



TRABAJO FIN DE GRADO
GRADO EN INGENIERÍA INFORMÁTICA

MoniSport

Aplicación móvil de soporte al análisis y monitorización de la actividad deportiva mediante vídeo y wearables

Autor

José M^a Ruano Fernández

Director

Manuel Noguera García



Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación

—
Granada, Noviembre de 2017



ugr

Universidad
de Granada



MoniSport

Aplicación móvil de soporte al análisis y monitorización de la actividad deportiva mediante vídeo y wearables

Autor

José M^a Ruano Fernández

Director

Manuel Noguera García

MoniSport: Aplicación móvil de soporte al análisis y monitorización de la actividad deportiva mediante vídeo y wearables

José M^a Ruano Fernández

Palabras clave: Aplicaciones móviles, Android, Control del rendimiento deportivo, Conexión de sensores Bluetooth, Wearable.

Resumen

MoniSport es una aplicación móvil implementada en Android cuyo objetivo principal es registrar, mediante sensores externos, variables de la práctica deportiva y grabar al mismo tiempo en vídeo su ejecución para su posterior análisis. Con este doble fin, durante la realización de un ejercicio se monitorizan datos como el pulso cardíaco o la temperatura, al tiempo que guarda en vídeo la sesión deportiva. La aplicación permite de este modo analizar los entrenamientos, sincronizando el vídeo con los datos registrados, lo que facilita la labor de estudio de especialistas del deporte y de la salud.

Para almacenar los datos, la aplicación permite la conexión, a través de Bluetooth, a sensores avanzados, principalmente de tipo *wearable*. Además, aprovecha las ventajas de los sistemas móviles, por lo que no es necesario un sistema de grabación profesional y basta un dispositivo móvil de potencia media para la captura del vídeo, algo que normalmente requiere una instalación costosa y que suele ser fija una vez instalada y configurada, es decir, que no puede trasladarse o transportarse fácilmente. Esto presenta el inconveniente de requerir que la monitorización y grabación de un ejercicio haya de realizarse en un entorno o laboratorio específico, impidiendo cambiar el lugar de monitorización según pueda interesar cada vez. Esto, a su vez, dificulta la grabación y análisis de la actividad física realizada en aquellos lugares donde puede llevarse a cabo más naturalmente (pensemos, por ejemplo, en una carrera al aire libre o un parque). Por el contrario, MoniSport, es totalmente portable, requiriendo únicamente un dispositivo móvil con cámara de fotos que pueda grabar vídeo.

El diseño simple de la interfaz facilita y simplifica el uso de la aplicación, lo que permite una mayor rapidez y una mejor usabilidad.

MoniSport: Mobile App to Monitor and Analyze Sports Activity through video and wearables

José M^a Ruano Fernández

Keywords: Mobile Application, Android, Sports Performance Control, Bluetooth Sensors Connection, Wearable.

Abstract

MoniSport is a simple mobile application implemented on Android whose main objective is recording, through external sensors, variables of the sports practice, and recording at the same time in video its performance to analyze it later. Taking on this double purpose, while doing an exercise, specific data is monitored, for example, heart rate or temperature, while the application records in video the corresponding sports session. In this way, the application allows to analyze training sessions, synchronizing video with all the recorded data, making easier the study work to sports and health specialists. Finally, this data could be used by the different specialists to improve performance in the corresponding sport and to avoid health problems.

For data recording, the application allows the connection, through Bluetooth, to advanced sensors, mainly of wearable type. In addition, it benefits from the great advantages that the mobile systems provide, so you don't need a very professional recording system and mobile device with a medium power is enough to capture video. This usually requires an expensive facility and in general is fixed when installed and configured, i.e., it cannot be easily transferred or carried. The extra disadvantage of this is that the monitoring and recording of an exercise should be done in a specific environment or laboratory, preventing from changing the site where the monitoring is done according to specific interests each time. In turn, this makes more difficult to record and analyze physical activity done in places where it can be done in a more natural way (for example, a race outdoor or in a park). By contrast, MoniSport is completely portable and it only requires a mobile device with a built-in camera that can record video.

The user-friendly interface design makes it easy and straightforward to use the application, allowing it to be faster and more usable. Tracking is simple due to graph drawn against rate, breath, acceleration, etc. In addition, you do not need a major learning, making the application suitable for sports and health specialists that do not have in-depth knowledge in the technological area.

In this document, you can find detailed information about functions that the application does and its specific features, along with user manual. It also contains the context analysis that drive to developing this application.

Yo, **José M^a Ruano Fernández**, alumno de la titulación GRADO EN INGENIERÍA INFORMÁTICA de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 76442816Q, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: José M^a Ruano Fernández

Granada a 13 de Noviembre de 2017.

D. **Manuel Noguera García**, Profesor la Universidad del Área Lenguajes y Sistemas Informáticos del Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Granada.

Informa:

Que el presente trabajo, titulado ***MoniSport: Plataforma para la monitorización de la actividad deportiva***, ha sido realizado bajo su supervisión por José M^a Ruano Fernández, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 13 de noviembre de 2017.

El director:

Manuel Noguera García

Agradecimientos

En primer lugar quiero agradecer a mi familia y amigos por la paciencia que tienen conmigo y por darme su cariño.

A Manuel, tutor de este proyecto, por el apoyo y comprensión durante el desarrollo. Gracias.

Al mejor periodista y comunicador audiovisual en la faz de la Tierra, Javi, sin ti sería imposible.

A Salva, sus conocimientos y amor por el deporte no tienen rival, gracias por la ayuda.

Por último quiero agradecer a Angie por estar siempre a mi lado, apoyándome y ayudándome incluso sin yo saberlo.

Índice

Sección I: Descripción del proyecto.....	20
1. Contexto	21
1.1 Introducción y motivaciones.....	21
1.2 Objetivos	22
2. Antecedentes	25
2.1 Sistemas móviles.....	25
2.2 Bluetooth Low Energy	27
2.3 Sensores	27
2.3.1 Zephyr BioHarness 3	28
2.3.2 SensorTag CC2541.....	28
2.3.3 Otros sensores.....	29
2.4 Aplicaciones móviles.....	30
2.4.1 AirCasting	30
2.4.2 Internet of Things (IoT) Dashboard/Gateway – IoTTool	30
2.4.3 Simplelink SensorTag	31
2.4.4 BLE SensorTag	32
2.5 Conclusiones	33
3. Proyecto desarrollado	34
3.1 Descripción.....	34
3.2 Funcionalidades	34
3.3 Interfaz completa.....	40
Sección II: Desarrollo del proyecto.....	43
4. Planificación	44
5. Análisis inicial	45
5.1 Planificación	45
5.2 Estudio de aplicaciones similares y ámbito tecnológico:	45
5.3 Aprendizaje de Android y el entorno Android Studio:	45
5.4 Creación de documentación primaria:	45
5.5 Diseño del logo de la aplicación:.....	45
5.6 Análisis de las especificaciones del proyecto:	45
5.6.1 Escenarios.....	45
5.6.2 Actores	46
5.6.3 Requisitos	46

6.	Diseño.....	51
6.1	Diagrama de componentes principales	51
6.2	Arquitectura de la aplicación	51
6.2.1	Componentes de la aplicación	51
6.2.2	Interacción de los componentes	52
7.	Implementación	54
7.1	Sistema Bluetooth.....	54
7.2	Grabación y visualización a tiempo real	56
7.3	Recuperación y visualización	57
7.4	GUI	58
8.	Pruebas.....	59
9.	Herramientas utilizadas	60
	Sección III: Conclusiones y trabajo futuro	61
10.	Conclusiones.....	62
11.	Trabajo futuro.....	63
	Sección IV: Manual del usuario	64
	Bibliografía.....	71

Índice de figuras

Figura 1: Prueba de esfuerzo completa.....	21
Figura 2: Ventas mundiales de <i>Smartphones</i> a usuarios finales por sistema operativo en el primer cuarto de 2017 y el primer cuarto de 2016 (Miles de unidades)	25
Figura 3: Logo de Android	25
Figura 4: Tabla con las distribuciones de Android y su cuota de mercado	26
Figura 5: Zephyr BioHarness 3	28
Figura 6: Aplicación de escritorio de Zephyr BioHarness 3	28
Figura 7: SensorTag CC2541	29
Figura 8: Relojes inteligentes LG Watch Urbane	29
Figura 9: Panel principal de la aplicación AirCasting	30
Figura 10: Panel principal de la aplicación Internet of Things (IoT) Dashboard/Gateway – IoTool	31
Figura 11: Panel principal de la aplicación Simplelink SensorTag	32
Figura 12: BLE SensorTag.....	32
Figura 13: Icono de MoniSport	34
Figura 14: Formulario para guardar los datos personales del deportista	35
Figura 15: Cuadro de diálogo mostrado durante la búsqueda de dispositivos.....	35
Figura 16: Gráficas con los datos recibidos a tiempo real.....	36
Figura 17: Previsualización de la sesión deportiva en tiempo real	36
Figura 18: Directorio donde están almacenados los archivos de una sesión deportiva	37
Figura 19: Listado de sesiones almacenadas en el dispositivo.....	37
Figura 20: Reproductor de vídeo visualizando una sesión deportiva almacenada.....	38
Figura 21: Gráficas con los datos almacenados de una sesión deportiva.....	38
Figura 22: Gráficas con los datos almacenados, mostrando los valores del segundo	39
Figura 23: Botones para sincronizar o desincronizar las gráficas y el vídeo reproducidos	39
Figura 24: Botones para ocultar o mostrar la gráfica correspondiente	39
Figura 25: Pantalla de inicio de MoniSport	40
Figura 26: Pantalla de inicio con el formulario de datos personales del deportista.....	40
Figura 27: Pantalla de búsqueda y conexión de dispositivos	41
Figura 28: Pantalla de grabación de sesiones deportivas	41
Figura 29: Pantalla de selección de sesiones deportivas almacenadas para visualizar.	42
Figura 30: Pantalla de visualización de sesiones deportivas almacenadas.....	42
Figura 31: Planificación estimada del proyecto	44
Figura 32: Diagrama de componentes principales.....	51
Figura 33: Código encargado de buscar, mostrar y conectar los dispositivos Bluetooth	54
Figura 34: Código encargado de enviar los datos recibidos por el Zephyr Listener	55
Figura 35: Código encargado de enviar los datos recibidos por el BluetoothServices ..	56
Figura 36: Parte del código encargado de recibir los datos y mostrarlos	57
Figura 37: Parte del código encargado de actualizar el valor mostrado por la gráfica si está sincronizado	58
Figura 38: Directorio con los archivos que componen la GUI	58

Figura 39: Pantalla de inicio de MoniSport	65
Figura 40: Pantalla de inicio con el formulario de datos personales del deportista desplegado	65
Figura 41: Pantalla de búsqueda y conexión de dispositivos.....	66
Figura 42: Pantalla de búsqueda y conexión de dispositivos mientras realiza una búsqueda	66
Figura 43: Pantalla de grabación de sesiones deportivas	67
Figura 44: Archivo CSV con los datos de una sesión deportiva con una sola componente	68
Figura 45: Archivo CSV con los datos de una sesión deportiva con tres componentes	68
Figura 46: Pantalla con el listado de sesiones deportivas almacenadas.....	69
Figura 47: Pantalla de visualización de sesiones deportivas almacenadas.....	69

Sección I: Descripción del proyecto

1. Contexto

1.1 Introducción y motivaciones

Ante el creciente interés de la sociedad actual por el estilo de vida saludable, en el que el deporte juega un papel fundamental, se ha incrementado la demanda de profesionales de los campos de la salud y la educación física, así como, las necesidades de estos: requieren herramientas que les permitan ofrecer servicios eficaces y de calidad. Entre ellas destacan las aplicaciones informáticas a través de las cuales contrastar los datos registrados por sensores con el fin de caracterizar con precisión el movimiento de las articulaciones y sugerir, tras el análisis de los datos registrados, correcciones destinadas a lograr el máximo rendimiento en la actividad deportiva.

El desarrollo de aplicaciones de monitorización de la condición física y deportiva, profesional y de alto rendimiento, requiere de un proceso previo de observación y monitorización del movimiento. El objetivo último de este trabajo preparatorio es el de contrastar los numerosos parámetros medibles con la observación visual.

Este proceso ha de llevarse a cabo, en la mayoría de los casos, de forma manual y utilizando diversos sistemas. Esto dificulta y ralentiza los procedimientos de análisis y la validación de las herramientas de soporte. En definitiva, es necesario cotejar los datos con la observación natural a fin de asegurar que las mediciones se ajustan a lo simplemente observado y a la percepción de los propios deportistas.

Por otro lado, se da la circunstancia de que buena parte de las aplicaciones de las que se puede servir este sector sólo están disponibles en versión de escritorio. Sin embargo, en la mayoría de las situaciones no es necesario disponer de un equipo de grabación profesional, sino simplemente la posibilidad de grabar imágenes y coordinarlas con los registros de los sensores. Algo que es posible con dispositivos móviles de potencia media. La Figura 1 muestra un ejemplo de este tipo de escenarios para el caso de una prueba de esfuerzo completa.



Figura 1: Prueba de esfuerzo completa
(Fuente: www.letsport.es)

Otra problemática tiene que ver con la cantidad y variedad de sensores deportivos disponibles en el mercado, por lo general, cada uno de ellos con su propia aplicación móvil o de escritorio. Esto dificulta la recolecta de datos desde diversos sensores con el objetivo de abordar un análisis conjunto de la actividad deportiva. Además, disminuye el espacio de almacenamiento disponible en el dispositivo ante la necesidad de instalar varias aplicaciones y se fuerza a los profesionales en la materia a familiarizarse con las distintas interfaces de cada aplicación. Todo ello unido a que las son poco personalizables, por lo que los valores recabados no resultan de utilidad suficiente al usuario.

En conclusión, una plataforma de observación móvil y portables para analizar y registrar sesiones de entrenamiento, en laboratorio o exteriores, facilitaría la labor de estudio de los profesionales del deporte convirtiéndose en una herramienta útil para los mismos, sin depender de un tipo concreto de sensor o marca. A su vez, esto evita ciertos inconvenientes derivados de tener que reservar el uso de un laboratorio, como horarios de entrenamiento menos flexibles y movibles, desplazamientos hasta dicho laboratorio, simulación y recreación de entornos, etc.

Por un lado, gracias a los dispositivos *wearables* y a la plataforma móvil, no será necesario desarrollar los entrenamientos en laboratorios (ver Figura 1). Por otro, los valores recopilados y los estudios desarrollados con la plataforma se podrán consultar de manera deslocalizada gracias a las funcionalidades y servicios que ofrecen las tecnologías móviles, siendo fácilmente trasladables.

1.2 Objetivos

Se pretende crear una aplicación móvil capaz de grabar en vídeo una sesión de entrenamiento a la vez que almacena los datos de los sensores conectados vía Bluetooth para, posteriormente, reproducirlos sincronizando el vídeo y los datos recogidos, con el fin de mejorar el análisis de las sesiones deportivas, y así optimizar el rendimiento deportivo. Actualmente no hay ninguna aplicación que combine la recogida de datos de sensores de diferentes marcas y la grabación de vídeo, lo que provoca que los especialistas deban usar diferentes softwares o equipos pesados a la vez.

Se pueden resumir todos los objetivos de este proyecto en los siguientes puntos:

- **Aplicar los conocimientos adquiridos:** A lo largo del Grado en Ingeniería Informática, se han adquirido una serie de conocimientos y técnicas en las diferentes asignaturas cursadas. En este proyecto se van a intentar aplicar todos los conocimientos posibles para su desarrollo, en cualquiera de las fases que lo componen.
- **Visualizar sesiones deportivas a tiempo real:** La aplicación debe permitir visualizar una sesión deportiva mientras ocurre, cumpliendo con los siguientes

objetivos:

- **Conectar con sensores:** Conectar con diferentes tipos de dispositivos que envíen datos vía Bluetooth. Estos dispositivos podrán ser *wearables*, otras aplicaciones móviles u otros dispositivos mientras usen el estándar de Bluetooth Low Energy.
- **Visualizar a tiempo real de datos:** Los datos recibidos por los sensores se deben visualizar a modo de gráficas, mostrando el valor actual recibido.
- **Visualizar a tiempo real de vídeo:** A través de la cámara del dispositivo, se podrá visualizar la sesión de entrenamiento a tiempo real.
- **Almacenar sesiones deportivas:** Como se ha mencionado antes, la aplicación debe ser capaz de guardar las sesiones deportivas, para ello debe cumplir estos objetivos:
 - **Almacenar datos:** Almacenar los datos recibidos por los sensores mencionados anteriormente en archivos de fácil entendimiento y los datos con información personal del deportista.
 - **Grabar en vídeo:** Grabar el vídeo de la sesión de entrenamiento en un archivo que se pueda reproducir en diferentes dispositivos. Este vídeo debe tener una calidad buena para su posterior reproducción y análisis.
- **Reproducir sesiones deportivas:** La aplicación debe ser capaz de reproducir las sesiones deportivas guardadas con anterioridad. Sin embargo, no es necesario que se hayan registrado con la propia aplicación, pudiendo añadir a la aplicación por separado, los archivos de vídeo y de datos, simplemente incluyéndolos en el directorio con el formato necesario. Este objetivo se divide en:
 - **Buscar archivos:** Los archivos guardados en el correspondiente directorio del dispositivo serán encontrados y, si tienen el formato correcto, accesibles como sesiones deportivas para visualizar.
 - **Reproducir datos:** Reproducir los datos de los sensores almacenados como gráficas y mostrando el valor en un tiempo dado, lo que permite seleccionar un punto en la gráfica para saber el valor medido en ese tiempo.
 - **Reproducir vídeo:** Reproducir el vídeo almacenado para su análisis, permitiendo avanzar, retroceder y pausar el vídeo.
 - **Sincronizar datos y vídeo:** La aplicación tendrá la opción de sincronizar y desincronizar los datos y el vídeo para mejorar su análisis.

- **Usabilidad:** La aplicación debe ser sencilla de usar y de fácil aprendizaje para ayudar a la labor de los especialistas.
- **Móvil:** La aplicación debe ser móvil para no dificultar el desplazamiento y no limitar el uso a zonas concretas.
- **Eficiencia:** Las aplicaciones móviles deben consumir pocos recursos (memoria, energía, etc.). Las aplicaciones nativas mejoran la eficiencia de éstas.

2. Antecedentes

Este apartado tiene como fin analizar los diferentes sensores y aplicaciones, además del contexto tecnológico, en el mercado con propósitos similares a los de este proyecto. Este análisis sirvió durante el desarrollo del proyecto y servirá en el futuro para mejorar la aplicación y no cometer los mismos errores que las soluciones estudiadas.

2.1 Sistemas móviles

Actualmente la mayor parte del mercado (99.8%) se encuentra repartido entre dos sistemas operativos, como se puede ver en la Figura 2, por lo que solo se analizarán dichos sistemas, al no tener los demás suficiente cuota de mercado como para ser relevantes.

Operating System	1Q17 Units	1Q17 Market Share (%)	1Q16 Units	1Q16 Market Share (%)
Android	327,163.6	86.1	292,746.9	84.1
iOS	51,992.5	13.7	51,629.5	14.8
Other OS	821.2	0.2	3,847.8	1.1
Total	379,977.3	100.0	348,224.2	100.0

Source: Gartner (May 2017)

Figura 2: Ventas mundiales de *Smartphones* a usuarios finales por sistema operativo en el primer cuarto de 2017 y el primer cuarto de 2016 (Miles de unidades)
(Fuente: www.gartner.com)

- **Android**

Android (Figura 3) es un sistema operativo inicialmente dirigido a dispositivos móviles con pantalla táctil como teléfonos inteligentes, tabletas y relojes inteligentes. Actualmente también se utiliza en televisiones inteligentes y vehículos.



Figura 3: Logo de Android

Android vio la luz en el año 2007 de la mano de la Open Handset Alliance (OHA) y respaldado por el sello Google. La fundación OHA se dedica al desarrollo de estándares abiertos para dispositivos móviles.

Este sistema operativo está basado en un núcleo Linux. Y a diferencia de otros sistemas operativos del mercado, Android se desarrolla de forma abierta y permite acceder tanto al código fuente como a la lista de incidencias gracias a la cual se pueden reportar problemas encontrados y conocer aquellos que aún se encuentran sin resolver. Una de las características destacables de este

sistema operativo es que es altamente personalizable.

Las aplicaciones nativas para el sistema se desarrollan habitualmente en el lenguaje Java a través *Android Software Development Kit* (Android SDK) junto con XML como lenguaje de descripción para archivos de configuración e interfaces de usuario.

Desde su versión inicial en pruebas, lanzada en septiembre de 2008, el sistema operativo ha ido avanzando y actualizándose a través de un gran número de versiones. En la siguiente imagen (ver Figura 4) se muestra en una tabla cómo se distribuyen las versiones en el mercado.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.3%
4.2.x		17	3.3%
4.3		18	1.0%
4.4	KitKat	19	14.5%
5.0	Lollipop	21	6.7%
5.1		22	21.0%
6.0	Marshmallow	23	32.0%
7.0	Nougat	24	15.8%
7.1		25	2.0%
8.0	Oreo	26	0.2%

Figura 4: Tabla con las distribuciones de Android y su cuota de mercado
(Fuente: www.xatakandroid.com)

En la imagen se puede observar cómo la versión más utilizada, a fecha octubre de 2017, es la versión 6.0 Marshmallow.

Actualmente, Android lidera el mercado de sistemas operativos móviles con mucha diferencia en número de usuarios con respecto al resto de sus competidores (como iOS u otros sistemas), tal y como se puede observar en la Figura 2. KitKat es con diferencia la versión de Android previa a la 5 más presente actualmente en el mercado.

La solución que se propone en esta memoria ha sido desarrollada sobre la versión 4.4 KitKat debido a las limitaciones del dispositivo de desarrollo con el que se contaba, aunque este hecho tenía como aspecto positivo que también permitía validar el uso de la aplicación sobre dispositivos de potencia limitada y compatibilidad hacia atrás.

- **iOS**

iOS es el sistema operativo exclusivo de los dispositivos de Apple. Inicialmente fue desarrollado para el dispositivo iPhone, por entonces era más conocido como iPhone OS, más tarde fue incorporado en el resto de dispositivos de la marca. Fue entonces cuando cambio su nombre a iOS.

Una de sus características principales es la exclusividad, ya que no permite la instalación en hardware de terceros. Incluso a la hora de crear aplicaciones nativas, la herramienta de desarrollo, X-Code, solo se encuentra disponible para sistemas operativos de Apple. Estas aplicaciones se desarrollan en el lenguaje Objective-C.

La interacción con el sistema operativo incluye gestos como deslizamientos, toques, pellizcos, los cuales tienen diferentes definiciones dependiendo del contexto de la interfaz. Actualmente este sistema operativo se encuentra en su versión 11.

- **Conclusión**

Se puede llegar a la conclusión de que Android es una mejor opción para desarrollar la aplicación al no presentar la desventaja de imponer el requerimiento de necesitar equipos específicos para el desarrollo, además de estar respaldada por una gran comunidad de soporte y ayuda.

2.2 Bluetooth Low Energy

El protocolo Bluetooth original penalizaba mucho la autonomía de los terminales. Debido a la evolución de la tecnología surge una nueva especificación de Bluetooth que se fundamenta en la reducción del consumo. Para esto, minimiza la potencia de transmisión de la señal de radio utilizada y el radio de cobertura. Esto desemboca en un aumento de la autonomía convirtiéndolo en el estándar clave para dar soporte a los múltiples dispositivos *wearables* que se encuentran a día de hoy en el mercado. Esto también se traduce en una mejora en el desarrollo de los sensores. Actualmente esta tecnología es la base de los dispositivos deportivos como los sensores para *runners*.

Fue propuesta por Nokia en 2006 bajo el nombre de Wibree y tuvo una gran acogida. Muchas empresas apoyaron esta propuesta con la intención de que esta tecnología tuviera una rápida disponibilidad. Fue en 2010 cuando Wibree fue incluida en el estándar Bluetooth con la aceptación de la especificación para Bluetooth 4.0, pasando a nombrarse Bluetooth Low Energy (BLE).

2.3 Sensores

Hay una gran cantidad de dispositivos con sus correspondientes aplicaciones, además de aplicaciones propias. Se ha seleccionado una muestra, basada en la disponibilidad que se tiene de los sensores, para analizarlos:

2.3.1 Zephyr BioHarness 3

Tal y como aparece recogido en su manual de uso, Zephyr BioHarness 3 (Figura 5) es un dispositivo *wearable* de monitorización fisiológica en tiempo real. El aparato consiste en un arnés para el pecho y un módulo electrónico unido al arnés. El dispositivo almacena y transmite, vía Bluetooth y USB, los signos vitales incluyendo el pulso cardíaco, frecuencia respiratoria, aceleración y orientación corporal.



Figura 5: Zephyr BioHarness 3
(Fuente: www.zephyranywhere.com)

Este dispositivo tiene su propia aplicación de escritorio, (Figura 6) la cual es muy completa pero no cumple el objetivo de movilidad y eficiencia. Además, esta aplicación está desarrollada específicamente para este aparato, lo que impide poder usar otros sensores dificultando la labor de los profesionales a la hora de recopilar y analizar los datos. Como punto final, no permite grabar en vídeo, siendo este uno de los requisitos principales del proyecto.



Figura 6: Aplicación de escritorio de Zephyr BioHarness 3
(Fuente: www.zephyranywhere.com)

Debido a la limitación de material, este dispositivo fue aportado por el departamento de Lenguaje y Sistemas Informáticos para el desarrollo de la solución **MoniSport**.

2.3.2 SensorTag CC2541

Aunque no ya no está disponible en el mercado, el kit de desarrollo SensorTag CC2541

(Figura 7), es uno de los dispositivos aportados por el departamento de Lenguajes y Sistemas Informáticos para el desarrollo del proyecto. Este kit hace más sencillo y ágil el desarrollo de aplicaciones Bluetooth Low Energy. Esto se debe a la combinación en una misma placa electrónica individual de todos los sensores utilizados habitualmente, lo cual permite realizar evaluaciones y demostraciones con rapidez. SensorTag incluye acelerómetro, un sensor de humedad, magnetómetro, sensor de presión, giroscopio y un sensor de temperatura.

Dispone de algunas aplicaciones móviles de las que se hablará en próximos apartados.



Figura 7: SensorTag CC2541
(Fuente: www.ti.com)

2.3.3 Otros sensores

Gracias a la tecnología Bluetooth Low Energy, no es complicado desarrollar aplicaciones móviles que se aprovechen de los sensores a los que tienen acceso los dispositivos. Por ello, se podrían desarrollar aplicaciones para conectar con la solución y recopilar los datos de sus sensores sin la necesidad de un dispositivo específico como los mencionados con anterioridad.

También cabe destacar que los dispositivos *wearables*, como lo son los relojes inteligentes (Figura 8), disponen de las mismas ventajas que los dispositivos móviles, pudiendo también desarrollar aplicaciones específicas para estos aparatos. De esta manera, se pueden conseguir otros tipos de sensores con los que recopilar datos para la solución.



Figura 8: Relojes inteligentes LG Watch Urbane
(Fuente: elmejorsmartwatch.es)

2.4 Aplicaciones móviles

Hay diversas aplicaciones móviles para la recopilación de datos transmitidos por dispositivos como los mencionados anteriormente. En este apartado se hablará acerca de algunas de estas aplicaciones, mencionando algunas ventajas e inconvenientes de éstas. Las valoraciones, la autoría, la fecha de actualización y las estadísticas de descarga se han obtenido de la plataforma de descargas Google Play Store. Las puntuaciones corresponden a un rango de valoración de entre 0 y 5.

2.4.1 AirCasting

- Autor: HabitatMap
- Descargas: 10.000 - 50.000
- Actualizado: 22 de junio de 2016
- Valoración / votos: 4 / 101

AirCasting (Figura 9) es de código abierto y permite guardar, mostrar y compartir datos ambientales y de salud usando un dispositivo móvil. Es capaz de conectarse a Zephyr BioHarness 3 pero la mayoría de datos que recoge vienen de otras aplicaciones que hay que instalar en el mismo dispositivo. Otro punto en contra es que no permite la grabación en vídeo.



Figura 9: Panel principal de la aplicación AirCasting
(Fuente: play.google.com)

2.4.2 Internet of Things (IoT) Dashboard/Gateway – IoTTool

- Autor: SenLab
- Descargas: 5.000 - 10.000
- Actualizado: 13 de octubre de 2017
- Valoración / votos: 4,3 / 11

Esta aplicación (Figura 10) permite monitorizar y compartir los datos de más de 100 sensores del mercado incluyendo los pertenecientes a las marcas SensorTag y Zephyr, entre otras. La problemática encontrada es que no permite la captura de vídeo, lo que es un requisito indispensable de este proyecto.

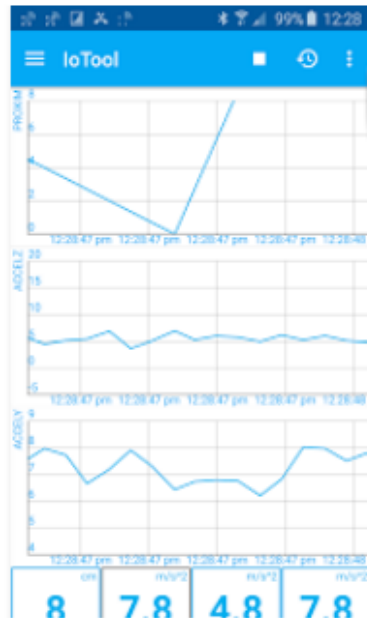


Figura 10: Panel principal de la aplicación Internet of Things (IoT) Dashboard/Gateway – IoTool
(Fuente: play.google.com)

2.4.3 Simplelink SensorTag

- Autor: Texas Instruments Inc.
- Descargas: 10.000 - 50.000
- Actualizado: 25 de mayo de 2017
- Valoración / votos: 2,9 / 280

Simplelink SensorTag (Figura 11) es la aplicación oficial de SensorTag, que permite conectar su gama de dispositivos para mostrar en tiempo real los datos recibidos. Entre las características de esta aplicación no está la de almacenar los datos de los sensores ni exportarlos. Tampoco permite grabar vídeo, es incapaz de conectarse con dispositivos no pertenecientes a su empresa y no ofrece la posibilidad de visualizar los datos guardados con anterioridad. Asimismo, cabe destacar la gran cantidad de opiniones negativas aportadas por los usuarios. Como punto final, la versión mínima de Android con la que es compatible es la 5.0 (Lollipop), lo que impide la descarga en el dispositivo facilitado para este proyecto por el departamento de Lenguajes y Sistemas de la Información.

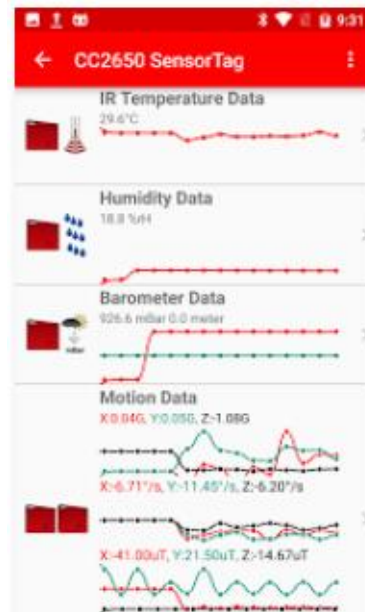


Figura 11: Panel principal de la aplicación Simplelink SensorTag
(Fuente: play.google.com/)

2.4.4 BLE SensorTag

- Autor: Steven Rudenko
- Descargas: 10.000 - 50.000
- Actualizado: 20 de julio de 2016
- Valoración / votos: 3,8 / 53

Esta es una aplicación (Figura 12) no oficial para SensorTag. Su código se puede encontrar en GitHub y es una aplicación usada para demostraciones. Al igual que la anterior, no permite grabar vídeo, almacenar y exportar datos, visualizar datos guardados y conectarse a otro tipo de dispositivo.



Figura 12: BLE SensorTag
(Fuente: play.google.com)

2.5 Conclusiones

En este apartado se han mostrado los principales problemas con las soluciones actuales que no cumplen los requisitos para este proyecto (vistos en el apartado 1.2). Con este análisis ha sido posible seleccionar funcionalidades parecidas e incluir algunas nuevas. Viendo las interfaces del resto de aplicaciones se ha podido mejorar la interfaz, ofreciendo a los usuarios una mejor usabilidad.

3. Proyecto desarrollado

En este apartado se presenta el funcionamiento de **MoniSport**: una aplicación móvil implementada en Android que se ha desarrollado para cumplir con los objetivos mencionados con anterioridad, basándose en el análisis tecnológico y solventando las problemáticas de los sistemas presentados en el apartado anterior. Además, se muestra el aspecto final de dicha aplicación.

3.1 Descripción

MoniSport (Figura 13) es una aplicación móvil con la que los especialistas podrán grabar y visualizar, en tiempo real o en diferido, sesiones de entrenamiento con sus respectivos parámetros medidos por los sensores convenientes.



Figura 13: Icono de MoniSport

Cabe destacar que el proyecto es complejo al tener conexiones con dispositivos externos que tienen un funcionamiento específico y cerrado. Otro añadido es la poca experiencia que se tiene en desarrollo en Android por parte del desarrollador. Esto complica el proyecto al tener que entender y practicar conceptos básicos y peculiaridades del sistema operativo.

La aplicación se conectará con los sensores externos que porte el deportista para así mostrar, mediante gráficas y valores, los datos transmitidos por estos. Estos datos se guardarán en el dispositivo junto con un vídeo de la actividad del deportista. El especialista podrá revisar, en cualquier momento, las sesiones guardadas, pudiendo así analizar posibles defectos o encontrar mejoras para la optimización del ejercicio del deportista.

Como se ha mencionado en apartados anteriores (2.1 Sistemas móviles), se ha escogido Android como sistema operativo en el que implementar la aplicación debido a la gran cantidad de dispositivos, de diferentes fabricantes, que utilizan dicho sistema operativo y el porcentaje de venta de dispositivos que tiene en el mercado. Además, no requiere un equipo específico para desarrollar en él, y los dispositivos aportados para el proyecto por parte del departamento de Lenguajes y Sistemas de la Información usan este sistema.

Se utilizarán los dispositivos Zephyr BioHarness 3 y SensorTag CC2541, ya que también fueron cedidos por el mismo departamento para la recopilación de los datos de los sensores a través de Bluetooth.

3.2 Funcionalidades

La aplicación dispone de las siguientes funcionalidades:

- **Almacenar datos personales referentes al deportista:**

La aplicación permitirá guardar datos personales del deportista en el propio dispositivo (Figura 14). Se almacenarán en la memoria interna del dispositivo (Figura 18). Estos datos serán:

- Nombre y apellidos.
- Edad.
- Altura.
- Peso.

El formulario tiene un título "Introduce los datos del deportista" en azul. Contiene cuatro campos de texto con etiquetas "Nombre y Apellidos:", "Edad:", "Peso:" y "Altura:". En la parte inferior hay dos botones: "Guardar" y "Cancelar".

Figura 14: Formulario para guardar los datos personales del deportista

- **Buscar dispositivos cercanos:**

Gracias a la tecnología Bluetooth, **MoniSport** permite la búsqueda y visualización de dispositivos cercanos que estén transmitiendo a través de dicha tecnología (Figura 15).

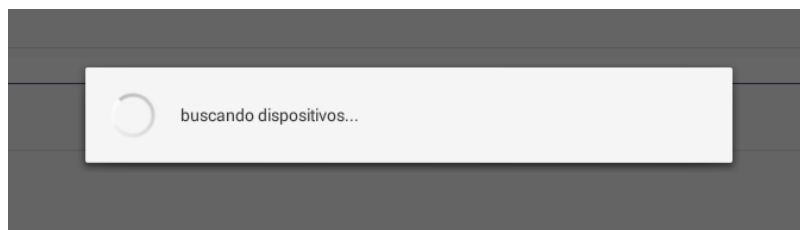


Figura 15: Cuadro de diálogo mostrado durante la búsqueda de dispositivos

- **Conectar con dispositivos cercanos:**

La aplicación está capacitada para conectarse con los dispositivos cercanos descubiertos con la funcionalidad anterior a través de Bluetooth.

- **Recibir datos de dispositivos externos:**

Tras realizar la conexión con los dispositivos externos vía Bluetooth, la aplicación permite recibir los datos que están midiendo, en tiempo real, los

sensores que disponen dichos dispositivos.

- **Visualizar los datos recibidos:**

La aplicación ofrece una visualización de los datos a modo de gráfica en el tiempo, actualizándose a tiempo real. Además permite ver el valor actual recibido de cada sensor (Figura 16).

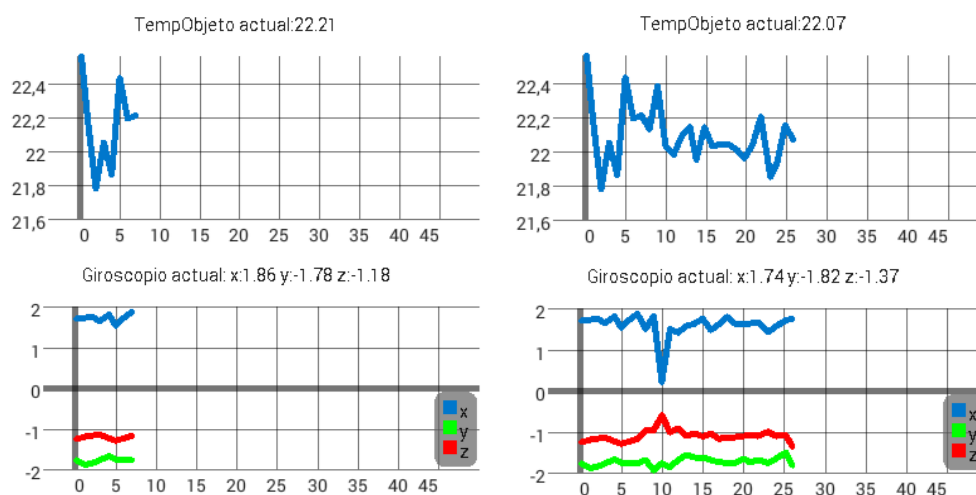


Figura 16: Gráficas con los datos recibidos a tiempo real

- **Almacenar los datos recibidos:**

La solución **MoniSport** guarda los datos recibidos en el propio dispositivo en un archivo, lo que le permite al usuario tener un completo control de ellos (Figura 18).

- **Visualizar la sesión deportiva a través de la cámara:**

La aplicación está preparada para la visualización de vídeo en directo que se sirve de la propia cámara del dispositivo y permite ver lo que sucede durante la sesión deportiva (Figura 17).



Figura 17: Previsualización de la sesión deportiva en tiempo real

- **Almacenar el vídeo de la sesión deportiva:**

El vídeo registrado por la cámara queda almacenado en la memoria interna del dispositivo inteligente para su posterior consulta (Figura 18).

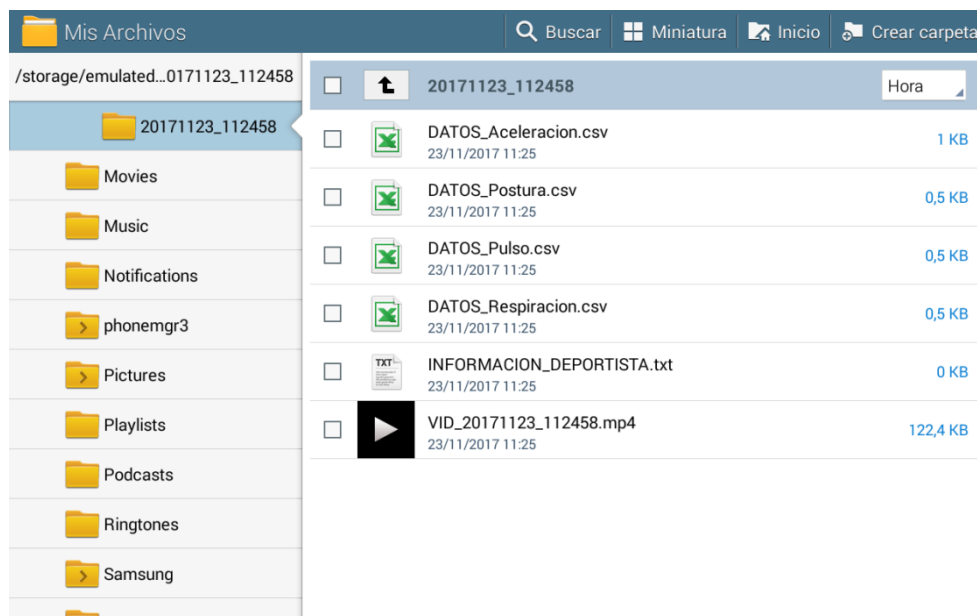


Figura 18: Directorio donde están almacenados los archivos de una sesión deportiva

- **Listar sesiones deportivas en el almacenadas en el dispositivo:**

La aplicación permite listar las sesiones deportivas almacenadas en el dispositivo, independientemente de que hayan sido obtenidas con el propio dispositivo, siempre que tengan el formato correcto (Figura 19). Los datos que se listarán son:

- Fecha y hora a la que se obtuvo la sesión.
- Datos que se guardaron durante la sesión, con el vídeo incluido.
- Dónde está almacenada la sesión deportiva.
- Datos personales del deportista.

<p>Sesión del 23/11/2017 a las 12:05 Datos: Video, Postura, Aceleracion, Respiracion, Pulso Almacenada en: /storage/emulated/0/MoniSport/Session/20171123_120512/ Información del deportista: Nombre y apellidos: chema, Edad: 25, Altura: 1,79y Peso: 77</p>	>
<p>Sesión del 23/11/2017 a las 12:05 Datos: Video, Postura, Aceleracion, Respiracion, Pulso Almacenada en: /storage/emulated/0/MoniSport/Session/20171123_120509/ Información del deportista: Nombre y apellidos: chema, Edad: 25, Altura: 1,79y Peso: 77</p>	>
<p>Sesión del 23/11/2017 a las 12:05 Datos: Video, Postura, Aceleracion, Respiracion, Pulso Almacenada en: /storage/emulated/0/MoniSport/Session/20171123_120505/ Información del deportista: Nombre y apellidos: chema, Edad: 25, Altura: 1,79y Peso: 77</p>	>

Figura 19: Listado de sesiones almacenadas en el dispositivo

- **Reproducir el vídeo de una sesión almacenada en el dispositivo:**

MoniSport reproduce el vídeo de una sesión grabada con anterioridad. El reproductor de la aplicación dispone de las habituales funcionalidades de un visor de vídeo (Figura 20):

- Pausar el vídeo.

- Reanudar el vídeo.
- Avanzar 15 segundos en el vídeo.
- Retroceder 5 segundos en el vídeo.
- Volver al inicio del vídeo.
- Navegador que muestra la duración del vídeo y permite moverse a distintos puntos del vídeo desplazando hacia adelante y hacia atrás.



Figura 20: Reproductor de vídeo visualizando una sesión deportiva almacenada

- **Visualizar los datos almacenados en el dispositivo:**

Con esta aplicación es posible visualizar los datos numéricos almacenados en el dispositivo a modo de gráfica en el tiempo, lo cual permite observar de un vistazo la evolución de los datos del entrenamiento registrados a lo largo de la duración del mismo. Es posible arrastrar la gráfica para visualizarla en su totalidad (Figura 21).

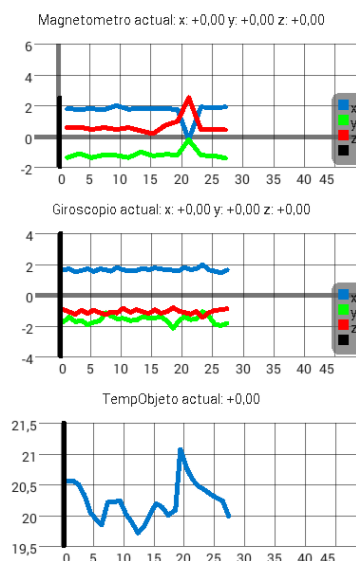


Figura 21: Gráficas con los datos almacenados de una sesión deportiva

- **Consultar los datos en un tiempo dado:**

En las gráficas para cada uno de los datos es posible seleccionar un punto para consultar el valor exacto arrojado por los sensores en ese momento del entrenamiento. (Figura 22).

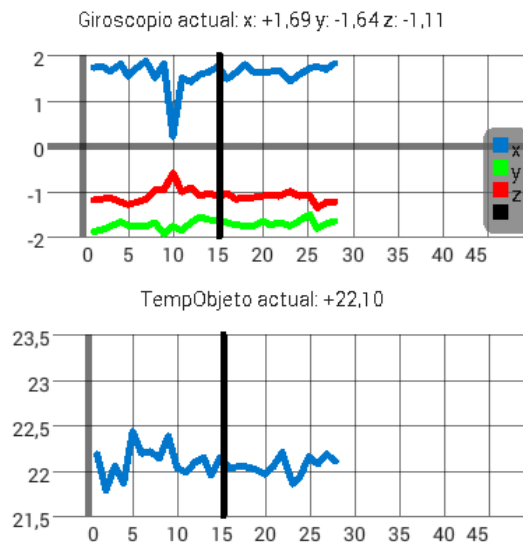


Figura 22: Gráficas con los datos almacenados, mostrando los valores del segundo 15

- **Sincronizar/Desincronizar el vídeo y las gráficas:**

La aplicación permite que las gráficas muestren el valor en cada segundo del vídeo al estar sincronizados, lo que permite que al consultar los datos pulsando en la gráfica, se mueva a ese segundo el vídeo; o viceversa: que al desplazarse por el vídeo, la gráfica muestre el valor correspondiente. La sincronización entre las gráficas y el audiovisual se puede activar o desactivar en cualquier momento (Figura 23).



Figura 23: Botones para sincronizar o desincronizar las gráficas y el vídeo reproducidos

- **Mostrar/Ocultar gráficas en la reproducción y visualización a tiempo real:**

Las gráficas que muestran los datos en la reproducción o visualización a tiempo real se podrán ocultar, cada una, pulsando su correspondiente botón. Al volver a pulsar dicho botón, aparecerá de nuevo la gráfica (Figura 24).



Figura 24: Botones para ocultar o mostrar la gráfica correspondiente

3.3 Interfaz completa

En este apartado se muestran las diferentes pantallas que componen la aplicación **MoniSport**.

- **Pantalla de inicio.**

En esta pantalla (Figura 25) se puede encontrar el logo de la aplicación. También se pueden ver dos botones. Por un lado, un botón despliega el formulario para introducir los datos del deportista (Figura 26) y una vez rellenado, lleva a la **pantalla de búsqueda y conexión de dispositivos** a través de Bluetooth. Por otro lado, un botón que permite ir a la **pantalla con el listado de sesiones deportivas almacenadas** en el dispositivo.

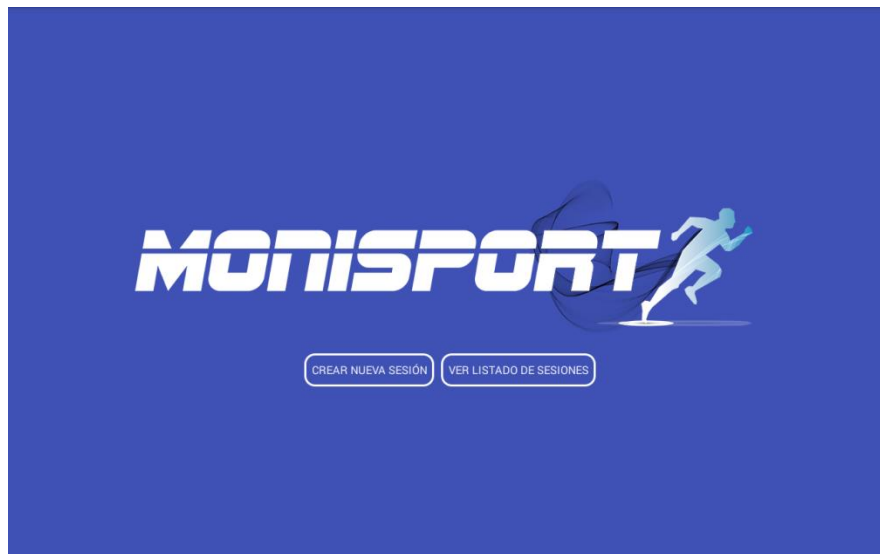


Figura 25: Pantalla de inicio de MoniSport

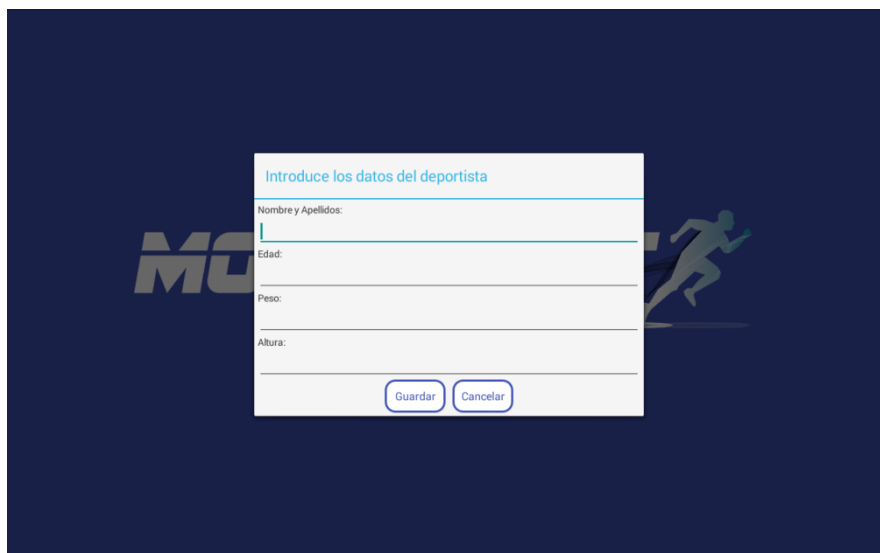


Figura 26: Pantalla de inicio con el formulario de datos personales del deportista

- **Pantalla de búsqueda y conexión de dispositivos.**

En esta pantalla (Figura 27) se despliega un listado con los dispositivos externos

vinculados con el que está ejecutando la aplicación. También aparece un botón para iniciar una búsqueda de dispositivos cercanos con el Bluetooth activado. Tras realizar dicha búsqueda, aparecerá un listado con los dispositivos encontrados. Por último, aparece un botón que conduce a la ***pantalla de grabación de sesiones deportivas***.

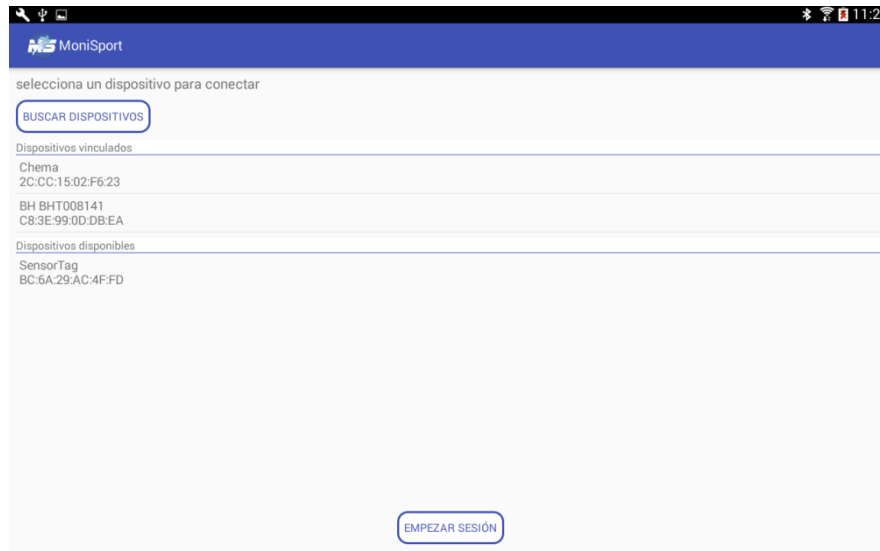


Figura 27: Pantalla de búsqueda y conexión de dispositivos

- **Pantalla de grabación de sesiones deportivas.**

A la derecha de la pantalla (Figura 28) se sitúan las gráficas, en principio vacías, de los diferentes sensores de los que se recibe señal del dispositivo seleccionado con anterioridad. También hay una previsualización de lo que esté enfocando la cámara. Finalmente, se puede ver un botón que permite empezar o finalizar la grabación.

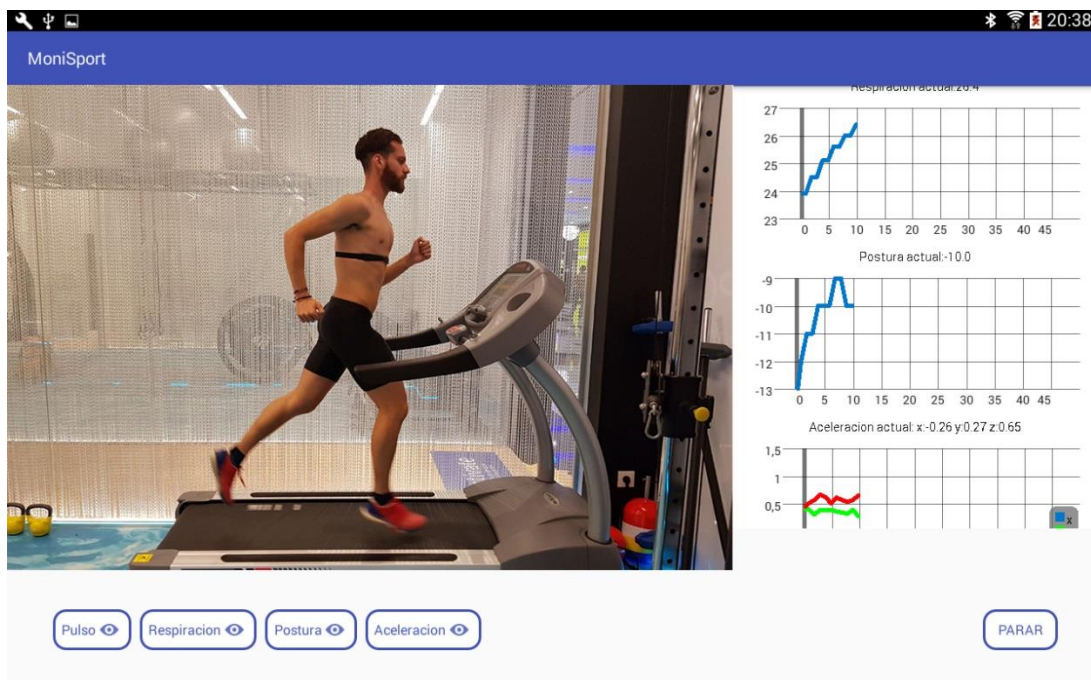


Figura 28: Pantalla de grabación de sesiones deportivas

- **Pantalla con el listado de sesiones deportivas almacenadas.**
En esta pantalla (Figura 29) es posible consultar el listado de las diferentes sesiones deportivas localizadas en el directorio de la aplicación con el formato correspondiente. Pulsar sobre alguna de ellas conducirá a la **pantalla de visualización de sesiones deportivas**.

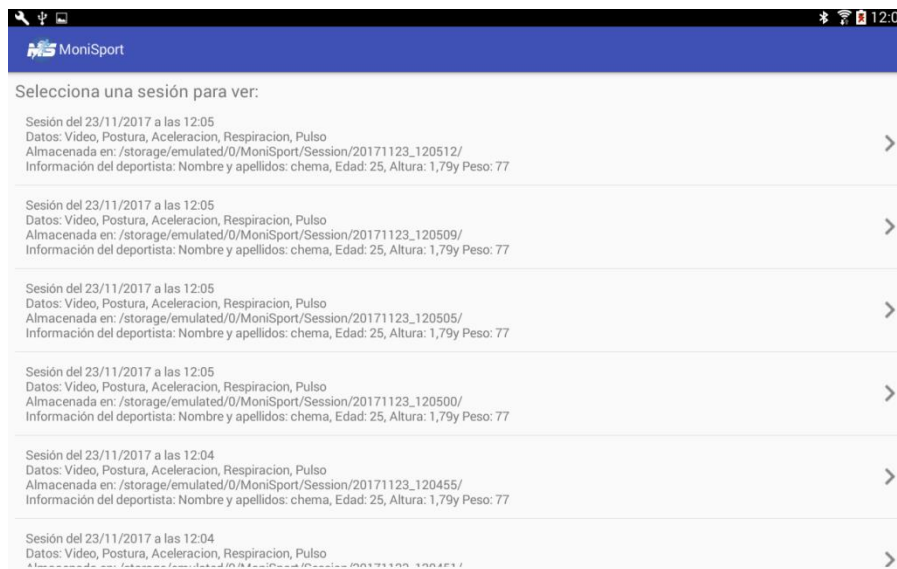


Figura 29: Pantalla de selección de sesiones deportivas almacenadas para visualizar.

- **Pantalla de visualización de sesiones deportivas almacenadas.**
En esta pantalla (Figura 30), se puede ver un reproductor del vídeo almacenado con sus respectivos controles. También están las gráficas con los datos cargados de la sesión. Por último, hay un botón para desactivar o activar la sincronización del vídeo con las gráficas.

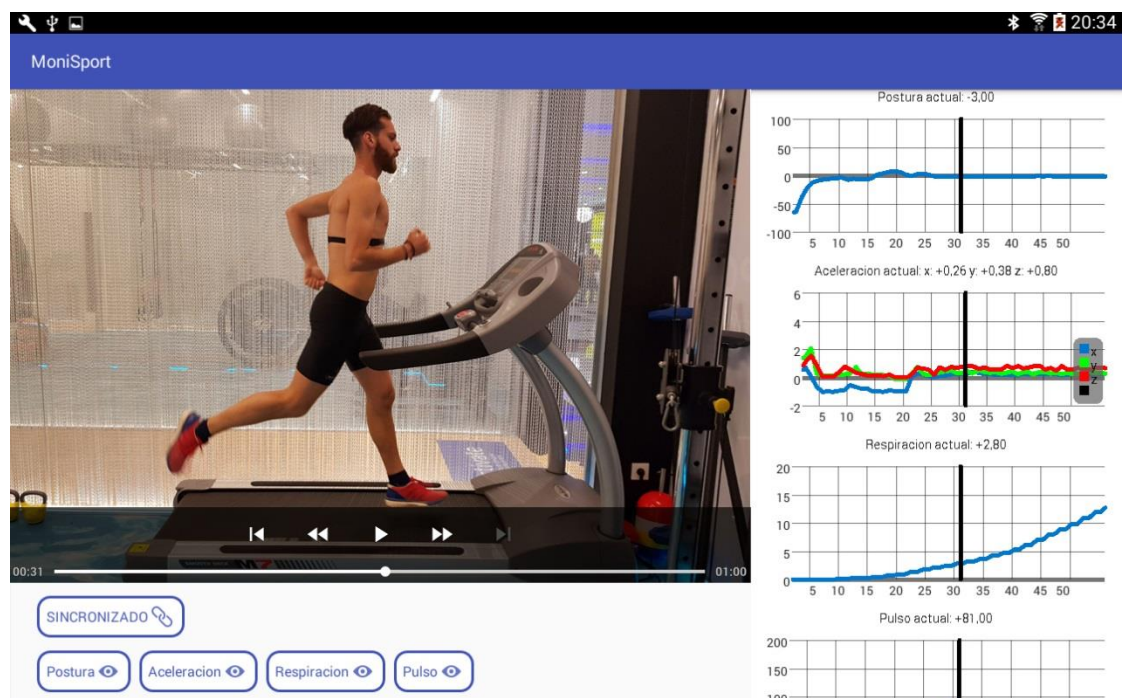


Figura 30: Pantalla de visualización de sesiones deportivas almacenadas

Sección II: Desarrollo del proyecto

4. Planificación

La planificación está dividida en las siguientes fases:

- **Fase 1: Análisis inicial.**

Esta primera fase pretende analizar y definir los objetivos y el alcance de la aplicación, a la vez que se comienza con la documentación del proyecto. Además, en esta fase se profundiza sobre el uso del entorno de desarrollo Android Studio y el desarrollo en Android que será usado para la implementación de la solución. Esta fase se compone de los siguientes objetivos:

- Planificar el proyecto.
- Estudiar aplicaciones existentes.
- Aprendizaje de Android y el entorno Android Studio.
- Creación de la documentación primaria.
- Diseño del logo de la aplicación.
- Análisis de las especificaciones del proyecto

- **Fase 2: Diseño.**

Tras el análisis de las especificaciones del proyecto, se genera un diseño que facilitará la fase de implementación y mejorará el resultado de la aplicación final. También se obtienen los bocetos sobre la interfaz de **MoniSport**.

- **Fase 3: Implementación.**

La fase de implementación corresponde al desarrollo de la propia aplicación siguiendo el diseño definido en la etapa anterior.

- **Fase 4: Pruebas.**

Como última fase, se prueba el prototipo desarrollado, permitiendo mejorar el producto volviendo a la fase anterior de ser necesario.

- **Fase 5: Documentación.**

Durante las fases anteriores se han generado todo tipo de documentos relativos al mismo. En esta última fase se organiza la información recabada, que queda recopilada en la documentación final, añadiendo lo necesario para completarla.

La planificación estimada resultante se puede ver en la siguiente figura (Figura 31):

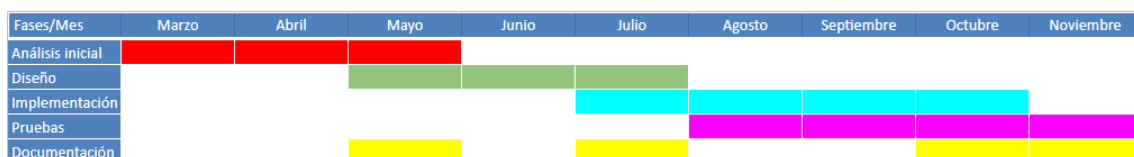


Figura 31: Planificación estimada del proyecto

5. Análisis inicial

Gran parte de lo desarrollado anteriormente conforma la fase inicial del proyecto, por lo que se resumirá brevemente.

Esta fase se compone de:

5.1 Planificación

Viene detallada en el punto 4 (Planificación), dividiéndose en:

- Fase 1: Análisis inicial
- Fase 2: Diseño
- Fase 3: Implementación
- Fase 4: Pruebas
- Fase 5: Documentación

5.2 Estudio de aplicaciones similares y ámbito tecnológico:

Viene detallado en el punto 2 (Antecedentes), se analizan las soluciones en el mercado, las tecnologías y los dispositivos disponibles.

5.3 Aprendizaje de Android y el entorno Android Studio:

Se realizaron ejercicios de prueba para la familiarización con el entorno de Android Studio y con Android siguiendo el libro 'Manual imprescindible de desarrollo de aplicaciones para Android: edición 2016', de Joan Ribas Lequerica.

5.4 Creación de documentación primaria:

Con los datos recopilados a lo largo del proyecto, en lo que se lleva de proyecto se empieza a realizar la documentación.

5.5 Diseño del logo de la aplicación:

Se diseñan varios bocetos para utilizar como logotipo e icono de la aplicación y se selecciona uno de ellos para su funcionalidad visual.

5.6 Análisis de las especificaciones del proyecto:

Se eligió utilizar una metodología ágil basada en SCRUM. Debido a que es un proyecto individual y sin un cliente definido, se ha optado por no seguir completamente la metodología, utilizando las partes que se han considerado más útiles y relevantes para este proyecto. Este apartado se puede subdividir en:

5.6.1 Escenarios

Se realizaron varios escenarios. Un escenario es un ejemplo del uso de la aplicación desde el punto de vista del usuario con un estilo narrativo, dotándole de un contexto

determinado. Con ello, conseguimos extraer funcionalidades y factores que de otro modo podrían pasar inadvertidos. Un ejemplo que se usó para este proyecto es:

Actor	
Datos	Nombre: Salva Edad: 25 años Objetivo: Mejorar el rendimiento deportivo de sus clientes.
Contexto de uso	Cuando: Cada dos días. Donde: Actívate Gym Nevada Shopping Dispositivos: Samsung Galaxy Tab 3 y Zephyr BioHarness 3
Escenario	
<p>Salva es un entrenador personal contratado en Actívate Gym. Dispone de una tablet y un dispositivo Zephyr para la monitorización deportiva. Decide descargarse la aplicación MoniSport para poder hacer un análisis de sus clientes y así mejorar su rendimiento deportivo a la vez que previene lesiones.</p> <p>A cada cliente le coloca el dispositivo Zephyr en el pecho y conecta la tablet con él vía Bluetooth. El cliente empieza a correr en la cinta ergonómica y Salva comienza la grabación de la sesión deportiva. Tras finalizar la sesión, Salva analiza con ayuda de la aplicación los datos recopilados y el vídeo detectando pequeños errores a la hora de realizar el ejercicio físico que impiden que el cliente obtenga el máximo rendimiento en la sesión deportiva.</p>	

5.6.2 Actores

En este apartado listaremos los diferentes actores que tienen interacción con la aplicación:

- **Usuario:** es el actor principal en la aplicación e interactuará a través de la interfaz de usuario.
- **Dispositivo externo con Bluetooth:** responsable de enviar los datos que recopile a la aplicación a través de Bluetooth.
- **Sistema Android:** es el sistema operativo del dispositivo con la ejecución de la aplicación. Se encargará de enviar los eventos del sistema a la aplicación.

5.6.3 Requisitos

- **Requisitos funcionales**
 - **RF1 Visualización de sesiones deportivas.**
 - **RF1.1 Búsqueda de dispositivos.**
La aplicación detectará dispositivos cercanos que tengan activado el Bluetooth, listándolos.
 - **RF1.2 Conexión con dispositivos.**

La aplicación se conectará con dispositivos a través de Bluetooth.

- **RF1.3 Recepción de datos.**

La aplicación recibirá los datos transmitidos por los dispositivos conectados vía Bluetooth.

- **RF1.4 Visualización de datos.**

La aplicación mostrará gráficas con los datos recibidos y el valor actual en tiempo real.

- **RF1.5 Visualización de sesión deportiva a tiempo real.**

La aplicación mostrará la previsualización de la imagen obtenida a través de la cámara del dispositivo en tiempo real.

- **RF1.6 Ocultado de gráficas.**

La aplicación permitirá ocultar las gráficas.

- **RF1.7 Mostrado de gráficas.**

La aplicación permitirá mostrar las gráficas que fueron ocultadas con anterioridad.

- **RF2 Almacenado de sesiones deportivas.**

- **RF2.1 Almacenado de datos.**

La aplicación almacenará los datos recibidos por los dispositivos externos en archivos dentro del dispositivo.

- **RF2.2 Almacenado de vídeo.**

La aplicación almacenará en vídeo las imágenes recogidas por la cámara.

- **RF2.3 Almacenado de datos personales del deportista.**

La aplicación almacenará los datos personales del deportista a través de un formulario.

- **RF3 Reproducción de sesiones deportivas.**

- **RF3.1 Listado con las sesiones deportivas almacenadas.**

La aplicación generará un listado con las sesiones deportivas almacenadas en el dispositivo y los datos que contienen.

- **RF3.2 Reproducción de vídeo almacenado.**

La aplicación reproducirá el vídeo de las sesiones deportivas almacenadas en el dispositivo.

- **RF3.2.1 Pausa del vídeo.**
La aplicación pausará el vídeo de la sesión deportiva que se esté reproduciendo.
- **RF3.2.2 Reanudación del vídeo.**
La aplicación reanudará el vídeo de la sesión deportiva que se esté reproduciendo.
- **RF3.2.3 Avanzado del vídeo.**
La aplicación hará avanzar el vídeo que se esté reproduciendo en el tiempo 15 segundos.
- **RF3.2.4 Retroceso del vídeo.**
La aplicación hará retroceder el vídeo que se esté reproduciendo en el tiempo 5 segundos.
- **RF3.2.5 Vuelta al inicio del vídeo.**
La aplicación hará volver al inicio del vídeo que se esté reproduciendo.
- **RF3.2.6 Navegación del vídeo.**
La aplicación avanzará y retrocederá en el vídeo a un punto seleccionado gracias a la barra de navegación.
- **RF3.3 Visualización de los datos almacenados.**
La aplicación mostrará las gráficas de los datos numéricos almacenados en el dispositivo en forma de gráficas en el tiempo.
 - **RF3.3.1 Desplazamiento horizontal.**
La aplicación se desplazará horizontalmente para la completa visualización de la gráfica.
- **RF3.4 Consulta de datos.**
La aplicación mostrará el valor de las gráficas en un segundo dado al pulsar sobre una de ellas.
- **RF3.5 Sincronización del vídeo y las gráficas.**
La aplicación sincronizará el vídeo con las gráficas para que muestren el valor en cada momento durante la reproducción del vídeo.
 - **RF3.5.1 Sincronización vídeo-gráficas**
La aplicación sincronizará el vídeo con las gráficas de manera que cualquier cambio en el tiempo actual del vídeo mostrará el valor que tienen las gráficas en ese nuevo instante seleccionado.

- **RF3.5.2 Sincronización gráficas-vídeo**

La aplicación sincronizará las gráficas con el vídeo, de manera que, al seleccionar un valor en las gráficas, el vídeo irá a ese instante de tiempo.

▪ **RF3.6 Desincronización del vídeo y las gráficas**

La aplicación desincronizará el vídeo con las gráficas para que los cambios en cualquiera de los componentes, vídeo y gráficas, no afecten.

▪ **RF3.7 Ocultado de gráficas.**

La aplicación permitirá ocultar las gráficas.

▪ **RF3.8 Mostrado de gráficas.**

La aplicación permitirá mostrar las gráficas que fueron ocultas con anterioridad.

• **Requisitos no funcionales**

- **RNF1** El dispositivo en que sea usada la aplicación dispondrá de Bluetooth.
- **RNF2** La aplicación podrá ejecutarse de forma nativa en dispositivos con sistema operativo 4.2.2 o superior.
- **RNF3** El dispositivo en que sea usada la aplicación dispondrá de cámara de grabación de vídeo.
- **RNF4** La aplicación será compatible con los dispositivos SensorTag CC2541 de Texas Instruments.
- **RNF5** La aplicación será compatible con los dispositivos Zephyr BioHarness 3.
- **RNF6** La arquitectura de la aplicación permitirá extender, de manera sencilla, los dispositivos compatibles.
- **RNF7** La aplicación se podrá conectar con otros dispositivos inteligentes vía Bluetooth para recibir datos de aplicaciones instaladas en ellos.
- **RNF8** La aplicación tendrá una interfaz simple para mejorar la usabilidad.
- **RNF9** Los archivos con los datos recibidos de los sensores se guardarán en formato CSV.
- **RNF10** Los archivos de vídeo de las sesiones deportivas se guardarán con formato MP4.

- **RNF11** Los archivos con los datos personales del deportista se almacenarán con formato TXT.

6. Diseño

6.1 Diagrama de componentes principales

En base al análisis anterior, se ha desarrollado este diagrama. En él se puede ver la estructura que tendrá la aplicación. La clase con más importancia será la ConexiónBluetooth que permitirá buscar y conectar con los dispositivos externos. Con la clase MainActivity se iniciará la aplicación, permitiendo navegar al resto de clases y sus respectivas funcionalidades.

Como se puede observar, hay diversos paquetes externos que han sido usados en el proyecto. Esto se debe a que cumplen ciertas funcionalidades necesarias para la aplicación. En apartados posteriores, se analizarán la implementación y funciones de cada clase.

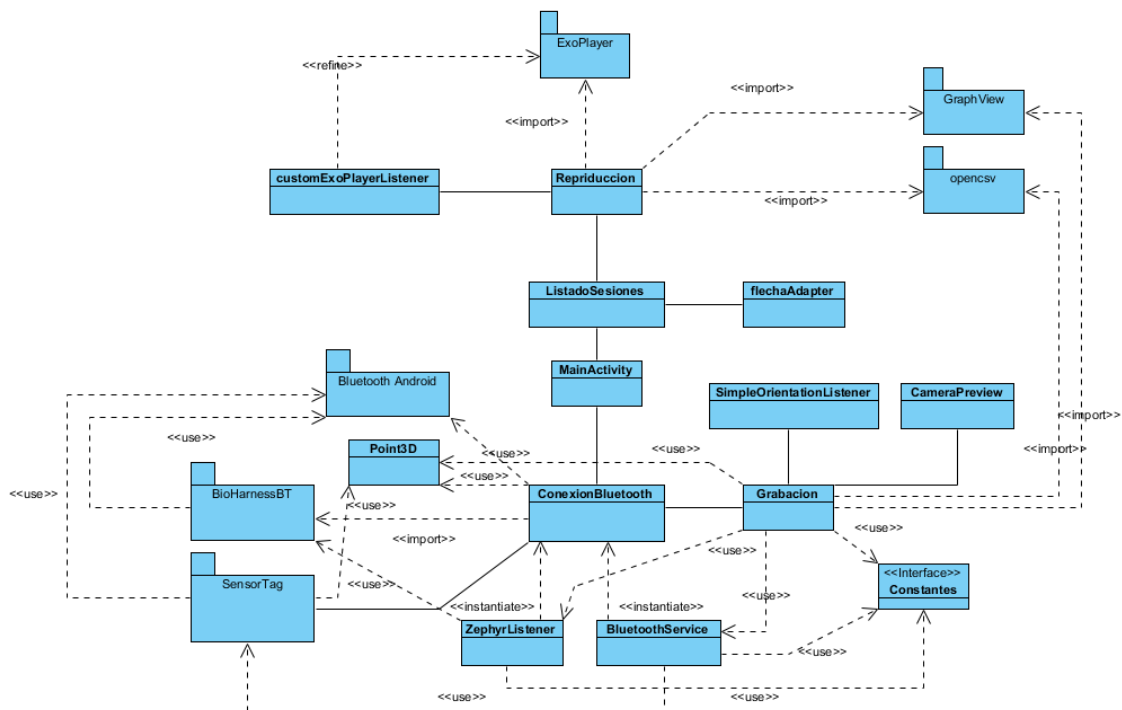


Figura 32: Diagrama de componentes principales

6.2 Arquitectura de la aplicación

En este apartado se describirán los componentes básicos y la interacción que se puede dar entre ellos.

6.2.1 Componentes de la aplicación

Los componentes básicos de la aplicación se pueden dividir en tres grupos:

- **Sistema Bluetooth.** Será el encargado de buscar, conectar y recibir los datos de

los dispositivos externos.

- **Grabación y visualización de vídeo y datos.** Se encargará de mostrar los datos y el vídeo a tiempo real y de grabarlos posteriormente.
- **Recuperación y reproducción de datos y vídeo.** Se encargará de cargar los datos de los archivos almacenados y mostrarlos.
- **GUI.** Al ser una aplicación en Android, la interfaz gráfica del usuario tiene que ser construida a través de archivos XML, formando un componente en la arquitectura de la aplicación.

6.2.2 Interacción de los componentes

Se van a mencionar brevemente, en este apartado, las interacciones que tienen los componentes entre sí, ya que en los apartados siguientes se especificará más sobre la implementación de cada uno de estos componentes y sus respectivas interacciones.

- **GUI**
A través de la **GUI** se podrá navegar al resto de los componentes y se mostrará la información recabada por cada uno de ellos.
 - **Recuperación y reproducción de datos y vídeo:** el listado de las sesiones deportivas guardadas, las gráficas con los datos numéricos y el vídeo almacenado.
 - **Sistema Bluetooth:** el listado de dispositivos vinculados, los resultados de la búsqueda de dispositivos y la conexión con el dispositivo.
 - **Grabación y visualización de vídeo y datos:** las gráficas con los datos numéricos recibidos en tiempo real y la previsualización de la cámara del dispositivo.
- **Sistema Bluetooth**
Además de lo mencionado anteriormente, el **Sistema Bluetooth** interactuará con la **Grabación y visualización de vídeo y datos** para facilitarle los datos transmitidos por los dispositivos externos conectados.
- **Grabación y visualización de vídeo y datos**
En los puntos anteriores se mencionan las interacciones de este componente con los expuestos.
- **Recuperación y reproducción de datos y vídeo**
Aunque de manera indirecta, este componente es capaz de realizar su función gracias a los datos que fueron almacenados con anterioridad por la **Grabación y visualización de vídeo y datos**, aunque no es estrictamente necesario ya que,

mientras lo archivos estén en el directorio correcto y tengan el formato usado, la aplicación será capaz de recuperarlos y reproducirlos.

7. Implementación

En esta sección se explican detalladamente los componentes de la aplicación. Para ello se pormenorizan las clases desarrolladas durante la ejecución del proyecto, profundizando en las funcionalidades que cumplen.

- **MainActivity**

Al no pertenecer a ningún componente concreto se comenzará hablando de la clase MainActivity. Esta clase es la primera a la cual se llama durante la ejecución de la aplicación y su cometido es permitir al usuario navegar a las principales funcionalidades como son la conexión con Bluetooth para guardar una sesión deportiva o visualizar las sesiones ya almacenadas. También mostrará un formulario para introducir la información personal del deportista antes de crear una nueva sesión.

7.1 Sistema Bluetooth

- **ConexionBluetooth**

Esta clase es la más importante del proyecto. Gracias a ella es posible la búsqueda de dispositivos externos con Bluetooth activado cerca de la aplicación. Además, muestra la lista de dispositivos externos vinculados al que ejecuta la aplicación para, en caso de que estén transmitiendo, poder conectarse a ellos. En la siguiente figura () se puede ver la función encargada de la búsqueda de dispositivos y su conexión.

```
private final BroadcastReceiver mReceiver = (context, intent) -> {
    String action = intent.getAction();
    if (BluetoothDevice.ACTION_FOUND.equals(action)) {
        BluetoothDevice device = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
        if (device.getBondState() != BluetoothDevice.BOND_BONDED) {
            mNewDevicesArrayAdapter.add(device.getName() + "\n" + device.getAddress());
        }
    } else if (BluetoothAdapter.ACTION_DISCOVERY_FINISHED.equals(action)) {
        texto_info.setText("selecciona un dispositivo para conectar");
        if (mNewDevicesArrayAdapter.getCount() == 0) {
            String noDevices = "No se encuentran dispositivos".toString();
            mNewDevicesArrayAdapter.add(noDevices);
        }
        dialog.hide();
    }
    else if (BluetoothService.ACTION_GATT_CONNECTED.equals(action)) {
        // GATT connect
        int status = intent.getIntExtra(BluetoothService.EXTRA_STATUS,
            BluetoothGatt.GATT_FAILURE);
        if (status == BluetoothGatt.GATT_SUCCESS) {
            mBtGatt = BluetoothService.getBtGatt();
            mBtGatt.discoverServices();
        } else {
            Toast.makeText(getApplication(), "Connect failed. Status: " + status,
                Toast.LENGTH_LONG).show();
        }
    }
};
```

Figura 33: Código encargado de buscar, mostrar y conectar los dispositivos Bluetooth

Esta clase se vale de los paquetes **BioHarnessBT** y **SensorTag** para gestionar la conexión con los dispositivos Zephy BioHarness 3 y SensorTag CC2541.

Tras la conexión con un dispositivo, instanciará un servicio y un listener para que el componente de grabación pueda acceder a los datos transmitidos por los dispositivos externos.

- **ZephyrListener**

Este listener es el encargado de transmitir los datos de los dispositivos Zephyr al componente de grabación y visualización en tiempo real, valiéndose del paquete **BioHarnessBT** para la recepción y procesamiento de los datos recibidos. En la siguiente figura (Figura 34) se puede ver parte del código que se encarga de transmitir los datos recibidos por el sensor.

```
public void Connected(ConnectedEvent<BTClient> eventArgs) {
    Log.d(TAG, String.format("Connected to BioHarness %s.", eventArgs.getSource().getDevice().getName()));
    RqPacketType.GP_ENABLE = true;
    RqPacketType.BREATHING_ENABLE = true;
    RqPacketType.LOGGING_ENABLE = true;

    ZephyrProtocol _protocol = new ZephyrProtocol(eventArgs.getSource().getComms(), RqPacketType);
    _protocol.addZephyrPacketEventListener((eventArgs) -> {
        ZephyrPacketArgs msg = eventArgs.getPacket();
        byte CRCFailStatus;
        byte RcvdBytes;

        CRCFailStatus = msg.getCRCStatus();
        RcvdBytes = msg.getNumRcvdBytes();
        int MsgID = msg.getMsgID();
        byte [] DataArray = msg.getBytes();
        switch (MsgID)
        {
            case GP_MSG_ID:

                //*****Displaying the Heart Rate*****
                int HRate = GPInfo.GetHeartRate(DataArray);
                Message text1 = _aNewHandler.obtainMessage(HEART_RATE);
                Bundle b1 = new Bundle();
                b1.putString("HeartRate", String.valueOf(HRate));
                text1.setData(b1);
                _aNewHandler.sendMessage(text1);
            // ...
        }
    });
}
```

Figura 34: Código encargado de enviar los datos recibidos por el Zephyr Listener

- **BluetoothServices**

Este servicio transmite a la grabación y visualización en tiempo real los datos recibidos por el dispositivo SensorTag o en general por dispositivos que usen tecnología Bluetooth Low Energy. Esto es una gran ventaja ya que se pueden desarrollar aplicaciones específicas para dispositivos inteligentes y conectarlas a **MoniSport**. En la siguiente figura (Figura 35) se puede ver la función que se encarga de transmitir los datos recibidos por el BluetoothServices.

```

private void broadcastUpdate(final String action,
                            final BluetoothGattCharacteristic characteristic, final int status) {

    final Intent intent = new Intent(action);
    intent.putExtra(EXTRA_UUID, characteristic.getUuid().toString());
    intent.putExtra(EXTRA_DATA, characteristic.getValue());
    intent.putExtra(EXTRA_STATUS, status);
    sendBroadcast(intent);
    Message text1 = new Message();
    Bundle b1 = new Bundle();
    if (characteristic.getUuid().toString().equals(SensorTagGatt.UUID_ACC_DATA.toString())) {
        text1 = _aNewHandler.obtainMessage(Constants.ACCELERATION_SENSORTAG);
        b1.putByteArray("Acelerometro", characteristic.getValue());
    }
    if (characteristic.getUuid().toString().equals(SensorTagGatt.UUID_MAG_DATA.toString())) {
        text1 = _aNewHandler.obtainMessage(Constants.MAGNETOMETER_SENSORTAG);
        b1.putByteArray("MAGNETOMETER", characteristic.getValue());
    }
    if (characteristic.getUuid().toString().equals(SensorTagGatt.UUID_GYR_DATA.toString())) {
        text1 = _aNewHandler.obtainMessage(Constants.GYROSCOPE_SENSORTAG);
        b1.putByteArray("GYROSCOPE", characteristic.getValue());
    }
    if (characteristic.getUuid().toString().equals(SensorTagGatt.UUID_IRT_DATA.toString())) {
        text1 = _aNewHandler.obtainMessage(Constants.IR_TEMPERATURE_SENSORTAG);
        b1.putByteArray("IR_TEMPERATURE", characteristic.getValue());
    }
    if (characteristic.getUuid().toString().equals(SensorTagGatt.UUID_HUM_DATA.toString())) {
        text1 = _aNewHandler.obtainMessage(Constants.HUMIDITY_SENSORTAG);
        b1.putByteArray("HUMIDITY", characteristic.getValue());
    }
    if (characteristic.getUuid().toString().equals(SensorTagGatt.UUID_BAR_DATA.toString())) {
        text1 = _aNewHandler.obtainMessage(Constants.BAROMETER_SENSORTAG);
        b1.putByteArray("BAROMETER", characteristic.getValue());
    }
    text1.setData(b1);
    mBusy = false;

    _aNewHandler.sendMessage(text1);
}

```

Figura 35: Código encargado de enviar los datos recibidos por el BluetoothServices

7.2 Grabación y visualización a tiempo real

- **Grabación**

Esta clase es la más importante de este componente. Se encarga de la gestión de la cámara del dispositivo, permitiendo que se graben en vídeo las imágenes obtenidas.

Gracias al **ZephyrListener** y **BluetoothServices**, genera las gráficas para los datos recibidos por los sensores usando la librería **GraphView**. En la siguiente imagen (Figura 36) se puede ver parte del código encargado de recibir los datos y mostrarlos.


```

private final Handler handlerDatos = handleMessage(msg) → {
    if (viendose) {
        byte[] rawValue;
        Point3D v;
        String mensaje;
        switch (msg.what) {
            case Constantes.HEART_RATE_ZEPHYR:
                if (!graficas.containsKey("Pulso")) {
                    crearGrafica("Pulso");
                }
                if (isRecording) {
                    String HeartRateText = msg.getData().getString("HeartRate");
                    setDatosGrafica("Pulso", HeartRateText);
                }
                //Log.d(TAG, "Heart Rate Info is " + HeartRateText);
                break;
            case Constantes.RESPIRATION_RATE_ZEPHYR:
                if (!graficas.containsKey("Respiracion")) {
                    crearGrafica("Respiracion");
                }
                if (isRecording) {
                    String PeakAccText = msg.getData().getString("RespirationRate");
                    setDatosGrafica("Respiracion", PeakAccText);
                }
                //Log.d(TAG, "Respiration Rate Info is " + RespirationRateText);
                break;
        }
    }
}

```

Figura 36: Parte del código encargado de recibir los datos y mostrarlos

Por último, guarda los datos obtenidos anteriormente en archivos CSV usando la librería **opencsv**.

- **CameraPreview**

Es la responsable de mostrar en tiempo real lo que está captando el objetivo de la cámara del dispositivo. Es usando en la clase **Grabación**.

- **SimpleOrientationListener**

Debido al control que requiere la cámara en Android, se implementa este listener para poder interpretar los cambios de orientación en el dispositivo mientras se está grabando. Se utiliza en la clase **Grabación**.

7.3 Recuperación y visualización

- **ListadoSesiones**

Se encarga de obtener un listado con las sesiones almacenadas en el dispositivo. Muestra información relativa a ellas, como los datos que contiene, cuándo fue obtenida y, por último, la ubicación de la misma. Además, al seleccionar una de ellas conducirá a la pantalla para poder visualizarla.

- **Reproducción**

Esta clase se encarga de mostrar el reproductor con el vídeo de la sesión deportiva a través de la librería **ExoPlayer**. También tiene como función construir las gráficas con los datos almacenados en los archivos CSV utilizando las librerías **opencsv** y **GraphView**. En la siguiente imagen (Figura 37) se puede ver parte del código encargado de actualizar las gráficas siguiendo al vídeo, si está sincronizado.

```

public void actualizarGráficas(){
    if(sincronizar){
        Iterator it=graficas.entrySet().iterator();
        while (it.hasNext()) {
            HashMap.Entry pair = (HashMap.Entry) it.next();
            GraphView g = (GraphView) pair.getValue();
            List<Series> todasSeries=g.getSeries();
            Viewport viewport=g.getViewport();
            LineGraphSeries<DataPoint> series = (LineGraphSeries<DataPoint>) todasSeries.get(g.getSeries().size() - 1);
            LineGraphSeries<DataPoint> seriesPrimera=(LineGraphSeries<DataPoint>)todasSeries.get(0);
            if(seriesPrimera.getHighestValueX() > 50){
                if(tiempo_actual < 25){
                    if(viewport.getMinX(false) != -4){
                        viewport.setMinX(-4);
                        viewport.setMaxX(50);
                    }
                } else if (tiempo_actual >= 25){
                    if(tiempo_actual+27<=seriesPrimera.getHighestValueX()+4) {
                        viewport.setMinX(tiempo_actual-29);
                        viewport.setMaxX(tiempo_actual+25);
                    } else if (viewport.getMaxX(false) != seriesPrimera.getHighestValueX()+4){
                        viewport.setMinX(seriesPrimera.getHighestValueX()-50);
                        viewport.setMaxX(seriesPrimera.getHighestValueX()+4);
                    }
                }
            }
        }

        DataPoint[] data=new DataPoint[2];
        data[0]= new DataPoint(tiempo_actual,viewport.getMinY(false));
        data[1]= new DataPoint(tiempo_actual,viewport.getMaxY(false));
        series.resetData(data);

        int tamano= todasSeries.size();
        String valor="";
        for(int i=0; i<tamano-1;i++){
            if(tamano==2) {
                valor += getValorInterpolado((LineGraphSeries<DataPoint>) todasSeries.get(i), tiempo_actual);
            } else {
                switch (i){
                    case 0:
                        valor += "x: " + getValorInterpolado((LineGraphSeries<DataPoint>) todasSeries.get(i), tiempo_actual);
                }
            }
        }
    }
}

```

Figura 37: Parte del código encargado de actualizar el valor mostrado por la gráfica si está sincronizado

Por último, permite la sincronización de las gráficas con el vídeo valiéndose de la clase **customExoPlayerListener** para controlar las interacciones del usuario con el reproductor y así poder avanzar en las gráficas o viceversa.

7.4 GUI

La GUI ha sido diseñada e implementada para que el uso de la aplicación sea intuitivo y fácil de aprender. En la siguiente figura (Figura 38) se pueden ver los archivos que la componen.

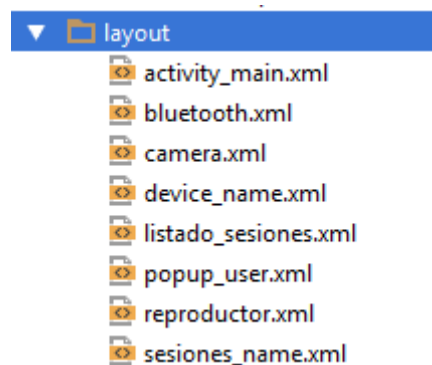


Figura 38: Directorio con los archivos que componen la GUI

8. Pruebas

Las pruebas de la aplicación se distribuyeron a lo largo de todo el proceso de desarrollo del proyecto.

Estas pruebas se han realizado mediante el depurador de Android Studio en cada ejecución. Cada vez que se cumplía un objetivo del proyecto o se implementaba una nueva funcionalidad se lanzaba una ejecución de la aplicación esperando tener ciertos resultados de la misma. Si los resultados eran correctos, se daba por validada la funcionalidad. En caso contrario, se analizaban los errores para modificar la implementación y llegar a un correcto funcionamiento.

Cuando se comprobó el correcto funcionamiento de todas las características de la aplicación (conexión con dispositivos externos, grabación y visualización a tiempo real), se testeó en una situación real, gracias a la colaboración de un entrenador personal.

No se han realizado pruebas unitarias por la prioridad de cumplir las funcionalidades definidas. Esto se tiene en cuenta para las posibles mejoras del proyecto en el futuro.

9. Herramientas utilizadas

Para el desarrollo de la aplicación se han utilizado las siguientes herramientas:

- **Samsung Galaxy Tab 3**

Esta tablet fue facilitada por el departamento de Lenguajes y Sistemas de la Información y ha sido sobre la que se han realizado la mayoría de pruebas de la aplicación. El sistema operativo que utiliza es Android 4.4.2 KitKat.

- **Zephyr BioHarness 3**

Las características y uso de este dispositivo se han detallado en anteriores apartados (ver 2.3.1 Zephyr BioHarness 3). Fue cedido por el departamento de Lenguajes y Sistemas de la Información y ha sido usado para las pruebas de conexión, grabación y visualización a tiempo real de datos.

- **SensorTag CC2541**

En apartados anteriores (ver 2.3.2 SensorTag CC2541) se ha profundizado en este dispositivo. Al igual que el anterior, fue aportado por el departamento de Lenguajes y Sistemas de la Información y utilizado para las pruebas de conexión, grabación y visualización a tiempo real de datos.

- **Android Studio 2.3.3**

Este entorno de desarrollo es el oficial para la plataforma Android. Es gratuito y está basado en el software de IntelliJ IDEA de JetBrains.

Sección III: Conclusiones y trabajo futuro

10. Conclusiones

MoniSport es una aplicación que ofrece ayuda a especialistas en ejercicio físico con el objetivo de monitorizar las sesiones de entrenamiento de los deportistas para su posterior análisis. Bajo esta premisa, se han cumplido los objetivos que se propusieron al comienzo del proyecto.

En primer lugar, durante la elaboración de este proyecto se han podido aplicar los conocimientos adquiridos a lo largo de la etapa formativa en el Grado, generando un software útil y funcional. Debido a los pocos conocimientos sobre Android al comienzo del proyecto, es posible que la implementación se pueda mejorar, pero la aplicación resultante es capaz de realizar las funciones para las que se desarrolló.

MoniSport es capaz de visualizar sesiones deportivas a tiempo real. Para ello permite conectarse con dispositivos externos como lo son Zephyr BioHarness 3 y SensorTag, recibir datos de estos y mostrarlos en la interfaz del usuario a la vez que se visualiza la sesión de entrenamiento a través de la cámara del dispositivo.

Permite almacenar las sesiones deportivas para su posterior uso, ya sea con la aplicación o sin ella, al tener los archivos en formatos útiles para otro tipo de softwares del mercado, lo que no limita a los especialistas.

Es capaz de reproducir las sesiones almacenadas, hayan sido grabadas o no con el propio dispositivo. **MoniSport** muestra las gráficas con los datos almacenados y permite reproducir el vídeo, siendo capaz de sincronizar ambos. Gracias a esto, los especialistas no necesitan diferentes dispositivos ni software para realizar análisis que ayuden al deportista.

La interfaz no está sobrecargada y es intuitiva para que aprender a utilizar la aplicación no sea complejo.

Por último, al ser una aplicación móvil, **MoniSport** no limita el uso a laboratorios y no impide o dificulta su desplazamiento como otros equipos del mercado.

A nivel personal, el alumno ha adquirido conocimientos en la programación móvil que no se tenían antes del comienzo del proyecto, pero es capaz de comprender que necesita más práctica para mejorar en ello.

11. Trabajo futuro

Tras el análisis del proyecto, se pudo ver que es un proyecto completo y puede tener mejoras en funcionalidades o sencillamente incorporar nuevas. Se mencionan algunas de las que han ido surgiendo a lo largo del proyecto:

- **Traducción a diferentes idiomas.**
No existe la necesidad de limitarse al mercado de habla hispana, la aplicación podría permitir seleccionar el idioma de uso.
- **Gestión avanzada de perfiles de deportistas.**
Ya que se guardan los datos de los deportistas, poder gestionar esos perfiles ayudaría a los especialistas en su labor de análisis.
- **Filtrado de sesiones deportivas o búsqueda.**
Para mayor utilidad para el especialista, las sesiones deportivas que se muestran se podrían filtrar, ordenar o buscar a través de un formulario.
- **Sistema de autenticación.**
Debido a que varios especialistas pueden usar el mismo dispositivo, un sistema de autenticación permitiría la separación de las sesiones guardadas.
- **Conexión con la nube.**
Con el sistema de autenticación, se podría implementar la conexión con la nube para poder guardar las sesiones deportivas y utilizarlas en diferentes dispositivos.
- **Interfaz para importación de datos.**
Ya que la aplicación no necesita que los datos se hayan guardado con la misma, una interfaz para seleccionar archivos e importarlos agilizaría el proceso.
- **Web.**
Al estar la aplicación conectada a la nube, se podría desarrollar una web que accediera a los datos para que el usuario pudiera consultarlos a través de un navegador. También permitiría la importación y exportación de los datos y el control de los datos del deportista.
- **Ampliación de sensores compatibles.**
Debido a la gran cantidad de sensores y marcas en el mercado, es necesario aumentar la conectividad con ellos para ampliar los clientes potenciales.
- **Manual de usuario integrado.**
Un manual de usuario en la propia aplicación permitiría al especialista aprender rápidamente el funcionamiento de la misma.

Sección IV: Manual del usuario

- **Pantalla de inicio**

Cuando se inicia la aplicación se puede ver la **pantalla de inicio** (Figura 39). Sirve de unión para las demás pantallas. En ella, aparece el logo de la aplicación junto con dos botones.

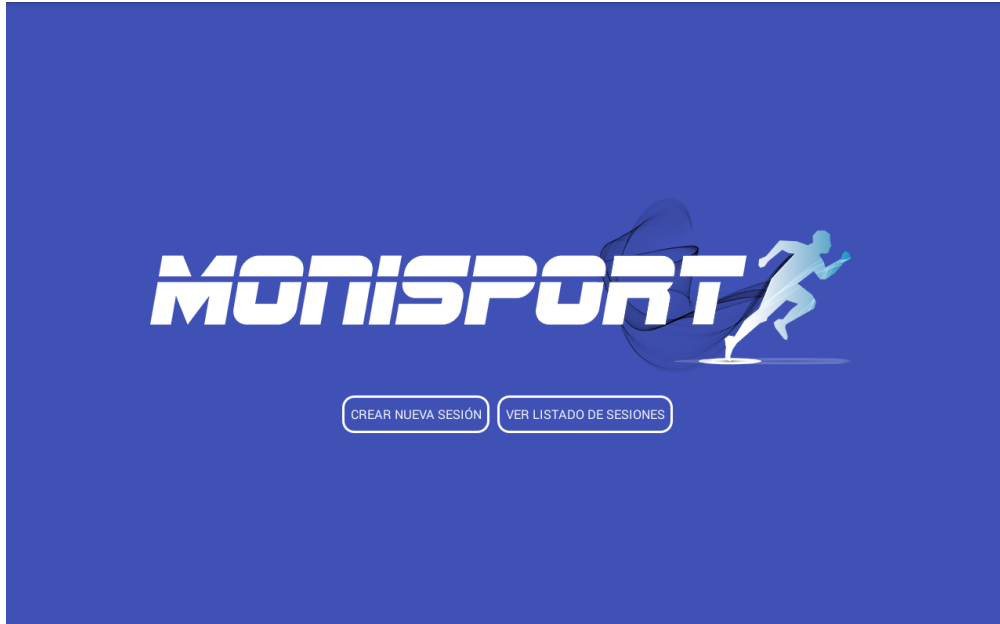


Figura 39: Pantalla de inicio de MoniSport

Pulsando el botón Crear nueva sesión, aparecerá un formulario para rellenar los datos personales del deportista (Figura 40Figura 40). Tras rellenarlo y pulsar Guardar, se accederá a la **pantalla búsqueda y conexión de dispositivos**. Si se pulsa Cancelar, desaparecerá el formulario.

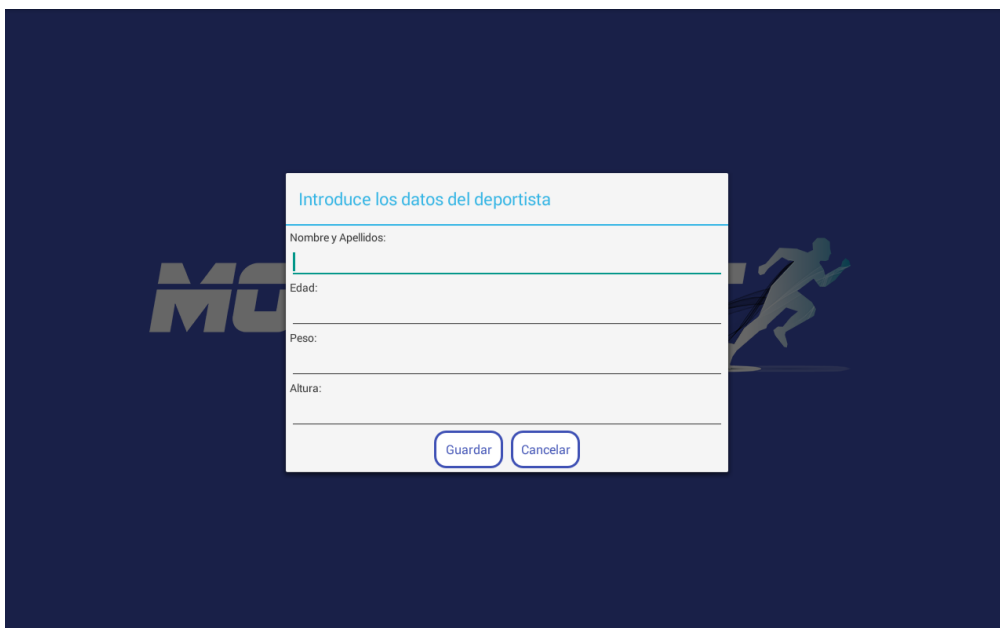


Figura 40: Pantalla de inicio con el formulario de datos perosnales del deportista desplegado

Pulsando el botón Ver listado de sesiones, se accederá a la **pantalla con el listado de sesiones deportivas almacenadas**.

- **Pantalla de búsqueda y conexión de dispositivos**

En esta pantalla (Figura 41) aparecerán dos botones, un listado con los dispositivos vinculados y otro con la lista de dispositivos cercanos encontrados tras realizar una búsqueda.

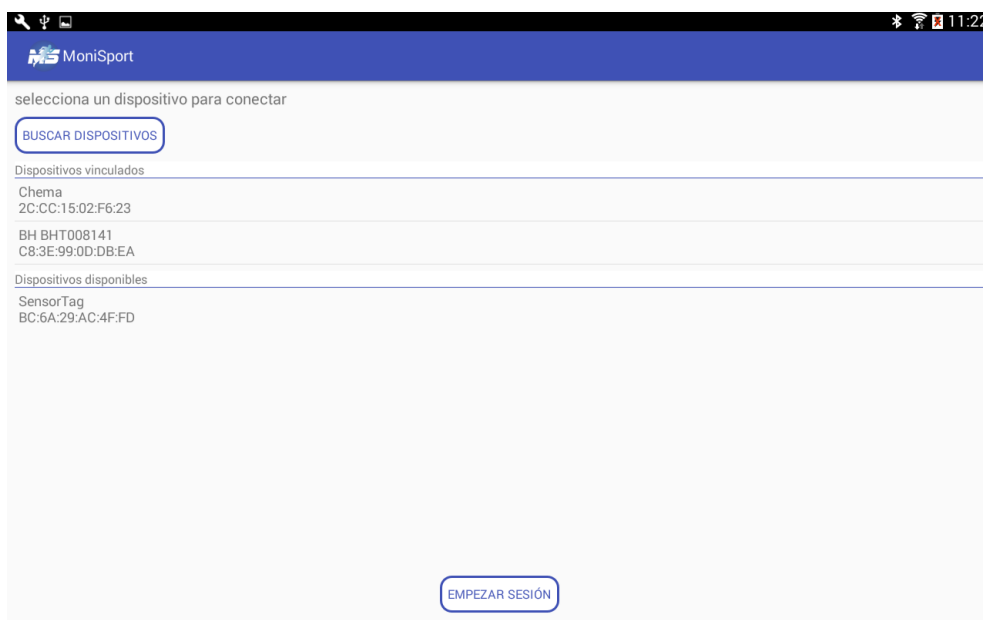


Figura 41: Pantalla de búsqueda y conexión de dispositivos

Pulsando el botón Buscar dispositivos, se realizará una búsqueda de dispositivos cercanos con Bluetooth activado. Esto bloqueará durante 12 segundos la interacción para poder realizar la búsqueda (Figura 42).

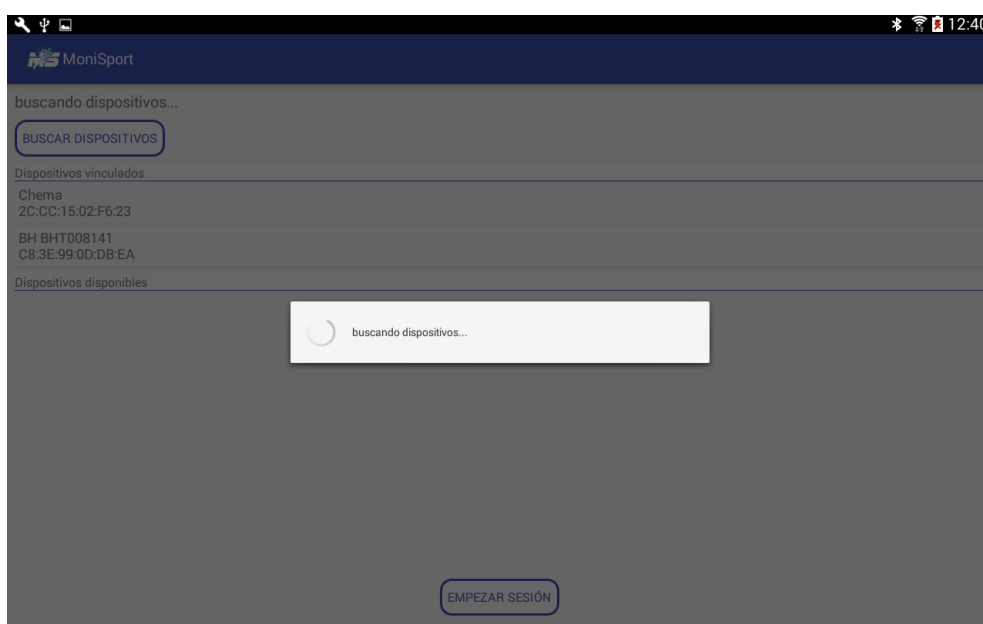


Figura 42: Pantalla de búsqueda y conexión de dispositivos mientras realiza una búsqueda

Pulsando cualquiera de los dispositivos en ambas listas, intentará conectar con ellos mostrando un mensaje si consigue realizar la conexión o si no ha sido capaz.

Pulsando el botón Ir a la cámara, se accederá a la **pantalla de grabación de sesiones deportivas**.

- **Pantalla de grabación de sesiones deportivas**

En esta pantalla (Figura 43) aparecerá una previsualización de lo que esté enfocando la cámara, las gráficas de cada sensor detectado y un botón.

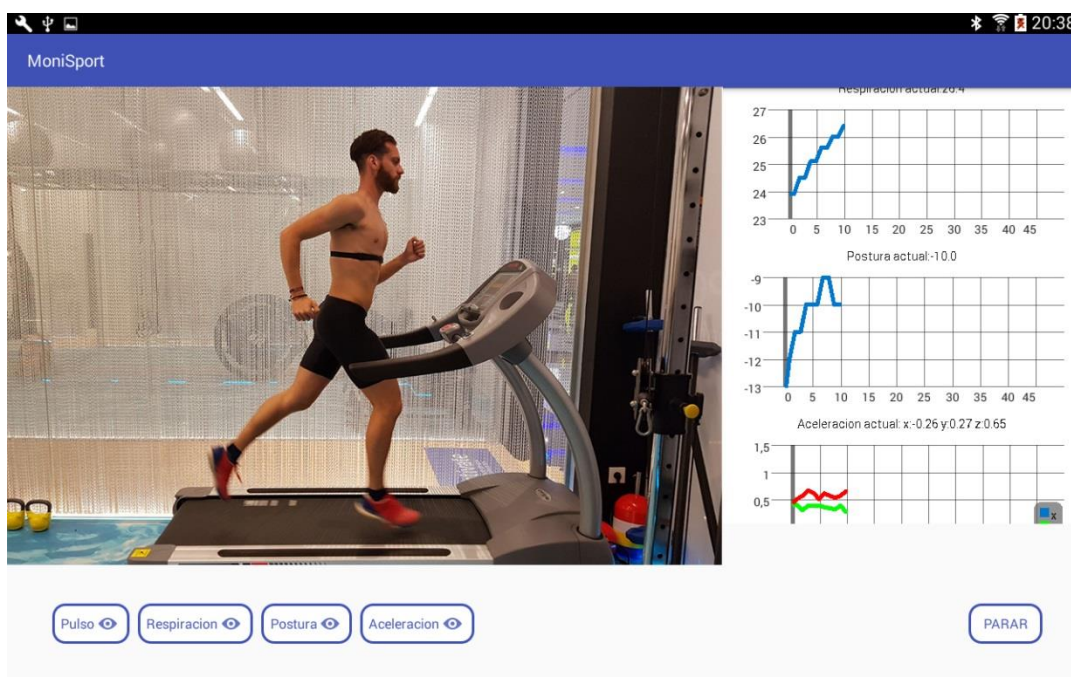




Figura 43: Pantalla de grabación de sesiones deportivas

Si la cantidad de gráficas es mayor de dos, se podrá desplazar hacia arriba y hacia abajo arrastrando con el dedo.

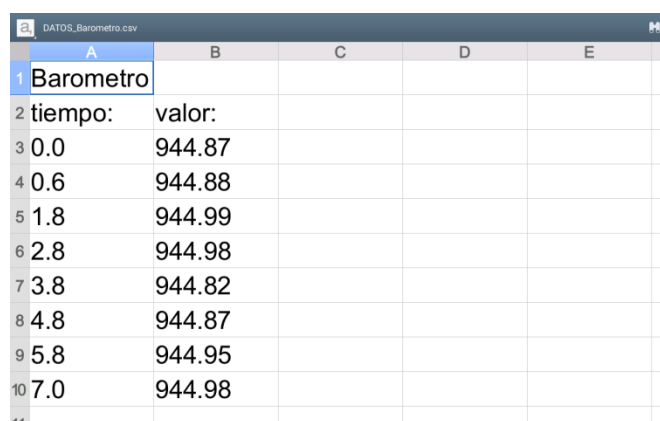
Pulsando el botón Grabar, se empezarán a mostrar los datos recibidos por los sensores en las gráficas y comenzará la grabación del vídeo. El botón cambiará a Parar.

Pulsando el botón Parar, se detendrá la grabación del vídeo y limpiará las gráficas para poder grabar de nuevo. El botón cambiará a Grabar.

Pulsando sobre los botones con los nombres de los sensores, desaparecerán o aparecerán las gráficas correspondientes a ese sensor. El símbolo del ojo  significa que la gráfica está visible. El símbolo del ojo tachado  significa que la gráfica está oculta.

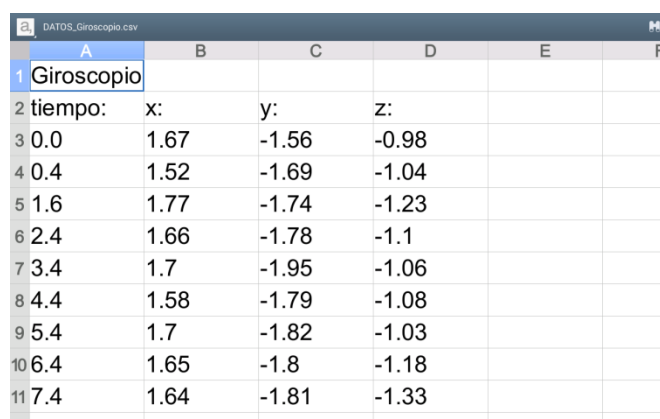
Al detener la grabación de vídeo, se guardará el archivo resultante en el dispositivo con nombre: “VID_ año mes día_hora minuto segundo.mp4”. También se guardarán los datos transmitidos por los sensores en formato CSV siguiendo las estructuras que se muestran en la (Figura 44), en caso de tener una sola componente y en la (Figura 45) en caso de tener varias componentes. El nombre de estos archivos será: “DATOS_tipodato.csv” Esto se guardará en el directorio:

“/storage/emulated/0/MoniSport/Session/año mes día_hora minuto segundo/”.
 Por ejemplo: DATOS_Giroscopio.csv y VID_20171119_193934.mp4 se almacenan en “/storage/emulated/0/MoniSport/Session/20171119_193934/”.



	A	B	C	D	E
1	Barometro				
2	tiempo:	valor:			
3	0.0	944.87			
4	0.6	944.88			
5	1.8	944.99			
6	2.8	944.98			
7	3.8	944.82			
8	4.8	944.87			
9	5.8	944.95			
10	7.0	944.98			

Figura 44: Archivo CSV con los datos de una sesión deportiva con una sola componente



	A	B	C	D	E	F
1	Giroscopio					
2	tiempo:	x:	y:	z:		
3	0.0	1.67	-1.56	-0.98		
4	0.4	1.52	-1.69	-1.04		
5	1.6	1.77	-1.74	-1.23		
6	2.4	1.66	-1.78	-1.1		
7	3.4	1.7	-1.95	-1.06		
8	4.4	1.58	-1.79	-1.08		
9	5.4	1.7	-1.82	-1.03		
10	6.4	1.65	-1.8	-1.18		
11	7.4	1.64	-1.81	-1.33		

Figura 45: Archivo CSV con los datos de una sesión deportiva con tres componentes

- **Importar datos no grabados con MoniSport o desde otro dispositivo**

Para realizar esta función, solamente hay que crear una carpeta en el directorio: “/storage/emulated/0/MoniSport/Session/” con el nombre de “año mes día_hora minuto segundo” y pegar ahí los archivos de datos y vídeo. Estos archivos deben nombrarse como se ha mencionado en el punto anterior y deben tener el formato estipulado también en el punto anterior.

- **Pantalla con el listado de sesiones deportivas almacenadas**

En esta pantalla (Figura 46) aparecerá un listado de sesiones deportivas almacenadas en el dispositivo en caso de que existieran. Este listado contendrá

la información de cuando se almacenó la sesión, los datos que tiene esa sesión, incluyendo el vídeo, y donde está almacenada.



Figura 46: Pantalla con el listado de sesiones deportivas almacenadas

Pulsar sobre cualquiera de las sesiones nos conducirá a la **Pantalla de visualización de sesiones deportivas almacenadas**.

- **Pantalla de visualización de sesiones deportivas almacenadas**

En esta pantalla (Figura 47) aparecerá un reproductor de vídeo para ver el vídeo almacenado de la sesión. Este reproductor tendrá los típicos botones de un reproductor: el botón de pausa/reanudación del vídeo, el botón de avanzar en el vídeo, el botón de retroceder en el vídeo y el botón para volver al inicio del vídeo. Además, incluye una barra de navegación que muestra el segundo actual del vídeo y la duración total de este.

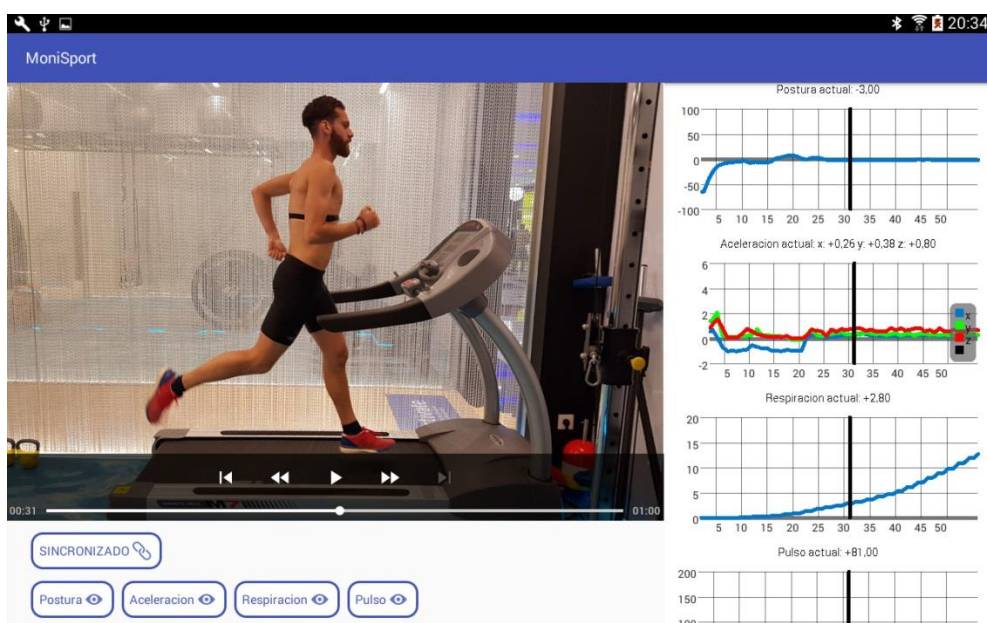




Figura 47: Pantalla de visualización de sesiones deportivas almacenadas

También se puede observar las gráficas con los datos almacenados de la sesión. Pulsar sobre las gráficas mostrará el valor que tienen en el segundo seleccionado. Si hay más de tres gráficas, se podrá desplazar verticalmente para poder visualizar las demás. Si la gráfica tiene más de 50 valores, se podrá desplazar horizontalmente para ver el resto de valores.

Habrà un botón Sincronizado. Al estar en activado, los movimientos en la línea temporal del vídeo actualizarán el valor que muestran las gráficas desplazándolas si es necesario y pulsado sobre las gráficas llevará al vídeo al segundo seleccionado.

Al pulsar el botón Sincronizado, lo explicado en el párrafo anterior dejará de surtir efecto y el botón cambiará a Desincronizado con este símbolo .

Al pulsar el botón Desincronizado, el botón cambiará a Sincronizado con este símbolo  y volverá a funcionar la sincronización entre vídeo y gráficas.

Bibliografía

- ¿En qué consiste Bluetooth LE?* (s.f.). Recuperado el Marzo de 2017, de hipertextual: <https://hipertextual.com/2013/12/que-es-bluetooth-le>
- AirCasting*. (s.f.). Recuperado el Abril de 2017, de Google Play: <https://play.google.com/store/apps/details?id=pl.llp.aircasting&hl=en>
- An extensible media player for Android*. (s.f.). Recuperado el Octubre de 2017, de Github: <https://github.com/google/ExoPlayer>
- Android*. (s.f.). Recuperado el Mayo de 2017, de Wikipedia: <https://es.wikipedia.org/wiki/Android>
- Android + Pair devices via bluetooth programmatically*. (s.f.). Recuperado el 2017 de Agosto, de StackOverFlow: <https://stackoverflow.com/questions/14228289/android-pair-devices-via-bluetooth-programmatically>
- Android 8.0 Oreo sólo ha llegado a un 0,2% de los dispositivos en su primer mes*. (s.f.). Recuperado el Noviembre de 2017, de Xataka: <https://www.xatakandroid.com/sistema-operativo/android-8-0-oreo-solo-ha-llegado-a-un-0-2-de-los-dispositivos-en-su-primer-mes>
- Android Custom ListView with Image and Text*. (s.f.). Recuperado el Octubre de 2017, de Androidhive: <https://www.androidhive.info/2012/02/android-custom-listview-with-image-and-text/>
- Android Studio*. (s.f.). Recuperado el Mayo de 2017, de Wikipedia: https://es.wikipedia.org/wiki/Android_Studio
- ArrayAdapter*. (s.f.). Recuperado el Octubre de 2017, de Android Developer: <https://developer.android.com/reference/android/widget/ArrayAdapter.html>
- Ballesteros, N. V. (Julio de 2015). *Granadian: Una guía turística para móviles*. Granada.
- BioHarness 3 User Manual*. (s.f.). Recuperado el Abril de 2017, de Zephyr Technology: <https://www.zephyranywhere.com/media/download/bioharness3-user-manual.pdf>
- BLE SensorTag*. (s.f.). Recuperado el Mayo de 2017, de Google Play: <https://play.google.com/store/apps/details?id=sample.ble.sensortag&hl=es>
- Bluetooth Low Energy*. (s.f.). Recuperado el Agosto de 2017, de Android Developer: <https://developer.android.com/guide/topics/connectivity/bluetooth-le.html?hl=es-419>
- CC2541 SensorTag Development Kit*. (s.f.). Recuperado el Mayo de 2017, de Texas Instruments: <http://www.ti.com/tool/CC2541DK-SENSOR>
- CC2541 SensorTag Quick Start Guide*. (s.f.). Recuperado el 2017 de Mayo, de Texas Instruments: <http://www.ti.com/lit/ug/swru324b/swru324b.pdf>
- ExoPlayer*. (s.f.). Recuperado el Octubre de 2017, de Android Developer: <https://developer.android.com/guide/topics/media/exoplayer.html>
- Gartner Says Worldwide Sales of Smartphones Grew 9 Percent in First Quarter of 2017*. (Mayo de 2017). Recuperado el Mayo de 2017, de Gartner: <https://www.gartner.com/newsroom/id/3725117>
- González, J. L. (Julio de 2012). *Smart Transport Guide*. Granada.
- Guijarro, A. J. (Septiembre de 2017). *Food4living*. Granada.
- How can I set camera preview size to squared aspect ratio in a squared SurfaceView*. (s.f.). Recuperado el Agosto de 2017, de StackOverFlow:

<https://stackoverflow.com/questions/11121963/how-can-i-set-camera-preview-size-to-squared-aspect-ratio-in-a-squared-surfaceview>

How to create a simple Graph. (s.f.). Recuperado el Agosto de 2017, de GraphView: <http://www.android-graphview.org/simple-graph/>

How to export data to csv file in Android? (s.f.). Recuperado el Agosto de 2017, de StackOverFlow: <https://stackoverflow.com/questions/27772011/how-to-export-data-to-csv-file-in-android>

How to receive serial data using android bluetooth. (s.f.). Recuperado el Julio de 2017, de StackOverFlow: <https://stackoverflow.com/questions/13450406/how-to-receive-serial-data-using-android-bluetooth>

How to record video of particular width and height on samsung device android? (s.f.). Recuperado el Julio de 2017, de StackOverFlow: <https://stackoverflow.com/questions/20245625/how-to-record-video-of-particular-width-and-height-on-samsung-device-android>

How to remove title bar from the android activity? (s.f.). Recuperado el Noviembre de 2017, de StackOverFlow: <https://stackoverflow.com/questions/36236181/how-to-remove-title-bar-from-the-android-activity>

Icons8. (s.f.). Recuperado el Noviembre de 2017, de Icons8: <https://icons8.com/>

Internet of Things(IoT) Dashboard/Gateway - IoTTool. (s.f.). Recuperado el Mayo de 2017, de Google Play: <https://play.google.com/store/apps/details?id=io.senlab.iotoolapp>

iOS. (s.f.). Recuperado el Abril de 2017, de Wikipedia: <https://es.wikipedia.org/wiki/IOS>

Kit de desarrollo SensorTag CC2541 de Texas Instruments. (s.f.). Recuperado el Noviembre de 2017, de Mouser Electronics: <https://www.mouser.es/new/Texas-Instruments/ti-cc2541-sensortag/>

Lequerica, J. R. (2016). *Manual imprescindible de desarrollo de aplicaciones para Android: edición 2016*. ANAYA MULTIMEDIA.

MediaRecorder. (s.f.). Recuperado el Agosto de 2017, de Android Developer: <https://developer.android.com/guide/topics/media/mediarecorder.html>

PRUEBA DE ESFUERZO COMPLETA: Electrocardiograma, monitorización y análisis de gases. (s.f.). Recuperado el Noviembre de 2017, de Let'Sport: <https://letsport.es/prueba-de-esfuerzo-completa-electrocardiograma-monitorizacion-y-analisis-de-gases-694>

Quesada, A. C. (Septiembre de 2016). Integración de aplicaciones y servicios para pulseras inteligentes en CloudFit. Granada.

Realtime updates / Live chart. (s.f.). Recuperado el Agosto de 2017, de GraphView: <http://www.android-graphview.org/realtime-chart/>

Salguero, A. C. (Septiembre de 2014). Gait Analyzer: App para el análisis del modo de caminar. Granada.

SensorTag-Android-2541. (s.f.). Recuperado el Agosto de 2017, de Gitorius TI: <https://git.ti.com/sensortag-android>

Simplelink SensorTag. (s.f.). Recuperado el Mayo de 2017, de Google Play: <https://play.google.com/store/apps/details?id=com.ti.ble.sensortag&hl=es>

SimpleOrientationListener. (s.f.). Recuperado el Julio de 2017, de GithubGist: <https://gist.github.com/PavelSynek/64dab386e4930e55c39e>

Technology, Z. (s.f.). BioHarness BT Example Android Project.

What's The Difference Between Bluetooth Low Energy And ANT? (s.f.). Recuperado el

Abril de 2017, de ElectronicDesign:
<http://www.electronicdesign.com/mobile/what-s-difference-between-bluetooth-low-energy-and-ant>
ZephyrBioHarnessRegistrator.java. (s.f.). Recuperado el Agosto de 2017, de GitHub:
https://github.com/senseobservationsystems/sense-android-library/blob/master/sense-android-library/src/nl/sense_os/service/external_sensors/ZephyrBioHarnessRegistrator.java
ZEPHYR™ PERFORMANCE SYSTEMS: COMPONENTS. (s.f.). Recuperado el Mayo de 2017, de Medtronic: <https://www.zephyranywhere.com/system/components>