







UNIVERSIDAD  
DE MÁLAGA



E.T.S.  
INGENIERÍA  
INFORMÁTICA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADO EN INGENIERÍA DE LA SALUD

**PULSERA INALÁMBRICA PARA  
MONITORIZACIÓN DEL PULSO CARDIACO**

**WIRELESS BRACALET FOR CARDIAC PULSE  
MONITORING**

Realizado por  
**Nelia de las Nieves Rivas Muñoz**

Tutorizado por  
**Antonio Jesús Bandera Rubio**

Departamento  
**Tecnología electrónica**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, FEBRERO 2019

Fecha defensa:

Fdo. El/la Secretario/a del Tribunal





*A mis padres, por su paciencia y apoyo, por los buenos momentos y sobre todo por los no tan buenos donde no me han dejado rendirme. Simplemente, gracias por estar.*

*A aquellos familiares y amigos, que han sido y son un apoyo incondicional.*

*Y a mí tutor, Antonio, por su labor y tiempo dedicado a este proyecto.*



## Resumen

A lo largo de este trabajo se diseñará un sistema inteligente basado en un ARM Cortex M-0 dotado de transceptor Bluetooth de baja energía, que podría incluirse en el ámbito de la telemedicina. Aunque se trata de un elemento que ya se puede encontrar en el mercado, con este proyecto se consigue un aprendizaje personal en profundidad sobre el estudio y captura de bioseñales. En este proyecto se ha hecho uso de una plataforma online, que trabaja en la nube, para la creación del código que captará las señales de interés, en nuestro caso el pulso cardíaco, usando fotoplestimografía. El elemento básico del hardware de nuestro sistema será la placa Tiny-BLE de Seeed Studio, construída en torno al chip nRF51822 de Nordic. El diseño incluye la etapa final de diseño de envoltente para nuestra propuesta. Por último, y acercando este proyecto a la telemedicina, se ha realizado una encuesta para valorar los conocimientos de las personas en un ámbito que está en pleno auge y expansión.

**Palabras clave:** fotoplestimografía, sensor de pulso, tiny-Ble, bluetooth, MBED, telemedicina, telemonitorización.

## Abstract

In this project we will design an intelligent system based on ARM Cortex M-0, it has a Bluetooth Low Energy (BLE) transceiver, this project can be included in the telemedicine area. Although it is an element that you can find on the current market, but with this project we can achieve a personal learning in deep about the research and capture of biosignals. In this project, we have used an online platform called MBED, that work in the cloud, to create the code that we will use to capture the signals of interest, in our case the cardiac pulse, using the photoplethysmography. The basic hardware element of our system will be the Tiny-BLE badge from Seeed Studio, built around the Nordic chip nRF51822. The design includes a final phase where we design the envelope for our proposal. And finally, and approaching this

project to the telemedicine's area, we have done a survey to value the knowledge in the society because the telemedicine is in full boom and expansion.

**Keywords:** photoplethysmography, pulse sensor, Tiny-BLE, bluetooth, MBED, telemedicine, telemonitoring.

# Índice

<b>Capítulo 1. Introducción.....</b>	<b>15</b>
1.1. Motivación y objetivos.....	15
1.2. Fundamentos previos.....	17
1.2.1 La placa tiny-BLE.....	17
1.2.2. El sensor de pulso por fotoplestimografía de pulse_sensor.....	18
1.3. Trabajos previos.....	19
1.4. Metodología y organización de la memoria.....	20
<b>Capítulo 2. Conceptos básicos.....</b>	<b>23</b>
2.1. Pulsioximetría basada en fotoplestimografía.....	23
2.2. Arquitectura hardware del sistema de captura.....	25
2.2.1. La placa Tiny-BLE.....	26
2.2.2. El Chip de Nordic nRF51822.....	27
2.2.3. El sensor de pulso.....	28
2.2.4. Conexión del sensor de pulso al chip de Nordic.....	30
2.3. Diseño Software.....	31
2.3.1. La plataforma MBED.....	31
2.4. Comunicaciones Bluetooth.....	33
2.4.1. Introducción a la tecnología Bluetooth.....	33
2.4.2. Especificaciones: protocolos y perfiles.....	34
2.4.3. Tecnología BLE.....	36
2.4.4. Servicios Bluetooth: GATT y UUID.....	38
<b>Capítulo 3. Sistema de captura y verificación de datos.....</b>	<b>41</b>
3.1. Organización del código.....	41
3.1.1. Captura de la señal de pulso.....	41
3.1.1.1. Variables e interrupciones.....	41
3.1.1.2. Algoritmo de estimación del pulso.....	44
3.1.2. Detección de golpecito ( <i>tap</i> ) en la placa.....	48

3.1.2.1. Acelerometro.h: instrucciones y funciones.....	49
3.1.2.2. Algoritmo de funcionamiento del acelerómetro.....	50
3.1.2.3. Librería EMPL_MPU6050.....	54
3.1.2.3. Librería nRF51822.....	54
3.1.3. BLE_part.....	55
3.1.3.1. Análisis de custom_service.....	56
3.1.3.2. Librería BLE_API.....	56
3.1.4. Código principal y archivos de cabecera.....	57
3.1.4.1. Archivo de cabecera InputOutput.h.....	57
3.1.4.2. Archivo de cabecera tiny_ble.h.....	57
3.1.4.3. Función main.....	58
3.2. Validación del código.....	61
3.2.1. TeraTerm.....	61
3.2.2. nRF Connect.....	62
3.2.3. Integración del código.....	62
3.2.4. Captura de datos.....	63
3.2.4.1. Captura de datos del sensor de pulso.....	64
3.2.4.2. Captura de datos del acelerómetro.....	66
<b>Capítulo 4. Implementación de la cubierta.....</b>	<b>71</b>
4.1. Introducción.....	71
4.2. SolidEdge, la opción más intuitiva.....	71
4.3. Desarrollo de prototipos.....	73
4.3.1. Desarrollo del primer modelo.....	73
4.3.2. Desarrollo del segundo modelo.....	74
<b>Capítulo 5. Telemedicina: actualidad y futuro.....</b>	<b>79</b>
5.1. ¿Qué es la telemedicina? Estado actual y perspectivas de futuro.....	79
5.1.1. Ventajas y desventajas en pacientes, profesionales y centros de atención.....	81
5.1.2. Aplicaciones de la telemedicina: actualidad y perspectivas de futuro.....	83

5.2. Impacto de la telemedicina en la sociedad.....	85
5.2.1. Resultados de la encuesta.....	85
5.2.2. Conclusión general sobre la encuesta.....	91
<b>Capítulo 6. Conclusiones y trabajo futuro.....</b>	<b>93</b>
<b>Referencias Bibliográficas.....</b>	<b>97</b>
<b>Anexos</b>	
Anexo I.- Cuestionario online sobre telemedicina.....	101



# Capítulo 1

## Introducción

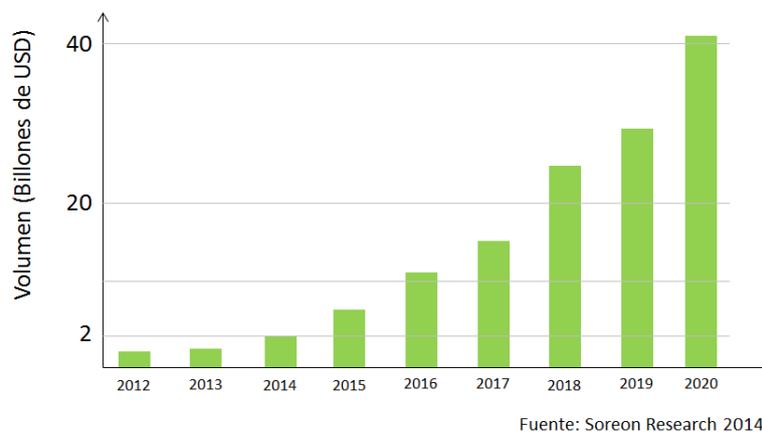
***A lo largo de este capítulo desarrollaremos conceptos básicos para el buen entendimiento de este Trabajo Fin de Grado, los cuales se ampliarán en el resto de Capítulos que componen esta memoria.***

### 1.1. Motivación y objetivos

En los últimos años la tecnología en electrónica y comunicaciones vive una profunda y continua revolución, que ha implicado una evolución transversal en diferentes áreas. Positivamente afectadas por esta evolución, las áreas de la biomedicina y la salud han sufrido importantes cambios, moviéndose de un entorno más hospitalario y local a uno más cercano a cada individuo y global. En la Figura 1.1 se muestra, a modo de ejemplo, la predicción de gasto en aplicaciones versátiles en el entorno de la salud a nivel mundial (no incluyéndose los denominados productos de consumo, como relojes o pulseras, que también pueden albergar aplicaciones adicionales relacionadas con el cuidado de la salud). El mercado se mueve desde los 2 Billones de USD en 2014 a los más de 40 estimados para 2020. Dentro de este segmento del mercado será posible encontrar una gran variedad de instrumentos con un amplio abanico de usos, que podrán ir desde la medición del pulso hasta pulseras inteligentes, capaces de predecir ataques epilépticos.

Nuestro proyecto se incluye en este marco. El objetivo es desarrollar un dispositivo responsable de la monitorización y preprocesado de datos que permiten estimar el pulso cardiaco de una persona. Se trata de una aplicación resuelta ya en el mercado, pero que nos obligará a recorrer personalmente el ciclo de diseño y desarrollo de un sistema de captura de biopotenciales. En términos generales, el pulso cardiaco es una señal médica cuyos parámetros pueden ser medidos, para su captura y envío, mediante el uso de un sensor y un transceptor adecuado. En nuestro caso, ambos procesos vendrán condicionados por su integración en un

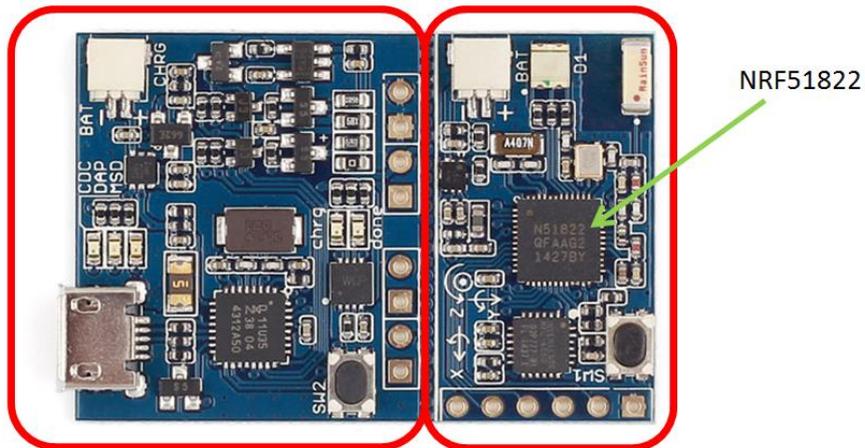
dispositivo *vestible*, lo que obligará a reducir el consumo energético y a emplear una comunicación inalámbrica.



**Figura 1.1.** Estimación de mercado para dispositivos vestibles para el cuidado de la salud (Fuente: Soreon Research 2014)

Perseguiremos a lo largo de esta memoria, el diseño y desarrollo de un dispositivo electromédico empotrable en una pulsera inteligente añadiendo un sensor de pulso basado en fotoplestimografía para poder captar la medida. Esta pulsera, hará uso de servicios Bluetooth de bajo consumo (*Bluetooth Low Energy*, BLE) para el envío de los datos recogidos. La conexión se realizará a un dispositivo móvil inteligente. La pulsera deberá ser ligera, para que no moleste en las actividades cotidianas, además de poseer cierta autonomía.

El elemento central del hardware de la pulsera será el chip de Nordic NRF51822, un ARM Cortex M-0, que programaremos usando el sistema operativo MBED, cuyo entorno de trabajo es proporcionado gratuitamente por ARM TechCon. Para incluir el chip en nuestro diseño emplearemos la placa Tiny-BLE de Seeed Studio. Este *break-out* consta de dos placas unidas por un troquelado (ver Figura 1.2). La placa que se muestra a la derecha sería la que, una vez cargado el fichero .hex con la implementación final del código, se cortaría del resto y formaría parte de nuestra pulsera inalámbrica.



**Figura 1.2.** El entorno Tiny-BLE de Seeed Studio. En la derecha se muestra el break-out que contiene al chip de Nordic NRF51822. A la izquierda se muestra el sistema de carga, que permite conectar la placa al PC usando un cable USB y cargar los ficheros .hex que se ejecutarán en el chip de Nordic.

Como extra al diseño software, y como paso final, haremos uso de un programa de CAD para el diseño de la envoltura de la pulsera.

## 1.2. Fundamentos previo

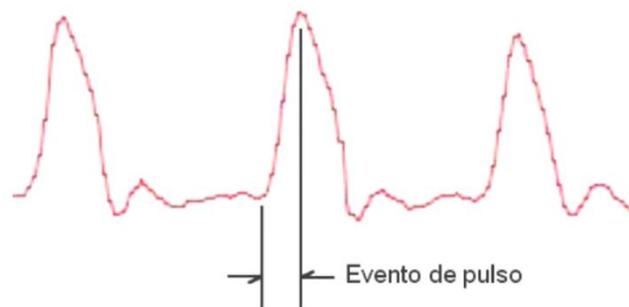
### 1.2.1 La placa Tiny-BLE

La placa Tiny-BLE es una propuesta de Seeed Studio (Seeed), diseñada en torno al chip NRF51822 de Nordic. La placa trabaja con el transceptor montado en este chip, que soporta, entre otras, la tecnología inalámbrica BLE. Como se mencionó en el Apartado 1.1, la placa consta de dos partes: la parte activa incluye el mencionado chip de Nordic, mientras que la otra parte funciona como un cargador, permitiendo cargar en el chip de Nordic el archivo .hex de ejecución. La placa activa puede medir el voltaje de la batería, lo que proporciona información sobre el consumo de energía, y alberga una unidad inercial (la MPU6050), que nos permitiría obtener datos relacionados con el movimiento de la propia placa.

La placa es compatible con el ARM IDE, basado en la plataforma MBED con C/C++ SDK, dando acceso a un gran número de librerías que hace que la creación de prototipos y su desarrollo sea mucho más fácil (Seeed).

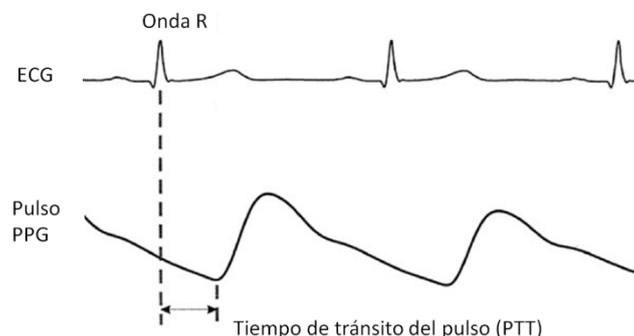
### 1.2.2. El sensor de pulso por fotoplestimografía de pulse\_sensor

Para la adquisición del pulso se empleará un sensor basado en fotoplestimografía. La fotoplestimografía es una técnica no invasiva basada en la medida y análisis de una señal óptica relacionada con los cambios en el volumen sanguíneo. Entre otras aplicaciones, permite la monitorización del pulso cardiaco. Sin entrar en los fundamentos del método, se puede comentar que la señal de pulso que se captura usándolo es una fluctuación analógica en tensión, que presenta una forma de onda en cierta forma predecible, cuyo perfil será similar al mostrado en la Figura 1.3. De hecho, es esa representación la que se denomina fotoplestisgrama o PPG.



**Figura 1.3.** Forma del pulso en un PPG

El pulso de la Figura 1.3 no se corresponderá, con exactitud con el pulso (pico de la onda R) del electrocardiograma (ECG), existiendo una diferencia de tiempo entre ambos debido a la transmisión de la señal por el sistema circulatorio (Tiempo de tránsito de pulso, PTT, en la Figura 1.4).



**Figura 1.4.** ECG vs. PPG (J.Allen, A.Murray, 2002)

Existen dos formas de capturar el PPG. En la configuración por transmisión el emisor y el receptor se colocan de forma que están enfrentados. Sólo se puede

aplicar en zonas estrechas, periféricas y muy específicas. La señal adquirida es normalmente más limpia, esto es, permite identificar mejor el pulso asociado al pico de la onda R. En reflexión, emisor y receptor se colocan sobre la misma superficie. La señal es más débil y existe el riesgo de que el receptor detecte directamente parte de la luz emitida por el emisor. Esto dificultará el diseño y obligará a trabajar un mejor acondicionamiento (es decir, todo el circuito o software dedicado a conseguir una señal adecuada, donde los pulsos se identifiquen claramente).

Nosotros haremos uso del diseño en reflexión, para lo cual se emitirá luz desde un diodo emisor y se hará uso de un fotodetector para recibir la luz reflejada por el tejido. Conforme el flujo de sangre cutáneo aumenta también lo hará la cantidad de luz recibida. La idea es medir el tiempo entre picos, denominado intervalo entre pulsaciones (IBI, *Inter Beat Interval*). En la Figura 1.5, se muestra el sensor empleado en nuestro proyecto, fabricado por pulse\_sensor (Pulse sensor).



**Figura 1.5.** El sensor de pulso de *pulse\_sensor*

### 1.3. Trabajos previos

El presente Trabajo Fin de Grado se ha construido sobre los resultados del proyecto ASIST-E, en el que la Universidad de Málaga fue contratada por la empresa MAGTEL para el diseño de una pulsera inteligente. La pulsera diseñada en este proyecto incluía sensores de pulso y temperatura, así como de estimación de la actividad física de la persona y, adicionalmente, el uso de un sensor GPS para poder seguir su movimiento. El sistema permitía detectar caídas.

La propuesta era compleja y, en algunos casos, los datos proporcionados no eran robustos (era, por ejemplo, el caso de la temperatura o de la detección de caídas).

Por ello, se propuso la simplificación del diseño pues, por ejemplo, la función de seguimiento podía usarse empleando el propio *smartphone* (presente como cliente de los datos proporcionados por la pulsera). De forma similar, los algoritmos de detección de caídas podían emplear los datos de los acelerómetros presentes en el *smartphone*. La idea de simplificar la pulsera para que solo enviara la información de pulso dio lugar al presente Trabajo Fin de Grado. Esto ha supuesto personalmente el analizar y comprender cómo funcionaba ASIST-E, el personalizar el software, eliminando todo aquello que resultaba innecesario, y, finalmente, adaptarlo a nuestro problema concreto (por ejemplo, el envío de datos por BLE mezclaba datos de pulso con otros parámetros, en un servicio a medida, que ha sido sustituido por el más simple servicio estándar BLE del *Heart Rate*).

#### **1.4. Metodología y organización de la memoria.**

Para la realización del presente Trabajo Fin de Grado seguiremos un esquema de trabajo que se divide en cuatro fases fundamentales:

- Documentación y recopilación de información. En esta fase se hará una profundización en el tema y se capturarán los requisitos de nuestro diseño.
- Diseño del hardware y del código de nuestro proyecto. Esta fase es la que tiene previsto una mayor duración debido a su complejidad: el diseño del hardware se puede catalogar como un proceso de selección, resultado del cual son los elementos presentados en el Apartado 1.2. El código exige un trabajo de desarrollo en el cual se configure correctamente el chip de Nordic, se configure y conecte el sensor de pulso, y se configuren los servicios BLE que necesite nuestra aplicación.
- Comprobación del diseño. Esta será la tercera fase, donde se verificará el cumplimiento de cada uno de los requisitos capturados, llevando a la práctica las pruebas previstas inicialmente.
- Validación del prototipo y redacción de la memoria. Esta será la cuarta y última fase de este proyecto. Pese a especificarse como cuarta fase, esta redacción no se deja, sin embargo, para el final del proyecto, sino que se entiende como un portafolio, que se irá completando conforme se avance en el proyecto.

Por otra parte, la organización de la memoria vendrá dada en capítulos, comenzando con éste como introducción, y desarrollándose posteriormente el resto para la buena comprensión y entendimiento de cada una de las técnicas que se van a usar para su desarrollo.



## Capítulo 2

### Conceptos básicos

***A lo largo de este capítulo se desarrollarán tanto los conceptos básicos, necesarios para el entendimiento de este Trabajo Fin de Grado, como la explicación de cada uno de los elementos empleados y sus fundamentos***

#### **2.1. Pulsioximetría basada en fotopletismografía.**

La pletismografía es una técnica óptica no invasiva, es decir, que no perturba el comportamiento de la variable que estamos midiendo. De forma muy resumida, su objetivo es detectar los cambios de volumen de sangre en la capa microvascular. La onda asociada a estos cambios, originada en los propios latidos del corazón, puede estimarse inicialmente empleando un método pulsátil. Junto a esta componente de mayor fuerza, esta onda tendrá asociadas componentes de baja frecuencia, relacionadas con el sistema respiratorio, el sistema nervioso simpático y los sistemas de termorregulación.

En fotopletismografía, esta onda se trata de estimar usando las variaciones cromáticas internas que provoca el flujo sanguíneo. Así, este tipo de técnica necesita pocos componentes para su funcionamiento: solo es necesario una fuente de luz de un color determinado, la cual es aplicada sobre la piel, y un fotodetector, encargado de medir las variaciones de intensidad de luz provocadas por el flujo sanguíneo, es decir, por el cambio de volumen de sangre que pasa por la zona en la que se está aplicando la iluminación. La sangre, en su movimiento por esta zona, absorberá más o menos la iluminación aplicada. Y, por ello, el fotodetector detectará más o menos iluminación. Este esquema se puede aplicar de dos formas distintas:

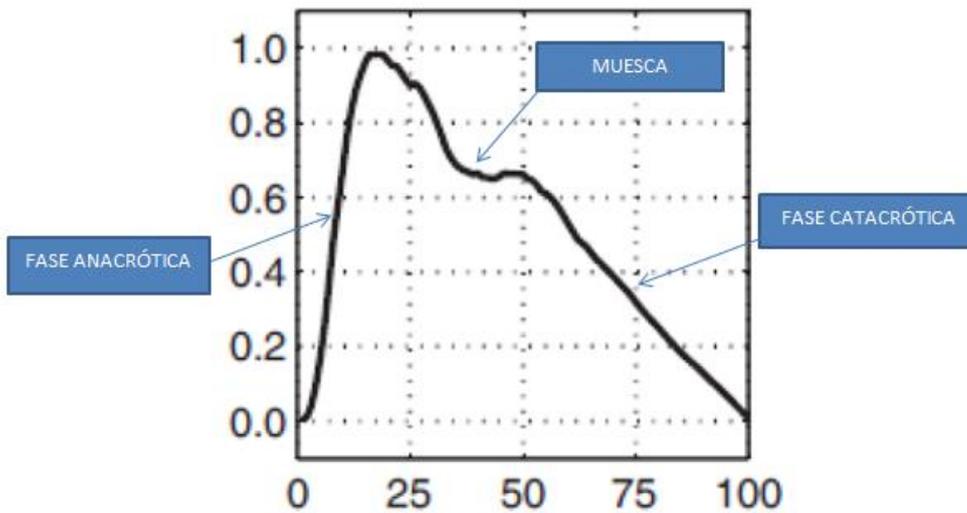
- Por transmisión. En este primer caso el tejido se encontrará entre la fuente de luz, la cual es típicamente un diodo LED, y el fotodetector. En este caso, la iluminación se transmite por el tejido y, el paso del flujo sanguíneo, hace que se absorba más o menos iluminación. El fotodetector extrae por ello una información directa de la iluminación absorbida.

- Por reflexión. En este caso encontraremos la fuente y el detector uno junto al otro, estudiándose la variación de la luz reflejada en el tejido. Nuevamente, el cambio de flujo sanguíneo provoca una mayor absorción de iluminación, y de esta fuente se extrae la información. El esquema es más ruidoso, pues la propia iluminación del diodo LED puede alcanzar al fotodetector. Sin embargo, puede aplicarse sobre la piel, en principio, en cualquier parte del cuerpo de la persona, y no en regiones como yemas de dedos u oreja. En este proyecto el sensor con el que se trabajará hace uso de este tipo de tecnología.

Se debe de tener en cuenta que se trata de una técnica óptica aplicada sobre tejidos vivos y que, por lo tanto, pueden darse fenómenos relacionados con este tipo de técnicas como son la dispersión o reflexión de luz. Por otro lado, se debe saber que la cantidad de luz que registra el detector puede verse afectado por distintos factores como, por ejemplo, el movimiento de las paredes arteriales o la posición en el que se encuentra el conjunto emisor-receptor con respecto a la zona en la que se realiza el estudio. Otro concepto de interés es lo que se conoce como ventana óptica, que varía con la longitud de onda de radiación. Por ventana óptica se define el margen de longitudes de onda para los cuales se proporciona una iluminación prácticamente transparente ante la interacción con los tejidos en seres vivos (López Silva, S.M. et al., 2005).

Analizando un poco la onda que proporciona la fotoplestimografía, podemos diferenciar dos zonas. La primera de ellas es la fase anacrótica, que se relaciona con la sístole o de subida de pulso. La segunda fase se le conoce como catacrótica y se relaciona con la diástole o flanco de bajada. En esta última fase es en la que se detecta una pequeña muesca, la cual aparece en aquellos sujetos sometidos al estudio cuyas arterias son saludables.

En la Figura 2.1 se puede observar lo que sería una representación de las anteriores fases nombradas.



**Figura 2.1.** Forma general que tiene una onda de fotoplestígrafía (Barea Cañizares, A., 2016).

Esta onda va variando conforme se desplaza hacia la periferia. En las aplicaciones médicas que hacen uso de ella es habitual tomar medidas bilaterales, es decir, tomar dos medidas simultáneas en la misma zona del cuerpo, por ejemplo, en el mismo dedo de la mano derecha e izquierda a la vez con el objetivo de comparar las ondas y poder dar un diagnóstico más preciso. También existen medidas multi-bilaterales, tomando medidas simultáneas en ambos lados y en varias zonas.

## 2.2. Arquitectura hardware del sistema de captura

Nuestra pulsera inteligente contará con dos elementos principales: un sensor de pulso basado en fotoplestígrafía y la placa Tiny-BLE, en la cual se incorpora el microprocesador en el cual se ejecutará nuestro programa.

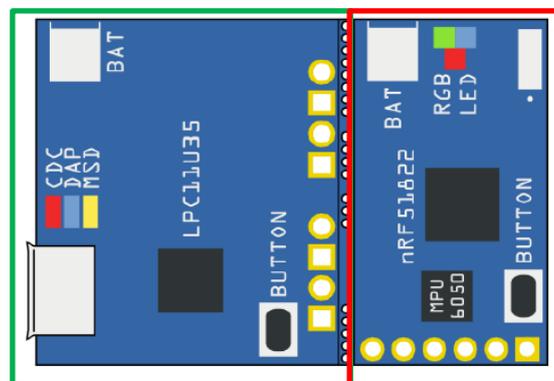


**Figura 2.2.** Principales componentes de la pulsera inteligente

### 2.2.1. La placa Tiny-BLE

La placa Tiny-BLE es una tarjeta de desarrollo que incluye el chip de Nordic nRF51822. Este chip dispone de un transceptor inalámbrico, con capacidad para trabajar con el estándar de Bluetooth de baja energía (*Bluetooth Low Energy*, BLE). Según la firma Seeed Studio, que la comercializa, la placa fue creada para proporcionar una solución al Internet de las Cosas basado en BLE, y está caracterizada por su bajo consumo y su reducido tamaño. Además de esto tiene la capacidad de proporcionar datos sobre su consumo en tiempo real, algo de gran utilidad en el desarrollo de un software puesto que siempre se busca la optimización y mejora, de tal manera que se implementen dispositivos con una larga vida útil de batería.

La placa en sí se puede dividir en dos partes bien diferenciadas. El núcleo de la parte BLE es el mencionado chip de Nordic (nRF51822). Una vez que nuestro programa esté cargado en el ARM, será esta parte de la placa la que incluyamos en nuestro prototipo final. Además en esta primera parte, la cual podemos observar enmarcada en rojo en la Figura 2.3, se incluyen también una antena, tres diodos leds, la Unidad Inercial (IMU) MPU6050, un pulsador y un conector para la batería de alimentación externa. En la segunda parte de la placa, la cual podemos observar enmarcada en verde en la Figura 2.3, encontramos un microcontrolador LPC1114, que es el encargado de controlar la interfaz de entrada al chip de Nordic, permitiendo cargar archivos .hex en él. También se incluye en esta parte, el conector USB, por el que conectamos la Tiny-BLE con el entorno del PC (Seeed).



**Figura 2.3.** División de Tiny-BLE en dos partes. En rojo está señalada la parte activa, que entra dentro de nuestro diseño y donde se incluye el chip de Nordic. En

verde podemos observar el segundo bloque de la placa, donde se encuentra el LPC que funciona como cargador.

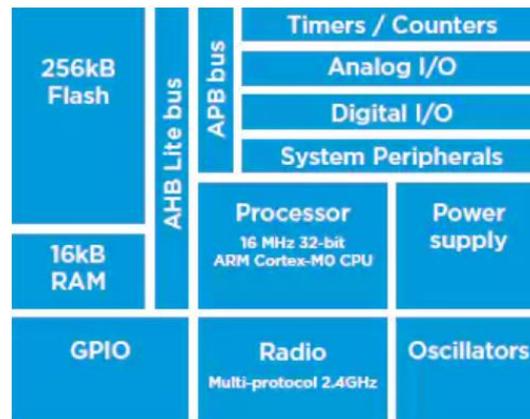
Nuestra forma de trabajar con la placa será conectándola por USB al PC, donde desarrollaremos nuestro programa y, tal y como se ha explicado anteriormente, una vez generemos el fichero con extensión .hex, lo cargaremos en el nRF51822 a través del LPC.

A continuación se analiza con más detalle el chip nRF51822, que constituye el elemento central en nuestro sistema.

### **2.2.2. El Chip de Nordic nRF51822**

El chip nRF51822 de Nordic, al igual que todos los de la serie nRF51, contienen una serie de elementos comunes como son el control de reloj, la alimentación, el reinicio o la interconexión periférica programable. Si éstas son características comunes de esta serie de Nordic, nosotros nos centraremos en los principales bloques que componen nuestro chip y sus características específicas.

En primer lugar debemos analizar las características principales de su procesador, el cual podemos observar en la Figura 2.4, en la parte central de la imagen. En concreto, este procesador se organiza con una arquitectura ARM Cortex-M0. Respecto a lo que se refiere a memoria, todos los registros se encuentran en el mismo espacio de direcciones, en una memoria tipo Flash. Existen regiones de la memoria flash que se pueden proteger de lectura, escritura y borrado mediante la unidad de protección de memoria o MPU. Esta MPU se puede configurar para proteger toda la memoria flash del dispositivo del modo lectura, o proteger la memoria por bloques según lo que se quiera. Adicionalmente se dispone de una memoria RAM. Desde la página de Nordic te dan una serie de recomendaciones para poder aprovechar al máximo la memoria RAM.



**Figura 2.4.** Diagrama de bloques del chip nRF51822

En lo que respecta al suministro de energía, bloque Power supply en la Figura 2.4, admite tres configuraciones:

- Configuración interna LDO. En este caso se genera la potencia del sistema directamente desde el voltaje de suministro.
- Configuración del convertidor DC/DC. Transforma el voltaje de la batería reduciendo así el voltaje interno y produciendo una pérdida de potencia mínima.
- Configuración de modo de baja tensión. Para dispositivos que usan un bajo voltaje, siendo este constante a 1.8V.

Por último se debe destacar el bloque que permite la interconexión periférica programable, también conocida como PPI, y que, en la Figura 2.4, podemos identificar con el bloque *System Peripherals*. Este bloque permite a los periféricos interactuar usando tareas y eventos de forma independiente a la CPU (Nordic Semiconductor ASA, 2014).

### 2.2.3. El sensor de pulso

El sensor de pulso empleado en nuestro diseño se muestra en la Figura 2.5. Es un sensor de pulso comercializado por pulse\_sensor. La medida se basa en la técnica de fotoplestimografía por reflexión, lo que permite usarlo en cualquier zona. Como desventaja su señal será más débil que aquellos que usan técnicas de transmisión. En concreto, el sensor es conocido comercialmente como *Pulse Sensor Amped* y se basa en tres componentes:

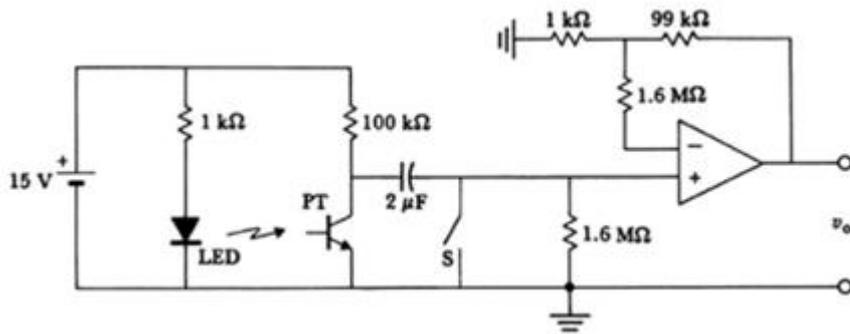
- El fotodetector de Avago APDS-9008.
- El fotodiodo de Kingbright AM2520ZGC09.
- El amplificador operacional MCP6001 de Microchip Technology.



**Figura 2.5.** Sensor para medir el pulso por PPG de pulsesensor. En esta imagen se puede apreciar que el emisor y el detector se encuentran en configuración de reflexión.

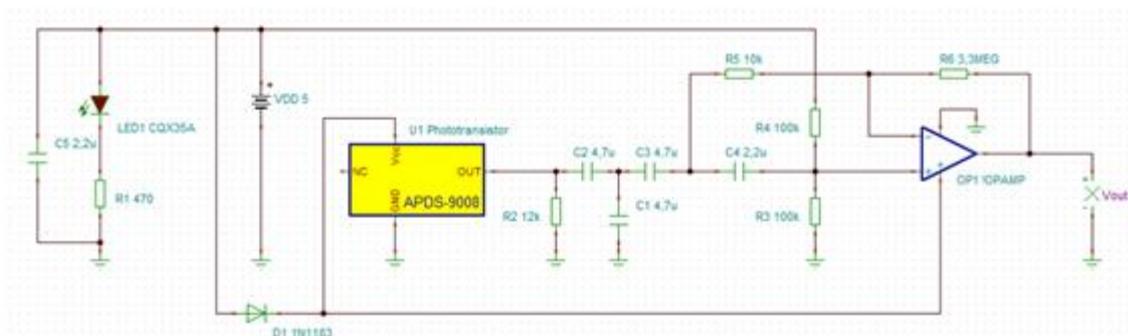
Como se describe en el Apartado 2.1, la técnica PPG se basa en el distinto comportamiento exhibido por la sangre ante una determinada iluminación. Así, el propio color de la sangre se debe a que refleja el color rojo y, por lo tanto, absorbe la luz verde. Por ello, este sensor hará uso de un diodo de luz verde para iluminar la zona de piel usada para medir el pulso, y se hará uso de fotodiodos que puedan evaluar la luz reflejada. Cuanta más sangre bombee el corazón, mayor será la cantidad que circule por las venas y, por lo tanto, la luz verde que será absorbida. El sensor lo que hace es lanzar varias ráfagas de luz por segundo, a la vez que capta la cantidad de luz que se ha absorbido. Es por esto, que se puede saber con bastante precisión cuantas veces por segundo se está bombeando la sangre, es decir, cuantas pulsaciones por minuto tiene el corazón.

La señal capturada por el fotodetector es muy débil y puede estar afectada por ruidos. Por ello, una vez que se obtiene la señal, el sistema deberá amplificarla, invertirla y filtrarla. Esta señal de salida será enviada al chip de Nordic, que dispone de conversores A/D (analógico/digital) para recibir señales analógicas. Dado que estos conversores necesitan recibir una señal en tensión, y nuestro fotodetector proporciona sin embargo una señal de corriente, la etapa de amplificación deberá además transformar esta corriente en tensión. Esto es, debe funcionar como amplificador de transimpedancia (Figura 2.6).



**Figura 2.6.** Amplificador de transimpedancia en configuración no inversora.

Sobre la etapa de amplificación se añadirán los elementos que permiten filtrar las señales generadas. Para ello se deberá tener en cuenta que el objetivo es medir pulso cardíaco, por lo que la señal deseada estará entre los 60-120 lpm (esto es, 1-3 Hz). Fuera de este rango es interesante que la señal sea anulada pues no corresponde a medida del pulso. Como muestra la Figura 2.7, el sensor de pulso lo que hace es añadir unos condensadores antes del amplificador operacional.



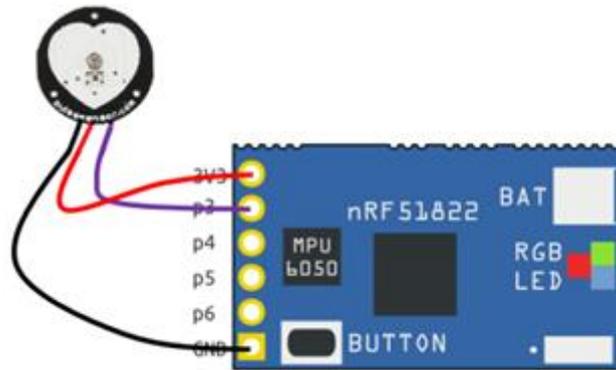
**Figura 2.7.** Esquema del sensor de pulso.

Se debe de tener en cuenta la calibración, puesto que una mala calibración junto con otra serie de artefactos, como el movimiento del sensor, puede dar lugar a una mala recogida de datos y por lo tanto a su mala interpretación (Pulse sensor).

#### 2.2.4. Conexión del sensor de pulso al chip de Nordic

El sensor ira conectado al chip de Nordic ubicado en la Tiny-BLE tal y como se muestra en la Figura 2.8. Básicamente, lo que hacemos es conectarlo al puerto de

entrada p3, que dispone del mencionado convertor A/D. El sensor se alimenta también desde la Tiny-BLE (cables rojo y negro, a 3,3V y GND, respectivamente).



**Figura 2.8.** Esquema de la conexión entre nuestra placa y el sensor de pulso

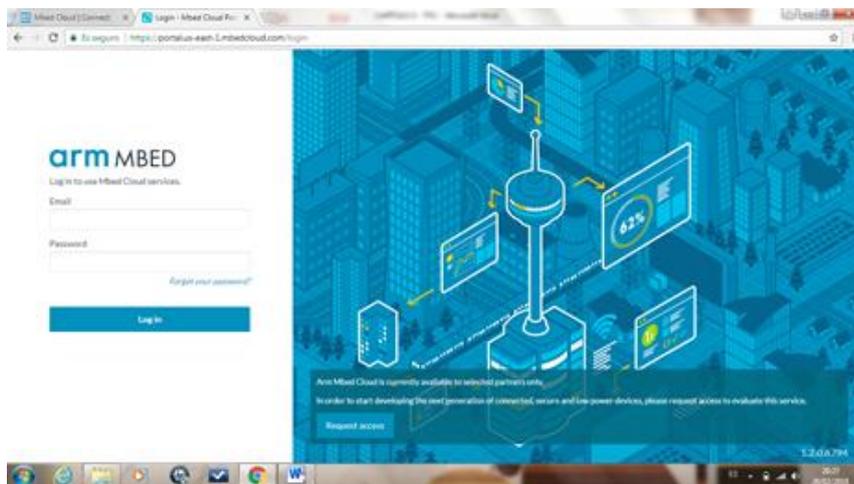
## 2.3. Diseño software

### 2.3.1. La Plataforma MBED

La plataforma MBED es proporcionada por ARM, y será con la que trabajaremos a lo largo de este proyecto. Es una plataforma orientada al diseño y desarrollo de sistemas basados en dispositivos ARM Cortex, lo que permite la creación rápida de nuevos productos o dispositivos que añaden, a productos ya cotidianos, nuevas capacidades. Junto al abaratamiento de la tecnología, y el que cada vez más personas tengan conocimientos suficientes para hacer uso de éstas, estas plataformas son, en parte, responsables de la popularización de la denominada internet de las cosas. Así, hasta hace unos años, el desarrollo tecnológico solo se centraba en centros I+D. Hoy en día el avance tecnológico surge frecuentemente de pequeñas *start-ups*. El objetivo es la creación rápida de prototipos, lo que es posible, al no ser difícil acceder a dispositivos y plataformas de trabajo, si se dispone de una cierta base electrónica y de programación. Una de las plataformas que hoy en día permite crear y desarrollar este tipo de proyectos es MBED.

Aunque están disponibles versiones para trabajar de forma local, la plataforma MBED se caracteriza por funcionar de forma online. Para su desarrollo, ARM se asocia con NXP. Su gran ventaja es, por tanto, que no requiere de instalaciones o descargas, y es accesible desde cualquier sistema operativo. Para su uso solo es necesario tener conexión a internet y la creación de una cuenta de acceso.

Su desarrollo ha sido rápido y está en continuo crecimiento debido a su sencillez para crear programas sin necesidad de usar otro software adicional. Simplemente debemos de compilar nuestro programa directamente en la nube y generar un ejecutable que, por último, se enviará al dispositivo encargado de ejecutarlo. En el caso de la placa Tiny-BLE, conectada mediante el puerto USB a nuestro ordenador y dotada de un sistema de carga, basta con copiar el ejecutable (de extensión .hex) en la carpeta que, en el ordenador, se asocia a la placa.



**Figura 2.9.** Página principal de MBED

Además de las nombradas anteriormente, MBED también ofrece otras facilidades a la hora de programar, puesto que disponemos tanto de librerías como de ejemplos, que permiten que, en el transcurso de poco tiempo, se adquieran conocimientos básicos. Dichas librerías y códigos están disponibles de forma online y gratuita, ofreciendo funciones que nos permiten comenzar a trabajar con nuestros dispositivos. Para que esta plataforma pueda funcionar correctamente incluye:

- C/C++ Software Development Kit (SDK). Permite un desarrollo rápido de aplicaciones a partir de las librerías creadas por otros usuarios.
- HDK. Es usado por el microcontrolador y da soporte a distintas placas.
- Compilador y herramientas online para un desarrollo comunitario, permitiendo que te puedas conectar desde cualquier parte sin la instalación de un software en un dispositivo.

Como conclusión, podríamos decir que vamos a hacer uso de una plataforma que nos va a permitir desarrollar nuestro prototipo de una forma fácil y rápida. Además, de permitirnos trabajar con una placa la cual lleva un ARM Cortex-M integrado como es la Tiny-BLE( Macias Macias, M. et al. , 2014).

## **2.4. Comunicaciones Bluetooth**

### **2.4.1. Introducción a la tecnología bluetooth**

En 1994 comienza la investigación para la creación de interfaces que hicieran uso de radio de baja frecuencia, y con un coste mínimo, entre teléfonos y sus diferentes accesorios. Con este objetivo nace SIG Bluetooth, un conjunto de empresas que crearán el primer estándar relacionado con esta tecnología. En principio se trataba de un estándar global, que recogía requerimientos, tanto de hardware como de software, para hacer uso de comunicaciones inalámbricas para transmitir tanto voz como datos entre diferentes dispositivos. Sus tres objetivos principales eran: facilitar la comunicación, eliminar cables, y tener la posibilidad de crear redes inalámbricas facilitando la sincronización de datos entre equipos.

La tecnología Bluetooth trabaja en la banda de radio ISM, siglas que identifican aplicaciones en el ámbito *Industrial, Scientific, Medical*. El uso de esta banda hace que se pueda utilizar en cualquier parte del mundo, permitiendo que los dispositivos puedan enlazar con otros similares sin importar donde nos encontremos.

Esto da lugar a un gran salto en lo que se refiere a gestión de información, ya que se da acceso e intercambio de una forma remota. La tecnología inalámbrica desde su creación siempre ha estado en continua evolución, dando lugar a nuevos estándares, diseñados para la comunicación a media y corta distancia, y buscando el bajo coste y consumo.

CARACTERÍSTICAS/FUNCIÓN	FUNCIONAMIENTO
Tipo de conexión	Expansión de espectro (saltos de frecuencia)
Espectro	Banda ISM de 2,4 GHz.
Potencia de transmisión	1 milivatio (mw)
Velocidad de datos total	1 Mbps utilizando saltos de frecuencia
Alcance	Hasta 100 m
Estaciones soportadas	Hasta ocho dispositivos por picorred
Canales de voz	Hasta tres
Seguridad de datos	Para autenticación, una clave de 128 bits; para cifrado, el tamaño de la clave es configurable de 8 a 128 bits
Direccionamiento	Cada dispositivo tiene una dirección (MAC) de 48 bits que se utiliza para establecer una conexión con otro dispositivo

**Figura 2.10.** Resumen de las principales características de Bluetooth

Otras características destacables son la inmunidad frente a interferencias, debido a los saltos de frecuencias, o la seguridad que da el fabricante respecto a esta tecnología, puesto que tiene que demostrar que su producto cumple el estándar Bluetooth.

#### **2.4.2. Especificaciones: protocolos y perfiles**

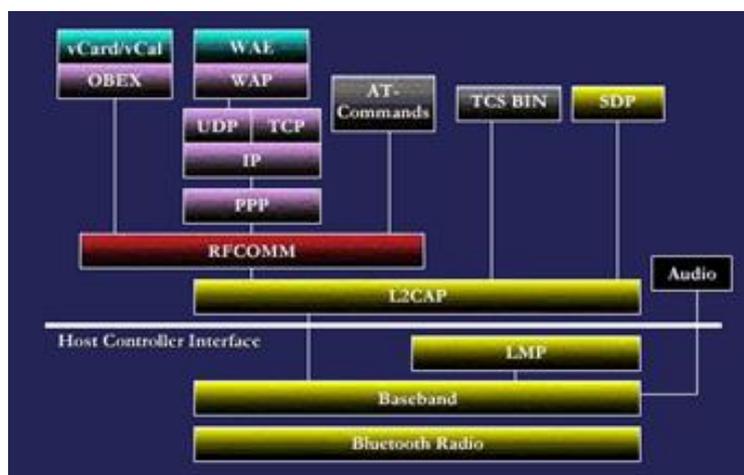
Uno de los puntos más importantes para entender el funcionamiento de la tecnología BLE son las especificaciones, puesto que definen el comportamiento inalámbrico que asegura la compatibilidad entre dispositivos que hagan uso de esta tecnología, definiéndose desde el nivel de radio hasta el nivel de aplicación. Dividiremos las especificaciones en dos bloques, encuadrando en el primero protocolos y, en el segundo, los perfiles, esto es, los modelos de uso donde se describe como las distintas aplicaciones deben de hacer uso de los protocolos.

Algunos de los protocolos más importantes de los que hace uso Bluetooth son los siguientes, los cuales están acompañados de una breve explicación para entender que se pretende con cada uno:

- Link Manager Protocol (LMP). Encargado de la autenticación, encriptación, control y configuración del enlace.
- Host Controller Interface (HCI). Da una interfaz permitiendo acceder a recursos del hardware.

- Logical Link Control and Adaptation (L2CAP). Correspondiente a la capa de enlace, brinda servicios a datos orientados y no orientados a conexión.
- Service Discovery Protocol (SDP). Actúa como una aplicación de un cliente, con la misión de descubrir servicios disponibles.
- RFCOMM. Este protocolo ofrece emulación del puerto serie sobre el protocolo.

Además de estos protocolos, las especificaciones tienen otros adicionales como son el protocolo punto a punto, o protocolos como UDP/TCP-IP.



**Figura 2.11.** Pila de protocolos de Bluetooth. Desde abajo hacia arriba, avanzamos desde el nivel de radio al nivel de aplicación.

Como es sabido, esta tecnología está diseñada para que sea usada por un gran número de usuarios y, por lo tanto, para que su implementación fuera ilimitada. Para ello, se definieron modelos de usuarios y perfiles de protocolo, definiéndose a cada perfil con una selección de mensajes y especificaciones de Bluetooth, y dando lugar a la una descripción clara de la interfaz.

Existen cuatro perfiles generales: de acceso (GAP), perfil de aplicación de descubrimiento de servicios (SDAP), de puerto serie (SPP) y de intercambio de objetos (GOEP). Una breve explicación de cada uno de ellos sería la siguiente:

- GAP. Encargado del descubrimiento y establecimiento de conexiones.
- SPP. Define los requerimientos necesarios para establecer una conexión haciendo uso del protocolo anteriormente nombrado como RFCOMM.

- SDAP. Se definen los procedimientos para una aplicación con tecnología bluetooth, donde deseamos recuperar información de servicios relacionados con servicios en otros dispositivos.
- GOEP. Ofrece características de intercambios de objetos, como por ejemplo la transferencia de archivos.

Para ver las mejoras que incluye esta tecnología, la podemos comparar con alguna más antigua como es el infrarrojo, el cual se basaba en una conexión óptica mediante luz infrarroja. Su principal desventaja era que se necesita una trayectoria clara y directa entre los dispositivos que se quieran conectar, que no existiera interferencias entre ambos dispositivos y solo funcionaba en distancias muy cortas. Todas estas desventajas se han podido solventar con la tecnología Bluetooth, cubriendo así aplicaciones en la que el infrarrojo no era eficiente.

Como resumen, podemos decir que Bluetooth es una herramienta para conexiones inalámbricas económica, fiable y de fácil uso, además de que se puede incluir en dispositivos ya existentes.

### **2.4.3. Tecnología BLE**

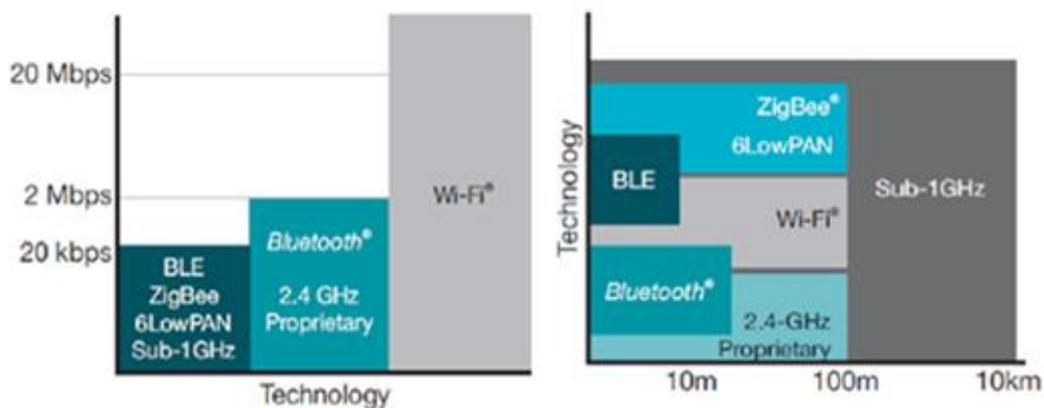
Bluetooth de baja energía (BLE) es una tecnología complementaria a Bluetooth, que garantiza unos niveles de bajo consumo y tiempo de establecimiento de conexión menor. Aunque, al igual que Bluetooth clásico, hace uso de la misma banda de frecuencias y tienen varias similitudes, BLE es considerada un nuevo estándar con unas aplicaciones y objetivos diferentes a los que tenía Bluetooth.

Su diseño tiene como objetivo la transmisión de pequeñas cantidades de datos, y no está pensado para mantener conexiones con otros dispositivos durante un largo periodo de tiempo. Esto da lugar a un bajo consumo, puesto que solo se encontrarán activos cuando se les pide una transmisión de datos. De esta mentalidad nacen aplicaciones de gran utilidad en distintas áreas, entre las que se encuentra el ámbito de la medicina.

Las características de BLE está unas ligadas con otras, es decir, la tasa de envío de datos está ligada con el consumo de energía del dispositivo y, por lo tanto, lo está con la alimentación externa necesaria. La nueva pila de protocolos que incluye BLE se puede dividir en tres grandes grupos:

- Aplicación: donde se cubre determinados casos de uso.
- Host: capas superiores de la pila de protocolos.
- Controlador: capas inferiores de la pila de protocolos.

A la hora de usar estas tecnologías se debe de tener en cuenta ciertos aspectos que pueden limitar la creación de proyectos como por ejemplo el rango de cobertura ya que esta es de 10 metros. Otro aspecto nombrado anteriormente es la tasa de transmisión de datos, que es baja, enviando pocas cantidades de datos.



**Figura 2.12.** En la gráfica de la izquierda se puede observar la comparativa de tasa de envío de datos de diferentes tecnologías, mientras que en la gráfica de la derecha se puede observar una comparativa del rango de cobertura que tienen las distintas tecnologías inalámbricas.

Hablando sobre la arquitectura de esta tecnología, podemos decir que la topología de la red de BLE tiene una configuración de estrella, donde los dispositivos denominados como Master pueden tener varias conexiones de capa de enlace con periféricos y realizar búsquedas de dispositivos. En los extremos de la estrella nos encontramos a aquellos que tienen el rol de esclavo, los cuales solo podrán establecer una única conexión en la capa de enlace con un Master.

La capa de enlace, nombrada en el párrafo anterior, la podemos definir, a grandes rasgos, como la responsable de la estructura de los paquetes (Garijo L., 2016). Como cierre de este punto podemos definir cuatro puntos principales que determinan la elección de este tipo de tecnología:

- Bajo consumo de potencia.
- Fácil de implementar y de instalar. Se trata de dispositivos de reducido tamaño, configurables, y a los que se les puede instalar una serie de accesorios según nuestras necesidades.
- Soportado por Smartphones. Ya que nuestro proyecto para poder implementarlo se debe de conectar a uno de estos dispositivos para la transmisión de datos.
- Larga vida de batería. Esto reduce en gran medida el coste de mantenimiento.

#### **2.4.4. Servicios Bluetooth: GATT y UUID**

La tecnología inalámbrica da lugar a la conexión de una gran variedad de dispositivos, lo que hace posible la monitorización y obtención de datos de interés. Por lo tanto, se obtiene información mediante la interacción con diferentes componentes sin intervención humana. Para esto BLE hace uso de servicios, perfiles y protocolos.

Esto podría permitir grandes avances en la ingeniería biomédica influyendo en el ámbito, como es el caso de este proyecto, de monitorización de constantes. La comunicación de la pulsera con el dispositivo, el cual recibe la información, se realizará mediante la creación de una pequeña red inalámbrica, lo que permitirá el envío y recepción de datos e información.

Dicha transmisión se realizará mediante el uso de perfiles, los cuales describen aspectos relacionados con la comunicación. Por lo tanto, para que se establezca una conexión entre dos dispositivos, ambos deben de estar haciendo uso del mismo perfil.

El perfil de BLE es *Generic Attribute Profile* más conocido como GATT, en él se define que dos dispositivos que hagan uso de ella pueden tanto enviar como recibir datos además de realizar las operaciones básicas necesarias para que esto ocurra. Este perfil a su vez se compone de varios servicios que suelen ser datos básicos que proporciona el dispositivo con el que se quiere crear la conexión.

Los servicios usados se pueden distinguir ya que cada uno de ellos hace uso de un UUID, es decir, un identificador único universal. Dentro de UUID tenemos dos divisiones: en primer lugar, aquellos que están estandarizados, compuestos por un total de 16 bits, y en segundo lugar aquellos que son propios del fabricante, cuyos UUID se componen de un total de 128 bits. Por lo tanto, en cada servicio nos encontramos su definición y su identificación.

GATT está compuesto por los datos que se envían o reciben mediante un dispositivo BLE, a esto se le denomina característica. Una característica, se puede definir como los datos usados por un servicio, además de la información sobre las operaciones básicas que puede realizar los datos transmitidos. La conexión entre dispositivos se realiza mediante una relación cliente-servidor y atiende al protocolo de conexión Attribute Protocol, más conocido como ATT, el cual determina la forma de transferir las características de los datos.

Un dispositivo debe actuar como servidor GATT, donde se almacena información junto con sus servicios y características. Si lo vemos paso a paso, lo primero que se hace es la conexión entre los dispositivos que se vayan a usar, realizándose así una unión entre un cliente GATT y un servidor GATT. En este momento, el servidor se hace invisible para otros clientes, impidiendo que se conecte otro cliente. Lo segundo que observaremos serán los servicios que ofrece nuestro servidor, y, seguidamente, será posible el envío y recepción de datos. El último paso es la desconexión entre dispositivos, dejando libre el servidor GATT para que otros dispositivos puedan hacer uso de éste (Tintín Durán, 2015).

En nuestro caso, tendremos una aplicación instalada en un *smarthpone*, el cual se encarga de descubrir y comunicarse con otros dispositivos haciendo uso de la

tecnología BLE. Por lo tanto, deberemos de implementar una solución para que esta conexión entre el dispositivo móvil y la pulsera inteligente sea posible. Todo ello se analiza ya en el Capítulo 3 de la presente memoria.

## Capítulo 3.

### Sistema de captura y verificación de datos.

**Este Capítulo introduce la estructura de nuestro código, detallando cada proceso que se realiza, librerías empleadas... etc. Además se realiza la validación del código y su correspondiente verificación mediante la captura de datos.**

#### 3.1. Organización del código.

En este Apartado se analiza el código desarrollado. En este proyecto se ha trabajado en Lenguaje C/C++, y se desarrollara al completo en la plataforma MBED, mencionada en el Capítulo 2. Por ello, como se describe a lo largo de este Capítulo, se hará uso intensivo de sus librerías. También es importante destacar que el diseño se enlazará por completo a la placa concreta empleada (en nuestro caso la Tiny-BLE). Eso obliga a usar ficheros de configuración concretos y a enlazar determinadas entradas o salidas a pines concretos de la placa. Una vez que se implementó el software asociado a la detección de pulso, y para aprovechar la existencia de la unidad inercial MPU6050 en la Tiny-BLE, se ha diseñado e implementado también un código para detectar que se da un pequeño golpecito a la placa (*tap*) que emplea esta unidad inercial, y permite sustituir la pulsación del botón en la Tiny-BLE. Ambas funciones se analizan a continuación por separado, en los Apartados 3.1.1 y 3.1.2 respectivamente, para finalmente unir las en un único programa principal.

##### 3.1.1. Captura de la señal de pulso

###### 3.1.1.1. Variables e interrupciones

El algoritmo para capturar la señal de pulso se implementa en el fichero `pulse.cpp`. Además de incluir la cabecera necesaria de MBED, lo primero que se han realizado son una serie de asignaciones. En la Figura 3.1 se muestran cuatro variables globales. En la variable *millis* se almacenan las decenas de milisegundos entre pulsaciones. El pin p22 está físicamente conectado al led rojo, con el objetivo de que

su iluminación intermitente proporcione información visual sobre el ritmo del pulso. El led se conecta a la variable *red* (una salida digital). También como salida digital, el pin p30 habilita la alimentación externa del sensor de pulso (*HeartLight*). Como entrada analógica, el pin p3 de la Tiny-BLE se emplea para recoger la señal de pulso aún sin procesar por el ARM (*HeartSignal*).

```
volatile unsigned long millis;

DigitalOut red(p22);

DigitalOut HeartLight(p30);
AnalogIn HeartSignal(p3);
```

**Figura 3.1.** Inicialización de *millis* y asignación de pines.

Seguidamente declaramos dos interrupciones. La primera de ellas detecta cambios en el botón disponible en la Tiny-BLE, que usaremos para resetear el sistema de detección de pulso. Se implementa con el tipo *InterruptIn*. La segunda interrupción es de tipo *Ticker*, y se encarga de hacer un muestreo periódico de la señal recogida. Como se muestra en la Figura 3.2 se crean dos variables. La primera de ellas, *HeartSample*, almacenará los datos del sensor. Este dato, proveniente del sensor y capturado en *HeartSignal*, será digitalizado en el rango 0-1024. La segunda, *newsample*, es de tipo booleano, y será *true* cuando llegue un nuevo dato al sensor. El interfaz pc que aparece en la Figura 3.2 permite el debugueo del software mediante la emulación de un puerto serie (ver Apartado 3.2).

```
InterruptIn button(p17);

Ticker Sampler;

volatile uint16_t HeartSample;
volatile bool newsample;

Serial pc(UART_TX, UART_RX);
```

**Figura 3.2.** Interrupciones y variables del sensor.

A continuación se describen las diferentes funciones de las que se hace uso para la toma de la señal de pulso. En la Figura 3.3 se muestra la que denominamos como *TakeSample*. Esta función se encarga de tomar nuevas muestras haciendo uso de la función *read* del tipo *AnalogIn* (en este caso se captura un *unsigned int* de 2 bytes (16 bits)), actualizando a su vez el valor de *millis* y poniéndose a *true* el booleano

*newsample*. El valor de *millis* se actualiza con la misma cadencia que el *Ticker* definirá (10 ms). Esta variable es especialmente importante, pues será la que finalmente determinará los latidos por minuto. Como se verá al analizar el algoritmo de estimación del pulso, este valor se va incrementando con cada captura en *TakeSample*, salvo que haya pasado un tiempo muy grande sin detectar pico de nuevo pulso o que se detecte la llegada de un nuevo pico de pulso. En ambos casos la señal se inicializa a cero (ver Apartado 3.1.1.2).

```
void TakeSample()
{
    HeartSample=HeartSignal.read_u16();
    millis=millis+10;
    newsample=true;
}
```

**Figura 3.3.** Función encargada de la toma de datos.

Como se ha comentado anteriormente, para poder reinicializar nuestra pulsera se empleará la opción de apretar un botón. Esto viene implementado por las funciones *buttonPressedCallback* y *buttonReleasedCallback*. La primera de ellas interrumpe el proceso de toma de datos mientras se deja pulsado el botón. Cuando se detecta que se ha soltado el botón se volverá a inicializar la toma de datos (el *TakeSample* se ejecuta cada 10.000 us (10 ms)). Estas dos funciones se muestran en la Figura 3.4.

```
void buttonPressedCallback(void)
{
    Sampler.detach();
    sleep();
}

void buttonReleasedCallback(void)
{
    Sampler.attach_us(&TakeSample,10000);
}
```

**Figura 3.4.** Funciones relacionadas con la detención de toma de datos.

Finalmente, la Figura 3.5 muestra la inicialización de las diferentes variables necesarias para la medida del pulso (se describen en el Apartado 3.1.1.2). Además también se define los baudios a los cuales podremos observar los datos en nuestro ordenador usando el mencionado terminal serie (ver Apartado 3.2). En este caso se

fija a 115200 baudios. También muestra la asignación de las funciones anteriormente descritas a las interrupciones *Sampler* y *button* (Figura 3.2).

```
float baseline=512.0, HFoutput=0.0, LFoutput=0.0, HFoutput2=0.0, hysteresis=1.0;
millis=0;
static int binOut=0, lastBinOut, BPM;
static unsigned long beat;
float signalSize, valley=0.0, peak=0.0, smoothPeak=1.0, smoothValley=-1.0;

red = 0;
HeartLight=1;
newsample= false;
pc.baud(115200);
Sampler.attach_us(&TakeSample,10000);
button.fall(buttonPressedCallback);
button.rise(buttonReleasedCallback);
```

**Figura 3.5.** Inicialización de variables necesarias para la creación de la onda de pulso.

### 3.1.1.2. Algoritmo de estimación del pulso

A continuación se describe el algoritmo empleado para estimar el pulso. Está basado en un algoritmo proporcionado por los fabricantes del sensor de pulso *pulsesensor*, que ha sido modificado para compilarlo en MBED para la Tiny-BLE. El algoritmo emplea distintas variables auxiliares que se irán describiendo en este apartado, Pero, junto al dato capturado, el algoritmo emplea dos variables importantes. La primera es *signalSize*, que determina la distancia entre los valores de pico y valle en la señal detectada. Cuando este valor supera un umbral prefijado se asume que estamos ante un pico de pulso o latido. La segunda es la ya mencionada *millis*. Esta señal almacena la distancia entre dos latidos consecutivos en decenas de milisegundos, permitiendo estimar la medida de latidos por minuto ( $6000/millis$ ).

Ambos datos son la salida natural del sistema. El primero (*signalSize*) nos permite visualizar si el sistema devuelve picos de valor relevante. Si son de valor pequeño, la medida será, en principio, menos fiable. El segundo (no directamente *millis*, sino *BPM*, como se muestra en la Figura 3.6) serán los latidos por minuto. En la Figura 3.6 se aprecian otras variables: *peak* y *smoothPeak*, *valley* y *smoothValley*. Su actualización se describe a continuación. En la Figura 3.6 se aprecia también el uso de una función *smooth* que se describirá también más adelante. La impresión en el puerto serie usa la variable *pc* y emplea, en el PC, el programa TeraTerm, como se describe en el Apartado 3.2.

```

if (lastBinOut == 0 && binOut == 1 && signalSize>20 && millis > 240) {
    beat = millis;
    BPM = 60000 / (beat);
    pc.printf("BPM: %d \n",BPM);
    pc.printf("Size: %.3f\n",signalSize);
    //actualiza valores
    smoothValley = smooth(valley, 0.6, smoothValley); // smooth valleys
    smoothPeak = smooth(peak, 0.6, smoothPeak); // smooth peaks
    //reset values
    peak = 0;
    valley = 0;
    millis=0;
}

```

**Figura 3.6.** Actualización de los valores cada toma de datos.

Los valores de pico y valle, que determinan cuando tenemos un latido en la señal de pulso, se evalúan usando la entrada *HeartSample* y una señal que es prácticamente la media de los datos capturados (*baseline*), que se inicializa en principio a 512 (Figura 3.5) pues la señal capturada y digitalizada puede variar entre 0 y 1023. Como muestra la Figura 3.7, con cada nueva entrada se actualiza el valor de *baseline* (aunque, como se muestra en la Figura 3.8, el parámetro aplicado al *smooth* hace que el nuevo dato influya sólo levemente en la media). Este valor de *baseline* sirve para normalizar *HeartSample* entre -512 y 512 (*HeartSample - baseline*). El algoritmo que calcula el pulso se muestra en la Figura 3.7.

```

1 while (1) {
2   if (newsample==true) {
3     newsample=false;
4     baseline = smooth((float)HeartSample, 0.99, baseline); // Valor medio (DC)
5     HFoutput = smooth(((float)HeartSample - baseline), 0.2, HFoutput);
6     HFoutput2 = HFoutput + hysteresis;
7     LFoutput = smooth(((float)HeartSample - baseline), 0.95, LFoutput);

// Se detectan picos y valles para escalar la salida. También se escala la variable
// histeresis.
8   if (HFoutput > peak) peak = HFoutput; // detector pico
9   if (HFoutput < valley) valley = HFoutput; // detector valle
10  if (millis > 1800)
11  {

```

```

// si pasa demasiado tiempo sin un latido, reinicia los valores para el tamaño
12     smoothPeak = smooth(peak, 0.6, smoothPeak); // smooth peaks
13     smoothValley = smooth(valley, 0.6, smoothValley); // smooth valleys
14     valley = 0;
15     peak = 0;
16     millis=0;
17     }
18     signalSize = smoothPeak - smoothValley; // tamaño de la señal de pulso HF
19     if (HFoutput2 < LFoutput)
20     {
21         lastBinOut = binOut;
22         binOut = 1; // por debajo de la referencia
23         hysteresis = - constrain((signalSize / 6.0), 1.0, 30.0) ; // escalado de histeresis
24     }
25     else
26     {
27         lastBinOut = binOut;
28         binOut = 0;
29         hysteresis = constrain((signalSize / 6.0), 1.0, 30.0) ; // escalado de histeresis
30     }
31     if (lastBinOut == 1 && binOut == 0) {
32         red = !red;
33     }
34     if (lastBinOut == 0 && binOut == 1 && signalSize>20 && millis > 240)
35     {
36         beat = millis;
37         BPM = 60000 / (beat);
38         pc.printf("BPM: %d \n",BPM);
39         pc.printf("Size: %.3f\n",signalSize);
//actualiza valores
40         smoothValley = smooth(valley, 0.6, smoothValley); // smooth valleys
41         smoothPeak = smooth(peak, 0.6, smoothPeak); // smooth peaks
//reset values
42         peak = 0;

```

```

43     valley = 0;
44     millis=0;
45     }
46 }
47 }

```

**Figura 3.7.** Algoritmo de estimación del pulso

Las función *smooth*, muy empleada en el algoritmo, permite suavizar la medida teniendo en cuenta, en cierta medida, los valores previos de la señal. La función implementada se muestra en la Figura 3.8. En ella podemos ver detalladamente cómo se calcula el valor final del filtro, *smoothedVal*. Un bucle inicial chequea que los valores recogidos por el filtro se mantienen dentro del rango correcto.

```

float smooth(float data, float filterVal, float smoothedVal)
{
    if (filterVal > 1) {
        filterVal = .99;
    } else if (filterVal <= 0.0) {
        filterVal = 0.01;
    }

    smoothedVal = (data * (1.0 - filterVal)) + (smoothedVal * filterVal);
    return smoothedVal;
}

```

**Figura 3.8.** Función de filtrado de señal.

Volviendo al algoritmo de la Figura 3.7, al comienzo del mismo se actualizan dos variables intermedias: *HFoutput* y *LFoutput*. La señal *HFoutput* será (*HeartSample - baseline*) sólo levemente suavizada. La segunda variable es la señal (*HeartSample - baseline*) muy suavizada (prácticamente una señal de referencia). *HFoutput2* se genera desde *HFoutput*, a la que se suma un valor de histéresis. Cuando *HFoutput2* es inferior a *LFoutput* podemos estar en un valle (línea 19 del código). En caso contrario, en un pico. Cuando se detecta un pico, el valor de histéresis se hace positivo. Cuando se detecta un valle, el valor de histéresis es negativo. Esto hace que, para considerar un valor como pico o valle, éste tenga que estar separado de la referencia por, al menos, ese valor de histéresis. Dado que la amplitud pico-valle de la señal puede ser distinta en función del lugar del cuerpo donde se emplea el

sensor o las condiciones de captura, el valor de histéresis se actualiza más adelante en el algoritmo en función de *signalSize* (líneas 23 y 29 en el código de la Figura 3.7). Los valores de picos y valles se suavizan por separado. Y, en cualquier caso, la existencia de un valle se valida finalmente con la condición expresada en la línea 34 del código. Si se detecta dicho valle, se asume que se ha producido un pulso y se inicializan todas las variables. En la siguiente iteración, los valores de *peak* y *valley* se actualizan con *HFoutput* (líneas 8 y 9 en el código).

Para el mencionado cálculo del valor de histéresis se usa la función *constrain*. Se trata de un limitador, el cual nos asegura que el valor que nos va a devolver se encuentre entre unos valores. Dicha función es la mostrada en la Figura 3.9.

```
float constrain(float value, float min, float max)
{
    if (value > max) return max;
    else if (value < min) return min;
    else return value;
}
```

**Figura 3.9.** Función limitadora.

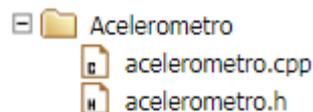
### 3.1.2. Detección de golpecito (*tap*) en la placa

Si incluimos la Tiny-BLE en una envolvente, construyendo la pulsera, el botón empleado para resetear el sistema quedaría dentro de dicha envolvente. Sacarla fuera supone una dificultad para la envolvente, aumentar la posible permeabilidad de la misma, y una fuente de problemas. Es por ello interesante sustituir dicho botón por otro sensor que no necesite romper la impermeabilidad de la envolvente. Una opción es sustituirlo por detectar que se ha dado un pequeño golpecito al sistema. Para implementar esta detección de movimiento podemos usar la unidad inercial presente en la Tiny-BLE. La MPU6050 incluye acelerómetros y giróscopos, que permiten estimar la aceleración o cambios de movimiento a la que está sometida el sistema.

El acelerómetro con el que vamos a trabajar se encuentra incluido en el MPU-6050, se trata de un acelerómetro digital con capacidad para tomar medidas en los tres ejes coordenados. Tiene habilitados dos protocolos de comunicación: SPI (*Serial Peripheral Interface*) o I2C (*Inter-Integrated Circuit*) [21,22]. Este modelo digital tiene

la ventaja de ser más funcional y de verse menos afectado por el ruido que los acelerómetros analógicos.

Trabajando sobre el código, tendremos una carpeta específica para el acelerómetro, Figura 3.10, compuesta por dos archivos: *acelerometro.cpp* y *acelerometro.h*. Dichos archivos serán analizados por separado y descritos paso a paso a continuación.



**Figura 3.10.** División de la carpeta Acelerómetro.

### 3.1.2.1. Acelerometro.h : instrucciones y funciones.

Como indica la Figura 3.11, el uso del acelerómetro obliga a emplear distintas librerías, proporcionadas por MBED y el fabricante de la unidad inercial. En concreto, como se observa en la Figura 3.11., es necesario incluir *mbed\_i2c.h*, para poder hacer uso del protocolo de comunicación; *inv\_mpu.h* e *inv\_mpu\_dmp\_motion\_driver.h* son necesarios para el funcionamiento del MPU que contiene la placa. El archivo *mbed.h* es necesario para trabajar con las funciones básicas proporcionadas por la plataforma MBED. Finalmente, el archivo *InputOutput.h* es de diseño propio y necesario para definir la información referida a entradas y salidas.

```
4 //para el acelerometro
5 #include "mbed_i2c.h"
6 #include "inv_mpu.h"
7 #include "inv_mpu_dmp_motion_driver.h"
8
9 #include "mbed.h"
10 #include "InputOutput.h"
```

**Figura 3.11.** Instrucciones necesarias para la compilación.

Continuando con el análisis de nuestro archivo tenemos dos variables (Figura 3.12): *acelerometro\_init()* y *acelerometro\_error* (*short sensors*). Ambas declaradas como tipo entero (*int*) y que serán desarrolladas en el archivo *acelerometro.cpp*. La primera de ellas se encarga de reiniciar el acelerómetro y la segunda de la detección de errores o fallos en el sistema a la hora de tomar la medida.

```

14 int acelerometro_init();
15 int acelerometro_error(short sensors);

```

**Figura 3.12.** Variables de inicialización y detección de errores.

Por último en este archivo se encuentran los prototipos de tres funciones (Figura 3.13). La primera de ellas está encargada de convertir la matriz de orientación en un escalar (*unsigned short inv\_orientation\_matrix\_to\_scalar (const signed char\*mtx)*). Las dos últimas funciones se ejecutarán como interrupciones y ambas deberán de ser definidas en el programa main.

```

17 //funciones prototipo para cuestiones de acelerometro
18 unsigned short inv_orientation_matrix_to_scalar( const signed char *mtx);
19
20 //estas 2 funciones que se ejecutan como interrupciones
21 void tap_cb(unsigned char direction, unsigned char count);
22 void android_orient_cb(unsigned char orientation);

```

**Figura 3.13.** Funciones del archivo acelerómetro.h

### 3.1.2.2. Algoritmo de funcionamiento del acelerómetro.

A continuación se describe detalladamente todas las funciones, variables y elementos que han sido necesarias para la implementación del algoritmo que le da funcionalidad al acelerómetro en el archivo acelerometro.cpp. La Figura 3.14 muestra que, lo que primero necesitamos, es definir la matriz de orientación (*static signed char board\_orientation(9)*), la cual es inicialmente una matriz identidad de 3x3. La variable de tipo entero *Volatile int read\_none\_count*, que también se muestra en la figura, es la encargada de contar los reintentos de inicializar el acelerómetro en caso de fallo.

```

3 static signed char board_orientation[9] = {
4     1, 0, 0,
5     0, 1, 0,
6     0, 0, 1
7 }; //orientacion
8
9 //reintentos en caso de fallo del acelerometro
10 volatile int read_none_count = 0;

```

**Figura 3.14.** Matriz de orientación y contador de intentos de inicialización.

La inicialización del acelerómetro (Figura 3.15) se implementa en la función *acelerometro\_init()*. Básicamente, lo que se hace es limpiar las conexiones que haya anteriormente, *mbed\_i2c\_clear()*, y así poder empezar las conexiones desde cero e inicializar el MPU.

```
int acelerometro_init()
{
    mbed_i2c_clear(MPU6050_SDA, MPU6050_SCL);
    mbed_i2c_init(MPU6050_SDA, MPU6050_SCL);
    if (mpu_init(0)) {
        return 1;
    }
}
```

**Figura 3.15.** Inicialización del protocolo de comunicación y MPU

La siguiente parte del código, Figura 3.16, sin entrar en gran profundidad, se encarga de la introducción de los datos tomados por el acelerómetro y giroscopio manteniendo una configuración FIFO. Se establece también una frecuencia de muestreo de datos, *mpu\_set\_sample\_rate()*, la cual viene definida por aquella que tiene por defecto el MPU.

```
mpu_set_sensors(INV_XYZ_GYRO | INV_XYZ_ACCEL);
/* Push both gyro and accel data into the FIFO. */
mpu_configure_fifo(INV_XYZ_GYRO | INV_XYZ_ACCEL);
mpu_set_sample_rate(DEFAULT_MPU_HZ);
mpu_set_accel_fsr(4);
```

**Figura 3.16.** Estructura FIFO para la toma de datos.

Se desarrollan una serie de funciones para volver a leer el paso anterior en caso de que los valores establecidos no sean los apropiados, Figura 3.17, para ello se han creado tres variables: *accel\_fsr*, *gyro\_rate* y *gyro\_fsr*. Con dichas variables, y sus métodos *get* asociados, podremos obtener los valores que se están buscando.

```
unsigned char accel_fsr;
unsigned short gyro_rate, gyro_fsr;
mpu_get_sample_rate(&gyro_rate);
mpu_get_gyro_fsr(&gyro_fsr);
mpu_get_accel_fsr(&accel_fsr);
```

**Figura 3.17.** Procedimiento cuando se detectan valores establecidos inapropiados.

Para leer y cargar los datos de movimiento y orientación se harán uso de funciones presentes en las librerías del sensor. En concreto, son las funciones

`dmp_load_motion_driver_firmware()` y `dmp_set_orientation()`, que establece la orientación haciendo uso de la función `inv_orientation_matrix_to_scalar()`. Las dos funciones que se muestran en la Figura 3.18 están encargadas, respectivamente, del registro de golpes y de la orientación: `dmp_register_tap_cb()` y `dmp_register_android_orient_cb()`.

```
dmp_load_motion_driver_firmware();
dmp_set_orientation(
    inv_orientation_matrix_to_scalar(board_orientation));
dmp_register_tap_cb(tap_cb);
dmp_register_android_orient_cb(android_orient_cb);
```

**Figura 3.18.** Funciones de lectura de movimiento y registro.

Una vez capturados los datos, se construye (Figura 3.19) una variable, `uint16_t dmp_features`, donde se recogen las características de la detección de movimiento. Se aprecia como la primera característica (`DMP_FEATURE_TAP`) es la asociada al golpecito en el sistema. También es necesario habilitar las características disponibles, `dmp_enable_feature()`, así como establecer el rango de entrada de datos en la fifo, `dmp_set_fifo_rate(10)`, o el modo de interrupción, `dmp_set_interrupt_mode(DMP_INT_GESTURE)`.

```
uint16_t dmp_features = DMP_FEATURE_TAP | DMP_FEATURE_PEDOMETER |
    DMP_FEATURE_ANDROID_ORIENT | DMP_FEATURE_SEND_RAW_ACCEL;

dmp_enable_feature(dmp_features);

dmp_set_fifo_rate(10);
mpu_set_dmp_state(1);

dmp_set_interrupt_mode(DMP_INT_GESTURE);
dmp_set_tap_thresh(TAP_XYZ, 300);

return 0;
}
```

**Figura 3.19.** Características definidas dentro de la función de inicialización.

El sistema incluye una función para detección de errores. La función se muestra en la Figura 3.20. En esta función actualizaremos la variable `read_none_count`, la cual es un contador y mientras que se mantenga a cero significará que nuestro acelerómetro se ha iniciado correctamente. Si se detecta un error, se harán tres intentos de reiniciar el acelerómetro. Si no arranca el sistema, lo reiniciaremos llevando de nuevo la variable a cero y limpiando todas las conexiones que el

protocolo de comunicación i2c haya podido realizar durante este tiempo. Si estos reintentos no fueran superiores a tres el acelerómetro comenzaría a funcionar sin necesidad de reiniciar el sistema.

```
56 int acelerometro_error(short sensors)
57 {
58     if (sensors) {
59         read_none_count = 0;
60         return 0;
61     } else {
62         read_none_count++;
63         //aquí ha sucedido algo pero vamos a intentar 3 veces
64         if (read_none_count > 3) {
65             read_none_count = 0;
66             mbed_i2c_clear(MPU6050_SDA, MPU6050_SCL);
67             return 1;
68         } else {
69             return 0;
70         }
71     }
72 }
```

**Figura 3.20.** Función de detección de errores en el acelerómetro.

Por último, tenemos dos funciones las cuales han sido directamente tomadas del fabricante invensense para la gestión del MPU6050. Estas funciones se encargan, básicamente, de convertir el valor de la matriz en un escalar. En la Figura 3.21, podemos ver que se trata de la función definida en el programa acelerometro.h. Esta función hace uso de otra, *inv\_row\_2\_scale*, y tiene definida una variable, *unsigned short scalar*, que será la que nos devuelva el valor del escalar.

```
unsigned short inv_orientation_matrix_to_scalar(
    const signed char *mtx)
{
    unsigned short scalar;

    scalar = inv_row_2_scale(mtx);
    scalar |= inv_row_2_scale(mtx + 3) << 3;
    scalar |= inv_row_2_scale(mtx + 6) << 6;

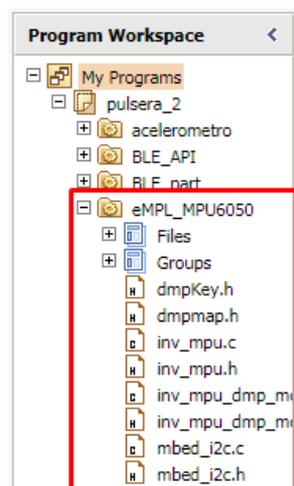
    return scalar;
}
```

**Figura 3.21.** Función encargada de transformar la matriz de orientación en un escalar.

### 3.1.2.3. Librería EMPL\_MPU6050.

El programa descrito en el Apartado anterior hace uso de librerías proporcionadas por MBED. En la librería EMPL\_MPU6050 se proporcionan funciones para trabajar con el acelerómetro y el giroscopio. El giroscopio simplemente se encarga de la medición de velocidades angulares basándose en impulsos de rotación. Por su parte el acelerómetro mide la aceleración, inclinación o vibración. Estos datos se transforman en una magnitud eléctrica que será lo que se emplea en los equipos de adquisición de señales.

En este caso dentro de esta librería nos encontramos archivos como un controlador basado en comunicación I2C para el giroscopio, imagen y funciones de interfaz, así como funciones de comunicación necesarias para poderse comunicar con el MPU de la placa.



**Figura 3.22.** Composición de la librería para MPU6050

### 3.1.3.3. Librería nRF51822

Al igual que en el caso anterior, esta librería ha sido directamente exportada desde la plataforma MBED. En este caso se compone, Figura 3.23, principalmente por dos carpetas y tres archivos. Estos son los que le dan funcionalidad a dicha librería y hacen que su funcionamiento sea el correcto.

Library: /Pulsometro/nRF51822			
Type to filter the list ...		<input type="checkbox"/> Match Case	<input type="checkbox"/> Whole Word
Name	Size	Type	
Classes		Classes Documentation	
Files		Files Documentation	
Groups		Grouped Documentation	
Structs		Structs Documentation	
bootloader		Program Folder	
source		Program Folder	
apache-2.0.txt	0.5 kB	Text File	
module.json	0.8 kB	Script File	
softdevice_nrf51822_licence_ag	1.5 kB	Text File	

**Figura 3.23.** Estructura interna de la librería nRF51822

Si exploramos los archivos que lo componen podemos encontrar desde las características del nRF5x hasta todas las funciones que son necesarias para la buena comunicación de la placa con otro dispositivo mediante el uso de la tecnología BLE.

### 3.1.3. BLE\_part.

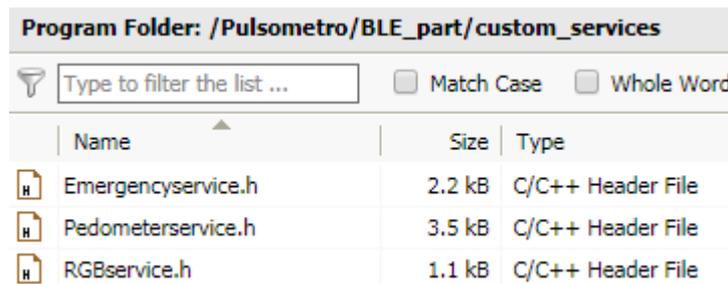
Como se muestra en el Apartado 3.1.3.2, el uso del BLE se fundamenta en una API proporcionada por MBED. Aparte de esa API, y siguiendo el esquema propuesto por MBED, básicamente el uso del BLE se construye entorno a dos ficheros (BLE\_handling.cpp y BLE\_handling.h), que deberemos adaptar a nuestro proyecto, y una carpeta, denominada como “custom\_services”, donde estarán definidos los servicios BLE que definamos nosotros personalmente (Figura 3.24).

Program Folder: /Pulsometro/BLE_part			
Type to filter the list ...		<input type="checkbox"/> Match Case	<input type="checkbox"/> Whole Word
Name	Size	Type	
custom_services		Program Folder	
BLE_handling.cpp	4.5 kB	C/C++ Source File	
BLE_handling.h	1.3 kB	C/C++ Header File	

**Figura 3.24.** Composición de la carpeta BLE\_part.

### 3.1.3.1. Análisis de custom\_service.

Una vez abierta la carpeta de servicios podemos observar que está compuesta por tres archivos de cabecera, Figura 3.25, las cuales cada una de ellas proporciona un servicio. El que nos interesa para el podómetro es el PedometerService.h. El primero, Emergencyservice.h, permite enviar determinadas alertas por BLE (pulsación de botón por ejemplo). El tercero, RGBservice.h, permite activar los leds.



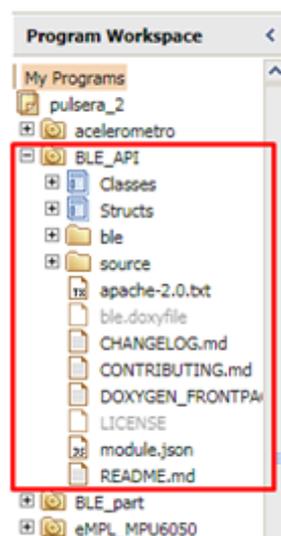
Program Folder: /Pulsometro/BLE_part/custom_services		
Type to filter the list ...		
<input type="checkbox"/> Match Case <input type="checkbox"/> Whole Word		
Name	Size	Type
Emergencyservice.h	2.2 kB	C/C++ Header File
Pedometerservice.h	3.5 kB	C/C++ Header File
RGBservice.h	1.1 kB	C/C++ Header File

Figura 3.25. Elementos incluidos en la carpeta custom\_services.

### 3.1.3.2. Librería BLE\_API.

La plataforma MBED proporciona distintas librerías que podemos usar en nuestros proyectos. Es el caso de la librería BLE\_API.

Esta librería contiene toda clase de códigos y estructuras necesarios para dar soporte a la tecnología BLE. En la Figura 3.26, podemos observar todos los elementos que lo componen, cada uno de ellos aportado algo imprescindible para su buen funcionamiento.



**Figura 3.26.** Composición de la librería BLE\_API

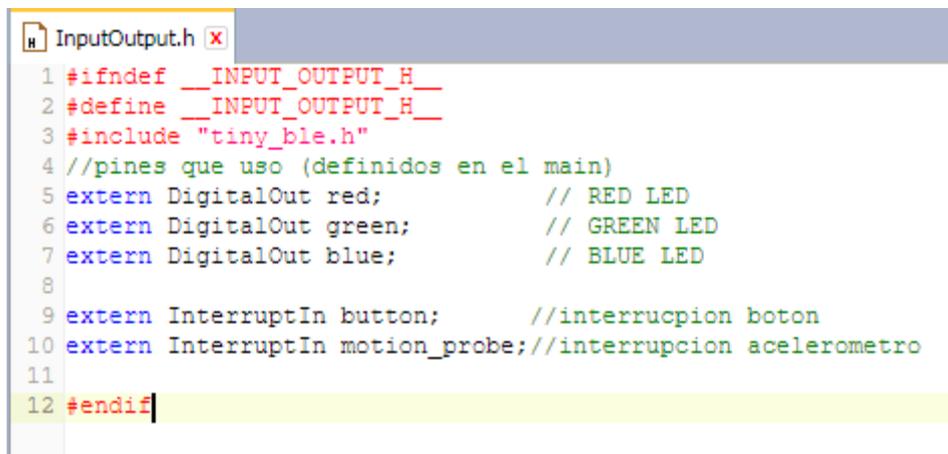
Según la propia definición que podemos encontrar en MBED, nos dice que esta librería es una “abstracción de alto nivel para hacer uso de la tecnología BLE en múltiples plataformas”. Los archivos presentes en esta librería crean las estructuras, comunicaciones y características necesarias para que la tecnología BLE funcione sin problemas en nuestro proyecto.

### 3.1.4 Código principal y archivos de cabecera.

#### 3.1.4.1. Archivo de cabecera InputOutput.h.

En este caso se trata de un archivo, Figura 3.27, relacionado con las entradas y salidas que se producen en nuestro conjunto. Se trata de un archivo simple donde se definen los leds externos compuestos por tres colores: rojo, verde y azul.

Por otro lado nos encontramos las cabeceras de dos funciones a implementar. La primera de ellas es de la interrupción del botón, y la segunda la interrupción de funcionamiento del acelerómetro.



```
InputOutput.h x
1 #ifndef __INPUT_OUTPUT_H__
2 #define __INPUT_OUTPUT_H__
3 #include "tiny_ble.h"
4 //pines que uso (definidos en el main)
5 extern DigitalOut red;           // RED LED
6 extern DigitalOut green;        // GREEN LED
7 extern DigitalOut blue;         // BLUE LED
8
9 extern InterruptIn button;      //interruccion boton
10 extern InterruptIn motion_probe;//interruccion acelerometro
11
12 #endif
```

**Figura 3.27.** Código del archivo InputOutput.h.

#### 3.1.4.2. Archivo de cabecera tiny\_ble.h.

Este archivo define a qué pin va asociada una determinada función. Por ejemplo, en la Figura 3.28 podemos observar que el pin 1 irá asociado con la batería. El pin 14 se encuentra relacionado con la interrupción del MPU, mientras que del pin 21 al 23 se definen los tres leds presentes en nuestra placa. Este archivo enlaza el hardware de la placa con nuestro diseño software.

```
tiny_ble.h x
1
2 #ifndef __TINY_BLE_H__
3 #define __TINY_BLE_H__
4
5 #define LED_GREEN    p21
6 #define LED_RED     p22
7 #define LED_BLUE    p23
8 #define BUTTON_PIN  p17
9 #define BATTERY_PIN p1
10
11 #define MPU6050_SDA p12
12 #define MPU6050_SCL p13
13 #define MPU6050_INTERRUPT p14
14
15 #define UART_TX     p9
16 #define UART_RX     p11
17 #define UART_CTS   p8
18 #define UART_RTS   p10
19
20 #endif // __TINY_BLE_H__
21
```

**Figura 3.28.** Definición de pines en el archivo tiny\_ble.h

### 3.1.4.3. Función main.

Tras la inclusión de las cabeceras necesarias, nos encontraremos las definiciones de los tres led que posee tiny ble. A continuación se definen, Figura 3.29, dos interrupciones la primera de ella provocada por presionar un botón y la segunda interrupción que gestiona la función del acelerómetro.

```
DigitalOut red(LED_RED);
DigitalOut green(LED_GREEN);
DigitalOut blue(LED_BLUE);

InterruptIn button(BUTTON_PIN);

InterruptIn motion_probe(MPU6050_INTERRUPT);
```

**Figura 3.29.** Inicialización de pines y de interrupciones.

A continuación tendremos las variables necesarias para la gestión de interrupciones inicializadas a cero. Seguidamente una llamada a las funciones que se encargan de las interrupciones. La primera de estas funciones es para la interrupción manual y hace que la variable que lo maneja inicializada anteriormente a cero pase ahora a un valor de uno. La segunda de estas funciones es la interrupción de la pulsera tras un

golpe, esto activa un nuevo estado de emergencia, donde se almacena la acción de golpeo leve (tap) de la placa. Todo esto se muestra en Figura 3.30.

```
volatile uint8_t motion_event = 0;
volatile uint8_t periodic_event = 0;

Serial pc(UART_TX, UART_RX);
#define LOG(...) { pc.printf(__VA_ARGS__); }

volatile uint8_t new_emergency_status;
volatile uint8_t emergency_status;

////////////////////
//callback para interrupciones
void motion_interrupt_handle(void)
{
    motion_event = 1;
}

void tap_cb(unsigned char direction, unsigned char count)
{
    // LOG("Tap motion detected\n");
    // LOG("%d\r\n",direction);
    new_emergency_status = emergency_status | EMERGENCY_SERVICE_IMPACT;
    green=~green;
}
}
```

**Figura 3.30.** Funciones para el control de interrupciones.

Antes de comenzar con el main, se han definido tres funciones más, encargadas de la orientación de la pulsera, actualizar el vector de estado en caso de pulsar el botón y la detección de eventos periódicos, Figura 3.31.

```
void android_orient_cb(unsigned char orientation)
{
    LOG("Orientation changed\n");
}

void buttonPressedCallback(void)
{
    new_emergency_status = emergency_status | EMERGENCY_SERVICE_BUTTON;
}
void periodic(void)
{
    periodic_event = 1;
}
}
```

**Figura 3.31.** Control de orientación, llamada ante pulsar el botón y detección de eventos periódicos.

Dentro de nuestra función main lo primero que nos encontramos, Figura 3.32, es la inicialización de las variables y del MPU.

```

int main()
{
    pc.baud(115200);
    Ticker PeriodicTrigger;

    //////////////// inicializacion de variables
    red = 1;
    green =1;
    blue=0;
    emergency_status=0x00;
    ////////////////
    //inicializar mpu
    if (acelerometro_init()) {
        LOG("failed to initialize mpu6050\r\n");
    }
}

```

**Figura 3.32.** Inicialización de variables y del acelerómetro.

A continuación nos encontramos, con el enganche a las interrupciones de las que hemos estado hablando hasta el momento y que se mostrará en la Figura 3.33 y de la inicialización de ble.

```

button.fall(buttonPressedCallback);//boton
motion_probe.fall(motion_interrupt_handle); //acelerometro
PeriodicTrigger.attach(periodic, 2.0); //eventos periodicos, cada 2 segundos
////////////////
wait(3);
BLE &ble = BLE::Instance();//declaro una instancia para el BLE
ble.init(bleInitComplete);//inicio el BLE

// espero que el ble inicie correctamente
while (ble.hasInitialized() == false) {

}

```

**Figura 3.33.** Llamada a interrupciones e inicialización de ble.

Por último, nos encontraremos ante un bucle while con el cual se gestionan los eventos relacionados con los cambios de status de emergencia, y los eventos de movimiento y periódicos.

El primero de ellos, cambio de estado de emergencia, dentro de este bucle while lo podremos modelar con un bucle de decisión, Figura 3.34. En el cual, cuando se produce un nuevo estado de emergencia se actualizan los datos necesarias para que se produzca la alarma.

```

emergency_status=readEmergencyState();
if ((new_emergency_status & EMERGENCY_SERVICE_MASK) != 0x00) {
    emergency_status = emergency_status|new_emergency_status;
    new_emergency_status=0x00;
    updateEmergencyService(emergency_status);
}

```

**Figura 3.34.** Actualización de los datos del estado de emergencia.

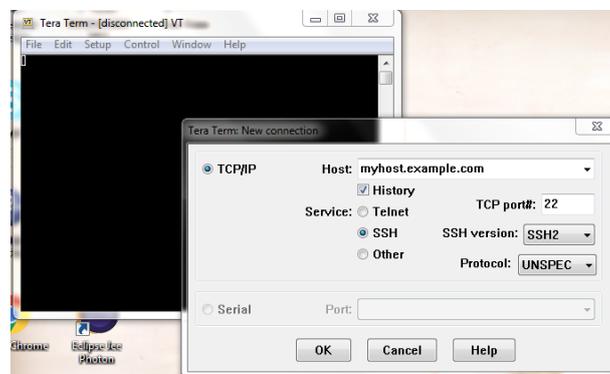
### 3.2. Validación del código.

La verificación de cada función y la validación final del prototipo se ha llevado a cabo mediante la realización de distintas pruebas. El objetivo final es validar que nuestro proyecto captura los valores apropiados para este tipo de dispositivos y que no ocurre ningún problema cuando es ejecutado.

Nos ayudaremos de varios programas, El programa TeraTerm nos ayudará a visualizar en pantalla los datos recogidos por nuestro sensor, funcionando como un terminal de salida conectado a la Tiny-BLE como un puerto serie. La aplicación nRF Connect nos permite detectar y conectarnos a dispositivos BLE, permitiendo chequear el estado de la conexión bluetooth.

#### 3.2.1. TeraTerm

También conocido como “TeraTerm Project”, básicamente es el encargado de emular un terminal. Es un proyecto de código abierto y su descarga es gratuita. Tiene la capacidad de emular diferentes terminales y admite conexiones de puerto serie (TeraTerm).



**Figura 3.35.** Emulador TeraTerm.

### 3.2.2. nRF Connect

Se trata de una herramienta que nos permitirá realizar las distintas pruebas que necesitamos para verificar la conexión BLE de la que hacemos uso. Nos permite realizar conexiones fáciles con otros dispositivos y donde se puede leer y escribir datos. Es una aplicación de descarga gratuita y compatible tanto para ordenadores (Windows, Linux y MacOS) como para Smartphones (IOS y Android).

Esta aplicación está preparada para usarse en conjunto con algunos de los chips de Nordic como, por ejemplo, es el nRF51, del cual estamos haciendo uso en nuestro proyecto. Esto hace que la aplicación detecte automáticamente a qué kit se está conectando (Nordic nRF Connect).



**Figura 3.36.** Logo principal de la aplicación.

### 3.2.3. Integración del código.

Las medidas de pulso y de podómetro se han verificado en programas independientes. Esto ha facilitado la verificación, pero, antes de validar, ha sido necesario integrar ambos, eliminando variables o datos innecesarios, que solo se habían incorporado al proyecto para verificación parcial. Como resultado obtenemos un proyecto con las aplicaciones anteriormente citadas y que tendrá la estructura mostrada en la Figura 3.37.

Program: /TFG\_proyecto

Type to filter the list ...  Match Case  Whole Word

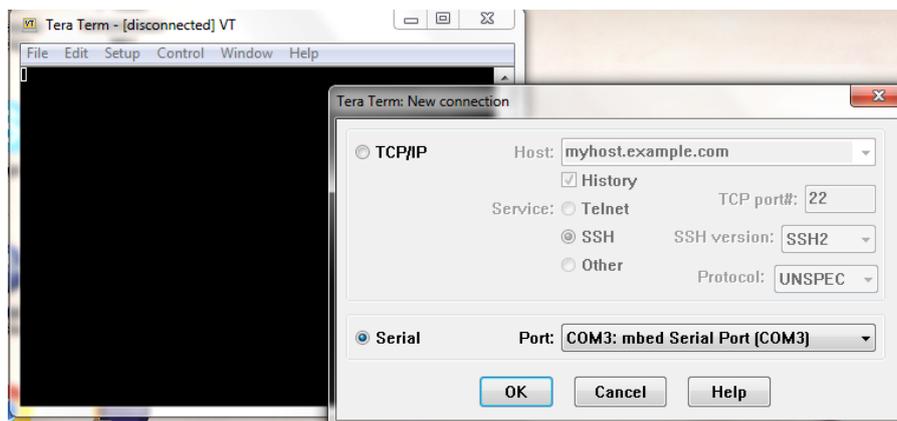
Name	Size	Type
Acelerometro		Program Folder
BLE_API		Published Library
BLE_part		Program Folder
eMPL_MPU6050		Published Library
nRF51822		Published Library
Pulsometro		Program Folder
battery.h	1.3 kB	C/C++ Header File
InputOutput.h	0.0 kB	C/C++ Header File
main.cpp	8.5 kB	C/C++ Source File
tiny_ble.h	0.3 kB	C/C++ Header File
mbed		Library Build

**Figura 3.37.** Estructura final del proyecto.

### 3.2.4. Captura de datos.

En este apartado se describe detalladamente el proceso a seguir para la obtención de los datos que nos han permitido verificar nuestro proyecto. Para poder comenzar con la toma de datos primero es necesario configurar TeraTerm para mostrar datos por pantalla, para ello debemos de descargarnos los driver necesarios para poder manejar el microcontrolador mediante un puerto USB.

Simplemente se trata de descargar los driver e instalarlos. Para ello es necesario tener cerrada cualquier ventana del explorador, y conectada nuestra placa al ordenador. Una vez hecho esto ejecutaremos el archivo descargado y la instalación será automática. Tal y como se muestra en la Figura 3.38, será necesario cambiar la configuración de TeraTerm, que pasará de TCP/IP a estar configurado con “Serial”.

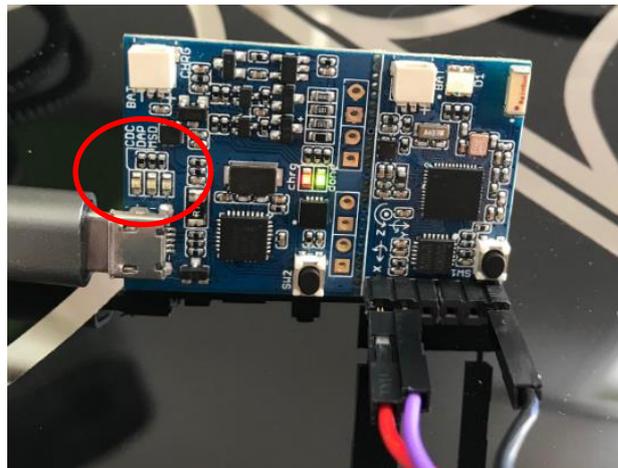


**Figura 3.38.** Configuración de TeraTerm que conecta con el puerto serie de mbed.

Una vez que está listo, conectaremos nuestro sensor y compilaremos nuestro programa en mbed para obtener el archivo de nuestro programa e insertarlo en nuestro microcontrolador para comenzar con la toma de datos. Descargaremos también en nuestro Smartphone la aplicación de “*nRF Connect*” para tenerla disponible para el intercambio de datos.

Para una mejor descripción dividiremos la captura de datos en dos apartados: captura de datos solo con el sensor de pulso en funcionamiento, y captura de datos solo del acelerómetro.

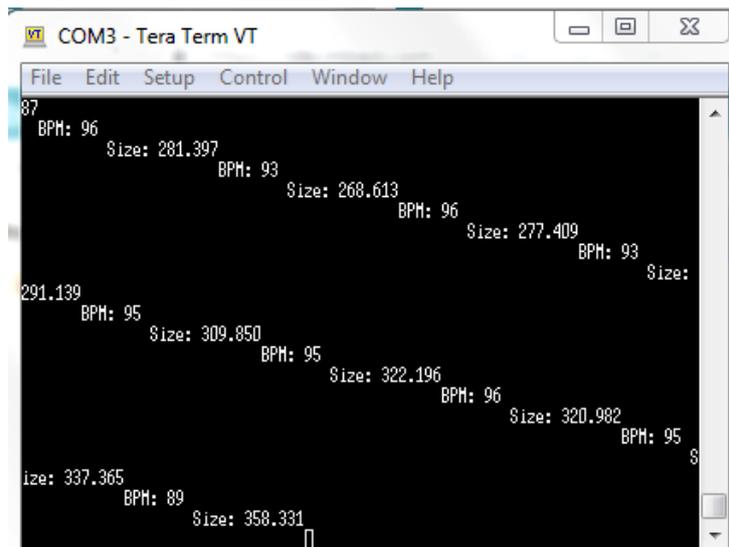
Cuando carguemos nuestro archivo en la placa debemos de fijarnos que los led marcados en la Figura 3.39, se encuentren parpadeando.



**Figura 3.39.** Placa en proceso de carga de un programa .hex.

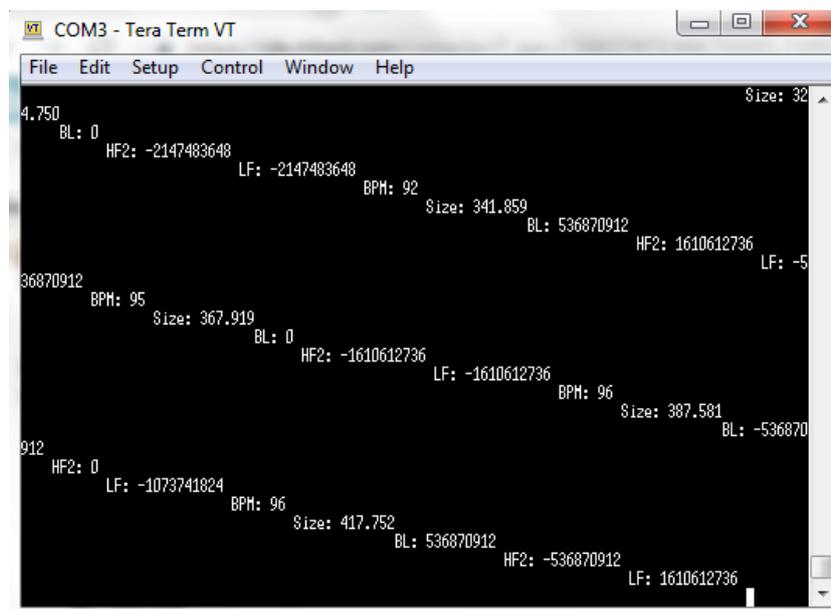
#### **3.2.4.1. Captura de datos del sensor de pulso.**

Para hacer la captura de datos por parte del pulsómetro, trabajaremos con su archivo .hex. Tras una correcta carga en la placa, podremos empezar con la toma de pulso. La captura de datos la podremos observar de dos formas diferentes. En primer lugar podemos tomar los datos que TeraTerm nos muestra por pantalla (Figura 3.40). Para obtenerlos debemos de configurar primeramente los baudios a los que TeraTerm está trabajando, y que correspondan con los indicados en la implementación del código. En la Figura 3.40, se muestran por pantalla dos valores: pulso (*BPM*) y tamaño de señal o muestra (*Size*).



**Figura 3.40.** Datos recogidos con el sensor de pulso.

Aunque estos valores son los de salida, en la verificación de funcionamiento, también se han llegado a visualizar variables intermedias como *LFoutput* y *HFoutput2*. Esa situación se muestra en la Figura 3.41.



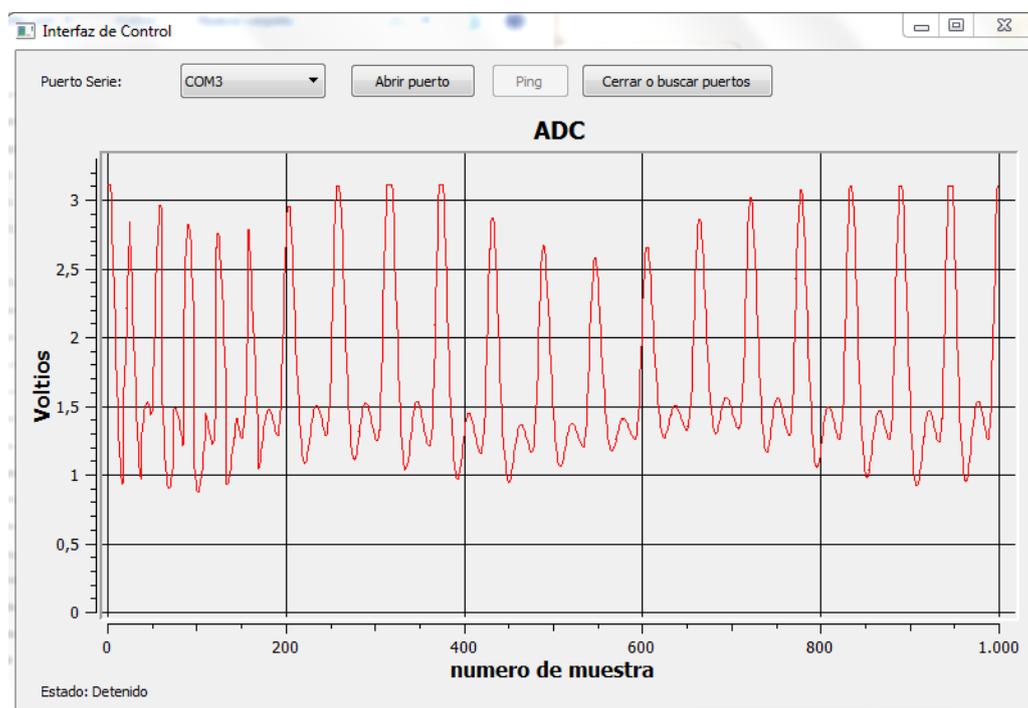
**Figura 3.41.** Impresión por pantalla de las variables *LFoutput* (LF) y *HFoutput2* (HF2).

Sin embargo la forma más intuitiva de ver estos parámetros es mediante una gráfica. Para ello podemos o bien extraer los valores de las imágenes anteriores, para lo cual necesitaríamos hacer un log desde Teraterm y guardar un archivo .csv con los

datos. Posteriormente podemos abrirlo con algún programa tipo Matlab o Excel y, seleccionando los datos de interés, podemos obtener la gráfica buscada.

En nuestro caso se ha elegido una segunda opción, haciendo uso de una interfaz ya programada para visualizar este tipo de gráficas y la cual fue usada durante el curso en las prácticas de la asignatura de electromedicina.

Como resultado, en la Figura 3.42 se muestra un ejemplo de gráfica de pulso, donde nuestro eje X son el número de muestras tomadas y el eje Y los voltios entre los que se encuentra la señal.



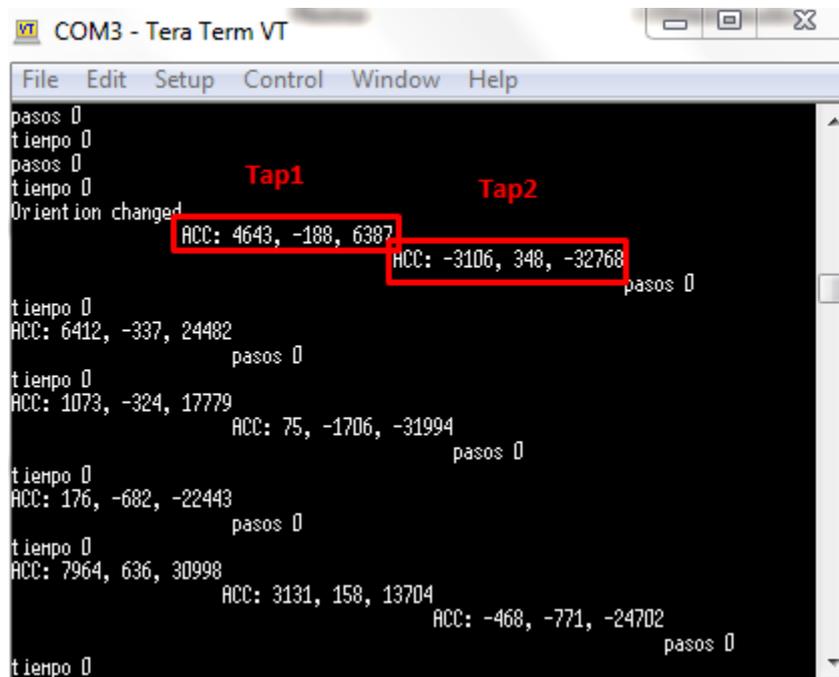
**Figura 3.42.** Gráfica de pulso mediante interfaz.

Con esto quedaría demostrado que el pulsómetro anteriormente implementado funciona de la forma correcta y esperada.

### **3.2.4.2. Captura de datos del acelerómetro.**

Para comprobar su funcionalidad trabajaremos con dos aplicaciones. Por un lado tendremos a TeraTerm, donde se mostraran los cambios que se produce cuando la placa detecta un golpecito (*tap*). Por otro lado, también haremos uso de la aplicación móvil nRF Connect, para verificar que la conexión de bluetooth ha sido establecida y que se produce un flujo de información entre el Smartphone y Tiny-BLE.

Empezaremos trabajando con TeraTerm. Para ello comenzaremos con el mismo proceso que en el apartado anterior, descargando el archivo .hex y comprobando que se carga correctamente en tiny-BLE. Una vez cargado el archivo podemos empezar con la toma de datos, como se muestra en la Figura 3.43.



**Figura 3.43.** Detección de golpecitos (tap).

Cada vez que se ha producido un toque en la placa, ésta lo ha mostrado por pantalla como un cambio de orientación y queda reflejado como se puede ver en la Figura 3.43. Con respecto a lo que aparece respecto a pasos y tiempo, pertenecen a la implementación del podómetro, el cual en este caso no se ha realizado. El tiempo se actualizará cada dos segundos y con él se actualizará a la vez los pasos.

Por otra parte tenemos la conexión a la app nRF Connect, en la Figura 3.44 y Figura 3.45 podremos observar, respectivamente, la detección de nuestro sistema y su posterior conexión y servicios que ofrece.

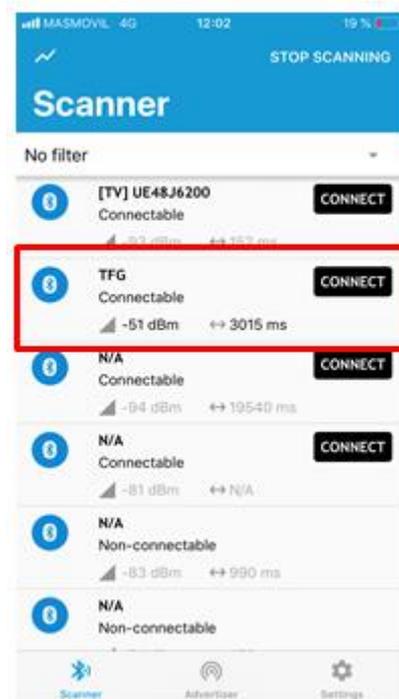


Figura 3.44. Detección de nuestro proyecto por parte de la app.

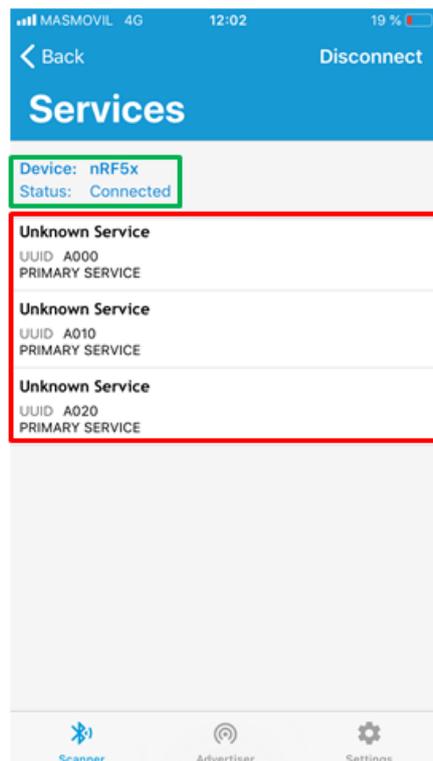
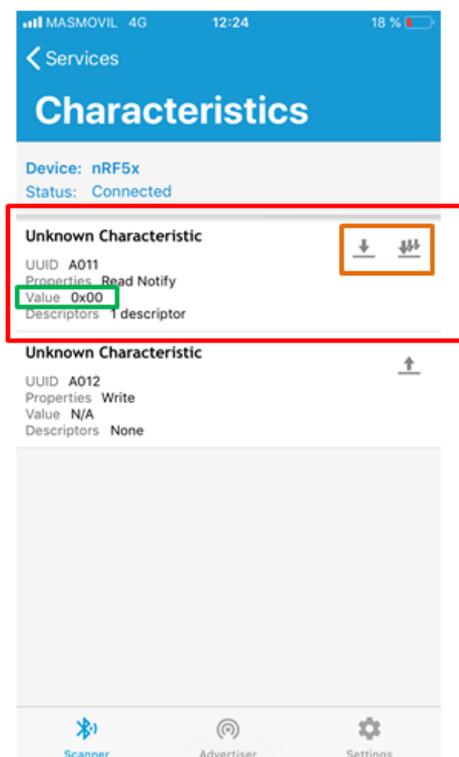


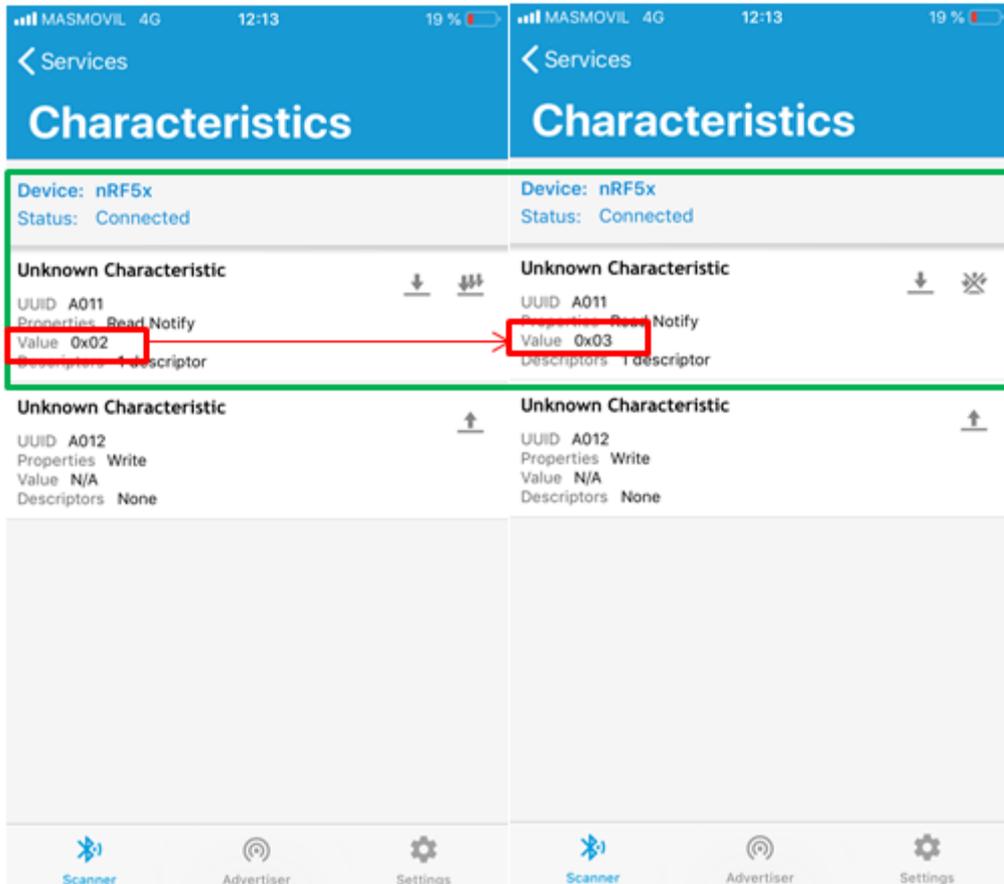
Figura 3.45. Servicios ofrecidos por el dispositivo.

En la figura 3.45 podemos ver, en un cajetín verde, que nuestro dispositivo se encuentra conectado y en rojo otro con los servicios que el dispositivo ofrece. Para

demostrar que esta conexión es real hemos ido accediendo a estos servicios y mediante el envío de notificaciones comprobando a qué correspondía. El ejemplo que vamos a poner viene dado por la detección de movimiento y la activación del estatus de emergencia. En la Figura 3.45, hemos accedido al servicio cuya UUID corresponde al A010, segundo servicio que se muestra en la Figura 3.45. Al acceder a este servicio nos encontraremos una primera característica con la propiedad de lectura y notificación, Figura 3.46, cuyo valor será de 0x00, pero que al detectar un golpe, es decir, la activación del acelerómetro, su valor cambiará a 0x02 y, al pulsar el botón, éste cambia ahora a 0x03, como se observa en la Figura 3.47.



**Figura 3.46.** Características ofrecidas por el segundo servicio. En rojo marcado la característica que varía con los tap e interrupción por el servicio de emergencia, en marrón marcadas las propiedades de lectura y notificación, y en verde marcado su valor inicial a cero.



**Figura 3.47.** Cambio de valores tras la detección de movimiento y tras pulsar el botón de emergencia. En la imagen de la izquierda se puede comprobar que nuestro valor a cambiado a 2 y esto es debido a que ha detectado algún tap, en la imagen de la derecha el valor cambia a 3 y esto es debido a que se ha activado el servicio de emergencia.

Con estas últimas figuras quedaría demostrado que nuestro proyecto tiene capacidad para hacer uso de las propiedades que ofrece Tiny-BLE con respecto a las conexiones inalámbricas.

# Capítulo 4

## Diseño de la envolvente

***En este capítulo se expondrán las dos soluciones que hemos diseñado para la implementar la envolvente de nuestro proyecto.***

### 4.1. Introducción

A lo largo de este capítulo, se realizará el diseño de la envolvente para nuestra pulsera inteligente. Para esta tarea, se hará uso de un programa de diseño CAD (*Computer Aided Design*). En este caso se han planteado, inicialmente, dos opciones para desarrollarla: SolidWorks y SolidEdge. Ambos entornos de trabajo permiten una implementación rápida y cómoda para nuestro objetivo de diseño en 3D y, además, poseen una interfaz similar. Sin embargo, la interfaz de SolidEdge nos ha parecido más intuitiva. Es por ello que, finalmente, nos hemos decantado por el uso de este.

En cuanto a los requisitos del diseño, se debe de tener en cuenta que esta pulsera está pensada para un uso diario y que, por lo tanto, no puede interferir en la realización de las tareas cotidianas. Deberá ser por ello una pulsera ligera y de fácil adaptación, que no modifique nuestra rutina.

### 4.2. SolidEdge, la opción más intuitiva.

La elección del entorno de trabajo SolidEdge frente a SolidWorks ha sido finalmente debido a su interfaz mucho más intuitiva. El programa seleccionado, al igual que SolidWorks, hace uso de métodos paramétricos para la creación de los diseños en 3D. SolidEdge incluye un modelador de sólidos, dibujos acotados y creación de ensamblajes. Además, ayuda a la eliminación de los obstáculos que cualquier programa de diseño puede presentar al pasar de un diseño en 2D a 3D. Es decir, podemos crear de una manera rápida nuestro diseño 3D desde un 2D.

Cabe destacar que el núcleo de modelado geométrico es muy potente, aportando esto una ayuda a la hora de la creación de un prototipo en 3D. Si hablamos del

ensamblado, podemos decir que este programa hace uso de varios métodos y de herramientas exclusivas que garantizan el almacenamiento del diseño original.



**Figura 4.1.** Imagen principal del programa empleado en el desarrollo de nuestros prototipos.

Como se ha comentado, SolidEdge tiene una interfaz muy intuitiva, lo que permite que se pueda manejar el programa, en nuestra opinión, de una forma bastante sencilla. Esto ha resultado ideal para personas que, como era nuestro caso, se están iniciando en el diseño 3D. Antes de comenzar con un diseño en uno de estos programas, es necesario precisar varias etapas, las cuales podemos dividir de la siguiente forma:

- Identificar y definir los requisitos de nuestros modelos. Una vez obtenido esto se realizará la conceptualización del modelo.
- Desarrollo y análisis del modelo desarrollado.
- Creación del prototipo. En nuestro caso se ha realizado en papel y a mano alzada el diseño de nuestros prototipos, para posteriormente introducir los datos en SolidEdge.
- Construcción real del modelo (3D).
- Últimas modificaciones para realizar mejoras o corregir errores.

En los siguientes apartados se presenta el diseño de la envolvente. Se presentarán dos diseños. En el primero de ellos no se ha modelado la correa, pues se empleará la empleada por otra solución comercial. En el segundo modelo se diseñan envolvente y correa. Gracias a las herramientas que posee este programa se podrá observar mediante una imagen renderizada como podría quedar nuestro diseño en la vida real (SolidEdge).

### 4.3. Desarrollo de prototipos.

Como ya se ha dicho anteriormente, en este apartado se hará el desarrollo e implementación de las dos propuestas que se ofrecen como contenedor de nuestra pulsera. La primera opción no modela la correa, pues se emplea una comercial disponible en el mercado. Nuestro diseño deberá por tanto adaptarse a esta correa. La segunda es una propuesta de diseño completa (correa y envolvente), con similitud a la gran oferta de pulseras inteligentes que nos podemos encontrar en el mercado.

#### 4.3.1. Desarrollo del primer modelo.

En el primer modelo que implementamos solo debemos de implementar la envolvente de la electrónica, que incluye nuestra placa y el sensor. Como correa se emplea una solución comercial, mostrada en la Figura 4.2, que se puede encontrar fácilmente. En concreto, dicha correa es la Polar A300. Por lo tanto, solo debemos de crear un contenedor que se amolde a dicha correa.



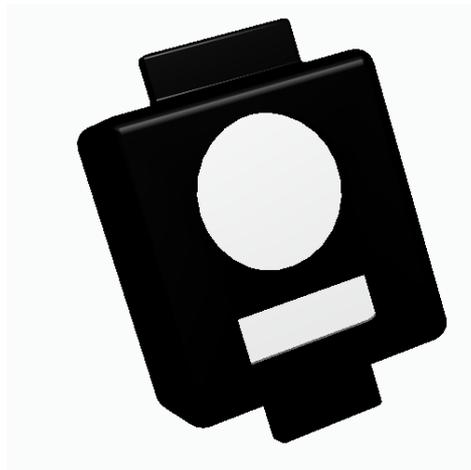
**Figura 4.2.** Correa Polar A300 de silicona negra

Se trata de una correa flexible, hecha en silicona y disponible en color negro. Para la creación de nuestro contenedor nos debemos de adaptar a la medida del hueco en la correa. El diseño realizado presenta unas dimensiones que le permiten quedar bien fijado en el hueco. Este diseño, mostrado en la Figura 4.3, carece de botones .



**Figura 4.3.** Prototipo de cubierta creado con SolidEdge, en el cual se respetan las medidas de la correa Polar A300.

En la Figura 4.3 se muestra el modelo final del contenedor de nuestra pulsera. Básicamente está compuesto de dos partes: un contenedor y su “tapadera”. Dentro de dicho contenedor es donde se colocarán la placa y el sensor. En la parte posterior del contenedor, como se muestra en la Figura 4.4, se han creado dos orificios el superior con forma redondeada para nuestro sensor y el inferior para el cargador.



**Figura 4.4.** Parte trasera del contenedor.

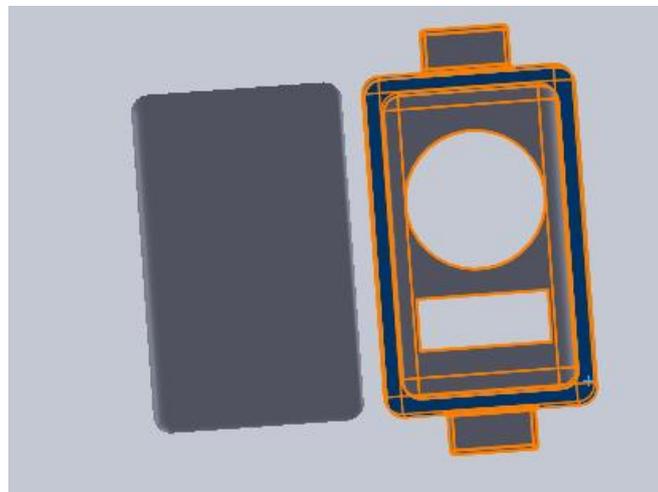
#### **4.3.2. Desarrollo del segundo modelo.**

En este segundo modelo no se condiciona el diseño por una correa comercial. Además, frente al primer diseño, nos hemos marcado como objetivo la creación de

un modelo más simple, pequeño y estético que el anteriormente creado. Como con el primer diseño, el objetivo es que el dispositivo no perturbe las actividades cotidianas.

En este caso hemos tenido que desarrollar la pulsera al completo, es decir, tanto la correa como el contenedor. Dicho contenedor, al igual que el anterior, está compuesto por dos partes y en él tienen que tener cabida tanto la placa, como el sensor y el sistema cargador.

En este diseño se eliminan totalmente los botones, siendo todas sus funciones táctiles. En primer lugar hemos comenzado con lo que, en mi opinión, puede resultar más fácil que en este caso es la implementación del contenedor.



**Figura 4.5.** Composición del contenedor, donde se pueden observar las dos piezas que lo componen.

Tal y como se ha descrito anteriormente, y como se muestra en la Figura 4.5, se pueden observar de forma diferenciada los dos componentes del contenedor. Un diseño sencillo, en el que nuestra placa y sensor quedan ajustados perfectamente. Al igual que en el diseño anterior el contenedor también contendrá dos orificios en la parte posterior con el mismo fin que en el modelo 1, albergar el sensor y el cargador de nuestra pulsera. Lo podemos observar también en la Figura 4.5.

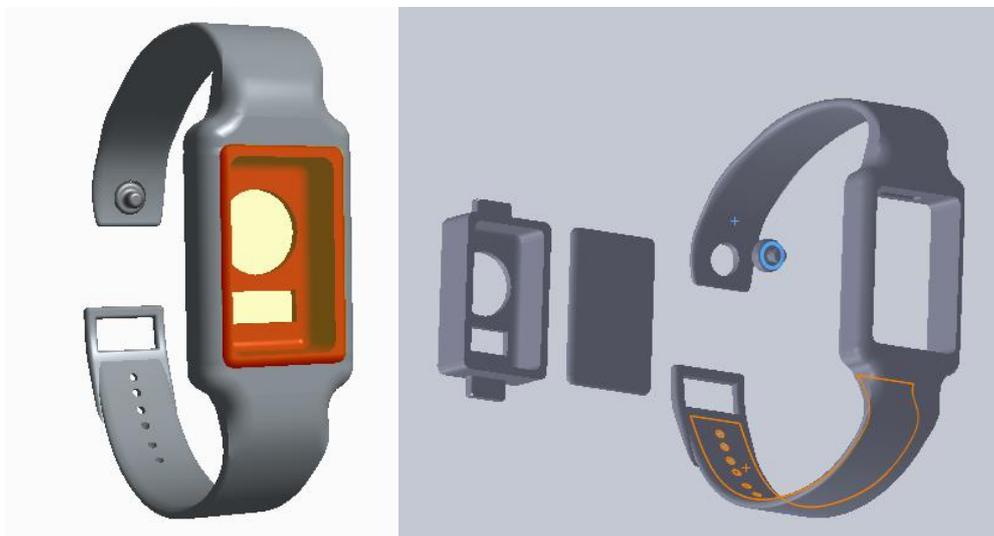
Por último se ha procedido a la creación de la parte en la que han surgido más complicaciones debido a las curvas que tiene. Se trata de la correa de nuestra pulsera. Se ha creado un diseño simple, que se adapta a la forma anatómica de la

muñeca para no interrumpir ninguna actividad y con una estética diferente a la opción anterior. El diseño de la pulsera es el que podemos observar en la Figura 4.6.



**Figura 4.6.** Capturas tomadas desde el programa una vez finalizado el proceso de implementación.

Para finalizar se muestra el conjunto de la correa con el contenedor, tanto de forma desglosada como con el contenedor ya empotrado en la correa. En la Figura 4.7, se puede ver el empotramiento de las tres piezas y nos podemos hacer una idea general de cómo quedaría.



**Figura 4.7.** A la izquierda se puede observar ya todas las piezas empotradas y obtenemos el resultado final, a la derecha podemos observar el desglose en piezas.

En la Figura 4.8 se puede observar la composición de tres piezas de nuestro diseño. La vista muestra nuestro modelo explosionado. Esto se ha podido conseguir gracias a las herramientas que nos ofrece SolidEdge.



**Figura 4.8.** Explosionado de nuestra pulsera, se puede observar claramente las piezas por las que está compuesta.

Por último, y gracias a las técnicas de renderizado de imagen que ofrece SolidEdge, se puede observar, en la Figura 4.9, cómo podría quedar nuestro prototipo de una forma real.



**Figura 4.9.** Prototipo real de la pulsera desarrollada, imagen renderizada a partir del modelo creado en SolidEdge.



## Capítulo 5

### Telemedicina: actualidad y futuro.

***A lo largo de este capítulo desarrollaremos el concepto general de telemedicina, y se hablará de lo que esta técnica ha proporcionado hasta el presente y de su posible evolución en un futuro. También se abordará el impacto que esta tecnología está teniendo en la sociedad.***

#### 5.1. ¿Qué es la telemedicina? Estado actual y perspectivas de futuro.

El término telemedicina empieza a emplearse durante los 70, y llega de la mano del “boom” tecnológico que suponen el ordenador, internet, etc. Nos encontrábamos ante una nueva tecnología capaz de romper barreras y crear una mayor accesibilidad a la sanidad, en especial para zonas rurales o en desarrollo.

La telemedicina se puede definir de muchas formas y cada una de ellas tendrá distintos matices, es por ello que haremos uso de la definición dada por la Organización Mundial de la Salud (OMS): *“La telemedicina es el suministro de servicios de atención sanitaria, en cuanto la distancia constituye un factor crítico, por profesionales que apelan a las tecnologías de la información y de la comunicación con el objetivo de intercambiar datos para hacer diagnósticos, preconizar tratamientos y prevenir enfermedades y heridas, así como para la formación permanente de los profesionales de atención a la salud y en actividades de investigación y de evaluación, con el fin de mejorar la salud de las personas y de las comunidades en las que viven”* (Organización Mundial de la Salud, OMS).

Este tipo de tecnología es hoy en día uno de los campos de investigación en los cuales se ha avanzado más en los últimos años. Por ello, se ha dividido en diferentes ramas para poder dar un mayor alcance, como son: teleeducación, telecirugía, teleconsulta o telemonitoreo. Dentro de esta última rama es donde podríamos encajar nuestro proyecto. Aunque a simple vista se podría decir que esta tecnología proporciona una gran cantidad de ventajas, también da lugar a una serie de desventajas. Para un mejor estudio de las ventajas/desventajas en este sector

debemos considerar tres tipos de actores: pacientes, profesionales y los propios centros de atención (Ruiz Ibañez, 2007).

A rasgos generales, el desarrollo de la telemedicina ha dado lugar a una serie de beneficios en el área de la salud como son los descritos a continuación:

- Acceso e intercambio de información. Una mejora del conocimiento, tanto de los facultativos como de la población, con respecto a las patologías.
- Acceso a la prestación de servicios en la salud. Sobre todo en aquellas zonas de difícil acceso o que carecen de una asistencia mínima, ha dado lugar al desarrollo de programas para ofrecer una atención sanitaria adecuada y eficiente.
- Mayor calidad de los servicios de salud. Se ha producido una disminución de costes, rápida disponibilidad, eficiencia (tanto en el tiempo de espera como en los profesionales) etc.
- Acceso a la educación continuada. Los profesionales están continuamente actualizando sus conocimientos y, por lo tanto, dan lugar a una mejor prestación de servicios.

Aun teniendo estas ventajas que a simple vista son favorables, la telemedicina también tiene hoy en día una serie de limitaciones como son:

- Disminución de la relación paciente-profesional. Provocado por el uso de las herramientas que nos ofrece la telemedicina.
- Disminución de la relación entre profesionales de la salud. Esto es debido a que ahora un profesional puede interactuar con otro directamente mediante el uso de esta tecnología.
- Tecnología impersonal. Así catalogan esta tecnología los pacientes que aún no tienen una gran confianza en este programa.
- Dificultades organizacionales y burocráticas. Con esto nos referimos a las distintas dificultades que te puedes encontrar para aplicar esta tecnología en diferentes partes del mundo, puesto que en cada país está regido por una serie de normas.
- Falta de desarrollo de protocolos y calidad de la información médica. No existe aún un protocolo específico para tratar a las personas mediante telemedicina. Por otra parte, aún se sigue pensando que la información

médica obtenida mediante esta tecnología es de menor calidad que la obtenida de forma presencial.

### **5.1.1. Ventajas y desventajas en pacientes, profesionales y centros de atención.**

A rasgos generales son tres los colectivos que hacen uso de este tipo de tecnologías: pacientes, profesionales y los centros de atención. Así, en este punto, se va a desarrollar de forma breve y directa las principales ventajas y desventajas que cada uno de estos colectivos puede encontrar al hacer uso de la telemedicina.

#### Pacientes

Comenzaremos hablando de nuestro primer colectivo: los pacientes. Las principales ventajas que se encuentran son:

- Reducción de las desigualdades por accesibilidad. Es decir, para personas que viven en zonas rurales o poco desarrolladas y para personas con movilidad reducida, esta tecnología supone un gran cambio debido a que tendrán acceso a algo que anteriormente no tenían o que suponía una gran dificultad para ellos.
- Diagnóstico y tratamiento más rápidos, reducción de tiempos de espera y ofrece la posibilidad de tener a los familiares cerca.
- Se crean nuevos modelos de relación profesional-paciente, dando lugar a un acercamiento a las nuevas tecnologías y al uso de Internet.

Las principales desventajas que encuentra son:

- Privacidad en la relación médico-paciente por problemas de seguridad en el manejo de datos.
- Bajo acceso a tecnologías en zonas menos desarrolladas o personas de la tercera edad.

#### Profesionales

El segundo colectivo en el cual repercute esta tecnología son en los profesionales que deben hacer uso de ella. Al igual que antes, para ellos tiene una serie de

ventajas y desventajas. Comenzamos hablando de alguna de las ventajas que ofrece, como son:

- Mejora en el diagnóstico, puesto que tiene más elementos a juicio para tomar decisiones. Además, el médico de familia o especialistas, pueden tener acceso a especialistas de otros centros que le ayuden en toma de decisión.
- Evita el desplazamiento del profesional a otro entorno. La imagen de innovación da lugar a una mayor confianza en el profesional provocando un aumento en la satisfacción de los usuarios.
- Acceso a programas de entrenamiento y educación médica, dando lugar a un aumento en el uso de las nuevas tecnologías.
- Acceso a la información de una forma fácil, rápida, barata y actualizada a través de esta nueva herramienta.
- Un mayor acceso a recursos y ahorro de tiempo en distintos proyectos.

Las limitaciones que ellos contemplan son:

- La responsabilidad del profesional se puede ver difuminada puesto que pueden actuar varios profesionales a la vez desde distintos sitios.
- Poca formación y falta de disponibilidad tecnológica.
- Mala adaptación de las tecnologías (por ejemplo disponer de una conexión de red lenta). Además requiere que las rutinas de trabajo y procesos se adapten a esta tecnología. Es por ello que, para su adaptación, se necesita bastante tiempo.
- Los profesionales deben de aceptarla. Es decir, en ocasiones este tipo de método carece de reconocimiento y requiere de tiempo extra (fuera de horario laboral) para poder adaptarse a ella.

### Centros

El último estudio sería las ventajas y desventajas que se producen sobre un centro de salud que posee este tipo de tecnología, pues éste se ve afectado por el uso que el resto de colectivos le están dando a esta tecnología. Al igual que antes, comenzamos hablando sobre las ventajas que ofrece a los centros como son:

- Se reduce la pérdida de exámenes o diagnósticos, puesto que todo queda automáticamente guardado en el sistema. Esto provoca una mejora en la gestión de salud y transparencia en el sistema.
- Mejor comunicación entre los distintos servicios de un centro de atención. Se produce una mejor utilización y aprovechamiento de los recursos existentes.
- Se produce una gran reducción de costes en transporte. Se renueva la imagen y se produce una utilización más eficaz de los equipos.

Al igual que el resto de colectivos también se presentan una serie de desventajas:

- Se requiere de recursos específicos, es decir, se necesita de un cambio en la estructura de gestión. Esto implica también la necesidad de formación para poder llevar este nuevo sistema.
- Resistencia del personal al cambio que se ofrece.
- Falta de infraestructura, lo que supone que adaptarse a esta tecnología sea una gran inversión de capital a lo que se le une una escasa financiación.

### **5.1.2. Aplicaciones de la telemedicina: actualidad y perspectivas de futuro.**

Aun teniendo en cuenta todas las desventajas o limitaciones que se presentan ante esta tecnología, la telemedicina está en auge y su desarrollo está siendo cada vez más rápido. Esta evolución también ha hecho que cambie la forma de envío de datos en esta tecnología, lo que empezó como un simple envío de datos y textos es, hoy en día, un envío por ejemplo de videos con audio, algo que mejora aún más la relación médico-paciente.

Otro papel importante es la infraestructura tecnológica de la que se esté haciendo uso, puesto que puede variar desde redes telefónicas públicas a tecnologías como telecomunicaciones vía satélite, microondas o transferencia asíncrona. Dentro de estas infraestructuras también encontramos la tecnología Bluetooth, de la que nosotros hacemos uso en este proyecto. Hoy en día, este estándar es muy usado y demandado entre los profesionales, puesto que ofrece una comunicación rápida de dispositivos con sensores (dispuestos en la denominada área corporal) con periféricos que sí pueden poner los datos en la red de área global. Con esta tecnología se puede insertar, acceder o monitorear, cómo es nuestro caso, información clínica del paciente.

Las aplicaciones médicas de la telemedicina se pueden extender a todas las ramas de la salud, desde la teleradiología o telepatología hasta la teleneurofisiología o teleoncología. Si queremos enmarcar nuestro proyecto en algún área lo podríamos hacer en telecardiología, una de las áreas que hoy en día tiene un mayor impacto. De esta forma, el dispositivo es capaz de ofertar desde teleconsulta a tiempo real hasta monitorización en su propio hogar como hace nuestro proyecto. En telecardiología podemos clasificar las aplicaciones en pre-hospitalarias, hospitalarias y post-hospitalarias, todos ellos basados en dispositivos de ECG y con capacidad de transmitir datos e información vía Web, telefónica o por redes inalámbricas. Con esto lo que se está consiguiendo son unos mejores diagnósticos y seguimiento de pacientes. Además de que se favorece la transmisión de información a lugares lejanos y siempre teniendo en cuenta su fácil almacenamiento. Nuestro dispositivo necesita de la existencia de un segundo dispositivo, al que nos conectamos, y que sí dispone de conexión a Internet.

Hablando de perspectivas de futuro, estamos hablando de una tecnología en auge y con un continuo crecimiento y desarrollo. Se realiza un constante planteamiento de herramientas y de modelos para la atención a pacientes con cualquier tipo de patologías.

El objetivo principal es la creación del conjunto de centro de referencia-centro satélite, es decir, una conexión remota entre dos puntos localizados en zonas geográficas distintas pudiéndose llegar a realizar un diagnóstico tanto del sistema como clínico. Cuando hablamos de diagnósticos de sistema, nos referimos a todas aquellas personas que necesitan verificar el estado o buen funcionamiento de un sistema, tales como implantes o marcapasos, siendo el propio paciente el que pueda verificarlo siguiendo los pasos indicados por el fabricante. Nos referiremos a diagnóstico clínico cuando su fin es verificar que los resultados obtenidos anteriormente son correctos, y que los datos que obtenemos se encuentran dentro de lo esperado. Uno de los ejemplos claros de este desarrollo nos lo encontramos en el área de teleaudiología. En 2016 encontramos un proyecto ya en fase de validación, con el que se pretende dar apoyo a aquellos pacientes que, por una serie de motivos, tienen dificultades para acceder a una consulta presencial (Dra. Marisela Cardier, 2016).

## **5.2. Impacto de la telemedicina en la sociedad.**

Puesto que estamos hablando de una tecnología en auge, que proporciona una mejora en la calidad de vida, nos ha parecido interesante ponernos en contacto con la sociedad para ver si lo conocen y que opinan sobre este tipo de tecnología. Para poder hacer esto se ha realizado una encuesta por Internet a la que tiene acceso todo el mundo.

Nos hemos intentado basar en una encuesta ya validada al español sobre este tema, pero no se ha encontrado ninguna que se adaptará a lo que buscábamos. Por lo tanto, hemos creado nuestra propia encuesta la cual se divide en tres partes:

- Primera parte. Datos de la población, se ha hecho un pequeño estudio de la población participante.
- Segunda parte. Preguntas generales sobre telemedicina, y conocimientos sobre algún tipo de proyecto o programa que hoy en día esté en funcionamiento.
- Tercera parte. Se presenta, de una forma breve y general nuestro proyecto, y preguntamos si saben de gente que este proyecto le podría ser beneficioso. Por último, se le pregunta si se encuentra alguna ventaja o desventaja, y en qué áreas creen que la telemedicina debería de avanzar.

El objetivo de esta encuesta, es saber si la sociedad está al tanto de las diferentes ayudas o programas con los que la sanidad puede contar actualmente y comprobar la opinión acerca de estas. Para tener un mayor alcance esta encuesta ha sido publicada por redes sociales donde todo el mundo ha tenido acceso y ha existido la posibilidad de que otras personas lo compartan para tener una mayor propaganda, también se han hecho encuestas a personas que han participado voluntariamente.

### **5.2.1. Resultado de la encuesta**

Como hemos descrito anteriormente, hemos dividido la encuesta en tres partes. Por lo tanto, iremos analizando y extrayendo los datos de cada una de ellas por separado.

## Primera parte

En esta sección simplemente hemos extraído datos de la población que ha participado en nuestra encuesta. Viendo los resultados, mostrados en la Tabla 1, podemos decir que la población femenina ha sido mucho más activa que la masculina.

Sexo	% de participación
Hombre	20,3%
Mujer	79,7%

**Tabla 1.** División por sexos de la población participante

También se han tenido en cuenta los diferentes rangos de edad de la población que ha participado, nuestra división y sus motivos son los siguientes:

- Menores de 20 años. Población en su mayoría estudiantil, que acaba de acceder a estudios universitarios o que está cursando otra serie de estudios, pero los cuales, en su mayoría, no se encuentran aún en el mundo laboral.
- Entre 20 a 30 años. Población la cual en su mayoría se encuentra cursando estudios superiores (FP, grados, máster...) o que, si aún no han salido al mundo laboral, lo hará en breve.
- Entre 30 a 50 años. Población activa trabajando.
- Más de 50 años. En este grupo la mayoría de la población lleva bastante tiempo en el mundo laboral.

Los resultados que hemos obtenido son los mostrados en la Tabla 2.

Edades	%de participación
Menos de 20 años	6,8%
Entre 20 a 30 años	60,8%
Entre 30 a 50 años	18,9%
Más de 50 años	13,5%

**Tabla 2.** División de la población participante por edades

Como conclusión de esta primera parte, hemos obtenido que la población más activa ha sido la femenina y en un rango de edades de 20 a 30 años.

## Segunda parte

Lo primero por lo que nos interesamos, es por saber si la población participante tiene alguna relación con el mundo sanitario. Esto lo hacemos, puesto que, si pertenecen a esta rama, la posibilidad de conocer de sistemas de telemedicina o de proyectos relacionados con éstos es mayor. Los resultados que obtenemos, Tabla 3, nos dan los siguientes datos:

Relacionados con la sanidad	% de la población
Pertenecientes a la sanidad	25.7%
No pertenecientes	75.7%

**Tabla 3.** Población relacionada con alguna rama de la sanidad

De toda nuestra población participante solo un 25.7% pertenece a la sanidad. La siguiente pregunta qué le hacemos es sobre la telemedicina: simplemente si saben lo que es o no. En la Tabla 4 podemos observar que la población que sabe que es la telemedicina es aún menor que la población perteneciente a la sanidad.

Con esto queremos decir que, de las personas que saben lo que es la telemedicina, no todas tienen que pertenecer al sector sanitario. Además, de aquí podemos sacar otro dato interesante y es que no toda la población que pertenece a la rama de la sanidad sabe a qué nos referimos con telemedicina.

Conocen la telemedicina	% de la población
Sí	24%
No	76%

**Tabla 4.** Tanto por ciento que sabe que es la telemedicina

Para todas estas personas que nos han contestado que sí saben que es la telemedicina, se les ha pedido que den una breve definición de lo que es. Como resumen hemos sacado estas tres definiciones:

- Respuesta médica ante algún problema de salud haciendo uso de los medios de comunicación.
- Medicina online y no presencial.

- Obtención de diagnóstico sin la necesidad de asistir presencialmente a una consulta.

Lo siguiente que preguntamos a nuestra población es si conocen algún sistema o proyecto relacionado con la telemedicina. En este punto podemos observar que la población que conoce alguno de estos sistemas se reduce aún más, tal y como se muestra en la Tabla 5.

Conoce algún sistema/proyecto	% de población
Sí	12%
No	88%

**Tabla 5.** Personas que conocen algún sistema/proyecto relacionado con la telemedicina

Trabajando con este 12% de la población que conocen algún sistema de telemedicina, se les ha pedido que nombre alguno de los que conozca y nos den una pequeña definición de la función que tienen los sistemas que nombren. Los resultados más obtenidos son:

- Teleasistencia, “ley de dependencia”. También conocido por el “botón rojo”. Se trata de un sistema que se oferta, en su mayoría, a personas mayores las cuales tienen cierto rango de dependencia. Es un medallón el cual incorpora un botón que va conectado con un sistema de emergencia. Al pulsarlo, la persona y el sistema de emergencia se ponen en contacto y son estos los encargados de seguir con el resto de protocolo como por ejemplo avisar a la ambulancia en caso de ser necesario o ponerse en contacto con los familiares.
- Programas de tv. Programa de televisión dedicados a la teleconsulta, donde una persona expone unos síntomas y se le da un posible diagnóstico.
- Sistema ADA. Aplicación en la que se introducen unos síntomas iniciales, esta te va haciendo una serie de preguntas hasta dar con el diagnóstico adecuado.
- TherapyChat. Se trata de una aplicación de ayuda psicológica.

Como conclusión de esta segunda parte podemos obtener, que, aunque la telemedicina sea una tecnología en auge, muy poca población tiene conocimiento

sobre ella y las posibilidades que ofrece. Incluso se puede observar que, aunque una parte de nuestra población en estudio pertenezca al sector sanitario, el tanto por ciento de personas que conocen la telemedicina es menor. Con esto, queremos decir que incluso una población dedicada a la sanidad no sabe nada acerca de la telemedicina.

### Tercera parte

En esta última parte se ha expuesto brevemente el enfoque de nuestro proyecto hacia la telecardiología, puesto que puede ser un sistema de seguimiento del paciente desde su propio hogar. Se ha introducido a la población de una forma breve en nuestro proyecto para obtener una opinión sobre éste.

En primer lugar nos hemos interesado por saber si alguien en nuestra población conoce de personas que necesitan un seguimiento regular debido a alguna patología cardíaca. Tal y como se muestra en la Tabla 6, casi la mitad de nuestra población participante sabe de alguien que necesite asistir constantemente a un centro sanitario para monitoreo.

<b>Conoce a personas con patologías cardíacas</b>	<b>% de la población</b>
Sí	45.3%
No	54.7%

**Tabla 6.** Muestra el % de población que conoce a alguien que se ve afectado por patologías cardíacas, las cuales necesitan de un seguimiento de sus constantes.

Es por ello, que una vez presentado nuestro proyecto el cual se puede ver como una rama de la telemedicina que sería la telemonitorización con especialización en telecardiología, queremos saber si la población vería un proyecto así beneficioso para aquellas personas afectadas por estas patologías. Lo que estamos ofreciendo es poder realizar una telemonitorización desde su propio hogar, sin tener que acudir tan a menudo a un centro de salud y reduciendo el tiempo de espera del paciente. Las respuestas obtenidas, Tabla 7, son las siguientes:

<b>Considera este sistema beneficioso</b>	<b>% de la población</b>
Sí	94,7%
No	5,3%

**Tabla 7.** Muestra los tantos por ciento de la población que opina que sí/no nuestro sistema puede ser beneficioso para personas con patologías cardíacas.

Como podemos comprobar la gran mayoría de la población participante haría uso de este tipo de tecnología si la tuvieran a su disposición. Pero como sabemos la tecnología hoy en día sigue causando cierta incertidumbre, por lo tanto también hemos querido saber qué ventajas y desventajas son las que se le encuentra a este proyecto. Las principales ventajas que destaca la población son las siguientes:

- Rapidez en el diagnóstico, eficiencia e intervención directa.
- Comodidad para el paciente puesto que se realiza la monitorización desde su propio hogar bajo las indicaciones médicas. También ahorro de tiempo, tanto por parte del paciente al tener que estar esperando, como para el centro sanitario. Y todo esto conduce a un aumento en la calidad de vida.
- Es el sistema el que se adapta al entorno del paciente y no el paciente el que se adapta al entorno.
- Control de las constantes indicadas sin que sea necesario el desplazamiento del paciente, sobre todo cuando se trata de paciente en áreas rurales o mal comunicadas o pacientes de movilidad reducida.
- Vigilancia y seguimiento constante del paciente en su propio hogar.
- Reducción de las colas de espera en los centros de salud, al que apuntan como uno de los grandes problemas de la sanidad hoy en día.

Pero todo aquello que tiene unas ventajas también produce unas desventajas, en este caso las más repetidas son:

- Puede suponer un gran coste para la sanidad o para el paciente.
- Las personas mayores pueden tener dificultad a la hora de manejar las nuevas tecnologías.
- Pérdida del trato médico-paciente, pasando a tener una relación impersonal.
- Rotura, mal funcionamiento o desconfiguración del sistema, ya sea por fallo del sistema, por mal uso o por no seguir las directrices marcadas por el especialista.
- Fallo en la conexión Bluetooth y por lo tanto en el envío de datos.
- Fallo en la detección de una anomalía y por lo tanto se produzca un mal diagnóstico.

- Desconfianza por parte del usuario sobre si este sistema puede ser efectivo o pensar que tiene un alto porcentaje de fallo.

Para finalizar nuestro cuestionario, sabiendo ya o teniendo una ligera idea de que es la telemedicina, le hemos preguntado en qué áreas creen que podría ser beneficioso este tipo de tecnología. En su mayoría han contestado que se deberían de implantar este tipo de sistemas en todas las áreas en general para mejorar en todo aquel aspecto en el que el paciente esté sujeto a un centro médico, puesto que creen que da lugar a un aumento en la calidad de vida. Pero también se han recibido respuestas más específicas como las siguientes:

- Geriátrica. Este tipo de instrumentos podría aumentar la calidad de vida de la tercera edad, por ejemplo, reduciendo la cantidad de veces que deben de ser trasladados a un centro de salud.
- Salud mental. Hablan sobre la creación de algún tipo de sistema que ayude a las personas de este área a tener un control, por ejemplo, de los brotes que sufran.
- Neurología, oncología y obstetricia.
- Cardiología. Sin ninguna duda el área más repetida debido al aumento en la sociedad de este tipo de patologías. Tras la presentación de este tipo de proyecto, hace plantearse a la gente porque aún no se hace uso de ello.

Como conclusión de esta tercera parte podríamos decir, que el aumento de la población con cardiopatologías hace que las personas cada vez estén confiando más en estas tecnologías, de las que opinan que, ante todo, se gana calidad de vida.

### **5.2.2. Conclusión general sobre la encuesta.**

Aunque hemos sacado pequeñas conclusiones sobre cada una de las partes de la encuesta, podemos obtener una conclusión general más elaborada.

Aunque la telemedicina está en auge y en pleno desarrollo, tal y como podemos observar es muy poca gente la que tiene conciencia de que es y a qué niveles puede llegar. Como se ha podido observar en las tablas anteriores, sólo una pequeña población pertenecía al sector sanitario, los cuales tienen una mayor oportunidad de estar en contacto con este tipo de tecnología.

Sin embargo, cuando se les preguntaba si conocían qué es la telemedicina, nuestra población se reducía aún más, es decir, personas que trabajan en el propio sector sanitario no sabían sobre lo que se les estaba hablando. Pero aún se sigue reduciendo mucho más nuestra población cuando se les pregunta sobre algún proyecto relacionado con esta tecnología.

Si es cierto, en la última parte, cuando se presenta el proyecto, que la población se ve receptiva a la hora de aceptar este tipo de proyectos y en los que ven grandes ventajas para la sociedad. En su mayoría, lo ven algo favorable pero aún se nota la desconfianza existente en la tecnología y en la posibilidad de fallo. Las ganas de adaptarse a las nuevas tecnologías y la falta de confianza.

Podemos decir por lo tanto, que aunque la telemedicina este en auge y vivamos en una era tecnológica, aún necesita dar un gran salto a la sociedad y que todos se hagan eco de la tecnología que tenemos y de la que no se está haciendo uso. De las posibilidades que puede ofrecer, bien por desconocimiento o falta de confianza, no se está haciendo un uso al 100%. Su desarrollo e implantación podría llegar a aumentar nuestra calidad de vida.

## Capítulo 6

### Conclusiones y trabajo futuro.

***Este Capítulo incluye las conclusiones que hemos ido extrayendo a lo largo del desarrollo de este Trabajo Fin de Grado. Por último, se expondrán varios puntos de estudio para trabajos futuros.***

Con este trabajo se ha querido realizar un estudio paso a paso del diseño de una pulsera inteligente, un dispositivo que, en la actualidad, ha llegado a convertirse en un instrumento de uso cotidiano. El trabajo integra en un mismo elemento el mundo de la medicina con el de la tecnología, dos grandes industrias que hoy en día avanzan a pasos agigantados.

Es importante destacar que el presente trabajo no arranca desde cero. Además de los conocimientos adquiridos en distintas asignaturas cursadas durante la carrera, el sistema se basa en el proyecto ASIST-E, financiado por el CDTI/CTA y llevado a cabo por la Universidad de Málaga en colaboración con la empresa MAGTEL. En dicho proyecto se diseñó una pulsera inteligente que ya empleaba algunos de los elementos básicos del presente Trabajo Fin de Grado (fundamentalmente el sensor nRF51822 de Nordic). Con estas bases, se ha llevado a cabo un proyecto personal de aprendizaje que, como principales contribuciones incluye:

- La inclusión de un servicio estándar BLE para enviar la señal del pulso (servicio *Heart Rate*). En ASIST-E se incluía esta medida en un servicio a medida, donde se integraba con otros parámetros.
- Eliminación del diseño original del sensor de temperatura, que no proporcionaba una medida fiable.
- El diseño de la envolvente usando SolidEdge
- Un estudio sobre la sociedad y la telemedicina.

Considerándolo un proyecto de aprendizaje, este Trabajo nos ha permitido introducirnos en el mundo de la electrónica y las comunicaciones inalámbricas, nos

ha obligado a entender el sistema ASIST-E original en el que nos basamos, y a modificarlo finalmente en función de nuestros propios objetivos. El resultado es una pulsera más barata que el original.

Con lo que respecta al pequeño análisis que se ha hecho de la telemedicina en la sociedad, reflejado en el Capítulo 4, se ha extraído una conclusión principal, en este caso se ve que, pese a los grandes avances de la tecnología y su repercusión en la medicina, la sociedad aún no está al día sobre las posibilidades, métodos y técnicas que ofrece la telemedicina. Y aunque sí que una pequeña parte de la población en estudio ha sido capaz de dar una descripción breve de qué significa este término, aún menos han sido capaces de nombrar o describir un proyecto que se base en esta tecnología.

También en el capítulo anteriormente nombrado, queda reflejada la gran desconfianza que aún tiene la población en la tecnología, que aún se nota más cuando se habla de poner una vida en manos de ésta. Esto ha quedado reflejado en el cuestionario: cuando se ha preguntado sobre las desventajas que pueden tener este tipo de sistemas, la respuesta más repetida ha sido *“la posibilidad de fallo en cualquier momento y la repercusión que esto tendría sobre una vida humana”*.

A gran escala podemos decir que las conclusiones extraídas de la realización de este proyecto son:

- Ampliación de conocimientos en lo que respecta a las tecnologías inalámbricas como ha sido BLE. Estudio a fondo de esta tecnología: en que se basa, protocolos, perfiles...
- Adquisición de nuevos conocimientos y elementos mediante el uso de la placa Tiny-BLE y el sensor de pulso. En especial gracias al trabajo con el sensor, se ha realizado un estudio más en profundidad sobre las características del pulso cardíaco y se ha tratado con una técnica, con la que hasta entonces no había tratado en profundidad, como es la fotoplestimografía.
- Estudio del proceso de creación del código que implementa las funciones del pulsera inteligente, basándonos en el proyecto ASIST-E.
- Creación de una encuesta para la obtención de datos de la población y sus conocimientos sobre la telemedicina y la gran oferta que tiene.

- Implementación del contenedor para nuestra pulsera, en la cual se ha hecho uso de una herramienta que anteriormente no había usado como ha sido SolidEdge. Para ello nos hemos basado en los conocimientos anteriormente adquiridos en la carrera con SolidWorks y gracias a esto hemos podido llegar a manejar esta nueva herramienta sin problemas.

En lo referente a los objetivos planteado para la realización de este proyecto, los cuales se citarán a continuación, podemos decir que se han cumplido:

- Estudio de requisitos mínimos para la realización del proyecto.
- Estudio en profundidad de los elementos y tecnologías empleados para su desarrollo.
- Estudio del código paso a paso en el que nos hemos basado, en este caso el creado para el proyecto ASIST-E.
- Hacer uso de la técnica de fotoplestimografía para la obtención de datos y envío de estos mediante tecnología BLE a un servidor, que será típicamente un Smartphone pero podría ser otro dispositivo (tablet o PC).
- Como objetivo optativo se había incluido la creación del contenedor de la pulsera, en este caso también ha sido cumplido y para el cual se ha usado un programa de CAD como es SolidEdge.

Por último, y para cerrar este capítulo, hacemos referencia a las perspectivas y trabajo de futuro. Al tratar con tecnología que está en constante cambio y crecimiento, y aplicarla además en un escenario, el de la telemedicina, que no para de crecer debido a los nuevos avances, estamos delante de un proyecto cuyo marco general estará en continuo avance y ligado a los cambios de futuro. Las líneas de investigación futuras pueden ser numerosas, y deben de tenerse en cuenta diferentes aspectos. Se plantean aquí algunas opciones, como pueden ser:

- Desarrollo del código para otros tipos de sensores que sean también necesarios para el estudio de otras patologías.
- Tanto la opción de eliminar la Tiny-BLE y sustituirla por los componentes que, de ella, se necesitan, como la continua miniaturización de estos componentes, obligarían a cambiar la envolvente para adaptarla a estas nuevas dimensiones.

- Implementación de códigos más sencillos, que sean flexibles y fáciles de mantener.
- Estudio de nuevas tecnologías inalámbricas aplicable a este tipo de proyectos.

## Referencias bibliográficas

Seeed Studio, Tiny BLE (<https://www.seeedstudio.com/Seeed-Tiny-BLE-BLE-%2B-6DOF-Mbed-Platform-p-2268.html>, Recuperado: 25 de Mayo de 2018)

J. Allen, A.Murray. "Age-related changes in peripheral pulse timing characteristic at the ears, fingers and toes". Journal of Human Hypertension (2002) 16, pp. 711-717.

Pulse sensor (<https://pulsesensor.com/>, Recuperado: 25 de Mayo de 2018)

Puras Mallagray, E. "Plestimografía. Presiones segmentarias". Sección V. Fundación hospital Alcorcón (Madrid).

López Silva, S.M. et al. (2005). "Fotoplestimografía por transmisión con múltiples diodos láser en el infrarrojo cercano durante el ejercicio físico". Óptica pura y aplicada, Vol. 38 num. 1.

Martín Sánchez, D. "Diseño de un dispositivo para la detección del estrés a partir de la señal de fotoplestimografía". Trabajo fin de grado, Universidad de Sevilla.

Angulo Duato, C. (2013-14). "TFG: Diseño e implementación de un pulsómetro digital basado en la fotoplestimografía". Universidad Politécnica de Valencia.

Barea Cañizares, A (2016) " Diseño, desarrollo y test de un prototipo de pulsera para adquirir el electrocardiograma y la onda de pulso" . Universidad Politécnica de Barcelona.

Melchor Rodríguez, A. Ramos Castro, J. "Análisis de la variabilidad de la frecuencia cardiaca mediante fotoplestimografía por imagen". Grupo de Instrumentación y bioingeniería, Campus Nord C4, Universidad Politecnica de Cataluña, Barcelona, España.

Tintín Durán, I. (2015). "Diseño y elaboración de un prototipo de monitor de signos vitales aplicando métodos no invasivos con comunicación de datos a dispositivos móviles". Universidad Politécnica Salesiana (Cuenca, Ecuador), pp.6-21.

Nordic Semiconductor ASA (2014). "Manual: nRF51822 Multiprotocol Bluetooth Low Energy/2.4GHz RF System on Chip".

Cousseau, I. (Agosto 2015). "Diseño de filtros analógicos". Universidad nacional del sur, Bahía blanca (Argentina).

Bandera Rubio, A. (2017). "Electromedicina: Diseño de un dispositivo wearable". Grado en Ingeniería de la Salud, Universidad de Málaga.

Macias Macias, M. et al. (Julio 2014). "La plataforma MBED para la enseñanza electrónica aplicada al diseño de productos". XI Congreso de Tecnología, Aprendizaje y Enseñanza de la electrónica.

Flores Carbajal, E. (Octubre de 2012). "Redes y sensores inalámbricos aplicados a la medicina". Trabajo fin de máster, Universidad de Cantabria.

Yera Pemán, I. Reinoso, A. (2016). "Aplicación Android para la gestión de notificaciones con un dispositivo tipo miband". Revista de Ciencia, Tecnología y Medio ambiente, Vol. XIV. Universidad Alfonso X el Sabio (Madrid).

Técnica industrial especial electricidad y electrónica (Octubre 2014). "Comunicación inalámbrica con Bluetooth". (pp. 18-23).

Akhayad, Y. (8 de Febrero de 2016). "Título del TFG: Bluetooth 4.0 Low Energy: Analisis de las prestaciones y aplicaciones para la automoción". Universidad Politecnica de Cataluña.

Garijo, L. Falone, F. (30 de Junio de 2016). "TFG: Diseño y estudio de un sistema de comunicación inalámbrica basado en tecnología Bluetooth Low Energy con desarrollo de protocolo propio de enrutamiento". Universidad Pública de Navarra, Pamplona.

"Sensor medidor de aceleración: acelerómetro". Universidad de Sevilla (pp.39-40).

Acelerómetro. Electrónica y mecatrónica. (URL: [electronicaestudio.com](http://electronicaestudio.com), Recuperado: 16 de Septiembre de 2018)

TeraTerm (URL: <https://ttssh2.osdn.jp/index.html.en>, Recuperada: 16 de Septiembre de 2018)

Nordic nRF Connect (URL: <https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF-Connect-for-Desktop>, Recuperada: 16 de Septiembre de 2018)

SolidEdge. Diferencias entre SolidWork y SolidEdge (URL: <https://esquivaesto.com/diferencias-entre-solidworks-y-solid-edge/>, Recuperada: 9 de Julio de 2018)

Organización Mundial de la Salud, OMS. (URL: <http://www.who.int/es>, Recuperado: 15 de Septiembre de 2018).

Ruiz Ibañez, Carlos. (Enero-Junio 2007) “Telemedicina: Introducción, aplicación y principios de desarrollo”. Medellín, Colombia (pp. 77-93).

Prados Castillejo, Jose A. “Telemedicina, una herramienta también para el médico de familia”. Unidad de Gestión clínica Lucano (pp. 129-132).

Dra. Marisela Cardier (2016). “Telemedicina. Estado actual y perspectivas futuras en audiología y otología”. Universidad de Navarra (pp. 840-847).



## ANEXO I – CUESTIONARIO ONLINE SOBRE TELEMEDICINA

### PARTE 1.- DATOS GENERALES

Grupo de edad al que pertenece: \*

- menos de 20 años
- entre 20 y 30 años
- entre 30 y 50 años
- más de 50

Sexo \*

- Hombre
- Mujer

Últimos estudios cursados: \*

¿Pertenece o ha estudiado algo relacionado con el ámbito sanitario? \*

- Sí
- No

### PARTE 2.- TELEMEDICINA

¿Sabe que es la telemedicina? \*

- Sí
- No

En caso de haber contestado "Sí" a la pregunta anterior, ¿podría dar una breve descripción de lo que entiende por telemedicina?

La telemedicina se divide en diferentes áreas tales como: teleprevención, telediagnóstico, telemonitoreo, teleconsulta... ¿Conoce de algún proyecto o sistema ya desarrollado que pertenezca a este área? \*

- Sí
- No

En caso de haber marcado "Sí" en la pregunta anterior ¿Podría indicar el nombre del proyecto o sistema?  
¿Podría explicar de forma breve en que consiste?

¿Conoce o sabe de alguien que haga uso de algún sistema relacionado con la telemedicina? \*

- Sí  
 No

### PARTE 3.- PROYECTO BASADO EN TELEMONITOREO

Se presenta un proyecto de telemonitoreo, con esto queremos decir que se pueden controlar una serie de constantes fisiológicas sin necesidad de asistir a un centro de atención sanitaria y por lo tanto, haciéndolo desde tu propio hogar mediante las pautas anteriormente indicadas por un facultativo. En este caso se crea un proyecto de telecardiología, un sistema que mide el pulso cardíaco y que puede hacer un envío de datos mediante el uso de Bluetooth, permitiendo así una rápida detección del problema en caso de anomalía. El sistema que se crea contempla no estorbar en las tareas cotidianas, de fácil manejo y de bajo consumo.

¿Conoces a alguien con algún tipo de patología cardíaca que deba de estar asistiendo a un centro de salud para un seguimiento de las constantes? \*

- Sí  
 No

¿Crees que este tipo de proyecto podría ser beneficioso para estas personas? \*

- Sí  
 No

A rasgos generales, ¿Le encuentras alguna ventaja a este sistema?¿Cuales? \*

**A rasgos generales, ¿Le encuentras alguna desventaja a este sistema? ¿Cuales? \***



**Por último, puesto que la telemedicina es algo que en este momento se encuentra en auge ¿En que área/especialidad crees que debería de avanzar para ofrecer unos mayores beneficios a la sociedad?**

