

VRBike: Realidad virtual en ciclismo indoor

Grupo 2

AUTORES:

Santiago Agustín ARRIBÉRE,
Matías Santiago FRANCOIS,
Agustín Ignacio GALDEMAN,
Joaquín Oscar GAYTAN,
Pablo Martín SCHEINFELD

PROFESORES:

Ricardo Alejandro PINGITORE
Walter Hugo ORCHESSI
Ignacio GASPARINI
Alejandro Rubén UGARTE

CIUDAD AUTÓNOMA DE BUENOS AIRES

Julio 2022

Contenido

1. Agradecimientos	5
2. Acrónimos y Significados	6
2.1. Acrónimos	6
2.2. Definiciones	7
3. Resumen	10
4. Introducción	11
4.1. Antecedentes	11
4.2. Contexto del proyecto	12
4.3. Tipo de cambio	12
5. Objetivos del proyecto	13
5.1. Finalidad del proyecto	13
5.2. Planteamiento del problema por resolver	14
5.3. Alcance	14
6. Definición del producto	18
6.1. Requerimientos del cliente	18
6.1.1. Relevamiento de datos	19
6.1.2. Construcción de la casa de calidad	22
6.2. Diagrama funcional de interfaces	23
6.3. Especificaciones de diseño	24
6.3.1. Especificaciones Funcionales	25
6.3.2. Especificaciones de interfaz	25
6.3.3. Especificaciones de performance	27
6.3.4. Especificaciones de servicio (RAMS)	27
7. Plan de validación	28
7.1. Diseño de bancos de prueba	28
7.1.1. Banco de pruebas 1	28
7.1.2. Banco de pruebas 2	29
7.1.3. Banco de pruebas 3	31
7.1.4. Banco de pruebas 4	33
7.1.5. Banco de pruebas 5	34
7.1.6. Banco de pruebas 6	34
7.2. Especificaciones de tests	35
7.2.1. Banco de pruebas 1	37
7.2.2. Banco de pruebas 2	37
7.2.3. Banco de pruebas 3	38

7.2.4. Banco de pruebas 4	39
7.2.5. Banco de pruebas 5	40
7.2.6. Banco de pruebas 6	41
7.3. Plan de test	42
8. Factibilidad tecnológica	43
8.1. Alternativas de diseño	43
8.1.1. Sensor de velocidad	43
8.1.2. Interfaz de comunicación local	45
8.1.3. Dispositivo de visualización	47
8.1.4. Diseño de entornos virtuales	49
8.1.5. Networking	51
8.1.6. Métricas	57
8.1.7. Regulador de fuerza	60
8.1.8. Resumen	63
8.2. DFMEA	63
9. Factibilidad Económica	67
9.1. Dimensionamiento del mercado y modelo de negocio	67
9.2. Análisis de competencia	67
9.3. Estimación de costos	68
9.4. Flujo de fondos	70
10. Factibilidad de tiempos	71
10.1. Programación. Diagrama de Gantt	73
11. Factibilidad legal y responsabilidad civil	74
11.1. Tipo societario	74
11.2. Licencia de software	75
11.3. Garantía y soporte	75
11.4. Banco de datos	76
11.5. Modificación de las bicicletas	76
12. Ingeniería de Detalle	77
12.1. Hardware	77
12.1.1. Bicicleta fija	78
12.1.2. Microcontrolador	79
12.1.3. Regulación de fuerza	80
12.1.4. Sensor de velocidad	84
12.1.5. Vibración	85
12.1.6. Alimentación	87
12.2. Software	88
12.2.1. Networking	88

12.2.2. Aplicación de VR	90
12.2.3. Servidor	95
12.2.4. Aplicación del coach	99
12.2.5. Aplicación del usuario	106
12.2.6. Base de datos	123
13. Diseño de circuitos impresos	130
14. Diseño Mecánico	131
14.1. Diseño del sistema de control de fuerza	132
14.2. Diseño de soportes	133
14.2.1. Soporte del PCB	133
14.2.2. Soporte del sensor de velocidad	134
14.2.3. Soporte del motor de vibración	135
15. Estudio de Confiabilidad	137
15.1. Diagrama de confiabilidad	137
15.2. Confiabilidad de Hardware	137
15.2.1. Módulos bajo análisis	138
15.2.2. Valor final	140
15.3. Confiabilidad de Software	141
16. Conclusiones	146
17. Anexo	148
17.1. Anexo A: Casa de calidad	148
17.1.1. Costo de adquisición	148
17.1.2. Costo mensual	149
17.1.3. Comodidad	149
17.1.4. Facilidad de uso	150
17.1.5. Realismo de los escenarios	151
17.1.6. Baja emisión sonora	151
17.1.7. Multijugador	152
17.1.8. Comunicación con el entrenador	152
17.1.9. Variedad de los escenarios	153
17.1.10. Calidad de la inmersión	154

1. Agradecimientos

Queremos agradecer a todas las personas que nos acompañaron en el proceso de desarrollo del presente trabajo, siendo fundamentales en esta etapa. En primer lugar, se agradece la colaboración del profesor Marc Ressler por apoyar la idea desde el principio. Además, fue valioso el aporte de Diego Pedroarias, Facundo Esteves y Gustavo Romero de Megatlon para definir las características del producto. También se agradece a Jorge, Gabriel y Nelson del pañol del laboratorio de electrónica.

Por último, se agradece especialmente a nuestros familiares y amigos que siguieron de cerca el proyecto y nos apoyaron incondicionalmente en el camino.

2. Acrónimos y Significados

2.1. Acrónimos

Acrónimo	Descripción
A	Amperes
AC	Corriente Alterna (<i>Alternating Current</i>)
ADC	Conversor analógica-digital (<i>Analog-to-Digital Conversion</i>)
BLE	Bluetooth de baja energía (<i>Bluetooth Low Energy</i>)
BPM	Latidos por minuto (<i>Beats per minute</i>)
CPU	Unidad de procesamiento central (<i>Central Processing Unit</i>)
DAC	Conversor digital-analógica (<i>Digital-to-Analog Conversion</i>)
DFMEA	Análisis modal de fallos y efectos (<i>Design Failure Mode Effect Analysis</i>)
DUT	Dispositivo bajo prueba (<i>Device Under Test</i>)
FDM	Modelado por deposición de filamento (<i>Filament Deposition Modeling</i>)
FSM	Máquina de estados finitos (<i>Finite State Machine</i>)
FTP	Umbral de potencia funcional (<i>Functional Threshold Power</i>)
HD	Alta definición (<i>High Definition</i>)
I2C	Circuito inter-integrado (<i>Inter-Integrated Circuit</i>)
ID	Identificador único
IHRSA	Asociación Internacional de Salud, Raqueta y Clubes Deportivos (<i>International Health, Racquet & Sportsclub Association</i>)
IP	Protocolo de Internet (<i>Internet Protocol</i>)
LAN	Red de área local (<i>Local Area Network</i>)
NEMA	Asociación Nacional de Fabricantes Eléctricos (<i>National Electrical Manufacturers Association</i>)
PID	Controlador proporcional, integrativo y derivativo (<i>proportional–integral–derivative controller</i>)
PWM	Modulación por ancho de pulso (<i>Pulse Width Modulation</i>)
RFID	Identificador por radiofrecuencia (<i>Radio Frequency Identification</i>)
RPM	Revoluciones por minuto
RPN	Número de Prioridad de Riesgo (<i>Risk Priority Number</i>)
SDK	Kit de desarrollo de software (<i>Software Development Kit</i>)
SPI	Interfaz Periférica Serial (<i>Serial Peripheral Interface</i>)
TREMA	Tasa de Rendimiento Mínimo Aceptable
UART	Transmisor-Receptor Asíncrono Universal (<i>Universal Asynchronous Receiver-Transmitter</i>)
URL	Localizador de recursos uniforme (<i>Uniform Resource Locator</i>)
USB	Bus serie universal (<i>Universal Serial Bus</i>)
V	Volts
VR	Realidad Virtual (<i>Virtual Reality</i>)
W	Watts
WAN	Red de área amplia (<i>Wide Area Network</i>)

2.2. Definiciones

Access Point

Dispositivo autónomo que permite a los dispositivos inalámbricos conectarse y comunicarse con una red informática por cable.

Aplicación móvil

Programa informático diseñado para ejecutarse en un dispositivo móvil, como un teléfono o una tablet.

Backend

Partes de una aplicación informática o del código de un programa que permiten su funcionamiento y a las que no puede acceder un usuario.

Bug

Error en un programa informático que ocasiona un resultado distinto al deseado cuando se lo ejecuta.

Ciclismo Indoor

El ciclismo indoor o spinning, como se le conoce comúnmente, se define como una actividad física colectiva, realizada sobre una bicicleta especialmente diseñada para tal efecto, a un ritmo determinado, en la que se efectúa un trabajo predominantemente cardiovascular.

Coach

Entrenado a cargo de la dirección, la instrucción y el entrenamiento de una clase de ciclismo indoor.

Contenedor

Componente ejecutable estandarizado que combina el código fuente de la aplicación con las bibliotecas del sistema operativo (SO) y las dependencias necesarias para ejecutarlo en cualquier entorno.

Corriente de Foucault

La corriente de Foucault es un fenómeno eléctrico descubierto por el físico francés Léon Foucault en 1851. Se produce cuando un conductor atraviesa un campo magnético variable, o viceversa.

Data Logger

Dispositivo capaz de capturar datos.

Docker

Docker es una plataforma de código abierto que se utiliza para desarrollar, desplegar y gestionar aplicaciones en entornos virtuales ligeros llamados contenedores.

Debugging

Proviene de la palabra inglesa *bug*, acción de detectar y corregir los errores en un programa informático.

Driver

Tipo de programa informático que controla un dispositivo de hardware.

Ethernet

Tipo de protocolo de comunicación utilizada para conectar dispositivos en una red LAN o en una red WAN por cable.

Frontend

Partes de un programa informático o de un sitio web con las que el usuario interactúa.

Gaming

Ejecución de aplicaciones especializadas conocidas como juegos electrónicos o videojuegos en consolas de videojuegos como Xbox y Playstation o en computadoras personales.

Glitch

Fallo, error en el ámbito de los videojuegos.

Half HD

Half HD, 720p o 1K es el nombre corto para una de las categorías de los modos de vídeo, se habla en estos términos cuando la pantalla tiene una resolución de 1280x720 píxeles.

Host

Servidor de una partida multijugador.

Módem

Dispositivo de red que permite el acceso a Internet. En un extremo, se conecta al cable coaxial, telefónico o de fibra óptica que llega a la casa o al edificio desde el proveedor de servicios de Internet. En el otro extremo, se conecta al Router a través de Ethernet. Puede ser un dispositivo independiente o combinarse con un Router.

Networking

Vinculación de los ordenadores para que funcionen de forma interactiva.

Nube

Uso de recursos informáticos -como servidores, gestión de bases de datos, almacenamiento de datos, redes, aplicaciones de software y capacidades especiales- a través de Internet.

On Premise

Software o tecnología que se encuentra dentro de los límites físicos de una empresa.

Realidad Virtual

Conjunto de técnicas informáticas que permiten crear imágenes y espacios simulados en los que una persona, mediante un dispositivo visual, tiene la sensación estar y poder desenvolverse dentro de estos entornos.

Renderización

Representación gráfica.

Router

Dispositivo de red que dirige paquetes de datos entre redes informáticas

Ruido

Todas aquellas señales de interferencias no deseadas y que están unidas a la señal principal (útil) de manera que la pueden alterar, produciendo efectos que pueden ser más o menos perjudiciales.

Servidor

Computadora o programa informático que gestiona el acceso a un recurso o servicio centralizado en una red.

Stepper

Motor eléctrico cuya característica principal es que su eje gira realizando pasos, es decir, desplazándose una cantidad fija de grados.

Switch

Hardware de red que conecta los dispositivos de una red informática utilizando la conmutación de paquetes para recibir y reenviar los datos al dispositivo de destino.

Tethered VR headset

Visor de realidad virtual que precisa de una unidad de procesamiento externo para funcionar.

Workstation

Computadora encargada de recibir, enviar y visualizar los datos.

3. Resumen

El presente proyecto, llamado VRBike, corresponde al proyecto final de la carrera Ingeniería Electrónica. VRBike consiste en la integración de realidad virtual en las sesiones de ciclismo indoor. El proyecto responde a la necesidad de mejorar la interacción entre las maquinarias de los gimnasios y las personas que las utilizan, permitiendo con esto mejorar la experiencia del usuario e incentivar el uso de los equipos para el entrenamiento.

Para este proyecto se busca transformar las sesiones tradicionales en experiencias inmersivas en términos visuales, físicos y sociales. Para ello, se plantea la instalación de sensores, actuadores, elementos de procesamiento y elementos de visualización en las maquinarias de los gimnasios. En concreto, VRBike consiste en adaptar las bicicletas fijas de las sesiones de ciclismo indoor, pero los beneficios de la integración de realidad virtual permitirían que esta idea sea extensible hacia muchos otros campos, por ejemplo las sesiones de running o de remo en interiores.

Actualmente, algunos gimnasios incorporan proyectores delante de los participantes con rutas o distintos escenarios buscando aumentar la inmersión. Sin embargo, esto resulta en una experiencia incompleta, dado que lo que el participante realiza no se ve reflejado en la pantalla. VRBike busca mejorar la experiencia principalmente en este punto.

Para su realización se considera la implementación de una bicicleta fija que transmita al usuario los estímulos físicos de una rutina de ciclismo, tales como vibración del manubrio y la sensación de movimiento. Estos estímulos, generados mediante instrumentos añadidos a la bicicleta, estarán coordinados con lo que el usuario percibe en los diversos mapas que ofrece el software.

A su vez, se considera implementar la conexión entre diversos usuarios en un ecosistema de entrenamiento comunicado mediante un servidor, que permita interactuar de manera multijugador entre los participantes. Además, se busca que las estadísticas y los datos del entrenamiento que realicen los participantes estén disponibles al entrenador en tiempo real y ofrecer un mecanismo para que los usuarios puedan acceder a ellos tras la sesión mediante el uso de la aplicación de celulares que dejaremos disponible para su instalación por parte de los usuarios.

Por último, se detallan todas las características y los pasos evaluados a partir del análisis de la idea hasta concluir en el prototipo para validación. Dicho análisis se lleva a cabo desde distintas áreas, evaluando la factibilidad del mismo desde el punto de vista tecnológico, temporal, económico y legal.

4. Introducción

4.1. Antecedentes

Actualmente, existen empresas extranjeras dedicadas al desarrollo de sistemas que integran gráficos de escenarios con el fin de mejorar el entrenamiento deportivo. Entre estas empresas se pueden mencionar a Holodia y a Zwift, entre otras. Dichas empresas se encuentran mayoritariamente enfocadas en el entrenamiento hogareño, en algunos casos modificando bicicletas de interior.

Sin embargo, el mercado de ciclismo indoor en gimnasios se encuentra cubierto en mayor parte por bicicletas fijas sin inteligencia añadida, más allá de un control de revoluciones y sensores de ritmo cardíaco en los modelos de alta gama. Megatlon, una de las cadenas de gimnasios más grandes del país, solamente cuenta en 2 de más de 30 sedes con un sistema capaz de enviar las métricas de los participantes de las sesiones al entrenador en tiempo real. Esto es muy útil en las clases de spinning, pero sigue sin motivar a los participantes de la sesión más allá de los comentarios que pueda realizar el entrenador. A su vez, esta falta de inteligencia asociada a los equipos que se utilizan se repite en las demás cadenas de gimnasios.

Es importante notar que existe un gran interés por parte de los gimnasios en reconquistar clientes a través de bicicletas indoor. Dirigentes de dos de las cadenas de gimnasios más importantes de la Ciudad Autónoma de Buenos Aires manifestaron que este rubro de entrenamiento les trajo muchos beneficios en el pasado, pero el interés fue quedando atrás por falta de innovaciones.

Dentro del segmento de bicicletas indoor, algunas de las empresas que se encargan de la venta de estos equipos con enfoque en los gimnasios son Technogym, Lifefitness, Nordictrack y Matrix, esta última dedicada mayoritariamente al mercado oriental.

El costo de adquisición de estas bicicletas es muy superior con relación a las bicicletas convencionales. Esto es así a nivel global, pero, en concreto para Argentina, el hecho de que dichas empresas sean de origen extranjero hace que los costos de adquisición resulten incluso más elevados debido a los costos de envío, a las tasas de impuestos de importación y al tipo de cambio. Dentro de las bicicletas que estas empresas ofrecen existe un segmento que integra funcionalidades como pantallas que permiten el uso de Netflix y funcionalidades como audio para el entrenamiento, dichas bicicletas se encuentran en costos muy elevados para ser adquiridas de forma masiva en Argentina, por lo que los gimnasios del país consistentemente buscan opciones más económicas con capacidades similares.

A fin de suplantar las tecnologías de alto costo mencionadas, en los gimnasios se suele utilizar un proyector general para las personas que se encuentran entrenando, de forma tal de ver en él un camino en movimiento para hacer la experiencia más amena. Esto si bien es algo que mejora la experiencia, no alcanza a lograr una mejora consistente debido a que el nivel de inmersión de los participantes es bajo, principalmente porque sus acciones no se ven reflejadas en el video que ven en la pantalla.

Las sesiones actuales en los gimnasios considerados cuentan con alrededor de 15 personas y se

realizan entre 3 y 5 veces al día. Estos valores se encuentran limitados por la cantidad de clientes y una mejora en la experiencia de los usuarios podría aumentar dicha cantidad y fidelizar a aquellos con los que ya cuentan los gimnasios.

Ante este análisis, se evidencia una necesidad en los gimnasios de incorporar mayor inmersividad a un costo accesible. A su vez, existen cadenas de gimnasios en Argentina con sedes en todo el país e incluso algunas como SmartFit que se encuentran en múltiples países. De esta forma, una vez validados los beneficios de VRBike en la Ciudad Autónoma de Buenos Aires y alrededores (región elegida para el comienzo del proyecto debido a su cercanía), el mercado podría ampliarse a otras regiones. Por otra parte, comenzar con el foco en los gimnasios, donde el mercado de realidad virtual no se encuentra en gran medida explotado, podría hacer que a medida que el proyecto escale se pueda hacer posible competir en el mercado hogareño, en el cual ya existen actualmente varias opciones.

4.2. Contexto del proyecto

La idea del proyecto surge a partir de la materia 23.15 Realidad Virtual, en la que dialogando con el profesor Ing. Marc S. Ressler comentó que le parecía una opción muy interesante que posee gran potencial. Por esto, se tomó la decisión de implementarla como proyecto final de tesis de la carrera de Ingeniería Electrónica. Se dialogó con dirigentes de los gimnasios Megatlon y Fitter, quienes manifestaron su apoyo y buena predisposición en lo que se necesite, lo cual tiene alta consideración para los posibles problemas que surjan durante el proyecto que se desarrolla.

Por último, ya que se considera que el proyecto puede causar un impacto positivo en quienes lo utilicen, se cuenta con el objetivo de eventualmente lanzarlo al mercado.

4.3. Tipo de cambio

En el desarrollo del presente se expresan todas las cantidades monetarias en dólares estadounidenses (USD) para independizar el análisis económico realizado de la inestabilidad económica de nuestro país. Respecto del tipo de cambio, se usa dólar MEP (o *dólar bolsa*), dado que es el que se puede comprar sin restricciones.

5. Objetivos del proyecto

5.1. Finalidad del proyecto

De acuerdo a IHRSA (International Health, Racquet & Sportsclub Association) la industria de la salud y el fitness en EE.UU. ha crecido al menos entre un 3 % y un 4 % anual durante los últimos diez años y no muestra signos de desaceleración a corto plazo. En principio se podría atribuir este fenómeno al exponencial crecimiento de las redes sociales, pero también al aumento en el interés de mejorar su salud y calidad de vida. Estos hechos generan que las herramientas que permitan mejorar estas facetas de sus vidas cobren especial relevancia.

Además, debido a factores como la inseguridad, principalmente en las grandes ciudades, y factores climáticos, generan circunstancias idóneas para el entrenamiento indoor y la expansión de cadenas de gimnasios destinadas a suplir estas necesidades no resueltas.

Todos estos factores se incrementaron debido a la pandemia de COVID-19, que reclusó a las personas en sus hogares por un largo período de tiempo, por lo que muchas debieron optar por entrenamientos alternativos que no implicaran abandonar el hogar.

Sin embargo, el entrenamiento en interiores cuenta con la debilidad de su falta de dinamismo y de distracciones visuales, factores muy presentes en el exterior. Dicha escasez genera que el entrenamiento indoor pueda resultar repetitivo y que se pierda interés a lo largo del tiempo.

Por todo lo expuesto, se puede concluir que existe una demanda insatisfecha por parte de la población de herramientas que permitan mejorar el entrenamiento indoor, combinando los beneficios del exterior (dinamismo y atractivos visuales) con los beneficios del interior (confort y seguridad). La combinación de estos beneficios podría implementarse mediante un aumento en el grado de inmersión en el entrenamiento indoor, generando mayor motivación en los participantes al momento de ejercitarse.

Nuestro producto busca brindar una solución más motivadora que las opciones actuales, implementando una bicicleta que, integrando realidad virtual, sense los movimientos y las acciones del usuario para comunicarlas con un entorno virtual en el que el usuario puede recorrer el escenario o tomar sus tiempos en una pista de carreras con la finalidad de batir su marca personal, compararse con los demás participantes de la sesión, entre otras opciones.

Todas estas características permiten que el usuario se encuentre motivado mientras entrena en la bicicleta, sintiendo que forma parte del entorno gracias a herramientas de comunicación del software con la persona, como ser el control activo de fuerza según la pendiente del terreno virtual o las vibraciones en el manubrio según el grado de rugosidad del terreno.

Esto logrará que la experiencia de hacer gimnasia se torne más motivadora, entretenida y amena para los ciclistas, logrando con esto una mejora en su salud y calidad de vidas en las personas mediante el ejercicio físico.

5.2. Planteamiento del problema por resolver

La principal problemática que se intenta resolver es la falta de motivación en las personas a la hora de utilizar los equipos de ciclismo indoor para realizar ejercicio en interiores. Esta problemática está presente tanto en gimnasios como en el entrenamiento hogareño.

En el caso del ciclismo indoor en los gimnasios, no existen actualmente opciones para solucionar la problemática. Como fue mencionado, esto es algo que los gimnasios intentan solucionar, ya que conocen que las personas entrenan más y mejor cuando se sienten motivadas y felices de realizar la rutina que le corresponde, algo que se puede incentivar mediante el entrenamiento recreativo o competitivo permitido a través de productos tecnológicos como el del presente proyecto.

A su vez, como se mencionó en la sección de antecedentes, las opciones que actualmente apuntan a los gimnasios son muy costosas. Por no lograr una buena relación costo-beneficio, dichas opciones son poco utilizadas y solo se evidencian en pocos gimnasios en Argentina (dos sedes de Megatlon).

Como consecuencia, se plantea como problemática a resolver aumentar la motivación de los participantes de sesiones de ciclismo indoor a un costo accesible. Para hacerlo, el presente proyecto busca, a partir de la modificación de bicicletas fijas, posibilitar una experiencia inmersiva en un ambiente virtual que logre causar impacto en el usuario a través de diferentes mecanismos de interacción.

Todo esto debe a su vez ser logrado respetando los márgenes de precios que se consideran convenientes para lograr un flujo de ventas acorde con los valores esperados en función de los costes de productos similares brindados por otras empresas.

5.3. Alcance

Para resolver la problemática comentada anteriormente, este proyecto propone la adaptación de una bicicleta fija, implementando diferentes sistemas para la comunicación desde el usuario al software como desde el software al usuario (pasando ambas comunicaciones por medios físicos en la bicicleta). En la figura 5.1 se presentan los elementos de hardware a incorporar o adaptar en las bicicletas sobre las que se trabaja.



FIGURA 5.1: Elementos de hardware a incorporar en cada bicicleta.

En primer lugar, el dispositivo de visualización es aquel en el que cada participante de la sesión de spinning visualiza a los distintos escenarios provistos por el software. Dicho dispositivo podría consistir de un visor de realidad virtual o de una pantalla delante del participante, en particular para nuestro prototipo se utilizará un visor de realidad virtual, ya que se considera que puede generar un mayor impacto en los usuarios.

Mediante el dispositivo de visualización, el usuario debe percibir visualmente ciertas acciones realizadas físicamente. Entre estas se encuentran la velocidad de pedaleo del ciclista. Para esto, es necesario incorporar un sensor de velocidad capaz de medir la velocidad del volante de inercia de la bicicleta.

Además de los sensores mencionados, VRBike propone que distintas componentes del terreno del mundo virtual en el que se ejercite el participante influyan en lo que percibe físicamente. Entre estas componentes se encuentran la inclinación, el tipo de terreno y las irregularidades del mismo. Las irregularidades donde la bicicleta se mueve influyen en la vibración percibida por el ciclista, por lo cual se busca replicar dicho comportamiento incorporando generadores de vibración en el manubrio de la bicicleta. De esta manera, por ejemplo, si la persona está andando en un terreno de ripio, el software envía esa información al control de vibraciones de la bicicleta para que se active y genere en el manubrio las sensaciones de vibración características de estos terrenos. Por otra parte, la inclinación y el tipo de terreno hacen que se requiera realizar una mayor fuerza para avanzar, por ejemplo en el caso de un terreno barroso con un ángulo de ascenso implicaría la necesidad de pedalear con mayor fuerza. Para esto, se debe incorporar un regulador de fuerza en la bicicleta, el cual se basa en la intervención del sistema de frenado que se encuentra en la bicicleta.

Todos estos elementos deben ser sincronizados y procesados en tiempo real. Para realizarlo, es necesario un dispositivo de comunicación y uno de cómputo, aunque estos se podrían integrar en

un único dispositivo.

Por último, para alimentar a todos los elementos de hardware mencionados se debe incorporar en cada bicicleta una fuente de alimentación capaz de brindar al sistema de las tensiones y corrientes necesarias.

VRBike propone que los usuarios puedan participar en sesiones multijugador. En ellas, podrán observar al resto de ciclistas en el ambiente virtual y podrán competir entre ellos. Con respecto al alcance de las sesiones, se plantea que VRBike soporte sesiones multijugador de hasta 15 participantes. Además, las métricas de los participantes deben ser visualizables por el entrenador a cargo de la sesión, así como también por los usuarios, luego de finalizada. Para realizar este punto, se resumen en la figura 5.2 los elementos necesarios a tener en cuenta de software y de networking.

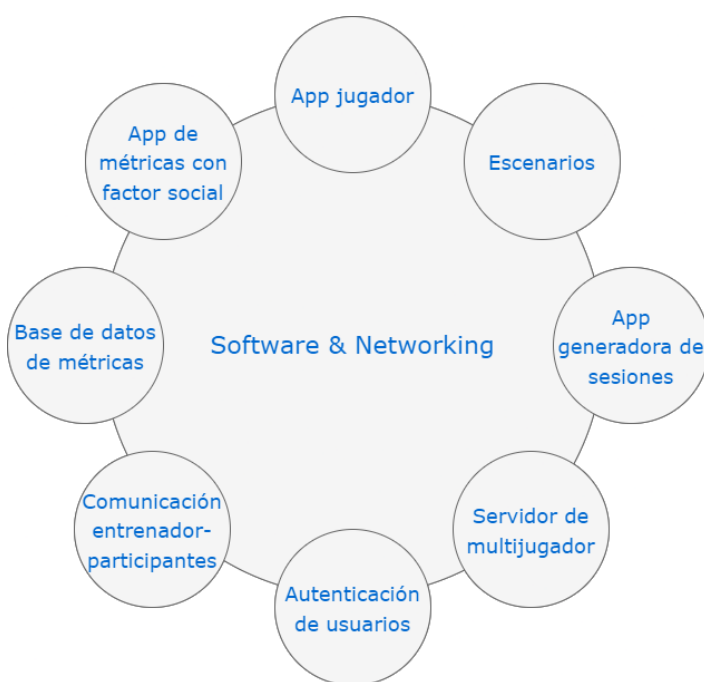


FIGURA 5.2: Elementos de software y de networking.

Como punto principal, cada participante debe visualizar una aplicación en su dispositivo de visualización. En ella se muestra el escenario sobre el cual el usuario realiza el ejercicio y las distintas métricas de interés. VRBike propone la opción de seleccionar entre distintos escenarios, entre los que se consideran ambientes de rutas, bosques y ciudades.

Además de la aplicación en cada dispositivo, el entrenador a cargo de la sesión debe contar con una aplicación capaz de visualizar a los dispositivos que se conectan, seleccionar un escenario, comenzar una sesión y obtener las métricas de los distintos participantes.

A modo de sincronizar a todos los usuarios y poder comunicar sus métricas actuales, es necesaria la presencia de un servidor que gestione la sesión de multijugador. Por otra parte, cada participante, si lo desea, debe ser identificable en la sesión y para ello es necesario un sistema de vinculación entre la bicicleta y el perfil del usuario. Es decir, un sistema de autenticación. Si el usuario no

desea autenticarse, podrá participar de la sesión con un alias "Invitado" o similar.

A su vez, en las sesiones tradicionales, el entrenador le indica a los participantes bajo que intensidad realizar el ejercicio, dependiendo de cada momento y de la forma física y del estado en el que esté ejercitándose cada individuo. Por ello, es necesario tener en cuenta esto en VRBike. El entrenador debe ser capaz de comunicarse con los participantes y dicha comunicación debe transitar por medio del servidor. A su vez, la generación de los mensajes a enviar debe estar integrada en la app del entrenador.

Tras la sesión, VRBike incluye la opción de verificar las métricas obtenidas una vez finalizada. A fin de almacenar las métricas de cada participante, se necesita contar con una base de datos externa. El paso ya mencionado de autenticación permite vincular a cada participante con los registros obtenidos. Y, si el participante decidió no autenticarse, podrá realizar el ejercicio pero sin almacenar sus métricas.

Una vez almacenadas, cada usuario podrá acceder a las métricas mediante una aplicación móvil. En esta aplicación podrá verificar sus propias métricas y también contrastarlas con las de otros usuarios. De esta forma, se busca que el factor social aumente la motivación de los participantes en superar los valores obtenidos por el resto en una forma de ranking. Como consecuencia, cada participante buscaría asistir a más sesiones, aumentando el interés en VRBike.

Como fue mencionado anteriormente, la idea es trabajar en primera instancia con gimnasios, debido a que no se evidencian soluciones similares ante las problemáticas planteadas. Además, sabemos que existen cadenas interesadas en productos innovadores en este ámbito, por lo que es algo que podría llegar a interesar como producto de comercialización. Sin embargo, una vez que el producto escale y se perfeccione, no se descarta adaptarlo a una versión para hogares.

En cuanto al alcance geográfico, como también se mencionó, se contempla en principio los gimnasios presentes en la Ciudad Autónoma de Buenos Aires y alrededores debido a su cercanía y, una vez incorporado el producto, escalar hacia el interior y hacia el exterior del país.

6. Definición del producto

6.1. Requerimientos del cliente

Luego de varias videollamadas con los responsables de dos de los gimnasios más importantes de la Ciudad de Buenos Aires, pudimos tomar ciertos datos que son de interés para ellos respecto de nuestro producto, como son la necesidad de transmitir sensaciones realistas a los usuarios, la importancia en la calidad de los materiales, la necesidad de escenarios realistas y la importancia de las comunicaciones con el entrenador. Mediante estos puntos se desarrollaron los requerimientos agregando un ID a cada uno para su posterior trazabilidad, así como también su origen, el cual puede ser desde el cliente, desprenderse de la ley Argentina, o bien ser tácito.

ID	Descripción	Origen
REQ-01	Los escenarios del entorno virtual deben estar basados en ambientes realistas (no fantásticos).	Cliente
REQ-02	El usuario debe percibir sensaciones realistas en base a la inclinación y tipo del terreno del entorno virtual.	Cliente
REQ-03	El usuario debe poder realizar cambios de velocidad y percibirlos en el entorno virtual.	Cliente
REQ-04	Los entornos virtuales deben tener buena calidad gráfica.	Cliente
REQ-05	Se debe mantener la estética y calidad de las bicicletas convencionales.	Cliente
REQ-06	El sistema debe permitir sesiones de entrenamiento de hasta 15 participantes.	Cliente
REQ-07	Cada usuario debe poder observar al resto de los participantes en el entorno virtual.	Cliente
REQ-08	El entrenador debe poder recibir las métricas de todos los participantes para monitorear el desempeño.	Cliente
REQ-09	El entrenador debe poder interactuar con los participantes una vez inmersos en el entorno virtual.	Cliente
REQ-10	El usuario debe poder acceder a sus estadísticas por medio de una app móvil.	Cliente
REQ-11	El usuario debe poder contar con un perfil y poder autenticarse al utilizar las bicicletas.	Cliente
REQ-12	El usuario debe poder utilizar el sistema de VRBike sin autenticarse.	Tácito
REQ-13	El usuario debe poder comparar sus métricas con las de otros usuarios mediante la app móvil.	Cliente
REQ-14	El entrenador debe poder elegir el escenario en función de la actividad que se desee realizar.	Cliente
REQ-15	El producto debe funcionar al ser alimentado con la red eléctrica argentina.	Tácito
REQ-16	El sistema debe ser fácil de utilizar para el usuario.	Tácito
REQ-17	El producto debe operar de forma correcta en condiciones normales de temperatura y humedad en salas de gimnasios en el territorio argentino.	Tácito
REQ-18	Se debe poder brindar las métricas típicas provistas en las bicicletas fijas.	Tácito
REQ-19	El producto debe cumplir con la resolución N° 1038/2021.	Ley Argentina

TABLA 6.1: Requerimientos del cliente

6.1.1. Relevamiento de datos

El proyecto propuesto busca, por medio de un aumento en el grado de inmersividad, aumentar la motivación de los participantes de las sesiones de ciclismo indoor. Existen distintos estudios científicos que respaldan dicha relación entre inmersividad y motivación. Por ejemplo, según un

estudio de la Universidad Técnica de Eindhoven (Eindhoven University of Technology), "[...] ofrecer un entorno más inmersivo en el que el usuario se sienta presente aumenta la diversión del usuario y, por tanto, tiene un efecto beneficioso en la motivación del usuario. En el entorno altamente inmersivo, en el que la experiencia de la presencia era más fuerte, los participantes declararon tener más interés y disfrute, más competencia y control percibidos y -quizá lo más importante- pedalearon más rápido."¹.

El aumento en el grado de inmersividad puede ser logrado por distintos medios. Actualmente, en el mercado de ciclismo indoor, como ya se ha mencionado, esto se realiza mediante la colocación de proyectores en donde los ciclistas puedan visualizar paisajes de rutas o bosques. Sin embargo, esta opción no permite interactuar con lo que sucede en los escenarios. Así, la inclusión de realidad virtual permitiría maximizar el nivel de inmersión y lograr los resultados buscados.

Existen datos prometedores con respecto al crecimiento del campo de realidad virtual y a su inserción en la vida habitual de la población. El tamaño del mercado mundial de la realidad virtual se valoró en 15.810 millones de dólares en 2020 y se espera que crezca a una tasa de crecimiento anual compuesta (CAGR) del 18 % de 2021 a 2028². Dicho crecimiento puede ser beneficioso para la introducción de VRBike en el mercado, generando que progresivamente las personas se encuentren más familiarizadas con esta tecnología.

Ante el planteo de introducir la realidad virtual en el deporte, específicamente en el spinning, surgen ciertas dudas con respecto a su comodidad y si esto podría afectar los resultados buscados. Un estudio realizado por la Universidad Estatal de Arizona (Arizona State University)³ muestra los efectos de realizar el entrenamiento con y sin visores de realidad virtual. Entre los resultados obtenidos se destaca en primer lugar que "La implementación de la realidad virtual para "gamificar" las máquinas de fitness ayuda a mejorar el entrenamiento en general. De los usuarios analizados, el 75 % se encontró interesado en volver a utilizar las Oculus Rift para entrenar porque lo disfrutaron y les pareció divertido.". Además, en dicho estudio se analiza la comodidad de utilizar el visor durante el ejercicio. Tomando una escala de 1 a 5, donde 1 es lo menos confortable, ningún participante seleccionó los valores de 1 y 2. La mayor parte de ellos lo hicieron en torno a 3 y 4, mientras que otro grupo de participantes le asignó un 5, considerando que no tuvieron ningún problema de comodidad.

Además del factor de la comodidad, un punto a tener en cuenta para el desarrollo de los escenarios es que en el estudio referenciado se indica que el 87,5 % de los participantes estuvieron de acuerdo en que si el juego tuviera más escenarios, lo utilizarían más.

Los beneficios obtenidos con realidad virtual, respaldados por distintos estudios, generan que actualmente existan en el mercado algunas opciones de características similares a VRBike. Entre las más destacadas se encuentra Wahoo Fitness, que ofrece distintos productos para transformar

¹Ijsselstein, Wijnand & De Kort, Yvonne & Bonants, R & Westerink, Joyce & de Jager, Marko. (2004). Virtual Cycling: Effects of immersion and a virtual coach on motivation and presence in a home fitness application.

²Grand View Research

³Amresh, Ashish & Salla, Rahul. (2017). Towards a Home-based Virtual Reality Game System to Promote Exercise. 10.24251/HICSS.2017.091.

una bicicleta convencional (de exterior) en una bicicleta inteligente. A partir de esto, la bicicleta puede conectarse con aplicaciones de software como Zwift o Strava, que proveen los escenarios necesarios. Esta empresa logró también establecerse como uno de los líderes en el sector hogareño, actualmente contando con oficinas en 4 continentes.

Como ya se comentó, se realizaron diferentes reuniones virtuales con los ejecutivos de 2 gimnasios muy importantes de la Ciudad Autónoma de Buenos Aires en las que se explicó la idea y se obtuvo en el proceso un muy valioso feedback sobre lo que les interesaría que tenga el producto, manifestando a su vez su fuerte apoyo a la idea y su disposición para lo que se necesite en el proceso.

6.1.2. Construcción de la casa de calidad

A continuación se muestra el gráfico correspondiente a la casa de calidad que fue realizada para el proyecto.

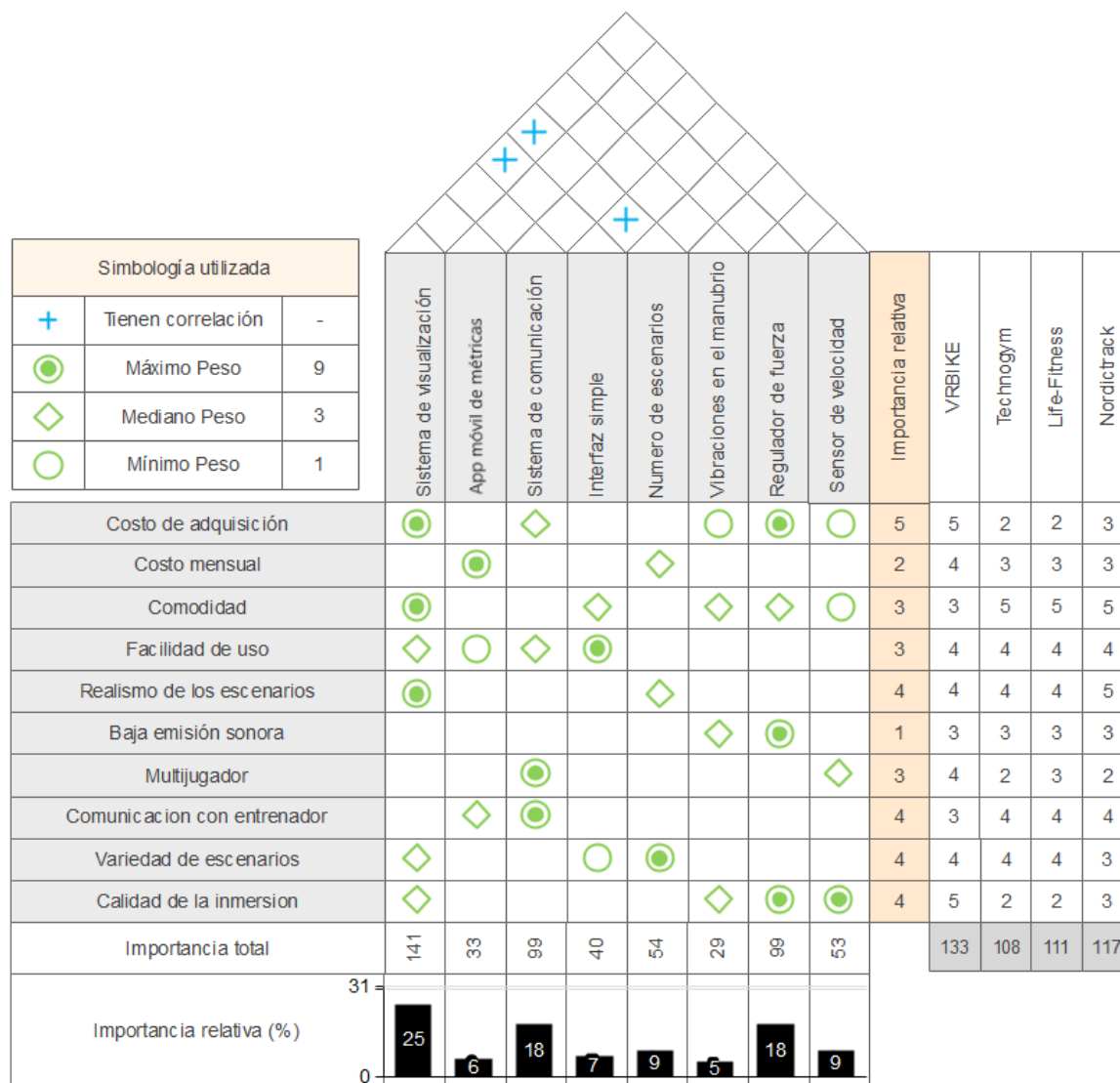


FIGURA 6.1: Casa de calidad.

Los valores incluidos en la casa de calidad se encuentran explicados en la sección 'Casa de Calidad' del Anexo del presente informe (ver 17.1).

6.2. Diagrama funcional de interfaces

A continuación se muestra un diagrama con las principales interfaces del proyecto, ubicando en el centro en el cubo amarillo al producto en desarrollo. Cada interfaz presentará luego las especificaciones de interfaz correspondientes según las tareas que lleven a cabo.

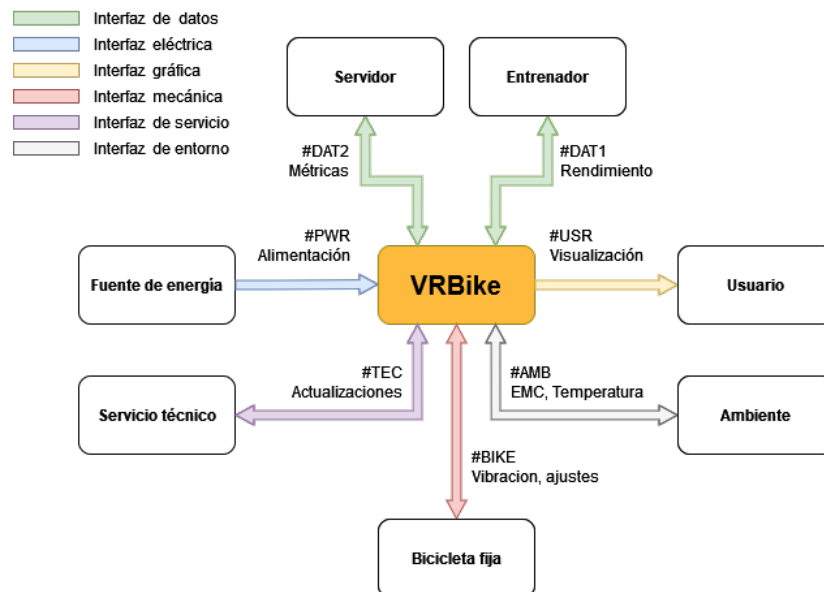


FIGURA 6.2: Interfaces que forman parte del proyecto.

Aunque luego se desarrollará en más detalle, cabe aclarar que el entrenador cuenta con una interfaz específica, dado que no necesita estar inmerso en el entorno virtual para las funciones que realiza. Una de sus funciones principales es controlar el estado de cada individuo y verificar que permanezcan en buen estado de salud mientras realizan el ejercicio. Es decir, verificar que no se produzcan mareos, náuseas o descompensaciones. Así, si el entrenador se encontrara inmerso mediante un visor de realidad virtual, perdería su capacidad para controlar lo mencionado. Además, el entrenador debe poder visualizar las métricas detalladas de todos los participantes sin necesidad de estar andando en una bicicleta. Por lo tanto, se considera que a través de una interfaz especial (una aplicación con una limpia interfaz de usuario) podrá realizar sus tareas adecuadamente.

Asimismo, en la figura 6.3 se aprecia un diagrama que representa la estructura general del proyecto, mostrando la interconexión entre distintos dispositivos y la estructura interna de cada uno. Como así también la conexión con el dispositivo maestro y el servidor.

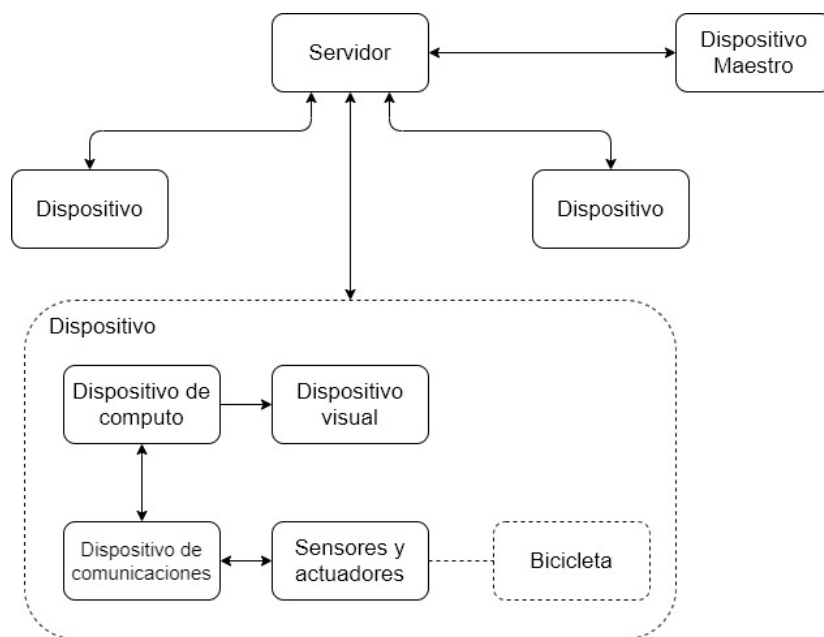


FIGURA 6.3: Estructura general del proyecto.

6.3. Especificaciones de diseño

A continuación se describen las especificaciones de diseño desarrolladas a partir de los requerimientos presentados previamente. En cada especificación se incorpora un ID único para su posterior trazabilidad y a su vez se indican los requerimientos que dan origen a la misma, como así también se menciona a qué instancia corresponde su aplicación y la forma en que se validará.

Leyenda para Especificaciones	
Aplicabilidad	Validación
P: Prototipo	I: Inspección visual
F: Producto final	D: Documentación
	S: Simulación
	T: Test

TABLA 6.2: Leyenda de uso en especificaciones

6.3.1. Especificaciones Funcionales

ID	Descripción	Origen	Aplicabilidad Validación
FUN-OP-01	El sistema debe ajustar de forma automática la carga en función de la inclinación y el terreno del entorno virtual.	REQ-02	P, F / T
FUN-OP-02	El sistema debe variar (o desactivar) la vibración del manubrio según el terreno por el que se circule.	REQ-02	P, F / T
FUN-OP-03	Se debe medir la velocidad del volante de inercia de la bicicleta.	REQ-18	P, F / T
FUN-OP-06	Se deben transmitir las mediciones al entrenador.	REQ-08	F / S
FUN-OP-07	Las mediciones deben quedar almacenadas para su acceso de forma remota.	REQ-10 REQ-11 REQ-12	F / S
FUN-OP-08	Se debe mostrar al usuario la distancia recorrida y la velocidad.	REQ-16 REQ-18	P,F / I, S
FUN-OP-08	El producto debe incluir un sistema de visualización de escenarios.	REQ-04	P, F / I

TABLA 6.3: Especificaciones funcionales de operación

6.3.2. Especificaciones de interfaz

ID	Descripción	Origen	Aplicabilidad Validación
I-BIKE-01	Los sistemas electrónicos incorporados a la bicicleta deben soportar sus vibraciones típicas.	REQ-05	P, F / I, D
I-BIKE-02	Los sistemas agregados no deben generar cambios grandes o grotescos en las dimensiones de la bicicleta fija.	REQ-05	F / I
I-BIKE-03	La velocidad de pedaleo debe ser medible en el intervalo de 0 a 400 RPM, con una precisión del 5 % (valor típico).	REQ-08 REQ-10 REQ-13 REQ-18	P, F / D, T
I-BIKE-04	Debe ser posible regular la carga con una precisión del 10 % basada en los niveles de giro de la varilla roscada para las bicicletas con freno a rozamiento o niveles de corriente para las que incorporan freno magnético.	REQ-02 REQ-04	P, F / D, T

TABLA 6.4: Especificaciones de interfaz mecánica

ID	Descripción	Origen	Aplicabilidad Validación
I-AMB-01	El dispositivo deberá operar normalmente con temperatura ambiente entre 15°C y 45°C	REQ-17	P, F / D

TABLA 6.8: Especificaciones de ambiente

ID	Descripción	Origen	Aplicabilidad Validación
I-DAT-01	La comunicación entre diversos dispositivos (bicicletas distintas dentro de una sala) se realizará de forma inalámbrica	REQ-06 REQ-07 REQ-08 REQ-09	P, F / D, T
I-DAT-02	Las mediciones al entrenador se deben transmitir en intervalos de como máximo dos segundos	REQ-08	P, F / I, D, T
I-DAT-03	Las métricas se deben almacenar en una base de datos en intervalos de un minuto.	REQ-10 REQ-11 REQ-13	P, F / I, D, T

TABLA 6.5: Especificaciones de interfaz de datos

ID	Descripción	Origen	Aplicabilidad Validación
I-GRAF-01	El dispositivo de visualización debe poder renderizar y reproducir video en calidad HD (720p) a 30FPS sostenidos.	REQ-04	P, F / I, D, T
I-GRAF-02	Se debe permitir la navegación por los menús si los hubiera (p. ej. de configuración de exigencia, peso, etc.) de forma intuitiva y eficiente para el usuario.	REQ-16	P, F / I, D

TABLA 6.6: Especificaciones de interfaz gráfica

ID	Descripción	Origen	Aplicabilidad Validación
I-POW-01	El producto funciona alimentado con una fuente de tensión 220V AC input - 12V/3A output	REQ-15	P, F / I, D
I-POW-02	La fuente de alimentación debe cumplir con normas IRAM 2292	REQ-19	P, F / D

TABLA 6.7: Especificaciones de interfaz eléctrica

6.3.3. Especificaciones de performance

ID	Descripción	Origen	Aplicabilidad Validación
PER-01	Se debe poder mantener la conexión con 15 jugadores (tamaño típico de una sala de spinning) en simultáneo, sin implicaciones en las funcionalidades o en el apartado gráfico	REQ-06 REQ-07 REQ-08 REQ-09	F / S, D, T
PER-02	El casco de realidad virtual utilizado debe contar con al menos dos horas de autonomía en uso continuo	REQ-04 REQ-16	P, F / D
PER-03	El retardo entre las acciones del usuario y sus efectos en el entorno virtual debe ser menor a 300ms.	REQ-03 REQ-04	P / I, S
PER-04	La latencia de conexión con el servidor debe ser menor a 70ms	REQ-03 REQ-04	P, F / I, T

TABLA 6.9: Especificaciones de performance

6.3.4. Especificaciones de servicio (RAMS)

ID	Descripción	Origen	Aplicabilidad Validación
SER-01	Se debe contar con una aplicación que sea utilizada por el entrenador para seleccionar el escenario y comenzar la sesión.	REQ-14	P, F / I, D
SER-02	En la aplicación del entrenador se debe recibir y visualizar las métricas de los integrantes de una sala	REQ-08 REQ-09	P, F / I, D
SER-03	Las métricas deben ser accesibles por medio de una app móvil al finalizar una sesión de entrenamiento	REQ-10 REQ-11 REQ-13	P, F / I, D

TABLA 6.10: Especificaciones de servicio

7. Plan de validación

En este apartado se busca definir el plan de validación del presente diseño. Esta tarea se realiza de forma *bottom-up*, por lo que en primera instancia es necesario dividir al diseño en capas distribuidas jerárquicamente, en función de su nivel de abstracción respecto del mundo físico.

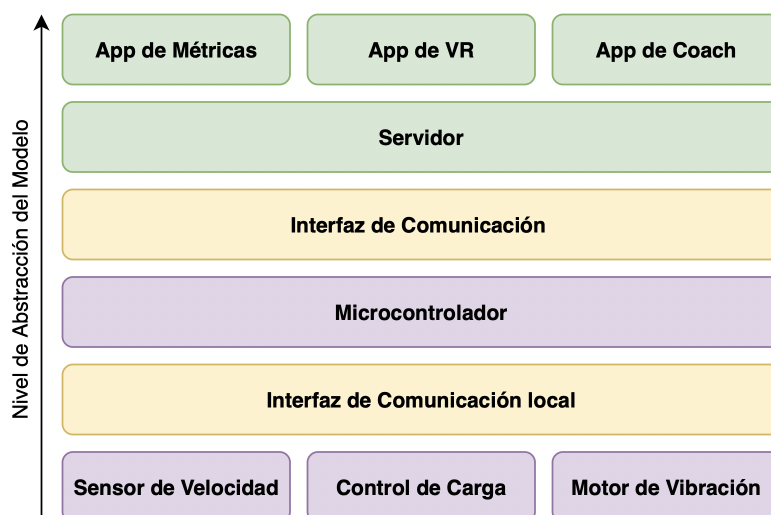


FIGURA 7.1: Segmentación del diseño de validación.

En la figura 7.1 se aprecia la segmentación antes mencionada. En su capa inferior se destacan los tres módulos que se encargan de interactuar con el medio físico (sensor de velocidad, control de carga y actuador de vibración). Estos módulos deben ser validados en forma separada.

Asimismo, deben validarse las interfaces de comunicación local entre el hardware y el microcontrolador y los canales entre el microcontrolador, el visor y el servidor local. Por último, es necesario validar que las distintas aplicaciones cumplan con los requerimientos planteados.

7.1. Diseño de bancos de prueba

A continuación se describen aquellos bancos de prueba a emplear en el proceso de validación del diseño.

7.1.1. Banco de pruebas 1

El banco de pruebas 1 corresponde a las validaciones necesarias para la regulación de la fuerza en la bicicleta mediante el motor stepper y la medición de la velocidad mediante el sensor de velocidad. Los rangos y desviaciones máximas permitidas se detallan en la tabla 7.1, en función de las especificaciones mencionadas en la tabla 6.4.

Variable	Rango a validar	Desviación máxima admitida
Velocidad	0 RPM - 400 RPM	5 %
Potencia	5 niveles con graduaciones internas	15 %

TABLA 7.1: Banco de pruebas 1. Rangos y tolerancias.

Para realizar la validación de cada uno de estos parámetros, se emplea un modelo de bicicleta fija que sea representativo del parque objetivo, de forma tal de luego poder extrapolar estos resultados a bicicletas con ligeras variaciones mecánicas.

Para la validación de la velocidad se utiliza un sensor de velocidad Magene s3+ que se anexa al pedal de la bicicleta para la medición de la velocidad. De esta manera, si las velocidades marcadas entre el sensor del sistema y el sensor de validación difieren en la cota de error propuesta (5 %), entonces el resultado de la validación es positivo.

Por otro lado, para la validación de la fuerza que el usuario tiene que realizar en los diferentes niveles de fuerza según el ángulo de giro del sistema que controla la fuerza, lo que se decide realizar es la prueba en diferentes grados de giro de la varilla selectora de fuerza en los casos con freno mecánico basándose en zapatas o para diferentes corrientes en el caso de los frenos magnéticos. Para validar esto lo que se hace es utilizar una llave torquimétrica que se puede conectar a los pedales de la bicicleta. De esta manera, la llave torquimétrica con resorte puede medir la fuerza que se ha de realizar para mover el pedal de la bicicleta en las diferentes posiciones de giro de la varilla o diferentes corrientes en el inductor del freno magnético según el modelo de bicicleta. Así lo que se puede comprobar es si se cumple con los requerimientos de fuerza impuestos, de no ser el caso se han de regular los ángulos de giro en la bicicleta para que esto se cumpla.

Señal	Descripción
load_data	Nivel de carga impuesto por el software en la bicicleta.
bike_speed_data	Velocidad angular de pedaleo medida por el sensor montado en la bicicleta.

TABLA 7.2: Señales presentes en el banco de pruebas 1.

7.1.2. Banco de pruebas 2

El banco de pruebas número 2 tiene como objetivo validar el funcionamiento del sistema de vibración. Para lograr este objetivo, se debe validar que la intensidad de la vibración supere el umbral impuesto en las especificaciones y además debe ser posible medir su direccionalidad. Lo anterior puede lograrse incorporando un arreglo de sensores piezoeléctricos direccionales, que entregan una señal analógica proporcional a la intensidad de la vibración que reciben. La orientación de los sensores determina la dirección de la vibración medida.

En las figuras 7.2 y 7.3 se observa el banco de pruebas propuesto.

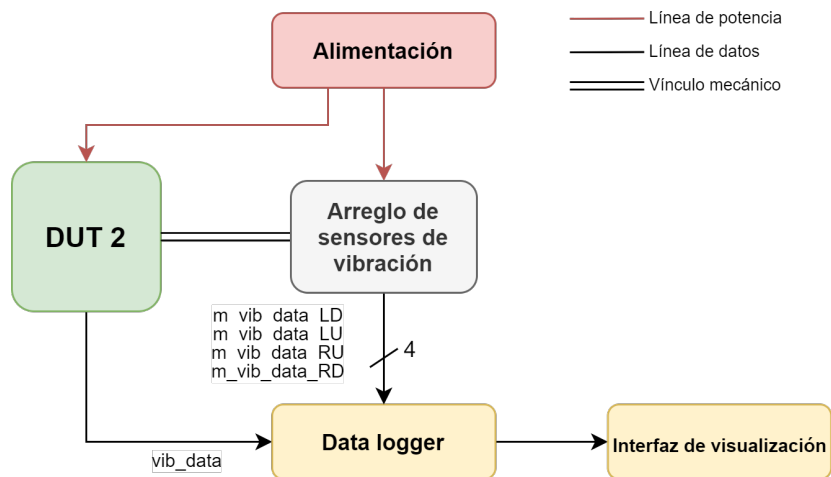


FIGURA 7.2: Banco de pruebas para vibración - Diagrama en bloques.

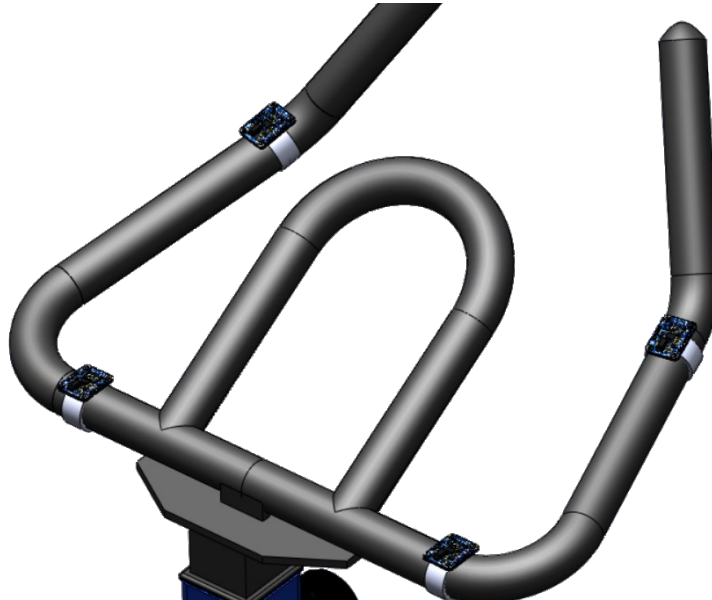


FIGURA 7.3: Banco de pruebas para vibración.

Este banco de pruebas contiene:

1. **Alimentación.** Se suministra alimentación de red (220V - 50Hz).
2. **Data logger.** Interfaz capaz de comunicarse con distintos módulos. Su función principal es la de recolectar datos para su análisis (ver señales en tabla 7.3).
3. **Arreglo de sensores de vibración.** La distribución del arreglo se observa en la figura 7.3. En este bloque se incluye la fuente para poder alimentar al arreglo de sensores. Se emplean cuatro sensores, colocando dos de cada lado del manubrio. El arreglo de sensores de vibración busca detectar las vibraciones generadas por los actuadores instalados en el manubrio.
4. **Interfaz de visualización.** Los datos de la prueba se observan desde una PC. En este dispositivo se realiza el preprocesamiento correspondiente de cada una de las señales y se

computan las métricas necesarias para poder evaluar la performance del diseño en cada una de las pruebas.

5. **DUT 2.** Este dispositivo se compone de la bicicleta con sus respectivos motores de vibración colocados en el manubrio y su dispositivo de cómputo corriendo la aplicación. Además, posee un modo especial de software para poder medir la vibración.

En la tabla 7.3 se muestran las distintas señales presentes en este banco de pruebas.

Señal	Descripción
vib_data	Nivel de vibración impuesto por el software en la bicicleta.
m_vib_data_LU	Salida del sensor de vibración izquierdo de la parte superior.
m_vib_data_RU	Salida del sensor de vibración derecho de la parte superior.
m_vib_data_LD	Salida del sensor de vibración izquierdo de la parte inferior.
m_vib_data_RD	Salida del sensor de vibración derecho de la parte inferior.

TABLA 7.3: Señales presentes en el banco de pruebas 2.

De esta manera, lo que se realiza es la prueba con el arreglo de sensores dispuesto en la bicicleta. Si los sensores detectan que la vibración se encuentra dentro del umbral necesario, la validación es aceptada, en caso contrario se modifica el PWM de los motores de vibración con la finalidad de mejorar las mediciones de los sensores de vibración.

7.1.3. Banco de pruebas 3

El banco de pruebas desarrollado para el testeo de la aplicación de realidad virtual puede verse en la figura 7.4. En este banco de pruebas se testeará la performance de este software y los niveles bajo diferentes tipos de condiciones.

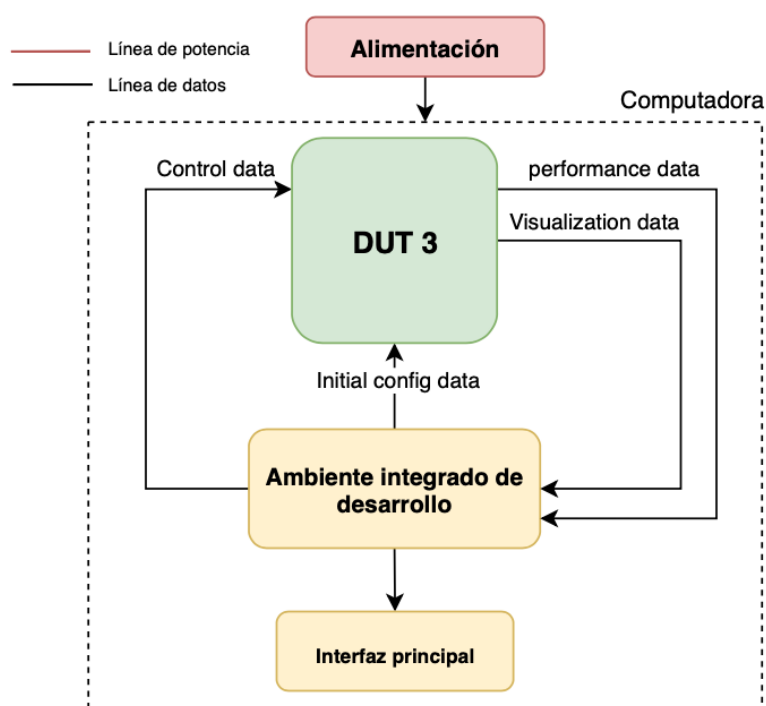


FIGURA 7.4: Banco de pruebas para validación de software - Diagrama en bloques.

A continuación se detallan sus componentes:

1. **Alimentación.** Se suministra alimentación de red (220V - 50Hz).
2. **Ambiente integrado de desarrollo.** Es el elemento principal para el control de las señales de testeo (se hace un *mock* de las señales de entrada de la aplicación). Este ambiente se comunica con el DUT, obteniendo la capacidad de *debugging* mediante la herramienta correspondiente (*Oculus Debug Tool* si la aplicación corriera sobre un visor VR Oculus).
3. **Interfaz principal.** En la interfaz principal se visualiza el funcionamiento de la aplicación y se reflejan los datos recolectados con la performance de esta.
4. **DUT 3.** Software a testear (Aplicación de VR).

En la siguiente tabla se explican cada una de las señales que componen el banco de pruebas.

Señal	Descripción
Control data	Información necesaria para usar el programa.
initial config data	Datos de configuración inicial para comienzo de tests.
Visualization data	Información necesaria para el visualizado del programa.
Performance data	Señal con métricas sobre el desempeño y uso de recursos del programa.

TABLA 7.4: Señales presentes en el banco de pruebas 3.

7.1.4. Banco de pruebas 4

En la figura 7.5 se esquematiza el banco de pruebas correspondiente a la interfaz de comunicación sensores-dispositivos.

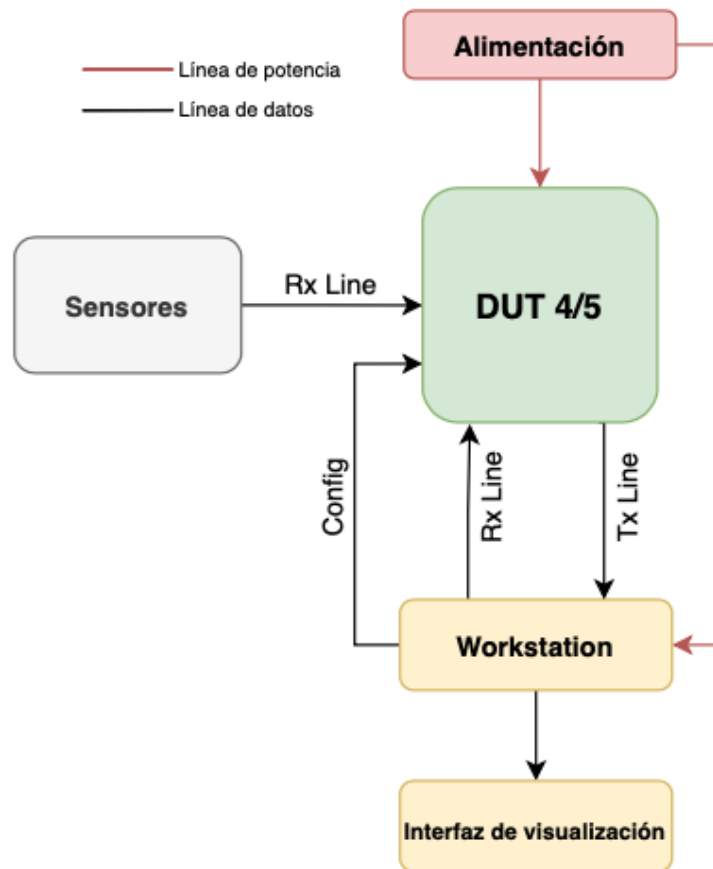


FIGURA 7.5: Banco de pruebas para validación de comunicación sensores-dispositivo - Diagrama en bloques.

A continuación se detallan sus componentes:

1. **Alimentación.** Se suministra alimentación de red (220V - 50Hz).
2. **Workstation.** Posee doble funcionalidad, es capaz de simular los paquetes de datos de los sensores y además puede recibir la data proveniente del DUT. Por otro lado, también permite la configuración del DUT y enviar los resultados a una interfaz de visualización.
3. **Interfaz de visualización.** Los datos de visualización en la pantalla conectada a la workstation. Los datos pre procesados y las métricas necesarias son mostradas en ella.
4. **Sensores.** Sensores montados en la bicicleta. Para tests posteriores debe ser posible medir la performance del DUT utilizando las señales reales y no simulaciones.
5. **DUT 4/5.** Los dispositivos a testear, pueden intercambiarse los módulos de comunicación sensores-dispositivo y dispositivo-servidor.

Es importante en el caso de las comunicaciones prestar atención a los retardos en el envío y recepción de las señales, estos se han de medir con la finalidad de garantizar que el sistema funciona correctamente.

7.1.5. Banco de pruebas 5

En la figura 7.6 se aprecia el diagrama en bloques del banco de pruebas para la validación de la aplicación de usuario.

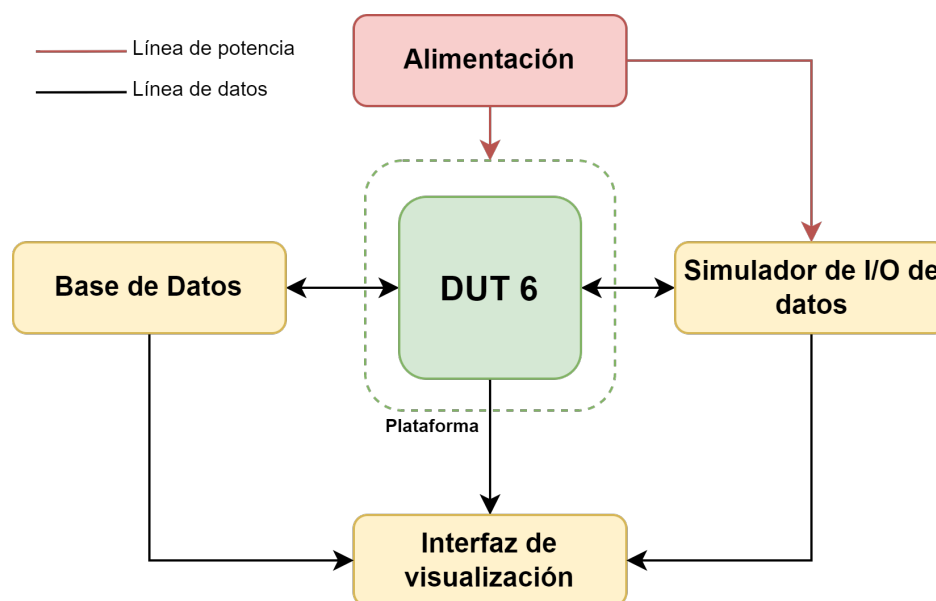


FIGURA 7.6: Banco de pruebas para validación de aplicación de usuario - Diagrama en bloques.

Los componentes de este banco de pruebas son:

1. **Alimentación.** Se suministra alimentación de red ($220V - 50Hz$).
2. **Base de Datos.** Base de datos ubicada en la nube. Interfaz de red con el DUT.
3. **Simulador de I/O de datos.** Dispositivo que corre un script que emula el flujo de datos de entrada del DUT, y también recibe las respuestas del mismo.
4. **Interfaz de visualización.** Interfaz que presenta de forma gráfica la salida de cada uno de los módulos a los que se la conecte.
5. **DUT 6.** Dispositivo a testear (Aplicación de usuario).

7.1.6. Banco de pruebas 6

El último banco de pruebas es el correspondiente al testeo de la aplicación del entrenador. En la figura 7.7 pueden verse sus partes:

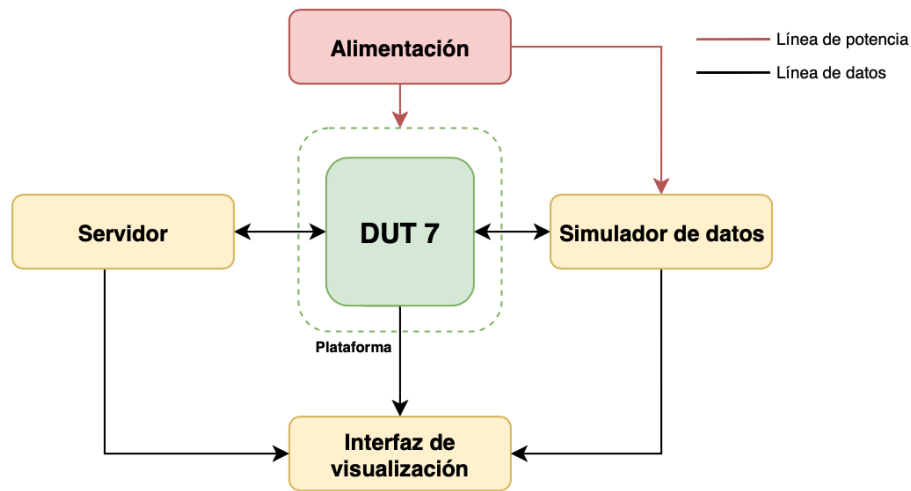


FIGURA 7.7: Banco de pruebas para validación de la aplicación del entrenador.

A continuación se detallan sus componentes:

1. **Alimentación.** Se suministra alimentación de red (220V - 50Hz).
2. **Servidor.** Servidor que comunica la aplicación del entrenador con la base de datos y el visor VR. Hay comunicación bidireccional entre el DUT y el servidor.
3. **Interfaz de visualización.** Interfaz que permite visualizar por separado la aplicación del entrenador, los datos simulados y los datos del servidor.
4. **Simulador de datos.** Módulo que ejecuta un programa que emula los datos que ingresan al DUT, además debe recibir los datos correspondientes.
5. **DUT 7.** Dispositivo a testear (Aplicación de coach).

7.2. Especificaciones de tests

A continuación se detallan los tests que han de ser llevados a cabo, acorde a cada uno de los bancos de pruebas propuestos en la sección anterior. La tabla 7.5 indica el identificador de cada test junto con una breve descripción.

ID	Tipo	Descripción
T-HW01	Hardware	Test de sensado de velocidad de pedaleo.
T-HW02	Hardware	Test de simulación de carga de la bicicleta.
T-HW03	Hardware	Test de respuesta de vibración.
T-VR01	App VR	Test de uso normal del sistema.
T-VR02	App VR	Test de sistema con picos de carga.
T-VR03	App VR	Test de fallas (máxima carga admisible).
T-VR04	App VR	Test de múltiples usuarios.
T-COM01	Comunicación	Test de ping.
T-COM02	Comunicación	Test de retardo.
T-COM03	Comunicación	Test de formato.
T-COM04	Comunicación	Idem T-COM03 con sensor.
T-COM05	Comunicación	Idem T-COM01, para el servidor.
T-COM06	Comunicación	Idem T-COM02, para el servidor.
T-COM06	Comunicación	Idem T-COM03, para el servidor.
T-UA01	App usuario	Test de base de datos.
T-UA02	App usuario	Test de autenticación/seguridad.
T-CA01	App del coach	Test de envío de datos a aplicación.
T-CA02	App del coach	Test de envío de datos con carga elevada.
T-CA03	App del coach	Test de envío de datos al servidor.
T-CA04	App del coach	Test de visualización de datos en tiempo real.

TABLA 7.5: Tests a realizar

7.2.1. Banco de pruebas 1

Precondiciones: Ninguna		
ID Aplicabilidad	Procedimiento	Criterio
T-HW01 P, F	1. Configurar el sensor de velocidad patrón para que mida la velocidad del volante de inercia en conjunto con la medición de velocidad del volante de inercia propio de nuestra solución 2. Anotar las medidas de velocidad de ambos sensores en diferentes velocidades.	Comprobar que las medidas sean iguales considerando un 10 % de error.
T-HW02 P, F	1. Realizar un barrido en el ángulo de posición en el motor solidario a la varilla roscada. Para cada posición revisar la fuerza en la llave torquimétrica. 2. Luego de realizado el barrido, observar y procesar los datos	Comprobar que el promedio de los errores porcentuales para cada dato no exceda el 15 % del valor esperado

TABLA 7.6: Tests - Banco de pruebas 1

7.2.2. Banco de pruebas 2

Precondiciones: DUT 2 y banco de pruebas deben estar correctamente anclados al piso.		
Procedimiento general: Conexión del DUT 2 al banco de pruebas N°2.		
ID Aplicabilidad	Procedimiento	Criterio
T-HW03 P, F	1. Comprobar conexión del DUT 2 a la alimentación. Verificar encendido. 2. Verificar conexión del data logger al sensor de vibración. 3. Encender el data logger. 4. Configurar el DUT 2 en modo prueba de vibración. Esto causa que el DUT 2 emita un barrido de vibraciones con intensidad creciente. 5. Luego de realizado el barrido, procesar y observar los datos recolectados desde la interfaz de visualización. Comparar los datos medidos con los obtenidos del barrido.	Comprobar que el promedio de los errores porcentuales para cada dato no exceda el 15 %.

TABLA 7.7: Tests - Banco de pruebas 2

7.2.3. Banco de pruebas 3

<p>Precondiciones: Ninguna.</p> <p>Procedimiento general: Para el siguiente set de tests se debe conectar la computadora utilizada para ejecutar el ambiente de desarrollo integrado. El primer paso es comprobar la conexión de la computadora a alimentación y al DUT, verificar su encendido y abrir el ambiente de desarrollo.</p>		
ID Aplicabilidad	Procedimiento	Criterio
T-VR01 P, F	<ol style="list-style-type: none"> 1. Configurar escenario de prueba 1 (load testing). 2. Realizar 3 recorridos del escenario con una definición de 2K y variando las configuraciones gráficas de menos a más demandantes. 3. Repetir con una definición de 4K. 4. Recopilar los datos y comparar entre las distintas configuraciones. 	Comprobar que la tasa media de marcos por segundo sea superior a 30 para cualquier configuración con un error del 10 %.
T-VR02 P, F	<ol style="list-style-type: none"> 1. Configurar escenario de prueba 2 (spike testing). 2. Recorrer el escenario en ambas definiciones (2K y 4K). 3. Recopilar los datos y observar las métricas en los 5 puntos dentro del escenario en los que se aumenta la carga sobre la GPU y la CPU. 	Comprobar que la tasa media de marcos por segundo sea superior a 30 con un error del 10 % y no caiga debajo de 20 por un tiempo superior a 5 segundos en las zonas de mayor demanda.
T-VR03 P, F	<ol style="list-style-type: none"> 1. Configurar escenario de prueba 3 (stress testing). 2. Recorrer el escenario, en caso de que el escenario falle reiniciar la prueba. 3. Recopilar los datos y notar en que etapas del recorrido se da la peor tasa de refresco. 	<p>Comprobar que el programa no falle.</p> <p>Comprobar que la tasa media de marcos por segundo no caiga por debajo de 5 durante un tiempo mayor a 2 segundos.</p>
T-VR04 P, F	<ol style="list-style-type: none"> 1. Configurar escenario de prueba 4 (load testing 2). 2. Recorrer el escenario por completo 3 veces, aumentando la cantidad de bicicletas simuladas en el escenario. 3. Recopilar los datos y analizar los gráficos de performance. 	Comprobar que la tasa media de marcos por segundo sea superior a 30 para cualquier configuración con un error del 10 %.

TABLA 7.8: Tests - Banco de pruebas 3

7.2.4. Banco de pruebas 4

<p>Precondiciones: Ninguna.</p> <p>Procedimiento general: Para el siguiente set de tests debe ser posible utilizar el mismo banco de pruebas para cualquiera de los dos DUTs. Los primeros 3 tests utilizan el DUT 4 en modo de simulación de input. Los últimos utilizan el DUT 5.</p>		
ID Aplicabilidad	Procedimiento	Criterio
T-COM01 P, F	1. Enviar paquetes de datos de prueba. 2. Esperar respuesta en la interfaz.	Comprobar que se reciba una respuesta no nula de parte del dispositivo de comunicación.
T-COM02 P, F	1. Enviar paquetes de datos de prueba. 2. Esperar respuesta en la interfaz.	Comprobar que el tiempo de retardo sea menor o igual a 50 ms.
T-COM03 P, F	1. Enviar paquetes de datos de prueba. 2. Esperar respuesta en la interfaz.	Comprobar que el retardo sea menor o igual a 50ms y que los datos enviados sean iguales a los recibidos.
T-COM04 P, F	1. Configurar DUT 4 en modo de testeo real. 2. Enviar señales con los sensores de la bicicleta. 3. Esperar respuesta en la interfaz.	Comprobar que el retardo sea menor o igual a 50 ms y que los datos enviados sean iguales a los recibidos.
T-COM05 P, F	1. Enviar paquetes de datos. 2. Esperar respuesta en la interfaz.	Comprobar que se reciba una respuesta no nula de parte del dispositivo de comunicación.
T-COM06 P, F	1. Enviar paquetes de datos. 2. Esperar respuesta en la interfaz.	Comprobar que el retardo sea menor o igual a 50 ms.
T-COM07 P, F	1. Enviar paquetes de datos. 2. Esperar respuesta en la interfaz	Comprobar que los datos recibidos sean iguales a los enviados.

TABLA 7.9: Tests - Banco de pruebas 4

7.2.5. Banco de pruebas 5

Precondiciones: Disponer de acceso a Internet. Procedimiento general: Conexión del DUT 6 al banco de pruebas N°5.		
ID Aplicabilidad	Procedimiento	Criterio
T-UA01 P, F	1. Comprobar conexión de la plataforma del DUT a la alimentación. Verificar encendido. 2. Comprobar conexión del simulador de I/O a la alimentación. Verificar encendido. 3. Comprobar conexión con la base de datos en la nube. 4. Configurar el simulador de I/O para que simule una carga de usuarios de prueba en una sesión ficticia. 5. Inspeccionar mediante la interfaz de visualización los datos que contiene la base de datos de la nube.	Los datos observados en la interfaz de visualización deben coincidir en estructura con los enviados por el simulador.
T-UA02 P, F	1. Comprobar conexión de la plataforma del DUT a la alimentación. Verificar encendido. 2. Comprobar conexión del simulador de I/O a la alimentación. Verificar encendido. 3. Comprobar conexión con la base de datos en la nube. 4. Configurar el simulador de I/O para publicar en la base de datos sesiones con métricas y datos de usuarios ficticios autenticados. 5. <i>Login</i> con credenciales de uno de los usuarios autenticados. 6. Descargar datos de la base de datos. 7. Repetir pasos 5 y 6 con todas las cuentas descritas en paso 4.	Comprobar que el los datos descargados corresponden a sus respectivas cuentas de usuario.

TABLA 7.10: Tests - Banco de pruebas 5

7.2.6. Banco de pruebas 6

Precondiciones: Ninguna. Procedimiento general: Para el siguiente set de tests debe estar disponible el servidor para recibir los datos provenientes de la aplicación. Además, debe alimentarse el dispositivo que ejecuta la aplicación y el que simula los datos.		
ID Aplicabilidad	Procedimiento	Criterio
T-CA01 P, F	1. Enviar paquetes de datos de prueba desde el simulador de datos. 2. Esperar respuesta en la interfaz.	Comprobar que se reciba una respuesta no nula de parte del dispositivo .
T-CA02 P, F	1. Enviar paquetes de datos de prueba simulando 15 usuarios. 2. Esperar respuesta en la interfaz.	Comprobar que se reciba una respuesta no nula de parte del dispositivo con una carga alta.
T-CA03 P, F	1. Enviar comandos desde la app hacia el servidor. 2. Esperar respuesta en la interfaz.	Comprobar que los datos enviados al servidor sean correctos.
T-CA04 P, F	1. Enviar paquetes de datos de prueba simulando 15 usuarios. 2. Esperar respuesta en la interfaz..	Comprobar que los datos se visualicen en tiempo real con una carga alta.

TABLA 7.11: Tests - Banco de pruebas 6

7.3. Plan de test

Se propone como plan de testeo el siguiente diagrama:

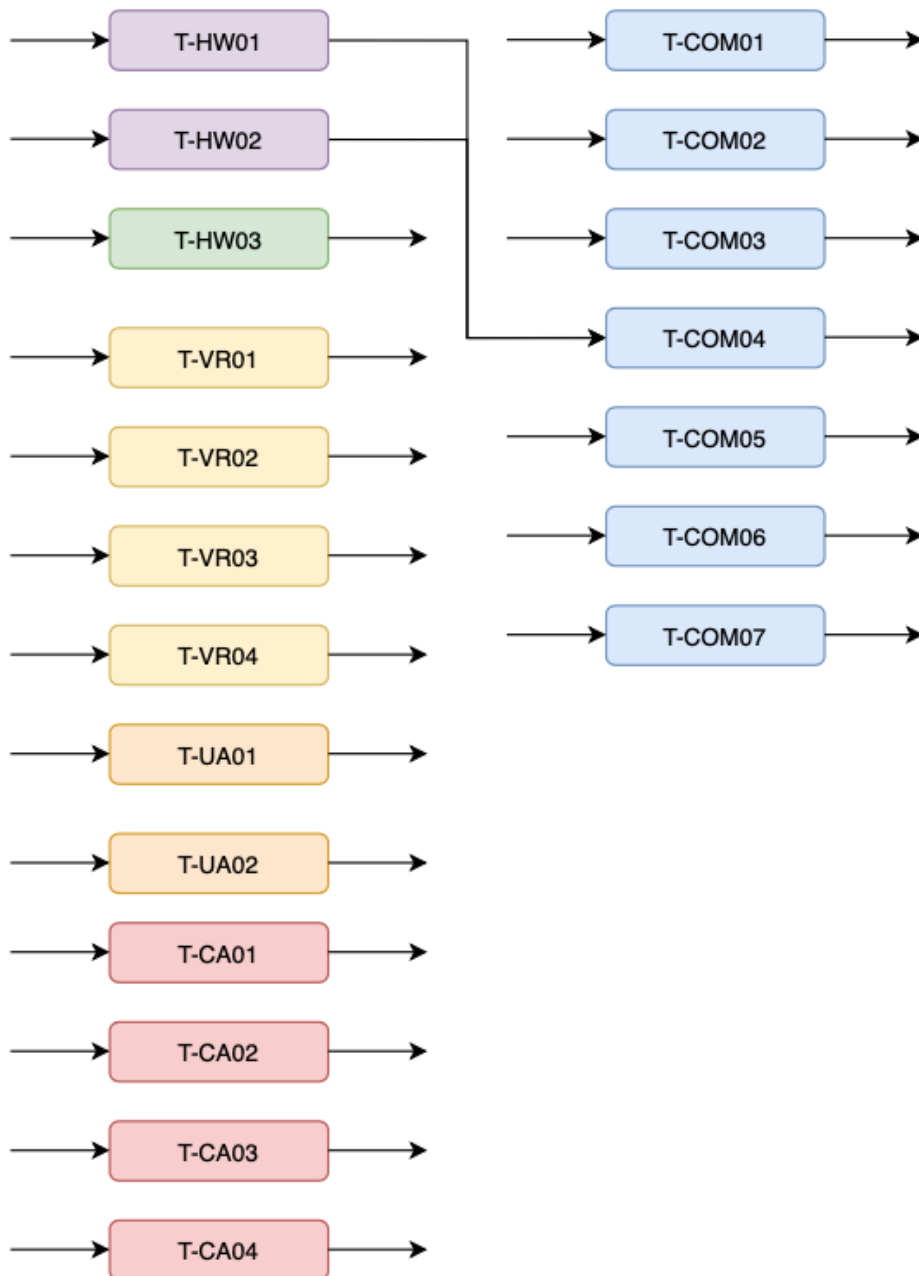


FIGURA 7.8: Diagrama de etapas para plan de testeo.

Es importante notar que de la manera en que está planeado es posible testear el software de manera independiente al resto de los sistemas, mientras que para el testeo de del módulo de comunicación es necesario validar módulos de hardware.

8. Factibilidad tecnológica

En este apartado se desglosa el diseño en módulos, y se evalúa cada una de las posibilidades de diseño de estos módulos a nivel técnico/tecnológico. Se elige una alternativa en función de diversos criterios, que pueden variar de un módulo a otro.

8.1. Alternativas de diseño

En esta sección se detallan las diferentes alternativas de diseño que se tienen en consideración para el desarrollo del proyecto, comparándolas entre sí con la finalidad de encontrar de esta manera la mejor configuración para los fines del proyecto.

8.1.1. Sensor de velocidad

Este sensor es el encargado de medir la velocidad del ciclista. Su importancia es crítica, ya que es la principal interfaz de movimiento que presenta el software de realidad virtual, afectando considerablemente a la experiencia percibida por el usuario. A continuación se presentan las diferentes alternativas para la medición de la velocidad angular.



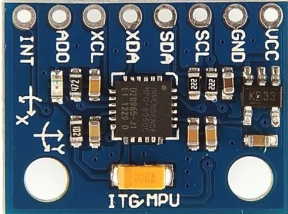
			
Modelo	TCRT5000	OMRON E6B2-CWZ6C	MPU6050
Tipo	Óptico	Encoder rotativo	Acelerómetro + Giróscopo
Montaje	Radial	Axial	Solidario al giro
Alimentación	5 VCC	5-24 VCC	5 VCC
Precio (ARG)	USD 0,45	USD 75	USD 1,65
Ventajas	<ul style="list-style-type: none"> - Montaje simple - Baja intervención mecánica - Costo - Disponibilidad 	<ul style="list-style-type: none"> - Precisión alta 	<ul style="list-style-type: none"> - Montaje simple - Baja intervención mecánica - Colocación en pedal - Costo
Desventajas	<ul style="list-style-type: none"> - Medición de velocidades cercanas a 0 	<ul style="list-style-type: none"> - Montaje complejo - Invasivo - Costo 	<ul style="list-style-type: none"> - Ruido - Poca precisión

TABLA 8.1: Comparación entre sensores de velocidad.

Analizando la investigación volcada en la tabla 8.1, se concluye que la opción más conveniente es el sensor óptico TCRT5000. Este sensor se monta en la periferia del volante de inercia, sensando su rotación. Se colocan en el volante de inercia zonas con cinta reflectiva de forma tal de poder identificar cada uno de los giros. La precisión de este sistema se regula variando el ángulo entre marcas reflectivas (y por ende, se varía también su cantidad), suavizando el movimiento percibido por el ciclista ante cambios en su velocidad de pedaleo, mejorando notablemente la experiencia de usuario. Uno de los principales factores de elección de este dispositivo corresponde a su facilidad de montaje, reduciendo la intervención a realizar sobre la bicicleta objeto de la adaptación. Asimismo, su bajo costo no tiene impacto significativo sobre los costos del producto final.

El encoder rotativo es descartado por su complejidad de montaje, tarea que implica desmontar el sistema de transmisión de la bicicleta y acoplar mecánicamente el eje del encoder al eje de giro del volante de inercia, sumado a los problemas que incorpora en relación con el mantenimiento y a la fiabilidad. Por último, el conjunto acelerómetro-giróscopo no es elegido como alternativa óptima, dado que la medición de la velocidad angular en este caso se hace de una forma más indirecta

respecto de los otros dos métodos, introduciendo ruido en la estimación. Asimismo, dado que este conjunto debe girar junto con el volante de inercia (giro no concéntrico), luego se complejiza la conexión eléctrica al sensor.

8.1.2. Interfaz de comunicación local

La interfaz de comunicación local es un dispositivo que se encarga de la adquisición y procesamiento de los datos recolectados por los sensores, y de estimular a los actuadores en función de lo determinado en la aplicación de realidad virtual, por lo que debe ser lo suficientemente rápido como para adquirir y enviar los datos en tiempo real, para no comprometer la experiencia de usuario. Este módulo se comunica con los sensores y actuadores en forma cableada, y al dispositivo que corre la aplicación de realidad virtual mediante un enlace inalámbrico. En este punto, se destaca que el dispositivo que ejecuta la aplicación de realidad virtual ya se encuentra conectado por red LAN al servidor. Otra característica relevante que debe ofrecer la solución de conectividad es lograr un bajo consumo de energía, para optimizar el uso de la batería del dispositivo que *renderiza* la aplicación de realidad virtual.

Habiendo definido la topología de la red, se emplea un enlace Bluetooth para la comunicación del microcontrolador con el visor de realidad virtual.

Nombre	Modo	Tipo	Descripción
SPD_IN	Entrada	Analógica	Sensor de velocidad
VIB_OUT	Salida	Digital (PWM)	Motores de vibración
LOAD_OUT	Salida	Digital (PWM)	Control de carga
APP_COM	Bidireccional	Bluetooth	Comunicación con app de VR.

TABLA 8.2: Señales de sensores/actuadores.

En la tabla 8.2 se resumen las entradas y salidas requeridas para este módulo.

		
Modelo	ESP32	RP2040 + HM10
Dimensiones	48mm x 26mm x 11,5mm	51mm x 21mm x 1mm (RPI) 37mm x 16mm x 2,2mm (HM10)
Procesador	Tensilica Xtensa LX6 (32bit dual-core @ 240 MHz)	RP2040 (32bit ARM dual-core @ 133 MHz)
SRAM	520 KB	264 KB
Flash	4 MB	2 MB
Pines GPIO	36 ADC (x15) DAC (x2)	26 ADC (x3)
Corriente máxima salida (GPIO)	40mA	16mA
Plataforma	C++	MicroPython, C, C++
Alimentación	3,3V (5V vía microUSB)	3,6V a 6V (HM10) 1,8V a 5,5V (RP2040) (5V vía microUSB)
Interfaces de comunicación	UART (x3) SPI (x4) I2C (x2) Bluetooth 4.2 + BLE Wifi 802.11 b/g/n (2,4GHz)	UART (x2) (uno usado por HM10) SPI (x2) I2C (x2) Bluetooth 4.0 + BLE (HM10)
Precio (ARG)	USD 8,5	USD 4,2 (RP2040) USD 4 (HM10) USD 8,2 (Total)
Ventajas	<ul style="list-style-type: none"> - Amplia comunidad - Bibliotecas - Tamaño pequeño - Disponibilidad - CPU potente 	<ul style="list-style-type: none"> - Bajo costo - Tamaño pequeño - Disponibilidad
Desventajas	<ul style="list-style-type: none"> - Costo superior 	<ul style="list-style-type: none"> - Menor potencia - En etapa de prototipado se usan dos módulos

TABLA 8.3: Alternativas de microcontrolador

Analizando la tabla 8.3, se concluye que el ESP32 es apropiado para la aplicación en cuestión, dado que es un conjunto que cuenta con mucho tiempo en el mercado, haciéndolo más confiable. Por otro lado, a un costo similar, la opción del ESP32 brinda el doble de recursos (tanto en memoria como en velocidad de la CPU) respecto al RP2040. Además, el ESP32 tiene una placa de desarrollo que integra tanto la CPU como el módulo de comunicación Bluetooth, por lo que se facilita la etapa de prototipado.

8.1.3. Dispositivo de visualización


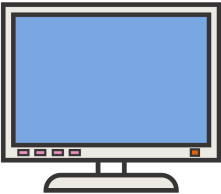
El dispositivo de visualización es un componente clave, ya que es la manera en la que el usuario interactúa con el mundo virtual y dependiendo de la calidad de la interfaz, se logrará una mejor o peor experiencia de usuario.

Hay dos caminos diferentes que pueden tomarse para elegir un tipo de interfaz, existen soluciones que no precisan de procesamiento externo y otras que sí. A su vez, podemos hacer otra división entre experiencias más o menos inmersivas. Estas divisiones llevan a plantear dos tipos de alternativas: utilizar alguna clase de visor de realidad virtual o integrar una pantalla sobre la bicicleta, la cual requerirá de una PC que se encargará de procesar y renderizar los niveles.

Si se comienza analizando el caso de los visores, una opción muy atractiva podría ser la utilización de dispositivos como el Samsung Gear VR, el cual utiliza un teléfono celular como pantalla y procesador. La complicación que tienen este tipo de dispositivos es, en primer lugar, que no existe un solo visor de este estilo compatible con toda clase de teléfonos móviles. En segundo lugar, las dos principales opciones, el Samsung Gear VR y el Google Daydream han sido discontinuados hace ya más de tres años, por lo que no existen actualizaciones de software ni soporte de ningún tipo. Por esta razón y el hecho de que no se puede tener un visor de este tipo que sea compatible con cualquier celular, se decidió descartar los visores de esta clase.

Dentro de la otra categoría de visores existen visores que precisan de un procesamiento externo y visores con procesadores integrados. Hay diferentes marcas y modelos, pero uno que destaca de entre el resto es el Oculus Quest 2, debido a su precio, a que no es necesaria una PC para utilizarlo y su cuota de mercado (80 %). Se analizaron otras marcas y modelos, pero la relación entre las prestaciones y el precio del visor Oculus eran superiores a las de la competencia (a esto hay que añadirle que puede conseguirse fácilmente en el mercado local).

Hablando ya acerca de la opción de montar un monitor sobre la bicicleta, esto trae dos inconvenientes: es necesario armar una PC para procesar los niveles y además la experiencia ofrecida no tiene el mismo nivel de inmersión que el visor. A continuación se detallan ambas opciones en una tabla comparativa:

	 <p>Oculus Quest 2</p>	 <p>Pantalla + PC</p>
Tipo	Standalone/Tethered	Tethered
Conectividad	Bluetooth y WiFi	Bluetooth y WiFi
SO	Android 10 modificado.	Windows
Pantalla	1832x1920, 60-120 Hz.	1920x1080, 60 Hz
Almacenamiento	128 Gb.	256 Gb
RAM	6 Gb.	8 Gb
Procesador	Qualcom XR2.	Intel Core i3 1010S
Bateria	2,5 - 3 horas	-
Carga	100 % en 2,5 horas.	-
Peso	500 gramos.	2.5 Kg
Audio	Parlante y micrófono integrados	No.
Precio (Sin IVA)	400 USD ¹	500 USD
Adicionales	Incluye controles, obligatorio ingresar a Facebook.	Costos adicionales, motherboard, fuente, gabinete, etc.

¹Makkax. Última consulta: 24/11/2021

TABLA 8.4: Comparación: visor-PC

Para la tabla comparativa anterior se configuró una PC con las características adecuadas para renderizar los niveles y teniendo en cuenta que se logre una potencia de procesamiento similar a la del visor Oculus. Los componentes utilizados son los siguientes:

Componente	Modelo	Precio (USD)
Placa Madre	Gigabyte H510M	55,35
Procesador	Intel Core i3 1010S	112,5
Cooler	Deepcool Gammaxx 400 V2	22,5
Memoria RAM	2x Crucial DDR4 4Gb	26,5
Placa de Video	AFOX GeForce GTX 750	120
Almacenamiento	SSD PNY 240 Gb	20
Gabinete	Kolink K6 mini ATX	31
Fuente de Poder	Deepcool 600W	35
Monitor	LG 20" 20MK400H-B	125
Armado y Envío	-	15
Total		562,85

TABLA 8.5: Componentes PC

Los dispositivos presentados poseen una capacidad de procesamiento similar, pero una desventaja clara del visor de realidad virtual es la necesidad de carga. La batería tiene una duración de hasta 3 horas con un uso moderado, pero debido a que las clases de spinning tienen una duración de 45 minutos y nunca se hacen más de dos clases seguidas sin pausa, la autonomía del visor no debería ser un inconveniente siempre y cuando se carguen durante dichas pausas. Además de la batería, la cuestión higiénica es importante. Por esos se propone la utilización de una funda de silicona lavable que va por sobre la goma espuma del visor y así se evita el contacto entre usos.

A pesar de que el precio de una PC más una pantalla es superior al del visor, finalmente este no fue el criterio determinante a la hora de la elección final, debido a que la diferencia es solamente del 15 % y se espera que los precios del hardware bajen a medida que pase el tiempo. Algo que sí fue un criterio determinante es la experiencia que podría tener el usuario realizando ejercicio.

Para lograr cumplir con el objetivo principal del proyecto, es decir, potenciar la actividad física mediante la inmersión propia de un ambiente virtual, un visor de VR ofrece una experiencia muy superior a la de una pantalla. Además, la integración del ejercicio real con lo que sucede en el mundo virtual es el factor diferenciador del proyecto. Por eso, mientras se pueda mantener una buena experiencia de usuario, se considera que la opción del visor cumple más adecuadamente con las especificaciones propuestas.

8.1.4. Diseño de entornos virtuales

El diseño del entorno virtual en que el usuario se verá inmerso es fundamental para aprovechar al máximo la experiencia de realidad virtual que se propone. Por esto mismo, se plantean dos alternativas de diseño: utilizar ambientes reales, es decir, grabaciones similares a la tecnología de Google StreetView, o diseñar, programar e implementar mapas propios, con ambientes sintéticos parecidos a los niveles que poseen los videojuegos modernos. A continuación se hace un análisis al respecto:



	 Escenarios reales	 Escenarios sintéticos
Calidad	Dependiente de la calidad de la fuente.	Programable, pero limitado por el hardware.
Uso de Memoria	Depende del tamaño y calidad gráfica del nivel requerido.	Depende del tamaño y calidad gráfica del nivel requerido.
Tipos de entornos	Imágenes/Videos de lugares reales.	Realismo variable de acuerdo al diseño de producción de los niveles.
Tiempo de desarrollo	Inicialmente medio, luego bajo.	Alto.
Grado de libertad del nivel	Ruta fija o recorrido libre en las calles.	Recorrido fijo o libre.
Ventajas	<ul style="list-style-type: none"> - Desarrollar nuevos niveles es rápido. - Navegar por calles con detalles reales puede mejorar la inmersión. - Se pueden lograr recorridos en ciudades y lugares turísticos. 	<ul style="list-style-type: none"> - El nivel de detalle es controlable. - Los niveles son optimizables - Los niveles son continuos.
Desventajas	<ul style="list-style-type: none"> - Capacidad de almacenamiento mayor (para muchos niveles). 	<ul style="list-style-type: none"> - Mayor tiempo de desarrollo. - La calidad de los niveles depende de la habilidad del programador. - Pueden generarse errores que empeoren la experiencia.

TABLA 8.6: Comparación entre tipos de entornos virtuales.

La calidad es una cualidad esencial para asegurar la inmersión, por lo que los escenarios sintéticos tienen el problema de tener gráficos encarados a videojuegos que no necesariamente le va a gustar a todas las personas que utilicen el equipo. Por lo que la opción de escenarios reales basados en videos 360 presentan un mejor desempeño en este apartado.

Si se analiza la utilización de mapas sintéticos, la principal ventaja es que son completamente programables, pero, por otro lado, requieren de grandes tiempos de desarrollo para generar nuevos niveles que funcionen correctamente sin errores en los mismos.

La elevación y el tipo de superficie son otros dos factores fundamentales a la hora de lograr una experiencia completa, y para el caso de los escenarios reales deben ser agregados, ya que no se posee la información acerca de la pendiente y elevación, pero, por otro lado, una vez desarrollado el pipeline del etiquetado de las características del terreno en cada momento, luego este paso es rápidamente extrapolable a otros videos.

Por todo lo anterior se decide elegir el uso de escenarios generados mediante videos 360, debido principalmente al realismo que los mismos aportan y a la facilidad de desarrollo que permite generar

nuevos escenarios en poco tiempo. La posibilidad de utilizar escenarios reales también aporta a la experiencia, esto si bien necesita la generación del video 360 con la correspondiente inversión en la cámara 360, en el largo plazo esta inversión termina siendo más rentable que la opción de los escenarios sintéticos.

Sumado a lo anteriormente mencionado, se consultó con los posibles compradores (encargados de dos de las principales cadenas de gimnasios de Buenos Aires), expresando estos que la opción de escenarios basados en videos 360 podría tener más futuro para las clases de spinning.

8.1.5. Networking

En la presente sección se analizan los distintos aspectos de la red implementada a fin de lograr una experiencia multijugador. Esto incluye la presencia de un servidor, sus características, el conexionado entre los nodos y el alojamiento de los elementos necesarios. Este servidor tiene las funciones de sincronizar a los distintos participantes de la sesión multijugador, transmitir las estadísticas de cada uno de ellos, transmitir los mensajes entre el entrenador y los participantes, y enviar a una base de datos las métricas de los participantes de la sesión.

Servidor En primer lugar, es necesario analizar la forma que toma el servidor. Este puede tratarse tanto de un servidor alojado en uno de los participantes de la sesión (servidor jugador), o tratarse de un dispositivo independiente, el cual únicamente cumple las funciones de servidor. Para ello, se los compara en la tabla 8.7.

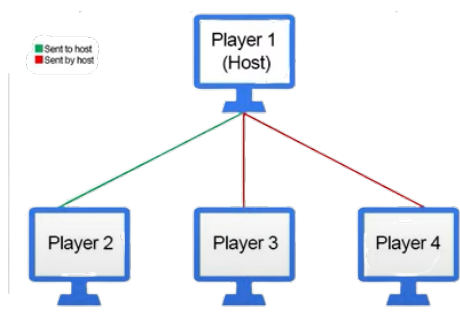
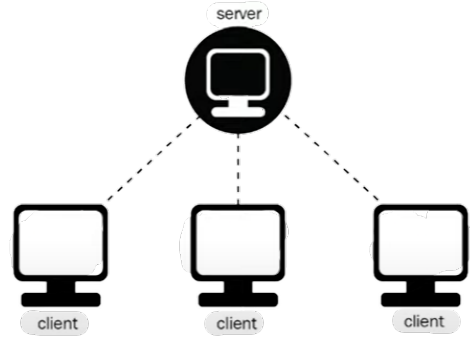
	Servidor jugador	Servidor dedicado
		
Tráfico	Centralizado.	Centralizado.
Costo	-	Costo del servidor.
Sesión	Depende de uno de los participantes (jugador host).	Depende del servidor.
Ventajas	- Costo	- Performance - Seguridad
Desventajas	<ul style="list-style-type: none"> - El jugador que actúa como servidor (jugador host) puede tener ventaja. - Problemas por desconexión del jugador host. - Vulnerable a fallas en el jugador host. 	- Costo

TABLA 8.7: Comparación entre tipos de servidor.

Analizando las dos alternativas presentadas, se decide utilizar un servidor dedicado dado que resulta considerablemente superior al contrastarlos. Como se menciona en la tabla, el servidor jugador es muy vulnerable a desconexiones o fallas en el dispositivo donde se encuentra actuando como servidor. Esto implica que la partida finalice abruptamente o comience un proceso de migración de *host*, algo que resulta muy complejo. Además, el jugador que sea servidor presenta una clara ventaja en cuanto a latencia. Adicionalmente, para que un jugador puede actuar como servidor debe tener una capacidad de red y de cómputo suficiente, lo cual genera que sea poco escalable e incorpora un requerimiento en dichos dispositivos. Por el contrario, un servidor dedicado puede escalar fácilmente, ya que basta con agregar capacidad de red o de cómputo. Por otro lado, es más seguro y presenta una mayor performance debido a que el dispositivo está únicamente planificado para tal fin.

Alojamiento del servidor Luego de determinar que se utiliza un servidor dedicado, el próximo paso es analizar si dicho servidor se encuentra en la misma sala en la que se realiza la sesión de spinning (servidor *on premise*) o si se utiliza un servidor alojado en la nube (servidor *cloud*). Para determinarlo, se contrastan en la tabla 8.8 las características de cada uno.

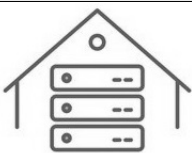

		
Tipo	Servidor <i>On Premise</i>	Servidor <i>Cloud</i>
Costo	Adquisición del servidor.	Pago mensual por uso.
Latencia	<30ms	De 50ms a 200ms
Ventajas	<ul style="list-style-type: none"> - Baja latencia. - Independencia del proveedor de Internet. 	<ul style="list-style-type: none"> - Alta escalabilidad. - Alta disponibilidad. - Alcance a múltiples salas.
Desventajas	<ul style="list-style-type: none"> - Niveles de disponibilidad. - Baja escalabilidad. 	<ul style="list-style-type: none"> - Latencia. - Dependencia del proveedor de Internet.

TABLA 8.8: Comparación entre alojamiento del servidor.

Tras contrastar las características de los servidores alojados en la nube y los servidores *on premise*, se concluye que la independencia del proveedor de Internet y la baja latencia de un servidor *on premise* resultan en una mejor experiencia de usuario. Por lo tanto, se opta por este tipo de alojamiento.

Sin embargo, un servidor alojado en la nube podría ampliar el alcance del proyecto a futuro. Con un servidor en la nube sería posible realizar sesiones multijugador entre distintos gimnasios y eventualmente incorporar particulares a dichas sesiones. Si bien esto no forma parte del alcance actual de VRBike, se lo tiene en cuenta a fin de evitar un gasto innecesario en el futuro debido a un rediseño completo del servidor. Por ello, se considera utilizar la tecnología de Docker, la cual permite independizar al servidor del entorno donde se ejecute. Así, se podría ejecutar el servidor *on premise* y, en un futuro, si se quisiera ampliar el alcance, ejecutar otra instancia del servidor, pero en este caso en la nube.

Docker permite empaquetar las aplicaciones en contenedores -componentes ejecutables estandarizados que combinan el código fuente de la aplicación con las bibliotecas del sistema operativo (SO) y las dependencias necesarias para ejecutar ese código en cualquier entorno.

En conclusión, se desarrollará el código del servidor y se lo empaquetará junto con sus dependencias en un contenedor Docker. Este contenedor se ejecutará en un entorno *on premise* con la finalidad de maximizar la experiencia de usuario. Implementarlo de esta forma permite que, si en el futuro o en casos particulares se lo necesita, dicho contenedor pueda ser ejecutado en un entorno de nube.

Switch - Access Point El sistema tiene que poder conectar los diferentes visores de realidad virtual con el servidor local que conecta a todos los miembros de la red. Para esto se hace uso de las diferentes direcciones IP que brinda el proveedor del servicio al gimnasio. Cada uno de los módems presenta por defecto un número limitado de dispositivos con los que se puede conectar.

Esto presenta el siguiente problema: si existen muchos dispositivos conectados al módem de internet Wifi, como suele ocurrir en los gimnasios, debido a que tanto la recepción como los empleados, como muchos de los clientes están conectados a dicha red, se puede saturar la red al agregar la necesidad de conectar las bicicletas (es decir, se sobrepasa el número máximo de dispositivos conectados). Esto se debe a que cada una de las bicicletas es considerada como un dispositivo extra que se añade a la red y como el número de bicicletas a agregar en el gimnasio podría ser grande (15, por ejemplo), la red podría colapsar.

Para solucionar esto podemos plantear dos enfoques diferentes, el primero es modificar las configuraciones del módem para que soporte más dispositivos conectados. Esto es algo que se puede realizar desde la configuración de la mayoría de los módems actuales. El problema de esta opción es que más allá de la modificación en las configuraciones de un módem de un tercero, con la consecuente variabilidad en el modelo y funciones del mismo, también genera el hecho de que al aumentar la cantidad máxima de conexiones con otros dispositivos se reduzca el ancho de banda que cada uno de los mismos puede utilizar en la red, haciendo que por ejemplo la computadora en la recepción funcione con un peor Internet que el que tenía antes de la modificación.

De esta manera, como segunda alternativa para enfrentar este problema, se considera el uso de un Switch que se encargue de trabajar como una interfase entre la comunicación entre las bicicletas, el servidor de multijugador y el módem del gimnasio. De esta manera logramos reducir las hipotéticamente 15 conexiones que comentamos en el ejemplo a 1 sola conexión, la que se realiza entre el Switch y el módem, permitiendo así evitar el problema de saturación que antes comentamos. Sumado al Switch se utiliza un Access Point (integrado) para permitir la conexión a través de Wifi en lugar de cableado, logrando obtener de esta manera un esquema como el de la figura 8.1

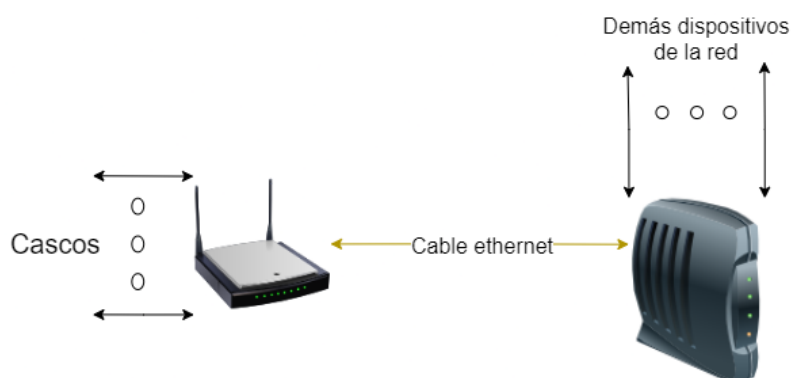


FIGURA 8.1: Esquema de conexionado del Switch + Access Point con la red de visores de realidad virtual y el módem Wifi.

Comunicación con la bicicleta Cada visor de realidad virtual se debe poder conectar a la bicicleta con la que se encuentra vinculada a la hora de ser utilizados.

Para lograr esto se deben considerar dos enfoques posibles: una comunicación cableada desde el casco hasta la bicicleta, o una comunicación inalámbrica. Considerando que la persona que utiliza

el producto (la bicicleta fija) se encuentra realizando actividad física, se cree que resultaría más conveniente recurrir a conexiones inalámbricas, siempre y cuando estas permitan obtener la latencia necesaria para el correcto funcionamiento del sistema.

Dentro de las opciones inalámbricas podemos diferenciar dos opciones diferentes: comunicación por Bluetooth o comunicación mediante Wifi. Ambas opciones presentan sus beneficios, pero teniendo en cuenta que la comunicación se realiza entre el casco y el dispositivo de comunicaciones instalado en la bicicleta, y, por otro lado, el visor debe también conectarse a través de Wifi con el Switch + Access Point para la comunicación con el servidor, se concluye que para que la comunicación entre casco-Switch y casco-bicicleta ocurran, no pueden ser ambas realizadas a partir del Wifi. Esto es debido a que si esto ocurriera de esa manera, el casco debería utilizar dos placas de red (que no posee) o enviar los datos a través del Módem a la bicicleta, lo cual añade complejidad innecesariamente. Por esta razón, la opción más apropiada es conectar el visor a la bicicleta a través de Bluetooth para permitir que, por otro lado, el visor se conecte por Wifi al Access Point + Switch.

De esta manera, se puede concluir que es importante que la comunicación entre la bicicleta y el visor se realice a través de una conexión Bluetooth, y la conexión del visor con el servidor a través de Wifi, mediante un Access Point y un Switch. Como se mencionó anteriormente, estos dos dispositivos se añaden a la red para reducir la cantidad de señales que debe manejar el Módem.

Un detalle a tener en cuenta es el hecho de que muchos de los visores de realidad virtual utilizan como medio de conexión BLE. Comparado con el Bluetooth clásico, BLE está diseñado para proporcionar, como su nombre lo indica, un bajo consumo de energía, manteniendo un rango de alcance de comunicación similar, por lo que se entiende que el uso de esta tecnología en el diseño no debería tener un impacto tan dramático en la duración de la batería del equipo.

Dispositivo del coach El coach en una sesión de ciclismo indoor es el profesor encargado de administrar la clase. Entre sus funciones se encuentran configurar la intensidad de las distintas etapas de la sesión, configurar la fuerza que se debe realizar para pedalear en la bicicleta y observar el estado de cada uno de ellos, verificando que no sufran descompensaciones, mareos o similares. Por estas razones, es un factor muy importante en las sesiones y es necesario tenerlo en cuenta para VRBike.

Funciones Las funciones mencionadas anteriormente deben ser posibles de realizar en una sesión de ciclismo indoor donde se utilice VRBike. Por ello, se plantea que el coach cuente con una interfaz para tal fin. Además, esta interfaz debe seguir permitiendo que realice su función de verificar el estado de salud de los participantes de la sesión, por lo cual se descarta que utilice un visor de realidad virtual. Así, se considera que debe contar con una interfaz especial. En la tabla 8.9 se detallan las funciones que debe proveer dicha interfaz.

Función	Descripción
Configuración de la sesión.	- Duración. - Selección del escenario.
Métricas	- Observar las métricas de los participantes en tiempo real.

TABLA 8.9: Funciones del coach.

En primer lugar, la configuración inicial de la sesión se puede realizar utilizando una simple interfaz de usuario que le permita seleccionar los distintos parámetros. Por otra parte, la observación de las métricas se trata de obtener una lista actualizada en tiempo real del nombre del participante y sus métricas, como velocidad actual, velocidad promedio y posición en el escenario. A su vez, esta observación se puede realizar con una interfaz de usuario en forma de tablero.

Interfaz del coach Una vez listadas las funciones necesarias, se determina con qué tipo de dispositivo cuenta el coach durante la sesión. Entre las alternativas se consideran dispositivos inmersivos, como un visor de realidad virtual, o dispositivos convencionales, como tablets y computadoras.

	Visor VR	Tablet	Computadora
Costo	≈ USD 400	≈ USD 300	≈ USD 400
Visualización de datos	3D con interacción	2D	2D
Facilidad de Uso	Muy Baja	Alta	Alta

TABLA 8.10: Comparación entre tipos de interfaces para el coach.

En la tabla 8.10 se contrastan las características a tener en cuenta. En primer lugar, el visor de realidad virtual únicamente se destaca en la visualización de datos, sin embargo, esto no resultaría óptimo para VRBike, donde las métricas de los participantes no requieren una visualización 3D. Por otro lado, como se mencionó, un visor no permite observar físicamente a los participantes de la sesión mientras se lo utiliza, algo fundamental en un gimnasio donde se realiza una actividad física intensa. Teniendo en cuenta estos factores, sumado a su costo y su baja facilidad de uso, se descarta dicha opción.

Luego, se analiza la opción correspondiente a una tablet, su facilidad de uso, y el hecho de que su limitación en la capacidad de cómputo no es de determinante interés para el tipo de aplicación requerida, hace que sea una opción atractiva, pero, por otro lado, al tener un servidor local, ya se debe utilizar una computadora para correr el servidor, por lo que se podría aprovechar esa misma computadora para correr el servidor y la app del coach en el mismo dispositivo.

Así, se concluye que la mejor opción es la correspondiente al uso de una computadora para la app del coach. Por otro lado, se desarrollan la aplicación para que, en caso de que el cliente deseara incorporar una tablet, pueda hacerlo minimizando las modificaciones necesarias para esto.

8.1.6. Métricas

Autenticación En la presente sección se evalúan distintas alternativas que permiten al usuario autenticarse en el sistema. Esto es necesario para poder registrar las métricas de la sesión con la cuenta del usuario que las obtuvo. También es importante para que los participantes puedan conocer los nombres de los presentes en la partida multijugador. En la tabla 8.11 se presentan las opciones tenidas en cuenta para la implementación de la autenticación.

			
Alternativa	Celular + QR	Llave de ingreso	Visor VR
Facilidad de uso	Media	Alta	Baja
Ventajas	<ul style="list-style-type: none"> - Independencia del gimnasio. - Popularidad de los códigos QR. - Facilidad de desarrollo 	<ul style="list-style-type: none"> - Facilidad para el usuario. 	<ul style="list-style-type: none"> - Facilidad de Desarrollo
Desventajas	<ul style="list-style-type: none"> - Necesidad del celular. 	<ul style="list-style-type: none"> - Depende del gimnasio. - Dificultad de desarrollo. 	<ul style="list-style-type: none"> - Dificultad para el usuario.

TABLA 8.11: Comparación entre opciones de autenticación de usuarios.

La primera opción analizada es utilizar las llaves de ingreso a los gimnasios como forma de autenticar al usuario en el sistema de VRBike. En la mayor parte de los casos, estas funcionan bajo la tecnología RFID y sería necesario incluir un lector RFID en las bicicletas VRBike. Al ser escaneadas, el sistema accede a un *tag* que varía en cada individuo. Si se implementara esta opción, sería necesario que cada usuario fuera poseedor de una llave única identificadora y concurra al gimnasio con ella, algo que no resulta atractivo. Se podría solucionar este inconveniente si se utilizaran las tarjetas de identificación de los gimnasios que poseen los usuarios, pero de esta manera sería necesario interactuar con las bases de datos del gimnasio para poder acceder a los datos asociados, tales como el nombre del participante. Como consecuencia, esto implica tener que adaptar el sistema de autenticación para el gimnasio en el que se implemente el producto. Por ello, se decide descartar esta opción debido a su complejidad y gran variabilidad según la tecnología y método de ingreso del gimnasio y se consideran otras alternativas que funcionen de forma más autónoma.

De esta manera, otra alternativa tenida en cuenta es el inicio de sesión mediante el visor de realidad virtual. Si bien esta opción es la más fácil en términos de facilidad de implementación y desarrollo, es la que más perjudica la experiencia de usuario debido a la complejidad que acarrea en los usuarios el uso de un sistema de inicio de sesión dentro del visor, algo con lo que los usuarios no están

familiarizados. Dado que se toma como prioridad que el sistema desarrollado sea fácil de usar, se descarta esta opción.

Por último, otra opción tenida en cuenta es el escaneo de un código QR con el celular del usuario. Bajo esta alternativa, los pasos necesarios para autenticar al usuario son los siguientes.

1. El usuario inicia sesión en la app de métricas de VRBike.
2. El usuario escanea el código QR único de cada bicicleta VRBike.
3. El servidor registra la vinculación usuario-bicicleta hasta que finalice la sesión.

Si bien el uso de códigos QR es elevado y se volvió aún más popular en los últimos años, existen casos en los que se tienen dificultades para utilizarlos, por ejemplo aquellas personas que poseen dispositivos móviles antiguos que no son capaces de reconocer con la cámara el código QR. Para poder integrar a estas personas, esta alternativa incluye la opción de ingreso de un código único de identificación escrito en la bicicleta (también visible en la bicicleta VRBike) que suplanta al QR al ser escrito por el usuario dentro de la aplicación en el campo correspondiente.

Cabe destacar que en los casos donde el usuario no cuente con su celular en el gimnasio, se debe permitir que participe de la sesión de spinning de todas formas. Para ello, se plantea que, si esto sucede, el usuario tenga la posibilidad de participar de la sesión bajo la opción de 'invitado', seleccionada automáticamente si el servidor no detecta una autenticación, esta función le permite al usuario utilizar el equipo sin haber conectado la bicicleta con su cuenta. Es importante remarcar que en este caso el usuario no podrá almacenar las métricas de la sesión en su cuenta.

Como conclusión, la alternativa por la que se opta es el escaneo de un código QR (o ingreso de un código escrito) único de cada bicicleta VRBike mediante la aplicación de métricas en el celular de cada usuario. La desventaja principal de esta opción es que el usuario requiere contar con su celular para poder autenticarse. Sin embargo, solamente sucede en pocas ocasiones que el individuo no cuente con él y, en dichos casos, no le impedirá participar de la sesión. Esta opción permite tener una abstracción sobre el gimnasio en el que se implemente el sistema y no necesita la adición de sensores en la bicicleta. En cuanto al *backend*, se debe asociar al código único de la bicicleta VR con un identificador del usuario en una base de datos mientras se realice la sesión.

Aplicación móvil Para la aplicación móvil en la que el usuario puede analizar sus métricas y compararlas con las del resto de ciclistas, consideramos algunas opciones posibles para su desarrollo. Estas se muestran a continuación en la siguiente tabla.


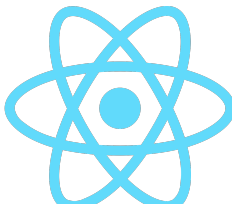
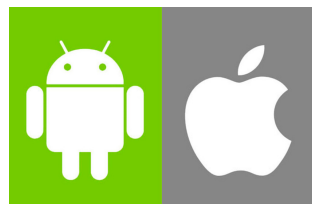
			
Alternativa	Flutter	React Native	Android/iOS nativo
Facilidad de Desarrollo	Alta	Media	Baja
Ventajas	<ul style="list-style-type: none"> - Desarrollo multiplataforma para iOS y Android - Fácil desarrollo - Permite desarrollo web 	<ul style="list-style-type: none"> - Desarrollo multiplataforma para iOS y Android 	<ul style="list-style-type: none"> - Acceso a servicios nativos del sistema. - Performance
Desventajas	<ul style="list-style-type: none"> - Emplea Dart como lenguaje de desarrollo. 	<ul style="list-style-type: none"> - Desarrollo web no integrado. 	<ul style="list-style-type: none"> - Dificultad de desarrollo - Desarrollo por separado para ambas plataformas. - Tiempo de desarrollo.

TABLA 8.12: Comparación entre opciones de *frameworks* para el desarrollo de la aplicación de celulares.

En la tabla anterior podemos ver que cada opción presenta sus ventajas y desventajas respecto a las demás. En el caso de desarrollo nativo para Android o para iOS tenemos que la facilidad de desarrollo es considerablemente baja y sumado a eso se necesita de dos equipos diferentes para el desarrollo de la aplicación, el equipo de desarrollo para iOS y el encargado de Android. De todas maneras permite utilizar servicios nativos del sistema, algo que podría ser útil en alguna aplicación.

React Native es otra posibilidad para el desarrollo de la aplicación, la facilidad de desarrollo y la posibilidad de desarrollo web son sus ventajas asociadas, pero el problema es que requiere desarrollo adicional a través de React para utilizar la aplicación como una página web.

Flutter es la última de las posibilidades que se consideran en este trabajo, su facilidad de desarrollo y la posibilidad de desarrollar aplicaciones tanto para iOS como para Android con el mismo equipo de desarrolladores la hace una opción muy atractiva, a esto se suma la posibilidad de desarrollo web, algo que puede ser útil. El lenguaje de desarrollo de Flutter es Dart, el cual es un lenguaje de desarrollo no tan común, lo cual podría en principio ser una desventaja, aunque es bastante sencillo y fácil de aprender y utilizar debido a la gran cantidad de recursos que se pueden encontrar en línea.

Considerando que la dificultad de desarrollo es alta y que no se necesita para esta aplicación utilizar las ventajas en cuanto a los servicios que provee, es que se decide despreciar la opción de desarrollo en Android y iOS nativo. Respecto a React Native se la considera una buena opción para el desarrollo, pero tiene el problema de que requiere de un desarrollo extra para pasar la aplicación desde la web hasta el celular mediante una *web app*.

Es por lo dicho que se decide por implementar la aplicación con Flutter como esquema de desarrollo, aprovechando las ventajas que provee y considerando que si bien utiliza Dart al ser fácil de utilizar, representa una desventaja asumible frente a las ventajas que esta opción trae consigo.

8.1.7. Regulador de fuerza

Una de las características más importantes del producto es la simulación de carga de pedaleo en función a las condiciones del terreno reflejado en la aplicación de realidad virtual, como por ejemplo el ángulo de la pendiente sobre la que se está circulando. Asimismo, el nivel de carga puede ser variado, por el entrenador, con el propósito de aumentar la exigencia del entrenamiento, asignando una determinada potencia a cada ciclista.

Cabe destacar que el sistema de simulación de carga es propio de la bicicleta que el cliente entrega para su adaptación, por ende este ensamble mecánico no es objeto de diseño. Sin embargo, se debe poder variar el ajuste de este módulo para poder adaptar la carga. De esta forma, se debe intervenir este sistema mecánico para brindar una interfaz que pueda ser accionada por el software, mediante el actuador correspondiente. El tipo de intervención sobre el sistema y el actuador a utilizar dependen del tipo de sistema de simulación de carga incluido en la bicicleta. En principio se diferencian dos casos.

Control de fuerza magnético. Este tipo de control de la carga aplicada sobre los pedales de la bicicleta se basa en un conjunto de imanes permanentes que pueden variar su posición respecto a una polea metálica, conectada mediante una correa a los pedales. Al girar esta polea, existe una velocidad relativa entre ella y el conjunto de imanes (que se encuentran en reposo). Cuando la polea se encuentra girando, se inducen en ella corrientes de Foucault por la variación del campo magnético, lo que resulta en una cupla de oposición que es reflejada en los pedales de la bicicleta. La intensidad de esta fuerza de resistencia es inversamente proporcional a la distancia entre los imanes y la polea.

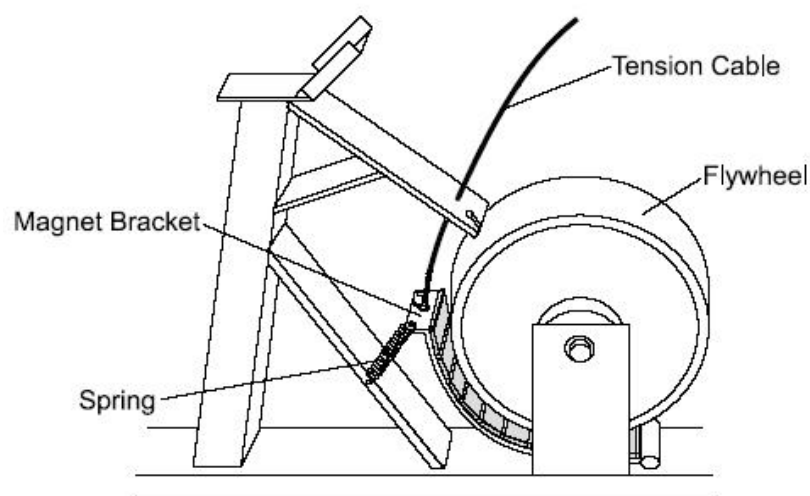


FIGURA 8.2: Esquema magnético de regulación de carga.

En la figura 8.2 se observa un esquemático del sistema, en donde se aprecia que la resistencia de pedaleo se regula mediante un cable de acero. Este cable normalmente se conecta a una rueda selectora, en donde el usuario puede seleccionar el nivel de resistencia adecuado. Para adaptar este sistema a la aplicación propuesta puede reemplazarse la rueda selectora por un motor paso a paso con un carrete en donde se fije el cable de acero mencionado anteriormente. Luego, al accionar el motor, se enrolla el cable, variando la distancia entre los imanes y la polea, y consecuentemente la carga presentada por la bicicleta. En principio, este motor debe montarse en las cercanías de la posición de la rueda selectora de carga. Como se mencionó anteriormente, esta característica depende específicamente de la marca y modelo de la bicicleta fija a adaptar. A su vez, el motor paso a paso es comandado por una placa controladora que a su vez recibe órdenes de la interfaz de comunicación local.

Se elige un motor paso a paso dado que este tipo de motores se caracteriza por su precisión en pequeñas variaciones. Para determinar las características eléctricas de este motor, se hace necesario medir el torque necesario para ajustar la carga, sobre la rueda de selección montada en la bicicleta. Nuevamente, esta característica es particular de cada marca y modelo de bicicleta.

Otra posible opción para controlar la carga es la de colocar un motor de corriente continua con un encoder rotativo en su mismo eje de giro, de forma tal de poder controlar el movimiento en forma precisa.

Otra alternativa de sistema de regulación de carga magnético es el mostrado en la figura 8.3. El principio de funcionamiento es el mismo que se explicó anteriormente, con la diferencia que en lugar de variar la distancia de un conjunto de imanes fijos a una polea metálica, se reemplaza el conjunto de imanes por un estator bobinado que se encuentra fijo en el espacio. Luego, variando la intensidad de corriente sobre el bobinado, se modifica la intensidad de campo magnético sobre la polea giratoria, aumentando la carga percibida por el ciclista. De esta forma, esto configura una interfaz eléctrica (en lugar de la mecánica), que dependerá de cómo fue diseñado este sistema por parte del fabricante de la bicicleta. Por ende, para controlar la carga, en este caso se emplea un generador de señales, que puede estar compuesto por un microcontrolador.

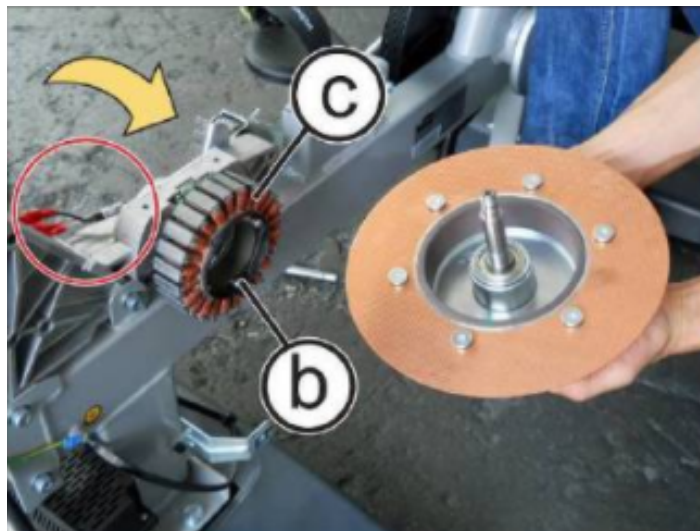


FIGURA 8.3: Esquema magnético de regulación de carga con interfaz eléctrica.

Control de fuerza por fricción. En esta clase de regulador de fuerza se controla la carga percibida por el ciclista al pedalear de forma mecánica, por fricción entre dos superficies. Se coloca una pieza de material con un alto coeficiente de fricción (denominada *zapata*) sobre el volante de inercia de la bicicleta. Esta zapata tiene la capacidad de poder bajar o subir, siendo ajustada por un tornillo accesible por el usuario. A medida que se ajusta el tornillo, la zapata presiona sobre la superficie lateral del volante de inercia, aumentando la carga de fricción sobre esta. De esta forma, se aumenta la resistencia percibida por el ciclista.



FIGURA 8.4: Esquema de regulación de carga por fricción.

En la figura 8.4 se muestra un esquema del sistema en cuestión. En este caso, la interfaz de regulación de carga es mecánica, siendo similar a la expuesta en la figura 8.2. Se puede modificar el tornillo que ajusta la presión acoplándole un motor paso a paso, o un motor de corriente continua con lazo de realimentación. De esta forma se controla en forma precisa el giro del tornillo, y por ende la carga aplicada. En este punto cabe destacar que el sistema se diseña con el objetivo de poder ser operado por una persona sin mayores esfuerzos, por lo que se considera que el torque

de accionamiento es bajo, y constante para todo nivel de carga. Sin embargo, al momento de seleccionar el motor a utilizar se debe medir el torque necesario para el ajuste con la herramienta correspondiente.

8.1.8. Resumen

A continuación se muestra una tabla que resume las opciones elegidas para cada una de las decisiones que se consideraron en el análisis de las alternativas de diseño planteadas.







Alternativa	Elección	Imagen
Sensor de velocidad	Sensor óptico TCRT5000	
Interfaz de Comunicación	ESP32	
Dispositivo de visualización	Oculus Quest 2	
Disposición del servidor	Servidor On Premise	
Autenticación con base de datos	Celular + QR	
Framework de desarrollo app	Flutter	

TABLA 8.13: Elecciones realizadas para cada análisis comparativo.

8.2. DFMEA

En esta sección se efectúa el análisis modal de fallas y efectos para el diseño en cuestión. Para ello, se realizaron sucesivas reuniones de corta duración para determinar las posibles fallas, efectos, causas y las acciones a tomar en cuenta. Se repitió este proceso para cada módulo, resultando en la información plasmada en las figuras 8.5 y 8.6. Para estimar el valor de RPN, se emplean las escalas mostradas en la tabla 8.14.

Valor	Severidad	Probabilidad	Detectabilidad
1	Insignificante	Remota	Completa
2	Poco significativo	Poco probable	Mayor
3	Moderado	Media	Moderada
4	Grave	Alta	Pequeña
5	Muy grave	Muy alta	Mínima

TABLA 8.14: Escala del DFMEA

Nivel de RPN (IC)	
Aceptable	$IC \leq 27$
Bajar hasta razonablemente práctico	$27 < IC \leq 48$
No aceptable	$IC > 48$

TABLA 8.15: Escala de RPN

N°	Módulo / Función	Fechas:		Original	22/10/2021		Revisión		1/6/2022		Aceptabilidad		
		Efectos de la falla	Modo de la falla		Causas de la falla	NS	PO	DE	IC	Acción	NS	PO	DE
1	Multijugador	Pérdida de inmersión.	Glitches en el movimiento de otros jugadores	Microcorte en conexión a servidor	5	3	2	30	Extrapolación del movimiento de los jugadores utilizando la velocidad y dirección	5	2	2	20
2	Multijugador	Pérdida de inmersión.	Glitches en el movimiento de otros jugadores	Ancho de banda de conexión insuficiente	5	1	4	20	Ninguna	5	1	4	20
3	Multijugador	Pérdida de inmersión.	Glitches en el movimiento de otros jugadores	Capacidad de servidor excedida	5	2	3	30	Escalamiento de recursos del servidor	5	1	3	15
4	Multijugador	El jugador no puede ver al resto de los participantes ni viceversa	Jugador fuera de la sesión multijugador	Reinicio accidental del casco antes de comenzar la sesión	4	2	4	32	Agregar "Número de usuarios conectados: X" a la interfaz del coach antes de comenzar la sesión	4	2	2	16
5	Multijugador	Interrupción en la sesión de entrenamiento.	Jugador fuera de la sesión multijugador	Pérdida de conexión del casco al servidor	5	2	3	30	Reubicar router más cerca de la bicicleta	5	1	3	15
6	Sensor de velocidad	Pérdida de inmersión.	Movimiento del ciclista no fluido	Velocidad de pedaleo baja. Delay en el sensado de velocidad.	4	2	3	24	Ninguna	4	2	2	16
7	Sensor de velocidad	Pérdida de inmersión	Desincronización físico-visual	Sensor corrido accidentalmente de posición	4	2	4	32	Volver a acomodar el módulo del sensor de velocidad a la bicicleta, acomodando el soporte o reemplazándolo	4	1	4	16
8	Sensor de velocidad	Pérdida de inmersión.	Movimiento del ciclista no fluido	Sensor de velocidad averiado por cortocircuito provocado por la aparición de sudor en sus contactos.	4	2	4	32	Reemplazar contactos por soldadura. Colocación de protección del módulo respecto del sudor.	4	1	4	16
9	Sensor de velocidad	Pérdida de inmersión.	Movimiento del ciclista no fluido	Sensor ubicado a mayor distancia de la detección	4	3	3	36	Reubicar sensor más cerca del volante de inercia	4	2	3	24
10	Regulador de Fuerza	Pérdida de inmersión.	Fuerza real no relacionada con entorno virtual	Motor descalibrado.	4	3	2	24	Ninguna	4	3	2	24
11	Regulador de Fuerza	Pérdida de inmersión	El ciclista percibe ruido cuando varía la carga	Juego excesivo en la varilla de ajuste de fuerza	5	3	2	30	Cambiar tuerca base de la varilla roscada por una con inserto plástico para evitar su desajuste. Agregar buje a la zona de la varilla que se inserta en el soporte superior	5	2	2	20

FIGURA 8.5: DFMEA (Parte 1).

N°	Módulo / Función	Fechas:		Original	22/10/2021		Revisión		1/6/2022		Aceptabilidad			
		Efectos de la falla	Modo de la falla		Causas de la falla	NS	PO	DE	IC	Acción	NS	PO	DE	IC
12	Regulador de Fuerza	Pérdida de inmersión. Dispositivo inutilizable para entrenamiento.	No se varía la Fuerza.	Motor inutilizado por sobrecorriente.	5	2	3	30	Agregar procedimiento de calibración del rango de movimiento del motor. Añadir fusible. y reemplazar motor	4	2	2	16	
13	Alimentación	Daños al usuario.	Electrocución de usuario	Falla a chasis de la bicicleta por parte de cable de alimentación	5	3	4	60	Reforzar aislación de cables de alimentación. Conexión a tierra de la bicicleta.	5	1	4	20	
14	Alimentación		El sistema no prende	Modulo de alimentación inutilizado por cortocircuito, causado por sudor del ciclista.	4	3	4	48	Colocar fuente en carcasa estanca. Cambiar de lugar la ubicación de la fuente y agregar un fusible.	4	2	2	16	
15	Alimentación	Dispositivo inutilizable.	El sistema no prende	Desconexión del cable de alimentación de la fuente.	5	2	2	20	Ninguna	5	2	2	20	
16	Visor VR	Abandono del sistema de VR	Apagado del visor	Batería insuficiente	5	3	2	30	Incorporar un cable USB en la bicicleta para cargar al visor entre clases.	5	2	2	20	
17	Modulo de vibración	Pérdida de inmersión.	Intensidad de vibración no satisface el umbral requerido	Señal de control errónea, módulo mal ubicado.	5	2	3	30	Calibrar ubicación del modulo de vibración mediante arreglo de sensores.	5	3	1	15	
18	Alimentación	Dispositivo inutilizable	El sistema no prende	La corriente de salida del convertidor buck es inferior a la mínima necesaria para funcionar en modo continuo	4	3	4	48	Incorporar LED de encendido para aumentar la corriente mínima de salida del convertidor	4	2	2	16	
19	Aplicación del usuario	Abandono de la aplicación	No se muestran las métricas del usuario	El usuario no cuenta con conexión a Internet	4	2	4	32	Guardar en caché las últimas métricas obtenidas y cargarlas en caso de no poder conectarse a la base de datos.	1	2	4	8	

FIGURA 8.6: DFMEA (Parte 2).

Para filtrar los valores de RPN y determinar las acciones a tomar, se adopta el criterio mostrado en la tabla 8.15.

9. Factibilidad Económica

9.1. Dimensionamiento del mercado y modelo de negocio

Según un estudio de la firma especializada en investigación de mercados IMARC Group, el ciclismo indoor representó un mercado valuado en 600 millones de dólares en el año 2020. Apoyado por la implantación de nuevas tecnologías, se espera que el mercado crezca un 5 % año a año, durante los próximos 6 años, alcanzando una valoración total de 800 millones de dólares en el año 2026.

Si se analiza el caso particular de Argentina, en el año 2017 era el quinto país del mundo con mayor cantidad de gimnasios (7900) a los que asisten más de 2,8 millones de usuarios cada año, con unos ingresos totales de mil millones de dólares. De los 2,8 millones de usuarios anteriores, cerca de 180 mil realizan clases de spinning anualmente en las más de 900 salas disponibles.

El modelo de negocio planteado trata de capturar parte de las personas que realizan spinning en los gimnasios. Para eso se propone un modelo en el que se vende la adaptación de las bicicletas, ajustándose así al modelo que cada gimnasio posea. Se cobra un monto fijo por cada bicicleta adaptada, y se vende de a salas. Además, hay una inversión inicial por parte del gimnasio correspondiente al sistema de Networking necesario para cada sesión de spinning. Por el precio que se le cobra a cada gimnasio se contempla también el costo de los servidores y soporte de software. No se descarta la implementación de un sistema de suscripción a futuro.

9.2. Análisis de competencia

En el mercado actual existen diferentes tipos de bicicletas fijas que dependiendo de sus características y prestaciones varían ampliamente en precio. En primer lugar, las bicicletas fijas más comunes que están presentes en la mayoría de los gimnasios de barrio tienen un valor de entre 600 y 700 dólares. Estas últimas no poseen ningún tipo de componente electrónico o consola y el frenado se hace mediante fricción.

En segundo lugar, uno de los modelos que algunos de los gimnasios de gama alta en Argentina poseen es el Life Fitness IC6, que tiene un valor de alrededor de 2600 dólares. Estas suponen una inversión considerable si se quiere llenar varias salas con ellas, pero también ofrecen una experiencia totalmente distinta a las bicicletas de gama baja, ya que cuentan con una consola que indica la velocidad, potencia, frecuencia cardíaca, y se comunica con la consola central del entrenador. Además, poseen frenos magnéticos y la calidad de los materiales y terminaciones es superior. La principal motivación de compra de estas bicicletas por parte de los gimnasios es la de poder brindar una experiencia diferenciada en clases de ciclismo indoor, atrayendo más clientes y fidelizándolos a ese gimnasio en particular.

Por último, existen bicicletas inteligentes como la TechnoGym Ride, que por un valor que ronda los 7000 dólares ofrecen una experiencia completa e integrada, con diversos tipos de sensores, partes ajustables y una consola interactiva que permite seguir varias métricas relevantes para el entrenamiento. Este tipo de bicicletas no existen actualmente en los gimnasios argentinos, debido a que no es económicamente viable armar salas con ellas.

Respecto al mercado de bicicletas fijas en el segmento hogareño, existen soluciones como VirZOOM o Wahoo Fitness, que apuntan a transformar una bicicleta tradicional en una fija para el uso en el hogar. Asimismo, aplicaciones como Peloton complementan el entrenamiento de ciclismo con diversas clases grabadas, accesibles por el usuario.

Desde el punto de vista de software, se pueden destacar algunas empresas que brindan una experiencia de realidad virtual en cuanto a diversos entrenamientos, como lo son Zwift, BKOOL o Holofit. Estas aplicaciones en general se compatibilizan con hardware específico, correspondiente a las bicicletas fijas de las marcas más reconocidas. En general, todas las empresas referentes a este rubro adoptan un modelo de negocios por suscripción, en donde el usuario abona una cuota mensual y a cambio obtiene acceso ilimitado a las plataformas de entrenamiento en VR.

Habiendo resumido las áreas y empresas competidoras del desarrollo propio, se concluye que no existe al día de hoy una empresa que brinde una solución cerrada al usuario, sino que aquella persona que pretenda experimentar el entrenamiento en realidad virtual debe adquirir el software y la bicicleta en forma separada. En el caso del proyecto en cuestión, la diferenciación respecto de la competencia radica en que se amplía notablemente el espectro de bicicletas aptas para entrenamiento VR, adaptando bicicletas fijas que no fueron concebidas a tal fin. De esta forma, se generaliza el acceso a este tipo de entrenamiento.

9.3. Estimación de costos

En la tabla 9.1 se muestra un resumen de la inversión inicial que debe hacerse para llevar a cabo el proyecto.

Elemento	Costo
Motor TB vel/carga (400W)	USD 200,00
Osciloscopio + puntas	USD 300,00
Multímetro industrial x1	USD 10,00
Herramientas	USD 100,00
Sensores TB vibración x4	USD 8,00
Impresora 3D	USD 300,00
Mesas/Escritorios x2	USD 140,00
Mesa de trabajo	USD 45,00
PC	USD 1.000,00
Estantería metálica x2	USD 85,00
Cafetera	USD 15,00
Set de mates	USD 30,00
Oculus (x2)	USD 800,00
Desarrollo	USD 8.000,00
Total inversión inicial	USD 11.233,00

TABLA 9.1: Inversión inicial

En la tabla anterior se tienen en cuenta todos los costos en materia de hardware y herramientas y además se considera el costo de las horas de desarrollo iniciales necesarias para terminar los elementos de software.

A continuación se muestra una tabla detallando los costos de fabricación para cada bicicleta. En la categoría "otros", se tienen en cuenta componentes electrónicos varios y otros gastos menores.

Bloque	Componente	Costo unitario	Cantidad	Costo total
Visor	Oculus Quest 2	USD 400,00	1	USD 400,00
Sensores	TCRT5000	USD 0,45	1	USD 0,45
Motor de vibración	Motor 1120	USD 1,14	1	USD 1,14
Control de carga	Motor paso a paso Nema 17	USD 7,72	1	USD 7,72
Comunicación	NodeMCU-ESP32	USD 8,50	1	USD 8,50
Alimentación	Fuente switching 12V 30W	USD 3,60	1	USD 3,60
Componentes varios	Soportes para componentes	USD 10,00	1	USD 10,00
	Otros	USD 20,00	1	USD 20,00
Total costos de fabricación				USD 451,41

TABLA 9.2: Costos de adaptación para una bicicleta

Observando la tabla 9.2, se hace evidente que el gasto principal es en el visor Oculus Quest 2, el cual hace cuatro quintas partes del costo total. Por esta razón, una posibilidad para proyectar el modelo de negocios a futuro y considerando que los costos de los visores de realidad virtual

disminuirán, es encargar la adquisición de dichos visores a cada cadena de gimnasios. Esto es algo que no se considera en un principio, ya que se busca facilitar el ingreso al mercado brindando una solución cerrada para cada cliente. Por último, algo que no se está considerando es el costo de la bicicleta, ya que los gimnasios proveerán de sus modelos para la adaptación.

9.4. Flujo de fondos

Para la realización del flujo de fondos del proyecto se considera un precio de venta de 1210 dólares por unidad, lo que supone un margen superior del 63 % por bicicleta y posiciona al producto como una opción de *smart bike* mucho más asequible que las bicicletas presentes en algunos gimnasios *prémium* en Argentina.

Interes	9 %					
Año	0	1	2	3	4	5
Bicicletas Vendidas	0	45	75	105	195	270
Cantidad de Salas	0	3	5	7	13	18
Ingresos	\$ -	\$ 37.562	\$ 62.603	\$ 87.645	\$ 162.769	\$ 225.372
Gastos	\$ -	\$ 31.864	\$ 50.789	\$ 84.913	\$ 138.687	\$ 191.999
Ingresos Brutos	\$ -	\$ 2.629	\$ 4.382	\$ 6.135	\$ 11.394	\$ 15.776
Costos Fijos	\$ -	\$ 3.477	\$ 3.477	\$ 3.477	\$ 3.477	\$ 3.477
Costos Variables	\$ -	\$ 20.658	\$ 34.429	\$ 56.601	\$ 105.117	\$ 145.546
Sueldos	\$ -	\$ 5.100	\$ 8.500	\$ 18.700	\$ 18.700	\$ 27.200
EBITDA	\$ -	\$ 5.698	\$ 11.815	\$ 2.731	\$ 24.081	\$ 33.373
IIGG	\$ -	\$ 1.994	\$ 4.135	\$ 956	\$ 8.428	\$ 11.680
Resultado Neto	\$ -	\$ 3.704	\$ 7.679	\$ 1.775	\$ 15.653	\$ 21.692
Inversión Inicial	\$ 9.283			\$ 1.000		\$ 2.000
IVA Ventas	\$ -	\$ 7.888	\$ 13.147	\$ 18.405	\$ 34.181	\$ 47.328
IVA Compras	\$ 1.950	\$ 5.068	\$ 7.960	\$ 12.826	\$ 22.805	\$ 31.715
IVA Neto	-\$ 1.950	\$ 2.820	\$ 5.186	\$ 5.579	\$ 11.377	\$ 15.613
Flujo de Caja	-\$ 11.233	\$ 5.653	\$ 7.679	\$ 775	\$ 15.653	\$ 19.692
Flujo Descontado	-\$ 11.233	\$ 5.186	\$ 6.464	\$ 599	\$ 11.089	\$ 12.799
Flujo Desc Acum.	-\$ 11.233	-\$ 6.047	\$ 417	\$ 1.016	\$ 12.104	\$ 24.903

VAN	\$ 24.903
TIR	36 %

TABLA 9.3: Flujo de fondos

En la tabla 9.3 se observa el flujo de fondos calculado para el proyecto. En principio, se limita el horizonte de proyecto en cinco años, de forma tal de poder predecir con mayor precisión los movimientos económicos. Para el cálculo del volumen de ventas, se fija como mercado objetivo aquellas cadenas de gimnasios de alto volumen de sucursales (como por ejemplo Megatlon, SportClub, OnFit, etc.), dado que se deduce que son las que mayor poder adquisitivo tendrán al momento

de realizar la erogación de capital para obtener la experiencia de realidad virtual. Asimismo, se considera que cada operación de venta se realiza por salas de ciclismo, estimando el tamaño de cada sala en quince bicicletas (valor surgido de investigación de mercado).

Cabe aclarar que a medida que se aumenta el volumen de producción, se tiene en cuenta la contratación de una mayor cantidad de empleados de forma tal de poder brindar el adecuado soporte al crecimiento en las ventas. En la tabla 9.4 se plasma el salario neto anual de cada uno de los empleados en función de su rango (para considerar cargas impositivas, aportes y cobertura médica, se toma que el salario bruto equivale a 1,7 veces el salario neto), y la progresión de los recursos humanos en la empresa con su respectiva carga económica, a través de los años.

	Sueldo Anual (USD)	Año 1	Año 2	Año 3	Año 4	Año 5
Técnicos	3.000	1	1	2	2	3
Administrativo	2.000		1	1	1	2
Developer	3.000			1	1	1
Total (USD)		3.000	5.000	11.000	11.000	16.000

TABLA 9.4: Evolución de empleados

Se calculan dos indicadores económicos para el proyecto. Por un lado, se computa el valor actual neto del proyecto (VAN), llegando a un valor de USD 24.903 al quinto año de ejercicio. Además, se calcula la tasa interna de retorno (o TIR), que indica un aproximado de la eficiencia del flujo de fondos en cuestión, resultando en 36 %. Para el cálculo de esta tasa se emplea una TREMA de 9 % en dólares.

10. Factibilidad de tiempos

A continuación se presentan las actividades que comprenden la realización del proyecto VRBike en 189 días, lo cual implica su finalización para el 31 de mayo de 2022, junto con la duración estimada para cada actividad, sus predecesoras y sus respectivas fechas de inicio y finalización.

ID	Tarea	Duración	Predecesora	Comienzo	Fin
1	VRBike	189		23/11/2021	31/5/2022
2	Bicicleta	55		23/11/2021	17/1/2022
3	Gestion de la bicicleta	55		23/11/2021	17/1/2022
4	Control de Fuerza	50		18/1/2022	9/3/2022
5	Identificación de la interfaz	8	3	18/1/2022	26/1/2022
6	Adaptación de la Interfaz	27	5	27/1/2022	23/2/2022
7	Instalación	8	6	24/2/2022	4/3/2022
8	Validación	2	7	7/3/2022	9/3/2022
9	Vibración	8		18/1/2022	26/1/2022
10	Determinación del umbral	2	3	18/1/2022	20/1/2022
11	Instalación	3	10	21/1/2022	24/1/2022
12	Validación	1	11	25/1/2022	26/1/2022
13	Velocidad	9		18/1/2022	27/1/2022
14	Drivers	2	3	18/1/2022	20/1/2022
15	Instalación	4	14	21/1/2022	25/1/2022
16	Validación	1	15	26/1/2022	27/1/2022
17	ESP	32		24/2/2022	28/3/2022
18	Instalación y cableado	8	8;12;16	10/3/2022	18/3/2022
19	Desarrollo intefaz HID	20	6;10;14	24/2/2022	16/3/2022
20	Validación de interfaces	7	18;19	21/3/2022	28/3/2022
21	BackEnd	64		23/11/2021	26/1/2022
22	Contratación de servicio de nube	2		23/11/2021	25/11/2021
23	Creación de intefaz front-back métricas	10	22	26/11/2021	6/12/2021
24	Creación de interfaz front-back multijugador	10	22	26/11/2021	6/12/2021
25	Desarrollo backend multijugador	42	24	7/12/2021	18/1/2022
26	Validación	7	25	19/1/2022	26/1/2022
27	Desarrollo de backend métricas	15	23	7/12/2021	22/12/2021
28	Validación	19	27	23/12/2021	11/1/2022
29	FrontEnd	104		7/12/2021	21/3/2022
30	Desarrollo de app de métricas	10	23	7/12/2021	17/12/2021
31	Desarrollo de app de coach	10	23;24	7/12/2021	17/12/2021
32	Desarrollo de app de VR	104	23;24	7/12/2021	21/3/2022
33	Networking	41		22/3/2022	2/5/2022
34	Integración entre frontend y backend	20	26;28;30; 31;32	22/3/2022	11/4/2022
35	Validación de integración	20	34	12/4/2022	2/5/2022
36	VR headset	146		23/11/2021	18/4/2022
37	Importación	51		23/11/2021	13/1/2022
38	Validación de potencia gráfica	6	32;37	22/3/2022	28/3/2022
39	Conexión con sensores	20	20;37	29/3/2022	18/4/2022
40	Integración	28		3/5/2022	31/5/2022
41	Integración app y sensores en el headset	13	35;38;39	3/5/2022	16/5/2022
42	Validación final	14	41	17/5/2022	31/5/2022

TABLA 10.1: Planificación de tareas

10.1. Programación. Diagrama de Gantt

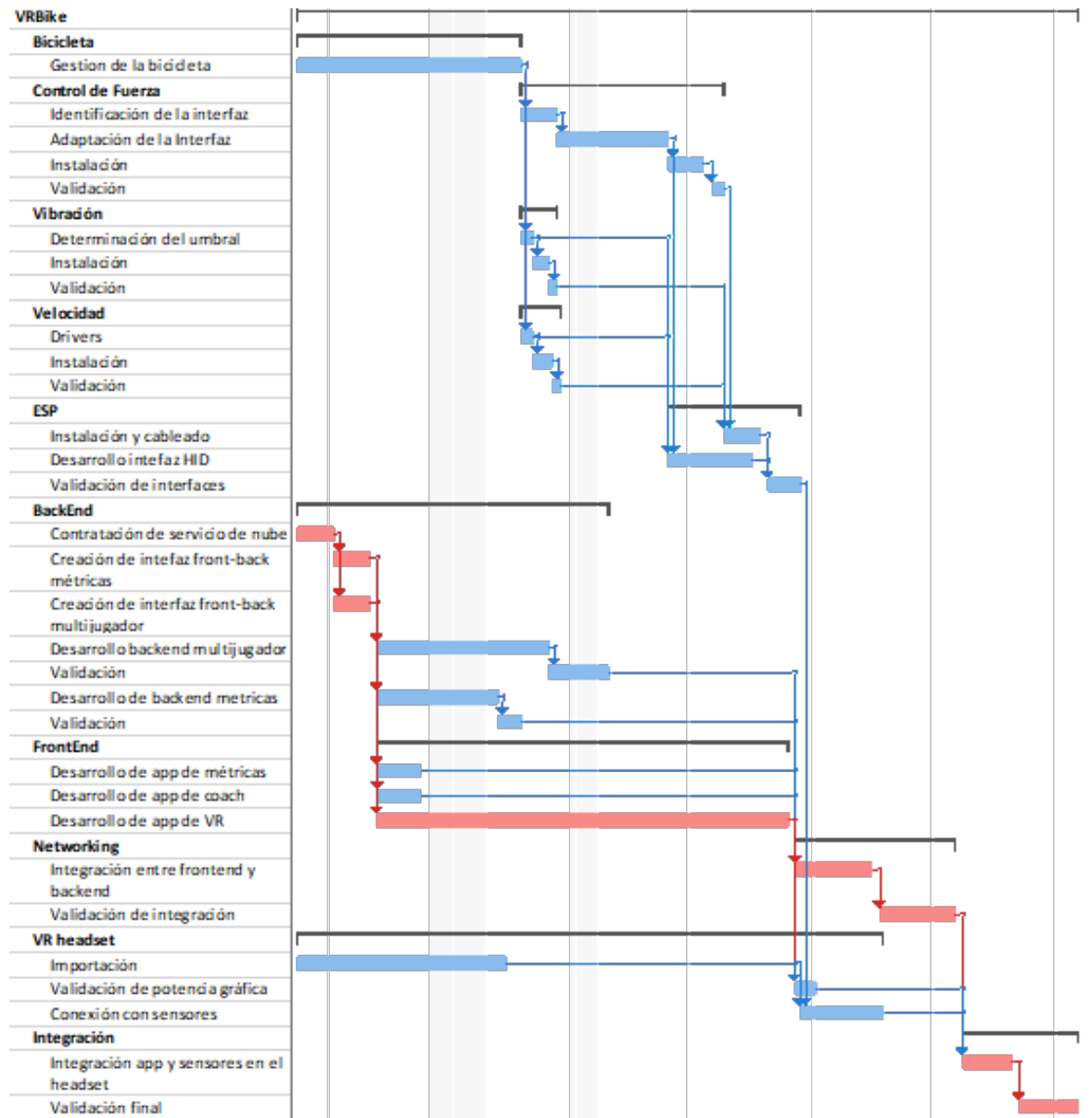


FIGURA 10.1: Diagrama de Gantt

Una vez definidas las actividades del proyecto fue posible realizar el diagrama de Gantt del mismo, expuesto en la figura 10.1, en el cual se representan las actividades del camino crítico en color rojo. Las mismas son principalmente del apartado de software, como puede ser las interfases entre *frontend* y *backend* de las aplicaciones a desarrollar, seguido del desarrollo de la aplicación de realidad virtual, la cual posteriormente requiere de la integración y validación del sistema de red, para finalmente concluir el proyecto con la integración de la aplicación y los sensores de la bicicleta física.

11. Factibilidad legal y responsabilidad civil

En este apartado se expone lo vinculado a la relación entre el producto objeto de este diseño y las leyes vigentes en Argentina.

11.1. Tipo societario

Para la fabricación, comercialización y soporte del producto, se busca establecer una sociedad en términos legales. Esta se define como la organización de una o más personas que realicen aportes para aplicarlos a la producción (o intercambio) de bienes y servicios, soportando tanto las ganancias como las pérdidas que se pudieran ocasionar. Los tipos de sociedad posibles se encuentran estipulados en el artículo 1° de la Ley General de Sociedades N° 19.550.

El objetivo del establecimiento de una sociedad es el de imponer una separación patrimonial entre el patrimonio de la persona jurídica y el patrimonio particular de los constituyentes de la misma sociedad. El tipo de sociedad se define en función de ciertos parámetros propios de la empresa, como son por ejemplo el capital disponible y el número de personas que integran la sociedad, entre otras. En este caso se acota el análisis a tres tipos de sociedades comerciales, que se exponen a continuación.

- **Sociedad de hecho (SH).** Esta sociedad se caracteriza por no tener un instrumento constitutivo. Esto brinda una mayor facilidad de constitución y flexibilidad, pero también carga con igual responsabilidad para cada uno de sus socios. De esta forma, este tipo societario es recomendable en actividades con poco caudal de facturación, o en aquellas que se involucre un bajo riesgo. A diferencia de los tipos subsiguientes, cada integrante debe responder con su propio patrimonio.
- **Sociedad anónima (SA).** Este tipo de sociedad se vincula a un gran volumen de capital y un alto número de socios. En general, es necesario constituir una SA cuando se trabaja con negocios complejos y su constitución suele incluir un alto costo. Una sociedad anónima se constituye a través de un estatuto, y su capital social se divide en acciones que, en función de su tipo, regulan el grado de participación de cada accionista en las decisiones de la empresa. Asimismo, existen tres órganos fundamentales en una sociedad anónima: la asamblea de accionistas (órgano de gobierno), el directorio (órgano de administración) y el consejo de vigilancia (órgano de fiscalización).
- **Sociedad de responsabilidad limitada (SRL).** Una SRL es constituida a través de un contrato social, al igual que en el caso de las sociedades anónimas. Respecto de las SA, el esquema de funcionamiento es distinto: se toman decisiones en reuniones entre los socios, quedando registradas y asentadas en un libro de actas. Luego, los gerentes de la sociedad deben ejecutar estas decisiones previamente documentadas. La constitución de una SRL insume un menor costo respecto de las sociedades anónimas. Luego, es recomendable este tipo de sociedad siempre que haya pocos socios con un aporte de capital bajo.

Las últimas dos sociedades deben ser inscriptas ante el Registro Público de Comercio. Luego, deben ser inscriptas en la Administración Federal de Ingresos Públicos (AFIP) y ante la municipalidad donde esté declarado el domicilio de giro comercial.

Habiendo expuesto las opciones de sociedad, se toma la decisión de constituir una sociedad de responsabilidad limitada. La principal ventaja es brindar la separación patrimonial respecto del patrimonio de los socios, con un costo bajo de constitución. Asimismo, en principio la empresa dispondrá de un número reducido de socios (cinco).

11.2. Licencia de software

Dado que el factor preponderante del producto es el entorno de realidad virtual, se deben tener en cuenta las legislaciones vigentes en cuanto a la comercialización de software. Respecto de esto, se vincula a la distribución de software con derechos de autor. En este caso, se otorga el código objeto de la aplicación (es decir, la aplicación compilada) al usuario final mediante su descarga de la tienda de aplicaciones correspondiente a la plataforma. Al descargarla, el usuario se compromete a usarla tal como se la fue entregada, sin mediar ningún tipo de modificación sobre esta. Luego, se aplica una licencia del tipo comercial, dado que el principal objetivo en la distribución del software es el de lograr un lucro económico.

11.3. Garantía y soporte

La garantía de un producto hace referencia a que el vendedor debe responder ante fallos de funcionamiento en un período de tiempo determinado, siempre y cuando estos fallos no sean responsabilidad estricta del usuario (por ejemplo, usando al producto de una forma no adecuada o no especificada por el fabricante). Este aspecto es tratado en la Ley de Nacional de Defensa del Consumidor (Ley N° 22.240) en su artículo undécimo, citado a continuación.

"La garantía legal tendrá vigencia por tres meses cuando se trate de bienes muebles usados y por seis meses en los demás casos a partir de la entrega, pudiendo las partes convenir un plazo mayor. En caso de que la cosa deba trasladarse a fábrica o taller habilitado, el transporte será realizado por el responsable de la garantía y serán a su cargo los gastos de flete y seguros y cualquier otro que deba realizarse para la ejecución del mismo"

Dado que se hace referencia a *bienes muebles*, entonces se deberá establecer un plazo de garantía mínimo de seis meses para aquellas partes de hardware que componen la adaptación de la bicicleta fija (como lo son sensores, dispositivos de cómputo, dispositivos de visualización, etc.). Por otro lado, se destaca que el software se comercializa con un modelo de negocios de suscripción. Esto implica que el usuario no solo contrata los servicios de la aplicación (es decir, su código objeto y el derecho a la instalación y uso), sino que también se le otorga el derecho a soporte técnico. Esto resulta en que se debe mantener correctamente la aplicación, suministrando un servicio de actualizaciones que sean lo suficientemente periódicas como para solucionar aquellos fallos de software que sean o no detectados por los usuarios. Asimismo, se debe mantener actualizado el

hardware relativo a los servicios en la nube, escalando los recursos contratados de forma tal de mantener invariante la experiencia de usuario frente al aumento de demanda de recursos originado por la conexión de una mayor cantidad de usuarios.

11.4. Banco de datos

Dentro de las capacidades del producto se encuentra el disponer de métricas de entrenamiento personalizadas, en tiempo real. Además, estas son almacenadas en una base de datos remota para luego poder ser consultadas por el ciclista mediante una aplicación móvil cuando lo desee. De lo anterior se desprende que es fundamental el rol de la base de datos en la propuesta del producto. Dado que los datos almacenados son propios de cada persona (y se encuentran asociados a esta por algún tipo de identificador), luego aplica la Ley N°25.326 de Protección de Datos Personales. El espíritu de esta ley se resume con el siguiente fragmento extraído de su artículo primero.

"La presente ley tiene por objeto la protección integral de los datos personales asentados en archivos, registros, bancos de datos, u otros medios técnicos de tratamiento de datos, sean estos públicos, o privados, destinados a dar informes, para garantizar el derecho al honor y a la intimidad de las personas, así como también el acceso a la información que sobre las mismas se registre, de conformidad a lo establecido en el artículo 43, párrafo tercero de la Constitución Nacional."

Además, se introduce el concepto de *habeas data*, mencionado en el artículo cuadragésimo tercero de la Constitución Nacional, citado a continuación.

"Toda persona podrá interponer esta acción para tomar conocimiento de los datos a ella referidos y de su finalidad, que consten en registros o bancos de datos públicos, o los privados destinados a proveer informes, y en caso de falsedad o discriminación, para exigir la supresión, rectificación, confidencialidad o actualización de aquellos. No podrá afectarse el secreto de las fuentes de información periodística."

De lo anteriormente expuesto se concluye que todo dato personal almacenado en bases de datos referentes al funcionamiento de la aplicación de realidad virtual no puede ser en ningún caso usado para otro objetivo que no fuere el propósito para el que fue recolectado. Asimismo, mediante el recurso de *habeas data* cualquier usuario puede exigir las acciones necesarias para tomar conocimiento de sus datos almacenados en el servidor.

11.5. Modificación de las bicicletas

Por lo expuesto en el presente, se deduce que se necesita modificar la bicicleta provista por el cliente. Esta modificación puede incluir intervenir un sistema, cambiando su diseño respecto al pensado por el fabricante. Esto se traduce en una pérdida de la garantía de la bicicleta fija, si esta estuviera todavía dentro del plazo, por lo que se recomienda que la bicicleta a adaptar esté fuera de este plazo. De acuerdo a las conversaciones mantenidas con los principales dirigentes de los gimnasios, esto no debería ser un problema.

12. Ingeniería de Detalle

En esta sección se realiza un análisis detallado de cada subsistema. Se lo divide entre aquellos aspectos vinculados al hardware y los relativos al software. Se parte del siguiente diagrama en bloques.

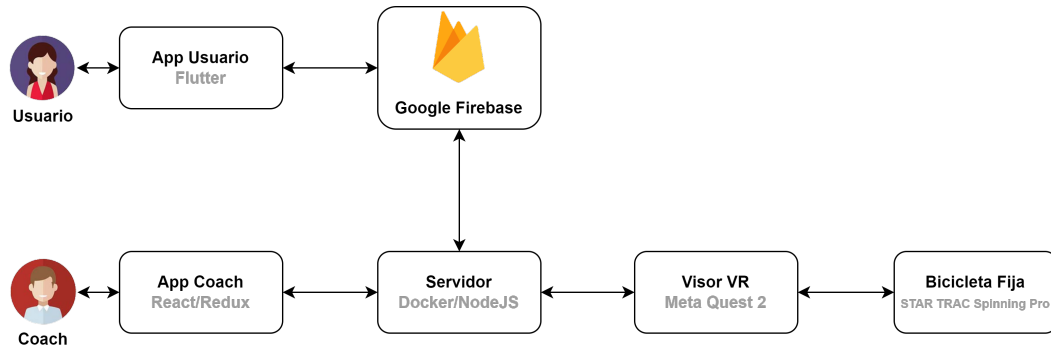


FIGURA 12.1: Diagrama en bloques general de la solución

En los sucesivos apartados se detallan los componentes y las funciones de cada uno de los bloques de 12.1.

12.1. Hardware

El área de hardware hace foco en la interfaz electromecánica que controla o adquiere las variables físicas en relación con la bicicleta. Por ende, es el principal nexo entre la experiencia virtual que percibe el ciclista, y el mundo "real" o físico. Un esquema detallado del hardware de la solución se puede apreciar en la figura.

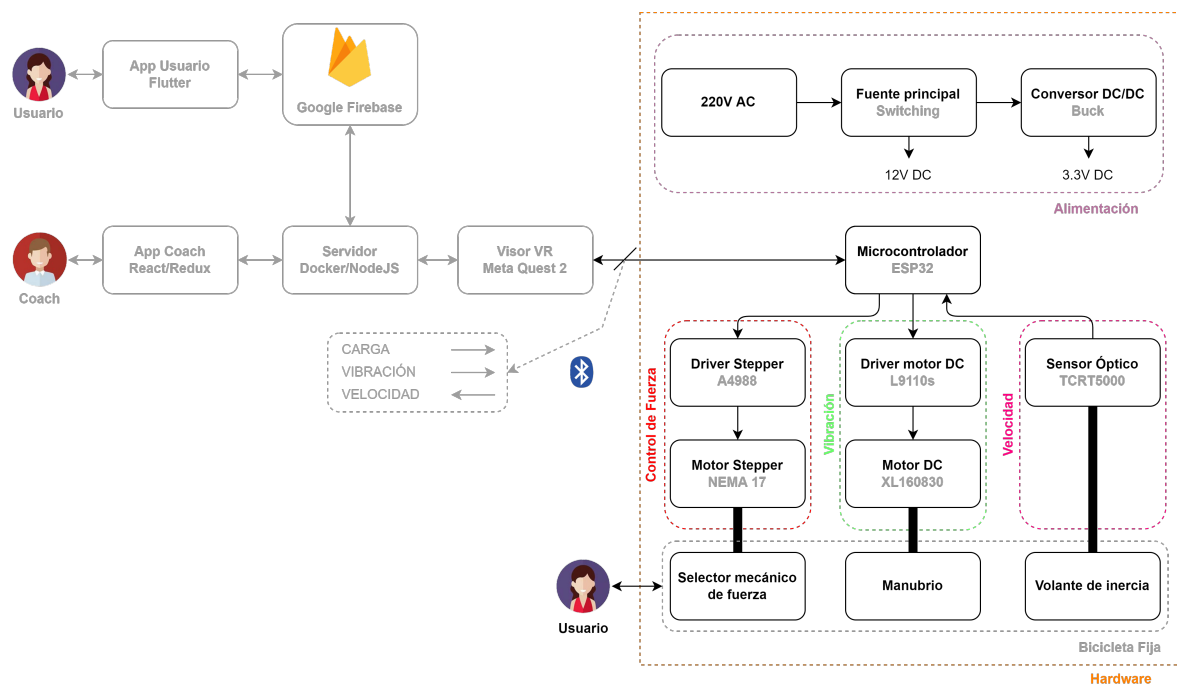


FIGURA 12.2: Diagrama en bloques general del área de software de la solución

Se pueden diferenciar seis submódulos, de acuerdo a sus funcionalidades.

- Bicicleta fija
- Microcontrolador
- Regulación de fuerza
- Sensor de velocidad
- Vibración
- Alimentación

En lo sucesivo, se profundiza acerca de cada uno de estos.

12.1.1.1. Bicicleta fija

La bicicleta fija es el eje principal del entrenamiento, ya que es el elemento sobre el cual se hace el ejercicio. En lo que respecta a la solución propuesta, se puede identificar a este elemento como un bloque en el cual tenemos tres señales características.

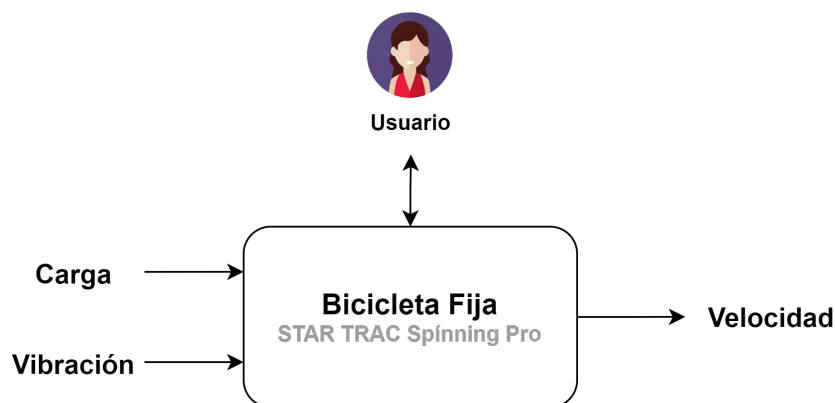


FIGURA 12.3: Bicicleta fija como bloque

Las tres señales que se identifican en la figura 12.2 son variables físicas que están vinculadas con la acción y la percepción del usuario ciclista. Para poder interactuar con estas variables físicas de manera electrónica, es necesario establecer un esquema de sensores y actuadores cuya instalación y características particulares dependen del modelo de bicicleta específico que se esté adaptando. Es así como se hace necesario fijar un modelo de bicicleta fija para poder realizar la adaptación, buscando que sea altamente genérica y representativa del parque de bicicletas fijas de gimnasios, de forma tal de poder extrapolar esta adaptación particular al resto de los modelos con la menor cantidad de modificaciones posibles, lo que deriva en un bajo costo.

En concordancia con lo anteriormente expuesto, la bicicleta objeto de adaptación del presente diseño es una STAR TRAC SPINNER PRO, provista por la cadena de gimnasios Megatlon.



FIGURA 12.4: STAR TRAC SPINNING PRO

En la figura 12.4 se observa la bicicleta en cuestión, destacando los siguientes aspectos, que influyen en la interfaz electromecánica referida con anterioridad:

- **Volante de inercia descubierto.** Esta característica permite estimar la velocidad de movimiento de la bicicleta a través de la medición de la velocidad del volante de inercia.
- **Interfaz de fuerza.** En el caso particular de esta bicicleta, el usuario controla la fuerza girando una perilla que ejerce presión sobre una zapata, que por rozamiento con el perímetro del volante de inercia causa una mayor resistencia.

Luego los principales focos de intervención son:

- **Volante de inercia**, en donde se mide la velocidad.
- **Perilla de fuerza**, desde donde se controla la fuerza percibida por el ciclista.
- **Manubrio**, en donde se ubica el módulo de vibración.

12.1.2. Microcontrolador

El microcontrolador elegido para esta solución es el ESP32. Dicho microcontrolador es el encargado de recibir e interpretar las mediciones del sensor de velocidad montado en la bicicleta que llegan como interrupciones a dicho microcontrolador, a su vez se encarga de entregar al driver del motor stepper la información necesaria sobre el ángulo de giro que se desea realizar sobre el motor, dicha información la envía haciendo uso de uno de los pines GPIO en la placa, y también se encarga de controlar las señales que van al driver para el control de los motores de vibración mediante una salida PWM que permite el control de la vibración con el circuito full bridge que constituye el driver del sistema de vibración.

Además de actuar como interfaz para los demás subsistemas que se encuentran montados en la bicicleta, se utiliza también la funcionalidad Bluetooth del ESP32 para la comunicación del sistema montado en la bicicleta con el visor de realidad virtual. Para esto se hace uso del protocolo HID sobre BLE, el cual es ampliamente utilizado en los controles de videojuegos. De esta manera se definen todos los parámetros que corresponden a la configuración de dicho controlador para luego poder utilizar al mismo para comunicar los datos obtenidos en la bicicleta al visor de realidad virtual, que luego se encarga de comunicar los datos a los demás subsistemas del equipo como se podrá ver más en detalle en el apartado que habla sobre el networking.

USAGE_PAGE(1)	#Desktop
USAGE(1)	#Gamepad
Collection(1)	#Application
Collection(1)	#Physical
Report ID(1)	#Gamepad
USAGE_ID(3)	#Vendor defined
LOGICAL_MINIMUM (0)	
LOGICAL_MAXIMUM (65535)	
REPORT_SIZE (16)	
REPORT_COUNT (1)	#Speed
INPUT (Data, Var, Abs)	
USAGE_ID(3)	#Vendor defined
LOGICAL_MINIMUM (0)	
LOGICAL_MAXIMUM (65535)	
REPORT_SIZE (16)	
REPORT_COUNT (2)	#Stepper & Vibration
OUTPUT (Data,Var,Abs)	
END_COLLECTION(0)	
END_COLLECTION(0)	

TABLA 12.1: Descriptor de contenido HID

12.1.3. Regulación de fuerza

La regulación de la carga de la bicicleta es un factor importante en el entrenamiento de ciclismo indoor con realidad virtual, no solo porque brinda inmersión al usuario, sino porque también suele ajustarse manualmente en reiteradas ocasiones durante las sesiones de spinning no inmersivas, de modo tal de modificar la potencia entregada por el ciclista a velocidades constantes. Es así como es imprescindible contar con este ajuste dentro de la solución propuesta.

Como se expuso en la primera parte del presente informe, existen en el mercado una gran cantidad modelos de bicicletas con diversos sistemas de ajuste de carga o fuerza. En particular, la bicicleta STAR TRAC SPINNING PRO mostrada en la figura 12.4 posee un sistema de frenado mecánico, por fricción. La intensidad del ajuste se regula mediante una perilla que gira solidaria a una varilla

roscada, que a su vez hace presión sobre la zapata mecánica que se encuentra debajo (tal como se detalla en la sección 8.1.7) y como se puede ver en la figura 12.5.



FIGURA 12.5: Perilla de regulación de fuerza de forma manual

Es por esto que, con la finalidad de intervenir lo menos posible la bicicleta, se propone un sistema mecánico compuesto por un motor paso a paso (también conocido con el nombre *stepper*) y por un sistema de engranajes con el objetivo de acoplar el motor con la varilla selectora de fuerza, un diseño 3D de este sistema se puede apreciar en la figura 12.6

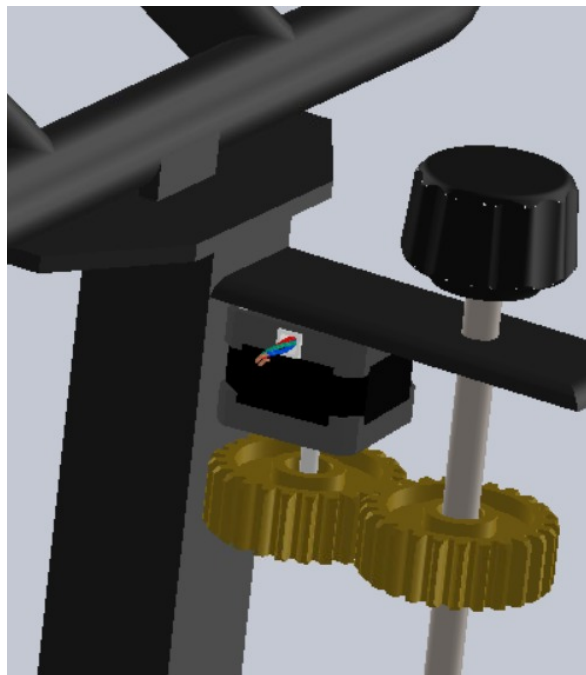


FIGURA 12.6: Control de fuerza propuesto

Respecto del motor, se analizaron dos alternativas distintas, siendo ambas de la misma familia de modelos NEMA 17. NEMA tiene un estándar de motores stepper que se caracteriza por fijar y tabular las dimensiones en términos de sección, teniendo como grado de libertad el largo del cuerpo de los mismos. En este sentido, una mayor longitud del rotor y estator de estos motores repercute en una mayor capacidad de torque.

Por esta razón, en primera instancia se consideró el uso de un motor NEMA 17 de 12V y 1A con una longitud de 38 mm y un torque de 42Ncm.



FIGURA 12.7: Motor NEMA 17

Luego de realizadas las pruebas sobre dicho motor, se pudo comprobar que la fuerza que el mismo proveía, si bien era suficiente para mover el eje, no lograba ajustar bien en la zona de la rosca en donde se debía realizar más fuerza sobre el volante de inercia (es decir, en la zona de mayor carga para el ciclista). Por esta razón se consideró el uso de un segundo motor, también NEMA 17 pero esta vez de 60mm de largo y con un torque de 91Ncm. Con este motor las pruebas realizadas fueron satisfactorias, logrando frenar la bicicleta en todos los casos sin esfuerzos aparentes ni picos

de tensión o corrientes.

Luego, en la figura 12.8, se aprecia el modo de conexión del motor elegido con el sistema de engranajes previamente diseñado en 3D.

Por otro lado, para el diseño de los engranajes se decidió en primera instancia utilizar una relación de engranajes 1:1 para el sistema. Pero luego se comprobó que las velocidades alcanzadas por el motor cumplían con los requisitos establecidos con creces, teniendo por ende un margen que puede ser ajustado. Por otra parte, el requerimiento de la fuerza necesaria en el motor para hacer girar la varilla no tenía tal intervalo de margen, por lo que se decidió implementar como segunda iteración del sistema de transmisión mecánica unos engranajes fabricados con impresión 3D con relación de reducción de 2:1. De esta manera, se pierde en velocidad de giro, pero se gana en la fuerza que se es capaz de transmitir para lograr el movimiento del eje.

El montaje final del motor realizado en el prototipo se muestra en la figura 12.8.

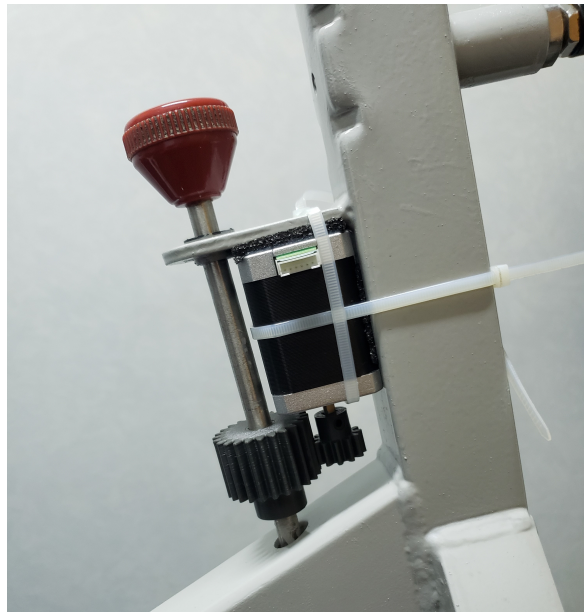


FIGURA 12.8: Motor montado en el prototipo

Este conjunto mecánico debe ser comandado por un driver apropiado que le permita al microcontrolador accionar el motor stepper en forma eficiente, sin necesidad de alimentarlo con la tensión de 3,3V. Es así como se elige el driver A4988, dado que es un estándar respecto del control de este tipo de motores, por lo que se considera una alternativa confiable.

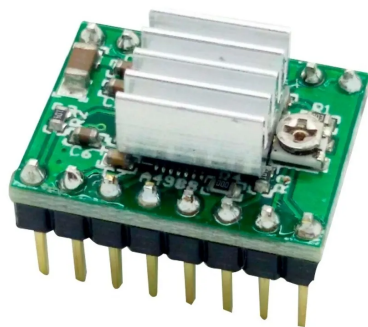


FIGURA 12.9: Driver A4988

Esto permite brindarle al microcontrolador una interfaz del tipo PWM, aprovechando los pines dedicados que tiene a tal fin. Asimismo, respecto de las características de este driver, se destaca que posee una protección frente a sobrecorrientes, regulable mediante un preset. Esto permite proteger al motor paso a paso ante eventos tales como un atascamiento mecánico.

Plan de Pruebas Para verificar el funcionamiento del módulo que controla la fuerza sobre la bicicleta, en primera instancia se prueba el motor *stepper* fuera del conjunto mecánico, conectado a un microcontrolador que ejecuta un programa *mock* que simula giros del motor consecutivos en ambos sentidos. Una vez superada esta etapa, se monta el motor en la bicicleta junto al conjunto de engranajes y se repite la prueba anterior. Por último, otro programa hace variar la carga percibida por el ciclista entre su mínimo y máximo, y se verifica que efectivamente el motor es capaz de accionar el mecanismo de ajuste de fuerza en todo su rango de forma satisfactoria.

12.1.4. Sensor de velocidad

Resulta importante integrar dentro de la bicicleta un sensor de velocidad que permita medir la velocidad de pedaleo del ciclista para transmitir estos datos al sistema que integra realidad virtual y así tener correlación entre el movimiento del usuario en el mundo físico y en el mundo virtual. Esta correlación es uno de los aspectos más importantes para la inmersividad que se intenta lograr con este proyecto.

Para la medición de dicha velocidad se decidió integrar un sensor que mida la velocidad de rotación del volante de inercia, debido a que la instalación del sensor es más simple y robusta si se realiza allí en lugar de realizarse sobre los pedales. Esto se puede realizar debido a que la relación entre la velocidad de pedaleo y el giro del volante de inercia es fija y lineal.

Es por esto que se integran marcas dentro del volante de inercia para luego medir la señal de salida del sensor óptico que variará digitalmente en función de si refleja la luz en el volante de inercia construido en metal reflectivo o si no la refleja debido al uso del material de baja reflectividad añadido.

De esta manera, la señal de salida del sensor óptico será un tren de pulsos cuya frecuencia cambiará en función de la velocidad de rotación del volante de inercia.

Para la medición de dicha frecuencia se utilizan interrupciones en el microprocesador que son ejecutadas en cada flanco ascendente de la señal del sensor. El uso de interrupciones permite utilizar el procesador para otras operaciones y salir de su flujo principal solo cuando su uso cuando sea necesario, esto el uso distribuido de las funciones del sistema dentro del ESP32, pero implica consideraciones adicionales debido a que es importante que las interrupciones sean rápidas y su uso sea relativamente limitado.

Esto fija un número máximo de marcas que se pueden utilizar dentro del volante de inercia, debido a que la cantidad de marcadores está directamente relacionada con la cantidad de interrupciones por vuelta del volante.

Por otro lado, se puede analizar el mínimo número de marcadores necesarios en el volante de inercia, debido a la necesidad de resolución en bajas velocidades, principalmente durante el inicio del pedaleo. Se comprobó que con un solo marcador el lag presente en la bicicleta es suficiente para entorpecer la experiencia. Con dos marcadores es suficiente, pero notorio, con tres marcadores espaciados a intervalos regulares, en cambio, el sistema se comportó de forma aceptable. En la implementación se decidió utilizar cuatro marcadores debido a que para ese número el retardo en la primera parte del pedaleo es prácticamente imperceptible y, por otro lado, el sistema funciona muy bien sin tener demasiadas interrupciones dentro del esp que pudieran entorpecer el funcionamiento del mismo.

Plan de Pruebas En primera instancia, se efectúan lecturas apuntando el sensor hacia el volante de inercia (reflectivo), y luego hacia las marcas previamente realizadas sobre el mismo (material de baja reflectividad). En este punto se verifica que el sensor conmute entre ambas lecturas. Luego, se ejecutan pruebas con el sensor montado en las proximidades del volante de inercia, con la bicicleta girando. Se imprimen las velocidades registradas en pantalla de forma tal de poder verificar la estabilidad de la medición, así como el cambio en esta a medida que se varía la velocidad de pedaleo.

12.1.5. Vibración

Para el sistema de vibración se tuvieron en consideración diferentes opciones de motores que permitieran transmitir al usuario la sensación de vibración necesaria para la simulación en terreno rugoso con la bicicleta en el mundo virtual.

En un principio se consideró el uso de motores de vibración pequeños que permitieran su instalación de manera simple, basándose en sistemas de vibración como la utilizada en los dispositivos de telefonía celular que pueden verse actualmente. A continuación, en la figura 12.10 se puede apreciar el motor de vibración probado en la bicicleta.

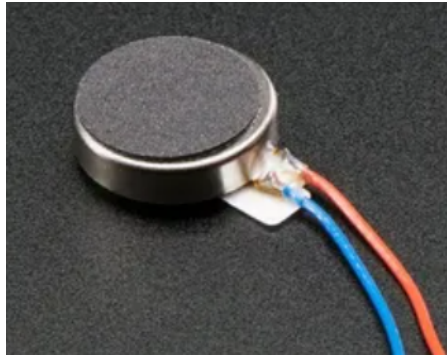


FIGURA 12.10: Motor de vibración tipo pila.

Luego de realizada la prueba se pudo comprobar empíricamente que no cumplía con los requisitos necesarios. Por lo que se decidió utilizar otros tipos de motores de vibración considerados durante la primera etapa de este desarrollo.

De esta manera se empleó un motor de vibración DC XL160830 como el que se puede ver en la figura 12.11. Dicho motor tiene una tensión de 5V y una corriente nominal de funcionamiento de 450mA.



FIGURA 12.11: Motor de vibración utilizado en la bicicleta.

Dicho motor vibratorio se encuentra agarrado en el manubrio con la finalidad de transmitir en él las vibraciones características de la bicicleta. Para controlarlo, se emplea un doble puente H basado en el driver L9110 que permite operar a una tensión nominal de 2,5V a 9V con una corriente de 800mA por canal. Para simular superficies con distintas rugosidades se utilizan señales PWM y solamente se varía el *duty cycle* de las mismas hasta los valores requeridos.

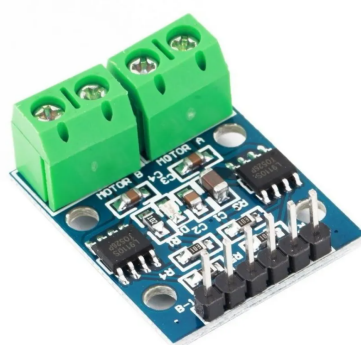


FIGURA 12.12: Driver L9110.

Plan de Pruebas Con el objetivo de validar el funcionamiento del sistema de vibración, en primer lugar se debe corroborar el funcionamiento del mismo fuera del montaje de la bicicleta. Para eso, se ejecuta un programa en el microcontrolador que envía las señales necesarias para accionar el motor de vibración con la fuerza requerida por las especificaciones. En segundo lugar, es necesario testear la vibración con el motor ya montado en el manubrio de la bicicleta fija. Para determinar el mejor punto de montura y para calibrar la señal que debe recibir el motor se utiliza el *testbench* desarrollado en la sección 7.1.2. Se miden las componentes de la vibración y se determina si las medidas resultantes cumplen con los requerimientos impuestos.

12.1.6. Alimentación

Para la alimentación del sistema se utiliza una fuente switching externa que se conecta a la red eléctrica (110V/220V), que permite alimentar al sistema con 12V de tensión constante.

El sistema requiere de una tensión de 12V para la alimentación del motor *stepper* que controla la fuerza, cuyo consumo es de hasta 2A. Asimismo, también se necesita proveer al equipo de una alimentación de 3,3V para la parte lógica que controla a dicho motor y para el sensor de velocidad. Por último, se necesita una alimentación de 5V para los motores de vibración, para el driver que los controla y para alimentar al ESP32. Por esta razón, se utiliza una fuente reductora de tensión tipo *Buck* para obtener este nivel de tensión a partir de los 12V entregados por la fuente principal, mientras que la alimentación de 3,3V es obtenida a partir del regulador lineal presente en el módulo del ESP32.

Respecto a las intensidades de corriente consumidas, el motor *stepper* tiene un consumo máximo de 1,7A que puede ser regulado por su *driver* (ver sección 12.1.3). Por otro lado, la alimentación del ESP32, que es de 5V, requiere de una corriente de 250 mA cuando está en modo activo. Otro subsistema que se vale de la alimentación de 5V es el motor de vibración, requiriendo una corriente de 450mA en operación normal. El módulo de sensado de velocidad solamente consume una corriente de 7 mA.

Considerando todo lo anterior, se pueden resumir los consumos de corriente en los dos niveles de alimentación propuestos, acorde a la tabla que se muestra a continuación:

Tensión de alimentación (V)	Consumo de corriente (A)	Potencia (W)
3.3	7×10^{-3}	0.023
5	0.7	3.5
12	1.7	20.4

Luego, se desprende que el consumo total de potencia del sistema es de aproximadamente 24W, sin considerar el consumo del visor de realidad virtual, dado que se alimenta de su propia batería. Para la tensión de 5V se utiliza un convertidor DC-DC del tipo Buck (reductora de tensión) que soporta una corriente máxima de 2A. Si se considera que la eficiencia de dicha fuente es del 80 % entonces el resultado en la corriente de entrada de 12V en dicha fuente resulta en 0,5A. Considerando un margen de seguridad del 20 % se obtiene un valor de 0,6A provenientes de la fuente de 12V, que alimentan a la fuente de tensión Buck.

Luego para el caso del motor, si se considera una corriente máxima de 1,7A limitada mediante el driver del motor, y si se prevé un margen de seguridad de 20 %, entonces el valor resultante en la corriente final de consumo en la etapa de 12V del motor stepper es de 2A.

Por lo tanto, la corriente que debe soportar la fuente principal de 12V que se elija debe ser de al menos 2,6A, por lo que se podría utilizar una fuente de 12V 3A en etapa de producción. Finalmente, se emplea una fuente de 12V 5A para el prototipo por razones de disponibilidad.

12.2. Software

En esta sección se analiza la ingeniería de detalle en lo que respecta a la componente de software de la solución propuesta. Este análisis gira en torno a los siguientes apartados.

- Networking
- Aplicación de VR
- Servidor
- Aplicación del coach
- Aplicación de usuario
- Base de datos

12.2.1. Networking

Con respecto al Networking, resulta conveniente recordar los módulos o subsistemas que forman parte del proyecto. Para esto se recomienda revisar la figura 12.1 en la que se pueden ver los módulos utilizados y la interconexión entre ellos.

Dentro del Networking existen comunicaciones de diferentes tipos. Entre estas se encuentran aquellas que se posibilitan mediante el uso de Internet, las que se efectúan a través de Bluetooth y las comunicaciones que ocurren en la red LAN del gimnasio utilizando el protocolo WebSockets.

Entre las comunicaciones mencionadas se presentan algunas que son críticas para el funcionamiento de VRBike y para la experiencia del usuario. Por esta razón, se categorizaron entre aquellas que necesariamente deben ser rápidas y aquellas donde la latencia no resulta prioritaria. Dicha categorización y la correspondiente selección de los tipos de comunicaciones permiten que el sistema de VRBike siga funcionando en caso de cortes de Internet o problemas con el servicio.

Entre las comunicaciones críticas se encuentran las comunicaciones entre la app del coach, el servidor, los visores y las bicicletas, ya que se debe sincronizar el estado de la sesión en tiempo real y sin cortes. Debido a esto, se seleccionaron conexiones vía Bluetooth y vía red LAN, las cuales por sus características permiten obtener muy bajos tiempos de latencia y no dependen de la conexión a Internet. Por otra parte, la actualización de las métricas una vez finalizada la sesión y la obtención de ellas desde la aplicación móvil no precisan darse instantáneamente y pueden ser generadas de forma no sincrónica, por lo que la necesidad de utilizar Internet obteniendo un tiempo de latencia mayor no resulta un problema en este apartado.

A continuación se resumen los diferentes tipos de comunicaciones y sus usos más frecuentes.

■ Comunicaciones por Internet

El sistema de VRBike funciona en su mayor parte sin necesitar acceso a Internet. Mediante una red LAN local (esto es, sin la necesidad de acceso a la red externa) los participantes se pueden conectar al servidor y participar de la sesión de spinning. Sin embargo, a fin de que los participantes puedan autenticarse y que las métricas queden almacenadas en la base de datos para que luego puedan ser mostradas desde la aplicación móvil, es necesario que ciertos módulos sí posean acceso a Internet. La aplicación móvil lo requiere para comunicarse con la base de datos (Google Firebase) y obtener los datos del usuario, así como también para registrar la vinculación con la bicicleta que se vaya a utilizar en la sesión. Por otra parte, el servidor tiene la capacidad de detectar nuevos usuarios vinculados a las bicicletas también mediante Internet. A su vez, al finalizar la sesión, puede actualizar las métricas de los usuarios en la base de datos haciendo uso de esta conexión. En el caso de que esta conexión no estuviera activa, simplemente se registran las métricas para mostrarlas durante el transcurso de la sesión de spinning, pero no se almacenan en la nube hasta que no se reconecte a Internet.

■ Comunicaciones por Bluetooth

La conexión entre la bicicleta fija y el visor de realidad virtual se realiza de manera inalámbrica mediante el uso de una conexión Bluetooth mediante BLE (Bluetooth Low Energy). De esta forma, se configura al ESP para que se comporte como un controlador externo del visor de realidad virtual. Aunque actúe como un controlador, la conexión también permite enviar y recibir datos e instrucciones desde el visor hacia la bicicleta y viceversa.

■ Comunicaciones por red LAN

Las comunicaciones entre la aplicación del coach y el servidor del gimnasio y la comunicación que se realizan entre dicho servidor y los visores de realidad virtual se realizan mediante el uso de una red LAN, en específico, mediante el protocolo WebSockets. Dicho protocolo permite la comunicación de datos de forma bidireccional de manera simultánea, lo que es muy útil para este tipo de aplicaciones donde es necesario transmitir datos y eventos entre los sistemas en tiempo real.

Subrutinas Si bien el Networking no ejecuta subrutinas, ya que no se trata de un programa, se las puede relacionar con la forma en la que conectan los dispositivos. Como se mencionó, se establecieron conexiones tanto por medio de red LAN como por Bluetooth. En las conexiones por medio de la red LAN se utilizó el protocolo WebSockets, el cual le dispone a los clientes las conexiones 1 a 1, o 1 a varios. Al utilizar este protocolo, se deben codificar en el cliente subrutinas a ejecutar para cuando llega cada tipo de mensaje por medio del protocolo utilizado. Lo mismo se debe desarrollar en los dispositivos que utilizan la conexión Bluetooth (visor de VR y bicicleta), pero solo se trata el caso de la conexión 1 a 1.

Complejidad La complejidad del Networking es, en el peor caso, $O(n^2)$. Este caso se da al recibir métricas instantáneas de cada participante de la sesión y reenviarlas a todo el resto de ellos. Sin embargo, dicha complejidad es aceptable, ya que la cantidad de participantes está limitado al tamaño de la sesión de spinning, el cual es en general de 15 personas.

Plan de Pruebas Como plan de pruebas para el Networking, se establece simular el envío de datos de 15 participantes, tal como ocurriría en una sesión de spinning. Esta simulación se realiza mediante un script desarrollado para tal fin que utiliza el Networking de VRBike para conectarse con el servidor, tal como lo haría una bicicleta. En esta prueba, se verifica que todos los dispositivos reciban un mensaje inicial de prueba ('ping') que, una vez que alcanza al cliente, retorna al servidor. Dicha acción verifica que todos los clientes se encuentren conectados al inicio y simula el envío de la acción de iniciar la sesión que realiza el coach. Además, se envían otros mensajes de prueba durante la sesión y se registra en cada dispositivo el momento en que son recibidos. En ambos casos, la latencia máxima aceptable es de 70ms, es decir, 140ms entre la ida y vuelta al servidor.

12.2.2. Aplicación de VR

Respecto a la aplicación que se ejecuta en los visores de realidad virtual, en primera medida se consideró el uso de escenarios 3D basados en el modelado de escenarios virtuales sintéticos realizados mediante el motor gráfico Unity. Por ello, se implementó una prueba de concepto en la cual se buscó determinar si dicho tipo de escenarios permiten obtener un ambiente realista. El escenario desarrollado se muestra en la figura 12.13.



FIGURA 12.13: Escenario 3D sintético diseñado.

El sistema desarrollado se pudo testear en computadoras y se comprobó que funcionaba de forma correcta y transmitía las sensaciones buscadas a los usuarios.

Una vez realizada, se dialogó con los potenciales compradores, a los cuales se les mostró la aplicación para obtener una retroalimentación de su parte. En esta etapa, se pudo comprobar que si bien les pareció interesante el contenido desarrollado, expresaron que la posibilidad de ofrecer movimiento libre por el escenario virtual a los ciclistas no es deseado. Esto se debe a que dicho movimiento genera distracciones innecesarias que no aportan al entrenamiento. En cambio, si el ciclista únicamente se concentra en pedalear, puede lograr mejores resultados.

Debido a esto, se optó por elegir otro rumbo para el desarrollo de la aplicación de realidad virtual, ya que al tener que restringirse el movimiento de los usuarios dentro del mundo virtual a un camino único, se abren otras opciones de desarrollo. En esta línea se decidió utilizar videos reales en 360 grados como mundos virtuales, en los cuales el usuario puede mover la cabeza para visualizar en diferentes direcciones durante el entrenamiento. A fin de que funcionen de forma correcta y lograr la inmersión, se modifica la velocidad de reproducción del video con base en la velocidad de rotación del volante de inercia detectada por el sensor de velocidad. En la figura 12.14 se puede ver una imagen de estos escenarios.



FIGURA 12.14: Escenario virtual 3D generado con videos en 360 grados.

Esta opción tiene muchas ventajas respecto a la anteriormente considerada. En primer lugar, al

tratarse de videos reales, se maximiza el realismo y la inmersión de los escenarios. Esto es indispensable para lograr usuarios recurrentes y, sobre todo, para interesar a aquellos no familiarizados con la tecnología de realidad virtual. A su vez, permite acortar los tiempos de desarrollo de los escenarios virtuales.

Una vez concluido el desarrollo de un escenario siguiendo este paradigma, se les mostró el desarrollo de la nueva versión a los potenciales clientes, quienes quedaron conformes con el resultado obtenido. Se manifestó una recepción positiva en cuanto a la evolución gráfica lograda, sin con esto haber reducido la inmersión ni ampliado los costos de desarrollo.

Comunicaciones El dispositivo de realidad virtual, además de mostrar el escenario al jugador, tiene la función de ser una parte activa dentro del Networking del sistema. Las comunicaciones que llegan y salen de la aplicación de realidad virtual se muestran en la figura 12.15.

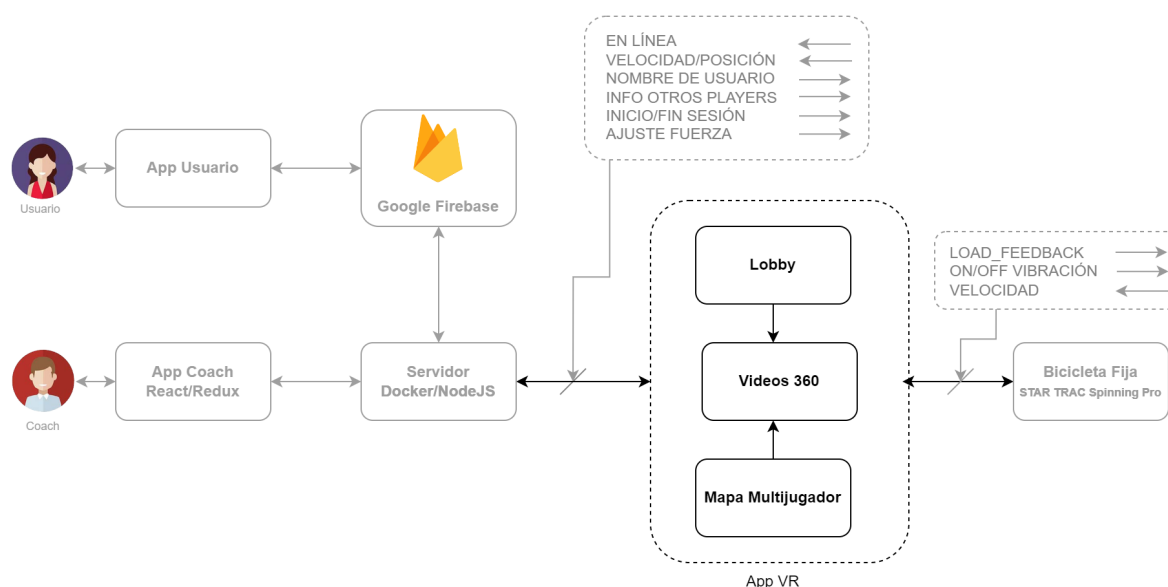


FIGURA 12.15: Diagrama esquemático de la aplicación de VR.

Subrutinas Con todo lo dicho se pueden mencionar a continuación algunas de las subrutinas que se implementan dentro de este software, esto puede ayudar a entender las tareas que esta celda tiene dentro del sistema.

- **LoadFeedback:** Esta función permite cambiar el nivel de fuerza que se aplica sobre el volante de inercia dependiendo del terreno en el que se encuentre el ciclista en el mundo virtual, modificando de esta manera la fuerza con la que el ciclista tiene que pedalear para lograr mover el volante de inercia.
- **VibrationFeedback:** Esta función permite cambiar el estado de una variable en el controlador de la bicicleta para con esto regular la vibración del motor de vibración integrado en la bicicleta fija en función del tipo de terreno en el que se esté transitando (a saber, asfalto, tierra, pasto, etc.)

- **OnServerChanges:** Esta función se ejecuta con los mensajes que llegan del servidor y permite sincronizar a los usuarios que forman parte de dicha sesión.
- **OnLine:** Esta función permite avisar al servidor cuando el visor de realidad virtual abre nuestra aplicación de ciclismo virtual. De esta manera, el servidor puede saber a quién enviar y de quien recibir datos.
- **OnNewSpeed:** Cada vez que llega un parámetro nuevo de velocidad (que puede ser igual al anterior, pero igualmente se activa esta función), esta función se activa y nos permite actualizar la velocidad de reproducción del video en función de la velocidad que llega a esta celda correspondiente al pedaleo del ciclista. Además, permite enviar dicho valor junto con la posición del jugador al server, de esta manera este puede saber en qué posición se encuentra cada usuario y a qué velocidad está pedaleando.
- **OnNewName:** El servidor enviará a la app VR el nombre de los demás jugadores y su propio nombre a medida que los vaya obteniendo de la autenticación realizada con Firebase, siempre que los usuarios hayan activado este proceso haciendo uso del código QR y de la aplicación de celulares. Esta función permite modificar los nombres de los demás jugadores y el del propio jugador dentro del mundo virtual.
- **OnNewLoad:** Esta función nos permite modificar el valor de fuerza que se ha de ejercer sobre el volante de inercia, en función del valor que llega desde el servidor proveniente inicialmente desde el parámetro ajustable en la app del coach. Esta función envía el valor de fuerza modificado a la bicicleta para que esta pueda con el motor stepper, regular la fuerza de accionamiento sobre el volante de inercia.
- **StartSesion:** Esta función permite el comienzo de la sesión de spinning virtual con base en un mensaje enviado desde el servidor enviado previamente a este desde la app del coach.
- **StopSesion:** Esta función permite detener la sesión de spinning virtual con base en un mensaje enviado desde el servidor enviado previamente a este desde la app del coach.

Además de estas funciones, la aplicación de VR tiene las funciones necesarias para mostrar el escenario virtual 3D que se está mostrando en pantalla. Con este fin se hace uso de subrutinas tales como `VisualizeMenu` o `VisualizeScenario` que funcionan de manera similar a los estados en una FSM donde es posible moverse entre los diferentes estados en función de las señales que se tengan en el momento.

En la figura 12.16 se puede ver el menú de inicio en el cual el usuario se encuentra cuando espera a que la sesión comience.



FIGURA 12.16: Imagen de la aplicación del casco en el menú antes de comenzar la actividad.

A continuación en la figura 12.17 se puede ver al sistema funcionando dentro del casco de realidad virtual, en esta figura se puede ver el escenario virtual en donde el usuario desarrolla su actividad.



FIGURA 12.17: Imagen correspondiente al mundo virtual que la persona observa mientras realiza el entrenamiento.

Complejidad En general, las funciones con las que se trabaja en este apartado presentan complejidad $O(n)$, en donde n es el número de datos que se han de utilizar para realizar la tarea. Por lo que en cuanto al poder de cómputo empleado, no es un problema grave con relación a la complejidad de las funciones, sino más bien un problema relacionado con el número de veces que dichas funciones se ejecutan en el tiempo disponible. Esto no constituye un problema para nuestra aplicación debido a que los tiempos de envío y recepción de paquetes de información son empleados de forma tal de lograr el tiempo real buscado, pero, por otro lado, no generar con esto

una reducción significativa en las imágenes por segundo que el visor puede mostrar en la pantalla.

Plan de Pruebas Para corroborar la funcionalidad del sistema se realizan pruebas en cuanto al número de datos que se pueden enviar por segundo sin que el sistema vea limitado en su capacidad de cómputo. Por otro lado, también se considera utilizar videos en diferentes resoluciones para poder comprobar cómo se desempeña el visor en tales situaciones. En ambos casos es importante validar que se alcancen los objetivos mínimos, no siendo necesario encontrar las cotas máximas de uso, debido a que vídeos de mucha calidad tienen el problema de que resultan más pesados en cuanto al espacio en memoria que consumen, algo que es complejo de resolver sin encarecer el costo de la solución (estamos hablando de resoluciones de video altas como 8k que en principio no entran dentro del alcance inicial del proyecto) Es por esto que se decidió por utilizar videos en resolución ultra HD (4K). Y, por otro lado, es necesario lograr una cantidad mínima de mensajes por segundo recibidos y enviados para lograr la sensación de real time buscada, pero no es necesario que este número de paquetes llegue a valores máximos posibles, siempre que la sensación buscada sea lograda.

12.2.3. Servidor

Una parte fundamental del sistema de VRBike es el servidor, el cual cuenta con las siguientes funcionalidades:

- Sincronización de sesión.

Las sesiones de spinning se caracterizan por contar con múltiples participantes realizando el ejercicio. Con el sistema de VRBike, las bicicletas estarán generando datos del desempeño del usuario que se deben compartir tanto con el coach de la sesión como con el resto de los participantes.

En mayor detalle, el coach debe en tiempo real obtener y evaluar indicadores de los participantes, tales como la velocidad y la potencia actual. Así también, debe poder encontrar en la aplicación a las personas que se suman a la sesión y poder dar inicio una vez que todos están listos. Por otro lado, los participantes, mientras realizan el ejercicio, contarán en la aplicación del visor de VR con una interfaz que compare, también en tiempo real, su desempeño con respecto al resto.

Aunque las funciones específicas de la aplicación del coach y de la aplicación del visor de VR se desarrollan en las secciones 12.2.4 y 12.2.2, es importante remarcar que se necesita una forma de sincronizar el estado de la sesión. Esta tarea es la función principal del servidor.

Por consiguiente, el servidor recibe datos de las bicicletas de los participantes y las sincroniza, haciéndolas disponibles a los otros participantes y al coach. Además, recibe del coach eventos a realizar sobre la sesión, tales como dar comienzo y finalizar, y envía estas señales a todos los participantes para que su sesión comience al mismo tiempo. Adicionalmente, a fin de realizar correctamente sus funciones, registra durante el tiempo total de la sesión a los participantes presentes.

- Detección de bicicletas vinculadas.

Además de la sincronización del estado de la sesión, el servidor detecta si una persona vinculó su bicicleta con su usuario de la aplicación de métricas (mecanismo explicado en la sección 12.2.5) y, en dicho caso, actualiza su nombre e informa sobre este cambio a todos los participantes.

- Ajuste de fuerza.

Así como se mencionó que el coach da inicio a la sesión y esto se sincroniza en todos los visores presentes, este también cuenta con la función de ajustar el nivel de fuerza que está ejerciendo una bicicleta sobre un ciclista. Se recuerda que este nivel depende del tipo e inclinación del terreno en el escenario virtual, pero se le añade la posibilidad al coach de modificarlo por distintas razones. Por ejemplo, podría detectar que un ciclista está ejercitándose de forma muy ligera sin realizar el debido esfuerzo o que una persona no está pudiendo completar el entrenamiento. En estos casos, el coach debería aumentar o disminuir respectivamente el nivel de fuerza de la bicicleta. Esta funcionalidad es descrita en la sección 12.2.5. Sin embargo, a consideraciones del servidor, este tiene la función de recibir estas acciones del coach y comunicarlas al participante correspondiente.

- Actualización de métricas en la base de datos.

Por último, el servidor almacena todas las métricas de los participantes, además de sincronizarlas. Esto se realiza para que, una vez finalizada la sesión, calcule métricas relevantes, como la velocidad y la potencia promedio de cada individuo y las envíe a la base de datos en la nube. Así, estas métricas quedan disponibles para que la aplicación móvil pueda mostrarlas (ver sección 12.2.5).

Con respecto al desarrollo del servidor, se utiliza al entorno Node.js y al lenguaje de programación TypeScript. Node.js fue elegido dado que es un estándar en el desarrollo de servidores y cuenta con múltiples ventajas como ser eficiente, escalable, multiplataforma y de código abierto. Por su lado, el lenguaje TypeScript se basa en el lenguaje JavaScript, uno de los más utilizados y ofrece características adicionales que solucionan problemas que se presentaban anteriormente, tal como la ausencia de tipos de datos. Además de Node.js y TypeScript, se hace uso de la biblioteca open source denominada [Colyseus](#). Esta biblioteca es ampliamente utilizada en el desarrollo de servidores multijugador, dado que permite implementar de una forma muy sencilla las conexiones necesarias y se encuentra basada en la transmisión por WebSockets. Adicionalmente, cuenta con SDKs tanto para Unity (utilizado para la aplicación del visor de VR) como para JavaScript (utilizado para la aplicación del coach), con lo cual resulta muy adecuada para el sistema de VRBike.

Otra característica del desarrollo del servidor es la utilización de la tecnología de contenedores Docker. Esta tecnología permite agrupar al código del servidor junto a todas sus dependencias en un único paquete, llamado contenedor. Mediante el uso de contenedores, se estandarizan las operaciones y se independiza la ejecución de las aplicaciones sobre la plataforma en la cual se realice. Así, se solucionan fallas tradicionales en las cuales una aplicación funcionaba en un ambiente, pero

no en otro. De esta forma, Docker permite independizarse del ambiente sobre el cual se despliegue el servidor, sea este una máquina local, virtual o un entorno en la nube. En la figura 12.18 se incluye una captura del contenedor Docker del servidor siendo ejecutado.

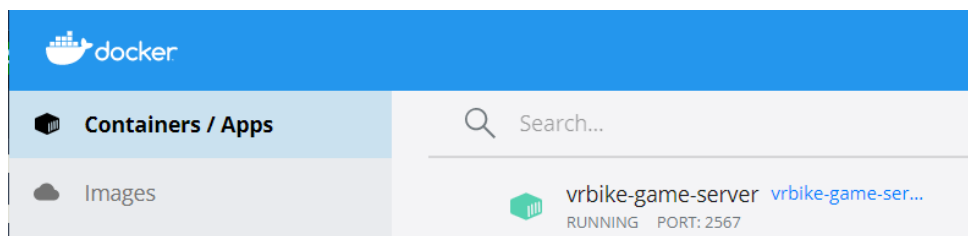


FIGURA 12.18: Contenedor Docker del servidor en ejecución.

Comunicación El servidor juega un papel central en las comunicaciones que son necesarias en el sistema, dado que es quien se encarga de comunicar las acciones del coach en su aplicación, conectar los dispositivos entre sí en la misma sala, y entregarle las métricas de los usuarios a la app del coach. A su vez, permite el guardado de los datos en la base de datos en la nube con Firebase. En este sentido, en la figura 12.19 se puede ver un diagrama con las comunicaciones en las que el servidor está involucrado.

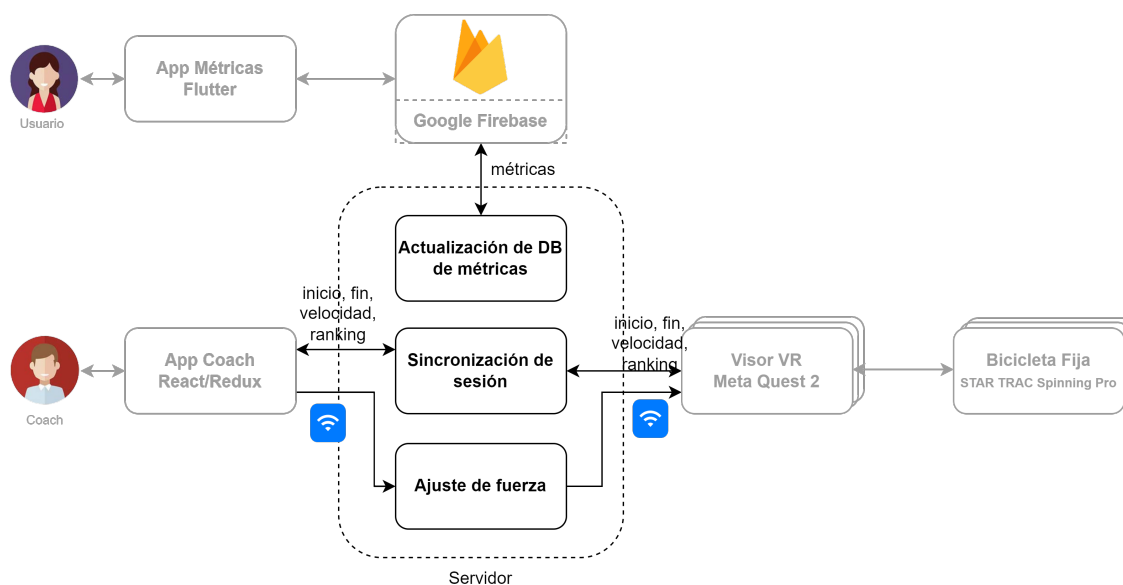


FIGURA 12.19: Gráfico esquemático que muestra las comunicaciones entre el servidor y las demás partes del sistema

Subrutinas A continuación se incluye una lista con las principales subrutinas ejecutadas por el servidor.

- **onCreate:** Subrutina ejecutada al iniciar una instancia del servidor y crear una sesión. En ella se inicializa el estado de la sesión y se inicializan las conexiones a la base de datos.
- **onPlayerJoin:** Subrutina que se ejecuta cuando un nuevo participante se une a la sesión. Esta rutina inicializa sus valores y verifica si se encuentra autenticado en la base de datos. Si se lo

encontrara, cambia su nombre de 'Invitado' a su nombre real y finalizada la sesión registra sus métricas en la base de datos.

- **onCoachJoin:** Subrutina que se ejecuta cuando el coach se conecta al servidor. Guarda su identificador a fin de enviarle los mensajes de actualización de las métricas de los participantes.
- **onLeave:** Subrutina ejecutada cuando un participante se desconecta de la partida para eliminarlo de la sesión.
- **onUserValueChange:** Subrutina que se ejecuta cuando se detecta una nueva vinculación entre un usuario y una bicicleta. Su función es cambiar el nombre del participante y registrarlo para almacenar sus métricas al finalizar la partida. También detectar las desvinculaciones de usuarios.
- **onMessage:** Subrutina ejecutada al recibir un mensaje por parte de las aplicaciones conectadas. Los posibles mensajes son:
 - **startSession:** Mensaje que indica iniciar la sesión. Este mensaje es enviado desde la aplicación del coach.
 - **endSession:** Mensaje que indica finalizar la sesión. También es enviado desde la aplicación del coach.
 - **restartSession:** Mensaje que indica reiniciar una sesión. Es otro de los mensajes enviados desde la aplicación del coach.
 - **updateMetrics:** Mensaje que contiene el último valor de las métricas de un usuario. Este mensaje es enviado desde la aplicación de VR de cada participante de la sesión.
 - **changeLevel:** Mensaje que indica cambiar el nivel de fuerza de un participante en particular. Es enviado desde la aplicación del coach.

Dependiendo del mensaje recibido, se ejecutan una serie de instrucciones determinadas que lo procesen.

Complejidad La complejidad computacional varía dependiendo de la subrutina del servidor que se analice. En algunas de ellas es de $O(n)$, donde n es el número de participantes de la sesión. Por ejemplo, al iniciar la sesión, debe configurar su estado inicial y enviar un mensaje a cada participante conectado. Por otra parte, en la subrutina que ajusta la fuerza de un participante en particular, la complejidad es de $O(1)$, ya que no se debe iterar por el resto de ellos.

Sin embargo, existe un caso donde la complejidad aumenta y es al recibir los datos actuales de métricas de los participantes de la sesión. En dicha situación, el servidor debe, por cada participante, recibir las métricas actuales, almacenarlas, y enviar el nuevo estado de la sesión a cada uno del resto de los participantes. De esta forma, la complejidad resulta de $O(n^2)$. No obstante, la biblioteca Colyseus, utilizada para gestionar la sesión multijugador, se encuentra optimizada tanto en cuanto a rendimiento, consumo de memoria y CPU tanto del lado del servidor como del lado del cliente. De

esta forma, permite establecer sesiones de una cantidad considerable de participantes y es utilizada en videojuegos de uso masivo. Por esta razón, y al estar limitados a la cantidad de participantes que caben en una sala de spinning, la complejidad no introduce problemas en las aplicaciones.

Plan de Pruebas Para probar el funcionamiento del servidor se estableció un plan de pruebas. En primer lugar, el servidor se ejecuta y se conecta un único participante y un coach. En esta prueba se verifica, primeramente, que el coach visible al participante. Luego, se envía una indicación de iniciar la sesión desde la aplicación del coach y se verifica en la aplicación de VR que la sesión haya comenzado. La prueba se desarrolla durante 10 minutos verificando que el coach reciba las métricas que registra el participante.

En segundo lugar, se repite la prueba aumentando el número de participantes a 15. Para ello, se utiliza un software que simula datos enviados por una bicicleta. En esta prueba se busca verificar el funcionamiento del servidor y no la latencia, ya que esta se verifica en el plan de pruebas del Networking. Por lo cual, todos los componentes se ejecutan en el mismo ordenador. Lo que se verifica en este caso es que tanto el coach como todos los participantes puedan acceder a las métricas de toda la sesión.

12.2.4. Aplicación del coach

Dentro de las aplicaciones que forman parte de este sistema se encuentra la app del coach. Esta aplicación está diseñada para que el entrenador que forma parte de la sesión de spinning pueda regular la sesión en la que los jugadores están inmersos, conociendo sus métricas en tiempo real.




Autenticación de entrenadores Dentro de la aplicación se integra una pantalla de login y otra para el registro de nuevos entrenadores. De esta manera, el sistema se asegura que los datos solo sean accesibles por el entrenador. A continuación en la figura 12.20 se muestran las opciones de inicio de sesión y de nuevo registros que visualiza un entrenador al iniciar la aplicación.

Login	Sign up
Si eres un Coach y tienes cuenta accede aquí	Si eres un Coach y no tienes cuenta accede aquí.
Ingresar	Registrate ahora
Olvidaste la contraseña?	

FIGURA 12.20: Opciones para ingresar a la aplicación siendo coach.

Lista de usuarios activos La aplicación cuenta con una pantalla en la cual se puede ver la cantidad de jugadores en la sesión en formato de tabla donde se listan los jugadores conectados e

información relevante sobre ellos. Esto le permite al coach conocer el estado de los participantes al inicio de una sesión a fin de poder ayudar a las personas que requieran cierta ayuda para configurar el equipo o arreglar algún problema que haya podido ocurrir. En la figura 12.21 se muestra el formato de tabla con los participantes que forman parte de la sesión.

Participantes						
	Ciclista	Velocidad	Metros Recorridos	Última Actividad	Nivel de Fuerza	Control de Fuerza
	Invitado 1	0.0	0.0	Último Entrenamiento -	1	<div>-1</div> <div>+1</div>
	Pablo	0.0	0.0	Último Entrenamiento Tue Mar 08 2022	1	<div>-1</div> <div>+1</div>

Reiniciar la sesión

Comenzar Entrenamiento!

FIGURA 12.21: Tabla con la información relevante de los usuarios de la sesión.

Niveles de potencia A su vez, también se puede observar otra pantalla en la que se muestran los diferentes niveles de potencia que son alcanzados por el usuario durante la sesión de entrenamiento. Dichos niveles de potencia son calculados con base en la fuerza y a la velocidad que se alcanzan en el ejercicio. Los gráficos de los niveles de potencia alcanzados por los usuarios se encuentran en la figura 12.22.

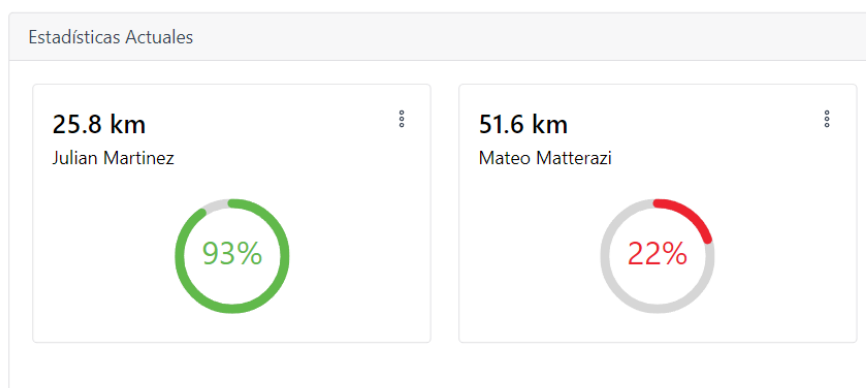


FIGURA 12.22: Gráficos de potencia en tiempo real de los usuarios que forman parte de la sesión.

En la aplicación también se integra una gráfica con las métricas de los jugadores que se actualiza en tiempo real y que le permite al entrenador tener información acerca de los valores de velocidad y potencia alcanzadas por los participantes en cada punto del camino virtual recorrido.

Interruptores de accionamiento rápido Otra característica de la aplicación es que cuenta con botones para la gestión de la sesión. En particular se destacan: uno para comenzar la sesión y otro para detener completamente el entrenamiento (Reiniciar la sesión), que se ha de utilizar en caso de que algo grave ocurra en la sala, como por ejemplo que una persona se descompuso o tuvo algún

problema grave durante la rutina por la que tenga que ser ayudada. Dichos botones se muestran en la figura 12.23



FIGURA 12.23: Botones de accionamiento rápido para comenzar o reiniciar una sesión de entrenamiento.

Control del nivel de fuerza individual A su vez, desde la aplicación el coach puede, a base de la información de los usuarios, aumentar la fuerza con la que realizan la rutina de entrenamientos. Para esto, se cuenta con un selector de nivel de fuerza que internamente actúa como un multiplicador, el cual se combina con el factor característico del terreno, para conocer el ángulo en el cual se debe posicionar al motor de fuerza según el terreno y la dificultad configurada desde la aplicación del coach en función del rendimiento del usuario.

Gráfico de velocidad en tiempo real Según lo conversado con los entrenadores de algunos de los gimnasios de Megatlon, algunos preferirían ver el valor actual de las métricas de los participantes como se muestra en la figura 12.22 (en ese caso se muestra la potencia instantánea desarrollada por los usuarios como porcentaje del nivel en el que se encuentra), mientras que otros prefieren poder ver la progresión de las métricas para cada jugador, para de esta manera tener una idea más general de cómo está realizando su entrenamiento. Es por esto que se decidió incluir además un gráfico con las métricas de los usuarios en tiempo real que se actualiza a medida que se reciben las métricas de posición y velocidad a la aplicación del coach. Dicha gráfica se puede ver en la figura 12.24



FIGURA 12.24: Gráficos de velocidad en tiempo real de los usuarios en función de la distancia recorrida.

Como se puede ver en la figura 12.24, las gráficas de los diferentes jugadores se muestran en celeste, y existen objetivos que se definen como metas por el coach, que son valores que se han de alcanzar durante la sesión. De esta manera, existe un contador en la parte inferior que le indica al

coach cuantos usuarios están alcanzando esa meta. A su vez, se define una zona de descanso que avisa al coach cuantos ciclistas se encuentran en ella en el momento dado.

La gráfica permite seleccionar un punto para tener más información de los usuarios en dicho instante. Así, el coach puede acceder a la información precisa en ese momento en particular de la sesión, como se puede ver en la figura 12.25.

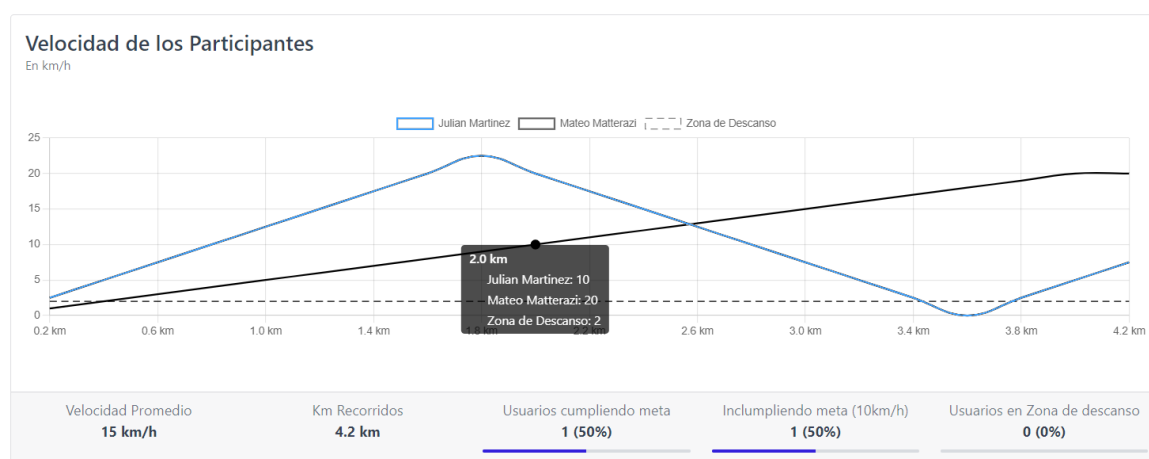


FIGURA 12.25: Menú desplegable sobre cualquier punto del gráfico de velocidad en tiempo real de la figura 12.24 en el que se muestran los datos precisos del punto seleccionado.

Barra lateral La aplicación del coach cuenta también con una barra lateral desplegable que le permite seleccionar la ventana que desea mostrar en pantalla en cada momento. Estas posibles pantallas a mostrar se listan a continuación y su objetivo es integrar algunas de las funcionalidades descritas anteriormente.

- **Participantes:** Pestaña que permite ver la tabla con los participantes que forman parte de la sesión, mostrada en la figura 12.21.
- **Estadísticas:** Pestaña en la que se incluyen los gráficos correspondientes al porcentaje del nivel de potencia en el que se encuentra cada usuario y los kilómetros recorridos hasta el momento, es decir, los gráficos tales como los de la figura 12.22.
- **Métricas:** Pestaña que muestra los diagramas de velocidad en tiempo real para cada uno de los usuarios, mostrados en la figura 12.24.
- **Apariencia:** Pestaña simple que permite modificar apartados de apariencia, como el color de la interfaz y la barra lateral.
- **Cuenta:** Pestaña que le permite al entrenador salir de su sesión al finalizar la clase de spinning.

A continuación, se muestra en la figura 12.26 una imagen de la barra lateral desplegable que está integrada dentro de la plataforma.

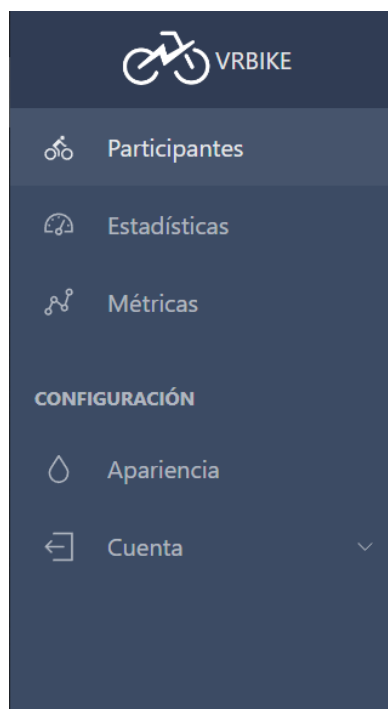


FIGURA 12.26: Barra lateral desplegable.

Responsive Design Durante el desarrollo de la aplicación del entrenador se tuvieron en cuenta las buenas prácticas de programación web para la interfaz de usuario, que forma parte del frontend. En particular, se prestó especial atención a la integración de elementos que permitan lo que en la industria se conoce como Responsive Design, que implica desarrollar la interfaz frontend de forma tal que funcione bien en dispositivos con diferentes tamaños de pantalla, adaptándose a sus interfaces.

De esta manera, se muestra en las figuras 12.27 y 12.28 la visualización de la misma pestaña en dos dispositivos diferentes, donde se puede apreciar cómo el sistema adapta sus componentes de forma que quepan de la mejor manera en el tamaño de pantalla disponible. Esto facilita utilizar el sistema en otros dispositivos en caso de que el usuario así lo requiera en un futuro.

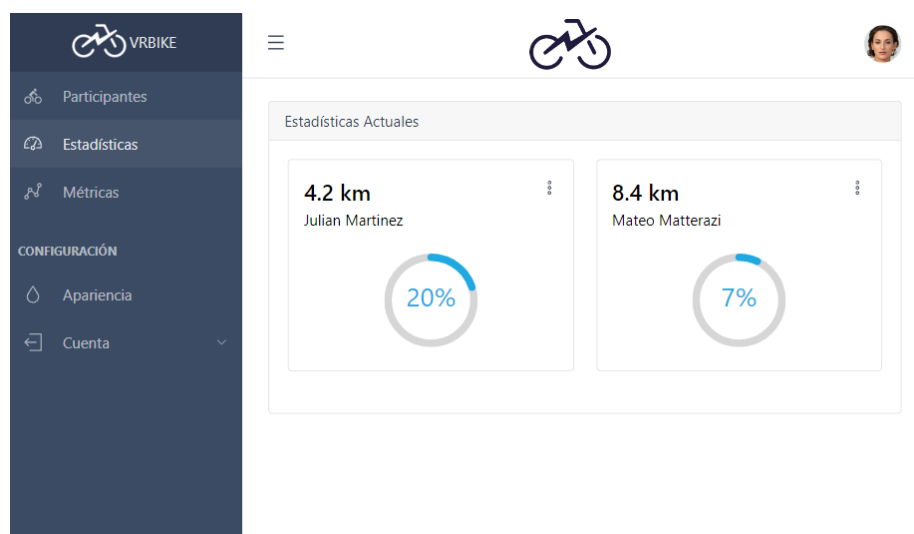


FIGURA 12.27: Visualización de la pestaña 'Estadísticas' en una computadora (resolución 1.920 x 1.080)

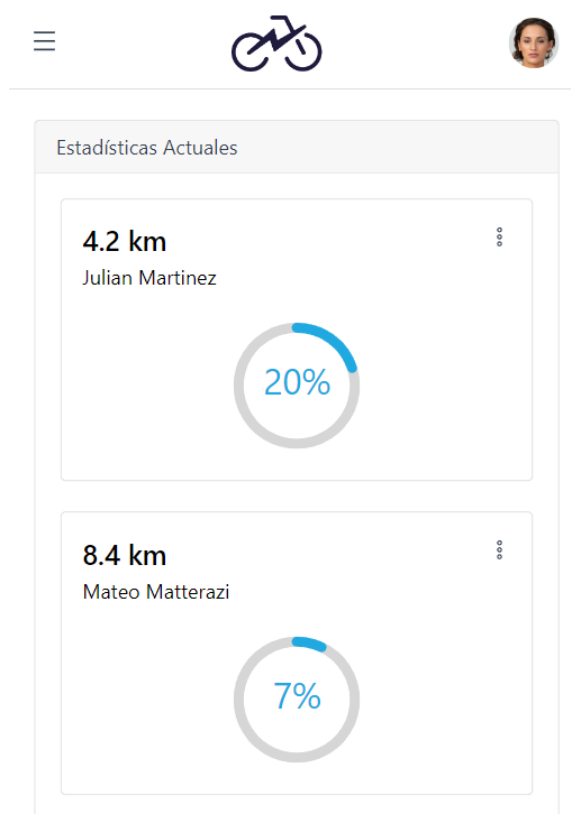


FIGURA 12.28: Visualización de la pestaña 'Estadísticas' en un celular.

Comunicación La aplicación del coach se debe comunicar con el servidor para obtener los datos de la sesión que se muestran en su pantalla. Estos datos corresponden a los participantes de la sesión e incluyen valores correspondientes a los kilómetros recorridos, a la velocidad instantánea y al nivel de potencia en el que se está trabajando.

Por otro lado, también se tienen que enviar los valores de regulación individual de fuerza que se modifican en la pantalla de la app del coach, y las acciones que el coach puede realizar como iniciar la sesión, reiniciarla o finalizarla.

Todas estas comunicaciones, que se realizan entre la aplicación del entrenador y el servidor, se realizan mediante el protocolo WebSockets haciendo uso de una conexión LAN. Estas comunicaciones se muestran en la imagen de la figura 12.29

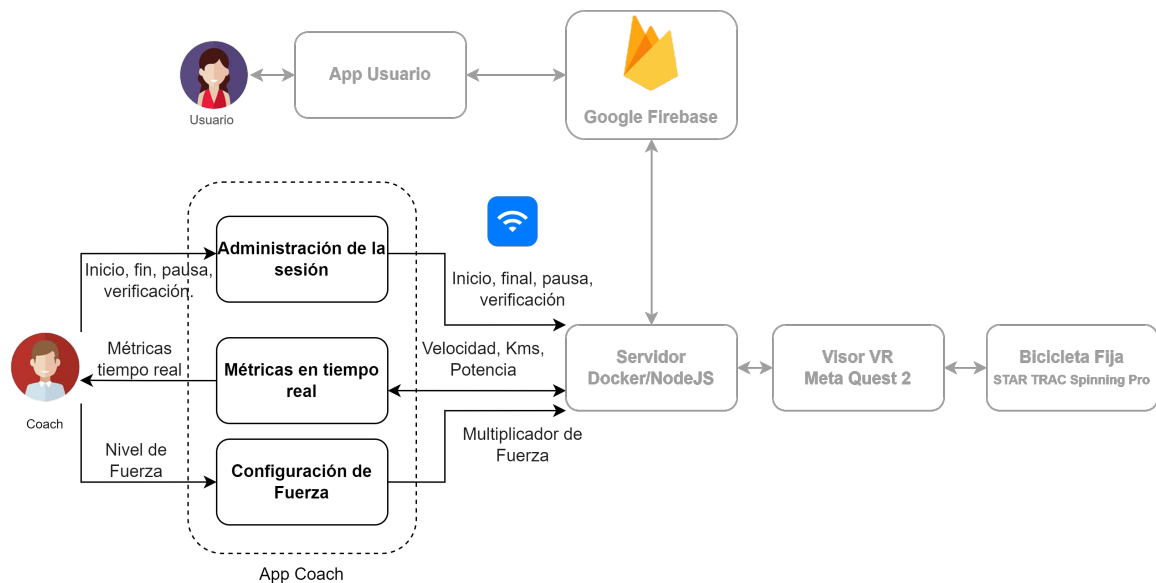


FIGURA 12.29: Gráfico esquemático que muestra las comunicaciones entre la aplicación del coach y las demás partes del sistema.

Subrutinas Con todo lo dicho, se pueden mencionar a continuación algunas de las subrutinas que se implementan dentro del software de la aplicación del coach.

- **StartSession:** Esta función se encarga de enviar el mensaje de inicio de sesión al servidor para que lo comunique al visor de realidad virtual cuando el entrenador pulsa el botón correspondiente.
- **RestartSession:** Esta función se encarga de enviar el mensaje que establece el reinicio la sesión, de forma tal que esta se detenga, pero que no se tengan en cuenta las métricas alcanzadas por los usuarios. Es decir, se comienza nuevamente reiniciando todos los valores almacenados hasta el momento.
- **OnNewData:** Esta función se encarga de actualizar los datos que se muestran en el frontend de la aplicación en función de los valores que llegan desde el servidor.
- **UpdateLoad:** Esta función se encarga de permitir la modificación de la fuerza por parte del coach para cada usuario de manera individual y enviar esta nueva información al servidor.
- **UpdateLook:** Esta función permite realizar los cambios visuales en la interfaz de usuario del coach que se encuentran presentes en la pestaña 'Apariencia'.

- **StartLogging:** Esta función se encarga de utilizar los datos provistos por el coach al ingresar a la aplicación para verificar en la base de datos de Firebase si la autenticación es correcta.
- **SignUp:** Esta función permite registrar a un nuevo entrenador dentro de la base de datos para permitirle acceder a la aplicación.
- **ToggleLateralMenu:** Esta función permite mostrar la barra lateral dentro de la interfaz o esconderla. Se activa cuando el entrenador pulsa el botón de menú destinado para tal fin en la esquina superior izquierda de la interfaz de la aplicación.

Complejidad En general, las funciones con las que se trabaja en este apartado presentan complejidad $O(n)$, en donde n es el número de datos que se han de utilizar para realizar la tarea. Dentro de las pestañas se encontró que el gráfico de la velocidad de cada participante en tiempo real generaba un problema de rendimiento cuando el número de datos a procesar era grande para cada actualización de la pantalla. Es por esto que se restringió el número de datos que se muestran en dicha página a 40 datos por usuario como máximo, lo cual es un valor que nos permite trabajar con la cantidad de usuarios en simultáneo que se considera para el gimnasio (aproximadamente 15 por gimnasio) y, además, 40 valores permite una granularidad suficiente para poder utilizar la aplicación de forma correcta y eficiente.

Plan de Pruebas Para corroborar la funcionalidad del sistema se realiza una prueba con el número de usuarios considerado para esta tarea, y con los valores de datos que se reciben por segundo (límites que se consideran para la situación real). Si se puede verificar que las métricas se actualizan en tiempo real y que no existe lag o problemas con la interfaz visual durante el testeo durante 10 minutos de envío y recepción de datos cambiando entre pestañas mientras esto sucede, se verifica que la prueba tiene resultado positivo y la interfaz de usuario funciona de manera apropiada.

12.2.5. Aplicación del usuario

La aplicación del usuario consiste de una aplicación móvil para los sistemas operativos Android y iOS disponible para todos los usuarios de VRBike a través de las tiendas de aplicaciones oficiales. Se encuentra desarrollada utilizando el entorno de trabajo Flutter y el lenguaje de programación Dart, con lo cual el código es multiplataforma y no es necesario modificarlo para que funcione en los sistemas operativos mencionados. A continuación se detallan sus funcionalidades.

- Autenticación de usuarios.
- Visualización de las métricas del usuario.
- Visualización de las últimas sesiones asistidas.
- Visualización de los logros obtenidos.
- Vinculación con una bicicleta VRBike.

A continuación en detalle cada una de estas funciones.

Autenticación de usuarios Dado que gran parte de la aplicación involucra acceder a datos propios de cada usuario, la aplicación debe contar con una forma de autenticar a quien la utiliza. A fin de maximizar la experiencia de usuario, se decidió realizar esta autenticación mediante redes sociales. Esto se debe a que este método es ampliamente utilizado actualmente e involucra muy pocos pasos a realizar por parte del usuario. El registro tradicional, mediante por ejemplo un nombre de usuario y una contraseña, cuenta con el riesgo de que el usuario abandone la aplicación durante el proceso. Adicionalmente, el usuario podría olvidarla y se debería contar con un complejo sistema de almacenamiento y recuperación de contraseñas. Por el contrario, la autenticación por medio de redes sociales requiere que el usuario simplemente seleccione el botón de la red social y ya podría acceder a la página principal. Además, este método cuenta con una ventaja tanto en recursos como en costos, ya que el sistema de almacenamiento y de recuperación de contraseñas es administrado por las empresas de las redes sociales.

En la figura 12.30 se muestra la página de inicio de sesión de la aplicación. Se pueden observar los botones correspondientes a las redes sociales con las cuales el usuario se autentifica para poder acceder.

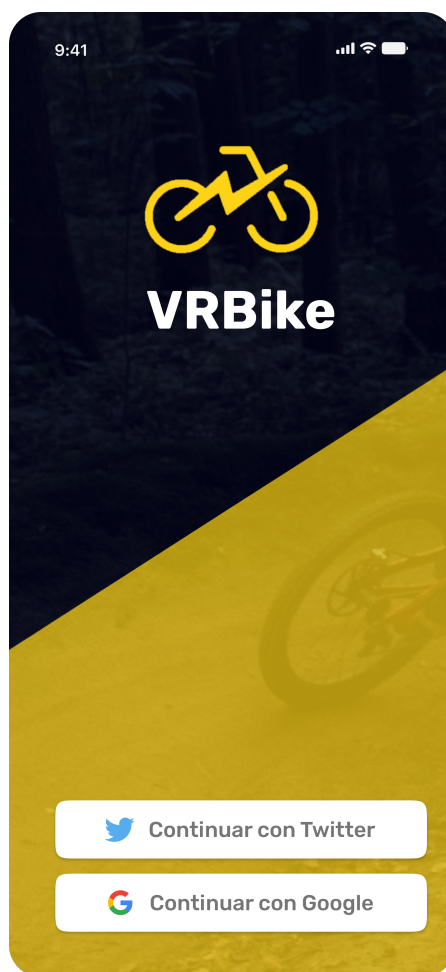


FIGURA 12.30: Página de inicio de sesión.

Una vez que el usuario inicia sesión, la red social utilizada devuelve un identificador único del

usuario. Este identificador se utiliza para guardar los datos del usuario en la base de datos y para poder vincularlo con una bicicleta al iniciar una sesión de spinning. En el caso de que el usuario no hubiera participado en ninguna sesión hasta el momento, se muestra una pantalla de inicio indicándole que debe participar en al menos una para poder visualizar sus métricas.

Visualización de las métricas del usuario La página principal de la aplicación muestra un resumen de las métricas que obtuvo el usuario en las sesiones en las que participó. En la figura 12.31 se muestra la métrica de distancia total recorrida como ejemplo de cómo son visualizados estos datos.

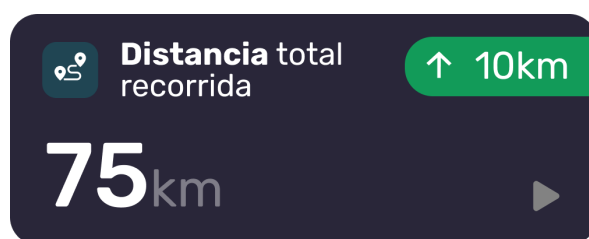


FIGURA 12.31: Métrica de distancia total recorrida.

A su vez, si el usuario de la aplicación selecciona en el resumen de una métrica en particular, puede acceder a su detalle. En la figura 12.32 se muestra una captura de la pantalla de detalle. En esta pantalla se visualiza información y gráficos adicionales a lo mostrado en la pantalla de resumen. Entre la información adicional se encuentra la fecha en la que se registró el último valor de la métrica y la descripción de su significado. Por otra parte, se incluyen distintos gráficos dependiendo de la métrica analizada. En el caso de la velocidad, un gráfico muestra la evolución del valor promedio sesión a sesión y otro gráfico muestra la evolución del valor máximo alcanzado en cada sesión. En cada uno de estos gráficos el color verde representa el máximo global y en color rojo, el mínimo. Además, si se mantiene presionado sobre una de las barras, aparece un indicador de la fecha en la que se registró dicho valor.



FIGURA 12.32: Pantalla de detalle de métrica.

A continuación se listan todas las métricas mostradas en la aplicación.

- Velocidad promedio.

Esta métrica refleja la velocidad en revoluciones por minuto que alcanzó el usuario en promedio durante la última sesión en la que participó. Además, se incluye un indicador del porcentaje de mejora o de empeoramiento de dicha sesión con respecto al promedio de las últimas 7 sesiones. En la figura 12.33 se muestra la forma en la que se visualiza esta métrica en la aplicación.



FIGURA 12.33: Métrica de velocidad promedio.

- Distancia total recorrida.

Este valor indica el total de la distancia (en kilómetros) que recorrió el usuario como suma de todas las sesiones en las que participó. Adicionalmente, un indicador marca el aumento que se realizó en la última sesión con respecto al valor anterior. En la figura 12.31 se muestra esta métrica.

- Umbral funcional de potencia promedio.

Este valor es el máximo funcional de potencia (FTP) en Watts del usuario. El FTP indica la máxima potencia que un ciclista es capaz de mantener durante un largo periodo de tiempo sin fatigarse.

El FTP se puede calcular mediante pruebas realizadas en gimnasios especializados, o bien estimarlo a partir de las características físicas del usuario (edad, masa y sexo). Al presionar sobre esta métrica, el usuario puede acceder a un formulario que permite modificar sus características físicas cargadas anteriormente para que el FTP se vuelva a estimar. A fin de estimarlo, se utiliza una fórmula estándar en la industria que consiste en duplicar la masa en kilogramos, restar el 0.5 % de cada año de edad a partir de los 35 años y, por último, restar un 10 % del valor resultante en el caso de que el sexo sea femenino. Cabe destacar que si el usuario contara con el valor exacto de FTP debido a haberlo calculado en una prueba en un gimnasio, puede cargarlo directamente en la aplicación. En la figura 12.34 se incluye una captura de esta métrica.

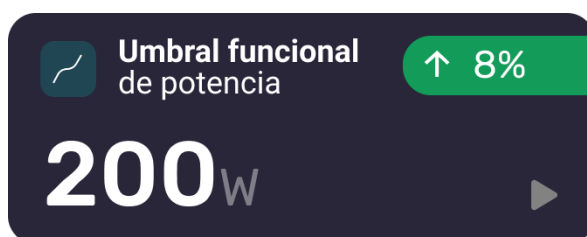


FIGURA 12.34: Métrica del umbral funcional de potencia.

En este caso, el indicador de evolución indica el porcentaje en que varió el último FTP registrado con respecto al anterior. Un aumento en dicho parámetro indica un aumento en la capacidad física del ciclista. Cabe destacar que si el usuario no hubiera ingresado ni los datos de sus características físicas ni el valor exacto de su FTP, se le asigna internamente el valor típico de 220W hasta que el usuario lo actualice.

- **Potencia promedio.**

La potencia promedio refleja el promedio de la potencia alcanzada durante la última sesión. La unidad seleccionada es porcentaje del máximo funcional del usuario. En la figura 12.35 se visualiza esta métrica mostrada en la aplicación. Se puede analizar que también incluye el indicador de la evolución de la última sesión con respecto al promedio de las últimas 7 sesiones.

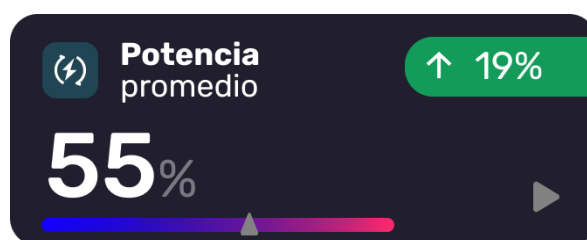


FIGURA 12.35: Métrica de potencia promedio.

A diferencia de las métricas descritas anteriormente, incluye una barra con el nivel de potencia (en a la escala de niveles conocidos como 'Coggan') que corresponde al valor alcanzado. Dicha escala es un estándar en la industria del ciclismo indoor y permite establecer distintas categorías en relación con el umbral funcional de potencia (FTP) del ciclista. La escala de niveles Coggan se divide en 7 zonas de entrenamiento de acuerdo al porcentaje de relación entre la potencia alcanzada y la de umbral. En la figura 12.2 se describe cada una las zonas, junto con el porcentaje del FTP al que corresponden.

Zona	Descripción	% FTP
1	Recuperación activa	<55
2	Resistencia	55-75
3	Tempo	76-91
4	Umbral del Lactato	91-105
5	VO ₂ máx.	106-120
6	Capacidad anaeróbica	121-150
7	Neuromuscular	>150

TABLA 12.2: Escala de niveles Coggan.

Así, VRBike registra la cantidad de tiempo en la que cada ciclista se desempeña en cada una de estas zonas y establece su promedio a lo largo de toda la sesión. Un valor más alto de este parámetro con respecto a su valor anterior está relacionada con una mejora en el entrenamiento llevado a cabo.

Visualización de las últimas sesiones asistidas Esta funcionalidad de la aplicación permite acceder en la pantalla 'Mis Sesiones' al listado las últimas sesiones en las que participó el usuario. En la figura 12.36 se incluye una captura de esta pantalla. En cada entrada de esta lista se resumen algunos datos de la sesión, tales como el gimnasio en el que se realizó, la fecha, la hora y la duración.



FIGURA 12.36: Listado de sesiones asistidas.

Si el usuario presiona sobre una de las sesiones mostradas en la lista de la pantalla 'Mis Sesiones', accede al detalle de dicha sesión. Este detalle consiste de tres pestañas, descritas a continuación.

- Info

Esta pestaña muestra valores generales tales como la fecha, la hora, la duración, la cantidad de participantes y la sede de la sesión. Una captura de esta pestaña se enseña en la figura 12.37.

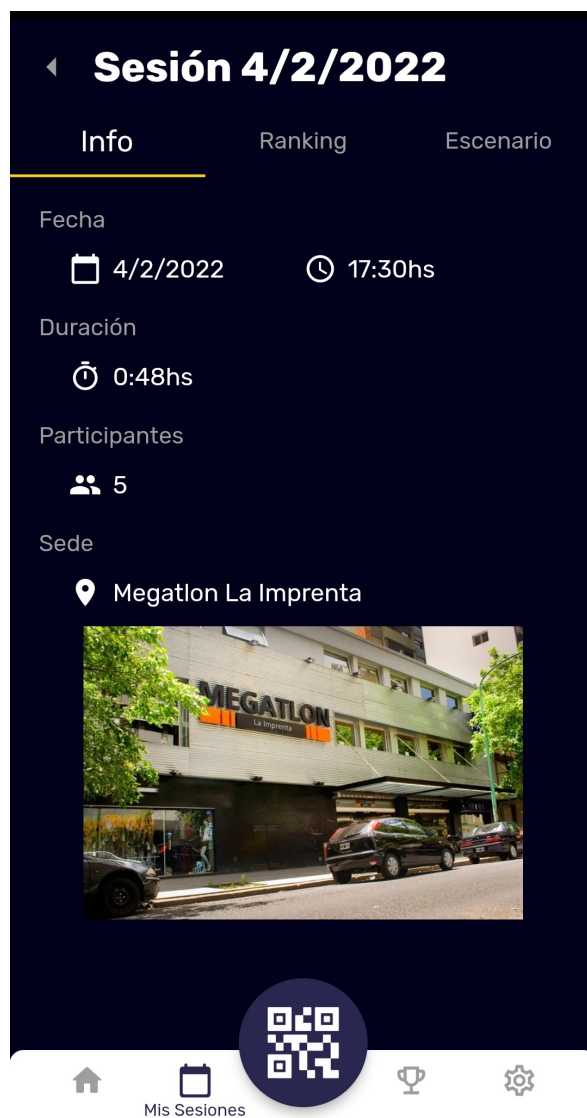


FIGURA 12.37: Pestaña de información de la pantalla de detalle de la sesión.

■ Ranking

Esta pestaña incluye un listado con todos las personas que participaron de la sesión. A dichos participantes se los ordena según las métricas que hayan obtenido, como la potencia promedio y la velocidad promedio. Así, el usuario puede analizar cómo fue su desempeño en comparación a los demás participantes. En la figura 12.38 se muestra una captura de esta pestaña.



FIGURA 12.38: Pestaña de ranking de la pantalla de detalle de la sesión.

■ Escenario

Por último, esta pestaña describe el escenario utilizado durante la sesión. Consta del nombre del escenario (por ejemplo, Recorrido por Londres"), junto con información como la distancia total del escenario y una imagen que lo caracterice.

Visualización de los logros obtenidos En la página de 'Mis Medallas' se introduce una serie de logros predefinidos que el usuario puede alcanzar. El objetivo de estos logros es actuar a modo de desafío para que los usuarios puedan superar sus propias métricas récord. A su vez, permiten una experiencia social al poder compartir los logros obtenidos mediante sus redes. Las medallas se dividen en tres categorías:

■ General

Las medallas de esta categoría representan aquellas que no se asocian a una métrica particular

sino a la consistencia del entrenamiento del usuario. Un ejemplo de estas es la medalla 'Novato', la cual se consigue al participar en la primera sesión de VRBike.

- **Potencia**

Estas medallas están relacionadas a los valores obtenidos de la métrica de potencia. Se presentan medallas para distintos niveles que se desbloquean a medida que el usuario mejora en las métricas alcanzadas. Por ejemplo, la métrica 'Potencia Nivel 2' se desbloquea al alcanzar una potencia promedio del 70 % del FTP en una sesión de spinning.

- **Velocidad**

Las medallas relacionadas a la métrica de velocidad se incluyen en esta categoría. A modo de ejemplo, la medalla 'Velocidad Nivel 1' establece el objetivo de lograr una velocidad promedio de 60rpm en una sesión de spinning.

En la figura 12.39 se muestra una captura de la pantalla 'Mis Medallas'. En la parte superior, es posible acceder a cada una de las categorías mencionadas al seleccionar sobre su correspondiente pestaña. En color claro se representan aquellas medallas ya obtenidas, mientras que en color oscuro y con una banda diagonal, aquellas que todavía no se desbloquearon.

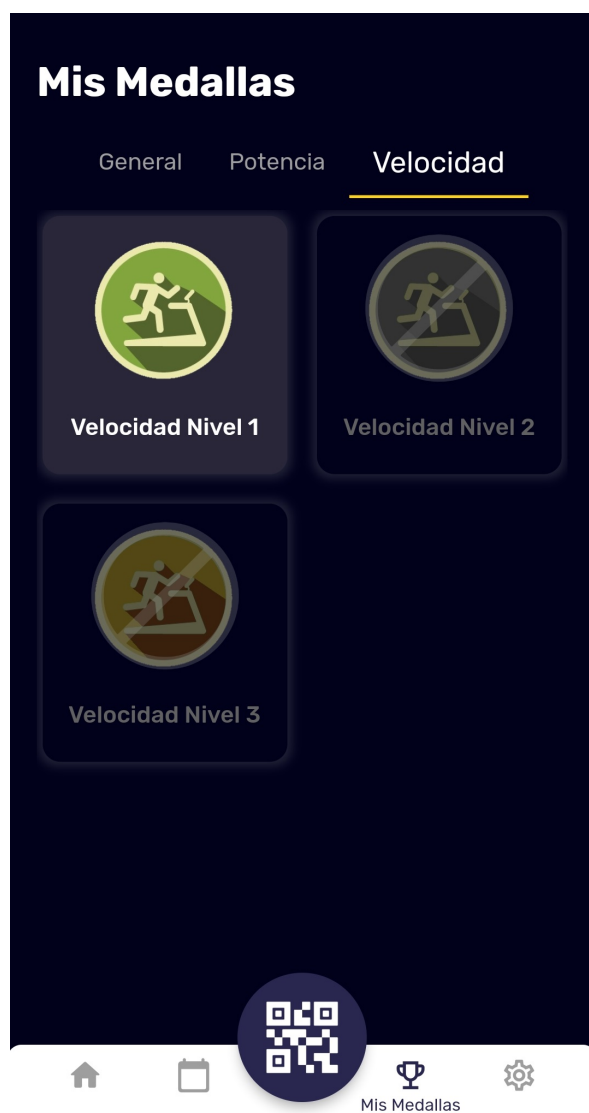


FIGURA 12.39: Pantalla de medallas.

Si se presiona sobre una de las medallas, se accede a la pantalla de detalle. En la figura 12.40 se muestra un ejemplo de la pantalla visualizada al visibilizar uno de estos logros. En ella, se muestra el objetivo necesario para desbloquearla, junto con la fecha en la que lo logró y un botón que permite compartir el logro mediante redes sociales.

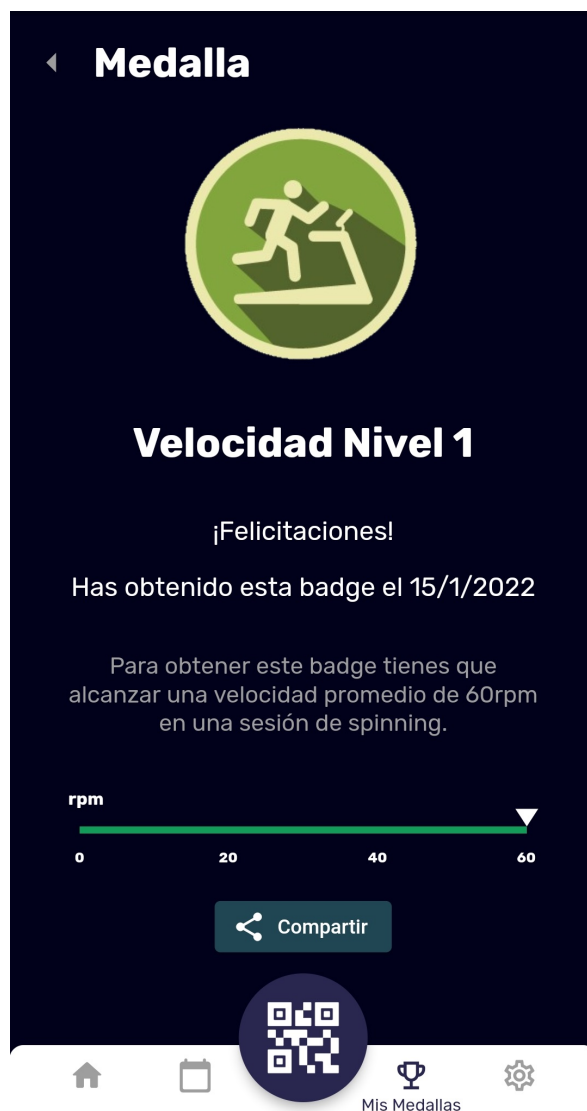


FIGURA 12.40: Medalla obtenida.

Por el contrario, en la figura 12.41 se visualiza el detalle de una medalla que todavía no fue desbloqueada por el usuario. Al igual que el detalle de la medalla obtenida, indica los requisitos para obtenerla junto con una barra de progreso.

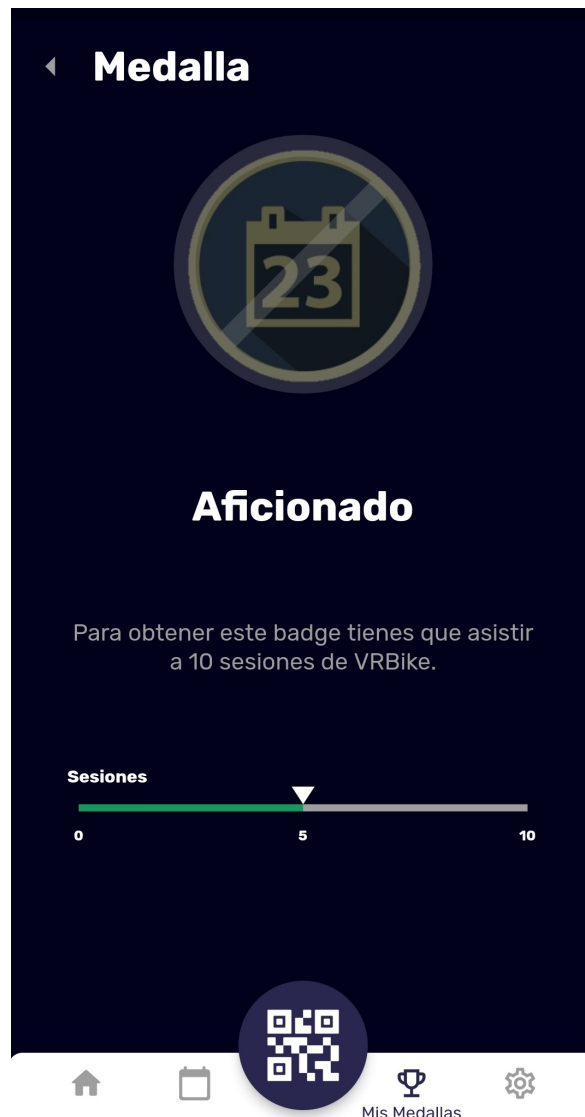


FIGURA 12.41: Medalla no obtenida.

Vinculación con una bicicleta VRBike Dado que la aplicación móvil describe métricas, detalles y logros obtenidos por los usuarios, es necesario definir una forma de detectar las sesiones en las que participaron. Es decir, se debe poder vincular las métricas que recolecta una bicicleta VRBike con un usuario determinado de la aplicación.

A fin de maximizar la experiencia del usuario, se priorizó que este proceso involucre la menor cantidad de pasos posible para vincularse. Luego de analizar distintas opciones mencionadas anteriormente, se seleccionó la vinculación por medio de un código QR presente en cada una de las bicicletas VRBike del gimnasio. Como resultado de este método, el usuario simplemente debe seleccionar un botón de la aplicación (ícono de código QR) y escanear con su cámara el código de la bicicleta que va a utilizar durante la sesión. Cabe destacar que este botón es el de mayor tamaño en la barra de botones y, por lo tanto, es detectable con facilidad. En la figura 12.42 se visibiliza un ejemplo de lo que vería el usuario al escanear el código QR.



FIGURA 12.42: Escaneo de QR.

Una vez escaneado, el usuario se vincula automáticamente sin requerir ningún paso adicional. En la figura 12.43 se muestra la pantalla que indica que el usuario se vinculó correctamente con una bicicleta.

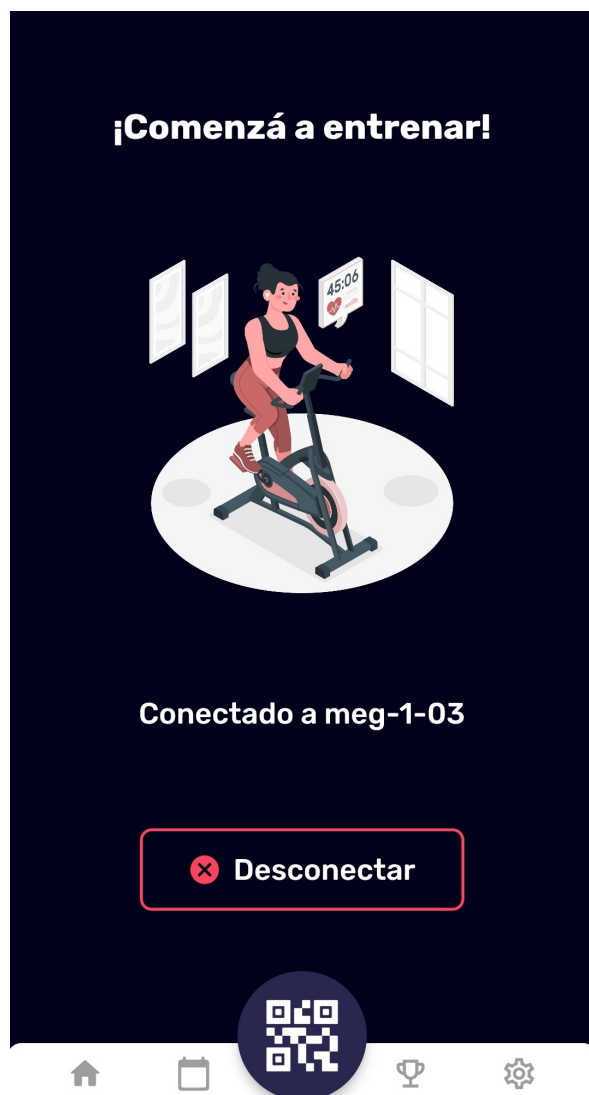


FIGURA 12.43: Usuario vinculado.

Internamente, el código QR le devuelve a la aplicación un código identificador único de la bicicleta VRBike. Por otra parte, la red social utilizada para autenticarse permite acceder a un identificador, también único, del usuario de la aplicación. Luego de obtenidos, la aplicación vincula estos dos identificadores en la base de datos en la nube. De esta forma, el servidor puede detectar que se encuentran vinculados y cargar las métricas logradas al usuario correspondiente al finalizar la sesión.

Cabe mencionar que la vinculación con la bicicleta es opcional. Si el usuario no se vincula, realizará el ejercicio bajo el nombre de 'Invitado' y las métricas no se reflejarán en la aplicación.

Comunicación En la figura 12.44 se muestran las funcionalidades de la aplicación del usuario y sus conexiones con el resto del sistema. Se aprecia que el único módulo con el que interactúa es con la base de datos de Google Firebase. Esto es así ya que es donde se encuentran almacenadas las métricas de los usuarios y es donde se establece la vinculación entre un usuario y una bicicleta.

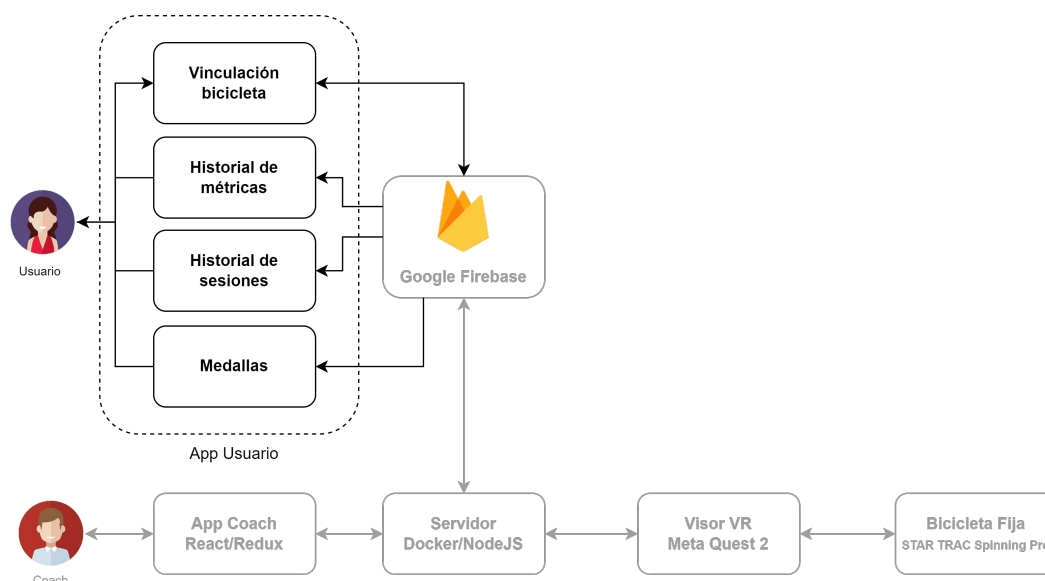


FIGURA 12.44: Gráfico esquemático que muestra las comunicaciones entre la aplicación del usuario y las demás partes del sistema

Subrutinas El proyecto de la aplicación del usuario cuenta con una estructura interna dividida de acuerdo a las funcionalidades requeridas. En cada parte, presenta subrutinas que establecen el comportamiento de la aplicación. A continuación, se listan las subrutinas más relevantes.

- **isUserLoggedIn:** Subrutina llamada al inicializar la aplicación que devuelve si el usuario ya inició sesión o bien debe realizarlo. Permite definir si se debe acceder a la pantalla de inicio de sesión o directamente a la pantalla de métricas.
- **signInWithGoogle:** Subrutina ejecutada cuando el usuario decide iniciar sesión mediante sus credenciales de Google.
- **signInWithTwitter:** Subrutina ejecutada cuando el usuario decide iniciar sesión mediante sus credenciales de Twitter.
- **signOut:** Subrutina utilizada para finalizar la sesión del usuario.
- **getUserMetricSnippets:** Subrutina que se conecta con la base de datos de Google Firebase para obtener el listado de los resúmenes de las métricas del usuario. Estos resúmenes son los mostrados en la pantalla de métricas, por ejemplo el de la figura 12.31.
- **getMetricDetails:** Subrutina que obtiene de la base de datos los detalles de una métrica particular. Por ejemplo, para el caso de la velocidad se obtienen la información que se muestra en la figura 12.32.
- **getUserSessions:** Subrutina que le solicita a la base de datos el listado de las sesiones en las que el usuario participó. Esta subrutina permite obtener los datos necesarios para la pantalla de la figura 12.36.

- `getSession`: Subrutina que obtiene, también de la base de datos de Google Firebase, los detalles de una sesión particulares, a mostrar en las pantallas de las figuras 12.37 y 12.38 y de la pestaña de escenario.
- `getUserBadges`: Subrutina que obtiene el listado de las medallas disponibles y, de ellas, cuáles ya obtuvo el usuario y cuáles le restan por desbloquear. De esta forma, se puede completar la pantalla 'Mis Medallas', mostrada en la figura 12.39.
- `build`: Cada pantalla de la aplicación se representa en Flutter (el entorno utilizado) mediante componentes. Por ejemplo, cada medalla de la pantalla de medallas es una instancia del componente 'BadgeCard'. Asimismo, cada sesión del listado de sesiones asistidas es una instancia del componente 'SessionListTile'. Flutter establece que cada uno de los componentes debe sobrecribir una subrutina 'build', en la que se programa la interfaz que toma dicho componente dependiendo de sus parámetros.
- `initState`: Es la subrutina que se ejecuta por única vez cuando la instancia del componente se muestra por primera vez en la pantalla. Se utiliza para hacer llamadas a bases de datos o inicializar otros elementos, tales como controladores, antes de que se renderice el componente.
- `dispose`: Subrutina ejecutada cuando el componente se elimina de la pantalla y que permite limpiar los recursos que utilizaba.
- `analyzeQr`: Subrutina que, una vez detectado un código QR para vincularse con una bicicleta, determina si es un código válido de VRBike. En el caso en que esto sea cierto, llama a la subrutina 'associateUser'.
- `associateUser`: Subrutina que vincula en la base de datos el usuario que utiliza la aplicación móvil con la bicicleta escaneada.
- `disassociateUser`: Subrutina que se ejecuta si el usuario decide cancelar su vinculación con una bicicleta.

Complejidad La complejidad computacional con la que se desempeñan la mayor parte de las subrutinas es $O(n)$, siendo n el número de datos. Esto se debe a que la base de datos de Google Firebase se encuentra estructurada a fin de reducir el cálculo que se debe realizar en la aplicación móvil. Así, los datos obtenidos de la base de datos ya se encuentran listos para ser mostrados. De esta forma, la complejidad depende simplemente de la cantidad de datos que se muestren, algo que no resulta problemático, ya que se cuenta con un número limitado de ellos. Dicho límite, por ejemplo, en el caso de las métricas, corresponde a los valores de las últimas sesiones.

La subrutina de mayor complejidad es la que ordena a los participantes de una sesión en el ranking de acuerdo a sus métricas. Este ordenamiento se realiza mediante el algoritmo de ordenamiento de burbuja, que conlleva una complejidad de $O(n^2)$, donde n es la cantidad de participantes. Sin embargo, al limitarse el número de participantes a la cantidad de bicicletas que caben en una sala

de spinning (alrededor de 15), la complejidad no resulta problemática para los teléfonos móviles actuales.

Plan de Pruebas El plan de pruebas de la aplicación móvil está conformado por distintas etapas. En primer lugar, se realizan pruebas unitarias sobre las subrutinas de la aplicación. Estas pruebas permiten verificar que las subrutinas cumplan su funcionalidad ante distintas condiciones que se imponen para cada una de ellas. Por otro lado, el entorno utilizado, Flutter, permite realizar pruebas sobre los componentes que conforman la aplicación. Estas pruebas determinan si los componentes se visualizan correctamente y si se interactúa con ellos de la forma esperada. Luego, las pruebas de integración realizadas sobre la aplicación completa indican si los distintos componentes y subrutinas interactúan conjuntamente de forma adecuada.

Además de las pruebas unitarias, de componentes y de integración, se realiza otra etapa de pruebas al momento de lanzar la aplicación a la tienda de aplicaciones. En primer lugar, la aplicación es lanzada en una versión alfa. Esta es la última de las pruebas internas realizadas y la realizan tanto el propio equipo de desarrollo como integrantes de los potenciales clientes. En este caso, estos últimos son los empleados de los gimnasios en los que se busca incorporar VRBike. Así, dichos empleados, tales como entrenadores y directivos, pueden reportar errores encontrados antes de que la aplicación sea lanzada a los verdaderos usuarios. Una vez finalizada la etapa de la prueba alfa, se lanza la versión beta de la aplicación. Esta fase involucra la prueba por parte de grupos de usuarios que no forman parte del equipo de desarrollo ni de los clientes, sino, en el caso de VRBike, a participantes de sesiones de spinning. Este grupo prueba la aplicación en su versión preliminar y detecta errores a ser corregidos antes de la versión de producción. Cabe destacar que las tiendas de aplicaciones, tanto de Android como de iOS, incluyen la administración de las versiones alfa, beta y producción de una forma muy sencilla, pudiendo configurar sus duraciones, los errores reportados y los usuarios que pueden acceder a ellas.

12.2.6. Base de datos

La base de datos es otro de los subsistemas desarrollados. Se encuentra desplegada en la nube en el ambiente de 'Backend como servicio' llamado Google Firebase. Dentro de este ambiente se creó una instancia de una base de datos para guardar la información correspondiente a las métricas de cada usuario de VRBike.

La base de datos utilizada cuenta con la particularidad de ser no relacional (o 'NoSQL') y estar basada en un modelo de documentos y colecciones de documentos. El hecho de ser no relacional involucra que, a diferencia de bases de datos tradicionales, la información puede ser almacenada de forma flexible sin necesidad de mantener un esquema definido por única vez. De esta forma, brinda el beneficio de ser más escalable al poder añadir nuevos campos en el futuro. En nuestro caso, esto se podría ver reflejado en la adición de una nueva métrica a registrar. Además, este tipo de bases de datos es más eficiente en cuanto a los recursos utilizados, ya que permite adaptar los documentos guardados a una serie variable de campos y esto evita almacenar campos vacíos.

Por otro lado, Firebase ofrece una moderna e intuitiva interfaz de usuario que posibilita acelerar el tiempo de desarrollo. Con respecto a la integración con el resto del sistema, Firebase brinda SDKs para Android, iOS, Web y Node.js, entre otros. De esta forma, se simplifica el desarrollo de la comunicación con el resto de los módulos.

Debido a que el modelo de pagos de Firebase funciona bajo la modalidad de costo por llamadas realizadas y a que cada llamada introduce un tiempo de carga en la búsqueda de los datos, se decidió utilizar un sistema diseñado a fin de minimizar la cantidad de llamadas a Firebase desde la aplicación móvil de métricas. Siguiendo las buenas prácticas del diseño de bases de datos, se optó por implementar una base de datos en la que en cada llamada se obtenga toda la información relevante al contenido de una página de la app. De esta manera, se logra disminuir la cantidad de llamadas a la base de datos y evitar demoras molestas por parte del usuario, ya que solamente debe realizar el pedido de información nueva en cada cambio de página, algo no tan frecuente en nuestra aplicación. Así, se logra reducir los costos operativos del proceso de almacenamiento de los datos y se mejora la experiencia de usuario al reducir el tiempo de espera durante el uso de la aplicación móvil.

Entrando en más detalle en cuanto a la organización de los datos, las colecciones en Firebase permiten, dentro de la misma base de datos, organizar a los documentos que se almacenan. Cada documento utiliza el formato json y, dentro de estos, se pueden guardar campos con un tipo de dato (a elección entre número, cadena de caracteres, mapa, arreglo, etc.) mediante un conjunto de clave-valor.

A continuación, se enumeran las colecciones creadas y una descripción de los campos almacenados en cada documento. Cabe destacar que a diferencia de las bases de datos tradicionales, en Google Firebase, al ser una base de datos no relacional, es una buena práctica repetir la información en distintas partes de la base de datos. El objetivo de este mecanismo es reducir la cantidad de llamadas necesarias para mostrar información en las interfases de usuario.

1. **userMetrics:** Colección que almacena el documento de cada usuario con la información de los resúmenes de sus métricas. Estos resúmenes guardan la información que se muestra en la aplicación móvil, por ejemplo, en la figura 12.31. Campos almacenados en los documentos:
 - **metricSnippets:** Resúmenes de métricas. Tipo: arreglo. Cada valor de este arreglo es de tipo mapa y representa una métrica determinada. Cada uno de estos mapas guarda:
 - **name:** Nombre de la métrica almacenada. Tipo: string.
 - **metricId:** Identificador de la métrica. Tipo: string.
 - **description:** Descripción de la métrica almacenada. Tipo: string.
 - **currentValue:** Valor actual de la métrica. Tipo: número.
 - **units:** Unidades en las que se mide la métrica. Tipo: string.
 - **secondaryCurrentValue** (opcional): Valor actual de la métrica medido en otra unidad. Tipo: número.

- **secondaryUnits** (opcional): Unidades secundarias en las que se mide la métrica. Tipo: string.
 - **iconCodePoint** (opcional): Id del ícono que representa a la métrica. Tipo: número.
 - **priority**: Prioridad de la métrica que permite ordenar la lista de métricas de acuerdo a su importancia -a mayor valor, menor importancia-. Tipo: número.
 - **isEvolutive**: Indicador de si la métrica es evolutiva, es decir, si interesa la variación que tome con respecto a sus valores anteriores. Tipo: boolean.
 - **evolutionData** (opcional): Información sobre la evolución de la métrica con respecto a sus valores anteriores. Tipo: mapa. Campos:
 - **description**: Descripción que explica el significado de la evolución. Tipo: string.
 - **evolution**: Valor de la evolución. Tipo: número.
 - **units**: Unidad en la que se mida la evolución. Tipo: string.
 - **isCategorizable**: Indicador de si la métrica se divide en rangos numéricos. Tipo: boolean.
 - **categoryData** (opcional): Información sobre la división en rangos numéricos. Tipo: mapa. Campos:
 - **scaleName**: Nombre de la escala de rangos numéricos utilizados. Tipo: string.
 - **scaleCode**: Identificador de la escala de rangos numéricos utilizados. Tipo: string.
 - **categoryRanges**: Rangos numéricos. Tipo: arreglo. Cada valor de este arreglo es un mapa que guarda:
 - ◇ **levelCode**: Valor que permite ordenar los niveles. Tipo: número.
 - ◇ **min**: Valor mínimo del rango numérico. Tipo: número.
 - ◇ **max**: Valor máximo del rango numérico. Tipo: número.
2. **metricDetails**: Subcolección de cada documento de la colección **userMetrics** que almacena los valores detallados de cada métrica, en un documento para cada una de ellas. A partir de esta subcolección se genera la pantalla de detalles mostrada en la figura 12.32. Cada documento de esta subcolección guarda los siguientes campos:
- **name**: Nombre de la métrica. Tipo: string.
 - **units**: Unidad en la que se mide la métrica. Tipo: string.
 - **values**: Valores alcanzados de la métrica en cada sesión. Tipo: arreglo. Cada valor de este arreglo representa una sesión y es de tipo mapa. Cada mapa guarda:
 - **dateTime**: Fecha y hora de la sesión. Tipo: timestamp.
 - **min**: Mínimo valor alcanzado en la sesión. Tipo: número.
 - **max**: Máximo valor alcanzado en la sesión. Tipo: número.
 - **value**: Valor promedio durante la sesión. Tipo: número.
 - **snippets**: Resúmenes detallados de la métrica. Tipo: arreglo. Cada uno de los valores del arreglo es un resumen similar a los de la colección **userMetrics**, pero con los detalles

adicionales que se muestran en la pantalla de detalles (como fecha y descripción). En general, solamente se necesita 1 resumen, pero se estableció que se almacene en un arreglo para tener la posibilidad de añadir más en el futuro.

3. **sessions**: Colección que almacena todas las sesiones llevadas a cabo, una en cada documento. En cada documento se incluyen los campos:

- **dateTime**: Fecha y hora de la sesión. Tipo: timestamp.
- **durationInSeconds**: Duración de la sesión en segundos. Tipo: número.
- **gym**: Información del gimnasio donde se realizó la sesión. Tipo: mapa. Campos:
 - **name**: Nombre del gimnasio. Tipo: string.
 - **location**: Ubicación geográfica del gimnasio. Tipo: string.
 - **imageUrl**: URL de la imagen del gimnasio. Tipo: string.
- **scene**: Información del escenario que se utilizó durante la sesión. Tipo: mapa. Campos:
 - **name**: Nombre del escenario. Tipo: string.
 - **mapImageUrl**: URL de la imagen del escenario. Tipo: string.
 - **previewImageUrl**: URL de la imagen de la ruta del escenario. Tipo: string.
- **sessionPlayers**: Participantes de la sesión. Tipo: arreglo. Cada valor del arreglo es de tipo mapa y corresponde a un participante. Cada mapa guarda:
 - **id (opcional)**: Identificador interno del usuario, definido si el usuario estuviera autenticado. Tipo: string.
 - **name**: Nombre del participante. Toma el valor de 'invitado' si el participante no estuviera autenticado. Tipo: string.
 - **imageUrl (opcional)**: URL de la imagen de perfil del usuario, si estuviera autenticado. Tipo: string.
 - **meanPower**: Valor promedio de la potencia en Watts alcanzada por el participante. Tipo: número.
 - **meanRpm**: Valor promedio de la velocidad en rpm alcanzada por el participante. Tipo: número.

4. **userSessions**: Colección que guarda los resúmenes de las sesiones en las que participó cada usuario, en un documento por cada usuario. Esta información permite mostrar el listado de sesiones de la figura 12.36. Cada documento está definido por:

- **sessions**: Sesiones en las que participó el usuario. Tipo: arreglo. Cada valor de este arreglo es de tipo mapa y corresponde a una sesión. Cada mapa guarda:
 - **sessionId**: Identificador de la sesión. Tipo: string.
 - **dateTime**: Fecha y hora de la sesión. Tipo: timestamp.
 - **gymName**: Nombre del gimnasio en el que se llevó a cabo la sesión. Tipo: string.
 - **durationInSeconds**: Duración de la sesión en segundos. Tipo: número.

5. `userBadges`: Colección que almacena las medallas que se le presentan a cada usuario, en un documento por usuario. Cada documento incluye:

- `badges`: Categorías en las que se dividen las medallas. Tipo: mapa. Campos:
 - General: Medallas correspondientes a la categoría 'General'. Tipo: arreglo. Cada valor del arreglo es un mapa que corresponde a una medalla. Cada mapa cuenta con:
 - `id`: Identificador de la medalla. Tipo: string.
 - `name`: Nombre de la medalla. Tipo: string.
 - `description`: Descripción de los requisitos para obtener la medalla. Tipo: string.
 - `image`: URL de la imagen de la medalla. Tipo: string.
 - `progress`: Progreso del objetivo para obtener la medalla. Tipo: mapa. Campos:
 - ◇ `objective`: Valor que se debe alcanzar. Tipo: número.
 - ◇ `currentCount`: Valor actual alcanzado por el usuario. Tipo: número.
 - ◇ `progressUnit`: Unidad en la que se mide el objetivo. Tipo: string.
 - ◇ `obtained`: Valor que indica si el usuario desbloqueó la medalla. Tipo: boolean.
 - ◇ `achievementDate` (opcional): Fecha en la que se obtuvo la medalla. Tipo: timestamp.
 - Potencia: Medallas correspondientes a la categoría 'Potencia'. Tipo: arreglo. Cada valor del arreglo es un mapa que corresponde a una medalla. Cada mapa sigue la misma estructura que las medallas de la categoría 'General'.
 - Velocidad: Medallas correspondientes a la categoría 'Velocidad'. Tipo: arreglo. Cada valor del arreglo es un mapa que corresponde a una medalla. Cada mapa sigue la misma estructura que las medallas de la categoría 'General'.

Google Firebase permite trabajar con los datos de la base de datos de distintas formas. Como se mencionó, incluye SDKs para poder interactuar con los subsistemas de VRBike y, además, una interfaz de usuario en la que se pueden crear colecciones y documentos visualmente. Así, resulta más sencillo definir la estructura de la base de datos. En la figura 12.45 se muestra uno de los documentos de la colección 'sessions' de la base de datos creada.

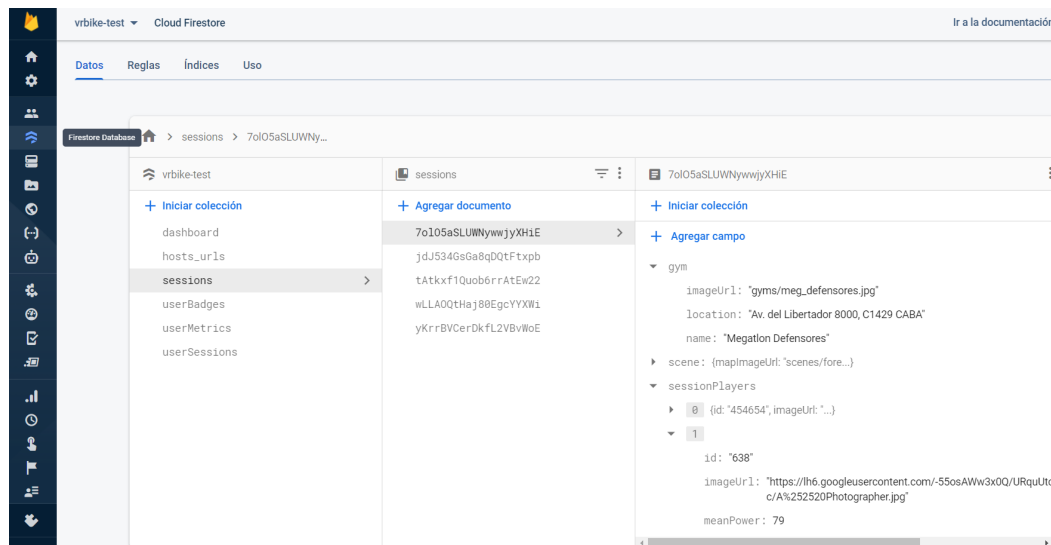


FIGURA 12.45: Base de datos.

El ejemplo mencionado permite entender el mecanismo mencionado a fin de minimizar la cantidad de llamadas a Firebase. En este caso, una vez que el usuario accede desde su aplicación a una sesión en particular del listado completo de las sesiones en las que participó, se realiza una única llamada a Firebase que devuelve la información completa de esta sesión. Así, se evita realizar múltiples llamadas en una misma página de la aplicación móvil de métricas y se evita que la aplicación móvil realice un procesamiento complejo.

Además de los datos enumerados anteriormente, Firebase cuenta con un componente, Realtime Database, que permite detectar cambios en los datos en tiempo real. Se hace uso de este para guardar la vinculación entre un usuario y una bicicleta, dado que el servidor requiere reaccionar con rapidez ante un nuevo usuario vinculado o desvinculado. Para ello, se creó una instancia de Realtime Database y, en un mapa llamado 'spinningRooms' se almacenan, como claves, los identificadores de cada gimnasio (por ejemplo, 'meg01'). Dentro de cada gimnasio se guarda el campo 'associatedBikes' que consiste de un arreglo de mapas, donde cada uno de ellos corresponde a una bicicleta asociada. La clave de estos mapas es el id de la bicicleta asociada (por ejemplo, 'meg-1-04') y, en su valor, guarda los campos 'uid', referido al identificador único del usuario; 'name', es decir, el nombre del usuario; y, 'imageUrl' que guarda, opcionalmente, la URL de la imagen de perfil del usuario. A fin de clarificar esta estructura, se incluye en la figura 12.46 una captura de ella.

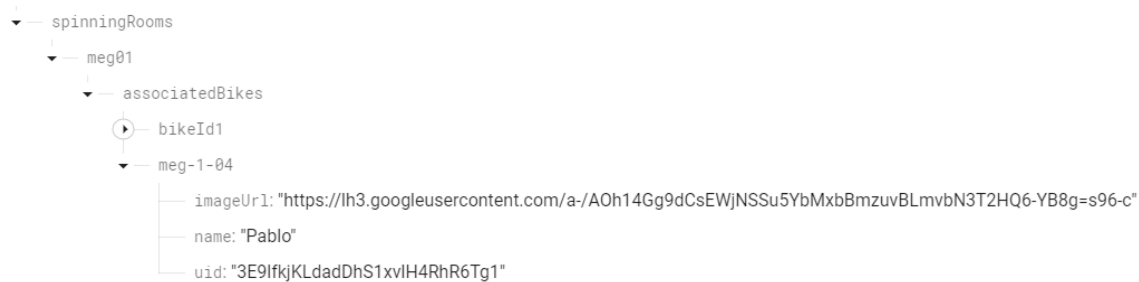


FIGURA 12.46: Almacenamiento de la vinculación usuario-bicicleta.

Subrutinas Si bien la base de datos por sí misma no ejecuta subrutinas, ya que su función es meramente la de almacenar, los SDKs de Firebase, que permiten conectar la base de datos con los distintos módulos, brindan diversas funcionalidades. A continuación, se listan las subrutinas utilizadas de los SDKs.

- **getDocument:** Subrutina que devuelve el contenido de un documento en particular. Se debe indicar la colección y el identificador del documento. Es utilizada tanto por la aplicación del usuario como por el servidor para obtener los datos referidos a los usuarios.
- **setDocument:** Subrutina que permite crear o actualizar el contenido de un documento de la base de datos. Se debe indicar la colección y el identificador del documento. Es utilizada por el servidor al finalizar la sesión para actualizar las métricas de los participantes.
- **setReference:** Subrutina que permite crear una nueva entrada en el componente Realtime Database. Es utilizada por la aplicación del usuario para vincular al usuario con la bicicleta a usar.
- **removeReference:** Subrutina utilizada para eliminar una entrada de la instancia de Realtime Database. Es utilizada por la aplicación del usuario en el caso de que el usuario decida cancelar su vínculo con una bicicleta y por el servidor para desvincular a los participantes cuando finaliza la sesión de spinning.
- **onValueChange:** Subrutina que se ejecuta cada vez que cambian los datos almacenados en la instancia de Realtime Database. Es utilizada en el servidor para detectar en tiempo real los nuevos participantes vinculados a las bicicletas.

Complejidad A diferencia de una base de datos SQL donde para realizar búsquedas complejas se requiere acceder a distintas tablas y relacionarlas, Firebase utiliza un mecanismo que indexa cada campo de cada documento. Esto permite obtener los datos buscados con una complejidad computacional de $O(1)$, sin importar los parámetros de la búsqueda. Dicha característica resulta en una ventaja considerable frente a otras opciones al brindar búsquedas de alta velocidad. No obstante, si bien la búsqueda es rápida, la transferencia de la información podría ser lenta si esta es considerable. En el caso de VRBike, se priorizó que la información para cada pantalla de la aplicación pueda ser obtenida a partir de un solo documento (o dos en casos particulares). Y,

dado que cada documento almacena únicamente texto limitado por la información de un usuario particular, la cantidad de información transmitida no resulta un problema. Adicionalmente, Firebase mejora su desempeño al almacenar en la caché de los dispositivos con los que se conecta una copia de los datos de forma activa. Esto no solo aumenta la velocidad de las llamadas más realizadas, sino que hace posible obtener datos, aunque podrían estar desactualizados, sin contar con conexión a Internet.

Plan de Pruebas El plan de pruebas de la base de datos está integrado por distintas etapas. En ellas, se busca verificar que la estructura de los datos, la conectividad y el funcionamiento sean adecuados.

En la primera de las etapas del plan de pruebas, se utilizan programas que simulan los datos que guardan en la base de datos los módulos de VRBike, pero sin la necesidad de estar realizando una sesión de spinning. Esto permite almacenar una mayor cantidad de datos con mayor agilidad que si se generaran los datos en sesiones de spinning en las cuales cada una dura aproximadamente 40 minutos. Los datos generados son inspeccionados visualmente en la interfaz de Firebase para verificar que presenten consistencia según la estructura definida.

La segunda etapa consiste en verificar que los SDKs se encuentren correctamente configurados. Para realizarla, se inicializan los SDKs y se ejecutan scripts que realizan pequeñas operaciones sobre la base de datos. Luego, se verifica, nuevamente visualmente, que los datos se hayan almacenado correctamente en Firebase.

El próximo paso de pruebas es la etapa de lectura de la base de datos y su correcta interpretación. El software de la primera etapa vuelve a ser ejecutado indicando como parámetro los identificadores de usuarios de prueba. Así, se almacenan en Firebase datos relacionados a usuarios a los cuales el equipo de desarrollo tiene acceso. Luego de ser ejecutado, se inicia sesión en las interfases de usuario con las credenciales de los usuarios de prueba y se verifica que se haya podido acceder a los datos almacenados y que estos se encuentren correctamente formateados en las aplicaciones.

Por último, se realiza una serie de 5 sesiones spinning completas con todos los módulos conectados. De esta forma, se prueba la funcionalidad general de la base de datos al utilizarse en un ambiente similar al de producción.

13. Diseño de circuitos impresos

Para la puesta en práctica del prototipo se diseñó una primera versión del circuito impreso basado en módulos comerciales para poder verificar el concepto. Este es la base del circuito impreso para el producto final, en el que se busca implementar las funcionalidades de todos los módulos en una única placa impresa, que pueda ser fabricada de forma masiva.

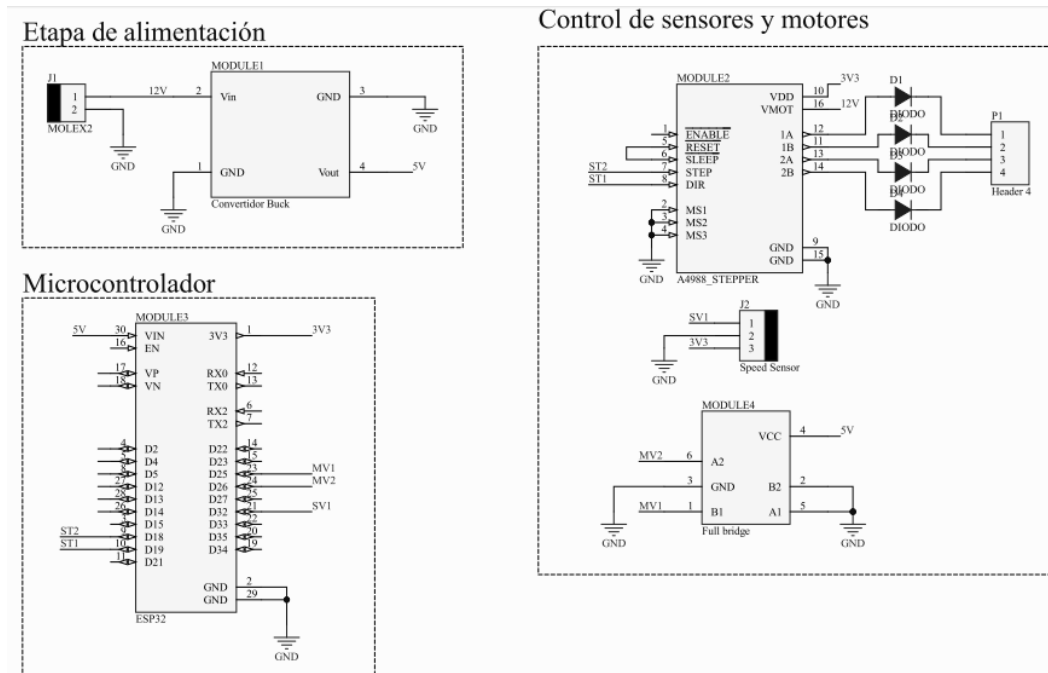


FIGURA 13.1: Esquemático del prototipo VRBike.

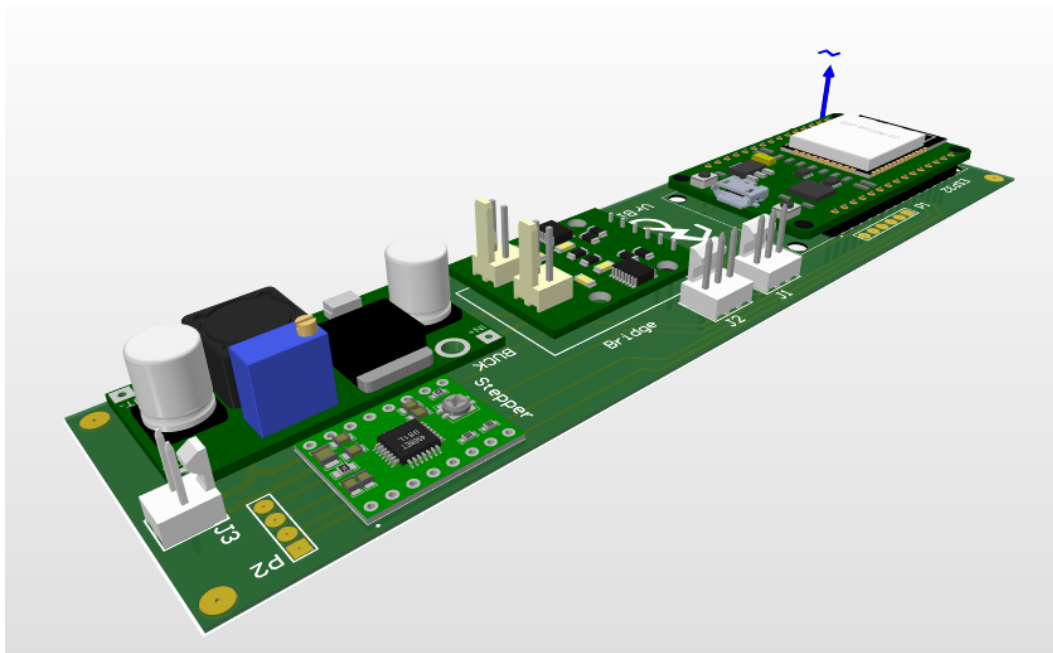


FIGURA 13.2: Modelo 3D PCB de prototipado

14. Diseño Mecánico

En esta sección se trata lo relacionado con el diseño mecánico del proyecto. La componente mecánica de la solución se encuentra en la adaptación de la bicicleta fija para poder utilizarla como controlador de la experiencia de realidad virtual. En este sentido, se destacan dos áreas dentro del diseño mecánico: el diseño del sistema de transmisión para regular la carga de la bicicleta, y la

realización de soportes carcasas protectoras para cada aspecto de intervención de la bicicleta.

14.1. Diseño del sistema de control de fuerza

En la sección 12.1.3 se explica el diseño del sistema de control de fuerza de la bicicleta. Uno de los componentes de dicho sistema es la relación de engranajes, que permite transmitir la potencia del motor a la varilla de control de fuerza provista por la bicicleta. Respecto de las características de esta relación de engranajes, se plantea lo siguiente.

- **Tipo de engranajes.** Uno de los principales requerimientos de este sistema es lograr un ruido de operación bajo, para no interferir con la experiencia de inmersión del usuario. En este sentido, se priorizaría usar engranajes helicoidales, para que el engrane entre ambos sea suave y silencioso. Por otro lado, la regulación de la fuerza implica un desplazamiento vertical de la varilla roscada que la controla, por lo que no sería posible usar los engranajes helicoidales anteriormente descritos. Luego, se eligen engranajes rectos para el sistema.
- **Relación de engranajes.** Como se describe en la sección 12.1.3, se emplea una relación de reducción de 2:1, de forma tal de aumentar la fuerza entregada a la varilla controladora de carga, en detrimento de la velocidad de cambio.
- **Tamaño de engranajes.** El tamaño de los engranajes está limitado por las dimensiones de la cavidad en donde se acomodan, como se aprecia en la figura 12.8. De esta forma, se llega a la relación de diámetros primitivos $\frac{\Phi_{p1} + \Phi_{p2}}{2} \approx 29,9mm$.

En la tabla 14.1 se muestran las características de los engranajes diseñados, considerando las características anteriormente mencionadas.

	Motor (Φ_1)	Varilla (Φ_2)
Módulo (m)	1,55mm	1,55mm
Diámetro externo (Φ_{ext})	21,7mm	40,3mm
Diámetro interno (Φ_{int})	14,725mm	33,325mm
Diámetro primitivo (Φ_p)	18,6mm	37,2mm
Número de dientes	12	24

TABLA 14.1: Características de los engranajes.

Los engranajes se realizan empleando impresión 3D plástica del tipo FDM, dado que es rápida y de bajo costo. Además, el material utilizado soporta los esfuerzos mecánicos (torsión) requeridos para esta aplicación. Los engranajes se fijan a los respectivos ejes usando tornillos, y el engranaje del motor aprovecha la chaveta provista en su eje. El resultado del diseño se aprecia en las figuras 14.1 y 14.2.

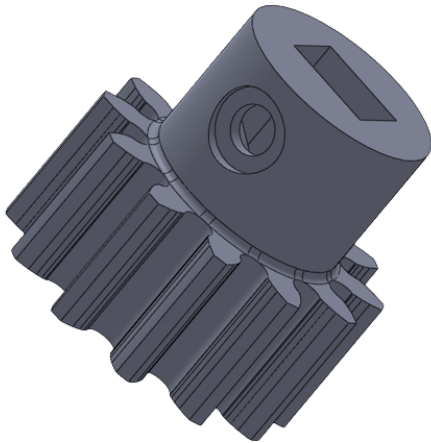


FIGURA 14.1: Engranaje del motor -
Diseño 3D

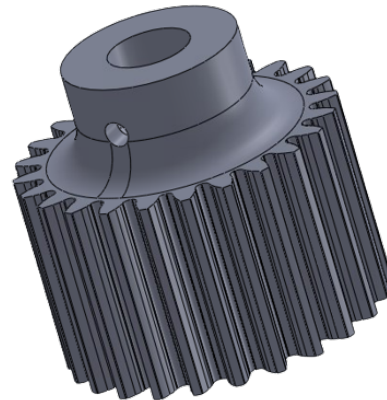


FIGURA 14.2: Engranaje de la varilla -
Diseño 3D

14.2. Diseño de soportes

En este apartado se diseñan los soportes cuya función es la de poder montar en la bicicleta el hardware necesario para la adaptación de la misma, así como proteger a los circuitos frente a golpes u otros fenómenos que pudieran dañar a los componentes. En principio, los soportes son diseñados utilizando como criterio primordial el de no afectar de forma "destructiva" a la bicicleta (por ejemplo, no hacer agujeros en la misma) para que el usuario pueda restaurar la bicicleta a su estado original si así lo deseara. Otro criterio importante para ubicar los diversos módulos de hardware es el de montarlos en lugares que no estén tan expuestos al usuario ciclista, de forma tal de protegerlos y extender su vida útil. Por último, se emplea un criterio estético para elegir la forma y lugar de montaje de cada uno de estos módulos. Se considera el diseño de soportes para los siguientes módulos.

- PCB
- Sensor de velocidad
- Motor de vibración

14.2.1. Soporte del PCB

El PCB nuclea los circuitos de control y comunicación de la solución. Es por ello que se debe elegir una ubicación de montaje que no esté al alcance del ciclista durante las sesiones. Luego, se decide colocar al PCB en la parte delantera de la bicicleta, dado que también es eficiente a nivel conexiones con los sensores y actuadores distribuidos en el cuerpo de la misma. En la figura 14.3 se observa el modelo 3D del soporte de montaje propuesto.

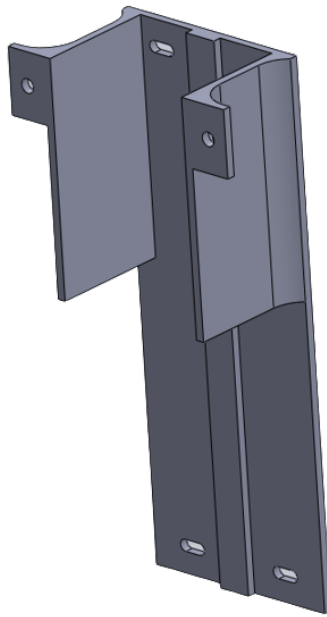


FIGURA 14.3: Soporte del PCB - Diseño 3D

Cabe destacar que el soporte anterior cuenta con una pieza rectangular que ayuda a fijar, mediante tornillos, el soporte al caño base del manubrio de la bicicleta.

14.2.2. Soporte del sensor de velocidad

El sensor de velocidad debe estar próximo al volante de inercia, y debe apuntar hacia el plano de dicho volante. En este sentido, se diseña un soporte que se ancla a uno de los caños laterales que componen la base del manubrio de la bicicleta, y se fija el sensor óptico de velocidad mediante un tornillo. El objetivo primordial del soporte es el de minimizar la distancia con el volante de inercia, de forma tal de obtener lecturas más precisas (para evitar la dispersión de los haces de luz infrarroja). La figura 14.4 refleja el diseño final del soporte.

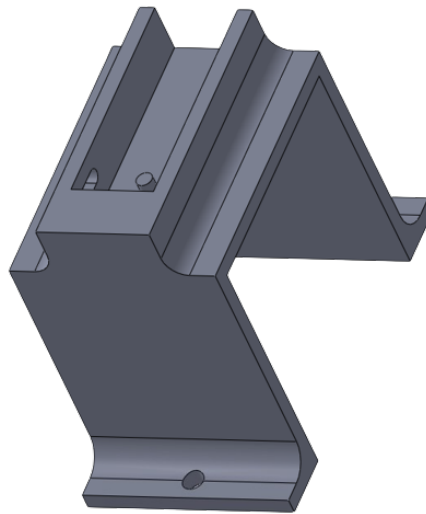


FIGURA 14.4: Soporte del sensor de velocidad - Diseño 3D

Al igual que con el soporte del PCB, en este caso se divide a la geometría en dos partes y se fija con tornillos en las pestañas diseñadas a tal fin.

14.2.3. Soporte del motor de vibración

El motor de vibración tiene la particularidad de que su ubicación en la bicicleta genera respuestas distintas en cuanto a la vibración percibida por el usuario ciclista. Se realizaron pruebas ubicando al motor en distintos lugares y se concluyó que el lugar en donde se maximiza la sensación de vibración es en el manubrio. Para elegir el lugar exacto de montaje del motor, se debe considerar que el manubrio es parte principal de contacto entre el ciclista y la bicicleta. Incluso más aún lo es en este caso, dado que el usuario tendrá ocluida su visión por el casco de realidad virtual. Es por ello que se decidió colocar al motor de vibración en la parte central del manubrio, que no está destinada para funcionar como agarre para el usuario, sino que es para alojar botellas de agua durante el entrenamiento. El caño central del manubrio presenta una sección pseudoelíptica que es considerada para el diseño del soporte. Dicho soporte consta de una abrazadera en dos partes (fijada por tornillos) y de una cápsula que contiene al motor, protegiendo a la excéntrica que genera la vibración de posibles contactos con el usuario. En las figuras 14.5 y 14.6 se observa el diseño propuesto.

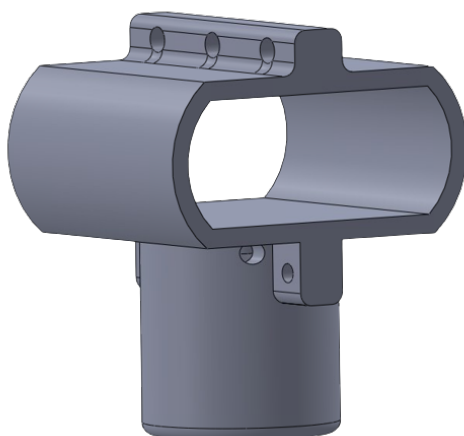


FIGURA 14.5: Soporte del motor de vibración - Diseño 3D

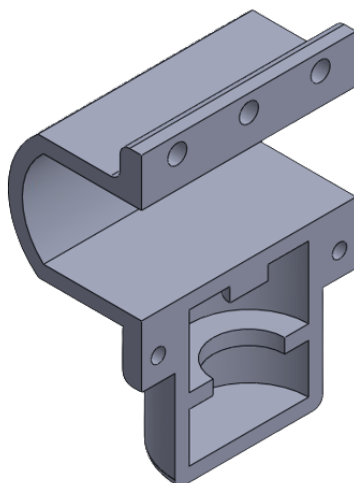


FIGURA 14.6: Soporte del motor de vibración (corte) - Diseño 3D

15. Estudio de Confiabilidad

15.1. Diagrama de confiabilidad

En el diagrama que se muestra a continuación en la figura 15.2, se pueden ver los diferentes bloques que hacen referencia a las partes o piezas del dispositivo, sin los cuales el mismo no podría funcionar adecuadamente, las cuales van en serie y en el caso de que tuvieran redundancia, las mismas irían en paralelo.

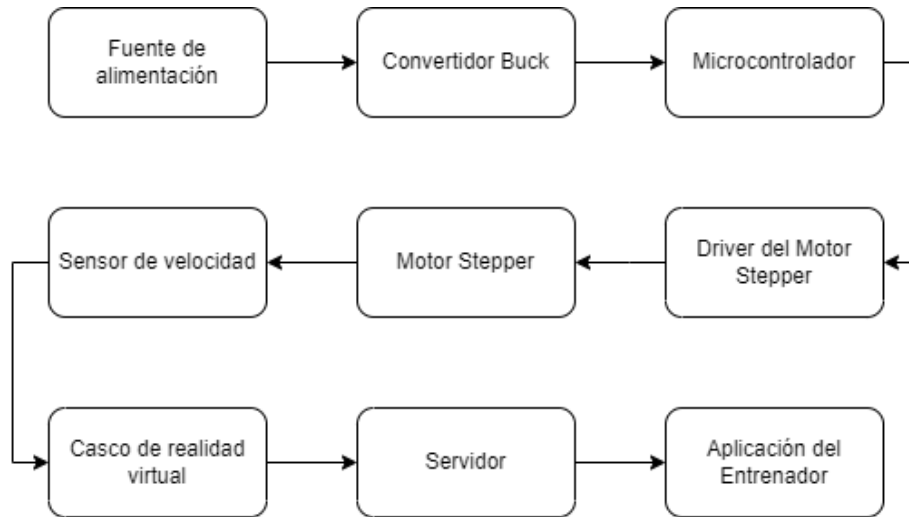


FIGURA 15.1: Diagrama de confiabilidad del sistema propuesto

De esta manera se procederá a analizar cuál es la confiabilidad de cada uno de esos módulos, obteniendo su lambda, su confiabilidad $R(t)$ y su tiempo medio entre fallas (MTBF).

15.2. Confiabilidad de Hardware

Se utilizará la norma militar MIL-HDBK-217F para el cálculo de los parámetros de confiabilidad. Se utiliza el valor λ_p como la cantidad de fallas esperada en un millón de horas.

A su vez, para la función de confiabilidad, se utiliza:

$$R(t) = e^{-\int_0^t \lambda_p(t) dt} \quad (15.1)$$

Asumiendo λ_p constante la ecuación anterior queda como:

$$R(t) = e^{-\lambda_p \frac{t}{10^6}} \quad (15.2)$$

De esta manera se obtiene el tiempo medio entre fallas (MTBF) en horas utilizando la ecuación que se muestra a continuación:

$$MTBF = \int_0^\infty R(t) dt = \frac{10^6}{\lambda_p} \quad (15.3)$$

15.2.1. Módulos bajo análisis

Para cada uno de los módulos de Hardware que se encuentran en el diagrama de bloques, se presentan las tablas con los coeficientes necesarios para calcular el λ_p , donde se asume que todos los componentes dentro de un bloque son críticos, por lo que el fallo de uno representa una falla en todo el equipo. De esta manera, la talla de fallos del módulo al completo resulta de la suma de la tasa de fallos de cada componente que lo conforma.

Convertidor Buck A continuación se muestra la tabla para el Convertidor Buck elegido.

Submódulo	Elemento	λ_b	π_T	π_E	π_Q	π_S	π_V	π_C	π_{SR}	π_{CV}	λ_p
Convertidor Buck	Capacitor	0.0004	1.1	1	10		1	0.54	0.33	1	0.0008
	Switch	0.1		1	1			1.3			0.13
	Inductor	0.00005	1.2	1	1						0.00006
	Diodo	0.0038	1.4	1	2.4	0.0054		1			0.0007
	Resistencia	0.0032		1	15						0.048

$$\lambda_{convertidor} = \sum \lambda = 0,1796 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.4)$$

Microcontrolador A continuación se muestra la tabla para el microcontrolador ESP32 elegido.

32 bits	C1 = 0,56
MOS	
Func Pins = 32	C2 = 0,00028(μP) ^{1.08} = 0.012
Tipo 1 (SMT)	
G_f	$\pi_L = 1$
Proveedor con más de dos años de experiencia	$\pi_Q = 10$
$T_j = T_c + \Theta_{jc} = 25^\circ\text{C} + 11^\circ\text{C}/\text{w} \cdot (0,2 \text{ w}) = 27,2^\circ\text{C}$	$\pi_T = 0.15$
$E_a(\text{eV}) = 0.6$	

TABLA 15.1: Coeficientes de cálculo Microcontrolador

$$\lambda_P = (C1.\pi_T + C2.\pi_E)\pi_Q.\pi_L = 1,08 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.5)$$

Fuente Con respecto a la fuente de 12 V utilizada, debido a que el fabricante no especifica el valor del MTBF correspondiente, y teniendo en cuenta que no se puede saber con certeza los componentes que se encuentran dentro del equipo porque no se tiene acceso al esquemático, se decide utilizar los valores usuales para este tipo de productos determinados por otros fabricantes para este tipo de productos.

Es así como se encontró que muchos fabricantes de fuentes similares a la utilizada en este proyecto, pronostican un MTBF de 200.000 horas, por lo que se decide tomar una medida más conservadora y considerar que la fuente empleada presenta un MTBF estimado de 100.000 horas.

De esta manera, el lambda pronosticado para la fuente es $\lambda_P = 10$

Driver del Motor Stepper El driver utilizado es el A4988 cuyos componentes principales son 8 transistores MOS de potencia en una configuración de doble *full bridge* para controlar los dos bobinados del motor paso a paso. Por lo tanto, para el cálculo de la confiabilidad del mismo se realizan los cálculos propuestos en la sección 6.4 de la norma antes mencionada, para luego aplicar la corrección por modelo híbrido correspondiente a varios dispositivos en un mismo encapsulado correspondiente a la sección 5.5.

Transistor FET en baja frecuencia	
MOSFET	$\lambda_b = 0.012$
$T_j = 100C$	$\pi_T = 3.7$
$5W < P < 50W$	$\pi_A = 4.0$
Factor de calidad	$\pi_Q = 1$
Factor ambiental	$\pi_E = 1$

TABLA 15.2: Coeficientes del transistor MOS de potencia

$$\lambda_P = \lambda_b \cdot \pi_T \cdot \pi_A \cdot \pi_Q \cdot \pi_E = 0.1776 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.6)$$

Modelo híbrido	
N	8 transistores
λ_C	0.1776
Factor ambiental	$p_i E = 0.5$
Circuito de potencia	$\pi_F = 21$
Factor de fabricación	$\pi_Q = 1$
Factor de aprendizaje >10 años	$\pi_L = 0.07$

TABLA 15.3: Coeficientes del modelo híbrido

$$\lambda_P = N \cdot \lambda_C \cdot (1 + 0.2 \cdot \pi_E) \cdot \pi_F \cdot \pi_Q \cdot \pi_L = 22.9 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.7)$$

Motor Stepper Como la función del motor stepper es regular la carga de la bicicleta en función del escenario virtual, su funcionamiento es intermitente y en baja velocidad, por lo que se considera que es una aplicación típica comercial del mismo, cuya tasa de fallas se muestra a continuación.

NEMA 17	
Aplicación	Comercial
Condiciones de operación	Bajo duty cycle, infrecuente
MTBF [horas]	1123600

TABLA 15.4: Coeficientes del motor tipo NEMA 17

$$\lambda_{P_{NEMA}} = 0,89 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.8)$$

Sensor de velocidad Para realizar el cálculo del sensor de velocidad, se puede pensar al mismo como un diodo LED y un fototransistor, de esta manera en la siguiente tabla se especifican los valores que la norma especifica para estos componentes.

Sensor de velocidad					
Dispositivo	λ_b	π_T	π_Q	π_E	λ_p
Fototransistor	0.013	1.4	5.5	1.0	0.1001
Diodo LED	0.00023	1.4	5.5	1.0	0.0018

TABLA 15.5: Coeficientes del sensor de velocidad

Por lo tanto, sumando los valores se obtiene:

$$\lambda_{\text{Sensor velocidad}} = \sum \lambda = 0.1019 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.9)$$

15.2.2. Valor final

Para obtener la tasa de fallas del sistema completo suman la tasa de fallas de cada módulo derivando en:

$$\lambda_{\text{total}} = \sum \lambda = 35.1515 \frac{\text{errores}}{10^6 \text{hs}} \quad (15.10)$$

Por lo que el tiempo medio entre fallas resulta:

$$MTBF = \int_0^{\infty} R(t) dt = \frac{10^6}{\lambda_{\text{total}}} \approx 28448 \text{ horas} \quad (15.11)$$

Luego la confiabilidad vendrá dada según nuestro modelo por la fórmula:

$$R(t) = e^{-\lambda_{\text{total}} \frac{t}{10^6}} \quad (15.12)$$

Así, se puede considerar que si se hace un uso intensivo del equipo dentro del gimnasio, el dispositivo estará encendido 8hrs, 6 días a la semana durante todo el año. por lo que se obtiene un total de 2400 horas de uso al año. Esto permite expresar las ecuaciones en años:

$$R(t_a) = e^{-\lambda_{\text{total}} \frac{t}{10^6} * 2400} \quad (15.13)$$

Si se grafica esta ecuación se puede obtener el gráfico que se muestra a continuación en la figura 15.2.

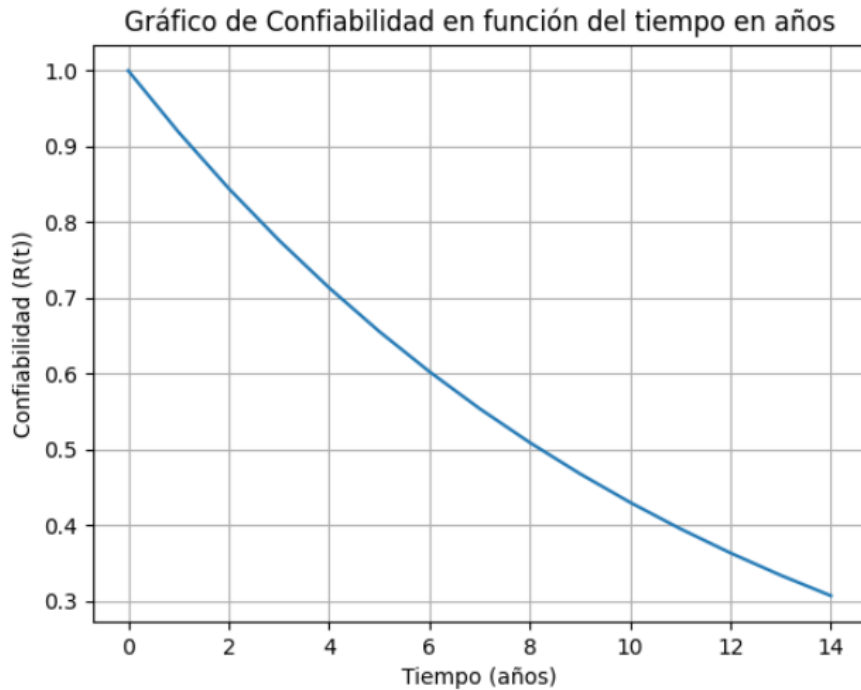


FIGURA 15.2: Confiabilidad expresada en función de los años de uso

Si se analiza el gráfico, se puede ver que la curva muestra que la confiabilidad es superior al 70 % en los primeros 4 años y luego baja hasta alcanzar el 50 % en el año 8.

Se considera de esta manera que los valores son razonables para la aplicación para la que se está desarrollando el producto. Además, se debe considerar que en el cálculo de dicha confiabilidad se tomaron parámetros muy conservadores, principalmente en el cálculo del driver del motor y de la fuente, en donde se utilizó el peor caso posible, y esos son los valores que elevan considerablemente el valor final del lambda en el que se basa el cálculo de la confiabilidad. Si bien este valor es un valor aceptable para nuestra aplicación, se deja constancia de que es calculado de forma conservadora y que puede ser menor al estimado.

15.3. Confiabilidad de Software

En la presente sección se desarrolla el análisis de confiabilidad para la parte de software del sistema. Este análisis se realiza mediante el método de estimación de Shooman, el cual normaliza las fallas por las líneas de código. Dicho método propone calcular la confiabilidad como:

$$R(t) = e^{-\lambda \cdot t} = e^{-k \cdot (\epsilon_T - \epsilon_c(\zeta)) \cdot t}$$

(15.14)

Donde ϵ_T es la tasa de errores totales y $\epsilon_c(\zeta)$ es la tasa de corrección de errores al tiempo ζ . Para hacer uso de este método, se utilizó un tiempo de depuración de 10 semanas, comenzando

en el mes de abril del año 2022. En este período, se registraron la cantidad de errores detectados y corregidos, dando lugar a los valores mostrados en la tabla 15.6

Semana	Errores detectados	E_D acum.	E. corregidos	E_C acum.	E. residuales
1 (Abril 22)	29	29	16	16	13
2 (Abril 22)	14	43	11	27	16
3 (Abril 22)	12	55	14	41	14
4 (Abril 22)	6	61	10	51	10
5 (Mayo 22)	9	70	8	59	11
6 (Mayo 22)	2	72	6	65	7
7 (Mayo 22)	1	73	4	69	4
8 (Mayo 22)	3	76	4	73	3
9 (Junio 22)	2	78	3	76	2
10 (Junio 22)	1	79	2	78	1

TABLA 15.6: Errores registrados en el período de depuración.

A partir de los valores registrados, se grafica su evolución en la figura 15.3. Se aprecia en este diagrama que la cantidad de errores residuales disminuye siguiendo una línea de tendencia exponencial negativa, a medida que aumenta el tiempo de depuración.

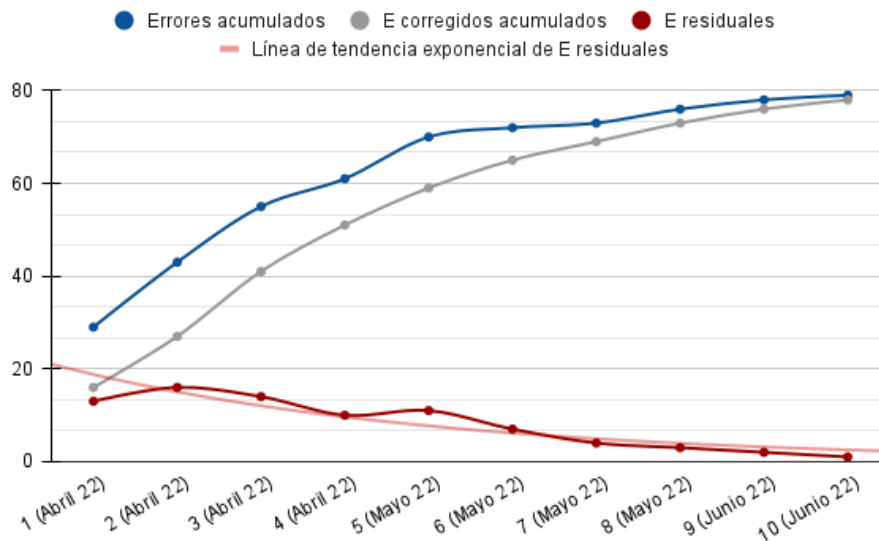


FIGURA 15.3: Evolución de los errores registrados.

Por su parte, en la figura 15.4 se muestra la evolución de la cantidad de errores detectados y se la compara con una exponencial negativa. Se verifica que la forma tomada se adapta correctamente al modelo, mediante un R^2 de 0,922. Además, dicho gráfico permite analizar que la etapa de depuración en la semana 10 toma valores estables que permiten comenzar la estimación de los parámetros del modelo.



FIGURA 15.4: Evolución de los errores acumulados, corregidos, acumulados y residuales.

Una vez registrados los valores, es posible calcular los parámetros del modelo de Shooman. Se parte de un λ calculado para cada semana como la relación entre los errores detectados acumulados y las horas totales. A partir de esto, se calcula la relación entre los λ de dos determinaciones, dando lugar a $\frac{\lambda_2}{\lambda_1}$. El valor resultante es utilizado para estimar E_T , la cantidad estimada total de errores, a partir de la siguiente fórmula:

$$\hat{E}_T = \frac{\frac{\lambda_2}{\lambda_1} \cdot E_{C_1} - E_{C_2}}{\frac{\lambda_2}{\lambda_1} - 1} \quad (15.15)$$

Luego, es posible obtener el otro parámetro del modelo de Shooman, que es el estimado de la constante k , calculado como:

$$\hat{k} = \lambda_1 \cdot \frac{I_T}{\hat{E}_T - E_C(\tau_1)} \quad (15.16)$$

Donde I_T es el número de instrucciones del software.

Semana	E. acum.	Eca	E. residuales	Horas	λ	λ_2/λ_1	\hat{E}_T	\hat{k}
1 (Abril 22)	29	16	13	168	0,173			
2 (Abril 22)	43	27	16	336	0,128	0,741		
3 (Abril 22)	55	41	14	504	0,109	0,853	122,053	12,117
4 (Abril 22)	61	51	10	672	0,091	0,832	100,459	16,518
5 (Mayo 22)	70	59	11	840	0,083	0,918	148,600	8,371
6 (Mayo 22)	72	65	7	1008	0,071	0,857	101,000	17,857
7 (Mayo 22)	73	69	4	1176	0,062	0,869	95,545	21,046
8 (Mayo 22)	76	73	3	1344	0,057	0,911	113,923	12,436
9 (Junio 22)	78	76	2	1512	0,052	0,912	107,200	14,881
10 (Junio 22)	79	78	1	1680	0,047	0,912	98,609	20,536

TABLA 15.7: Parámetros del modelo de Shooman calculados.

Luego de obtener los parámetros del modelo para cada semana, se calcula el valor final del estimador de λ del sistema. Esto se realiza utilizando la siguiente fórmula:

$$\lambda = k \cdot \epsilon_R = k \cdot \frac{E_R}{I_T} \quad (15.17)$$

Tras reemplazar por los valores de k y E_R de la última semana del período de depuración, se obtiene el valor:

λ	0,00228
-----------	---------

Este valor de λ corresponde MTTF de:

MTTF	438,2 horas
------	-------------

Por último, con el valor de λ obtenido, se reemplaza en la ecuación 15.14 y se obtiene el gráfico de la figura 15.5. En este gráfico se muestra la confiabilidad del software del sistema frente al tiempo de uso en horas. Se puede notar su característica exponencial decreciente a medida que el tiempo aumenta.

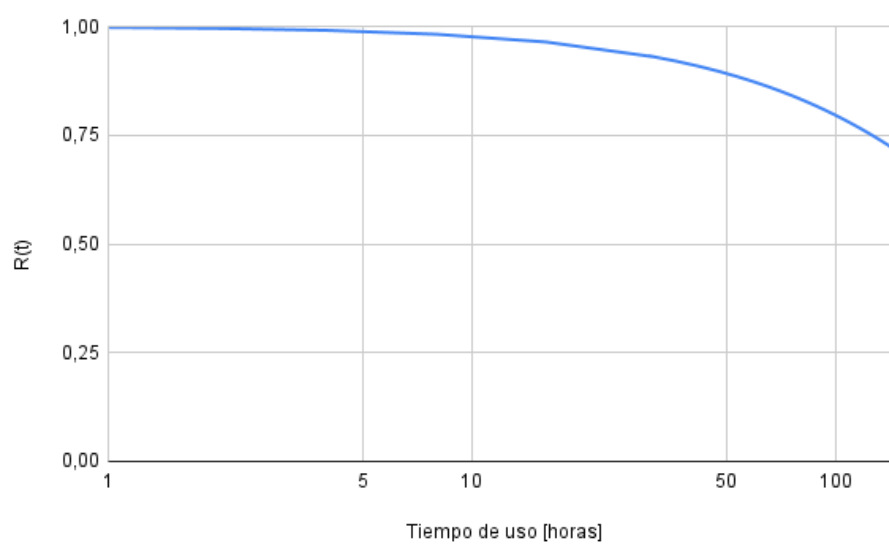


FIGURA 15.5: Confiabilidad del software frente al tiempo de uso.

16. Conclusiones

A modo de conclusión, se pudieron alcanzar los objetivos propuestos, y se logró cumplir con los requisitos definidos para el proyecto, desarrollando un sistema que integra la realidad virtual en las sesiones de ciclismo indoor, y un ecosistema que brinda a los usuarios de las sesiones todas las métricas sobre su desempeño una vez finalizado el entrenamiento.

En cuanto a la evolución del proyecto, a partir del prototipo realizado surgieron diferentes posibilidades de mejora para ser implementadas a futuro. Entre ellas se halla la mejora del acople mecánico del sistema de ajuste de la carga, a fin de reducir el sonido provocado al tener que variar la carga a lo largo de la sesión. Por otra parte, para el prototipo se utilizaron videos 360° de dominio público para los escenarios virtuales. Con el propósito de mejorar la experiencia se propuso desarrollar la infraestructura para llevar a cabo de manera propia las grabaciones de los videos, a fin de obtener, junto con las imágenes, mediciones precisas de variables como la velocidad, la aceleración y la inclinación, permitiendo mejorar la interacción entre el escenario virtual y la bicicleta.

Otra de las mejoras posibles se encuentra en el servidor, ya que migrando parte del servidor a un servicio en *la nube*, sería posible realizar sesiones con interacción entre diversos gimnasios, posibilitando de esta forma la competencia entre distintas sedes o las sesiones colaborativas. Por último, pensando en ampliar el modelo de negocios a futuro, la arquitectura con el servidor en la nube permite expandir el negocio hacia el mercado hogareño, brindando la posibilidad a los usuarios de realizar sesiones de ciclismo desde sus hogares utilizando toda la infraestructura diseñada por VRBike.

Referencias

- [1] Wijnand A Ijsselsteijn et al., "Virtual Cycling: Effects of immersion and a virtual coach on motivation and presence in a home fitness application", *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 19, núm. 2. SAGE Publications, pp. 159–171, ago. 20, 2021. doi: 10.1177/15485129211034586.
- [2] D. R. Mestre, V. Dagonneau, y C.-S. Mercier, "Does Virtual Reality Enhance Exercise Performance, Enjoyment, and Dissociation? An Exploratory Study on a Stationary Bike Apparatus", *Presence: Teleoperators and Virtual Environments*, vol. 20, núm. 1. MIT Press - Journals, pp. 1–14, feb. 01, 2011.
- [3] Chen, Ching Kong Chen, Fong-Jie Huang, jung tang Huang, Chi-Jung. (2007). Study of Interactive Bike Simulator in Application of Virtual Reality. *Journal of the Chinese Society of Mechanical Engineers, Transactions of the Chinese Institute of Engineers - Series C*. 28.
- [4] N. Xie et al., "Measurement of dynamic vibration in cycling using portable terminal measurement system", *IET Intelligent Transport Systems*, vol. 13, núm. 3. Institution of Engineering and Technology (IET), pp. 469–474, oct. 2018. doi: 10.1049/iet-its.2018.5181.
- [5] A. Matviienko et al., "Reducing Virtual Reality Sickness for Cyclists in VR Bicycle Simulators", *CHI Conference on Human Factors in Computing Systems*. ACM, abr. 29, 2022. doi: 10.1145/3491102.3501959.

17. Anexo

17.1. Anexo A: Casa de calidad

El análisis comparativo de la casa de calidad se realiza teniendo en cuenta distintos productos de la competencia. Por parte de Technogym, se tiene en cuenta la [Technogym Bike](#). Con respecto a LifeFitness, el modelo [IC7](#) es tenido en cuenta. Por último, por parte de Nordictrack se toma de referencia al modelo [Commercial S22i Studio Cycle](#).

17.1.1. Costo de adquisición

El costo de adquisición corresponde al costo inicial que debe abonar el comprador a fin de obtener el producto. A dicho requerimiento se le asigna una importancia relativa de 5 (valor máximo) debido a que en el mercado local se considera como la principal limitante a la hora de adquirir equipamiento. Esta consideración se obtuvo en comunicaciones con directivos de una cadena de gimnasios local, en las cuales se indicó la dificultad para mejorar las instalaciones debido a los costos.

Análisis comparativo Con respecto a la competencia, los precios a los que se venden sus productos resultan considerablemente altos para el mercado local. Al elevado precio de venta de estos equipamientos se le suman las tasas de importación requeridas, ya que se trata de empresas extranjeras. Tanto Technogym como LifeFitness se comercializan en torno a los US\$2500, mientras que NordicTrack lo realiza desde los US\$2000. Cabe destacar que estos productos incluyen a la bicicleta, a diferencia de VRBike, el producto propuesto. Sin embargo, si se comparan los precios anteriores con una bicicleta de calidad similar sin el sistema inmersivo (alrededor de US\$700), se aprecia que el diferencial económico incluido por el sistema de inmersión sigue siendo elevado. Por el lado de VRBike, se espera poder comercializarlo con un precio menor a los analizados. Este precio incluiría al visor de realidad virtual, a los sensores, actuadores, sistema de cómputo y todas las conexiones intermedias. Teniendo en cuenta que el visor de realidad virtual Oculus Quest 2 se puede adquirir por alrededor de US\$400 y que es uno de los componentes de mayor valor, el precio total resultante de cada instalación de VRBike sería preliminarmente de a lo sumo US\$1200. De esta forma, resulta muy conveniente económicamente y se le asigna un valor de 5 en la casa de calidad. A NordicTrack, al contar con un precio elevado, pero menor a los de Technogym y LifeFitness se le asigna un 3. Por último, a los restantes se les asigna un 2, ya que son los precios más elevados. Existen productos de otros fabricantes con precios aún mayores, como la [BZLLW](#), por lo que decide no asignarles un 1.

Relación con los "Cómo" El sistema de visualización, el cual se utiliza para visualizar el escenario virtual, junto con el regulador de la fuerza de pedaleo, son los factores más caros y, por lo tanto, los que más afectan al costo de adquisición. El sistema de comunicación que incluye todo el sistema de Networking se encuentra en un segundo escalón. Por último, el sensor de velocidad y las vibraciones en el manubrio, si bien influyen en el costo, no son factores fundamentales y tienen un mínimo peso en la relación.

17.1.2. Costo mensual

El costo mensual se refiere a un costo correspondiente al mantenimiento o suscripción de ciertos servicios asociados al producto principal. Se considera para este caso que, siempre y cuando el valor sea razonable, los gimnasios de gama media-alta, no lo aprecian como un limitante. Esto es debido a que a comparación de una adquisición de maquinaria, el costo mensual de un producto como el propuesto es reducido. A su vez, los gimnasios ya cuentan con servicios con altos costos mensuales como, para dar un ejemplo, la electricidad.

Análisis comparativo Tanto Technogym, como LifeFitness y NordicTrack, ofrecen una suscripción a sus aplicaciones móviles, en la cual el usuario puede encontrar las métricas obtenidas al realizar los entrenamientos, videos de clases grabadas por profesores y otras funcionalidades. El costo de suscripción mensual se encuentra para todas ellas en torno a los US\$40 por usuario. Así, se les asigna un 3 en la casa de calidad debido a que este precio, aunque es elevado, no es extremadamente caro y resulta accesible. En VRBike, el caso propuesto, se incluirían distintos servicios que se deberían cobrar en forma de suscripción. Entre estos se encuentran una app móvil para acceder a las métricas de los usuarios, el software local al que se conectaría cada participante, y una continua publicación de nuevos mapas. Si bien este conjunto incluye las mismas o más funcionalidades que la competencia, se espera reducir su precio al cobrarlo en conjunto para todo el gimnasio en el que se implemente. Es decir, en lugar de cobrarlo individualmente por usuario, determinar un precio único que, aunque sea mayor, resulte conveniente dada la cantidad de individuos. En consecuencia, al ser menor al de la competencia (por usuario) pero no ser considerablemente barato, se le asigna un valor de 4 en la casa de calidad.

Relación con los "Cómo" Los factores que se incorporan en forma de suscripción y afectan al costo mensual son la app móvil de métricas en primer lugar y el número de escenarios en un peso menor. La app móvil de métricas permite a los usuarios visualizar sus métricas históricas desde su celular, y compararlas con las del resto de competidores. Con respecto a la cantidad de escenarios, su costo sería algo menor que el de la app móvil, y permitiría actualizar al software incluyendo continuamente más mapas.

17.1.3. Comodidad

Este requerimiento está asociado a las características del producto que le permitan al usuario realizar el ejercicio naturalmente. Esta se puede ver dificultada por los sistemas añadidos para obtener una experiencia inmersiva. La importancia relativa dada a esta característica es de 3, ya que, aunque es necesario evaluarlo, no es un punto decisivo al analizar la compra de uno de los productos, teniendo en cuenta que las sesiones de spinning no suelen ser de larga duración. Además, la mayor parte de la comodidad que percibe quien realiza el ejercicio está dada por la bicicleta utilizada, la cual no forma parte del producto estudiado.

Análisis comparativo Los productos de la competencia (Technogym, LifeFitness y NordicTrack) no presentan elementos que le dificulten el ejercicio a quien lo está realizando. Sus productos

incluyen pantallas frente al individuo, las cuales no lo perjudican ni resultan invasivas. Por esta razón, se les asigna un valor máximo de 5 en la casa de calidad. Por el contrario, en el producto VRBike se requiere el uso de un visor de realidad virtual. Este visor incorporaría peso en la cabeza del participante, reduciendo su comodidad. Este efecto, aunque existe, no es determinante para la mayoría de las personas y ya existen muchas aplicaciones de visores de realidad virtual en prácticas deportivas. Por esta razón, se le asigna un 3 en la casa de calidad.

Relación con los "Cómo" La comodidad resulta muy influenciada por el sistema de visualización utilizado. Un visor de realidad virtual afecta directamente la comodidad del usuario al tener que soportar un peso en su cabeza durante el entrenamiento. Por ello, la relación es máxima con este factor. Por otra parte, en una relación media se encuentra la interfaz simple, debido a que si es muy dificultoso el uso del software se podría dificultar la realización del ejercicio. A su vez, los actuadores (vibraciones en el manubrio y regulador de la fuerza) afectan en un término medio a la comodidad. Una vibración o fuerza excesiva podrían reducir el confort del participante. Por último, el sensor de velocidad, sobre todo su precisión, afecta en forma mínima a la comodidad. Si el sensor de velocidad no es lo suficientemente preciso, se podrían generar mareos debido a la desincronización del ejercicio realizado con el movimiento visualizado en el software.

17.1.4. Facilidad de uso

La facilidad de uso es otra de las características analizadas. Con esto se refiere a la simplicidad para interactuar con el sistema por parte de un usuario sin conocimientos técnicos. Este parámetro no es un parámetro fundamental para la compra del producto. Sin embargo, es necesario evaluarlo dado que un sistema con mucha dificultad de uso podría provocar una negativa en los usuarios a utilizarlo. Por esta razón, se le asigna una importancia relativa de 3.

Análisis comparativo Con respecto a la competencia, sus productos resultan muy fáciles de utilizar y no se encuentran críticas comunes en este aspecto. Sin embargo, el uso de una pantalla táctil y la cantidad de funcionalidades hace que para ciertos individuos poco familiarizados con los sistemas tecnológicos actuales se pueda dificultar ligeramente la experiencia. En consecuencia, se les asigna un valor de 4 en la casa de calidad. En el caso de VRBike, se espera desarrollar una interfaz de usuario muy simple y que la mayor parte de las configuraciones, como la selección del mapa a visualizar, las realice de forma centralizada el entrenador a cargo de la clase de spinning donde se aplique. No obstante, nuevamente existe una pequeña porción de individuos que pueden verse perjudicados por el uso de un visor de realidad virtual. De esta forma, se le asigna un valor de 4.

Relación con los "Cómo" La facilidad de uso del producto propuesto se verá afectada por distintos factores. En primer lugar, por el sistema de visualización utilizado, ya que un visor de realidad virtual puede dificultar la experiencia para ciertos usuarios. También, el sistema de comunicación, el cual incluye la interacción con el entrenador, afectaría la sencillez de la experiencia de usuario. No obstante, el factor fundamental en la relación, y con el cual se espera maximizar la facilidad

de uso, es con la simplicidad de la interfaz de usuario. Se propone que el usuario deba realizar configuraciones mínimas al momento de utilizar el sistema y que sea el entrenador quien desde un dispositivo maestro configure la sala. Otro factor que afecta, aunque en un peso mínimo, es la app móvil para acceder a las métricas, ya que afecta en la forma en que el usuario las visualiza.

17.1.5. Realismo de los escenarios

Los escenarios son una parte fundamental de la experiencia del usuario en el uso de los sistemas inmersivos en ciclismo indoor. La calidad gráfica y la exactitud en la recreación de ambientes exteriores potencian el uso del sistema. En caso de no presentar escenarios reales, el participante de la sesión de ciclismo indoor podría preferir los sistemas tradicionales y decidir no continuar con la experiencia inmersiva. Dada esta importancia, se le asigna un valor de 4 en la casa de calidad. El hecho de no asignarle un valor de 5 se basa en que para un cierto grupo de usuarios, aquellos familiarizados con los videojuegos, el realismo podría no resultar un aspecto primordial.

Análisis comparativo La empresa NordicTrack es, de la competencia analizada, la que escenarios más realistas presenta. Su sistema cuenta con grabaciones de alta calidad realizadas en escenarios como bosques y rutas, por lo que logran recrear los espacios exteriores perfectamente. Acerca de Technogym, y de igual forma LifeFitness, la empresa también cuenta con grabaciones, pero en ellas no incluye la presencia de otros ciclistas. En términos del producto propuesto, se espera que VRBike cuente con escenarios sintéticos de alta calidad, donde se logre una correcta simulación de ambientes reales. Asimismo, en los escenarios se podrá apreciar a otros ciclistas. Debido a estas causas, se le asigna un valor de 5 a NordicTrack; y de 4 a LifeFitness, Technogym y VRBike.

Relación con los "Cómo" El factor que se relaciona en mayor medida con el realismo de los escenarios es el sistema de visualización. La capacidad gráfica de este sistema es el principal limitante para incluir escenarios realistas. Además del sistema de visualización, surge una relación media entre el realismo de los escenarios y la cantidad de escenarios. Esto se da en términos del tiempo y del costo de desarrollo; los recursos asignados a cada escenario disminuyen a medida que se requiere una mayor cantidad de ellos.

17.1.6. Baja emisión sonora

Teniendo en cuenta que en una clase de spinning el ruido es generalmente alto debido a la alta cantidad de participantes, el ruido de las bicicletas y las indicaciones del entrenador de la clase, la emisión sonora del producto analizado es el aspecto menos relevante a tener en cuenta. Así, se le asigna un valor de 1 en la casa de calidad.

Análisis comparativo Todos los productos estudiados, tanto VRBike como la competencia, introducen una emisión sonora media en las sesiones de ciclismo indoor, sin la apreciación de manifestaciones negativas por parte de los usuarios en este aspecto. Dada esta característica, se les asigna un valor de 3 en la casa de calidad.

Relación con los "Cómo" Los módulos del producto que mayores emisiones sonoras generan son los actuadores. En primer lugar, se encuentra el regulador de la fuerza, el cual al contar con un motor capaz de ajustar el freno de la bicicleta es el que más emisiones genera. Asimismo, las vibraciones en el manubrio se generan mediante otro motor, aunque este de menor tamaño y emisiones sonoras.

17.1.7. Multijugador

El deportista que participa de una sesión de ciclismo indoor, aunque no sea por lo general su principal objetivo, podría ver enriquecida su experiencia al poder comparar su progreso con el de los demás participantes. A su vez, esta característica le agrega dinámica a la experiencia, pudiendo motivar a los usuarios. Así, se considera que tiene una importancia relativa de 3.

Análisis comparativo Los sistemas analizados de Technogym y de NordicTrack apuntan a un uso individual y presentan ciertas funcionalidades para comparar el progreso con el de usuarios remotos. A pesar de esto, no permiten vincularse con otros individuos de una misma sala, algo fundamental en sesiones de spinning. Por ello, se les asigna un valor de 2 en la casa de calidad. LifeFitness, por su parte, sí permite la asociación y la comparación de las métricas internamente en un mismo grupo de entrenamiento. Pero, la apreciación de los múltiples jugadores es solamente en cuanto a números (velocidad media, bpm, etc.). Debido a esto, se le otorga un valor de 3. Se propone que VRBike permita la vinculación de hasta 15 participantes en una misma sesión y la visualización de los demás usuarios en la experiencia de realidad virtual. Dadas estas funciones, se considera sobre sus competidores y se le atribuye un valor de 4.

Relación con los "Cómo" Con respecto a la característica de multijugador del producto, el sistema de comunicación es el principal limitante y, por lo tanto, es el factor que mayor relación mantiene. El ancho de banda del sistema utilizado influye en la cantidad de paquetes de datos enviados al mismo tiempo y, por lo tanto, no permite la participación de una cantidad ilimitada de usuarios. A su vez, la precisión del sensor de velocidad puede afectar la participación de múltiples usuarios. Esto se da debido a que si se utiliza un sensor con mucha varianza, se podrían obtener estadísticas distintas en dos participantes que realizan exactamente el mismo ejercicio.

17.1.8. Comunicación con el entrenador

El entrenador en una clase de spinning cumple la función de indicarle a los participantes la forma de regular la intensidad con la que realizan el entrenamiento. Por ejemplo, si una persona se encuentra efectuando mucho esfuerzo en el inicio de la sesión, el entrenador podría indicarle que disminuya la intensidad a fin de poder tolerar el resto del entrenamiento. También, motivan a los participantes y buscan que no abandonen el ejercicio con mensajes de ánimo. Por estas razones, son una pieza fundamental en las clases de spinning y el producto analizado debe permitir esta interacción. Con estas razones, se le asigna una importancia relativa de 4 en la casa de calidad.

Análisis comparativo Dado que los productos de la competencia no incluyen un visor de realidad virtual, el entrenador puede interactuar con los participantes directamente y como lo realiza tradicionalmente. En VRBike se propone que el entrenador pueda enviar mensajes a los participantes desde un dispositivo maestro. Sin embargo, si bien esto sigue haciendo posible las funciones principales, limita en cierta medida las posibilidades del entrenador. Por tales motivos, se le atribuye un valor de 3 a VRBike y de 4 a sus competidores.

Relación con los "Cómo" La comunicación con el entrenador representa un valor importante que se puede apreciar en la importancia relativa que presenta. Podemos ver en el diagrama de la casa de calidad que tiene un peso medio respecto de la app móvil, ya que esta es la que nos permite lograr una comunicación entre las métricas de la bicicleta y las aplicaciones de los usuarios, almacenando los valores previos en una base de datos a la que el entrenador puede acceder con la finalidad de modificar la rutina de ejercicios programada. También presenta un peso fuerte respecto del sistema de comunicaciones debido a que es por dicho sistema que se envía la información en tiempo real con el entrenador.

17.1.9. Variedad de los escenarios

Así como se mencionó para el realismo de los escenarios, otra característica muy importante es que estos sean variados. Un poco variedad de escenarios podría generar que un participante habitual de sesiones de spinning se memorice el recorrido a realizar y el mapa visualizado, produciendo que este pueda perder motivación dada la continua repetición. Entonces, se considera que este aspecto presenta una importancia relativa de 4.

Análisis comparativo En cuanto al contraste con los competidores, se espera que VRBike cuente con una cantidad de escenarios similar a Technogym y LifeFitness, alrededor de 10 a 15 escenarios. Esta es una cantidad que hace posible que en un entrenamiento regular de entre una y tres veces por semana variando el escenario con el que se realice no produzca una memorización en el corto plazo. Por ello, se le asigna un valor de 4 a los tres productos mencionados. NordicTrack, por su parte, cuenta con una cantidad algo menor, lo que justifica que se le asigne un valor de 3.

Relación con los "Cómo" Respecto a la variedad de escenarios, se puede encontrar una relación media respecto del sistema de visualización, debido a que el nivel de detalles y posibilidades de procesamiento modifican las posibles diferencias y variabilidad de los escenarios diseñados. También podríamos encontrar una relación débil respecto de la interfaz simple, ya que una variedad muy alta de escenarios implicaría desafíos respecto de la interfaz para mantenerla simple y amigable. Por último, existe una relación de máximo peso entre la variedad de los escenarios y el número de escenarios posibles, si la variedad de escenarios que se puede alcanzar es alta entonces se podrán lograr un gran número de escenarios con buenos resultados.

17.1.10. Calidad de la inmersión

La característica de la calidad de la inmersión es otro de los puntos fundamentales en el análisis del producto, dado que este se basa en la inclusión de inmersión en el ciclismo indoor. Esta característica incluye los sensores y actuadores que permitan vincular el entrenamiento del usuario con lo visualizado y medido. Dado que su importancia, aunque sea muy alta, es algo menor a la del costo de adquisición -un gimnasio podría preferir no innovar añadiendo inmersión si su costo es muy elevado-, se le asigna una importancia relativa de 4 en la casa de calidad.

Análisis comparativo La calidad de la inmersión es el aspecto en donde más se destaca VRBike en comparación a sus competidores. Tanto Technogym como LifeFitness cuentan con sensores de velocidad, bpm, y sensores similares, pero no cuentan con actuadores que generen sensaciones percibidas en quien se entrena. De esta forma, como la inmersión se centra solamente en la visualización en un monitor del progreso del participante, se les asigna un valor de 2. Con respecto a NordicTrack, además de los sensores mencionados, cuenta con un adaptador de la inclinación y de la fuerza de la bicicleta dependiendo del terreno en el que se perciba que se está pedaleando. Así, se encuentra un punto por encima de los otros competidores, con un valor de 3. En el producto propuesto, VRBike no solo contaría con sensores, sino también con actuadores para introducir vibración en el manubrio y ajuste de la fuerza de pedaleo. Sumado a esto, el visor de realidad virtual incrementa considerablemente la calidad de la inmersión. Por estas razones, VRBike presenta un valor de 5 en la casa de calidad.

Relación con los "Cómo" Respecto a la calidad de inmersión, podemos encontrar en el diagrama de la casa de calidad que se considera que presenta una relación de mediano peso respecto del sistema de visualización, ya que los gráficos y la capacidad de procesamiento del sistema de visualización representan factores importantes a la hora de lograr un grado de inmersión alto en el ciclista. También presenta un grado de relación medio con las vibraciones que se puedan sentir en el manubrio, ya que estas se corresponderían con la superficie donde la bicicleta virtual se encuentre circulando. Respecto del regulador de fuerza, se puede ver que la calidad de inmersión es alta, ya que se considera que es un factor muy importante a la hora de aportar realismo a la experiencia de usuario, ya que se adapta en función de las pendientes del terreno en el que el usuario se encuentra circulando, o con base en el cambio seleccionado por el ciclista en la bicicleta fija. También se considera que existe una relación de máximo peso entre la calidad de inmersión y el sensor de velocidad, ya que el mismo es muy importante para transmitir esa sensación de realismo que se busca en la persona al utilizar la bicicleta.