



*ugr* | Universidad  
de **Granada**

TRABAJO FIN DE GRADO  
INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

# Extracción de complejos QRS fetales en señales ECG abdominales

---

**Autor**  
Pedro Álvarez Guirado

**Directores**  
Encarnación Castillo Morales  
Diego Pedro Morales Santos



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE  
TELECOMUNICACIÓN

Granada, Julio de 2016





---

El tribunal constituido para la evaluación del TFG titulado:

## **Extracción de complejos QRS fetales en señales ECG abdominales**

Realizado por el alumno: **Pedro Álvarez Guirado**

Y dirigido por los tutores: **Encarnación Castillo Morales y D. Pedro Morales Santos**

Ha resuelto asignarle la calificación de:

- SOBRESALIENTE (9 - 10 puntos)
- NOTABLE (7 - 8.9 puntos)
- APROBADO (5 - 6.9 puntos)
- SUSPENSO

Con la nota:                    puntos.

El Presidente:

El Secretario:

El Vocal:

Granada, Julio de 2016.





*ugr* | Universidad  
de **Granada**

INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

## Extracción de complejos QRS fetales en señales ECG abdominales

REALIZADO POR:

Pedro Álvarez Guirado

DIRIGIDO POR:

Encarnación Castillo Morales

Diego Pedro Morales Santos

DEPARTAMENTO:

Electrónica y Tecnología de Computadores

Granada, Julio de 2016



---

# Extracción de complejos QRS fetales en señales ECG abdominales

Trabajo Fin de Grado, Universidad de Granada

Pedro Álvarez Guirado

## PALABRAS CLAVE

Electrocardiograma fetal, frecuencia cardíaca fetal, *clustering*, clasificación no supervisada, ECG, transformada *wavelet*, extracción de complejos QRS fetales, distancia, *k-means*, *k-medoids*, *fuzzy c-means*, *dominant sets*, algoritmos jerárquicos, MATLAB, GUI, FPGA.

## RESUMEN

La frecuencia cardíaca fetal, tomada mediante técnicas no invasivas, es de gran importancia clínica para monitorizar el estado de salud del feto durante el embarazo. A día de hoy, el análisis de las señales ECG fetales continua en un estado inicial de desarrollo. Esto es debido principalmente a la baja relación señal a ruido de la señal ECG fetal, la presencia de artefactos, etc.

El presente Trabajo Fin de Grado presenta un resumen de los diferentes algoritmos de clasificación no supervisada que pueden ser utilizados en la extracción de complejos QRS fetales en señales ECG abdominales, las cuales primeramente se procesan mediante el uso de la transformada *wavelet*, para eliminar los artefactos, el *baseline wandering*, y las interferencias. Hemos utilizado estos algoritmos para clasificar estas señales en tres grupos: ruido, complejos QRS maternos y fetales. Algunos de los algoritmos utilizados en este proyecto han sido: *k-means*, *k-medoids*, *fuzzy c-means*, jerárquico, *dominant sets*. Además, se ha implementado una GUI, en el lenguaje de programación MATLAB, capaz de clasificar nuestras señales automáticamente, extrayendo su FHR y representando estas señales con los complejos QRS fetales marcados.

Por último, este trabajo se encuadra dentro de las líneas de investigación llevadas por los tutores de este TFG. Los resultados obtenidos permitirán una futura implementación hardware, compuesta por una FPGA, encargada de la eliminación del ruido de la señal mediante la transformada *wavelet* y de la extracción de los complejos QRS fetales a través de algoritmos de *Clustering*.



---

# Fetal QRS complex detection from abdominal ECG signals

Bachelor's Thesis, University of Granada

Pedro Álvarez Guirado

## KEYWORDS

Fetal electrocardiogram, fetal heart rate, clustering, unsupervised learning, ECG, wavelet transform, QRS complex, distance, k-means, k-medoids, fuzzy c-means, dominant sets, hierarchical clustering, MATLAB, GUI, FPGA

## ABSTRACT

Non-invasive fetal heart rate is of great relevance in clinical practice to monitor fetal health state during pregnancy. To date, however, the analysis of abdominal fetal ECG is considered a challenging problem for biomedical and signal processing communities. This is mainly due to the low signal-to-noise ratio of fetal ECG and the difficulties in cancellation of maternal QRS complexes, motion, etc.

In this Thesis we present a survey of different unsupervised algorithms for the detection of fetal QRS complex from abdominal ECG signals. These signals are first pre-processed using the wavelet transform to remove artefacts, baseline wandering and interferences. Later, we use clustering algorithms to classify our signals in three groups: noise, maternal QRS complexes and fetal QRS complexes. The algorithms used in this project are: hierarchical, k-means, k-medoids, fuzzy c-means, and dominant sets. We have developed a GUI in MATLAB able to classify our signals automatically and show us the FHR and the aECG signal with all the fetal QRS complexes marked.

Lastly, this work is embedded in the research framework carried out by this Thesis' advisors. The results of this Thesis will allow a future hardware implementation, based on FPGA technology.



---

Dª. Encarnación Castillo Morales, profesora del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada, como directora del Trabajo Fin de Grado de Pedro Álvarez Guirado

D. Diego Pedro Morales Santos, profesor del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada, como director del Trabajo Fin de Grado de Pedro Álvarez Guirado

Informan:

que el presente trabajo, titulado:

***Extracción de complejos QRS fetales en señales ECG abdominales***

Ha sido realizado y redactado por el mencionado alumno bajo nuestra dirección, y con esta fecha autorizamos a su presentación.

Granada, Julio de 2016.

Fdo. Encarnación Castillo Morales

Fdo. Diego Pedro Morales Santos



---

Los abajo firmantes autorizan a que la presente copia de Trabajo Fin de Grado se ubique en la Biblioteca del Centro y/o departamento para ser libremente consultada por las personas que lo deseen.

Granada, Julio de 2016.

Fdo. Encarnación Castillo Morales

Fdo. Diego Pedro Morales Santos

Fdo. Pedro Álvarez Guirado



---

## Agradecimientos

En primer lugar quisiera agradecer la realización del presente proyecto a mis tutores Encarnación Castillo Morales y Diego Pedro Morales Santos, por la labor y dedicación realizada. Sin su ayuda nada de esto habría sido posible.

Por encima de todo quiero dedicar este trabajo a mi familia. Especialmente a mis padres, Antonio y María del Carmen, hermano, Fernando, y *aquellos que se marcharon demasiado pronto*. Vosotros sois *la luz que guía mi camino*.

Quisiera también, agradecer este proyecto a la otra gran familia de nuestra vida, mis amigos, Juan Jesús, Juan Carlos, Tomás, Jesús, Álvaro y Rocío. A mis compañeros de clase durante esta etapa, quienes se han convertido en verdaderos amigos, Víctor, José, Fran, Juan David, Antonio, Carlos, Julio, Manolo y Jesús.

A todas aquellas personas que me han ayudado a aprender de mis errores y ser mejor persona.

*Gracias.*



---

*“Siempre que enseñas, enseña a la vez a dudar de lo que enseñas”*

— José Ortega y Gasset

# Índice general

|           |   |           |
|-----------|---|-----------|
| <b>I</b>  | <b>Introducción y antecedentes</b>  | <b>1</b>  |
| 1.        | Introducción  | 2         |
| 1.1.      | Motivación . . . . .  | 2         |
| 1.2.      | Objetivos . . . . .   | 3         |
| 1.3.      | Estructura de la memoria . . . . .  | 4         |
| 2.        | Teoría y Estado del Arte  | 5         |
| 2.1.      | Electrocardiografía . . . . .   | 5         |
| 2.2.      | Anatomía y fisiología del corazón . . . . .                                   | 6         |
| 2.2.1.    | Procesos celulares del ECG . . . . .  | 6         |
| 2.2.2.    | Bases físicas de la Electrocardiografía . . . . .                             | 9         |
| 2.2.3.    | El electrocardiograma en condiciones normales . . . . .                       | 11        |
| 2.3.      | Electrocardiograma Fetal . . . . .  | 19        |
| 2.3.1.    | Monitorización fetal . . . . .  | 19        |
| 2.3.2.    | Electrocardiografía fetal . . . . .   | 20        |
| 2.4.      | Métodos de extracción de parámetros del fECG . . . . .                        | 23        |
| <b>II</b> | <b>Trabajo realizado</b>  | <b>26</b> |
| 3.        | Organización del proyecto   | 27        |
| 3.1.      | Investigación sobre técnicas de extracción de complejos QRS fetales . . . . . | 27        |
| 3.2.      | Implementación . . . . .  | 28        |
| 4.        | Planificación y estimación de costes  | 30        |
| 4.1.      | Planificación . . . . .   | 30        |
| 4.1.1.    | Investigación . . . . .   | 30        |
| 4.1.1.1.  | Revisión del estado del arte . . . . .  | 30        |
| 4.1.1.2.  | Algoritmos de <i>Clustering</i> . . . . .                                     | 30        |

## ÍNDICE GENERAL

---

|                         |   |           |
|-------------------------|---|-----------|
| 4.1.2.                  | Implementación <i>software</i> . . . . .                              | 31        |
| 4.1.2.1.                | Implementación de los algoritmos de <i>Clustering</i> elegidos        | 31        |
| 4.1.2.2.                | Creación de la GUI . . . . .  | 31        |
| 4.1.3.                  | Implementación <i>hardware</i> . . . . .                              | 31        |
| 4.1.3.1.                | Diseño del sistema <i>hardware</i> . . . . .                          | 31        |
| 4.1.3.2.                | Diseño de la PCB de instrumentación . . . . .                         | 31        |
| 4.1.3.3.                | Programación en VHDL . . . . .  | 31        |
| 4.1.4.                  | Prototipado y testeo . . . . .  | 31        |
| 4.1.4.1.                | Construcción del prototipo . . . . .                                  | 31        |
| 4.1.4.2.                | Testeo en tiempo real . . . . .                                       | 32        |
| 4.1.5.                  | Documentación . . . . .   | 32        |
| 4.1.6.                  | Diagrama de Gantt . . . . .   | 32        |
| 4.2.                    | Recursos . . . . .  | 32        |
| 4.2.1.                  | Humanos . . . . .   | 32        |
| 4.2.2.                  | <i>Hardware</i> . . . . .   | 34        |
| 4.2.3.                  | <i>Software</i> . . . . .   | 34        |
| 4.3.                    | Estimación de costes . . . . .  | 34        |
| 4.3.1.                  | Recursos humanos . . . . .  | 34        |
| 4.3.2.                  | Herramientas . . . . .  | 34        |
| 4.4.                    | Presupuesto . . . . .   | 34        |
| <b>5. Investigación</b> |   | <b>36</b> |
| 5.1.                    | Planteamiento de partida . . . . .                                    | 36        |
| 5.1.1.                  | Preprocesado <i>wavelet</i> de un único paso . . . . .                | 37        |
| 5.1.1.1.                | Transformada <i>wavelet</i> . . . . .                                 | 39        |
| 5.1.1.2.                | Niveles de descomposición <i>L</i> y <i>M</i> . . . . .               | 39        |
| 5.1.2.                  | Extracción y detección de características de la señal aECG . . . . .  | 41        |
| 5.1.2.1.                | Detección de los complejos QRS maternos . . . . .                     | 41        |
| 5.1.2.2.                | Detección de los complejos QRS fetales . . . . .                      | 41        |
| 5.1.2.3.                | Corrección de falsos positivos y falsos negativos . . . . .           | 42        |
| 5.1.2.4.                | Extracción de la FHR y MHR . . . . .                                  | 43        |
| 5.2.                    | Trabajo a realizar . . . . .  | 45        |
| 5.3.                    | El problema de la utilización de umbrales . . . . .                   | 46        |
| 5.4.                    | El problema de la clasificación . . . . .                             | 48        |
| 5.5.                    | Clasificación no supervisada: <i>Clustering</i> . . . . .             | 52        |
| 5.6.                    | Búsqueda de las características y patrones de la señal aECG . . . . . | 55        |

## ÍNDICE GENERAL

---

|   |            |
|---|------------|
| 5.7. Búsqueda de las herramientas de medida de similitud . . . . .        | 59         |
| 5.8. Algoritmos de <i>Clustering</i> . . . . .                            | 64         |
| 5.8.1. Algoritmos jerárquicos . . . . .                                   | 66         |
| 5.8.2. Algoritmos particionales . . . . .                                 | 70         |
| 5.8.2.1. Algoritmo <i>k-means</i> . . . . .                               | 72         |
| 5.8.2.2. Algoritmo <i>k-means++</i> . . . . .                             | 73         |
| 5.8.2.3. Algoritmo <i>k-medoids</i> . . . . .                             | 74         |
| 5.8.2.4. Algoritmo <i>fuzzy c-means</i> . . . . .                         | 75         |
| 5.8.2.5. Algoritmo de conjuntos dominantes . . . . .                      | 76         |
| <b>6. Simulación</b>  | <b>81</b>  |
| 6.1. Bases de datos y señales aECG . . . . .                              | 81         |
| 6.1.1. PhysioNet . . . . .  | 83         |
| 6.2. Implementación software en MATLAB . . . . .                          | 89         |
| 6.2.1. Primeros pasos . . . . .   | 89         |
| 6.2.2. Algoritmo <i>k-means</i> . . . . .                                 | 90         |
| 6.2.3. Algoritmo <i>k-medoids</i> . . . . .                               | 104        |
| 6.2.4. Algoritmo <i>fuzzy c-means</i> . . . . .                           | 107        |
| 6.2.5. Algoritmo de clasificación jerárquica . . . . .                    | 111        |
| 6.3. Criterios de evaluación del número correcto de clusteres . . . . .   | 116        |
| 6.3.1. Criterio de Calinski-Harabasz . . . . .                            | 116        |
| 6.3.2. Criterio de Davies-Bouldin . . . . .                               | 118        |
| 6.3.3. Valor del espacio . . . . .  | 120        |
| 6.3.4. Valor de la silueta . . . . .                                      | 121        |
| 6.3.5. Criterios a utilizar . . . . .                                     | 125        |
| <b>7. Evaluación de resultados</b>  | <b>127</b> |
| 7.1. GUI evaluadora de algoritmos de Clustering en señales aECG . . . . . | 127        |
| 7.1.1. Interfaz y secciones . . . . .                                     | 131        |
| 7.1.2. Pasos de utilización de la GUI y consideraciones a tener en cuenta | 137        |
| 7.2. Resultados . . . . .   | 139        |
| 7.2.1. Corrección de FP y FN . . . . .                                    | 144        |

---

## ÍNDICE GENERAL

|  |            |
|--|------------|
| <b>III Conclusiones y trabajo futuro</b>   | <b>165</b> |
| <b>8. Conclusiones y líneas de investigación futuras</b>                                 | <b>166</b> |
| 8.1. Conclusiones . . . . .  | 166        |
| 8.2. Futura implementación hardware . . . . .  | 167        |
| 8.3. Líneas de investigación futuras . . . . .   | 167        |
| <b>A. Implementación hardware</b>  | <b>171</b> |
| A.1. Arquitectura hardware . . . . .   | 171        |
| A.2. Hardware de la parte de adquisición del ECG . . . . .                               | 174        |
| A.2.1. Arquitectura de un sólo canal . . . . .   | 175        |
| A.2.2. Protección y aislamiento . . . . .  | 176        |
| A.2.3. Reducción de ruido en el modo-común primario: circuito de tierra activo . . . . . | 177        |
| A.2.4. Aumento de la impedancia de entrada: etapa CMOS . . . . .                         | 178        |
| A.2.5. Preamplificación y aislamiento . . . . .  | 178        |
| A.2.6. Filtrado paso alto . . . . .  | 179        |
| A.2.7. Amplificación secundaria . . . . .  | 179        |
| A.2.8. Filtrado paso baja y sobremuestreo . . . . .                                      | 180        |
| A.2.9. Diseño de la PCB . . . . .  | 182        |
| <b>Referencias</b>   | <b>187</b> |

# Índice de figuras

|  |    |
|--|----|
| 2.1. Potencial de acción típico de una célula ventricular de miocardio. Las diferentes Fases están denotadas en la Figura [18]. . . . .  | 6  |
| 2.2. Dipolo eléctrico debido al flujo de corriente en la célula de miocardio en la despolarización. $V_m$ hace referencia al potencial de la transmembrana [18]. . . . .   | 10 |
| 2.3. El torso esférico ideal con la fuente cardiaca centralizada [18]. . . . .   | 11 |
| 2.4. Anatomía del corazón fetal [22]. . . . .  | 12 |
| 2.5. Trayectoria del vector cardíaco en condiciones normales de funcionamiento [18]. . . . .   | 13 |
| 2.6. Colocación en el torso humano de los electrodos [18]. . . . .   | 15 |
| 2.7. Colocación de electrodos representados en el plano frontal [18]. . . . .  | 16 |
| 2.8. Patrón temporal del vector del corazón combinado con la geometría de los electrodos situados sobre el plano frontal [18]. . . . .   | 17 |
| 2.9. Representación gráfica y características normales del ECG [18]. . . . .   | 18 |
| 2.10. Muestra de una señal fECG tomada invasivamente (rojo) a través de un electrodo en el escalpedo, y no-invasivamente (trazas inferiores) mediante electrodos en el abdomen materno.[20] . . . . .  | 21 |
| 2.11. Rangos de amplitud y de frecuencia de las distintas señales biológicas, algunas de las cuales interfieren con la señal cardiaca fetal. Las etiquetas en esta Figura hacen referencia al electrocardiograma materno (mECG), electroencefalograma (mEEG), electrooculograma (mEOG), electromiografía (mEMG), electrohistograma (mEHG), y electrocardiograma fetal (fECG). Tener en cuenta que la amplitud de las señales depende del lugar en el que se toman los datos.[20] . . . . . | 22 |
| 2.12. Representación general de la señal, artefactos y ruido presente en el ECG en el dominio de la frecuencia. La región en la cuál se manifiesta el complejo QRS del fECG está delimitada en la Figura. [20] . . . . .   | 23 |

## ÍNDICE DE FIGURAS

---

|  |    |
|--|----|
| 2.13. Representación del cuerpo materno, en el cuál se ve cómo la medida del ECG abdominal contiene la suma de la señal amECG y fECG, además de la presencia de ruido. . . . .           | 24 |
| 2.14. Etapas para la extracción de parámetros del fECG [8] . . . . .   | 25 |
| 5.1. Diagrama de flujo del método inicial de partida. [6] . . . . .  | 37 |
| 5.2. Señal de la base de datos <i>Daisy</i> , con la estimación de BW y la corrección del ruido de la señal utilizando el método propuesto en [6]. . . . .                               | 43 |
| 5.3. Detección de fQRS sobre la derivada, con presencia de falsos positivos y negativos. . . . .   | 44 |
| 5.4. Izquierda: clasificación binaria. Derecha: clasificación en 3 clases. . . . .   | 49 |
| 5.5. Nubes de puntos en el espacio Euclídeo bidimensional. [21] . . . . .  | 49 |
| 5.6. Soluciones para la clasificación en dos grupos de los datos anteriores. [21] .  | 50 |
| 5.7. Procedimiento de clasificación mediante <i>Clustering</i> . [28] . . . . .  | 53 |
| 5.8. Señal aECG <i>r04.edf</i> de Physionet, canal 4, muestras 5001 a 10000. Azul: señal original sin procesar. Rojo: señal preprocesada mediante transformada <i>wavelet</i> . . . . .  | 56 |
| 5.9. Datos de distancia máximos seguidos de mínimos de la señal aECG ordenados según orden de muestreo . . . . .   | 57 |
| 5.10. Datos de distancia máximos seguidos de mínimos de la señal aECG ordenados según orden de muestreo . . . . .  | 58 |
| 5.11. Datos de distancia máximos seguidos de mínimos de la señal aECG ordenados según orden de muestreo . . . . .  | 59 |
| 5.12. Representación para $\mathbb{R}^2$ de la distancia entre dos puntos mediante uso de la distancia <i>Euclídea</i> (verde) y distancia de <i>Manhattan</i> (el resto). . . . .       | 62 |
| 5.13. Datos representados mediante dos líneas horizontales. [21] . . . . .   | 64 |
| 5.14. Datos clasificados horizontalmente mediante el algoritmo jerárquico <i>Single Linkage</i> . [21] . . . . .   | 65 |
| 5.15. Datos clasificados verticalmente mediante el algoritmo <i>2-means</i> . [21] . . . . .   | 65 |
| 5.16. Dendograma que muestra la diferencia entre los algoritmos jerárquicos <i>divisivos</i> y <i>aglomerativos</i> . . . . .  | 66 |
| 5.17. Representación de datos en el espacio y dendrograma (derecha) resultante de la clasificación de los mismos mediante <i>Clustering</i> jerárquico. [21] . . . . .                   | 68 |
| 5.18. Representación de los vértices pertenecientes a un conjunto dominante, $S$ , marcados dentro de la elipse, con respecto a un vértice externo a este grafo, $V(j)$ . [16] . . . . . | 77 |

## ÍNDICE DE FIGURAS

---

|  |     |
|--|-----|
| 5.19. Representación de los vértices pertenecientes a dos grafos distintos, $S$ y $S - \{i\}$ . [16] . . . . .   | 79  |
| 5.20. Ejemplo del valor del peso entre vértices en un grafo. En el grafo de la izquierda, el peso nos indicará que el vértice 1 pertenece al resto de vértices, mientras que en el grafo de la izquierda, el vértice uno no pertenecerá a este grafo. [16] . . . . . | 80  |
| 6.1. Señal <i>r01.edf</i> de la base de datos <i>Abdominal and Direct Fetal Electrocardiogram</i> . . . . .  | 86  |
| 6.2. Señal <i>ecgca906.edf</i> de la base de datos <i>Non-Invasive Fetal ECG</i> de <i>PhysioNet</i> . . . . .   | 88  |
| 6.3. Suma total de distancias a los centroides en la señal <i>r01.edf</i> realizando varias repeticiones del algoritmo <i>k-means</i> . . . . .  | 93  |
| 6.4. Datos de distancia máximos seguidos de mínimos de la señal <i>r04.edf</i> según orden de muestreo, clasificados con $k = 3$ utilizando el algoritmo <i>k-means++</i> y la distancia Euclídea al cuadrado. . . . .   | 94  |
| 6.5. Datos de distancia máximos seguidos de mínimos de la señal <i>r04.edf</i> según orden de muestreo, clasificados con $k = 3$ utilizando el algoritmo <i>k-means++</i> y la distancia <i>Cityblock</i> . . . . .  | 95  |
| 6.6. Datos de distancia máximos seguidos de mínimos de la señal <i>r04.edf</i> ordenados en orden creciente, clasificados con $k = 3$ utilizando el algoritmo <i>k-means++</i> y la distancia Euclídea al cuadrado. . . . .  | 95  |
| 6.7. Datos de distancia máximos seguidos de mínimos de la señal <i>r04.edf</i> ordenados en orden creciente, clasificados con $k = 3$ utilizando el algoritmo <i>k-means++</i> y la distancia <i>Cityblock</i> . . . . .   | 96  |
| 6.8. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-means</i> . . . . .  | 97  |
| 6.9. Señal aECG <i>r10.edf</i> clasificada utilizando el algoritmo <i>3-means</i> utilizando la distancia Euclídea al cuadrado en datos ordenados temporalmente. . . . .   | 98  |
| 6.10. Señal aECG <i>r10.edf</i> clasificada utilizando el algoritmo <i>3-means</i> utilizando la distancia <i>Cityblock</i> o Euclídea en datos ordenados temporalmente . . . . .  | 99  |
| 6.11. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-means</i> utilizando la distancia Euclídea al cuadrado en datos ordenados por amplitud. . . . .   | 99  |
| 6.12. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-means</i> utilizando la distancia <i>Cityblock</i> o Euclídea en datos ordenados por amplitud. . . . .  | 100 |
| 6.13. Señal aECG <i>r10.edf</i> clasificada utilizando el algoritmo <i>3-means</i> con distancia Euclídea al cuadrado . . . . .  | 101 |

## ÍNDICE DE FIGURAS

---

|   |     |
|---|-----|
| 6.14. Señal aECG <i>r10.edf</i> clasificada utilizando el algoritmo <i>4-means</i> con distancia <i>Cityblock</i> o Euclídea sin elevar al cuadrado ordenando los datos mediante amplitud máx-mín . . . . .                   | 103 |
| 6.15. Señal aECG <i>r10.edf</i> clasificada utilizando el algoritmo <i>4-means</i> con distancia <i>Cityblock</i> o Euclídea sin elevar al cuadrado, ordenando los datos mediante amplitud máx-mín . . . . .                  | 103 |
| 6.16. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-medoids</i> con distancia Euclídea al cuadrado con los datos ordenados . . . . .   | 105 |
| 6.17. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-medoids</i> con distancia Euclídea al cuadrado ordenando los datos mediante amplitud máx-mín .   | 106 |
| 6.18. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-medoids</i> con distancia <i>Cityblock</i> con los datos ordenados temporalmente. . . . .  | 106 |
| 6.19. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>3-medoids</i> con distancia <i>Cityblock</i> ordenando los datos mediante amplitud máx-mín . . . . .   | 107 |
| 6.20. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>fuzzy 3-means</i> ordenando los datos mediante amplitud máx-mín . . . . .  | 108 |
| 6.21. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>fuzzy 3-means</i> , ordenando los datos mediante amplitud máx-mín y con la representación de la <i>pmf</i> de cada dato. . . . .                       | 110 |
| 6.22. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>fuzzy 3-means</i> con distancia Euclídea ordenando los datos mediante amplitud máx-mín, y con la representación de la <i>pmf</i> de cada dato. . . . . | 110 |
| 6.23. Señal aECG <i>r04.edf</i> clasificada utilizando el algoritmo <i>fuzzy 3-means</i> . . .  | 111 |
| 6.24. Dendograma de la señal <i>r04.edf</i> . . . . .   | 114 |
| 6.25. Dendograma de la señal <i>r04.edf</i> , mostrando ejemplos de uniones de inconsistencia y de consistencia. . . . .  | 115 |
| 6.26. Dendograma de la señal <i>r04.edf</i> , dividiendo el árbol en tres grupos . . . .  | 115 |
| 6.27. Dendograma de la señal <i>r04.edf</i> , dividiendo el árbol en 4 grupos . . . .   | 116 |
| 6.28. Valores <i>VRC<sub>k</sub></i> en función del número de grupos, obtenidos de calcular el criterio de <i>Calinski-Harabasz</i> en la señal <i>r04.edf</i> . . . . .  | 118 |
| 6.29. Valores <i>DB</i> en función del número de grupos, obtenidos de calcular el criterio de <i>Davies-Bouldin</i> en la señal <i>r04.edf</i> . . . . .  | 119 |
| 6.30. Valores <i>DB</i> en función del número de grupos, obtenidos de calcular el criterio de <i>Davies-Bouldin</i> en la señal <i>r04.edf</i> . . . . .  | 120 |
| 6.31. Valores <i>gap</i> en función del número de grupos, obtenidos de calcular el criterio de <i>gap</i> en la señal <i>r04.edf</i> . . . . .  | 121 |

## ÍNDICE DE FIGURAS

---

|   |     |
|---|-----|
| 6.32. Evaluación de la señal <i>r04.edf</i> con el criterio de la silueta para $k = 3$ . . . . .  | 123 |
| 6.33. Evaluación de la señal <i>r04.edf</i> con el criterio de la silueta para $k = 4$ . . . . .  | 123 |
| 6.34. Evaluación de la señal <i>r04.edf</i> con el criterio de la silueta para $k = 8$ . . . . .  | 124 |
| 6.35. Valores $S_i$ en función del número de grupos, obtenidos de calcular el criterio de la silueta en la señal <i>r04.edf</i> . . . . .   | 124 |
| 6.36. Valores $S_i$ en función del número de grupos acotados, obtenidos de calcular el criterio de la silueta en la señal <i>r04.edf</i> . . . . .  | 125 |
| <br>7.1. Aspecto de la GUI evaluadora de algoritmos de Clustering en señales aECG antes de ejecutarse . . . . .   | 129 |
| 7.2. Aspecto de la GUI evaluadora de algoritmos de Clustering en señales aECG una vez ejecutada, con los resultados y gráficos de una señal. . . . .  | 130 |
| 7.3. Panel de parámetros para el preprocesamiento de la señal de nuestra GUI .  | 131 |
| 7.4. Panel de parámetros para la clasificación de la señal de nuestra GUI. Aparece también, la casilla de activación de los resultados de <i>Acc<sub>media</sub></i> . . . . .  | 132 |
| 7.5. Cuadros de resultados de nuestra GUI: Cuadro de resultados instantáneos (Cuadro superior), Cuadro de resultados medios (Cuadro inferior izquierda) y Cuadro con los resultados tras la corrección de FP y FN (Cuadro inferior derecha). . . . .  | 134 |
| 7.6. Cuadro con los valores del vector de frecuencia cardíaca en (latidos/min) .  | 135 |
| 7.7. Panel con los gráficos resultantes de la clasificación de nuestra señal: clasificación mediante umbral (esq. superior izq.), clasificación mediante clustering (esq. superior der.), representación de los grupos clasificados mediante clustering (esq. inferior izq.) y dendograma jerárquico de clasificación (esq. inferior der.). . . . . | 136 |
| 7.8. Valores medios para el FHR medidos en bpm . . . . .  | 144 |
| 7.9. Señal aECG <i>r08.edf</i> entret = 5s y t = 10s clasificada tras la corrección de FP y FN. . . . .   | 145 |
| <br>A.1. Diagrama de bloques de la arquitectura hardware inicial. [24] . . . . .  | 172 |
| A.2. Diagrama de adquisición de la señal ECG simplificado. [18] . . . . .   | 175 |
| A.3. Circuito para adquirir la señal ECG a partir de tres derivaciones, realizada con tres electrodos. [18] . . . . .   | 176 |
| A.4. Circuito de tierra activa utilizado para la reducción de ruido en modo común. [18] . . . . .   | 178 |
| A.5. Amplificador operacional en configuración no inversora con retroalimentación negativa . . . . .  | 180 |

## ÍNDICE DE FIGURAS

---

|  |     |
|--|-----|
| A.6. Señal limitada en banda a aproximadamente 225 Hz (debido a la frecuencia de corte de 1 kHz del filtro <i>antialiasing</i> ) muestreada a 300 kHz, la cuál contiene copias espectrales en los múltiplos de 300 Hz y sufre de <i>aliasing</i> (superposición) . . . . . | 181 |
| A.7. La misma señal, limitada en banda a 225 Hz pero muestreada a 1.5 kHz (5 veces el muestreo anterior) posee copias espectrales en múltiplos de 1.5 kHz y no sufre de aliasing. . . . .  | 181 |
| A.8. Después de aplicar un filtro digital LPF de orden elevado con una decimación de un factor de 5, el espectro está espaciado por múltiplos de 300 Hz pero sin aliasing . . . . .  | 181 |
| A.9. PCB diseñada para la adquisición de la señal aECG. Capa <i>top</i> con plano de masa. . . . .   | 184 |
| A.10. PCB diseñada para la adquisición de la señal aECG. Capa <i>top</i> con plano de masa vista en 3D. . . . .  | 185 |
| A.11. PCB diseñada para la adquisición de la señal aECG. Capa <i>bottom</i> vista en 3D  | 186 |

# Índice de cuadros

|       |   |     |
|-------|---|-----|
| 4.1.  | Diagrama de Gantt del proyecto . . . . .  | 33  |
| 4.2.  | Coste temporal del proyecto. . . . .  | 35  |
| 4.3.  | Presupuesto. . . . .  | 35  |
| 5.1.  | Medidas de disimilitud y similitud para características cuantitativas. . . . .  | 61  |
| 6.1.  | Propiedades de adquisición de la base de datos <i>Abdominal and Direct Fetal Electrocardiogram</i> de <i>PhysioNet</i> . . . . .  | 84  |
| 6.2.  | Señal <i>r01.edf</i> del <i>Non-Invasive Fetal ECG</i> de <i>PhysioNet</i> en formato <i>.mat</i> .   | 85  |
| 6.3.  | Propiedades de adquisición de la base de datos <i>Non-Invasive Fetal ECG</i> de <i>PhysioNet</i> . . . . .  | 85  |
| 6.4.  | Señal <i>ecgca906.edf</i> del <i>Non-Invasive Fetal ECG</i> de <i>PhysioNet</i> en formato <i>.mat</i> . . . . .  | 87  |
| 6.5.  | Resultados de la clasificación de la señal aECG <i>r10.edf</i> en el intervalo $t_0 = 15\text{ s}$ , $t_1 = 20\text{ s}$ , utilizando diferentes metodologías . . . . . | 102 |
| 6.6.  | Criterios de evaluación del número de grupos, $k$ , para los diferentes algoritmos de <i>Clustering</i> . . . . .   | 126 |
| 7.1.  | Valores medios de precisión y coste computacional de los algoritmos de Clustering. . . . .  | 140 |
| 7.2.  | Valores medios de exactitud tras la corrección de FP y FN. . . . .  | 144 |
| 7.3.  | Resultados de clasificación en la señal <i>r01.edf</i> . . . . .  | 147 |
| 7.4.  | Resultados de clasificación en la señal <i>r01.edf</i> . . . . .  | 148 |
| 7.5.  | Resultados de clasificación en la señal <i>r01.edf</i> . . . . .  | 149 |
| 7.6.  | Resultados de clasificación en la señal <i>r01.edf</i> . . . . .  | 150 |
| 7.7.  | Resultados de clasificación en la señal <i>r04.edf</i> . . . . .  | 151 |
| 7.8.  | Resultados de clasificación en la señal <i>r04.edf</i> . . . . .  | 152 |
| 7.9.  | Resultados de clasificación en la señal <i>r04.edf</i> . . . . .  | 153 |
| 7.10. | Resultados de clasificación en la señal <i>r04.edf</i> . . . . .  | 154 |

## ÍNDICE DE CUADROS

---

|  |     |
|--|-----|
| 7.11. Resultados de clasificación en la señal <i>r07.edf</i> | 155 |
| 7.12. Resultados de clasificación en la señal <i>r07.edf</i> | 156 |
| 7.13. Resultados de clasificación en la señal <i>r07.edf</i> | 157 |
| 7.14. Resultados de clasificación en la señal <i>r07.edf</i> | 158 |
| 7.15. Resultados de clasificación en la señal <i>r08.edf</i> | 159 |
| 7.16. Resultados de clasificación en la señal <i>r08.edf</i> | 160 |
| 7.17. Resultados de clasificación en la señal <i>r08.edf</i> | 161 |
| 7.18. Resultados de clasificación en la señal <i>r08.edf</i> | 162 |
| 7.19. Resultados de clasificación en la señal <i>r10.edf</i> | 163 |
| 7.20. Resultados de clasificación en la señal <i>r10.edf</i> | 164 |

# Algoritmos

|      |  |    |
|------|--|----|
| 5.1. | Algoritmo jerárquico de <i>Clustering</i> basado en la unión ( <i>Linkage-based Clustering</i> ) . . . . . | 67 |
| 5.2. | Algoritmo <i>k-means</i> . . . . .   | 72 |
| 5.3. | Algoritmo <i>k-means++</i> . . . . .   | 73 |
| 5.4. | Algoritmo <i>k-medoids con la técnica PAM</i> . . . . .  | 75 |
| 5.5. | Algoritmo <i>fuzzy c-means</i> . . . . .   | 76 |
| 5.6. | Algoritmo <i>dominant sets</i> . . . . .   | 78 |

# Nomenclatura

- Acc      precisión
- AD      Analógico-Digital
- AO      amplificador operacional
- AV      atrio-ventricular
- BW      base wandering
- COIT Colegio Oficial de Ingenieros de Telecomunicación
- DBSCAN Density-Based Spatial Clustering of Applications with Noise
- ECG electrocardiograma
- EEG electroencefalograma
- eHealth electronics-Health
- EHG electrohistograma
- EMG electromiografía
- ETSE Escuela Técnica Superior de Ingeniería de la Universidad de Valencia
- fECG electrocardiograma fetal
- FIFO First In-First Out
- FIR      Finite Impulse Response
- FN      Falsos Negativos
- FP      Falsos Positivos
- FPAA Field-Programmable Analog Array

## **ALGORITMOS**

---

FPGA Field-Programmable Gate Array

fQRS complejos QRS fetales

GUI interfaz gráfica de usuario

HDL Hardware Description Language

HPF High Pass Filter

IA inteligencia artificial

IEEE Institute of Electrical and Electronics Engineers

IP Intellectual Property

LA brazo izquierdo

LL pierna izquierda

LPF Low Pass Filter

MCG magnetocardiograma

mQRS complejos QRS maternos

máx-min máximo local seguido de mínimo local

NIBIB National Institute of Biomedical Imaging and Bioengineering

NIGMS National Institute of General Medical Sciences

OPTICS Ordering Points To Identify the Clustering Structure

PAM Partitioning Around Medoids

PCB Printed Circuit Board

PDV Positive Diagnostic Value

RA brazo derecho

RAM Random Access Memory

ROM Read-Only Memory

SA sino-auricular

SMD Surface Mount Device

SNR Signal-to-Noise Ratio

SVM Support Vector Machines

VHDL VHSIC Hardware Description Language

VHSIC Very High Speed Integrated Circuit

WT transformada wavelet

## **ALGORITMOS**

---

# Parte I

## Introducción y antecedentes

# Capítulo 1

## Introducción

### 1.1. Motivación

El electrocardiograma (ECG), palabra proveniente de las palabras griegas *élektron*; *kardía*; y *grámma*), es la representación gráfica de la actividad eléctrica del corazón. El ECG registra el tamaño, la cadencia y la naturaleza de los impulsos eléctricos presentes en el corazón del paciente. De este modo, el médico puede definir la frecuencia y el ritmo cardíaco y saber, por ejemplo, si el corazón late a un ritmo por encima o por debajo del normal. Además, estos registros facilitan información importante sobre la posibilidad de que exista algún problema en el flujo sanguíneo del músculo cardíaco, hipertrofia del tamaño, o daños en el corazón.

Una de las principales causas por la que se ha escogido el presente trabajo es la gran utilidad del mismo de cara a una posible implementación clínica. Debido a esto, podríamos definir como increíble o anormal, el cómo, en un mundo en el que las nuevas tecnologías a menudo se quedan obsoletas, antes incluso de que lleguen a estar disponibles, un simple registro eléctrico de más de un siglo de antigüedad continúe siendo la llave para prevenir y diagnosticar importantes desordenes clínicos relacionados con la actividad cardíaca.

Es por ello, que el presente documento se centra en la investigación sobre una prueba rutinaria, cuyo bajo coste, facilidad para su implementación y característica no invasiva del paciente la convierten en la prueba “estrella” para detectar anomalías cardíacas en todo el mundo.

Además de las causas anteriores, nos centraremos en una línea de investigación que aún continua en una etapa de investigación inicial, la investigación sobre el corazón fetal, o clínicamente conocida como electrocardiograma fetal (fECG). Razón por la cuál estimamos que el presente proyecto, se convierte en un proyecto de gran repercusión y utilidad de cara a futuras implementaciones clínicas.

La falta de desarrollo sobre el fECG es debida en parte, a la falta de disponibilidad de bases de datos estándares, a una baja SNR (*Signal-to-Noise Ratio*) en el ECG fetal comparado con el materno, causa debida a las interferencias existentes entre el corazón fetal y los electrodos abdominales, así como por el hecho de que el corazón fetal posee un menor tamaño que el materno, lo que implica una menor amplitud en el voltaje extraído en el ECG.

Además, también cabe destacar el menor conocimiento clínico respecto a la función cardíaca fetal, como una de las posibles causas del desarrollo arcaico que ha sufrido hasta ahora el fECG.

Por lo que, concluimos que, el fECG es una técnica de monitorización de la actividad eléctrica del corazón del feto que proporciona valiosa información acerca del estado fisiológico del mismo.

Desde el punto de vista médico, la monitorización fetal es una herramienta esencial, ya que permite evaluar el estado del feto y tomar decisiones durante el parto, y en general, durante todo el período de gestación. Su realización es tan simple como colocar unos electrodos sobre el vientre materno, sin riesgo para la madre ni para el feto.

De entre todas las variables que podemos monitorizar, el análisis de la frecuencia cardíaca fetal (FHR) es una buena forma de evaluar el bienestar del feto, permitiendo la detección de patologías durante el embarazo, con la consiguiente reducción de la mortalidad fetal.

## 1.2. Objetivos

El objetivo principal del presente documento es el siguiente:

- Investigar y explorar los distintos algoritmos de clasificación basados en técnicas de aprendizaje no supervisado para la búsqueda de complejos QRS fetales en señales abdominales del ECG.
- Evaluar los distintos algoritmos, así como sus parámetros óptimos, en términos de precisión (*Acc*) para detectar tanto complejos QRS fetales como maternos, eliminando falsos positivos (FP) y falsos negativos (FN).
- Extraer de forma automática los parámetros de mayor interés clínico, a partir de la señal abdominal del ECG.
  - Frecuencia cardíaca fetal (FHR) instantánea.

## **1. INTRODUCCIÓN**

---

- Variabilidad de la FHR.
- Representación gráfica de los QRS complejos en los que se encuentra las señales fetales

Como objetivos secundarios del presente documento se establecen los siguientes:

- Diseñar un prototipo hardware de extracción de FHR en tiempo real, integrando el algoritmo con mayor eficiencia.
- Mostrar los resultados obtenidos mediante algún tipo de soporte físico.

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

El presente documento se estructura en tres partes, las cuales se detallan a continuación.

#### **INTRODUCCIÓN Y ANTECEDENTES.**

Esta primera parte se centra en la introducción y exposición de las bases teóricas que permiten la correcta comprensión del proyecto. Primero introduciremos los principios de electrocardiografía, necesarios para el entendimiento del fECG. Como última parte de esta sección, comentaremos el trabajo previo sobre el que continua nuestro proyecto, así como a la introducción teórica y estado del arte del mismo.

#### **TRABAJO REALIZADO.**

En esta segunda parte desarrollaremos en profundidad el trabajo e investigación llevada a cabo durante el realizamiento del presente proyecto. Explicaremos brevemente la estructura que hemos seguido a cabo para la realización del proyecto, para después continuar con la exposición de los algoritmos sobre los que se ha investigado, su implementación en software, simulación y pruebas realizadas que se han llevado a cabo. En esta sección comentamos también los resultados obtenidos en las distintas fases del trabajo.

#### **CONCLUSIONES Y TRABAJO FUTURO**

Como última parte del presente documento, presentaremos las conclusiones obtenidas y los diferentes frentes y líneas de investigación que quedan abiertos tras la conclusión de este trabajo. También se presentará una posible implementación *hardware* para el diseño de un futuro prototipo

# Capítulo 2

## Teoría y Estado del Arte

### 2.1. Electrocardiografía

La electrocardiografía (del griego *electro*, relacionado con la actividad eléctrica, *kardio*, corazón, y *graph* una raíz griega que significa escribir) es una rama de la medicina que se encarga de estudiar la actividad eléctrica del corazón, con el fin de diagnosticar y prevenir enfermedades cardiovasculares.

El corazón está comprendido por un músculo, el miocardio, que es conducido rítmicamente por las contracciones y este, a su vez, es el encargado de la circulación sanguínea en el cuerpo. Antes de cada latido, o sístole, una onda eléctrica pasa a través de él, lo cuál conlleva a una contracción miocárdica. El patrón seguido por esta onda eléctrica no es aleatorio, ya que se extiende sobre la estructura cardíaca de manera coordinada, lo que conduce a una sístole coordinada. Este resultado se muestra como un cambio en la diferencia de potencial en la superficie del cuerpo humano. La señal resultante, amplificada y filtrada es conocida como ECG.

Pero, por desgracia, un extenso número de factores pueden afectar al ECG, incluyendo anomalías cardíacas, metabólicas (incluyendo falta de oxígeno, o isquemia), del miocardio, y anomalías macroscópicas de la geometría normal del corazón. El análisis del ECG es una prueba rutinaria de cualquier evaluación médica, debido al papel importante que juega el corazón en la salud humana, y además por su manera no-invasiva de analizarse.

El entendimiento de los principios básicos de un ECG requieren la apreciación de cuatro fenómenos: la electrofisiología de las células, la propagación del pulso eléctrico a través del miocardio, la fisiología de las estructuras específicas del corazón a través de las cuales viajará la señal eléctrica, y por último, la forma en la que este pulso eléctrico conduce a una señal medible en la superficie del cuerpo humano, produciendo el conocido ECG.

## 2. TEORÍA Y ESTADO DEL ARTE

---

### 2.2. Anatomía y fisiología del corazón

#### 2.2.1. Procesos celulares del ECG

Cada latido del corazón es disparado por un potencial, que se origina a partir de un marca-pasos rítmico dentro del corazón y es conducido a través de los órganos para producir la contracción. Al igual que ocurre con otros tejidos eléctricamente activos, la célula miocárdica posee una transmembrana potencial típica,  $V_m$ , de en torno a -80 hasta -90 mV con respecto al fluido extracelular que la rodea. [9].

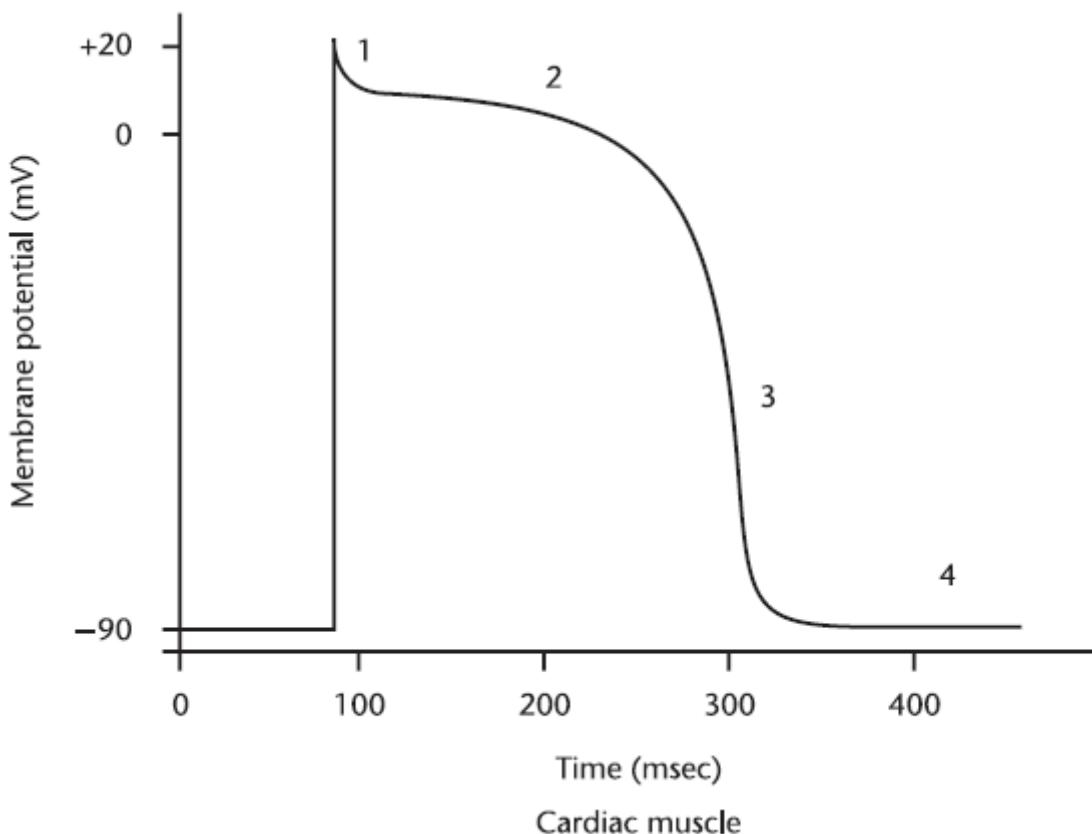


Figura 2.1: Potencial de acción típico de una célula ventricular de miocardio. Las diferentes Fases están denotadas en la Figura [18].

El potencial cardíaco puede ser almacenado por medio del uso de electrodos. La membrana controla la permeabilidad de un número de iones, incluido sodio, potasio, calcio y cloruro. Estos iones pasan la membrana a través de canales específicos de iones que pueden abrirse o cerrarse. Por lo tanto, se dice que estos canales actúan como puertas o canales dependientes de voltaje (*gated channels*), y su apertura y clausura ocurre en función de los cambios de voltaje, o a través de la activación de receptores (*receptor gated channels*).

## 2.2 Anatomía y fisiología del corazón

---

La variación de la conductancia de la membrana debido a la apertura o clausura de los canales de iones, genera cambios en el potencial de la transmembrana a lo largo del tiempo. La variación temporal y su polarización y despolarización para una célula ventricular se ilustra en la siguiente Figura 2.1, con las cinco fases convencionales.

Cuando las células cardíacas se despolarizan a un tensión umbral de unos -70 mV (debido a otra acción de potencial conducida) se produce una rápida despolarización (Fase 0 - ascenso rápido del potencial de acción) causada por un incremento transitorio en la conductancia del canal de sodio. La **Fase 1** representa una re-polarización inicial, causada por la apertura del canal de potasio. Durante la **Fase 2** hay un balance entre la corriente de calcio entrante, y la corriente saliente de potasio, causando un *efecto Plateau* en el potencial de acción, y un retraso en la re-polarización. Este movimiento entrante de iones de calcio se produce a través de canales de calcio que se abren cuando el potencial de la membrana despolarizada ronda los -40 mV. La re-polarización, **Fase 3**, es un proceso complejo y en él están presentes diversos mecanismos. Los incrementos de la conductancia del potasio, tienden a re-polarizar la célula a través de una corriente media saliente de potasio. Además, hay un decremento dependiente del tiempo en la conductividad del calcio, el cuál, además contribuye a la re-polarización celular.

La **Fase 4**, la última fase, es caracterizada por abrir los canales de potasio y el potencial negativo de la transmembrana. Después de la Fase 0, hay un conjunto de procesos moleculares y celulares conocidos como procesos de acoplamiento de excitación-contracción (*excitation-contraction coupling*). La despolarización de las células da lugar a altas concentraciones de calcio intracelular, que a su vez, desbloquea las contracciones enérgico-dependientes de la célula (a través de un cambio del complejo de la proteína troponina).

Antes de que se propague el potencial de acción, este debe de ser iniciado por la acción del marca-pasos cardíaco, el cuál no es más que células cardíacas que poseen la propiedad de la automaticidad. Esto es, las células tienen la habilidad para despolarizarse espontáneamente, por ello funcionan como marca-pasos para el resto de células del corazón. Estas células se encuentran en el nodo sino-auricular (SA), el cuál recibe el nombre de “marca-pasos del corazón”, en el nodo atrio-ventricular (AV) y en ciertos sistemas de conducción especializados dentro de las aurículas y los ventrículos. En las células automáticas, el potencial restante (Fase 4) no es estable, pero muestra una despolarización espontánea: el potencial de la transmembrana aumenta lentamente hacia alcanzar el cero, debido al goteo de iones de sodio y calcio que entran a través de las células marca-pasos. Cuando el potencial de las células alcanza el umbral, estas

## **2. TEORÍA Y ESTADO DEL ARTE**

---

introducen un potencial de acción similar a la Fase 0, descrita anteriormente, pero mediada por el intercambio de calcio a una tasa menor. Tras el potencial de acción, el potencial de la membrana vuelve a los niveles restantes y el ciclo se repite. Cabe destacar que hay distintos niveles de automaticidad en el corazón. La tasa intrínseca del nodo SA es la más alta (alrededor de 60 a 100 latidos por minuto), seguida de la tasa del nodo AV (entre 40 y 50 latidos por minuto), y por último, el músculo ventricular (entre 20 y 40 latidos por minuto). Bajo condiciones de operación normal, el nodo SA determina la frecuencia cardíaca, los marca-pasos más bajos son reseteados durante cada ciclo cardíaco. Sin embargo, en algunas circunstancias patológicas, la tasa de los marca-pasos inferiores puede exceder de la del nodo SA, y por ende, determinar esta la frecuencia cardíaca general.

El potencial de acción, una vez inicializado en la célula cardíaca, se propaga a través de la membrana celular hasta que la célula es despolarizada. Las células de miocardio son las únicas que transmiten los potenciales de acción de una célula a la adyacente por medio de la corriente de difusión. Las corrientes iónicas fluyen de una célula a otra a través de estas conexiones intercelulares, y el corazón se comporta eléctricamente como un sincitio funcional. Entonces, un impulso originado en cualquier parte del miocardio se propagará a través del corazón, resultando en una contracción mecánica por parte de este. Un marca-pasos artificial, por ejemplo, introduce impulsos eléctricos despolarizados a través de un electrodo normalmente localizado en el ventrículo derecho. Los marca-pasos inducen potenciales de acción que excitan todo el miocardio ventricular resultando esto en contracciones mecánicas.

Sin embargo, las principales estructuras intentan modificar la propagación del potencial de acción entremezclándose en todo el corazón. En primer lugar, estás no son más que bandas de fibras de conductancia especializadas a través del potencial de acción, que viaja más rápido comparado con la conducción a través del miocardio. Esto se debe a que viajan a través de una combinación conductible de fibras y del propio miocardio en el que el potencial de acción puede propagarse a todas las regiones de los ventrículos en menos de 100 ms. En sujetos con problemas del sistema de conducción, el tiempo de propagación se prolonga porque el potencial de acción únicamente se propaga a través del propio miocardio. Este movimiento des-sincronizado de varias partes del corazón puede causar un leve deterioro de la eficacia del bombeo. En adición a las fibras conductoras especializadas, hay tejidos que aislan eléctricamente los ventrículos de la aurícula. En un corazón sano, la única manera de que el potencial de acción pase de la aurícula al ventrículo es a través de otra estructura especializada llamada el nodo AV, cuya función es retrasar la conducción, así que la aurícula

puede contraerse completamente antes de que comiencen a hacerlo los ventrículos. La función y la estructura de un corazón sano se discute en profundidad en la siguiente sección.

### 2.2.2. Bases físicas de la Electrocardiografía

Como resultado de la actividad eléctrica en las células, la corriente fluye a través del cuerpo humano, y se establecen diferencias de potencial en la superficie de la piel, las cuales pueden ser medidas utilizando el instrumental necesario. El registro gráfico de estos potenciales, medidos en la superficie corporal, en función del tiempo produce lo que se conoce como electrocardiograma. El modelo matemático más sencillo para relacionar el generador cardíaco con los potenciales de la superficie corporal es el modelo de dipolo (*dipole model*).<sup>[12]</sup> Este modelo simple, es extremadamente útil para proveer un marco de referencia para el estudio de la electrocardiografía clínica.<sup>[17]</sup>

El modelo de dipolo tiene dos componentes, una representación de la actividad eléctrica del corazón (el propio dipolo), y la geometría y propiedades eléctricas del cuerpo circundante. En primer lugar, se realiza la representación de la actividad eléctrica del corazón, como un potencial de acción que se propaga a través de las células (en el miocardio), donde se genera una corriente intracelular asociada en la dirección de propagación, en la interfaz de reposo y en el tejido despolarizante. Esta es la principal fuente eléctrica de la superficie del ECG, referida como corriente de dipolo. Además, hay otra corriente extra-celular igual fluyendo en contra de la dirección de propagación, por lo que la carga se conservará. Todos los bucles de corriente en los medios conductores están cerca sí mismos, formando un dipolo eléctrico, tal y como aparece en la Figura 2.2. La actividad eléctrica total del corazón en cualquier instante de tiempo puede ser representada por una distribución de los dipolos de corriente activos. En general, estos se hallan sobre una superficie irregular correspondiente al límite entre el tejido despolarizado y el polarizado.

Si el corazón fuese suspendido en un medio homogéneo, conductor e isotrópico y fuese observado desde una distancia lo suficientemente grande comparado con su tamaño, se podría suponer que todos estos dipolos de corriente se originan en un único punto en el espacio, y la actividad eléctrica total del mismo puede ser representada como un único dipolo de corriente equivalente cuya magnitud y dirección sea la suma de los vectores de todos los dipolos individuales. El momento dipolar neto es comúnmente referido como **vector del corazón**  $M(t)$ . A medida que cada onda

## 2. TEORÍA Y ESTADO DEL ARTE

---

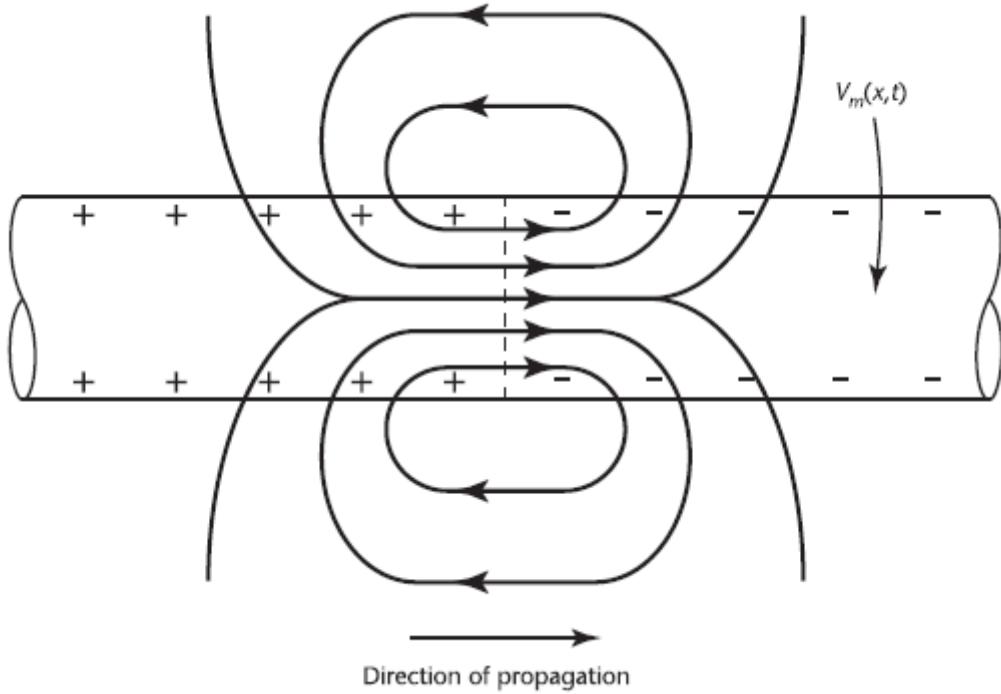


Figura 2.2: Dipolo eléctrico debido al flujo de corriente en la célula de miocardio en la despolarización.  $V_m$  hace referencia al potencial de la transmembrana [18].

de despolarización se propaga a través del corazón, el vector del corazón cambia en magnitud y dirección en función del tiempo.

La distribución de corrientes y potenciales resultante en la superficie depende de las propiedades eléctricas del torso, o abdomen, en el caso del aECG. Como aproximación razonable, el modelo de dipolo ignora la anisotropía y la falta de homogeneidad del torso o abdomen, tratando al cuerpo humano como un sistema lineal, isotrópico, homogéneo, conductor, esférico de radio,  $R$ , y con conductividad,  $\sigma$ . La fuente es representada como un único dipolo de corriente variante en el tiempo, localizado en el centro de la esfera. El campo eléctrico estático, la densidad de corriente, y el potencial eléctrico en cualquier parte del torso están relacionados de manera no dinámica con el vector del corazón en cualquier tiempo dado. El término reactivo debido a la impedancia del tejido puede ser obviado. La ecuación de Laplace puede ser resuelta para dar el potencial de distribución en el torso como:

$$\Phi(t) = \cos\theta(t) 3[M(t)] / 4\pi\sigma R^2 \quad (2.1)$$

donde  $\theta(t)$  es el ángulo entre la dirección del vector del corazón  $M(t)$ , y  $OA$  el vector que une el centro de la esfera,  $O$ , al punto de observación,  $A$  (ver Figura 2.3).

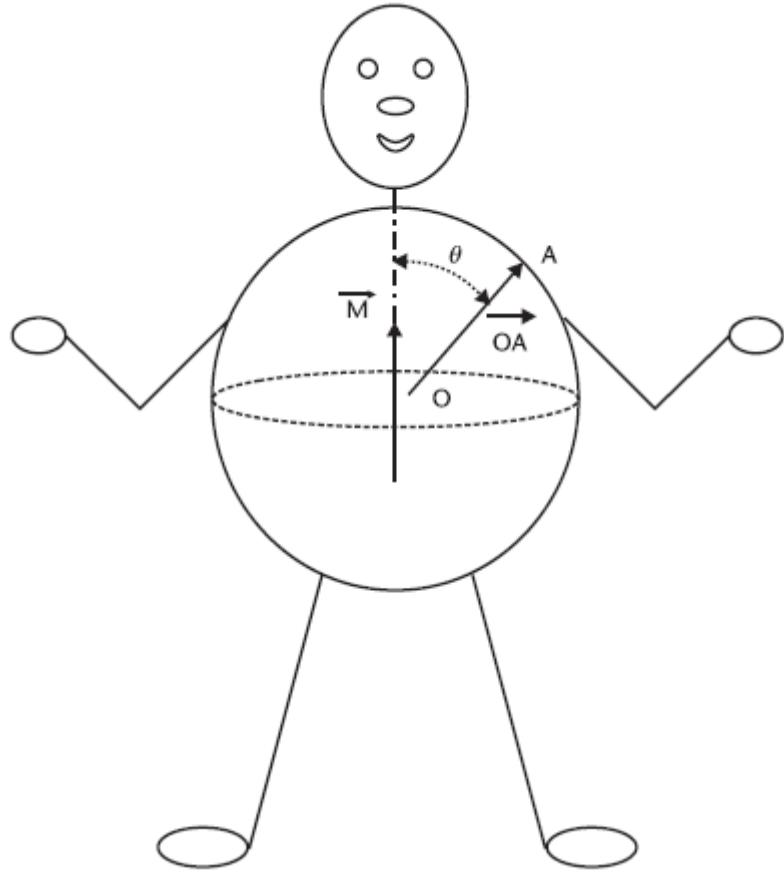


Figura 2.3: El torso esférico ideal con la fuente cardíaca centralizada [18].

$|M|$  es por lo tanto la magnitud del vector del corazón. Generalmente, la diferencia de potencial entre dos puntos de la superficie del torso puede expresarse como:

$$\phi_{CT}(t) = \phi_{RA}(t) + \phi_{LA}(t) + \phi_{LL}(t) \quad (2.2)$$

donde RA hace referencia al brazo derecho, LA al brazo izquierdo y LL hace referencia a la pierna izquierda. Tener en cuenta que  $\phi_{CT}$  deberá ser siempre cero. La siguiente subsección describe la derivación clínica de un ECG en condiciones normales, sin presentar anomalías.

### 2.2.3. El electrocardiograma en condiciones normales

El funcionamiento del corazón como bomba depende principalmente de las propiedades de contracción y relajación del miocardio. Además de este, han de tenerse en cuenta otros factores como: la organización geométrica de las células de miocardio, las propiedades del tejido cardíaco, el ritmo eléctrico del corazón, la función valvular

## 2. TEORÍA Y ESTADO DEL ARTE

---

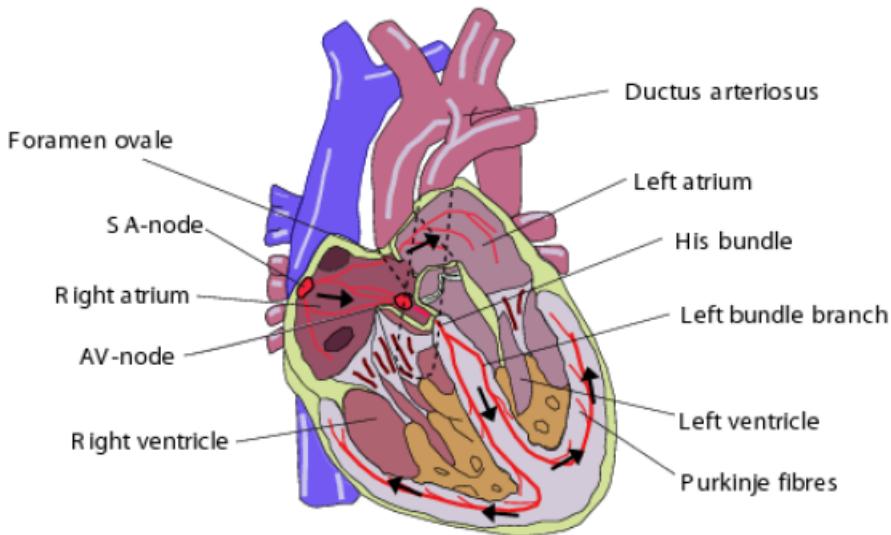


Figura 2.4: Anatomía del corazón fetal [22].

del mismo, y la adecuación de la entrega de sangre oxigenada a través de las arterias coronarias para conocer la demanda metabólica del miocardio. El corazón tiene cuatro cámaras cavitarias cuyas paredes consisten en un sincitio de células miocárdicas. A la salida de cada cámara hay una válvula que se cierra tras cada contracción, previniendo este el flujo retrógrado cuando la cámara realice la acción de relajación y las presiones que van hacia abajo excedan de las presiones de la cámara. El *corazón derecho* incluye una pequeña aurícula, lo que conduce a un mayor ventrículo derecho.<sup>1</sup> La aurícula derecha recibe sangre de la mayoría de partes del cuerpo y la suministra al ventrículo derecho. Cuando el ventrículo derecho se contrae, este impulsa debidamente la sangre a los pulmones, donde la sangre es oxigenada y liberada de dióxido de carbono.<sup>2</sup> La aurícula izquierda recibe sangre de los pulmones y la conduce hasta el ventrículo izquierdo.<sup>3</sup> Las fuertes contracciones del ventrículo izquierdo impulsan la sangre a través de la aorta hacia el resto del cuerpo, con la suficiente presión para llegar a irrigar incluso los cerebros de las personas más altas.<sup>4</sup> La aurícula izquierda y el ventrículo izquierdo forman el *corazón izquierdo*. Como se indicó anteriormente, en condiciones normales las aurículas terminan de contraerse antes de que los ventrículos comiencen a hacerlo.

La Figura 2.5 muestra la geometría de un corazón en condiciones normales de funcionamiento y los vectores eléctricos instantáneos resultantes del corazón a través del ciclo cardíaco. La figura muestra el origen de los latidos cardíacos (en el nodo SA), un retraso en el nodo AV (de modo que las aurículas, teóricamente, finalizan la contracción antes de que la comiencen los ventrículos), y la conducción acelerada de

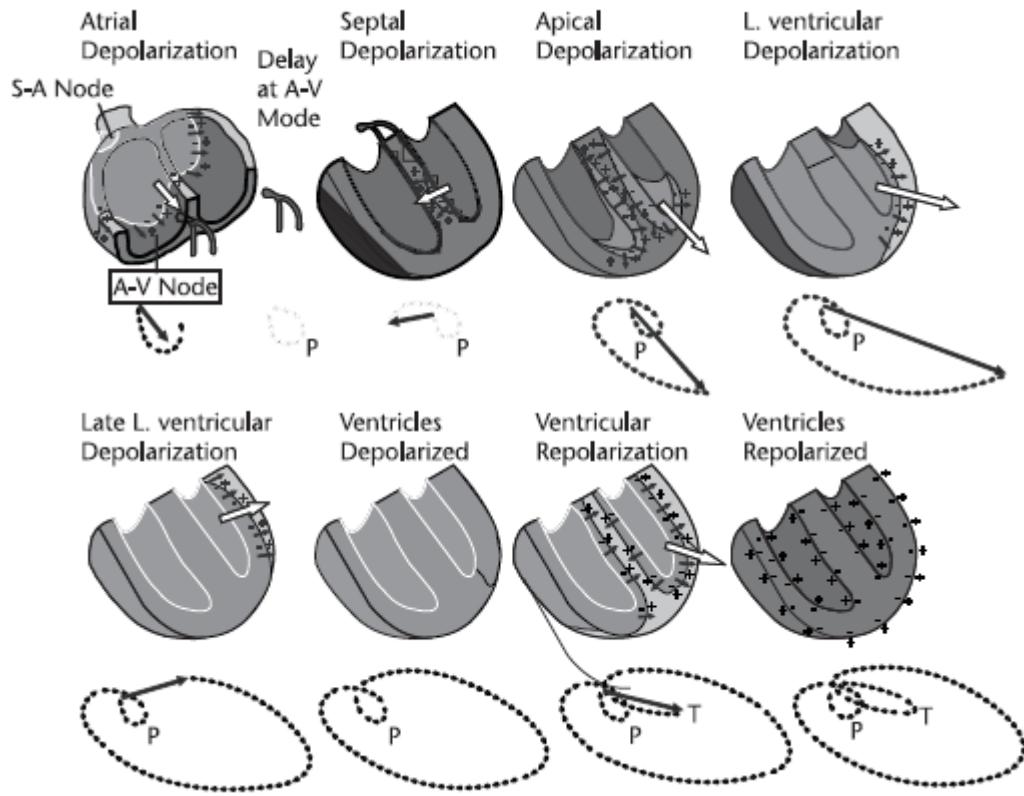


Figura 2.5: Trayectoria del vector cardíaco en condiciones normales de funcionamiento [18].

la onda de despolarización a través de las fibras conductoras especializadas (de modo que diversas partes del corazón son despolarizadas de una manera más sincronizada). Se muestran además, nueve estados temporales diferentes. La línea de puntos debajo de cada estado resume la trayectoria anterior de los vectores del corazón. Primariamente se muestra la *despolarización auricular*. A medida que desciende la onda de despolarización a través de ambas aurículas, el vector suma está apuntando hacia abajo (a los pies del sujeto) y a la izquierda del sujeto. A continuación, hay un *retraso en el nodo AV*, discutido anteriormente, durante ese tiempo no hay actividad eléctrica medible en la superficie humana a menos que se utilicen técnicas especiales de promediación. Después surge la actividad del nodo AV, la cuál despolariza al Fascículo atrioventricular,<sup>5</sup> seguido de una agrupación de las ramificaciones. El siguiente paso es la *despolarización septal*. El septum o tabique es la pared que se encuentra entre los ventrículos, y se dispone de una mayor agrupación de fibras conductoras a lo largo del lado izquierdo del tabique. Cuando la onda del potencial de acción entra, el tabique miocárdico tiende a propagarse de izquierda a derecha. Entonces, el vector del corazón resultante apunta hacia el lado derecho del sujeto. A continuación se encuentra

## 2. TEORÍA Y ESTADO DEL ARTE

---

la *despolarización apical*, y la onda de despolarización moviéndose a la izquierda se equilibra con la onda moviéndose hacia la derecha. El vector resultante apunta hacia el ápice del corazón, el cuál se encuentra apuntando hacia abajo, hacia la izquierda del sujeto. En la *despolarización ventricular izquierda*, también hay actividad eléctrica en el ventrículo derecho, pero debido a que el ventrículo izquierdo posee más masa, la actividad de este es la que domina. Después de las diversas partes de despolarización del miocardio, estas se contraen a través del proceso de excitación-contracción descrito anteriormente. Nos encontramos con que aparece un *Efecto Plateau* periódico durante la despolarización del miocardio (*ventrículos despolarizados*) donde no se propaga ningún potencial de acción, y por lo tanto, no hay vector cardíaco medible. Finalmente, las células individuales comienzan a re-polarizarse y otra onda de carga pasa a través del corazón, ahora originada desde los dipolos hasta la interfaz de *despolarización* y el *tejido re-polarizado*. Es entonces cuando el corazón vuelve a su estado inicial de reposo (en el cuál los ventrículos son re-polarizados), esperando a un estímulo eléctrico por el cuál comience el ciclo de nuevo. Tener en cuenta que la polaridad y la dirección de propagación de la fase de re-polarización se invierten respecto de las de depolarización. Como resultado, las ondas de re-polarización del ECG, generalmente son de la misma polaridad que las ondas de despolarización.

Para completar la revisión de los principios de la superficie del ECG, se describe a continuación una descripción de cómo la trayectoria del vector cardíaco resulta en un patrón escalar del ECG en condiciones normales. El vector cardíaco, el cual se expande, contrae, y rota en un espacio tridimensional, es proyectado en 12 líneas diferentes con una orientación bien definida. Cada derivación revela la magnitud del vector cardíaco en la dirección de cada rama en cada instante de tiempo. Las seis derivaciones precordiales informan de la actividad en el plano *horizontal*. En la práctica, esto requiere que se coloquen seis electrodos alrededor del torso. (Figura 2.6), y por lo tanto, el ECG representa la diferencia entre cada uno de estos electrodos (V1-6) y el *terminal central*.

Los electrodos adicionales, las derivaciones de las extremidades (*leads*), se colocan en cada una de las cuatro extremidades del sujeto, y el terminal central es la media de los potenciales de las extremidades. Las diferencias de potencial entre los electrodos de las extremidades y el terminal central son los fundamentos para otras tres derivaciones , tal y como aparecen (Figura 2.7):

1. **Derivación I**, la diferencia entre el brazo izquierdo (LA) y el brazo derecho (RA);

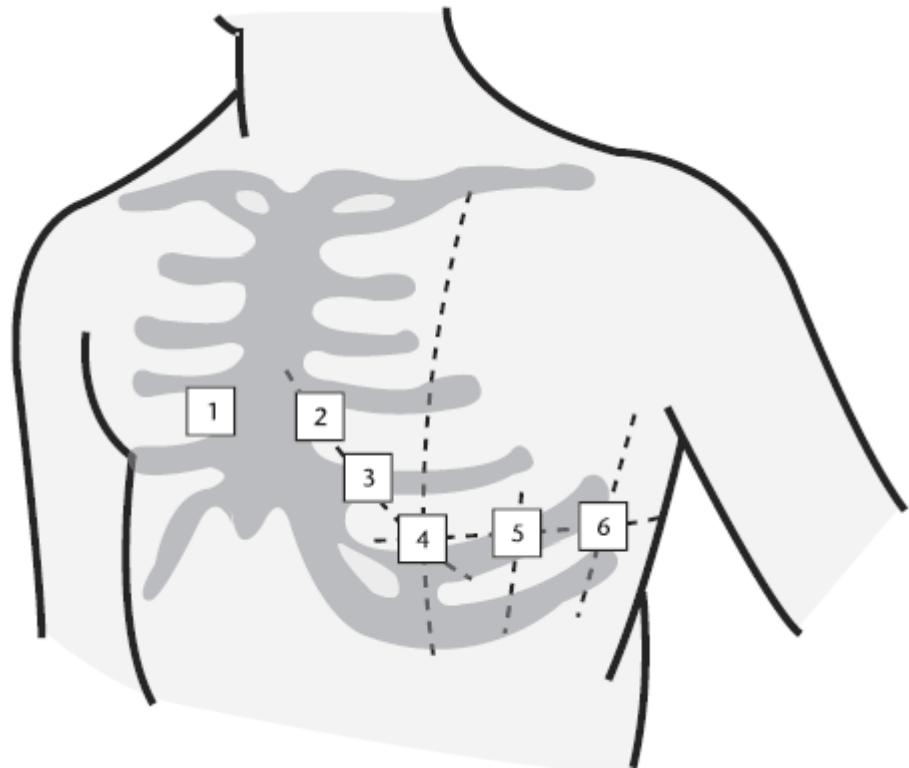


Figura 2.6: Colocación en el torso humano de los electrodos [18].

$$V_I = V_{LA} - V_{RA}$$

2. **Derivación II**, la diferencia entre la pierna izquierda (LL) y la RA;

$$V_{II} = V_{LL} - V_{RA}$$

3. **Derivación III**, la diferencia entre la LL y el LA.

$$V_{III} = V_{LL} - V_{LA}$$

Notar también que las derivaciones de las extremidades aumentadas (denotadas por “a”) representan el potencial de una derivación dada con respecto a la media de los potenciales de las otras dos derivaciones. Por lo que,  $V_F$  es la diferencia entre la LL y la media de las derivaciones del brazo;  $VR$  es la diferencia entre el RA y la media de la LL y la LA, y  $VL$  es la diferencia entre la LA y la media del RA y la LL.

$$aV_L = V_L - \frac{1}{2}(V_R + V_F)$$

$$aV_R = V_R - \frac{1}{2}(V_L + V_F)$$

## 2. TEORÍA Y ESTADO DEL ARTE

---

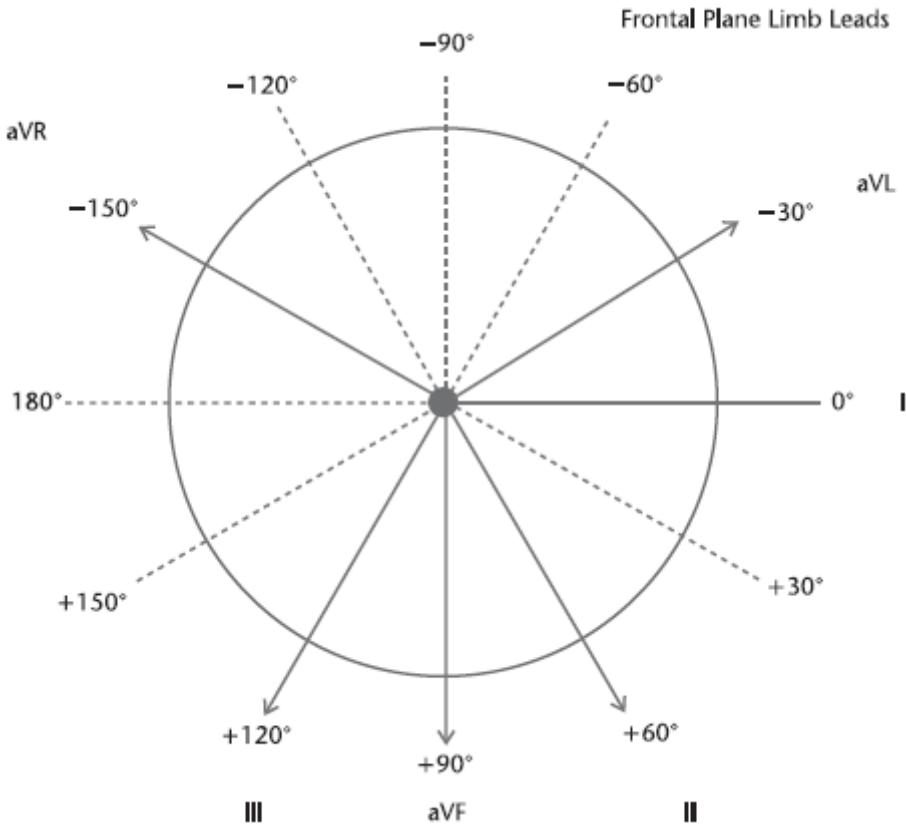


Figura 2.7: Colocación de electrodos representados en el plano frontal [18].

$$aV_F = V_F - \frac{1}{2}(V_L + V_R)$$

Hay 12 derivaciones para representar una actividad tridimensional, por lo que existe información redundante en este tipo de configuración. Sin embargo, este “sobremuestreo” parcial de proyectar el vector cardíaco en ejes no-ortogonales, tiende hacia una representación más fácil de la interpretación humana y compensa las inconsistencias que se puedan presentar a la hora de colocar los electrodos. (después de todo, las formas del cuerpo humano carecen de coherencia geométrica, por lo que la colocación de los electrodos varía en función del sujeto y de la técnica). Además, el cuerpo humano no es una esfera homogénea.

El patrón temporal del vector del corazón humano se combina con la geometría de las derivaciones de las extremidades en el plano frontal en la Figura 2.8. En negro se muestra la trayectoria temporal del vector del corazón. Las derivaciones ECG frontales están superpuestas en su orientación convencional (Figura 2.7). El patrón resultante para las tres derivaciones ECG de la superficie, I, II y III se muestran en la misma Figura. Notar que los ejes QRS son perpendiculares a la derivación isoeléctrica (la

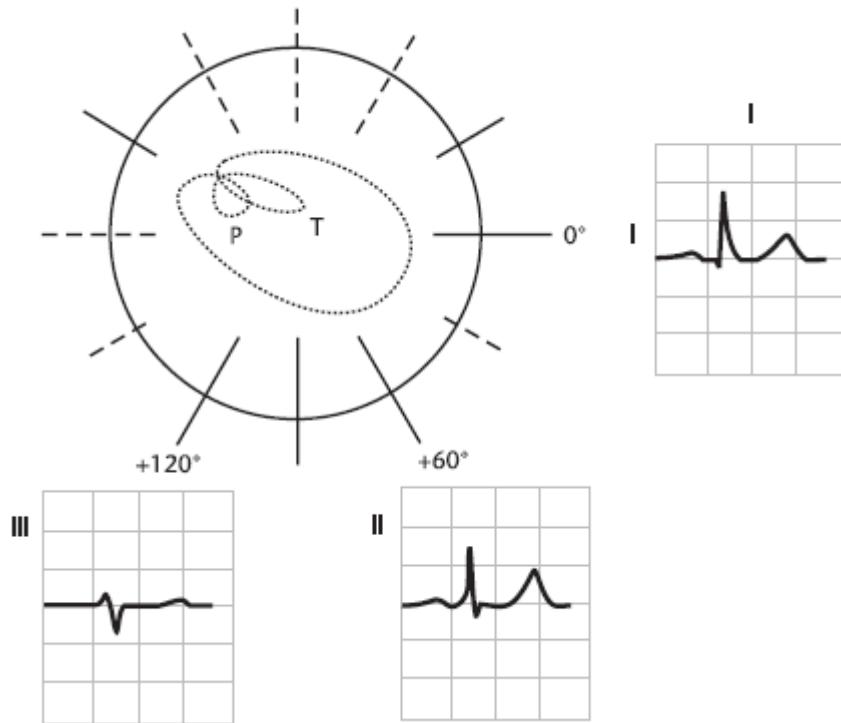


Figura 2.8: Patrón temporal del vector del corazón combinado con la geometría de los electrodos situados sobre el plano frontal [18].

derivación con fuerzas iguales en la dirección positiva y negativa). La aparición de cambios significantes en los ejes QRS pueden indicar problemas cardíacos.

La Figura 2.9 muestra las características clínicas de un ECG normal, las cuáles incluyen amplitudes de onda y tiempos de interondas. La localización de diferentes ondas en el ECG se encuentran señaladas arbitrariamente por las letras P, Q, R, S y T (y a veces U, a pesar de que esta onda es muy difícil de identificar, ya que puede estar ausente, tener una baja amplitud, o ser enmascarada por un latido subsecuente). El tiempo entre latidos (intervalo RR) no está señalado en la Figura. Los valores de las características clínicas indicadas en la Figura 2.9 son valores típicos, a pesar de que puedan variar en función del género, edad, actividad y estado de salud.

A continuación, introduciremos los Intervalos y segmentos más característicos del ECG, los cuáles resultan de un gran interés clínico.

**Intervalo RR:** intervalo entre dos complejos QRS consecutivos. Permite deducir directamente la frecuencia cardíaca instantánea ( $HR = 1/RR$ ). La frecuencia cardíaca media y su varianza son fundamentales para cualquier diagnóstico, y en especial para el caso de las taquicardias y bradicardias. El intervalo normal entre dos latidos consecutivos es de **0.83s** lo que supone una frecuencia cardíaca  $HR = 72 \text{ bpm}$ . Las

## 2. TEORÍA Y ESTADO DEL ARTE

---

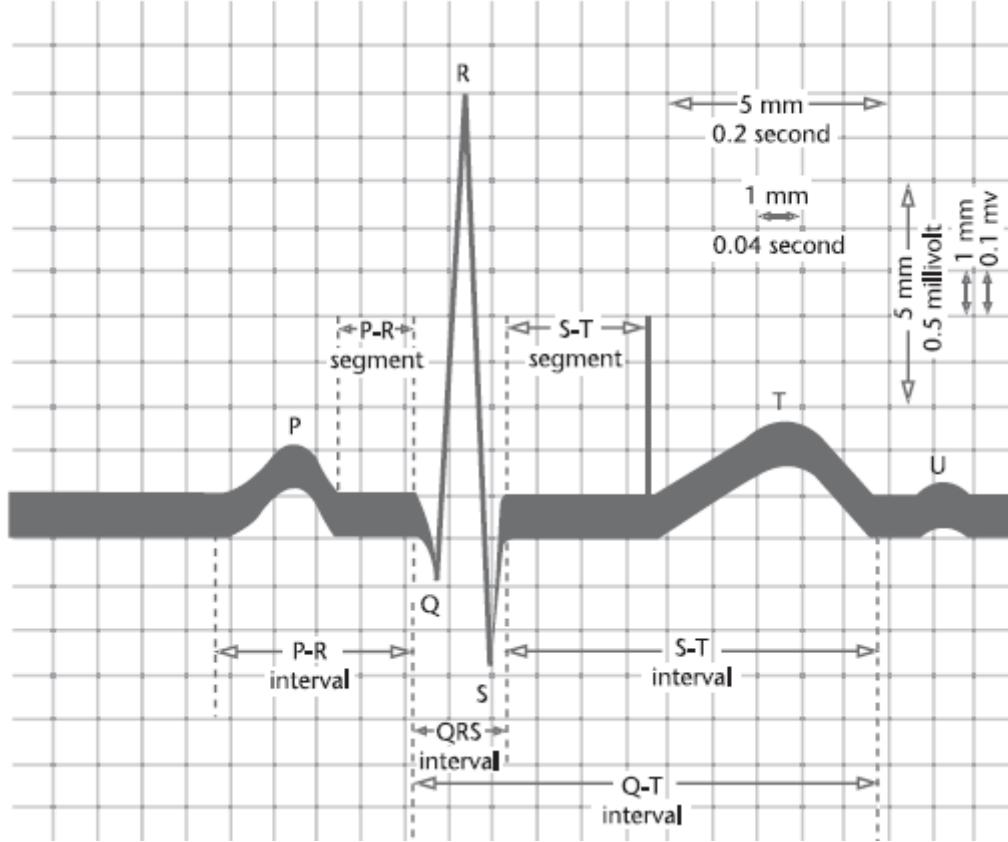


Figura 2.9: Representación gráfica y características normales del ECG [18].

unidades *bpm* hacen referencia a pulsos/minuto.

**Intervalo QT:** tiempo transcurrido entre el comienzo de la onda Q (o del complejo QRS) y el final de la onda T. En este intervalo, el ventrículo se excita eléctricamente y a continuación se contrae, por lo que esta medida permite evaluar la salud del mismo, siendo crucial en el diagnóstico de arritmias o de alteraciones electrolíticas como la hipocaldemia. La duración normal de este intervalo es de 0.35s.

**Segmento ST:** tiempo transcurrido entre el final de la onda S (final del complejo QRS) y comienzo de la onda T. Durante este intervalo, el ventrículo se contrae totalmente, expulsando la sangre hacia los tejidos. Este segmento se ve alterado en caso de isquemia, pero también en las primeras fases de infarto de miocardio, por lo que es útil para el diagnóstico en caso de angina. [19]

**Intervalo PR:** tiempo entre el comienzo de la onda P y el comienzo del complejo QRS. Se trata del intervalo entre el comienzo de la excitación eléctrica auricular y el comienzo de la excitación eléctrica ventricular. Permite evaluar el sistema eléctrico cardíaco, posibilitando el diagnóstico de bloqueos atrio-ventriculares. La duración

normal del intervalo PR es de 0.16s.

### 2.3. Electrocardiograma Fetal

La monitorización fetal hoy en día únicamente se basa en la extracción de la frecuencia cardíaca del feto (FHR), sin incluir características de la señal fetal, características que son el pilar básico de la evaluación cardíaca tanto materna como fetal. La razón principal de este suceso es la falta de tecnología disponible para poder distinguir exactamente donde se encuentran las señales maternas del ECG y las fetales del mismo, problema sobre el cuál se estructura este documento. Es decir, todavía no existe ninguna técnica de procesado de señal que sea capaz de extraer la señal fECG de los electrodos colocados en el abdomen materno sin distorsión, debido a la baja SNR de la señal fetal.

#### 2.3.1. Monitorización fetal

A demás de la electrocardiografía existen otras muchas técnicas para la adquisición de la FHR y de la señal proveniente del corazón fetal. De entre todas las técnicas de monitorización de la frecuencia cardíaca fetal existentes caben destacar:

- *Ecocardiografía*[25] , conocida como sonografía del corazón, la cuál se basa en técnicas de ultrasonidos.
- *Fonocardiografía*[7], la cuál no es más que el registro gráfico de los sonidos procedentes del corazón, producidos por las propias contracciones, transformados mediante cristales de cuarzo en ondas eléctricas respecto a la presión producida por los propios sonidos registrados.
- *Magnetocardiografía*[13], la cuál es la técnica que se encarga de medir el campo magnético de la señal cardíaca MCG (magnetocardiograma) utilizando dispositivos ultra sensibles como por ejemplo dispositivos SQUID (*Superconducting Quantum Interference Device*).

De todos estos métodos, la ecocardiografía es la técnica más utilizada para la monitorización de la FHR. Sin embargo, el ECG y el MCG contienen más información, debido a que la mayoría de los problemas cardíacos no se manifiestan únicamente mediante la FHR, por lo que necesitaremos fijarnos en los intervalos RR de tiempo. Se debe de tener en cuenta que debido a la similaridad morfológica del MCG y el ECG, los métodos de procesado de señal en ambos casos son idénticos; a pesar de

## **2. TEORÍA Y ESTADO DEL ARTE**

---

que la amplitud de las señales fetales, fECG, es mayor en el MCG que en el ECG debido al uso de dispositivos SQUIDs, los aparatos para la extracción del ECG son más simples en cuanto a electrónica, y por lo tanto más asequibles que los sistemas de extracción MCG.

Dentro del fECG existen dos modalidades, atendiendo a la forma en la que se adquiere la señal eléctrica del corazón fetal. Así es posible diferencias entre electrocardiograma fetal directo e indirecto, siendo el primero un método invasivo. A pesar de esto es el que ofrece una medición más fiable y estable.

Es por ello, que actualmente la aplicación de la electrocardiografía fetal únicamente se limita a la utilización de técnicas de monitorización invasivas, colocando la colocación de un electrodo en el escalpedo del feto durante el parto. A pesar de que actualmente esta es la única forma de medir completamente el fECG, únicamente se realiza bajo ciertas circunstancias clínicas, presentando un importante riesgo para la salud del feto e incluso de la madre.

Por el contrario, el fECG no-invasivo supone la adquisición y tratamiento de la señal mediante un método inocuo, indoloro y mucho más cómodo de realizar. En contraposición, las técnicas no-invasivas están sujetas a un mayor número de errores, debido a la menor amplitud de la señal registrada.

### **2.3.2. Electrocardiografía fetal**

El presente documento se centra en la utilización de técnicas no invasivas para la monitorización fetal, evitando así que tanto el feto como la madre sufran daños mediante el uso de técnicas de medida intrusivas. En concreto, debido a la simplicidad en cuanto a electrónica y al coste del instrumental necesario, nos centraremos en la monitorización fetal del ECG mediante electrodos en el abdomen materno (fECG), conocido como Electrocardiografía fetal

La Figura 2.10 muestra una señal fECG registrada mediante técnicas invasivas, como la medida mediante electrodos en el escalpedo (señal de color rojo), y técnicas no invasivas (señales inferiores, verde y amarilla), tomando la señal directamente desde el abdomen materno. Se han señalado cuatro latidos fetales (señalados como a, b, c y d) en las tres trazas. Como se ve en la Figura, las trazas provenientes del abdomen materno tienen menor amplitud que la tomada directamente del escalpedo, además de incluir más ruido y artefactos debido a la presencia del corazón materno. Tenga en cuenta que los artefactos se manifiestan en ambas trazas con una morfología similar a los latidos fetales, lo cuál dificulta la clasificación de estas señales.

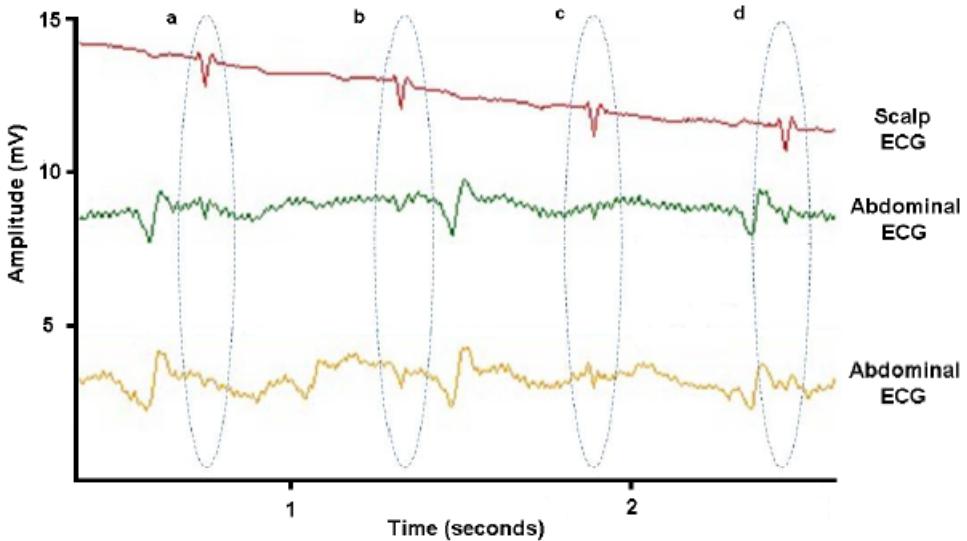


Figura 2.10: Muestra de una señal fECG tomada invasivamente (rojo) a través de un electrodo en el escalpido, y no-invasivamente (trazas inferiores) mediante electrodos en el abdomen materno.[20]

En la Figura 2.11 se ha comparado la amplitud y el rango de frecuencias de la señal fECG con otras señales biológicas y artefactos. Tal y como observamos, el fECG es mucho más débil que las demás señales biológicas. Además desde el punto de vista del procesamiento de señal, la señal fECG no posee un dominio específico (en el tiempo, frecuencia, espacio, etc.) en la cuál pueda ser totalmente separado del resto de señales, lo cuál dificulta su separación y tratamiento.

Además del resto de señales biológicas presentes en el cuerpo humano, hay que añadir la presencia de ruido, lo cuál dificulta aún más su extracción. Un ejemplo de ello, es la presencia de ruido electrónico, propias del proceso de adquisición de la señal. Ruidos tales como ruido de baja frecuencia (*lfNoise*), ruido dentro de la banda de frecuencias (*ibNoise*) y ruido de alta frecuencia (*hfNoise*). El *lfNoise*, también denominado *base wandering* (BW), es debido mayormente a los movimientos respiratorios y está situado en la banda de frecuencias que va desde los **0 Hz** hasta los **0.5 Hz**. El *hfNoise*, en cambio, incluye interferencias tanto biológicas como electrónicas, normalmente por encima de los 40 Hz. Finalmente el *ibNoise* comprende todas las interferencias coincidentes con el ancho de banda del fECG siendo por tanto estas las más difíciles de eliminar. [3]

La Figura 2.12 muestra el hecho de que la mayor parte de los latidos del corazón fetal en el electrocardiograma (el complejo QRS) reside en el mismo dominio de frecuencia que el complejo QRS del corazón adulto, así como del ancho de banda del

## 2. TEORÍA Y ESTADO DEL ARTE

---

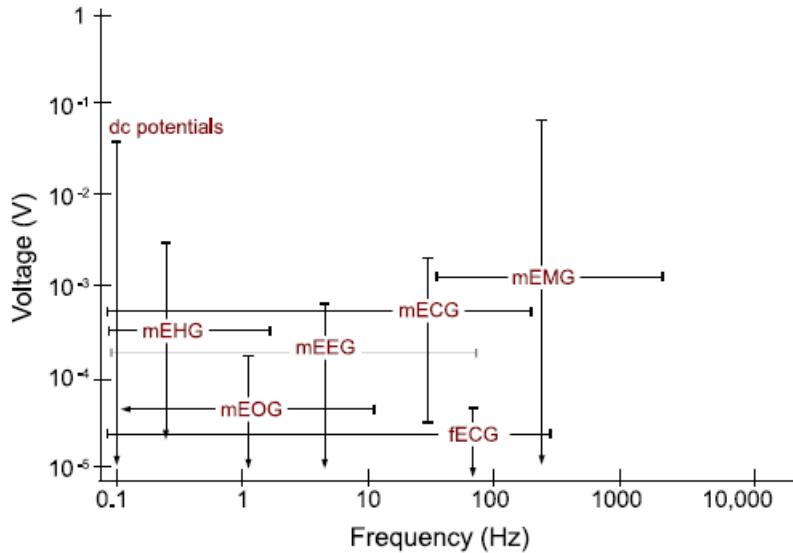


Figura 2.11: Rangos de amplitud y de frecuencia de las distintas señales biológicas, algunas de las cuáles interfieren con la señal cardiaca fetal. Las etiquetas en esta Figura hacen referencia al electrocardiograma materno (mECG), electroencefalograma (mEEG), electrooculograma (mEOG), electromiografía (mEMG), electrohistograma (mEHG), y electrocardiograma fetal (fECG). Tener en cuenta que la amplitud de las señales depende del lugar en el que se toman los datos.[20]

ruido muscular del mismo. La información del fECG se superpone con otras señales no deseadas en el tiempo, espacio, etc; por lo tanto, el uso de métodos que únicamente trabajen con un solo dominio del tiempo no es capaz de separar las señales fetales cardíacas completamente

Por lo tanto, desde el punto de vista matemático, la señal medida en el abdomen materno, sería la suma de tres componentes: el *fECG*, el mECG abdominal (*amECG*) y el ruido abdominal, (*aNoise*) (Figura 2.13):

$$aLead = fECG + amECG + aNoise$$

Tal y como observamos en la figura 2.11 la banda de frecuencia en la que se encuentra el fECG va desde los **0 Hz** hasta aproximadamente poco más de **100 Hz**. Mientras que la amplitud, la cuál depende del tiempo de gestación del feto, posición del mismo, posición en la cuál se coloca el feto, impedancia abdominal materna, etc. aunque no suele sobrepasar los **60  $\mu$ V**.

En la Figura 2.12 se puede observar como la señal materna, es la señal predominante en cualquier registro abdominal. Por lo tanto, tal y como se observa en la Figura 2.11 la banda de frecuencias es algo menor que la fetal, sobrepasando ligeramente los

## 2.4 Métodos de extracción de parámetros del fECG

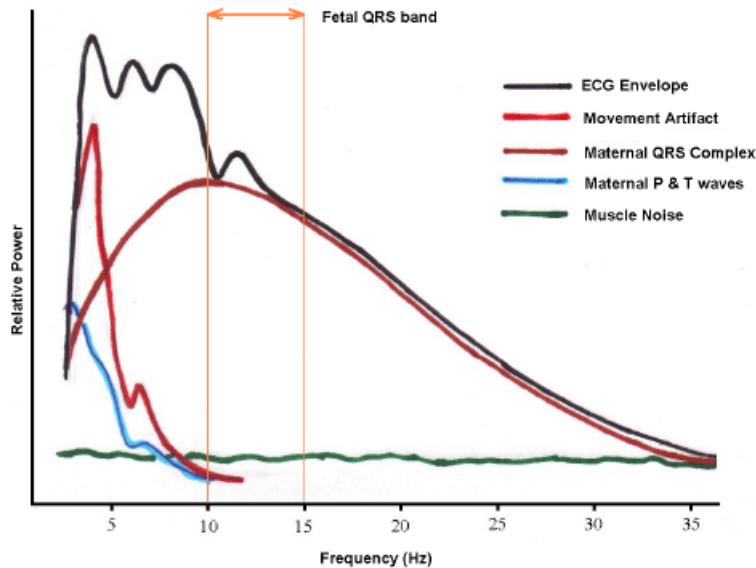


Figura 2.12: Representación general de la señal, artefactos y ruido presente en el ECG en el dominio de la frecuencia. La región en la cuál se manifiesta el complejo QRS del fECG está delimitada en la Figura. [20]

100 Hz, llegando en algunos casos hasta **150 Hz**. Por su parte la amplitud de la señal materna, puede variar entre **50  $\mu$ V** hasta llegar en algunos casos hasta **1 mV**. No obstante, el rango normal está comprendido entre **100  $\mu$ V** y **150  $\mu$ V**.

Todo estas dificultades hacen que la obtención de la señal fECG sea una tarea difícil, debido a la dificultad de separar y clasificarla independientemente de las demás señales biológicas y el ruido.

## 2.4. Métodos de extracción de parámetros del fECG

Para extraer correctamente los parámetros del fECG es necesario una etapa de acondicionamiento y filtrado de la señal biológica adquirida del vientre materno. Es por ello, que la adquisición, procesado, y representación de nuestra señal fECG se estructura en los siguientes bloques:

1. **Adquisición de señales:** El proceso empieza con la adquisición de la señal eléctrica proveniente del abdomen materno. Esta señal, como se ha comentado anteriormente, contendrá tanto la señal amECG como la señal fECG, así como ruido. Para adquirir la señal se utilizan unos electrodos colocados en el

## 2. TEORÍA Y ESTADO DEL ARTE

---

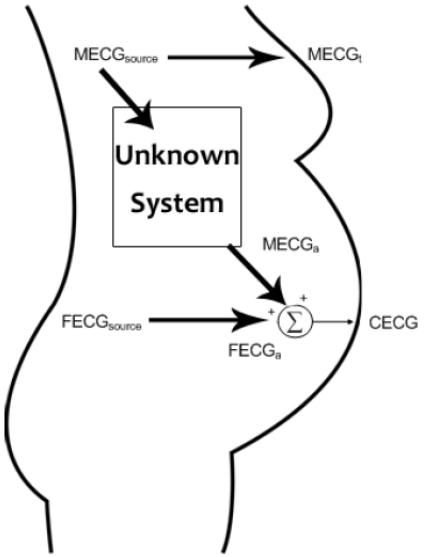


Figura 2.13: Representación del cuerpo materno, en el cuál se ve cómo la medida del ECG abdominal contiene la suma de la señal amECG y fECG, además de la presencia de ruido.

vientre materno, siendo los más utilizados los de Ag/AgCl por sus buenas características eléctricas y la facilidad en su fabricación.[15] Además, la electrónica de instrumentación actual permite una gran reducción en las interferencias de origen puramente electrónico, utilizando filtros analógicos para eliminar la interferencia de la línea de alimentación, la cual se encuentra entre 50 Hz y 60 Hz, así como filtros paso bajos (LPF) para rechazar frecuencias superiores a 100 Hz para evitar el hfNoise, y el *aliasing* a la salida del ADC.

2. **Preprocesamiento de la señal:** Una vez que disponemos de las señales digitalizadas, procedemos a la eliminación del ruido de las mismas, eliminando ruidos e interferencias presentes en ellas. En concreto en este proyecto se utiliza un filtrado *wavelet*.
3. **Detección de complejos mQRS y fQRS:** Una vez obtenida la señal aECG limpia de ruido, se procede a detectar los complejos maternos y fetales de la señal.
4. **Extracción de parámetros:** Finalmente, haciendo uso de la localización de los fQRS se realizan una serie de medidas, tales como la frecuencia cardíaca o la longitud del segmento ST. Estas medidas pueden llevarse a cabo por el propio médico si se dispone de una señal lo suficientemente limpia, o incluso pueden

## 2.4 Métodos de extracción de parámetros del fECG

---

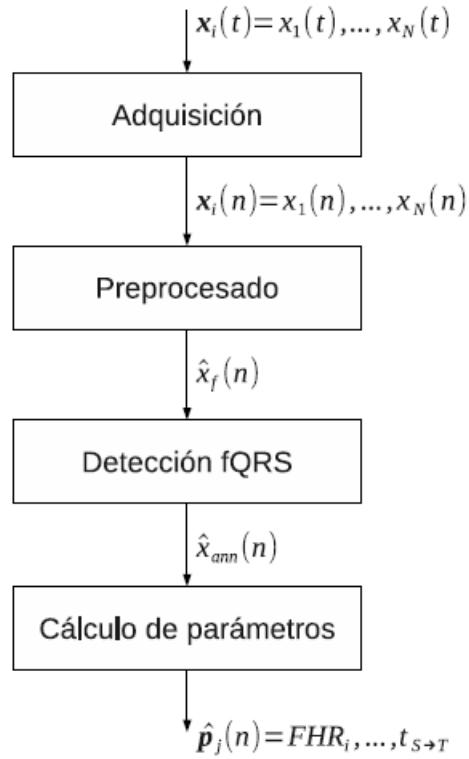


Figura 2.14: Etapas para la extracción de parámetros del fECG [8]

llegar a automatizarse, tal y como se procederá a realizar en el presente documento, evitando así la presencia de un *experto* para poder realizar las medidas en caso necesario.

## Parte II

### Trabajo realizado

# Capítulo 3

## Organización del proyecto

En el capítulo anterior se han visto los antecedentes y fundamentos teóricos sobre los que se basa el actual proyecto. En este capítulo se expone la organización y flujos de trabajo seguidos para la realización del trabajo. De forma global, el trabajo se ha organizado en dos grandes bloques, los cuáles se corresponden con la investigación de técnicas para la extracción de los complejos QRS fetales, y su posterior implementación en un soporte hardware.

### 3.1. Investigación sobre técnicas de extracción de complejos QRS fetales

La primera parte, y además la más importante del proyecto, se ha centrado en la investigación acerca de técnicas para la extracción de los complejos QRS fetales. Para ello, se ha investigado sobre diversos algoritmos que permitiesen mejorar los resultados obtenidos en [6] realizado por el grupo de investigación ECSens, de la Universidad de Granada.

Para la realización de esta parte de trabajo se ha utilizado una metodología típica de investigación, consistente en la revisión de trabajos científicos. Cabe destacar que en el presente proyecto los algoritmos estudiados no habían sido aplicados con anterioridad al campo ni a la temática de la extracción de complejos QRS fetales, por lo que se ha trabajado sobre una línea de investigación novedosa y partiendo desde cero, en cuanto a la clasificación y extracción de parámetros se refiere.

Tras la investigación inicial de los futuros algoritmos que mejorasen la eficiencia del algoritmo inicial, se procedió a la implementación de los mismos en software para poder comparar la eficiencia de los distintos algoritmos en función de las señales aECG a analizar. Implementándose una interfaz gráfica de usuario (GUI) que nos

### **3. ORGANIZACIÓN DEL PROYECTO**

---

permitiera extraer los complejos fQRS de forma automática en cada señal con los diversos algoritmos implementados en software.

La herramienta más relevante en esta etapa de simulación ha sido MATLAB, un lenguaje de cálculo técnico de alto nivel y un entorno interactivo para el desarrollo de algoritmos, la visualización de datos, el análisis de datos, y cálculo numérico. Se utiliza para aprendizaje automático, procesamiento de señales, procesamiento de imágenes, visión artificial, comunicaciones, finanzas computacionales, diseño de control, robótica y muchos otros campos.

Para la evaluación de la eficacia y fiabilidad de los distintos algoritmos estudiados, hemos elegido señales con una calidad aceptable en la medida y que han sido estudiadas por médicos y expertos en cardiología. Es por eso que se han utilizado señales biológicas procedentes de la base de datos de *PhysioNet*, las cuales contienen indicados los complejos fQRS detectados por cardiólogos.

#### **3.2. Implementación**

En esta etapa, el flujo de trabajo ha sido diferente al de la primera parte, debido a que en este proceso se ha tratado de implementar una solución específica en hardware.

Utilizando el sistema implementado inicialmente por el grupo ECSens, el cuál consiste en un sistema de adquisición de la señal, en este caso una FPAA (*Field-Programmable Analog Array*), para después realizar el procesado de señal en una FPGA (*Field Programmable Gate Array*). En el presente documento, se ha estudiado la alternativa de utilizar una PCB (*Printed Circuit Board*) específica para la adquisición de las señales en la parte analógica, en sustitución de la FPAA, debido a que únicamente se aprovecharía un bajo porcentaje de la funcionalidad y elementos de la FPAA, llevando esto a una ineficiencia económica en el actual proyecto.

Una vez adquirida la señal gracias a la PCB diseñada, se procedió a realizar el procesamiento y clasificación de la señal en una FPGA. La futura implementación de esta parte del proyecto se llevará a cabo utilizando lenguajes de descripción hardware (*Hardware Description Language*, HDL), en concreto, VHSIC *Hardware Description Language* (VHDL) y la herramienta de desarrollo *ISE Design Suite* del fabricante Xilinx. [27]

VHDL es un lenguaje de descripción hardware definido por el *Institute of Electrical and Electronics Engineers* (IEEE). VHDL es el acrónimo que representa la combinación de otros dos acrónimos *Very High Speed Integrated Circuit* (VHSIC) y HDL por lo que este lenguaje permite describir circuitos de gran complejidad. Para lograr este

### **3.2 Implementación**

---

fin, el diseño en VHDL se basa en una estructura modular y jerárquica. El circuito completo y cada uno de estos módulos puede diseñarse mediante una descripción tanto estructural, en el que se define la interconexión de las distintas funciones lógicas o componentes, como funcional, en la que se define la funcionalidad del circuito.

El diseño comenzará con la descripción en HDL de la solución que mejores prestaciones y precisión presente en la extracción de complejos fQRS. A continuación, se realizaría la síntesis del código, con lo que se conseguiría una nueva descripción en la que se emplean elementos de más bajo nivel, como funciones lógicas básicas, registros, etc.

En la futura etapa de implementación, los componentes lógicos que forman parte de esta nueva descripción deberán de ubicarse en la plataforma sobre la que se va a implementar nuestro proyecto, en este caso se utilizará una FPGA, en concreto una de la familia Spartan-6 de Xilinx, el modelo XC6SLX45-3FGG484, disponible en el departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada.

# Capítulo 4

## Planificación y estimación de costes

En el presente Capítulo se exponen los recursos, planificación y costes del presente proyecto. Se han tenido en cuenta todos los factores que han influido en mayor o menor medida en la realización del mismo.

### 4.1. Planificación

La distribución temporal de los cinco paquetes de trabajo pueden verse en el diagrama de Gantt del Cuadro 4.1. A continuación se procederá a enumerar estos paquetes, junto con las tareas que los componen:

#### 4.1.1. Investigación

##### 4.1.1.1. Revisión del estado del arte

En esta tarea se ha realizado un repaso de las soluciones existentes hasta el momento, totales o parciales, sobre el mismo ámbito de aplicación o relacionado con este. En nuestro caso, en las soluciones previas, no se encontraron soluciones totales al problema, sino soluciones parciales enfocadas a la clasificación de complejos QRS maternos, tarea menos compleja de realizar, que la separación entre maternos y fetales.

##### 4.1.1.2. Algoritmos de *Clustering*

Además, de la revisión del estado del arte de la extracción de los complejos fQRS, se realizó una revisión de las técnicas de *Clustering* existentes aplicadas a diversos campos (biomedicina, economía, minería de datos, etc).

### **4.1.2. Implementación *software***

#### **4.1.2.1. Implementación de los algoritmos de *Clustering* elegidos**

Una vez estudiados los diferentes algoritmos de *Clustering* existentes, decidimos implementar aquellos más representativos o de menor coste computacional en el lenguaje de programación MATLAB, para proceder a la extracción de los complejos fQRS.

#### **4.1.2.2. Creación de la GUI**

Una vez implementados los algoritmos de clasificación en MATLAB se procedió a implementar una GUI que nos facilitara la búsqueda del algoritmo que mejor precisión nos ofreciera, así como el de menor coste computacional.

### **4.1.3. Implementación *hardware***

#### **4.1.3.1. Diseño del sistema *hardware***

Se estudio la posibilidad de implementar nuestro sistema de detección de complejos fQRS en un sistema *hardware*, utilizando una PCB para la adquisición de la señal aECG, y una FPGA para el procesamiento y clasificación.

#### **4.1.3.2. Diseño de la PCB de instrumentación**

En esta tarea se realizó la versión inicial de la PCB encargada de adquirir la señal aECG a través de electrodos en el abdomen materno.

#### **4.1.3.3. Programación en VHDL**

Con nuestro sistema final diseñado, se procedería a la programación en VHDL del mismo para la utilización de la FPGA en la parte digital del sistema final.

### **4.1.4. Prototipado y testeo**

#### **4.1.4.1. Construcción del prototipo**

Diseñado el sistema final, y programado en VHDL, en esta tarea nos encargaremos de construir el prototipo final

## **4. PLANIFICACIÓN Y ESTIMACIÓN DE COSTES**

---

### **4.1.4.2. Testeo en tiempo real**

Por último, fabricado el prototipo final, se procedería al testeo del mismo. Para ello se probaría el sistema con pacientes reales, con diferentes semanas de gestación, para garantizarnos una correcta extracción de los parámetros fetales.

### **4.1.5. Documentación**

Como última tarea, se ha realizado la redacción de la documentación, una tarea llevada a cabo de forma paralelo al resto de tareas.

### **4.1.6. Diagrama de Gantt**

En el Cuadro 4.1 se expone el Diagrama de Gantt, una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo del tiempo total estimado de duración del proyecto. El Diagrama de Gantt no indica las relaciones existentes entre actividades.

Leyenda del cuadro 4.1:

- El símbolo ■ indica que el trabajo se ha realizado conforme a la planificación prevista inicialmente.
- Por otra parte, el símbolo ● indica la no realización de la actividad conforme a la planificación prevista inicialmente.

## **4.2. Recursos**

En el apartado de recursos se detallan a continuación todos aquellos que han sido de utilidad para la realización del proyecto, tanto recursos humanos como materiales.

### **4.2.1. Humanos**

- D.<sup>a</sup> Encarnación Castillo Morales, profesora del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada, en calidad de tutora del proyecto.
- D. Diego Pedro Morales Santos, profesor del Departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada, en calidad de co-tutor del proyecto.

|                                | N 15 | D 15 | E 16 | F 16 | M 16 | A 16 | M 16 | J 16 | J 16 |
|--------------------------------|------|------|------|------|------|------|------|------|------|
|                                | 15   | 30   | 15   | 30   | 15   | 30   | 15   | 30   | 15   |
| <b>Investigación</b>           |      |      |      |      |      |      |      |      |      |
| A)                             | ■    |      |      |      |      |      |      |      |      |
| B)                             | ■    | ■    | ■    | ■    | ■    | ■    | ■    | ■    |      |
| C)                             |      |      |      | ■    | ■    | ■    | ■    |      |      |
| D)                             |      |      |      |      | ■    | ■    | ■    | ■    |      |
| <b>Implementación Software</b> |      |      |      |      |      |      |      |      |      |
| E)                             |      |      |      |      | ■    |      |      |      |      |
| F)                             |      |      |      |      | ■    | ■    |      |      |      |
| G)                             |      |      |      |      |      | ●    | ●    | ●    |      |
| <b>Implementación hardware</b> |      |      |      |      |      |      |      |      |      |
| E)                             | ■    |      |      |      |      |      |      |      |      |
| F)                             | ■    |      |      |      |      |      |      |      |      |
| G)                             |      |      |      |      |      |      |      |      |      |
| <b>Prototipado y testeо</b>    |      |      |      |      |      |      |      |      |      |
| H)                             |      |      |      |      |      | ●    | ●    | ●    |      |
| I)                             |      |      |      |      |      |      |      |      |      |
| <b>Documentación</b>           |      |      |      |      |      |      |      |      |      |
| J)                             | ■    | ■    | ■    | ■    | ■    |      |      |      |      |
| K)                             | ■    | ■    | ■    | ■    | ■    |      |      |      |      |

Cuadro 4.1: Diagrama de Gantt del proyecto

## **4. PLANIFICACIÓN Y ESTIMACIÓN DE COSTES**

---

- Pedro Álvarez Guirado, alumno de la Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada, autor del proyecto.

### **4.2.2. *Hardware***

- Ordenador personal.
- FPGA Xilinx XC6SLX45-3FGG484.
- PCB de adquisición.

### **4.2.3. *Software***

- Sistema operativo Microsoft Windows 10.
- MATLAB y Simulink R2015b con la adquisición de Licencia para estudiante.
- Altium *Designer* v.16 con Licencia.
- Xilinx ISE *Design Suite* WebPACK con Licencia gratuita.
- Ofimática: LYX 2.1.0 y MiKTEX 2.9.

## **4.3. Estimación de costes**

### **4.3.1. Recursos humanos**

En el Cuadro 4.2 puede verse el coste temporal de los diferentes paquetes de trabajo. En total se computa en días, a razón de jornadas de 8 horas a 20 €/hora, precio por hora de un Ingeniero Técnico de Telecomunicaciones por el Colegio Oficial de Ingenieros de Telecomunicación (COIT).

### **4.3.2. Herramientas**

## **4.4. Presupuesto**

En el Cuadro 4.3 se registra el presupuesto total del proyecto, ascendiendo este a 16.979 €.

#### 4.4 Presupuesto

---

| Fase                         | Duración (días) |
|------------------------------|-----------------|
| Revisión del estado del arte | 10              |
| Búsqueda de algoritmos       | 15              |
| Implementación software      | 40              |
| Implementación hardware      | 5               |
| Documentación y Pruebas      | 15              |
| <b>TOTAL</b>                 | <b>85</b>       |

Cuadro 4.2: Coste temporal del proyecto.

| CONCEPTO   | CANTIDAD        |
|--|-----------------|
| <i>Recursos humanos</i>                                    |                 |
| Ingeniero Técnico de Telecomunicación x 680 h x 20 €/h     | 13.600 €        |
| <i>Herramientas</i>  |                 |
| Ordenador personal x 700 € unidad x 10 meses / 36 meses    | 195 €           |
| MATLAB® y Simulink R2015b                                  | 69 €            |
| Altium Designer v.16 x 7.200 € unidad x 2 meses / 12 meses | 1.200 €         |
| Xilinx ISE Design Suite WebPACK                            | 0 €             |
| Sistema operativo Microsoft® Windows 10                    | 120 €           |
| FPGA Xilinx XC6SLX45-3FGG484                               | 1.795 €         |
| <b>TOTAL</b>   | <b>16.979 €</b> |

Cuadro 4.3: Presupuesto.

# Capítulo 5

## Investigación

### 5.1. Planteamiento de partida

**Planteamiento del problema.** Tal y como vimos en la sección 2.4, la extracción de parámetros del fECG, y en general de cualquier señal biomédica, puede dividirse en tres grandes etapas: adquisición, disminución de ruido y detección de parámetros. Siguiendo este esquema, el presente trabajo se centra principalmente en la tercera etapa, esto es, la detección de los parámetros fetales en la señal aECG. El objetivo principal es, por lo tanto, obtener los parámetros más característicos de la señal aECG, es decir, separar correctamente aquellos parámetros que pertenecen a la señal fECG de los parámetros pertenecientes a la señal mECG y del ruido. En concreto, puesto que la FHR se calcula directamente a partir del intervalo RR fetal ( $FHR = \frac{1}{f_{RR}}$ ), nos centraremos en la extracción de las posiciones de los fQRS ( $FHR(k) = t_{fQRS(k)} - t_{fQRS(k-1)}$ ). Para lograr este fin, se investigará acerca de posibles algoritmos que nos permitan mejorar el trabajo presentado por [6], del cuál parte este trabajo.

Para ello, primeramente vamos a presentar el sistema inicial de partida del presente proyecto.

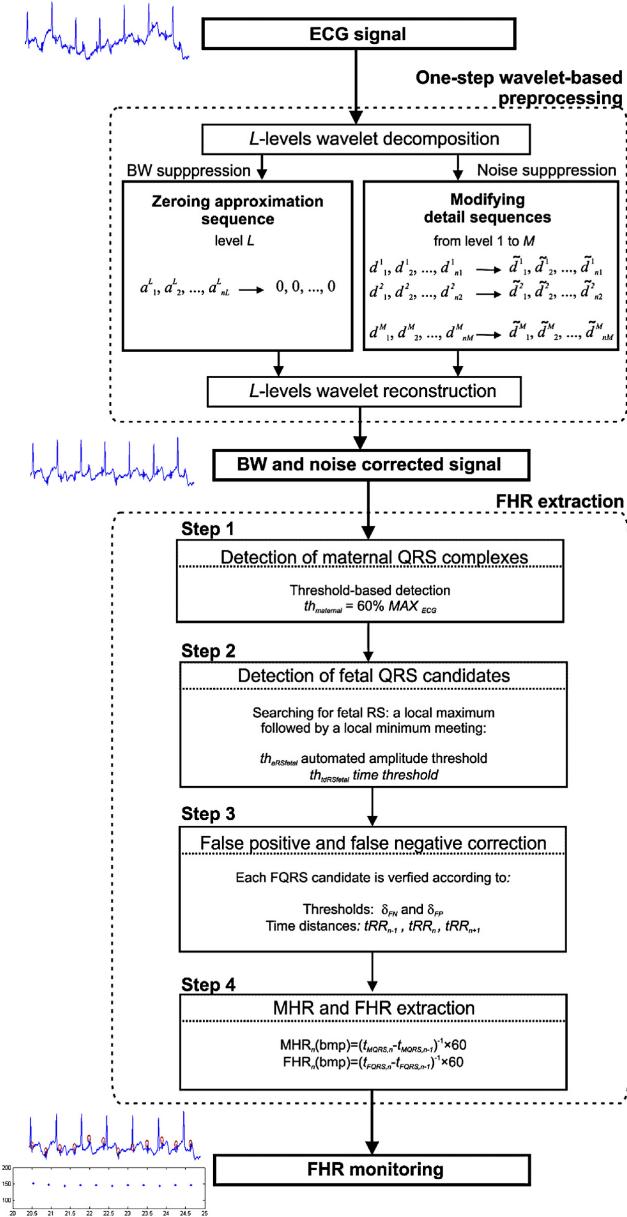


Figura 5.1: Diagrama de flujo del método inicial de partida. [6]

El método inicial propuesto para la monitorización del FHR está compuesto por dos etapas. Estas etapas son:

### 5.1.1. Preprocesado *wavelet* de un único paso

La primera etapa se centra en el *preprocesado* basado en la *transformada wavelet* (WT) de un solo paso, para la corrección del BW y la eliminación de ruido. El preprocesado de la WT elimina el BW y el ruido utilizando únicamente una descomposición *wavelet* y una estructura de reconstrucción de la señal. Esto nos permite implementar

## 5. INVESTIGACIÓN

---

el algoritmo en una estructura hardware optimizada, que ahorra recursos y tiempo de procesamiento, permitiendo así aplicaciones en tiempo real. Además, el método de medida consiste en la utilización de un único electrodo, colocado en el abdomen materno, sin necesidad de separar la señal fECG de la aECG, así que los complejos QRS fetales son directamente extraídos del preprocesamiento de la señal aECG. Este método optimiza al máximo los recursos hardware, evitando la utilización de procesado adicional utilizado en otra técnicas. Adicionalmente, el método propuesto no solo permite identificar los complejos QRS fetales, sino también los maternos (mQRS), permitiendo así monitorizar el FHR y el MHR.

Tal y como se ha detallado anteriormente, una descomposición *wavelet* adecuada puede utilizarse para la supresión del BW y del ruido. Además, debido a la estructura *wavelet* similar para ambas aplicaciones, BW y ruido, se propone la aplicación de ambas técnicas en un único paso. Gracias a esto, mejoraremos la eficiencia sin empeorar la precisión, ahorrando recursos y facilitando futuras implementaciones hardware. Los pasos requeridos para la aplicación de esta propuesta se detallan a continuación:

- **Descomposición:** la descomposición *wavelet* se aplica a un cierto nivel,  $L$ , para producir los coeficientes de aproximación  $a_n^{(L)}$  que capturan el BW.
- **Reducción a cero de las aproximaciones:** la aproximación  $a_n^{(L)}$  es sustituida por un vector de todo-ceros.
- **Umbral:** el nivel  $M$  (con  $M < L$ ) permite distinguir correctamente la presencia de cambios parciales en el ruido. Adicionalmente, por cada nivel desde  $i = 1$  hasta  $M$ , se aplican los umbrales límites apropiados a los coeficientes de detalle,  $d_n(i)$  para eliminar mejor el ruido.
- **Reconstrucción:** la reconstrucción *wavelet* basada en la reducción a cero de los coeficientes de aproximación de nivel  $L$ , se realiza para obtener el BW correcto y eliminar el ruido presente en la señal.

En consecuencia, la supresión de BW y de ruido es fácil de conseguir utilizando esta técnica basada en la WT, la cuál requiere seleccionar los parámetros apropiados. Se requiere de una selección especial de algunos parámetros para señales muy ruidosas o señales que contienen fECG con poca potencia.

### 5.1.1.1. Transformada *wavelet*

La selección de la WT se basa en las similitudes entre la estructura básica del ECG y la función *wavelet*, así como en el tipo de procesado aplicado. Para procesado de señal, los tipos de *wavelet* más interesantes son los ortogonales, entre los que destacan los algoritmos Haar, Daubechies, Coiflets, Symmlets y Biortogonal [23]. La *wavelet* ortogonal no es redundante y es adecuada para la compresión y eliminación de ruido en señales o imágenes. Además, poseen una fase lineal, y están adecuadas para la extracción de parámetros y características. En concreto, el método utilizada la familia *wavelet Daubechies 6* (db6), por su semejanza con la forma de onda del ECG y por haber demostrado buenos resultados en su implementación.

### 5.1.1.2. Niveles de descomposición $L$ y $M$

El nivel de descomposición para la supresión de BW,  $L$ , debería de capturar el BW. Tal y como se ha comentado, para eliminar el BW es necesario seleccionar un nivel de resolución *wavelet* que capture los componentes del ECG para frecuencias menores de 1 Hz. Por lo tanto, el método comienza con la descomposición de la señal abdominal por medio de una transformada *wavelet* en  $L$  niveles. Los coeficientes de aproximación del  $L$ -ésimo nivel,  $a_n^{(L)}$ , presentan las componentes de menor frecuencia de la señal analizada. El valor de  $L$  será elegido de forma que los coeficientes  $a_n^{(L)}$  recojan la información correspondiente a frecuencias inferiores a **1 Hz**, esto es:

$$1 \text{ Hz} = \frac{F_0}{2^L} \quad (5.1)$$

$$L = \lceil \log_2(F_0) \rceil \quad (5.2)$$

Donde  $F_0$  es la componente de máxima frecuencia de la señal, “normalmente” la mitad de la frecuencia de muestreo (si se ha procesado la señal analógica habrá que considerar la frecuencia de corte de este filtrado). De este modo, sustituir los coeficientes  $a_n^{(L)}$  por cero equivale a eliminar toda la información por debajo de 1 Hz, y, por tanto, supone la eliminación del ruido de baja frecuencia lfNoise o *base wandering*. El nivel máximo de umbral (*threshold*),  $M$ , depende de varios factores, como la SNR de la señal original o la frecuencia de muestreo. La presencia de ruido es mayor a altas frecuencias, aunque la mayoría de la energía espectral se encuentra en las sub-bandas de baja frecuencia. Por lo que, para evitar la perdida de componentes de la señal, como pueden ser la morfología PQRST, solo se tratarán las bandas de alta frecuencia para la eliminación del ruido.

## 5. INVESTIGACIÓN

---

Por lo tanto, la supresión de ruido de alta frecuencia *hfNoise* se consigue eliminando parte de la información de los  $M$  primeros niveles de descomposición, correspondientes con altas frecuencias. Para ello, se aplica una función matemática adecuada a los  $M$  primeros coeficientes de detalle  $d_n^{(i)}$ , siendo  $i = 1, \dots, M$  y  $M < L$ . Esta función anulará los coeficientes con un valor cercano a cero, respetando aquellos coeficientes de mayor amplitud. En concreto, los coeficientes de detalle menores que un determinado umbral  $d_n^{(i)} < T_h$  se igualan a cero, mientras que los coeficientes mayores a este umbral se mantienen intactos (*hard thresholding*,  $\tilde{d}_n^{(i)}$ ).

$$\tilde{d}_n^{(i)} = \begin{cases} d_n^{(i)} & \text{si } |d_n^{(i)}| \geq th_{i,r} \\ 0 & \text{si } |d_n^{(i)}| < th_{i,r} \end{cases} \quad (5.3)$$

o se modifican ligeramente para evitar discontinuidades bruscas (*soft thresholding*,  $\hat{d}_n^{(i)}$ ).

$$\hat{d}_n^{(i)} = \begin{cases} sign(d_n^{(i)})(|d_n^{(i)}| - th_{i,r}) & \text{si } |d_n^{(i)}| \geq th_{i,r} \\ 0 & \text{si } |d_n^{(i)}| < th_{i,r} \end{cases} \quad (5.4)$$

Con la aplicación de estos umbrales se consigue eliminar los rizados presentes en la señal, considerándose estos ruido. De entre los varios tipos de umbrales existentes destacamos los dos siguientes, por ser los que han demostrado mejores resultados en la eliminación del ruido *hfNoise* del aECG. Se trata de los umbrales *exponencial*, para el caso de *soft thresholding*, y *minimax*, usado con *hard thresholding*. Ambos umbrales han sido obtenidos de las definiciones de umbrales de la *teoría universal de Donoho*. [5]

$$Th_{exp} = 2^{\frac{i-M}{2}} \sqrt{2\log N} \quad (5.5)$$

$$Th_{minimax} = 0,3936 + 0,1829 \left( \frac{\log n_i}{\log 2} \right) \quad (5.6)$$

donde  $n_i$  representa el número de coeficientes  $d_n^{(i)}$  para cada nivel  $i$ , y  $N$  es la longitud total de la señal abdominal.

Tras la descomposición de la señal abdominal y la modificación adecuada de los coeficientes correspondientes, la señal abdominal es finalmente reconstruida aplicando la transformada *wavelet* inversa a estos nuevos coeficientes. El resultado de la aplicación de este método a una aECG es la eliminación eficaz de los ruidos de baja y alta frecuencia, conservando las características principales de las señales mECG abdominal y fECG.

### 5.1.2. Extracción y detección de características de la señal aECG

Después del procesado, la señal resultante es adecuada para la extracción del FHR. Las fases requeridas para la extracción se detallan a continuación:

#### 5.1.2.1. Detección de los complejos QRS maternos

En primer lugar se detecta el máximo de la señal procesada,  $MAX_{ECG}$ . Debido a que la señal está libre de BW, este valor puede identificarse como el valor máximo de los complejos mQRS (concretamente, la amplitud mayor del pico R materno). Entonces, se calcula un umbral como un cierto porcentaje del máximo,  $th_{maternal}$ , por lo que todos los puntos por encima de este umbral, pertenecerán a los complejos QRS maternos (mQRS).

#### 5.1.2.2. Detección de los complejos QRS fetales

Conociendo la morfología de los complejos QRS fetales, se realiza un escaneo entre cada dos complejos mQRS detectados buscando un máximo local, seguido de un mínimo local (máx-mín), denominado candidato a pico fRS. Para convertir cada candidato a pico fRS a un candidato a complejo fQRS es necesario que se cumplan dos requerimientos establecidos por la media de los dos umbrales.

El primer umbral es la distancia en amplitud entre los dos puntos que son candidatos a pico fRS, denominado  $th_{aRSfetal}$ . El valor óptimo de este umbral depende de varios parámetros como las semanas de gestación, la posición del feto, la posición del electrodo o el ruido que afecta a la señal aECG. Trabajando con diferentes bases de datos públicas, se ha encontrado que no es posible establecer un valor fijo para estos umbrales, incluso cuando los datos son normalizados, debido a que depende de los parámetros mencionados.

El segundo umbral, denominado  $th_{dRSfetal}$ , es la distancia de tiempos entre dos candidatos a picos fRS los cuales dependen principalmente del FHR. El valor propuesto para este umbral es la distancia de tiempos entre el intervalo R-S fetal, considerado como el límite para el comienzo de una baja FHR. Además, utilizando este umbral de tiempos se nos asegura la detección de FHR altos y en condiciones normales. Los candidatos a picos RS que cumplen estos dos umbrales serán almacenados como futuros candidatos a fQRS.

## 5. INVESTIGACIÓN

---

### 5.1.2.3. Corrección de falsos positivos y falsos negativos

En este paso, se ha llevado a cabo la detección de FP y de FN. Un FN es un complejo fQRS no detectado, mientras que un FP es un complejo fQRS inexistente, detectado. Los FP y FN son el resultado de diversas situaciones, como por ejemplo el solapamiento de complejos fQRS y mQRS, la baja energía de señales fQRS, un procesado no adecuado o un valor erróneo para los umbrales anteriormente descritos.

A pesar de esto, este algoritmo considera la posibilidad de un incremento del FHR repentino (o decremento) durante la captura de la señal aECG, esto nos asegura que estas situaciones no se confundan con FP (o FN). Para un candidato a fQRS dado en la iteración  $n$ , las distancias de tiempo RR son definidas como:

$$tRR_m = t_{detectedfRm} - t_{detectedfRm-1} \quad (5.7)$$

donde los valores considerados para  $m$  son  $n-1$ ,  $n$ ,  $n+1$ , y  $n+2$ . En consecuencia, para encontrar los FN y los FP o para detectar las variaciones en el FHR, estas distancias  $tRR$  son evaluadas de la siguiente manera:

$$\delta_{FP} \cdot tRR_{n-1} < tRR_n < \delta_{FN} \cdot tRR_{n-1} \quad (5.8)$$

$$tRR_n > \delta_{FN} \cdot tRR_{n-1} \quad (5.9)$$

$$tRR_n < \delta_{FP} \cdot tRR_{n-1} \quad (5.10)$$

donde  $\delta_{FN}$  y  $\delta_{FP}$  son los umbrales que determinan el rango de FN y FP.

Si se cumple la ecuación 5.8, el candidato a fQRS se convierte en un fQRS detectado y subsecuentemente se calcula la FHR. Si no, se evalúa la posibilidad de que sea un FP o FN. Para encontrar un FN, es necesario explorar la posibilidad de que haya un fQRS no detectado entre el candidato fQRS y el fQRS detectado anteriormente. Si se cumple la ecuación 5.9, se comprueba si existe un FN o el ritmo cardíaco fetal ha disminuido. Esto se evalúa comprobando las comparaciones  $tRR_n > \delta_{FN} \cdot tRR_{n+1}$  y  $tRR_n > \delta_{FN} \cdot tRR_{n+2}$ . A pesar de que  $tRR_{n+1}$  y  $tRR_{n+2}$  son calculados a partir de los fQRS que todavía no han sido procesados y verificados por la etapa 3, si estas condiciones se cumplen debido a la eficiencia de la etapa 2, es muy probable que los pulsos fetales permanezcan a un ritmo similar, por lo que se evalúa la posibilidad de un posible candidato a FN. Entonces se comprueba la existencia de un complejo mQRS entre el candidato fQRS y el fQRS detectado anteriormente. Por lo que se evalúan posibles

fQRS que se hayan superpuesto con los mQRS. Si estas comparaciones no se cumplen, se detecta un cambio en el FHR, un decremento, y no existen FN. Se ha llevado a cabo un procedimiento similar para la detección de los FPs. Si se cumple la ecuación 5.10, así como las comparaciones  $tRR_n < \delta_{FN} \cdot tRR_{n+1}$  y  $tRR_n < \delta_{FN} \cdot tRR_{n+2}$  existe un FP y el candidato a fQRS se elimina. Sin embargo, si se satisface la ecuación 5.10 pero alguna de las comparaciones anteriores no se cumple, se considera que el FHR ha aumentado.

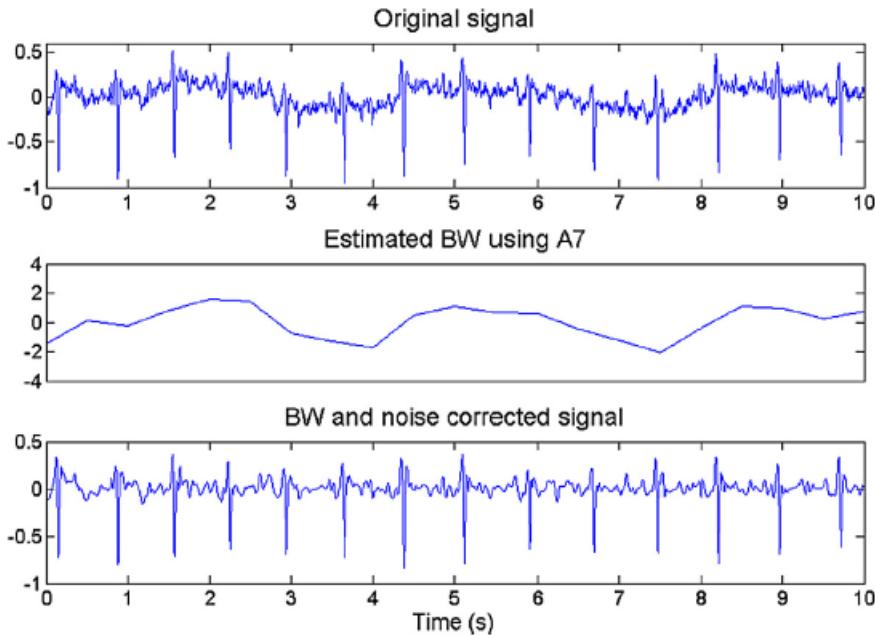


Figura 5.2: Señal de la base de datos *Daisy*, con la estimación de BW y la corrección del ruido de la señal utilizando el método propuesto en [6].

#### 5.1.2.4. Extracción de la FHR y MHR

El último paso consiste en la extracción de la MHR y la FHR, obtenidas respectivamente de las distancias entre cada par de complejos mQRS y fQRS detectados.

**Evaluación de la extracción de fQRS.** En el último término, la evaluación de cualquier técnica requiere de unas medidas de bondad, sin las cuales es imposible compararlo con otros métodos. Normalmente utilizaríamos la relación señal a ruido (SNR), pero para medirla necesitamos disponer de una señal original y de una señal con presencia de ruido, por lo que ninguna de las señales disponibles en las bases de datos pueden utilizarse para evaluar la SNR. Por otra parte, generar las señales de

## 5. INVESTIGACIÓN

---

forma artificial supone simular un sistema complejo, y desconocido en parte, falseando en cualquier caso los resultados.

Tendremos que recurrir a otras mediciones para evaluar la eficacia del método. Para ello, recurriremos a parámetros como la sensibilidad,  $S_e$ , *positive diagnostic value*,  $PDV$ , y precisión,  $Acc$ , presentadas en [6].

$$S_e = \frac{TD}{TD + FN} \quad (5.11)$$

$$PDV = \frac{TD}{TD + FP} \quad (5.12)$$

$$Acc = \frac{TD}{TD + FP + FN} \quad (5.13)$$

Como podemos observar en la ecuación 5.11, la sensibilidad está relacionada directamente con los falsos negativos y define la probabilidad de detectar un pulso fetal (fQRS), mientras que la  $PDV$  proporciona la probabilidad de que un fQRS detectado sea realmente un pulso fetal y no un falso positivo. La precisión es una medida que resume la eficacia o bondad de un método. En la figura 5.3 se muestra un ejemplo de estas mediciones.

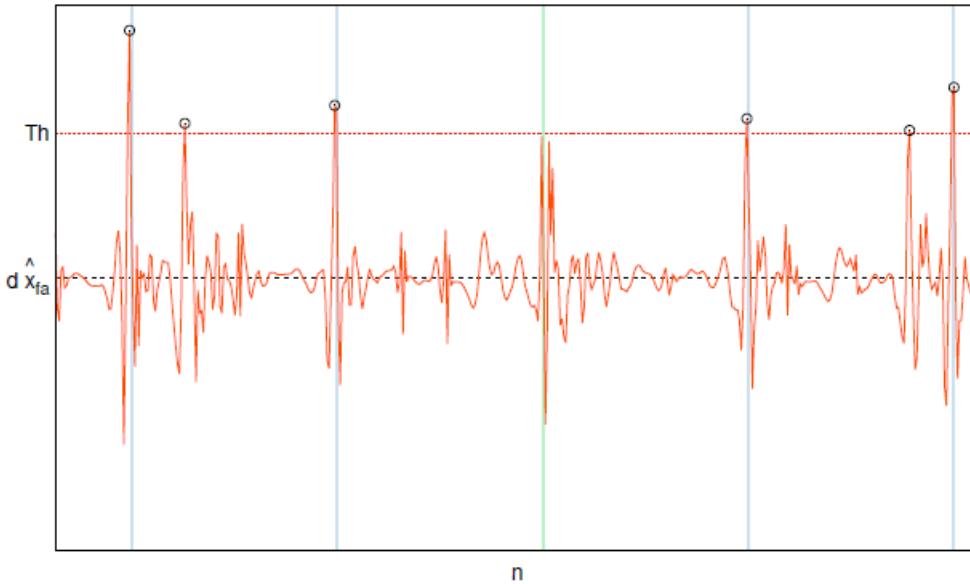


Figura 5.3: Detección de fQRS sobre la derivada, con presencia de falsos positivos y negativos.

## 5.2. Trabajo a realizar

A continuación se exponen las principales líneas sobre las que se ha trabajado en el presente proyecto. Nuestro objetivo es, una vez realizado el preprocesado de la señal, es decir, una vez que la señal ha sido filtrada por la *transformada wavelet*, detectar todos los complejos QRS fetales y maternos de una manera correcta, aprovechando al máximo la eficacia de la *transformada wavelet*, llegando incluso no solo a mostrar la FHR, sino también mostrar la señal fECG en la que el *especialista clínico o médico* pueda extraer conclusiones sobre futuras anomalías del paciente. En el caso de automatizarse completamente a detectarse complejidades de forma automática, sin la necesidad de recurrir a un especialista, pudiendo implementarse un producto *eHealth (electronics-Health)* económico, de pequeñas dimensiones y con una gran fiabilidad en los resultados.

Para ello, los pasos seguidos han sido los siguientes:

1. **Investigación sobre posibles mejoras en la extracción de fECG.** Debido a que nuestro anterior método de elección de complejos fQRS y mQRS necesita de la introducción de los valores umbrales, necesitando particularizar estos valores para las señales en la detección de complejos fQRS, se ha investigado sobre diferentes técnicas de clasificación de datos, para proceder a la extracción de las principales características de la señal sin necesidad de introducir umbrales ni semillas particulares. Utilizaremos en su lugar, algoritmos con inicializaciones completamente aleatorias.
2. **Implementación de las posibles soluciones al problema inicial y configuración de los parámetros óptimos para su aplicación.** Una vez estudiadas las posibles soluciones a nuestro problema inicial, se procedió a implementar los diferentes algoritmos que solucionaran este problema.
3. **Comparación entre todas las soluciones, probando con diferentes tipos de señales y de bases de datos para encontrar el algoritmo que mejor se adapte a las señales aECG.** Una vez teníamos los diferentes algoritmos implementados y listos para clasificar los complejos QRS de las señales aECG, se procedió a la comparación de los mismos, bajo diferentes bases de datos, señales, tamaño de ventanas, parámetros, etc. para encontrar de manera estadística el algoritmo que mejor se adaptase a las señales aECG.

## 5. INVESTIGACIÓN

---

**4. Implementación de un sistema en tiempo real con el algoritmo que mayores prestaciones ofrezca.** En este apartado se ha implementado una GUI que permita extraer los complejos fQRS y mQRS de nuestras señales en tiempo real, pudiendo incorporarse este sistema al sistema sanitario, ahorrando costes al mismo, y mejorando las prestaciones de los actuales sistemas de adquisición de señales ECG.

### 5.3. El problema de la utilización de umbrales

Nuestra solución inicial, la detección de complejos fQRS y mQRS mediante la utilización de umbrales conllevaba a diversos problemas. El método del valor umbral es un algoritmo cuya finalidad es segmentar una señal, es decir, separar partes de la misma, en función del foco de atención sobre el que nos centremos. Básicamente, podemos decidir que máximos locales pertenecen a los complejos QRS fetales, cuáles a los maternos, y cuáles son considerados ruido de la señal, gracias a la utilización de umbrales, los cuales nos ayudan a decidir a qué grupo pertenecerá cada señal.

Matemáticamente, podemos expresarlo como:

$$maternal_{QRS} = \begin{cases} \tilde{maternal}_{QRS} & \tilde{maternal}_{QRS} \geq th_{maternal} \\ 0 & \tilde{maternal}_{QRS} < th_{maternal} \end{cases} \quad (5.14)$$

$$fetal_{QRS} = \begin{cases} \tilde{fetal}_{QRS} & (\tilde{fetal}_{QRS} \geq th_{aRSfetal}) \& (\tilde{fetal}_{QRS} < th_{maternal}) \\ 0 & \tilde{fetal}_{QRS} < th_{aRSfetal} \end{cases} \quad (5.15)$$

Donde  $th_{maternal}$  hace referencia al umbral de referencia de los complejos QRS maternos, y  $th_{aRSfetal}$  hace referencia al umbral de referencia de los complejos QRS fetales.

El principal problema era la variabilidad de los resultados en función del umbral escogido, es decir, si se escoge un umbral demasiado alto, detectaremos menos complejos fQRS, llevando esto al exceso de aparición de FN y a la reducción de sensibilidad de nuestra señal. Mientras que si realizamos la acción contraria, escoger un umbral demasiado bajo, detectaremos más complejos fQRS de los existentes, aumentando en exceso nuestros FP, y por lo tanto reduciendo el *PDV* de la señal.

Es entonces cuando surge el problema de *¿qué valor de umbral debemos de escoger para cada señal? ¿Existe algún umbral general válido para todas las señales aECG que nos permita diferenciar correctamente las señales fECG del ruido, y de las mECG?*

### 5.3 El problema de la utilización de umbrales

---

La respuesta a estas cuestiones es una respuesta compleja, a la cuál únicamente podíamos llegar mediante un *razonamiento inductivo*. Este tipo de razonamiento posee una lógica estructural desde la especificación a la generalización, permitiéndonos medir la probabilidad de los argumentos. Al contrario que el deductivo, se centra en la creación de declaraciones generalizadas a partir de sucesos específicos. Por lo que, nos era imposible analizar todas las señales aECG existentes para adecuar nuestro umbral a todas ellas. No podemos encontrar un valor de umbral característico de las señales aECG, debido a que estas señales varían en función de diversos factores, como son las semanas de gestación, posición del feto, posición de los electrodos, ruido eléctrico, etc. En su defecto, podemos utilizar algoritmos que regulen mediante ecuaciones el nivel de umbral con respecto al resto de la señal. En definitiva, a pesar de que los datos estén normalizados, es imposible encontrar un valor fijo para estos umbrales, dependiendo este de los parámetros mencionados anteriormente.

Por lo que nos encontramos ante dos posibles metodologías para definir el umbral:

- **Asignación generalizada del umbral.** Este tipo de asignación plantea la problemática de *razonamiento inductivo* anteriormente desarrollada.
- **Asignación específica del umbral para cada señal mediante visualización de la señal.** Esta asignación, permite ajustar los parámetros de cada señal en función de la misma, logrando una mayor eficiencia que en el caso anterior, pero con la problemática de la implementación en tiempo real. Es decir, es ineficiente observar la señal, y a continuación establecer el umbral en función de la misma.

Por lo tanto, debido a los problemas de ambas soluciones, se implementó un método que buscarse una correlación entre ambas metodologías, aprovechándose de las ventajas de ambas y resolviendo así los inconvenientes presentes. Este método se basa en el uso de los siguientes umbrales que se describen a continuación:

- $th_{maternal}$ : usado para detectar los mQRS. Puede calcularse automáticamente como un cierto porcentaje del máximo de la señal aECG procesada. Este porcentaje se elige de acuerdo a un valor de referencia [10], el cuál demuestra que un valor por encima del 60 % del  $MAX_{ECG}$  puede ser considerado como un mQRS.
- $th_{dRSfetal}$  y  $th_{aRSfetal}$ : estos umbrales evalúan la distancia entre tiempos y la amplitud, respectivamente, entre los picos fR y fS seleccionados como posibles

## 5. INVESTIGACIÓN

---

candidatos. Basándose en el estudio de la señal fECG y de la dependencia de esta con diversos factores [10],  $th_{dRSfetal}$  se establece entre 0.020 y 0.035 segundos, y el valor exacto en cada caso es derivado de la base de datos de entrenamiento. Mientras tanto,  $th_{aRSfetal}$  se establece para variar entre un rango relacionado con el máximo de los complejos mQRS después de la eliminación de BW y del ruido, por lo que el valor para este umbral se calcula automáticamente y se adapta durante el proceso, basándose en el estudio de las distancias máximas seguidas de mínimos.

En la figura 5.3 podemos observar la aplicación del método de valor umbral propuesto, en ella las líneas verticales se corresponden con las anotaciones: en azul, fQRS correctamente detectados; en verde, fQRS no detectados. Los puntos negros indican máximos que superan el umbral, Th, marcado por la línea roja. Para el caso ilustrado en la gráfica,  $TD = 4$ ,  $FP = 2$ ,  $FN = 1$ ,  $S_e = 0,80$ ,  $PDV = 0,67$ ,  $A_{cc} = 57,14\%$

Una vez analizado en profundidad el método, se llegó a la conclusión de que era necesario investigar sobre algoritmos de clasificación para mejorar la eficacia de esta implementación. Por lo que se comenzará introduciendo las posibles soluciones estudiadas y los métodos y algoritmos propuestos.

### 5.4. El problema de la clasificación

El siguiente paso, fue encontrar un método o algoritmos que nos ayudaran a superar el problema de la utilización de umbrales para clasificar nuestra señal. Para ello, se investigó acerca de posibles soluciones empleadas en el campo de la biomedicina, clasificación de señales, etc.

Vivimos en un mundo rodeado de datos, en el que la información se almacena o representa como datos, para después proceder a su administración y análisis. Uno de los problemas más importantes con respecto al tratamiento de la información, es su clasificación o agrupamiento. Esta acción es una de las actividades más primitivas desarrolladas por los seres humanos, jugando esta un papel muy importante en el desarrollo histórico de la humanidad. Cuando se trata de aprender un nuevo objeto, se buscan características que pueda definirlo, y además se compara con otros objetos conocidos o fenómenos, basándose en la similitud o disimilitud, proximidad, respecto a algunos estándares o reglas.

El problema de la clasificación de señales es probablemente el tipo de problema más estudiado en el campo del aprendizaje máquina (**machine learning**), y ha sufrido un importante desarrollo algorítmico y teórico en el último siglo.

El problema más fácil de los pertenecientes a la clasificación, la **clasificación binaria**, se reduce a la siguiente cuestión: dado un patrón  $x$ , dibujado sobre un dominio  $X$ , estima que valor de la variable aleatoria  $y \in \{\pm 1\}$  asumirá. Básicamente, si queremos clasificar dos objetos bien diferenciados,  $A$  y  $B$ , queremos saber, a qué objeto pertenecerá nuestro dato. La habilidad para resolver este problema básico nos permite abordar una gran variedad de situaciones prácticas. Por otra parte nos encontramos la **clasificación de varias clases**, la extensión lógica de la clasificación binaria. La principal diferencia reside en que ahora  $y \in \{1, \dots, n\}$  asume un rango de diferentes valores, queremos clasificar más de dos grupos.

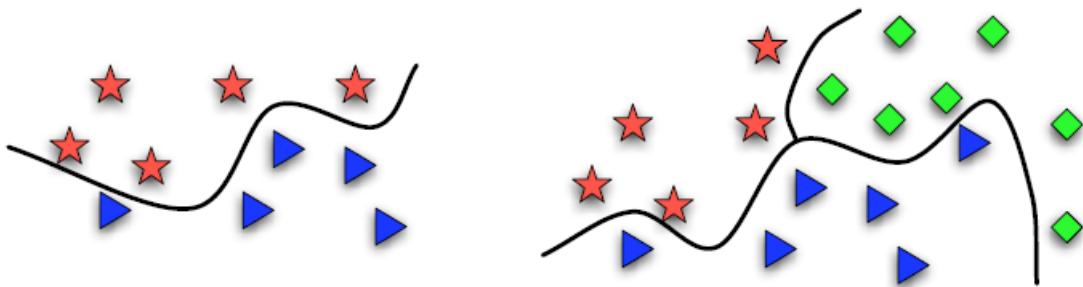


Figura 5.4: Izquierda: clasificación binaria. Derecha: clasificación en 3 clases.

En la figura 5.4 podemos ver la diferencia entre ambos tipos de clasificación. Debemos de tener en cuenta que en el caso de la derecha, hay un mayor grado de ambigüedad.

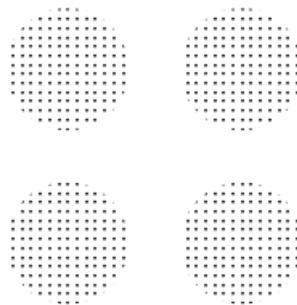


Figura 5.5: Nubes de puntos en el espacio Euclídeo bidimensional. [21]

**¿Qué es un grupo? ¿Qué diferencia a un grupo con respecto de otro?**

Básicamente, hay dos tipos de definición acerca de los grupos:

## 5. INVESTIGACIÓN

---

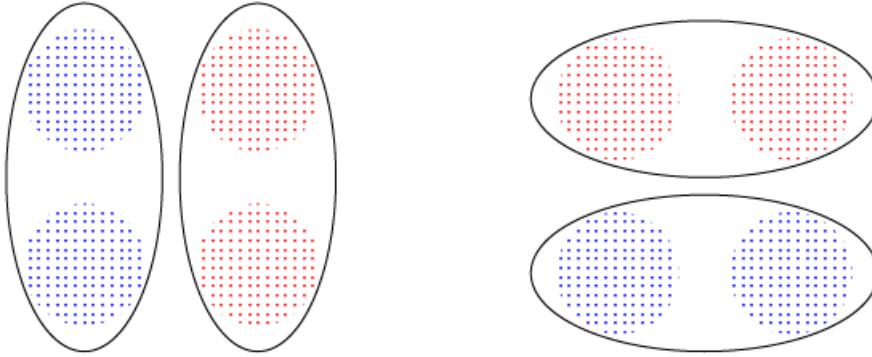


Figura 5.6: Soluciones para la clasificación en dos grupos de los datos anteriores. [21]

- **Definición interna:** un punto u objeto pertenece al mismo grupo si posee una alta similitud con el resto de puntos pertenecientes a ese grupo. Haciendo referencia a su homogeneidad interna.
- **Definición externa:** un punto no pertenece a un grupo o clase si existe un alto grado de disimilitud con el resto de puntos del grupo. Por lo que, dos grupos difieren entre sí, cuando ambos poseen un alto grado de disimilitud.

Con similitud, o disimilitud hacemos referencia a diferentes características de estos datos: forma, tamaño, distancia, etc. En este proyecto se ha trabajado principalmente con la definición de similitud como distancia entre dos datos u objetos.

En la Figura 5.6 vemos las diferentes posibilidades de agrupamiento para los datos de la Figura 5.5. Cada agrupamiento de datos, ellipse, corresponde con un grupo.

Pero, a pesar de encontrarnos antes diferentes problemas de clasificación, también nos encontramos con diferentes métodos de resolución. Básicamente, los sistemas de clasificación pueden ser supervisados o no supervisados, dependiendo de si estos asignan nuevas entradas a un grupo de números finitos de clases discretas supervisadas o categorías no supervisadas, respectivamente.

En la **clasificación o aprendizaje supervisada**, la cartografía o trazado de grupos, a partir de un grupo de vectores de entrada ( $x \in R^d$ , donde  $d$  es la dimensión del espacio de entrada) hacia un grupo finito de clases discretas, se modela en términos de una función matemática  $y = y(x, w)$ , donde  $w$  es el vector de parámetros ajustables. Los valores de estos parámetros están determinados (optimizados) por un algoritmo de aprendizaje inductivo, cuyo objetivo es minimizar el riesgo empírico (relacionado con el principio de inductivo) en un conjunto finito de ejemplos de entrada-salida,  $(x_i, y_i)$ ,  $i = 1, \dots, N$ , donde  $N$  es la cardinalidad finita de un conjunto de datos representativos.

Cuando el algoritmo de aprendizaje inductivo converge o termina, se genera una clasificación inducida. [28]

En la **clasificación o aprendizaje no supervisada**, denominada *Clustering* o análisis exploratorio de datos, no hay disponible ningún dato agrupado. El objetivo del *Clustering* es separar grupos de datos no clasificados en un conjunto finito, discreto con estructura “natural”, en lugar de proporcionar una caracterización precisa de muestras no observadas, generadas por la misma distribución de probabilidad. Esto puede hacer que la tarea del agrupamiento no funcione correctamente fuera del marco de los problemas predictivos de aprendizaje no supervisado, como por ejemplo la estimación de funciones de densidad de probabilidad, la maximización de la entropía. Tener en cuenta, que el *Clustering* difiere del escalamiento multidimensional (mapas perceptuales), cuyo objetivo es representar todos los objetos evaluados de una manera que se minimice la distorsión topográfica mientras se utilizan las menores dimensiones posibles. El agrupamiento no-predictivo es una proceso subjetivo de la naturaleza, la cuál se opone a un juicio absoluto en cuanto a la eficacia relativa de todas las técnicas de agrupamiento.

**¿Qué tipo de aprendizaje escoger?** En el aprendizaje no supervisado es posible aprender modelos y estructuras mayores y más complejas, que con el aprendizaje supervisado, debido a que en el aprendizaje supervisado se intentan encontrar las conexiones entre dos conjuntos de observaciones, por lo que, la dificultad de la tarea se incrementa exponencialmente con el número de pasos entre los dos conjuntos. Esto es por lo que la clasificación supervisada, no puede, en la práctica, aprender modelos con una jerarquía profunda.

Además, para la utilización de algoritmos de clasificación supervisada, es necesario una base de datos de entrenamiento, para “entrenar” al sistema. Por lo que en nuestro caso, estos datos debíamos de obtenerlos manualmente de una base de datos de señales aECG, escasas y con muy pocas anotaciones fetales, por lo que necesitaríamos la presencia de un especialista para realizar nuestra base de entrenamiento.

Debido a estas complejidades, y que estamos ante el estudio inicial de la clasificación de un problema, se decidió emplear técnicas de aprendizaje no supervisado, es decir, para la resolución de nuestro problema, clasificar las señales fECG y mECG presentes en las señales aECG, utilizamos una clasificación mediante *Clustering*.

## 5. INVESTIGACIÓN

---

### 5.5. Clasificación no supervisada: *Clustering*

Las tareas correspondientes al *Clustering* pueden variar en función del tipo de entrada que tengamos y del tipo de salida esperada. Concretamente, nos centraremos en la siguiente configuración común:

- **Entrada:** un grupo de datos,  $X$ , y una función de disimilitud o distancia entre ellos, que cumpla con todas las propiedades descritas en la sección 5.7, simetría, desigualdad triangular, etc. Adicionalmente, algunos algoritmos de *Clustering*, requieren la entrada de un parámetro  $k$  (el cuál determina el número de clústeres).
- **Salida:** se dividen los datos de entrada,  $X$ , en diferentes grupos. Esto es,  $C = (C_1, \dots, C_k)$ , donde  $\bigcup_{i=1}^k C_i = X$ , y para todo  $i \neq j$ ,  $C_i \cap C_j = \emptyset$ . Otra posible opción de salida es el *dendograma* (del griego *dendron*, árbol, y *gramma*, dibujo).

Por lo tanto, los algoritmos de *Clustering* dividen los datos en un cierto número de grupos (**clústeres**). No hay una definición universal de este tipo de aprendizaje, aunque, la mayoría de las definiciones coinciden con la definición dada anteriormente sobre grupos, haciendo mención a la similitud, es decir, los patrones de un mismo grupo deberán ser similares, mientras que en grupos diferentes no. Tanto, la similitud como la disimilitud deben de ser descritas de una manera más legible, por lo que a continuación, expondremos diferentes descripciones matemáticas sobre distintos tipos de *Clustering*:

Dado un conjunto de patrones de entrada,  $\mathbf{X} = \{x_1, \dots, x_j, \dots, x_N\}$ , donde  $x_j = (x_{j1}, x_{j2}, \dots, x_{jd})^T \in R^d$  y cada medida  $x_{ji}$  es una característica. Se definen las siguientes clases de *Clustering*:

- **Clustering particional (hard):** intenta dividir  $\mathbf{X}$ , en  $k$ -particiones,  $C = \{C_1, \dots, C_k\}$  ( $K \leq N$ ) tal qué:
  1.  $C_{i_K} \neq \phi$ ,  $i = 1, \dots, K$ ;
  2.  $\bigcup_{i=1}^k C_i = \mathbf{X}$ ;
  3.  $C_i \cap C_j = \emptyset$ ,  $i, j = 1, \dots, K$  y  $i \neq j$ .
- **Clustering jerárquico:** este tipo de *Clustering* intenta construir una partición como un árbol de estructura anidada de  $\mathbf{X}$ ,  $H = \{H_1, \dots, H_Q\}$  ( $Q \leq N$ ), tal que  $C_i \in H_m$ ,  $C_j \in H_l$ , y  $m > l$  lo cuál, implica que  $C_i \subseteq C_j$  o  $C_i \cap C_j = \emptyset$  para todo  $i, j \neq i, m, l = 1, \dots, Q$ .

En el *Clustering* particional, cada dato pertenece únicamente a un grupo. Sin embargo, cada dato puede pertenecer a todos los grupos con un grado de pertenencia a cada grupo,  $u_{i,j} \in [0, 1]$ , el cuál representa el coeficiente de pertenencia del objeto  $j$ -ésimo en el cluster  $i$ -ésimo y satisface las siguientes dos restricciones:

$$\sum_{i=1}^c u_{i,j} = 1, \quad \forall j \quad \text{y} \quad \sum_{i=1}^c u_{i,j} < N, \quad \forall i$$

tal y como está definida en la teoría de conjuntos difusos (*fuzzy set theory*). Esto es conocido como *Clustering* difuso o *fuzzy Clustering*.

### Procedimiento de clasificación mediante *Clustering*

A continuación, vamos proceder a explicar el procedimiento del análisis de la clasificación de nuestras señales mediante *Clustering*:

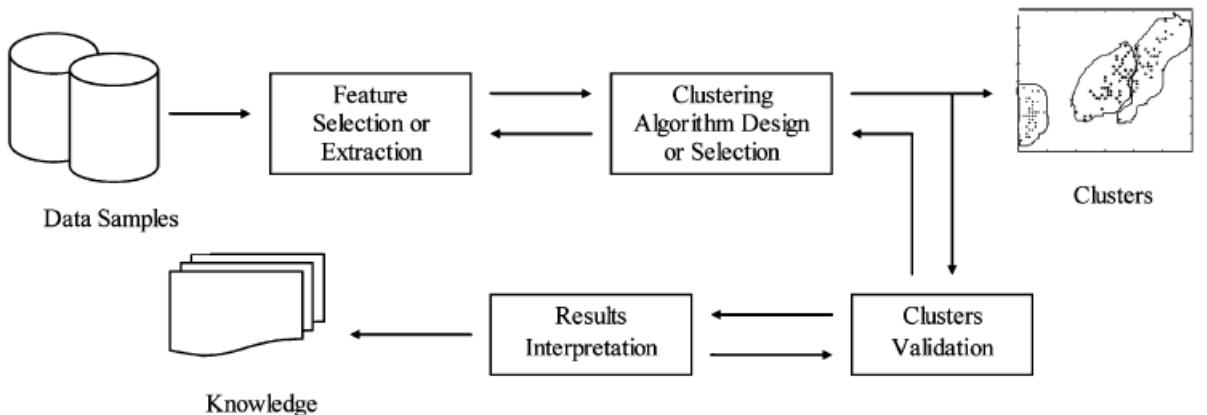


Figura 5.7: Procedimiento de clasificación mediante *Clustering*. [28]

- Selección de características o extracción.** La selección de características elige características distintivas de un conjunto de candidatos, mientras que la extracción de características utiliza algunas transformaciones para generar características novedosas y útiles de las características originales. Ambas son cruciales para la efectividad de la aplicación del *Clustering*. Una buena elección de las características puede reducir la carga de trabajo, y simplificar el proceso de diseño subsecuente. Generalmente, las características ideales deberán ser utilizadas para distinguir correctamente patrones que pertenezcan a diferentes grupos, su inmunidad al ruido y fáciles de extraer e interpretar.
- Diseño del algoritmo de *Clustering* o selección.** Este paso se combina a menudo con la selección de una medida de proximidad y la construcción de una

## 5. INVESTIGACIÓN

---

función de criterio. Los datos se agrupan en función de su parecido. Obviamente, la medida que escojamos como medida de proximidad, afectará a la formación de los clusteres resultantes, aunque casi todos los algoritmos están explícitamente o implícitamente unidos a algunas definiciones de medidas de proximidad. Una vez escogida la medida de proximidad o distancia, la construcción de una función de criterio para el agrupamiento hace que la partición en clusteres sea un problema de optimización. El *Clustering* es un problema ubicuo, es decir, cada algoritmo está ideado para solucionar diferentes problemas en diversos campos, por lo que no existe ningún algoritmo universal de *Clustering* que pueda resolver todos los problemas. Por esta razón, es necesario investigar acerca de las características que tenemos en los datos de nuestro problema, para así, poder seleccionar un algoritmo de clasificación correcto, que se adecue perfectamente a nuestros datos.

3. **Validación de los clusteres.** Dado un conjunto de datos, cada algoritmo de *Clustering* genera una división, sin importar el tipo de estructura que exista en ellos. Por otra parte, diferentes enfoques conducen a diferentes clusteres o grupos; incluso con la utilización del mismo algoritmo, la identificación de diferentes parámetros o el orden de colocación de los mismos puede afectar al resultado final. Por lo tanto, los estándares de evaluación son importantes para proveer al usuario de un cierto grado de confianza en los resultados, derivados de la utilización del algoritmo elegido. Estas evaluaciones deben de ser objetivas y no tener preferencia por ningún algoritmo. Además, deben de ser útiles para responder a cuestiones como *¿cuantos clusteres hay en los datos?*, *¿son los grupos obtenidos correctos, o un error de los algoritmos?* o *¿por qué escogemos un algoritmo en lugar de otro?*

En general, hay tres categorías para evaluar la validez del algoritmo utilizado: índices externos, internos y relativos. Estos están definidos sobre tres tipos de estructuras de *Clustering*, conocidas como *Clustering* particional, jerárquico y individual. Los *índices externos* se basan en estructuras predefinidas, que son el reflejo de la información previa de los datos, y se utilizan como estándares para evaluar las diferentes soluciones de clasificación. Los *índices internos* no dependen de información externa, por el contrario, examinan la estructura del *Clustering* directamente de los datos originales. Por su parte, los *criterios relativos* se encargan de la comparación entre diferentes estructuras de clusteres,

## 5.6 Búsqueda de las características y patrones de la señal aECG

---

proveyendo una referencia, para decidir cual de los algoritmos se ajusta mejor a las características de nuestros datos.

4. **Interpretación de los resultados.** El objetivo final del *Clustering* es proveer al usuario de las particiones deseadas, clasificando correctamente sus datos, y por lo tanto resolviendo de una manera efectiva el problema planteado.

Hay que tener en cuenta que, en ocasiones, es necesario realizar varias repeticiones para lograr la clasificación deseada. Además, no existen criterios universales y efectivos para guiarnos acerca de la selección de algoritmos o características. Los criterios de validación nos pueden proporcionar indicios acerca de la efectividad de nuestras soluciones, pero incluso, escoger el criterio adecuado es un problema adicional que requiere mayor carga de trabajo o coste computacional.

## 5.6. Búsqueda de las características y patrones de la señal aECG

Antes de proceder a aplicar los diferentes algoritmos de clasificación mediante *Clustering*, era necesario encontrar los patrones y características propias de nuestra señal ECG abdominal.

Nuestra señal, está constituida principalmente por tres grupos o señales independientes, que son: complejos QRS maternos, complejos QRS fetales y ruido. Por lo que para clasificar estos datos necesitamos la presencia de cada uno de ellos en la misma señal. Estos datos, deben de diferir entre sí, es decir, las características o característica de cada uno de los grupos deben de ser diferentes, o tener un alto grado de disimilitud con respecto a las características de los otros grupos.

Por lo que, realizando el preprocesado de nuestra señal original (Figura 5.8) mediante la WT, obteníamos una señal con menor presencia de ruido, en el que se diferencian los complejos QRS de una mejor manera, pero que aún contaba con gran presencia de este. Básicamente nuestro principal problema, antes de aplicar ningún algoritmo era: *¿Qué características son propias de cada grupo? ¿Qué herramienta puedo utilizar para diferenciarlas?*

Las características de la señal aECG tienen una particularidad interesante, que permite diferenciar de manera visual claramente los diferentes grupos, es decir, si nos fijamos en los complejos QRS de la señal preprocesada vemos como hay dos características principales sobre las que guiarnos:

## 5. INVESTIGACIÓN

---

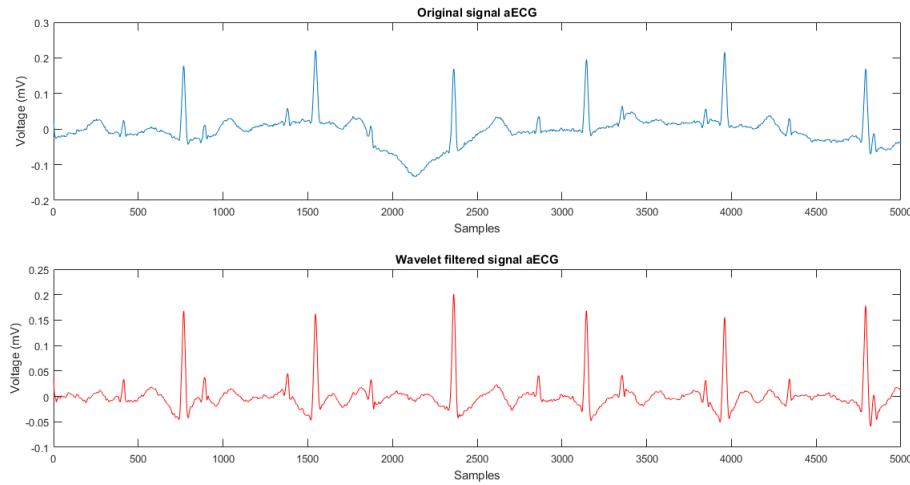


Figura 5.8: Señal aECG *r04.edf* de Physionet, canal 4, muestras 5001 a 10000. Azul: señal original sin procesar. Rojo: señal preprocesada mediante transformada *wavelet*.

- **Amplitud de la señal:** tal y como se observa en la señal preprocesada de la figura 5.8, los complejos QRS maternos, poseen mayor amplitud que el resto de señales de la señal aECG. Esto es debido, entre otras cosas al mayor tamaño del corazón materno con respecto del fetal, y con respecto del ruido. Si observamos los complejos QRS fetales, vemos como estos, a veces, se pueden confundir con el ruido, si únicamente observamos la amplitud de la señal, ya que el menor tamaño del corazón, unido a diversos problemas de medida y señales aleatorias presentes en nuestras mediciones, originan un ruido, que a veces puede confundirse fácilmente con nuestros complejos QRS fetales atendiendo únicamente a la amplitud de la señal.
- **Distancia entre el pico RS de los complejos:** es decir, además de la amplitud, se observa claramente en la señal la diferencia entre el pico RS, la diferencia entre un máximo local, seguido de un mínimo local puede proporcionarnos información acerca de los diferentes grupos. Por lo que, una mayor diferencia entre un máximo local seguido de un mínimo, nos informa de que estamos ante un posible candidato a complejo QRS materno. Por otra parte, si la diferencia es menor que la del materno, pero es mayor que la del ruido, estaremos ante un candidato a complejo QRS fetal. Por último, la menor diferencia entre máximo seguido de mínimo, nos informa ante la posibilidad de un candidato a ruido.

En nuestro caso, hemos utilizado como datos para nuestra clasificación la distancia entre el pico RS de los complejos, debido a que su utilización englobará a la utilización

## 5.6 Búsqueda de las características y patrones de la señal aECG

de amplitudes. Por lo que, nuestros datos serán un vector unidimensional con todas las distancias entre posibles picos RS.

Atendiendo a estas dos características, la siguiente cuestión era: *¿Qué disposición deben de tener los datos? ¿Cómo deben de estar organizados?*

Básicamente, nos encontramos ante el problema de la estructuración y/o ordenación de los datos. Tal y como se ha mencionado anteriormente, el posicionamiento de los datos sobre el espacio es crucial para obtener una correcta clasificación de nuestra señal, debido a que básicamente, como veremos más adelante, la principal medida de similitud utilizada en este tipo de algoritmos es la distancia entre los diversos datos. Por lo que, una mala estructuración espacial del problema conllevaría a una mala clasificación de nuestros datos. Atendiendo a esto, principalmente nos encontramos ante dos posibilidades principales de organización espacial:

1. **Organización de los datos de manera temporal, según orden de muestreo:** la manera más intuitiva de estructurar los datos era según se iban obteniendo en el muestreo. Por lo que, a medida que íbamos muestreando se calculaba la amplitud (en mV) de cada dato, y si se detectaba un máximo local seguido de mínimo local, estábamos ante la presencia de un posible complejo QRS, se almacenaba. Por lo que, estructurando los datos según este procedimiento obtendríamos un posicionamiento de los mismos en el espacio tal y como aparece en la figura 5.9

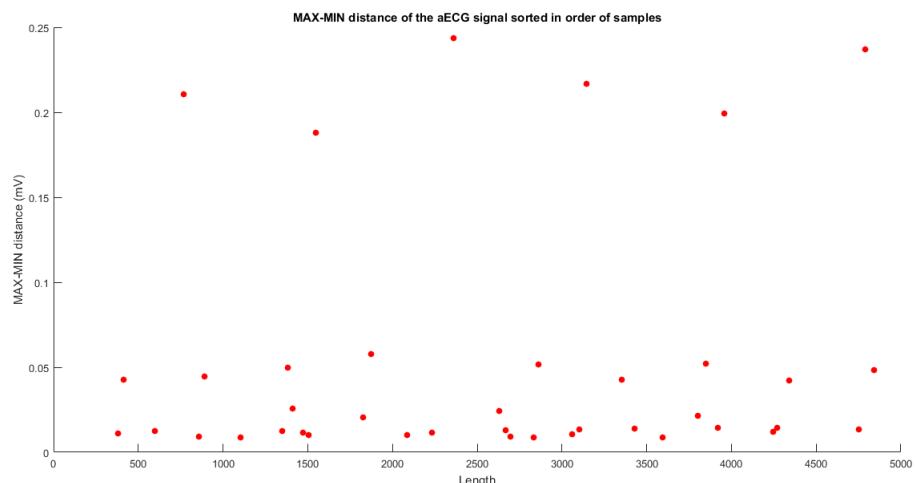


Figura 5.9: Datos de distancia máximos seguidos de mínimos de la señal aECG ordenados según orden de muestreo

## 5. INVESTIGACIÓN

---

2. **Organización de los datos en orden creciente o decreciente:** una manera menos intuitiva de organizar los datos, es utilizar una organización creciente o decreciente atendiendo a la amplitud de la distancia de máximos locales seguidos de mínimos locales. Esta ordenación se puede ordenar en un orden ascendente como en uno descendente. A continuación se añaden dos figuras con la ordenación de los datos en el espacio atendiendo a este tipo de organización, siendo la figura 5.10 organizada en orden creciente, mientras que 5.11 está en orden decreciente.

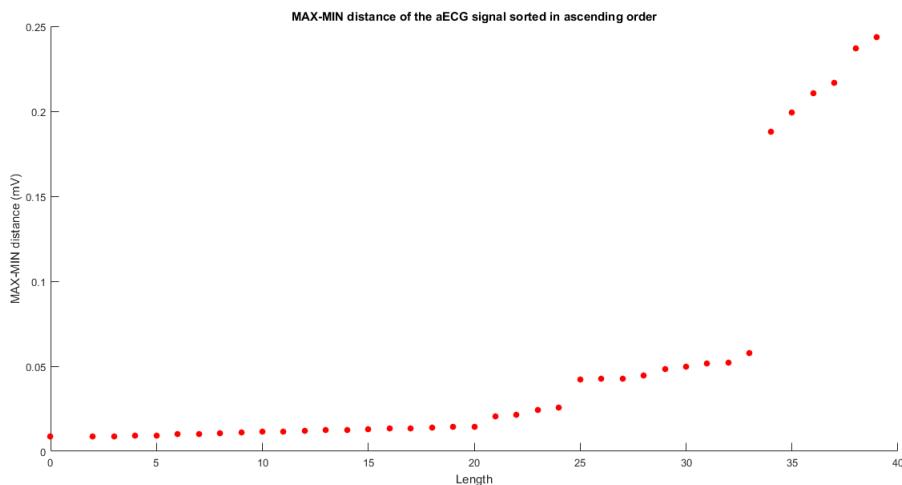


Figura 5.10: Datos de distancia máximos seguidos de mínimos de la señal aECG ordenados según orden de muestreo

Ante estas dos posibles organizaciones de resolución de nuestro problema, estábamos ante el problema de *¿cuál de todas ellas escoger?*

Un problema que no era fácil de resolver. Por lo que deberíamos de probar con los diferentes algoritmos de clasificación en estas distintas estructuras, para ver qué algoritmo se ajustaba mejor a cada una de ellas, y cuál de todas, proporcionaba una mejor clasificación. Además, *atendiendo a la eficiencia computacional*, cabe destacar que la organización según orden de muestreo, no necesitaba ordenar las señales, lo cuál era una ventaja competitiva con respecto al otro tipo de agrupación de datos. Esto es debido a que los algoritmos de ordenamiento, poseen un mayor coste computacional que la no presencia de estos, además del uso de memoria y otros recursos computacionales. Por lo que si los algoritmos de clasificación funcionan bien en este tipo de organización, la implementación del algoritmo en un sistema hardware tendría

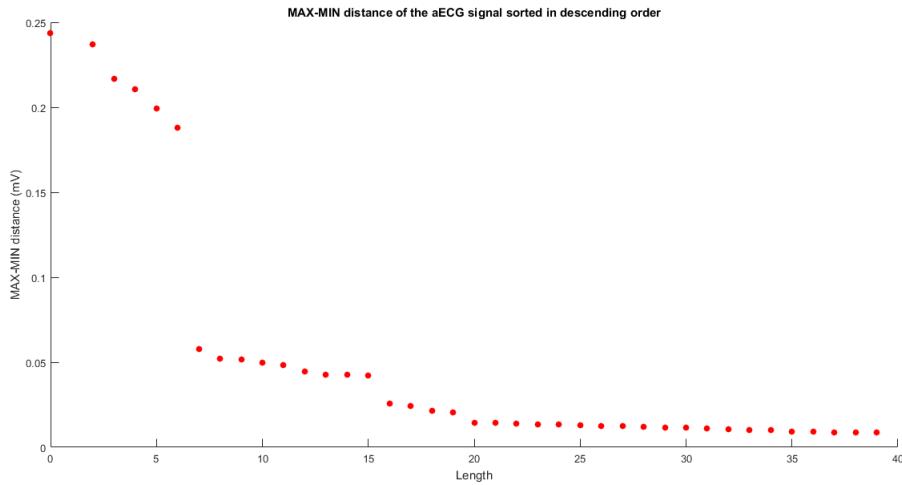


Figura 5.11: Datos de distancia máximos seguidos de mínimos de la señal aECG ordenados según orden de muestreo

mayor eficiencia computacional. Aunque no debemos de olvidar que, para la primera fase de desarrollo, nuestro principal propósito es conseguir una correcta clasificación, independientemente del coste computacional de la misma. Obviamente a igualdad de resultados, se implementará el algoritmo con menor coste computacional.

## 5.7. Búsqueda de las herramientas de medida de similitud

Una vez estudiadas las características de los datos que forman la señal aECG y sus posibles clasificaciones procedemos a investigar las herramientas o herramienta para poder medir la similitud entre los datos de la señal aECG.

La principal herramienta utilizada para medir similitud entre datos es el uso de distancias entre datos. *¿Por qué?* La distancia está directamente relacionada con la similitud (o disimilitud) entre objetos, debido a que la mayoría de los algoritmos de agrupamiento mediante *Clustering* se centran en ella, es decir, *¿si dos datos están muy cercanos el uno del otro, y ambos están muy alejados del tercero, cuales pertenecen a un grupo y cuál a otro?*, y *¿qué tipo de herramienta utilizamos para averiguarlo?* Los dos primeros datos pertenecerán al primer grupo, mientras que el tercero pertenecerá al otro grupo. Esto es posible gracias a la utilización del concepto de distancia entre datos (vectores).

Básicamente, un dato perteneciente a un conjunto de datos es descrito por un conjunto de características, por lo general representado como un vector multidimensional.

## 5. INVESTIGACIÓN

---

Estas características pueden ser cuantitativas o cualitativas, continuas o binarias, etc, las cuales determinan los mecanismos de medida correspondientes.

Matemáticamente, una **función de disimilitud** o distancia de un conjunto de datos  $M$  se define para satisfacer las siguientes condiciones:

1. *Simetría*:  $D(x_i, x_j) = D(x_j, x_i)$ ;
2. *Positividad*:  $D(x_i, x_j) \geq 0$  para toda  $x_i$  y  $x_j$ .
3. *Desigualdad de Minkowski*: En todo triángulo la suma de las longitudes de dos lados cualquiera es siempre mayor a la longitud del lado restante.

$$D(x_i, x_j) \leq D(x_i, x_k) + D(x_k, x_j) \quad \forall x_i, x_j, x_k \quad (5.16)$$

4. *Reflexividad* .  $D(x_i, x_j) = 0$  si  $x_i = x_j$ .

Igualmente, una **función de similitud** se define para satisfacer las siguientes condiciones:

1. *Simetría*.  $S(x_i, x_j) = S(x_j, x_i)$ ;
2. *Positividad*.  $0 \leq S(x_i, x_j) \leq 1$  para toda  $x_i$  y  $x_j$ .
3.  $S(x_i, x_j)S(x_j, x_k) \leq [S(x_i, x_j) + S(x_j, x_k)]S(x_i, x_k) \quad \forall x_i, x_j, x_k$
4. *Métrica de similitud* .  $S(x_i, x_j) = 1$  si  $x_i = x_j$

Por norma general, las funciones de distancia o disimilitud son utilizadas para medir características cuantitativas, mientras que las medidas de similitud son más importantes para variables cualitativas. Por ejemplo, en el caso de datos de tipo binario (0/1) se utilizan medidas de similitud, por que ambos grupos o tipos de datos son completamente diferentes (sistemas todo-nada). Debido a que en nuestros tipos de datos, poseen características cualitativas muy parecidas (no estamos ante el caso de sistemas todo-nada), nos guiaremos a través de las características cuantitativas, por lo que utilizaremos medidas de distancia en la clasificación de nuestras señales.

Por lo tanto, en el Cuadro 5.1 podemos observar distintas medidas de disimilitud (distancias) y similitud de nuestros datos, definidas para un espacio Euclídeo de  $n$ -dimensiones,  $\mathbb{R}^n$ , las cuales utilizaremos para nuestros algoritmos de clasificación. Volviendo a las propiedades de las funciones de disimilitud, el axioma de la desigualdad triangular, o de *Minkowski* es el axioma que hace que todas las medidas

## 5.7 Búsqueda de las herramientas de medida de similitud

---

| MEDIDAS                                       | ECUACIONES   | COMENTARIOS  |
|---|--|--|
| Distancia de <i>Minkowski</i>                 | $D_{xy} = \left( \sum_{l=1}^d  x_l - y_l ^n \right)^{1/n}$                               | Métrica. Invariante ante cualquier rotación y traslación para $n = 2$ (Euclídea). Los datos con mayores valores y varianzas tienden a dominar sobre los demás. |
| Distancia Euclídea                            | $D_{xy} = \left( \sum_{l=1}^d  x_l - y_l ^2 \right)^{1/2}$                               | La distancia métrica más utilizada. Caso especial de la distancia de <i>Minkowski</i> para $n = 2$ . Tiende a formar clusteres hiperesféricos.                 |
| Distancia <i>Cityblock</i> o <i>Manhattan</i> | $D_{xy} = \sum_{l=1}^d  x_l - y_l $  | Caso especial de la distancia de <i>Minkowski</i> para $n = 1$ . Tiende a formar clusteres hiperesféricos.   |
| Distancia de <i>Mahalanobis</i>               | $D_{xy} = (x - y)^T S^{-1} (x - y)$ donde $S$ es la matriz de covarianza del mismo grupo | Tiende a formar clusteres hiperelipsoidales. Cuando $S$ es la matriz identidad, la distancia de <i>Mahalanobis</i> es la distancia Euclídea                    |
| Distancia de <i>Chebychev</i>                 | $D_{xy} = \max_{1 \leq l \leq d}  x_l - y_l $  | Caso especial de la distancia de <i>Minkowski</i> para $n \rightarrow \infty$ .  |
| Similitud del coseno                          | $D_{xy} = \cos\alpha = \frac{\mathbf{x}^T \mathbf{y}}{\ \mathbf{x}\  \ \mathbf{y}\ }$    | Independiente de la longitud del vector. Invariante frente a la rotación, pero no a transformaciones lineales.   |

Cuadro 5.1: Medidas de disimilitud y similitud para características cuantitativas.

de distancia tengan la propiedad de describir el camino más corto de un punto a otro. A continuación, exponemos el tipo de distancia más utilizada, la que nos describe de manera matemática el concepto “natural” de distancia que todos tenemos en mente.

Un *espacio Euclídeo de n-dimensiones* es un espacio en donde todos sus puntos son vectores de  $n$  números reales. La medida de distancia convencional en este espacio es la que aparece en el Cuadro 5.1 (distancia Euclídea). Esto es, hacemos el cuadrado de la distancia en cada dimensión, sumamos todos sus cuadrados, y tomamos el valor positivo de la raíz.

Es fácil comprobar que se cumplen los requisitos para que sea una función de distancia. La *desigualdad triangular* es la más difícil de comprobar. Sin embargo, esta es una propiedad en sí del espacio Euclídeo.

Otra distancia muy común utilizada en algoritmos de clasificación, es la distancia  $L_1$ -normalizada o distancia de *Manhattan* o *Cityblock*. En ella, la distancia entre dos puntos es la suma de las magnitudes de las diferencias de cada dimensión. Es

## 5. INVESTIGACIÓN

---

nombrada como distancia de *Manhattan* ya que es la distancia que habría que recorrer entre los puntos si nos viésemos obligados a viajar a lo largo de las líneas de cuadrícula.

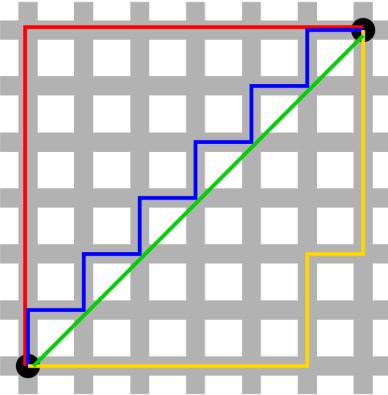


Figura 5.12: Representación para  $\mathbb{R}^2$  de la distancia entre dos puntos mediante uso de la distancia *Euclídea* (verde) y distancia de *Manhattan* (el resto).

En la Figura 5.12 vemos la diferencia entre el uso de la distancia Euclídea y de *Manhattan* para medir la distancia entre dos puntos. Tal y como observamos, mientras que la distancia Euclídea, únicamente posee un camino más corto para llegar de un punto al otro, la distancia de *Manhattan* puede utilizar diferentes caminos de la misma longitud para lograr su objetivo.

Por otra parte, la distancia de *Mahalanobis* utiliza la matriz de covarianza para calcular la distancia entre dos puntos. Si la matriz de covarianza es la matriz identidad, la distancia en este caso será la propia distancia Euclídea. Si por otra parte, la matriz de covarianza es diagonal, entonces la medida de distancia resultante es la distancia Euclídea normalizada:

$$D_{xy} = \sqrt{\sum_{l=1}^d \frac{(x_l - y_l)^2}{s_{xy}^2}} \quad (5.17)$$

donde  $s_{xy}$  es la desviación estándar de  $x$  e  $y$  sobre las muestras. Básicamente, se diferencia de la distancia Euclídea en que esta distancia tiene en cuenta la correlación entre las variables aleatorias.

## Caracterización del espacio de clasificación en señales aECG: Espacio Euclídeo unidimensional

Tal y como se observan en las Figuras 5.9 y 5.10 nos encontramos ante un espacio Euclídeo unidimensional,  $\mathbb{R}$ , debido a que se cumplen las propiedades de espacio Euclídeo para  $n = 1$ . Esta dimensión es comúnmente denominada *longitud*, haciendo referencia a un único eje cartesiano. Debido a que nuestros datos serán datos representados por un vector unidimensional, todas las medidas de distancia las aplicaremos al espacio  $\mathbb{R}$ . Por lo tanto, la definición de la distancia Euclídea en un espacio unidimensional será:

$$D_{xy} = \left( \sum_{l=1}^1 |x_l - y_l| \right)^{1/1} = \sqrt{(x - y)^2} = x - y \quad (5.18)$$

A pesar de ser la distancia Euclídea, la medida de disimilitud por defecto, la operación de raíz cuadrada conlleva un alto coste computacional (en el caso unidimensional no será necesario calcularla, pero, el algoritmo por defecto, elevaría la diferencia al cuadrado para proceder posteriormente al cálculo de la raíz), por lo que es frecuente encontrar en los algoritmos de *Clustering* una medida alternativa de esta distancia, la **distancia Euclídea al cuadrado** (5.19), la cuál, no es más que la definición de distancia Euclídea pero elevado al cuadrado, con el objetivo de colocar una diferencia mayor en los datos que se encuentran más alejados entre sí. El problema es que este tipo de distancia no cumple con la desigualdad triangular, por lo que no puede ser definida como función de disimilitud. A pesar de esto, es utilizada frecuentemente en problemas de optimización en los cuales, únicamente atendemos a la distancia entre los datos. Este tipo de distancia se podrá aplicar perfectamente a nuestro caso, llegando a implementarse en el sistema hardware final en caso de ofrecer buenos resultados de clasificación, debido a su mayor eficiencia computacional.

$$D_{xy}^2 = (x - y)^2 \quad (5.19)$$

Por último, la aplicación del espacio Euclídeo unidimensional al resto de distancias, definidas matemáticamente en el Cuadro 5.1, resultarán en la definición dada por la ecuación 5.18. Demostramos este resultado en la ecuación 5.20:

$$D_{xy} = \left( \sum_{l=1}^1 |x_l - y_l|^n \right)^{1/n} = \sqrt{(x - y)^n} = x - y \quad (5.20)$$

En concreto, podemos ver como para cualquier caso de la distancia de *Minkowski*, ya sea, Euclídea, *Manhattan* o *Chebyshev*, la distancia será la mostrada en la ecuación

## 5. INVESTIGACIÓN

---

5.20. La distancia Euclídea, en el caso unidimensional, será la única distancia medible. En el caso de la distancia *Chebychev*, si quisieramos calcular la distancia entre dos puntos, utilizaríamos la máxima distancia medible, en cualquiera de nuestros ejes. Debido a que únicamente dispondremos de un eje, la distancia *Chebychev* se reducirá a la distancia Euclídea. Por el contrario, en el caso de la distancia *Manhattan*, tal como hemos explicado anteriormente, la distancia entre dos puntos es la suma de las diferencias absolutas de sus coordenadas. En conclusión, únicamente dispondremos de tres medidas de disimilitud en el espacio Euclídeo unidimensional, la distancia Euclídea, la distancia Euclídea al cuadrado y la distancia de *Mahalanobis*. Además de estas medidas de disimilitud, se utilizará la similitud del coseno como medida de similitud.

### 5.8. Algoritmos de *Clustering*

En la siguiente sección se van a introducir todos los algoritmos para la clasificación de nuestras señales, los cuales se implementarán posteriormente mediante el software MATLAB.

Tal y como se describió en la sección 5.5 la propia definición de *Clustering* es quizás un poco imprecisa y ambigua, y es difícil llegar a una definición más rigurosa. Por lo que hay diversas dificultades en la clasificación de las señales mediante los algoritmos de *Clustering*. Uno de los problemas más básicos es que tanto la similitud como la disimilitud se confunden en muchos casos. Matemáticamente hablando, la similitud o proximidad no es una relación transitiva, mientras que la relación entre clusteres es una relación de equivalencia, en particular, una relación transitiva. Concretamente, puede darse el caso en el que haya una larga secuencia de datos,  $x_1, \dots, x_m$  en el que cada  $x_i$  sea muy similar a sus vecinos,  $x_{i-1}$  y  $x_{i+1}$ , pero a su vez  $x_1$  sea muy diferente a  $x_m$ . Sin embargo, en este caso, nos encontramos con que elementos diferentes ( $x_1$  y  $x_m$ ) comparten el mismo clúster, violando de esta manera el segundo requisito. Para ilustrar este ejemplo, suponemos que queremos agrupar los puntos de la siguiente Figura 5.13 en dos clusteres.



Figura 5.13: Datos representados mediante dos líneas horizontales. [21]

Si para clasificar estos datos, utilizamos un algoritmo que no hace hincapié en separar los puntos más cercanos (por ejemplo, el algoritmo *Single Linkage* que se describirá en la sección 5.8.1) este algoritmo clasificará nuestros datos separandolos horizontalmente en dos líneas, tal y como podemos ver en la Figura 5.14

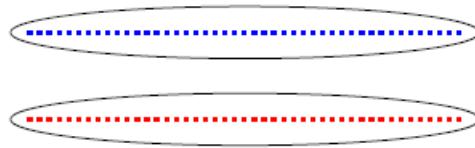


Figura 5.14: Datos clasificados horizontalmente mediante el algoritmo jerárquico *Single Linkage*. [21]

Por el contrario, si utilizamos un método que hace hincapié en que los puntos más lejanos no estén en el mismo grupo (por ejemplo, el algoritmo *2-means* que será descrito más adelante) clasificará nuestros datos dividiéndolos verticalmente entre la mitad derecha e izquierda.

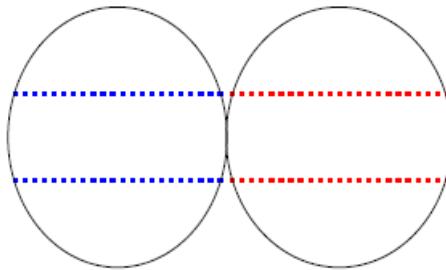


Figura 5.15: Datos clasificados verticalmente mediante el algoritmo *2-means*. [21]

Por lo que en función de nuestro tipo de datos, habrá algoritmos que nos ofrezcan mayor eficacia que otros. Antes de proceder a implementarlos en MATLAB y proceder a la clasificación de nuestras señales vamos a proceder a describir los algoritmos de clasificación utilizados. En la sección 5.5 vimos como principalmente hay dos grupos de tipos de algoritmos de *Clustering*: *Clustering* particional y jerárquico. Comenzaremos describiendo los principales algoritmos del grupo jerárquico para posteriormente explicar los pertenecientes al *Clustering* particional.

## 5. INVESTIGACIÓN

---

### 5.8.1. Algoritmos jerárquicos

Por una parte nos encontramos con los **algoritmos jerárquicos divisivos**, es decir, aquellos que comienzan asignando a todos los datos el mismo grupo, para posteriormente proceder a realizar las divisiones del mismo según criterios de similitud hasta que tengamos grupos formados por un único punto. Pero este tipo de algoritmos tienen la problemática de ser computacionalmente más complejos que los algoritmos aglomerativos, debido a que necesitamos un segundo algoritmo de *Clustering* como “subrutina” para proceder a su clasificación. [1]

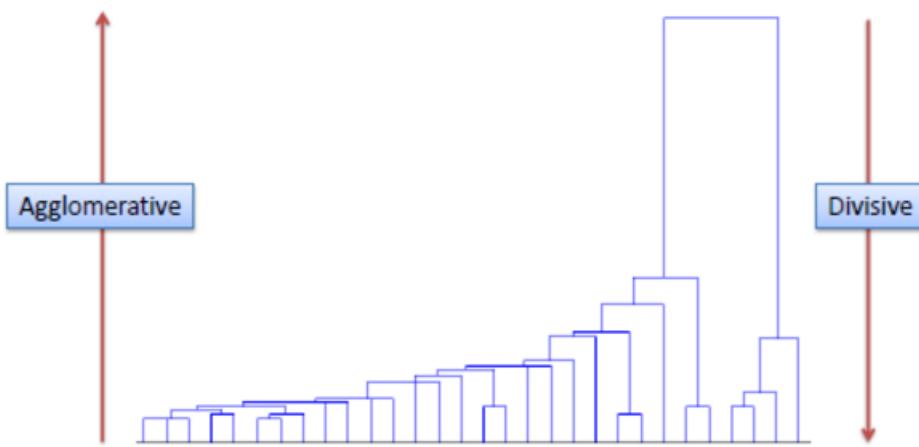


Figura 5.16: Dendograma que muestra la diferencia entre los algoritmos jerárquicos *divisivos* y *aglomerativos*.

Por otra parte, nos encontramos con los **algoritmos jerárquicos aglomerativos**, en los cuáles nos centraremos en el presente proyecto, debido a su menor coste computacional y su posible implementación hardware. Este tipo de algoritmos comienzan asignando a cada punto un clúster. A continuación, se combinan los clusteres en función de la “cercanía” de unos a otros, usando alguna de las definiciones estudiadas anteriormente de distancia. La combinación finaliza cuando una combinación más de clusteres conducirá a clusteres indeseables por alguna razón. Por ejemplo, deberemos de terminar cuando tengamos un número predeterminado de clusteres, o utilizar una medida de la compacidad de los grupos.

Para comenzar asumimos que trabajaremos en el espacio Euclídeo. Esto nos permitirá representar cada clúster por su centroide o media de los puntos de cada clúster.

**Algoritmo 5.1** Algoritmo jerárquico de *Clustering* basado en la unión (*Linkage-based Clustering*)

---

1. Dados: un conjunto de datos  $X = \{x_1, \dots, x_n\}$  y una función de distancia  $dist(c_1, c_2)$ .
  2. Asignamos a cada punto un grupo y su correspondiente centroide:  $c_i = \{x_i\}$ .
  3. Calculamos las distancias entre grupos, utilizando como medida sus centroides  $dist(c_i, c_j)$  para todo  $i, j$  pertenecientes a  $X$ ,  $i \neq j$ ,  $i, j < N$ .
  4. Creamos nuevos grupos, resultantes cada uno de la unión de los dos grupos con mayor similitud, según la distancia elegida. En cada nueva iteración el número de grupos se va reduciendo
  5. Volvemos a (3) y volvemos a realizar los pasos hasta que según el criterio utilizado, sea necesario finalizar la clasificación.
- 

Realizando la clasificación del algoritmo 5.8.1, necesitamos definir dos parámetros para terminar de implementar nuestro algoritmo. Estos dos parámetros son: en primer lugar, la **distancia** elegida para realizar las mediadas de disimilitud entre clusteres, y en segundo lugar, debemos decidir el **criterio de parada** de la unión de grupos. Recalcar que uno de los datos de entrada que le introducimos a un algoritmo de clasificación, es la distancia entre puntos, por lo que hay varias formas de extender  $d$ , como una medida de la distancia entre los diferentes grupos. Las más comunes son:

1. *Single Linkage Clustering*: en este tipo de clasificación la distancia entre clusteres se define como la mínima distancia entre miembros de los dos clusteres, es decir:

$$D(A, B) = \min\{d(x, y) : x \in A, y \in B\} \quad (5.21)$$

2. *Average Linkage Clustering*, en este tipo de clustering, la distancia entre dos clusteres se define como el promedio de la distancia entre un punto perteneciente a un clúster, y otro punto perteneciente al otro.

$$D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y) \quad (5.22)$$

## 5. INVESTIGACIÓN

---

3. *Max Linkage Clustering*, la distancia entre dos clusteres se define como la distancia máxima entre sus elementos, es decir, la máxima distancia entre los puntos más lejanos de cada grupo.

$$D(A, B) = \max\{d(x, y) : x \in A, y \in B\} \quad (5.23)$$

Si no empleamos ninguna regla para parar nuestra clasificación, el resultado del algoritmo se puede describir por un *dendograma* de clasificación: esto es, un árbol de subconjuntos de dominio, el cuál tiene los conjuntos formados por un único punto en su parte inferior, hojas, mientras que el dominio completo se encuentra en su raíz. Por ejemplo, si tenemos un conjunto de datos  $X = \{a, b, c, d, e\} \subset \mathbb{R}^2$ , utilizando la distancia Euclídea, representados a la izquierda de la Figura 5.17, el dendrograma resultante se representará a la derecha de la Figura.

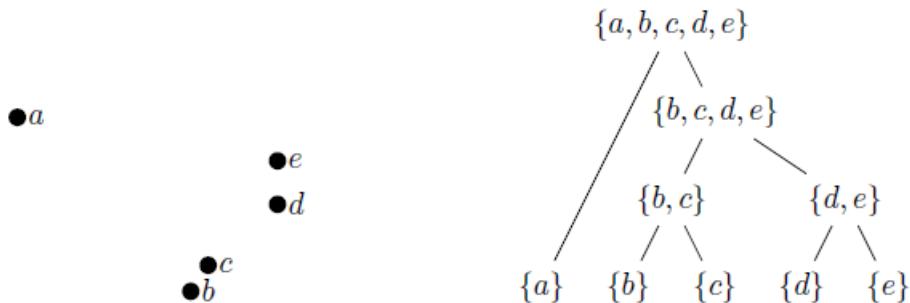


Figura 5.17: Representación de datos en el espacio y dendrograma (derecha) resultante de la clasificación de los mismos mediante *Clustering* jerárquico. [21]

El algoritmo *Single Linkage* está estrechamente relacionado con el **algoritmo de Kruskal**, con el objetivo de encontrar el árbol de expansión mínimo en un grafo ponderado. En efecto, considere el grafo completo cuyos vértices son elementos de  $X$  y el peso de un borde  $(x, y)$  es la distancia  $d(x, y)$ . Cada unión de dos clusteres realizada por el algoritmo *Single Linkage* corresponde a la elección de un borde en el grafo mencionado. También es posible demostrar que el conjunto de aristas del algoritmo forma un árbol de expansión mínima.

Si se quisiera dividir el dendrograma en particiones (clusteres) necesitaríamos emplear un criterio de parada. Los criterios de parada más comunes incluyen:

- **Número prefijado de clusteres:** introduciendo el número de grupos que deseamos,  $k$ , nuestro algoritmo finalizará tan pronto como el número de clusteres sea  $k$ .
- **Distancia de la cota superior:** establecemos un parámetro  $r \in \mathbb{R}_+$ . El algoritmo dejará de funcionar cuando todas las distancias entre clusteres sean más grandes que  $r$ . Además, podemos fijar  $r$  para que sea  $\alpha \max\{d(x, y) : x, y \in X\}$  para una constante  $\alpha < 1$ . En ese caso el criterio de parada se denomina distancia escalada de la cota superior (*scaled distance upper bound*).

Debido a que nuestros datos de la señal aECG están formados por la unión de otras tres señales: mECG, fECG y ruido, utilizaremos para nuestros datos un número prefijado de clusteres, siendo este  $k = 3$ .

Pero los algoritmos jerárquicos clásicos carecen de robustez, por lo que son sensibles al ruido y a los datos en los extremos (*outliers*). Una vez que un dato es asignado a un clúster, este no podrá ser considerado de nuevo, lo que significa que este tipo de algoritmos no son capaces de corregir posibles errores realizados anteriormente en la clasificación. Otro de los problemas de los algoritmos jerárquicos es su tendencia a formar formas esféricas, en la que se distorsiona la estructura jerárquica normal.

Atendiendo a su eficiencia, nos encontramos con que el algoritmo jerárquico no es muy eficiente. En cada paso, necesitamos calcular las distancias entre cada par de clusteres, para encontrar los que proporcionan una mejor unión entre ellos. El paso inicial consume un tiempo de  $O(n^2)$ , pero los pasos subsecuentes tardarán un tiempo proporcional a  $(n - 1)^2$ ,  $(n - 2)^2$ , .etc. La suma de los cuadrados hasta llegar a  $n$  es  $O(n^3)$ , este algoritmo es cúbico. Por lo que únicamente podrá ser implementado para pequeñas bases de datos.

Sin embargo, hay una implementación más eficiente para este algoritmo:

1. Comenzamos calculando las distancias entre todos las parejas de puntos, por lo que tardaremos  $O(n^2)$ .
2. Añadimos los pares y sus distancias a una cola de prioridad, por lo que siempre podremos encontrar la menor distancia en un solo paso. Esta operación consume un tiempo de  $O(\log n)$ .
3. Cuando decidimos unir dos clusteres,  $C$  y  $D$ , eliminamos todas las entradas de la cola de prioridad que involucran a alguno de estos dos clusteres; esto requiere un tiempo de trabajo de  $O(n \log n)$  ya que en la mayoría necesitamos

## 5. INVESTIGACIÓN

---

realizar  $2n$  eliminaciones, y el tiempo de eliminación de la cola de prioridad puede implementarse en  $O(\log n)$ .

4. Por último, calculamos todas las distancias entre el nuevo clúster y los restantes.

Este cálculo conlleva un tiempo de  $O(n \log n)$ , debido a que hay al menos  $n$  entradas para ser insertadas en la cola de prioridad, y esa inserción, de todas las entradas, conllevará un tiempo de  $O(\log n)$ .

Debido a que los dos últimos pasos se ejecutarán por lo menos  $n$  veces, y los dos primeros pasos se ejecutarán solo una vez, el tiempo total que empleará nuestro algoritmo será  $O(n^2 \log n)$ . Este tiempo es mejor que el tiempo inicial, sin aplicar las mejoras, el cuál era  $O(n^3)$ , pero aún sigue resultando un tiempo elevado, por lo que no podrá ser utilizado en grandes bases de datos.

### 5.8.2. Algoritmos particionales

Tal y como se describió en la sección 5.5 los algoritmos de *Clustering* particionales son aquellos que asignan a cada dato un único grupo. Definimos una función de perdidas sobre un conjunto de posibles grupos y el objetivo del algoritmo de *Clustering* es encontrar la partición con menores perdidas. Bajo este paradigma, el problema de la clasificación se convierte básicamente en un **problema de optimización**. La función de perdidas es una función que se define mediante unos parámetros de entrada  $(X, d)$  y una solución de clasificación  $C = (C_1, \dots, C_k)$ . Dada una función objetivo o de pérdidas, la cuál denotamos por  $G$ , el objetivo del algoritmo de *Clustering* se define como, encontrar, para una entrada dada  $(X, d)$ , un *Clustering*  $C$  tal que  $G((X, d), C)$  sea minimizado. Para alcanzar este objetivo es necesario aplicar algoritmos de búsqueda apropiados.

Sin embargo, cuando se habla por ejemplo, del *Clustering k-means*, nos referimos principalmente al algoritmo, en vez de la correspondiente solución exacta del problema de minimización. Muchos de estos algoritmos necesitan como parámetro de entrada el número de clusteres,  $k$ . En la práctica es deseable que el usuario escoja el parámetro  $k$  que mejor se ajuste al problema.

A continuación, vamos a comenzar describiendo algunas de las funciones objetivo:

- **Función objetivo *k-means*:** esta es una de las funciones objetivas de *Clustering* más conocidas. En *k-means* los datos son divididos en diferentes conjuntos  $C_1, \dots, C_k$  donde cada  $C_i$  se representa por un centroide  $\mu_1$ . Se asume por defecto que la entrada  $X$  pertenece a un espacio métrico  $(X', d)$  (por lo que  $X \subseteq X'$ ) y

los centroides son miembros de  $X'$ . La función objetivo *k-means* mide la distancia al cuadrado entre cada punto en  $X$  y el centroide de su clúster. El centroide de  $C_i$  se define como:

$$\mu_i(C_i) = \underset{\mu \in X'_i}{\operatorname{argmin}} \sum_{x \in C_i} d(x, \mu)^2 \quad (5.24)$$

Por lo que la función objetivo *k-means* se puede escribir como:

$$G_{k-means}((X, d), (C_1, \dots, C_k)) = \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu_i(C_i))^2 \quad (5.25)$$

Que a su vez, puede reescribirse como:

$$G_{k-means}((X, d), (C_1, \dots, C_k)) = \min_{\mu_1, \dots, \mu_k \in X'} \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu_i)^2 \quad (5.26)$$

- **Función objetivo *k-medoids*:** similar a la función objetivo *k-means*, excepto que en este caso necesitamos que el centroide sea un punto propio del propio clúster. Esta función se define como:

$$G_{k-medoids}((X, d), (C_1, \dots, C_k)) = \min_{\mu_1, \dots, \mu_k \in X} \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu_i)^2$$

- **Función objetivo *k-median*:** similar a la función *k-medoids*, excepto que la “distorsión” entre los datos y el centroide del clúster se miden por distancia, en vez de por distancia al cuadrado:

$$G_{k-median}((X, d), (C_1, \dots, C_k)) = \min_{\mu_1, \dots, \mu_k \in X} \sum_{i=1}^k \sum_{x \in C_i} d(x, \mu_i)$$

- **Función objetivo *fuzzy c-means*:** similar a la función *k-means*, excepto con la única diferencia de que ahora tenemos un grado de pertenencia de cada punto a cada grupo,  $\omega_{ij}^m$ :

$$G_{fuzzy\,C-means}((X, d, \omega), (C_1, \dots, C_k)) = \min_{\mu_1, \dots, \mu_k \in X} \sum_{i=1}^k \sum_{j=1}^C \omega_{ij}^m d(x, \mu_i(C_i))^2 \quad (5.27)$$

## 5. INVESTIGACIÓN

---

Todas las funciones objetivo anteriores son funciones basadas en centroides. Su solución es determinada por un conjunto de centroides. Generalmente, los algoritmos basados en los centroides son determinados por una función monótona  $f : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  y definidos como:

$$G_f((X, d), (C_1, \dots, C_k)) = \min_{\mu_1, \dots, \mu_k \in X'} \sum_{i=1}^k \sum_{x \in C_i} f(d(x, \mu_i))$$

donde  $X'$  es, o bien  $X$  o bien un superconjunto de  $X$ .

A continuación vamos a describir los algoritmos que pertenecen al tipo de algoritmos particionales.

### 5.8.2.1. Algoritmo *k-means*

La función objetivo *k-means* es muy popular en aplicaciones prácticas de *Clustering*. Sin embargo, resulta que encontrar la solución óptima a esta ecuación es a menudo inviable (el problema es un problema de tipo NP-hard, quiere decir al menos tan complejo como NP, pero no necesariamente en NP).<sup>6</sup> Como alternativa, se implementa un algoritmo más simple para encontrar la solución, por lo que a menudo el término *k-means* se refiere al resultado de este algoritmo en vez de a la propia función objetivo que minimiza el coste objetivo en *k-means*. A continuación describimos el algoritmo *k-means*. En él podemos aplicar algunas de las distancias vistas anteriormente, aunque por defecto se utiliza la distancia Euclídea.

---

#### Algoritmo 5.2 Algoritmo *k-means*

---

1. **Entrada:** un conjunto de datos  $X = \{x_1, \dots, x_n\}$  y un número de clusteres,  $k$ .
  2. **Inicialización:** elegimos aleatoriamente  $k$  centroides de cada clúster  $C = c_1, \dots, c_k$ .
  3. Para cada  $i \in \{1, \dots, k\}$ , establecer el clúster  $C_i$  para que esté en el conjunto de los puntos en  $X$  que están mas cercanos a  $c_i$  de lo que están a  $c_j$  para todo  $j \neq i$ .
  4. Para cada  $i \in \{1, \dots, k\}$ , establecer  $c_i$  para que sea el centroide de todos los puntos en  $C_i$  :  $c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$
  5. Repetir los pasos 3 y 4 hasta que  $C$  deje de cambiar.
-

### 5.8.2.2. Algoritmo *k-means++*

En la práctica, se escogen los centroides aleatoriamente de  $X$ . La idea es que los pasos 2 y 3 garantizan que nuestra función objetivo se minimiza, por lo que el algoritmo va realizando mejoras locales a un clúster hasta que no es posible hacer más.

A pesar de que este algoritmo es más eficiente que implementar la propia función objetivo *k-means* y minimizarla, sigue siendo complejo de resolver. Para solucionar esto se utiliza el algoritmo *k-means++*, el cuál utiliza un proceso heurístico para encontrar los centroides para el algoritmo *k-means*. De acuerdo con Arthur y Vassilvitskii [4], el algoritmo *k-means++* mejora el tiempo de funcionamiento del algoritmo de Lloyd, y la calidad de la solución final.

---

#### **Algoritmo 5.3** Algoritmo *k-means++*

---

1. **Entrada:** un conjunto de datos  $X = \{x_1, \dots, x_n\}$ , un número de clusteres,  $k$ .
  2. **Inicialización:** elegimos aleatoriamente una observación de los datos,  $X$ . La observación elegida será nuestro primer centroide, denotado por  $c_1$ .
  3. Calculamos las distancias desde cada observación o dato hasta  $c_1$ . La distancia entre  $c_j$  y cada observación  $m$  se denota como  $d(x_m, c_j)$ .
  4. Seleccionamos el nuevo centroide  $c_2$  aleatoriamente del conjunto de datos  $X$  con probabilidad: 
$$\frac{d^2(x_m, c_1)}{\sum_{j=1}^n d^2(x_j, c_1)}$$
  5. Para elegir el centroide j:
    1. Calculamos las distancias desde cada observación hasta cada centroide, y asignamos cada observación a su centroide más cercano.
    2. Para  $m = 1, \dots, n$  y  $p = 1, \dots, j - 1$ , seleccionamos el centroide  $j$  aleatoriamente de  $X$  con probabilidad: 
$$\frac{d^2(x_m, c_p)}{\sum_{\{h; x_h \in C_p\}} d^2(x_h, c_p)}$$
 donde  $C_p$  es el grupo de observaciones cercanas al centroide  $c_p$  y  $x_m$  pertenece a  $C_p$ . Esto es, seleccionar cada centroide subsecuente con una probabilidad proporcional a la distancia del mismo con el centroide elegido anteriormente.
  6. Repetimos el paso 5 hasta que se hayan elegido los  $k$  centroides.
-

## 5. INVESTIGACIÓN

---

Tal y como observamos en el algoritmo *k-means++*, la principal diferencia con el algoritmo *k-means*, es que, mientras que *k-means* inicializa aleatoriamente todos los centroides al principio (Paso 2) el algoritmo *k-means++* inicializa únicamente un centroide y después, calcula todos los centroides asignándoles una probabilidad. Esta probabilidad es proporcional a la distancia, es decir, en el denominador tenemos la sumatoria de todas las distancias, de todos los puntos respecto con el primer centroide, mientras que en el numerador tenemos la distancia de ese punto exacto con el mismo centroide. Por lo que, a mayor distancia con respecto al primer centroide asignado, mayor probabilidad tendremos de tener otro nuevo centroide, que representará a un nuevo grupo. Este procedimiento es mucho más intuitivo que el primero, lo que dará lugar a menor número de iteraciones para alcanzar la solución óptima, necesitando un menor coste computacional. Es por ello, que en nuestra implementación en MATLAB, utilizaremos el algoritmo *k-means++* en lugar del propio *k-means*.

A pesar de encontrarnos con numerosos algoritmos en el campo del *Clustering*, el algoritmo *k-means* es uno de los más eficientes, robustos y sencillos de implementar. Por lo que su uso se ha extendido en diversos campos científicos. Además, fue clasificado como uno de los 10 mejores algoritmos de clasificación en [26].

### 5.8.2.3. Algoritmo *k-medoids*

El algoritmo *k-medoids* es un método de partición utilizado comúnmente en dominios que requieren robustez, tanto para el ruido como para datos que se encuentran en las fronteras.

Es muy similar al algoritmo *k-means*, y el objetivo de ambos algoritmos es dividir un conjunto de observaciones en  $k$  grupos o clusteres por lo que estos subconjuntos minimizan la suma de las distancias entre las medidas y el centro del clúster. En el algoritmo *k-means*, este centro es llamado centroide, el cuál es la media de las medidas del propio subconjunto. En el algoritmo *k-medoids*, este centro es un propio punto del subconjunto, llamado medoide.

Este algoritmo devuelve medoides que son puntos pertenecientes a los clusteres. Esto permite utilizar el algoritmo en situaciones donde la media de los datos no exista sin los propios datos en sí. Esta es la principal diferencia entre *k-means* y *k-medoids*, que los centroides dados por *k-means* no tienen por qué ser puntos propios del clúster. Por lo que este algoritmo, *k-medoids*, es útil para clasificar datos en donde el concepto de “media” sea imposible de definir o interpretar. Por lo tanto, su algoritmo será prácticamente igual que el algoritmo *k-means*, con un mayor coste computacional que este, por lo que a pesar de ser más eficiente para algunos tipos de datos,

este implicará mayor tiempo de ejecución. Para ello utilizaremos la técnica *Partitioning Around Medoids* (PAM), ayudándonos de la inicialización de los medoides del inicio del algoritmo *k-means++*. Este algoritmo puede producir mejores resultados que el *k-means++* en algunas situaciones, pero con el problema de un mayor coste computacional. .

---

**Algoritmo 5.4** Algoritmo *k-medoids* con la técnica PAM
 

---

1. **Entrada:** un conjunto de datos  $X = \{x_1, \dots, x_n\}$ , un número de clusteres,  $k$ .
  2. **Inicialización:** elegimos aleatoriamente una observación de los datos,  $X$ . La observación elegida será nuestro primer medoide, denotado por  $m_1$
  3. Calculamos las distancias desde cada observación o dato hasta  $m_1$ . La distancia entre  $m_j$  y cada observación  $m$  se denota como  $d(x_m, m_j)$ .
  4. Seleccionamos los nuevos medoides  $m_j$  aleatoriamente del conjunto de datos  $X$  con probabilidad: 
$$\frac{d^2(x_m, m_1)}{\sum_{j=1}^n d^2(x_j, m_1)}$$
  5. En cada clúster comprobamos cada punto como posible medoide, comprobando si la suma de las distancias intra-clúster se vuelven menores utilizando este punto como medoide. Si se cumple, este punto se define como nuevo medoide del grupo. Por lo que cada punto es asignado a un clúster en función de la distancia de cada punto al medoide más cercano.
  6. El algoritmo continua ejecutándose hasta que los medoides del mismo no cambien.
- 

#### 5.8.2.4. Algoritmo *fuzzy c-means*

Los algoritmos estudiados anteriormente asignaban cada punto a un único clúster. Al contrario que estos algoritmos, el algoritmo *fuzzy c-means* asigna a cada dato una probabilidad de pertenecer a todos los grupos, por lo que realizamos una partición difusa de los datos.<sup>[2]</sup> Para ello, este algoritmo proporciona un método que muestra cómo agrupar los datos con una probabilidad de pertenecer a diferentes grupos. A diferencia del algoritmo *k-means* el algoritmo *fuzzy c-mean* minimiza la varianza interna del grupo (*intra-cluster*) pero tiene el mismo problema que *k-means*, es decir, en la optimización de la función objetivo estamos ante un mínimo local, no absoluto, por lo que el resultado dependerá de elección inicial de los centroides. Matemáticamente:

## 5. INVESTIGACIÓN

---

Sea  $X = \{x_1, \dots, x_n\}$  un subconjunto del espacio Euclídeo, y un número de clusteres,  $k$ . Una partición difusa de  $X$  en  $k$  clusteres o grupos, es una tupla de  $k$  funciones de pertenencia  $\mu = (\mu_1, \dots, \mu_k)$  tal que:

1.  $0 \leq \mu_i(x) \leq 1, \quad \forall i = 1, \dots, k$
2.  $0 < \sum_{j=1}^n \mu_i(x_j) < 1, \quad \forall i, \forall j = 1, \dots, n$
3.  $\sum_{i=1}^c \mu_i(x_j) < 1, \quad \forall j.$

Las particiones difusas se representan como una matriz, asociando cada fila a uno de los  $k$  grupos y cada columna a uno de los elementos de  $X$ , de tal forma que el valor en la fila  $i$  y la columna  $j$  indique la pertenencia del elemento  $j$  al grupo  $i$ . Es decir la matriz difusa se define como:

$$W = w_{i,j} \in [0, 1], \quad i = 1, \dots, k, \quad j = 1, \dots, n \quad (5.28)$$

---

### **Algoritmo 5.5 Algoritmo *fuzzy c-means***

---

1. **Entrada:** un conjunto de datos  $X = \{x_1, \dots, x_n\}$ , un número de clusteres,  $k$ .
  2. **Inicialización:** inicializamos la matriz  $W = [w_{ij}]$ ,  $W^{(0)}$ .
  3. En el paso- $k$ : calculamos el vector de centroides  $C^{(k)} = [c_j]$  con la matriz  $W^{(k)}$
  4. Actualizamos la matriz  $W^{(k)}$ ,  $W^{(k+1)}$ : siendo  $m$  un coeficiente de ponderación  $1 \leq m < \infty$
- $$w_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{2/(m-1)}}$$
5. Sí  $\|W^{(k+1)} - W^{(k)}\| < \varepsilon$ , es decir, si la diferencia entre la matriz anterior y la actual es menor que un cierto criterio  $\varepsilon$ , con valores entre 0 y 1, nuestro algoritmo finalizará.
- 

#### 5.8.2.5. Algoritmo de conjuntos dominantes

Como último algoritmo utilizado para clasificar nuestras señales aECG utilizamos el algoritmo ideado por Marcelo Pelillo. Para clasificar nuestra señal aECG haremos uso del *Clustering* basado en teoría de grafos, conocido como *dominant set Clustering*. [16]

Primeramente, convertiremos la tarea de clasificación del ECG en un problema de clasificación de grafos. Para lograr tal fin, hacemos de nuestros datos,  $G = (V, w)$  un grafo completo, ponderado en sus ejes, donde  $V = \{1, \dots, n\}$  es el conjunto de vértices, cada vértice  $i \in V$  relacionado con cada dato,  $x_i$ , y con  $w : V \times V \rightarrow \mathbb{R}_+$  siendo una función de similitud, en nuestro caso, la distancia Euclídea. En la Figura 5.18, podemos ver la clasificación de los vértices en función de la similitud con el resto de vértices vecinos.

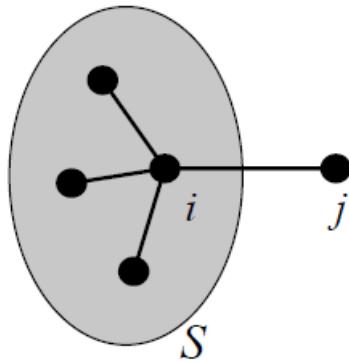


Figura 5.18: Representación de los vértices pertenecientes a un conjunto dominante,  $S$ , marcados dentro de la elipse, con respecto a un vértice externo a este grafo,  $V(j)$ . [16]

El grado medio se define como:

$$awdeg_s(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij} \quad (5.29)$$

Además, si  $j \notin S$  podemos definir:

$$\phi_S(i, j) = a_{ij} - awdeg_s(i) \quad (5.30)$$

Intuitivamente,  $\phi_S(i, j)$  mide la similitud entre los vértices  $j$  e  $i$ , con respecto a la similitud (media) entre el vértice  $i$  y sus vecinos en  $S$ .

Por lo que, siendo  $S \subseteq V$  un conjunto no nulo de vértices, e  $i \in S$ . El peso de  $i$  con respecto a  $S$  se define como:

$$w_S(i) = \begin{cases} 1 & \text{si } |S| = 1 \\ \sum_{j \in S - \{i\}} \phi_{S - \{i\}}(i, j) w_{S - \{i\}}(j) & \text{resto} \end{cases} \quad (5.31)$$

## 5. INVESTIGACIÓN

---

### Algoritmo 5.6 Algoritmo *dominant sets*

---

1. **Entrada:** un grafo completo,  $G = (V, w)$  así como un número de clusteres,  $k$ .
  2. Calculamos el peso de cada vértice,  $V_i$  con respecto a los demás vértices:  

$$w_S(i) = \begin{cases} 1 & \text{si } |S| = 1 \\ \sum_{j \in S - \{i\}} \phi_{S - \{i\}}(i, j) w_{S - \{i\}}(j) & \text{resto} \end{cases}.$$
  3. Por lo que, si  $w_S(i) > 0$ , este vértice pertenecerá a nuestro grafo  $S$ .
  4. Una vez obtenido el grafo o conjunto dominante, los vértices pertenecientes a este se eliminaran del grafo, y se volverá a calcular el peso para el resto de vértices del grafo, creando un nuevo conjunto dominante. Así hasta  $k$  iteraciones.
  5. Si quedasen vértices sin asignar a grupos dominantes, estos se asignarían a los grupos más cercanos en función a la distancia de disimilitud utilizada.
- 

Un ejemplo del calculo de pesos o distancias entre vértices puede observarse en la Figura 5.20.

Por lo que, el peso total de  $S$  se define como:

$$W(S) = \sum_{i \in S} w_S(i) \quad (5.32)$$

Intuitivamente,  $w_S(i)$  nos ofrece una medida de la similitud total entre el vértice  $i$  y el resto de vértices pertenecientes al grafo  $S - \{i\}$  con respecto a la similitud total entre vértices de  $S - \{i\}$ . En la Figura 5.19 observamos vértices pertenecientes a dos conjuntos dominantes diferentes.

En el marco de referencia de este algoritmo, los clusteres o grupos se caracterizan en términos de dominios dominantes, subconjuntos de vértices para grafos ponderados en los ejes que nos aseguran las dos propiedades fundamentales de un clúster: coherencia interna y incoherencia externa. Desde una perspectiva combinatoria, un conjunto dominante se define como un subconjunto de vértices  $D \subseteq V$  satisfaciendo las siguientes condiciones:

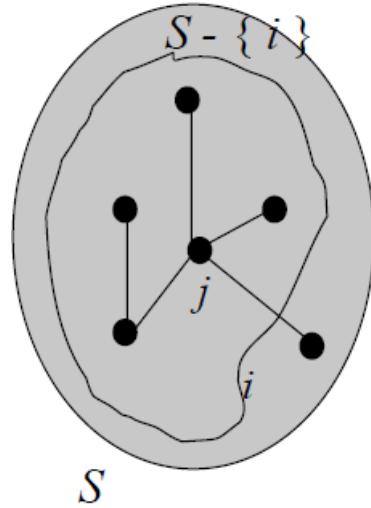


Figura 5.19: Representación de los vértices pertenecientes a dos grafos distintos,  $S$  y  $S - \{i\}$ .[16]

$$w_S(i) > 0, \quad \forall i \in D$$

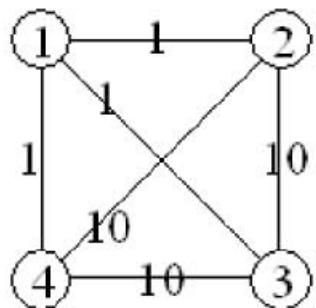
$$w_{S \cup \{i\}}(i) < 0, \quad \forall i \in V \setminus D$$

Donde,  $w_S(i)$  es una caracterización recursiva de la distancia asignada a cada elemento  $i \in V$  con respecto del conjunto  $D \subseteq V$ .

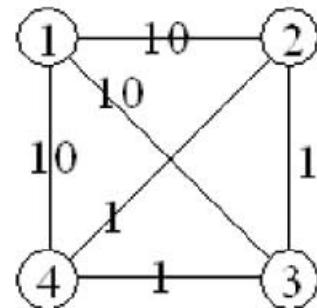
Para clasificar por tanto nuestros datos en diferentes particiones de grupos, se utilizará una estrategia *peel-off*, es decir, en cada iteración encontraremos el conjunto dominante de vértices, eliminándolos a su vez del grafo, hasta que todos los vértices hayan sido clasificados. Por lo que, si queremos clasificar nuestros datos en 3 grupos, únicamente necesitaremos 3 iteraciones, calculando por tanto 3 conjuntos dominantes.

## 5. INVESTIGACIÓN

---



$$w_{\{1,2,3,4\}}(1) < 0$$



$$w_{\{1,2,3,4\}}(1) > 0$$

Figura 5.20: Ejemplo del valor del peso entre vértices en un grafo. En el grafo de la izquierda, el peso nos indicará que el vértice 1 pertenece al resto de vértices, mientras que en el grafo de la izquierda, el vértice uno no pertenecerá a este grafo. [16]

# Capítulo 6

## Simulación

El presente capítulo tiene por objetivo realizar la clasificación de nuestras señales aECG, clasificando en cada una de ellas mECG, fECG y ruido. Para ello nos hemos ayudado de la herramienta de software MATLAB, en donde hemos implementado todos los algoritmos de *Clustering* descritos anteriormente en el capítulo 5. En este capítulo comenzaremos introduciendo las señales que hemos utilizado, el banco de datos de donde han sido extraídas, etc. para después explicar la metodología seguida para clasificar nuestras señales.

### 6.1. Bases de datos y señales aECG

Antes de proceder a realizar cualquier simulación de la exactitud y eficiencia de los algoritmos era necesario proveer de la “materia prima” sobre la cuál íbamos a trabajar en el presente proyecto. Por lo que, en nuestro caso, necesitábamos de las señales aECG para proceder a su preprocesado y posterior clasificación. Para lograr este objetivo se tuvieron en cuenta varias posibilidades a la hora de adquirir y/o trabajar con las señales aECG. Estas posibilidades eran:

- **Adquisición de las señales aECG por medio de un sistema de adquisición electrónico.** Este tipo de solución es la más lógica para una futura implementación hardware portable, debido a que la finalidad de este proyecto es la creación de un sistema portable que ayude a clasificar las señales en tiempo real sin necesidad de un especialista médico. Pero a su vez, es también la que más problemas daba a la hora de adquirir correctamente una señal sobre la que utilizar nuestro algoritmo. Estos problemas podían ser:
  - Problemas relacionados con los sensores de adquisición utilizados en la adquisición de la señal aECG: es decir, los artefactos utilizados para medir la

## 6. SIMULACIÓN

---

señal del abdomen materno deben de ser de la mayor calidad posible, evitando la introducción de ruido externo a la propia señal debido a artefactos y sensibilidad al movimiento o posición del feto.

- Problemas relacionados con el propio circuito de adquisición de la señal. debido a que este circuito, reprogramable o no, es un prototipo, no es eficiente realizar la clasificación de nuestras señales con un sistema que aún no ha sido verificado como útil para este fin. Por lo que en la etapa de evaluación de los algoritmos de clasificación es conveniente utilizar sistemas de adquisición en fases finales, que hayan demostrado su correcto funcionamiento ante diversos factores internos y/o externos.
  - Problemas relacionados con la propia medición de la señal: en el momento de medir nuestra señal aECG es posible que surjan problemas relacionados con el propio acto de medir la señal, como pueden ser: movimientos bruscos, posición del feto, estado fetal (cuanto mayor es el tiempo de gestación, mayor amplitud poseerá la señal del corazón proveniente del feto, por lo que, nuestra clasificación se podrá hacer de una manera más sencilla), etc.
- **Utilización de bases de datos en las que se encuentren señales aECG.**  
Esta opción ha sido la escogida para desarrollar nuestro proyecto y evaluar la exactitud de nuestros algoritmos, debido a que, las señales han sido tomadas por profesionales en el campo de la salud, con un instrumental adecuado, en el que se ha comprobado que en ninguna de ellas hay más ruido del habitual en una sesión normal de ECG. Además, a la hora de comparar la eficiencia de los algoritmos implementados en este proyecto con las posibles implementaciones de futuros algoritmos, es necesario que ambos se evalúen sobre las mismas señales. Por lo que gracias a la utilización de bases de datos estándares, facilitamos la labor de investigación de los futuros investigadores, que intenten profundizar en la búsqueda de algoritmos que mejoren la exactitud y eficacia de los utilizados en este documento.

Una vez aclarado cuál será la metodología utilizada para proveer a nuestros algoritmos de las señales aECG es necesario aclarar que base de datos en concreto hemos utilizado, así como todas las señales pertenecientes a esta base de datos. Para ello, tras realizar una investigación acerca de las principales bases de datos, nos decantamos por la base de datos perteneciente a la fuente ***PhysioNet.[11]***

### 6.1.1. PhysioNet

PhysioNet es una base de datos de señales fisiológicas creada por la *National Institute of General Medical Sciences* (NIGMS) y el *National Institute of Biomedical Imaging and Bioengineering* (NIBIB). La característica principal de esta base de datos es que todas las señales han sido supervisadas por expertos en el campo de la medicina, además de su constante crecimiento, y la posibilidad de descargar señales de gran calidad en un tamaño reducido y adaptadas para ser trabajadas en diferentes herramientas software, como por ejemplo, en nuestro caso MATLAB.

Esta base de datos incluye señales del EEG, ECG, etc. abarcando diferentes tipos de señales, desde señales adquiridas de pacientes completamente sanos, hasta señales adquiridas directamente de pacientes con una gran variedad de patologías, tales como muerte cardíaca súbita, arritmias cardíacas, desórdenes motores, etc. Por lo que es una base de datos que incluye diferentes tipos de señales, algo muy importante para demostrar la validez de los métodos utilizados, debido a que no es factible utilizar un método de resolución inductiva, que únicamente funcione para un tipo de señal muy específico. Por lo que, en el presente documento se ha trabajado con todas las señales disponibles en esta base de datos que se ajustaban perfectamente a nuestro estudio. Este tipo de señales han sido señales ECG provenientes del abdomen materno, que incluyen tanto los propios latidos del corazón materno, como los fetales y la presencia de ruido en la medición. Además de estas características, nos era necesario que vinieran anotados los complejos QRS, especialmente los fetales (los más difíciles de encontrar). Por lo que, nos centramos en una base de datos que incluía todos los fetales anotados por un especialista en señales ECG.

Para ello, nos servimos de la ayuda del *PhysioBank ATM* el software en el que se encuentra el banco de datos de *PhysioNet*. En él, podremos descargar las señales pertenecientes a la base de datos deseada, además de otras opciones, como por ejemplo, ver la forma de onda de la señal, su intervalo RR, etc. Además, nos permite descargar las señales en diferentes formatos, interesándonos especialmente el formato *.mat*, para nuestro uso en MATLAB de las señales.

La principales bases de datos que hemos utilizado en nuestro estudio han sido:

- ***Abdominal and Direct Fetal Electrocardiogram Database***: esta base de datos contiene señales fECG multicanal, es decir, adquiridas mediante diferentes canales. Las señales se han obtenido de 5 mujeres diferentes, entre 38 y 41 semanas de gestación. Estas grabaciones fueron adquiridas por el Departamento

## 6. SIMULACIÓN

---

| Propiedades                  | Valor   |
|------------------------------|---------|
| Resolución (bits)            | 16      |
| Ancho de Banda (Hz)          | 1 - 150 |
| Frecuencia de muestreo (kHz) | 1       |

Cuadro 6.1: Propiedades de adquisición de la base de datos *Abdominal and Direct Fetal Electrocardiogram* de *PhysioNet*.

de Obstetricia de la Universidad Médica de Silesia, por medio del sistema KOMPOREL para la adquisición y el análisis del fECG. Cada señal está compuesta de cuatro señales diferentes, adquiridas del abdomen materno y del fECG de referencia, medido en la propia cabeza del feto. La configuración de los electrodos, está formada por cuatro electrodos alrededor del ombligo materno, un electrodo de referencia colocado por encima del pubis y un electrodo de referencia de modo común (con una señal a tierra) colocada en la pierna izquierda. Para reducir la impedancia de la piel, se rasparon las áreas bajo los electrodos de *Ag – AgCl* (plata - Cloruro de plata). Además de esta configuración, se colocó un electrodo en el escalpedo fetal.

La adquisición del fECG directo se llevó a cabo mediante un electrodo típico en espiral, utilizado en la monitorización de señales fECG. Las localizaciones de la onda *R* en la señal fECG directa fueron determinadas automáticamente por medio del análisis *online* aplicado por el propio sistema. Una vez anotadas estas ondas, estas fueron comprobadas por un grupo de cardiólogos, marcando correctamente todas las localizaciones de la onda *R* en las señales. Estas marcas están anotadas en las señales, por lo que esto nos ayudará en el siguiente capítulo para calcular la precisión y eficacia de nuestro algoritmo.

Esta base de datos contiene 5 señales, las cuales están designadas con el formato *r0x.edf* donde *x* representa el número designado a cada paquete de señal, *x* = {1, 4, 7, 8, 10}. Cada paquete contiene a su vez diferentes señales, siendo estas:

- Señal directa del escalpedo fetal: *Direct\_1*
- Señales provenientes de los cuatro electrodos colocados en el abdomen materno: *Abdomen\_1*, *Abdomen\_2*, *Abdomen\_3*, *Abdomen\_4*.
- Anotaciones de los complejos QRS fetales: un fichero aparte con formato *.qrs* en el que se incluyen los complejos QRS fetales.

En la Figura 6.1 proveniente del paquete *r01.edf* con una duración de 10 segundos, e pueden observar las señales descritas anteriormente. Los puntos azules,

## 6.1 Bases de datos y señales aECG

---

| Hora     | Fecha    | Direct_1<br>(mV) | Ab_1 (uV) | Ab_2 (uV) | Ab_3 (uV) | Ab_4 (uV) |
|----------|----------|------------------|-----------|-----------|-----------|-----------|
| 00:00:00 | 01/01/11 | 28.700           | -8.900    | 13.300    | 22.000    | 32.500    |
| ...      | ....     | ...              | ...       | ...       | ...       | ...       |

Cuadro 6.2: Señal *r01.edf* del *Non-Invasive Fetal ECG* de *PhysioNet* en formato *.mat*

| Propiedades                  | Valor      |
|------------------------------|------------|
| Resolución (bits)            | 16         |
| Ancho de Banda (Hz)          | 0.01 - 100 |
| Frecuencia de muestreo (kHz) | 1          |

Cuadro 6.3: Propiedades de adquisición de la base de datos *Non-Invasive Fetal ECG* de *PhysioNet*.

así como las líneas azules verticales, designan que en la señal existe un complejo QRS fetal. Utilizaremos estas anotaciones para comparar nuestros candidatos a complejos QRS fetales con los complejos QRS fetales verdaderos, anotados en azul.

- ***Non-Invasive Fetal ECG Database***: esta base de datos contiene 55 señales abdominales multicanal fECG, tomadas de una única paciente entre 21 y 40 semanas de gestación. Las señales tienen duraciones variables, desde minutos hasta semanas. Esta adquisición ha sido realizada por la Escuela Técnica Superior de Ingeniería de la Universidad de Valencia (ETSE). La principal diferencia de estas señales es que en ellas, únicamente vienen anotados los complejos QRS maternos, por lo que esta base de datos no nos es del todo útil para realizar la evaluación del proyecto. Aún así, se utilizará para clasificar en tiempo real los complejos QRS fetales una vez hayamos escogido el mejor algoritmo de clasificación (deducido utilizando la base de datos *Abdominal and Direct Fetal Electrocardiogram*). Utilizaremos algunas señales pertenecientes a esta base de datos, en las cuáles se anotaron los complejos QRS fetales con la presencia de un cardiólogo, tal y como se detalla en [6].

En esta base de datos, las señales se designan con el siguiente formato *ecg-caxyz.edf* siendo *xyz* un número que designa a la señal en concreto. Cada paquete contiene a su vez diferentes señales, las cuales son:

- Señales provenientes del tórax materno: dos señales provenientes de dos electrodos colocados en el torax materno, designadas como *Thorax\_1* y *Thorax\_2*.

## 6. SIMULACIÓN

---

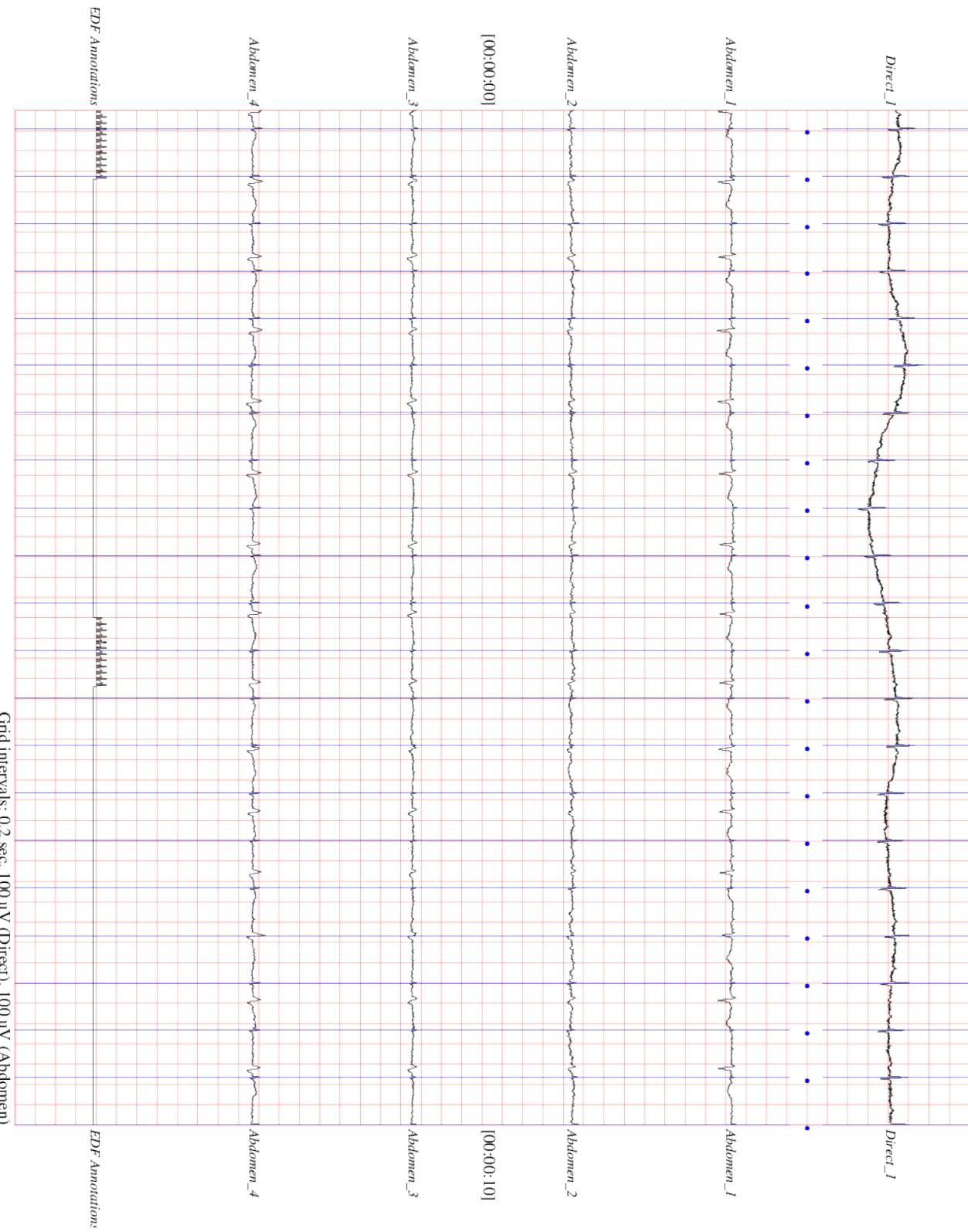


Figura 6.1: Señal *r01.edf* de la base de datos *Abdominal and Direct Fetal Electrocardiogram*

## 6.1 Bases de datos y señales aECG

---

| Hora     | Fecha    | Thorax_1<br>(mV) | Thorax_2<br>(mV) | Ab_1 (uV) | Ab_2 (uV) | Ab_3 (uV) | Ab_4 (uV) |
|----------|----------|------------------|------------------|-----------|-----------|-----------|-----------|
| 23:31:05 | 24/03/04 | 0.951            | 0.260            | 51.987    | 52.049    | 52.049    | 51.942    |
| ...      | ....     | ...              | ...              | ...       | ...       | ...       | ...       |

Cuadro 6.4: Señal *ecgca906.edf* del *Non-Invasive Fetal ECG* de *PhysioNet* en formato *.mat*

- Señales provenientes de los cuatro electrodos colocados en el abdomen materno: *Abdomen\_1*, *Abdomen\_2*, *Abdomen\_3*, *Abdomen\_4*.
- Anotaciones del complejo QRS materno: un fichero aparte con formato *.qrs* en el que se incluyen los complejos QRS maternos.

## 6. SIMULACIÓN

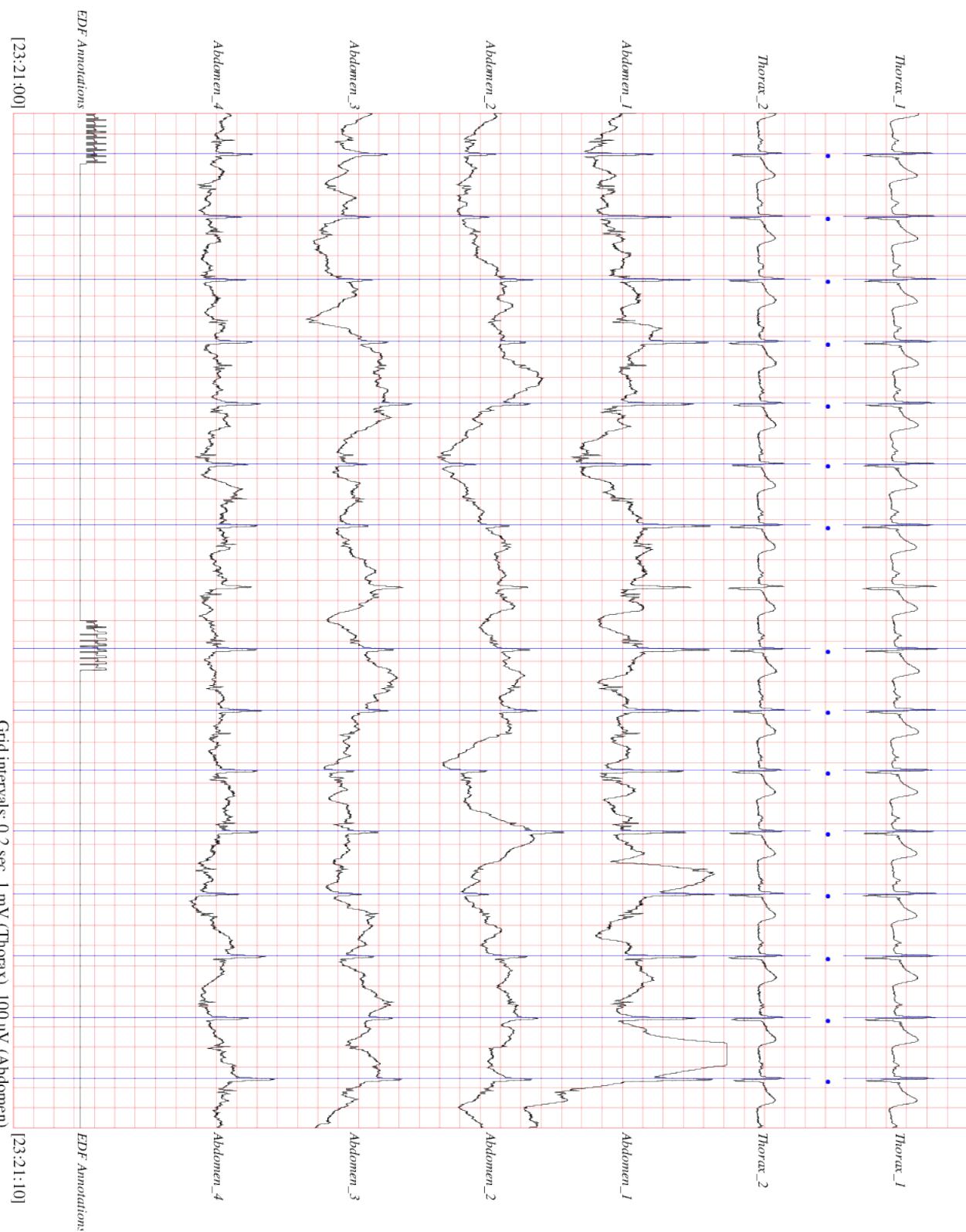


Figura 6.2: Señal *ecgca906.edf* de la base de datos *Non-Invasive Fetal ECG* de *PhysioNet*

## 6.2. Implementación software en MATLAB

En esta sección vamos a exponer el proceso llevado a cabo para clasificar nuestras señales aECG, y obtener los resultados de precisión y exactitud que se detallarán en el siguiente capítulo.

### 6.2.1. Primeros pasos

Antes de comenzar a desarrollar la GUI, era necesario implementar los algoritmos de clasificación detallados en el capítulo 5. Todas las funciones estarán disponibles libremente para la comunidad científica. Con estas funciones, scripts y GUIs pretendemos fomentar la investigación en el campo de las señales fECG, para descubrir nuevos algoritmos que mejoren a los aquí implementados, de cara a una futura implementación hardware portable, que pueda estar disponible para cualquier persona y/o hospital clínico a bajo coste.

A continuación, vamos a ir mostrando las diferentes implementaciones software y resultados gráficos de los algoritmos expuestos en el capítulo anterior. Para ello, antes de proceder a la clasificación, era necesario descargar nuestras señales del repositorio *PhysioBank ATM*, seleccionando el paquete de señal que se quiere descargar, indicando que se descargue también el fichero con las anotaciones de los complejos QRS y descargando estas señales en formato *.mat*. Los ficheros *.mat* descargados, son básicamente una matriz con la misma disposición en columnas que la mostrada en los Cuadros 6.4 y 6.2, conteniendo cada archivo las diferentes señales en cada columna, mientras que en las filas de esta matriz se encuentran las muestras de cada señal, muestreadas a 1 kHz.

Para proceder a la utilización de estas señales en MATLAB era necesario realizar la apertura de las mismas correctamente. Para ello nos ayudamos de la herramienta *PhysioToolkit* desarrollada por el propio *PhysioNet*, la cuál no es más que una librería de diferentes funciones para MATLAB y GNU OCTAVE que nos facilitan la utilización y la representación de todas las señales fisiológicas disponibles en el propio banco de datos de *PhysioNet*. Por lo que antes de procesar nuestra señal, procedíamos a su apertura mediante las funciones disponibles en esta herramienta software. Tener en cuenta, que además de obtener las muestras de la señal en los distintos electrodos, también era necesario obtener los complejos QRS marcados en la señal, por lo que una vez extraída nuestra señal, extraímos el vector que indicaba la posición en la que se encontraban estos complejos.

## 6. SIMULACIÓN

---

Una vez teníamos nuestra señal en el *workspace*, era necesario realizar el preprocesado de la señal, para eliminar el ruido mediante la WT. Por lo que para proceder a realizar el preprocesado era necesario indicar el canal sobre el cuál queríamos trabajar (principalmente cualquiera de las señales provenientes de los electrodos colocados en el abdomen materno), así como la longitud de la ventana de muestreo, es decir, el número de muestras de la señal que queríamos obtener, y la posición de la ventana. Más adelante se aclarará el valor posible que puede tomar la ventana rectangular para el preprocesado de nuestra señal.

Realizado esto, necesitábamos convertir nuestra señal preprocesada con la WT, la cuál se encontraba en punto fijo, a doble precisión. Esta conversión se realizó porque facilitaba los cálculos, y el coste computacional en MATLAB. Finalizado esto, podíamos proceder a la clasificación de la señal mediante los diferentes algoritmos descritos en el capítulo 5.

### 6.2.2. Algoritmo *k-means*

Tal y como se detalló en el capítulo anterior, el algoritmo *k-means* no es más que un método particional, en el que necesitamos introducir la señal preprocesada y el número de clusteres o grupos en los que queremos dividir esta señal. Para utilizar este algoritmo se han implementado diferentes posibilidades de utilización, que van desde la utilización de diferentes funciones de disimilitud, hasta diferentes maneras de inicializar los centroides de nuestra señal.

#### Inicialización del algoritmo *k-means*

Para inicializar nuestros centroides, y por tanto, el algoritmo, tenemos dos variantes. Estas variantes son los algoritmos explicados anteriormente, 5.8.2.1 y 5.8.2.2. Por lo que podremos:

- Seleccionar  $k$  semillas aleatoriamente implementando el algoritmo *k-means++* para la inicialización de los centroides.
- Seleccionar  $k$  semillas aleatoriamente implementando el algoritmo de *Lloyd* o *k-means*.

Las diferencias de estos algoritmos se explicaron anteriormente, pero básicamente, el algoritmo *k-means++* nos ayudaba a inicializar los centroides con menor coste computacional, evitando realizar más iteraciones en el algoritmo.

### Distancias

Nuestro algoritmo *k-means* utilizada diferentes medidas de disimilaridad o distancias, introducidas en el Cuadro 5.1, estas distancias se utilizarán en espacios Euclídeos de *p-dimensiones*, siendo en nuestro caso concreto el *Espacio Euclídeo* unidimensional. Por lo que nuestro algoritmo, calculará los centroides de distinta manera según la función de distancia que le introduzcamos.

Por defecto, nuestro algoritmo utilizará la distancia Euclídea al cuadrado, qué, aunque no cumpla la desigualdad de *Minkowski*, en el caso de un espacio unidimensional se podrá considerar válida como función de disimilaridad. Tal y como comentamos en el capítulo anterior, se utiliza esta distancia por defecto debido a que reduce significativamente el coste computacional de nuestro algoritmo. Básicamente las dos distancias más importantes que podrán ser utilizadas en nuestro algoritmo serán la distancia Euclídea al cuadrado y la distancia de *Manhattan*, aunque en el caso del espacio unidimensional, la distancia de *Manhattan* sea la distancia Euclídea. Se ha probado este algoritmo con otras distancias, como por ejemplo, la distancia del coseno, no ofreciendo buenos resultados, debido a que esta distancia está más ideada como medida de similaridad que como medida de disimilaridad, por lo que no se implementará en la GUI final.

### El problema del mínimo local

En el capítulo 5 vimos, como los problemas de Clustering básicamente eran problemas de optimización, en los cuáles era necesario alcanzar un mínimo absoluto de una función objetivo, en concreto la función objetivo *k-means* 5.26. Debido a la complejidad de resolución de este problema, se implementa un algoritmo equivalente, 5.8.2.1, que nos ofrece como solución un mínimo, pudiendo ser mínimo relativo o absoluto. El problema es que la utilización de este algoritmo no nos garantiza siempre que el mínimo sea absoluto. Por lo que, puede ser que se trate de un mínimo local, el cuál no nos será válido para nuestra implementación. Esto ocurre debido a que este algoritmo inicializa los centroides de manera aleatoria, donde la reasignación de un punto cualquiera a un nuevo grupo aumentaría la suma total de las distancias de los puntos a los centroides, siendo por lo tanto la suma total un mínimo relativo y no absoluto.

Para evitar esto, con cada señal a clasificar, se repite la función *k-means* varias veces, hasta asegurarnos que hemos encontrado un mínimo absoluto, el cuál vendrá dado por la sumatoria de distancias de menor valor. Por tanto, podemos concluir, que

## 6. SIMULACIÓN

---

esto nos asegura encontrar siempre el mínimo absoluto de nuestra función objetivo *k-means*. A pesar de tener que repetir varias veces el algoritmo, esto no nos supone una gran implicación en el coste computacional del mismo, debido a que como veremos en el siguiente capítulo la propia WT consume más tiempo y recursos que nuestros algoritmos de clasificación.

Realizando una prueba con la señal *r01.edf* perteneciente al banco de datos *Abdominal and Direct Fetal Electrocardiogram*, con una ventana de longitud  $l = 5000$  muestras, con posición de la ventana en las muestras:  $inicio = 5001$ ,  $final = 10000$ , clasificando nuestra señal en 3 grupos,  $k = 3$ , y realizando 10 repeticiones del algoritmo, obtenemos los siguientes resultados:

```
Repetición 1, 6 iteraciones, suma total de distancias = 0.00841159.  
Repetición 2, 6 iteraciones, suma total de distancias = 0.00841159.  
Repetición 3, 7 iteraciones, suma total de distancias = 0.00841159.  
Repetición 4, 4 iteraciones, suma total de distancias = 0.00841159.  
Repetición 5, 8 iteraciones, suma total de distancias = 0.00411869.  
Repetición 6, 5 iteraciones, suma total de distancias = 0.00841159.  
Repetición 7, 5 iteraciones, suma total de distancias = 0.00841159.  
Repetición 8, 6 iteraciones, suma total de distancias = 0.00841159.  
Repetición 9, 2 iteraciones, suma total de distancias = 0.00331758.  
Repetición 10, 6 iteraciones, suma total de distancias = 0.00841159.  
Mínimo absoluto de la suma total de distancias = 0.00331758
```

Tener en cuenta que la suma total de distancias de los puntos a los centroides, es la suma de todos los puntos a los distintos centroides. En nuestro caso, para el mínimo absoluto 0,00331758, este valor se descompone en otros 3 valores, es decir, la distancia de cada punto a cada centroide ( $k = 3$ ). Siendo en este caso  $mínimo_{abs} = 0,00331758 = 0,00271758 + 0,0 + 0,0006$ .

Tal y como podemos observar en la Figura 6.3, el algoritmo *k-means* varía en cada iteración, por lo que según hemos comentado, es necesario repetir el algoritmo varias veces hasta que nos aseguremos de haber encontrado un mínimo absoluto.

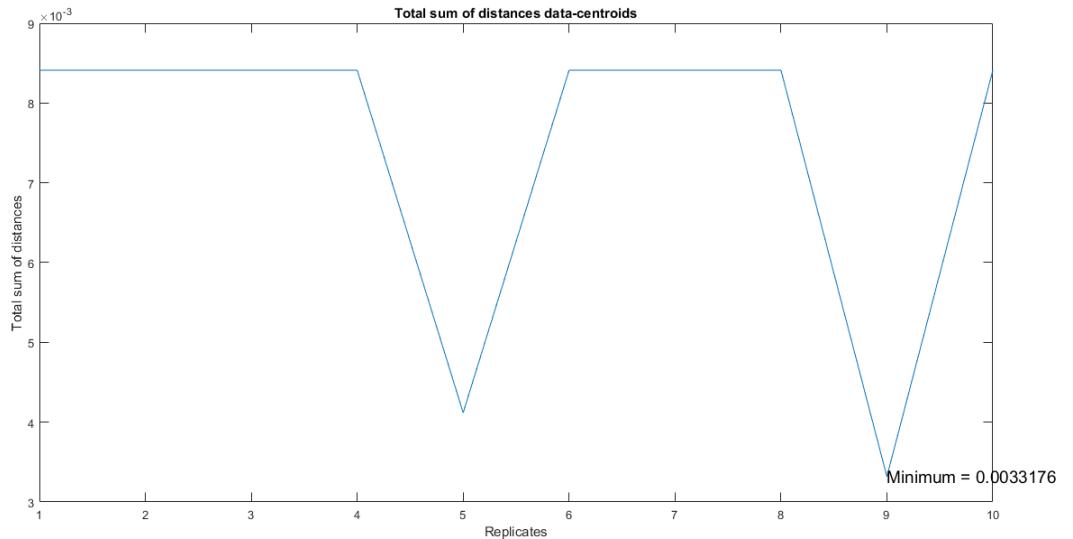


Figura 6.3: Suma total de distancias a los centroides en la señal *r01.edf* realizando varias repeticiones del algoritmo *k-means*.

### Implementación y resultados gráficos

A continuación, vamos a proceder a utilizar este algoritmo para ver los resultados del mismo en diferentes señales y configuraciones de los pulsos cardíacos. Primera-mente comenzaremos con la configuración de las amplitudes máximo-mínimo de la señal aECG de una manera temporal, según orden de muestreo, para así evitar tener que ejecutar un algoritmo de ordenamiento que consuma más recursos.

En la figura 6.4 podemos ver como el algoritmo *k-means++* ha clasificado nuestra señal *r04.edf* con longitud de ventana  $l = 5000$ , en la posición  $inicio = 5001$ ,  $final = 10000$  ( $t_0 = 5\text{ s}$ ,  $t_1 = 10\text{ s}$ ). Esta señal se ha clasificado con un parámetro  $k = 3$ , utilizando la distancia Euclídea al cuadrado. En la imagen, se pueden observar los distintos grupos o clusteres creados por el algoritmo, representados en diferentes colores, amarillo, verde y azul. Además, se han representado los centroides de cada grupo mediante una  $x$  de color rojo. En el interior de la elipse representada en la figura con el texto fECG QRS complex, estarán los pulsos cardíacos fetales, mientras que en la elipse superior se encontrarán los pulsos maternos. Por otra parte, el resto de datos no encuadrados en ninguna de las dos elipses pertenecerá al grupo del ruido.

En esta Figura podemos observar que prácticamente se han clasificado a la perfección nuestras señales, aún sin haber pasado la corrección de FP y FN. Por lo que los valores obtenidos en esta clasificación han sido:

## 6. SIMULACIÓN

---

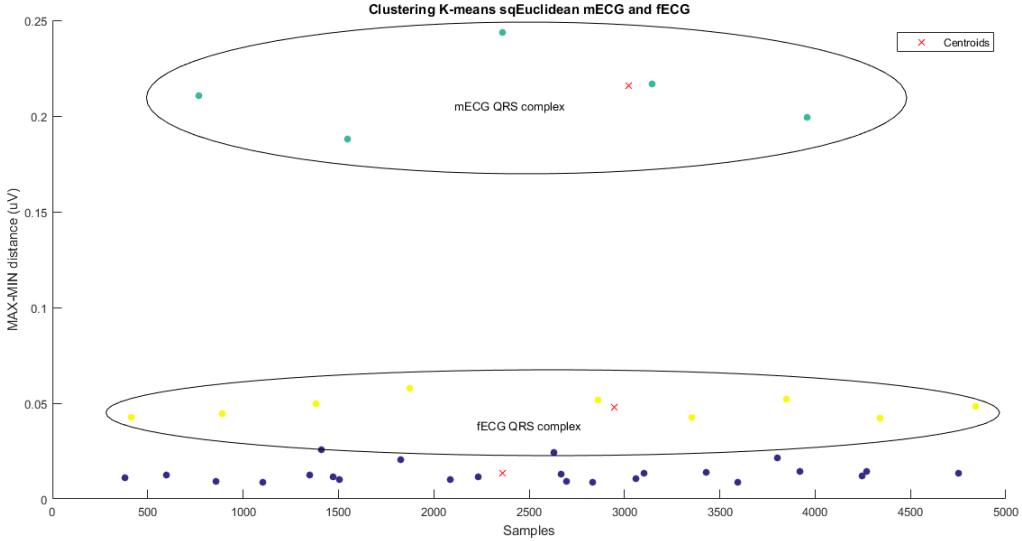


Figura 6.4: Datos de distancia máximos seguidos de mínimos de la señal *r04.edf* según orden de muestreo, clasificados con  $k = 3$  utilizando el algoritmo *k-means++* y la distancia Euclídea al cuadrado.

$$\begin{aligned} TD &= 9 & FP &= 0 & FN &= 1 \\ Acc &= 0,9 & PDV &= 1 & S_e &= 0,9 \end{aligned} \tag{6.1}$$

Si comparamos este resultado con nuestro anterior método, la clasificación mediante umbrales, obtenemos:

$$\begin{aligned} TD_{umbral} &= 8 & FP_{umbral} &= 0 & FN_{umbral} &= 2 \\ Acc_{umbral} &= 0,8 & PDV_{umbral} &= 1 & S_{e_{umbral}} &= 0,8 \end{aligned} \tag{6.2}$$

Por lo que, en esta primera prueba, gracias a la utilización del algoritmo *k-means++* con distancia Euclídea al cuadrado, hemos logrado obtener mejor precisión que con el método de los umbrales. Viendo este resultado, podemos afirmar que parece que el algoritmo funciona bien con las señales aECG, y con la disposición de los datos de manera temporal. A pesar de esto, no queremos extraer resultados únicamente de una sola prueba, por lo que realizaremos varias pruebas para intentar evitar en la medida de lo posible, una deducción inductiva.

## 6.2 Implementación software en MATLAB

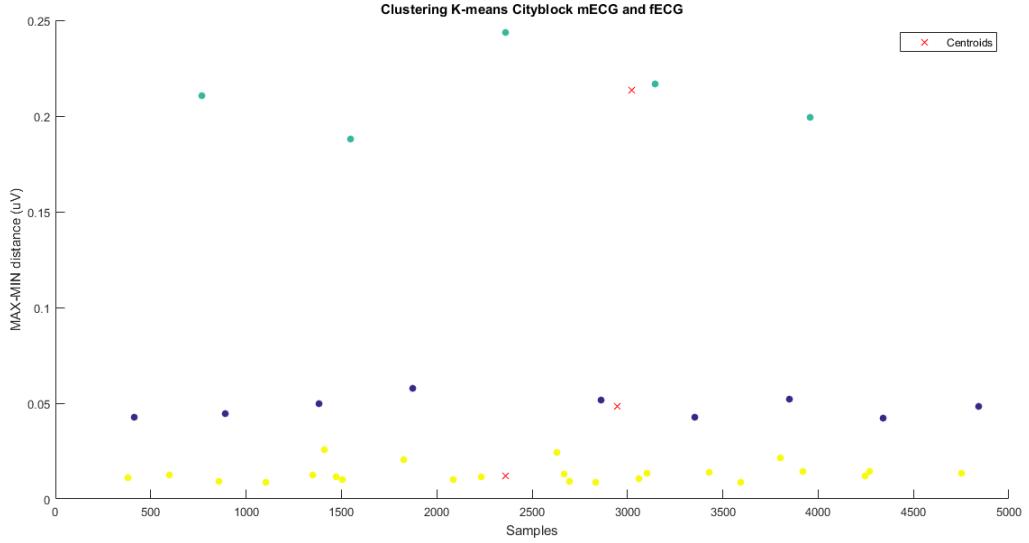


Figura 6.5: Datos de distancia máximos seguidos de mínimos de la señal *r04.edf* según orden de muestreo, clasificados con  $k = 3$  utilizando el algoritmo *k-means++* y la distancia *Cityblock*.

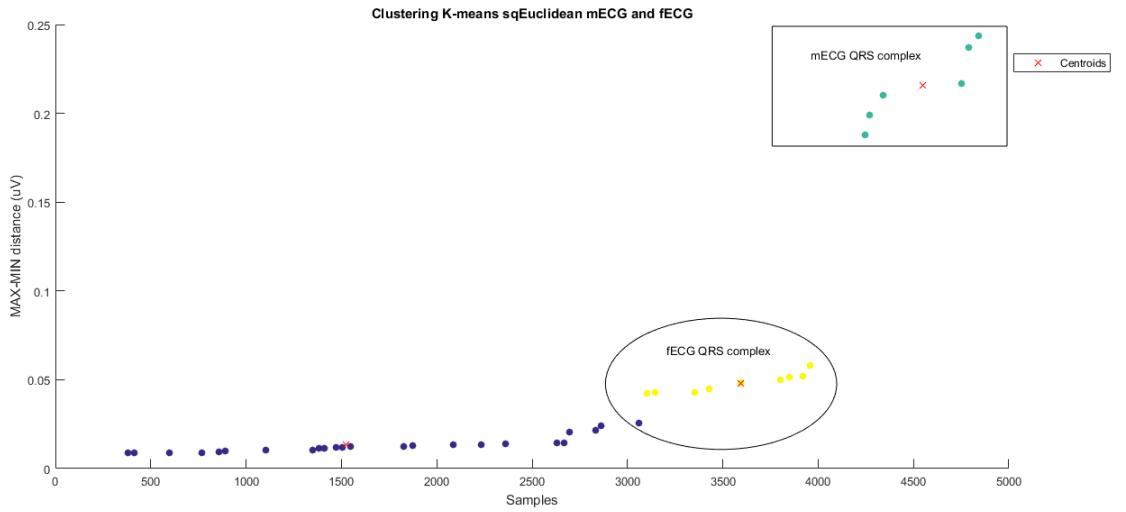


Figura 6.6: Datos de distancia máximos seguidos de mínimos de la señal *r04.edf* ordenados en orden creciente, clasificados con  $k = 3$  utilizando el algoritmo *k-means++* y la distancia Euclídea al cuadrado.

Si clasificamos de nuevo nuestra señal con el mismo algoritmo *β-means++* pero con la utilización de la distancia *Cityblock*, vemos como en la Figura 6.5 hemos obtenido la misma clasificación, independientemente de la distancia escogida. Esto es debido a que tenemos un espacio Euclídeo unidimensional con pocos datos en él, gracias a

## 6. SIMULACIÓN

---

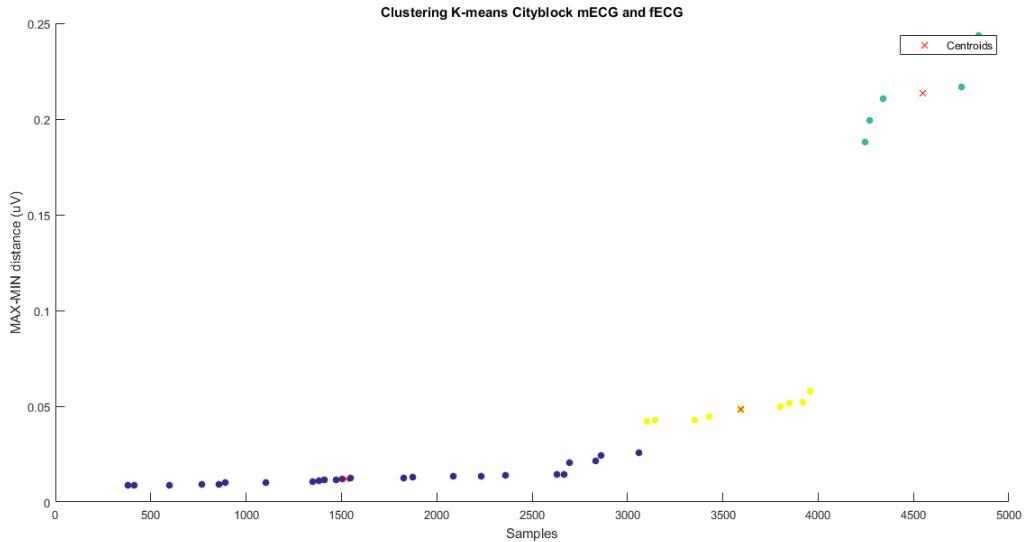


Figura 6.7: Datos de distancia máximos seguidos de mínimos de la señal *r04.edf* ordenados en orden creciente, clasificados con  $k = 3$  utilizando el algoritmo *k-means++* y la *distancia Cityblock*

esto los puntos serán fácilmente clasificables. Los resultados obtenidos coinciden con los mostrados anteriormente gracias a la distancia Euclídea al cuadrado, mostrados en 6.1.

Procedemos a continuación a re-ordenar los datos según su amplitud en la distancia máx-mín. Se ordenaran los datos en esta configuración para ver si la disposición de los mismos en el espacio nos ayuda a mejorar la clasificación anterior. Tanto la configuración en orden ascendente como la descendente, ofrecerán la misma predisposición de los datos de nuestra señal, sin diferencias a la hora de clasificarlas. Comenzaremos utilizando la configuración en sentido ascendente.

Por lo tanto, en la Figura 6.6 vemos nuestros datos clasificados utilizando el algoritmo *k-means++* para  $k = 3$  y con distancia Euclídea al cuadrado. En esta imagen, podemos observar como el grupo de color azul se corresponde con el ruido de nuestra señal. La elipse de la parte inferior derecha, la cuál contiene a los datos de color amarillo, se corresponderá con los complejos QRS fetales. Mientras que, el Cuadro superior derecho, el cuál contiene los datos de color verde, contendrá los complejos QRS maternos. Calculando la eficiencia de esta clasificación, obtenemos los mismos resultados que en los dos casos anteriores, los resultados mostrados en 6.1. Si utilizamos el mismo caso, pero con la distancia *Cityblock*, vemos como también obtenemos el mismo resultado, Figura 6.7.

## 6.2 Implementación software en MATLAB

---

A continuación, vamos a representar en la Figura 6.8 nuestra señal aECG tomada del cuarto electrodo colocado en el abdomen materno, con los fetales clasificados por los algoritmos probados, así como también se representarán los fetales anotados por el grupo de cardiólogos de *PhysioNet*. Además, se han representado los fetales extraídos gracias a la utilización del antiguo algoritmo, mediante detección de umbrales, utilizado y sobre el cuál se basa este proyecto.

Tener en cuenta que, el complejo QRS fetal no detectado no se ha podido detectar debido a que está enmascarado con un complejo QRS materno, por lo que en estos casos, ningún algoritmo es capaz de detectarlo. La única solución, será calcular su FHR y ver si su cambia su variabilidad, por lo que en ese caso, puede considerarse añadir como posible complejo QRS fetal, un complejo QRS materno.

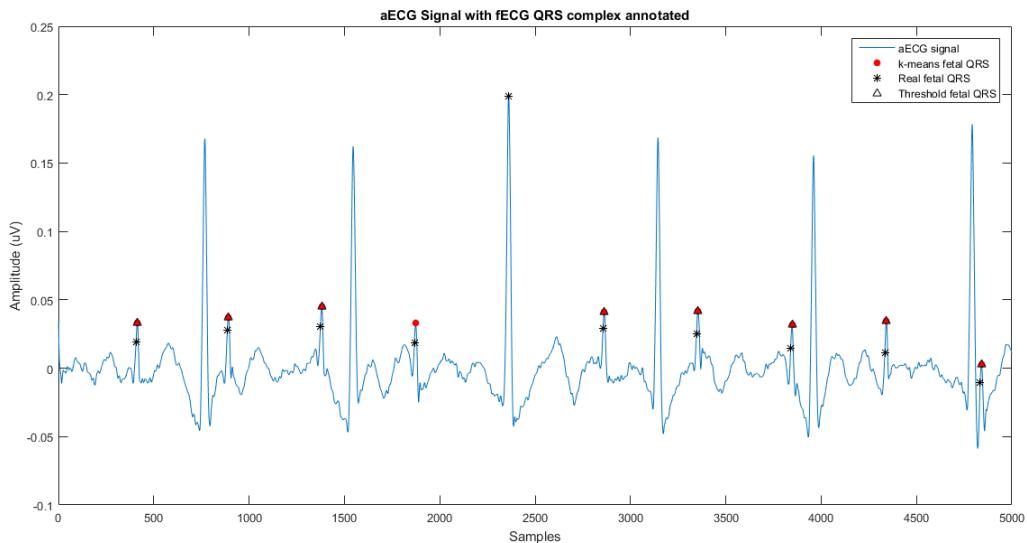


Figura 6.8: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-means*.

A pesar de estos buenos resultados, vamos a implementar este algoritmo para un tipo de señal más ruidosa, para apreciar cómo funcionaría en estos casos. Es por ello, que seleccionamos a continuación una nueva señal de la misma base de datos, *Abdominal and Direct Fetal Electrocardiogram*, en concreto la señal *r10.edf*, muestreada con una ventana de 5000 muestras, eligiendo la ventana posicionada entre 15001 y 20000 muestras ( $t_0 = 15\text{ s}$ ,  $t_1 = 20\text{ s}$ ). En este caso, se observa que según la distancia utilizada como métrica, *Euclídea al cuadrado*, o *Cityblock*, nuestros resultados cambian completamente, debido a la elevación al cuadrado.

En la Figura 6.9, podemos ver como nuestro algoritmo ha vuelto a clasificar de una manera aceptable nuestros datos utilizando la distancia Euclídea al cuadrado.

## 6. SIMULACIÓN

---

Únicamente, ha dejado sin clasificar correctamente 3 complejos QRS fetales, uno de estos fetales no detectados ha sido debido a su baja amplitud de distancia máx-mín, por lo que se ha clasificado dentro del grupo de ruido. Los otros dos fetales, no han podido ser detectados ya que se encuentran enmascarados por dos complejos QRS maternos, en este caso no existe algoritmo de *Clustering* que clasifique nuestros datos correctamente. El algoritmo por detección de umbrales utilizado tampoco clasificaría correctamente estas pulsaciones.

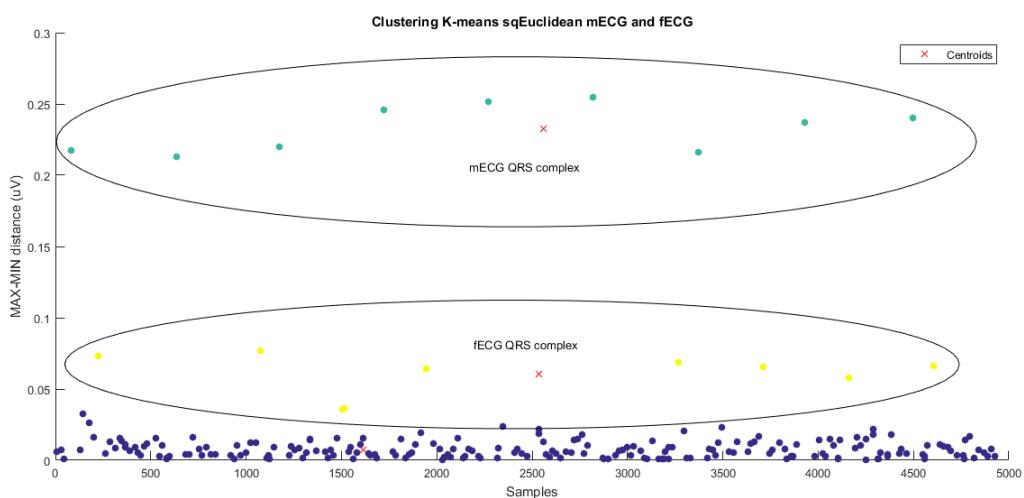


Figura 6.9: Señal aECG *r10.edf* clasificada utilizando el algoritmo *3-means* utilizando la distancia Euclídea al cuadrado en datos ordenados temporalmente.

Si por otra parte, utilizásemos la distancia Cityblock, en este caso si que habría diferencia entre la clasificación de nuestros datos. Como podemos ver en la Figura 6.10 nuestros datos fECG no se han clasificado correctamente. Esto es debido a que las ecuaciones de ambas distancias, aunque basadas ambas en la distancia de *Minkowski*, difieren entre sí. Premiando la distancia al cuadrado la mayor distancia entre puntos, método por el que obtendremos mejores resultados.

A continuación, en la Figura 6.11 volvemos a utilizar la clasificación *k-means* pero esta vez con los datos ordenados por amplitud de distancia máx-mín. Una vez más volvemos a conseguir los mismos resultados que en el caso de no ordenar nuestros datos. Por lo que, gracias a la no utilización de algoritmos de ordenamiento, podemos aumentar la eficiencia de nuestro algoritmo al implementarlo en un sistema hardware. Como podemos ver, hemos vuelto a tener el problema en la clasificación de los fECG, clasificando tres fetales incorrectamente.

## 6.2 Implementación software en MATLAB

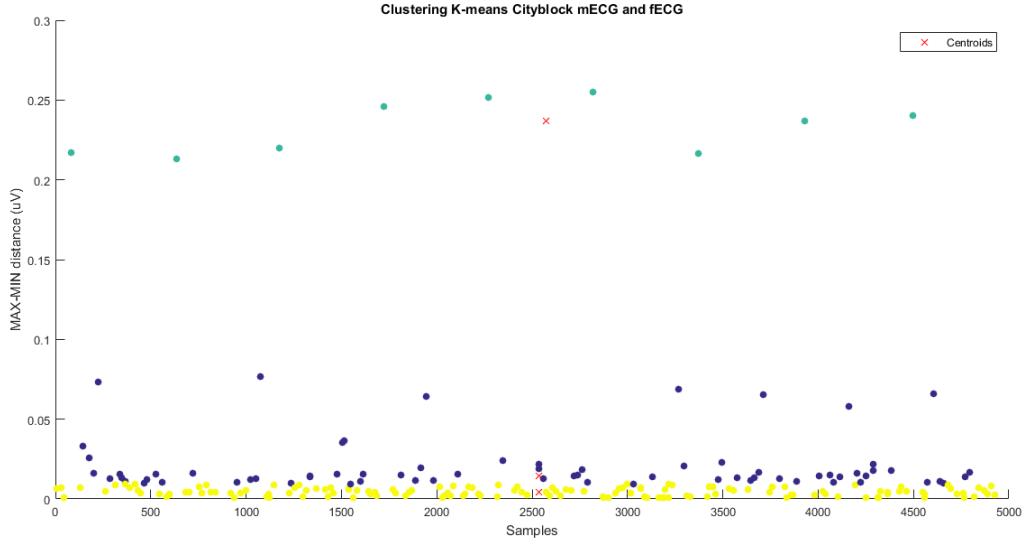


Figura 6.10: Señal aECG *r10.edf* clasificada utilizando el algoritmo *3-means* utilizando la distancia *Cityblock* o Euclídea en datos ordenados temporalmente

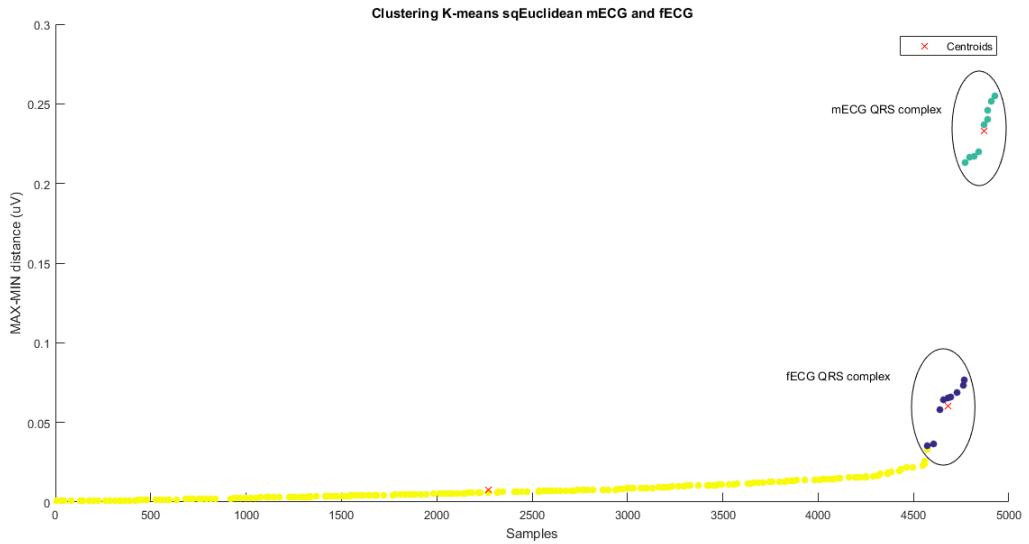


Figura 6.11: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-means* utilizando la distancia Euclídea al cuadrado en datos ordenados por amplitud.

Tal y como hemos en la Figura 6.12 realizado con el caso de la distancia Euclídea al cuadrado, hemos vuelto a clasificar nuestra señal, ordenada en amplitud, con la distancia *Cityblock*. Al igual que en el caso anterior, nuestro algoritmo de clasificación ha clasificado de nuevo los datos de la misma manera, clasificando erróneamente ruido en el grupo fECG, y a su vez, no clasificando correctamente los tres pulsos cardíacos

## 6. SIMULACIÓN

---

que deben de pertenecer al grupo fECG.

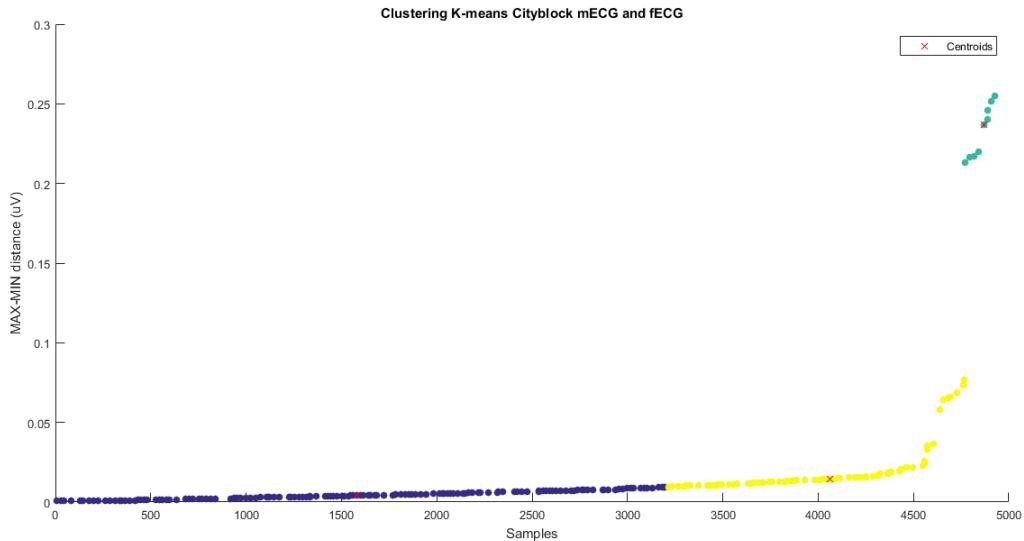


Figura 6.12: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-means* utilizando la distancia *Cityblock* o Euclídea en datos ordenados por amplitud.

En la siguiente Figura, 6.13 podemos ver la propia señal aECG tomada desde un electrodo en el abdomen materno. Tal y como podemos observar, hay 3 pulsos que nuestro algoritmo no ha reconocido correctamente, dos de ellos, están superpuestos (“camuflados”) con los maternos, por lo que nos es imposible clasificarlos con este o cualquier otro algoritmo. Más adelante propondremos una futura solución a este problema con la corrección de FP y FN.

## 6.2 Implementación software en MATLAB

---

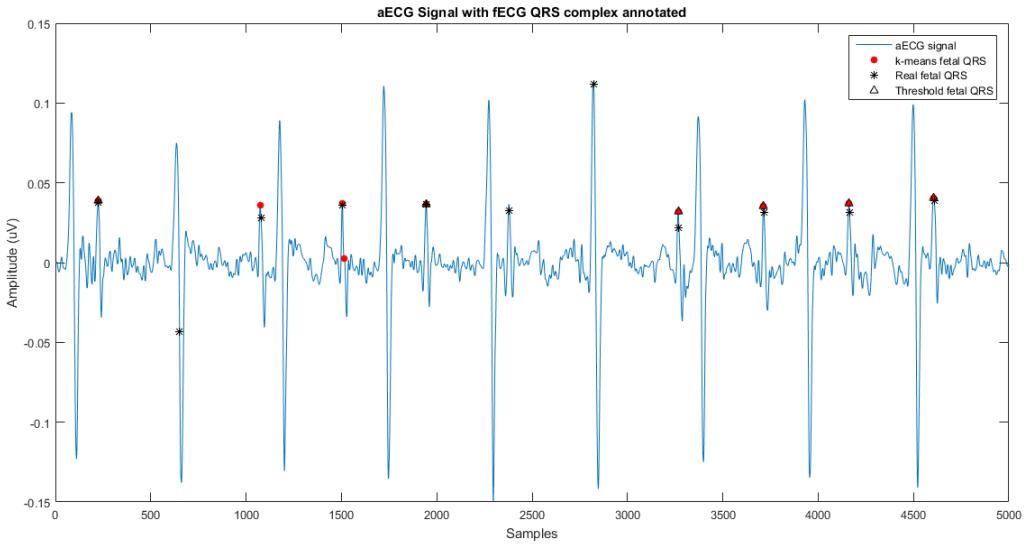


Figura 6.13: Señal aECG *r10.edf* clasificada utilizando el algoritmo *3-means* con distancia Euclídea al cuadrado

En la siguiente Cuadro, 6.5, podemos ver los resultados de las distintas clasificaciones empleadas anteriormente. Tal y como podemos observar, no existe diferencia entre ordenar los datos por amplitud de máx-mín y no hacerlo, debido a que trabajamos en un espacio unidimensional, en el que la diferencia entre datos será la misma independiente de su disposición en el vector. Por otra parte, tal y como hemos comentado, si que hay unas diferencias notables entre la aplicación de nuestro algoritmo *3-means*, con el uso de la distancia Euclídea al cuadrado y con el uso de la distancia *Cityblock*. La distancia Euclídea nos proporciona mejores resultados que la *Cityblock*, obteniendo una eficiencia  $\eta_{sgeuclidean} = 0,667/0,114 = 5,850 \times \eta_{cityblock}$ . Es decir, la eficiencia de *k-means* utilizando en este caso la distancia Euclídea al cuadrado es casi seis veces superior a la utilización de la distancia *Cityblock* o Euclídea en este caso. Además, destacamos un aspecto importante, la presencia de 59 *FP* en el caso de utilizar distancia *Cityblock* o Euclídea, por contra de 1 *FP* en el caso de la Euclídea al cuadrado.

## 6. SIMULACIÓN

---

| <i>r10.edf</i><br>3-means++ | Orden<br>temporal,<br>distancia<br><i>Euclídea</i><br><i>al</i><br><i>cuadrado</i> | Orden<br>temporal,<br>distancia<br><i>Cityblock</i> | Orden<br>creciente,<br>distancia<br><i>Euclídea</i><br><i>al</i><br><i>cuadrado</i> | Orden<br>creciente,<br>distancia<br><i>Cityblock</i> |
|-----------------------------|--|---|---|--|
| <i>TD</i>                   | 8  | 8   | 8   | 8  |
| <i>FP</i>                   | 1  | 59  | 1   | 59   |
| <i>FN</i>                   | 3  | 3   | 3   | 3  |
| <i>Acc</i>                  | 0.667  | 0.114   | 0.667   | 0.114  |
| <i>PDV</i>                  | 0.889  | 0.119   | 0.889   | 0.119  |
| <i>S<sub>e</sub></i>        | 0.727  | 0.727   | 0.727   | 0.727  |

Cuadro 6.5: Resultados de la clasificación de la señal aECG *r10.edf* en el intervalo  $t_0 = 15\text{ s}$ ,  $t_1 = 20\text{ s}$ , utilizando diferentes metodologías

Pero, existe una solución, para la cuál nuestro algoritmo *k-means* con la utilización de la distancia Euclídea o *Cityblock* puede ofrecer buenos resultados sin recurrir a configuraciones externas. Esta solución, consiste en encontrar el valor de *k* con el que mejor se clasifiquen nuestros datos. Para ello, podemos averiguarlo de manera no eficiente, probando diferentes valores del parámetro *k* hasta que obtengamos el resultado que nos garantice una mejor clasificación de nuestros fECG, o averiguar cuál el valor correcto de este parámetro para este tipo de datos mediante algunos métodos que se introducirán más adelante.

Por ejemplo, si ahora probáramos *k-means* con  $k = 4$  utilizando la distancia *Cityblock*, nuestros resultados serían distintos a los anteriores, obteniendo la clasificación correspondiente a las Figuras 6.14 y 6.15.

## 6.2 Implementación software en MATLAB

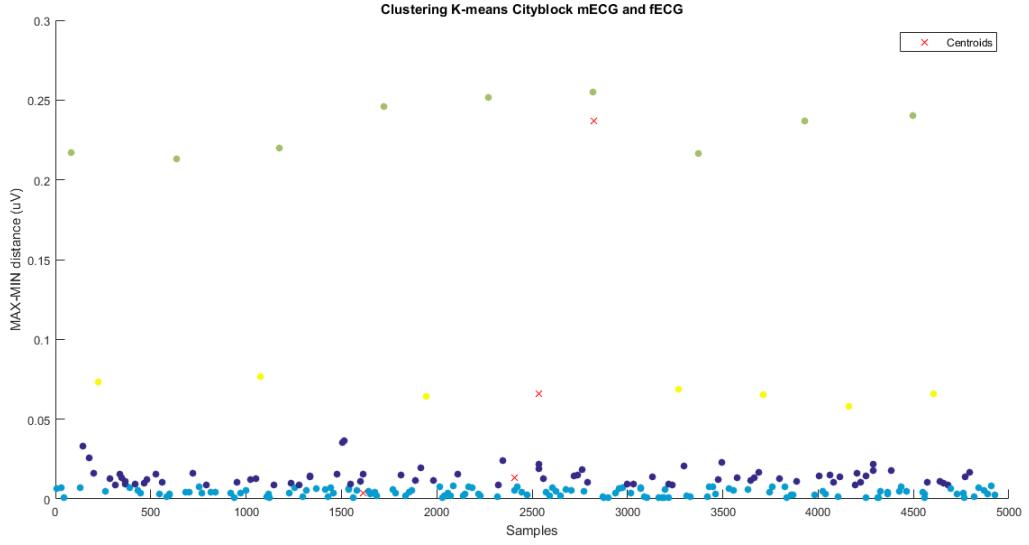


Figura 6.14: Señal aECG *r10.edf* clasificada utilizando el algoritmo *4-means* con distancia *Cityblock* o Euclídea sin elevar al cuadrado ordenando los datos mediante amplitud máx-mín

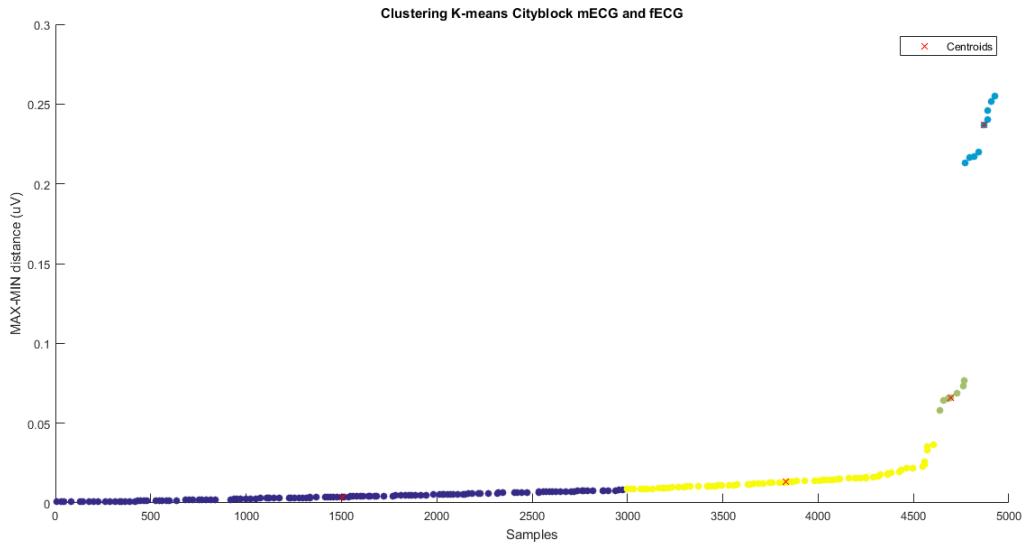


Figura 6.15: Señal aECG *r10.edf* clasificada utilizando el algoritmo *4-means* con distancia *Cityblock* o Euclídea sin elevar al cuadrado, ordenando los datos mediante amplitud máx-mín

En este caso obtendríamos los siguientes resultados:

$$\begin{aligned} TD &= 7 & FP &= 0 & FN &= 4 \\ Acc &= 0,640 & PDV &= 1 & S_e &= 0,640 \end{aligned} \tag{6.3}$$

Por lo que la elección de un parámetro  $k$  correcto aumenta considerablemente

## 6. SIMULACIÓN

---

nuestra eficiencia en la clasificación. En el siguiente capítulo introduciremos los métodos posibles a emplear para conseguir un parámetro  $k$  adecuado.

### Debilidades del algoritmo $k$ -means

A continuación, vamos a exponer los principales problemas de la utilización de este algoritmo, y en que caso se podrán utilizar y en cuales no será conveniente:

- Relacionado con la propia definición de este algoritmo, este es únicamente aplicable si la **media** está definida en nuestro espacio vectorial. Por lo que, especificando para nuestro caso, en nuestro espacio vectorial Euclídeo, la media está definida. Por lo que no tendremos problemas a la hora de utilizarlo.
- Este algoritmo es sensible a los **datos en las fronteras**. Es decir, no es recomendable utilizar  $k$ -means para datos frontera, datos que se encuentran muy alejados del resto de datos, pero que pertenezcan al mismo grupo. Esto puede llevar a errores en la clasificación, clasificando esos datos frontera como datos no propios del grupo.
- La necesidad de especificar el número de clusteres o grupos a realizar,  $k$ .

En nuestro proyecto, no tendremos en cuenta la primera de las posibles causas de fallo del empleo de  $k$ -means (media), debido a que sí que estará definida. Pero por otra parte, tendremos que tener muy en cuenta los dos otros factores, debido a que tendremos datos frontera en algunos casos (errores en la medición, artefactos, etc.), además de la necesidad de especificar a nuestro algoritmo el número de grupos,  $k$ .

### 6.2.3. Algoritmo $k$ -medoids

Este algoritmo, tal y como se vio en la sección 5.8.2.3, es un algoritmo muy similar a  $k$ -means, con la diferencia de que este algoritmo utiliza la definición de **medoides** como puntos medios de los propios grupos, en lugar de utilizar centroides. Esto implicará resultados diferentes en algunas configuraciones en el espacio de los datos a clasificar. Por ejemplo, utilizaremos el algoritmo  $k$ -medoids para clasificar de nuevo la señal *r04.edf*, en el intervalo de tiempos  $t_0 = 5\text{ s}$  y  $t_1 = 10\text{ s}$ . Para ello vamos a realizar el mismo procedimiento anterior, implementando el algoritmo un número repetido de veces hasta alcanzar el mínimo absoluto de las distancias entre puntos y medoides.

Comenzaremos utilizando la distancia Euclídea al cuadrado, con los datos sin ordenar en amplitud, para ver las diferencias entre este algoritmo y  $k$ -means, tal

## 6.2 Implementación software en MATLAB

y como vemos en la Figura 6.16. Podemos apreciar que hemos obtenido la misma clasificación que la obtenida mediante el uso del algoritmo *k-means* en esta señal. Cabe destacar que, no hay una gran diferencia significativa entre ambos algoritmos, más allá de la propia definición de punto medio de cada grupo en cada uno de ellos (centroide vs medoide). Además, tal y como ocurría en el caso de la clasificación con *k-medoids*, no existe diferencia entre clasificar nuestros datos de forma ordenada en amplitud, que no hacerlo. ( Figuras 6.16 y 6.17). Esto es debido a que nuestros datos forman un vector unidimensional, por lo que la distancia entre un punto y el resto de puntos será la misma distancia independientemente de su ordenación.

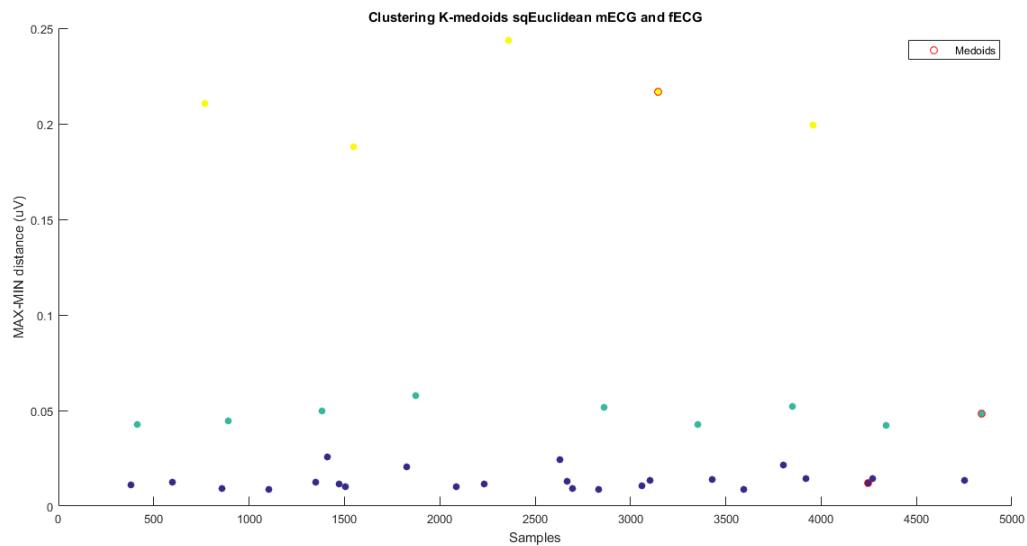


Figura 6.16: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-medoids* con distancia Euclídea al cuadrado con los datos ordenados

Si volvemos a clasificar nuestros datos utilizando en este caso la distancia *Cityblock* o Euclídea, obtendríamos las clasificaciones mostradas en las Figuras 6.18 y 6.19. Volvemos a observar que no hay diferencia en la distribución de los datos en el espacio, debido a que nuestros datos formarán un vector unidimensional.

## 6. SIMULACIÓN

---

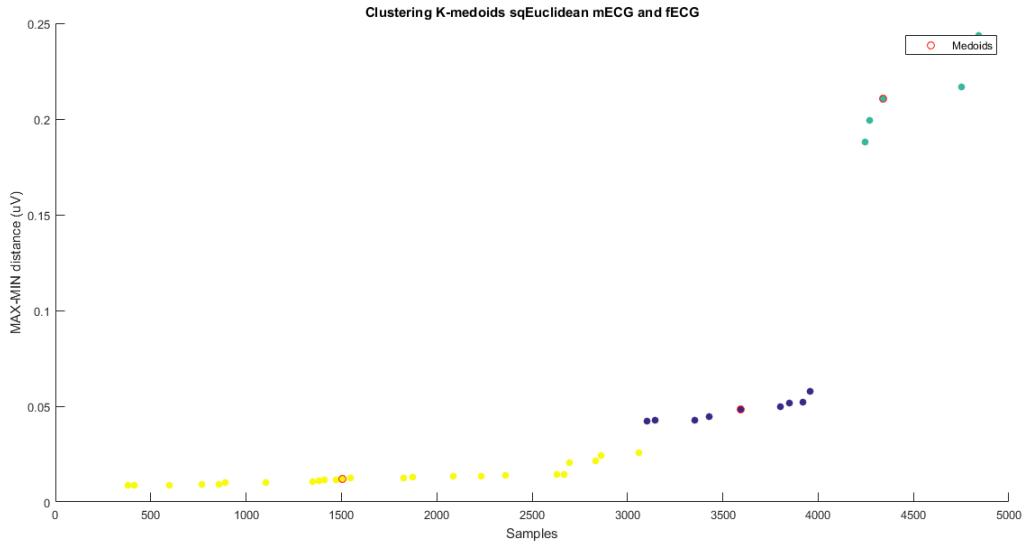


Figura 6.17: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-medoids* con distancia Euclídea al cuadrado ordenando los datos mediante amplitud máx-mín

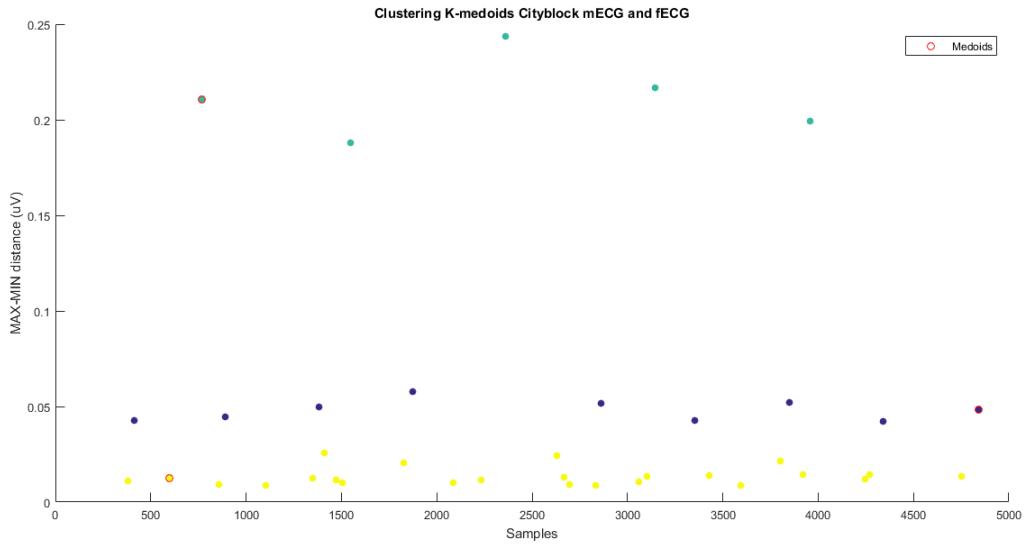


Figura 6.18: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-medoids* con distancia *Cityblock* con los datos ordenados temporalmente.

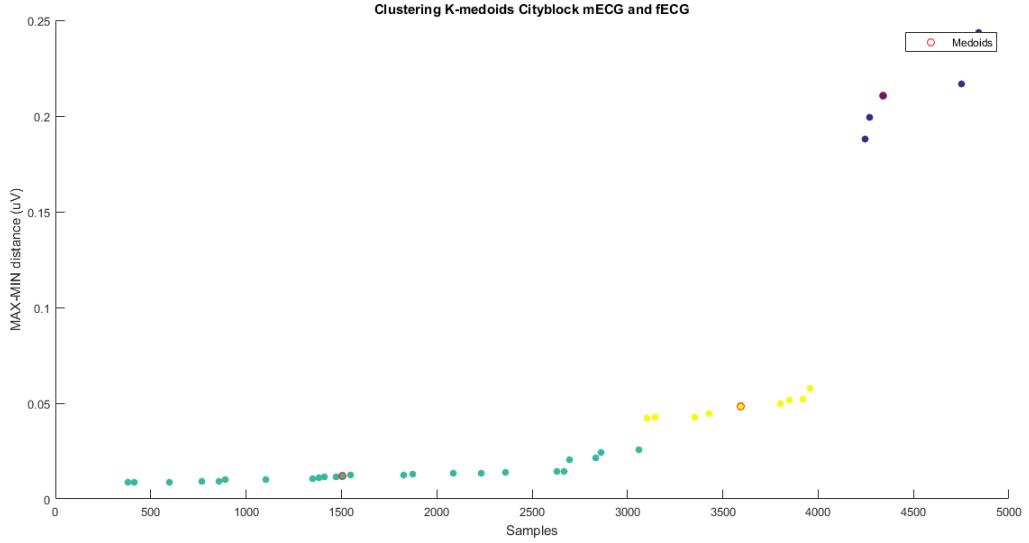


Figura 6.19: Señal aECG *r04.edf* clasificada utilizando el algoritmo *3-medoids* con distancia *Cityblock* ordenando los datos mediante amplitud máx-mín

Además, se han obtenido los mismos resultados que los obtenidos con la utilización del algoritmo *k-means*, esto es debido al parecido de ambos algoritmos, tanto en su funcionamiento como en la distancia utilizada. Estos resultados son los mostrados en la ecuación 6.4.

$$\begin{array}{lll} TD = 9 & FP = 0 & FN = 1 \\ Acc = 0,9 & PDV = 1 & S_e = 0,9 \end{array} \quad (6.4)$$

#### 6.2.4. Algoritmo *fuzzy c-means*

A continuación vamos a implementar en software el algoritmo de clasificación difusa, *fuzzy c-means* (5.8.2.4). Tal y como se comentó, este algoritmo utiliza como función objetivo la descrita en la ecuación 5.27, una función objetivo parecida a la función propia de *k-means* pero con la única diferencia, de que ahora, cada dato, poseerá un grado de pertenencia a cada grupo.

Aplicando este algoritmo a nuestros datos, en concreto la señal *r04.edf* utilizada anteriormente, en el rango de *5 s* a *10 s*, obtendremos la clasificación mostrada en la Figura 6.21.

Pero antes de comenzar con la propia clasificación en sí, vamos a mostrar el coste en iteraciones de nuestro algoritmo. Para ello se ha representado en la Figura 6.20 los valores obtenidos en la función objetivo *fuzzy c-means*. Para ello, se han representado los valores de la función objetivo en función del número de iteraciones necesarias para

## 6. SIMULACIÓN

---

llegar al momento de parada de nuestro algoritmo, es decir, aquél en el que el valor de la función objetivo en la siguiente iteración no cambia de valor. Como podemos observar en la imagen, únicamente hemos necesitado 7 iteraciones para clasificar nuestros datos correctamente. Esto nos deja dos evidencias de nuestro algoritmo: por una parte, que este converge rápidamente en nuestro caso, debido sobre todo a que hemos utilizado una metodología (máx-mín) en la que consideramos los mínimos candidatos posibles, para facilitarnos nuestra clasificación y evitar sobrecostes computacionales innecesarios. También, podemos extraer como conclusión, que una vez hemos llegado al punto de inflexión en el que nuestra función objetivo ha descendido linealmente (iteración 4 en nuestra figura), cada iteración minimizará la función objetivo mínimamente. Por lo que si por necesidad, o por error, nuestro algoritmo no pudiese ser ejecutado repetidas veces hasta asegurarnos de haber alcanzado un mínimo absoluto, la clasificación de nuestros datos no variaría de una forma muy drástica. Esto podría facilitarnos la utilización de los algoritmos en una plataforma hardware.

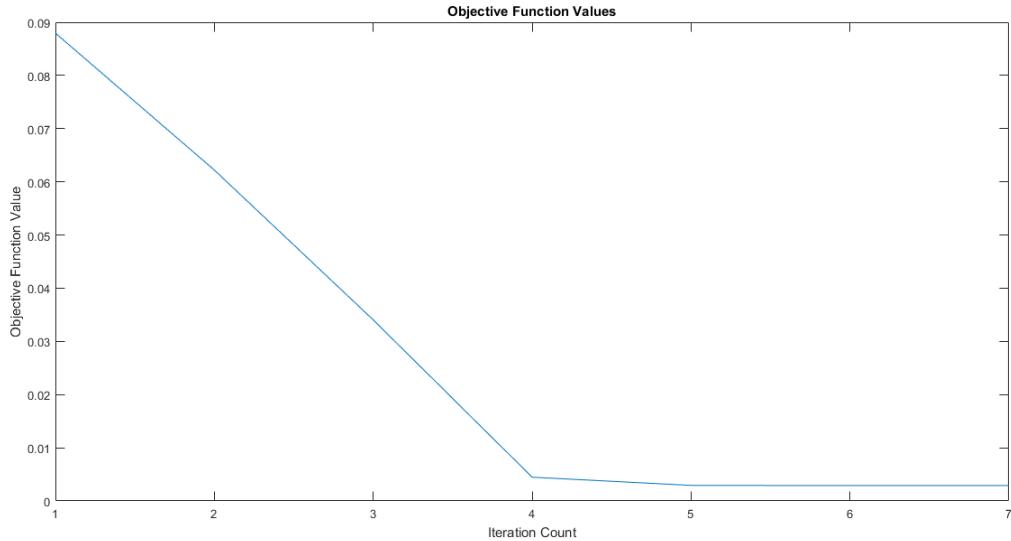


Figura 6.20: Señal aECG *r04.edf* clasificada utilizando el algoritmo *fuzzy 3-means* ordenando los datos mediante amplitud máx-mín

Clasificados nuestros datos, tal y como aparece en la Figura 6.21 podemos ver una diferencia significativa con el resto de nuestras anteriores figuras. Esta diferencia es la aparición de las probabilidades de pertenencia de cada punto a cada uno de los grupos, representadas en esta figura mediante tres gráficas diferentes de distintos colores (azul, rojo y verde). Para una mejor apreciación de las probabilidades se han representado como diagrama de barras de probabilidad en la Figura 6.22. En

estas figuras podemos apreciar la aparición de la *función de probabilidad (pmf)* de nuestros datos. Teóricamente, estas distribuciones son la propia *pmf* de nuestros datos, ya que cumplen:

- $P_j(x_i) = p_{ij}$
- $\sum_{j=1}^C P_j(x_i) = 1 \quad \forall x \in X$

Es decir, que se cumpla que la función de probabilidad  $P$  asociada a cada  $X$  sea la propia probabilidad  $p_i$  del suceso  $X = x_i$ . Además, debe de cumplirse de que la suma de todas estas probabilidades, debe de ser uno. Por lo que, si observamos cualquier punto discreto de datos, la suma de todas las probabilidades en ese punto, probabilidad de pertenecer al grupo primero, al segundo y al tercero, dará la unidad.

$$\sum_{j=1}^C P_j(x_i) = P_1(x_i) + P_2(x_i) + P_3(x_i) = 1$$

Para clasificar nuestros datos, hemos escogido de cada punto, la probabilidad mayor, asignando este punto al grupo con el que presentaba una probabilidad mayor de pertenencia. En la Figura 6.21, podemos observar como por ejemplo, la muestra 24 tiene unas probabilidades  $P_1(x_{24}) = 0,74$ ,  $P_2(x_{24}) = 0,25$ ,  $P_3(x_{24}) = 0,01$ . Por lo que se vé claramente que el grupo al que mayor probabilidad tiene de pertenecer es al primero (color azul verdoso). En ningún dato hemos obtenido valores cercanos de probabilidad, es decir, probabilidades muy cercanas entre sí (el peor caso sería aquel en el que  $P_j(x_i) = 1/C$ , aquel caso en el que todas las probabilidades sean equiprobables). En ese caso elegiríamos un grupo al azar.

La utilización de este algoritmo, es útil para los valores de frontera, como por ejemplo los del caso de probabilidades iguales. Pero en nuestros datos, va a ser difícil encontrarnos con este caso, lo que nos beneficiará en gran medida la clasificación. Denotar que gracias a esto, no existirá una gran diferencia entre la utilización del algoritmo *fuzzy c-means* y el propio *k-means*.

## 6. SIMULACIÓN

---

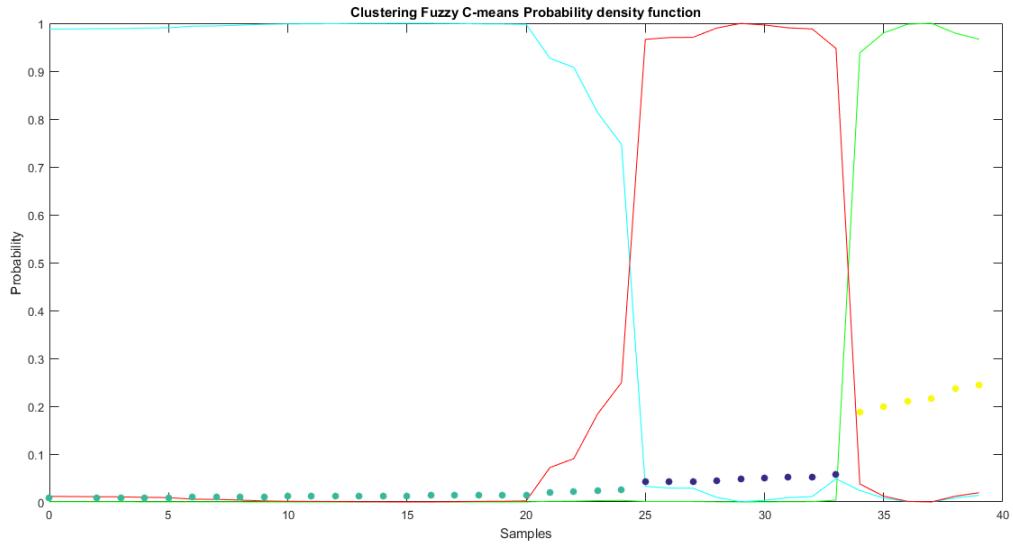


Figura 6.21: Señal aECG *r04.edf* clasificada utilizando el algoritmo *fuzzy 3-means*, ordenando los datos mediante amplitud máx-mín y con la representación de la *pmf* de cada dato.

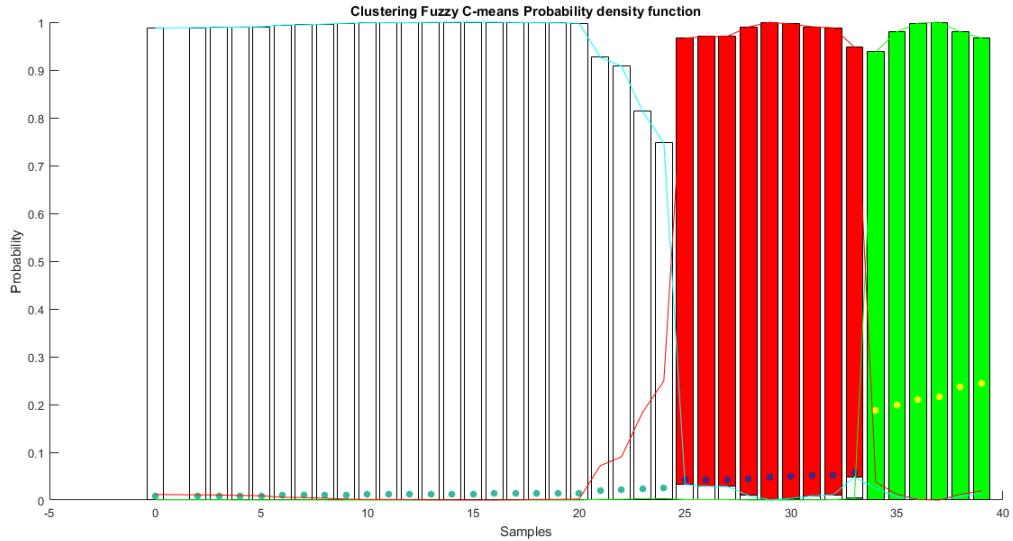


Figura 6.22: Señal aECG *r04.edf* clasificada utilizando el algoritmo *fuzzy 3-means* con distancia Euclídea ordenando los datos mediante amplitud máx-mín, y con la representación de la *pmf* de cada dato.

La representación de nuestros complejos QRS fetales en la propia señal aECG preprocesada con la WT se muestra en la Figura 6.23. Como podemos ver, en esta clasificación hemos obtenido los mismos resultados que los obtenidos utilizando el

algoritmo *k-means*. Esto es tal y como hemos comentado anteriormente, debido a que nuestros datos raramente serán equiprobables, y que las funciones objetivo de ambos algoritmos son muy parecidas, el resultado será prácticamente el mismo utilizando ambas. Los resultados pueden verse en la ecuación 6.1.

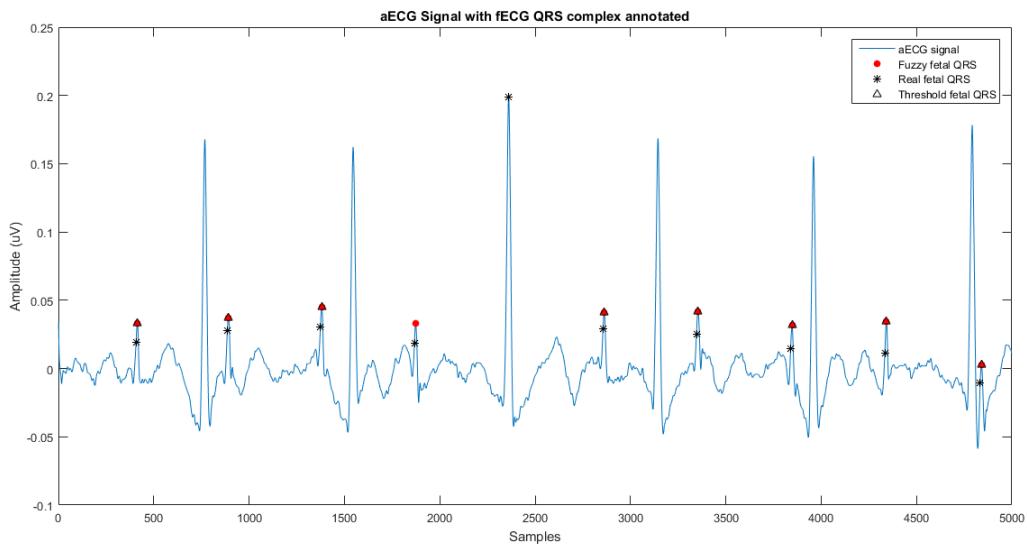


Figura 6.23: Señal aECG *r04.edf* clasificada utilizando el algoritmo *fuzzy 3-means*

### 6.2.5. Algoritmo de clasificación jerárquica

En esta sección, vamos a utilizar la clasificación jerárquica para clasificar nuestros datos, y representar de una forma jerárquica esta propia clasificación. Para ello, en nuestra implementación crearemos un árbol jerárquico de grupos o *dendograma*. Este árbol no posee una única clasificación, sino una jerarquía de niveles, donde los clusteres de un nivel se unen para formar un nuevo clúster en el siguiente nivel. Para ello, emplearemos el algoritmo descrito en 5.8.1.

Básicamente nuestro proceso consistirá en:

- Encontrar la disimilitud o distancia entre cada par de datos del conjunto
- Agrupar estos datos en un árbol binario jerárquico de grupos
- Determinar por donde dividir el árbol jerárquico para obtener los grupos necesarios.

Por lo que, primeramente calcularemos las distancias entre todos los datos de nuestro conjunto, calculando la matriz de distancias de nuestros datos. Esta matriz, será una

## 6. SIMULACIÓN

---

matriz cuadrada y simétrica, conteniendo en cada posición  $ij$  la distancia entre el punto  $i$  y el punto  $j$ . Tener en cuenta que la diagonal de esta matriz será nula, ya que representará la distancia entre el punto y si mismo.

Una vez que hemos calculado la matriz de distancias determinaremos que datos deberán ser agrupados en clusteres. Para ello, tomaremos la matriz de distancias y uniremos los pares de datos que están cercanos entre sí (cada grupo se hará con la unión de dos datos). Así, iremos creando grupos mediante uniones de pares de datos cercanos entre sí, hasta que todos los datos pertenezcan al mismo grupo.

Una vez creados los grupos, se creará el *dendograma*. En el dendograma, el eje  $X$  representará el índice de cada objeto en los datos originales. Las uniones entre los datos se representan como una  $\prod$ . En el eje Y, la altura de cada  $\prod$  indica la distancia entre los objetos.

Una vez realizada la clasificación gracias al dendrograma, verificaremos la validez de nuestra clasificación. Para ello, verificaremos la disimilaridad de nuestros datos. La distancia entre cada par de datos, es conocida como *distancia cofenética*. Una manera de medir, que tan bien ha sido clasificado nuestro árbol jerárquico es comparando las distancias cofenéticas con la distancia original calculada en la matriz de distancias. Si el clustering es válido, la unión de los datos en el árbol tendrán una fuerte correlación con las distancias entre datos calculadas en la matriz de distancias. Por lo que comparando todas las distancias, obtendremos un valor final, el *coeficiente de correlación cofenético*. Cuanto más cerca de 1 esté, mayor exactitud tendrá nuestro agrupamiento. Tener en cuenta que este coeficiente, dependerá del tipo de unión escogida, y de la función de distancia.

Una manera de determinar las divisiones de grupos naturales en un conjunto de datos es comparando la altura de cada unión en el árbol jerárquico con las alturas de las uniones vecinas debajo de esta unión. Una unión con aproximadamente la misma altura que la unión vecina, indica que no hay divisiones distintivas entre los datos unidos en este nivel de jerarquía. Por lo que estas uniones poseen un alto grado de consistencia, debido a que la distancia entre datos unidos es aproximadamente la misma que las distancias entre los objetos que ellas contienen. Por otra parte, una unión cuya altura difiera considerablemente de las uniones por debajo de ella indicará que los objetos unidos en ese nivel están mas alejados uno del otro que los objetos a partir de los cuales fueron unidos. Esta unión se dice que es inconsistente con las uniones bajo ella.

En *Clustering*, las uniones inconsistentes pueden indicar la frontera de una división natural en un conjunto de datos. Es decir, nos indicarán el lugar por donde cortar

nuestro árbol jerárquico.

En la Figura 6.24 hemos implementado el dendograma de los datos de la señal *r04.edf* utilizada anteriormente. En esta figura podemos observar lo comentado anteriormente, distancias entre puntos, uniones, etc. Si observamos la Figura 6.25, vemos como la unión superior, aquella que se encarga de unir todos nuestros datos en un único grupo, posee un alto valor de inconsistencia, ya que todos los demás grupos bajo ella, poseen una amplitud mucho menor. Esto indica que ante una posible separación, esta sería la primera rama del árbol jerárquico que dividiríamos. Si por otra parte, observamos las uniones inferiores (aquellas en torno a 0.01) vemos como estas uniones representarán una gran consistencia, lo que quiere decir, que la amplitud de sus uniones inferiores es muy parecida a la unión superior. No es recomendable separar nuestros datos por estas uniones, ya que indican que los datos de estas uniones pertenecen al mismo grupo.

A continuación, una vez realizado nuestro dendograma, vamos a proceder a separar nuestros datos. Para ello, lo primero es fijarnos en las uniones con mayor inconsistencia, ordenadas a su vez jerárquicamente. Esto es, la primera unión representará la primera separación entre datos, separando nuestro clúster único, en dos clusteres. La siguiente separación se realizará en la subsiguiente unión por debajo de esta (aquella unión con mayor amplitud por debajo de la recién separada). Pero, tal y como veremos, nuestros árbol jerárquico, no realizará la clasificación correctamente si lo dividimos únicamente en tres grupos.

Fijémonos en la Figura 6.26, hemos decidido separar nuestro dendrograma en tres grupos. Si observamos correctamente nuestros datos, vemos como la parte de la derecha ( $x_1 = \{5, 7, 11, 12, 19, 27\}$ ) serán los complejos QRS maternos de nuestra señal. Debido a la mayor separación entre estos puntos que el resto de puntos de los otros grupos, su unión poseerá una mayor amplitud, debido a las distancias entre ellos. Esto nos supondrá un problema complejo de resolver, ya que si dividiéramos el árbol por la línea marcada en la figura, indicada con *maxclust* = 3 obtendríamos una clasificación en la que los fetales y el ruido pertenecerán al mismo grupo, y se dividirán los complejos QRS maternos en dos subgrupos. Esta clasificación no nos ayudará a clasificar los datos, por lo que necesitaremos dividir nuestro árbol por otra unión.

El siguiente paso, viene representado en la Figura 6.27, en el que ahora, hemos decidido dividir nuestro árbol en 4 grupos. Ahora, si observamos la imagen, vemos como subdividiendo la siguiente unión, dividiremos por un lado los complejos QRS fetales, y por otro el ruido de nuestra señal. *¿Problema?* Nuestros complejos QRS maternos permanecerán en dos grupos, *¿cómo automatizar el algoritmo para que una los*

## 6. SIMULACIÓN

---

maternos y siga separando correctamente los fetales? Esta es una cuestión difícil de resolver, que podría resolverse mediante el uso de **aprendizaje supervisado**. Pero esto no es algo que nos deba de preocupar, nuestro objetivo es clasificar correctamente los complejos QRS fetales, independientemente de las clasificaciones de los otros grupos. Por un lado el ruido no nos interesaría, por lo que los grupos o subgrupos del mismo serán indiferentes. Además, la clasificación de complejos QRS maternos es mucho más sencilla, si quisiéramos averiguar cuales serán estos, podríamos utilizar el método de los umbrales anterior, o simplemente dividir nuestros datos en dos grupos, por lo que se realizaría esta clasificación correctamente.

Realizando por tanto, esta clasificación y dividiendo el árbol en 4 grupos, obtendremos los siguientes parámetros de clasificación:

$$TD = 9 \quad FP = 0 \quad FN = 1 \quad (6.5)$$
$$Acc = 0,9 \quad PDV = 1 \quad S_e = 0,9$$

Los datos, coinciden con los obtenidos mediante el algoritmo *k-means* y el *fuzzy c-means*. La única diferencia reside en que, antes hemos utilizado como parámetro  $k = 3$ , mientras que ahora hemos necesitado que sea  $k = 4$  el número de grupos.

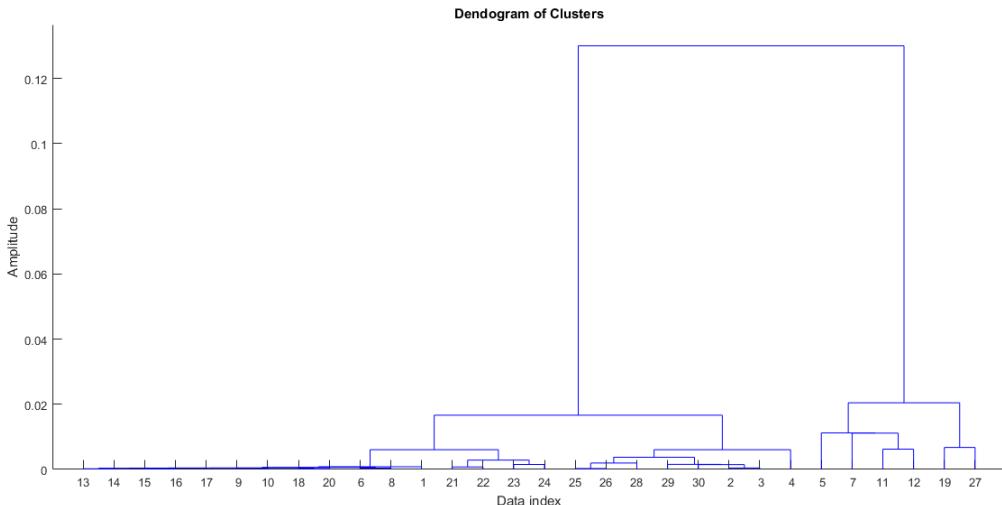


Figura 6.24: Dendograma de la señal *r04.edf*

## 6.2 Implementación software en MATLAB

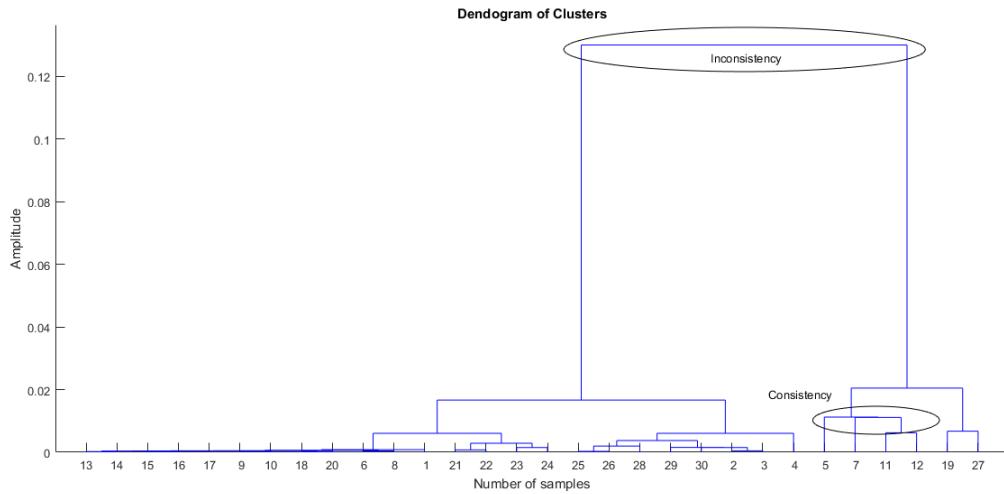


Figura 6.25: Dendograma de la señal *r04.edf*, mostrando ejemplos de uniones de inconsistencia y de consistencia.

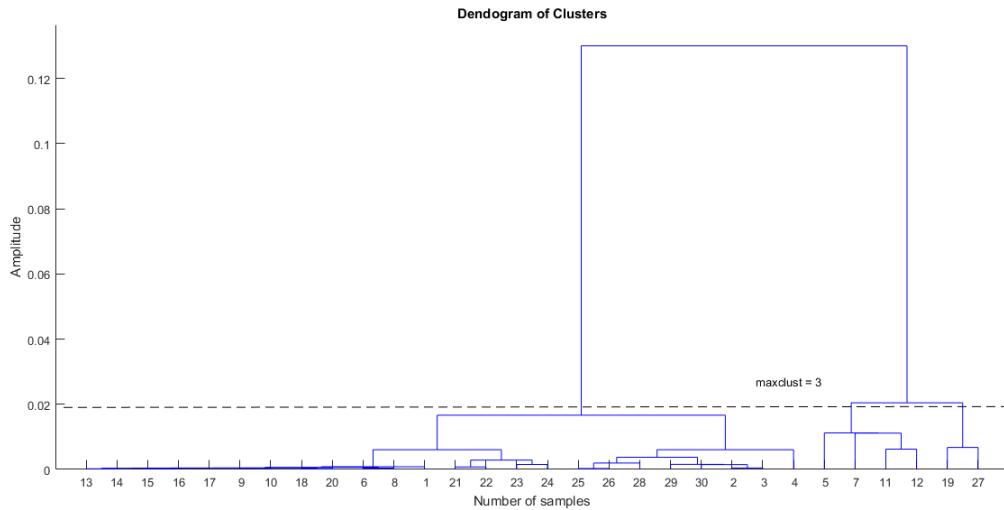


Figura 6.26: Dendograma de la señal *r04.edf*, dividiendo el árbol en tres grupos

## 6. SIMULACIÓN

---

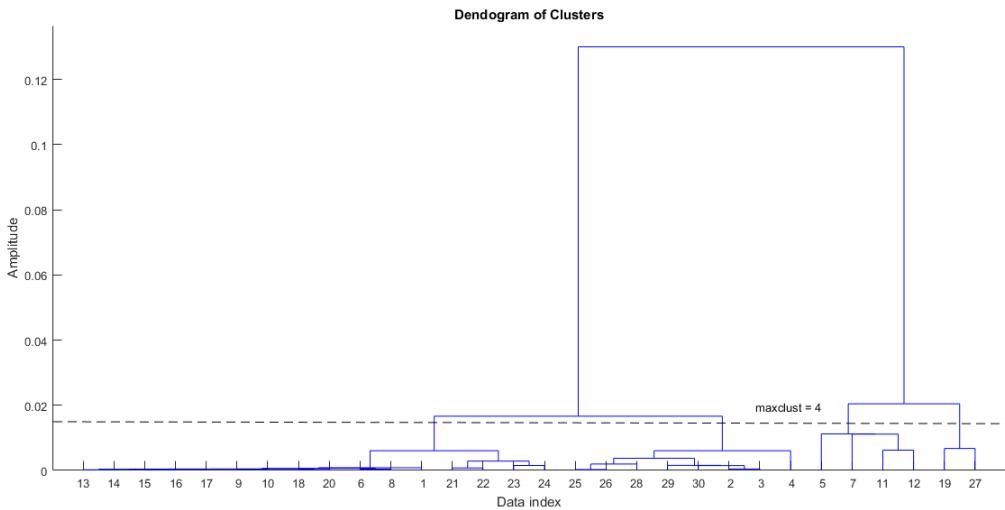


Figura 6.27: Dendograma de la señal *r04.edf*, dividiendo el árbol en 4 grupos

### 6.3. Criterios de evaluación del número correcto de clusteres

Tal y como hemos visto, uno de los mayores problemas al que nos enfrentamos con la clasificación de nuestros datos es la necesidad de indicar el número de grupos en los que queremos clasificar los datos. Sabemos que, el mínimo número de grupos que tendremos si nuestro algoritmo ha funcionado bien serán 3 (ruido, complejos QRS fetales y maternos). Pero, debido a la complejidad de estos algoritmos, a veces no clasificaremos correctamente los datos con el parámetro de grupos mínimo. Para ello es necesario utilizar diferentes algoritmos que nos indiquen la bondad de nuestros datos, y el número óptimo de grupos que necesitamos aplicar a ellos. Para ello a continuación, vamos a introducir diferentes criterios de evaluación.

#### 6.3.1. Criterio de Calinski-Harabasz

El criterio de *Calinski-Harabasz*, a menudo es llamado criterio de relación de varianza (VRC). El índice de *Calinski-Harabasz* se define como:

$$VRC_k = \frac{SS_B}{SS_W} \times \frac{(N - k)}{(k - 1)} \quad (6.6)$$

donde  $SS_B$  es la varianza total entre clusteres,  $SS_W$  es la varianza total intra-clúster,  $k$  es el número de clusteres, y  $N$  el número de observaciones.

La varianza total entre clusteres,  $SS_B$ , se define como:

### 6.3 Criterios de evaluación del número correcto de clusteres

---

$$SS_B = \sum_{i=1}^k n_i \|m_i - m\|^2 \quad (6.7)$$

donde  $k$  es el número de clusteres,  $m_i$  es el centroide de cada clúster  $i$ ,  $m$  es la media total de las muestras, y  $\|m_i - m\|$  es la distancia normalizada  $L^2$  (distancia Euclídea) entre los dos vectores.

La varianza total intra-cluster  $SS_W$ , se define como:

$$SS_W = \sum_{i=1}^k \sum_{x \in c_i} \|x - m_i\|^2 \quad (6.8)$$

donde  $k$  es el número de clusteres,  $x$  es el dato en cuestión,  $c_i$  es el clúster  $i$ -ésimo,  $m_i$  es el centroide del clúster  $i$ , y  $\|x - m_i\|$  es la distancia Euclídea entre los dos vectores.

Los clusteres que están bien definidos y diferenciados poseen un alto valor de  $SS_B$ , y un valor pequeño de  $SS_W$ . A mayor ratio  $VRC_k$ , mejor partición de los datos. Para determinar el valor óptimo de grupos, maximizamos  $VRC_k$  con respecto a  $k$ . Es decir, el valor óptimo de clusteres será la solución con el valor más alto de  $VRC_k$ .

Destacar que este criterio está optimizado para el algoritmo *k-means* con el uso de la distancia Euclídea al cuadrado.

A continuación, vamos a ver la bondad de nuestra clasificación en nuestros datos utilizando este criterio. Para ello, volviendo a la señal *r04.edf* con ventana en los instantes  $t_0 = 5\text{ s}$  y  $t_1 = 10\text{ s}$ , vamos a calcular el valor  $VRC_k$  de los datos.

Si calculamos este criterio con nuestros datos obtendremos los valores en función del número de grupos, representados en la Figura 6.28. Si observamos esta gráfica, vemos como nuestros datos siguen una representación prácticamente lineal, es decir, a mayor  $k$ , mayor  $VRC_k$ . Por lo que, el valor máximo de nuestro algoritmo se obtiene para  $k = 39$ , el número máximo de clusteres a realizar, es decir, asigna cada punto con un grupo único. Es decir, nuestros datos según el criterio de *Calinski-Harabasz* serán clasificados óptimamente en el máximo local o absoluto evaluado, en nuestro caso  $k = 39$ . Por lo que, este método no nos ofrece buenos resultados en estos datos. En nuestro caso, tras haber comprobado correctamente la extracción de complejos QRS fetales, el parámetro óptimo real de ellos será  $k = 3$ , por lo que, a priori, la utilización de este criterio no nos garantiza una buena clasificación de datos. El tiempo computacional para realizar esta evaluación ha sido de  $t = 1,87\text{ s}$ .

## 6. SIMULACIÓN

---

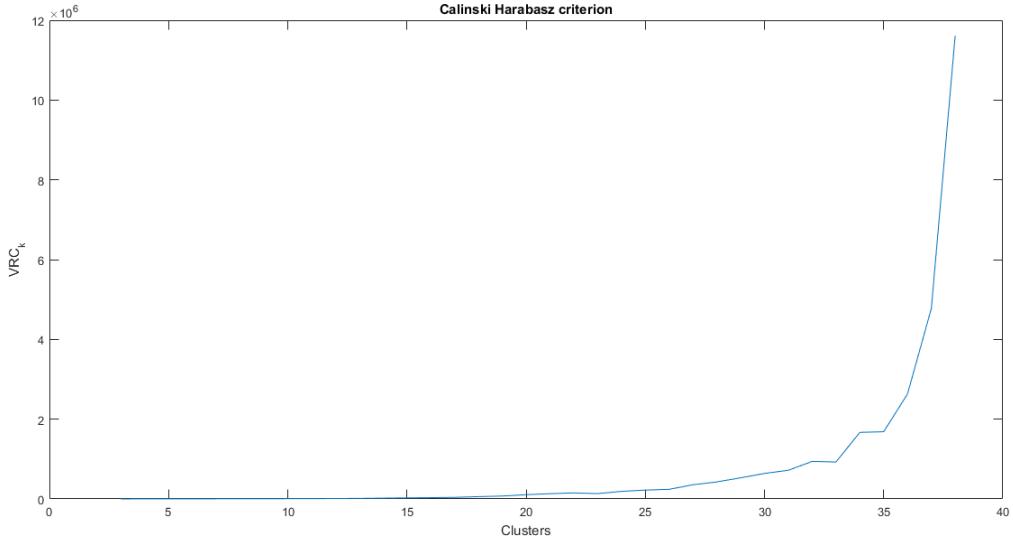


Figura 6.28: Valores  $VRC_k$  en función del número de grupos, obtenidos de calcular el criterio de *Calinski-Harabasz* en la señal *r04.edf*

### 6.3.2. Criterio de Davies-Bouldin

El criterio de *Davies-Bouldin* (DB)[14] se basa en una relación de distancias intra-clúster y entre-clusteres. El índice del método DB se define como:

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \{D_{i,j}\} \quad (6.9)$$

donde  $D_{i,j}$ , es la relación de distancias entre clusteres e intra-clúster para los clusteres  $i - \text{ésimo}$  y  $j - \text{ésimo}$ . Matemáticamente:

$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}} \quad (6.10)$$

$\bar{d}_i$  es la distancia media entre cada punto del clúster  $i - \text{ésimo}$  y el centroide del clúster  $i - \text{ésimo}$ .

$\bar{d}_j$  es la distancia media entre cada punto del clúster  $j - \text{ésimo}$  y el centroide del clúster  $j - \text{ésimo}$ .

$d_{i,j}$  es la distancia Euclídea entre los centroides de los clusteres  $i - \text{ésimo}$  y  $j - \text{ésimo}$ .

El valor máximo de  $D_{i,j}$  representará el peor caso de la relación de distancias entre clusteres e intra-clúster del clúster  $i$ . Por lo que, la solución óptima de clustering poseerá el índice  $DB$  con el menor valor posible. Se tratará de encontrar el mínimo absoluto de esta función.

### 6.3 Criterios de evaluación del número correcto de clusteres

---

Analizando de nuevo los datos de la señal *r04.edf* extraídos anteriormente, y evaluando el criterio *DB* para los naturales comprendidos entre  $3 \leq k \leq 39$ , obtendremos los resultados mostrados en la Figura 6.29. El tiempo computacional consumido en este caso ha sido de  $t = 1,51\text{ s}$ . Tal y como ocurría con el criterio de *Calinski-Harabasz*, el mínimo absoluto de este criterio ocurre para  $k = 39$ , por lo que este criterio no será válido tal y como se ha implementado para nuestros datos. Si observamos bien la figura, podremos optimizarlo para nuestro caso. Es decir, tal y como aparece en la Figura 6.30 si únicamente aplicamos el criterio para el rango  $0 < k \leq 16$  nuestro mínimo (relativo en este caso) será  $k = 3$ , por lo que el criterio será válido para su utilización en el presente proyecto.

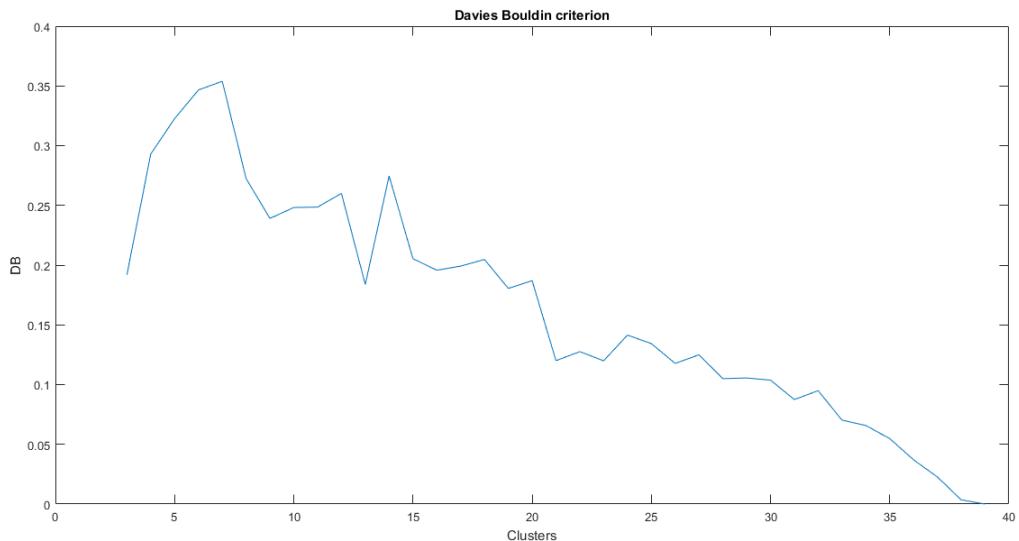


Figura 6.29: Valores *DB* en función del número de grupos, obtenidos de calcular el criterio de *Davies-Bouldin* en la señal *r04.edf*

## 6. SIMULACIÓN

---

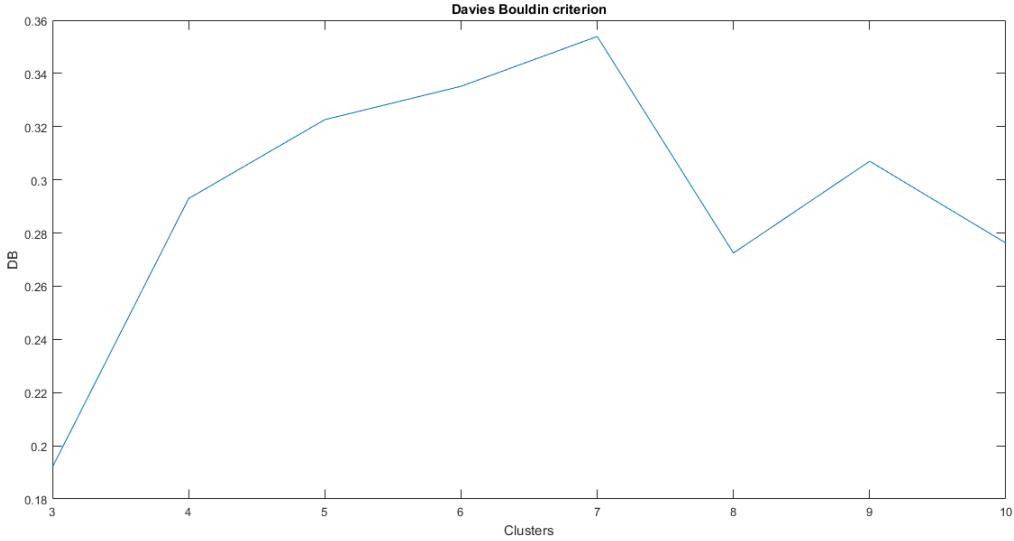


Figura 6.30: Valores  $DB$  en función del número de grupos, obtenidos de calcular el criterio de *Davies-Bouldin* en la señal *r04.edf*

### 6.3.3. Valor del espacio

Un enfoque gráfico común para la evaluación de los grupos consiste en trazar una medida del error en función de los distintos grupos propuestos,  $k$ , y localizar el máximo local o absoluto de esta gráfica, Figura 6.31. Este máximo se da para la disminución más dramática de la medida de error. El criterio del espacio formaliza este enfoque para estimar la localización del máximo, así como el número de grupos con el valor más grande de *gap*. Por lo tanto, bajo el criterio del *gap*, el número óptimo de clusteres será aquel en el que se alcance el máximo local o absoluto de este criterio.

El valor de *gap* se define como:

$$Gap_n(k) = E_n^*\{\log(W_k)\} - \log(W_k) \quad (6.11)$$

donde  $n$  es el tamaño de las muestras,  $k$  es el número de clusteres evaluados, y  $W_k$  es la medida de dispersión intra-clúster agrupada:

$$W_k = \sum_{r=1}^k \frac{1}{2n_r} D_r \quad (6.12)$$

donde  $n_r$  es el número de datos en el clúster  $r$ , y  $D_r$  es la suma de las distancias por parejas para todos los puntos del clúster  $r$ . El valor de  $E_n^*\{\log(W_k)\}$  será determinado por el método de Monte Carlo mediante el muestreo de una distribución de referencia.

### 6.3 Criterios de evaluación del número correcto de clusteres

Este criterio está definido para todo tipo de *Clustering*, incluso para aquellos que únicamente contienen un grupo, y puede ser utilizado por cualquier métrica o distancia. Sin embargo, este método es computacionalmente más costoso que los demás, debido a que el algoritmo de clustering debe de ser aplicado a unos datos de referencia para cada solución propuesta.

Debido al alto coste computacional de este algoritmo, únicamente lo hemos aplicado para  $3 \leq k \leq 15$ . Aún así, hemos obtenido un tiempo de coste computacional  $t = 52\text{ s}$ , por lo que no será deseable la utilización de este algoritmo, debido a su alto coste computacional. Los resultados de este criterio se muestran en la Figura 6.31. En este caso, el número de clusteres que mejor se adaptará a nuestros datos según este criterio es aquel con un máximo relativo o absoluto en nuestra gráfica,  $k = 15$ . Una vez más, este criterio no nos será válido para la correcta clasificación de nuestros complejos QRS fetales.

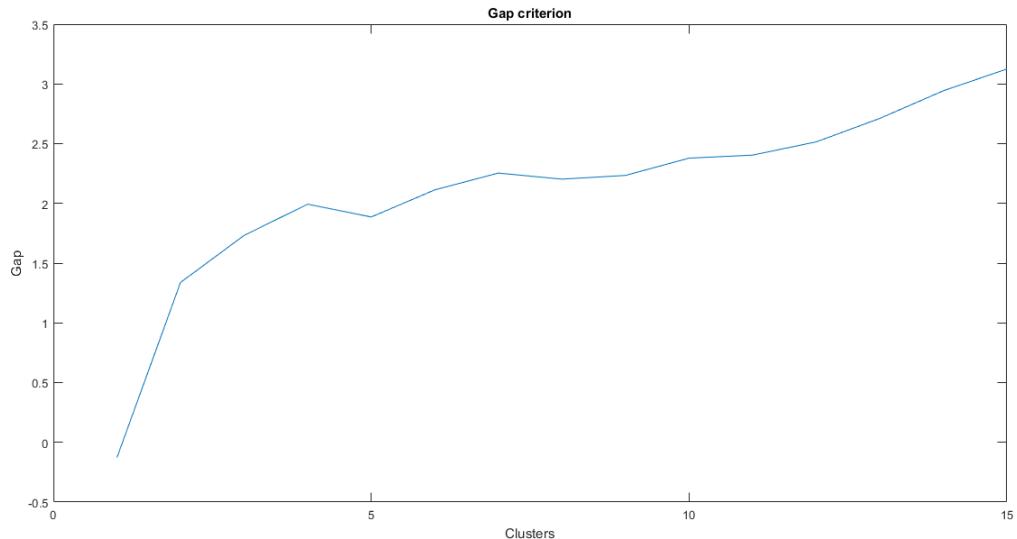


Figura 6.31: Valores *gap* en función del número de grupos, obtenidos de calcular el criterio de *gap* en la señal *r04.edf*

#### 6.3.4. Valor de la silueta

El valor de la silueta para cada punto es la medida de como de similares son ese punto con los puntos propios del propio clúster, cuando son comparados con puntos en otros clusteres. El valor de silueta para el punto  $i$ -ésimo,  $S_i$ , se define como:

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \quad (6.13)$$

## 6. SIMULACIÓN

---

donde  $a_i$  representa la distancia media del punto  $i$  –ésimo hacia otros puntos en el mismo clúster,  $i$ , y  $b_i$  es la mínima distancia media del punto  $i$  –ésimo con respecto a los puntos pertenecientes a otros grupos, minimizando el exceso de grupos.

El rango de valores que puede tomar  $S_i$  va desde  $-1$  hasta  $+1$ . Un valor mayor del índice  $S_i$  indicará que el dato  $i$  está bien agrupado con los datos de su propio clúster, y mal agrupado con los grupos vecinos. Si la mayoría de los datos tienen un alto valor de silueta, la solución de clasificación será apropiada. Si por el contrario, la mayoría de los puntos poseen un valor bajo o negativo de silueta, entonces la solución tendrá más o menos grupos de los apropiados. El criterio de silueta puede ser utilizado para cualquier métrica y algoritmo.

Si aplicamos este criterio para diferentes valores de  $k$ , por ejemplo,  $k = 3$ , representado en la Figura 6.32,  $k = 4$ , representado en la Figura 6.33 y  $k = 8$ , Figura 6.34 podemos observar lo expuesto anteriormente. Como el valor de cada punto  $S_i$  estará entre el rango  $-1, +1$ . Para obtener una representación global de todos los valores  $S_i$  calcularemos la media de todos los valores  $S_i$ , obteniendo así un valor medio de estos valores.

$$S_{i_{mean}} = \frac{1}{N} \sum_{i=1}^N S_i \quad (6.14)$$

Por lo que, aplicando este criterio y calculando el valor de  $S_{i_{mean}}$  para nuestros datos de la señal *r04.edf*, hemos obtenido los siguientes resultados, mostrados en la Figura 6.35. Como podemos observar, el valor óptimo de grupos para nuestros datos ha sido  $k = 39$ , un valor que no se ajustará a nuestra clasificación, con un coste computacional  $t = 5\text{ s}$ . Pero tal y como ocurría con el criterio de DB, podemos implementar nuestro algoritmo en una franja que nos permita obtener nuestro máximo relativo en los primeros valores de  $k$ . Para ello, en la Figura 6.36 hemos vuelto a utilizar este criterio con datos comprendidos entre  $3 \leq k \leq 10$ , obteniendo ahora  $k = 3$  como número óptimo de grupos para clasificar nuestros datos. Como conclusión, si acotamos nuestro número de parámetros, este criterio será válido para nuestra clasificación.

### 6.3 Criterios de evaluación del número correcto de clusteres

---

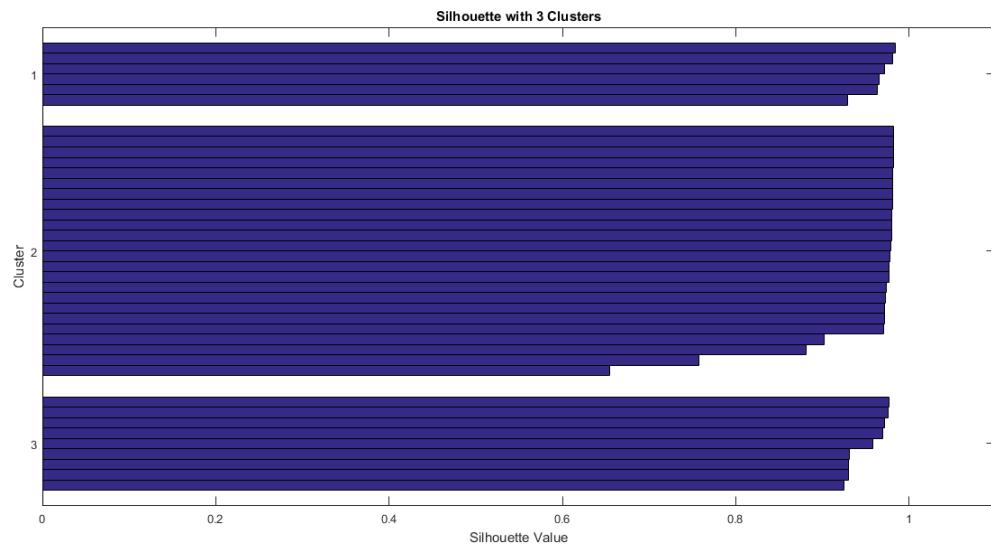


Figura 6.32: Evaluación de la señal *r04.edf* con el criterio de la silueta para  $k = 3$ .

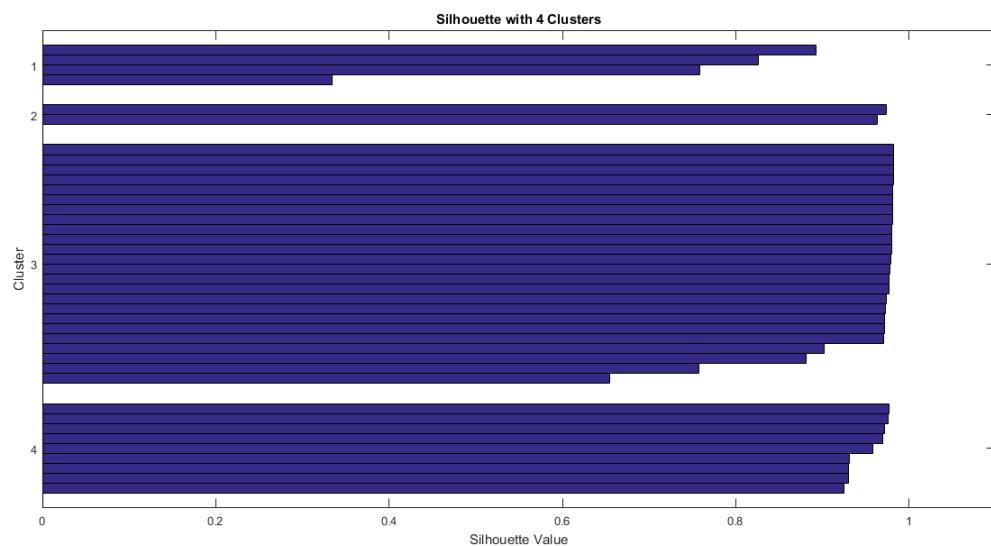


Figura 6.33: Evaluación de la señal *r04.edf* con el criterio de la silueta para  $k = 4$ .

## 6. SIMULACIÓN

---

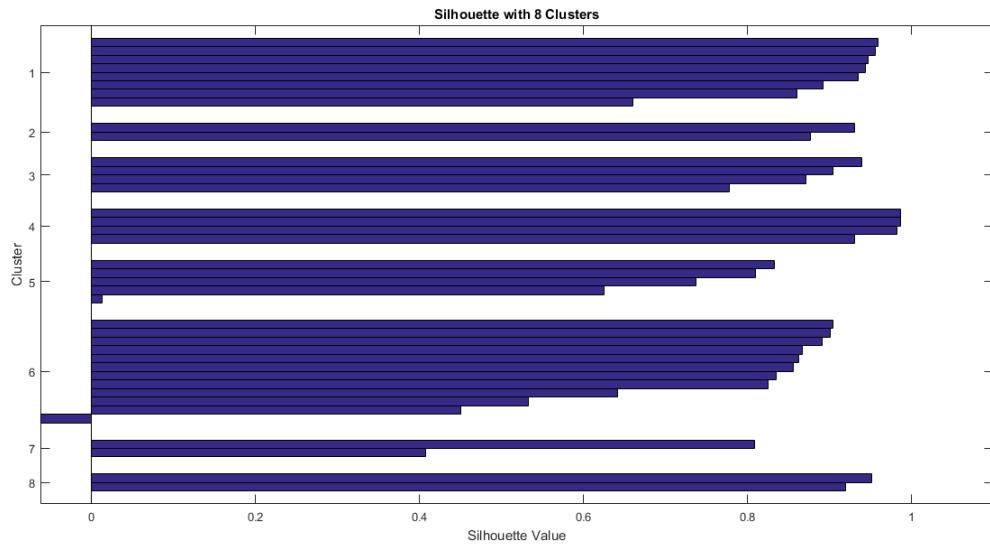


Figura 6.34: Evaluación de la señal  $r04.edf$  con el criterio de la silueta para  $k = 8$ .

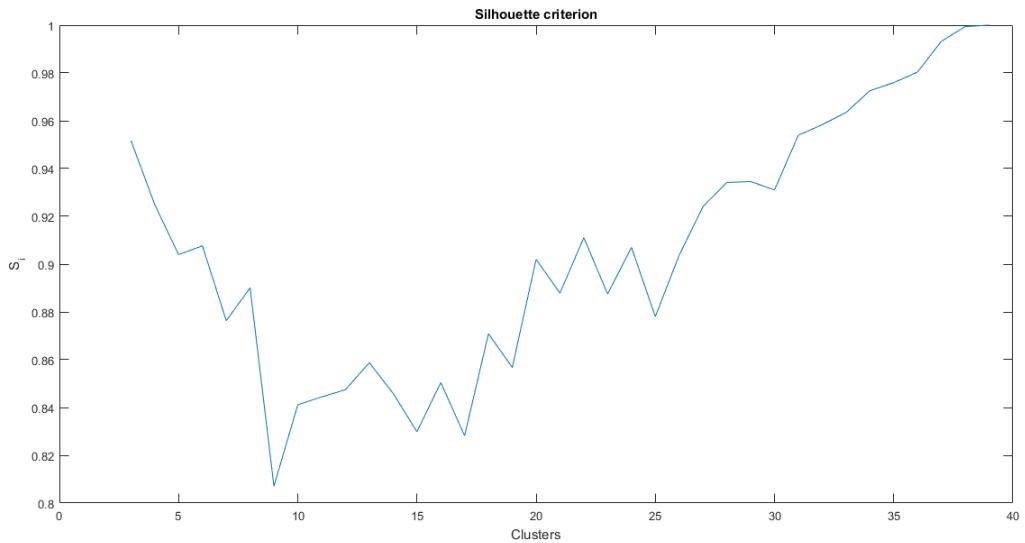


Figura 6.35: Valores  $S_i$  en función del número de grupos, obtenidos de calcular el criterio de la silueta en la señal  $r04.edf$

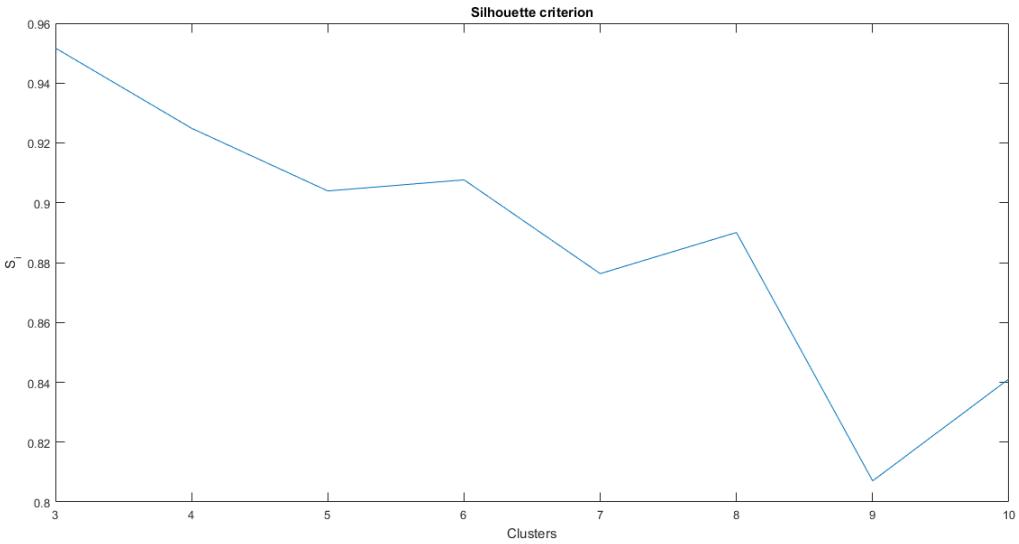


Figura 6.36: Valores  $S_i$  en función del número de grupos acotados, obtenidos de calcular el criterio de la silueta en la señal *r04.edf*

### 6.3.5. Criterios a utilizar

Una vez implementados los diferentes criterios para evaluar el número de grupos óptimos para nuestros datos, debíamos de decidir que criterio o criterios utilizar. Para ello, primeramente se analizó la bondad de este criterio con nuestro tipo de datos, es decir, que el parámetro  $k$  dado, nos ofrezca una buena clasificación de nuestros datos. También atendimos el coste computacional del mismo, para ahorrar operaciones innecesarias en nuestro algoritmo. En el Cuadro 6.6 podemos observar los resultados y características de cada uno de los criterios utilizados.

## 6. SIMULACIÓN

---

| CRITERIO                      | ECUACIONES   | COMENTARIOS  | COSTE COMP. |
|-------------------------------|--|--|-------------|
| Criterio de Calinski-Harabasz | $VRC_k = \frac{SS_B}{SS_W} \times \frac{(N - k)}{(k - 1)}$                   | Método no válido para la clasificación de complejos QRS fetales. Optimizado para <i>k-means</i> con el uso de la distancia $L^2$                   | Bajo        |
| Criterio de Davies-Bouldin    | $DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \{D_{i,j}\}$                  | Método válido únicamente acotando en un rango adecuado los grupos $k$ a analizar.  | Bajo        |
| Valor del espacio             | $Gap_n(k) = E_n^* \{ \log(W_k) \} - \log(W_k)$                               | Método no válido para la clasificación de complejos QRS fetales. Válido para para cualquier métrica y algoritmo de clasificación.                  | Muy alto    |
| Valor de la silueta           | $S_{i_{mean}} = \frac{1}{N} \sum_{i=1}^N \frac{(b_i - a_i)}{\max(a_i, b_i)}$ | Método válido únicamente acotando en un rango adecuado los grupos $k$ a analizar. Válido para para cualquier métrica y algoritmo de clasificación. | Bajo        |

Cuadro 6.6: Criterios de evaluación del número de grupos,  $k$ , para los diferentes algoritmos de *Clustering*

Para el calculo de nuestro parámetro  $k$  en cada uno de nuestros algoritmos de clasificación, utilizaremos principalmente el valor de la *silueta* y el criterio de *Davies-Bouldin*, debido a que han sido los criterios que mejor se han ajustado a nuestra clasificación. Tener en cuenta que necesitamos acotar estos dos criterios para obtener un valor óptimo para nuestra clasificación. Como conclusión, no será conveniente analizar estos criterios para  $k > 10$ .

# Capítulo 7

## Evaluación de resultados

Una vez implementados todos los algoritmos de clasificación en MATLAB y analizadas las características de todos ellos, era necesario implementar un programa que nos permitiera clasificar todas nuestras señales con los diferentes algoritmos de manera automatizada. También era deseable que obtuviéramos de manera automática los parámetros de bondad de nuestra clasificación, *Acc*, *FP*, *FN*, *TD*, etc. Por lo que, para analizar el algoritmo que mejor clasificaba todas nuestras señales era necesario analizar todas estas señales con todos los algoritmos y calcular cuál nos ofrecía mejor clasificación. Para lograr este objetivo se creó una GUI que automatizaba todo este procedimiento, almacenando todos nuestros datos en un archivo en formato *excel*. Con todos estos datos almacenados mostraríamos la media aritmética de los mismos, para así conocer el algoritmo que mejores resultados ofrece en función de los valores medios de sus parámetros de bondad.

### 7.1. GUI evaluadora de algoritmos de Clustering en señales aECG

En la primera parte de nuestra evaluación, se pretendía hallar el algoritmo que mejor clasificara nuestros complejos QRS fetales. Para ello, se implementaron todos y cada uno de los algoritmos que íbamos a necesitar para lograr una correcta clasificación. Una vez realizado esto, se procedió a comparar paso a paso todos los algoritmos en los diferentes trozos de señales procedentes del *Physionet*. El problema de esto, era la necesidad de guardar todos los resultados manualmente, y ejecutar todos los algoritmos por separado en el mismo trozo de señal hasta que se hubiera analizado esta correctamente. Esta metodología carecía de eficiencia a la hora de analizar las señales, por lo que se decidió realizar una GUI en MATLAB, (Figuras 7.1 y 7.2), que

## 7. EVALUACIÓN DE RESULTADOS

---

clasificara las señales de manera automática comparando la efectividad de todos los algoritmos, mostrando en un Cuadro todos los parámetros de bondad de la clasificación (Acc, Se, etc.). A continuación, describimos el diagrama de entradas y salidas de nuestra GUI.

- **Entradas:** para poder clasificar nuestras señales es necesario introducir los siguientes parámetros:

- *Archivo de la señal*
- *Canal*
- *Longitud de las muestras de la ventana*
- *Muestra de inicio de la ventana*
- *Muestra final de la ventana*
- *Mínimo número de grupos a clasificar*
- *Máximo número de grupos a clasificar*
- *Pasos para clasificar nuestra señal entre el mínimo y máximo número de grupos a clasificar.*

- **Salidas:** una vez clasificada nuestra señal obtendremos como salidas:

- *Parámetros de bondad, instantáneos y medios, de la clasificación de nuestra señal (Acc, Se, PDV, TD, FP, FN, etc)*
- *Coste computacional de nuestros algoritmos*
- *Gráficas de las clasificaciones realizadas.*
- Vector de FHR

## 7.1 GUI evaluadora de algoritmos de Clustering en señales aECG

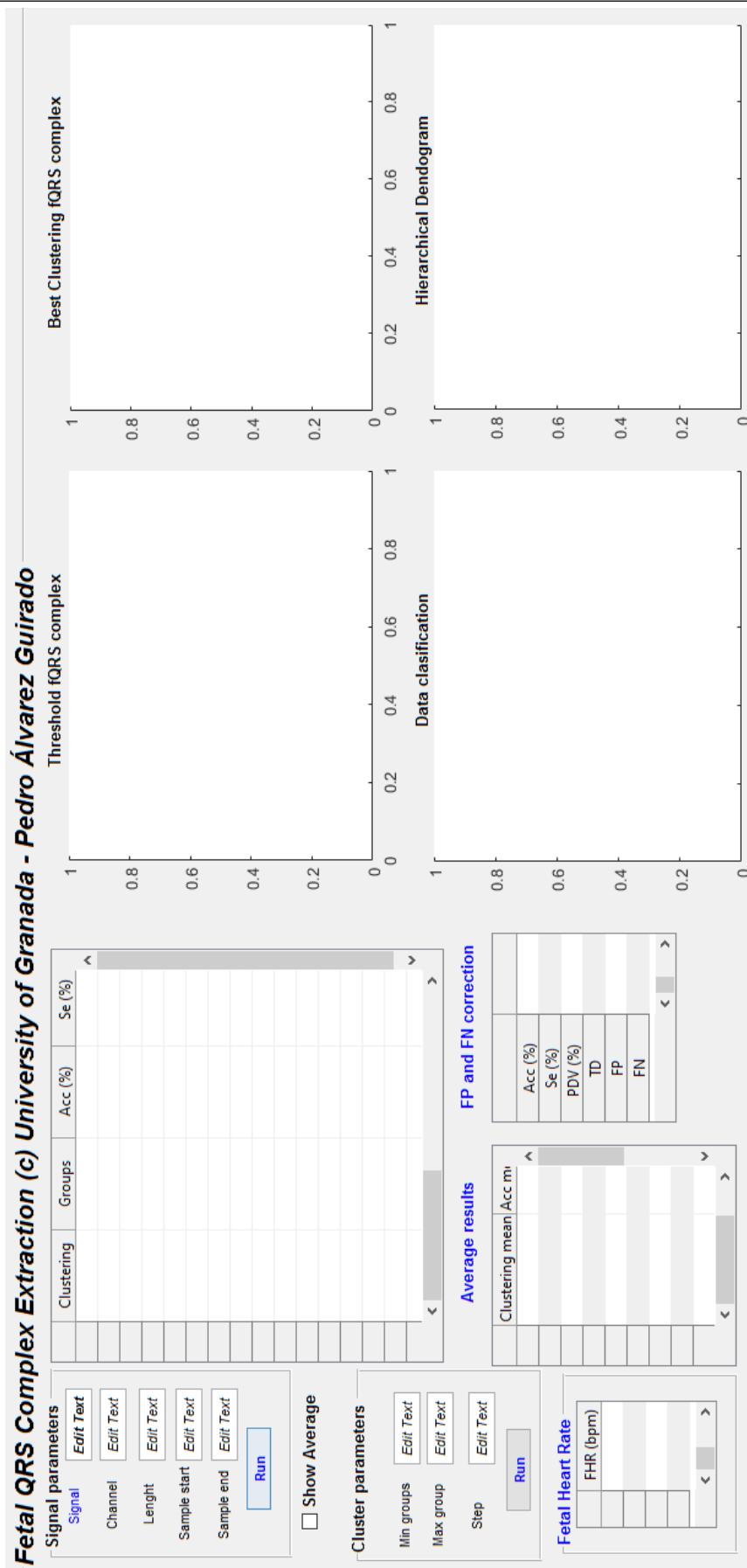


Figura 7.1: Aspecto de la GUI evaluadora de algoritmos de Clustering en señales aECG antes de ejecutarse

## 7. EVALUACIÓN DE RESULTADOS

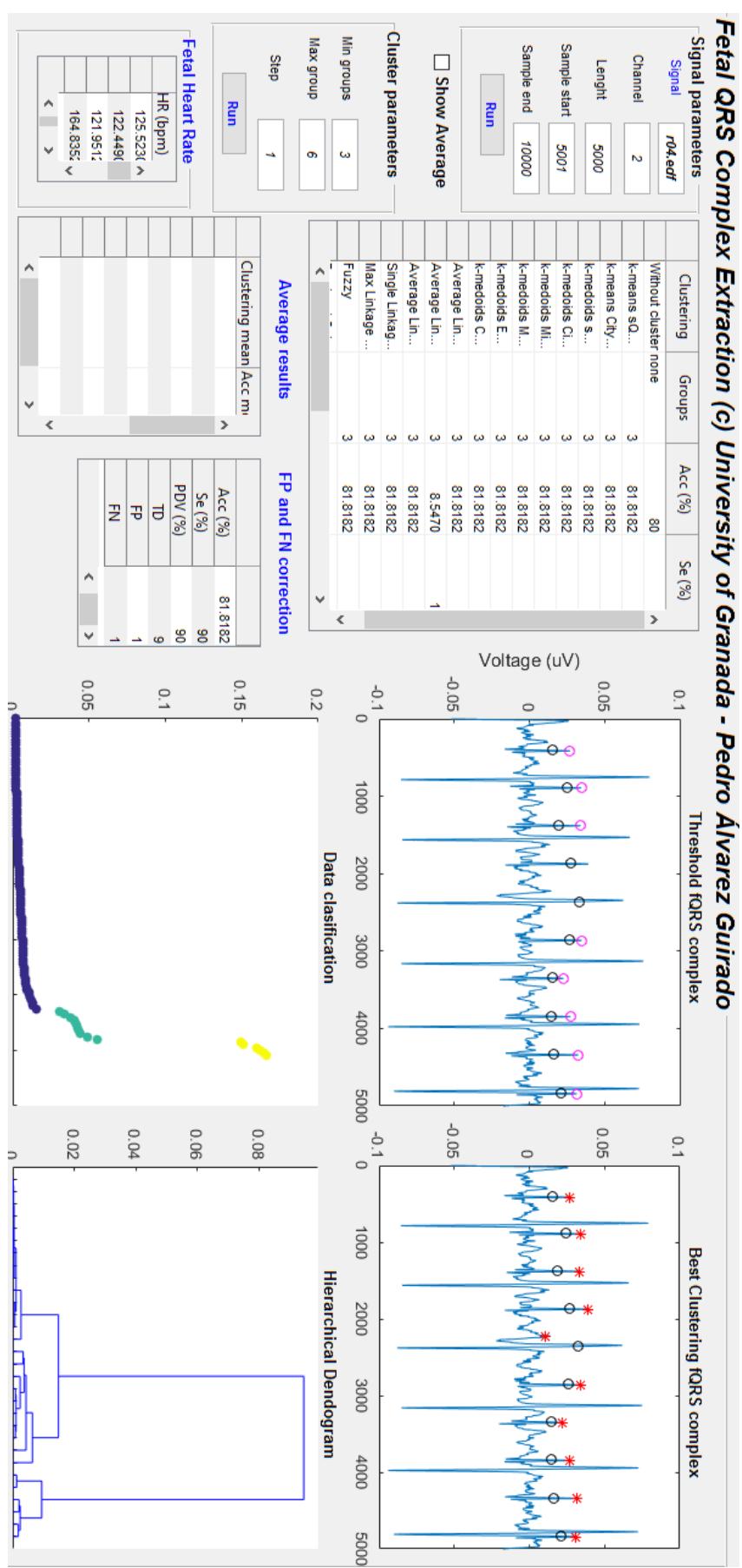


Figura 7.2: Aspecto de la GUI evaluadora de algoritmos de Clustering en señales aECG una vez ejecutada, con los resultados y gráficos de una señal.

### 7.1.1. Interfaz y secciones

Nuestra GUI está dividida principalmente en dos paneles, un panel encargado de preprocessar la señal utilizando la WT y extraer los complejos QRS fetales anotados por el equipo de cardiólogos de *Physionet*, y otro panel encargado de la clasificación en sí de nuestras señales. Detalladamente:

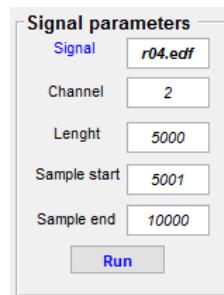


Figura 7.3: Panel de parámetros para el preprocessamiento de la señal de nuestra GUI

- **Parámetros de la señal (*Signal parameters*):** en este panel, mostrado en la Figura 7.3, nos encargamos de introducir los parámetros de la señal que deseamos clasificar. Los recuadros de texto a completar son los siguientes:

- Señal (*Signal*): aquí introduciremos la señal deseada a analizar. Para ello, primeramente deberemos de introducir el nombre del archivo que contiene todas las muestras de la señal que queremos analizar. Tener en cuenta que es necesario ejecutar la GUI en el mismo directorio en el que se encuentren los archivos de señal aECG. Este nombre se introducirá de la siguiente forma: *nombrearchivo.edf*. Por ejemplo, *r04.edf*. No necesitamos introducir comillas ni otros signos de puntuación.
- Canal (*Channel*): en este recuadro introduciremos el canal en el que queremos analizar nuestra señal. Cada canal se corresponderá con un electrodo colocado en el abdomen materno, tal y como se detalló en la Figura 6.2. El número introducido se corresponderá con el número de columna de esta figura.
- Longitud (*Length*): aquí configuraremos la longitud de la ventana de muestreo utilizada para preprocessar la señal mediante la WT. Nuestros algoritmos están pensados mayormente para clasificar señales con una longitud

## 7. EVALUACIÓN DE RESULTADOS

---

de ventana  $l = 5000$ , no siendo posible reducir esta longitud. Sí será posible aumentar la longitud del tamaño de nuestra ventana, pero debido a la necesidad de implementar estos algoritmos en hardware y ser capaz de clasificar las señales aECG en tiempo real, necesitaremos una ventana lo menor posible. El número a introducir aquí ,será, por ejemplo, 5000.

- Muestra de comienzo de ventana (*Sample start*): en el recuadro se introducirá la muestra por la que queremos comenzar nuestra señal.
- Muestra de final de ventana (*Sample end*): aquí introduciremos la muestra en la que queremos que acabe nuestra ventana. Es necesario que la diferencia entre ambas, la muestra de inicio y la final sea igual a la longitud de la ventana para que nuestro algoritmo pueda operar correctamente.  
 $sample_{end} - sample_{start} = length$ .
- Botón de ejecución (*Run*): una vez completados todos los cuadros anteriores con sus parámetros correctamente, pulsaremos este botón para preprocesar nuestra señal y obtener los complejos fQRS anotados en la señal.

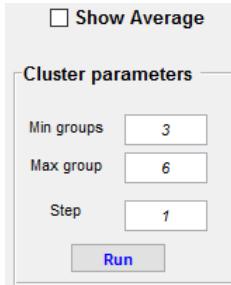


Figura 7.4: Panel de parámetros para la clasificación de la señal de nuestra GUI. Aparece también, la casilla de activación de los resultados de *Accmedia*.

- **Parámetros de clasificación** (*Cluster parameters*): una vez preprocesada nuestra señal correctamente, es necesario realizar la clasificación de la misma para hallar sus complejos fQRS. En este panel no es posible elegir los métodos a clasificar, ya que es necesario probar todos los algoritmos expuestos en esta memoria, para proceder al hallazgo de resultados. Si que podremos optimizar los parámetros de clasificación para las señales, es decir, introduciremos los parámetros para clasificar nuestra señal, y en base a estos los métodos para hallar

## 7.1 GUI evaluadora de algoritmos de Clustering en señales aECG

---

correctamente el número de grupos (criterio de DB y de la silueta) determinaran el número de grupos correctos para cada señal. Básicamente, tenemos como parámetros:

- Mínimo número de grupos (*Min groups*): en este cuadro introduciremos el mínimo número de grupos que queremos clasificar. Tener en cuenta que si queremos clasificar fQRS es necesario introducir en este parámetro el número 3. Si por otra parte, únicamente quisiéramos clasificar complejos QRS maternos, podríamos introducir  $\text{mín} = 2$ .
- Máximo número de grupos (*Max groups*): introduciremos el número máximo de grupos que deseamos clasificar. Tener en cuenta que, debemos de tener cuidado a la hora de introducir un número muy alto en este cuadro, debido a que corremos el riesgo de que, tal y como vimos en el capítulo anterior, nuestra señal no se clasifique correctamente. Además, aumentaremos el tiempo de ejecución de nuestros algoritmos, debido a la proporcionalidad de estos con el número de grupos.
- Pasos para clasificar nuestra señal entre el mínimo y máximo número de grupos a clasificar (*Steps*): aquí, introduciremos el número de pasos que queremos dar entre los dos parámetros anteriores. Normalmente este valor será igual a uno.

## 7. EVALUACIÓN DE RESULTADOS

---

The screenshot displays three windows from a software application:

- Top Window:** A table titled "Clustering" with columns "Groups", "Acc (%)", and "Se (%)". It lists various clustering methods and their performance metrics. One row shows "Without cluster none" with Acc (%) at 80.
- Middle-Left Window:** A table titled "Average results" with columns "Clustering mean" and "Acc mean". It shows the average accuracy for each method.
- Middle-Right Window:** A table titled "FP and FN correction" with columns "Acc (%)", "Se (%)", "PDV (%)", "TD", "FP", and "FN". It provides detailed metrics for the overall classification performance.

Figura 7.5: Cuadros de resultados de nuestra GUI: Cuadro de resultados instantáneos (Cuadro superior), Cuadro de resultados medios (Cuadro inferior izquierda) y Cuadro con los resultados tras la corrección de FP y FN (Cuadro inferior derecha).

- **Casilla de activación de los valores medios de los resultados (Show Average):** activando esta casilla mostraremos en el Cuadro 2 la media aritmética de todas las ejecuciones guardadas, desde la primera vez que se ejecuto este programa, en nuestro archivo *excel*. Se deberá de activar la casilla antes o después de ejecutar la clasificación, debido a que si se realiza la clasificación y en ese instante se activa esta casilla, podremos tener problemas a la hora de leer nuestros datos. Tener en cuenta que la activación de esta casilla aumenta el tiempo de ejecución de nuestro programa, por lo que si se desean tomar medidas rápidas, o únicamente llevamos pocas medidas es recomendable no activarla.
- **Cuadro de resultados instantáneos :** este Cuadro mostrará los resultados de bondad de la clasificación realizada en la señal elegida. Entre sus parámetros podemos encontrar los siguientes: *Acc (%)*, *Se (%)*, *PDV (%)*, *TD*, *FP*, *FN*, *k* y *coste computacional (s)*.

## 7.1 GUI evaluadora de algoritmos de Clustering en señales aECG

---

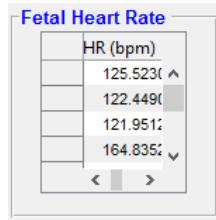


Figura 7.6: Cuadro con los valores del vector de frecuencia cardíaca en (latidos/min)

- **Cuadro de valores medios de los parámetros de bondad (Average results):** en este Cuadro se mostrará únicamente el método utilizado y su *Acc media*, es decir, la media aritmética de todos los valores de *Acc* obtenidos durante nuestras simulaciones. Se ha decidido mostrar únicamente la media del parámetro *Acc* debido al alto coste computacional que acarreaba leer todos los parámetros del fichero *excel*. Relacionado con esto, se ha añadido un botón de activación para poder mostrar el valor medio de *Acc*.
- **Cuadro con los parámetros de bondad tras la realización de la corrección de FP y FN (FP and FN correction):** con el método que mejores resultados de *Acc<sub>instantanea</sub>*, realizaremos su corrección de FP y FN. Más adelante explicaremos como se ha realizado la corrección en sí.
- **Cuadro del vector de pulso cardíaco (Fetal Heart Rate):** en este Cuadro se mostrará, una vez realizada la corrección de FP y FN de nuestra señal, el vector con todas las FHR extraídas cada dos pulsos cardíacos.
- **Gráfica de la señal aECG con los complejos fQRS extraídos del método por umbrales:** en la gráfica se muestran los complejos fQRS extraídos, marcados en la señal mediante una circunferencia de color magenta, comparándose con los complejos fQRS reales marcados por *Physionet*, marcados en la señal mediante una circunferencia de color negro.

## 7. EVALUACIÓN DE RESULTADOS

---

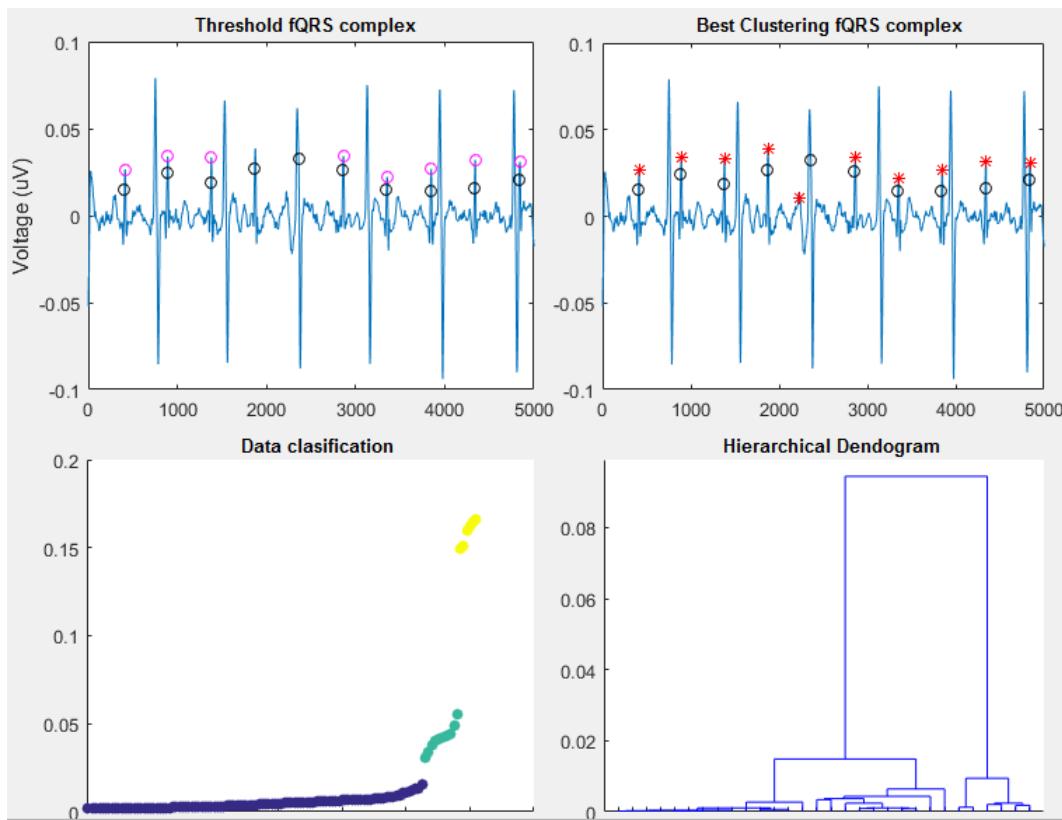


Figura 7.7: Panel con los gráficos resultantes de la clasificación de nuestra señal: clasificación mediante umbral (esq. superior izq.), clasificación mediante clustering (esq. superior der.), representación de los grupos clasificados mediante clustering (esq. inferior izq.) y dendograma jerárquico de clasificación (esq. inferior der.).

- **Gráfica de la señal aECG con los complejos fQRS extraídos mediante el algoritmo con mejor  $Acc_{instantánea}$** : en la gráfica se muestran los complejos fQRS extraídos mediante *Clustering*, marcados en la señal mediante un asterisco rojo, comparándose con los complejos fQRS reales marcados por *Physionet*, marcados en la señal mediante una circunferencia de color negro.
- **Clasificación de los datos máx-mín realizada con el algoritmo con mejor  $Acc_{instantánea}$** : aquí se muestran los datos máx-mín, ordenados por amplitud, clasificados en los distintos grupos. Para una visualización más sencilla se han asignado diferentes colores a los distintos grupos clasificados. Tener en cuenta que la clasificación realizada con los datos sin ordenar proporcionará los mismos resultados, pero por cuestiones de fácil diferenciación de grupos se ha representado de esta forma.

- **Dendograma de los datos máx-mín de la señal realizado mediante el algoritmo jerárquico:** como último gráfico, hemos representado el dendograma de nuestra señal mediante el algoritmo de clasificación Jerárquico con distancia media. Esta gráfico nos será de gran utilidad a la hora de ver la jerarquía de clasificación de nuestros datos.

### 7.1.2. Pasos de utilización de la GUI y consideraciones a tener en cuenta

Antes de proceder a la utilización de nuestra herramienta es necesario que nos encontremos en el directorio en el que se encuentren todas las funciones necesarias así como el propio fichero del ejecutable de la GUI. Una vez tengamos los ficheros y archivos descargados procederemos a clasificar nuestras señales. Para ello, seguir los siguientes pasos:

#### Signal parameters

1. Introducir el nombre del fichero que contiene la señal deseada a clasificar en el cuadro *Signal*. Para añadir nuevas señales a nuestro banco de datos únicamente es necesario guardarlas en esta ubicación.
2. Introducir el canal que deseamos analizar en el texto disponible en el cuadro *Channel*. En nuestras señales únicamente dispondremos de 4 canales abdominales, por lo que solamente podremos utilizar cuatro canales.
3. A continuación, debemos de introducir la longitud de nuestra ventana, *Length*. Tal y como se comentó anteriormente el valor típico y a su vez mínimo, para este campo es 5000. Aunque podremos aumentar este valor en periodos del mismo, 10000, 15000, etc, es siempre recomendable trabajar con este valor.
4. Una vez introducida nuestra longitud, es necesario especificar la muestra o punto por el que queremos que comience la ventana, por lo que en el cuadro *Sample start* introduciremos este valor.
5. Necesitaremos además, introducir la muestra final de nuestra ventana. Por lo que, cumpliendo con  $sample_{end} - sample_{start} = length$ , introduciremos este valor, el cuál no será más que:  $sample_{end} = length - sample_{start}$ .

## 7. EVALUACIÓN DE RESULTADOS

---

6. Con todos nuestros datos introducidos es el momento de pulsar el botón *Run* perteneciente al panel *Signal parameters*, para que podamos preprocesar nuestra señal. Durante un tiempo aproximado de  $t = 120\text{ s}$  nuestra GUI preprocesará nuestra señal hasta mostrar en el Gráfico 7.7 la señal aECG con los complejos fQRS marcados por el algoritmo inicial del umbral.
7. Cuando esta gráfica ha aparecido, es el momento de realizar la clasificación de nuestra señal. Para ello, el primer parámetro a introducir en el cuadro *Min groups* es el mínimo número de grupos que querramos clasificar, en nuestro caso  $\min = 3$ .
8. También se debe de especificar el máximo número de grupos a clasificar. Este parámetro puede ser variable, pero siempre tendremos que tener presente la ineficiencia de los algoritmos de averiguamiento del parámetro  $k$  para valores altos de  $k$ . En nuestro caso, un valor frecuente será  $k = 6$ .
9. Como último parámetro a introducir, introduciremos en *Step* el valor de los pasos a dar en nuestra clasificación. Este valor deberá ser por defecto  $step = 1$ , ya que, debido a que probaremos con un rango pequeño, no es necesario aumentar el número de pasos.
10. Como último paso pulsaremos el botón *Run* perteneciente al panel *Cluster parameters*, para proceder a la clasificación de nuestras señales. Este paso supondrá un coste aproximado temporal de  $t = 60\text{ s}$ , tiempo menor que el tiempo de preprocesado de la señal. Finalizará cuando se muestren el resto de gráficos en nuestra GUI, así como el contenido de los Cuadros de datos.
11. Como parámetro opcional, podremos activar la casilla *Show Average*, la cuál mostrará los resultados medios de la Acc procedentes del fichero de excel en el que se han guardado todos los resultados.
12. Por último, se nos mostrará una Figura con la señal aECG con los complejos fQRS finales, tras la corrección de FP y FN

## 7.2. Resultados

Una vez implementada nuestra GUI, el siguiente paso fue realizar la clasificación de todas las señales posibles. Para ello, únicamente introdujimos los parámetros necesarios de cada señal en nuestra GUI, almacenándose automáticamente los resultados generados. Relacionado con los criterios de evaluación del número correcto de grupos, hemos implementado únicamente el criterio de Davies-Bouldin y el de la silueta, debido a que se ha comprobado que estos dos criterios eran los que mayor Acc nos aseguraban en nuestra clasificación. En el algoritmo de decisión, ante resultados diferentes para la estimación de  $k$  nos hemos decidido por el menor de ellos.

Relacionado con el tiempo de ejecución de los algoritmos, tener en cuenta que todos los algoritmos de clustering se ejecutan en un promedio de 20 repeticiones por algoritmo, para asegurarnos del hallazgo del mínimo absoluto de la función objetivo del algoritmo en cuestión. Algunos algoritmos, como por ejemplo los jerárquicos y el algoritmo de Conjuntos dominantes se ejecutan únicamente una vez.

Por lo tanto, procedimos a realizar una evaluación completa en todos los rangos de las señales de la base da datos. En la memoria, únicamente se muestran varias trozos de cada señal clasificados, no mostrándose todos.

Como podemos observar en los resultados mostrados, nos encontramos ante una ardua tarea a la hora de definir que algoritmo ofrece una mejor clasificación para todas nuestras señales. No podremos definir un único algoritmo como “ideal” para nuestros datos, debido a que dependiendo de diversos factores, entre los que destacará el ruido presente en nuestra señal y la diferencia de amplitudes entre los complejos QRS maternos y fetales, habrá algoritmos que funcionen mejor bajo ciertas condiciones. Pero sí que podemos definir el valor medio de *Acc* de nuestros resultados, asegurándonos que, en término medio, podremos encontrar un algoritmo que funcione mejor que el resto.

## 7. EVALUACIÓN DE RESULTADOS

---

| ALGORITMO   | $Acc_{medio}$ (%) | Coste computacional (s) |
|---|-------------------|-------------------------|
| <i>Algoritmo mediante umbral</i>                    | 68.13             | 1.3713                  |
| <i>k-means con distancia Euclidea al cuadrado</i>   | 72.64             | 0.0713                  |
| <i>k-means con distancia Cityblock</i>              | 55.18             | 0.1166                  |
| <i>k-medoids con distancia Euclidea al cuadrado</i> | 74.58             | 0.3856                  |
| <i>k-medoids con distancia Cityblock</i>            | 61.84             | 0.3429                  |
| <i>k-medoids con distancia de Minkowski</i>         | 55.24             | 0.3102                  |
| <i>k-medoids con distancia de Mahalanobis</i>       | 55.26             | 0.4022                  |
| <i>k-medoids con distancia Euclidea</i>             | 55.24             | 0.3368                  |
| <i>k-medoids con distancia de Chebychev</i>         | 55.24             | 0.3294                  |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      | 65.34             | 0.0044                  |
| <i>Jerárquico Linkage con distancia del coseno</i>  | 9.78              | 0.0060                  |
| <i>Jerárquico Linkage con distancia Cityblock</i>   | 23.67             | 0.0038                  |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   | 23.67             | 0.0041                  |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   | 71.23             | 0.0034                  |
| <i>Fuzzy c-means</i>                                | 70.27             | 0.0083                  |
| <i>Conjuntos dominantes</i>                         | 19.02             | 0.1040                  |

Cuadro 7.1: Valores medios de precisión y coste computacional de los algoritmos de Clustering.

Por ello, si observamos los resultados medios del Cuadro 7.1 vemos como hay varios algoritmos que ofrecen unos resultados no válidos. Especialmente dos algoritmos muy por debajo de los resultados esperados, los cuales son:

- Algoritmo jerárquico Linkage con distancia del coseno: queda claro que la utilización de una medida de similitud, como es la similitud del coseno, no es válida para nuestro propósito, debido a que esta medida calcula la similitud existente entre dos vectores o puntos, en el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar. Por otra parte, la existencia de cualquier ángulo entre los vectores proporcionará un valor menor que 1. En nuestros datos, el resultado será igual a cero en la mayoría de los casos. No siendo válida esta métrica para clasificar nuestros datos.
- Conjuntos dominantes: el algoritmo de Marcello Pelillo, el cual se basa en conjuntos dominantes según la teoría de grafos, no ha sido válido para clasificar nuestra señal aECG.

Descartados estos dos algoritmos, observamos como hay una serie de algoritmos con valores por debajo del algoritmo de clasificación mediante umbrales. Estos algoritmos son:

- *k-means* con distancia *Cityblock*
- *k-medoids* con distancia *Cityblock*, de *Minkowski*, *Mahalanobis*, *Chebychev* y Euclídea
- Jerárquico *Linkage* con distancia *Cityblock*
- Jerárquico *Linkage* con distancia mínima Euclídea al cuadrado

Concluimos por tanto, que la utilización de distancias, que no sean la **distancia Euclídea al cuadrado**, empeoran nuestra clasificación con respecto al método mediante umbrales. En el caso particular de la distancia de *Minkowski* con  $n = 3$ , los resultados coinciden con los medidos con el caso de distancia de *Minkowski*  $n = 2$ , es decir la distancia Euclídea. Esto es debido a que al ser nuestros datos unidimensionales, ambas métricas quedarán iguales. En teoría, la distancia Euclídea al ser una medida de similitud completa, es decir, cumple todas las condiciones para ello, incluido la desigualdad triangular, debería de funcionar mejor que el caso de la distancia Euclídea al cuadrado. Pero, según los resultados mostrados, esto no será así. Una sencilla explicación viene dada por el mero hecho de elevar al cuadrado las distancias. Esto quiere decir que, si nos encontramos con puntos muy distantes, si elevamos su distancia Euclídea al cuadrado, la distancia aumentará cuadráticamente, pero esta distancia a su vez será mucho menor que la elevación al cuadrado de la distancia entre puntos distantes. Así que, comparando ambas métricas, distancia Euclídea contra Euclídea al cuadrado, habrá más diferencia entre las distancias de puntos disimilares en el caso de la Euclídea al cuadrado. Este aspecto favorecerá nuestra extracción, ya que a mayor distancia entre puntos diferentes, más facilidades para nuestra clasificación. Debido a que trabajamos en el espacio Euclídeo unidimensional, el resto de distancias nos ofrecerán los mismos resultados, ya que básicamente, todas las distancias excepto la distancia de *Mahalanobis*, serán la distancia Euclídea. La distancia de *Mahalanobis* es utilizada principalmente para identificar valores atípicos, debido a que toma en cuenta las diferentes escalas entre las variables y las correlaciones entre éstas. El problema es que en nuestros datos, rara vez hay valores atípicos entre los maternos y fetales. Por lo que esta clasificación, desfavorecerá el hecho de que aparezcan complejos QRS con una amplitud diferente a la varianza del resto, pero que a su vez pertenezca a este

## 7. EVALUACIÓN DE RESULTADOS

---

grupo. Esta es la razón por la que esta distancia no ofrecerá resultados válidos para nuestra clasificación en estas señales aECG.

Atendiendo a los algoritmos en sí, y obviando las métricas, el algoritmo jerárquico con la utilización de la distancia mínima entre puntos para clasificar datos, es el que peores resultados en comparación con los demás algoritmos jerárquicos. Esto es debido a que, si únicamente computamos la distancia mínima entre dos grupos, tendremos más probabilidades de agrupar los datos incorrectamente que si computamos la máxima distancia entre grupos, ya que al obviar datos, podemos agrupar un grupo con otro porque ambos tengan dos puntos muy cercanos, aunque la distancia media fuera menor si se agrupara este con otro grupo.

Por otra parte, los algoritmos que ofrecen una precisión igual o superior al método mediante umbrales son:

- ***k-means* con distancia Euclídea al cuadrado:** se ajusta bastante bien a la mayoría de los casos de clasificación, tanto para amplitudes de los complejos fQRS pequeñas como para amplitudes con valores en torno a los complejos maternos. Es necesario ejecutarlo varias veces para asegurarnos del hallazgo del mínimo absoluto de la suma total de distancias entre puntos y centroides.
- ***k-medoids* con distancia Euclídea al cuadrado:** comparte las ventajas del algoritmo *k-means*, pero con la particularidad de la ubicación del centroide. Este es el algoritmo que mejores resultados nos ofrece, debido principalmente a la ubicación del centroide, en este caso, medoide, el cuál pertenecerá expresamente a nuestros datos, lo que facilitará la clasificación y la agrupación de datos.
- **Jerárquico *Linkage* con distancia Euclídea al cuadrado:** en este algoritmo se mide la distancia media entre puntos, siendo un algoritmo aglomerativo. La ventaja es que únicamente es necesario ejecutarlo una vez, ahorrando en coste computacional. Su mayor problema es que no permite la reordenación de datos una vez efectuados.
- **Jerárquico *Linkage* con distancia máxima Euclídea al cuadrado:** este algoritmo ofrece mejores resultados que el Jerárquico basado en la distancia media, ya que en este caso medimos la máxima distancia entre clusteres, evitando así desagrupar datos lejanos que pertenezcan a un mismo grupo.
- ***Fuzzy c-means* con distancia Euclídea al cuadrado:** algoritmo parecido al *k-means* pero con la particularidad de la pertenencia probabilística de cada dato

a todos los puntos. Este algoritmo funciona bien con datos que se encuentran en las fronteras de los clusteres.

Como vemos, la semejanza principal de estos algoritmos es el uso que hacen de la **distancia Euclídea al cuadrado** como función de disimilitud. Cualquiera de estos algoritmos puede ofrecernos mejores resultados que el método de los umbrales, cada uno con sus características particulares. Pero, destacaremos especialmente, los algoritmo ***k-means*** y ***k-medoids*** con distancia Euclídea al cuadrado, debido a que serán estos los que nos ofrezcan mejores resultados ( $Acc_{media_{k-means}} = 72,64$ ,  $Acc_{media_{k-medoids}} = 74,58$ ).

Estos resultados, confirman el porqué de la utilización de estos dos métodos en diversos campos, siendo por tanto los dos algoritmos más utilizados a día de hoy, tanto por su bajo coste computacional como por sus prestaciones. Destacaremos también, la reducción importante llevada a cabo en el apartado de complejidad computacional, reduciendo nuestro coste computacional de  $1,37\text{ s}$  de media, utilizando el algoritmo mediante umbrales, a en torno a  $0,40\text{ s}$  en el caso de utilización del algoritmo ***k-medoids***. Por lo que lograremos un incremento de la eficiencia :

$$Eficiencia_{comp}(\%) = \frac{Coste_{umbrales}}{Coste_{k-medoids}} = \frac{1,37\text{ s}}{0,40\text{ s}} \times 100 = 342\%$$

La implementación de este nuevo algoritmo consumirá 3,42 veces menos tiempo que el algoritmo inicial, lo que facilitará nuestra implementación hardware, y a su vez, mejorará la percepción de clasificación en tiempo real en las pruebas del futuro prototipo final.

Aunque no se ha especificado en la memoria explícitamente, se ha comprobado que utilizando diferentes longitudes de ventanas para preprocesar nuestra señal, a mayor longitud de la ventana utilizada, mayor *Acc* obtenida. Esto es debido a que cuanto más datos haya disponibles, más probabilidades tendremos de clasificar estos correctamente. También hay que tener en cuenta que el aumento de la longitud de nuestra ventana aumentará el tiempo de procesamiento y clasificación de nuestro sistema. Debido a esto, se ha decidido utilizar **ventanas de 5000 muestras** para extraer los complejos de nuestra señal, reduciendo al máximo el coste computacional de nuestros algoritmos.

## 7. EVALUACIÓN DE RESULTADOS

---

### 7.2.1. Corrección de FP y FN

Debemos de tener en cuenta que, estos resultados han sido proporcionados sin realizar una corrección de FP y FN, la cuál se realiza una vez escogido el mejor método, aumentando nuestra precisión notablemente. Una vez escogidos ambos métodos, los resultados de precisión media con la corrección realizada fueron los mostrados en el Cuadro 7.2.

| ALGORITMO   | $Acc_{medioFPyFN} (\%)$ |
|---|-------------------------|
| <i>k-means con distancia Euclidea al cuadrado</i>   | 86.64                   |
| <i>k-medoids con distancia Euclidea al cuadrado</i> | 88.58                   |

Cuadro 7.2: Valores medios de exactitud tras la corrección de FP y FN.

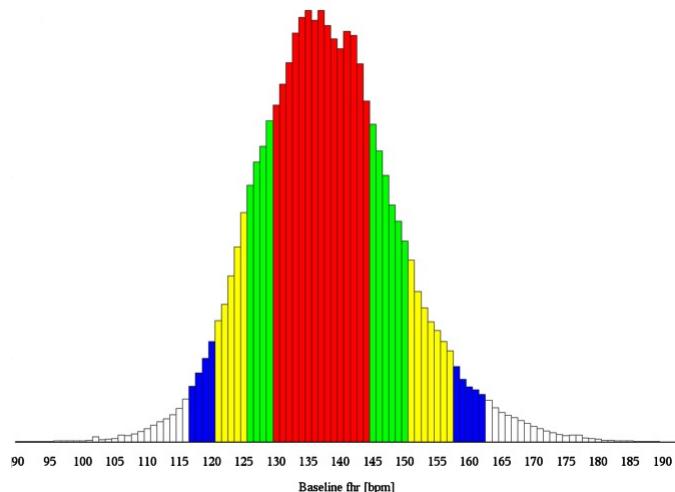


Figura 7.8: Valores medios para el FHR medidos en bpm

La corrección de FP y FN se efectuó de la siguiente manera:

1. Una vez clasificada nuestra señal, observábamos su vector FHR, midiendo las diferencias entre los complejos fQRS detectados, sabiendo que su rango, el cual cumple una campana de Gauss, oscila entre 115 y 165 bpm. Si el valor medido fuera mayor o menor que este rango se aplicaría la corrección a la señal.

2. Esta corrección comenzaría con la eliminación de los FP de nuestra señal. Es decir, si el valor FHR instantáneo fuera menor que 115 bpm eliminaríamos el valor con menor amplitud de los detectados en ese tramo de señal, conservando el dato con mayor amplitud máx-mín.
3. Si por el contrario, el valor FHR fuera superior al límite superior, 165 bpm, introduciríamos un nuevo complejo fQRS en nuestra señal. El método para introducirlo es el siguiente: localizaríamos el punto medio entre los dos complejos fQRS que nos han dado un valor FHR superior, y estableceríamos un rango de muestras en torno al cuál buscaríamos el dato con mayor amplitud. En muchas ocasiones se seleccionará un complejo QRS materno, debido a que este será el que mayor amplitud posea, pero en función de la experiencia, destacamos que en todas las señales hay al menos un complejo QRS materno superpuesto en amplitud a uno fetal. Por lo que en la mayoría de los casos, este complejo fQRS añadido se detectará como un punto correctamente clasificado.

Una vez realizada la corrección de FP y FN se mostrará el resultado final, marcando los complejos fQRS en la señal aECG, tal y como podemos observar en la Figura 7.2.1.

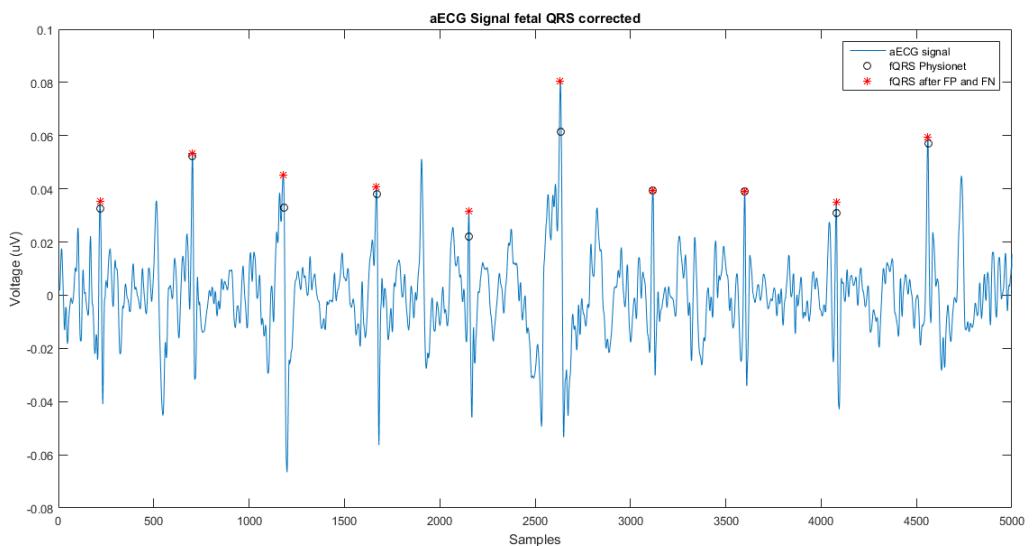


Figura 7.9: Señal aECG *r08.edf* entret = 5s y t = 10s clasificada tras la corrección de FP y FN.

A continuación se exponen una serie de resultados obtenidos de la simulación en varios trozos de las señales analizadas. Por motivos de espacio, únicamente se han expuesto cuatro ejemplos de resultados por señal.

## **7. EVALUACIÓN DE RESULTADOS**

---

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r01.edf</i>                                      | 1      | 5000  | 3     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 76.32   | 90.91  | 83.33   | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 19.65   | 100    | 19.65   | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 78,57   | 100    | 78,57   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 68.75   | 100    | 68.75   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 78,57   | 78     | 50      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 61.12   | 100    | 61.12   | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

Cuadro 7.3: Resultados de clasificación en la señal *r01.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |            |           |            |        |
|---|--------|-------|-------|------------|-----------|------------|--------|
| <i>r01.edf</i>                                      | 5001   | 10000 | 4     |            |           |            |        |
| ALGORITMO   |        |       |       | Acc<br>(%) | Se<br>(%) | PDV<br>(%) | Grupos |
|   |        |       |       |            |           |            |        |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 0          | 0         | 0          | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 75         | 81.82     | 90         | 3      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 75         | 81.82     | 90         | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 75         | 81.82     | 90         | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 75         | 81.82     | 90         | 3      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 75         | 81.82     | 90         | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 75         | 81.82     | 90         | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 21.74      | 90.91     | 22.22      | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 70.1       | 81.82     | 50         | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 52.94      | 81.82     | 60         | 3      |
| <br>  |        |       |       |            |           |            |        |
| <i>Corrección FP y FN</i>                           |        |       |       | 90.91      | 90.91     | 100        | 3      |

Cuadro 7.4: Resultados de clasificación en la señal *r01.edf*

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r01.edf</i>                                      | 20001  | 25000 | 4     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 0       | 0      | 0       | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 91.67   | 100    | 91.67   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 100     | 100    | 100     | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 4.16    | 100    | 4.16    | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 78.57   | 100    | 78.57   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 78.57   | 100    | 78.57   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 100     | 100    | 100     | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 61.8    | 61.11  | 50      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 6.11    | 18.18  | 10      | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

 Cuadro 7.5: Resultados de clasificación en la señal *r01.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |            |           |            |        |
|---|--------|-------|-------|------------|-----------|------------|--------|
| <i>r01.edf</i>                                      | 70001  | 75000 | 4     |            |           |            |        |
| ALGORITMO   |        |       |       | Acc<br>(%) | Se<br>(%) | PDV<br>(%) | Grupos |
|   |        |       |       |            |           |            |        |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 6.67       | 9.09      | 20         | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 29.17      | 63.64     | 35         | 6      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 29.17      | 63.64     | 36.84      | 6      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 30.43      | 63.64     | 36.84      | 6      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 9.43       | 9.09      | 100        | 6      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 29.17      | 63.64     | 35         | 6      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 29.17      | 63.64     | 35         | 6      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 30.43      | 35        | 36.84      | 6      |
| <i>Fuzzy c-means</i>                                |        |       |       | 29.17      | 63.64     | 50         | 5      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 29.17      | 63.64     | 35         | 5      |
| <i>Corrección FP y FN</i>                           |        |       |       | 43.75      | 63.63     | 58.4       | 6      |

Cuadro 7.6: Resultados de clasificación en la señal *r01.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r04.edf</i>                                      | 20001  | 25000 | 4     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 72.73   | 9.09   | 100     | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 75      | 81.81  | 90      | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 15.78   | 81.81  | 16.36   | 3      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 75      | 81.81  | 90      | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 15.78   | 81.81  | 16.36   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 15.78   | 81.81  | 16.36   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 15.78   | 81.81  | 16.36   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 15.78   | 81.81  | 16.36   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 15.78   | 81.81  | 16.36   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 75      | 81.81  | 90      | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 5.64    | 90.90  | 5.68    | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 6.25    | 9.09   | 16.67   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 6.25    | 9.09   | 16.67   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 75      | 90     | 50      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 2.50    | 9.09   | 3.33    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 90.91   | 90.91  | 100     | 3      |

Cuadro 7.7: Resultados de clasificación en la señal *r04.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r04.edf</i>                                      | 40001  | 45000 | 4     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 80      | 80     | 100     | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 64.28   | 90     | 69.23   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 90      | 90     | 100     | 4      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 69.23   | 90     | 75      | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 90      | 90     | 100     | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 10.46   | 90     | 10.58   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 10.46   | 90     | 10.58   | 3      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 10.46   | 90     | 10.58   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 10.46   | 90     | 10.5    | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 3.40    | 100    | 3.401   | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 90      | 90     | 100     | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 60      | 64.28  | 50      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 6.06    | 57.40  | 6.66    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 90.91   | 90.91  | 100     | 3      |

Cuadro 7.8: Resultados de clasificación en la señal *r04.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r04.edf</i>                                      | 5001   | 10000 | 3     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 80      | 80     | 100     | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 90      | 90     | 100     | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 15.78   | 90     | 16.07   | 4      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 90      | 90     | 100     | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 90      | 90     | 100     | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 90      | 90     | 100     | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 90      | 90     | 100     | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 90      | 90     | 100     | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 90      | 90     | 100     | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 90      | 90     | 100     | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 5.20    | 100    | 5.20    | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 90      | 90     | 100     | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 90      | 100    | 50      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 4       | 10     | 6.25    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

Cuadro 7.9: Resultados de clasificación en la señal *r04.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r04.edf</i>                                      | 15001  | 20000 | 3     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 81.81   | 81.81  | 100     | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 83.33   | 90.90  | 90.90   | 4      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 83.33   | 90.90  | 90.90   | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 13.33   | 90.90  | 13.51   | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 14.28   | 90.90  | 14.49   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 14.28   | 90.90  | 14.49   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 83.33   | 90.90  | 90.90   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 83.33   | 90.90  | 50      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 58.33   | 63.63  | 87.5    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 81.81   | 81.81  | 100     | 3      |

Cuadro 7.10: Resultados de clasificación en la señal *r04.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r07.edf</i>                                      | 5001   | 10000 | 3     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 90.90   | 100    | 90.90   | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 90.90   | 100    | 90.90   | 4      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 90.90   | 100    | 90.90   | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 35.71   | 100    | 35.71   | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 90.90   | 100    | 90.90   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 90.90   | 90.90  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 40      | 100    | 40      | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

Cuadro 7.11: Resultados de clasificación en la señal *r07.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r07.edf</i>                                      | 30001  | 35000 | 3     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 70      | 70     | 100     | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 81.81   | 90     | 90.90   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 21.73   | 100    | 90.90   | 4      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 81.81   | 90     | 90      | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 81.81   | 90     | 90      | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 81.81   | 90     | 90      | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 81.81   | 90     | 90      | 3      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 81.81   | 90     | 90      | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 81.81   | 90     | 90      | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 5.23    | 100    | 5.23    | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 5.12    | 100    | 5.12    | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 5.23    | 100    | 5.23    | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 5.23    | 100    | 5.23    | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 5.23    | 100    | 5.23    | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 81.81   | 90     | 90      | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 8.56    | 40     | 9.76    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 81.81   | 90     | 90      | 3      |

Cuadro 7.12: Resultados de clasificación en la señal *r07.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r07.edf</i>                                      | 50001  | 55000 | 3     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 72.72   | 72.72  | 100     | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 81.81   | 81.81  | 100     | 4      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 81.81   | 81.81  | 100     | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 11.90   | 90.90  | 12.04   | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 81.81   | 81.81  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 72.72   | 72.72  | 100     | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 90.91   | 90.91  | 100     | 3      |

Cuadro 7.13: Resultados de clasificación en la señal *r07.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio  | Final  | Canal   |        |
|---|---------|--------|---------|--------|
| <i>r07.edf</i>                                      | 85001   | 90000  | 3       |        |
| ALGORITMO   | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    | 80      | 80     | 100     | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   | 60      | 90     | 64.28   | 3      |
| <i>k-means con distancia Cityblock</i>              | 60      | 90     | 64.28   | 4      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> | 56.25   | 90     | 60      | 3      |
| <i>k-medoids con distancia Cityblock</i>            | 53.33   | 80     | 61.53   | 4      |
| <i>k-medoids con distancia de Minkowski</i>         | 12.50   | 90     | 12.67   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       | 12.50   | 20     | 25      | 3      |
| <i>k-medoids con distancia Euclidea</i>             | 12.50   | 20     | 25      | 3      |
| <i>k-medoids con distancia de Chebychev</i>         | 12.50   | 90     | 12.67   | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      | 72.72   | 80     | 88.88   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  | 4.46    | 100    | 4.46    | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   | 72.72   | 80     | 88.88   | 3      |
| <i>Fuzzy c-means</i>                                | 52.94   | 56.25  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         | 3.57    | 10     | 5.26    | 3      |
| <i>Corrección FP y FN</i>                           | 81.81   | 90     | 90      | 3      |

Cuadro 7.14: Resultados de clasificación en la señal *r07.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r08.edf</i>                                      | 5001   | 10000 | 2     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 0       | 80     | 100     | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 83.33   | 100    | 83.33   | 4      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 71.42   | 100    | 71.42   | 3      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 50      | 70     | 63.63   | 4      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 71.42   | 100    | 71.42   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 71.42   | 100    | 71.42   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 71.42   | 100    | 71.42   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 71.42   | 100    | 71.42   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 71.42   | 100    | 71.42   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 75      | 90     | 81.81   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 10      | 10     | 100     | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 58.33   | 70     | 77.77   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 76.92   | 76.92  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 4.71    | 50     | 4.95    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

Cuadro 7.15: Resultados de clasificación en la señal *r08.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r08.edf</i>                                      | 35001  | 40000 | 4     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 0       | 0      | 0       | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 84.61   | 100    | 84.61   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 84.64   | 100    | 84.64   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 84.61   | 100    | 84.61   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 9.09    | 9.09   | 100     | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 76.92   | 90.91  | 83.33   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 76.92   | 90.91  | 83.33   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 84.65   | 100    | 84.65   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 61.11   | 61.11  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 4.16    | 9.09   | 7.14    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

Cuadro 7.16: Resultados de clasificación en la señal *r08.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r08.edf</i>                                      | 35001  | 40000 | 4     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 0       | 0      | 0       | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 84.61   | 100    | 84.61   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 84.64   | 100    | 84.64   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 61.11   | 100    | 61.11   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 84.61   | 100    | 84.61   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 9.09    | 9.09   | 100     | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 76.92   | 90.91  | 83.33   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 76.92   | 90.91  | 83.33   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 84.65   | 100    | 84.65   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 61.11   | 61.11  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 4.16    | 9.09   | 7.14    | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 100     | 100    | 100     | 3      |

Cuadro 7.17: Resultados de clasificación en la señal *r08.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio  | Final  | Canal   |        |
|---|---------|--------|---------|--------|
| <i>r08.edf</i>                                      | 90001   | 95000  | 4       |        |
| ALGORITMO   | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    | 0       | 0      | 0       | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   | 68.75   | 100    | 68.75   | 3      |
| <i>k-means con distancia Cityblock</i>              | 68.75   | 100    | 68.75   | 3      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> | 68.75   | 100    | 68.75   | 3      |
| <i>k-medoids con distancia Cityblock</i>            | 68.75   | 100    | 68.75   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         | 68.75   | 100    | 68.75   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       | 68.75   | 100    | 68.75   | 3      |
| <i>k-medoids con distancia Euclidea</i>             | 68.75   | 100    | 68.75   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         | 68.75   | 100    | 68.75   | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      | 68.75   | 100    | 68.75   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  | 10.67   | 100    | 10.67   | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   | 57.89   | 100    | 57.89   | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   | 57.89   | 100    | 57.89   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   | 71.42   | 90.90  | 76.92   | 3      |
| <i>Fuzzy c-means</i>                                | 68.75   | 68.75  | 100     | 3      |
| <i>Conjuntos dominantes</i>                         | 83.33   | 90.91  | 90.91   | 4      |
| <i>Corrección FP y FN</i>                           | 90      | 100    | 90      | 4      |

Cuadro 7.18: Resultados de clasificación en la señal *r08.edf*

## 7.2 Resultados

---

| SEÑAL   | Inicio | Final | Canal |         |        |         |        |
|---|--------|-------|-------|---------|--------|---------|--------|
| <i>r10.edf</i>                                      | 90001  | 95000 | 4     |         |        |         |        |
| ALGORITMO   |        |       |       | Acc (%) | Se (%) | PDV (%) | Grupos |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 55.56   | 90.90  | 58.82   | 0      |
| <i>k-means con distancia Euclídea al cuadrado</i>   |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 25      | 90.90  | 25.64   | 3      |
| <i>k-medoids con distancia Euclídea al cuadrado</i> |        |       |       | 55.56   | 90.90  | 58.80   | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>k-medoids con distancia Euclídea</i>             |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>Jerárquico Linkage Euclídea al cuadrado</i>      |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 5.02    | 90.90  | 5.05    | 3      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 5.18    | 90.90  | 5.20    | 3      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 5.18    | 90.90  | 55.55   | 3      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 52.63   | 90.90  | 55.55   | 3      |
| <i>Fuzzy c-means</i>                                |        |       |       | 52.63   | 55.55  | 8.33    | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 4.54    | 9.09   | 90.91   | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 72.72   | 100    | 72.72   | 3      |

Cuadro 7.19: Resultados de clasificación en la señal *r10.edf*

## 7. EVALUACIÓN DE RESULTADOS

---

| SEÑAL   | Inicio | Final | Canal |            |           |            |        |
|---|--------|-------|-------|------------|-----------|------------|--------|
| r10.edf   | 55001  | 60000 | 3     |            |           |            |        |
|   |        |       |       | Acc<br>(%) | Se<br>(%) | PDV<br>(%) | Grupos |
| <b>ALGORITMO</b>                                    |        |       |       |            |           |            |        |
| <i>Algoritmo mediante umbral</i>                    |        |       |       | 36.36      | 36.36     | 100        | 0      |
| <i>k-means con distancia Euclidea al cuadrado</i>   |        |       |       | 16.67      | 81.81     | 17.30      | 3      |
| <i>k-means con distancia Cityblock</i>              |        |       |       | 27.27      | 81.81     | 17.30      | 4      |
| <i>k-medoids con distancia Euclidea al cuadrado</i> |        |       |       | 18.36      | 81.81     | 29.03      | 3      |
| <i>k-medoids con distancia Cityblock</i>            |        |       |       | 27.27      | 81.81     | 17.30      | 4      |
| <i>k-medoids con distancia de Minkowski</i>         |        |       |       | 9.43       | 90.90     | 9.52       | 3      |
| <i>k-medoids con distancia de Mahalanobis</i>       |        |       |       | 9.43       | 90.90     | 9.52       | 3      |
| <i>k-medoids con distancia Euclidea</i>             |        |       |       | 9.43       | 90.90     | 9.52       | 3      |
| <i>k-medoids con distancia de Chebychev</i>         |        |       |       | 9.43       | 90.90     | 9.52       | 3      |
| <i>Jerárquico Linkage Euclidea al cuadrado</i>      |        |       |       | 9.43       | 90.90     | 9.52       | 4      |
| <i>Jerárquico Linkage con distancia del coseno</i>  |        |       |       | 21.42      | 81.81     | 22.50      | 4      |
| <i>Jerárquico Linkage con distancia Cityblock</i>   |        |       |       | 5.18       | 100       | 5.18       | 4      |
| <i>Jerárquico Linkage con distancia mínima sQ</i>   |        |       |       | 9.09       | 9.09      | 100        | 4      |
| <i>Jerárquico Linkage con distancia máxima sQ</i>   |        |       |       | 9.09       | 9.09      | 100        | 4      |
| <i>Fuzzy c-means</i>                                |        |       |       | 40.90      | 81.81     | 45         | 3      |
| <i>Conjuntos dominantes</i>                         |        |       |       | 12.85      | 13.23     | 100        | 3      |
| <i>Corrección FP y FN</i>                           |        |       |       | 90.90      | 90.90     | 100        | 4      |

Cuadro 7.20: Resultados de clasificación en la señal r10.edf

# Parte III

## Conclusiones y trabajo futuro

# Capítulo 8

## Conclusiones y líneas de investigación futuras

### 8.1. Conclusiones

En el presente proyecto se ha realizado un estudio de diversas técnicas de *Clustering* para mejorar la clasificación de complejos QRS fetales, detectados anteriormente mediante una técnica basada en umbrales de amplitud. Se han analizado diversos algoritmos de clasificación en todas las señales con complejos fQRS marcados por especialistas, disponibles en el banco de datos de *Physionet*.

El principal problema de la utilización de la técnica basada en umbrales para detectar nuestros complejos fQRS era la variabilidad de los resultados en función del umbral escogido, ya que no es posible encontrar un valor fijo para estos umbrales, dependiendo estos del nivel de ruido, semanas de gestación, posición del feto, etc. En el estudio realizado en este proyecto se ha pretendido evitar la utilización de una deducción inductiva, la cuál posee una lógica estructural desde la especificación a la generalización, centrándose en la creación de declaraciones generalizadas a partir de sucesos específicos.

Por lo que, para evitar el razonamiento inductivo en la clasificación de complejos fQRS se han utilizado algoritmos estadísticos, sin ningún tipo de semillas, con una inicialización aleatoria para proceder a la clasificación de nuestros datos. Los algoritmos de *Clustering* comparados en el presente documento han sido tanto algoritmos jerárquicos como particionales (*k-means*, *k-medoids*, *fuzzy c-means*, etc.) utilizando diversas funciones de disimilitud o distancia.

Los resultados aquí obtenidos, han sido claramente satisfactorios, ofreciendo un rendimiento superior al algoritmo de detección de complejos fQRS mediante umbrales. Concretamente, ha habido dos algoritmos que han ofrecido una precisión por

encima del resto. Estos dos algoritmos han sido: el algoritmo *k-means* y el algoritmo *k-medoids*, ambos con la utilización de la distancia Euclídea al cuadrado como medida de disimilitud entre datos. Además, de una mayor precisión, el coste computacional de los mismos ha sido menor que el anterior algoritmo inicial de clasificación, mejorándose por tanto la percepción de tiempo ante una posible implementación en tiempo real.

Se ha implementado una GUI en MATLAB para clasificar las señales aECG en tiempo real. Esta GUI muestra los parámetros de bondad de la clasificación realizada por los diversos algoritmos estudiados, ofreciéndonos los resultados instantáneos de estos parámetros, así como los resultados medios de todas las ejecuciones realizadas. Cada clasificación es almacenada automáticamente en un fichero *excel*. Una vez clasificada nuestra señal, la GUI nos mostrará los complejos QRS fetales finales, gracias a la corrección de FP y FN realizada a partir del algoritmo de *Clustering* con el que hemos obtenido mayor precisión instantánea.

## **8.2. Futura implementación hardware**

En el Apéndice A se presenta una futura arquitectura hardware necesaria para la obtención de la FHR a partir de la señal aECG medida en el abdomen materno. En este proyecto se han estudiado diferentes soluciones para la etapa de extracción de los complejos fQRS, contemplando las ventajas e inconvenientes de cada una de ellas, eligiendo finalmente la futura implementación de los algoritmos de clasificación *k-means* y *k-medoids* basados en la distancia Euclídea al cuadrado. Esta arquitectura se basa en el trabajo realizado por [24], utilizando una FPGA, en concreto una de la familia Spartan-6 de Xilinx, el modelo XC6SLX45-3FGG484, disponible en el departamento de Electrónica y Tecnología de Computadores de la Universidad de Granada. Además se presentará una PCB de adquisición de la señal analógica a partir de sensores colocados en el abdomen materno. Como futura mejora a esta implementación, se podría eliminar la utilización de la pantalla LCD (*Liquid Crystal Display*) en el prototipo para ahorrar costes, visualizando los datos en una pantalla externa al prototipo, como puede ser el caso de un *smartphone*.

## **8.3. Líneas de investigación futuras**

En el marco de este Trabajo Fin de Grado se han implementado varios algoritmos de *Clustering* para la extracción de complejos QRS fetales, obteniéndose resultados

## 8. CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS

---

que mejoran los descritos en la bibliografía, basados en otras técnicas de procesado, siendo el presente proyecto innovador en el uso de los algoritmos de *Clustering*.

Este trabajo permitirá el uso de nuevos algoritmos de *Clustering* para extraer estos complejos fatales, como son, los algoritmos de *Clustering* basados en densidades. En ellos, los grupos están definidos en función de las densidades de los datos. En estos algoritmos los datos en áreas esparcidas son clasificados como ruido o puntos frontera. El método de agrupamiento más popular en este grupo, es conocido como **DBSCAN** (*Density-Based Spatial Clustering of Applications with Noise*), el cuál realiza una estimación de la distribución de densidad de los datos, agrupándolos en función de esta estimación. Como característica más importante, destacar que no necesita de la especificación del número de clusteres,  $k$ , pero en su lugar necesita un valor para especificar el umbral de distancia  $\varepsilon$ , y el mínimo número de datos para formar un grupo. Otra propiedad interesante de DBSCAN es que posee una complejidad relativamente reducida. Requiere un número lineal de operaciones,  $O(N \log N)$ , y producirá los mismos resultados en cada iteración, por lo que únicamente será necesario una repetición para clasificar nuestros datos. Por su parte, el algoritmo **OPTICS** (*Ordering Points To Identify the Clustering Structure*) es una generalización de DBSCAN que elimina el problema más significativo de este, la detección de grupos en datos de densidad variable, produciendo un resultado jerárquico de la clasificación.

También, se podrán utilizar algoritmos basados en distribuciones. En este tipo de algoritmos, los grupos se definen como los datos que pertenecen a la misma distribución de probabilidad. Uno de los métodos más utilizados es conocido como **Clustering basado en distribuciones Gaussianas**, en el que los datos se clasifican en torno a diferentes distribuciones Gaussianas, inicializadas aleatoriamente, cuyos parámetros,  $\mu$  y  $\sigma$  se optimizan iterativamente hasta alcanzar el conjunto de datos que mejor se ajusta a esta distribución. Pero como la mayoría de algoritmos de *Clustering*, es necesario especificar el número de grupos a clasificar,  $k$ .

Con respecto a la GUI, una futura mejora sería su implementación en otros lenguajes de programación, tales como R, un lenguaje de programación con enfoque al análisis estadístico, utilizado especialmente en investigaciones biomédicas, y Phyton, los cuales contienen diversas librerías de algoritmos de clasificación, pudiendo aumentar la cantidad de algoritmos comparados en este proyecto. La curva de aprendizaje de estos lenguajes, ha impedido la implementación de una GUI de clasificación en ellos.

Debido a las limitaciones en cuanto a cantidad y variabilidad de señales aECG de las que disponemos en bases de datos, no nos ha sido posible investigar sobre

### 8.3 Líneas de investigación futuras

---

diversos problemas relacionados con la extracción de los complejos fQRS. Por ejemplo, no hemos encontrado una base de datos fiable que contenga los complejos fetales QRS marcados en el caso de un embarazo múltiple, aquel que desarrolla más de un feto, para una posible detección de los mismos. Tampoco se dispone de bases de datos de entrenamiento para extraer estos complejos. Una posible solución a estos problemas puede ser la contratación de un experto en cardiología, el cuál nos ayude a marcar los complejos fQRS correctamente, para así verificar que realmente nuestros algoritmos los han extraído correctamente, y proceder a la creación de una base de datos fiable, utilizable por la comunidad científica. Concretamente, las posibles líneas de investigación futuras serán:

- Ante el posible caso de un **embarazo múltiple**, podríamos aplicar técnicas de *Clustering* para clasificar los diferentes FHR de cada feto. Es decir, si utilizamos el método por detección por umbrales, este nos extraerá los complejos QRS fetales de ambos fetos, como complejos QRS fetales del mismo grupo, por lo tanto realizará una clasificación errónea. Una posible solución sería la implementación de las técnicas de clustering aquí presentadas, así como otras técnicas, para la resolución de este problema.
- Tal y como hemos comentado, la falta de base de datos de entrenamiento, no nos ha permitido realizar una clasificación de nuestras señales mediante Aprendizaje supervisado, razón por la cuál, tuvimos que decantarnos por la utilización de técnicas de aprendizaje no supervisado. Por lo que, en el caso futuro que estas bases de datos existieran, o se crearan, se podrían aplicar técnicas de este tipo para la extracción de complejos QRS fetales. Por lo que, podríamos utilizar **redes neuronales** o **SVM** (*Support Vector Machines*) para extraer estas características. Básicamente, su funcionamiento es el siguiente: dado un conjunto de muestras de entrenamiento, podemos clasificar nuestros datos y entrenar una SVM para construir un modelo que prediga la clase de una nueva muestra. A medida que se aumentan los entrenamientos, mayor precisión adquiriremos en nuestra clasificación. En conclusión, la utilización de redes neuronales es una posible herramienta para aumentar la eficiencia obtenida en este proyecto.
- Además, la aplicación de métodos de **inteligencia artificial** (IA) es importante en la detección y clasificación de diferentes tipos de arritmias y diversas patologías, las cuales se detectan principalmente a partir del complejo QRS. Una vez se llegase a una clasificación con unos parámetros de bondad aceptables, superiores al 90 %, se podría implementar un sistema de detección ante

## **8. CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS**

---

posibles patologías cardíacas. Es decir, podríamos calcular los valores entre los distintos intervalos y segmentos (S-T, R-R, QRS, etc.) de nuestra señal, y en el caso de la obtención de valores anormales en estos intervalos, se podría avisar al paciente de posibles patologías. Actualmente, en el campo del fECG, no es posible la implementación de este sistema, debido a que es necesario obtener una alta precisión para evitar el aviso de avisos innecesarios al paciente (por la presencia de FP), así como la no notificación de patologías (por culpa de la presencia de FN en la señal).

# Apéndice A

## Implementación hardware

El siguiente capítulo presenta la arquitectura hardware necesaria para la obtención de la FHR a partir de la señal aECG medida en el abdomen materno. Se han estudiado diferentes soluciones para la etapa de preprocesamiento, contemplando las ventajas e inconvenientes de cada una de ellas, decantandonos finalmente por la arquitectura que implementa la solución WT. Esta arquitectura se basa en el trabajo realizado por [24], utilizando una FPGA, en concreto una de la familia Spartan-6 de Xilinx, el modelo XC6SLX45-3FGG484. Añadiremos al procesador principal, la implementación de los algoritmos de clasificación *k-means* y *k-medoids* basados en la distancia Euclídea al cuadrado. Además se presentará un método de adquisición de la señal analógica a partir de sensores colocados en el abdomen materno.

### A.1. Arquitectura hardware

La solución hardware desarrollada en [24] implementa el bloque de preprocesamiento descrito por [6], que hace uso de un filtrado no lineal basado en la WT para eliminar gran parte de los ruidos de baja y alta frecuencia de la señal aECG.

La solución hardware propuesta se implementará en una FPGA utilizando el lenguaje de programación VHDL, programada gracias a la utilización del software ISE 13.4 de Xilinx. La exportación del preprocesado y los algoritmos de clasificación se realizará gracias a la herramienta HDL *Coder* integrada en MATLAB, la cuál exportará nuestras funciones a VHDL. Para ello es necesario optimizar nuestro código, así como implementar nuestros algoritmos en punto fijo, tarea que se ha realizado previamente. De cara a una futura implementación final, se programaría nuestro código completamente en VHDL, evitando la utilización de traductores de código, para así aumentar la eficiencia del mismo.

## A. IMPLEMENTACIÓN HARDWARE

---

Este diseño incluye un microprocesador, el cuál actúa como controlador del flujo de la señal, mientras que los coprocesadores *wavelet* eliminan el *wandering* y ruido de la señal. El diagrama de bloques de esta arquitectura se presenta en la Figura A.1.

Esta arquitectura ha sido diseñada para aplicaciones en tiempo real basadas en la adquisición y procesado de datos ECG agrupados en ventanas de 5000 muestras.

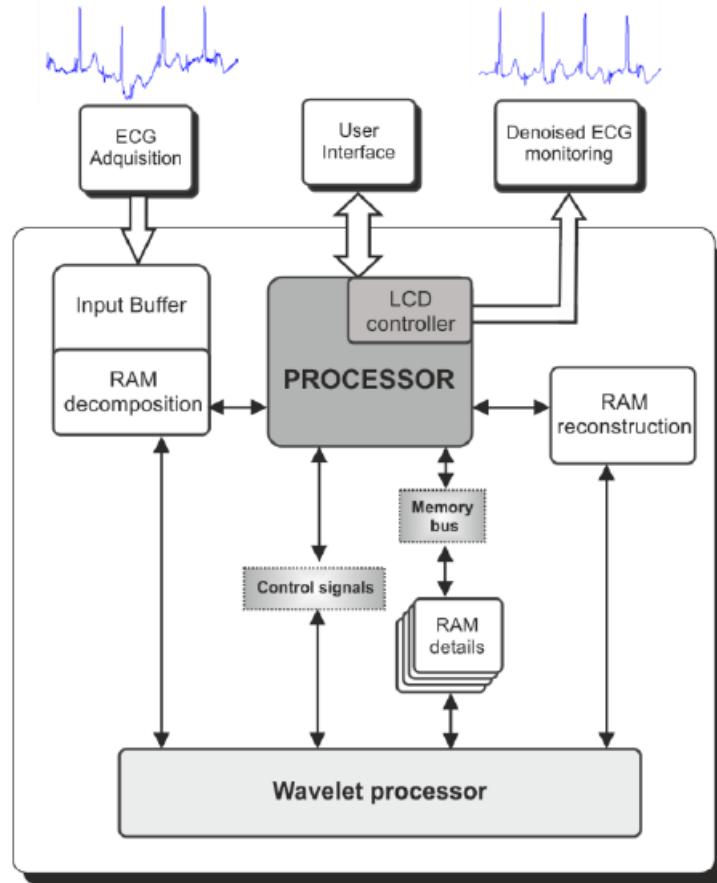


Figura A.1: Diagrama de bloques de la arquitectura hardware inicial. [24]

La arquitectura propuesta se compone de los siguientes bloques:

**Buffer de entrada** El bloque de la memoria RAM (*Random Access Memory*) de descomposición se utiliza para la adquisición y procesado de cada ventana de datos aECG. Este bloque hace uso de dos memorias, denominadas *buffer* de entrada y RAM de descomposición. El buffer de entrada almacena los datos adquiridos como una pila. Esta ventana de datos se copia en la memoria RAM de descomposición. Entonces, la memoria RAM es refrescada con los coeficientes de aproximación obtenidos de la descomposición de la *wavelet*. Esta arquitectura evita los conflictos de acceso a memoria

durante el tiempo en el que la ventana de datos actuales esta siendo procesada y se están adquiriendo las muestras de la siguiente ventana.

**Procesador *wavelet*** Este módulo esta constituido por un *core IP* (*Intellectual Property*), el cuál realiza las operaciones de descomposición y reconstrucción de la WT. Este *core IP* está formado por tres unidades de control, dos bloques de procesamiento (bloques de descomposición y reconstrucción) y una memoria FIFO (*First In-First Out*). Las dos unidades de control primeras permiten configurar los bloques de descomposición y reconstrucción de la WT, mientras que la tercera permite la sincronización de ambos bloques de procesos, controlando además el acceso a memoria. La arquitectura de los bloques de descomposición se basa en una estructura de bucle, incluyendo dos bancos de filtros FIR utilizados para calcular los  $L$  niveles de descomposición de la *wavelet*. Cada filtro FIR calculará una de las dos secuencias de la WT. Una estructura similar se utiliza para el bloque de reconstrucción. El procesador *wavelet* del *core IP* involucra el uso de tres bloques de memoria. La primera memoria es la memoria RAM del bloque de descomposición de entrada. Las otras dos memorias son:

**RAMs** Cada una de estas memorias se utiliza para almacenar los coeficientes de cada nivel de descomposición. Como es necesario  $L$  niveles de descomposición *wavelet*, necesitaremos por tanto,  $L$  RAMs. Estas memorias serán direccionadas durante el proceso de reconstrucción para acceder a las distintas secuencias de niveles  $L$ .

**Memoria de reconstrucción** La señal reconstruida en cada nivel se almacenará en una memoria. Cuando el proceso de reconstrucción ha finalizado, esta memoria almacenará la señal preprocesada.

**Bloque de procesos** El procesador implementará las operaciones de control necesarias, seleccionando los parámetros adecuados involucrados en la eliminación de BW y del ruido, aplicando el procesado propuesto para la supresión de BW y ruido, para finalmente inspeccionar la señal procesada. Este bloque integra un sistema compuesto por un microcontrolador, una memoria ROM (*Read-Only Memory*), una memoria RAM, un controlador LCD para la representación gráfica en tiempo real de la señal, puertos de entrada y salida, la lógica necesaria para actuar como interfaz con las memorias externas que almacenan los datos del procesado *wavelet* y la lógica para las conexiones entre estos módulos.

## A. IMPLEMENTACIÓN HARDWARE

---

Los algoritmos de clasificación mediante *Clustering* se implementarán en el microprocesador principal, sustituyendo al algoritmo de clasificación mediante umbrales. En la implementación de estos dos algoritmos, *k-means* y *k-medoids* únicamente será necesario reservar  $n^2$  posiciones de memoria RAM, siendo  $n$  el número de datos máx-mín. Estas posiciones de memoria se refrescarán automáticamente en cada iteración, sobrescribiendo los datos sobre los datos anteriores, sin necesidad de reservar más posiciones de memoria. Mientras, que si atendemos al número de iteraciones medias hasta conseguir la conversión de nuestro algoritmo, este ha sido de 5, por lo que necesitaremos de media 5 iteraciones hasta alcanzar la convergencia de nuestro algoritmo. A su vez, es necesario realizar varias repeticiones del algoritmo para encontrar el mínimo absoluto de distancias, por lo que si repetimos nuestro algoritmo  $z$  repeticiones, únicamente necesitaremos  $n^2$  posiciones de memoria RAM, más  $z$  posiciones para almacenar el valor total de la suma de distancias entre clusteres, siendo el total de posiciones ocupadas  $total_{posRAM} = n^2 + z$ .

### A.2. Hardware de la parte de adquisición del ECG

En esta sección, se discutirán los problemas surgidos alrededor del diseño y fabricación del sistema hardware para el acondicionamiento de la señal aECG. En una primera implementación del prototipo final, se considerará la implementación sobre una tarjeta FPA, un dispositivo analógico programable, el cuál contiene bloques de circuitos analógicos configurables interconectados entre sí. Entre sus ventajas con respecto a el *layout* de circuitos analógicos nos encontramos con:

- Prototipado muy rápido, mediante entornos gráficos sencillos. Ofrece una mayor rapidez en el diseño y verificación del circuito.
- Flexibilidad para reconfigurar el diseño.
- Reducción del número de integrados necesarios para realizar la aplicación.
- Menos problemas de estabilidad y ruidos.

Pero a pesar de estas ventajas, este tipo de solución posee una desventaja principal contra el diseño del circuito de adquisición en una PCB, su coste. Por lo que para la fabricación de un producto final, es necesario optimizar al máximo el coste de fabricación del mismo, por lo que, el diseño y fabricación de una PCB para la adquisición del sistema será el sistema elegido para la adquisición de nuestra señal aECG. En el presente capítulo, se detallará el diseño del circuito a realizar para este objetivo.

### A.2.1. Arquitectura de un sólo canal

La Figura A.2 muestra el proceso general de adquisición de la señal ECG de un sujeto. Las fluctuaciones (en mV) de la diferencia de potencial (DP) entre las derivaciones diferenciales del ECG en la superficie corporal son amplificadas con un amplificador de instrumentación aislado (ver Figura A.3). Notar que, generalmente son requeridas tres derivaciones para obtener una señal diferencial del sujeto, además del requerimiento de un electrodo que actuará como tierra del circuito. La diferencia de voltajes entre los otros electrodos (entrada A y B) sirven como señal de entrada que es amplificada a través de los amplificadores operacionales (AO) U1A y U1B. Estas señales son amplificadas diferencialmente y pasadas a través de un filtro paso alta (HPF, *High Pass Filter*).

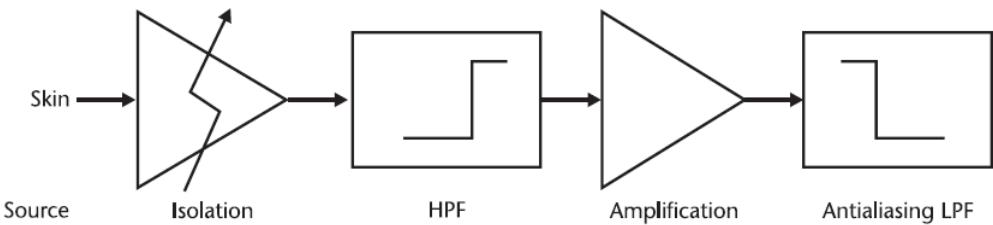


Figura A.2: Diagrama de adquisición de la señal ECG simplificado. [18]

## A. IMPLEMENTACIÓN HARDWARE

---

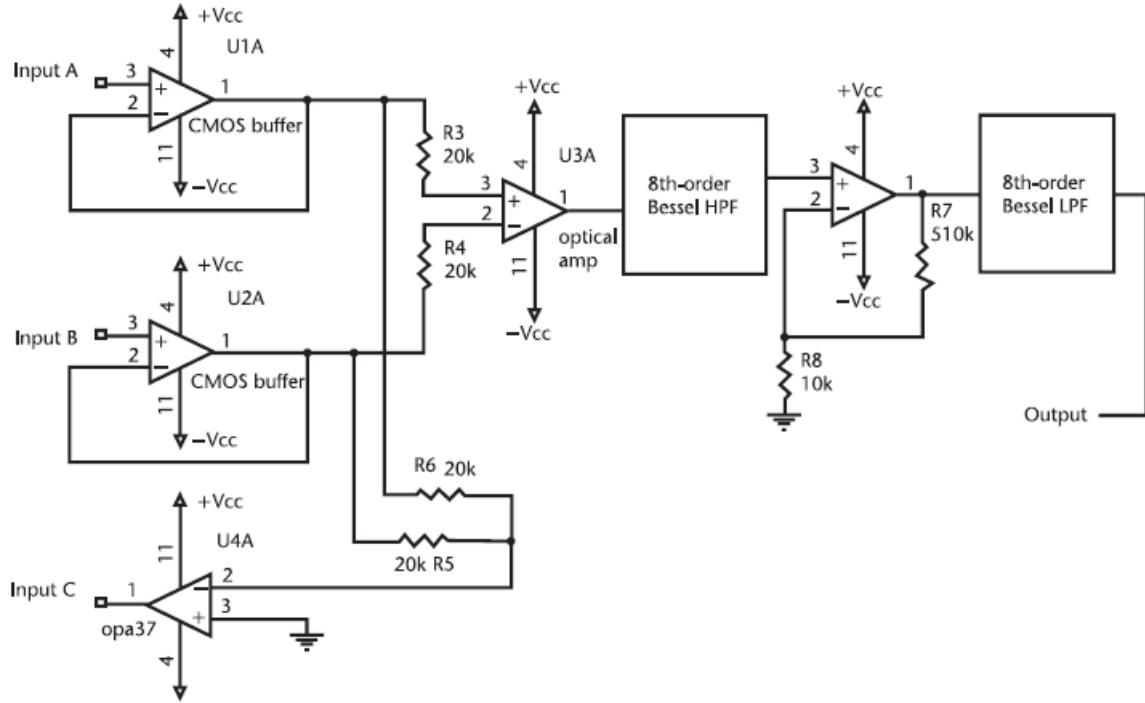


Figura A.3: Circuito para adquirir la señal ECG a partir de tres derivaciones, realizada con tres electrodos. [18]

### A.2.2. Protección y aislamiento

Para cualquier circuito que utilice una fuente de alimentación y tenga contacto directo con el cuerpo humano, este debe de ser segmentado en secciones aisladas y no-aisladas. Estas secciones deben de estar separadas aproximadamente 10 mm la una de la otra (dependiendo de la constante dieléctrica de la placa). Incluso aún cuando hayan pequeñas descargas de corriente (menores de 100  $\mu$ A) en el sujeto, estas pueden inducir una fibrilación ventricular letal en humanos.

La potencia de la parte no-aislada de la placa es transferida directamente a la parte aislada utilizando convertidores DC-DC. El uso de un transformador magnético inducido para transferir la potencia, realiza una transferencia de fotones en lugar de electrones (corriente) hacia la parte aislada. Por lo tanto, no hay corriente que llegue al sujeto desde la red eléctrica.

De manera similar, la información es transmitida de vuelta desde la parte aislada (paciente) del circuito hacia la parte no-aislada a través de luz en los aisladores ópticos. Estos aisladores-ópticos convierten los electrones en fotones y de nuevo en electrones, transmitiendo así únicamente luz (y no corriente) a través del *gap* de aislamiento. Los aisladores-ópticos están colocados con una distancia mínima de 10-mm entre la

## A.2 Hardware de la parte de adquisición del ECG

---

parte aislada y la no aislada, y están alimentados en ambos lados, ya sea por la salida aislada de los convertidores DC-DC o conectados directamente a la red eléctrica, respectivamente.

Después de la etapa de aislamiento óptico, la señal es filtrada por un HPF, una segunda etapa de amplificación, y después un filtro anti-aliasing paso baja (*Low Pass Filter*, LPF). Finalmente, la señal es muestreada por un conversor analógico digital (AD).

Tener en cuenta que las resistencias con altos valores deben de ser posicionadas entre cada entrada y tierra para ofrecer al circuito protección ante un posible voltaje de desfibrilación. Además, es necesario colocar una resistencia que limite la corriente a la salida en caso de que fallen los AO. Señalar también que, el aislamiento óptico en una etapa de amplificación inicial puede introducir ruido significante. Por lo tanto, es preferible aislar directamente después de digitalizar la señal.

### A.2.3. Reducción de ruido en el modo-común primario: circuito de tierra activo

Cabe destacar la presencia de ruido electromagnético en la red eléctrica utilizada para alimentar al sistema o en el propio sistema de alimentación. El espectro de algunos ECGs puede expandirse desde corriente en DC hasta aproximadamente 1 kHz, y además, utilizan un filtro Notch o elimina banda de entre 50 Hz y 60 Hz (debido a que el ruido eléctrico de la red eléctrica puede variar en ese rango) para eliminar el ruido, aunque esto además implique la eliminación de información de la señal. La utilización de un circuito de tierra activo es la opción preferida para eliminar el ruido en modo común.

El circuito activo de tierra, tal y como se muestra en la Figura A.4, trabaja utilizando la media del Modo Común de los voltajes de los dos terminales de entrada de la etapa de preamplificación. Esta etapa amplifica e invierte la señal, para después utilizarla como tierra o voltaje de referencia del circuito. Este circuito no elimina la señal diferencial pero sí que elimina el ruido en modo-común. Esto es, eliminará las partes de señal que se presentan simultáneamente en ambos electrodos.

## A. IMPLEMENTACIÓN HARDWARE

---

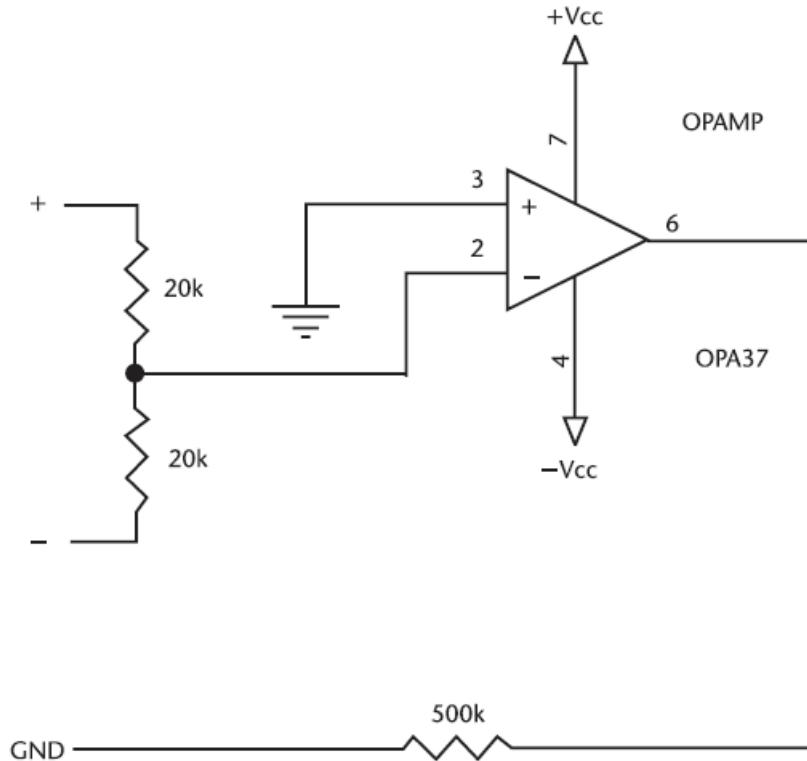


Figura A.4: Circuito de tierra activa utilizado para la reducción de ruido en modo común. [18]

### A.2.4. Aumento de la impedancia de entrada: etapa CMOS

Es sabido que en el diseño de instrumentación biomédica es necesario una alta impedancia de entrada, debido a que la señal de interés (señales fisió-eléctricas) son extremadamente débiles (del orden de cientos de mV) y, en consecuencia, no pueden suministrar corriente adecuadamente. Una impedancia de entrada extremadamente alta así como la correspondiente amplificación de potencia de la señal son propiedades propias del circuito seguidor de tensión CMOS. Un preamplificador operacional CMOS hace la función de etapa ideal de desacoplamiento entre la señal electrofisiológica y la subsiguiente etapa de procesado de señal analógica.

### A.2.5. Preamplificación y aislamiento

A pesar de que es preferible colocar la etapa de aislamiento después de los amplificadores, esto significa tener que escribir *drivers* propios para los controladores AD. Por lo que, si la señal aECG no presenta anomalías importantes, podremos proporcionar aislamiento óptico en la etapa de preamplificación. Esto asegura que una

## A.2 Hardware de la parte de adquisición del ECG

---

sobretensión eléctrica sea incapaz de electrocutar al sujeto, y a la inversa, que un sobrevoltaje en los terminales de entrada no pueda dañar al circuito de instrumentación más allá del preamplificador. La mayor amplitud de corriente origina un acoplamiento capacitivo desde la fuente de alimentación hasta el chasis de instrumentación de tierra. Sin embargo, si el chasis que alberga la parte hardware del sistema de adquisición del ECG está adecuadamente conectado a tierra, la resistencia mínima desde el chasis hasta tierra llevará la mayor parte de corriente hacia tierra, a través de este camino. El amplificador de aislamiento óptico provee de una alta interrupción dieléctrica , o equivalentemente una pequeña capacidad, protegiendo al sujeto de actuar como camino para las corrientes de retorno hacia la tierra.

Los voltajes fisiológicos producidos por el corazón van desde los  $\mu\text{V}$  hasta varios mV, y el rango dinámico del preamplificador es usualmente  $\pm 12\text{V DC}$ . La contabilización de los diferentes potenciales de las células en los electrodos pueden producir un voltaje diferencial tan alto como 100 mV. Por lo que estableceremos la ganancia en función de este valor, logrando no saturar nuestra señal. Sin embargo, se deben de tomar precauciones a medida que aumenta la diferencia de potencial, como por ejemplo reducir la ganancia.

### A.2.6. Filtrado paso alto

La señal de salida del amplificador de instrumentación es la entrada del filtro paso alta con una frecuencia de corte de 0.1 Hz. El filtro pasa alto sirve para eliminar el *offset* de DC debido a las diferencias de potencial de las células en los electrodos así como otros ruidos de baja frecuencia presentes en la señal. La elección de un filtro de Bessel está motivada por el hecho de que este posee una respuesta de fase óptima. Esto es, este tipo de filtro posee un retardo de grupo casi constante, y una distorsión de fase despreciable. Esta característica óptima en la respuesta de fase se paga con una disminución en la pendiente de caída en la banda de transición comparándose con otras funciones de transferencia.

### A.2.7. Amplificación secundaria

Después de filtrar la señal mediante el filtro HPF, la señal es amplificada de nuevo. Esta es la última etapa de amplificación en el acondicionamiento de la señal para proceder a su utilización. Esta segunda etapa, incrementa la SNR de la señal y establece el voltaje de la misma en un rango apropiado para el muestreo con un convertidor A/D con un rango dinámico de  $\pm 10\text{V}$ . El circuito de amplificación (FiguraA.5) es

## A. IMPLEMENTACIÓN HARDWARE

---

un amplificador operacional (AO) en configuración no inversora, con una ganancia de:  $1 + R_f/R_{in}$ . La señal de entrada de esta etapa, al contrario que en la etapa de preamplificado, no posee *offset* debido al filtrado HPF realizado en la etapa anterior. Por lo que, esta etapa de amplificación puede amplificar la señal sin llegar a saturarla.

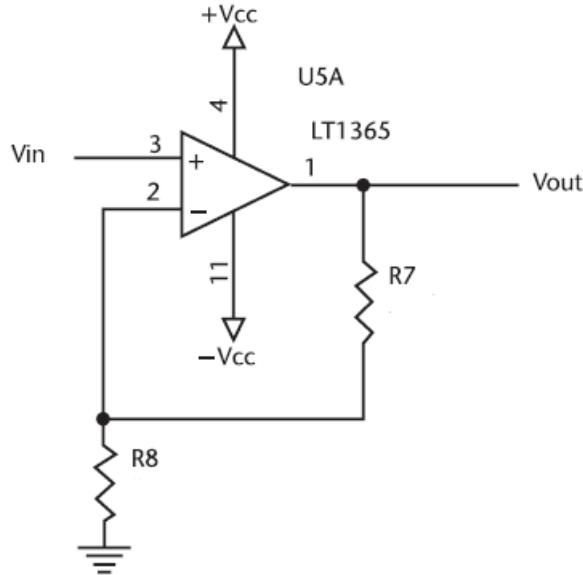


Figura A.5: Amplificador operacional en configuración no inversora con retroalimentación negativa

### A.2.8. Filtrado paso baja y sobremuestreo

Como se ha indicado anteriormente, el espectro del ECG tendrá un BW que irá, desde la frecuencia DC, es decir, desde 0 Hz hasta aproximadamente 150 Hz, por lo tanto utilizando el **Teorema de muestreo de Nyquist**, observamos cómo con un filtro LPF ideal con una banda pasante de 0 a 150 Hz, necesitaríamos un muestreo de la señal de 300 Hz para evitar el aliasing.

Sabemos que la implementación real de un circuito nunca es exactamente “ideal”, se debe de cumplir una relación entre el tipo de filtro, la frecuencia de corte del filtro, y la frecuencia de muestreo del AD que produzca una pequeña cantidad aceptable de aliasing.

El *sobremuestreo* es una técnica a menudo empleada en sistemas que utilizan un filtro *antialiasing* con una baja pendiente de caída. Se demuestra que el muestreo de una señal analógica produce copias especulares del espectro analógico en los múltiplos de las muestras de frecuencia,  $f_s$ , en el dominio discreto del tiempo. Considerar el

## A.2 Hardware de la parte de adquisición del ECG

---

ejemplo en el que una señal es filtrada en el dominio analógico con un LPF no-ideal con frecuencia de corte de 150 Hz, muestreado a 300 Hz. El resultado es una señal que padece de aliasing, la cuál se muestra en las regiones espectrales superpuestas de la Figura A.6. Una posible solución podría ser construir un filtro analógico de mayor orden para reducir la banda de transición, lo cuál puede resultar costoso y consumir más recursos para mitigar los efectos del aliasing. Alternativamente, se podría muestrear la señal a una frecuencia de muestreo mayor, separando entre sí las copias espectrales, tal y como se muestra en la Figura A.7. Se podría utilizar como filtro LPF, para filtrar la señal digitalizada, un filtro digital de mayor orden, el cuál es más barato y sencillo de implementar. Esto conseguiría el mismo efecto que la introducción de un filtro *antialiasing* de mayor orden, sin la complejidad hardware anterior.

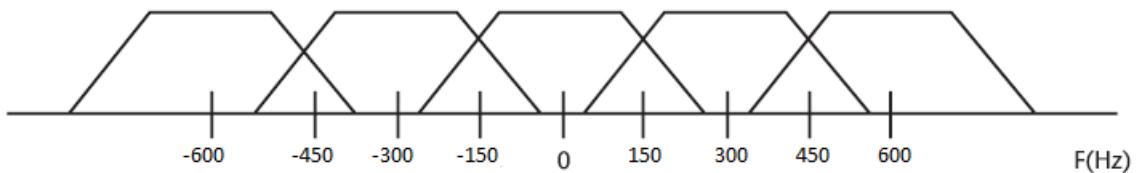


Figura A.6: Señal limitada en banda a aproximadamente 225 Hz (debido a la frecuencia de corte de 1 kHz del filtro *antialiasing*) muestreada a 300 kHz, la cuál contiene copias espectrales en los múltiplos de 300 Hz y sufre de *aliasing* (superposición).

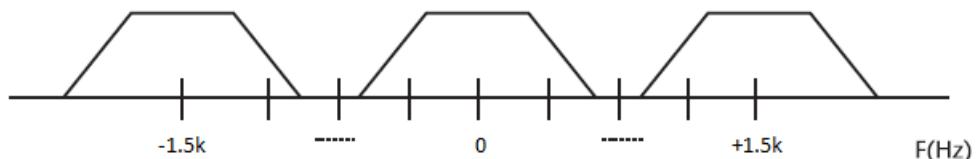


Figura A.7: La misma señal, limitada en banda a 225 Hz pero muestreada a 1.5 kHz (5 veces el muestreo anterior) posee copias espectrales en múltiplos de 1.5 kHz y no sufre de aliasing.

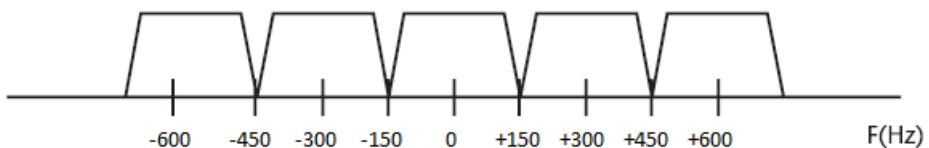


Figura A.8: Después de aplicar un filtro digital LPF de orden elevado con una decimación de un factor de 5, el espectro está espaciado por múltiplos de 300 Hz pero sin aliasing .

## A. IMPLEMENTACIÓN HARDWARE

---

Dicha técnica, conocida como *sobremuestreo*, es utilizada a menudo en sistemas de muestreo de datos para minimizar la complejidad y el coste de los circuitos analógicos, así como para aprovechar la potencia proveniente del procesado de la señal digital. En el caso de este sistema, utilizaremos un filtro *antialiasing* de orden elevado. Sin embargo, a pesar de que el filtro está optimizado para una distorsión de fase mínima, su caída es similar a la que ofrece un filtro de menor orden. Una aproximación razonable para este tipo de señal es emplear un sobremuestreo cinco veces mayor, por lo que la señal es muestreada por un factor de cinco, para dar una tasa de muestreo efectiva de 300 Hz. Un LPF con respuesta finita al impulso (*Finite Impulse Response*, FIR) simétrico digital precede al diezmado para evitar el aliasing. El uso inicial del sobremuestreo además minimiza el aliasing, y subsecuentemente inframuestrea (después del filtrado LPF), proporcionando el requisito mínimo sin pérdidas permisible para el almacenamiento de datos sin tener que recurrir a la compresión. La Figura A.8 muestra el contenido espectral de la señal después de haber sido sobremuestreada (cinco veces mayor), filtrada paso baja, y diezmada por un factor de cinco. Las copias espectrales no se superponen, indicando esto que no ha ocurrido aliasing. Sin embargo, el espectro está relativamente muy poco espaciado, indicando esto que la señal no está sobremuestreada, y que la memoria utilizada es por lo tanto mínima.

### A.2.9. Diseño de la PCB

Para el diseño de nuestra PCB seguimos las siguientes pautas:

1. Separar la parte analógica de la digital, y a su vez la baja de la alta frecuencia.
2. Utilizar un plano de masa siempre que sea posible, para reducir las interferencias electromagnéticas.
3. Las pistas de masa y alimentación, deberán de ser más anchas que el resto de pistas del circuito, para que tengamos la máxima capacidad de corriente, y a altas frecuencias nuestra impedancia sea lo menor posible
4. Las pistas de alta frecuencia, deberán colocarse por los extremos de la PCB, así como evitar la formación de ángulos de 90º en su ruteado.
5. Trazar las pistas tan cortas como sea posible.
6. Evitar la formación de bucles.

## A.2 Hardware de la parte de adquisición del ECG

---

Una vez seguidas todas estas indicaciones, se diseño una versión inicial de la PCB que se encargará de adquirir la señal aECG del abdomen materno. Este diseño ha seguido como circuito guía el circuito presentado en la Figura A.3. Nuestra PCB ha sido diseñada en dos capas, la capa *top* o superior, y la capa *bottom* o inferior, montándose los componentes de tecnología SMD (*Surface Mount Device*) en la capa superior. Debido a la gran cantidad de conexiones entre las pistas, ha sido necesario implementar vías en nuestro diseño, para poder trazar pistas en la capa *bottom*. Respecto al orden de ruteo de nuestra placa, primeramente se han ruteado las pistas de alta frecuencia (debido a que estas son mas susceptibles al ruido), para en segundo lugar hacer las conexiones a tierra y alimentación. Respecto a nuestros componentes, se eligió como LPF el integrado MAX7405 de Maxim, el cuál implementa un Filtro Bessel de orden elevado en el que podremos cambiar la frecuencia de corte del filtro introduciendo una señal de reloj de la frecuencia deseada. Este integrado cumplirá con nuestro rango de frecuencias. En las siguientes Figuras se muestran la PCB diseñada, tanto la capa *top* como la *bottom*.

## A. IMPLEMENTACIÓN HARDWARE

---

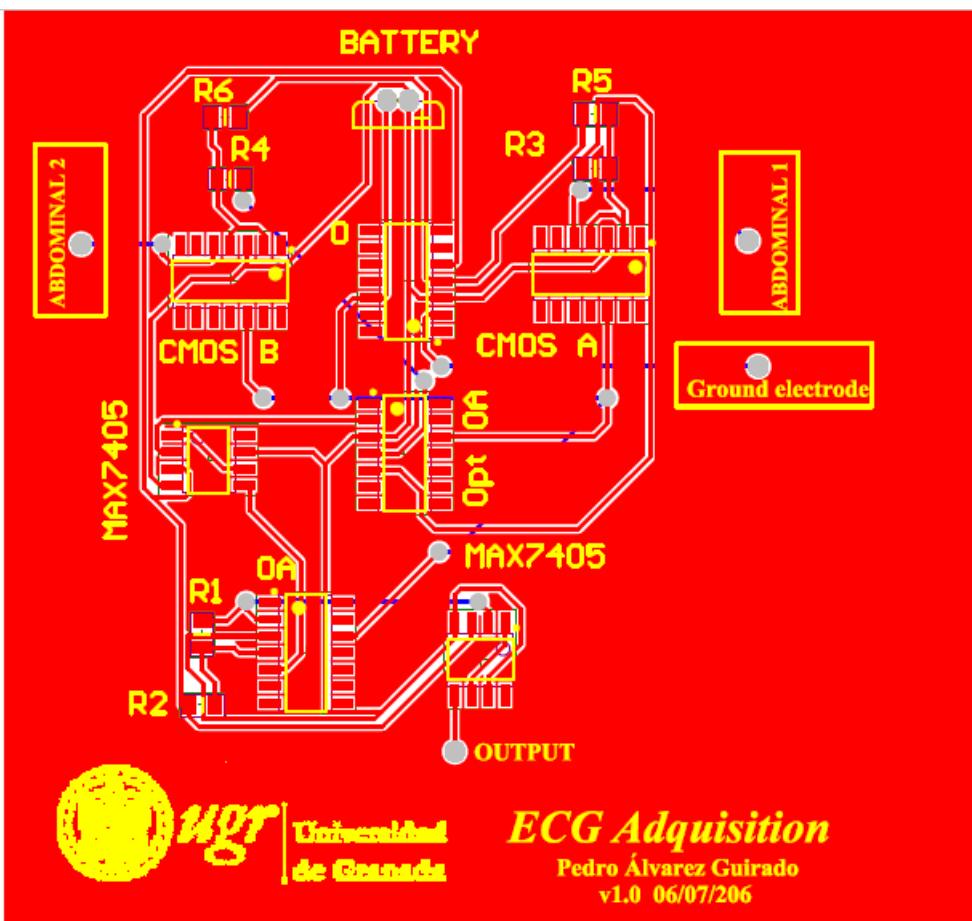


Figura A.9: PCB diseñada para la adquisición de la señal aECG. Capa *top* con plano de masa.

## A.2 Hardware de la parte de adquisición del ECG

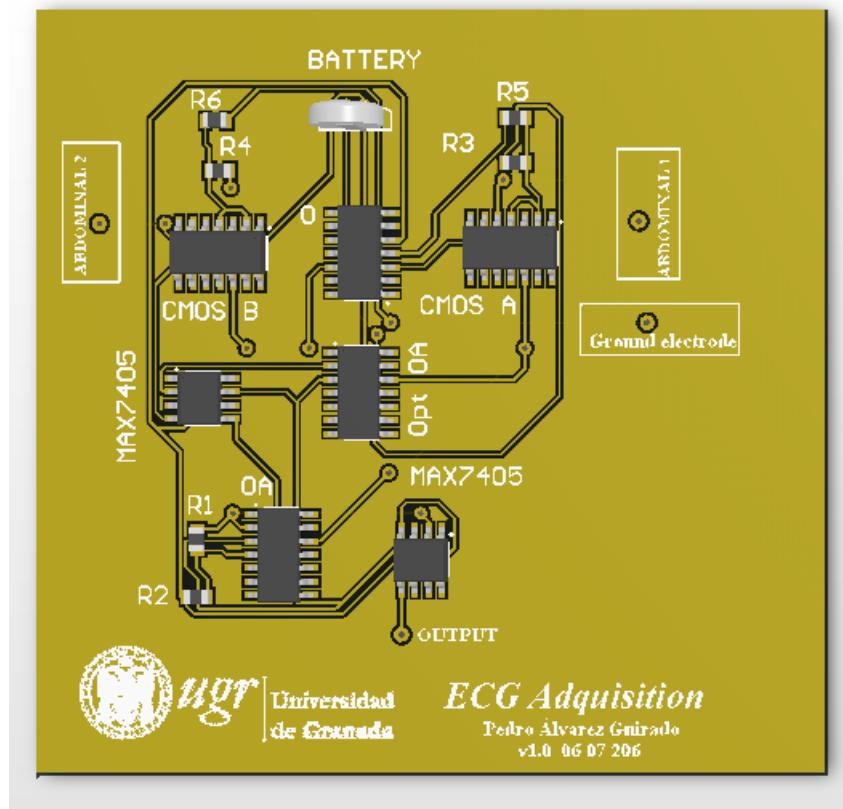


Figura A.10: PCB diseñada para la adquisición de la señal aECG. Capa *top* con plano de masa vista en 3D.

## A. IMPLEMENTACIÓN HARDWARE

---

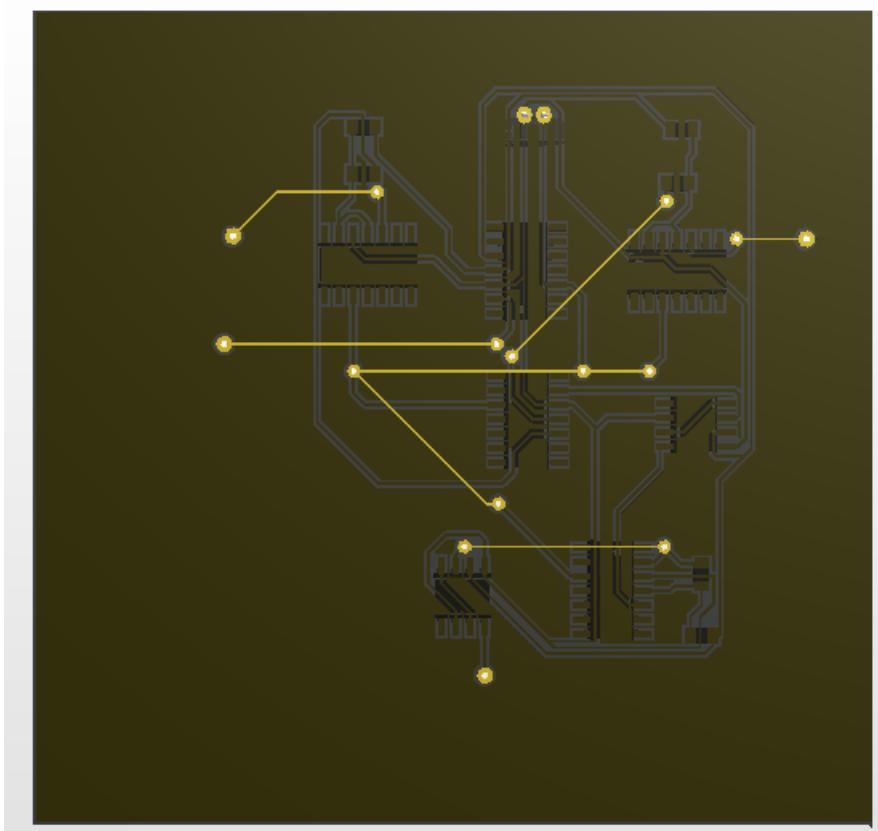


Figura A.11: PCB diseñada para la adquisición de la señal aECG. Capa *bottom* vista en 3D

# Referencias

- [1] P. BERKHIN. *Survey of clustering data mining techniques*. 2001.
- [2] J.C BEZDEC. Pattern recognition with fuzzy objective function algorithms. *Plenum Press*, 1981.
- [3] L.A BURATTINI. Noninvasive fetal electrocardiography: An overview of the signal electrophysiological meaning recording procedures, and processing techniques. *Annals of Noninvasive Electrocardiology*, vol. 20, pp. 303-313, 2015.
- [4] ARTHUR DAVID AND SERGI VASSILVITSKII. K-means++: The advantages of careful seeding. *SODA 07: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.
- [5] D. DONOHO. Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc*, Vol. 90, pp: 1200-1224, 1995.
- [6] D.P MORALES E. CASTILO AND A. GARCÍA. Efficient wavelet-based ecg processing for single-lead fhr extraction. *Digital Signal Processing: A Review Journal*, Vol. 23, pp. 1897-1909, 2013.
- [7] M. TOROK F. KOVACS AND I. HABERMAJER. A rule-based phonocardiographic method for long-term fetal heart rate monitoring. *IEE Trans. Biomed. Eng.*, vol. 47, pp. 124-130, 2000.
- [8] CRISTIAN G. G. Sistema digital de extracción del ritmo cardíaco fetal mediante filtrado adaptativo. *Bachelor's Thesis, Universidad de Granada*, 2015.
- [9] PATRICK E. MCSHARRY GARI D. CLIFFORD, FRANCISCO AZUAJE. *Advanced Methods and Tools for ECG Data Analysis*. Artech House, 2006.
- [10] E.G.M GOLBACH AND STINSTRA J.G. Reference values for fetal mcg/ecg recording in uncomplicated pregnancies. *12th International Conference on Biomagnetism, Espoo, Finland*, pp. 13-17, 2000.

## REFERENCIAS

---

- [11] AMARAL LAN GOLDBERGER AL. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation* 101(23):e215-e220, 2000.
- [12] ITO H. AND L. GLASS. Spiral breakup in a new model of discrete excitable media. *Phys. Rev. Lett.*, Vol. 66, No. 5, pp. 671-674, 1991.
- [13] V. KARINIEMI AND K. HUKKINEN. Quantification of fetal heart rate variability by magnetocardiography and direct electrocardiography. *Am J Obstet Gynecol*, vol. 128, pp. 526-30, 1977.
- [14] DAVIES D. L AND D. W. BOULDIN. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 1, pp 224-227, 1979.
- [15] STEPHEN LEE AND JOHN KRUSE. *Biopotential electrode sensors in ecg/eeg/emg systems*. Analog Devices, Inc., 2008.
- [16] PAVAN M AND PELILLO M. Dominant sets and pairwise clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2007.
- [17] J. MALMIVUO AND R. PLONSEY. *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*. Oxford University Press, 1995.
- [18] R. G. MARK. Quantitative physiology: Organ transport systems. *MIT Open Courseware*, 2004.
- [19] ELIZABETH A. MARTIN. *Concise Medical Dictionary, 8th edition*. Oxford University Press, 2010.
- [20] REZA SAMENI AND GARI D. CLIFFORD. A review of fetal ecg signal processing; issues and promising directions. *Open Pacing Electrophysiology Theraphy Journal*, Vol 3, pp: 4-20, 2010.
- [21] SHAI SHALEV-SHWARTZ AND SHAI BEN-DAVID. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
- [22] J. STINSTRA. *Reliability of the fetal magnetocardiogram*. PhD thesis, University of Twente, 2001.

## REFERENCIAS

---

- [23] ARVIND K. TIWARI AND BRIJ N. SINGH. Optimal selection of wavelet basis function aplied to ecg signal denoising. *Digit. Signal Process.* Vol. 16, pp. 275-287, 2006.
- [24] D.P. MORALES A. GARCÍA L. PARRILLA D. GONZALÍA F.S. MOLINA V.U RUIZ, E. CASTILLO AND J. FLORIDO. Ecg processing on reconfigurable hardware for efficient artifact reduction. *Experimental and Clinical Cardiology*, Vol. 20, pp 3023-3028, 2014.
- [25] J. WLADIMIROFF AND G. PILU. *Ultrasound and the Fetal Heart*. NY: Parthenon Publishing, 1996.
- [26] KUMARV. WU X. AND J.R QUINLAN. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 2008.
- [27] XILINX. *Syntheis and Simulation Design Guide, v12.3 edition*. 2010.
- [28] RUI XU AND DONALD WUSCH. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, Vol. 16, pp 645-676, 2005.