



TRABAJO FIN DE GRADO
INGENIERÍA INFORMÁTICA

WATCH-YOUR-ARM

Tracking your gym arm activity through smartwatch technologies

Autor

Andrés López Joya

Directores

Oresti Baños Legrán



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE
TELECOMUNICACIÓN

Granada, Julio de 2020

FITCONTROL

WATCH-YOUR-ARM

Tracking your gym arm activity through smartwatch technologies.

Autor

Andrés López Joya

Directores

Oresti Baños Legrán

WATCH-YOUR-ARM: tracking your gym arm activity through smartwatch technologies

Andrés López Joya

Palabras clave: sensor, reloj inteligente, deporte, gimnasio, ejercicio, entrenamiento, actividad física

Resumen

El presente trabajo tiene como objetivo crear una aplicación para Android, la cual reconozca el principio y el final de las repeticiones realizadas de un ejercicio de gimnasio durante un entrenamiento anaeróbico(a través del reloj inteligente) y clasifique dichas repeticiones según los datos recabados entre el principio y el final, de manera correcta o incorrecta proporcionando una justificación de la misma. Para ello se ha desarrollado un sistema que consiste de dos partes una en dispositivo móvil y la otra en reloj inteligente.

El sistema hace uso de los sensores disponibles en los relojes inteligentes (acelerómetro y giroscopio).

Para evaluar la precisión del sistema se ha realizado un estudio sobre 10 personas con edades comprendidas entre 21-58, y con un nivel de experiencia en el entrenamiento variado (principiantes y avanzados).

Los resultados muestran que es posible identificar y clasificar las repeticiones según su ejecución, por lo que a raíz de esos datos se ha creado un algoritmo capaz de clasificar dichas repeticiones.

WATCH-YOUR-ARM: tracking your gym arm activity through smartwatch technologies

Andrés López Joya

Keywords: sensor,wereable,smartwatch, sport, gym, fitness,exercise,training,physical activity

Abstract

The present work aims to create an application for Android, which recognizes the beginning and end of the repetitions performed of a gym exercise during an anaerobic training (through the smart watch) and classifies said repetitions according to the data collected between the beginning and end, correctly or incorrectly providing a justification for it. For this, a system has been developed that consists of two parts, one in a mobile device and the other in a smart watch.

The system makes use of the sensors available in smart watches (accelerometer and gyroscope).

To evaluate the precision of the system, a study was carried out on 10 people between the ages of 21-58, and with a varied level of experience in training (beginners and advanced).

The results show that it is possible to identify and classify the repetitions according to their execution, which is why an algorithm capable of classifying said repetitions has been created as a result of these data.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Planteamiento del problema	2
1.3. Motivación	2
1.4. Solución propuesta	3
1.5. Objetivos	3
2. Estado del arte	5
3. Metodología	9
3.1. Planificación temporal y estimación de recursos necesarios	9
3.1.1. Presupuesto	11
3.2. Diseño	11
3.2.1. Requisitos	11
3.2.2. Arquitectura	16
3.3. Implementación	18
3.3.1. FitControl User	18
3.3.2. FitControl	31
3.4. Algoritmo	32
4. Evaluación	39
4.1. Resultados	39
4.1.1. Estudio	39
4.1.2. Matriz de confusión	40
4.1.3. Encuesta post-uso	40
4.2. Discusión	45
5. Conclusiones y Trabajos Futuros	47
5.1. Conclusiones	47
5.2. Trabajos Futuros	47
Bibliografía	50

Índice de figuras

1.1. Índice de popularidad en la búsqueda de <i>rutinas en casa</i> (ratio de menos popular, 0, a más popular, 100). Fuente Google Trend [10].	1
1.2. Envíos estimados de dispositivos portátiles en todo el mundo (en millones de unidades). Fuente [26]	2
3.1. Diagrama de Gantt para la planificación del proyecto. Fuente propia	10
3.2. Crear nuevo perfil. Fuente propia.	12
3.3. Iniciar conexión entre smartwatch y smartphone. Fuente propia.	12
3.4. Captar datos del reloj inteligente y enviarlos a smartphone. Fuente propia.	13
3.5. Procesar datos obtenidos y clasificar las repeticiones. Fuente propia.	13
3.6. Consultar repeticiones totales de un movimiento. Fuente propia.	14
3.7. Consultar repeticiones de un usuario en una fecha. Fuente propia.	14
3.8. Cambiar foto de perfil. Fuente propia.	15
3.9. Cambiar datos del perfil. Fuente propia.	15
3.10. Esquema de la arquitectura del proyecto. Fuente propia.	16
3.11. Diagrama UML de las tablas de la base de datos de FitControl User. Fuente propia.	19
3.12. Búsqueda de dispositivos por el dispositivo móvil. Fuente propia	21
3.13. Búsqueda de dispositivos por el dispositivo móvil. Fuente propia	22
3.14. Primera actividad de la aplicación. Fuente propia.	24
3.15. Actividad para crear nuevo perfil de usuario. Fuente propia.	24
3.16. Apartado home de la aplicación en un perfil concreto. Fuente propia.	25
3.17. Actividad para modificar nuestro perfil. Fuente propia.	25
3.18. Apartado Movements de la aplicación. Fuente propia.	26
3.19. Apartado en el que se muestran todas las repeticiones de elevaciones laterales. Fuente propia.	26
3.20. Apartado Register de la aplicación sin seleccionar fecha. Fuente propia.	27
3.21. Apartado Register de la aplicación seleccionando fecha. Fuente propia.	27
3.22. Apartado Register de la aplicación en una fecha. Fuente propia.	28
3.23. Apartado Register de la aplicación en una fecha y un movimiento seleccionado. Fuente propia.	28
3.24. Apartado Reciver de la aplicación buscando dispositivo. Fuente propia.	29
3.25. Apartado Reciver de la aplicación con dispositivo encontrado. Fuente propia.	29
3.26. Apartado Reciver una vez procesadas las repeticiones. Fuente propia.	30
3.27. Actividad principal de la aplicación en el reloj inteligente. Fuente propia. Fuente propia.	32
3.28. Mala ejecución de una elevación lateral por exceso de flexión de codo. Fuente propia.	33

3.29. Mala ejecución de una elevación lateral por ausencia de flexión de codo.	
Fuente propia.	33
3.30. Correcta ejecución de una elevación lateral por ligera de flexión de codo.	
Fuente propia.	33
3.31. Datos del acelerómetro al realizar 6 repeticiones de elevaciones laterales.	
Fuente propia.	34
3.32. Datos del giroscopio al realizar 6 repeticiones de elevaciones laterales. Fuente	
propia.	34
3.33. Datos del acelerómetro al realizar 6 repeticiones de elevaciones laterales con	
las repeticiones localizadas. Fuente propia.	35
3.34. Datos del giroscopio al realizar 6 repeticiones de elevaciones laterales con	
las repeticiones localizadas. Fuente propia.	36
3.35. Árbol de decisión creado a partir de los datos de 10 usuarios para elevaciones	
laterales. Fuente propia.	36
4.1. Primera pregunta de la encuesta post-uso. Fuente propia.	41
4.2. Segunda pregunta de la encuesta post-uso. Fuente propia.	41
4.3. Tercera pregunta de la encuesta post-uso. Fuente propia.	42
4.4. Cuarta pregunta de la encuesta post-uso. Fuente propia.	42
4.5. Quinta pregunta de la encuesta post-uso. Fuente propia.	42
4.6. Sexta pregunta de la encuesta post-uso. Fuente propia.	43
4.7. Séptima pregunta de la encuesta post-uso. Fuente propia.	43
4.8. Octava pregunta de la encuesta post-uso. Fuente propia.	43
4.9. Novena pregunta de la encuesta post-uso. Fuente propia.	44
4.10. Décima pregunta de la encuesta post-uso. Fuente propia.	44

Índice de tablas

2.1. Tabla comparativa entre relojes inteligentes de distintas marcas. Fuente propia	7
3.1. Tabla con presupuesto del proyecto. Fuente propia.	11
4.1. Tabla con todas las repeticiones realizadas por cada usuario en el estudio.	
Fuente propia.	39
4.2. Matriz de confusión del algoritmo. Fuente propia.	40
4.3. Tabla con los percentiles de cada usuario. Fuente propia.	44

Capítulo 1

Introducción

1.1. Contexto

En estos últimos años la gran mayoría de la población esta adquiriendo conciencia sobre la importancia del deporte en la calidad de vida y la prevención de enfermedades.

Esta corriente social relacionada con la vida saludable y la actividad física está llevando a un gran número de personas a apuntarse a gimnasios o a comprar material deportivo para ejercitarse, este último más frecuente tras la aparición del corona-virus.

El sector deportivo actualmente está abriendo la puerta a numerosos negocios relacionados con él, ya que es un nicho adecuado para poder sacar un beneficio económico.

Cabe recalcar que tras la aparición del corona-virus muchas personas se han tenido que mantener en sus casas por el bien común y este suceso ha provocado un movimiento en las redes a favor del deporte y la vida activa. Esto ha desencadenado que muchas personas hayan empezado a realizar ejercicio en casa durante la cuarentena.



Figura 1.1: Índice de popularidad en la búsqueda de *rutinas en casa* (ratio de menos popular, 0, a más popular, 100). Fuente Google Trend [\[10\]](#).

Como se aprecia en la figura [1.1](#) la popularidad de *rutinas en casa* en las búsquedas de Google al empezar la cuarentena (en el mes de marzo) se disparó, lo que indica el interés por el entrenamiento en casa.

1.2. Planteamiento del problema

Muchas de estas personas carecen de información previa sobre musculación o sobre la correcta ejecución de ciertos ejercicios y no disponen de la ayuda de una persona especializada que esté completamente pendiente de su técnica y sus entrenamientos. La gran mayoría de los gimnasios son comerciales y no tienen una atención personalizada si no es mediante pago.

Además esta situación se ha visto acentuada por la pandemia, pues incluso la gente que puede permitirse un asesor o entrenador personal, no puede recurrir al trato personal de un especialista por el confinamiento.

Por todo ello un gran número de individuos realiza ejercicios de una forma potencialmente lesiva, con lo cual puede que no obtengan los beneficios deseados, y por consiguiente se frustran y acaban abandonando el entrenamiento.

1.3. Motivación

La idea de este proyecto surge como consecuencia del gran avance tecnológico actual debido a la aparición de nuevas aplicaciones y dispositivos para monitorizar toda actividad física de manera cómoda y sencilla. Los dispositivos móviles portátiles como los relojes inteligentes, las pulseras para la actividad física, entre otros, están adquiriendo gran importancia en el sector y están teniendo un auge de ventas. Dichos dispositivos pueden proporcionar información sobre el entrenamiento aeróbico o anaeróbico, y pueden almacenar información e incluso compartirla con otras personas.

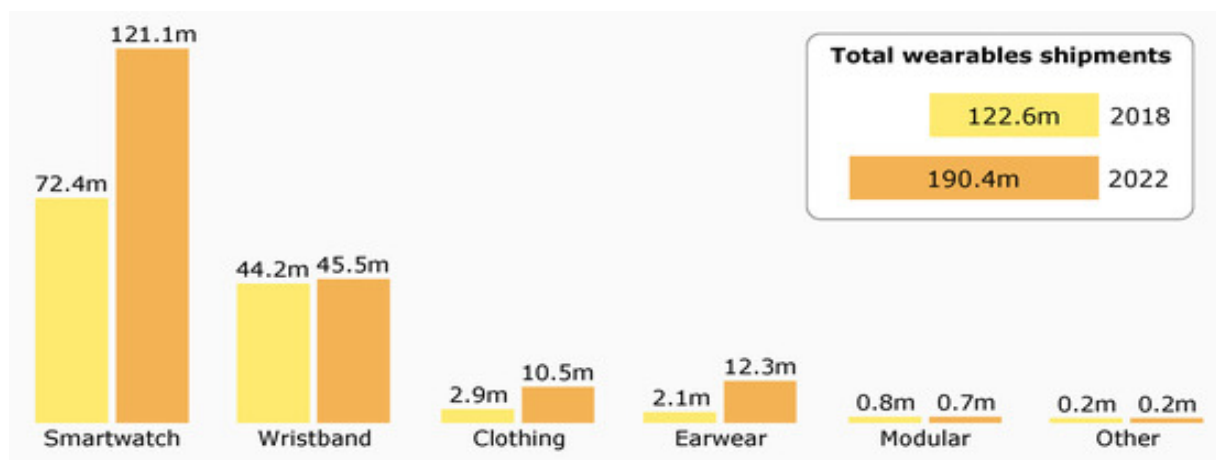


Figura 1.2: Envíos estimados de dispositivos portátiles en todo el mundo (en millones de unidades). Fuente [\[26\]](#)

La figura [1.2](#) ilustra bien que la venta de este tipo de productos ha aumentado y va a seguir aumentando. Destaca sobre el resto de productos el reloj inteligente, que es interesante para el desarrollo del proyecto.

1.4. Solución propuesta

Mi propuesta para este trabajo es crear una aplicación que identifique las repeticiones de un determinado ejercicio una vez terminado el entrenamiento. Estas repeticiones serán clasificadas y explicadas según la calidad de la ejecución. Además se creará un registro de ellas en el que el usuario podrá observar la evolución realizada.

El propósito de esta aplicación es aprovechar el auge que están teniendo los relojes inteligentes llegando por consiguiente al máximo número de usuarios. Esto se hará para mejorar el entrenamientos de los usuarios y para garantizar la máxima activación muscular para cada ejercicio.

1.5. Objetivos

Objetivo Principal: El objetivo principal de este proyecto es desarrollar un sistema inteligente que permita detectar y cuantificar la realización del ejercicio *elevación lateral* mediante el uso de mancuernas. El sistema permitirá además medir las repeticiones y delimitar si están bien ejecutadas o no.

Objetivos Específicos:

- Crear una arquitectura software que permita conectar el dispositivo móvil con el reloj inteligente para enviar datos, procesarlos y visualizarlos.
- Crear un algoritmo que sea capaz de clasificar las repeticiones.
- Evaluar el nivel de acierto del algoritmo en cuestión.
- Desarrollar una interfaz gráfica que permita al usuario manejar todas las funcionalidades anteriormente mencionadas y visualizar los datos obtenidos.

Capítulo 2

Estado del arte

Aunque sea un sector en auge, todavía no hay muchos artículos de investigación referentes a la correcta ejecución de los movimientos utilizando dispositivos como los relojes inteligentes. Primeramente se ha realizado una investigación a través de la web Google Scholar [9], con el objetivo de exponer cómo está este sector. La búsqueda se ha realizado bajo esta consulta:

(sensor OR wearable OR smartwatch OR smart watch)
AND
(wrist* OR bracelet)
AND
(powerlift* OR fitness OR gym* OR exercise OR athlete* OR training OR
workout OR physical activity)

Gracias a ella se han encontrado varios artículos relevantes para el proyecto; uno de ellos se titula: *Analysis of wearable and smartphone-based technologies for the measurement of barbell velocity in different resistance training exercises!* [20] que fue un proyecto realizado por Balsalobre-Fernández y David Marchante. Consiste en la comparativa entre un sensor ya establecido y una aplicación desarrollada por ellos con el propósito de analizar la validez, confiabilidad y precisión de la nueva tecnología portátil, basada en teléfonos inteligentes para la medición de la velocidad de la barra en los ejercicios de entrenamiento de resistencia. Esta aplicación es interesante ya que usa una cámara para medir los datos concluyentes a diferencia de mi proyecto en el cual se utiliza un reloj inteligente. Esta aplicación fue desarrollada en IOS, denominándose *IOS PowerLift*. Para el caso práctico se hicieron pruebas con 10 participantes, 6 varones y 4 mujeres con edades comprendidas entre los 22-30 y con un nivel de entrenamiento físico alto. Para este caso se han utilizado dos receptores de datos, por un lado un sensor en la muñeca izquierda de los participantes marca *Beast*, obteniendo así datos del acelerómetro, giroscopio y magnetómetro, además de una cámara para grabar la ejecución del ejercicio y comparar los datos obtenidos por cada sensor. En este caso los participantes realizaron 3 ejercicios (press banca, sentadilla profunda, puente de glúteo) y se vio una relación entre los datos obtenidos en el sensor *Beast* y en la aplicación la cual usaba la cámara. Se puede recalcar que esta aplicación tiene un uso exclusivo para la medición de la velocidad de ejecución y no sobre la correcta ejecución del movimiento, ya que los participantes tienen un nivel avanzado.

Otro artículo interesante para explicar el contexto del sector se titula *FitCoach: virtual fitness coach empowered by wearable mobile devices* [22], este proyecto consiste en la crea-

ción de una aplicación la cual reconoce los diferentes tipos de ejercicios e interpreta datos de condición física de grano fino (es decir, fuerza de movimiento y velocidad) a un puntaje de revisión de ejercicio fácil de entender, que proporciona una evaluación y recomendación de rendimiento de entrenamiento integral. Los autores de este proyecto son: Xiaonan Guo, Jian Liu y Yingying Chen. Para esta aplicación se ha usado un reloj inteligente, concretamente: reloj inteligente de LG *Watch Urbane*, del cual se han obtenido los datos de los sensores (acelerómetro y giroscopio) que lleva integrados. La aplicación se llama *FitCoach* y se ha desarrollado en *Samsung Galaxy Note 3* con android y para el desarrollo de la misma se han tomado datos de 12 participantes, de los cuales 7 tenían un nivel intermedio y 5 eran principiantes. Para esta aplicación se han extraído valores estadísticos de los datos de los sensores y a partir de ellos una actividad física se ha podido clasificar usando un algoritmo *SVM*. Entre las posibles actividades a clasificar están: press de banca, remo (aeróbico), press de banca con pesas, correr, extensiones lumbares en máquina, aperturas para pecho, press de banca en la máquina, cruce de polea (hacia abajo), curl bíceps con soga, curl bíceps con mancuernas, extensión de tríceps y elevaciones laterales. Este proyecto usa un algoritmo de aprendizaje automático supervisado para clasificar los movimientos, este método va a ser usado en nuestro proyecto con el objetivo de clasificar las repeticiones en correctas o incorrectas, pero con un algoritmo diferente. Ambos proyectos mencionados datan del 2017.

Siguiendo con el hilo del anterior artículo encontramos otro titulado *Tracking of strength-training: validation of a motion-recognition algorithm and a pilot towards 1RM, muscle loading and fatigue index using a smartwatch app* [23] publicado en 2018, y cuyos autores son Silvio René Lorenzetti y Dominik Huber. Este artículo sigue la temática del anterior [22] el cual consistía en crear una aplicación que determinara el tipo de ejercicio que se estaba realizando, pero con la particularidad adicional de que éste, aparte de lo mencionado, también evaluaba su precisión en conteo de repeticiones y se aplicaba un método para estimar la repetición máxima posible, el estrés muscular y la fatiga. Este proyecto fue desarrollado con el reloj inteligente: *Applewatch* y para la recolección de datos solo se hizo uso del acelerómetro a diferencia del anterior. El número de ejercicios que puede clasificar es menor al antes mencionado (press banca, curl biceps, prensa, sentadillas, elevaciones laterales) y el algoritmo de clasificación no es mencionado sino que es un algoritmo previamente creado por *FocusMotion* [7].

Otro artículo también interesante pero que no sigue la línea de los dos últimos es el artículo titulado: *Determining the single best axis for exercise repetition recognition and counting with smartwatches* [25] cuyos autores son Bobak Jack Mortazavi, Mohammad Pourhomayoun y Gabriel Alsheikh entre otros, este proyecto tiene como objetivo tratar de seleccionar el mejor eje único para cada ejercicio extrayendo únicamente las características más específicas de la actividad, a fin de minimizar la carga computacional y el consumo de energía en el conteo de repeticiones, además tiene la peculiaridad de que el conteo de repeticiones se realiza en tiempo real. Para localizar el eje más significativo se han recogido datos de 12 participantes con edades comprendidas entre los 23-38, que han realizado repeticiones de los ejercicios (curls biceps, encogimientos de abdomen, jumping jacks, flexiones y elevaciones laterales) a través de los sensores (acelerómetro y giroscopio) localizados en el reloj inteligente (*Samsung GalaxyGear*) en la muñeca izquierda. Una vez recolectados esos datos se ha utilizado el algoritmo de aprendizaje automático supervisado, denominado *Random Forest* [16] y a partir del modelo obtenido se puede obtener el eje más relevante.

Teniendo en cuenta todos estos artículos se puede ver que no hay ningún proyecto que

se encargue de verificar la correcta ejecución del ejercicio con un reloj inteligente, por lo que puede ser un buen nicho para desarrollar. Podemos observar como se utilizan modelos de aprendizaje automático para conseguir una experiencia de entrenamiento cómoda y eficaz a nivel de registro de actividades como se propone en los artículos *FitCoach* [22] y *Tracking of strength training* [23].

Además podemos apreciar que los datos obtenidos de los dispositivos suelen proceder del acelerómetro, giroscopio y magnetómetro, este último en el caso del primer artículo mencionado [20]. El uso de este último sensor puede ocasionar problemas a la hora de crear un software compatible con bastantes relojes inteligentes, ya que no todos lo poseen. Se puede observar en la tabla 2 las especificaciones que poseen algunos de los relojes del mercado. Podemos ver que 5 de 6 relojes poseen el acelerómetro y giroscopio, en cambio 1 solo posee magnetómetro, por lo que el uso de este limitará mucho la aplicación en la mayoría de los dispositivos.

En este caso vamos a utilizar el *Moto 360 2ª Gen* [13] ya que posee los sensores requeridos y ofrece un precio asequible.

NOMBRE	SISTEMA OPERATIVO	SENSORES RELEVANTES	MARCA	PRECIO EN EUROS
Apple Watch Serie 5 [5]	watchOS 6	Acelerómetro y Giroscopio	Appel	449
Moto 360 2ª Gen [13]	Android Wear OS	Acelerómetro y Giroscopio	Motorola	249
Amazfit GTR [1]	AmazFit OS	Acelerómetro, Giroscopio y Magnetómetro	Xiaomi	132,96
Galaxy Watch Active2 [8]	Tizen	Acelerómetro y Giroscopio	Samsung	299
Huawei Watch 2 [11]	Android Wear OS	Acelerómetro y Giroscopio	Huawei	449
Versa 2 [18]	Fitbit OS	Acelerómetro	Fitbit	199

Tabla 2.1: Tabla comparativa entre relojes inteligentes de distintas marcas. Fuente propia

Este proyecto, a diferencia del resto, va a tener en cuenta un solo ejercicio, las elevaciones laterales. Esto es así ya que: es un ejercicio muy común en los gimnasios, se puede medir con el reloj inteligente por el movimiento de brazos que se produce al realizarlo y además tiene la peculiaridad de que al ser un ejercicio accesorio para el entrenamiento, es decir, no es un ejercicio principal y compuesto, se debería realizar con una carga de peso liviana, pero no suele verse así lo que puede ser potencialmente lesivo.

Capítulo 3

Metodología

3.1. Planificación temporal y estimación de recursos necesarios

Para la planificación del proyecto se ha tenido en cuenta los dispositivos y herramientas software a utilizar, así como un tiempo de investigación sobre el sector para conocer los proyectos, artículos y aplicaciones ya existentes y si mi proyecto podía ofrecer alguna característica diferenciativa respecto al resto. La herramienta principal para el desarrollo del proyecto es *Android Studio*, usando java para su implementación, esta herramienta es totalmente gratuita por lo que no supone ningún costo.

Se ha tenido en cuenta el personal utilizado para este proyecto, que ha sido diseñado e implementado íntegramente por Andrés López Joya, teniendo en cuenta la participación del tutor a la hora de guiarlo. El hardware necesario es un dispositivo móvil con Android y el reloj inteligente.

Teniendo en cuenta lo dicho la planificación se va a dividir en dos etapas:

- Aprendizaje
 - Investigación del sector.
 - Aprendizaje básico de Android Studio.
 - Aprendizaje básico sobre los sensores del Smartwatch.
- Implementación
 - Implementación de la conexión bluetooth y envío de datos entre dispositivos.
 - Implementación de la base de datos en el dispositivo móvil.
 - Interfaz básica para pruebas y visualización de los datos.
 - Pruebas con sujetos y creación de algoritmo.
 - Evaluación del algoritmo.
 - Interfaz gráfica funcional y encuestas post-uso.

Cada etapa llevará un tiempo establecido para su desarrollo, para mostrarlo se va a hacer uso de un diagrama de Gantt (figura 3.1). Hay que destacar que la creación de la memoria se irá haciendo a la par con el proyecto.

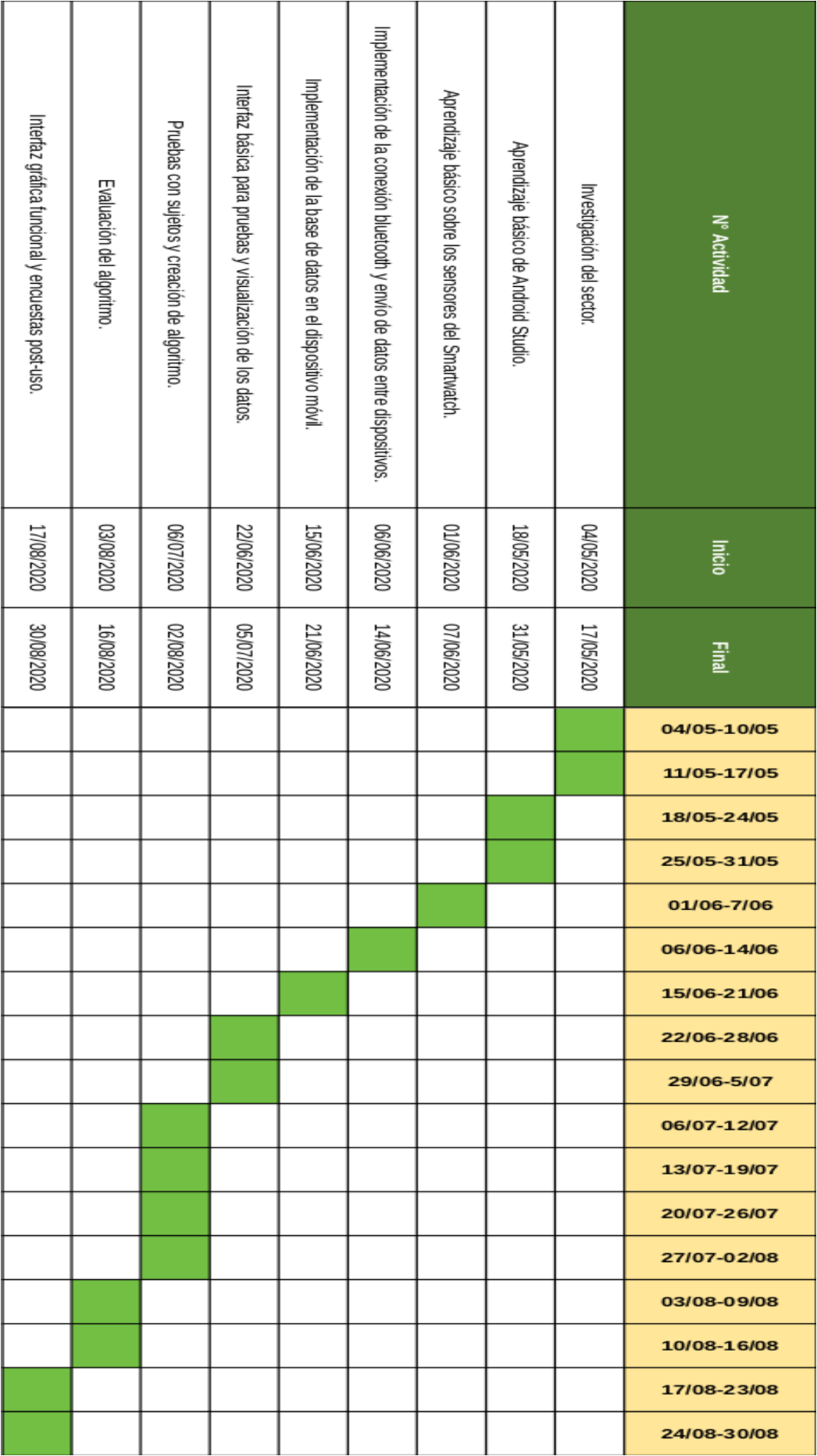


Figura 3.1: Diagrama de Gantt para la planificación del proyecto. Fuente propia

3.1.1. Presupuesto

Tipo del gasto	Precio unidad (Euros)	Importe total (Euros)
PARTIDA DE PERSONAL		
Sueldo desarrollador Junior	2000/mes	8000
PARTIDA FUNGIBLE		
-	-	-
PARTIDA INVENTARIABLE		
Herramientas, librerías y programas	Gratuitos	0
Reloj inteligente	249	249
Dispositivo Móvil	127,99	127,99
PARTIDA SERVICIOS TECNICOS		
-		
PARTIDA VIAJES Y DIETAS		
Desplazamientos	Gratuitos	0
PARTIDA DE OTROS		
Diseño aplicación	Gratuitos	0
Diseño logos	Gratuitos	0
GASTOS TOTALES		8376,99

Tabla 3.1: Tabla con presupuesto del proyecto. Fuente propia.

En el presupuesto se han tenido en cuenta varios datos, como el salario medio en España de un desarrollador, dato obtenido de *campusmvp* [6]. Además del costo en hardware necesario para desarrollar la aplicación, en este caso el dispositivo móvil y el reloj inteligente utilizado, también se han incluido los factores gratuitos, como son las librería y programas utilizados, desplazamientos y diseño de la aplicación y logos. Se puede apreciar el presupuesto más detallado en la tabla 3.1.

3.2. Diseño

3.2.1. Requisitos

Para definir el diseño se va a hacer uso de la priorización de requisitos software con *MoSCoW* [12] (M viene de Must en inglés y quiere decir que son obligatorios para el funcionamiento de la aplicación, la S representa Should, esto quiere decir que deberían estar para el uso del sistema pero que no son esenciales, la C hace referencia a Could y quiere decir que podrían estar, es decir, que no son importantes pero pueden estar y por último la W de Will not que quiere decir que no estarán en el sistema), este método es interesante para dividir los requisitos según sean más o menos importantes para el desarrollo del proyecto, en este caso no hay requisitos que no estarán (Will not).

Must - Obligatorios

- Crear perfil de usuario. Figura 3.2
- Iniciar conexión entre el reloj inteligente y dispositivo móvil. Figura 3.3

- Captar datos de los sensores del reloj inteligente, enviar los datos al dispositivo móvil, organizarlos y guardarlos. Figura 3.4
- Interpretar los datos guardados, obtener datos estadísticos de las repeticiones y clasificarlas dependiendo de los datos. Figura 3.5.

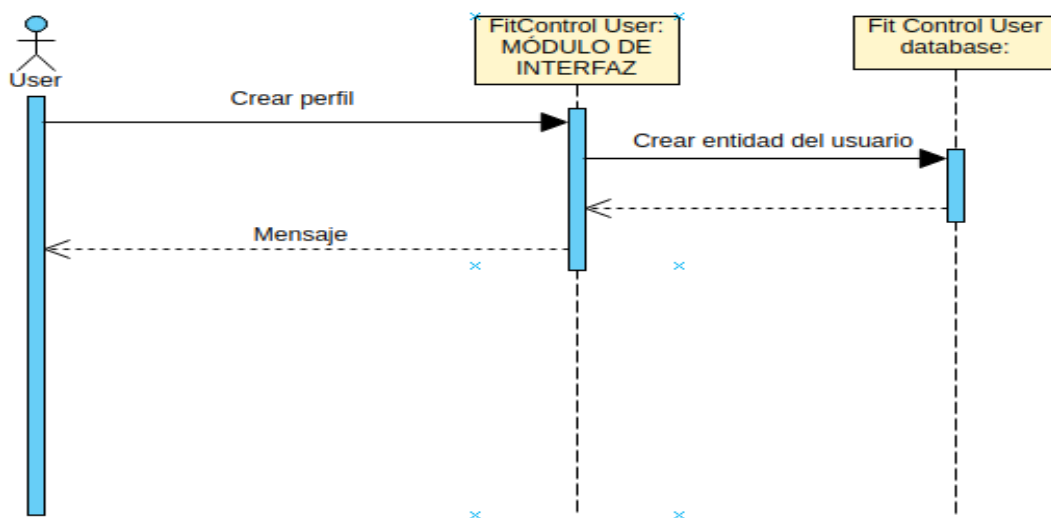


Figura 3.2: Crear nuevo perfil. Fuente propia.

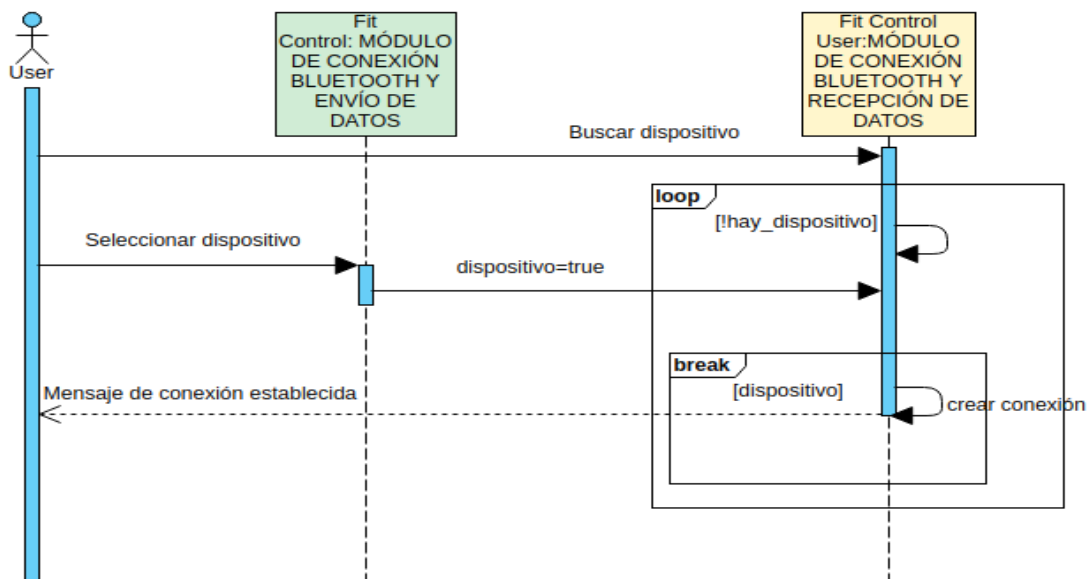


Figura 3.3: Iniciar conexión entre smartwatch y smartphone. Fuente propia.

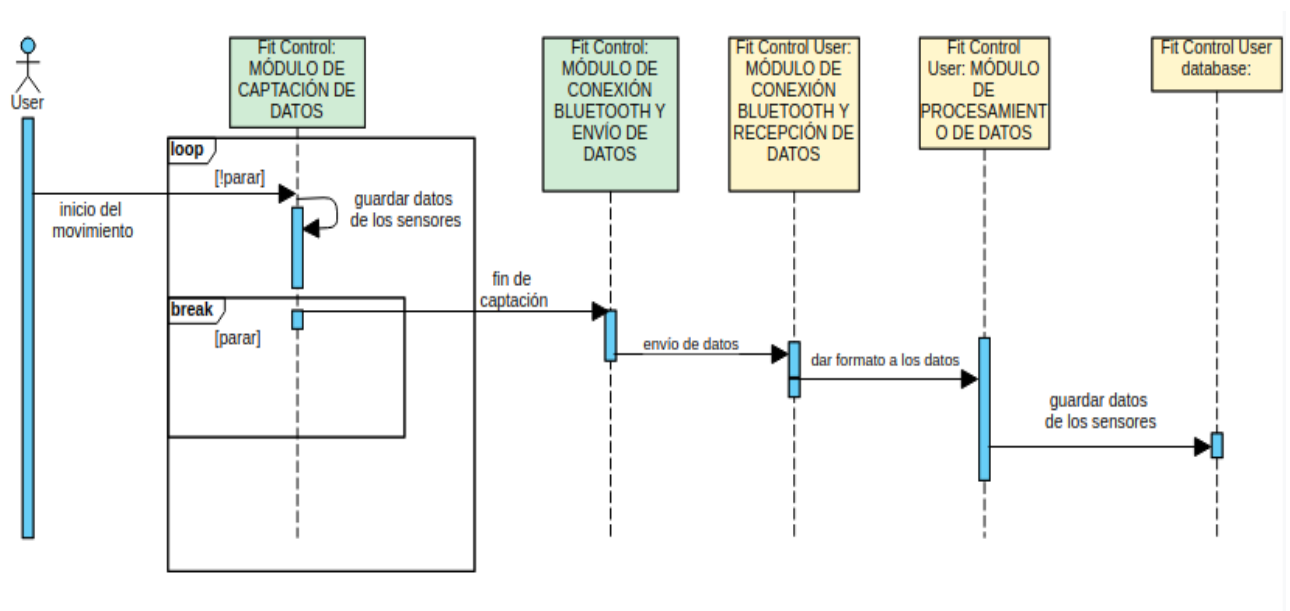


Figura 3.4: Captar datos del reloj inteligente y enviarlos a smartphone. Fuente propia.

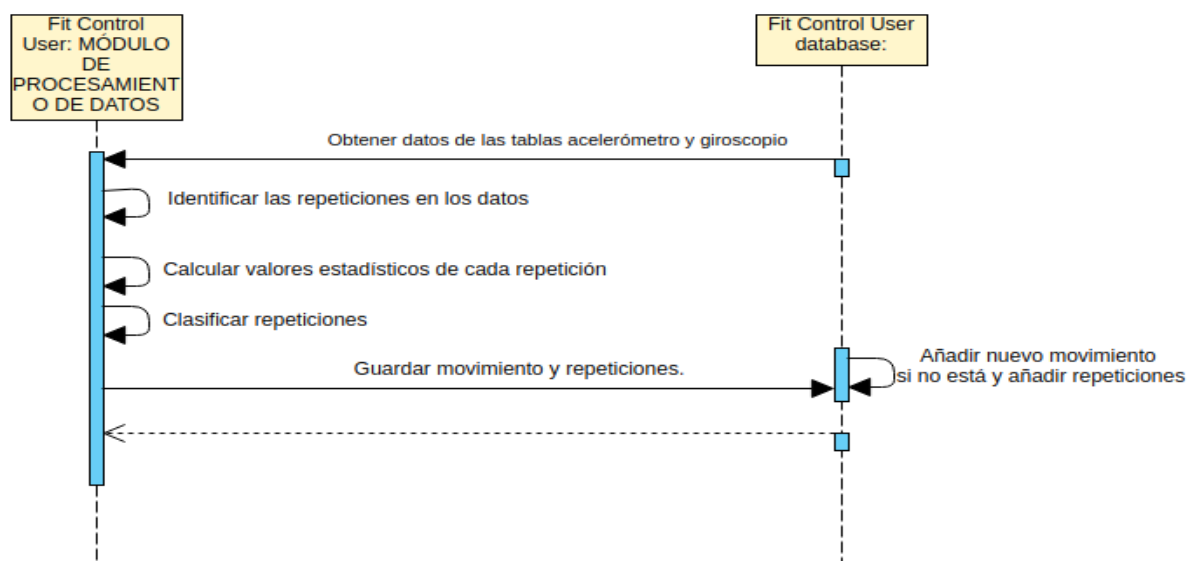


Figura 3.5: Procesar datos obtenidos y clasificar las repeticiones. Fuente propia.

Should - Deberían estar

- Consultar repeticiones totales de un ejercicio realizadas por un usuario. Figura 3.6.
- Consultar repeticiones de un usuario en una fecha. Figura 3.7

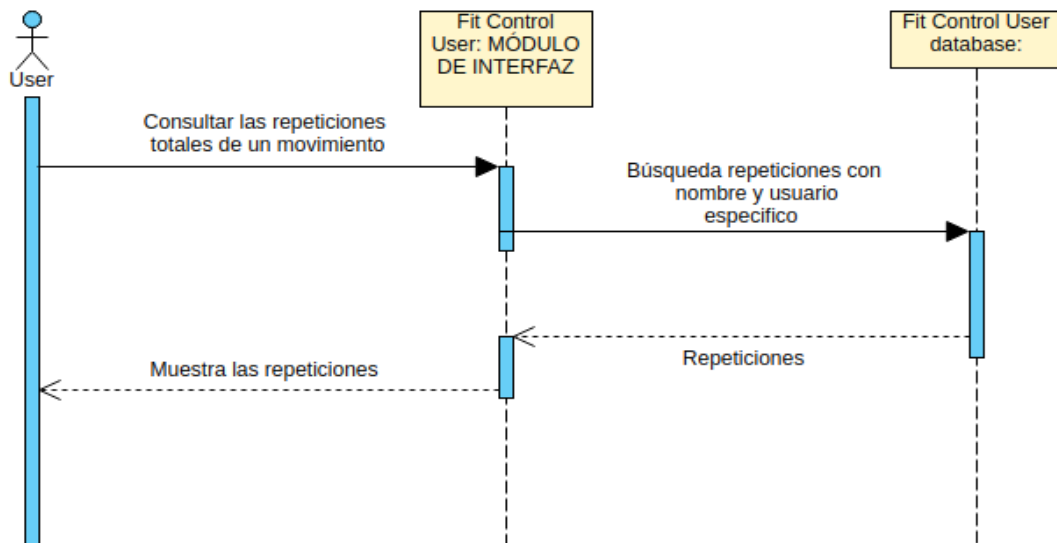


Figura 3.6: Consultar repeticiones totales de un movimiento. Fuente propia.

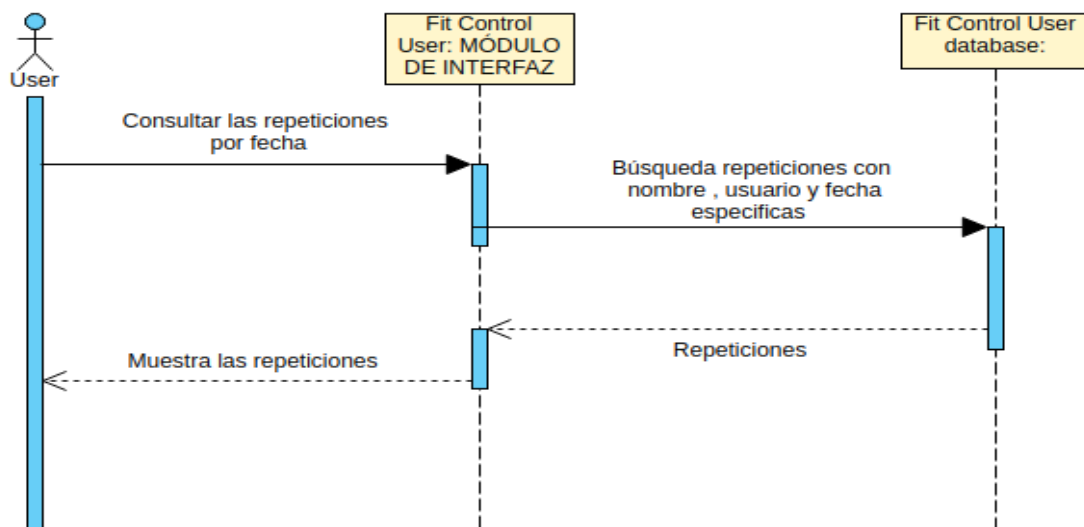


Figura 3.7: Consultar repeticiones de un usuario en una fecha. Fuente propia.

Could - Podrían estar

- Cambiar foto de perfil de usuario. Figura 3.8.
- Modificar datos de perfil de usuario. Figura 3.9.

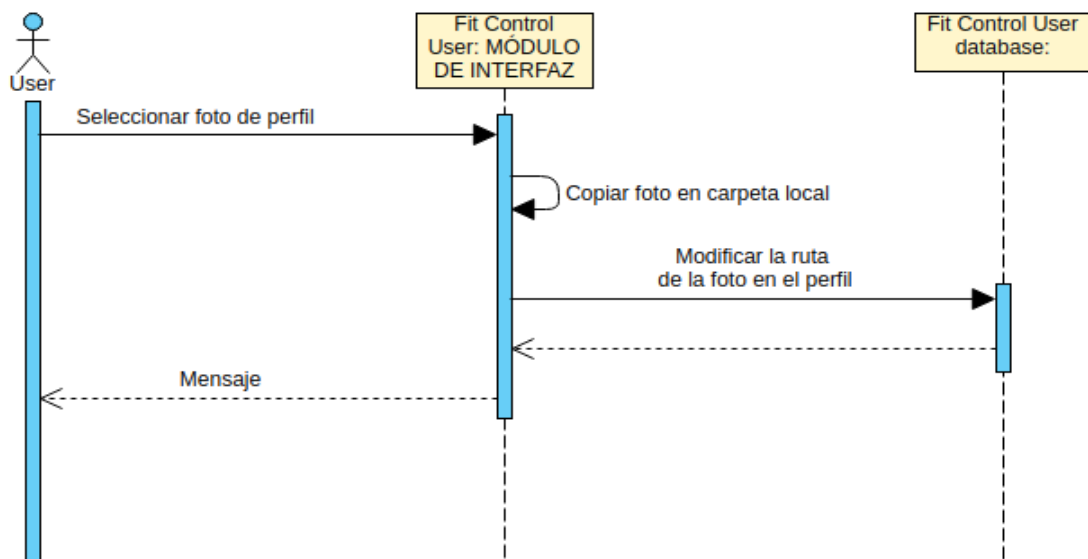


Figura 3.8: Cambiar foto de perfil. Fuente propia.

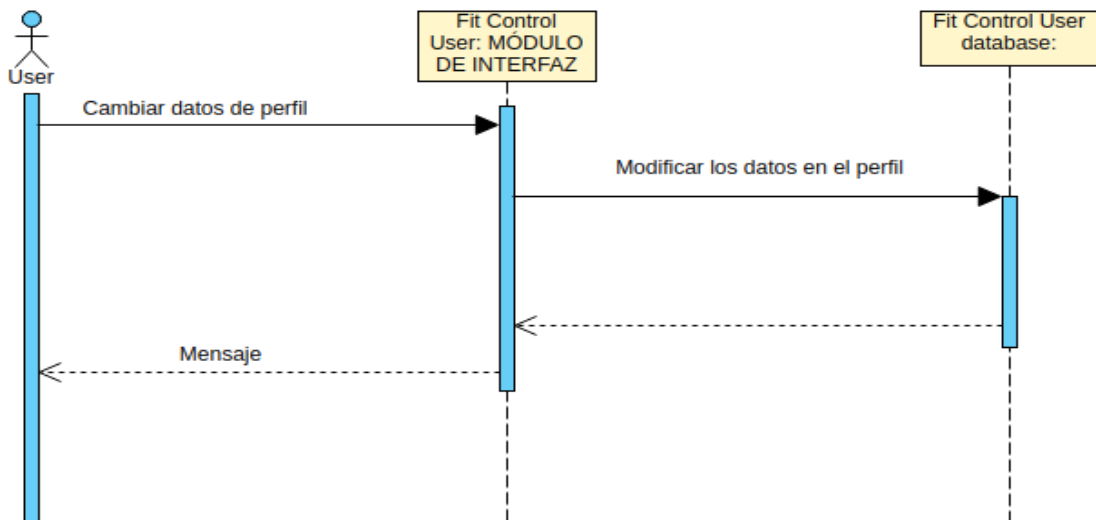


Figura 3.9: Cambiar datos del perfil. Fuente propia.

3.2.2. Arquitectura

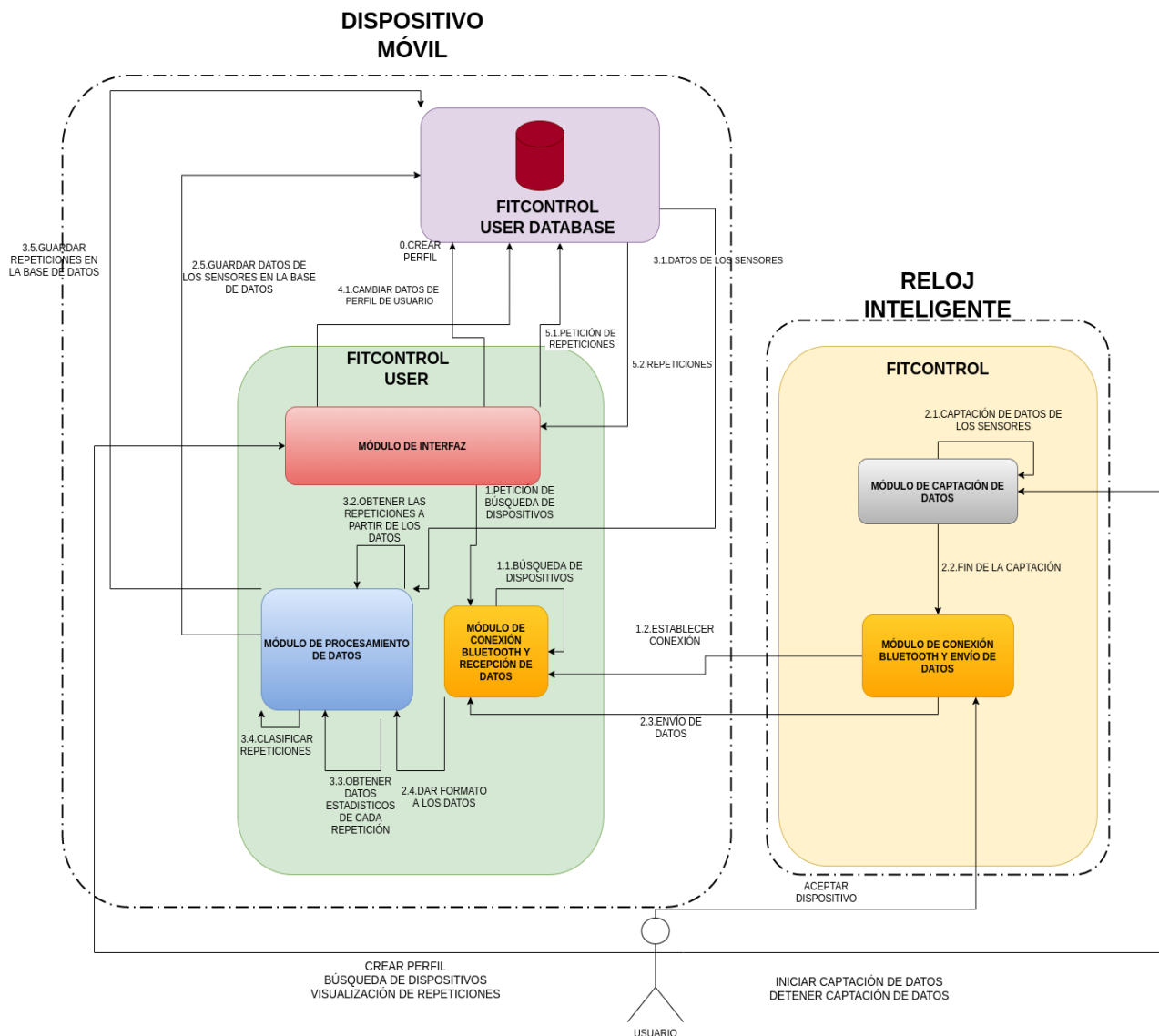


Figura 3.10: Esquema de la arquitectura del proyecto. Fuente propia.

Como se puede ver en la figura [3.10](#) la arquitectura del proyecto consta de dos grandes bloques (FitControl y FitControl User) y la base de datos de FitControl User. FitControl hace referencia a la aplicación desarrollada en el reloj inteligente y consta de dos módulos principales:

- **Módulo de captación de datos:** Este módulo se va a encargar de obtener los datos de los sensores en el reloj inteligente y mandarlos al *módulo de conexión bluetooth y envío de datos* una vez se haya indicado el fin de la captación y de selección del movimiento a realizar.
- **Módulo de conexión bluetooth y envío de datos:** En este módulo se van a desarrollar dos funciones. Por una parte se va a establecer la conexión con el *módulo de conexión bluetooth y recepción de datos* de FitControl User actuando de cliente y por otra se van a enviar los datos captados a dicho módulo.

- **Módulo de conexión bluetooth y recepción de datos:** Este módulo se va a encargar de establecer la conexión con el *módulo de conexión bluetooth y envío de datos* de FitControl actuando como servidor y por otro lado se va a encargar de recibir todos los datos enviados por FitControl.
- **Módulo de procesamiento de datos:** Este es el módulo más complejo del proyecto. Sus funciones son 4: darle formato a los datos recibidos para poder guardarlos en la base de datos, interpretar dichos datos para identificar las repeticiones realizadas, obtener datos estadísticos de cada repetición registrada y por último clasificar cada repetición usando dichos datos. Una vez realizado todo el proceso se guardan las repeticiones en la base de datos.
- **Módulo de interfaz:** En este módulo se van a realizar todas las actividades que van a ser funcionalidades de la interfaz como crear usuarios o consultas a la base de datos.
- **Usuario:** Este es el actor del sistema y tiene dos funciones una en el reloj inteligente y otra en el dispositivo móvil: en la parte del reloj inteligente puede iniciar y detener la captación de datos del reloj, para ello interactúa con el *módulo de captación de datos* y en la parte del dispositivo móvil interactúa con el *módulo de interfaz* para realizar las distintas funcionalidades disponibles (crear perfil, búsqueda de dispositivos y visualización de repeticiones).

El flujo correcto para el funcionamiento del sistema es:

1. **Crear un perfil**, para ello el usuario interactúa con el *módulo de interfaz* en el dispositivo móvil, que este a su vez guarda una nueva entidad de perfil en *FitControl User Database* con los datos ofrecidos por el usuario.
2. **Buscar dispositivos Bluetooth en el dispositivo móvil.** El usuario interactúa con el *módulo de interfaz* y este manda una petición de búsqueda a el *módulo de conexión bluetooth y recepción de datos*, lo que inicia la búsqueda.
3. **Aceptar la conexión en el reloj inteligente.** El usuario manda la petición al *módulo de conexión bluetooth y envío de datos*, que este a su vez acepta la conexión entre el dispositivo móvil y el reloj inteligente.
4. **Captación y envío de datos.** Con la conexión ya creada el usuario puede iniciar la captación de datos en el reloj inteligente, para ello se hace uso del *módulo de captación de datos*. Con todos los datos necesarios el usuario detiene la captación y manda una señal al *módulo de conexión bluetooth y envío de datos*. Cuando la señal es recibida el módulo mencionado envía los datos recabados a el *módulo de conexión bluetooth y recepción de datos*. Desde este último módulo se mandan los datos a el *módulo de procesamiento de datos* donde se le da el formato necesario y posteriormente son guardados en la base de datos.
5. **Procesamiento de los datos.** Para poder realizar este procesamiento se envían los datos previamente guardados al el *módulo de procesamiento de datos*, donde se obtienen las repeticiones. A partir de estas se calculan los datos estadísticos necesarios y se realiza la clasificación de las repeticiones a partir de estos datos. Posteriormente estas se guardan en la base de datos.

6. **Visualización de los datos.** Este sería el último paso en el flujo principal de la aplicación. El usuario manda una petición al módulo de interfaz y este a su vez obtiene las repeticiones pertinentes de la base de datos.

3.3. Implementación

El nombre dado a la aplicación desarrollada para dispositivo móvil es FitControl User y FitControl es el nombre que tiene la aplicación desarrollada para reloj inteligente.

3.3.1. FitControl User

Esta parte del proyecto ha sido desarrollada en el dispositivo móvil (Redmi Note 5).

Base de datos

La base de datos está implementada en *Room* [17] la cual proporciona una capa de abstracción sobre *SQLite*, ya que esta proporciona unas facilidades y comodidades bastante notorias a la hora de crear nuevas tablas y hacer consultas, entre otras funcionalidades. Esta base de datos va a constar de 5 tablas Acelerómetro(A), Giroscopio(G), Perfil(P), Movimiento(M), Repeticiones(R). Figura 3.11.

Las tablas A y G representan el estado de los sensores que han sido enviados desde FitControl en un momento exacto durante la ejecución de las repeticiones, las variables de esta tabla van a representar:

- El **instante** en nanosegundos en el que se ha captado.
- El **valor** en m/s al cuadrado de los **3 ejes** x,y,z en ese instante.
- Una **etiqueta** que indicará:
 - Si en ese momento se encuentra realizando una repetición (mayor que 0).
 - Si en ese momento no se ha realizado ninguna repetición o si en ese momento se han realizado todas (-1).
 - Si es un momento en el que se encuentra en un estado de reposo entre una repetición y otra.

Estas dos tablas están creadas para un fácil manejo de los datos. A medida que reciben información de FitControl, las tablas se irán vaciando para evitar el exceso de información innecesario y la saturación de la base de datos.

Por otra parte tenemos las tablas P, M y R, estas tablas van a estar relacionadas tal y como se muestra en la figura 3.11.

Cada perfil puede tener varias entidades de la clase Movimiento, pero solo una por día. Así se pueden agrupar las repeticiones de manera más cómoda y sin tener tantas entidades de la clase Movimiento y ahorrando en espacio de almacenamiento.

La tabla perfil posee :

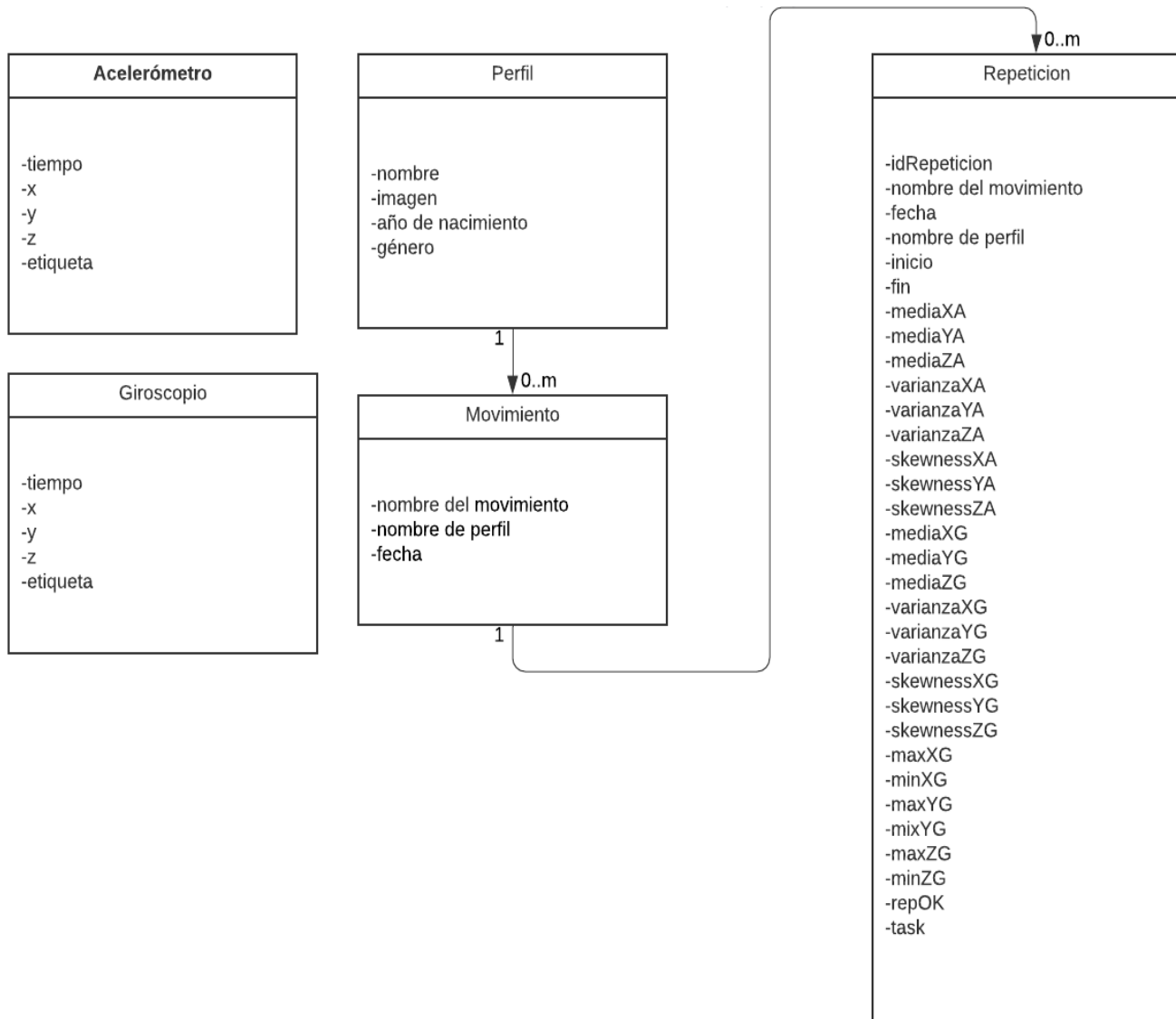


Figura 3.11: Diagrama UML de las tablas de la base de datos de FitControl User. Fuente propia.

- Un **nombre de usuario**, el cual va a ser la clave primaria de nuestra tabla. Este nombre es único por lo que no puede haber dos que se llamen igual.
- Una **imagen** de perfil, cuya dirección de almacenamiento está guardada en forma de cadena de texto en la variable image.
- Dos variables **año de nacimiento y género**, las cuales están para posibles distinciones de genero o de edad en algún punto futuro.

Cada entidad de la tabla Movimiento agrupa las repeticiones de ese movimiento realizadas por un usuario concreto en una fecha concreta, por lo que en un mismo día no puede haber dos entidades del mismo usuario y del mismo movimiento. La tabla Movimiento posee:

- El **nombre del movimiento** del cual se han ejecutado las repeticiones.

- El **nombre de usuario** que ha realizado el movimiento en cuestión.
- La **fecha** de realización del movimiento ya que solo puede haber una entidad del mismo perfil y mismo movimiento por día.

Por último tenemos la tabla Repetición. Esta tabla es extensa ya que posee todos los datos estadísticos de cada repetición realizada, junto con los valores necesarios para localizar la repetición en el sistema. La tabla Repetición posee:

- Un **identificador** auto-generado para cada repetición realizada.
- El **nombre del movimiento** del que se hace la repetición para poder localizarlo.
- El **nombre de usuario** que ha realizado dicha repetición.
- La **fecha de ejecución** de la repetición.
- La **hora** en nanosegundos en la que la repetición **empezó** a ejecutarse.
- La **hora** en nanosegundo en la que la repetición **terminó** de ejecutarse.
- Tres atributos que representan **la media** de todos los valores correspondientes a esa repetición para cada eje x,y,z.

$$\bar{x}_y = 1/n * \sum_{i=1}^n x_i$$

- Tres atributos que representan **la varianza** de todos los valores correspondientes a esa repetición para cada eje x,y,z.

$$\bar{\sigma}_y = 1/n * \sum_{i=1}^n (x_i - \bar{x}_y)^2$$

- Tres atributos que representan **el skewness** de todos los valores correspondientes a esa repetición para cada eje x, y, z, para estos valores he utilizado una librería ya implementada en math3.
- Dos atributos que van a representar el **máximo** y el **mínimo** valor en cada eje.
- Un atributo que va a definir la clasificación de como se ha realizado la repetición puede tomar valores desde el 0 hasta el 2 incluidos.

Conexión Bluetooth

Este proyecto está compuesto por dos aplicaciones android, una para reloj inteligente y otra para dispositivo móvil. Para el correcto funcionamiento de la aplicación deben interactuar entre si. Para hacer posible la conexión y la transferencia de datos entre los dos dispositivos se ha utilizado Bluetooth .

Hay que destacar que para que el reloj inteligente interactúe con el dispositivo móvil hay que hacer una conexión previa mediante la aplicación *Watch OS* que ofrece este reloj inteligente.

En este caso el dispositivo móvil actúa como servidor abriendo la búsqueda a otro dispositivo, tal y como se muestra en la figura 3.12 (la aplicación en el dispositivo móvil se llama FitControl User y la aplicación en reloj inteligente FitControl).



Figura 3.12: Búsqueda de dispositivos por el dispositivo móvil. Fuente propia

Con el dispositivo encontrado y el canal de comunicación abierto se pueden captar los datos de los sensores, guardarlos en el reloj y enviarlos a través del canal al dispositivo móvil donde se procesarán dichos datos.

En el dispositivo móvil a raíz de los datos recibidos se identifica cada repetición y se obtienen los diferentes datos estadísticos. Mediante estos datos se clasifican las repeticiones y se guardan en una base de datos. Figura 3.13.



Figura 3.13: Búsqueda de dispositivos por el dispositivo móvil. Fuente propia

Como primer paso del proceso de captación de datos, el dispositivo móvil tiene que aceptar al reloj inteligente, para ello se ha hecho uso de la librería que ofrece android para conexiones bluetooth [3].

Primero se inicializa una variable tipo *BluetoothAdapter* la cual va a representar el adaptador Bluetooth del dispositivo y con el que vamos a poder crear nuestro objeto de la clase *AcceptThread*.

```
1 acceptTread = new AcceptTread(blueetoothAdapter)
```

Al inicializar esta variable se va a crear un *socket*(tmp) con una UUID fija, a través de la función *.listenUsingRfcommWithServiceRecord()*, tal y como se muestra en el siguiente código.

```
1 tmp = bluetoothAdapter.listenUsingRfcommWithServiceRecord("conexion", UUID.  
    fromString("fa87c0d0-afac-11de-8a39-0800200c9a66"));
```

Esta UUID va a ser usada por el reloj inteligente para poder encontrar este canal creado y poder comunicarse.

Una vez creado el canal hay que aceptarlo. Para ello vamos hacer una llamada a la función *.run* del objeto *acceptThread*; esta inicia una búsqueda de dispositivos con la misma UUID previa petición de enlace.

```
1 acceptTread.run();
```

Una vez encontrado el dispositivo ya estaría el canal abierto para poder transferir datos, sin embargo falta la parte de comunicación entre ambos dispositivos, para ello se va a hacer uso de la clase *MyBluetoothService*, en este caso el dispositivo móvil actúa como servidor, por lo que solo va a recibir datos, no enviarlos. Al crear el objeto de esta clase se va a inicializar una variable con el flujo de datos de entrada del *socket* para poder obtener los datos que hay en él.

```
1 tmpIn = socket.getInputStream();
```

Una vez recibidos solo hay que pasar los datos que hay en el flujo a una variable, tal y como se muestra en el siguiente cuadro de código.

```
1 numBytes = mmInStream.read(mmBuffer);  
2 String readMessage = new String(mmBuffer, 0, numBytes);  
3 aux += readMessage;
```

Esta variable contiene todos los datos recibidos, pero no está localizada en la clase principal por lo que hay que pasarla a ella a través de un *handler* para poder manejarlos, esto se hace mediante la función *.run* del objeto de la clase.

```
1 Message readMsg = handler.obtainMessage(  
2 Utilidades.MESSAGE_READ, 0, -1, aux);  
3 readMsg.sendToTarget();
```

Procesamiento de datos

Hay que darle el formato adecuado a los datos en la clase principal (objetos de tipo acelerómetro y giroscopio respectivamente), ya que el valor obtenido es una cadena de texto. Esto es posible pues esta organizada por bloques, los cuales están separados por algún carácter especial. Mediante la función *.dividir()* se pueden fraccionar los datos de la cadena de texto.

El formato de la cadena de texto es el siguiente:

- Cadena de texto = Nombre del movimiento % Datos de los sensores
- Datos de los sensores = Datos acelerómetro # datos giroscopio
- Datos acelerómetro/giroscopio = Bloque1 / Bloque2 / ... / Bloque(N-1) / Bloque(N)
- Bloque(i) = Tiempo(i), x(i), y(i), z(i), etiqueta(i)

A partir de esos datos se crean objetos de tipo acelerómetro y giroscopio y se guardan en la base de datos para poder tener un manejo de estos más cómodo.

Una vez guardados los datos en su correcto formato se tratan para poder obtener la información relevante y así clasificarlos mediante el algoritmo que se explicará posteriormente.

Interfaz

Esta aplicación consta de una interfaz de usuario personalizable. Nada más entrar a la aplicación podemos ver los perfiles creados que hay en el dispositivo tal y como se puede apreciar en la figura 3.14, en este caso hay un solo perfil creado cuyo nombre es andres. Para crear un nuevo perfil debemos pulsar la cruz blanca en la que pone *Add Profile*, esta nos llevará a una nueva actividad (figura 3.2) donde puedes crear un nuevo perfil introduciendo los valores deseados siempre y cuando el no haya un perfil con un nombre de usuario igual al creado.

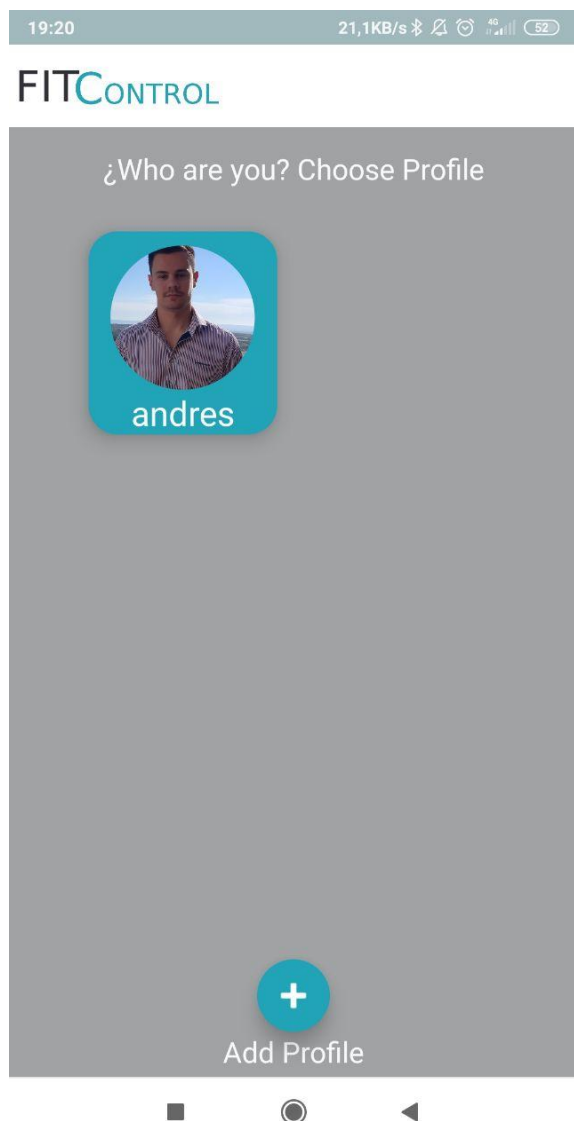


Figura 3.14: Primera actividad de la aplicación. Fuente propia.

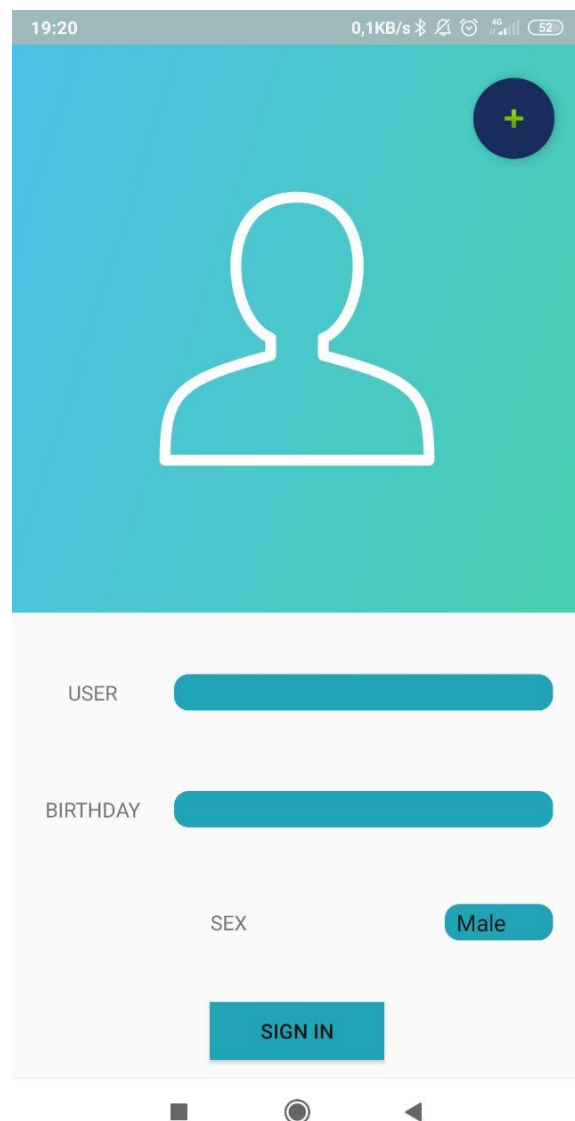


Figura 3.15: Actividad para crear nuevo perfil de usuario. Fuente propia.

Una vez creado nuestro perfil nos aparecerá el *home* (figura 3.16), donde hay un diagrama en pastel con el número de repeticiones totales que se han realizado de manera correcta y de manera incorrecta por ese perfil. Además esta actividad ofrece más información, como el nivel en el que se encuentra el usuario (jerarquía de niveles propia), el movimiento más frecuente y una consejo general. Si deseamos cambiar algún dato relacionado con nuestro perfil, como nuestra foto, fecha de nacimiento o sexo, basta con pulsar los tres botones que aparecen en la parte superior derecha a la derecha y se nos redirigirá a la siguiente actividad donde podremos modificarlos (figura 3.17) .

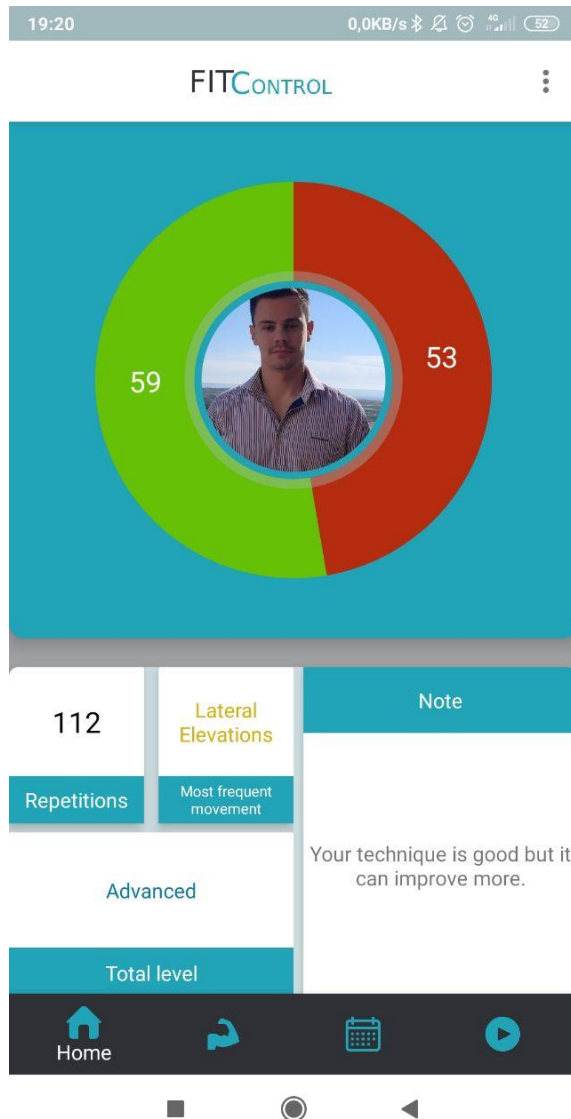


Figura 3.16: Apartado home de la aplicación en un perfil concreto. Fuente propia.

The screenshot shows the profile modification screen of the FITCONTROL app. At the top, the status bar displays the time 19:22, data usage 0,2KB/s, and battery level 51%. The screen features a large photo of a man. Below the photo, the name 'andres' is displayed. The 'Information' section includes fields for 'Birthday' (6 / 7 / 2020) and 'Sex' (Male). A 'SAVE CHANGES' button is located at the bottom.

Figura 3.17: Actividad para modificar nuestro perfil. Fuente propia.

La aplicación dispone de un navegador, en la parte inferior de la pantalla, el cual nos permite desplazarnos por los diferentes campos que tiene la aplicación. En el apartado *Movements* (figura 3.18) podemos ver los diferentes ejercicios que hemos realizado, cabe recalcar que en la aplicación solo está implementada la elevación lateral y que el curl de bíceps es simplemente visual. Al seleccionar el movimiento de elevaciones laterales nos aparecen todas las repeticiones que se han realizado de ese ejercicio por ese perfil (figura 3.19). Al seleccionar cualquier repetición en la pantalla resultante nos ofrece una explicación a la incorrecta realización de la repetición.

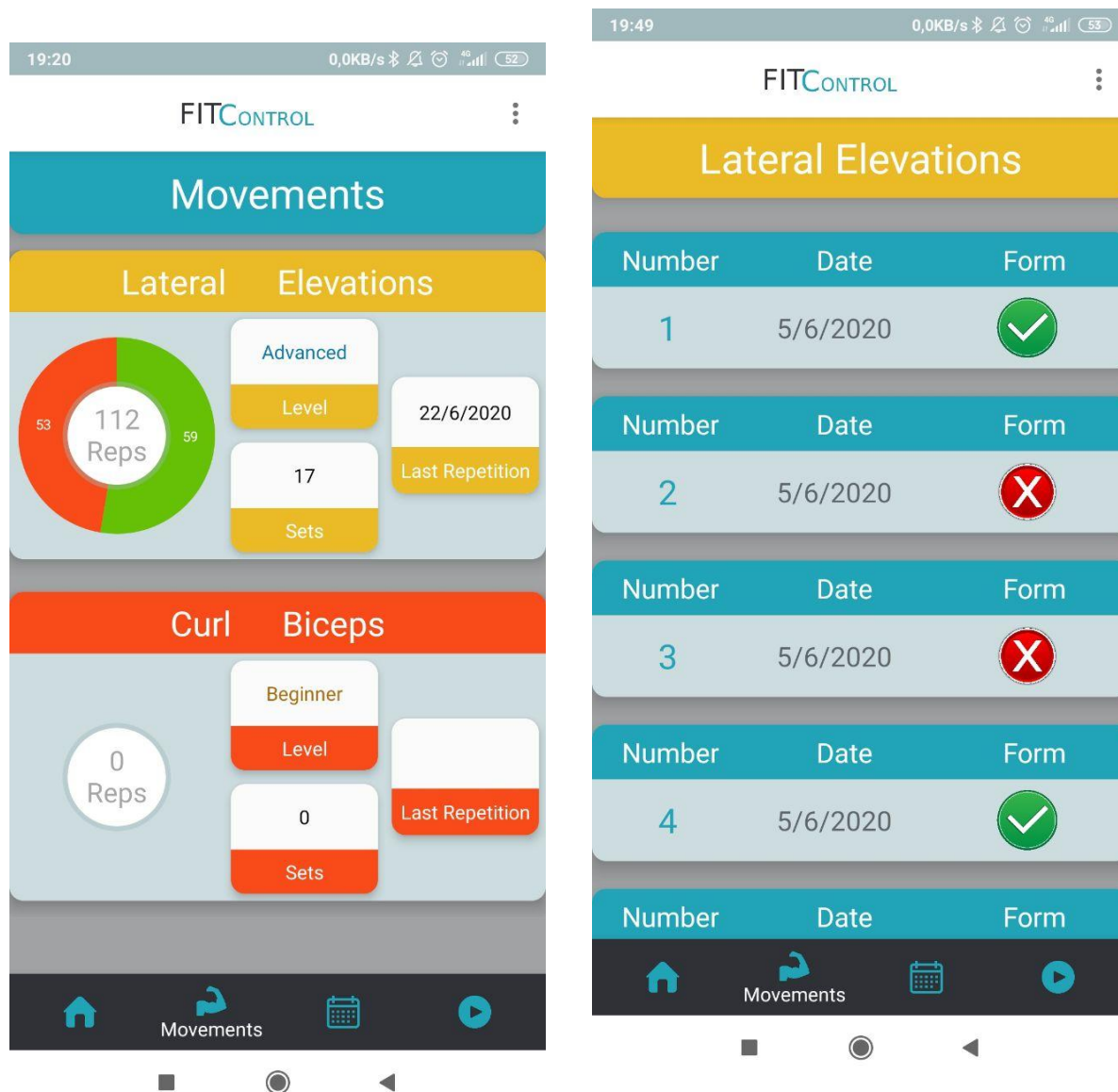


Figura 3.18: Apartado Movements de la aplicación. Fuente propia.

Figura 3.19: Apartado en el que se muestran todas las repeticiones de elevaciones laterales. Fuente propia.

Si seguimos avanzando en el navegador podemos ver un icono de un calendario, en el cual como vemos en la imagen [3.20](#) aparece un calendario en el que están todas las repeticiones divididas por días. Para obtener la cantidad de repeticiones que se han hecho, solo hay que seleccionar sobre la barra en el día que se desee tal y como se muestra en la figura [3.21](#).

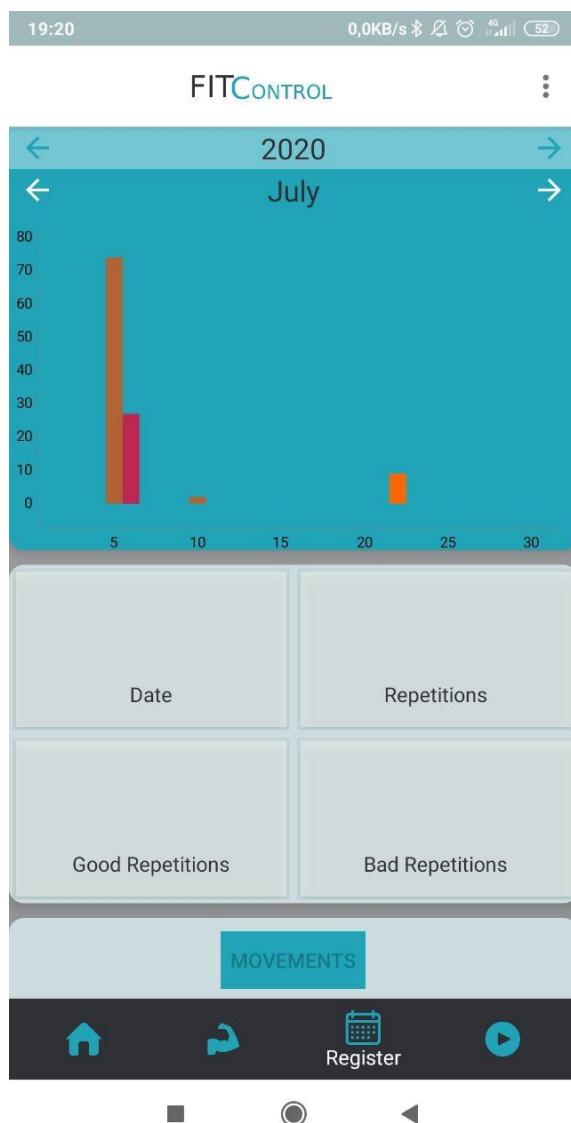


Figura 3.20: Apartado Register de la aplicación sin seleccionar fecha. Fuente propia.

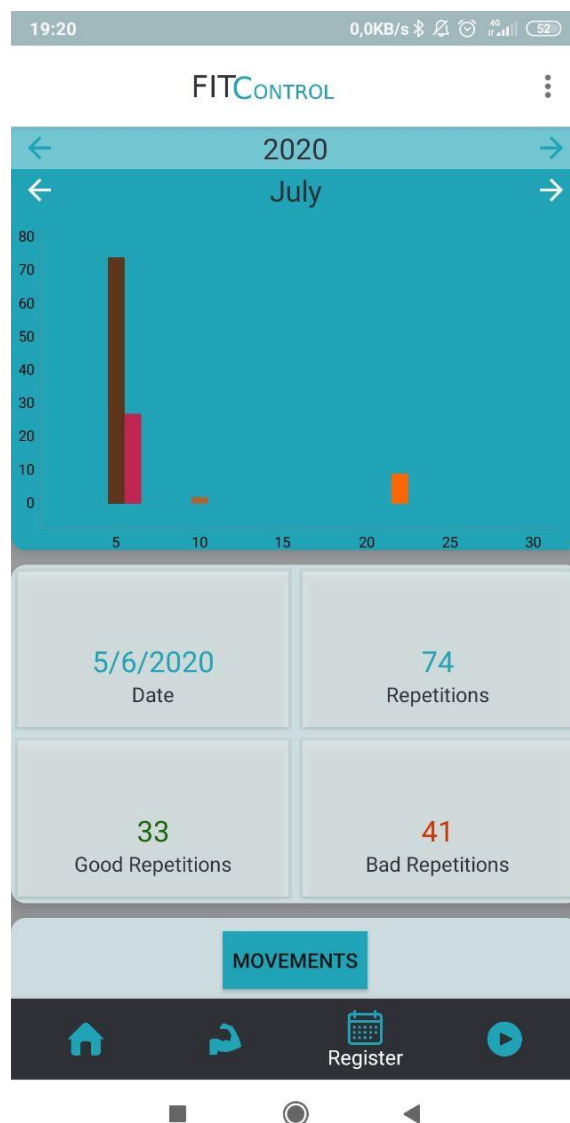


Figura 3.21: Apartado Register de la aplicación seleccionando fecha. Fuente propia.

Al seleccionar el botón *movements* se muestra un nuevo fragmento donde aparecen las diferentes repeticiones y series que se hicieron de ese ejercicio en ese día (figura 3.22), y de la misma forma que pasaba en el apartado *Movements* al seleccionar en el movimiento deseado se muestran todas las repeticiones realizadas con la posibilidad de seleccionar cada una lo que justificara su clasificación. Figura 3.23

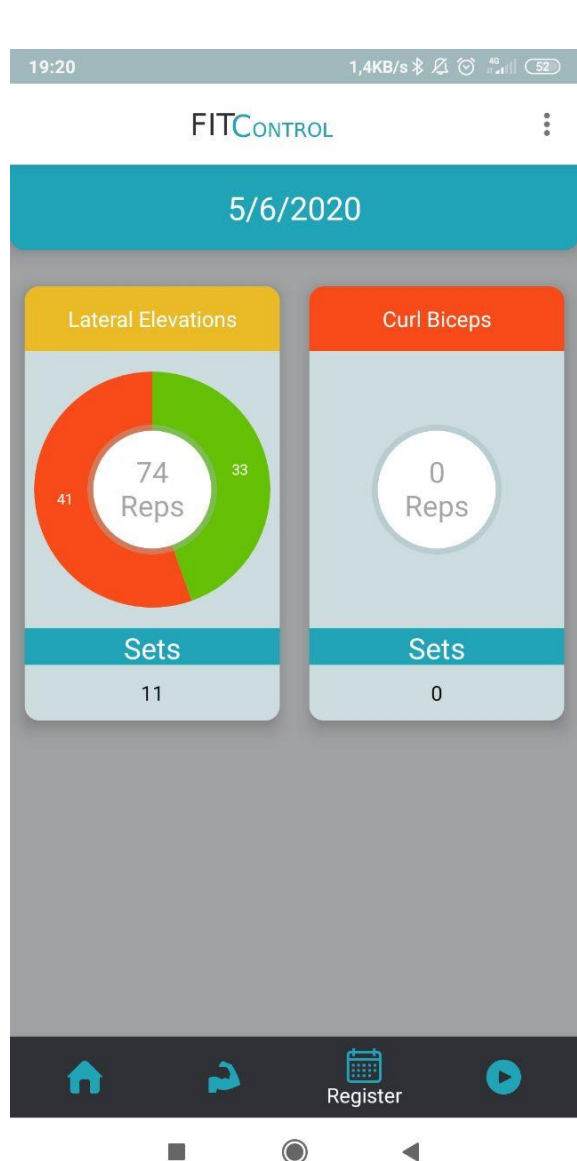


Figura 3.22: Apartado Register de la aplicación en una fecha. Fuente propia.



Figura 3.23: Apartado Register de la aplicación en una fecha y un movimiento seleccionado. Fuente propia.

Por último el apartado de *Receiver* inicia la búsqueda de dispositivo y se desarrolla el procesamiento de datos. Como se puede observar en la figura 3.24 cuando iniciamos este fragmento el dispositivo comienza a realizar la búsqueda del reloj y una vez la conexión establecida el punto rojo pasa a verde y recuerda que para que los datos se envíen se debe pulsar el botón verde en el reloj inteligente. Figura 3.25

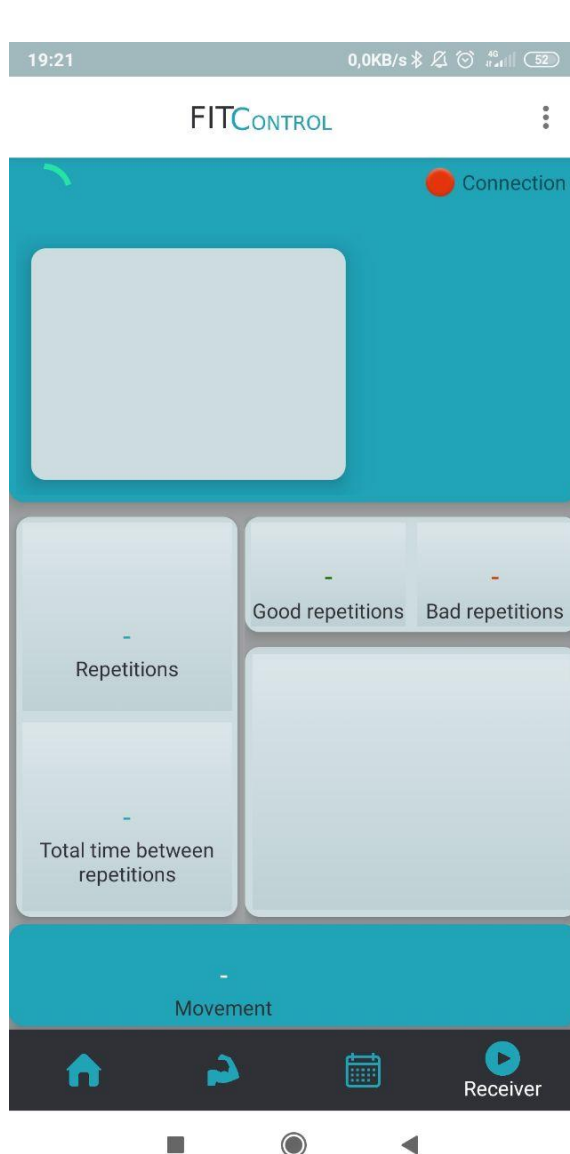


Figura 3.24: Apartado Reciver de la aplicación buscando dispositivo. Fuente propia.

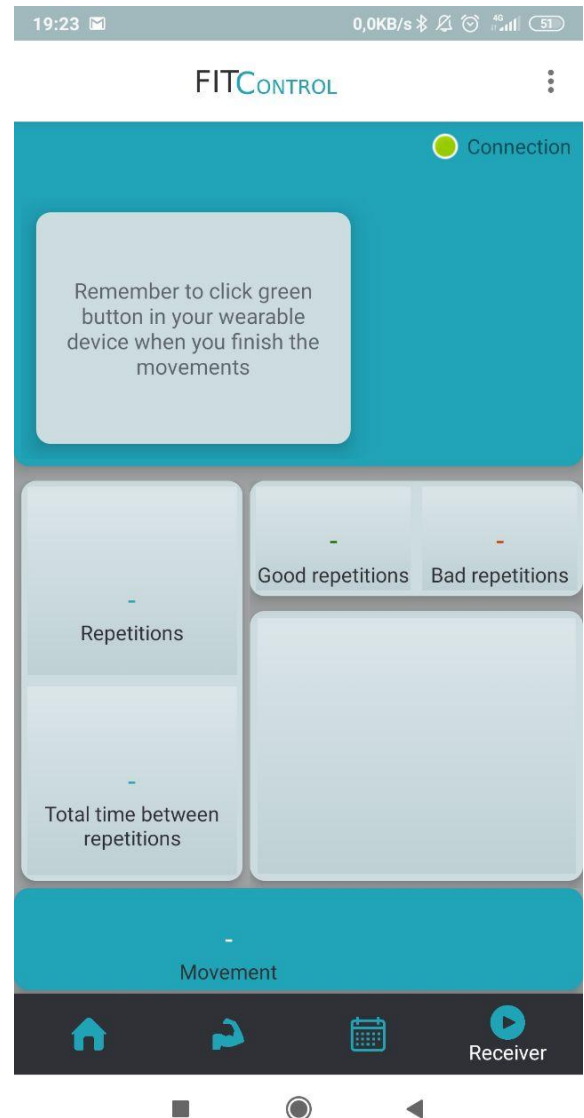


Figura 3.25: Apartado Reciver de la aplicación con dispositivo encontrado. Fuente propia.

Una vez que se ha producido la conexión y se han enviado los datos, el dispositivo los procesa y muestra la información relevante, como el número de repeticiones realizadas, el número de repeticiones realizadas correctamente e incorrectamente, pudiendo seleccionar cada una para saber el porqué de la clasificación correcta o incorrecta, el tiempo entre repeticiones (es decir el tiempo que esta la pesa en estado de reposo) y el nombre del movimiento realizado, tal y como lo ilustra la figura 3.26.

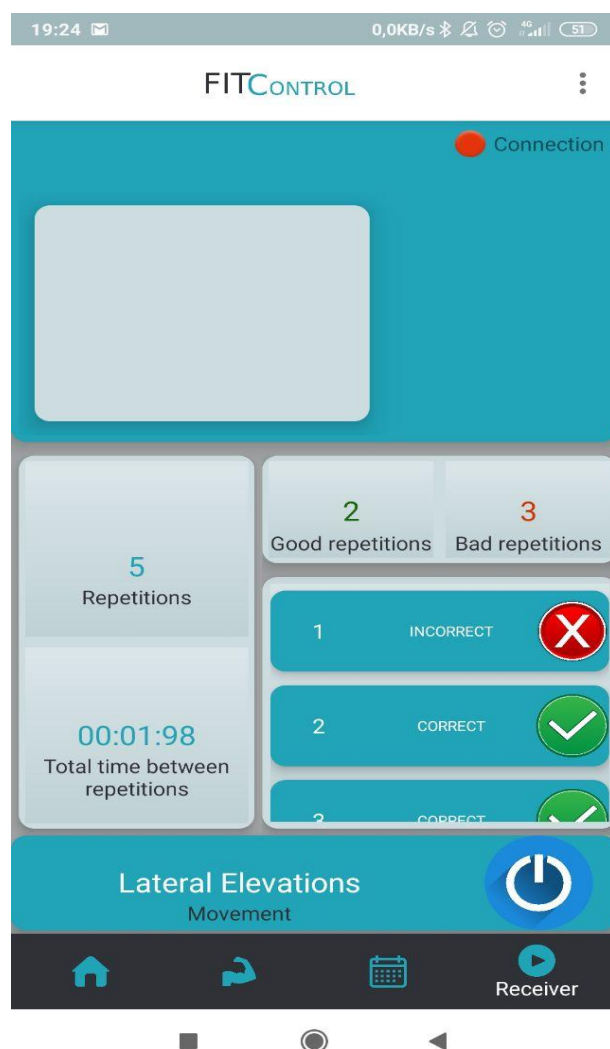


Figura 3.26: Apartado Reciver una vez procesadas las repeticiones. Fuente propia.

3.3.2. FitControl

Esta parte del proyecto ha sido desarrollada en el reloj inteligente *moto 360 2nd Generation*.

Conexión bluetooth y envío de datos

Este punto del proyecto es más simple que FitControl User al no poseer ningún tipo de procesamiento de datos.

FitControl identifica todos los dispositivos bluetooth de los que dispone, utilizando la misma librería que FitControl User que es la que ofrece android para el bluetooth. Seguidamente se los muestra al usuario para que pueda escoger el dispositivo deseado. Esto se realiza en la clase *ChooseDevice* cuya función es guardar la dirección MAC del dispositivo seleccionado en un archivo en preferencias. Para ello son utilizadas las siguientes líneas de código:

```
1  SharedPreferences prefs =getSharedPreferences("PreferenciasDevice", Context.  
    MODE_PRIVATE);  
2  SharedPreferences.Editor editor = prefs.edit();  
3  editor.putString("device",listOfDevice.get(value).getAddress());  
4  editor.commit();
```

Una vez guardada la MAC en el archivo se puede acceder en todo momento. Para comparar entre todos los dispositivos desde la clase principal hay para saber cual es el seleccionado y así tener el dispositivo deseado.

```
1  Set<BluetoothDevice> pairedDevices = bluetoothAdapter.getBondedDevices();  
2  SharedPreferences preferencias =getSharedPreferences("PreferenciasDevice", Context.  
    MODE_PRIVATE);  
3  mac=preferencias.getString("device", "device");  
4  
5  if (pairedDevices.size() > 0) {  
6  // There are paired devices. Get the name and address of each paired device.  
7      for (BluetoothDevice device1 : pairedDevices) {  
8          String deviceHardwareAddress = device1.getAddress(); // MAC address  
9              if(deviceHardwareAddress.equals(mac)){  
10                 device=device1;  
11             }  
12         }  
13     }
```

Una vez identificado el dispositivo se procede a crear la conexión que se debe hacer una vez que la aplicación FitControl User esté buscando dispositivos, para ello se crea un objeto de tipo *ConnectThread* pasandole el dispositivo y posteriormente llamando a la función *run*, tal y como se muestra en la siguiente figura .

```
1  conex = new ConnectThread(device);  
2  conex.run(bluetoothAdapter);
```

Establecido el canal hay que crear la parte de comunicación con el dispositivo móvil, para ello, al igual que antes, hacemos uso de la clase *MyBluetoothService* con la peculiaridad de que en vez de inicializar con el flujo de entrada la variable, se va a hacer con el flujo de salida para así poder escribir en el canal y que le sea posible al dispositivo móvil captar los datos.

```
1 tmpOut = socket.getOutputStream();
```

Para obtener los datos de los sensores y poder así enviarlos se ha usado la librería de android para hardware la cual permite trabajar con objetos de tipo sensor [4]. Esta librería permite además interactuar cuando cierto tipo de sensor varíe mediante la función *onSensorChanged(SensorEvent event)*, de esta manera se puede guardar la información necesaria del sensor en un momento determinado para posteriormente poder enviarla. Cabe recalcar que los *listeners* de los sensores están programados para que vayan captando valores con más frecuencia de normal, esto se hace mediante las líneas de código siguientes:

```
1 sensorManager.registerListener(this, accelerometerSensor, SensorManager.  
    SENSOR_DELAY_GAME);  
2 sensorManager.registerListener(this, gyroscopeSensor, SensorManager.  
    SENSOR_DELAY_GAME);
```

Cuando ya se han guardado todos los valores necesarios en los *Arraylist* del acelerómetro y giroscopio, estos valores son enviados a través del canal creado mediante la función *write()* del objeto de la clase *MyBluetoothService*.

Interfaz

La interfaz del reloj inteligente es muy básica, al iniciar la aplicación aparece una actividad con todos los dispositivos bluetooth que ofrecen conexión. Al seleccionar uno, la actividad cambia a otra lista para elegir el movimiento a realizar. Cuando eliges el movimiento aparece la actividad inicial 3.27 la cual posee dos botones azul y verde. El botón azul que aparece da pie a empezar el movimiento con una cuenta atrás, y del mismo modo, cuando se quiera parar de ejecutar repeticiones se seleccionará este y se volverá al estado de pausa. Por otro lado el botón verde es el que se encarga de enviar todos los datos recabados hasta ese momento.



Figura 3.27: Actividad principal de la aplicación en el reloj inteligente. Fuente propia.

3.4. Algoritmo

El primer paso antes de definir el algoritmo es delimitar lo que se entiende como una repetición correcta o incorrecta. Para ello se va a tomar como referencia el libro *Bodybuilding Anatomy* [21] y sobre todo el contenido audiovisual proporcionado por la página web

oficial de *PowerExplosive* [15], página oficial de un centro deportivo con personal cualificado liderado por David Marchante, profesional licenciado en Ciencias de la Actividad Física y del Deporte en la Universidad de Madrid. En este proyecto se van a clasificar las repeticiones según el nivel de flexión que va a presentar el codo durante el movimiento.

- Se va a clasificar una repetición correcta cuando el movimiento se produzca con una ligera flexión de codo. Figura 3.30
- Se va a clasificar una repetición incorrecta por dos motivos:
 - Al realizar el movimiento el usuario flexiona demasiado el codo llegando a los 90° , ya que esto implica que parte del trabajo recaiga sobre los rotadores externos del hombro. Figura 3.28
 - Al ejecutar la repetición el usuario no realiza ninguna flexión de codo lo que sería potencialmente lesivo. Figura 3.29



Figura 3.28: Mala ejecución de una elevación lateral por exceso de flexión de codo. Fuente propia.



Figura 3.29: Mala ejecución de una elevación lateral por ausencia de flexión de codo. Fuente propia.



Figura 3.30: Correcta ejecución de una elevación lateral por ligera flexión de codo. Fuente propia.

Con los datos ya organizados en la base de datos y para una mejor visualización de los mismos se extraen los valores de los sensores en formato CSV. Para ello se usa una función llamada *exportar* que crea un archivo en el almacenamiento interno del dispositivo móvil y carga los datos guardados en las tablas de acelerómetro y giroscopio.

Los datos exportados se pueden visualizar en un gráfico tal y como se ve en la figura 3.31 para el acelerómetro y en la figura 3.32 para el giroscopio, donde se pueden apreciar los valores tomados en los 3 ejes distintos.

Se puede observar tanto en la figura 3.31 como en la figura 3.32 una variación en los valores durante la ejecución de cada repetición. Esta se estabiliza en el momento en el que la mancuerna permanece en reposo, es decir, entre el fin de una repetición y el inicio de otra, por lo que la idea principal para empezar a tratar estos datos es localizar cada repetición identificando el inicio y fin de cada una. Para ello se van a usar los datos del acelerómetro.

En cambio los datos del giroscopio se van utilizar para delimitar la correcta ejecución o no del ejercicio en cuestión e informar sobre la causa.

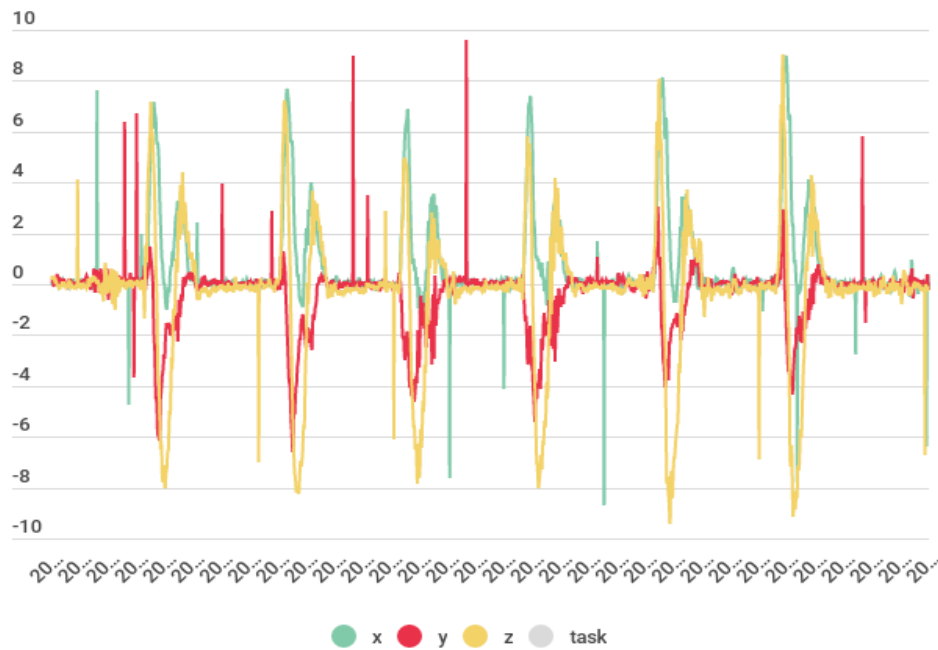


Figura 3.31: Datos del acelerómetro al realizar 6 repeticiones de elevaciones laterales. Fuente propia.

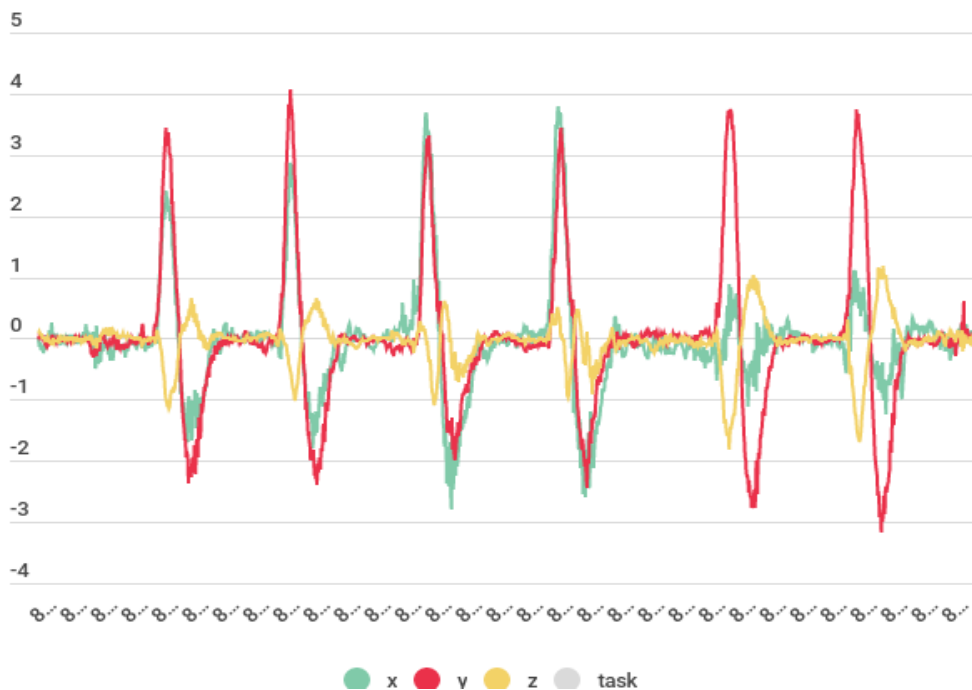


Figura 3.32: Datos del giroscopio al realizar 6 repeticiones de elevaciones laterales. Fuente propia.

Observando la figura 3.31 vemos que cuando se produce una repetición el valor del eje X aumenta, esto indica el inicio de una repetición y por tanto es el eje X el que se va a utilizar para delimitar el inicio de la repetición.

Para poder identificar correctamente el inicio de cada repetición se han obtenido los índices de todos los máximos relativos que hay en el eje X. Esto se ha realizado con una función llamada *maxRelativos* y cuya finalidad es recorrer todos los valores del eje y si encuentra un máximo relativo lo guarda a un *Arraylist*. Una vez que se han obtenido los máximos se van recorriendo uno a uno, restando su valor con el valor del máximo anterior.

$$ratio_n = MaxX_n - MaxX_{n-1}$$

Así se obtiene el ratio o la diferencia que hay entre los valores. Se ha establecido un valor mínimo (el valor 2) para el eje X, el cual si es sobrepasado se considera el inicio de una repetición. Todo este proceso y el siguiente van a ser desarrollados en una función denominada *load_acel.reps*.

Cuando se ha iniciado una repetición, no se puede iniciar otra distinta hasta que haya acabado esta y para saber si una repetición ha llegado a su fin, como vemos en la figura 3.31, tiene que haber una variación mínima en los valores de los sensores durante un corto periodo de tiempo (en el eje X), es decir, que para que se considere fin de repetición, los valores deben oscilar alrededor del valor 0. En este caso se ha indicado que si los valores de los máximos relativos no superan el valor de 0.4 y no son menores que -0.4 (a lo largo de 3 máximos relativos), se considera como fin de repetición. A partir de ahí comienza la búsqueda de un nuevo inicio, por lo que para que una repetición cuente debe haber un tiempo de reposo entre cada repetición. Una vez identificado el inicio y el final de cada repetición ya se tienen las repeticiones localizadas y estructuradas, tal y como ilustran las figuras 3.33 y 3.34, en las cuales cada escalón azul una repetición diferente.

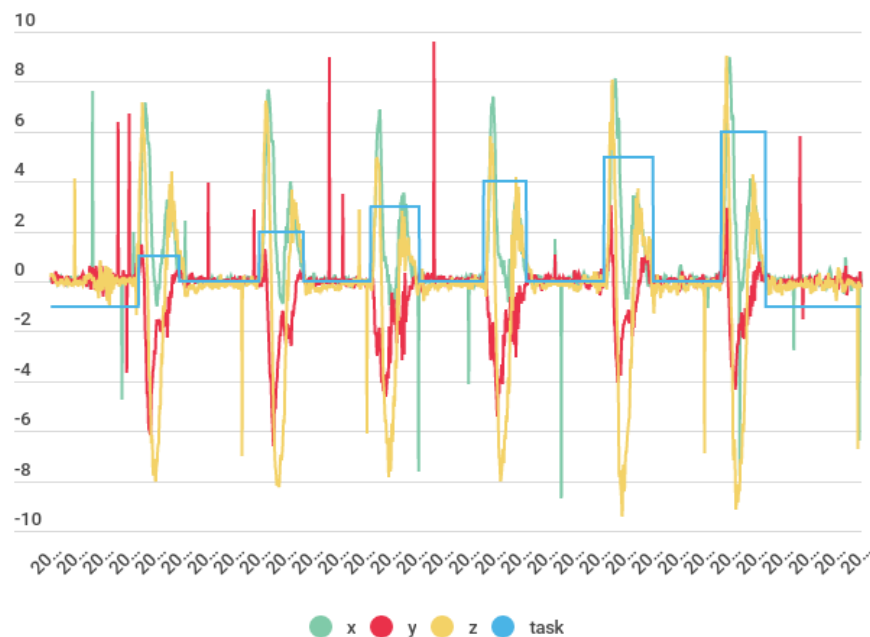


Figura 3.33: Datos del acelerómetro al realizar 6 repeticiones de elevaciones laterales con las repeticiones localizadas. Fuente propia.



Figura 3.34: Datos del giroscopio al realizar 6 repeticiones de elevaciones laterales con las repeticiones localizadas. Fuente propia.

Delimitada cada repetición solo resta obtener los datos estadísticos antes mencionados: media, varianza, skewness, máximo absoluto y mínimo absoluto de cada eje. Estos datos van a variar dependiendo de la intensidad con la que se realice el ejercicio y en nuestro caso no es lo que buscamos, por lo que necesitamos saber la relación que hay entre los valores de los 3 ejes en un mismo punto, para ello se normaliza cada valor entre 0 y 1 para así ver la relación que mantienen. Estos datos van a ser usados en la función *decisionTree*.

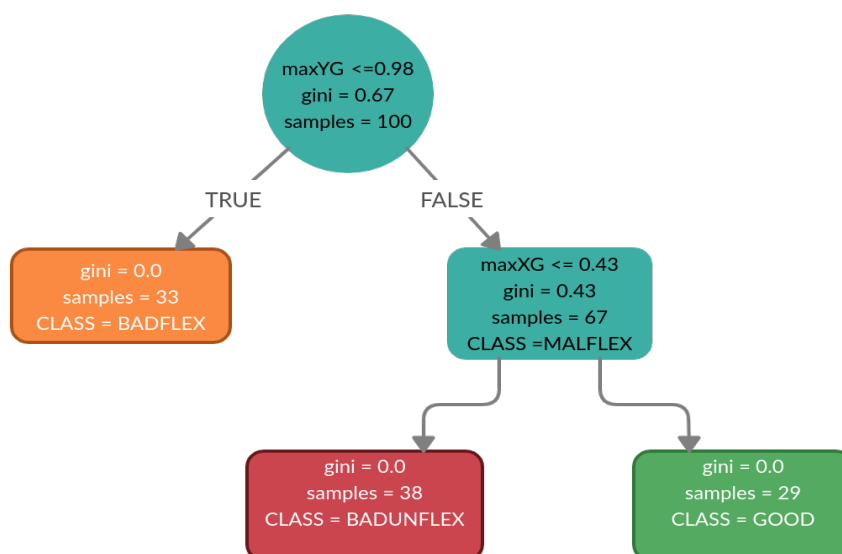


Figura 3.35: Árbol de decisión creado a partir de los datos de 10 usuarios para elevaciones laterales. Fuente propia.

```
1 public void decisionTree(List<Repetition> reps, String mov) {  
2     switch (mov) {  
3         case "Lateral Elevations":  
4             for (int i = 0; i < reps.size(); i++) {  
5  
6                 if (reps.get(i).getMaxYG() <= 0.98) {  
7                     //BAD YOU SHOULD FLEX LESS YOU ELBOW  
8                     reps.get(i).setRepOK(1);  
9                 } else if (reps.get(i).getMaxXG() <= 0.43) {  
10                    reps.get(i).setRepOK(2);  
11                } else {  
12                    reps.get(i).setRepOK(0);  
13                }  
14            }  
15        }  
16        break;  
17    }  
18 }
```

Esta función se encarga de clasificar las repeticiones en 3 tipos: correctas (figura 3.30), incorrectas por doblar demasiado el codo (figura 3.28) y incorrectas por no doblar suficiente el codo (figura 3.29). Esto se va a poder clasificar de esta manera ya que si observamos la figura 3.32 podemos diferenciar 3 patrones, las dos primeras repeticiones están realizadas de manera correcta, las dos siguientes de manera incorrecta doblando el codo a 90° y las dos últimas estirando el brazo completamente, por lo que se pueden apreciar las distintas ejecuciones del movimiento.

Para obtener esta función se ha recolectado información de 10 usuarios distintos, realizando elevaciones laterales de forma supervisada y clasificadas una a una de forma visual. Con estas repeticiones se han creado dos *datasets*, uno para entrenar un árbol de decisión y otro para probar dicho árbol, todo esto de manera *offline*, dando como resultado el árbol que vemos en la figura 3.35.

La aplicación además puede calcular el tiempo de reposo de la mancuerna entre series. Para ello lo que se ha hecho es establecer las etiquetas de los sensores a 0 en el tiempo que pasa entre el fin de una repetición y el momento en el que se inicia otra, para ello se ha hecho uso de la función *calculateRestTime* que se encarga de calcular el tiempo total que hay entre todas las instancias que posee la etiqueta 0 en nano-segundos, que posteriormente se convertirán en minutos, segundos y mili-segundos.

Capítulo 4

Evaluación

4.1. Resultados

4.1.1. Estudio

Para este estudio se han realizado pruebas sobre 10 usuarios distintos (hombres y mujeres) con edades comprendidas entre 21-58, y con un nivel de experiencia en el entrenamiento variado (principiantes y avanzados). Estas pruebas consistían en la realización de repeticiones del ejercicio elevación lateral. Para la ejecución de estas se ha hecho uso de una mancuerna de 6 kg (esta mancuerna ha sido la misma para todos los usuarios), estas se han realizado en un ámbito doméstico, ya que con la situación del COVID-19 los establecimientos deportivos permanecen cerrados y siempre con el brazo derecho. Para obtener un dataset equilibrado en sus clases se han realizado aproximadamente las mismas repeticiones para cada una de las tres clases de clasificación, cada usuario ha realizado 36 repeticiones totales (12 de cada clase) para la creación del dataset de entrenamiento y al rededor de 10 repeticiones para crear el dataset de test. En la tabla 4.1 podemos ver con más precisión todos los datos recabados durante el estudio.

Nº	Sexo	Edad	Nivel	Número de repeticiones			TOTAL
				GOOD	BADFLEX	BADUNFLEX	
1	V	21	Intermedio	15	16	16	47
2	V	26	Novato	16	16	15	47
3	V	30	Intermedio	14	17	14	45
4	F	24	Novato	15	15	15	45
5	V	41	Novato	15	15	14	44
6	F	58	Avanzado	15	16	16	47
7	V	23	Avanzado	17	17	14	48
8	V	31	Novato	16	16	15	47
9	V	28	Avanzado	15	15	15	45
10	V	28	Intermedio	15	15	15	45
TOTAL				153	158	149	460

Tabla 4.1: Tabla con todas las repeticiones realizadas por cada usuario en el estudio. Fuente propia.

4.1.2. Matriz de confusión

Los resultados obtenidos en el algoritmo muestran que clasifica las repeticiones en: buenas [3.30], malas por no flexionar el codo [3.29] y malas por flexionarlo demasiado [3.28] con una exactitud del 98 %. Este porcentaje ha sido obtenido al someter un *dataset* (de 100 repeticiones realizadas por distintas personas y con diferente forma) al algoritmo.

A raíz de los datos obtenidos en el algoritmo se ha creado una matriz de confusión, en la que aparecen las repeticiones clasificadas correctamente e incorrectamente [4.1.2]

		Predicción		
Observación		GOOD	BADFLEX	BADUNFLEX
	GOOD	31	2	0
	BADFLEX	0	38	0
	BADUNFLEX	0	0	29

Tabla 4.2: Matriz de confusión del algoritmo. Fuente propia.

4.1.3. Encuesta post-uso

Para conocer la opinión de la gente sobre la aplicación se ha realizado un cuestionario post-uso a través de la herramienta que proporciona google para realizar formularios. 10 participantes han realizado repeticiones utilizando la aplicación. Para medir el nivel de funcionalidad del sistema se va a usar la escala de usabilidad [19] del sistema, esto consiste en 10 preguntas que van a tener 5 posibles elecciones las cuales irán desde muy de acuerdo hasta muy en desacuerdo.

Las cuestiones son:

- Creo que me gustaría utilizar este sistema con frecuencia.
- Encontré el sistema innecesariamente complejo.
- Pensé que el sistema era fácil de usar.
- Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema.
- Encontré que las diversas funciones de este sistema estaban bien integradas.
- Pensé que había demasiada inconsistencia en este sistema.
- Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente.
- Encontré el sistema muy complicado de usar.
- Me sentí muy seguro al usar el sistema.
- Necesitaba aprender muchas cosas antes de poder ponerme en marcha con este sistema.

Cada respuesta va ser transformada a un número comprendido entre el 0 y el 4 y sumado al cómputo general de cada usuario, y posteriormente multiplicado por 2.5 para mantener la puntuación total sobre 100. Un puntaje del SUS por encima de 68 se consideraría por

encima del promedio y cualquier valor por debajo de 68 está por debajo del promedio, sin embargo, la mejor manera de interpretar sus resultados consiste en "normalizar" los puntajes para producir una clasificación percentil.

Los resultados obtenidos son los siguientes:

Creo que me gustaría utilizar este sistema con frecuencia



10 respuestas

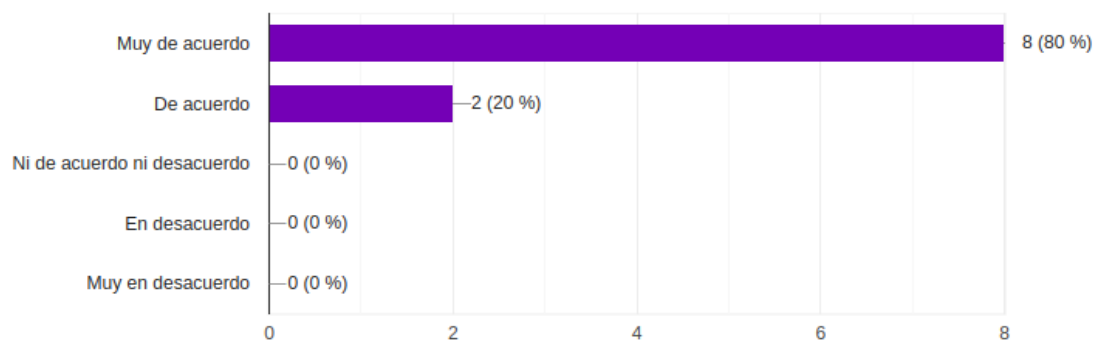


Figura 4.1: Primera pregunta de la encuesta post-uso. Fuente propia.

Encontré el sistema innecesariamente complejo



10 respuestas

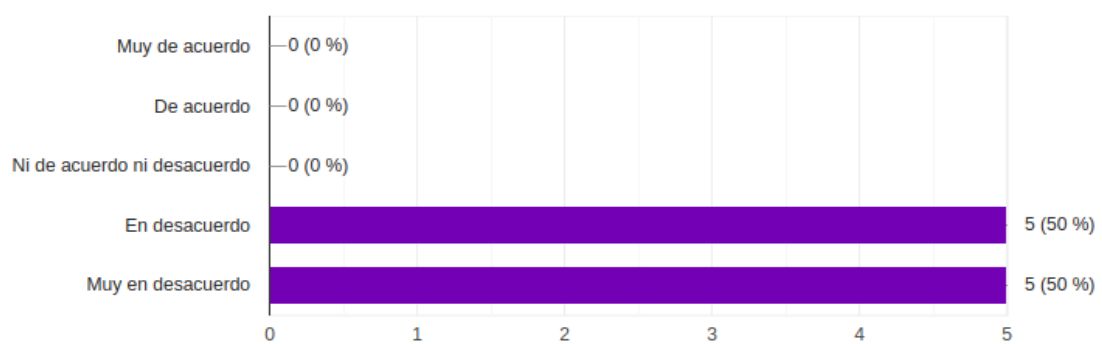


Figura 4.2: Segunda pregunta de la encuesta post-uso. Fuente propia.

Pensé que el sistema era fácil de usar

10 respuestas

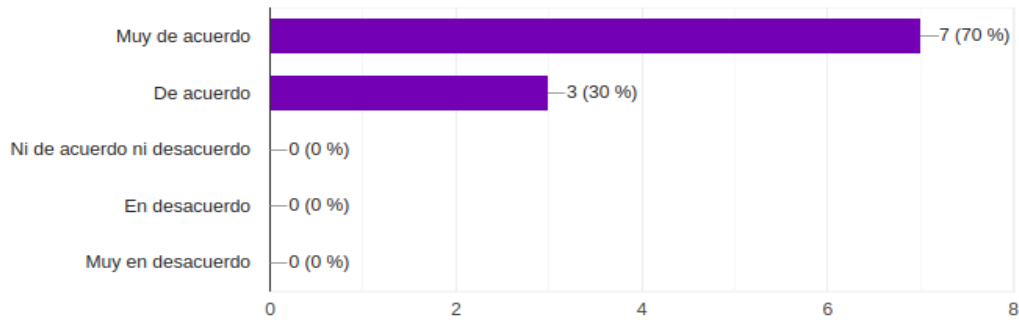


Figura 4.3: Tercera pregunta de la encuesta post-uso. Fuente propia.

Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema

10 respuestas

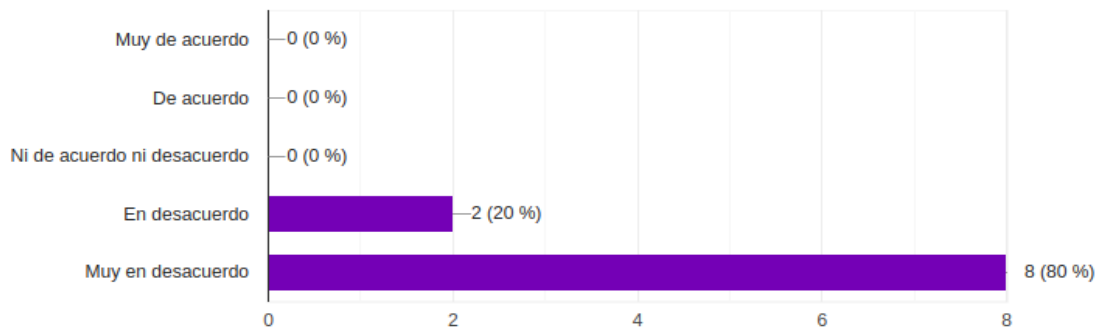


Figura 4.4: Cuarta pregunta de la encuesta post-uso. Fuente propia.

Encontré que las diversas funciones de este sistema estaban bien integradas

10 respuestas

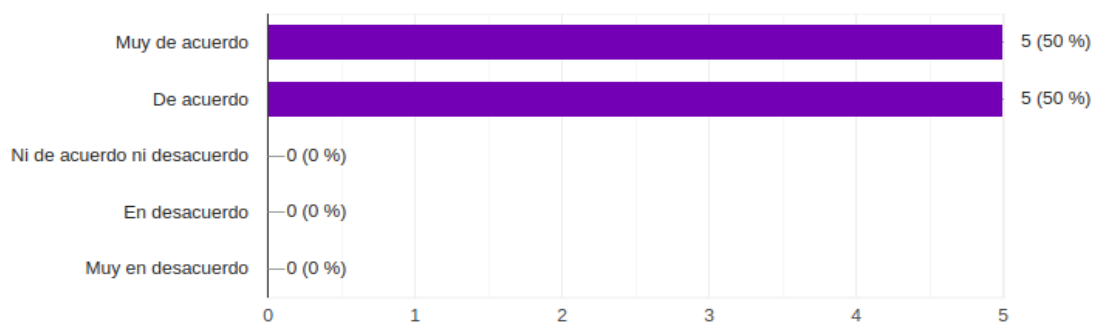


Figura 4.5: Quinta pregunta de la encuesta post-uso. Fuente propia.

Pensé que había demasiada inconsistencia en este sistema



10 respuestas

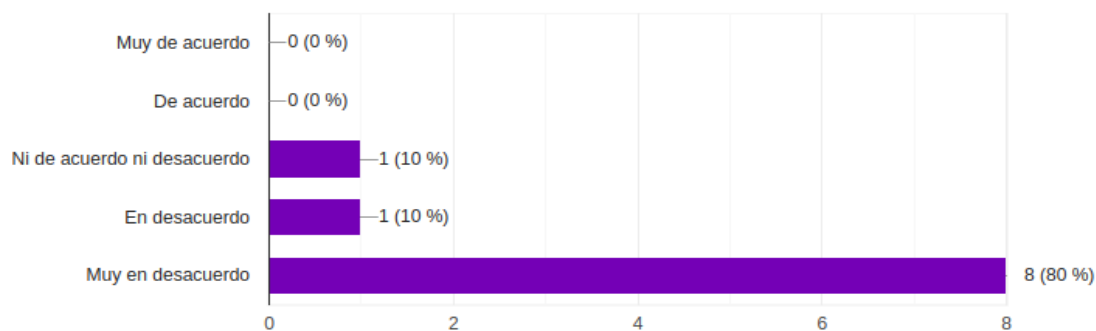


Figura 4.6: Sexta pregunta de la encuesta post-uso. Fuente propia.

Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente



10 respuestas

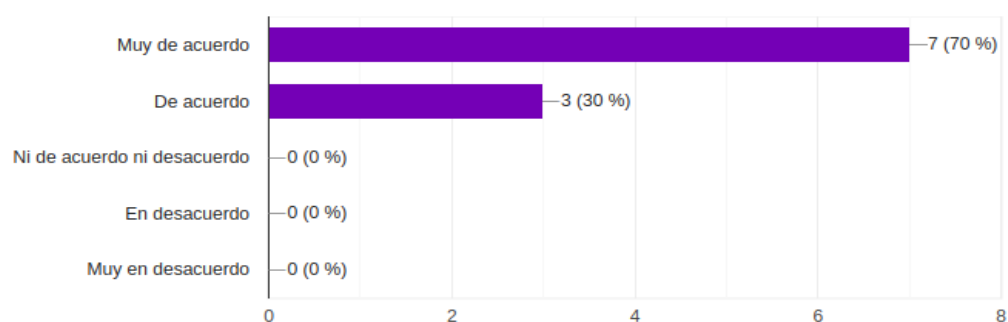


Figura 4.7: Séptima pregunta de la encuesta post-uso. Fuente propia.

Encontré el sistema muy complicado de usar



10 respuestas

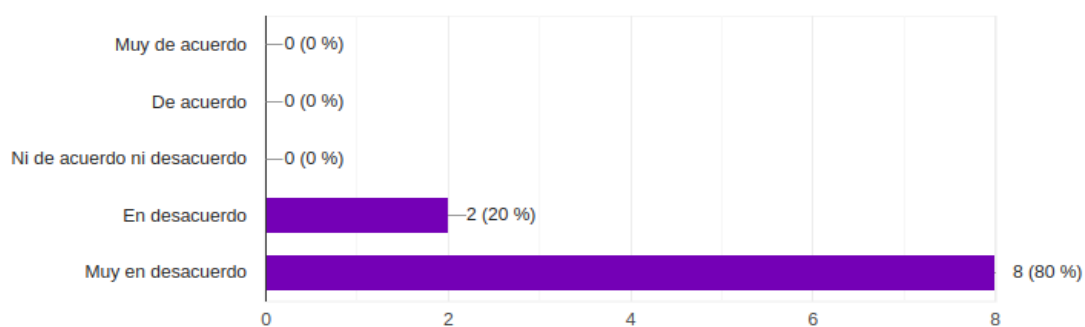


Figura 4.8: Octava pregunta de la encuesta post-uso. Fuente propia.

Me sentí muy seguro al usar el sistema

10 respuestas

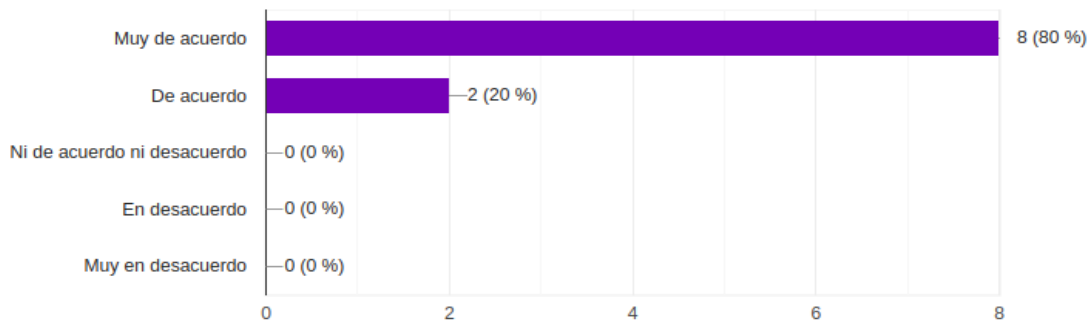


Figura 4.9: Novena pregunta de la encuesta post-uso. Fuente propia.

Necesitaba aprender muchas cosas antes de poder ponerme en marcha con este sistema

10 respuestas

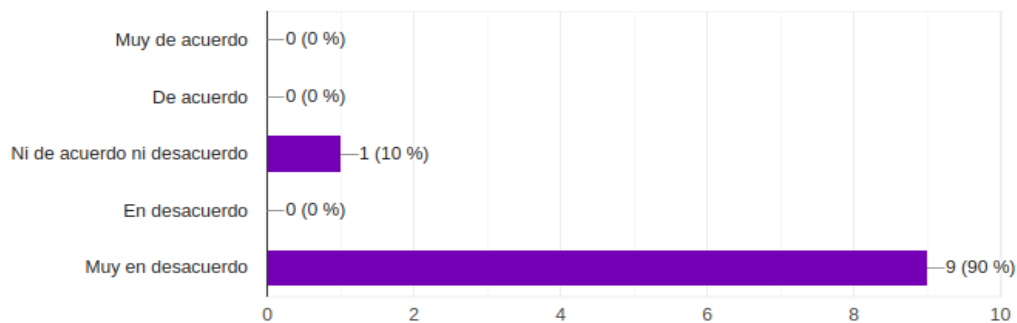


Figura 4.10: Décima pregunta de la encuesta post-uso. Fuente propia.

Los percentiles totales obtenidos son:

Usuario	Percentil total
1	95
2	97,5
3	85
4	75
5	97,5
6	92,5
7	97,5
8	92,5
9	100
10	95

Tabla 4.3: Tabla con los percentiles de cada usuario. Fuente propia.

Esto da como resultado un percentil global de 92,75.

4.2. Discusión

Ante los valores obtenidos que podemos visualizar en la matriz de confusión [4.1.2](#) se llega a la conclusión de que el algoritmo cumple bien su función. Solo dos de las 100 repeticiones han sido clasificadas erróneamente como mala cuando era una buena repetición. Esto nos indica que en una serie de repeticiones casi la totalidad de ellas serían clasificadas correctamente.

Como vemos en la tabla [4.1.3](#) el global de las encuestas es de 92.75 lo que indica que las respuestas a las preguntas han sido votadas positivamente, ya que ha superado con creces el 68. De esto se puede deducir que es una aplicación intuitiva, fácil de usar y práctica.

Este proyecto posee varias limitaciones claras que en un futuro se podrían solventar, entre ellas:

- La aplicación solo reconoce las repeticiones realizadas si el reloj está localizado en la mano derecha.
- No se puede saber si una repetición está realizada de manera correcta hasta el final de la serie, por lo que no se puede corregir la postura en tiempo real.
- Para que se identifiquen las repeticiones es necesario que entre cada repetición haya un tiempo de reposo en el que se mueva lo menos posible la mancuerna, por lo que si se ejecuta una serie realizando repeticiones consecutivas sin descansar la mancuerna, la aplicación identificará todas las repeticiones como una sola.
- La causa de la limitación más importante de este proyecto es que solo se pueden tomar los datos de un punto en concreto, en este caso la muñeca derecha, esto vale para delimitar la correcta ejecución de algunos ejercicios específicos, ya que ejercicios compuestos como, por ejemplo, en una sentadilla en la que se tienen que tener en cuenta la posición de las rodillas, la posición de la cadera y la posición de los tobillos, no es posible delimitar si se ha realizado correctamente solo con un reloj inteligente.

Capítulo 5

Conclusiones y Trabajos Futuros

5.1. Conclusiones

Se ha conseguido el objetivo principal de este proyecto, el cual era desarrollar un sistema inteligente capaz de detectar las repeticiones realizadas de elevaciones laterales en un entrenamiento y cuantificar la realización de las mismas para posteriormente poder clasificarlas dependiendo de la técnica realizada, en buenas y malas repeticiones así como indicar el posible motivo de aquellas que se realizan incorrectamente.

Además se ha desarrollado una arquitectura software capaz de comunicar el dispositivo móvil con el reloj inteligente, para así poder enviar datos del reloj al dispositivo móvil donde son procesados y visualizados.

También se han obtenido datos de diversos individuos con el objetivo de crear un nuevo algoritmo que sea capaz de llevar a cabo la clasificación de estas repeticiones en buenas o malas (justificando con el motivo) y se ha podido cuantificar el nivel de acierto de dicho algoritmo.

Por último se han creado dos interfaces gráficas, una para cada aplicación, las cuales además de satisfacer el objetivo principal de este proyecto ofrecen otro tipo de servicios, como selección de perfil, visualización de repeticiones realizadas en distintos días y personalización de perfil entre otros añadidos. Además se ha realizado una encuesta post-uso a cada usuario para así ver el grado de usabilidad que ha observado en la aplicación.

Por tanto todos los objetivos establecidos previamente se han alcanzado.

5.2. Trabajos Futuros

Este proyecto puede expandirse en numerosos aspectos:

- El proyecto actual solo contempla un movimiento, las elevaciones laterales. Como trabajo futuro se podría hacer un estudio de los numerosos ejercicios que se realizan en el gimnasio y ampliar el número de ejercicios disponibles.
- Esta aplicación tiene la peculiaridad de que clasifica las repeticiones, una vez rea-

lizada la serie completa. Sería interesante poder modelar el proyecto para que la respuesta de la aplicación fuera en tiempo real y así se pudiera corregir la forma del ejercicio mientras se va realizando.

- Como trabajo futuro sería interesante además expandir la aplicación para que no se tuviera que seleccionar el movimiento a realizar, sino que se clasificara el movimiento realizado tal y como se realiza en el proyecto titulado *FitCoach* [22] mencionado anteriormente en el estado del arte y posteriormente clasificar el movimiento según su forma.
- Para escalar la aplicación se podrían cambiar los usuarios por cuentas de usuario con contraseña, creando así una aplicación web en la que se almacenará toda la información de dichas cuentas y se podrá tener la información disponible desde cualquier dispositivo con acceso a Internet.

Bibliografía

- [1] Amazfit gtr. <https://www.pccomponentes.com/amazfit-gtr-reloj-smartwatch-42mm-coral-red>.
- [2] Amazfit gtr. <https://www.pccomponentes.com/amazfit-gtr-reloj-smartwatch-42mm-coral-red>.
- [3] Android bluetooth. <https://developer.android.com/guide/topics/connectivity/bluetooth?hl=es-419>.
- [4] Android sensors. <https://developer.android.com/guide/topics/sensors?hl=es-419>.
- [5] Appel watch. <https://www.apple.com/es/shop/buy-watch/apple-watch>.
- [6] Desarrolladores en españa. <https://www.campusmvp.es/recursos/post/el-mercado-laboral-para-desarrolladores-en-espana-y-europa-ano-2020.aspx>.
- [7] Focusmotion. <https://focusmotion.io/>.
- [8] Galaxy watch active2. <https://www.samsung.com/es/wearables/galaxy-watch-active2-sm-r830nz/SM-R830NZKAPHE/>.
- [9] Google scholar. <https://scholar.google.es/schhp?hl=es>.
- [10] Google trend. <https://trends.google.es/trends/?geo=ES>.
- [11] Huawei watch 2. <https://www.xataka.com/relojes-inteligentes/huawei-watch-2-analisis-mas-autonomo-y-deportivo-que-nunca-pero-no-necesariamente-mejor>.
- [12] Moscow. <https://unpocodejava.com/2019/01/14/que-es-el-metodo-moscow-para-toma-de-requerimientos/>.
- [13] Moto 360. <https://www.tusmartwatch.es/moto-360-2/>.
- [14] Motorola 360 2nd gen. <https://www.smartwatchspecifications.com/Products/motorola-360-2nd-gen/>.
- [15] Powerexplosive. <https://powerexplosive.com/5-trucos-para-hombros-gigantes-elevaciones-laterales/>.
- [16] Random forest. <https://bookdown.org/content/2031/ensambladores-random-forest-parte-i.html>.
- [17] Room. <https://developer.android.com/training/data-storage/room>.
- [18] Versa 2. <https://www.fitbit.com/es/versa>.

- [19] [www.usability.gov. https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html](https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html).
- [20] Carlos Balsalobre-Fernández, David Marchante, Eneko Baz-Valle, Iván Alonso-Molero, Sergio L. Jiménez, and Mario Muñoz-López. Analysis of wearable and smartphone-based technologies for the measurement of barbell velocity in different resistance training exercises. *Frontiers in Physiology*, 8:649, 2017.
- [21] Nick Evans. *Bodybuilding anatomy*. Human Kinetics, 2015.
- [22] X. Guo, J. Liu, and Y. Chen. Fitcoach: Virtual fitness coach empowered by wearable mobile devices. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pages 1–9, May 2017.
- [23] Silvio Lorenzetti and Dominik Huber. Tracking of strength training: Validation of a motion-recognition algorithm & a pilot towards 1rm, muscle loading and fatigue index using a smartwatch app. *ISBS Proceedings Archive*, 36(1):886, 2018.
- [24] Slobodan Milanko and Shubham Jain. Liftright: Quantifying strength training performance using a wearable sensor. 2019.
- [25] B. J. Mortazavi, M. Pourhomayoun, G. Alsheikh, N. Alshurafa, S. I. Lee, and M. Sarrafzadeh. Determining the single best axis for exercise repetition recognition and counting on smartwatches. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks*, pages 33–38, June 2014.
- [26] Juan M Santos-Gago, Mateo Ramos-Merino, Sonia Vallarades-Rodriguez, Luis M Álvarez-Sabucedo, Manuel J Fernández-Iglesias, and Jose L García-Soidán. Innovative use of wrist-worn wearable devices in the sports domain: A systematic review. *Electronics*, 8(11):1257, 2019.

