



UNIVERSIDAD DE MÁLAGA



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

DISEÑO DE UN ELECTROCARDIOGRAFO PARA EL
ESTUDIO DE SU APLICACIÓN EN IDENTIFICACIÓN
BIOMÉTRICA

DESIGN OF AN ELECTROCARDIOGRAPH FOR THE STUDY OF
ITS APPLICATION IN BIOMETRIC IDENTIFICATION

Realizado por
Marta González Aguilar

Tutorizado por
Antonio Jesús Bandera Rubio
Camilo Andrés Ruiz Beltrán

Departamento
Electromedicina
UNIVERSIDAD DE MÁLAGA

MÁLAGA, mayo de 2023



UNIVERSIDAD
DE MÁLAGA





ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
INFORMÁTICA

GRADUADA EN INGENIERÍA DE LA SALUD

**DISEÑO DE UN ELECTROCARDIOGRAFO
PARA EL ESTUDIO DE SU APLICACIÓN EN
IDENTIFICACIÓN BIOMÉTRICA**

**DESIGN OF AN ELECTROCARDIOGRAPH FOR
THE STUDY OF ITS APPLICATION IN BIOMETRIC
IDENTIFICATION**

Realizado por

Marta González Aguilar

Tutorizado por

Antonio Jesús Bandera Rubio

Camilo Andrés Ruiz Beltrán

Departamento

Electromedicina

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2023

Fecha defensa: Junio 2023



UNIVERSIDAD
DE MÁLAGA





Agradecimientos

Quiero agradecer, en primer lugar, a mi tutor Antonio Bandera por haberme guiado y acompañado, sin ti no hubiera podido desarrollar este proyecto. Tu compromiso y dedicación han sido una fuente constante de inspiración y motivación para mí. Gracias por compartir tus conocimientos y por creer en mi potencial. A mis cuatro voluntarios, no tengo palabras suficientes para expresar mi gratitud hacia vosotros. Vuestra generosidad, tiempo y paciencia han sido esenciales para llevar a cabo este proyecto. A mi familia, vuestra confianza plena en mis capacidades me ha dado la fuerza y la determinación necesarias para superar cualquier obstáculo. Gracias por estar a mi lado en cada paso del camino y por creer en mí incluso cuando ni yo lo hago. Y finalmente, a toda la gente buena que me rodea en mi día a día, quiero agradecerles de corazón. Vuestra presencia, apoyo y palabras de aliento me han dado la confianza necesaria para seguir adelante y creer en mí misma. Vuestra bondad y positividad me han recordado constantemente que no estoy sola y que siempre hay personas dispuestas a ayudar y creer en los demás.

Marta González Aguilar

Málaga, 2023



UNIVERSIDAD
DE MÁLAGA



Resumen

La situación actual en el control de acceso de la mayoría de las aplicaciones o servicios web, se encuentra regulada por técnicas convencionales basadas en contraseñas, tarjetas de crédito y firmas, que son susceptibles de suplantación. Una posible solución, dado que no se basa en las técnicas de uso común que utilizan características extrínsecas de las personas, es la biometría. En este momento, se están investigando y utilizando rasgos fisiológicos, como la huella dactilar, el iris y la cara. Pero estos rasgos son falsificables mediante lentes de contacto, impresión de huellas etc. Recientemente, se está buscando dar un enfoque más exclusivo mediante el uso de las señales eléctricas del corazón. Por ello, se ha sugerido el Electrocardiograma (ECG), una prueba que posee numerosas características interpretables para la identificación entre individuos. En el presente Trabajo de Fin de Grado se propone realizar un estudio de la aplicación del ECG como nuevo método de identificación, planteando un estudio desde el diseño de un electrocardiógrafo con transmisión Bluetooth, hasta la búsqueda de parámetros diferenciadores mediante algoritmos automáticos.

Dado que el estudio se enfoca en un análisis conceptual de la inserción del ECG como nuevo control de acceso, se toman pocas muestras de ECG reales, procesados y limpiados para aplicar posteriormente un algoritmo de aprendizaje supervisado como es el de los K vecinos más cercanos (K-Nearest Neighbors). Este busca clasificar una muestra desconocida basándose en la similitud con sus vecinos más cercanos, en este caso, una métrica de distancia entre señales.

Finalmente, y pese a tratarse de un prototipo, se logró un rendimiento de un 97,22% de precisión en la clasificación, con lo que abre la posibilidad a que en líneas futuras se pueda perfeccionar con otras técnicas. Llegando, a implementarlo en un dispositivo que pueda ser usado como método identificativo de uso diario.

Palabras clave: Electrocardiograma, Electrocardiógrafo, Procesamiento de señal, Biometría, Clasificación, Bluetooth

Abstract

The current situation in the access control of most web applications or services is governed by conventional techniques based on passwords, credit cards and signatures, which are susceptible to spoofing. One possible solution, since it does not rely on commonly used techniques using extrinsic characteristics of individuals, is biometrics. At the moment, physiological traits, such as fingerprint, iris and face, are being investigated and used. But these traits are falsifiable by means of contact lenses, fingerprinting etc. Recently, a more exclusive approach is being sought by using the electrical signals of the heart. Therefore, the Electrocardiogram (ECG), a test that has numerous interpretable features for identification between individuals, has been suggested. In this Final Degree Project we propose a study of the application of the ECG as a new method of identification, from the design of an electrocardiograph with Bluetooth transmission, to the search for differentiating parameters by means of automatic algorithms.

Since the study focuses on a conceptual analysis of the insertion of the ECG as a new access control, few samples of real ECGs are taken, processed and cleaned in order to subsequently apply a supervised learning algorithm such as K-Nearest Neighbors. This seeks to classify an unknown sample based on the similarity with its nearest neighbors, in this case, a distance metric between signals.

In the end, and despite being a prototype, a performance of 97.22% classification accuracy was achieved, which opens the possibility that in future lines it can be improved with other techniques. It could even be implemented in a device that could be used as an identification method for everyday use.

Keywords: Electrocardiogram, Electrocardiograph, Signal processing, Biometry, Classification, Bluetooth

Índice

Agradecimientos	5
Resumen	7
Abstract	8
Índice	9
Índice de figuras	12
Introducción.....	15
1.1 Motivación	15
1.2 Objetivos	16
1.3 Estructura de la memoria	17
Conceptos básicos	18
2.1. Relevancia del electrocardiograma	18
2.2. Métodos de captura del electrocardiograma.....	20
2.2.1 Electroodos.....	20
2.2.2. El Triángulo de Einthoven. Derivaciones.....	21
2.2.3. Tipos de electrocardiogramas	23
2.3. Mecánica y evaluación de las señales	24
2.4. Los electrocardiógrafos	26
2.4.1. Tipos y sistemas de captación de señales	26
2.5. La biometría.....	27
2.5.1. Modalidades biométricas	27
2.5.2. Aplicaciones de la biometría	28
Metodología y fases de trabajo	30
3.1. Requisitos de diseño del ECG.....	30
3.1.1. Requisitos funcionales.....	30
3.1.2. Requisitos no funcionales.....	31
3.1.3. Solución adoptada.....	31

3.2. Metodología de tratamiento de datos	32
Diseño del prototipo	38
4.1. Arquitectura física.....	39
4.1.1. Selección de componentes.....	39
4.1.2. Montaje del sistema.....	42
4.2. Arquitectura software	44
4.2.1. Captura de datos y envío usando Bluetooth de Baja Energía	44
4.2.2. Preprocesado de datos.....	50
4.2.3. Identificación de usuarios.....	54
Resultados obtenidos	56
5.1. Captura del ECG	56
5.2. Preprocesado del ECG en MATLAB.....	57
5.3. Pruebas de identificación	60
5.3.1. Mejora del algoritmo de clasificación	63
Conclusiones y trabajo futuro	67
6.1. Conclusiones.....	67
6.2. Trabajo futuro.....	68
Anexo 1 - Código Keil Studio Cloud	69
Anexo 2 - Código filtro Butterworth.....	71
Anexo 3 - Código segmentación	72
Anexo 4 - Código K-Nearest Neighbours (selección aleatoria).....	74
Anexo 5 - Código K-Nearest Neighbours (selección no aleatoria).....	76
Referencias	78

Índice de figuras

Figura 2.1. Factores de riesgo cardiovascular	18
Figura 2.2. Fallecimientos por enfermedad cardiovascular 2003-2020	19
Figura 2.3. Triángulo de Einthoven . En la parte superior se usa un electrodo positivo y otro negativo para obtener el ECG desde tres medidas (tres derivaciones). En la inferior, se usa un electrodo positivo y un punto virtual para obtener también el ECG desde tres medidas (tres derivaciones). Las derivaciones usadas en la parte superior son bipolares y, las de la parte inferior, unipolares.....	22
Figura 2.4. Derivaciones precordiales	23
Figura 2.5. Captura con un dispositivo 1L-ECG	24
Figura 2.6. a) Etapas señal ECG b) Partes del corazón implicadas.....	25
Figura 2.7. Clasificación modalidades biométricas	28
Figura 3.1. Conexión del Heart Monitor de Hailege (basado en el AD8232) con el Arduino DUE	33
Figura 3.2. Filtro Butterworth paso bajo de orden n	35
Figura 4.1. Arquitectura física del sistema. Se organiza en torno a los bloques de captura y acondicionamiento y la unidad digital de procesado.	39
Figura 4.2. Diagrama de bloques funcionales del AD8232	40
Figura 4.3. Configuración del AD8232 en modo monitorización cardiaca	40
Figura 4.4. Respuesta en frecuencia para el circuito en configuración monitorización cardiaca.....	41
Figura 4.5. Placa tiny BLE.....	42
Figura 4.6. Sistema construido en torno al AD8232 de Hailege conectada a la Tiny-BLE de Seeed Studio que contiene el nRF51822 de Nordic	43
Figura 4.7. Conexión de electrodos al sistema	43
Figura 4.8. ECG capturado usando el sistema propuesto	44
Figura 4.9. Página de entrada al IDE Keil Studio Cloud de ARM	45
Figura 4.10. Aspecto de las ventanas que conforman el IDE.....	46
Figura 4.11. Monitorización de la conexión BLE-UART usando nRF-Connect	49

Figura 4.12. Monitorización de la conexión BLE-UART usando nRF-Connect	50
Figura 4.13. Filtro Butterworth paso bajo, $n = 4$ y $F_c = 30$ diseñado	52
Figura 4.14. Señal ECG capturada y filtrada	52
Figura 4.15. Señal ECG representada por segmentos	53
Figura 4.16. Señales ECG antes y después de seleccionar las muestras dentro de los parámetros establecidos	54
Figura 5.1. Señales ECG originales de una de las personas estudiadas	57
Figura 5.2. Señales ECG filtradas con el filtro Butterworth desarrollado	58
Figura 5.3. Detección de complejos QRS proporcionada por la función <code>pan_tompkin</code>	59
Figura 5.4. Preprocesado de la señal del ECG llevado a cabo por la función <code>pan_tompkin</code>	60
Figura 5.5. Matriz de confusión cuando $K=5$, precisión= 82,35%	61
Figura 5.6. Matriz de confusión cuando $K=3$, precisión= 76,47%	61
Figura 5.7. Matriz de confusión cuando $K=3$, precisión = 70,58%	62
Figura 5.8. Matriz de confusión cuando $K=2$, precisión = 87.50%	63
Figura 5.9. Matriz de confusión cuando $K=2$, precisión= 82,85%	64
Figura 5.10. Matriz de confusión cuando $K=3$, precisión= 94,28%	65
Figura 5.11. Matriz de confusión cuando $K=5$, precisión=91,42%	65
Figura 5.12. Matriz de confusión cuando $K=4$, precisión= 97,22%	66

1

Introducción

1.1 Motivación

Hoy en día, hay una gran variedad de aplicaciones del mundo real que dependen, para protegernos, tanto a nosotros mismos como nuestras pertenencias (sean objetos físicos, datos o información personal), de sistemas de reconocimiento. Varias de estas aplicaciones siguen dependiendo de sistemas ya considerados como tradicionales, y basados en elementos o conocimientos extrínsecos, como tarjetas, llaves, pines o contraseñas. Sin embargo, estos mecanismos tradicionales presentan varios problemas de usabilidad y seguridad. Por ello, las representaciones sustitutas de la identidad real ya no se consideran suficientes en muchas ocasiones.

Como resultado de lo descrito anteriormente, se ha producido un aumento en el interés por el campo del reconocimiento biométrico. Por reconocimiento biométrico se entiende la identificación automática de personas basada en sus características fisiológicas distintivas (por ejemplo, cara, huella dactilar, iris, geometría de la mano) y de comportamiento (por ejemplo, modo de andar, firma, pulsación de teclas). Uno de los rasgos biométricos más comunes es la huella dactilar. Sin embargo, aunque su uso supuso un avance significativo, sigue habiendo problemas relacionados con la usabilidad y fiabilidad de la huella dactilar, originados principalmente porque la adquisición de una

huella dactilar suele ser de mala calidad y, además, puede ser fácilmente burlada por un especialista hábil.

En concreto, el electrocardiograma (ECG) es una señal originada internamente y única en cada persona, por lo que tiene el potencial de ser una fuente fiable para la biometría [1,2]. Usado profusamente para monitorizar el estado cardiovascular de una persona, los retos que debe afrontar el ECG para poder ser usado como rasgo biométrico son la extracción de características relevantes y fiables de las señales de ECG y el diseño de modelos precisos para la comparación de ECGs, que protejan al individuo contra ataques de identidad.

En este contexto surge el principal objetivo de este Trabajo Fin de Grado, que consistiría en diseñar un electrocardiógrafo de tamaño reducido que constaría de un número reducido de receptores, y con capacidad de transmitir los datos por Bluetooth. Además, en este Trabajo Fin de Grado se investigará el potencial de emplear información basada en ECG como rasgo biométrico para la identificación y autenticación de individuos.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado es diseñar y montar el prototipo de un electrocardiógrafo y evaluar si el ECG obtenido por este dispositivo permite, sin embargo, realizar un análisis lo suficientemente exhaustivo como para distinguir entre distintos individuos. Esta respuesta se obtendrá de manera no invasiva y en un breve intervalo de tiempo. Se usarán electrodos conectados a un *front-end* analógico, y se dispondrá de un microcontrolador como unidad central de proceso, encargada de gestionar la captura del dato y su transmisión usando Bluetooth de Baja Energía (BLE).

Este electrocardiógrafo se empleará para recopilar la información de diferentes personas y, con estos datos, se evaluará su capacidad para identificación biométrica. En primer lugar, se deberá hacer un estudio documental del comportamiento de las señales eléctricas del corazón, para así partir de una base sólida antes de analizar las respuestas reales. A continuación, evaluaremos las señales obtenidas, identificando puntos relevantes del ECG, y segmentando un patrón que pueda usarse para clasificación. Posteriormente, se empleará un clasificador basado en técnicas propias de la Inteligencia Artificial para identificar las principales diferencias entre los ECGs de distintos individuos. De esta forma se evaluarán los resultados obtenidos para su

aplicación al mundo de la identificación biométrica. Esta evaluación vendrá limitada por las características del estudio, restringido a un número reducido de usuarios pues estaríamos en una fase muy preliminar de diseño e implementación.

1.3 Estructura de la memoria

Para satisfacer el objetivo final de este Trabajo Fin de Grado se deberán cumplir los subobjetivos que han sido marcados en el Apartado 1.2. Dicho cumplimiento se refleja en esta memoria, estructurada de la siguiente manera:

- Capítulo 1. Introducción. Se detallan las motivaciones del proyecto y objetivos a cumplir.
- Capítulo 2. Conceptos básicos. Se hablará sobre el electrocardiograma. Su origen como método para monitorizar el estado del corazón. La mecánica y evaluación de las señales. Los electrocardiógrafos. Tipos y sistemas de obtención de señal. La biometría.
- Capítulo 3. Metodología y fases de trabajo. Requisitos de diseño del ECG. Pruebas y validaciones. Soluciones y propuesta adoptada.
- Capítulo 4. Diseño del prototipo. Se detalla la arquitectura física y software.
- Capítulo 5. Resultados obtenidos
- Capítulo 6. Conclusiones y trabajo futuro

Conceptos básicos

2.1. Relevancia del electrocardiograma

El corazón, órgano único y motor del aparato circulatorio en los animales, suele ser, sin embargo, poco cuidado en el día a día de las personas. En los últimos datos recopilados por la SEC (Sociedad Española de Cardiología) en 2020, se registró un aumento de la mortalidad cardiovascular, coincidiendo con la pandemia del COVID-19 [3]. Este estudio afirma que casi el 60% de los españoles tiene dos o más factores de riesgo vascular. Algunas de las enfermedades relacionadas son la hipertensión, la diabetes, el sobrepeso, la falta de sueño, el estrés... En la Figura 2.1 se muestra la proporción de sufrir una enfermedad cardiovascular en función del factor de riesgo.

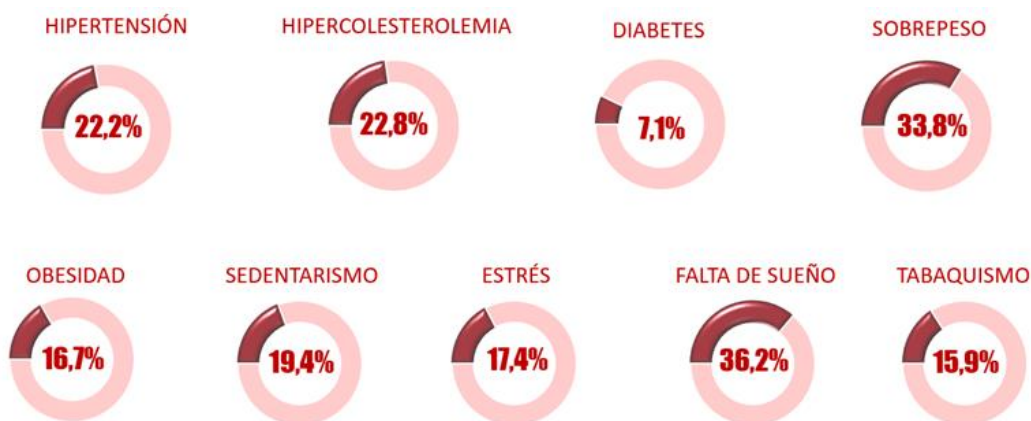


Figura 2.1. Factores de riesgo cardiovascular [2]

Padecer varios factores de riesgo vascular es más frecuente en la población de mayor edad. Por ejemplo, el 70,4% de los españoles con edades >55 años tienen dos o más factores de riesgo, dato que baja hasta el 53,9% en la franja de edad comprendida entre los 36 y los 54 años, y al 41,6% en los españoles de entre 16 y 35 años [1].

Desde el punto de vista temporal, en la Figura 2.2 se observa cómo, con el paso de los años, hay un claro descenso de las enfermedades cerebrovasculares pero un incremento de muertes por enfermedades hipertensivas, insuficiencia cardiaca y otras enfermedades del corazón. En estas primeras se registraron 14.271 muertes en 2020, con un relevante incremento respecto al año anterior, donde fueron 11.854.

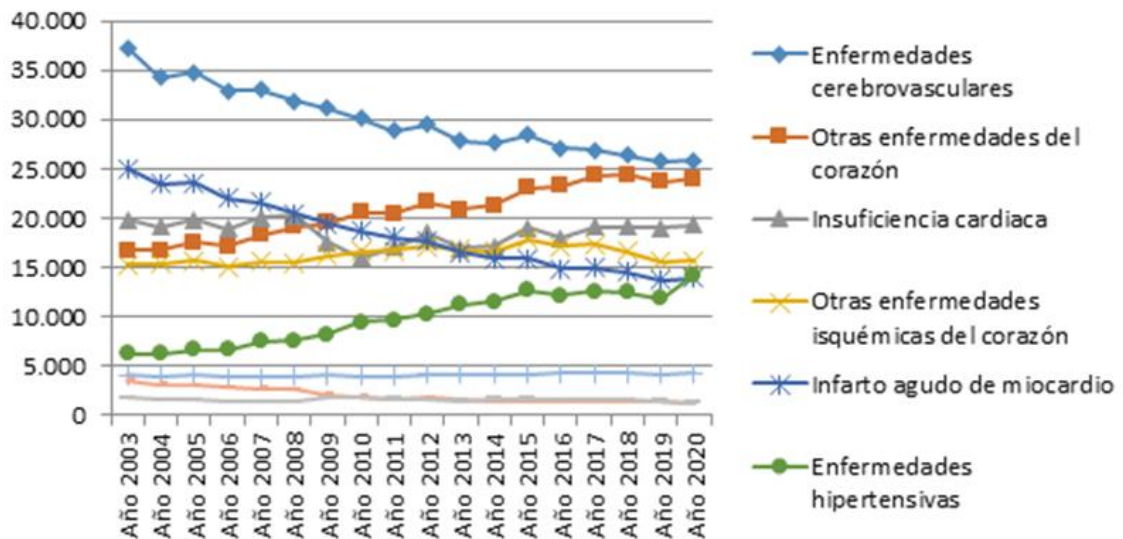


Figura 2.2. Fallecimientos por enfermedad cardiovascular 2003-2020 (Datos de la Sociedad Española de Cardiología)

Por lo tanto, tomando como base este estudio se concluye que la mortalidad no ha parado de crecer en los últimos 15 años por estas causas. Analizando los factores detrás de la misma, es importante destacar que, en gran parte, está en manos de la población contribuir a su descenso.

Algunos de los hábitos que contribuyen al bienestar del sistema cardiovascular son seguir una buena dieta, practicar ejercicio físico, abandono del hábito tabáquico, control de los niveles de estrés etc. Pero no todo puede controlarse con un cambio de hábitos, por lo que se deberá recurrir a la tecnología para monitorizar el estado del corazón en cada momento. En este caso, el electrocardiograma (ECG) es la señal más relevante para los profesionales. Este tipo de señales son obtenidas mediante el electrocardiógrafo. La electrocardiografía es una prueba médica que mide y registra la

actividad eléctrica del corazón, y se utiliza para detectar y diagnosticar problemas cardiacos. Un ECG puede mostrar la rapidez y regularidad de los latidos del corazón, pero también el tamaño y la posición de las cavidades cardíacas y la presencia de cualquier daño en el músculo cardíaco.

2.2. Métodos de captura del electrocardiograma

La captura del ECG se realiza mediante la colocación de electrodos en la piel del paciente, que registran la actividad eléctrica del corazón y la transmiten a un dispositivo de registro, como un electrocardiógrafo o un monitor de ECG. El método de colocación de los electrodos puede variar dependiendo del tipo de ECG que se quiera realizar. En este apartado se describe brevemente qué son los electrodos, la forma de colocarlos sobre el cuerpo, y los tipos de ECGs que existen.

2.2.1 Electrodos

Los electrodos son dispositivos conductores que se utilizan para conectar un circuito eléctrico a un medio conductor. Estos dispositivos se utilizan en una amplia variedad de aplicaciones, desde la medicina hasta la industria y la investigación científica.

En el ámbito de la medicina, los electrodos del ECG son dispositivos esenciales para medir la actividad eléctrica del corazón. Estos se colocan en diferentes posiciones en el pecho, los brazos y las piernas del paciente y pueden utilizar, o no, un gel conductor para transmitir las señales eléctricas desde el corazón hasta el dispositivo de registro. De hecho, se clasifican en función de si usan o no este gel, por lo que existen dos tipos principales: húmedos o secos.

- Los **electrodos húmedos** son aquellos que requieren un gel conductor para transmitir las señales eléctricas del cuerpo al dispositivo de registro. Este gel conductor, a menudo llamado gel de electrodo o pasta conductora, se aplica en la piel del paciente antes de colocar el electrodo, y ayuda a mejorar la calidad de la señal eléctrica capturada. Los electrodos húmedos se utilizan comúnmente en electrocardiogramas (ECG), electroencefalogramas (EEG), electromiografías (EMG) y otras aplicaciones médicas que requieren una buena calidad de señal.
- Los **electrodos secos**, por otro lado, no requieren de gel conductor para transmitir las señales eléctricas. Estos electrodos utilizan tecnología de

superficie seca, como sensores de contacto de metal o polímeros conductores, para detectar las señales eléctricas del cuerpo. A menudo se utilizan en aplicaciones de monitorización a largo plazo, como monitores cardíacos portátiles o monitores de sueño, ya que son más cómodos para el paciente y no requieren la aplicación de gel conductor en la piel.

En general, ambos tipos de electrodos tienen sus ventajas y desventajas. Los electrodos húmedos pueden proporcionar una mejor calidad de señal, pero pueden ser más incómodos para el paciente y requieren la aplicación de gel conductor. Los electrodos secos pueden ser más cómodos para el paciente, pero pueden tener asociados una menor calidad en la señal capturada, y requerir una mayor presión de contacto para detectar señales eléctricas precisas. En última instancia, la elección de un tipo de electrodo depende de la aplicación específica y de las necesidades del paciente.

2.2.2. El Triángulo de Einthoven. Derivaciones

En el diseño del primer electrocardiógrafo realmente práctico, Willen Einthoven (1860-1926) propuso utilizar tres puntos de medida para medir la actividad eléctrica del corazón: uno en el brazo derecho, otro en el brazo izquierdo y un tercero en la pierna izquierda (este patrón triangular (ver Figura 2.3) origina el nombre de 'triángulo' dado a esta configuración). Al no disponer de electrodos, en este primer dispositivo se sumergen las manos y un pie del paciente en unas cubetas de agua salada. La medida de la corriente se lleva a cabo con un galvanómetro de filamento o cuerda, verdadero avance tecnológico sobre el que descansaba el diseño del electrocardiógrafo.

El triángulo de Einthoven es importante porque permite medir la diferencia de potencial eléctrico entre dos puntos del corazón en cualquier momento dado. La actividad eléctrica del corazón se mide en forma de ondas, llamadas ondas P, QRS y T, y la diferencia de potencial eléctrico entre los electrodos se registra como una curva en el ECG. La medición de la actividad eléctrica del corazón a través del triángulo de Einthoven permite a los médicos diagnosticar trastornos cardíacos y monitorizar la salud cardíaca.

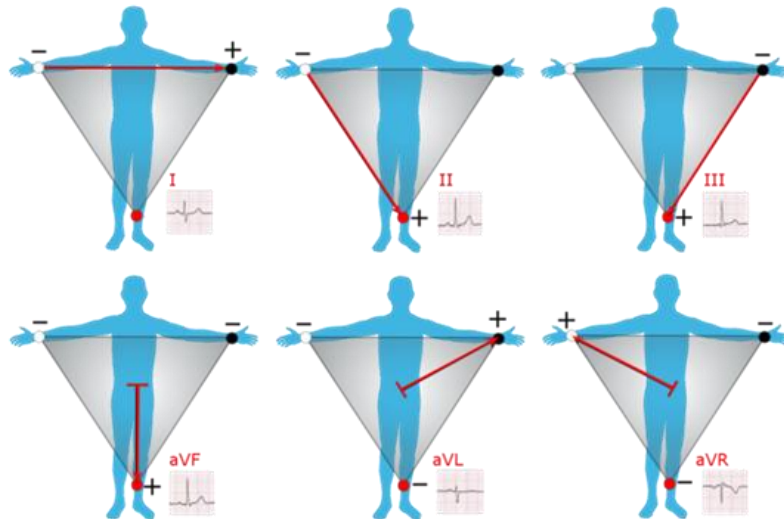


Figura 2.3. Triángulo de Einthoven [4]. En la parte superior se usa un electrodo positivo y otro negativo para obtener el ECG desde tres medidas (tres derivaciones). En la inferior, se usa un electrodo positivo y un punto virtual para obtener también el ECG desde tres medidas (tres derivaciones). Las derivaciones usadas en la parte superior son bipolares y, las de la parte inferior, unipolares.

El triángulo de Einthoven es una configuración, que requiere distintos electrodos colocados en distintas posiciones. Si cada medida concreta e individual es una derivación, existen diferentes derivaciones para ubicar sobre el cuerpo de la persona los electrodos utilizados en un electrocardiograma (ECG). Cada derivación permite el registro de una diferencia de potencial eléctrico entre dos puntos, sean los dos puntos realmente electrodos o solo uno de ellos (y el otro punto sea virtual, y obtenido de valores capturados con electrodos en otros puntos del cuerpo). Cada derivación no se debe analizar de forma independiente, sino en el conjunto de la configuración elegida, pues cada una de ellas permite obtener información sobre la misma actividad eléctrica del corazón desde una perspectiva diferente. Las principales derivaciones son:

- Derivaciones bipolares: son las más comunes y utilizan dos electrodos, uno positivo y otro negativo, para medir la actividad eléctrica entre dos puntos específicos del cuerpo. Las derivaciones bipolares más utilizadas son la DI, DII y DIII, que se obtienen colocando el electrodo positivo en diferentes posiciones del cuerpo (brazo izquierdo, pierna derecha, brazo derecho respectivamente) y el electrodo negativo en una posición común (pierna izquierda). Las tres derivaciones forman, en conjunto, el ya mencionado triángulo de Einthoven.
- Derivaciones unipolares: utilizan un solo electrodo positivo y una combinación de otros electrodos colocados en diferentes posiciones del cuerpo como

referencia para medir la actividad eléctrica del corazón. Las derivaciones unipolares más comunes son las de Wilson, que utilizan tres electrodos colocados en el brazo derecho, brazo izquierdo y pierna izquierda.

- Derivaciones precordiales: se colocan electrodos en diferentes posiciones en el pecho para medir la actividad eléctrica del corazón desde diferentes ángulos. Las derivaciones precordiales más comunes son V1, V2, V3, V4, V5 y V6, que se colocan en diferentes posiciones alrededor del pecho (Figura 2.4).

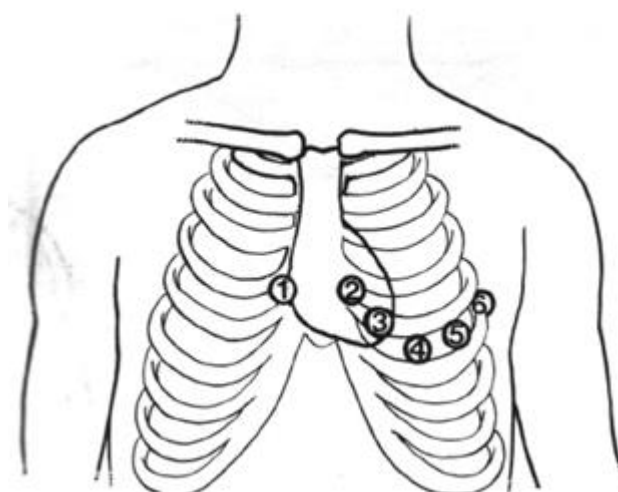


Figura 2.4. Derivaciones precordiales [5]

2.2.3. Tipos de electrocardiogramas

Existen diferentes tipos de ECG, fruto de usar distintas configuraciones de derivaciones. Ya no es común usar el triángulo de Einthoven como se muestra en la Figura 2.3. A continuación, se describen algunos de los ECG más comunes:

- ECG de 12 derivaciones: es el ECG más utilizado en la práctica clínica y consta de 12 derivaciones que permiten obtener información sobre la actividad eléctrica del corazón desde diferentes perspectivas. Estas derivaciones se dividen en tres grupos: las derivaciones bipolares estándar (DI, DII, DIII), las derivaciones unipolares precordiales (V1-V6), y las derivaciones unipolares de las extremidades (aVR, aVL, aVF).
- ECG de 3 derivaciones: es un ECG simplificado que utiliza solamente tres derivaciones para obtener información sobre la actividad eléctrica del corazón desde diferentes perspectivas. Las derivaciones utilizadas son DI, DII y aVF.
- ECG de 6 derivaciones: es otro ECG simplificado que utiliza seis derivaciones para obtener información sobre la actividad eléctrica del corazón desde

diferentes perspectivas. Las derivaciones utilizadas son DI, DII, DIII, aVL, aVR y aVF.

- ECG de Holter: es un tipo de ECG ambulatorio que registra la actividad eléctrica del corazón durante un periodo prolongado de tiempo (generalmente 24-48 horas) mientras el paciente realiza sus actividades diarias normales. Este tipo de ECG suele utilizar tres derivaciones: DI, DII y aVF.

Los electrocardiogramas de una sola derivación (1L-ECG) se utilizan cada vez más en entornos (pre)clínicos para la detección y el seguimiento de una serie de trastornos del ritmo y la conducción cardíacas [13]. En la actualidad existen múltiples dispositivos de 1 L-ECG con aprobación, por ejemplo, europea (CE) y/o americana (FDA), que están disponibles para los médicos, así como electrónica de consumo. Las diferencias entre estos dispositivos son múltiples e incluyen diferencias en el software, la conectividad del dispositivo (autónomo o controlado por el smartphone), y el hardware. Los dispositivos portátiles de 1 L-ECG se basan en la captura de una señal de ECG y un registro similar a la derivación DI de un ECG estándar de 12 derivaciones (ver Figura 2.3). Esto se obtiene sujetando dos electrodos metálicos con las dos manos o con varios dedos de cada mano, como se ilustra en la Figura 2.5. Es el método más simple de capturar el ECG, aunque también en ocasiones poco preciso. En un entorno de identificación biométrica, en la que se intenta disminuir la colaboración del usuario, debería ser la fórmula elegida para capturar la señal.

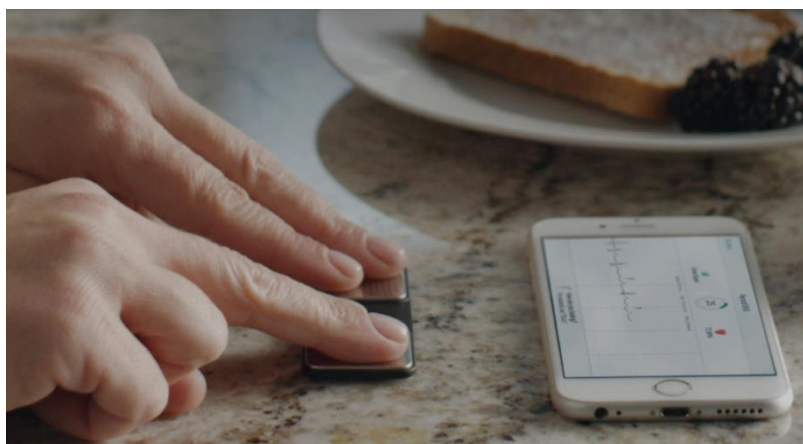


Figura 2.5. Captura con un dispositivo 1L-ECG [12]

2.3. Mecánica y evaluación de las señales

La forma convencional de un ECG está compuesta por diferentes ondas y complejos que representan la actividad eléctrica del corazón en las diferentes etapas

del ciclo cardíaco. El ciclo eléctrico cardíaco está formado por las siguientes etapas, representadas en la Figura 2.6 [6]:

- **Onda P:** Representa la activación del nodo sinoauricular (SA) y la dispersión de la señal eléctrica por todas las aurículas, lo que causa la despolarización de las aurículas.
- **Segmento PQ o PR:** Reproduce la transmisión de las señales eléctricas del nodo sinoauricular (SA) al auriculoventricular (AV).
- **Complejo QRS:** Indica la activación del nodo AV y la transmisión de la señal a través del haz de His. Representa la despolarización de los ventrículos (las cámaras inferiores del corazón)
 - **Onda Q:** Corresponde con la despolarización del tabique interventricular.
 - **Onda R:** Despolarización de la masa principal de los ventrículos.
 - **Onda S:** Última etapa de la despolarización ventricular en la base del corazón. La repolarización de las aurículas también ocurre en este momento, pero se oculta debido al gran tamaño del complejo QRS.
- **Segmento ST:** Momento en el que los ventrículos se contraen y bombean la sangre. Representa la etapa de equilibrio en los potenciales de acción miocárdicos.
- **Onda T:** Representa la repolarización de los ventrículos cuando se relajan después de la contracción o diástole ventricular.
- **Onda U:** La onda U representa la repolarización del músculo papilar. Pero esta no suele observarse y su ausencia suele ignorarse.

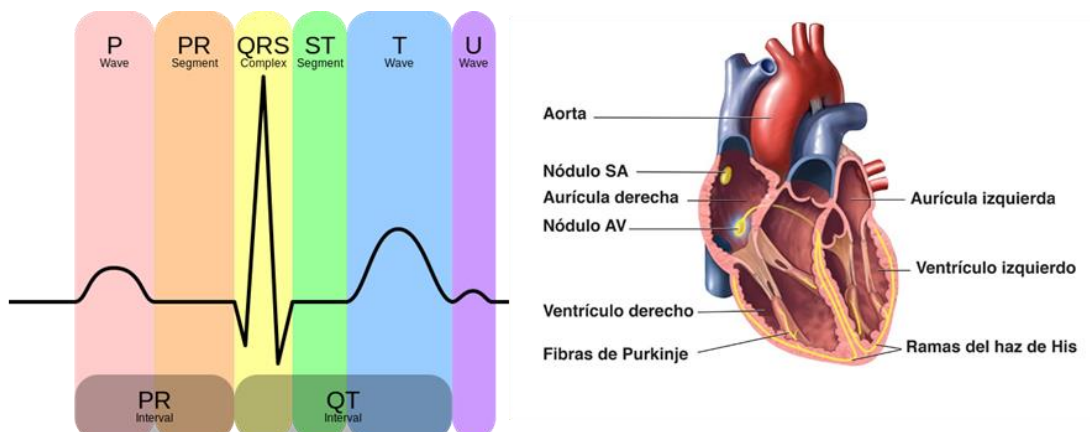


Figura 2.6. a) Etapas señal ECG [7] b) Partes del corazón implicadas [8]

2.4. Los electrocardiógrafos

Un electrocardiógrafo es un dispositivo médico utilizado para registrar la actividad eléctrica del corazón y producir una representación gráfica de esta actividad en forma de una señal electrocardiográfica o ECG. Es el sistema que recoge toda la información analizada en el apartado 2.3, y permite la visualización de los datos y el análisis al profesional médico.

Los electrocardiógrafos modernos a menudo también tienen características adicionales, como la capacidad de grabar y almacenar la señal ECG para su posterior análisis, y la capacidad de conectarse a sistemas de telemedicina, para permitir la monitorización remota de la actividad eléctrica del corazón.

2.4.1. Tipos y sistemas de captación de señales

Existen varios tipos de electrocardiógrafos, entre ellos:

1. **Electrocardiógrafos de un canal:** son los más básicos y solo registran un único trazo de la actividad eléctrica del corazón.
2. **Electrocardiógrafos de tres canales:** registran tres trazos simultáneamente, lo que permite ver diferentes aspectos de la actividad eléctrica del corazón.
3. **Electrocardiógrafos de 12 canales:** son los más avanzados y registran 12 trazos simultáneamente, proporcionando una vista más detallada de la actividad eléctrica del corazón y permitiendo la detección de patologías más complejas.
4. **Electrocardiógrafos portátiles:** son dispositivos pequeños y compactos que se pueden llevar a cualquier lugar y utilizados para realizar registros de la actividad eléctrica del corazón en el hogar o en ambientes no clínicos.
5. **Electrocardiógrafos inalámbricos:** son dispositivos que no requieren cables para conectarse a los electrodos y al equipo de registro, lo que proporciona mayor comodidad y libertad de movimiento al paciente.
6. **Electrocardiógrafos de alta resolución:** son electrocardiógrafos avanzados que utilizan técnicas especiales para detectar y registrar pequeñas variaciones en la actividad eléctrica del corazón, lo que permite la detección temprana de patologías cardíacas.

En este proyecto tomaremos como base el funcionamiento de un electrocardiógrafo de un canal por su simplicidad y portabilidad. Para este estudio los puntos de mayor complejidad se centran en la obtención de la señal de manera que sea fiable y con el menor nivel de ruido, de manera que su análisis sea lo más sencillo posible. Además, como ya se ha comentado al presentar los tipos de ECG, si queremos implementarlo como identificador, con una sola señal bastaría. El objetivo es demostrar que, a pesar de la simplicidad en la captura y la posible similitud entre las señales del ECG de cualquier individuo, existen rasgos diferenciadores.

2.5. La biometría

La biometría es un campo fascinante y en constante evolución que se dedica al estudio de la identificación de seres vivos mediante sus características físicas o de comportamiento. Esta disciplina utiliza tecnologías avanzadas para capturar y analizar datos biométricos únicos de cada individuo, como huellas dactilares, reconocimiento facial, iris, voz, y más. Estos datos se utilizan para establecer la identidad de una persona de manera precisa y confiable.

Una de las principales ventajas de la biometría es su alta precisión y seguridad. Las características biométricas son únicas para cada individuo, lo que significa que la probabilidad de error o suplantación es extremadamente baja. Esto hace que la biometría sea ampliamente utilizada en diversas aplicaciones, desde la seguridad informática y el control de acceso físico hasta la identificación personal en dispositivos móviles.

Además de su seguridad, la biometría también ofrece comodidad y conveniencia. A diferencia de las contraseñas o tarjetas de identificación, que se pueden perder u olvidar, los datos biométricos están siempre presentes en cada individuo. Esto permite un acceso rápido y sin problemas a lugares, servicios o dispositivos, lo que ahorra tiempo y simplifica los procesos de autenticación. En resumen, la biometría ofrece una forma confiable, precisa y conveniente de identificar a las personas, y su creciente adopción promete transformar la forma en que interactuamos con la tecnología y aseguramos nuestra información.

2.5.1. Modalidades biométricas

En biometría, la capacidad de distinguir entre individuos se logra mediante la utilización de cierta información física o conductual, lo cual se conoce como modalidad

biométrica. Según esto, las modalidades biométricas se dividen en dos tipos principales: biometrías fisiológicas y biometrías conductuales, como se ilustra en la Figura 2.7. Las biometrías fisiológicas se refieren a las características anatómicas o fisiológicas de los individuos, mientras que las biometrías conductuales se relacionan con el comportamiento o la forma en que el individuo lleva a cabo ciertas acciones [9].

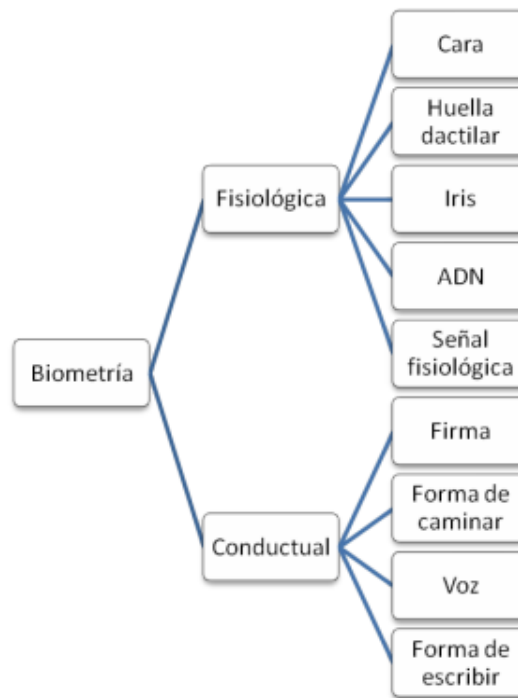


Figura 2.7. Clasificación modalidades biométricas

2.5.2. Aplicaciones de la biometría

Hoy en día, la biometría está presente en múltiples sectores como son la informática, las finanzas, la justicia etc.... Estas aplicaciones pueden ser agrupadas en cuatro categorías [10, 11]:

- **Control de acceso físico:** Se utiliza en áreas restringidas y permite el acceso mediante el uso de características biométricas. Empresas, hospitales, industrias y control de fronteras son algunos ejemplos de ubicaciones donde se ha implementado este tipo de control debido a su fiabilidad y eficiencia.
- **Control de acceso lógico:** Este tipo de control delimita el acceso a programas y datos, permitiendo que solo los usuarios autorizados puedan acceder a ellos. Es utilizado por empresas privadas y públicas, así como por servicios militares y gubernamentales.

- **Justicia y orden público:** Los sistemas biométricos se utilizan en aplicaciones como la identificación en escenas del crimen, la administración e ingreso en prisiones, y en otras aplicaciones forenses.
- **Control de asistencia:** La biometría también se puede utilizar para rastrear la ubicación de individuos o grupos de personas, lo que permite controlar el tiempo de trabajo y la ubicación de las personas.

3

Metodología y fases de trabajo

En este apartado, se evaluarán las necesidades y requisitos de diseño del prototipo de ECG de manera que permita la extracción y almacenamiento de señales. Posteriormente, se evaluará el criterio y metodología de tratamiento de los datos orientado a la identificación biométrica. Por último, se expondrá la solución adoptada.

3.1. Requisitos de diseño del ECG

En este apartado, se deben traducir las necesidades del proyecto a requisitos de diseño del sistema de obtención de señales ECG.

Puesto que se trata de un sistema para uso frecuente debe ser sencillo y económico, además de ser lo menos invasivo posible, intuitivo y manejable. Se deben cumplir con las necesidades mencionadas manteniendo la limpieza de los resultados. Estas necesidades se traducen en los siguientes requisitos funcionales y no funcionales:

3.1.1. Requisitos funcionales

- El sistema debe capturar señales ECG
- El sistema debe transmitir la señal capturada usando Bluetooth de Baja Energía

- El sistema debe representar las señales en PC
- El sistema debe obtener un patrón biométrico desde el ECG capturado
- El sistema debe comparar el patrón de una persona contra una base de datos.

3.1.2. Requisitos no funcionales

- El ancho de banda del ECG estará entre 0.5 y 40 Hz
- La frecuencia de muestreo será de 100 Hz
- La alimentación del sistema será inferior o igual a 5V
- La arquitectura física del dispositivo se diseña a nivel de módulos funcionales

3.1.3. Solución adoptada

Una vez evaluados los requisitos del sistema, y analizado el comportamiento anatómico del corazón y el entorno de funcionamiento del sistema, se diseña una metodología de trabajo que se basa en las siguientes premisas o tecnologías:

- **ECG de 1 canal:** Debido a que solo es necesaria una derivación, se usarán 3 electrodos ubicados según la teoría del Triángulo de Einthoven (apartado 2.2). Los electrodos se colocan en los dedos índices de las manos derecha e izquierda. Opcionalmente, un tercer electrodo se coloca en la pierna derecha (tobillo). Se evaluará la pérdida de nitidez en la señal si se omite este tercer electrodo.
- **Electrodos húmedos:** Aunque para una aplicación de identificación podrían ser más prácticos los electrodos secos (presentes en dispositivos comerciales como Kardia [12]), no ha sido factible hacerse con ellos para este proyecto. Por ello, se han usado electrodos húmedos. La solución prioriza por tanto la comodidad del paciente, con la reducción de electrodos a 3, pero también incrementa la calidad de la señal con la selección de electrodos húmedos frente a los secos.
- **Front-end analógico AD8232:** bloque de acondicionamiento de Analog Devices, específicamente creado para tratar con señales ECG. Está diseñado para extraer, amplificar y filtrar las señales en presencia de perturbaciones tanto externas como internas [15], aportando una reducción notoria del ruido. Se integra en nuestro sistema usando un módulo de Hailege (fabricado siguiendo el mismo esquema que propone Sparkfun [14]).
- **ARM nRF51822 de Nordic:** Este Cortex M-0 de Nordic es un dispositivo con las prestaciones suficientes para abordar el problema de la captura de la señal con

un consumo reducido. Cuenta con la ventaja de disponer de un transceptor integrado a 2.4GHz, que permite implementar el protocolo BLE. En nuestro sistema este ARM se integra usando la placa Tiny-BLE de SeeedStudio.

- **Keil Studio Cloud:** El software empotrado en el nRF51822, de captura de la señal analógica del ECG y envío usando BLE, se desarrolla usando el entorno online de programación ARM Keil Studio Cloud [18]. Este entorno evoluciona del sistema operativo MBED, empleado en la asignatura de Electromedicina del Grado en Ingeniería de la Salud, y obsoleto desde diciembre de 2022.
- **MATLAB:** Una vez se disponga de un conjunto de ECGs capturados usando nuestro sistema a un reducido número de personas, se deberá generar el patrón biométrico asociado al ECG y abordar el proceso de identificación. Para agilizar el trabajo en estas dos tareas se apuesta por emplear el entorno MATLAB de desarrollo, gracias al conjunto de toolboxes que ofrece para aplicaciones de reconocimiento de patrones.

Todos estos aspectos se ampliarán en el Capítulo 4 de la presente memoria. A continuación, en el apartado 3.2, se resume la metodología seguida, en lo referido al tratamiento de datos, para abordar el trabajo a desarrollar en este proyecto. Al documentar este apartado se proporcionan detalles obtenidos conforme se han ido cerrando las distintas fases.

3.2. Metodología de tratamiento de datos

En este proyecto se han seguido distintas fases para obtener un sistema de identificación basado en ECG. La metodología del análisis biométrico con ECG se resume en la adquisición de los datos de ECG de la persona, seguido por la extracción de características y la selección de un modelo para la identificación. Posteriormente, se lleva a cabo la comparación de las características extraídas. A continuación, se exponen paso a paso las pautas seguidas para cumplir dicho objetivo.

Fase 1: Obtención de datos

En esta fase se recopilaban las señales ECG de pocos individuos (en este caso se han tomado 10 señales de 362 valores de 4 personas). Como se trata de un estudio introductorio a la identificación biométrica bastará con estos datos. Para llevar a cabo la captura se usó una versión preliminar del prototipo que se describe en el Capítulo 4, basado en el mencionado AD8232 pero que empleaba un Arduino DUE para digitalizar

y almacenar los datos capturados (Figura 3.1). Esto nos permitía trabajar en paralelo en el procesamiento de los datos (la programación del Arduino se basa en el código propuesto en [12] y permitió disponer de datos de ECG de forma muy rápida), y en cerrar el diseño del dispositivo físico (la plataforma ARM Keil Studio Cloud era nueva para nosotros y debíamos migrar lo desarrollado en MBED para gestionar la transmisión BLE).

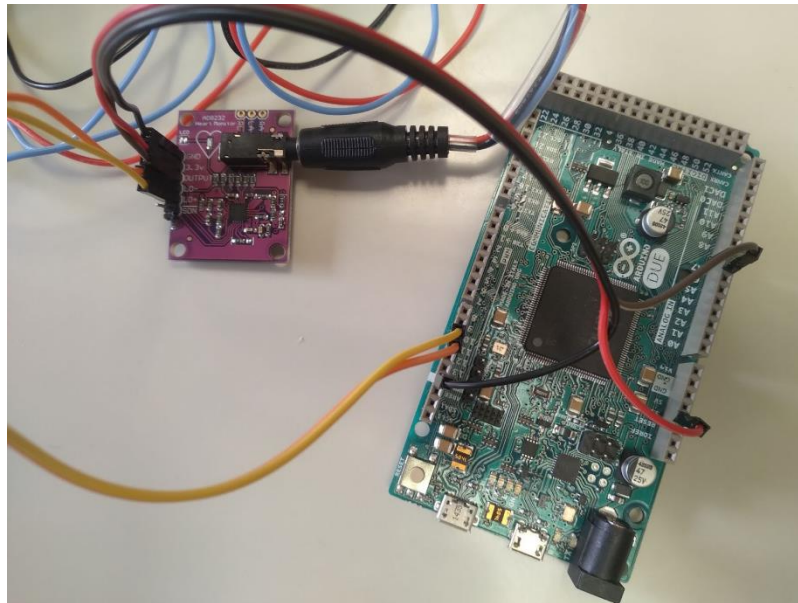


Figura 3.1. Conexión del Heart Monitor de Hailege (basado en el AD8232) con el Arduino DUE. El esquema de conexión sigue la propuesta en [19]

Los datos capturados se han pasado a formato “.mat” para analizarlos usando MATLAB. Como se ha mencionado, se escogió este software tras un análisis entre distintas opciones, en busca del entorno que tuviese más paquetes relacionados con el procesamiento de la señal del electrocardiograma.

Fase 2: Preprocesamiento de la señal

Antes de la identificación de los patrones ECG, uno de los puntos más importantes para el tratamiento de datos es el filtrado. Los filtros en este caso son utilizados para mejorar la calidad de la señal y reducir el ruido en los registros de la actividad eléctrica del corazón, lo que puede hacer que sea más fácil detectar anomalías en la señal y mejorar la precisión del diagnóstico. En este estudio se ha empleado un filtro Butterworth (ver Figura 3.2).

Filtro Butterworth

El filtro Butterworth es un filtro digital de uso frecuente en procesamiento de señales y filtrado de datos. Se caracteriza por tener una respuesta de frecuencia suave

y una caída gradual en la atenuación hacia las frecuencias de corte. Como ocurre con otros filtros, presenta múltiples opciones de diseño con diferentes órdenes, donde el orden se refiere a la cantidad de polos o pares de polos complejos conjugados presentes en la función de transferencia del filtro. A medida que aumenta el orden, se obtiene una respuesta de frecuencia más pronunciada y una atenuación más rápida en la banda de paso. Dentro de la multiplicidad de configuraciones del filtro, destacan dos principales categorías:

- Filtro Butterworth de paso bajo: Este filtro permite el paso de frecuencias inferiores a una frecuencia de corte determinada y atenúa las frecuencias por encima de esta.
- Filtro Butterworth de paso alto: Este filtro tiene un funcionamiento complementario al anterior. Es decir, deja pasar aquellas frecuencias que van por encima de la frecuencia de corte establecida y atenúa las frecuencias inferiores.

En resumen, la categoría de paso bajo tiene como principal objetivo eliminar el ruido de alta frecuencia y la variante de paso alto busca resaltar componentes de alta frecuencia en una señal, además, de eliminar el ruido de baja frecuencia.

En este caso, las señales ECG son señales de baja frecuencia muy fácilmente perturbables con interferencia electromagnética, artefactos musculares u otras interferencias ambientales. Por ello, para poder establecer un límite en el que solo se tomen la verdadera señal del corazón se aplicará un filtro Butterworth de paso bajo (ver figura 3.2).

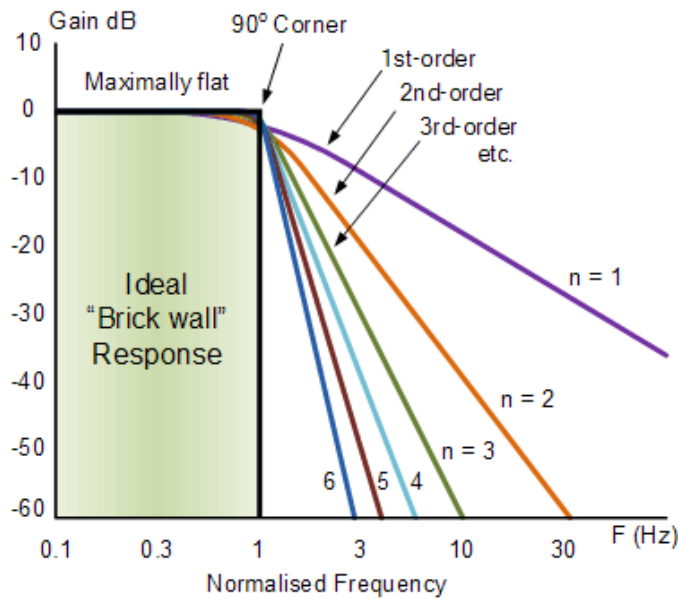


Figura 3.2. Filtro Butterworth paso bajo de orden n [17]

Fase 3: Segmentación de las señales ECG

Una vez filtradas las señales se realizó un proceso de segmentación. La segmentación del ECG es un proceso fundamental en el análisis de las señales cardíacas. Consiste en dividir la señal de ECG en sus diferentes componentes o partes principales, como las ondas P, QRS y T. Identificados los distintos ciclos presentes en el ECG y las partes que lo componen, podremos manejar todos los ciclos cardíacos como un conjunto coherente de datos.

Fase 4: Limpieza de señales

Para la limpieza de los ciclos cardíacos detectados por persona, se han superpuesto todos los ciclos cardíacos de cada persona en busca de perturbaciones presentes en ciertos instantes de tiempo durante la toma de datos. El objetivo de esta limpieza es, en la fase de entrenamiento, descartar aquellos ciclos que, para una persona, son muy diferentes del resto.

Fase 5: Selección de algoritmo de clasificación

La selección del algoritmo de clasificación para la identificación biométrica basada en el ECG es un proceso crucial para garantizar resultados precisos y confiables. En esta fase se explorará la selección de algoritmos de clasificación [20]:

- Máquina de vectores de soporte (Support Vector Machines, SVM): Son algoritmos de aprendizaje supervisado que buscan encontrar un hiperplano óptimo que pueda separar las clases en un espacio de características. En la

identificación biométrica por ECG, las características extraídas de las señales se utilizan para entrenar el modelo SVM. Este algoritmo busca maximizar la distancia entre los puntos más cercanos de diferentes clases, denominados vectores de soporte, para lograr una mejor separación entre las clases. Esto permite clasificar con precisión nuevas muestras desconocidas.

- **Análisis discriminante lineal (Linear Discriminant Analysis, LDA):** Es un algoritmo de aprendizaje supervisado que busca encontrar una combinación lineal de características que maximice la separabilidad entre las clases. En la identificación biométrica por ECG, el LDA se utiliza para proyectar las características del ECG en un espacio de menor dimensión, donde las clases son más fácilmente distinguibles. El LDA busca maximizar la relación entre la dispersión entre clases y la dispersión dentro de las clases, lo que resulta en una mejor separación de las clases en el nuevo espacio de características.
- **K-Nearest Neighbors (KNN):** El KNN es un algoritmo de aprendizaje supervisado que clasifica una muestra desconocida basándose en la similitud con sus vecinos más cercanos. En la identificación biométrica por ECG, las características del ECG se utilizan para calcular la distancia entre las muestras de entrenamiento y una muestra desconocida. Luego, el algoritmo encuentra los k vecinos más cercanos y clasifica la muestra desconocida en función de la mayoría de los votos de esos vecinos. El KNN es simple y fácil de implementar, y puede manejar tanto problemas lineales como no lineales. Además, es capaz de adaptarse rápidamente a nuevos datos sin necesidad de volver a entrenar todo el modelo.

Realizando un análisis de los algoritmos anteriores, teniendo en cuenta las condiciones del estudio y su alcance, se concluyó con la elección del KNN como algoritmo de clasificación para la identificación biométrica basada en el ECG. A continuación, se presentan algunas consideraciones que justifican la solución adoptada:

- **Similitud basada en vecinos cercanos:** En la identificación biométrica, la similitud entre diferentes ECGs puede ser medida utilizando una métrica de distancia entre señales. El KNN se basa en la idea de que las muestras con características similares tienden a agruparse en el mismo espacio utilizando la

distancia entre las características del ECG para encontrar vecinos más cercanos o clasificar una muestra desconocida.

- **Naturaleza no lineal:** El ECG es una señal altamente no lineal y su clasificación efectiva puede requerir un algoritmo que sea capaz de capturar relaciones no lineales entre las características del ECG. El KNN puede manejar datos no lineales y es adecuado para capturar las complejas relaciones presentes en las señales de ECG.
- **Adaptabilidad de nuevos datos y cambios de la señal:** Que el algoritmo KNN se base en las distancias significa que es capaz de adaptarse rápidamente a nuevos datos sin necesidad de volver a entrenar el modelo completo. Esto es beneficioso en la identificación biométrica, ya que los datos del ECG pueden variar en diferentes condiciones y momentos.
- **Número de datos de entrenamiento:** Como se trata de una prueba de conceptos con pocos datos es una buena opción elegir este algoritmo porque es capaz de realizar un entrenamiento exhaustivo con un número limitado de datos.

Fase 6: Entrenamiento y pruebas del algoritmo

El entrenamiento y prueba del algoritmo es una fase necesaria para evaluar el rendimiento del modelo. Este proceso implica dividir los datos en dos conjuntos: entrenamiento y prueba. Para la elección de la proporción de datos de cada conjunto es frecuente la elección de un 80-20%. Es decir, el 80% de los datos se utilizan para entrenar el algoritmo. Esto implica proporcionar al algoritmo las características del ECG y las etiquetas correspondientes (identificadores de las personas) para que aprenda a clasificar correctamente. Una vez se ha entrenado el modelo se utiliza el 20% de los datos restantes para evaluar el rendimiento del algoritmo.

4

Diseño del prototipo

En este Capítulo se describe el diseño del sistema. En nuestro caso, el sistema consta de un prototipo hardware, con el que se capturan los datos del ECG y se envían a un elemento externo (PC) usando Bluetooth de Baja Energía (BLE), y de un módulo software que se ejecutaría en este elemento externo, y cuya misión es procesar el ECG (filtrando, escalando, y segmentando éste en ciclos individuales), y gestionar su clasificación, implementando la identificación del usuario. Por ello, el Capítulo consta de dos grandes apartados. En el primero se describe el hardware implementado, justificando la selección de componentes y describiendo el montaje del sistema. Se proporciona algún resultado de verificación, pero las pruebas experimentales se recogen en el Capítulo 5. En el apartado 4.2 se describe la solución software. En este caso, el apartado se divide en tres secciones, que analizan (i) el software empotrado en el dispositivo (captura y envío usando BLE del ECG), (ii) el software desarrollado para el procesamiento del ECG; y (iii) el software evaluado para implementar la identificación del usuario. Los módulos desarrollados en los puntos (ii) y (iii) se han realizado en MATLAB, y son los que se ejecutarán en el dispositivo externo a nuestro prototipo hardware.

4.1. Arquitectura física

La Figura 4.1 muestra el diagrama de bloques de la arquitectura física del dispositivo. Básicamente, el sistema requiere un circuito de captura y acondicionamiento, que tenga como entradas los datos proporcionados por los electrodos ubicados sobre la persona y que genere como salida el ECG; y una unidad de procesamiento digital, que capture este ECG analógico y lo digitalice y envíe usando el protocolo BLE.

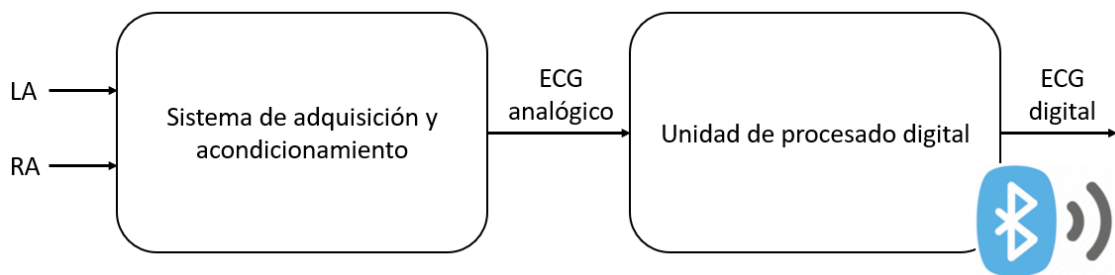


Figura 4.1. Arquitectura física del sistema. Se organiza en torno a los bloques de captura y acondicionamiento y la unidad digital de procesado.

4.1.1. Selección de componentes

Para el primero de los bloques en la Figura 4.1 existen distintas opciones en el mercado. En nuestro caso, se ha usado el front-end analógico AD8232 de Analog Devices. Perteneciente a la familia AD823X de front-ends analógicos integrados para adquisición y acondicionamiento de señales biológicas, el AD8232 es muy utilizado para proporcionar la salida analógica del ECG. Este front-end se caracteriza por ofrecer un rechazo al modo común adecuado para estas aplicaciones (80dB), y su reducido consumo y tamaño, lo que permite miniaturizar el diseño del electrocardiógrafo. Es por ello que es frecuente encontrarlo en sistemas de desarrollo de bajo coste, ofrecido por fabricantes como Sparkfun [14]. En la Figura 4.2 se muestra el diagrama de bloques funcional del AD8232. Básicamente, el dispositivo incluye un amplificador de instrumentación (IA), un amplificador operacional (A1), un amplificador para el circuito de pierna derecha (A2), y un seguidor de tensión (A3) para proporcionar una tensión de referencia. El AD8232 detecta automáticamente cuando se ha desconectado cualquiera de los electrodos de entrada, y dispone también de un circuito interno para generar de nuevo la señal de salida, de forma muy rápida, cuando el electrodo se vuelve a conectar.

El amplificador de instrumentación interno es un diseño a medida para trabajar con el ECG.

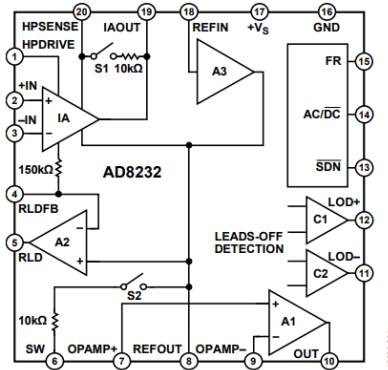


Figura 4.2. Diagrama de bloques funcionales del AD8232 [15]

En la placa empleada (aunque fabricada por Hailege, sigue el esquema propuesto por Sparkfun), el AD8232 se usa en la configuración denominada de monitorización cardiaca [15]. Esta configuración (Figura 4.3) es la adecuada para monitorizar la forma de la onda del ECG, y asume que el paciente estará en un reposo relativo durante el proceso de medida (el diseño no se centra por ello en eliminar los artefactos provocados por el movimiento).

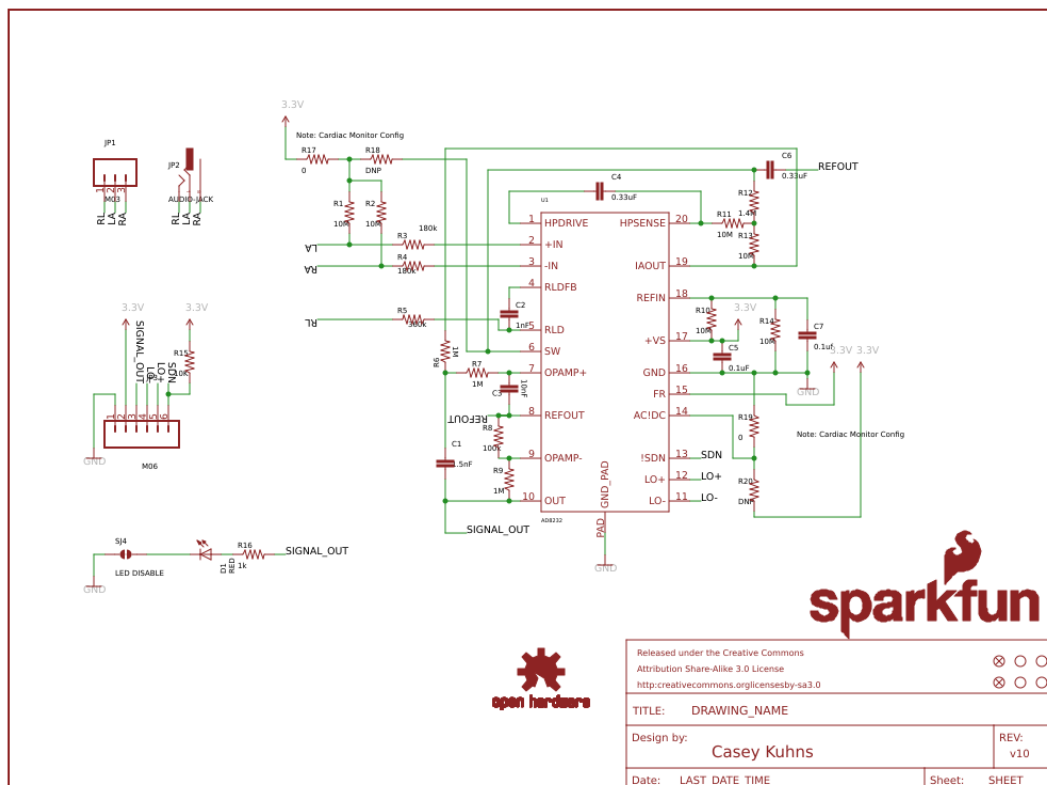


Figura 4.3. Configuración del AD8232 en modo monitorización cardiaca. El diseño de Sparkfun [14] sigue el propuesto por el fabricante en su hoja de especificaciones [15]

El AD8232 se configura con un filtro paso alto a 0.5 Hz seguido de un paso bajo a 40 Hz. El resultado es una configuración que proporciona una ganancia casi plana en el ancho de banda del ECG (0.5 - 40 Hz), con gran caída a frecuencias menores y mayores (ver Figura 4.4).

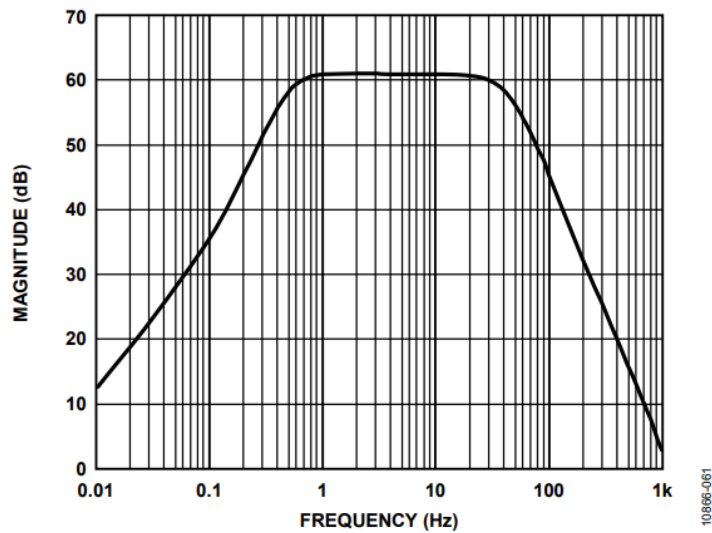


Figura 4.4. Respuesta en frecuencia para el circuito en configuración monitorización cardiaca [15]

En lo que respecta al módulo de procesado digital, el sistema usará el chip nRF51822 de Nordic. Este integrado implementa una arquitectura ARM Cortex M-0, suficiente para nuestra aplicación, e incluye en su diseño un transceptor a 2.4 GHz, que le permitirá implementar el protocolo BLE. En concreto, emplearemos un entorno de desarrollo que lo integra: la placa Tiny-BLE de Seeed Studio [21]. Aunque ya está obsoleto, este entorno resulta muy interesante y fácil de emplear, y su arquitectura ha sido replicada en otros entornos, como la Arch BLE de Seeed Studio.

La Tiny-BLE se divide en dos partes. La parte activa incluye como núcleo el nRF51822 de Nordic. Es la parte que ejecutará el código de nuestra aplicación. La otra parte se utiliza como cargador (CMSIS DAP), y permite conectar por USB la placa al ordenador, manejar la parte activa como si fuera una carpeta, y así cargar fácilmente el ejecutable al nRF51822. Como muestra la Figura 4.5, ambas partes se unen por un troquelado, que permite partir en dos fácilmente la placa. Esto permitiría incluir en un diseño real sólo la parte activa, con la consiguiente reducción en peso y tamaño.

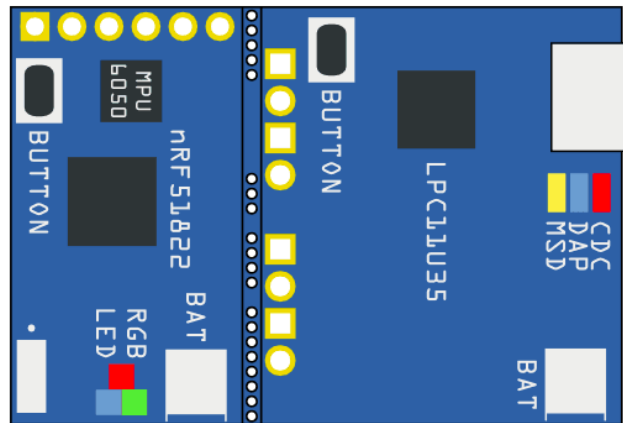


Figura 4.5. Placa tiny BLE [21]

En la parte activa de la placa, además del nRF51822, presenta un LED RGB, un pulsador, una unidad inercial que detecta el movimiento, y una antena Bluetooth. En este caso, solo se necesita el microcontrolador y la antena. El LED será también útil para mostrar visualmente que el programa está ejecutándose. También es importante notar que se dispondrá de seis pines accesibles en la parte activa. Dos de ellos proporcionan alimentación externa (3,3 V y GND). Los cuatro restantes (nombrados como p3, p4, p5 y p6) son terminales configurables como entrada o salida. En los cuatro casos, las entradas podrán ser digitales o analógicas.

4.1.2. Montaje del sistema

La Figura 4.6 muestra la conexión de los dos elementos que forman el dispositivo. En este caso, se han conectado entre los dos módulos las señales de alimentación (3,3 V y GND), y la señal de salida del AD8232 (OUTPUT) con el terminal p3 de la Tiny-BLE, configurado como de entrada analógica. Los terminales LO- y LO+ del AD8232, que informan de una desconexión en los electrodos, se conectan respectivamente a los terminales p5 y p6 de la Tiny-BLE, configurados como entradas digitales. Si se detecta un valor 1 en cualquiera de ellos es que se ha producido una desconexión.

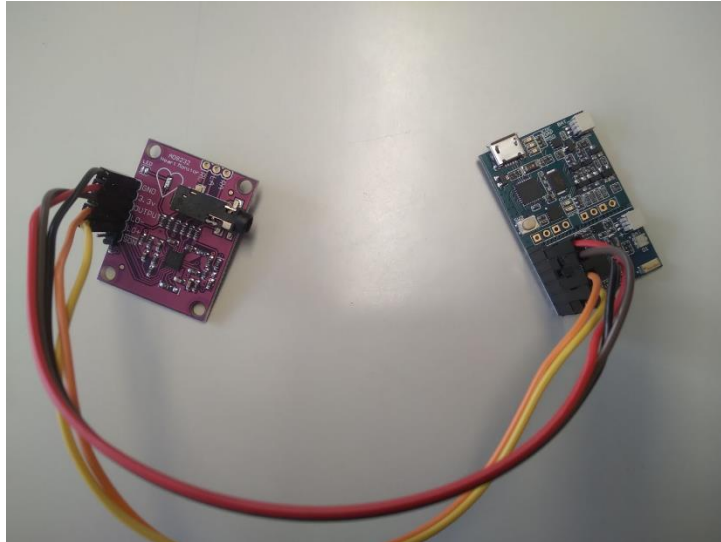


Figura 4.6. Sistema construido en torno al AD8232 de Hailege (izquierda) conectada a la Tiny-BLE de Seeed Studio (derecha) que contiene el nRF51822 de Nordic

Los electrodos se conectan directamente a la placa de Hailege. En principio se pueden conectar tres electrodos, lo que permite montar el triángulo de Einthoven. La Figura 4.7 muestra la conexión de los electrodos al sistema. Se montan electrodos húmedos.

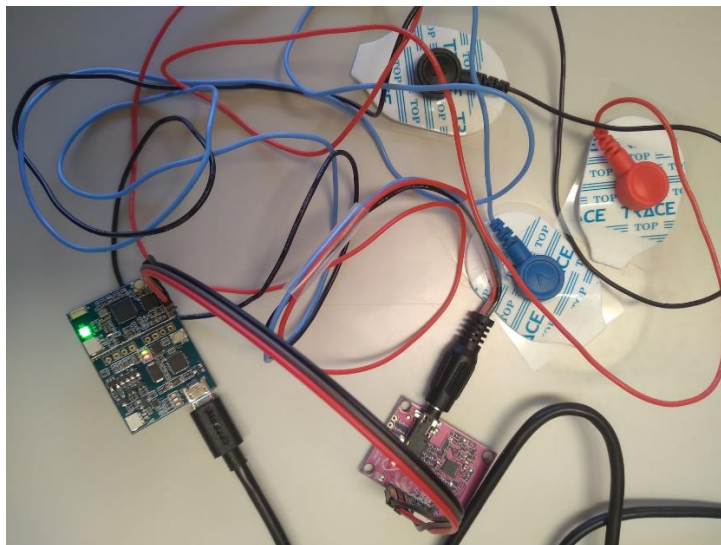


Figura 4.7. Conexión de electrodos al sistema

Finalmente, la Figura 4.8 muestra el aspecto de un ECG capturado usando este sistema. Se han conectado solo los electrodos de RA y LA, en la yema de los dedos índices de las manos derecha e izquierda, respectivamente. La señal recibida se digitaliza entre 0 y 1024 (convertidor analógico/digital de 10 bits). En la figura, el dato de valor es un flotante entre 0,0 y 1,024. Como se muestra en la Figura 4.7, para esta prueba no se usan las señales LO- y LO+, pues nos avisarían de una desconexión al no estar el

electrodo de pierna derecha (RL). El ECG es ruidoso, pero se aprecian las ondas y segmentos presentes en su forma típica (Figura 2.6).

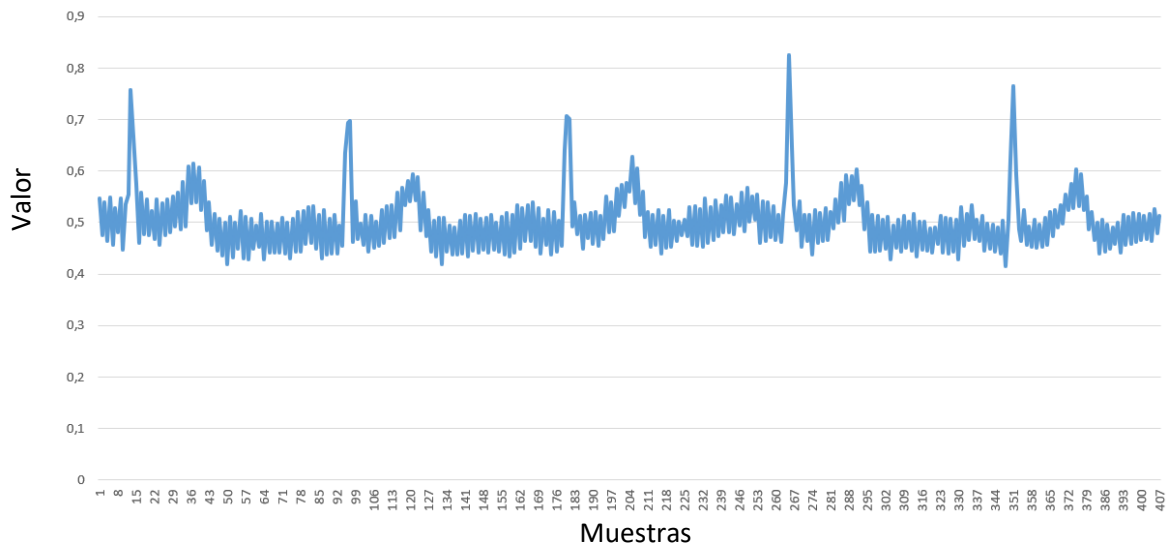


Figura 4.8. ECG capturado usando el sistema propuesto

4.2. Arquitectura software

4.2.1. Captura de datos y envío usando Bluetooth de Baja Energía

Keil Studio Cloud

Para programar el nRF51822 se carga en el mismo un fichero .hex generado externamente. En tal caso, para generar ese fichero se usará el compilador Keil Studio Cloud de ARM [18]. Keil Studio Cloud es un IDE (Integrated Development Environment) basado en navegador web. Es de uso gratuito, viene con un potente editor y es compatible con el amplio ecosistema de microcontroladores ARM CMSIS, así como con Mbed OS, el antiguo sistema operativo en línea recomendado por ARM para programar sus diseños. De hecho, Keil Studio Cloud se basa en la misma tecnología desarrollada para Mbed Studio y el compilador en línea Mbed, con el objetivo de proporcionar un entorno de creación rápida de prototipos fácil de usar, que se puede utilizar para desarrollo embebido en general.

La Figura 4.9 muestra el aspecto de la página de entrada al IDE Keil Studio Cloud. Como evolución de Mbed Studio, usa el mismo entorno de trabajo que éste, adaptado con soporte adicional para ampliar la gama de middlewares y permitir trabajar con el mayor número posible de dispositivos embebidos basados en ARM. En la Figura 4.10 se muestra el aspecto del entorno de trabajo. Se dispone de una ventana central, donde

se muestra el código de los ficheros abiertos (en nuestro caso, el main.cpp). Debajo de esta ventana, se encuentra el interfaz donde el sistema nos muestra problemas con el proyecto, resultados de la generación de un fichero de salida, o las librerías presentes en el proyecto. En las ventanas a la izquierda de la ventana principal, el IDE nos informa sobre el proyecto activo, el hardware (target) para el que se está generando el fichero de salida, y el hardware que tenemos conectado al PC en ese momento (en nuestro caso, se está generando código para la Seeed Arch BLE, y no hay hardware conectado al PC). Bajo estas tres pequeñas ventanas, se tiene una ventana con el directorio de proyectos. Sólo uno de ellos podrá ser seleccionado como activo (esto es importante porque las opciones de, por ejemplo, Construir proyecto, se ejecutan sobre el proyecto activo, sin importar qué fichero está abierto en la ventana principal).

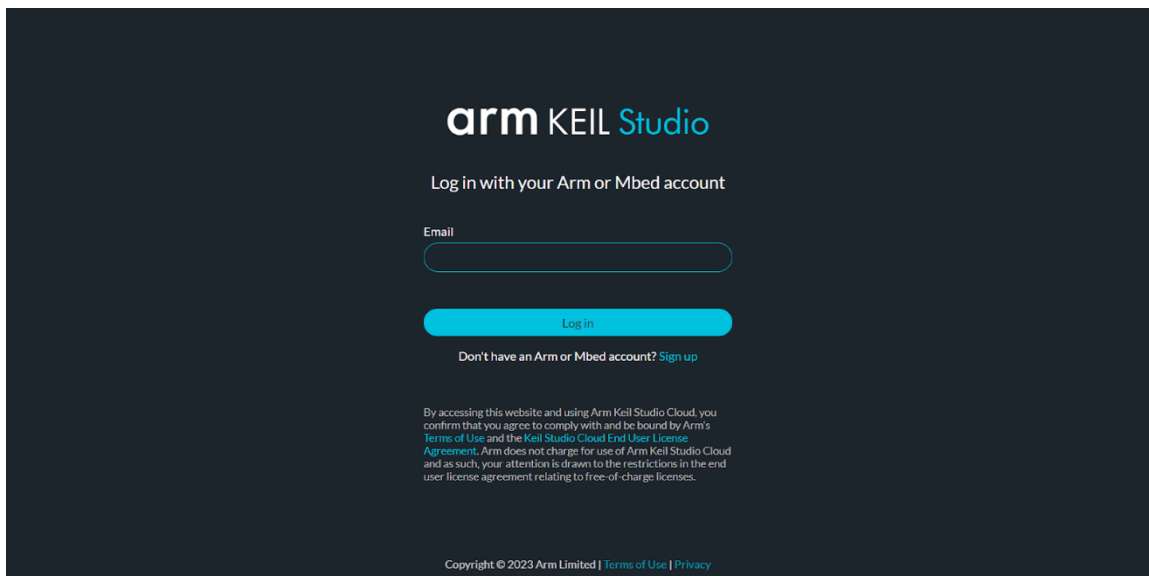


Figura 4.9. Página de entrada al IDE Keil Studio Cloud de ARM

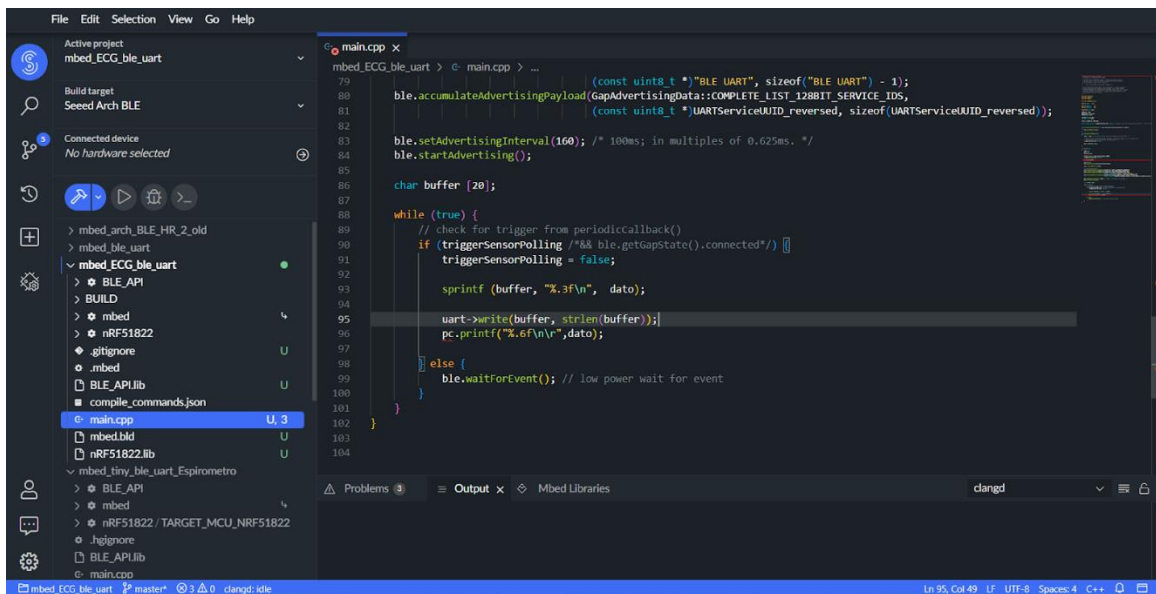


Figura 4.10. Aspecto de las ventanas que conforman el IDE (ver texto)

Sobre esta ventana con el directorio de proyectos se tienen botones rápidos para construir el proyecto, construir y cargar el proyecto en el hardware conectado al PC, ejecutar en modo debug, o abrir una consola, en la que se podrán visualizar los datos enviados por el terminal serie que, en muchas plataformas (como es la Tiny-BLE), se puede abrir entre el PC y el hardware conectado. En el ejemplo abierto en la Figura 4.10, es en este terminal donde se muestra el mensaje que enviamos con la instrucción `pc.printf()`.

Finalmente, el IDE ofrece unas líneas de menú tanto superior, principalmente ayuda en la edición de ficheros o para abrir y gestionar proyectos, como lateral, más relacionada con la gestión de más alto nivel (hardware conectado, búsqueda en proyectos, historial, o la gestión del perfil de usuario).

Software desarrollado

El software desarrollado para ejecutarlo en la placa Tiny-BLE se organiza en un único fichero `main.cpp` y utiliza tres librerías externas. La librería `BLE_API` es una abstracción de alto nivel que permite emplear Bluetooth Low Energy en distintas plataformas [24], mientras que la librería `nRF51822` carga la pila y controladores de Nordic para la API BLE en Mbed [25]. También es necesario cargar la librería `mbed`, que ofrece toda la funcionalidad del sistema operativo Mbed. Concretamente, se ha cargado la versión Mbed OS 2, pues era con la que se tenía experiencia previa al usarla en la asignatura de Electromedicina en el grado de Ingeniería de la Salud.

El programa desarrollado tiene dos características principales: gestiona la captura de datos del AD8232 usando una interrupción periódica (Ticker), y envía los datos usando el servicio UART de BLE. Para implementar la captura de datos el sistema implementa un Ticker.

```
ticker.attach_us(&periodicCallback,10000);
```

en el que se usan como parámetros la función a ejecutar y el tiempo entre llamadas a la interrupción. En este caso, este tiempo se define en microsegundos. La función llamada por la interrupción es muy simple

```
void periodicCallback(void)
{
    led1 = !led1;

    triggerSensorPolling = true;

    dato = Entrada.read();
}
```

hace parpadear un LED y pone a true una variable de contexto (triggerSensorPolling), y se lee el dato de la entrada analógica (AnalogIn Entrada(p3)). En el bucle que se ejecuta siempre en el main.cpp, poner a true esta variable nos hace entrar en la parte del código que gestiona el envío del dato recién leído usando BLE (también por el terminal serie al PC).

```
while (true) {
    // check for trigger from periodicCallback()
    if (triggerSensorPolling /*&& ble.getGapState().connected*/) {
        triggerSensorPolling = false;

        sprintf (buffer, "%.3f\n", dato);

        uart->write(buffer, strlen(buffer));
        pc.printf("%.6f\n\n",dato);

    } else {
        ble.waitForEvent(); // low power wait for event
    }
}
```

Al usarse el servicio BLE_UART de BLE, el dato a enviar debe convertirse en una cadena de caracteres. Esta cadena se envía usando `uart->write(buffer, strlen(buffer));` Finalmente, el código incluye la inicialización del servicio BLE, y su anuncio, usando para ello funciones de la librería BLE_API.

```

uart = new UARTService(ble);

/* setup advertising */
ble.accumulateAdvertisingPayload(GapAdvertisingData::BREDR_NOT_SUPPORTED);
ble.setAdvertisingType(GapAdvertisingParams::ADV_CONNECTABLE_UNDIRECTED);
ble.accumulateAdvertisingPayload(GapAdvertisingData::SHORTENED_LOCAL_NAME,
                                (const uint8_t *)"BLE UART", sizeof("BLE UART") - 1);
ble.accumulateAdvertisingPayload(
    GapAdvertisingData::COMPLETE_LIST_128BIT_SERVICE_IDS,
    (const uint8_t *)UARTServiceUUID_reversed,
    sizeof(UARTServiceUUID_reversed));

ble.setAdvertisingInterval(160); /* 100ms; in multiples of 0.625ms. */
ble.startAdvertising();

```

Para evaluar la correcta recepción de los datos en el smartphone se ha instalado en éste la aplicación nRF Connect, ofrecida por Nordic. En la Figura 4.11 se muestra una captura del resultado de la búsqueda de dispositivos BLE (nuestro dispositivo tiene el nombre de BLE_UART, definido en la función `ble.accumulateAdvertisingPayload`). Si se realiza la conexión al dispositivo, se observa que ofrece tres servicios (Figura 4.11 (derecha)). De ellos es de especial interés el Nordic UART Service. Si se pulsa sobre este servicio se abren las dos características que ofrece este servicio (Figura 4.12 (izquierda)). El primero es una característica de escritura, mientras que la segunda (TX Characteristic) es de lectura. De hecho, es de notificación: si se clic en ella se irá actualizando el valor recibido (si fuera de lectura, capturaríamos un único dato). Al pinchar en esta característica (Figura 4.12 (derecha)) aparece el campo Valor (Value), en el que se muestra el valor enviado por el dispositivo (0.551 en este caso). El descriptor de este servicio, que se muestra en las últimas filas, indica que se han habilitado las notificaciones.

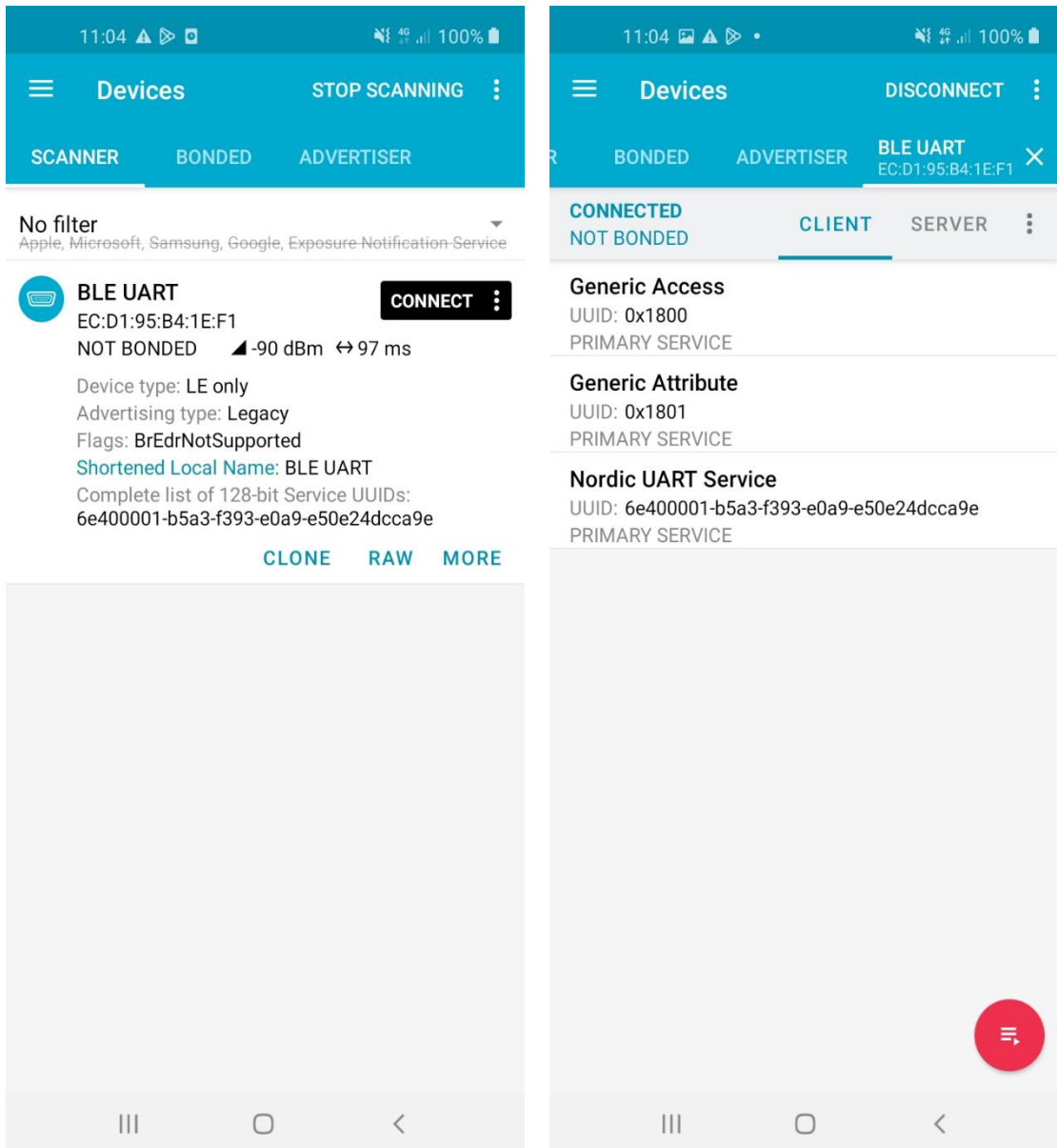


Figura 4.11. Monitorización de la conexión BLE-UART usando nRF-Connect (1)

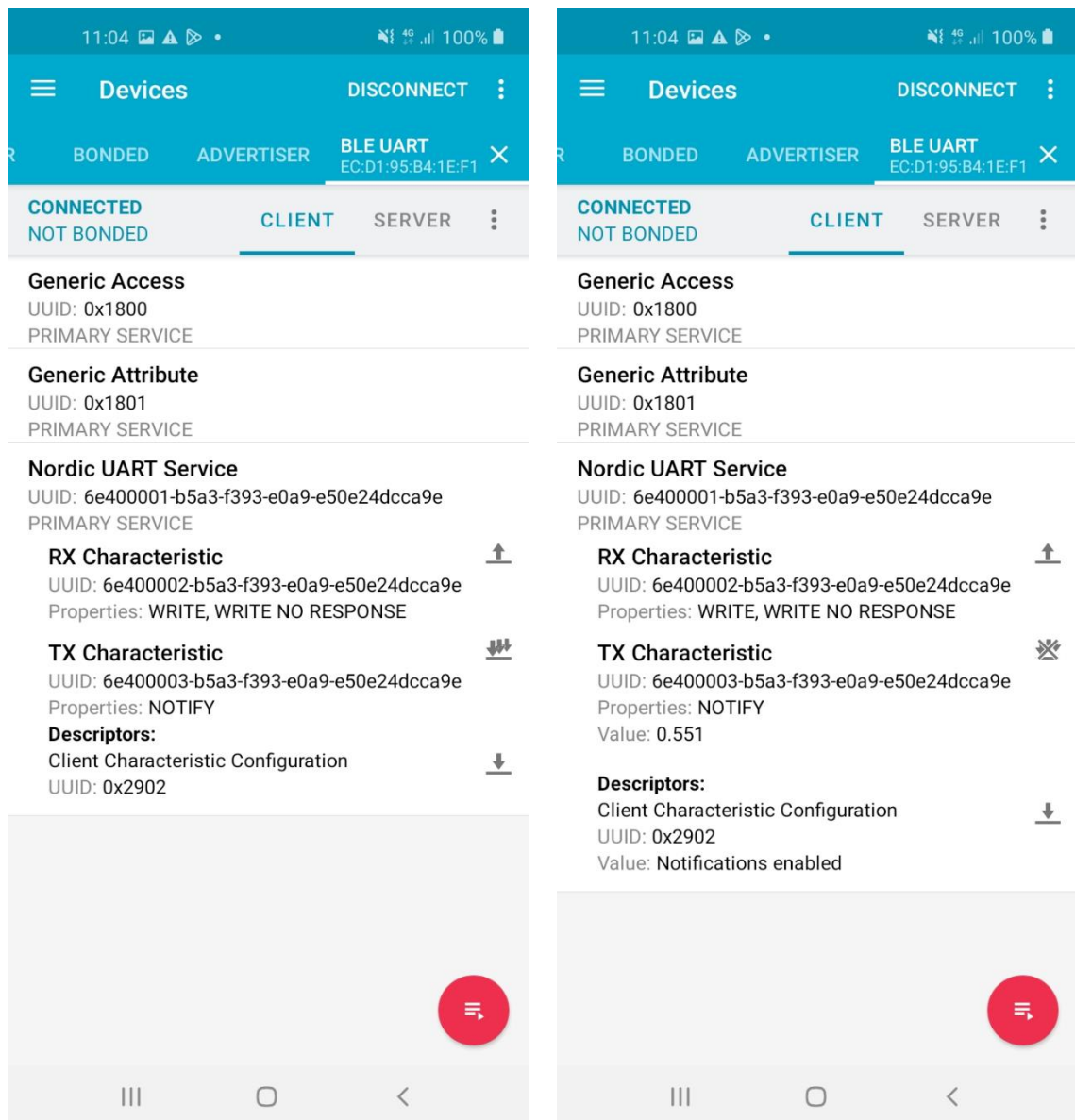


Figura 4.12. Monitorización de la conexión BLE-UART usando nRF-Connect (2)

4.2.2. Preprocesado de datos

Como se ha detallado en el Capítulo 3, el preprocesado de los datos es una parte crucial para este tipo de estudios. Ya en la parte del sistema que corre fuera del electrocardiógrafo, esta etapa comienza con la creación de un filtro Butterworth adecuado para las señales capturadas.

Para implementar este filtro se ha hecho uso del paquete de procesamiento de señales digitales que se encuentra en la Toolbox de Procesamiento de Señales de MATLAB (Signal Processing Toolbox) [16]. Concretamente se han empleado dos funciones, denominadas *butter* y *filtfilt*, que permiten generar el filtro y usarlo para aplicar el mismo filtro Butterworth a múltiples señales de ECG. La función *butter* se ha usado para dividir la señal en segmentos superpuestos, lo que facilita el procesamiento

y el cálculo del filtro. Luego, la función *filtfilt* se utilizó para aplicar el filtro de forma bidireccional, eliminando cualquier retraso de fase y garantizar una representación precisa de la señal filtrada. Previo a la aplicación de estas funciones hay que seleccionar adecuadamente los valores de la frecuencia de corte y de muestreo (30 Hz y 100 Hz respectivamente). A continuación, se exponen las razones de esta selección:

- Eliminación del ruido de alta frecuencia: Aunque el AD8232 está configurado para eliminar frecuencias fuera del margen entre 0.5 y 40 Hz, se aprecia en las capturas la presencia de ruido de alta frecuencia. La frecuencia de corte de 30 Hz permite atenuar o eliminar eficazmente el ruido de alta frecuencia no deseado que pueda estar presente en la señal de ECG. Esto incluye el ruido muscular y el ruido eléctrico, los cuales suelen tener componentes de frecuencia más alta. Al aplicar un filtro paso bajo con una frecuencia de corte de 30 Hz, se puede mejorar la calidad de la señal al eliminar estos componentes de alta frecuencia no deseados.
- Conservación de información relevante: La frecuencia de corte de 30 Hz se elige de manera que conserve la información relevante en la señal de ECG. La mayoría de los eventos cardíacos y características importantes del ECG, como los complejos QRS y las ondas P, se encuentran en frecuencias más bajas. Al utilizar una frecuencia de corte de 30 Hz, se retiene la información necesaria para el análisis y la interpretación precisa de estas características.
- Cumplimiento con el teorema de muestreo: La frecuencia de muestreo de 100 Hz es adecuada para capturar las componentes de frecuencia de la señal de ECG de manera efectiva, de acuerdo con el teorema de muestreo de Nyquist-Shannon. Según este teorema, para evitar la pérdida de información, la frecuencia de muestreo debe ser al menos el doble de la frecuencia máxima presente en la señal. En el caso del ECG, la frecuencia máxima típica de interés es de aproximadamente 40 Hz, por lo que una frecuencia de muestreo de 100 Hz cumple con este criterio.

Al combinar una frecuencia de corte de 30 Hz y una frecuencia de muestreo de 100 Hz, se puede obtener una señal de ECG filtrada con un buen equilibrio entre la eliminación del ruido de alta frecuencia y la conservación de la información relevante. Sin embargo, es importante tener en cuenta que la elección de estos parámetros puede

variar según las características específicas de la señal y los requisitos del análisis. En la Figura 4.13 se observa la respuesta de frecuencia del filtro.

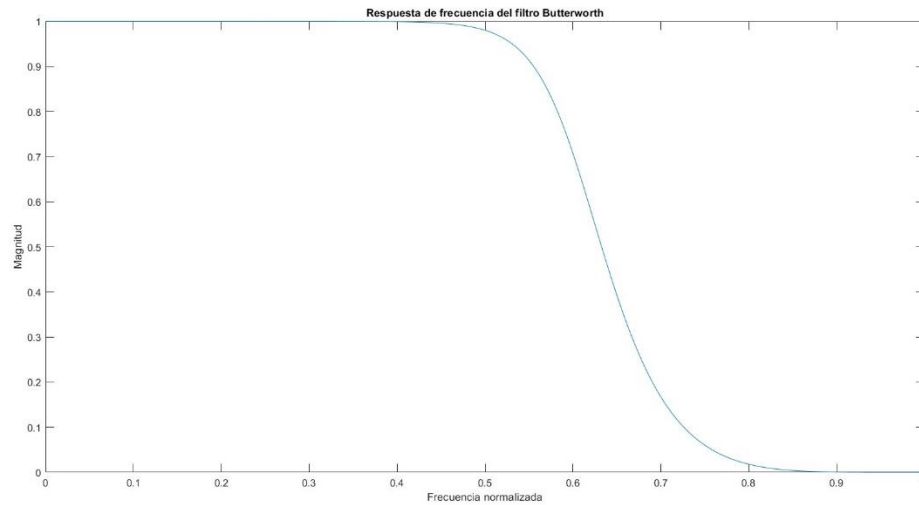


Figura 4.13. Filtro Butterworth paso bajo, $n = 4$ y $F_c = 30$ diseñado

En la Figura 4.14 se muestran la señal ECG capturada a una de las personas que participaron en nuestro estudio y el resultado de filtrar esta señal usando el filtro diseñado.

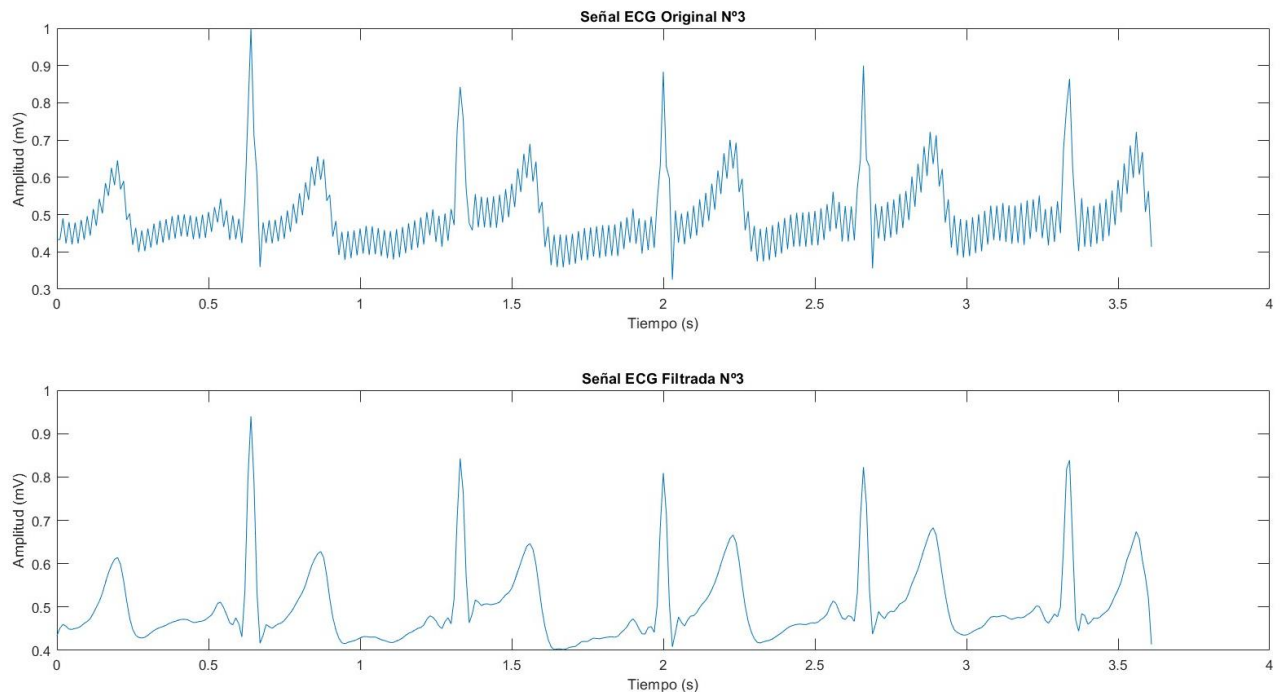


Figura 4.14. Señal ECG capturada (imagen superior) y filtrada (imagen inferior)

Una vez filtrada la señal del ECG, procederemos a segmentarla. El objetivo es dividir el ECG en ciclos, e identificar en cada uno de estos las distintas partes que lo forman. Para esta etapa MATLAB cuenta con el paquete capaz de detectar los complejos

QRS de cualquier señal ECG denominado “*Complete Pan Tompkins Implementation ECG QRS detector*”. De este paquete se extrae una de las principales funciones de este estudio, que es la función *pan_tompkin*, la cual se ha usado para detectar los complejos QRS ya que ha demostrado tener una detección precisa de los mismos. Una vez identificados se calcularon los intervalos de tiempo correspondientes a cada parte del ciclo cardiaco (P, Q, R, S, T) en función de la duración total del ciclo cardiaco. Estos intervalos se definen de manera proporcional y se ajustan según la frecuencia de muestreo de la señal. Esto permite una segmentación adecuada de las diferentes partes del ciclo cardíaco (ver Figura 4.15).

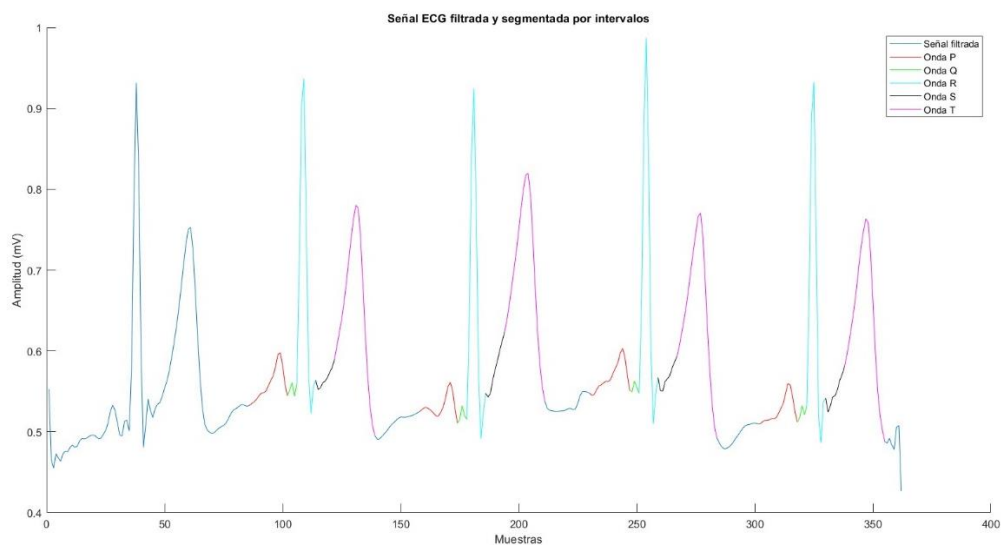


Figura 4.15. Señal ECG representada por segmentos

Una vez que tenemos los ciclos que componen cada ECG capturado a una determinada persona, agruparemos estos ciclos cardiacos en el conjunto de datos que caracterizan a esta persona. Una parte de este conjunto de ciclos se usará para entrenar el algoritmo de identificación y generar el modelo de persona. Por ello, es interesante eliminar aquellos ciclos que sean muy distintos del resto. Se ha utilizado un método de superposición de gráficas convencional, es decir, un *hold on* con todos los ciclos de las señales ECG almacenadas en la fase de segmentación mediante un bucle *for*. La eliminación será llevada a cabo manualmente. Evidentemente, en este caso, es posible hacerlo así debido a que el número de señales es reducido. El criterio de eliminación de ciclos se ajustó a las especificaciones de diseño estipuladas en el apartado 3.1 teniendo en cuenta los niveles de ruido permitidos (ver Figura 4.16).

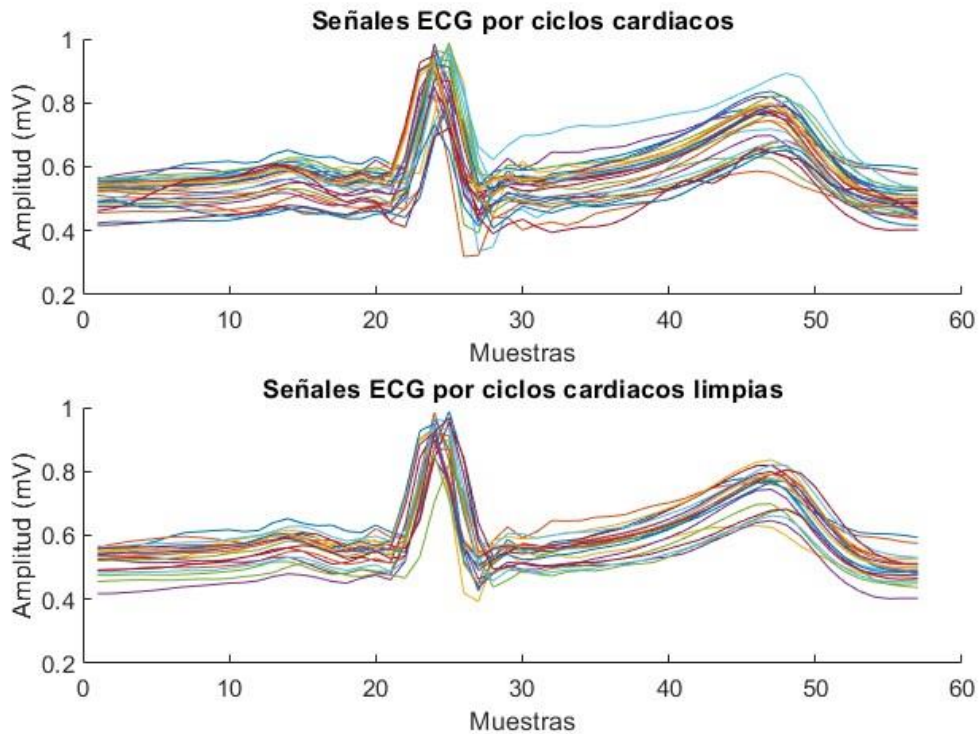


Figura 4.16. Señales ECG antes y después de seleccionar las muestras dentro de los parámetros establecidos

Tras este último paso se puede concluir que los datos están preparados para entrenar el clasificador.

4.2.3. Identificación de usuarios

Una vez preprocesados y organizados los datos como se describe en el apartado 4.2.2, se puso en marcha la implementación del algoritmo de clasificación KNN. Este algoritmo, como se expone en el Capítulo 3, es la mejor opción en nuestro caso para la búsqueda de diferencias entre señales ECGs. Para implementar un método de clasificación KNN, se han seguido los siguientes pasos:

1. **Cargar los datos:** Para iniciar el proceso, se cuentan con 21 señales limpias (ciclos del ECG) correspondientes a cada uno de los participantes en la investigación (4 personas). El primer paso consiste en preparar la información de manera adecuada para el algoritmo. En este caso, se ha creado una matriz de datos con dimensiones de 84x57. Esta matriz combina las 21 señales de cada persona (21 x 4), donde cada una de ellas está compuesta por 57 valores.
2. **División en conjuntos de entrenamiento y prueba:** Se divide el conjunto de datos normalizados en un conjunto de entrenamiento y un conjunto de prueba. El 80% de los datos se utilizarán para entrenar el modelo y el 20% restante se

utilizará para evaluar el rendimiento del modelo. Esta elección se realiza de manera aleatoria gracias a la función *randperm*.

3. **Cálculo de distancias:** Se define una función anónima (*distancia_euclidiana*) que calcula la distancia Euclidiana entre dos señales.
4. **Encuentra los vecinos más cercanos:** Se establece el número de vecinos más cercanos (K) que se tomarán en cuenta. Luego, para cada señal en el conjunto de prueba, se calculan las distancias entre esa señal y todas las señales en el conjunto de entrenamiento. Las distancias se ordenan de menor a mayor y se seleccionan los K vecinos más cercanos. Como se trata de unos conjuntos generados aleatoriamente dentro del conjunto de datos se ha elaborado un bucle de selección del mejor valor para K. Esta selección prueba con valores para K del 1 al 10.
5. **Realiza la clasificación:** Se asigna a la señal del conjunto de prueba la etiqueta de clase más frecuente entre sus vecinos más cercanos. Esto se logra mediante la función *mode*, que devuelve el valor más común en un conjunto de datos.
6. **Evalúa el rendimiento del algoritmo:** Se calcula la precisión y exactitud del modelo comparando las etiquetas predichas con las etiquetas reales del conjunto de prueba. La precisión se calcula dividiendo el número de verdaderos positivos entre la suma de verdaderos positivos y falsos positivos.

5

Resultados obtenidos

En este Capítulo se valida la funcionalidad del sistema propuesto, extendiendo las pruebas que se han presentado en el Capítulo 4 y proporcionando datos sobre el proceso de identificación.

5.1. Captura del ECG

En los Capítulos 3 y 4, al describir la captura de señales ECG, se ha hablado de dos sistemas hardware. Como se comentó en el Capítulo 3, se ha trabajado simultáneamente en ambos con el fin de optimizar el tiempo dedicado a este proyecto. El primero de los sistemas (Figura 3.1) empleaba la placa de Hailege con el AD8232 y un Arduino DUE. El segundo (Figura 4.6) sustituye el Arduino por una Tiny-BLE. Aparte de las diferencias en el código implementado debido a usar distintos entornos de programación, la principal diferencia entre ambos diseños es que el segundo implementa el envío BLE, una funcionalidad muy práctica, pero a la vez algo más compleja a nivel de configuración. Para poder trabajar el diseño software en MATLAB [23] mientras cerrábamos el trabajo con la plataforma Tiny-BLE, las señales ECG capturadas por persona, y que emplearemos a lo largo de este Capítulo, fueron capturadas usando el primero de los diseños. Una vez completado el segundo prototipo, no se han visto diferencias relevantes en la forma que adoptaba el ECG, más

condicionado por el uso del AD8232 (que sí era común a ambos diseños). Además, los convertidores analógico-digital de ambas plataformas son de 10 bits.

Por tanto, el prototipo usado en las pruebas descritas en este Capítulo es el formado por el Arduino DUE y la placa con el AD8232 fabricada por Hailege. El Arduino fue programado para la captura de ECGs mediante el entorno Arduino IDE versión 2.1.0 [22]. La implementación de este código es sencilla y breve gracias a la configuración del AD8232 [19]. Por tanto, con un simple bucle, que determinara la frecuencia de toma de datos y la ubicación de los pines, se obtienen buenos resultados. Los resultados obtenidos fueron transcritos a un fichero excel y, posteriormente, transformados en ficheros “.mat” para que pudieran ser empleados en la siguiente fase, implementada en MATLAB.

5.2. Preprocesado del ECG en MATLAB

Como se ha descrito en el Capítulo 4, la primera de las etapas implementadas en MATLAB tenía por finalidad el filtrado de los datos capturados. En las Figuras 5.1 y 5.2 se muestran, respectivamente, las 10 señales ECG capturadas a una de las personas que participaron en el estudio, y las correspondientes versiones filtradas. Se aprecia como el filtro permite eliminar el ruido de alta frecuencia, que era mayor o menor en función de las condiciones externas de captura del ECG.

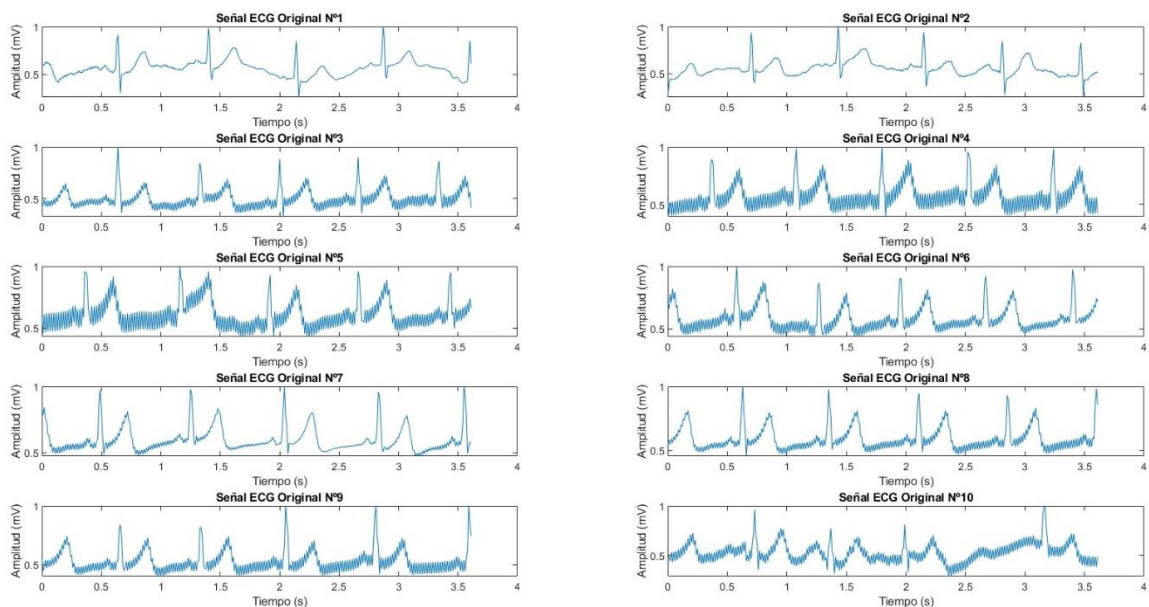


Figura 5.1. Señales ECG originales de una de las personas estudiadas

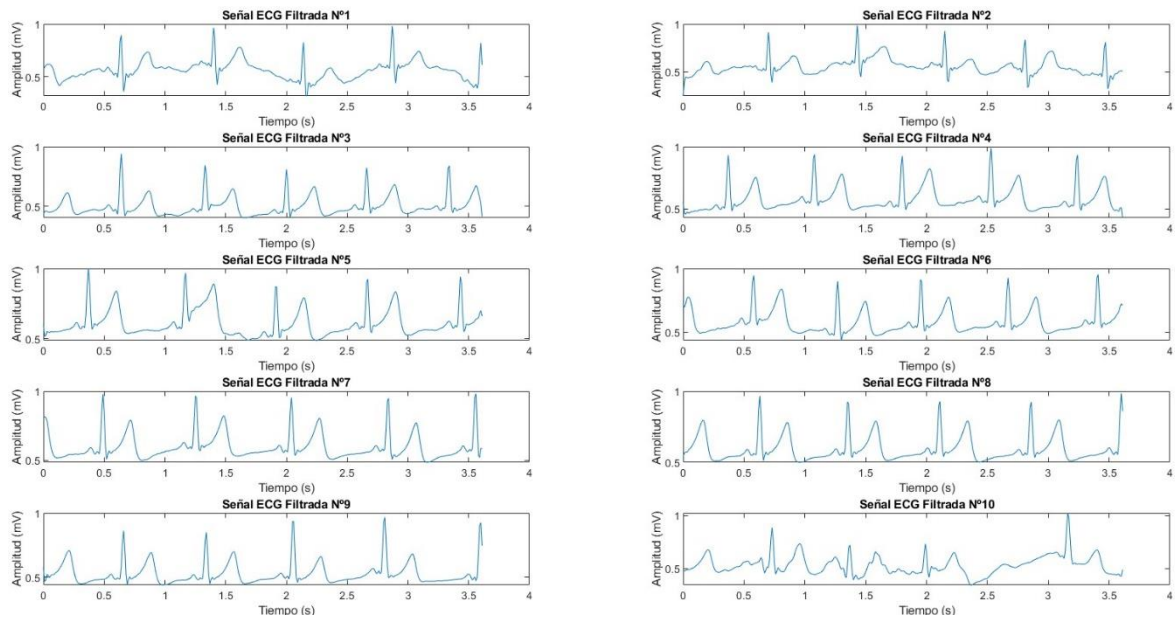


Figura 5.2. Señales ECG filtradas con el filtro Butterworth desarrollado

El siguiente paso era segmentar las señales ECG filtradas. Queda fuera del alcance de este trabajo el describir con detalle el algoritmo de **Pan-Tompkin**, empleado para llevar a cabo la segmentación. Brevemente, para detectar un complejo QRS, este algoritmo busca los picos locales de la señal ECG. Por pico se entiende un punto en el que la señal cambia de sentido ascendente a descendente. Además, se aplica la regla de que, después de un pico, no puede haber otro en los siguientes 200 ms. Finalmente, todo pico no tiene por qué ser un complejo QRS. Para reducir la posibilidad de seleccionar un pico debido al ruido como QRS, la amplitud del pico se compara con un umbral, que tiene en cuenta la información disponible (variará con el tiempo) sobre los complejos QRS ya detectados y el nivel de ruido. La Figura 5.3 muestra el proceso de detección de complejos QRS empleando el código disponible en MATLAB [26]. En las dos figuras superiores aparecen tres señales dibujadas sobre el ECG. La señal negra representa el nivel de ruido detectado, la señal roja el nivel de señal (QRS ya detectados), y la señal verde el umbral adaptativo. Se aprecia como la señal roja aumenta, y se aproxima a la altura de los QRS ya detectados. El valor de señal de ruido permanece prácticamente constante, y también el umbral. En la figura se aprecia que el algoritmo detecta los complejos QRS correctamente.

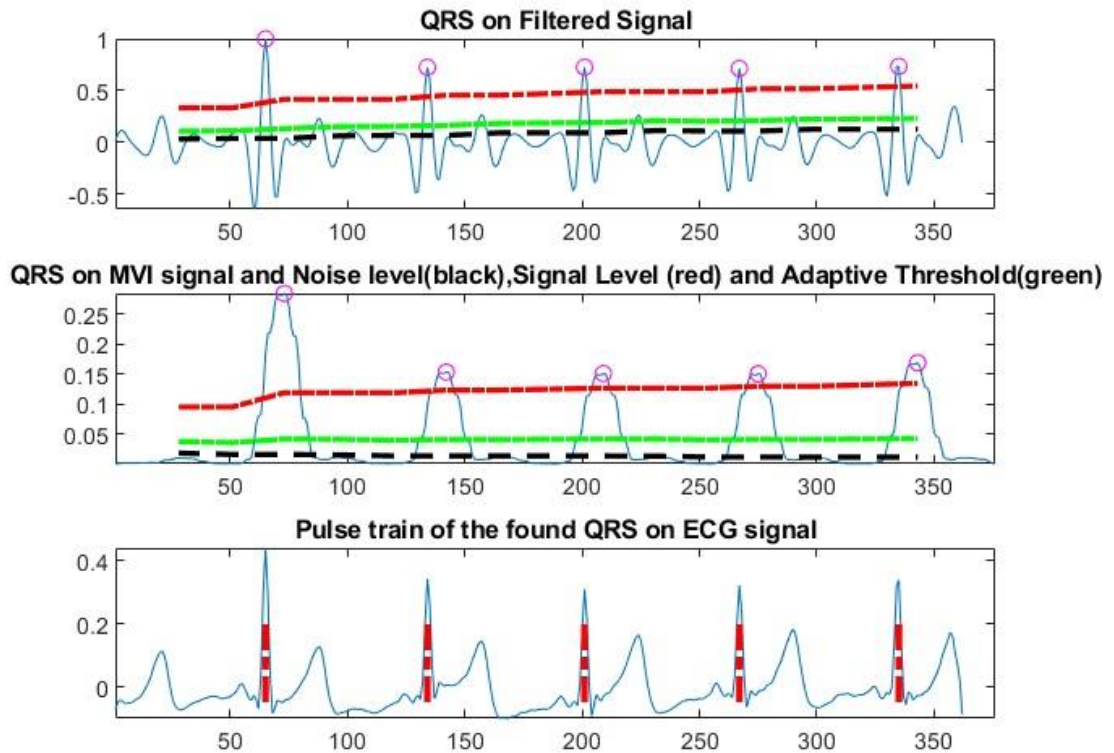


Figura 5.3. Detección de complejos QRS proporcionada por la función pan_tompkin [elaboración propia].

Internamente, el algoritmo aplica un preprocesado al ECG. Los distintos pasos de este preprocesado se muestran en la Figura 5.4. Así, la señal de entrada será (1) filtrada con un paso-banda, entre 5 y 15 Hz; (2) filtrada con un derivador para obtener información sobre la pendiente (más fuerte en torno al QRS); (3) elevada al cuadrado (esto dará más fuerza al QRS frente a las ondas T); y (4) integrada usando una ventana móvil. El algoritmo trabajará sobre esta señal integrada. En la Figura 5.3, por ejemplo, los valores de señal, ruido y umbral se muestran (imagen intermedia) sobre esta señal integrada. Esta figura se repite, en la posición inferior-derecha, en la Figura 5.4.

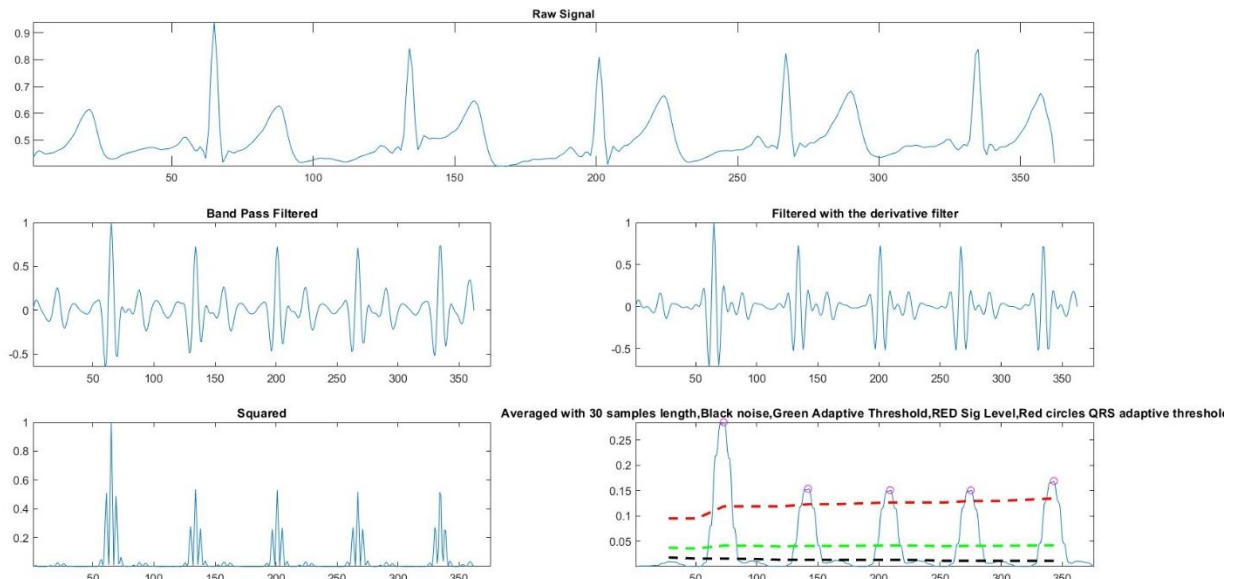


Figura 5.4. Preprocesado de la señal del ECG llevado a cabo por la función pan_tompkin [elaboración propia]

Como se describe en el Capítulo 4, una vez detectado el complejo QRS, podemos determinar el ciclo cardíaco e, incluso, identificar el resto de partes que lo componen.

5.3. Pruebas de identificación

Una vez que cada persona es caracterizada por un conjunto de ciclos cardíacos, el algoritmo KNN descrito en el Capítulo 4 fue aplicado a este conjunto de datos. Se diferencia en esta sección entre dos pruebas. En la primera se divide el conjunto de datos en entrenamiento y clasificación de forma aleatoria y se prueban distintos valores de K. En la segunda se usa siempre el mismo conjunto de datos para el entrenamiento.

Método aleatorio

Tras ejecutar la clasificación repetidas veces se observa que no se obtienen siempre los mismos resultados si no que estos van variando en cada ciclo. Esto se debe a que la división de datos se realiza aleatoriamente. Las Figuras 5.5-7 muestran la matriz de confusión en 3 ejecuciones diferentes (con K=5 (Figura 5.5) y K= 3 (Figuras 5.6 y 5.7)).

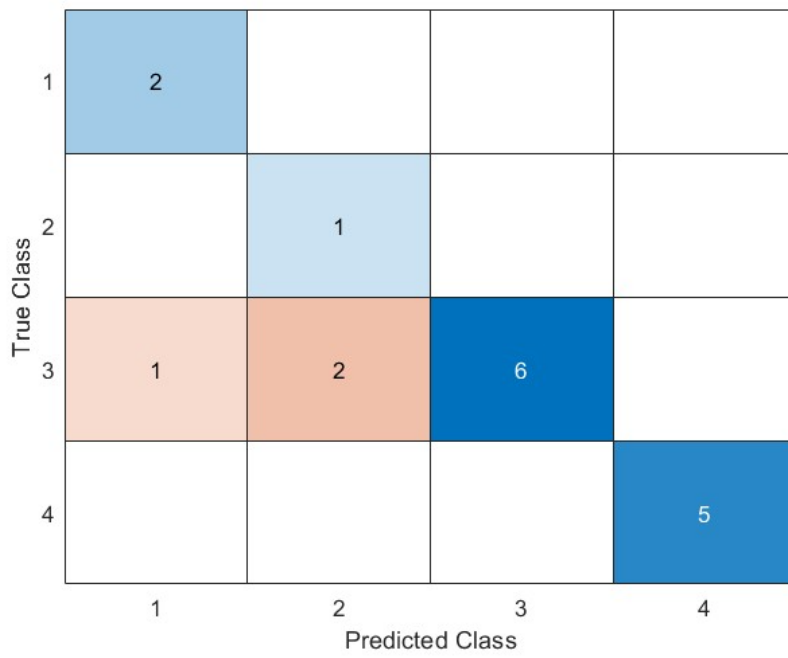


Figura 5.5. Matriz de confusión cuando K=5, precisión= 82,35%

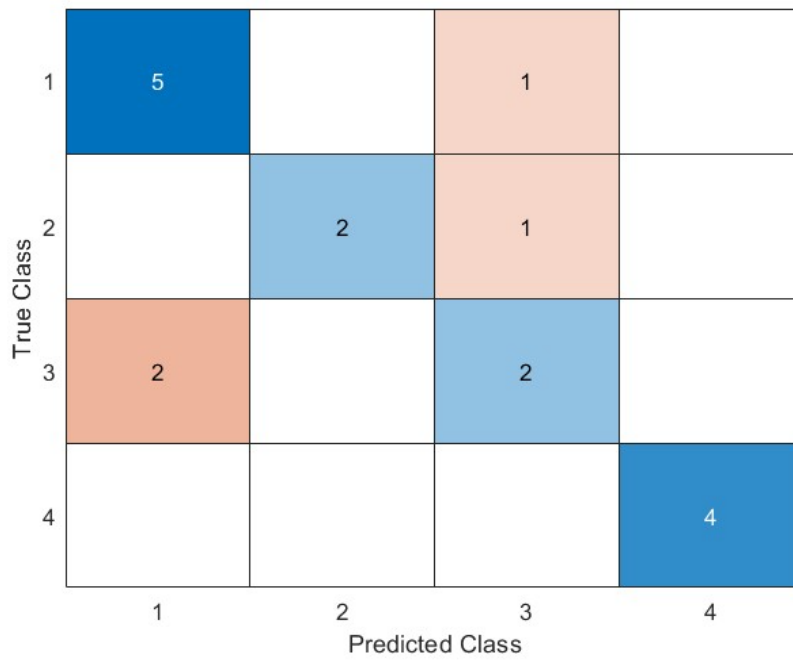


Figura 5.6. Matriz de confusión cuando K=3, precisión= 76,47%

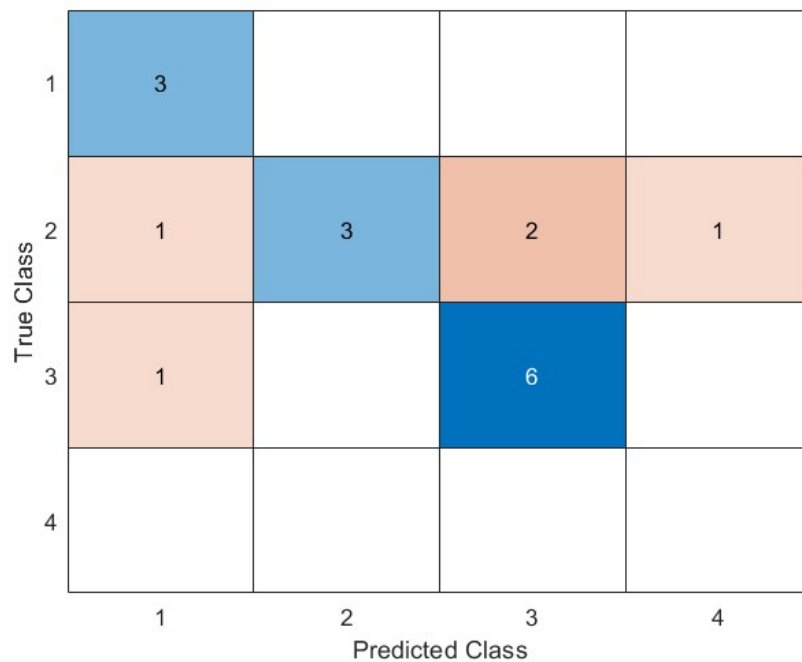


Figura 5.7. Matriz de confusión cuando K=3, precisión = 70,58%

Con este tipo de selección de datos aleatoria se obtiene una precisión que oscila entre el 70-85%, pero dependiendo de cómo se distribuyan, puede haber casos en los que el rendimiento del modelo parezca alto debido a una distribución afortunada de los datos, pero en realidad no sea generalizable. Ante estos resultados, se ha planteado una modificación del algoritmo en la que se seleccionen los datos de manera dirigida (no aleatoria).

Método no aleatorio

Para obtener un conjunto de datos de entrenamiento y prueba fijos se tomarán de 21 en 21 los datos correspondientes a cada persona. Y, a cada uno de los conjuntos se le aplicará una proporción del 80-20 tomando de esta manera siempre las mismas muestras de entrenamiento y prueba. Al seleccionar manualmente los datos, se tiene un mayor control sobre la distribución y características del conjunto de entrenamiento y prueba. Esto puede ser útil cuando se desea evaluar el modelo en escenarios específicos o garantizar una representación equilibrada de las clases, como en este caso. A continuación, la Figura 5.8 muestra el resultado usando esta nueva selección de datos.

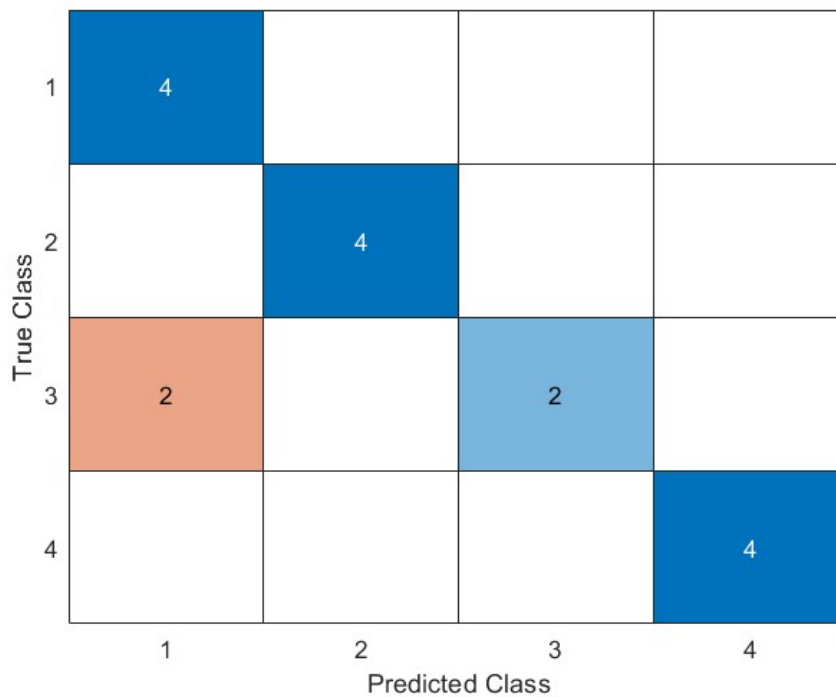


Figura 5.8. Matriz de confusión cuando K=2, precisión = 87.50%

En la Figura 5.8 se observa la matriz de confusión del algoritmo KNN implementado con selección dirigida, no aleatoria, de los datos. Se obtiene una clasificación con una precisión del 87,50%, un valor altamente satisfactorio considerando la limitada disponibilidad de datos. Del total de 16 señales de prueba se han clasificado correctamente 14. Son las que aparecen representadas en la diagonal de la matriz. Sin embargo, dos de las señales han resultado mal clasificadas ya que se predijo que debían pertenecer a la persona número 1 y, realmente, pertenecen al número 3.

5.3.1. Mejora del algoritmo de clasificación

Para mejorar el algoritmo KNN implementado es esencial encontrar el equilibrio entre la cantidad y la calidad de los datos para obtener resultados óptimos. Respecto a la calidad de los datos, se puede afirmar que han sido manipulados y limpiados con exactitud acorde a las características de las señales como bien se expone en el Capítulo 4 y 5. Por otro lado, como la calidad es la mejor que podíamos conseguir dado el tiempo de que disponíamos, sólo queda aumentar la cantidad de muestras.

Un conjunto de datos más grande mejora la generalización del modelo, ya que permite aprender patrones más representativos y robustos. Al tener una muestra más grande y diversa, se reduce el sesgo del modelo hacia patrones específicos presentes en

un subconjunto limitado de datos, lo que resulta en resultados más imparciales. Además, un tamaño de muestra más grande proporciona mayor poder estadístico, lo que permite detectar relaciones significativas entre las variables y realizar inferencias más precisas. También se obtiene una mayor estabilidad en las estimaciones del rendimiento del modelo, como la precisión o la sensibilidad, lo que proporciona resultados más consistentes y confiables.

Para ello, se pasó de 21 muestras por persona a 44. Este número de datos no es redondo debido a las pérdidas de muestras tras analizar, procesar y limpiar las nuevas muestras. Posteriormente, tras una reformulación del algoritmo con una simple modificación de parámetros de tamaño se realizaron, como en el apartado anterior, dos clasificaciones: una por métodos de selección automática entre datos de entrenamiento prueba y otra por selección dirigida.

Método aleatorio

A continuación, se muestran los resultados de varias ejecuciones (K = 2 (ver Figura 5.9), K = 3 (ver Figura 5.10) y K = 5 (ver Figura 5.11)).

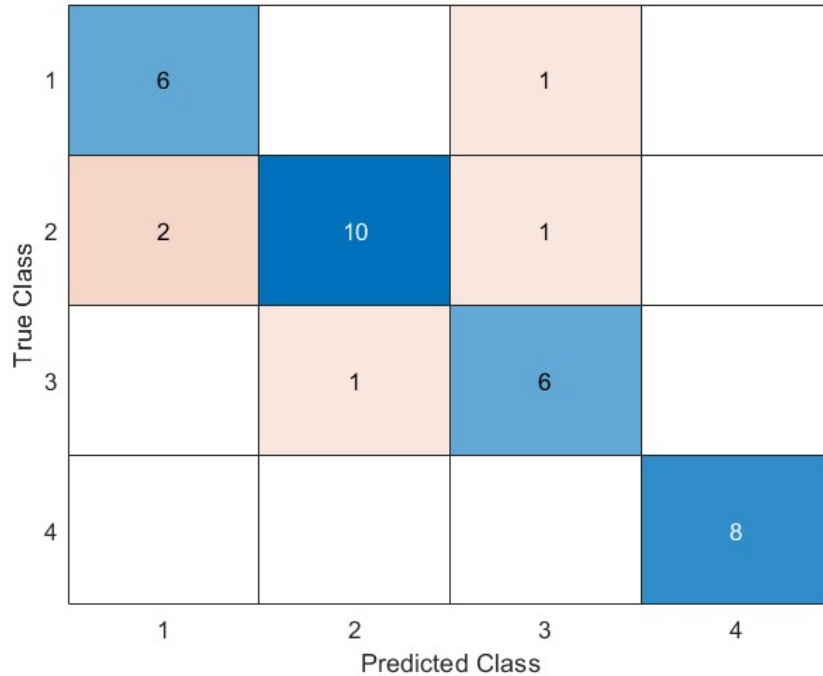


Figura 5.9. Matriz de confusión cuando K=2, precisión= 82,85%

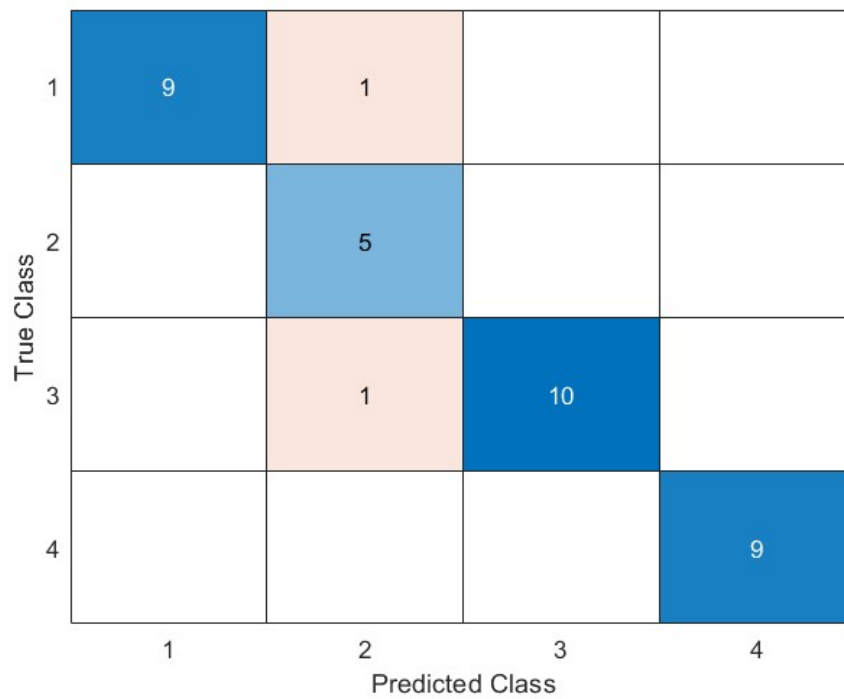


Figura 5.10. Matriz de confusión cuando K=3, precisión= 94,28%

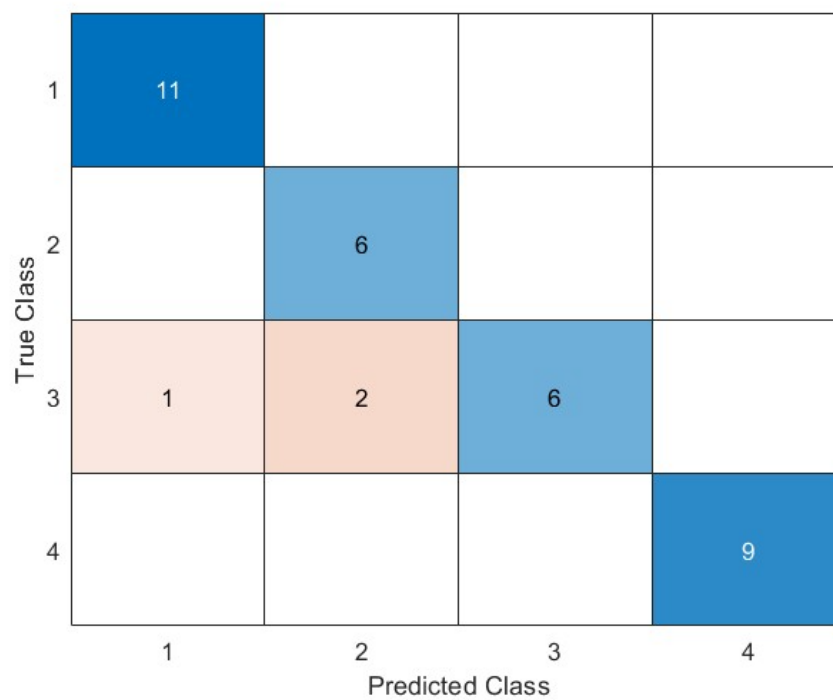


Figura 5.11. Matriz de confusión cuando K=5, precisión=91,42%

Con esta mejora se pasa de oscilar entre el 70-85% a una precisión entre el 85-95%, lo que supone una gran mejora de la clasificación y que sostiene con seguridad la acertada decisión de aumentar el número de muestras.

Método no aleatorio

En este caso, se sigue el mismo procedimiento que en el apartado anterior, pero tomando 44 muestras de cada persona y determinando unas muestras fijas de entrenamiento y prueba. A continuación, la Figura 5.12 muestra los resultados con K igual a 4.

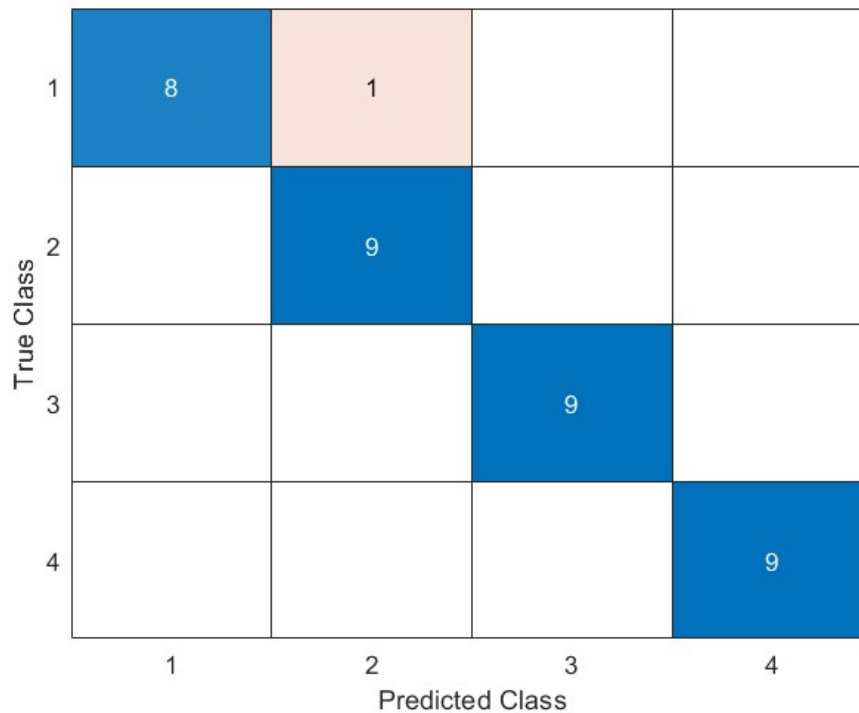


Figura 5.12. Matriz de confusión cuando K=4, precisión= 97,22%

Con estos resultados, se puede afirmar que el aumento de las muestras aporta una precisión mayor, alcanzando un 97,22% de precisión de correcta clasificación. Además, se confirma la calidad de las muestras extraídas con nuestro prototipo, ya que el aumento de las muestras ha permitido aumentar en mucho la precisión del resultado. Significativamente, las muestras no se han tomado en una única sesión, sino que han pasado casi 15 días entre la captura del paquete inicial de datos y el segundo. Esto es también indicativo del valor del ECG como identificador biométrico.

6

Conclusiones y trabajo futuro

6.1. Conclusiones

Las conclusiones obtenidas en este Trabajo Fin de Grado son gratamente satisfactorias ya que se han alcanzado todos los objetivos planteados.

- Se ha montado el sistema ECG para la captura de señales en dos configuraciones diferentes:
 - A partir de Arduino DUE y AD8232
 - A partir de Tiny-BLE de Seeed Studio y AD8232 con transmisión Bluetooth
- Se ha preprocesado la señal, generando un filtro Butterworth de paso bajo que elimina perfectamente el ruido conservando las frecuencias correspondientes al latido cardiaco.
- Se ha segmentado automáticamente las señales ECG permitiendo la identificación de las ondas P, Q, R, S y T.

- Se ha implementado un algoritmo de aprendizaje automático basado en los K vecinos más cercanos (KNN), donde se han planteado dos posibilidades: una selección de datos aleatoria y no aleatoria.
- Complementando el punto anterior, se ha confirmado como aumentar el número de muestras permitía tener mejores resultados.

Además de alcanzar los objetivos principales también se han podido cumplir los subobjetivos relacionados con la clasificación de señales con el algoritmo automático, con el cual se ha podido alcanzar un 97,22% de precisión. Es cierto que solo se han clasificado cuatro sujetos, pero este valor supera altamente las expectativas establecidas ya que, como bien se expone en el transcurso de este estudio, las señales ECG guardan una gran similitud entre diferentes personas.

6.2. Trabajo futuro

Este estudio se podría mejorar aumentando aún más la cantidad de muestras iniciales de cada persona. Este factor va directamente ligado al algoritmo de aprendizaje automático, debido a que, según la definición de este tipo de algoritmos, a mayor número de datos de entrenamiento, el sistema alcanza mayor veracidad y capacidad de decisión. Otra posible mejora se centraría en el desarrollo y estudio de otros algoritmos de clasificación, como pueden ser el LDA y el SVM. Además, a estos últimos algoritmos se le podrían añadir cálculos de características específicas de las señales ECG, como duración de las ondas, amplitud, frecuencia cardiaca etc. de manera que se pudiera especificar qué características y comportamientos exactos de la señal son exclusivos de cada sujeto.

Anexo 1 - Código Keil

Studio Cloud

```
/* mbed Microcontroller Library
 * Copyright (c) 2006-2013 ARM Limited
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */

#include <string.h>
#include "mbed.h"
#include "BLE.h"

#include "UARTService.h"

#define LED1      p21

#define UART_TX   p9
#define UART_RX   p11

DigitalOut pwr (p30);

BLEDevice ble;
DigitalOut led1(LED1);
UARTService *uart;

AnalogIn Entrada(p3);
DigitalIn LeadON1(p5);
DigitalIn LeadON2(p6);
volatile float dato;

Serial pc(UART_TX, UART_RX);

static volatile bool  triggerSensorPolling = false; // booleano: true (hay nuevo dato
del sensor) / false (no hay dato nuevo)

void disconnectionCallback(const Gap::DisconnectionCallbackParams_t *params)
{
    ble.startAdvertising();
}

void periodicCallback(void)
{
    led1 = !led1; /* Do blinky on LED1 while we're waiting for BLE events */
}
```

```

    triggerSensorPolling = true;

if (!LeadON1 && !LeadON2) /* Electrodo conectados*/
    dato = Entrada.read();
}

int main(void)
{
    led1 = 1;
    pwr = 1;
    Ticker ticker;

    ticker.attach_us(&periodicCallback,10000);
    triggerSensorPolling = false;

    pc.baud(115200);

    ble.init();
    ble.onDisconnection(disconnectionCallback);

    uart = new UARTService(ble);

    /* setup advertising */
    ble.accumulateAdvertisingPayload(GapAdvertisingData::BREDR_NOT_SUPPORTED);
    ble.setAdvertisingType(GapAdvertisingParams::ADV_CONNECTABLE_UNDIRECTED);
    ble.accumulateAdvertisingPayload(GapAdvertisingData::SHORTENED_LOCAL_NAME,
                                     (const uint8_t *)"BLE UART", sizeof("BLE UART") -
1);

    ble.accumulateAdvertisingPayload(GapAdvertisingData::COMPLETE_LIST_128BIT_SERVICE_IDS
, (const uint8_t *)UARTServiceUUID_reversed, sizeof(UARTServiceUUID_reversed));

    ble.setAdvertisingInterval(160); /* 100ms; in multiples of 0.625ms. */
    ble.startAdvertising();

    char buffer [20];

    while (true) {
        // check for trigger from periodicCallback()
        if (triggerSensorPolling /*&& ble.getGapState().connected*/) {
            triggerSensorPolling = false;

            sprintf (buffer, "%.3f\n", dato);

            uart->write(buffer, strlen(buffer));
            pc.printf("%.6f\n\n",dato);

        } else {
            ble.waitForEvent(); // low power wait for event
        }
    }
}

```

Anexo 2 - Código filtro

Butterworth

```
load('datosPersona1_Ajustada.mat'); % Cargar la señal ECG desde un archivo
fc = 30; % Frecuencia de corte del filtro (Hz)
fs = 100; % Frecuencia de muestreo de la señal (Hz)
ecg_signal = datosPersona1_Ajustada;
[b, a] = butter(5, fc/(fs/2), 'low'); % Coeficientes del filtro
ecg_filtered = filtfilt(b, a, datosPersona1_Ajustada); % Aplicar filtro a la señal
t = (0:length(datosPersona1_Ajustada)-1)/fs;% Vector de tiempo
figure
for i = 1:10
    subplot(5,2,i);
    plot(t, datosPersona1_Ajustada(:, i))
    xlabel('Tiempo (s)');
    ylabel('Amplitud (mV)');
    title(['Señal ECG Original N°',num2str(i)]);
end
figure
for i = 1:10
    subplot(5,2,i);
    plot(t, ecg_filtered(:, i))
    xlabel('Tiempo (s)');
    ylabel('Amplitud (mV)');
    title(['Señal ECG Filtrada N°',num2str(i)]);
end
```

Anexo 3 - Código

segmentación

```
load('datosPersona1_Filtrada.mat')
PruebaPersona1_XCicloCardiaco = [];
for j = 1:10
ecg_filtered = datosPersona1_Filtrada(:,4);
fs = 100;
% Detectar los complejos QRS de la señal ECG
[qrs_amp_raw,qrs_i_raw,delay] = pan_tompkin(ecg_filtered,fs,1); % Detalles de la
función pan_tompkin() omitidos por brevedad
% Calcular la duración de un ciclo cardiaco (en segundos)
cycle_duration = 0.8;
% Calcular los intervalos de tiempo correspondientes a las diferentes partes del ciclo
cardiaco
interval_p = floor((cycle_duration*fs)/5);
interval_q = floor((cycle_duration*fs)/10);
interval_r = floor((cycle_duration*fs)/10);
interval_s = floor((cycle_duration*fs)/10);
interval_t = floor((cycle_duration*fs)/5);

figure;
hold on;
plot(ecg_filtered);
for i = 2:(length(qrs_i_raw))
    cycle_start = qrs_i_raw(:,i)/100;
    p_start = cycle_start - interval_q/100 - interval_p/100;
    p_end = p_start + (interval_p/100);
    q_start = p_end;
    q_end = q_start + (interval_q/100) - (interval_r/100)/2;
    r_start = q_end;
    r_end = r_start + (interval_r/100);
    s_start = r_end;
    s_end = s_start + interval_s/100;
    t_start = s_end;
    t_end = t_start + interval_t/100;
    %p
    indiceS = floor(p_start * fs) + 1;
```

```

valorS = ecg_filtered(indiceS);
indiceE = floor(p_end * fs) + 1;
valorE = ecg_filtered(indiceE);
ecg_p = ecg_filtered(indiceS:(indiceS+17));
%q
indiceSq = floor(q_start * fs) + 1;
valorSq = ecg_filtered(indiceSq);
indiceEq = floor(q_end * fs) + 1;
valorEq = ecg_filtered(indiceEq);
ecg_q = ecg_filtered(indiceSq:indiceSq+5);
%r
indiceSr = floor(r_start * fs) + 1;
valorSr = ecg_filtered(indiceSr);
indiceEr = floor(r_end * fs) + 1;
valorEr = ecg_filtered(indiceEr);
ecg_r = ecg_filtered(indiceSr:indiceSr+9);
%s
indiceSs = floor(s_start * fs) + 1;
valorSs = ecg_filtered(indiceSs);
indiceEs = floor(s_end * fs) + 1;
valorEs = ecg_filtered(indiceEs);
ecg_s = ecg_filtered(indiceSs:indiceSs+9);
%t
indiceSt = floor(t_start * fs) + 1;
valorSt = ecg_filtered(indiceSt);
indiceEt = floor(t_end * fs) + 1;
valorEt = ecg_filtered(indiceEt);
ecg_t = ecg_filtered(indiceSt:indiceSt+17);

hold on;
plot(indiceS:indiceS+17 ,ecg_p,'r');
plot(indiceSq:indiceSq+5, ecg_q, 'g')
plot(indiceSr:indiceSr+9, ecg_r, 'c')
plot(indiceSs:indiceSs + 9, ecg_s, 'k')
plot(indiceSt:indiceSt +17, ecg_t, 'm')
PruebaPersona1_XCicloCardiaco = [ PruebaPersona1_XCicloCardiaco
ecg_filtered(indiceS:indiceS+56)];
title('Señal ECG filtrada y segmentada por intervalos');
xlabel('Muestras');
ylabel('Amplitud (mV)');
legend('Señal filtrada','Onda P','Onda Q', 'Onda R', 'Onda S', 'Onda T');
pause(2)
end
end
hold off;

```

Anexo 4 - Código K-Nearest Neighbours (selección aleatoria)

```
datos_norm = (X);
% División en conjuntos de entrenamiento y prueba (80% entrenamiento, 20% prueba)
porcentaje_entrenamiento = 0.8;
indices_entrenamiento = randperm(size(datos_norm, 1), round(porcentaje_entrenamiento *
size(datos_norm, 1)));
indices_prueba = setdiff(1:size(datos_norm, 1), indices_entrenamiento);
datos_entrenamiento = datos_norm(indices_entrenamiento, :);
datos_prueba = datos_norm(indices_prueba, :);
% Etiquetas de clase correspondientes a las señales ECG (asumiendo 4 personas)
etiquetas = [ones(21, 1); 2*ones(21, 1); 3*ones(21, 1); 4*ones(21, 1)];
etiquetas_entrenamiento = etiquetas(indices_entrenamiento);
etiquetas_prueba = etiquetas(indices_prueba);
% Paso 2: Cálculo de distancias
% Función para calcular la distancia Euclidiana entre dos señales
distancia_euclidiana = @(x1, x2) norm(x1 - x2);
% Paso 3: Encuentra los vecinos más cercanos
% Rango de valores de K a probar
k_values = 1:10;
% Variables para almacenar los resultados
mejor_precision = 0;
mejor_k = 0;
% Bucle para probar diferentes valores de K
for k = k_values
    % Entrenar el modelo con el valor actual de K
    modelo = fitcknn(datos_entrenamiento, etiquetas_entrenamiento, 'NumNeighbors', k);

    % Realizar la clasificación en los datos de prueba
    etiquetas_predichas = predict(modelo, datos_prueba);

    % Calcular la precision del modelo
```

```

precision = sum(etiquetas_predichas == etiquetas_prueba) / numel(etiquetas_prueba);

% Verificar si esta precisión es la mejor hasta ahora
if precision > mejor_precision
    mejor_k = k;
    mejor_precision = precision;

end

end
K = mejor_k;
etiquetas_predichas = zeros(size(datos_prueba, 1), 1);
for i = 1:size(datos_prueba, 1)
    distancias = zeros(size(datos_entrenamiento, 1), 1);
    for j = 1:size(datos_entrenamiento, 1)
        distancias(j) = distancia_euclidiana(datos_prueba(i, :),
datos_entrenamiento(j, :));
    end
    [~, indices_ordenados] = sort(distancias);
    k_indices = indices_ordenados(1:K);
    k_etiquetas = etiquetas_entrenamiento(k_indices);

    % Paso 4: Realiza la clasificación
    etiquetas_predichas(i) = mode(k_etiquetas);
end

% Paso 5: Evalúa el rendimiento del algoritmo
disp(['Mejor valor de K: ' num2str(mejor_k)]);
disp(['Precisión correspondiente: ' num2str(mejor_precision)]);
%Calcular la matriz de confusión
C = confusionmat(etiquetas_prueba, etiquetas_predichas);
% Visualizar la matriz de confusión en un gráfico
confusionchart(etiquetas_prueba, etiquetas_predichas);

```

Anexo 5 - Código K-Nearest Neighbours (selección no aleatoria)

```
datosnom = X;
semilla = 123;
rng(semilla);
num_personas = 4; % Número total de personas
muestras_por_persona = 21; % Número de muestras por persona
% División en conjuntos de entrenamiento y prueba para cada persona
porcentaje_entrenamiento = 0.8;
indices_entrenamiento = [];
indices_prueba = [];
for persona = 1:num_personas
    inicio_persona = (persona - 1) * muestras_por_persona + 1;
    fin_persona = persona * muestras_por_persona;
    indices_persona = randperm(muestras_por_persona, round(porcentaje_entrenamiento *
muestras_por_persona));

    indices_entrenamiento = [indices_entrenamiento, inicio_persona + indices_persona -
1];
    indices_prueba = [indices_prueba, setdiff(inicio_persona:fin_persona,
inicio_persona + indices_persona - 1)];
end
datos_entrenamiento = datos_norm(indices_entrenamiento, :);
datos_prueba = datos_norm(indices_prueba, :);
% Etiquetas de clase correspondientes a las señales ECG
etiquetas = repelem(1:num_personas, muestras_por_persona)';
etiquetas_entrenamiento = etiquetas(indices_entrenamiento);
etiquetas_prueba = etiquetas(indices_prueba);
% Paso 2: Cálculo de distancias
% Función para calcular la distancia Euclidiana entre dos señales
distancia_euclidiana = @(x1, x2) norm(x1 - x2);
% Paso 3: Encuentra los vecinos más cercanos
```

```

% Rango de valores de K a probar
k_values = 1:10;
% Variables para almacenar los resultados
mejor_precision = 0;
mejor_k = 0;
% Bucle para probar diferentes valores de K
for k = k_values
    % Entrenar el modelo con el valor actual de K
    modelo = fitcknn(datos_entrenamiento, etiquetas_entrenamiento, 'NumNeighbors', k);

    % Realizar la clasificación en los datos de prueba
    etiquetas_predichas = predict(modelo, datos_prueba);
    % Calcular la precisión del modelo
    precision = sum(etiquetas_predichas == etiquetas_prueba) / numel(etiquetas_prueba);
    % Verificar si esta precisión es la mejor hasta ahora
    if precision > mejor_precision
        mejor_precision = precision;
        mejor_k = k;
    end
end
K = mejor_k;
etiquetas_predichas = zeros(size(datos_prueba, 1), 1);
for i = 1:size(datos_prueba, 1)
    distancias = zeros(size(datos_entrenamiento, 1), 1);
    for j = 1:size(datos_entrenamiento, 1)
        distancias(j) = distancia_euclidiana(datos_prueba(i, :),
datos_entrenamiento(j, :));
    end
    [~, indices_ordenados] = sort(distancias);
    k_indices = indices_ordenados(1:K);
    k_etiquetas = etiquetas_entrenamiento(k_indices);

    % Paso 4: Realiza la clasificación
    etiquetas_predichas(i) = mode(k_etiquetas);
end
% Paso 5: Evalúa el rendimiento del algoritmo
% Mostrar el mejor valor de K y su precisión correspondiente
disp(['Mejor valor de K: ' num2str(mejor_k)]);
disp(['Precision correspondiente: ' num2str(mejor_precision)]);
figure;
%Calcular la matriz de confusión
C = confusionmat(etiquetas_prueba, etiquetas_predichas);
% Visualizar la matriz de confusión en un gráfico
confusionchart(etiquetas_prueba, etiquetas_predichas);

```

Referencias

[1] Pereira, T.M.C.; Conceição, R.C.; Sebastião, R. Initial Study Using Electrocardiogram for Authentication and Identification. *Sensors*, 22, 2202.

<https://doi.org/10.3390/s22062202>

[2] Efesalud, R. (2022). Cerca del 60 % de los españoles tiene dos o más factores de riesgo cardiovascular. EFE Salud. <https://efesalud.com/factores-riesgo-cardiovascular-resultados-encuesta-salud-fundacion-corazon/>

Acceso el 2 de mayo de 2023

[3] Sociedad Española de Cardiología

https://secardiologia.es/images/multimedia/infografias/MORTALIDAD_CARDIOVASCULAR_2020.pdf

Acceso el 1 de mayo de 2023

[4] Derivation of the limb leads

https://commons.wikimedia.org/wiki/File:Limb_leads_of_EKG.png

Acceso el 1 de mayo de 2023

[5] Doctors Network

https://www.medicinembbs.org/2011/02/electrocardiogram_21.html

Acceso el 1 de mayo de 2023

[6] Alila Medical Media en Español. (2016, 1 febrero). El Sistema de Conducción Cardíaca y la Relación con ECG, Animación. Alila Medical Media Español. [Vídeo].

YouTube. <https://www.youtube.com/watch?v=2v7T-O2CWWk>

Acceso el 1 de mayo de 2023

[7] Colaboradores de Wikipedia. (2023). Electrocardiograma. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Electrocardiograma>

Acceso el 2 de mayo de 2023

- [8] Trastornos de la conducción | NHLBI, NIH. (2022, 24 marzo). NHLBI, NIH.
<https://www.nhlbi.nih.gov/es/salud/trastornos-de-la-conduccion>
Acceso el 2 de mayo de 2023
- [9] Palestina, D. E. M. (2019). Análisis de características estadísticamente significativas en el dominio temporal de señales ECG y PPG para identificación biométrica bimodal.
https://www.lareferencia.info/vufind/Record/MX_1df5a4a737564f3bcec6df0f6122faac
Acceso el 4 de mayo de 2023
- [10] Jain, A. K., Ross, A., & Nandakumar, K. (2016). Introduction to biometrics. Springer. <https://www.springer.com/gp/book/9780387733838>
Acceso el 4 de mayo de 2023
- [11] NIST. (2019). Biometric recognition: Performance metrics and associated testing methodologies. Special Publication 800-76-2.
<https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-76-2.pdf>
Acceso el 4 de mayo de 2023
- [12] Kardia, by AliveCor
<https://www.kardia.com/kardiamobile/>
Acceso del 7 de mayo de 2023
- [13] M. Patrick Witvliet, Evert P.M. Karregat, Jelle C.L. Himmelreich, Jonas S.S.G. de Jong, Wim A.M. Lucassen, Ralf E. Harskamp, Usefulness, pitfalls and interpretation of handheld single-lead electrocardiograms, Journal of Electrocardiology 66, 2021, 33-37
<https://doi.org/10.1016/j.jelectrocard.2021.02.011>
- [14] SparkFun Single Lead Heart Rate Monitor - AD8232. (s. f.). SEN-12650 - SparkFun Electronics. <https://www.sparkfun.com/products/12650>
Acceso el 7 de mayo de 2023
- [15] AD8232 - Single Lead, Heart Rate Monitor Front End
<https://www.analog.com/media/en/technical-documentation/data-sheets/AD8232.pdf>
Acceso el 11 de mayo de 2023

[16] Signal Processing Toolbox. (s. f.). <https://es.mathworks.com/products/signal.html>
Acceso el 11 de mayo de 2023

[17] Storr, W. (2022). Butterworth Filter Design. Basic Electronics Tutorials.
https://www.electronics-tutorials.ws/filter/filter_8.html
Acceso el 11 de mayo de 2023

[18] ARM Keil Studio Cloud
<https://studio.keil.arm.com/>
Acceso el 12 de mayo de 2023

[19] Portable Signal Processing Kit with ECG
<https://www.hackster.io/celikthomas/portable-signal-processing-kit-with-ecg-d7e52c>
Acceso el 12 de mayo de 2023

[20] Sosa-Trejo, D., Bandera, A., González, M., & Hernández-León, S. (2023). Vision-based techniques for automatic marine plankton classification. Artificial Intelligence Review. <https://doi.org/10.1007/s10462-023-10456-w>
Acceso el 12 de mayo de 2023

[21] Tiny BLE | Seeed Studio Wiki
https://wiki.seeedstudio.com/Tiny_BLE/
Acceso el 12 de mayo de 2023

[22] Colaboradores de Wikipedia. (s. f.). Archivo:Arduino Logo Registered.svg - Wikipedia, la enciclopedia libre.
https://es.m.wikipedia.org/wiki/Archivo:Arduino_Logo_Registered.svg
Acceso el 12 de mayo de 2023

[23] MATLAB Logo. (2022, 1 junio). símbolo, significado logotipo, historia, PNG.
<https://1000marcas.net/matlab-logo/>
Acceso el 12 de mayo de 2023

[24] Bluetooth Low Energy / BLE_API
http://os.mbed.com/teams/Bluetooth-Low-Energy/code/BLE_API/
Acceso el 15 de mayo de 2023

[25] nRF51822 - Nordic stack and drivers for the mbed BLE API | Mbed

<https://os.mbed.com/teams/Nordic-Semiconductor/code/nRF51822/>

Acceso el 15 de mayo de 2023

[26] Hooman Sedghamiz (2023). Complete Pan Tompkins Implementation ECG QRS detector. MATLAB Central File Exchange.

<https://www.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector>

Acceso el 18 de mayo de 2023.



UNIVERSIDAD
DE MÁLAGA

| uma.es

E.T.S de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA