



## ***Wearable de medida de rendimiento en running***

**Diego Sánchez Llorach**

**Tutor: Marina Alonso Díaz**

**Cotutor: Salvador Coll Arnau**

Trabajo Fin de Grado presentado en la Escuela  
Técnica Superior de Ingenieros de  
Telecomunicación de la Universitat Politècnica de  
València, para la obtención del Título de Graduado  
en Ingeniería de Tecnologías y Servicios de  
Telecomunicación

Curso 2020-21

Valencia, 2 de diciembre de 2020



## Resumen

La finalidad de este trabajo es el desarrollo de una aplicación para dispositivos móviles de análisis de rendimiento en el deporte de *running*. Por lo tanto, se ha hecho uso de un prototipo *hardware* con las especificaciones necesarias -microprocesador, sensores de movimiento y comunicación con dispositivos móviles- para la obtención de los datos recogidos y envío al dispositivo móvil para el posterior análisis.

En cuanto al prototipo *hardware*, se han analizado los diferentes dispositivos *wearables* disponibles en el mercado, teniendo en cuenta que cumplan los requisitos necesarios y buscando siempre la mejor relación calidad-precio posible.

El dispositivo *wearable* colocado en el tobillo es capaz de medir la calidad de la pisada y enviar los consiguientes datos vía Bluetooth al dispositivo móvil.

En el dispositivo móvil, mediante la aplicación creada es posible visualizar gráficamente los resultados, además de poder gestionar y descargar los datos en un fichero.

Finalmente, los datos descargados se trasladan a un ordenador para ser analizados e interpretados mediante un programa de análisis matemático.

De esta manera, es posible ayudar al deportista a mejorar en este deporte que en los últimos años esta tan de moda.

**Palabras Clave:** *running*, rendimiento, *wearable*.

## Resum

La finalitat d'aquest treball es el desenvolupament d'una aplicació per a dispositius mòbils d'anàlisi de rendiment en l'esport de *running*. Per tant, s'ha fet ús d'un prototip *hardware* amb les especificacions necessàries -microprocessador, sensors de moviment i comunicació amb dispositius mòbils- per a l'obtenció de les dades arreglades i enviament al dispositiu mòbil per al posterior anàlisi.

Quant al prototip *hardware*, s'han analitzat els diferents dispositius *wearables* disponibles en el mercat, tenint en compte que complisquen els requisits necessaris i buscant sempre la millor relació-qualitat preu possible.

El dispositiu *wearable* col·locat en el turmell és capaç de mesurar la qualitat de la xafada i enviar les conseqüents dades via Bluetooth al dispositiu mòbil.



En el dispositiu mòbil, mitjançant l'aplicació creada es possible visualitzar gràficament els resultats, a més de poder gestionar y descarregar les dades en un fitxer.

Finalment, les dades descarregades es traslladen a un ordinador per a ser analitzades i interpretades mitjançant un programa d'anàlisi matemàtic.

Amb tot això, és possible trobar errors en la xafada i ajudar a l'esportista a millorar en aquest esport que en els últims anys es troba tant de moda.

**Paraules clau:** *running*, rendiment, *wearable*.

## **Abstract**

The purpose of this work is the development of an application for mobile devices for performance analysis in running sport. Therefore, a hardware prototype with the necessary specifications -microprocessor, motion sensors and communication with mobile devices- has been used to obtain the collected data and send it to the mobile device for further analysis. Respect to the hardware prototype, the different wearable devices available on the market have been analysed, taking into account that they meet the necessary requirements and always seeking the best possible quality-price ratio.

The wearable device placed on the ankle is capable of measuring the quality of the footprint and sending the resulting data via Bluetooth to the mobile device.

On the mobile device, using the application created, it is possible to graphically display the results, as well as to manage and download the data to a file.

Finally, the downloaded data is transferred to a computer to be analysed and interpreted by means of a mathematical analysis programme.

In this way, it is possible to find errors in the footprint and help the athlete to improve in this sport that in recent years is so fashionable.

**Key words:** *running*, performance, *wearable*.



## Índice

Capítulo 1.	Introducción.....	8
1.1	Motivación.....	8
1.2	Objetivos.....	9
1.3	Estructura de la memoria.....	10
Capítulo 2.	Tecnología disponible.....	11
2.1	Dispositivos <i>wearables</i> .....	11
2.1.1	SensorTag CC2650.....	11
2.1.2	SensorTag CC3200.....	13
2.1.3	STEVAL-WESU1.....	14
2.1.4	Otras alternativas.....	15
2.2	Sistemas Operativos.....	16
2.2.1	Android.....	16
2.2.2	iOS.....	16
Capítulo 3.	Descripción de la solución adoptada.....	18
3.1	Diagrama de bloques general.....	18
3.2	Hardware.....	18
3.2.1	Razones de elección.....	18
3.2.2	Diagrama de bloques.....	20
3.2.3	Detalles técnicos.....	20
3.3	Software.....	23
3.3.1	Razones de elección.....	23
3.3.2	Diagrama de bloques.....	24



Capítulo 4.	Fundamentos teóricos .....	25
4.1	Protocolo Bluetooth Low Energy .....	25
4.2	Representación de la orientación .....	25
4.2.1	Ángulos de Euler .....	26
4.2.2	Cuaterniones .....	28
4.2.3	Gimbal Lock .....	29
4.3	Medida de orientación .....	30
4.3.1	Unidad de Medición Inercial .....	30
4.3.2	Sistemas de Referencia de Actitud y Rumbo (AHRS) .....	31
4.3.3	Filtro de Kalman .....	32
4.3.4	Algoritmo de Madgwick .....	33
Capítulo 5.	Detalles de la implementación .....	35
5.1	Hardware .....	35
5.2	Software .....	35
5.2.1	Búsqueda del dispositivo .....	35
5.2.2	Obtención y modificación de los datos .....	40
5.2.3	Administración de los usuarios .....	48
5.2.4	Menú principal de usuario .....	48
5.2.5	Historial de actividades .....	48
5.2.6	Detalles de la actividad .....	49
5.2.7	Guardar la actividad en un fichero .....	49
5.2.8	Análisis de los datos .....	50
Capítulo 6.	Resultados obtenidos .....	60
6.1	Aplicación móvil .....	60



6.1.1	Inicio .....	60
6.1.2	Administración de usuarios .....	61
6.1.3	Menú principal.....	62
6.1.4	Selección del dispositivo .....	63
6.1.5	Registro de una nueva actividad.....	64
6.1.6	Detalles de la actividad.....	65
6.1.7	Guardar la actividad en un fichero .....	66
6.1.8	Historial de actividades .....	67
6.2	Análisis de los datos .....	68
Capítulo 7.	Conclusiones y propuesta de trabajo futuro .....	73
7.1	Conclusiones.....	73
7.2	Propuesta de trabajo futuro.....	74
Capítulo 8.	Bibliografía.....	76

## Índice de figuras

Figura 1. Millones de dispositivos wearables vendidos a nivel mundial. (Statista).....	9
Figura 2. Vista del dispositivo SensorTag CC2650 de Texas Instruments. (SimpleLink™ CC2650 SensorTag™) .....	12
Figura 3. Vista del dispositivo SensorTag CC3200 de Texas Instruments. (SimpleLink™ Wi-Fi® CC3200 SensorTag).....	13
Figura 4. Vista del dispositivo STEVAL-WESU1 de STMicroelectronics. (STEVAL- WESU1).....	14
Figura 5. Vista de una placa Arduino junto con el módulo MPU6050 (Conexión entre Arduino y MPU-6050).....	15
Figura 6. Diagrama de bloques general. ....	18
Figura 7. Diagrama de bloques del STEVAL-WESU1. (STEVAL-WESU1 - How to use).....	20
Figura 8. Componentes del STEVAL-WESU1 (STEVAL-WESU1 - How to use).....	20
Figura 9. Cuota de mercado de pedidos de smartphones a nivel mundial por sistema operativo. (Statista).....	23
Figura 10. Diagrama de bloques del software de la aplicación móvil.....	24
Figura 11. Estructura servidor GATT.....	26
Figura 12. Representación de los ángulos de Euler. (Ángulos de Euler) .....	27
Figura 13. Representación de los ángulos de Tait-Bryan. (Ángulos de Tait-Bryan) ....	28
Figura 14. Representación del principio del <i>gimbal lock</i> . (Gimbal lock).....	30
Figura 15. Representación del cabeceo, alabeo y guiñada en una aeronave. (Sociedad Aeronáutica Española).....	31
Figura 16. Etapas del filtro de Kalman. (Filtro de Kalman).....	33



Figura 17. Diagrama de bloques del algoritmo de Madgwick para una implementación MARG, incluyendo la distorsión magnética (Grupo 1) y la deriva compensada del giroscopio (Grupo 2). (Madgwick).....	34
Figura 18. Parte del manifiesto correspondiente al BLE.....	36
Figura 19. Configuración del adaptador y manejador. ....	36
Figura 20. Configuración del adaptador y manejador. ....	37
Figura 21. Búsqueda de dispositivos BLE.....	38
Figura 22. Manejar el resultado de la búsqueda de dispositivos. ....	39
Figura 23. Método para conectarse a un dispositivo. ....	39
Figura 24. Identificadores UUID.....	40
Figura 25. Método para habilitar los sensores configurados. ....	41
Figura 26. Método para configurar los sensores.....	42
Figura 27. Método para obtener los datos de la característica.....	42
Figura 28. Formato de los datos transmitidos. (STEVAL-WESU1 - How to use) .....	43
Figura 29. Función que permite la conversión de las unidades de los datos (1). ....	43
Figura 30. Función que permite la conversión de las unidades de los datos (2). ....	44
Figura 31. Relación entre las diferentes tablas que forman la base de datos. ....	45
Figura 32. Primera parte del código que define las acciones a realizar al crearse la vista .....	46
Figura 33. Segunda parte del código que define las acciones a realizar al crearse la vista. ....	47
Figura 34. Primera parte del código que define las acciones a realizar al crearse la vista. ....	47





Figura 35. Permiso de escritura en un almacenamiento externo del manifiesto de la aplicación.....	49
Figura 36. Ejemplo de un fichero CSS con datos que se van a analizar. ....	50
Figura 37. Código empleado para la lectura del fichero y separación en matrices. ....	51
Figura 38. Código empleado para eliminar el Gimbal Lock. ....	52
Figura 39. Código empleado para compensar la gravedad y poner a 0 g la aceleración de referencia en cada uno de los ejes. ....	53
Figura 40. Código empleado para representación de los datos. ....	54
Figura 41. Método para obtener los pasos mediante un comparador con histéresis con las medidas del giróscopo.....	56
Figura 42. Método para calcular el número de pasos con las medidas del acelerómetro. ....	57
Figura 43. Método para obtener la aceleración máxima de cada paso. ....	58
Figura 44. Método para obtener la matriz con el instante de tiempo y posición de cada aceleración máxima. ....	59
Figura 45. Pantalla de inicio de la aplicación. ....	60
Figura 46. Pantalla de selección de usuarios. ....	61
Figura 47. Menú para añadir un nuevo usuario. ....	62
Figura 48. Menú para editar un usuario.....	62
Figura 49. Menú para añadir un nuevo usuario. ....	63
Figura 50. Buscando dispositivos.....	64
Figura 51. No se han encontrado dispositivos.....	64
Figura 52. Dispositivo encontrado.....	64
Figura 53. Antes de empezar la actividad.....	65



Figura 54. En medio de la actividad. ....	65
Figura 55. Detalles de una actividad. ....	66
Figura 56. Exportar los datos a un fichero.....	67
Figura 57. Historial de actividades. ....	68
Figura 58. Posición y orientación del dispositivo.....	69
Figura 59. Medidas recogidas por los sensores en la actividad realizada corriendo. ....	70
Figura 60. Medidas recogidas por los sensores en la actividad realizada andando.....	70
Figura 61. Máxima aceleración de cada paso realizado corriendo.....	71
Figura 62. Máxima aceleración de cada paso realizado andando.....	71
Figura 63. Matriz con los datos obtenidos de cada paso realizado en la actividad corriendo.....	72
Figura 64. Matriz con los datos obtenidos de cada paso realizado en la actividad andando.....	72



## Capítulo 1. Introducción

En este capítulo se describen los motivos y los objetivos por los cuales se ha llevado a cabo este proyecto y la estructura de esta memoria.

### 1.1 Motivación

A escasos meses de finalizar la segunda década del siglo XXI, el *hobbie* favorito de la mayor parte de la sociedad es la práctica de algún deporte. Por otra parte, el desarrollo tecnológico crece muy rápidamente, por lo que no era de extrañar que llegado a cierto punto ambos conceptos pudieran unirse.

Hoy en día, la gran mayoría de deportes hacen uso de múltiples herramientas para perfeccionar la precisión de los resultados o para ayudar al usuario a mejorar, entre otros muchos propósitos.

Una de las herramientas con mayor repercusión en los últimos años es el uso de dispositivos *wearables*, capaces de recoger datos precisos del entrenamiento realizado para posteriormente analizarlos y observar donde se puede mejorar.

Tal es la repercusión de dichos dispositivos que grandes marcas de tecnología han creado su propia línea de productos *wearables*, como se observa en la Figura 1.

Cabe destacar que los últimos meses debido a la situación socio sanitaria en la que se encuentra el país (a causa del virus COVID-19), donde muchas zonas tienen restringido el número máximo de personas en reuniones, prohibido practicar deportes de contacto y/o practicar deportes colectivos; la práctica de deportes individuales ha crecido drásticamente. (Las Provincias)

Por ambos motivos, el proyecto está centrado en un deporte individual donde usuario pueda darse cuenta de sus errores y poder mejorar en próximos entrenamientos.

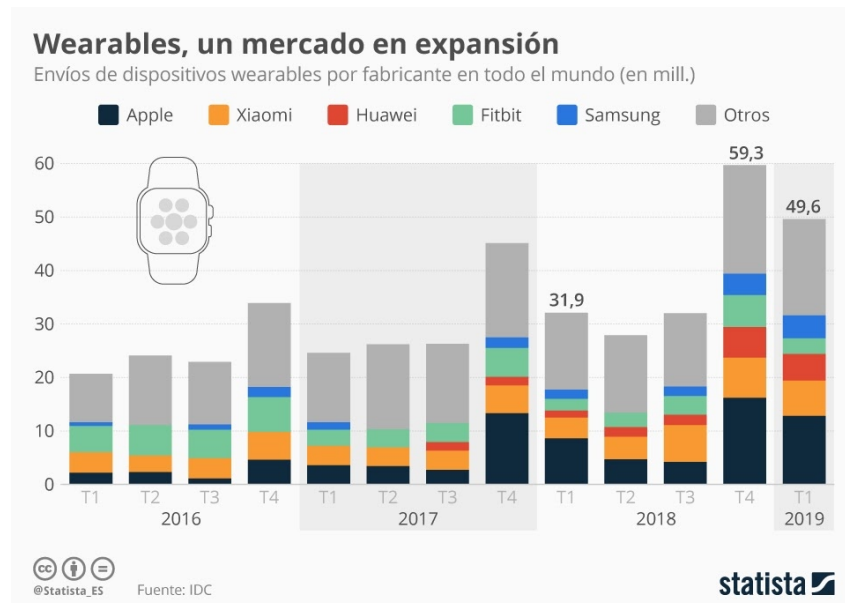


Figura 1. Millones de dispositivos wearables vendidos a nivel mundial. (Statista)

## 1.2 Objetivos

En primer lugar, el objetivo previo al diseño de la aplicación es encontrar el dispositivo *wearable* que más se ajuste a las necesidades del proyecto. Es decir, un dispositivo ligero que pueda ser colocado en el tobillo, que incorpore los sensores necesarios para obtener los datos lo más precisos posibles.

El segundo objetivo es comprender el protocolo de comunicación que permite enlazar y enviar la información captada por el *wearable* al dispositivo móvil.

El tercer objetivo consiste en el desarrollo de la aplicación móvil, esta debe tener todas las funcionalidades planteadas: registrar, representar y almacenar los datos de la actividad realizada. Además de poder descargar dichos datos en un fichero para su exportación a otros dispositivos.

Como cuarto y último objetivo, analizar los datos en un ordenador para obtener el número de pasos realizados, la aceleración máxima y la posición del tobillo en cada uno de ellos.



### **1.3 Estructura de la memoria**

La siguiente memoria se encuentra dividida en 8 capítulos, los cuales a su vez se dividen en secciones y subsecciones.

En el Capítulo 2 se exponen y analizan las diferentes tecnologías wearables disponibles en el mercado y en el Capítulo 3 se encuentra detallada la solución adoptada para llevar a cabo el proyecto.

En el Capítulo 4 se describen los fundamentos teóricos necesarios para el correcto desarrollo de este. En el Capítulo 5 la implementación hardware y software del prototipo y en el Capítulo 6 se exponen los resultados obtenidos tras el análisis de los datos.

Finalmente, en el Capítulo 7 se encuentran las conclusiones y seguidamente, en el Capítulo 8 la bibliografía empleada.



## Capítulo 2. Tecnología disponible

### 2.1 Dispositivos *wearables*

La tecnología *wearable* se basa en el uso de dispositivos electrónicos como accesorios del cuerpo que interactúan con el usuario y otros dispositivos. (Dispositivos *wearables*)

En este punto de la memoria, se analizarán los dispositivos *wearables* disponibles en el mercado que pueden ser de gran utilidad para este proyecto.

#### 2.1.1 *SensorTag CC2650*

Este dispositivo SimpleLink™ SensorTag CC2650 de la empresa Texas Instruments incluye diversos sensores de baja potencia capaces de obtener diferentes magnitudes como la aceleración, la temperatura, la humedad, el magnetismo y la presión. (Texas Instruments)

Emplea un microprocesador con la siguiente tecnología:

- Powerful ARM® Cortex®-M3
- Velocidad de reloj de hasta 48 MHz
- Memoria Flash de 128 KB

Este modelo de dispositivo tiene integrado una unidad de procesamiento de movimiento MPU-9250 fabricado por InvenSense, la cual incorpora los siguientes sensores: (InvenSense)

- Acelerómetro
  - Rango:  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  y  $\pm 16g$
  - Frecuencia de muestreo de hasta 4 kHz

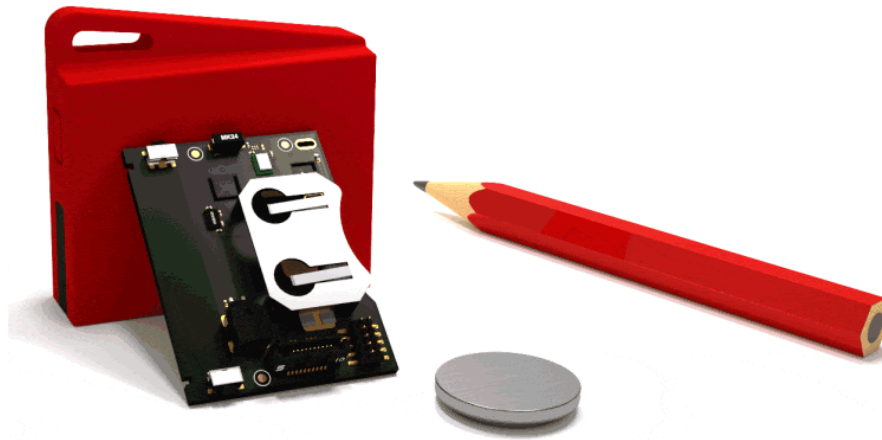


Figura 2. Vista del dispositivo SensorTag CC2650 de Texas Instruments. (SimpleLink™ CC2650 SensorTag™)

- Giróscopo
  - Rango:  $\pm 250^\circ/\text{s}$ ,  $\pm 500^\circ/\text{s}$ ,  $\pm 1000^\circ/\text{s}$  y  $\pm 2000^\circ/\text{s}$
  - Frecuencia de muestreo de hasta 1 kHz
  
- Magnetómetro
  - Rango:  $\pm 4800\text{uT}$
  - Frecuencia de muestreo de hasta 8 kHz

Este dispositivo tiene un consumo muy bajo y utiliza *Bluetooth Low Energy* para la comunicación con otros dispositivos (o su versión *iBeacon* para dispositivos de *Apple*). También es posible establecer comunicación mediante la tecnología *ZigBee®* o *6LoWpan*.

Otro aspecto para tener en cuenta de este dispositivo es que el fabricante nos ofrece mediante un *DevPack* la posibilidad de modificar el firmware, a la vez de poder incorporar sensores adicionales al dispositivo.

### 2.1.2 *SensorTag CC3200*

El dispositivo SimpleLink™ Wi-Fi® SensorTag CC2650 de la empresa Texas Instruments, comparte gran parte de las características y especificaciones con el modelo descrito en la Sección 2.1.1.

Una de las diferencias más significativas, es a la hora de la comunicación entre diferentes dispositivos. En vez de incorporar la tecnología *Bluetooth Low Energy*, este modelo realiza la comunicación mediante Wi-Fi®, por lo que permite conectar los sensores a una nube y poder tener los datos en cualquier lugar en poco tiempo.

En cuanto al microprocesador las especificaciones son prácticamente las mismas que las del modelo anterior. Sin embargo, este modelo utiliza un controlador ARM® Cortex®-M4 de 32 bits.

Respecto a los sensores de movimiento, sigue teniendo los mismos; incorporando una unidad inercial, en concreto el módulo MPU-9250 descrito en la Sección 2.1.1.



Figura 3. Vista del dispositivo SensorTag CC3200 de Texas Instruments. (SimpleLink™ Wi-Fi® CC3200 SensorTag)



### 2.1.3 STEVAL-WESU1

El dispositivo STEVAL-WESU1 de la empresa STMicroelectronics, incorpora diferentes sensores de movimiento de baja potencia capaces de medir la aceleración, magnetismo, giro y presión. (STMicroelectronics)



Figura 4. Vista del dispositivo STEVAL-WESU1 de STMicroelectronics. (STEVAL-WESU1)

La comunicación con dispositivos externos se realiza mediante *Bluetooth Low Energy*, al igual que el dispositivo descrito en la Sección 2.1.1. Sin embargo, utiliza un procesador BLE diseñado por la propia empresa llamado *BlueNRS-MS*.

En cuanto al microprocesador, STMicroelectronics utiliza su propio diseño STM32L151VEY6 con las características siguientes: (STM32L151VE)

- Acelerómetro
  - Rango:  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  y  $\pm 16g$
  - Frecuencia de muestreo de hasta 1.6 kHz
  
- Giróscopo
  - Rango:  $\pm 125^\circ/s$ ,  $\pm 245^\circ/s$ ,  $\pm 500^\circ/s$ ,  $\pm 1000^\circ/s$  y  $\pm 2000^\circ/s$

- Frecuencia de muestreo de hasta 1.6 kHz
- Magnetómetro
  - Rango:  $\pm 4$ gauss,  $\pm 8$ gauss,  $\pm 12$ gauss y  $\pm 16$ gauss
  - Frecuencia de muestreo de hasta 1 kHz

### 2.1.4 Otras alternativas

Aparte de los dispositivos descritos en los puntos anteriores, también es posible combinar diferentes productos para conseguir lo que se exige, un ejemplo perfecto es Arduino.

Esta alternativa consiste en la combinación de diferentes placas y módulos capaces de desarrollar casi cualquier aplicación. (Arduino)

Para este proyecto en específico, es necesario la combinación de dos módulos: el módulo MPU6050 o similar que incorpora un acelerómetro y un giróscopo; y un módulo Bluetooth para la conexión con el dispositivo móvil. (MPU6050)

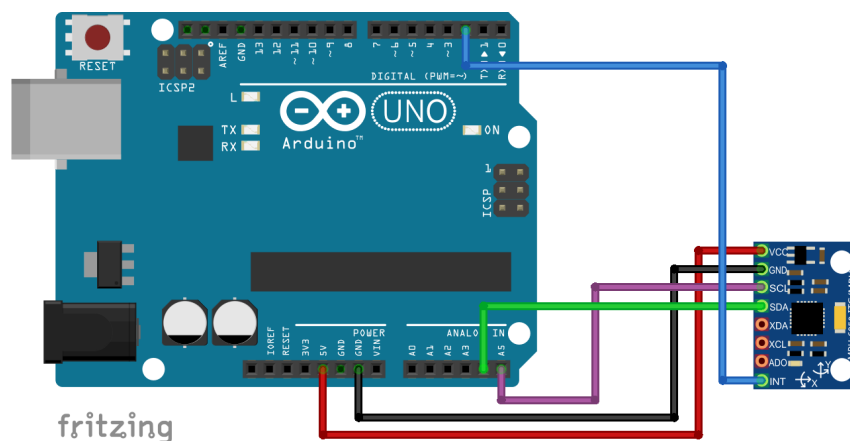


Figura 5. Vista de una placa Arduino junto con el módulo MPU6050 (Conexión entre Arduino y MPU-6050)



## 2.2 Sistemas Operativos

### 2.2.1 Android

Se trata del sistema operativo más utilizado en los *smartphones* durante los últimos años. Fue creado por *Android Inc.*, sin embargo, actualmente es propiedad de Google. (Android, 2020) (Google, 2020)

Las principales características de este dispositivo se encuentran a continuación:

- Desarrollo libre y código abierto
- Permite programación en Java, C, C++
- Núcleo basado en el Kernel de Linux
- Extenso catálogo de aplicaciones a través de Google Play
- Gran cantidad de formatos multimedia soportados
- Soporte de HTML, HTML5 y Adobe Flash Player
- Multitarea de aplicaciones
- Entorno de desarrollo oficial: Android Studio

### 2.2.2 iOS

Este sistema operativo creado por la empresa *Apple Inc.* es utilizado por todos sus *smartphones*. Sus características más importantes son las siguientes (Apple Inc., 2020; Arduino, 2020; Arduino, 2020; Arduino, 2020) (iOS):

- Código único y exclusivo de Apple Inc.
- Núcleo basado en XNU
- Multitarea de aplicaciones
- Soporte de HTML y HTML5
- Catálogo de aplicaciones mediante App Store



- Navegador web propio: Safari

## Capítulo 3. Descripción de la solución adoptada

En este tercer capítulo se presenta la solución adoptada y se encuentra dividido en tres partes: el diagrama de bloques general, la descripción detallada del hardware y posteriormente del software utilizado.

### 3.1 Diagrama de bloques general

En la Figura 6 que se encuentra a continuación se visualiza el diagrama de bloques. El dispositivo *wearable* seleccionado que recoge los pertinentes datos de los sensores y la conexión vía *Bluetooth Low Energy* entre el dispositivo y el *smartphone*.



Figura 6. Diagrama de bloques general.

### 3.2 Hardware

#### 3.2.1 Razones de elección

Entre las múltiples opciones de dispositivos *wearables* descritas en el Capítulo 2, se ha escogido el dispositivo STEVAL-WESU1 de la empresa STMicroelectronics. Uno de los principales motivos por el que este dispositivo ha sido elegido, es debido a sus dimensiones y peso. Esto es debido a que, teniendo las especificaciones internas bastante similares a la competencia, la diferencia en cuanto a las características físicas es bastante



grande. El peso del dispositivo es muy importante para el usuario, ya que un exceso de peso en uno de los tobillos puede llegar a modificar la manera de pisar para equilibrar el cuerpo.

En cuanto al segundo motivo, el rango de frecuencias de muestreo de este dispositivo es más elevado que en otros dispositivos. Por lo que puede ser utilizado para la monitorización de casi cualquier actividad física. A su vez, posee mayor cantidad de memoria que cualquiera de los dispositivos descritos.

En tercer lugar, este dispositivo incorpora en el hardware la posibilidad de comunicación mediante la tecnología *Bluetooth Low Energy*. Esta se ha utilizado en el proyecto debido a que el consumo energético es escaso. El único inconveniente que tiene es el rango de distancia de comunicación entre el dispositivo y el *smartphone*; sin embargo, en este caso ambos se encuentran a una corta distancia.

Finalmente, cabe destacar la extensa documentación del dispositivo, tanto de su funcionamiento como de los protocolos que utiliza. Toda la documentación necesaria del dispositivo se puede encontrar en la página web oficial de STMicroelectronics. También se encuentra detallada toda la información necesaria para actualizar y modificar el firmware. A su favor, hay que añadir que este dispositivo permite modificar casi cualquier configuración de los sensores, por lo que no es necesario modificar el código interno.

### 3.2.2 Diagrama de bloques

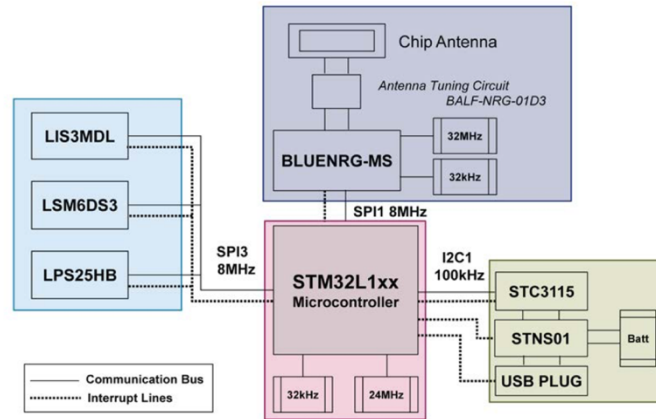


Figura 7. Diagrama de bloques del STEVAL-WESU1. (STEVAL-WESU1 - How to use)

### 3.2.3 Detalles técnicos

A continuación, se encuentran descritos los componentes más importantes del dispositivo *wearable* y señalizados en la Figura 8:

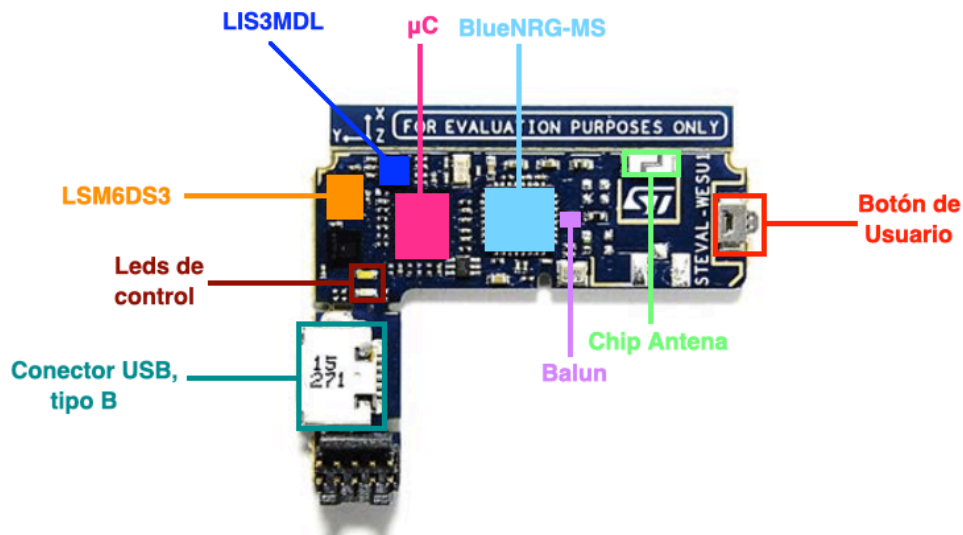


Figura 8. Componentes del STEVAL-WESU1 (STEVAL-WESU1 - How to use)



### 3.2.3.1 *Microcontrolador*

- Microcontrolador ARM® Cortex®-M3 de 32 bits
- Memoria Flash de 512 KB
- Memoria RAM de 48 KB
- Diseñado para aplicaciones de escasa potencia
- Bajo consumo de corriente

### 3.2.3.2 *Sensores*

El módulo LSM6DS3 incorpora el acelerómetro y el giróscopo y sus características son las siguientes: (LSM6DS3, 2017)

- Consumo de corriente 1.25 mA
- Hasta 1.6 kHz ODR (Output Data Rate)
- Rangos de escala del acelerómetro:  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  y  $\pm 16g$
- Rangos de escala del giróscopo:  $\pm 125^\circ/s$ ,  $\pm 245^\circ/s$ ,  $\pm 500^\circ/s$ ,  $\pm 1000^\circ/s$  y  $\pm 2000^\circ/s$

El módulo LIS3MDL incorpora el magnetómetro y sus características son las siguientes: (LIS3MDL)

- Hasta 1 kHz ODR (*Output Data Rate*)
- Resolución de 16 bits
- Rangos de escala del magnetómetro:  $\pm 4\text{gauss}$ ,  $\pm 8\text{gauss}$ ,  $\pm 12\text{gauss}$  y  $\pm 16\text{gauss}$





### 3.2.3.3 Comunicación Bluetooth Low Energy

El módulo de conexión Bluetooth se encuentra dividido entre el procesador de baja potencia Bluetooth Low Energy (*BLUENRG-MS*) y su par conjugado (*BALF-NRG-01D3*) cuya finalidad es filtrar armónicos. (*BLUENRG-MS*) (*BALF-NRG-01D3*)

#### 3.2.3.3.1 BLUENRG-MS

- Núcleo central basado en ARM Cortex-M0
- Cumple con las especificaciones del estándar Bluetooth v.4.1
- Es capaz de actuar tanto de maestro como de esclavo

#### 3.2.3.3.2 BALF-NRG-01D3

- Par conjugado de 50 Ohms del BLUENRG-MS
- Incorpora un transformador balun (balanceado a desbalanceado)
- Filtra los armónicos

### 3.2.3.4 Batería y gestión de la energía

La gestión de la batería se encuentra dividida en 4 subsistemas:

- USB STNS01: se trata del cargador de la batería el cual permite su carga.
- Módulo STC3115: proporciona información acerca del nivel de batería.
- Conector de la batería
- Conector USB, el cual acepta USB tipo B.

Señalar que el módulo STNS01 es un cargador lineal para baterías de litio de una celda. En este dispositivo se encuentra configurado como cargador de batería, además de funcionar como conmutador de la ruta de alimentación USB y la fuente de alimentación de la batería. (STNS01)

### 3.3 Software

#### 3.3.1 Razones de elección

Tras analizar las dos opciones de software más populares del mercado, el sistema operativo escogido para desarrollar la aplicación móvil ha sido *Android*.

La primera razón de porque ha sido escogido es básicamente por lo popular que es entre los *smartphones* de los últimos años como vemos en la Figura 9.

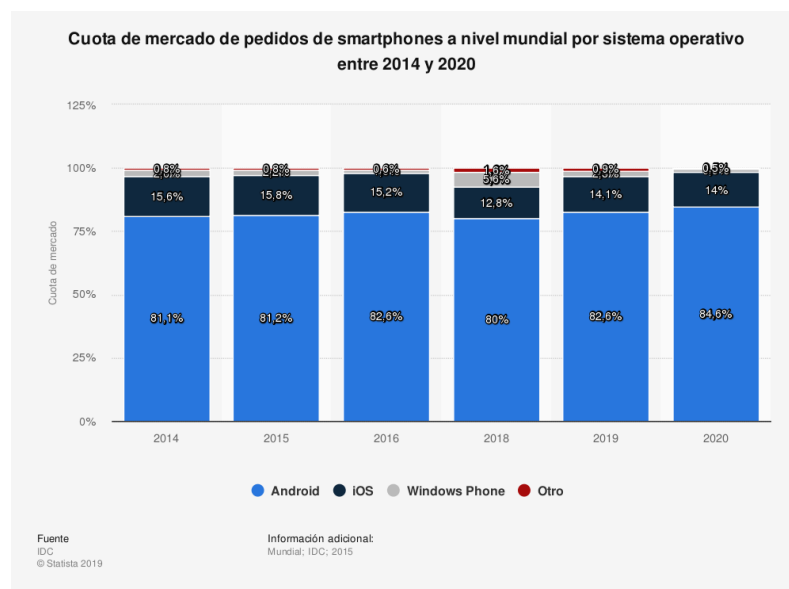


Figura 9. Cuota de mercado de pedidos de smartphones a nivel mundial por sistema operativo. (Statista)

En segundo lugar, como se ha mencionado anteriormente, este sistema operativo es de código abierto, por lo que tanto en Internet como físicamente se puede encontrar gran cantidad de información para el desarrollo de cualquier aplicación móvil.

En cuanto al desarrollo de la aplicación, el entorno escogido ha sido *Android Studio*, el oficial de *Android*. Éste puede ser descargado de forma gratuita desde su página web oficial para las principales plataformas (Windows, MacOs o Linux). (Android Studio)

### 3.3.2 Diagrama de bloques

A continuación, se muestra el diagrama de bloques correspondiente con la aplicación creada que se ejecutará en el *smartphone*.

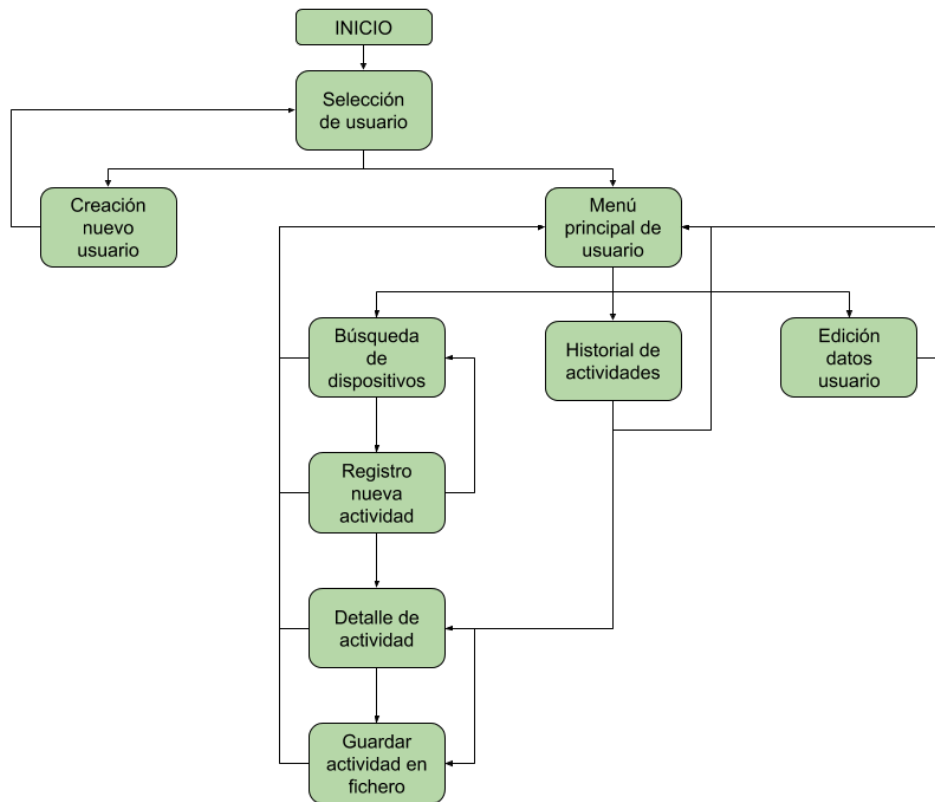


Figura 10. Diagrama de bloques del software de la aplicación móvil

## Capítulo 4. Fundamentos teóricos

En este cuarto capítulo se describen los conceptos teóricos indispensables para el desarrollo y entendimiento del proyecto. Este capítulo se encuentra dividido en dos partes: el protocolo de comunicación entre el dispositivo y el *smartphone*, y conceptos esenciales sobre la orientación de los objetos en el espacio.

### 4.1 Protocolo Bluetooth Low Energy

La comunicación BLE (Bluetooth Low Energy) está basada en una especificación llamada GATT (General ATtribute profile). Ésta detalla cómo se envía y recibe la información entre dispositivos (uno ejerce de servidor mientras que el otro como cliente). (Bluetooth Generic Attributes)

El GATT se crea sobre el protocolo de atributo ATT (ATtribute protocol), el cual se encuentra optimizado para la utilización en dispositivos BLE. Cada atributo se identifica mediante un identificador único universal (UUID) compuesto por 128 bits. Los atributos que se transfieren por ATT tienen el siguiente formato: servicio, característica y descriptor.

Un atributo puede tener más de un servicio, y a su vez un servicio puede tener más de una característica. Estas características pueden tener o no un descriptor.

En la Figura 11 se encuentra detallado mediante un diagrama la estructura de un servidor GATT.

### 4.2 Representación de la orientación

Para representar un cuerpo tridimensional en el espacio existen múltiples opciones. Para este proyecto se han utilizado los ángulos de Euler y los cuaterniones ya que son perfectos para el objetivo que se requiere, conocer la posición exacta de un cuerpo en el espacio. Ambos sistemas de representación se encuentran explicados a continuación:

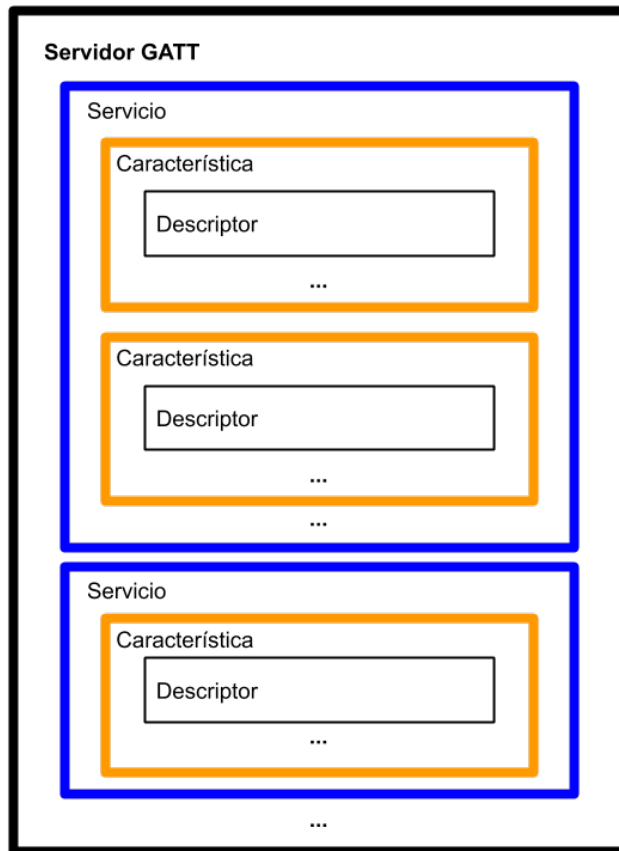


Figura 11. Estructura servidor GATT.

#### 4.2.1 *Ángulos de Euler*

Los ángulos de Euler constituyen un conjunto de tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia de ejes ortogonales, normalmente móvil, respecto a otro sistema de referencia de ejes ortogonales normalmente fijos. (Ángulos de Euler)

Dados dos sistemas de coordenadas  $xyz$  y  $XYZ$  con origen común, es posible especificar la posición de un sistema en términos del otro usando tres ángulos  $\alpha$ ,  $\beta$  y  $\gamma$ .

La definición matemática es estática y se basa en escoger dos planos, uno en el sistema de referencia y otro en el triedro rotado.

La intersección de los planos coordenados  $xy$  y  $XY$  escogidos se llama línea de nodos, y se usa para definir los tres ángulos:

- $\alpha$  es el ángulo entre el eje  $x$  y la línea de nodos.
- $\beta$  es el ángulo entre el eje  $z$  y el eje  $Z$ .
- $\gamma$  es el ángulo entre la línea de nodos y el eje  $X$ .

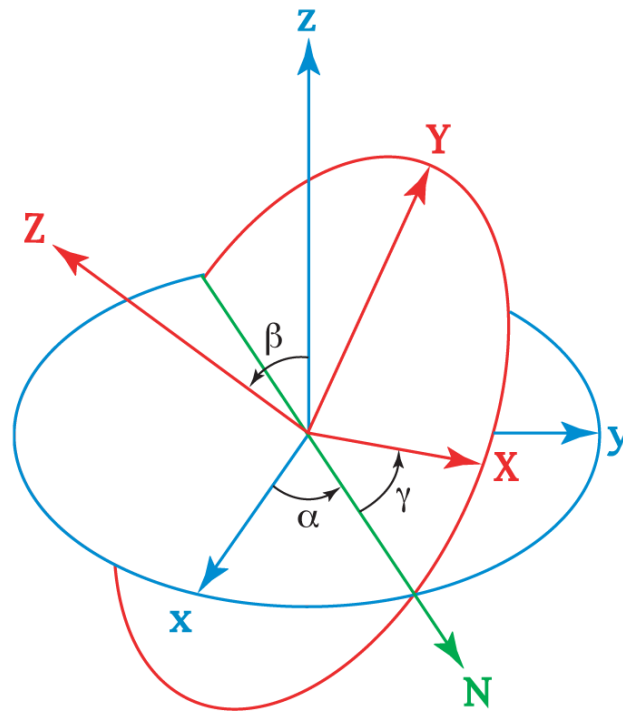


Figura 12. Representación de los ángulos de Euler. (Ángulos de Euler)

Escogiendo otros planos de referencia se obtendrían distintas convenciones alternativas, las cuales se llaman de Tait-Bryan cuando los planos de referencia son no-homogéneos (por ejemplo,  $xy$  y  $XY$  son homogéneos, mientras  $xy$  y  $XZ$  no lo son). (Ángulos de Tait-Bryan)

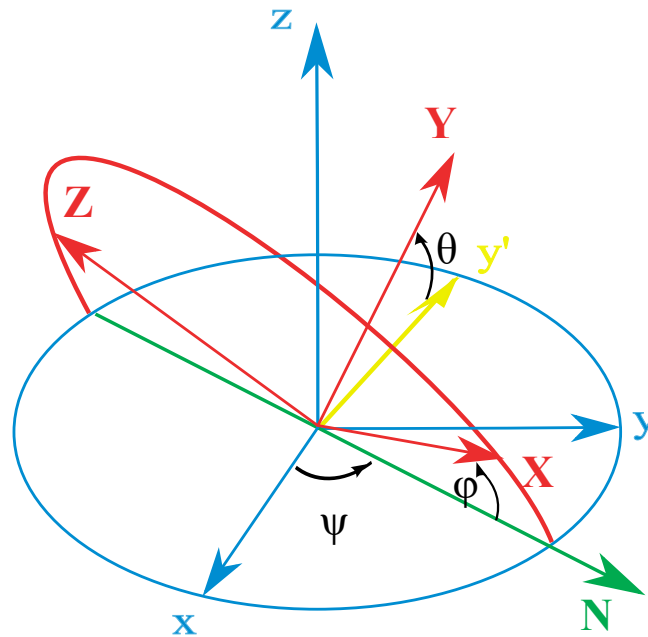


Figura 13. Representación de los ángulos de Tait-Bryan. (Ángulos de Tait-Bryan)

Este sistema de representación conlleva un problema que debe ser solucionado, el *gimbal lock*. Que se encuentra explicado en la Sección 4.2.3.

#### 4.2.2 Cuaterniones

Los cuaterniones son una extensión de los números reales, similar a la de los números complejos. Mientras que los números complejos son una extensión de los reales por la adición de la unidad imaginaria  $i$ , tal que  $i^2 = -1$ , los cuaterniones son una extensión generada de manera análoga añadiendo las unidades imaginarias  $i, j$  y  $k$  a los números reales tal que: (Cuaternion)

$$\hat{q} = q_0 + q_1i + q_2j + q_3k \quad (1)$$

$$i^2 = j^2 = k^2 = ijk = -1 \quad (2)$$

Generalmente su representación es la siguiente:

$$q = a + bi + cj + dk \quad (3)$$

Una peculiaridad de los cuaterniones es que su producto no es conmutativo. Dicho sistema de representación es utilizado frecuentemente en matemática teórica y aplicada, para describir las rotaciones y orientaciones de un cuerpo en el espacio. En áreas como la robótica, la navegación o gráficos por ordenador su uso es más que evidente.

La representación en cuaterniones tiene ciertas ventajas frente otros sistemas de representación. En comparación con las matrices de rotación, los cuaterniones no necesitan tanto tiempo para sus cálculos, por lo que, a la hora de seguir el desplazamiento de un objeto es un punto para tener en cuenta. La ventaja que tienen frente a los ángulos de Euler es la inexistencia del *gimbal lock*.

### 4.2.3 *Gimbal Lock*

El *gimbal lock* consiste en la pérdida de un grado de libertad en una suspensión cardán de tres rotores, que ocurre cuando los ejes de dos de los tres rotores se colocan en paralelo, bloqueando el sistema en una rotación en un espacio bidimensional degenerado. (Bloqueo del cardán)

Existen diferentes soluciones a este problema, una posible solución es rotar uno o más ejes a una posición arbitraria en el momento que se halle el *gimbal lock*, con ello restableceremos el dispositivo. La solución más utilizada es evitar la representación tridimensional con ángulos de Euler, y usar la representación en cuaterniones.



En este proyecto se han utilizado los ángulos de Euler, en concreto, la convención de ángulos de Tait-Bryan. En la sección 5.2.8.3 se encuentra explicada con detalle la solución implementada.

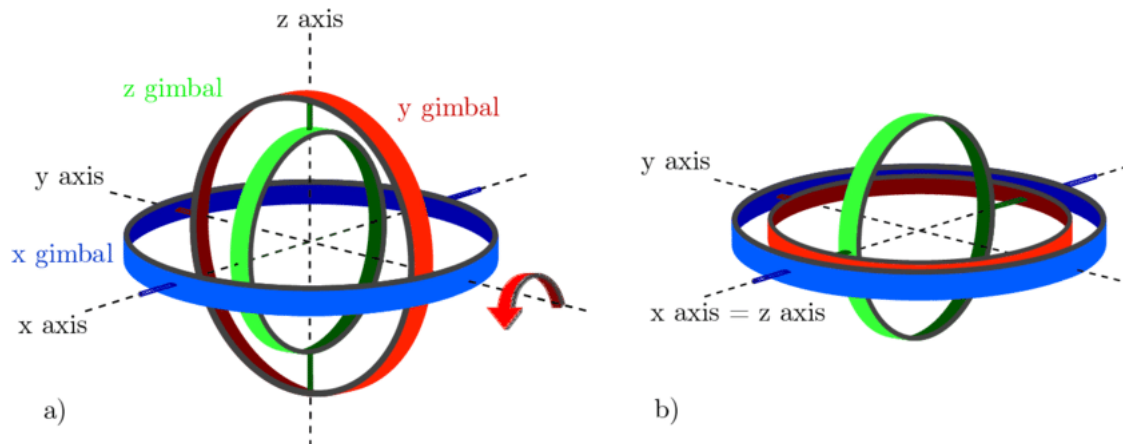


Figura 14. Representación del principio del *gimbal lock*. (Gimbal lock)

### 4.3 Medida de orientación

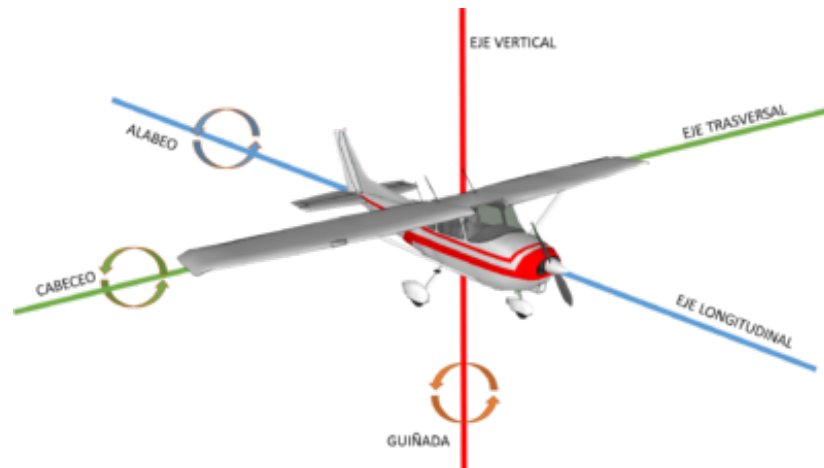
Para poder determinar la orientación de un cuerpo en el espacio es necesario combinar diferentes sensores. Para ello existen mecanismos capaces de obtenerla: las Unidades de Medición Inercial (IMU) y los Sistemas de Referencia de Actitud y Rumbo (AHRS).

#### 4.3.1 Unidad de Medición Inercial

Una unidad de medición inercial o IMU, es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giróscopos. (IMU)

Son normalmente utilizadas para maniobrar aviones, incluyendo vehículos aéreos no tripulados entre muchos otros usos, y además naves espaciales, incluyendo transbordadores, satélites y aterrizadores.

Funciona detectando la actual tasa de aceleración usando uno o más acelerómetros, y detecta los cambios en atributos rotacionales tales como cabeceo, alabeo y guiñada usando uno o más giróscopos.



**Figura 15. Representación del cabeceo, alabeo y guiñada en una aeronave. (Sociedad Aeronáutica Española)**

Una de las grandes desventajas de usar IMU para la navegación es que estas normalmente son afectadas por un error acumulativo. Debido a que el sistema de guía está continuamente agregando los cambios detectados a las posiciones previamente calculadas, cualquier error en la medición, no importando lo pequeña que sea, se van acumulando de punto a punto. Esto lleva a una “deriva”, o a una diferencia que aumenta siempre entre donde el sistema piensa que se encuentra localizado y la posición real.

#### **4.3.2 *Sistemas de Referencia de Actitud y Rumbo (AHRS)***

Se trata de sensores tridimensionales que proporcionan información acerca del rumbo, la actitud, y la guiñada de una aeronave. En ocasiones también son llamados sensores MARG y están formados por giróscopos, acelerómetros, y magnetómetros, que



proporcionan datos en los tres ejes del espacio. (Sistemas de referencia de Actitud y Rumbo)

Este sistema soluciona el problema del error acumulativo de IMU gracias al uso de un magnetómetro, con ello se reduce significativamente el error existente entre la orientación estimada y la real del dispositivo.

En AHRS se utilizan normalmente filtros de Kalman como método de estimación no lineal, para obtener una única solución a partir de las diferentes medidas que realizan los sensores.

#### **4.3.3 Filtro de Kalman**

El filtro de Kalman es un algoritmo desarrollado por Rudolf Emil Kalman en 1960 con el objetivo de filtrar y definir sistemas lineales. Se trata de un algoritmo recursivo que estima mediante mínimos cuadrados para procesar enormes cantidades de datos, sin importar la precisión de los datos de entrada. Con ello consigue calcular un estimador que proporcione valores actuales de las variables, con datos del tiempo actual y de tiempos anteriores. Para el correcto funcionamiento del filtro es necesario: (Filtro de Kalman, 2020)

- Conocimiento del sistema y de los instrumentos de medida
- Datos estadísticos de los instrumentos de medida (error de medición)
- Información disponible de las variables de interés (condiciones iniciales)

Este filtro de Kalman trabaja sobre medidas discretas en vez de tiempo continuo. Se trata de un algoritmo recursivo, por lo que no requiere el almacenamiento de todos los valores anteriores para volver a procesar.

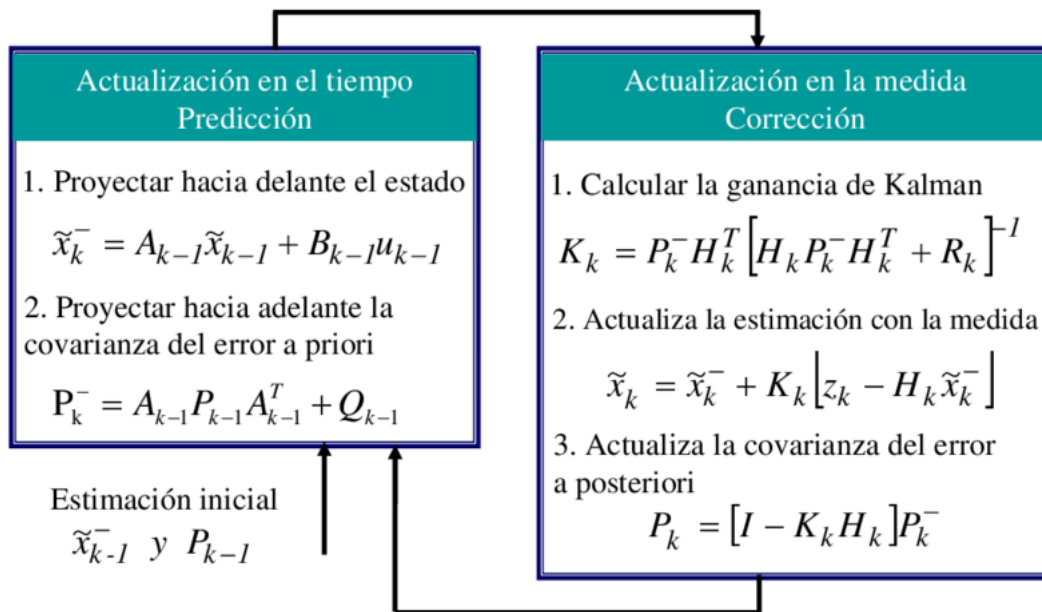


Figura 16. Etapas del filtro de Kalman. (Filtro de Kalman)

Este filtro ha sido una herramienta muy útil a la hora de solucionar problemas relacionados con el control y la estimación, con los que ha aportado gran precisión y eficiencia. Su principal uso ha sido en los sistemas de control modernos, en el seguimiento y navegación de todo tipo de vehículos y en el diseño predictivo de estimación de estos.

#### 4.3.4 Algoritmo de Madgwick

El algoritmo Madgwick es un filtro diseñado por O.H. Madgwick basándose en el filtro de Kalman. Puede ser aplicado tanto a una IMU como a un MARG. Utiliza los cuaterniones como sistema de representación para evitar el error del *gimbal lock* al utilizar los ángulos de Euler.

Para el desarrollo del proyecto se ha escogido utilizar este algoritmo en su versión MARG, ya que presenta diferentes mejoras en comparación con los derivados del filtro Kalman.

Una gran ventaja que tiene frente a otros algoritmos es su bajo coste computacional, tal como detalla el autor en su informe, únicamente requiere 109 operaciones aritméticas en el caso de los IMUs y 277 en el caso de los MARGs para cada actualización del filtro. La segunda ventaja es el buen rendimiento que ofrece a relativamente bajas frecuencias de muestreo, como es el caso de nuestro proyecto. En cuanto a la tercera, es posible ajustar uno o dos parámetros (uno en IMU y dos en MARG) como la ganancia  $\beta$ , permitiendo obtener resultados más precisos en función de las características del sistema. Finalmente permite implementar algoritmos para compensar la deriva del giroscopio y la distorsión magnética. (Madgwick)

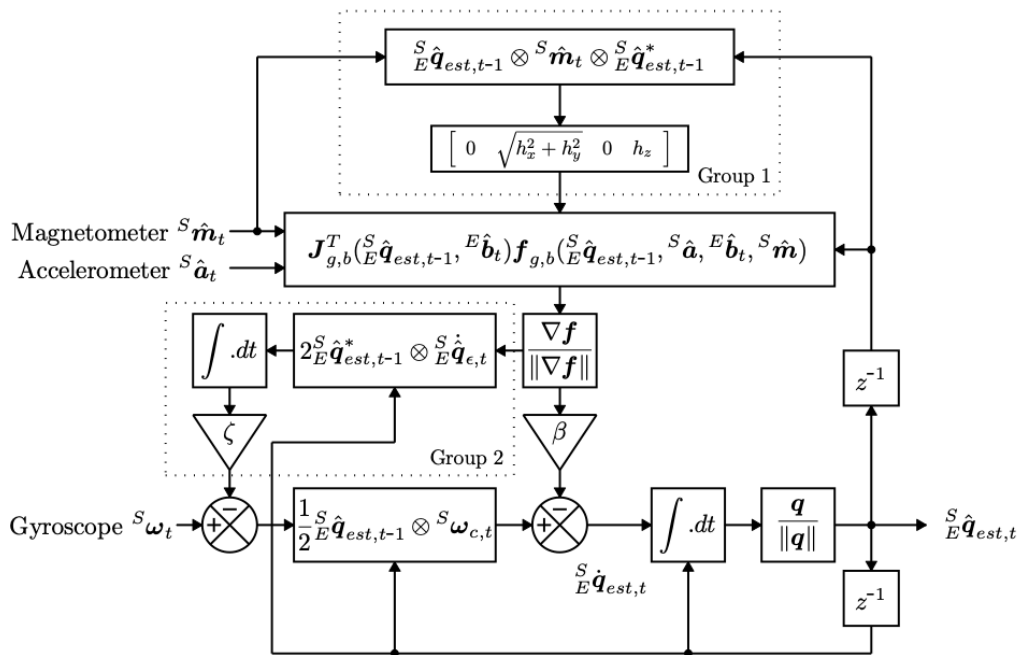


Figura 17. Diagrama de bloques del algoritmo de Madgwick para una implementación MARG, incluyendo la distorsión magnética (Grupo 1) y la deriva compensada del giroscopio (Grupo 2). (Madgwick)

## Capítulo 5. Detalles de la implementación

En este quinto capítulo se encuentran descritos con detalle los aspectos mas relevantes de la implementación hardware y software del proyecto. Para ello se ha seguido el desarrollo de una aplicación similar creada para la misma finalidad, la recogida de datos de un dispositivo *wearable*. (Viguer Soriano)

### 5.1 Hardware

No ha sido alterado ningún componente del hardware del dispositivo wearable. Este era uno de los propósitos del proyecto, no modificar ni el firmware ni el código interno del dispositivo. Por lo que, en cuanto a la configuración de los sensores, se ha realizado mediante software, siguiendo el manual de usuario proporcionado por los fabricantes.

### 5.2 Software

A continuación, se encuentra detallado el desarrollo de las diferentes partes del software. Para un mejor entendimiento de cada parte del software se ha hecho uso de nuevo del diagrama de bloques de la Sección 3.3.2.

#### 5.2.1 Búsqueda del dispositivo

Este apartado engloba la búsqueda, la selección y la conexión con el dispositivo wearable mediante BLE.

Para este tipo de comunicación es necesario definir tres permisos en el fichero *AndroidManifest.xml*. Los dos primeros corresponden con la configuración del servicio de *Bluetooth* y el tercero al permiso de ubicación, ya que es imprescindible para la comunicación BLE. También se ha definido el servicio BLE para los *smartphones* que dispongan del mismo, tal como se muestra en la Figura 18.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.acme.RunningApp"
4     android:installLocation="auto">
5
6     <uses-permission android:name="android.permission.BLUETOOTH" />
7     <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
8     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
9     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
10
11     <uses-feature
12         android:name="android.hardware.bluetooth_le"
13         android:required="true" />
```

Figura 18. Parte del manifiesto correspondiente al BLE.

En cuanto a la actividad que realiza la búsqueda (*ScanActivity*), lo más significativo es la definición del manejador y el adaptador (*mBleManger* y *mBleAdapter*). Cabe destacar la importancia de comprobar si el smartphone utilizado es capaz de soportar esta tecnología *Bluetooth*.

```
138 //Configuración del adaptador Bluetooth
139 mBleManager = (BluetoothManager) getSystemService(Context.BLUETOOTH_SERVICE);
140 mBleAdapter = mBleManager.getAdapter();
141
142 //Configuración del escáner Bluetooth
143 mBleScanner = mBleAdapter.getBluetoothLeScanner();
144
145 //Comprobar si el dispositivo soporta Low Energy Bluetooth
146 if (mBleAdapter == null) {
147     // El dispositivo no soporta Low Energy Bluetooth
148     Toast.makeText( context: this,
149         text: "El dispositivo no soporta Low Energy Bluetooth", Toast.LENGTH_LONG).show();
150 }
```

Figura 19. Configuración del adaptador y manejador.

Desde la versión SDK > 22, no vale solo con incluir las definiciones anteriormente mencionadas en el manifiesto, es necesario pedir explícitamente al usuario los permisos. Por lo que se le pide autorizar mediante un cuadro de diálogo la ubicación del dispositivo. Antes de empezar con la búsqueda de los dispositivos es necesario verificar la activación del *Bluetooth* y la ubicación GPS. En caso de no estar activados se pedirá su activación.

```
152 //Comprobar si posee permiso de ubicación y si no pedir al usuario
153 if (this.checkSelfPermission(Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
154     requestPermissions(
155         new String[] {Manifest.permission.ACCESS_COARSE_LOCATION },
156         PERMISSION_REQUEST_COARSE_LOCATION);
157     }
158
159 //Habilitación del Bluetooth
160 if (mBleAdapter != null && !mBleAdapter.isEnabled()) {
161     Intent enableIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
162     startActivityForResult(enableIntent, PERMISSION_REQUEST_BLUETOOTH_ENABLE);
163 }
164
165 //Encendido del sistema GPS si está apagado
166 locationManager = (LocationManager) getSystemService(LOCATION_SERVICE);
167 if (!locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
168     //Mostrar mensaje
169     dialogos(D_LOCATION_DISABLED);
170 }else{
171     //Empezar búsqueda de dispositivos
172     startScan( enable: true);
173 }
174
175 }
```

Figura 20. Configuración del adaptador y manejador.

Tras definir los permisos y verificar que ambos servicios están activados, ya es posible iniciar la búsqueda de dispositivos. Se encuentra configurado para que al empezar la actividad, la búsqueda empiece automáticamente y dure 3 segundos, tal como se muestra en la Figura 21. Sin embargo, existe un botón para volver a realizar la búsqueda de dispositivos las veces que sea necesario.





```
438 |  
439 |  
440 |  
441 | private ScanCallback mScanCallback = new ScanCallback() {  
442 |  
443 |     @Override  
444 |     public void onScanResult(int callbackType, ScanResult result) {  
445 |         BluetoothDevice mBleDevice;  
446 |         mBleDevice = result.getDevice();  
447 |         if (mBleDevice != null && mBleDevice.getName() != null &&  
448 |             mBleDevice.getName().contains("WeSU") && !mBleDevices.containsValue(mBleDevice)) {  
449 |             //Añadir dispositivo a la lista  
450 |             mBleDevices.put(mBleDevice.getAddress(), mBleDevice);  
451 |             devicesArray = mBleDevices.keySet().toArray(new String[mBleDevices.keySet().size()]);  
452 |         }  
453 |     }  
454 | };
```

Figura 22. Manejar el resultado de la búsqueda de dispositivos.

Una vez finalizada la búsqueda y visualizar en pantalla nuestro dispositivo, el usuario debe seleccionarlo para iniciar la conexión. Esto se realiza mediante el método *ConnectBLEDevice*.

```
459 |  
460 |  
461 |  
462 |  
463 |  
464 |  
465 |  
466 |  
467 |  
468 |  
469 |  
470 |  
471 |  
472 |  
473 |  
474 |
```

Figura 23. Método para conectarse a un dispositivo.

### 5.2.2 Obtención y modificación de los datos

En primer lugar, es necesario definir los roles en la comunicación, como se ha explicado en la sección 4.1, el *smartphone* actúa como cliente mientras que el dispositivo *wearable* como servidor.

Es necesario definir también diferentes servicios y características mediante un UUID. Dichos identificadores se encuentran detallados en el manual de usuario. Se observan dos servicios: de configuración y de lectura de los sensores. El primero de ellos contiene la característica para acceder a los registros y el segundo la característica para leer el acelerómetro, el giróscopo y el magnetómetro.

```
94     public static final UUID UUID_DESCRIPTOR = UUID.fromString("00002902-0000-1000-8000-00805f9b34fb");
95
96     public static final UUID UUID_SERVICE_SENSORS_DATA = UUID.fromString("00000000-0001-11e1-9ab4-0002a5d5c51b");
97     public static final UUID UUID_CHARACTERISTIC_SENSORS_DATA = UUID.fromString("00e00000-0001-11e1-ac36-0002a5d5c51b");
98
99     public static final UUID UUID_CONFIG_SERVICE = UUID.fromString("00000000-000f-11e1-9ab4-0002a5d5c51b");
100    public static final UUID UUID_REGISTER_ACCESS = UUID.fromString("00000001-000f-11e1-ac36-0002a5d5c51b");
```

Figura 24. Identificadores UUID.

Cuando la aplicación está lista para obtener datos de la actividad, tal como se ha explicado en la Sección 5.2.1, presionando el botón para empezar a recoger datos de los sensores, se llaman a dos métodos: *enableNotifications*, responsable de habilitar las notificaciones para poder recibir los datos de los sensores; y *configureSensors*, encargado de configurar los sensores según los parámetros requeridos.

```
582  |  /*
583  |  * Método para habilitar las notificaciones de los sensores configurados
584  |  * @param gatt GATT que se usará
585  |  */
586  |  public void enableNotifications(BluetoothGatt gatt) {
587  |
588  |      BluetoothGattService service, service_config;
589  |      BluetoothGattCharacteristic characteristic, characteristic_config;
590  |      BluetoothGattDescriptor descriptor;
591  |
592  |      try {
593  |          service = gatt.getService(UUID_SERVICE_SENSORS_DATA);
594  |          characteristic = service.getCharacteristic(UUID_CHARACTERISTIC_SENSORS_DATA);
595  |
596  |          // Habilitar notificaciones locales
597  |          gatt.setCharacteristicNotification(characteristic, enable: true);
598  |
599  |          // Habilitar notificaciones remotas
600  |          descriptor = characteristic.getDescriptor(UUID_DESCRIPTOR);
601  |          descriptor.setValue(BluetoothGattDescriptor.ENABLE_NOTIFICATION_VALUE);
602  |          gatt.writeDescriptor(descriptor);
603  |
604  |          service_config = gatt.getService(UUID_CONFIG_SERVICE);
605  |          characteristic_config = service_config.getCharacteristic(UUID_REGISTER_ACCESS);
606  |
607  |          // Habilitar notificaciones locales
608  |          gatt.setCharacteristicNotification(characteristic_config, enable: true);
609  |
610  |          // Habilitar notificaciones remotas
611  |          gatt.writeDescriptor(descriptor);
612  |
613  |      } catch (NullPointerException npe) {
614  |          gatt.disconnect();
615  |          dialogos(D_DEVICE_DISCONNECTED);
616  |      }
617  |
618  |
619  |
620  |  }
```

Figura 25. Método para habilitar los sensores configurados.

```
652 |#| /**
653 |#| * Método para configurar el periodo y los sensores
654 |#| */
655 |#| public void configureSensors() {
656 |#|     setPeriod(mBleGatt, value: 104); //Configurar frecuencia timer a 104 Hz
657 |#|     setSensors(mBleGatt, sensor: 0, config: false, value: 16); //Configurar ACC FS: 16 g
658 |#|     setSensors(mBleGatt, sensor: 0, config: true, value: 104); //Configurar ACC ODR: 104 Hz
659 |#|     setSensors(mBleGatt, sensor: 1, config: false, value: 2000); //Configurar GYR FS: 2000 dps
660 |#|     setSensors(mBleGatt, sensor: 1, config: true, value: 104); //Configurar GYR ODR: 104 Hz
661 |#|     setSensors(mBleGatt, sensor: 2, config: false, value: 16); //Configurar MAG FS: 16 gauss
662 |#|     setSensors(mBleGatt, sensor: 2, config: true, value: 80); //Configurar MAG ODR: 80 Hz
663 |#|
664 |#|     Log.i( tag: "Running App", msg: "Sensors configured");
665 |#| }
```

Figura 26. Método para configurar los sensores.

El sistema puede recibir información de una característica en el momento que se activan las notificaciones de esta, mediante un método de la clase *BluetoothGattCallback*, llamado *onCharacteristicRead*. Se ha modificado para que avise al manejador, el encargado de almacenar la información y convertir los datos para poder trabajar con ellos, de que se está registrando una actividad.

```
535 |#| @Override
536 |#| public void onCharacteristicRead(BluetoothGatt gatt, BluetoothGattCharacteristic characteristic, int status) {
537 |#|
538 |#|     if(characteristic.getUuid().equals(UUID_CHARACTERISTIC_SENSORS_DATA)){
539 |#|         mHandler.sendMessage(Message.obtain( h: null, MSG_MOVEMENT, characteristic));
540 |#|     }
541 |#| }
```

Figura 27. Método para obtener los datos de la característica.

### 5.2.2.1 Modificación de los datos

El dispositivo wearable se encuentra configurado de fábrica para transferir los datos recogidos mediante una palabra de 20 bytes, tal como se muestra en la Figura 28.

Destacar que todos los paquetes empiezan con dos bytes del formato *uint16* correspondientes al *timestamp*.

Octets LSB	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Name	TS		Accelerometer				Gyroscope				Magnetometer									
Axis	-		X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z	X	Y	Z
Value	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF	0xFFFF

**Accelerometer payload:** mg, signed int16

**Gyroscope payload:** tenth of dps, signed int16

**Magnetometer payload:** mGa, signed int16

Figura 28. Formato de los datos transmitidos. (STEVAL-WESU1 - How to use)

Para obtener los datos en la escala requerida se ha implementado la clase *Conversión*, la cual realiza los factores de conversión necesarios para obtener: g en el acelerómetro, rad/s para el giróscopo y Ga para el magnetómetro.

```

8  /**
9  * @param value array de bytes con los datos recibidos
10 * @return entero con el tiempo in milliseconds
11 */
12 @ public static Integer convertTs(byte[] value) {
13
14     //TIMESTAMP uint16
15     int ts = ByteBuffer.wrap(value, offset: 0, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() & 0xFFFF;
16     System.out.println("\n TIMESTAMP:" + ts);
17
18     return (ts);
19 }
20
21 /**
22 * @param value array de bytes con los datos recibidos
23 * @return array de reales con los valores de aceleración en g los 3 ejes
24 */
25 @ public static Double[] convertACC(byte[] value) {
26
27     //ACCELERATION int16
28     Double[] ACC = new Double[3];
29     //X axis
30     ACC[0] = (int) ByteBuffer.wrap(value, offset: 2, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 1000.0;
31
32     //Y axis
33     ACC[1] = (int) ByteBuffer.wrap(value, offset: 4, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 1000.0;
34
35     //Z axis
36     ACC[2] = (int) ByteBuffer.wrap(value, offset: 6, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 1000.0;
37
38     System.out.println("\n ACCELERACIÓN (g): [ " + ACC[0] + " , " + ACC[1] + " , " + ACC[2] + " ]");
39
40     return (ACC);
41 }

```

Figura 29. Función que permite la conversión de las unidades de los datos (1).

```
44  /**
45  * @param value array de bytes con los datos recibidos
46  * @return array de reales con los valores de giro en rad/s los 3 ejes
47  */
48  @ public static Double[] convertGYR(byte[] value) {
49
50      //GYROSCOPE int16
51      Double[] GYR = new Double[3];
52
53      //X axis
54      GYR[0] = ByteBuffer.wrap(value, offset: 8, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 10.0;
55
56      //Y axis
57      GYR[1] = ByteBuffer.wrap(value, offset: 10, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 10.0;
58
59      //Z axis
60      GYR[2] = ByteBuffer.wrap(value, offset: 12, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 10.0;
61
62      System.out.println("\n GIRÓSCOPO (rad/s): [ " + GYR[0] + " , " + GYR[1] + " , " + GYR[2] + " ]");
63
64      return new Double[] {toRadians(GYR[0]), toRadians(GYR[1]), toRadians(GYR[2])};
65  }
66
67  /**
68  * @param value array de bytes con los datos recibidos
69  * @return array of integers with the interpreted magnetometer values in Ga in the three axis
70  */
71  @ public static Double[] convertMAG(byte[] value) {
72
73      //MAGNETOMETER int16
74      Double[] MAG = new Double[3];
75
76      //X axis
77      MAG[0] = (int) ByteBuffer.wrap(value, offset: 14, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 1000.0;
78
79      //Y axis
80      MAG[1] = (int) ByteBuffer.wrap(value, offset: 16, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 1000.0;
81
82      //Z axis
83      MAG[2] = (int) ByteBuffer.wrap(value, offset: 18, length: 2).order(ByteOrder.LITTLE_ENDIAN).getShort() / 1000.0;
84
85      System.out.println("\n MAGNÉTOMETRO (Ga): [ " + MAG[0] + " , " + MAG[1] + " , " + MAG[2] + " ]");
86
87      return (MAG);
88  }
```

Figura 30. Función que permite la conversión de las unidades de los datos (2).

### 5.2.2.2 Base de datos

Para almacenar los datos recogidos por los sensores se ha creado una base de datos *SQLite*, esta permite almacenar la información estructurada en tablas.

Se ha hecho uso este tipo de base de datos porque se encuentra integrado en el entorno de desarrollo utilizado para este proyecto. Los métodos utilizados se encuentran en la clase *DatabaseSQLHelper* que extiende de la clase *SQLiteOpenHelper*.

Para este proyecto la solución tomada es crear tres tablas dentro de la base de datos: la primera tabla contiene la información de cada usuario y su correspondiente información; la segunda tabla contiene las características de las diferentes actividades; y finalmente la tercera tabla los datos modificados de cada sensor.

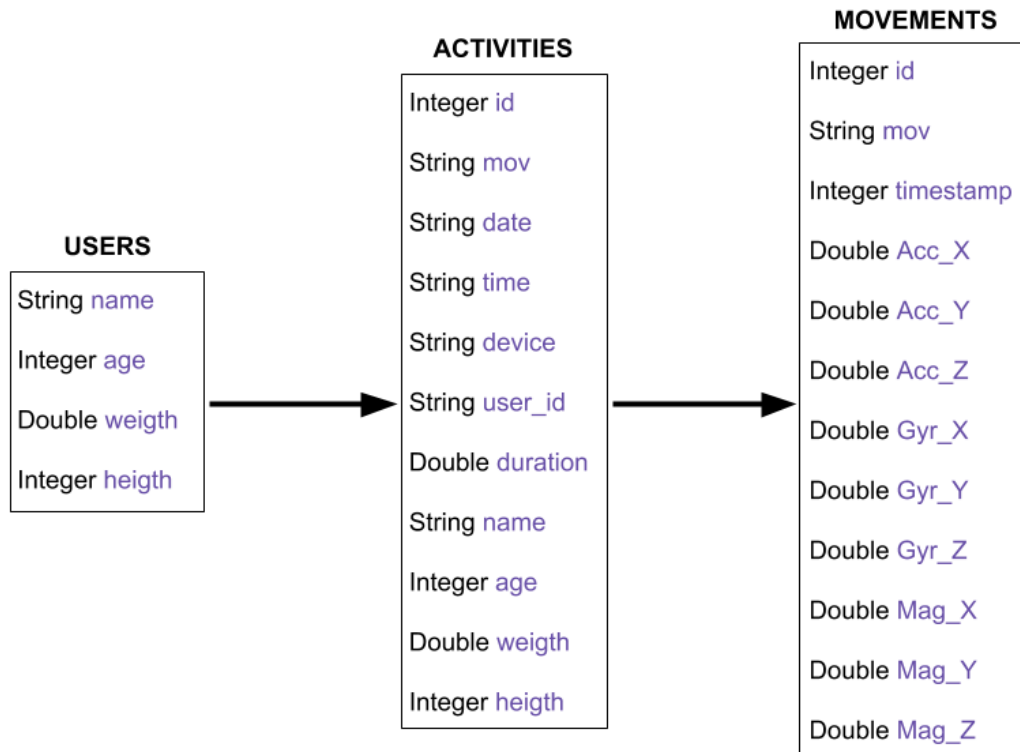


Figura 31. Relación entre las diferentes tablas que forman la base de datos.

### 5.2.2.3 Representación de los datos

Para la representación gráfica de los datos dentro de la aplicación se ha hecho uso de la librería *MPAndroidChart*. Se trata de una librería externa del entorno de *Android Studio* que permite insertar gran variedad de gráficos.



En la clase *MovimentDetailsFragment* definimos el objeto *LineChart*, el tipo de gráfico utilizado de la librería mencionada.

Los parámetros que vamos a representar corresponden con los obtenidos por el acelerómetro en sus tres ejes, por lo que es necesario dividirlos en tres columnas de datos diferentes. Para una buena visualización de los datos, mediante el objeto *LineDataSet* es posible definir el color o el nombre de los ejes, entre otros.

Finalmente, los datos de cada eje son almacenados en un objeto *ArrayList* y posteriormente en un objeto *LineData*.

```
44  /**
45   * Método que define las acciones a realizar al crearse la vista
46   */
47  @Override
48  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
49
50      //Definir el layout a usar
51      View view = inflater.inflate(R.layout.fragment_activity_detail, container, attachToRoot: false);
52
53      //Referencias a objetos del layout e inicialización
54      mDate = (TextView) view.findViewById(R.id.register_value);
55      mDate.setText(((MovementDetailsActivity) getActivity()).date);
56
57      mUser = (TextView) view.findViewById(R.id.by_value);
58      mUser.setText(((MovementDetailsActivity) getActivity()).user);
59
60      mDevice = (TextView) view.findViewById(R.id.device_value);
61      mDevice.setText(((MovementDetailsActivity) getActivity()).device);
62
63      mDuration = (TextView) view.findViewById(R.id.duration_value);
64      mDuration.setText(((MovementDetailsActivity) getActivity()).duration);
65
66      //Definición e inicialización de la gráfica con los datos a representar
67      mLineChart = (LineChart) view.findViewById(R.id.lineChart);
68
69      LineDataSet set1 = new LineDataSet(((MovementDetailsActivity) getActivity()).y_acc_x, label: "ACC X");
70      set1.setAxisDependency(YAxis.AxisDependency.LEFT);
71      set1.setDrawCircleHole(false);
72      set1.setColor(Color.BLUE);
```

Figura 32. Primera parte del código que define las acciones a realizar al crearse la vista

```
74 LineDataSet set2 = new LineDataSet(((MovementDetailsActivity) getActivity()).y_acc_y, label: "ACC Y")
75 set2.setAxisDependency(YAxis.AxisDependency.LEFT);
76 set2.setDrawCircleHole(false);
77 set2.setColor(Color.RED);
78
79 LineDataSet set3 = new LineDataSet(((MovementDetailsActivity) getActivity()).y_acc_z, label: "ACC Z")
80 set3.setAxisDependency(YAxis.AxisDependency.LEFT);
81 set3.setDrawCircleHole(false);
82 set3.setColor(Color.GREEN);
83
84 ArrayList<ILineDataSet> lineDataSets = new ArrayList<>();
85 lineDataSets.add(set1);
86 lineDataSets.add(set2);
87 lineDataSets.add(set3);
88
89 LineData data = new LineData(lineDataSets);
90 mLineChart.setData(data);
91 mLineChart.invalidate();
92
93 return view;
94 }
```

Figura 33. Segunda parte del código que define las acciones a realizar al crearse la vista.

Los datos obtenidos de cada eje del acelerómetro se encuentran definidos en un vector (uno por eje) dentro de la actividad *MovementDetailsFragment*, gracias al método *getMovements* implementado. Los datos se obtienen de la base de datos a partir del identificador de la actividad.

```
183 /**
184  * Método para cargar los movimientos a mostrar en un ArrayList
185  */
186 private void getMovements(){
187     Cursor c= mDatabaseSQLHelper.getMovementsByActivityId(ID);
188     if (c.moveToFirst()) {
189         //Recorrer el cursor hasta que no haya más registros
190         do {
191             Movimiento m = new Movimiento(c);
192             y_acc_x.add(new Entry(m.getTimestamp(),m.getAcc_X().floatValue()));
193             y_acc_y.add(new Entry(m.getTimestamp(),m.getAcc_Y().floatValue()));
194             y_acc_z.add(new Entry(m.getTimestamp(),m.getAcc_Z().floatValue()));
195         } while(c.moveToNext());
196     }
197 }
```

Figura 34. Primera parte del código que define las acciones a realizar al crearse la vista.

### 5.2.3 Administración de los usuarios

En cuanto a la administración de los usuarios, se han implementado diferentes métodos, actividades y fragments para poder crear, seleccionar, modificar y eliminar los diferentes usuarios.

Existen tres bloques en el diagrama de la Figura 10 que hacen referencia a esta sección: la selección, la creación y la edición de un usuario.

En primer lugar, los encargados de mostrar los usuarios registrados en la base de datos son la actividad *UsersActivity* y el fragment *Usersfragment*. Para ello es necesario un *ListView* y un cursor de la clase creada *UserCursorAdapter*. Esto es debido a que la base de datos devuelve objetos de la clase *Cursor*.

Siguiendo con la creación y edición de los usuarios, para su implementación se ha utilizado la clase *CreateNewUser* y el fragment *CreateNewUserFragment* junto con la base de datos.

### 5.2.4 Menú principal de usuario

Este bloque correspondiente al diagrama de bloques de la Figura 10 se encuentra reflejado en la clase *MainActivity*. En la Sección 5.2.4 es posible visualizar el resultado mediante una captura de pantalla de la aplicación. Además, posteriormente se detalla cada parte del menú.

### 5.2.5 Historial de actividades

Esta funcionalidad de la aplicación consiste en poder visualizar en el dispositivo móvil las actividades que un usuario ha realizado, ordenadas de más reciente a más antigua.

Para ello se ha implementado la clase *RecordsFragment*, procedente de la clase *MainActivity*, junto con la base de datos, donde se encuentran almacenadas las actividades realizadas.

El procedimiento de visualización y relleno de la lista con las actividades se encuentra detallado en la Sección 5.2.3, ya que se ha realizado de la misma forma para los usuarios.

### 5.2.6 *Detalles de la actividad*

La finalidad de esta parte de la aplicación es la visualización detallada de cada una de las actividades que ha realizado el usuario.

Para ello se ha implementado junto a la base de datos, la clase *MovementDetailsActivity* y el correspondiente fragment, *MovementDetailsFragments*.

Es posible acceder a esta pantalla de dos formas diferentes: la primera de ellas es tras finalizar una actividad, y la otra, mediante la selección de la actividad que se desea visualizar desde la pantalla del historial de actividades previamente explicada en la Sección 5.2.5.

En la Sección 6.1.6 se encuentra esta parte detalla con mas profundidad.

### 5.2.7 *Guardar la actividad en un fichero*

Finalmente, en este último bloque del diagrama de la Figura 10 se detalla como guardar los datos de una actividad en un fichero con la extensión CSV.

Se ha escogido este formato de archivo por la facilidad de importación para el análisis de datos, ya que se va a exportar a un ordenador con mayores recursos de procesamiento.

Para ello es necesario incluir en el manifiesto de la aplicación el permiso siguiente, que permite la escritura de datos en un almacenamiento externo.

```
10 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Figura 35. Permiso de escritura en un almacenamiento externo del manifiesto de la aplicación.

Por otra parte, es necesario comprobar que es posible escribir en el almacenamiento externo y que tiene espacio suficiente para admitir más archivos, por lo que se ha implementado el método *isExternalStorageWritable*.

Cabe destacar, la implantación del método *getPublicDocStorageDir* para obtener la ruta del directorio donde se van a escribir los archivos.

## 5.2.8 Análisis de los datos

Esta sección, parte de la importación de los datos a la herramienta de análisis matemático Matlab®, hasta finalmente la explicación detallada de los métodos de análisis utilizados. (Matlab®)

### 5.2.8.1 Importación de los datos

En primer lugar, se visualiza una pequeña parte del archivo donde se observa que los datos de interés se encuentran a partir de sexta fila del archivo.

	A	B	C	D	E	F	G	H	I	J
1	Actividad:,16,,,Usuario:,Diego									
2	Fecha y hora:,27 nov. 2020 , 18:24:53,,Edad:,25									
3	Dispositivo:,WeSU,,,Altura (cm):,180									
4	Duracion (s): ,16.606,,,Peso (kg):,85.0									
5										
6	ID, Tiempo, ACC_X ,ACC_Y ,ACC_Z, GYR_X, GYR_Y, GYR_Z, MAG_X, MAG_Y, MAG_Z									
7	10986,644,-0.067,-0.93,-0.288,0.05235987755982989,-0.11344640137963143,-0.0942477796076938,0.578,-0.041,0.693,									
8	10987,644,-0.067,-0.93,-0.288,0.05235987755982989,-0.11344640137963143,-0.0942477796076938,0.578,-0.041,0.693,									
9	10988,646,-0.071,-0.929,-0.293,0.061086523819801536,-0.11693705988362008,-0.09948376736367678,0.582,-0.047,0.687,									
10	10989,646,-0.071,-0.929,-0.293,0.061086523819801536,-0.11693705988362008,-0.09948376736367678,0.582,-0.047,0.687,									
11	10990,647,-0.073,-0.926,-0.295,0.061086523819801536,-0.12217304763960307,-0.10297442586766545,0.579,-0.047,0.687,									
12	10991,648,-0.07,-0.927,-0.291,0.0645771823237902,-0.1291543646475804,-0.1117010721276371,0.579,-0.04,0.69,									
13	10992,649,-0.06,-0.928,-0.285,0.05235987755982989,-0.143116998663535,-0.11868238913561441,0.578,-0.036,0.685,									
14	10993,650,-0.055,-0.935,-0.292,0.041887902047863905,-0.12566370614359174,-0.12566370614359174,0.574,-0.039,0.69,									
15	10994,651,-0.07,-0.922,-0.296,0.04014257279586958,-0.12566370614359174,-0.1291543646475804,0.574,-0.039,0.691,									
16	10995,652,-0.054,-0.928,-0.287,0.03665191429188092,-0.13264502315156904,-0.1291543646475804,0.58,-0.031,0.691,									
17	10996,653,-0.056,-0.932,-0.287,0.029670597283903602,-0.1308996938995747,-0.13264502315156904,0.58,-0.046,0.693,									
18	10997,654,-0.055,-0.925,-0.296,0.031415926535897934,-0.13788101090755203,-0.1413716694115407,0.578,-0.044,0.686,									
19	10998,655,-0.045,-0.925,-0.302,0.031415926535897934,-0.13788101090755203,-0.13962634015954636,0.573,-0.044,0.686,									
20	10999,656,-0.037,-0.923,-0.3,0.03665191429188092,-0.1361356816555577,-0.1361356816555577,0.573,-0.042,0.693,									
21	11000,657,-0.039,-0.931,-0.282,0.031415926535897934,-0.13264502315156904,-0.11693705988362008,0.57,-0.043,0.689,									
22	11001,659,-0.032,-0.938,-0.275,0.024434609527920613,-0.13264502315156904,-0.09948376736367678,0.57,-0.042,0.69,									
23	11002,659,-0.032,-0.938,-0.275,0.024434609527920613,-0.13264502315156904,-0.09948376736367678,0.57,-0.042,0.69,									
24	11003,660,-0.049,-0.931,-0.293,0.024434609527920613,-0.1361356816555577,-0.09250245035569947,0.563,-0.043,0.692,									
25	11004,661,-0.037,-0.931,-0.304,0.020943951023931952,-0.14835298641951802,-0.08552113334772216,0.567,-0.04,0.702,									
26	11005,662,-0.04,-0.93,-0.302,0.026179938779914945,-0.15533430342749532,-0.08028514559173916,0.573,-0.044,0.69,									
27	11006,663,-0.038,-0.922,-0.309,0.03316125578789226,-0.158824961931484,-0.07330382858376185,0.579,-0.044,0.69,									
28	11007,664,-0.024,-0.933,-0.299,0.04014257279586958,-0.13264502315156904,-0.0471238898038469,0.579,-0.037,0.694,									
29	11008,665,-0.027,-0.935,-0.296,0.05585053606381855,-0.11693705988362008,-0.031415926535897934,0.568,-0.034,0.696,									
30	11009,666,-0.036,-0.936,-0.298,0.05759586531581287,-0.1117010721276371,-0.019198621771937627,0.569,-0.034,0.69,									
31	11010,673,-0.047,-0.936,-0.282,0.10297442586766545,-0.1308996938995747,0.07853981633974483,0.578,-0.044,0.689,									

Figura 36. Ejemplo de un fichero CSS con datos que se van a analizar.

Mediante el siguiente código de Matlab es posible extraer los datos numéricos y separarlos en matrices para su posterior análisis:

```
M = csvread(filename,6,0);
time = M(:,2);
t=0;

for i=1:size(M,1)
    time(i)= t;
    t=t+(1/104);
end

Accelerometer = M(:, [3:5]);
Gyroscope      = M(:, [6:8]);
Magnetometer   = M(:, [9:11]);
```

Figura 37. Código empleado para la lectura del fichero y separación en matrices.

### 5.2.8.2 Algoritmo de Madgwick

Tras la importación de los datos, se implementa el algoritmo de Madgwick detallado en la Sección 4.3.4. Los parámetros que deben ser ajustados son la ganancia  $\beta$  y el periodo o frecuencia de muestreo. Este último parámetro debe coincidir con la frecuencia de muestreo con la que los sensores han realizado las medidas, 104 Hz.

La ganancia  $\beta$  hace referencia a la corrección de errores: el ruido, errores de calibración y cuantificación, entre otros. Tal como se encuentra detallado en el informe, el autor recomienda un valor  $\beta$  igual a 0.041 para una implementación óptima en MARG. (Madgwick)

También se ha definido el parámetro del cuaternión de orientación inicializado en (1, 0, 0, 0) tal como se indica en el informe previamente mencionado.

Todas las funciones utilizadas para su implementación que conforman el código completo del algoritmo, junto con su explicación detallada se encuentran en la página web. (Implementación del método)

### 5.2.8.3 Ángulos de Euler

La finalidad de esta sección es cambiar el sistema de representación de los datos de cuaterniones a ángulos de Euler, ya que este último facilita mucho la representación gráfica de los mismos. En concreto, se ha utilizado la convención de ángulos de Tait-Bryan detallada en la Sección 4.2.1.

El método utilizado para el cambio se encuentra en la web mencionada en la Sección 5.2.8.3; además de encontrarse explicado en el informe anterior. (Madgwick)

Debido a esta convención de ángulos adoptada, se ha solucionado el problema del *gimbal lock* explicado en la Sección 4.2.3 tal como se muestra en la Figura 38.

```
if((theta > 86) & (theta < 94))
    theta = pi/2* (180/pi);
    phi = 0* (180/pi);
    psi = 2 * atan2(q(:,2),q(:,1))* (180/pi);
end

if((theta > -94) & (theta < -86))
    theta = -pi/2* (180/pi);
    phi = 0* (180/pi);
    psi = -2 * atan2(q(:,2),q(:,1))* (180/pi);
end
```

Figura 38. Código empleado para eliminar el Gimbal Lock.

### 5.2.8.4 Compensación de la gravedad

En este apartado se encuentra explicado como se ha compensado la fuerza de la gravedad para obtener las medidas de la aceleración lo más real posible. Tal como explica el autor en su tesis, mediante la implementación de su método es posible remediar dicho error. (Varesano) (Compensar gravedad)

Sin embargo, partiendo del supuesto de que el usuario empieza la actividad sin moverse, para facilitar los cálculos posteriores se coge como valor de referencia 0 g en cada uno de

los tres ejes. Para ello se ha restado el valor inicial de cada uno de los ejes a cada uno de los valores del vector.

Los cálculos realizados se muestran en la Figura 39, donde “compensateACC” es la función que compensa la gravedad.

```
for j=1:1:(size(Accelerometer,1))
    acc(j,:) = compensateAcc(quaternion(j,:), Accelerometer(j,:));
    ACC(j,1) = acc(j,1);
    ACC(j,2) = acc(j,2);
    ACC(j,3) = acc(j,3);
end

for j=1:1:(size(Accelerometer,1))
    ACC(j,1) = Accelerometer(j,1) - Accelerometer(1,1);
    ACC(j,2) = Accelerometer(j,2) - Accelerometer(1,2);
    ACC(j,3) = Accelerometer(j,3) - Accelerometer(1,3);
end
```

Figura 39. Código empleado para compensar la gravedad y poner a 0 g la aceleración de referencia en cada uno de los ejes.

#### 5.2.8.5 Representación de los datos

Para una mejor representación y visualización de los datos se ha decidido crear una vista con cuatro gráficas: en primer lugar, la velocidad angular obtenida mediante el giróscopo en radianes por segundo; en segundo lugar, la aceleración obtenida por el acelerómetro en g (fuerza de la gravedad); seguidamente, la aceleración anterior tras compensar la gravedad; y finalmente, el flujo magnético obtenido por el magnetómetro en Gauss. En la Figura 40 se encuentra el código empleado para dichas representaciones.



```
%Subplot1 Giróscopo
axis(1) = subplot(4,1,1);
hold on;
plot(time, Gyroscope(:,1), 'r');
plot(time, Gyroscope(:,2), 'g');
plot(time, Gyroscope(:,3), 'b');
legend('X', 'Y', 'Z');
xlabel('Tiempo (s)');
ylabel('Velocidad angular (rad/s)');
title('Giróscopo');
hold off;

%Subplot2 Acelerómetro
axis(2) = subplot(4,1,2);
hold on;
plot(time, Accelerometer(:,1), 'r');
plot(time, Accelerometer(:,2), 'g');
plot(time, Accelerometer(:,3), 'b');
legend('X', 'Y', 'Z');
xlabel('Tiempo (s)');
ylabel('Aceleración (g)');
title('Acelerómetro');
hold off;

%Subplot3 Acelerómetro Corregido
axis(3) = subplot(4,1,3);
hold on;
plot(time, ACC(:,1), 'r');
plot(time, ACC(:,2), 'g');
plot(time, ACC(:,3), 'b');
legend('X', 'Y', 'Z');
xlabel('Tiempo (s)');
ylabel('Aceleración (g)');
title('Acelerómetro compensado');
hold off;

%Subplot4 Magnetómetro
axis(4) = subplot(4,1,4);
hold on;
plot(time, Magnetometer(:,1), 'r');
plot(time, Magnetometer(:,2), 'g');
plot(time, Magnetometer(:,3), 'b');
legend('X', 'Y', 'Z');
xlabel('Tiempo (s)');
ylabel('Flujo (G)');
title('Magnetómetro');
hold off;
```

Figura 40. Código empleado para representación de los datos.

#### 5.2.8.6 *Métodos de análisis empleados*

Se han implementado una serie de métodos en Matlab® para obtener diferentes parámetros de la actividad realizada que pueden ser muy útiles a la hora de analizar los movimientos efectuados.

Con los datos obtenidos por los sensores se ha logrado calcular el número de pasos que el usuario ha realizado, la aceleración máxima en cada uno de los pasos y a su vez la posición del tobillo empleando los ángulos de Euler.

En primer lugar, existen diferentes formas de interpretar y obtener los pasos que el usuario ha realizado.

Una forma de obtenerlo es mediante un comparador con histéresis. Se trata de fijar dos umbrales, en este caso se va a utilizar para comparar la velocidad angular en el eje Z obtenida por el giróscopo. El objetivo es definir ambos umbrales para calcular el número de pasos tanto si el usuario se encuentra andando como corriendo.

El resultado es una gráfica de 0's y 1's, con la que se obtienen el número de pasos. Se considera un paso cada vez que se realice la transición de un 1 a un 0, de izquierda a derecha. El método completo se muestra en la Figura 41.

También se ha intentado calcular el número de pasos mediante un comparador con histéresis cogiendo como datos de entrada el eje X de la aceleración compensada. Sin embargo, aparecen diferentes aceleraciones indeseadas tras la aceleración máxima que dificultan considerablemente el cálculo. Se ha ingeniado un método con solo un solo umbral capaz de obtener también los pasos.

Al haber un único umbral, se recogen tanto los valores de la aceleración deseados como las aceleraciones indeseadas que perjudican el cálculo. Debido a que estas últimas se encuentran después de una muestra nula (que no supera el umbral) se ha diseñado un filtro que elimine (iguale a 0) las 25 muestras posteriores al nulo que existe entre las aceleraciones deseadas y las que no.

```
um1=??
um2=??

vec_pasos=Gyroscope(:,3);

vec_comp=vec_pasos;

for u=1:length(vec_pasos)
    if vec_pasos(u) > um1
        vec_comp(u)=1;
    end

    if vec_pasos(u) < um2
        vec_comp(u)=0;
    end
end

for u=1:length(vec_pasos)
    if vec_pasos(u) < um1 && vec_pasos(u) > um2 && vec_pasos(u) ~= 1
        if u>1
            if vec_comp(u-1)==1
                vec_comp(u)=1;
            else
                vec_comp(u)=0;
            end
        end
    end
end

figure
stem(vec_comp)

Npasos=0;

for s=1:length(vec_comp)
    if s < length(vec_comp)
        if vec_comp(s)==1 && vec_comp(s+1)==0
            Npasos=Npasos+1;
        end
    end
end
```

Figura 41. Método para obtener los pasos mediante un comparador con histéresis con las medidas del giróscopo.

Con ello se consigue un resultado semejante al obtenido con el comparador con histéresis anterior. La cuenta de los pasos se realiza de la misma forma y el método se puede visualizar en la Figura 42.

```
Picos=ACC(:,1);  
  
Picos_comp=zeros(length(Picos),1);  
  
uml=2.5;  
  
for u=1:length(Picos)  
    if Picos(u) > uml  
        Picos_comp(u)=Picos(u);  
    end  
end  
  
for i=1:length(Picos_comp)  
    if Picos_comp(i)>0  
        Picos_comp(i)=1;  
    end  
end  
  
for i=1:length(Picos_comp)  
    if Picos_comp(i)==1 && Picos_comp(i+1)==0  
        Picos_comp((i+1):(i+25))=0;  
    end  
end  
  
figure  
stem(Picos_comp)
```

Figura 42. Método para calcular el número de pasos con las medidas del acelerómetro.

Otro de los parámetros más significativos es la aceleración máxima en cada uno de los pasos. Para ello se ha partido de las aceleraciones que superan los umbrales (descartando las aceleraciones indeseadas) obtenidas con el comparador de un solo umbral.

Los valores aceleraciones que superan el umbral se almacenan en un vector, en el que las posiciones no pertenecientes a las mismas se consideran nulas (es decir, 0).

Seguidamente se lee el vector hasta encontrar el primer valor no nulo, este y los siguientes valores consecutivos no nulos se almacenan en un vector auxiliar del que se calcula el mayor y se almacena en un vector de máximos. Dicho máximo corresponde con la aceleración máxima en el primer paso.

Se sigue leyendo el vector hasta que se vuelve a encontrar valores de aceleración rodeados de nulos, y así sucesivamente hasta obtener el máximo de cada paso.

Como resultado se obtiene un vector donde cada valor corresponde con el máximo de cada paso.

```
Picos_reales=zeros(length(Picos_comp),1);

for a=1:length(Picos_comp)
    if Picos_comp(a)==1
        Picos_reales(a)=ACC(a,1);
    end
end

Compar_max=zeros(length(Picos_reales),1);
Maximos= [];
m=1;

for b=1:length(Picos_reales)
    if Picos_reales(b) ~= 0
        Compar_max(b)=Picos_reales(b);

        if Picos_reales(b+1)==0
            Maximos(m)=max(Compar_max);
            m=m+1;
            Compar_max=zeros(length(Picos_reales),1);
        end
    end
end
```

Figura 43. Método para obtener la aceleración máxima de cada paso.

Finalmente, se han obtenido los ángulos de Euler para cada máximo de aceleración anterior. Para una mejor visualización de estos, se ha creado un método que genera una matriz donde cada fila corresponde con la aceleración máxima de cada paso, el instante en segundos de cuando se ha realizado, y cada una de las componentes de dichos ángulos.

```
Angulos=[];  
  
for a1=1:length(Maximos)  
    for a2=1:length(euler)  
        if Maximos(a1)==ACC(a2,1)  
            Angulos(a2,1)=time(a2);  
            Angulos(a2,2)=Maximos(a1);  
            Angulos(a2,3)=euler(a2,1);  
            Angulos(a2,4)=euler(a2,2);  
            Angulos(a2,5)=euler(a2,3);  
        end  
    end  
end  
  
Ang_Acc_Time_Max=zeros(length(Maximos),5);  
indice=1;  
  
for s=1:length(Angulos)  
    if Angulos(s,1) ~= 0  
        Ang_Acc_Time_Max(indice,:)=Angulos(s,:);  
        indice=indice+1;  
    end  
end
```

Figura 44. Método para obtener la matriz con el instante de tiempo y posición de cada aceleración máxima.

## Capítulo 6. Resultados obtenidos

Este capítulo se encuentra dividido en dos partes: el resultado final de la aplicación móvil y los resultados obtenidos tras el análisis de los datos en Matlab®.

### 6.1 Aplicación móvil

En esta sección se puede comprobar visualmente el resultado final de la aplicación. A su vez, se encuentran detalladas con más profundidad algunas de las secciones anteriores.

#### 6.1.1 Inicio

Al ejecutar la aplicación, en primer lugar, aparece el nombre de la aplicación en color blanco y su respectivo logo representativo sobre un fondo azul a modo de bienvenida.

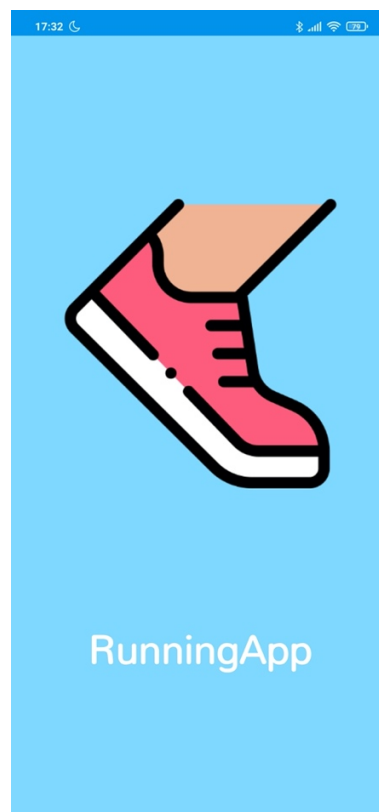
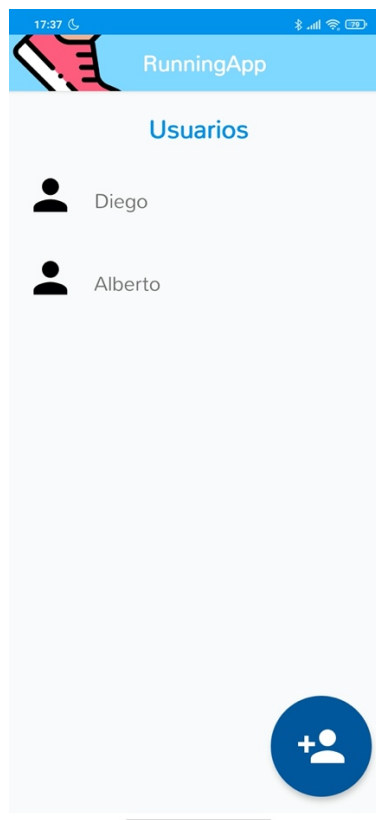


Figura 45. Pantalla de inicio de la aplicación.

### 6.1.2 Administración de usuarios

Al igual que en la sección 5.2.3 este apartado engloba los tres bloques que hacen referencia a la selección, creación y edición de usuarios.

En primer lugar, la selección de usuarios. A continuación, en la Figura 46, se visualiza dicho menú con los usuarios creados con anterioridad.



**Figura 46. Pantalla de selección de usuarios.**

En las dos siguientes capturas de pantalla se visualiza el menú de creación y edición de un usuario.



17:37

← Añadir usuario

Usuario

Nombre

Edad

Peso (kg)

Altura (cm)

Figura 47. Menú para añadir un nuevo usuario.

17:38

← Editar usuario

Usuario

Diego

23

85.0

179

Figura 48. Menú para editar un usuario.

### 6.1.3 Menú principal

A esta pantalla se accede tras elegir uno de los usuarios y se debe escoger una de las opciones que se presentan. Se puede seleccionar el menú de configuración, seleccionando el icono del engranaje situado en la parte superior derecha del menú, donde es posible cambiar de usuario, modificar los datos del usuario, borrar los datos o borrar los datos del usuario, tal como se muestra en la Figura 49.

La segunda opción es seleccionar el historial de actividades, situado en la parte inferior derecha de la pantalla; por último, la tercera opción, iniciar una nueva actividad presionando en el centro de la pantalla.

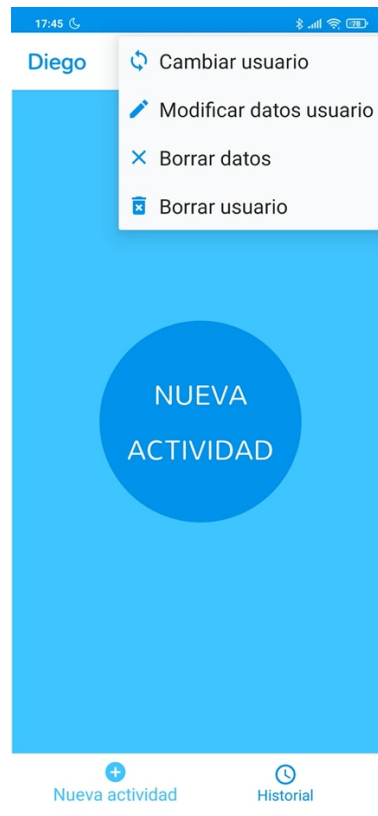


Figura 49. Menú para añadir un nuevo usuario.

#### 6.1.4 Selección del dispositivo

Esta parte de la aplicación es accesible tras seleccionar en el menú principal del usuario “NUEVA ACTIVIDAD”. Al entrar en la pantalla de selección, automáticamente empieza la búsqueda de dispositivos BLE y en una lista aparecerán los detectados. Para seleccionarlo simplemente hay que presionar sobre él.

En caso de no haber detectado el dispositivo deseado, es posible volver a realizar la búsqueda seleccionando el icono situado en la parte superior derecha de la pantalla.

En las tres siguientes figuras se visualizan los tres posibles estados: buscando dispositivos, no se han encontrado dispositivos y dispositivo encontrado.

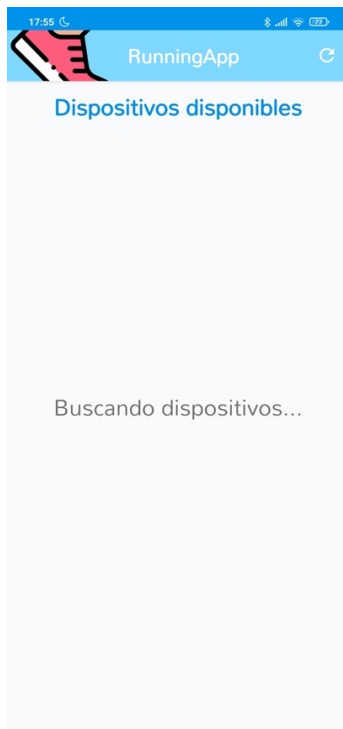


Figura 50. Buscando dispositivos.

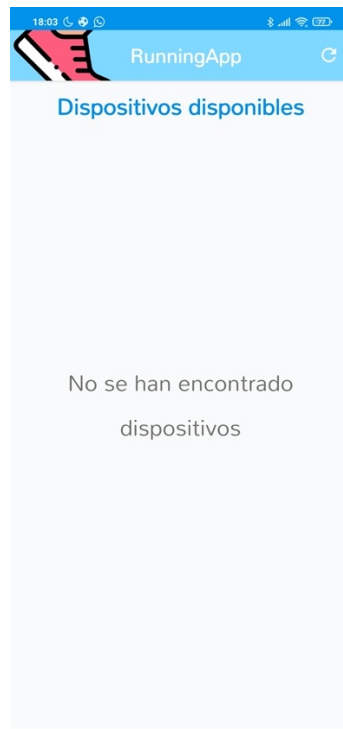


Figura 51. No se han encontrado dispositivos.

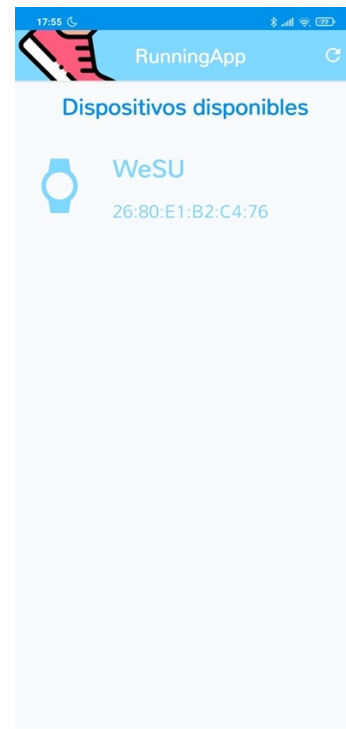


Figura 52. Dispositivo encontrado.

### 6.1.5 Registro de una nueva actividad

En este menú aparece de nuevo un círculo en medio de la pantalla con el texto “EMPEZAR CAPTURA”. En el momento que se presiona, el dispositivo comienza a recoger los datos obtenidos por los sensores.

Para parar la actividad se debe presionar de nuevo el círculo del medio, al cual le ha cambiado el nombre a “STOP”.

Por otro lado, existe la posibilidad de cancelar la captura y volver al menú anterior presionando sobre el botón de “CANCELAR”.



Figura 53. Antes de empezar la actividad.

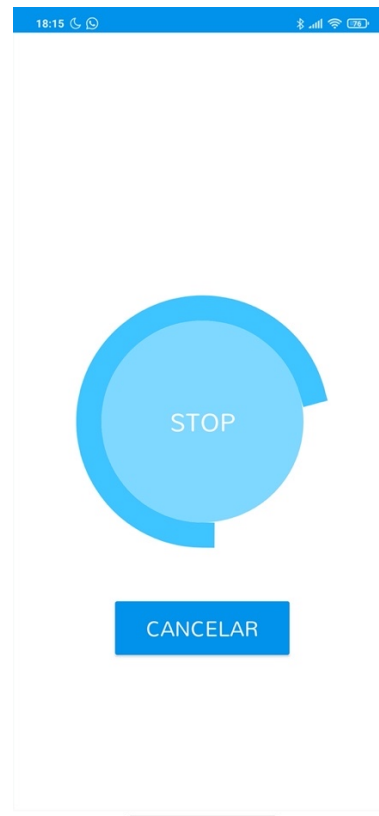


Figura 54. En medio de la actividad.

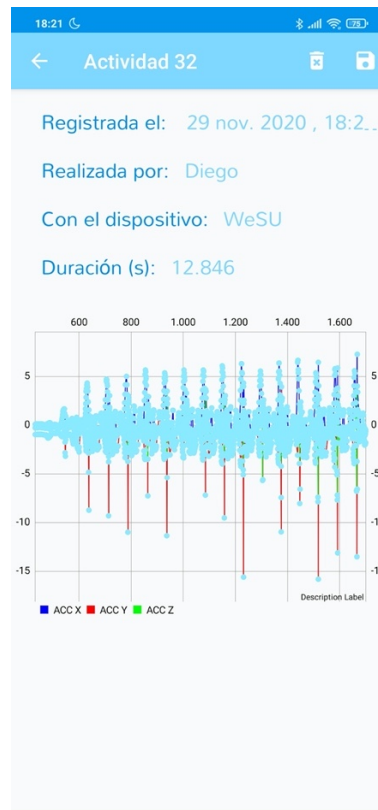
### 6.1.6 Detalles de la actividad

Es posible llegar a esta pantalla de dos maneras distintas: tras la captura de una actividad o desde el menú de historial de actividades que se encuentra explicado en la Sección 5.2.5

En esta pantalla se visualiza en la parte superior el número de la actividad, la fecha de realización, el usuario, el dispositivo utilizado y la duración de la misma.

En la parte inferior se encuentra una gráfica que representa la aceleración del dispositivo frente al número de muestras recogidas por el acelerómetro en cada uno de los tres ejes.

En la parte superior derecha de la pantalla aparecen dos iconos con los que (de izquierda a derecha) es posible borrar la actividad que se visualiza en pantalla o exportarla a un fichero CSV.



**Figura 55. Detalles de una actividad.**

### **6.1.7 Guardar la actividad en un fichero**

Para exportar la actividad a un fichero CSV únicamente es necesario especificar el nombre del fichero y presionar el botón de circular que se encuentra en la parte inferior derecha de la pantalla. Por defecto, el nombre que se le da corresponde con la fecha en que se ha realizado la actividad.

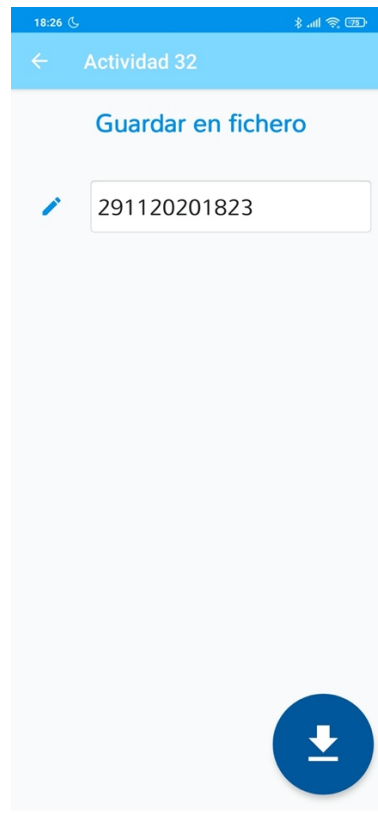


Figura 56. Exportar los datos a un fichero.

### 6.1.8 *Historial de actividades*

Se trata de una de las opciones posibles de selección explicadas en la Sección 6.1.3. En esta pantalla se visualizan en una lista las actividades realizadas y presionando sobre la deseada se avanza hasta los detalles de dicha actividad, detallada en la Sección 6.1.6.



Figura 57. Historial de actividades.

## 6.2 Análisis de los datos

A lo largo de esta sección se encuentran detallados los resultados obtenidos tras la implementación de los métodos mencionados en la Sección 5.2.8.6

En primer lugar, hay que destacar que se han realizado numerosas pruebas variando la posición de la pulsera en el tobillo, teniendo en cuenta siempre que encuentre en la parte exterior del tobillo derecho del usuario.

Los resultados se han obtenido con el dispositivo situado tal como se muestra en la Figura 58.



**Figura 58. Posición y orientación del dispositivo.**

A continuación, se encuentran los resultados de analizar dos actividades con un nivel de exigencia diferente entre ellas.

La primera de ellas consiste en un usuario corriendo a una velocidad de intensidad moderada durante unos escasos segundos tal como se visualiza en la combinación de graficas de la Figura 59.

La segunda actividad analizada consiste en un usuario andando con una intensidad moderada durante algunos segundos más que la actividad anterior, las medidas recogidas por los sensores se muestran en la Figura 60.

Ambas actividades han sido analizadas con los mismos métodos, ya que uno de los objetivos es que independientemente de la exigencia de la actividad se calculen los resultados sin modificar ningún parámetro del código de Matlab®.



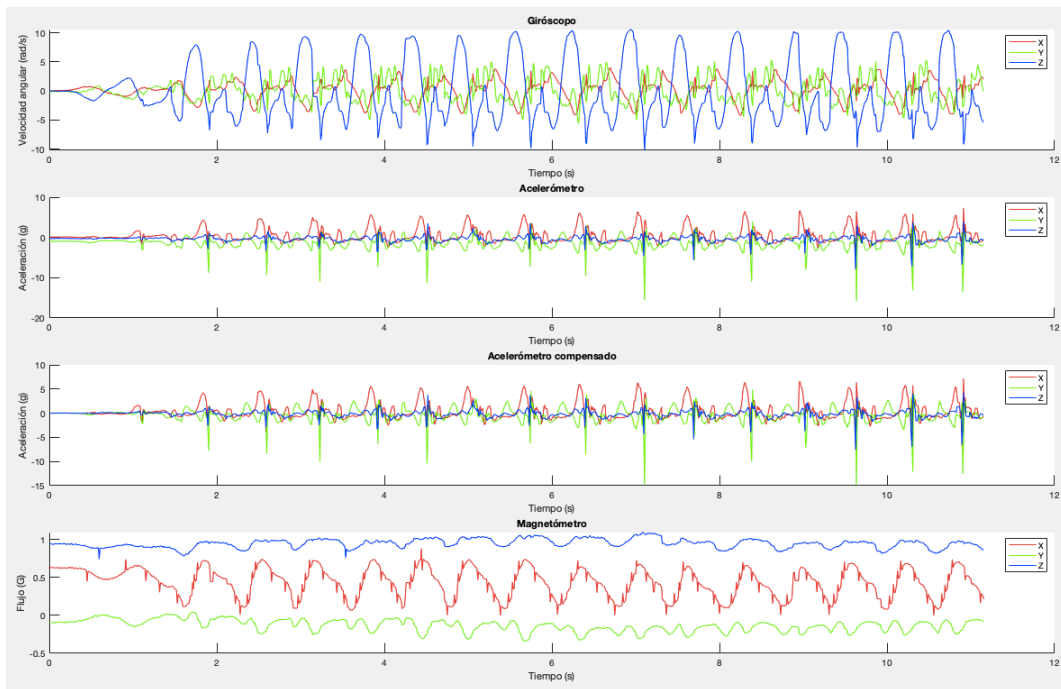


Figura 59. Medidas recogidas por los sensores en la actividad realizada corriendo.

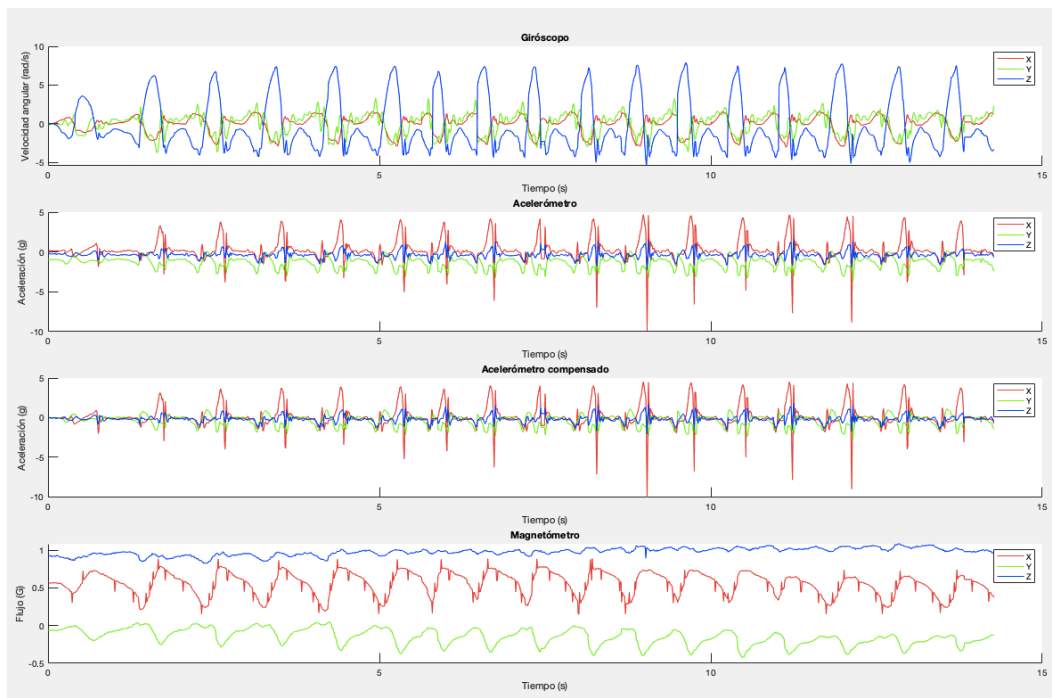


Figura 60. Medidas recogidas por los sensores en la actividad realizada andando.

De los dos métodos creados para el recuento de pasos, el comparador con histéresis sobre la velocidad angular es más seguro que el realizado con un único umbral sobre la aceleración compensada tras varias pruebas realizadas en diferentes actividades. Se han implementado estos métodos sobre las dos señales mencionadas por la facilidad de interpretación que tienen visualmente.

Los umbrales empleados en cada uno de los métodos son: en el comparador con histéresis el umbral superior son 4 radianes por segundo y el inferior 0 radianes por segundo; y para el comparador de un solo umbral son 4 g.

De este modo, para una mayor fiabilidad del recuento se recomienda llevar a cabo ambos métodos.

El método empleado para obtener la máxima aceleración en cada uno de los pasos parte del vector que se utiliza para calcular el número de pasos con los datos de la aceleración.

Aplicando dicho método se obtiene un vector con la longitud igual al número de pasos, en el que en la posición correspondiente a cada paso se visualiza la máxima aceleración.

En la Figura 61 y la Figura 62 se visualiza el vector de máxima aceleración del usuario corriendo y andando, respectivamente.

	1
1	4.2350
2	4.6550
3	4.9380
4	5.5670
5	5.2930
6	5.5840
7	5.5890
8	6.0220
9	6.2220
10	5.4630
11	6.2990
12	6.5950
13	5.3010
14	5.7130
15	5.4860

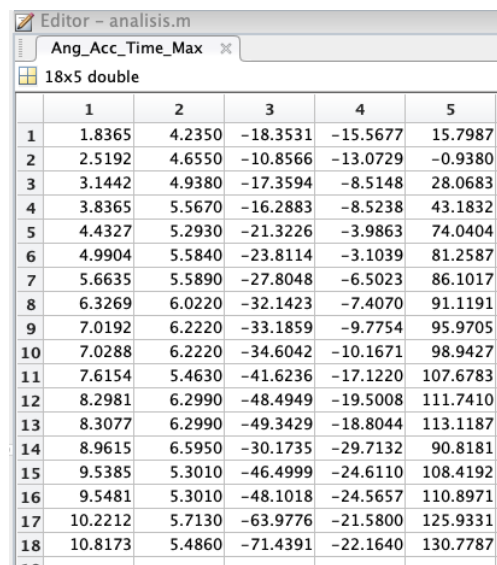
Figura 61. Máxima aceleración de cada paso realizado corriendo.

	1
1	3.1320
2	3.6610
3	3.7600
4	3.8730
5	3.9360
6	3.5970
7	4.0280
8	4.0290
9	4.0510
10	4.5300
11	4.4040
12	4.1590
13	4.5100
14	4.2960
15	4.1520
16	3.7560

Figura 62. Máxima aceleración de cada paso realizado andando

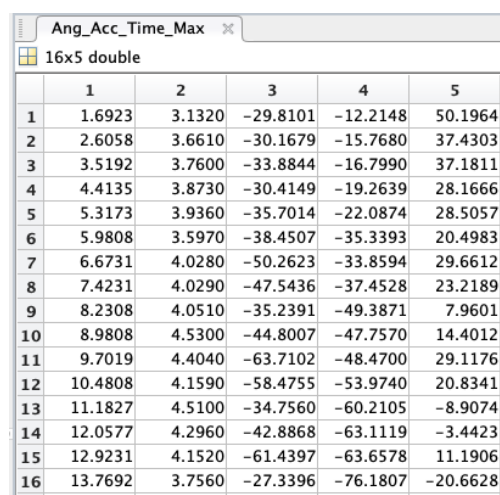
Finalmente, con el último método se crea una matriz de 5 columnas, la cual asigna respectivamente a cada fila el instante de tiempo, la aceleración máxima y cada una de las tres coordenadas de los ángulos de Euler, a cada uno de los pasos realizados.

Puede ocurrir que la aceleración máxima en un paso ocurra en dos o más instantes de tiempo diferentes (como se aprecia en la Figura 63), por lo que en la matriz no se debe descartar ninguna de las aceleraciones máximas repetidas.



	1	2	3	4	5
1	1.8365	4.2350	-18.3531	-15.5677	15.7987
2	2.5192	4.6550	-10.8566	-13.0729	-0.9380
3	3.1442	4.9380	-17.3594	-8.5148	28.0683
4	3.8365	5.5670	-16.2883	-8.5238	43.1832
5	4.4327	5.2930	-21.3226	-3.9863	74.0404
6	4.9904	5.5840	-23.8114	-3.1039	81.2587
7	5.6635	5.5890	-27.8048	-6.5023	86.1017
8	6.3269	6.0220	-32.1423	-7.4070	91.1191
9	7.0192	6.2220	-33.1859	-9.7754	95.9705
10	7.0288	6.2220	-34.6042	-10.1671	98.9427
11	7.6154	5.4630	-41.6236	-17.1220	107.6783
12	8.2981	6.2990	-48.4949	-19.5008	111.7410
13	8.3077	6.2990	-49.3429	-18.8044	113.1187
14	8.9615	6.5950	-30.1735	-29.7132	90.8181
15	9.5385	5.3010	-46.4999	-24.6110	108.4192
16	9.5481	5.3010	-48.1018	-24.5657	110.8971
17	10.2212	5.7130	-63.9776	-21.5800	125.9331
18	10.8173	5.4860	-71.4391	-22.1640	130.7787

Figura 63. Matriz con los datos obtenidos de cada paso realizado en la actividad corriendo



	1	2	3	4	5
1	1.6923	3.1320	-29.8101	-12.2148	50.1964
2	2.6058	3.6610	-30.1679	-15.7680	37.4303
3	3.5192	3.7600	-33.8844	-16.7990	37.1811
4	4.4135	3.8730	-30.4149	-19.2639	28.1666
5	5.3173	3.9360	-35.7014	-22.0874	28.5057
6	5.9808	3.5970	-38.4507	-35.3393	20.4983
7	6.6731	4.0280	-50.2623	-33.8594	29.6612
8	7.4231	4.0290	-47.5436	-37.4528	23.2189
9	8.2308	4.0510	-35.2391	-49.3871	7.9601
10	8.9808	4.5300	-44.8007	-47.7570	14.4012
11	9.7019	4.4040	-63.7102	-48.4700	29.1176
12	10.4808	4.1590	-58.4755	-53.9740	20.8341
13	11.1827	4.5100	-34.7560	-60.2105	-8.9074
14	12.0577	4.2960	-42.8868	-63.1119	-3.4423
15	12.9231	4.1520	-61.4397	-63.6578	11.1906
16	13.7692	3.7560	-27.3396	-76.1807	-20.6628

Figura 64. Matriz con los datos obtenidos de cada paso realizado en la actividad andando

## Capítulo 7. Conclusiones y propuesta de trabajo futuro

### 7.1 Conclusiones

En un principio, se propuso la realización de un proyecto que pudiera ayudar al deportista a mejorar en el deporte de *running*. La palabra “mejorar” se refiere en este caso a aumentar las capacidades del usuario a la hora de realizar este tipo de actividades. Por lo que el primer reto era focalizar cuales eran los puntos fuertes que le hacen al deportista prosperar en este deporte. Por ello, este proyecto se ha centrado en obtener la aceleración del deportista en los instantes que es máxima.

El primer objetivo, era escoger el dispositivo *wearable* más adecuado a nuestras necesidades. Vistos los resultados finales se puede afirmar que el STEVAL-WESU1 si que era adecuado y ha cumplido con las expectativas. Sin embargo, no es posible afirmar que sea el más adecuado porque únicamente hemos probado este en la realidad. Los demás dispositivos disponibles en el mercado han sido descartados por sus especificaciones (aun siendo algunos muy similares al empleado).

Otro de los objetivos planteados para este proyecto, era el diseño de una aplicación capaz de obtener los datos captados por los sensores para su posterior análisis. En cuanto a la aplicación, se ha diseñado a conciencia para que cumpla los requisitos que se planteaban, por lo que el objetivo está cumplido.

Para cumplir el objetivo anterior era necesario cumplir uno intermedio, el de comprender el protocolo de comunicación que permite enlazar el dispositivo *wearable* al *smartphone*. Se ha pasado de tener un conocimiento básico sobre el *Bluetooth Low Energy*, a entender con detalle como realizar una conexión, que tipo de paquetes transporta y que comporta esta tecnología.

Cabe destacar también, el conocimiento adquirido en relación con el entorno de desarrollo *Android Studio*, que antes de la realización del proyecto era mínimo.

Como último objetivo, el análisis de los datos, se han cumplido las expectativas iniciales, obtener la aceleración máxima de cada paso y los datos de utilidad en ese instante de



tiempo. Añadir que se ha desarrollado el análisis para cualquier nivel de exigencia, pensando en que cualquier persona, independientemente de su condición física, pueda hacer uso de este prototipo.

Se considera que este deporte llevado a un alto nivel requiere de gran técnica. Por lo que los resultados obtenidos pueden ayudar a un experto en la materia a mejorar el rendimiento del usuario. Por otro lado, esta aplicación puede ayudar a cualquier deportista a comprobar si ha mejorado a medida que realiza más entrenamientos.

Finalmente, mencionar un ligero inconveniente surgido a la hora de analizar los datos, a en concreto con la aceleración. Aparecen diferentes aceleraciones que no siguen el movimiento esperado que causan picos que dificultan el análisis. Estos han sido debidamente eliminados para obtener los resultados lo más real posibles.

## 7.2 Propuesta de trabajo futuro

Se podría estudiar la implementación en otros deportes como el atletismo, en concreto en la disciplina de salto de altura, salto de longitud, lanzamiento de disco, lanzamiento de martillo, entre otros. Donde el acelerómetro y el giróscopo juegan un papel muy importante.

En cuanto al *running*, se podría implementar en la aplicación el uso una nube donde almacenar todos entrenamientos realizados para no tener que exportarlos a un ordenador manualmente.

En cuanto al análisis, se podría profundizar más en los parámetros calculados a partir de los datos obtenidos por los sensores. No conformarse solo con la aceleración, obtener la distancia de la zancada del deportista, la velocidad a la que realiza la actividad o la distancia recorrida.

En caso de recorrer grandes distancias, implementar con la ayuda de la posición GPS una base de datos donde se almacene el lugar donde se ha realizado la actividad y el recorrido que se ha llevado a cabo.



También se podría implementar un sistema capaz de corregir la posición del pie al andar, comparando los ángulos de posición del tobillo con los de una base de datos para prevenir lesiones futuras.

En pocas palabras, se podría utilizar esta aplicación como punto de partida para futuras aplicaciones o implementar nuevas funcionalidades a la misma.



## Capítulo 8. Bibliografía

*Ángulos de Euler.* (2007). Obtenido de

[https://es.wikipedia.org/wiki/Ángulos\\_de\\_Euler#/media/Archivo:Euler.png](https://es.wikipedia.org/wiki/Ángulos_de_Euler#/media/Archivo:Euler.png)

*Ángulos de Euler.* (2020). Obtenido de [https://es.wikipedia.org/wiki/Ángulos\\_de\\_Euler](https://es.wikipedia.org/wiki/Ángulos_de_Euler)

*Ángulos de Tait-Bryan.* (2009). Obtenido de

[https://es.wikipedia.org/wiki/Ángulos\\_de\\_navegación#/media/Archivo:Taitbrian\\_angles.svg](https://es.wikipedia.org/wiki/Ángulos_de_navegación#/media/Archivo:Taitbrian_angles.svg)

*Ángulos de Tait-Bryan.* (2020). Obtenido de

[https://es.wikipedia.org/wiki/Ángulos\\_de\\_navegación](https://es.wikipedia.org/wiki/Ángulos_de_navegación)

*Android.* (2020). Obtenido de <https://es.wikipedia.org/wiki/Android>

*Android Studio.* (2020). Obtenido de <https://developer.android.com/studio>

*Apple Inc.* (2020). Obtenido de <https://www.apple.com>

*Arduino.* (2020). Obtenido de <https://www.arduino.cc>

*BALF-NRG-01D3.* (2017). Obtenido de <https://www.st.com/resource/en/datasheet/balf-nrg-01d3.pdf>

*Bloqueo del cardán.* (2019). Obtenido de

[https://es.wikipedia.org/wiki/Bloqueo\\_del\\_cardán](https://es.wikipedia.org/wiki/Bloqueo_del_cardán)

*BLUENRG-MS.* (2020). Obtenido de <https://www.st.com/en/wireless-transceivers-mcus-and-modules/bluenrg-ms.html>

*Bluetooth Generic Attributes.* (2020). Obtenido de

<https://www.bluetooth.com/specifications/gatt/>

*Compensar gravedad.* (2011). Obtenido de <http://www.varesano.net/blog/fabio/simple-gravity-compensation-9-dom-imus>

*Conexión entre Arduino y MPU-6050.* (2020). Obtenido de

<https://www.prometec.net/usando-el-mpu6050/>



- Cuaternion.* (2020). Obtenido de <https://es.wikipedia.org/wiki/Cuaternión>
- Dispositivos wearables.* (2014). Obtenido de <https://www.dispositivoswearables.net/>
- Filtro de Kalman.* (2005). Obtenido de [https://www.researchgate.net/figure/Figura-88-Etapas-del-filtro-de-Kalman\\_fig42\\_39425275](https://www.researchgate.net/figure/Figura-88-Etapas-del-filtro-de-Kalman_fig42_39425275)
- Filtro de Kalman.* (2020). Obtenido de [https://es.wikipedia.org/wiki/Filtro\\_de\\_Kalman](https://es.wikipedia.org/wiki/Filtro_de_Kalman)
- Gimbal lock.* (2019). Obtenido de [https://www.researchgate.net/figure/Ilustrates-the-principle-of-gimbal-lock-The-outer-blue-frame-represents-the-x-axis-the\\_fig4\\_338835648](https://www.researchgate.net/figure/Ilustrates-the-principle-of-gimbal-lock-The-outer-blue-frame-represents-the-x-axis-the_fig4_338835648)
- Google.* (2020). Obtenido de <https://about.google/intl/es/>
- Implementación del método.* (2012). Obtenido de <https://x-io.co.uk/quaternions/>
- IMU.* (2020). Obtenido de [https://es.wikipedia.org/wiki/Unidad\\_de\\_medición\\_inercial](https://es.wikipedia.org/wiki/Unidad_de_medición_inercial)
- Inverse Sense.* (2020). Obtenido de <https://inversense.tdk.com/>
- iOS.* (s.f.). Obtenido de <https://es.wikipedia.org/wiki/IOS>
- Las Provincias. (7 de Noviembre de 2020). *Nuevos límites para hacer deporte en Valencia.*
- LIS3MDL.* (2017). Obtenido de <https://www.st.com/resource/en/datasheet/lis3mdl.pdf>
- LSM6DS3.* (2017). Obtenido de <https://www.st.com/resource/en/datasheet/lsm6ds3.pdf>
- Madgwick, S. (2010). *An efficient orientation filter for inertial and inertial/magnetic sensor arrays.*
- Matlab®.* (2020). Obtenido de <https://es.mathworks.com/products/matlab.html>
- MPU6050.* (2016). Obtenido de [https://naylorpmechatronics.com/blog/45\\_Tutorial-MPU6050-Acelerómetro-y-Giroscopio.html](https://naylorpmechatronics.com/blog/45_Tutorial-MPU6050-Acelerómetro-y-Giroscopio.html)
- MPU-9250.* (2017). Obtenido de <https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2017/06/AN-IVS-0001EVB-00-v1.4.pdf>





*SimpleLink™ CC2650 SensorTag™*. (2015). Obtenido de

[https://www.ti.com/lit/ug/tidu862/tidu862.pdf?ts=1606826009702&ref\\_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FTIDC-CC2650STK-SENSORTAG](https://www.ti.com/lit/ug/tidu862/tidu862.pdf?ts=1606826009702&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FTIDC-CC2650STK-SENSORTAG)

*SimpleLink™ Wi-Fi® CC3200 SensorTag*. (2018). Obtenido de

[https://www.ti.com/lit/ug/swru514/swru514.pdf?ts=1606852282296&ref\\_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FCC3200STK-WIFIMK](https://www.ti.com/lit/ug/swru514/swru514.pdf?ts=1606852282296&ref_url=https%253A%252F%252Fwww.ti.com%252Ftool%252FCC3200STK-WIFIMK)

*Sistemas de referencia de Actitud y Rumbo*. (2019). Obtenido de

[https://es.wikipedia.org/wiki/Sistemas\\_de\\_Referencia\\_de\\_Actitud\\_y\\_Rumbo](https://es.wikipedia.org/wiki/Sistemas_de_Referencia_de_Actitud_y_Rumbo)

*Sociedad Aeronáutica Española*. (2020). Obtenido de

<https://www.sociedad aeronautica.org/control-aviones/>

*Statista*. (2016). Obtenido de [https://es.statista.com/estadisticas/600731/cuota-de-](https://es.statista.com/estadisticas/600731/cuota-de-mercado-de-sistemas-operativos-para-smartphones-por-pedidos--2020/)

[mercado-de-sistemas-operativos-para-smartphones-por-pedidos--2020/](https://es.statista.com/estadisticas/600731/cuota-de-mercado-de-sistemas-operativos-para-smartphones-por-pedidos--2020/)

*Statista*. (5 de Mayo de 2019). Obtenido de [https://es.statista.com/grafico/17572/envios-](https://es.statista.com/grafico/17572/envios-mundiales-de-dispositivos-wearables/)

[mundiales-de-dispositivos-wearables/](https://es.statista.com/grafico/17572/envios-mundiales-de-dispositivos-wearables/)

*STEVAL-WESU1 - How to use*. (2019). Obtenido de

[https://www.st.com/resource/en/user\\_manual/dm00279614-how-to-use-the-stevalwesu1-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/dm00279614-how-to-use-the-stevalwesu1-stmicroelectronics.pdf)

*STEVAL-WESU1*. (2016). Obtenido de

[https://www.st.com/resource/en/data\\_brief/steval-wesu1.pdf](https://www.st.com/resource/en/data_brief/steval-wesu1.pdf)

*STM32L151VE*. (s.f.). Obtenido de

<https://www.st.com/resource/en/datasheet/stm32l151ve.pdf>

*STM32L151VE*. (s.f.). Obtenido de

<https://www.st.com/resource/en/datasheet/stm32l151ve.pdf>

*STMicroelectronics*. (2020). Obtenido de [https://www.st.com/content/st\\_com/en.html](https://www.st.com/content/st_com/en.html)

*STNS01*. (2017). Obtenido de <https://www.st.com/resource/en/datasheet/stns01.pdf>



*Texas Instruments.* (2020). Obtenido de <https://www.ti.com/>

Varesano, F. (2011). <http://www.varesano.net/blog/fabio/my-mos-thesis-using-arduino-tangible-human-computer-interaction>.

Viguer Soriano, E. (2019). *Prototipo de dispositivo wearable de medida de rendimiento en deportes acrobáticos basado en unidad de medida inercial y comunicación a dispositivo móvil.*