



**UNIVERSIDAD AUTÓNOMA DE BAJA
CALIFORNIA**

FACULTAD DE INGENIERÍA

**MAESTRÍA Y DOCTORADO EN CIENCIAS E
INGENIERÍA**

**“Atenuación de la deriva de la línea basal de una señal de
ECG de un paciente en movimiento capturada por un
sistema de alta resolución”**

T E S I S

**que presenta para obtener el grado de
MAESTRO EN CIENCIAS**

Eladio Altamira Colado

**DIRECTOR DE TESIS:
Dr. Roberto López Avitia**

**CODIRECTOR DE TESIS:
Dr. Miguel Enrique Bravo Zanoguera**



**Facultad de
Ingeniería**

Mexicali, B. C.

Junio de 2022

DEDICATORIA

A mis padres Magdalena Colado Hernández y Eladio Altamira Peláez, quienes me han impulsado y apoyado a siempre continuar con mi formación. A Vanessa, quien siempre me recordó de lo que soy capaz, me motivó, y creyó en mí. A mis amistades, quienes de manera desinteresada me acompañaron en buenos y complicados momentos. A mi compañero, el Dr. Daniel Cuevas González, quien me brindó la mano y su disposición durante este trabajo. A mis tutores, el Dr. Roberto López Avitia y el Dr. Miguel Enrique Bravo Zanoguera, quienes me guiaron y apoyaron durante todo el proceso.

AGRADECIMIENTOS

A mi familia, por apoyarme durante toda mi trayectoria académica, permitirme el privilegio de estudiar, y transmitirme sus valores.

A Vanessa, por insistirme en nunca rendirme, en nunca conformarme, en seguir mis sueños, y en creer en mí mismo.

A mi director, el Dr. Roberto López Avitia y a mi comité de tesis quienes me impulsaron y motivaron a mejorar cada aspecto de este trabajo.

Al Dr. Miguel Enrique Bravo Zanoguera, por ser un modelo a seguir, por su compromiso, enseñanzas, confianza, y paciencia. Por permitirme involucrarme en su grupo de investigación, ayudándome a descubrir todo lo que soy capaz de aportar, dando como producto esta tesis.

A mis docentes, por su disposición dentro y fuera del aula a siempre apoyarme con sus conocimientos y mis inquietudes.

Al Dr. Daniel Cuevas González, con quien compartí muchísimos momentos durante todo mi posgrado, y quien fue un ejemplo para mí, por dedicarme desinteresadamente muchas horas de su tiempo, por aconsejarme y ayudarme en momentos de estrés.

A la Facultad de Ingeniería de la Universidad Autónoma de Baja California por permitirme ingresar al programa de Maestría y Doctorado en Ciencias e Ingeniería (MYDCI).

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico brindado para permitirme realizar mis estudios de posgrado.

RESUMEN

El avance en el desarrollo de dispositivos médicos en décadas recientes ha permitido que, además de utilizar estos dispositivos para aplicaciones en el cuidado de la salud de pacientes con enfermedades, se investigue acerca de sus aplicaciones en otros campos, por ejemplo, en la rehabilitación, o en la mejora del rendimiento deportivo [1].

La tecnología en electrocardiógrafos ha permitido paulatinamente crear dispositivos con menor susceptibilidad a fuentes de ruido y esto a su vez ha permitido realizar mediciones sin la exigencia de permanecer en reposo durante el estudio, como en pacientes que sufren convulsiones, o cuando el estudio deba ser realizado mientras se realizan actividad física; en estos casos se induce ruido a la señal de interés [2].

Mediante la presente investigación se busca contribuir al desarrollo de dispositivos médicos que logren capturar una señal de ECG mientras el paciente se encuentre en movimiento, proporcionando una señal procesada por el mismo dispositivo. Dos elementos operando conjuntamente fueron aportados en esta metodología:

- Una alta resolución (24-bits) para capturar señales de ECG con mayor detalle mientras se utiliza bajas ganancias de amplificación, lo cual evita que el sistema sea saturado por ruido proveniente de artefactos de movimiento [3].
- Si la señal está corrompida con ruido del artefacto de movimiento, pero no está saturada, se puede filtrar para recuperar la señal de interés. Aquí se propone el método de la plantilla de ciclo ECG en reposo, para eliminar la deriva de la línea basal en la señal de ECG del paciente en movimiento. En esto se supone que el usuario puede establecer un ECG de reposo durante sus actividades diarias.

Este nuevo método se compara con el del filtro paso alto y los métodos de interpolación spline en su capacidad para eliminar la deriva de la línea basal. Aunque existen algunas limitaciones del estudio, se encontró que la técnica de registro de ECG de alta-resolución a baja-ganancia junto al uso de la plantilla de un ciclo en reposo funciona bien para atenuar la deriva de la línea basal de una señal de ECG de un paciente en movimiento. Lograr

eliminar el ruido de una señal de ECG contaminada por el movimiento de un paciente, permitiría el uso de dispositivos portátiles durante las actividades diarias.

Palabras clave —**Artefacto de movimiento, ADC de alta resolución, procesamiento digital de señales, atenuación de deriva de línea basal, detector de complejos QRS.**

ABSTRACT

The recent decades growth of medical devices development has allowed, in addition to their use on healthcare applications, that research about their application in other fields of study such as rehabilitation, or improvement on sports performance is made [1].

The technology on electrocardiography devices has gradually allowed to create less noise-susceptible devices, and at the same time has allowed to make measurements without the requirement to stay in resting state during the study, such as on patients who suffer from seizures, or when acquisition must be made during physical activity; on these cases, noise is induced into the signal of interest [2].

It is aimed through this research to contribute on the development of medical devices able to acquire an ECG signal during movement from the patient, supplying a signal processed by the same device. In this methodology, two together operating elements were contributed:

- High resolution (24 bits) to acquire more detailed signals while low amplification gain is used, which prevents system saturation by motion artifact noise [3].
- If the signal is corrupted by motion artifacts, but is not saturated, it can be processed to recover the signal of interest. A method utilizing an ECG resting cycle template to eliminate baseline wander by movement of a patient is proposed. This supposes that the user is able to stay in resting position during their daily activities to produce an ECG resting cycle.

This new method is compared with a high-pass filter and spline interpolation methods on their capacity to eliminate baseline wander. Although some limitations on this study exist, it was found that the high-resolution ECG acquisition with low amplification gain, together with the usage of an ECG resting cycle template works well to attenuate baseline wander from a corrupted ECG signal from a patient in movement. Achieving to eliminate noise of an ECG signal corrupted by movement from a patient would allow to use portable devices during their daily activities.

Keywords —**Motion artifact, High-resolution ADC, digital signal processing, baseline wander attenuation, QRS complex detector**

ÍNDICE DE CONTENIDO TEMÁTICO

	Página
CAPÍTULO 1. PRESENTACIÓN DE LA INVESTIGACIÓN	1
1.1 Introducción	1
1.2 Planteamiento del problema	2
1.3 Pregunta problema	2
1.4 Hipótesis	2
1.5 Objetivos	3
1.5.1 Objetivo general	3
1.5.2 Objetivo específico	3
1.6 Justificación	3
CAPÍTULO 2. ANTECEDENTES Y MARCO TEÓRICO	5
2.1 Enfermedades Cardiovasculares	5
2.2 Electrocardiograma (ECG)	5
2.2.1 Descripción del electrocardiograma (ECG)	5
2.2.2 Derivaciones bipolares	6
2.3 Medición de bioseñales	7
2.3.1 Conceptos generales de medición de bioseñales	7
2.3.2 Fuentes de ruido	8
2.4 Electrocardiógrafo Holter	11
2.5 Circuito integrado AD8232	13
2.5.1 Tablero Sparkfun de monitoreo cardíaco de una derivación AD8232	14
2.6 Detectores de complejos QRS	15
2.6.1 Algoritmo Pan-Tompkins	16
2.7 Antecedentes de propuestas para atenuar DLB por postprocesamiento de una señal ECG	17
2.7.1 Filtros digitales	17
2.7.2 Replicar tendencia de DLB	18
2.8 Entorno de desarrollo MATLAB	19
2.9 Convertidor analógico-digital (ADC) Delta-Sigma	20
2.10 Microcontrolador C8051F350	22
2.10.1 Módulo ADC de 24 bits	22
2.10.2 Ambiente de programación y desarrollo Simplicity Studio	24
CAPÍTULO 3. METODOLOGÍA	26
3.1 Caracterización de ADC de 24 bits	26
3.1.1 Medición del ruido en modo común	26
3.1.2 Pruebas de captura de ECG y selección de frecuencia de muestreo	27
3.2 Modificación de tablero Sparkfun AD8232	28
3.2.1 Modificación de ganancia	28
3.2.2 Habilitación de terminal REFOUT	29
3.3 Módulo Bluetooth HC-10 y configuración del puerto UART	30

	Página
3.4 Integración de dispositivo portátil	31
3.5 Captura de señal ECG	32
3.5.1 Características de paciente	32
3.6 Reducción de DLB y artefactos de movimiento	35
3.6.1 Filtro digital	35
3.6.2 Interpolación spline	36
3.6.3 Generación de señal esperada	39
3.7 Comparación de métodos de atenuación de ruido	44
3.8 Generación de señal sintética de ECG por medio de generador de funciones	45
3.9 Generación de señal sintética con tren de plantillas	48
3.10 Diseño de detector de complejos QRS	49
3.11 Pruebas de detectores de complejos QRS	51
3.11.1 Algoritmo propio en MATLAB	52
3.11.2 Algoritmo Pan-Tompkins en MATLAB	53
3.11.3 Algoritmo SimpleRST	54
CAPÍTULO 4. RESULTADOS	56
4.1 Caracterización de ADC de 24 bits	56
4.2 Modificación de tablero Sparkfun AD8232	56
4.3 Integración de dispositivo portátil	58
4.4 Filtrado digital	58
4.5 Interpolación spline	59
4.6 Atenuación de DLB con plantilla de ECG en reposo	59
4.7 Comparación de métodos de reducción de ruido	60
4.8 Efectividad de algoritmo propio para detección de picos R de ECG	60
4.8.1 Señal 1	60
4.8.2 Señal 2	61
4.8.3 Señal 3	62
4.9 Efectividad de algoritmo Pan-Tompkins para detección de picos R de ECG	63
4.9.1 Señal 1	63
4.9.2 Señal 2	64
4.9.3 Señal 3	65
4.10 Efectividad de algoritmo SimpleRST para detección de picos R de ECG	66
4.10.1 Señal 1	66
4.10.2 Señal 2	67
4.10.3 Señal 3	68
4.11 Comparación de efectividad de detectores de picos R de ECG	69
CAPÍTULO 5. DISCUSIÓN DE RESULTADOS	70
5.1 Caracterización de ADC de 24 bits	70

	Página
5.2 Modificación de tablero Sparkfun AD8232	70
5.3 Integración de dispositivo portátil	71
5.4 Filtrado	71
5.5 Interpolación spline	71
5.6 Atenuación de DLB con plantilla de ECG en reposo	72
5.7 Efectividad de algoritmo propio para detección de picos R de ECG	72
5.8 Efectividad de algoritmo Pan-Tompkins para detección de picos R de ECG	73
5.9 Efectividad de algoritmo SimpleRST para detección de picos R de ECG	73
5.10 Comparación de efectividad de detectores de picos R de ECG	73
CONCLUSIONES	74
TRABAJO FUTURO	76
REFERENCIAS	77
APÉNDICE A. CÓDIGOS DE MATLAB	82
A1. Algoritmo propio para detección de complejos QRS	82
A2. Filtro digital para atenuar DLB de señal original	82
A3. Código para método por interpolación spline	83
A4. Atenuación de DLB por método propuesto con plantilla	83
A5. Generación de espectro de frecuencias	84
A6. Código para graficar señales en un mismo formato	84

ÍNDICE DE FIGURAS

	Página
CAPÍTULO 2	
2.1 Onda normal de ECG.	5
2.2 Derivaciones bipolares de ECG.	6
2.3 Diagrama general para medición de biopotenciales.	7
2.4 Esquema de bloques para adquisición de ECG.	8
2.5 Señal de ECG contaminada por deriva de línea basal.	9
2.6 Señal de ECG contaminada por interferencia de 60 Hz de la toma de alimentación.	9
2.7 Señal de ECG contaminada por ruido de EMG.	10
2.8 Señal de ECG contaminada por artefacto de movimiento.	11
2.9 Dispositivo Holter colocado en paciente.	12
2.10 Diagrama de bloques funcional de circuito integrado AD8232.	13
2.11 Diagrama electrónico de configuración “Cardiac Monitor” recomendada por el fabricante.	14
2.12 Tablero con configuración “Cardiac Monitor” de circuito integrado AD8232 fabricado por la compañía Sparkfun.	15
2.13 Etapas de algoritmo Pan-Thompkins para detección de complejos QRS de señal ECG implementado en MATLAB.	16
2.14 Esquema simple de un bloque de filtrado digital.	17
2.15 Comparación de onda ECG durante estado de reposo (azul) y onda ECG después de realizar actividad física (rojo).	19
2.16 Interfaz gráfica del usuario de aplicación “Filter Designer” de MATLAB.	20
2.17 Diagrama de funcionamiento de modulador delta-sigma.	21
2.18 Prueba de funcionamiento de modulador sigma-delta en Simulink. a) Esquema de bloques de modulador sigma-delta. b) Señal analógica de entrada. c) Señal de salida del diferenciador. d) Señal de salida del integrador. e) tren de pulsos generado por comparador.	22
2.19 Diagrama interno de bloques de módulo de convertidor ADC de 24 bits de microcontrolador C8051F350.	23
2.20 Tablero de desarrollo con microcontrolador C8051F350 fabricado por la compañía Silicon Labs.	24
2.21 Interfaz de usuario de ambiente de programación Simplicity Studio con programa ejemplo para utilización del módulo ADC de 24 bits.	25
CAPÍTULO 3	
3.1 Resultado de barrido de frecuencias para medición de ruido de cuantificación en modo común de ADC.	27
3.2 Señales de ECG muestreadas con ADC de microcontrolador C8051F350. a) Señal muestreada con ganancia 100 a 120 Hz. b) Señal muestreada con ganancia 100 a 360 Hz.	27
3.3 Diagrama electrónico de tablero Sparkfun AD8232 con configuración “cardiac monitor”.	28

	Página
3.4 Señal de ECG muestreada a 180 Hz con ganancia 1100. Los límites superior e inferior de saturación se muestran con franjas de color rojo.	29
3.5 Localización de terminal REFOUT indicada en tablero Sparkfun AD8232.	30
3.6 Módulo bluetooth HC-10.	31
3.7 Esquema de bloques de etapas del dispositivo portátil integrado.	31
3.8 Paciente en estado de reposo durante adquisición de señal ECG.	33
3.9 Señal de ECG adquirida de paciente en reposo a 180 Hz y ganancia 100×.	33
3.10 Paciente durante adquisición de señal en posición firme y movimiento repentino. a) Paciente en posición firme. b) paciente tras realizar movimiento repentino colocándose en posición de marcha.	34
3.11 Señal original adquirida.	34
3.12 Parámetros especificados en interfaz gráfica de herramienta “Filter designer” para diseño de filtro digital pasa-altas.	35
3.13 Respuesta en frecuencia de filtro pasa-altas digital generado con herramienta “Filter designer” (figura 3.12) visualizado con la función fvtool de MATLAB.	35
3.14 Puntos de señal original seleccionados para interpolar por interpolación spline.	36
3.15 a) Señal original. b) Señal generada tras interpolar puntos seleccionados de señal original.	38
3.16 Selección de primera onda de ECG presentada en la señal original en estado de reposo para utilizar como plantilla.	39
3.17 Plantilla de ECG en reposo seleccionada.	40
3.18 Tren de plantillas generado.	40
3.19 Señal resultante de sustraer tren de plantillas (figura 3.18) a señal original (figura 3.11).	41
3.20 Parámetros especificados en interfaz gráfica de herramienta “Filter designer” para diseño de filtro digital pasa-bajas.	41
3.21 Respuesta en frecuencia de filtro pasa-bajas digital generado con herramienta “Filter designer” (figura 3.20) visualizado con la función fvtool de MATLAB.	42
3.22 Línea de tendencia de DLB generada al filtrar la señal obtenida tras sustraer el tren de plantillas de la señal original (figura 3.19).	42
3.23 Señal generada tras sustraer la línea de tendencia de DLB de la señal original.	43
3.24 Espectro de frecuencias de señal generada al sustraer DLB de señal original (figura 3.23).	43
3.25 Parámetros especificados en interfaz gráfica de herramienta “Filter designer” para diseño de filtro digital notch para atenuar componente de frecuencia de 59 Hz.	44
3.26 Onda ECG diseñada en software ArbExpress para generar en generador de funciones de ondas arbitrarias.	45
3.27 Generador de funciones Tektronix AFG3021B con habilidad de generar funciones arbitrarias.	46

	Página
3.28 Generador de funciones Tektronix AFG3021B generando onda ECG diseñada.	46
3.29 Osciloscopio RIGOL SD1102D realizando monitoreo de onda ECG generada por generador de funciones.	47
3.30 a) Señal generada por función chirp. b) Señal generada tras añadir la señal generada por generador de funciones a la señal generada por función chirp.	48
3.31 Señal generada tras añadir la señal creada por la función chirp (figura 3.30a) al tren de plantillas (figura 3.18).	48
3.32 Diagrama de flujo de funcionamiento de algoritmo creado para detección de complejos QRS.	50
3.33 Ubicación de picos detectados de la señal 1 por algoritmo propio.	52
3.34 Ubicación de picos detectados de la señal 2 por algoritmo propio.	52
3.35 Ubicación de picos detectados de la señal 3 por algoritmo propio.	53
3.36 Ubicación de picos detectados de la señal 1 por algoritmo Pan-Tompkins.	53
3.37 Ubicación de picos detectados de la señal 2 por algoritmo Pan-Tompkins.	54
3.38 Ubicación de picos detectados de la señal 3 por algoritmo Pan-Tompkins.	54
3.39 Ubicación de picos detectados de la señal 1 por algoritmo SimpleRST.	54
3.40 Ubicación de picos detectados de la señal 2 por algoritmo SimpleRST.	55
3.41 Ubicación de picos detectados de la señal 3 por algoritmo SimpleRST.	55
 CAPÍTULO 4	
4.1 Tablero Sparkfun AD8232 con terminal REFOUT habilitada y ganancia 100.	57
4.2 Dispositivo portátil integrado para adquisición de ECG.	58
4.3 Señal generada tras utilizar con la señal original el filtro diseñado.	58
4.4 Señal generada tras sustraer a la señal original la línea de tendencia generada por interpolación spline.	59
4.5 Señal generada tras sustraer a la señal original la línea de tendencia generada en sección 3.6.3.	60

ÍNDICE DE TABLAS

	Página
CAPÍTULO 2	
2.1 Principales características del módulo ADC del microcontrolador C8051F350.	23
CAPÍTULO 3	
3.1 Frecuencias utilizadas para realizar barrido para caracterización del ADC.	26
3.2 Configuración de puerto UART de microcontrolador C8051F350 para transmisión serial de datos.	30
3.3 Características del paciente de cual se adquirió la señal de ECG.	32
3.4 Números de muestra de puntos de señal original seleccionados para interpolar por interpolación spline.	36
3.5 Números de muestra de cada señal en donde se encontraron picos R manualmente.	51
CAPÍTULO 4	
4.1 Valores de desviación estándar, SNR, y bits efectivos medidos tras realización de barrido de frecuencias durante caracterización de ADC.	56
4.2 Comparación entre cada método descrito en la investigación. Se comparan los valores de RMS y desviación estándar de cada señal procesada respecto a los valores de RMS y desviación estándar de la señal original. Se muestra la atenuación.	60
4.3 Comparación de ubicación de picos R detectados en señal 1 por algoritmo propio respecto a la ubicación de picos R detectados manualmente.	61
4.4 Comparación de ubicación de picos R detectados en señal 2 por algoritmo propio respecto a la ubicación de picos R detectados manualmente.	62
4.5 Comparación de ubicación de picos R detectados en señal 3 por algoritmo propio respecto a la ubicación de picos R detectados manualmente.	63
4.6 Comparación de ubicación de picos R detectados en señal 1 por algoritmo Pan-Tompkins respecto a la ubicación de picos R detectados manualmente.	64
4.7 Comparación de ubicación de picos R detectados en señal 2 por algoritmo Pan-Tompkins respecto a la ubicación de picos R detectados manualmente.	65
4.8 Comparación de ubicación de picos R detectados en señal 3 por algoritmo Pan-Tompkins respecto a la ubicación de picos R detectados manualmente.	66
4.9 Comparación de ubicación de picos R detectados en señal 1 por algoritmo SimpleRST respecto a la ubicación de picos R detectados manualmente.	66
4.10 Comparación de ubicación de picos R detectados en señal 2 por algoritmo SimpleRST respecto a la ubicación de picos R detectados manualmente.	68
4.11 Comparación de ubicación de picos R detectados en señal 3 por algoritmo SimpleRST respecto a la ubicación de picos R detectados manualmente.	69
4.12 Efectividad de detectores QRS probados con cada señal. Señal 1: Señal original del paciente. Señal 2: Señal sintética generada en MATLAB. Señal 3: Señal generada por generador de funciones.	69

CAPÍTULO 1. PRESENTACIÓN DE LA INVESTIGACIÓN

1.1 Introducción

El avance en el desarrollo de dispositivos médicos en décadas recientes ha permitido que, además de utilizar estos dispositivos para aplicaciones en el cuidado de la salud de pacientes con enfermedades, se investigue acerca de sus aplicaciones en otros campos, por ejemplo, en la rehabilitación, o en la mejora del rendimiento deportivo [1]. La electrocardiografía ha sido utilizada desde el siglo XX. El impacto del estudio de electrocardiografía hizo que se continuara realizando investigación para la mejora del estudio y creación de mejores dispositivos capaz de realizar el estudio.

La tecnología implementada paulatinamente en electrocardiógrafos ha permitido crear dispositivos con menor susceptibilidad a fuentes de ruido y esto a su vez ha permitido realizar mediciones en pacientes con necesidades distintas a las de un paciente que pueda permanecer en reposo durante el estudio, como pacientes que sufren convulsiones, realizan movimiento involuntario, o cuyo estudio deba ser realizado mientras realizan actividad física, entre otros [2]. Debido a que el movimiento del paciente puede introducir ruido de una amplitud similar a la amplitud de la señal de interés, es necesario contar un rango dinámico suficientemente amplio para evitar saturar el sistema encargado de realizar la captura de la señal, esto se puede lograr al utilizar un sistema de alta resolución [3].

La demanda de estos dispositivos crece con el tiempo y esto propicia a encontrar limitaciones en ellos. En muchos casos es necesaria la lectura del electrocardiograma (ECG) de un paciente durante periodos continuos prolongados, lo cual requiere en ciertos casos el uso de un dispositivo portátil [4]. Debido al ruido que provoca en la señal de interés el movimiento del paciente, se han estudiado y propuesto métodos para atenuar o eliminar el ruido de una señal de ECG contaminada por distintas fuentes existentes de ruido. Lograr eliminar el ruido de una señal de ECG contaminada por el movimiento del paciente, permitiría la medición del ECG de un paciente en constante movimiento durante la misma. Las técnicas de procesamiento digital permitirían procesar una señal contaminada previamente capturada para obtener información útil de esta, sin tener que erradicar el ruido durante la medición.

1.2 Planteamiento del problema

El incremento actual de las enfermedades cardiovasculares (ECV) ha provocado que la demanda de estudios de electrocardiografía (ECG) incremente también. Los dispositivos encargados de capturar una señal de ECG han sido desarrollados y mejorados a lo largo del tiempo encontrando diversas aplicaciones además de la detección e identificación de patologías. Algunas aplicaciones, como la medición de una señal de ECG durante la actividad física, requieren que la medición se realice mientras el paciente se encuentra en movimiento, lo cual provoca la aparición de artefactos de movimiento en la señal, contaminándola y dificultando su interpretación [5].

Otra necesidad ha sido la portabilidad de estos dispositivos, lo cual muestra también limitaciones debido al movimiento, uso de baterías, volumen físico, entre otros [6].

1.3 Pregunta problema

¿Se puede, mediante un adecuado procesamiento digital utilizando una plantilla de ECG capturada del paciente mientras se encontraba en reposo, lograr la obtención de una señal de ECG limpia y útil para interpretación, adquirida de un paciente en movimiento, por un dispositivo portátil de alta resolución (24 bits)?

1.4 Hipótesis

La utilización de una plantilla de ECG de un paciente en reposo para procesar una señal de ECG capturada por un dispositivo portátil de alta resolución permite la atenuación de artefactos de movimiento de una señal de ECG capturada durante el movimiento del paciente.

1.5 Objetivos

1.5.1 Objetivo general

Obtener una señal de ECG postprocesada limpia de un paciente en movimiento capturada por un dispositivo portátil de alta resolución

1.5.2 Objetivos específicos

- Caracterizar el convertidor analógico-digital (ADC) de 24 bits del microcontrolador C8051F350 para seleccionar frecuencia de muestreo.
- Modificar el tablero Sparkfun para habilitación de terminal REFOUT del chip AD8232 y lograr una medición diferencial con el ADC
- Acondicionar el módulo bluetooth HC-10 para transmitir datos del microcontrolador a una computadora de forma inalámbrica.
- Adquirir señal de ECG de paciente.
- Crear una señal sintética de ECG con plantilla de ECG de paciente en reposo para postprocesar señal adquirida durante el movimiento.

1.6 Justificación

Las necesidades actuales en el desarrollo de dispositivos de electrocardiografía aportan áreas de oportunidad en la innovación de dispositivos. Los dispositivos actualmente fabricados por compañías dedicadas a la fabricación de dispositivos médicos producen dispositivos cuya resolución no supera los 16 bits [7].

La utilización de un dispositivo de 24 bits para capturar señales de ECG permite realizar lecturas sin necesidad de utilizar ganancias demasiado elevadas y disminuir la susceptibilidad del sistema a saturación por artefactos de movimiento [3].

Las tecnologías existentes permiten el desarrollo de nuevos métodos de postprocesamiento de señales digitales que faciliten la atenuación de ruido en señales biofisiológicas como el ECG [8]. Al desarrollar un método capaz de atenuar artefactos de movimiento en una señal de ECG se crea un campo de oportunidad para investigar acerca de la implementación en tiempo real del método desarrollado.

Mediante la presente investigación se busca contribuir al desarrollo de dispositivos médicos que logren capturar una señal de ECG mientras el paciente se encuentre en movimiento, proporcionando una señal procesada por el mismo dispositivo.

CAPÍTULO 2. ANTECEDENTES Y MARCO TEÓRICO

2.1 Enfermedades Cardiovasculares

Las Enfermedades Cardiovasculares (ECV) son un grupo de enfermedades asociadas a trastornos del corazón y vasos sanguíneos. Las ECV han sido catalogadas como una de las principales causas de muerte en el pasado reciente, siendo la principal causa de muerte en algunos países del mundo. La Organización Mundial de la Salud (OMS) reportó que en 2019

murieron un estimado de 17.9 millones de personas a causa de ECV, lo cual representó un 32% del total de muertes a nivel mundial durante ese mismo año [9].

2.2 Electrocardiograma (ECG)

2.2.1 Descripción del electrocardiograma (ECG)

El electrocardiograma (ECG) es un estudio clínico que proporciona información acerca del comportamiento eléctrico del corazón del paciente. El ECG es el registro de las diferencias de potencial que existen entre varios puntos de la superficie del organismo a lo largo de un periodo de tiempo. La medición de un ECG puede ser realizada colocando electrodos en la superficie de la piel del organismo [10].

En cada ciclo cardiaco, se producen en el registro ondas y deflexiones. El eje de las ordenadas representa el voltaje, y el eje de las abscisas representa el tiempo.



Figura 2.1 Onda normal de ECG [11].

La primera onda es llamada onda P y se produce por la activación auricular. La despolarización ventricular es señalada por el complejo QRS, el cual inicia con una deflexión negativa llamada onda Q. A la primera deflexión positiva se le conoce como onda R, y la siguiente deflexión negativa es llamada onda S. Finalmente, se produce una onda tras un intervalo, llamada onda T, la cual señala la repolarización ventricular [10], [12], [13].

2.2.2 Derivaciones bipolares

Willem Einthoven colocó los electrodos en tres extremidades: brazo derecho, brazo izquierdo, y pierna izquierda, midiendo los potenciales del siguiente modo:

- Derivación I (DI): electrodo positivo en brazo izquierdo y electrodo negativo en brazo derecho.
- Derivación II (DII): electrodo positivo en pierna izquierda y electrodo negativo en brazo derecho.
- Derivación III (DIII): electrodo positivo en pierna izquierda, y electrodo negativo en brazo izquierdo [10], [13], [14].

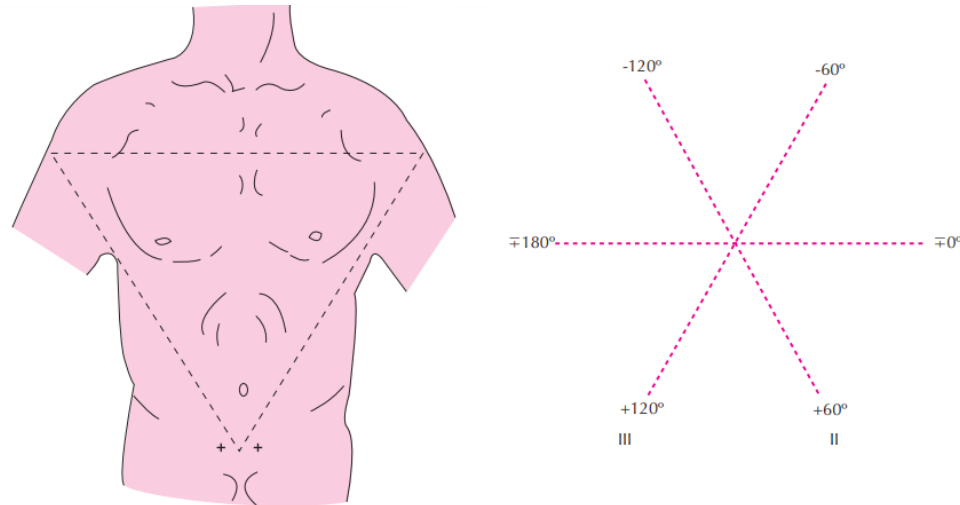


Figura 2.2 Derivaciones bipolares de ECG [10].

Estas derivaciones son utilizadas en la actualidad y cada derivación proporciona una perspectiva de la actividad eléctrica del corazón en dos dimensiones, según la colocación de los electrodos [10], [13], [15].

Al encontrar desviaciones de la morfología considerada “normal” de un ECG, se pueden identificar patologías en el órgano cardíaco del paciente [10], [13].

2.3 Medición de bioseñales

2.3.1 Conceptos generales de medición de bioseñales

Para realizar la medición de una señal biofisiológica, es necesaria una interfaz entre el organismo cuyo potencial desea ser medido, y el circuito electrónico de medición, la cual es lograda por electrodos de cloruro de plata al estar en contacto con la superficie de la piel.

Los electrodos realizan la función de un transductor y convierten la información biológica en una diferencia de potencial que puede ser medida y cuantificable [16], [17],[17].

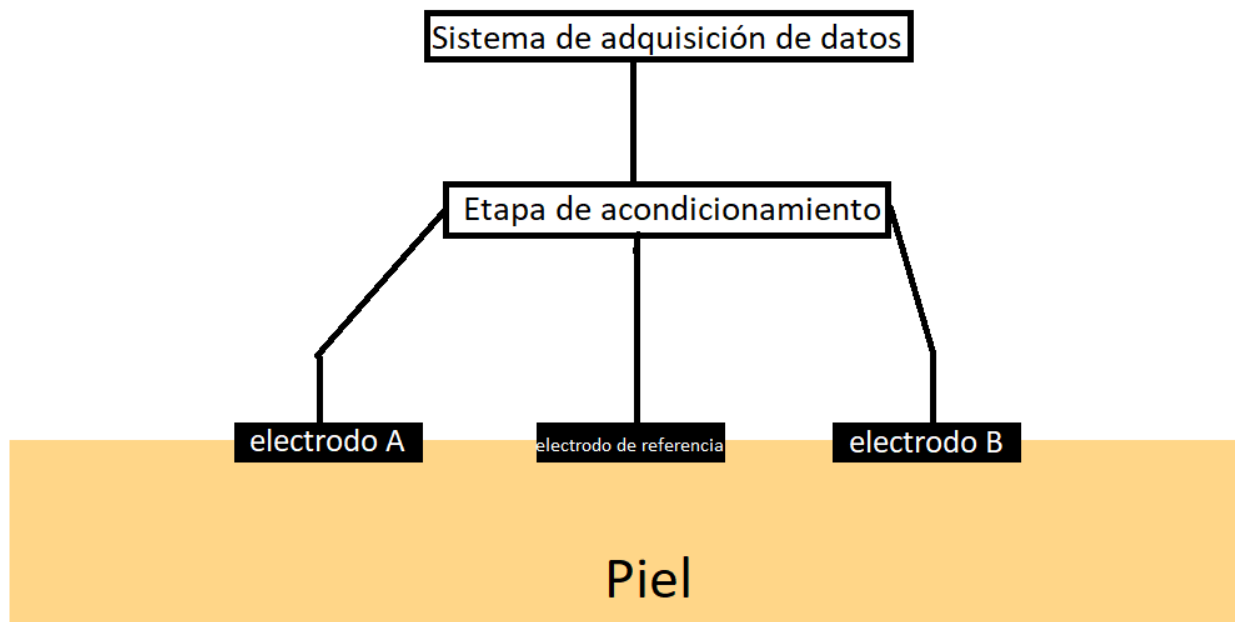


Figura 2.3 Diagrama general para medición de biopotenciales.

A diferencia de la medición de voltaje con un voltímetro de un sistema distinto al de un organismo vivo, los voltajes medidos de un organismo son de magnitud muy pequeña comparada con la mínima magnitud que un voltímetro convencional podría detectar, por lo tanto, es necesario amplificar los potenciales de los electrodos por medio de un circuito acondicionador de señal, de tal manera que pueda ser detectado por el convertidor analógico-digital (ADC) de un sistema de adquisición de datos a utilizar. Así mismo, en el circuito acondicionador de señal se incluye una etapa de filtrado [12].

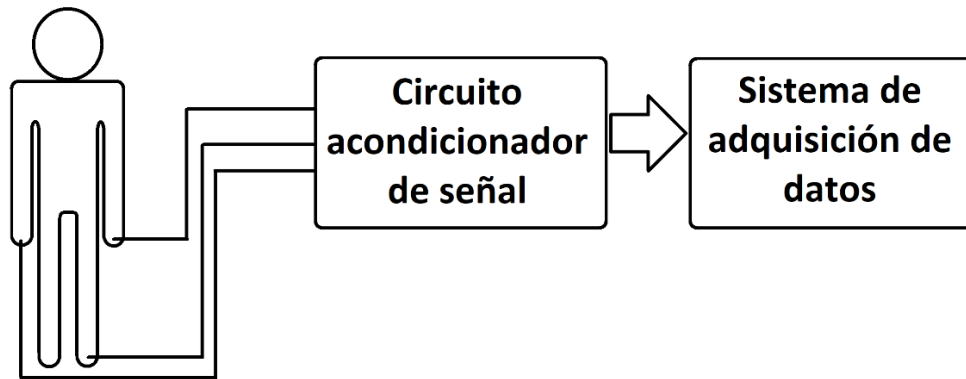


Figura 2.4 Esquema de bloques para adquisición de ECG.

2.3.2 Fuentes de ruido

Debido a que las señales biofisiológicas son de una magnitud de origen muy pequeña (del orden entre μV y mV) son señales muy susceptibles al ruido analógico [16].

Deriva de la línea basal (DLB): Es el efecto en el que el eje base (eje x) de una señal no permanece con un valor fijo (cero) a lo largo del tiempo. Esto provoca que la señal tenga un desplazamiento del estado basal. Las causas de este fenómeno suelen ser: movimiento del paciente, electrodos inadecuados, respiración [18].

La frecuencia de la DLB es baja y se encuentra cerca de 0.5 Hz, sin embargo, este valor puede incrementar si se presenta movimiento o respiración de alta intensidad como durante el ejercicio [18].

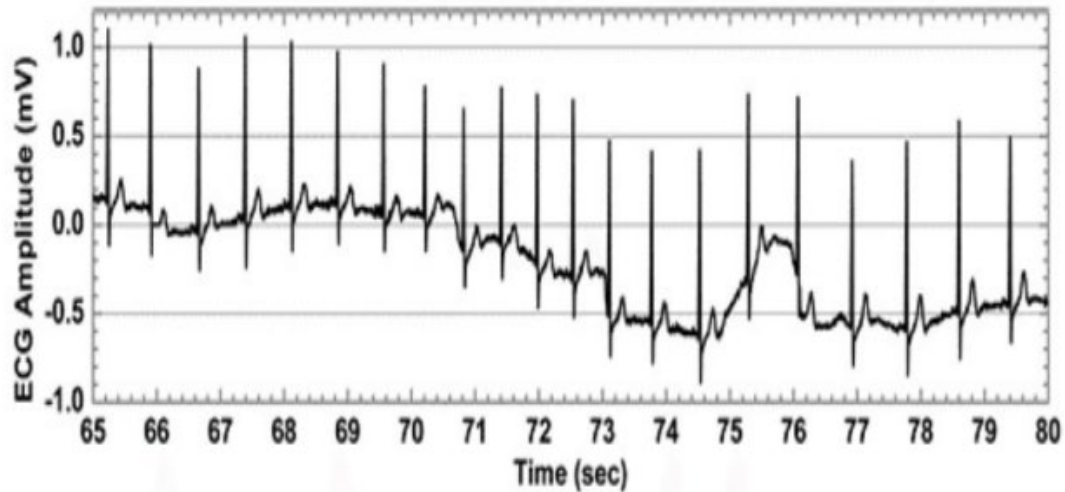


Figura 2.5 Señal de ECG contaminada por deriva de línea basal [18].

Debido a la baja frecuencia de la DLB, esta puede ser atenuada utilizando un filtro paso-altas.

Interferencia de la toma de alimentación: La toma de alimentación de una instalación produce campos magnéticos que representan una fuente de ruido bastante común para cualquier señal analógica. La presencia de este ruido se caracteriza por tener una forma senoidal de 50 Hz o 60 Hz (según la región geográfica). Debido a que la frecuencia del ruido es similar a la de componentes de la señal de interés, no resulta adecuado utilizar filtros pasa-altas ni filtros pasa-bajas para atenuar el ruido, sino que se requiere la utilización de un filtro notch que atenúe específicamente la frecuencia deseada [16], [18], [19].

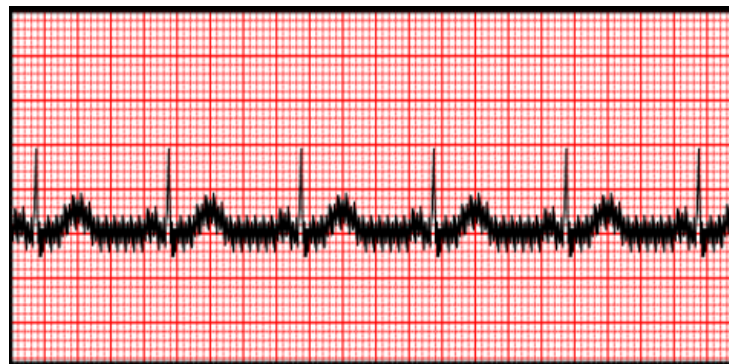


Figura 2.6 Señal de ECG contaminada por interferencia de 60 Hz de la toma de alimentación [18], [20].

Ruido electromiográfico (EMG): Esta fuente de ruido es provocada por la activación de las fibras musculares encontradas bajo la superficie de la piel en donde se encuentra colocado algún electrodo. Este ruido representa una mayor complicación que la interferencia de 50/60 Hz y la DLB, ya que el ruido EMG es aleatorio y puede alcanzar la amplitud del complejo QRS de la señal de interés [21], [22].

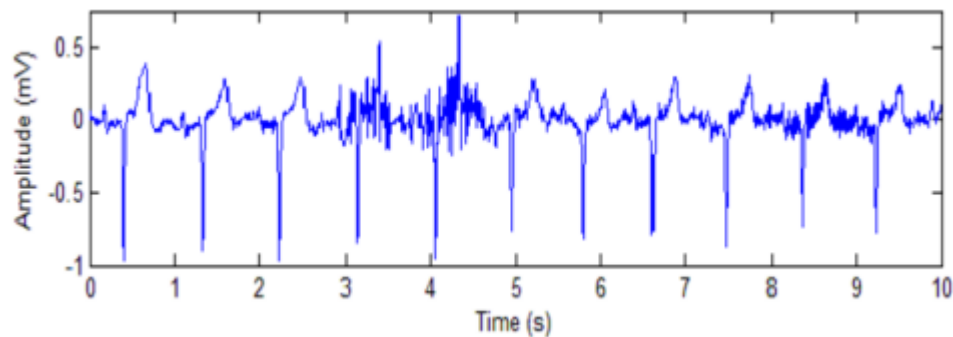


Figura 2.7 Señal de ECG contaminada por ruido de EMG [21].

Una técnica utilizada para reducir el ruido EMG es promediar la región en la que se presenta, sin embargo, esto está limitado a que todos los complejos QRS en la señal sean iguales y que hayan ocurrido ya varios latidos.

Es muy común encontrar ruido EMG al realizar mediciones durante el ejercicio debido a la contracción de distintos grupos musculares del cuerpo.

Artefactos de movimiento: Los artefactos de movimiento son provocados por el movimiento de los electrodos. Ocurren por diversas causas como estiramiento de la piel, provocando una alteración en la impedancia de la piel. La presencia de artefactos de movimiento se asemeja a la de la DLB, sin embargo, son más problemáticos ya que cuyo espectro de frecuencias se superpone al de la señal de interés. Suelen encontrarse entre el rango de frecuencias de 1 Hz y 10 Hz. En ocasiones son de mayor amplitud que la señal de ECG y pueden confundirse con complejos QRS [22].

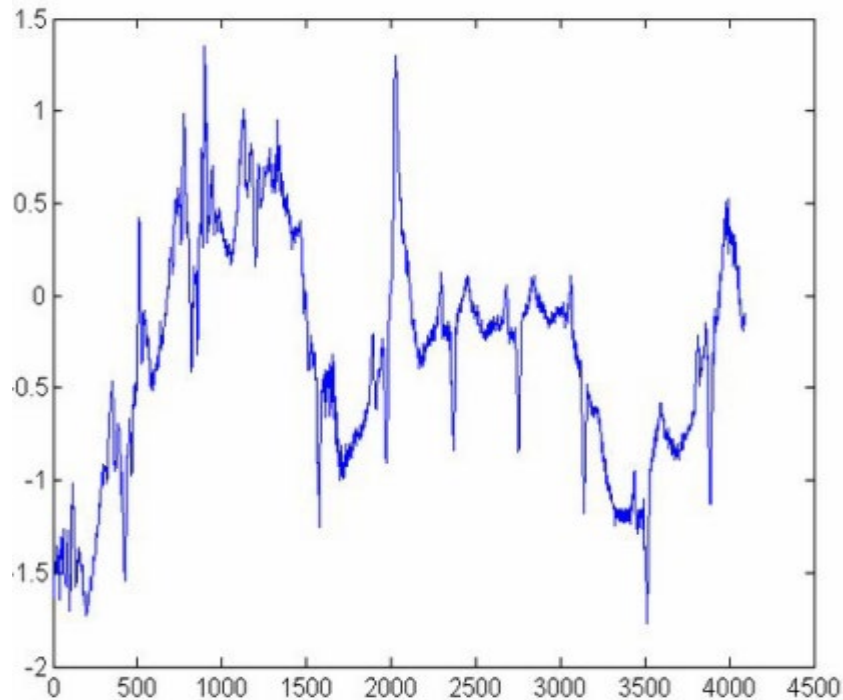


Figura 2.8 Señal de ECG contaminada por artefacto de movimiento [23].

2.4 Electrocardiógrafo Holter

El desarrollo de dispositivos portátiles de electrocardiografía ha incrementado debido a sus aplicaciones en pacientes ambulatorios [24].

El Holter es un electrocardiógrafo capaz de realizar la medición de un paciente mientras este realiza actividades habituales de su vida diaria [25].

El primer holter se creó en 1940 por el Doctor en Ciencias Físicas y Químicas Norman Holter, lo cual representó uno de los avances más importantes en la historia de la cardiología. El dispositivo pesaba alrededor de 40 kg. Este peso ha ido disminuyendo a lo largo de los años, así como su tamaño. También ha incrementado sus prestaciones.

Al principio, se indicaba que el Holter tenía el objetivo de detectar arritmias cardíacas en el paciente, ya que a simple vista se podrían detectar anomalías en la separación entre complejos QRS de la señal. Fue hasta 1970 que se comenzó a utilizar con más funcionalidades y más canales (más derivaciones).

En los años 80, contando ya con desarrollo de software y tecnología informática, se comenzó a utilizar el almacenamiento de datos medidos por un Holter. Lo cual permitió análisis posterior y post-procesamiento [5].

La resolución típica de un Holter es menor que la de un electrocardiógrafo no portátil. Un electrocardiógrafo convencional posee una resolución típica de 12 a 16 bits [7].

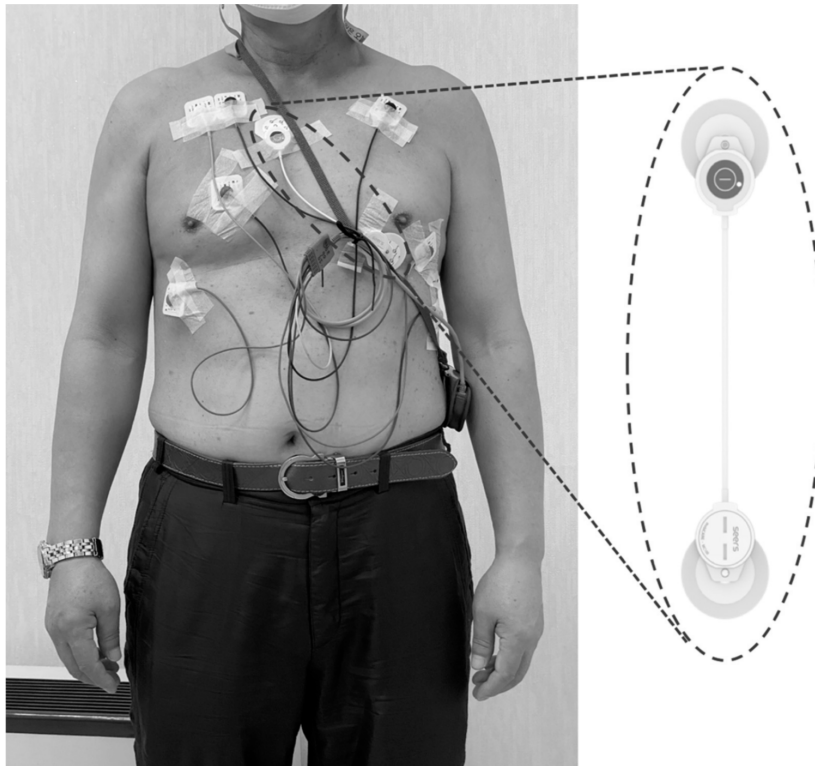


Figura 2.9 Dispositivo Holter colocado en paciente [26].

Una resolución menor que 17 bits vuelve a la señal más capaz de saturar al sistema de adquisición si la señal se encuentra contaminada por artefactos de movimiento.

Un sistema de mayor resolución permite la lectura de señales con baja ganancia sin necesidad de modificar los voltajes de referencia del ADC y así evitar la saturación del sistema por artefactos de movimiento [3].

2.5 Circuito integrado AD8232

El circuito integrado AD8232 es un dispositivo fabricado por la compañía Analog Devices.

El dispositivo se encuentra diseñado para extraer, amplificar, y filtrar señales biofisiológicas de baja amplitud capturadas bajo condiciones ruidosas como las que provocan los artefactos de movimiento, entre otras (ver sección 1.3.2).

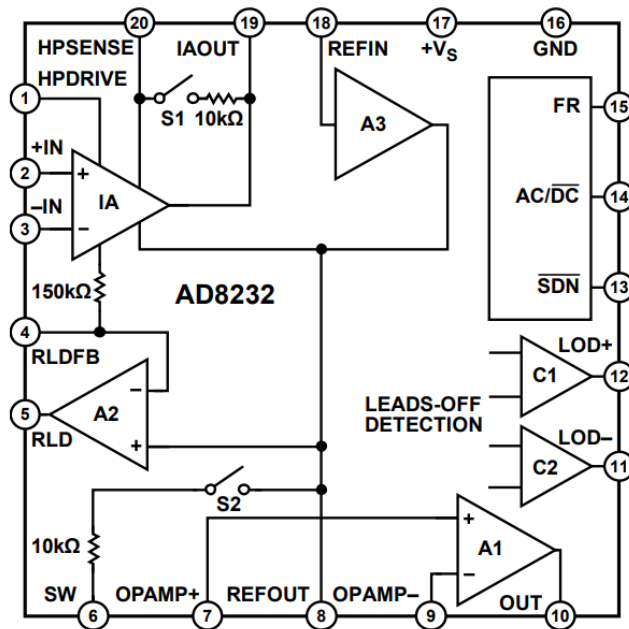


Figura 2.10 Diagrama de bloques funcional de circuito integrado AD8232 [27].

Las aplicaciones enlistadas por el fabricante son: monitores de ritmo cardiaco para actividad física, ECG portátil, Monitores de salud remotos, periféricos de videojuegos, adquisición de señales biofisiológicas.

Para el funcionamiento se requiere una polarización de entre 2.0 voltios y 3.5 voltios, valores que permiten polarizar también diversos microcontroladores. También indica el fabricante que su relación de rechazo al modo común es de 80 dB.

Internamente, el circuito proporciona una ganancia de la señal de entrada de $100\times$, la cual puede ser incrementada por medio de arreglos de resistencias utilizando las terminales

El fabricante proporciona diagramas de circuitos recomendados para distintas aplicaciones del dispositivo. Una de las aplicaciones que el fabricante recomienda, para la cual proporciona su diagrama electrónico, es la aplicación de monitor cardíaco [27].

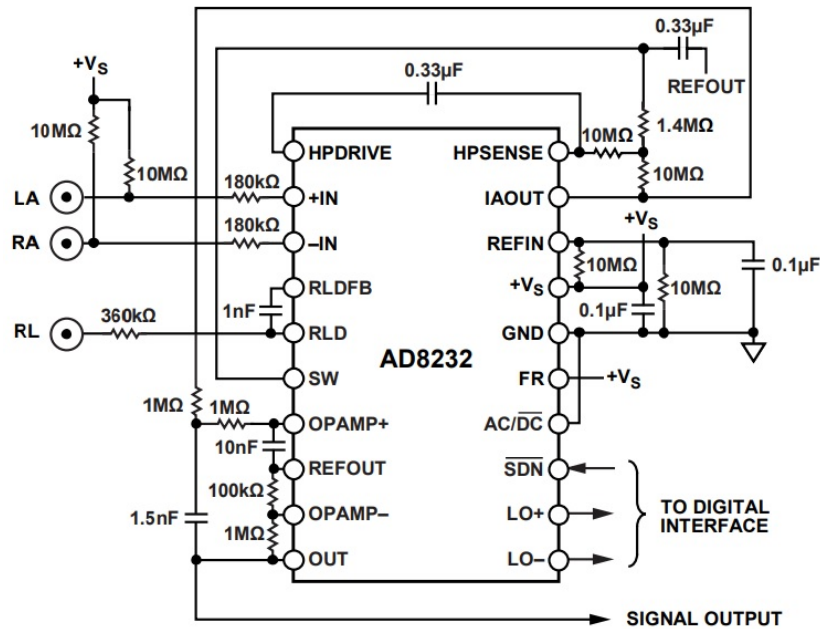


Figura 2.11 Diagrama electrónico de configuración “Cardiac Monitor” recomendada por el fabricante [27].

2.5.1 Tablero Sparkfun de monitoreo cardíaco de una derivación AD8232

La compañía Sparkfun fabrica un tablero de monitoreo cardiaco de una derivación utilizando la configuración de monitor cardiaco del circuito integrado AD8232 recomendada por el fabricante de este, proporcionando una ganancia total de $1100\times$ [27]–[29].

Este tablero cuenta con una entrada para cables de electrodo y utiliza la derivación bipolar DI, sin embargo, la derivación se puede modificar cambiando la colocación de los electrodos.

Este tablero tiene un costo de 21.50 dólares americanos [28].

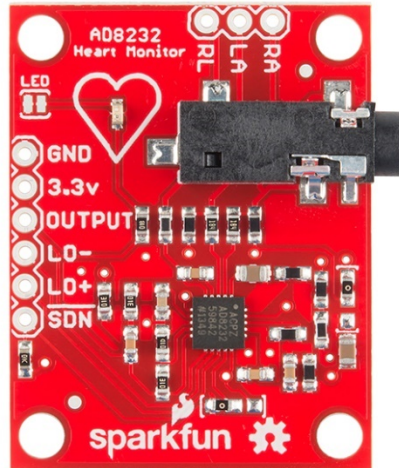


Figura 2.12 Tablero con configuración “Cardiac Monitor” de circuito integrado AD8232 fabricado por la compañía Sparkfun [28].

2.6 Detectores de complejos QRS

La detección de complejos QRS consiste en identificar los picos R de una señal de ECG. Detectar los picos R de un ECG proporciona información acerca del ritmo cardiaco del paciente. Desviaciones en la forma y tiempo de aparición de un complejo QRS de su forma y frecuencia normal refleja un trastorno en el órgano cardiaco [30], [31].

Una aplicación de la detección de complejos QRS es identificar en tiempo real el ritmo cardiaco de un paciente, lo cual facilita la detección de arritmias.

Es complicada la detección de complejos QRS debido a la gran variabilidad de estos, pero también por el ruido que pueda contener una señal de ECG. Por ejemplo, la presencia de ruido de alta frecuencia podría confundir a un detector de complejos QRS debido a la alta frecuencia con que el mismo complejo QRS varía. Además, según la derivación utilizada para medir la señal de ECG del paciente, la morfología de cada segmento de la señal es distinta, incluyendo al complejo QRS [32].

Debido a la utilidad de la detección de complejos QRS, se han creado diversos algoritmos para este propósito que ya son conocidos [30]–[33].

2.6.1 Algoritmo Pan-Tompkins

Propuesto en 1985 por Jiapu Pan y Willis J. Tompkins fue el algoritmo Pan-Tompkins. Este algoritmo consiste detectar complejos QRS después de realizar un preprocesamiento a la señal para atenuar el ruido que complique la detección. La detección se realiza analizando la pendiente, amplitud, y ancho de cada segmento de la señal a lo largo de esta.

El algoritmo es capaz de adaptarse a cambios de ritmo y morfología de la señal al automáticamente definir valores de umbral que indiquen si una alta pendiente, amplitud, y ancho de un segmento de la señal corresponde a los valores de estos en un complejo QRS.

En una primera etapa se aplica un filtro pasa-banda de 5 a 12 Hz aplicando un filtro pasa-bajas y un filtro pasa-altas en cascada.

Después de aplica un filtro derivativo que proporcione información acerca de la pendiente de la señal a lo largo de esta, identificando regiones de pendiente elevada.

Una vez filtrada la señal, se eleva al cuadrado la misma para reducir la probabilidad de confundir un pico R con una onda T debido a su amplitud.

Finalmente, para identificar la duración de un complejo QRS se promedian los valores de la señal presentados en segmentos iguales de la misma [31], [32].

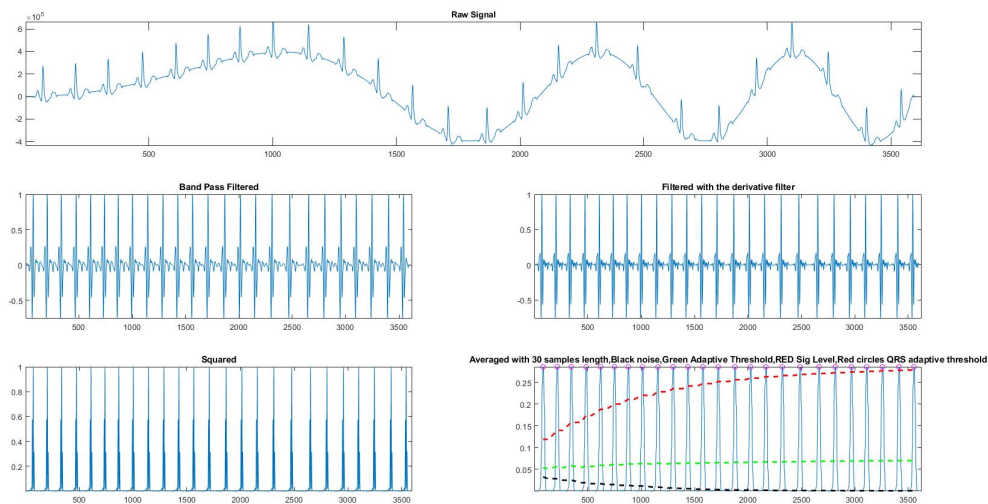


Figura 2.13 Etapas de algoritmo Pan-Thompkins para detección de complejos QRS de señal ECG implementado en MATLAB [34].

El algoritmo Pan-Tompkins ha sido implementado en distintas aplicaciones para la realización de pruebas de funcionamiento con señales de ECG [32], [34].

2.7 Antecedentes de propuestas para atenuar DLB por postprocesamiento de una señal ECG

2.7.1 Filtros digitales

Un filtro digital es un sistema de procesamiento digital de señales (PDS) que en su entrada recibe una señal digitalizada, y en su salida proporciona una señal distinta de la que recibió en su entrada. Para esto, se realizan operaciones matemáticas utilizando los valores de las distintas muestras que componen a la señal de entrada [35].



Figura 2.14 Esquema simple de un bloque de filtrado digital.

Las operaciones matemáticas realizadas por un filtro digital son indicadas por medio de una ecuación llamada “ecuación de diferencias” y tiene la siguiente forma general:

$$\begin{aligned}
 &y(n) + a_1y(n - 1) + \dots + a_Ny(n - N) \\
 &= b_0x(n) + b_1x(n - 1) + \dots + b_Mx(n - N)
 \end{aligned}
 \tag{2.1}$$

En donde $a_1, \dots, a_N$ y $b_0, b_1, \dots, b_M$ son los coeficientes de la ecuación, $y(n)$ es la señal de salida, y $x(n)$ es la señal de entrada. N es el número de muestras de la señal de entrada, y n es el número de muestra actual [36].

Aunque los filtros digitales se pueden utilizar en muchísimas aplicaciones distintas, uno de sus principales usos es atenuar y eliminar ciertos componentes de frecuencia como los filtros analógicos lo realizan. Por lo tanto, una señal digital ruidosa puede ser mejorada

con un filtro digital si se puede analizar la información de la señal en el dominio del tiempo y su frecuencia [35].

Para atenuar la DLB en una señal de ECG se diseña un filtro pasa-altas debido a que la DLB es de baja frecuencia, sin embargo, la señal de interés contiene también componentes de baja frecuencias, como las ondas P y T, por lo tanto, utilizar un filtro pasa-altas podría atenuar también componentes de la señal de interés, si la frecuencia de la DLB tiene un valor similar. Esto presenta limitaciones en la utilización de filtros digitales para limpiar señales de ECG [37], [38].

2.7.2 Replicar tendencia de DLB

Para atenuar la DLB de una señal de ECG se ha buscado replicarla por distintos métodos como la interpolación de puntos de la señal ruidosa.

Replicar la tendencia de la DLB permitiría sustraer la tendencia generada de la señal ruidosa para obtener como resultado la señal de ECG de interés. La clara ventaja de esta propuesta contra la utilización de filtros es que no se perdería información útil contenida en la señal.

Una propuesta para la reducción de artefactos de movimiento que se ha estudiado mayormente en señales de electroencefalograma (EEG) y otras señales biofisiológicas ha sido la utilización de acelerómetros para estimar el artefacto de movimiento provocado por el movimiento del paciente y así replicarle para sustraerle de una señal ruidosa. La medición del artefacto de movimiento se realiza de forma separada de la medición de la señal de ECG.

Un estudio midió el tiempo que dura el complejo QRS en dos señales de un mismo paciente: una señal capturada en estado de reposo, y otra señal capturada justo tras la realización de actividad física intensa. La duración de los complejos QRS en ambas señales fue la misma, por lo que se interpretó que el tiempo de despolarización ventricular es el mismo al encontrarse en reposo y durante el ejercicio. Saber esto permite asumir que los complejos QRS a lo largo de una señal de ECG idealmente presentan la misma morfología [39].

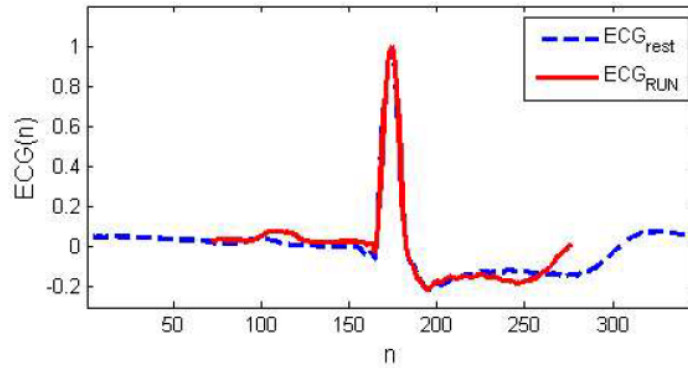


Figura 2.15 Comparación de onda ECG durante estado de reposo (azul) y onda ECG después de realizar actividad física (rojo) [39].

2.8 Entorno de desarrollo MATLAB

MATLAB es un ambiente de cómputo numérico que es desarrollado por la empresa MathWorks. Este software permite la manipulación de matrices, graficar funciones y datos, implementar algoritmos, crear interfaces de usuario, entre otros [40].

Además, MATLAB incluye un simulador llamado “Simulink” que permite simular sistemas digitales por medio de bloques de funcionamiento, conteniendo una amplia librería de bloques de salida, entrada, y de operación.

A lo largo de sus versiones, se han implementado diversas aplicaciones y herramientas que resultan de utilidad en muchos campos de estudio. Una de las herramientas incluidas en MATLAB es la aplicación “Filter Designer”, la cual permite el diseño de un filtro digital de acuerdo con las necesidades del usuario [41].

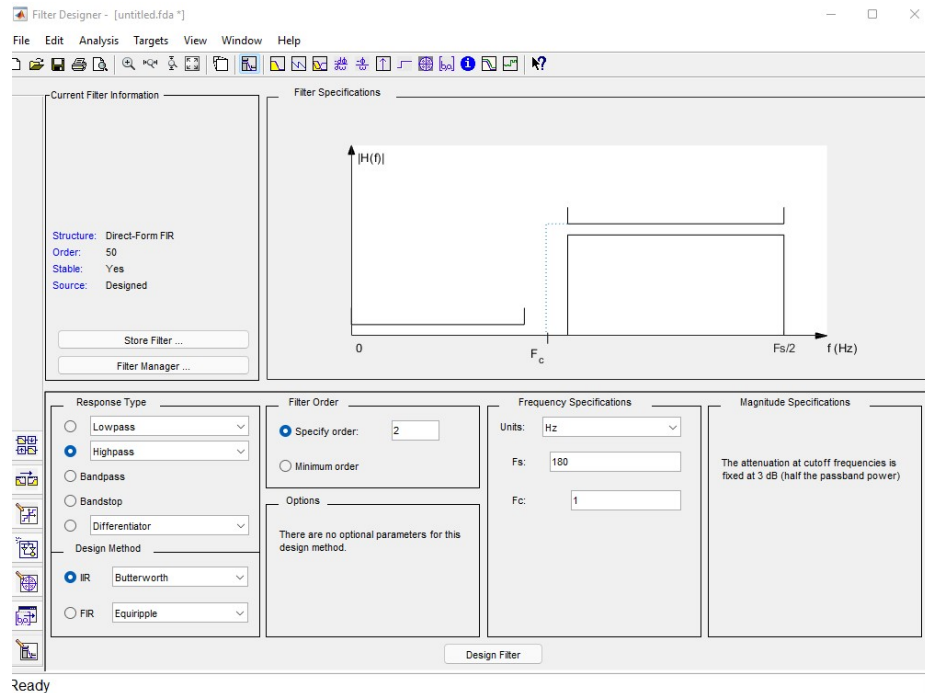


Figura 2.16 Interfaz gráfica del usuario de aplicación “Filter Designer” de MATLAB [41].

El diseño de filtros digitales con esta aplicación facilita la implementación de estos y reduce el tiempo necesario para su diseño [41].

2.9 Convertidor analógico-digital (ADC) Delta-Sigma

Una de las tecnologías que llamó la atención en la electrónica a principios del siglo fue la modulación delta-sigma para convertir señales analógicas en digitales.

La modulación delta-sigma permite crear ADCs de elevada resolución sin utilizar grandes cantidades de componentes electrónicos, pero no son buenos trabajando a elevadas tasas de muestreo [42].

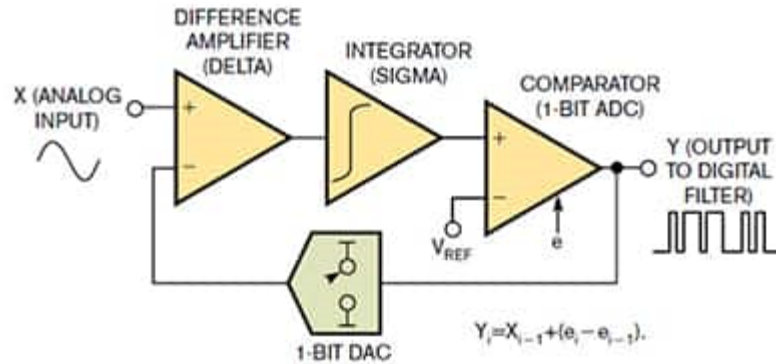
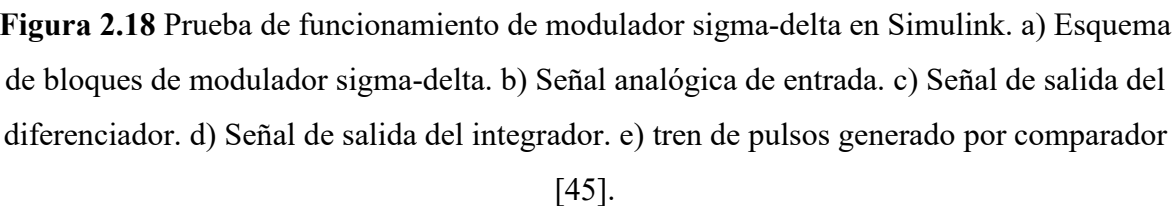


Figura 2.17 Diagrama de funcionamiento de modulador delta-sigma [42].

El modulador delta-sigma se compone de un amplificador diferencial, el cual transmite su señal de salida a un integrador para que este a su salida envíe su señal a un comparador que diferencia a una muy alta frecuencia el valor de salida del integrador contra un valor de voltaje de referencia. A la salida se produce un tren de pulsos de distinta anchura que al traducirse por medio de un convertidor digital-analógico (DAC) de 1 bit permite retroalimentar al modulador con un valor analógico teóricamente igual al de la señal de entrada en la iteración anterior, permitiendo calcular la diferencia entre el valor actual de la señal de entrada y su valor en la iteración anterior y compararle tras su integración con el voltaje de referencia nuevamente y así continuar con la generación del tren de pulsos [43], [44].

Debido a que se requiere cierta longitud del tren de pulsos, muestrear a una tasa muy elevada hará que la longitud del tren de pulsos no sea suficiente para que el DAC realice una conversión precisa, provocando así un mayor ruido de cuantificación [43].

Teóricamente, al promediar un número infinito de pulsos se obtendría el valor exacto de la señal de entrada tras ser convertido, provocando un error de cuantificación de cero.



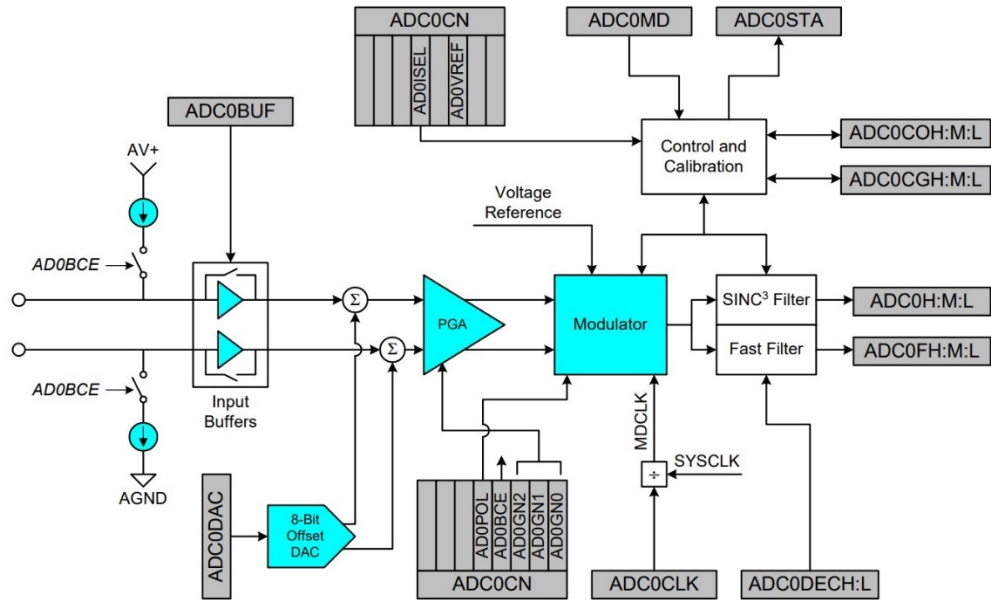


Figura 2.19 Diagrama interno de bloques de módulo de convertidor ADC de 24 bits de microcontrolador C8051F350 [46].

El ADC, además de realizar conversiones de 0 a $2^{24}-1$ es capaz de realizar mediciones en modo diferencial utilizando dos terminales analógicas como entrada, utilizando 1 bit para indicar el signo y 23 bits para el valor numérico de -2^{23} a $2^{23}-1$.

El fabricante indica que el ADC es capaz de realizar mediciones a una frecuencia de muestreo máxima de 1000 Hz. La tabla 1 enlista las principales características del módulo ADC [46].

Tabla 2.1. Principales características del módulo ADC del microcontrolador C8051F350 [46].

Frecuencia máxima de muestreo	1000 Hz
Resolución	24 bits
Tipo de ADC	Delta-Sigma
Tipo de entrada analógica	diferencial
Memoria digital	8kb de Flash y 768 bytes de RAM

El fabricante produce también tableros de desarrollo con el microcontrolador C8051F350 el cual contiene terminales y botones necesarios para el funcionamiento de los módulos del microcontrolador. Este tablero se alimenta con un conector que suministre 9 voltios de tensión, lo cual permite que una batería de 9 voltios pueda alimentarlo, además de la fuente de alimentación incluida que permite alimentarlo desde un tomacorriente [47], [48].

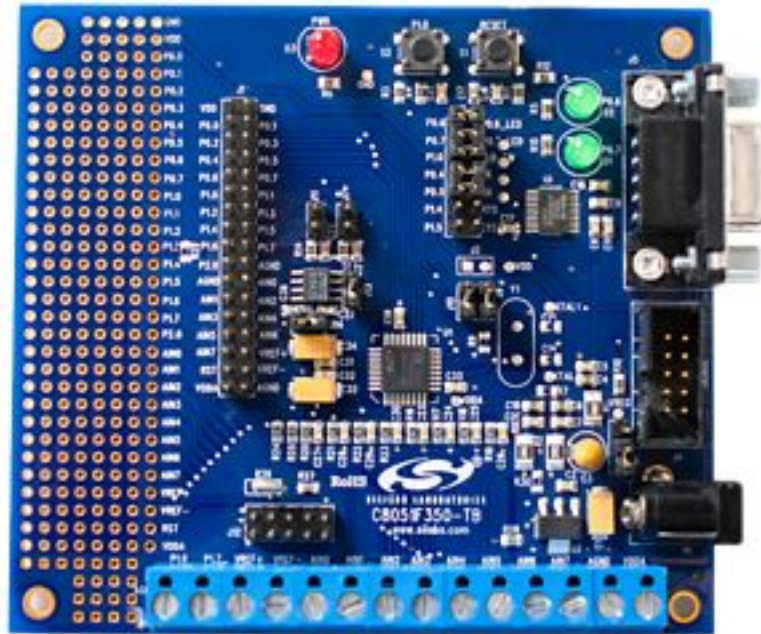


Figura 2.20 Tablero de desarrollo con microcontrolador C8051F350 fabricado por la compañía Silicon Labs [47].

2.10.2 Ambiente de programación y desarrollo Simplicity Studio

Simplicity Studio es un ambiente de desarrollo basado en Eclipse y en C/C++ diseñado para trabajar con todas las tecnologías que la empresa Silicon Labs produce. Es capaz de reconocer todos los dispositivos de la empresa, incluidos dispositivos que trabajan con la tecnología “Internet of Things” (IoT).

El ambiente se basa en Eclipse y en C/C++. El lenguaje de programación utilizado para programar dentro del ambiente de programación es C y se utiliza el compilador Keil 8051 [49].

Cuenta con programas ejemplos que permiten probar y utilizar sus módulos como el módulo ADC correctamente configurado, permitiendo modificar bits de configuración para ajustar la configuración a las necesidades del usuario [49], [50].

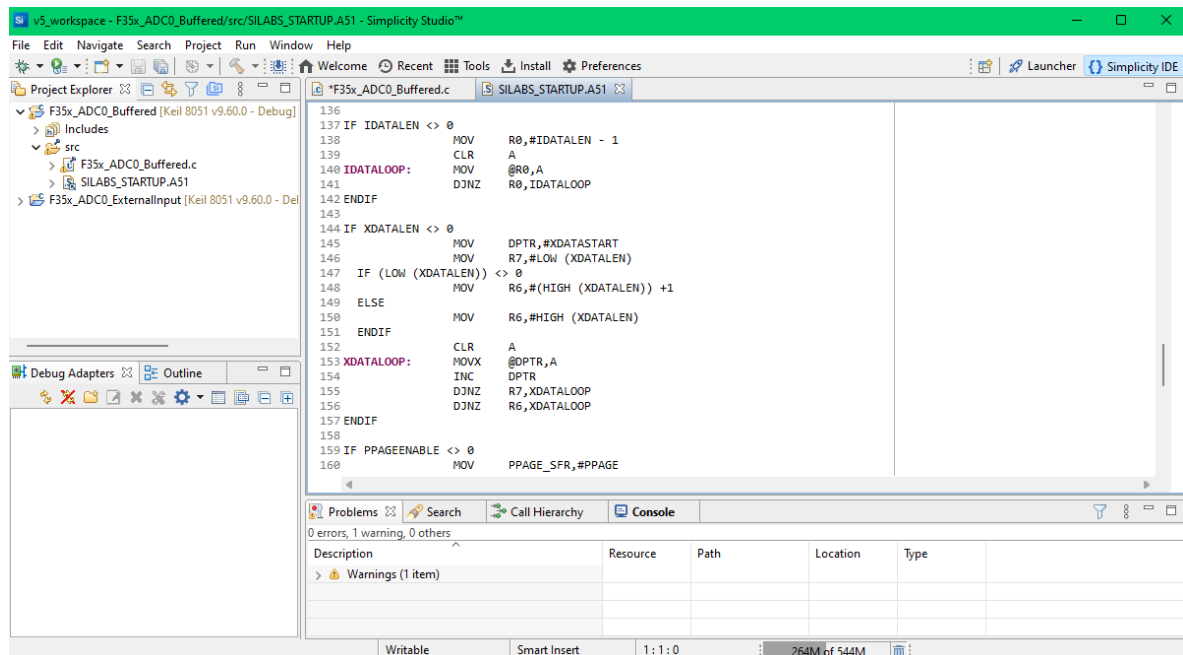


Figura 2.21 Interfaz de usuario de ambiente de programación Simplicity Studio con programa ejemplo para utilización del módulo ADC de 24 bits.

El ambiente de programación Simplicity Studio es utilizado en el desarrollo de esta investigación.

CAPÍTULO 3. METODOLOGÍA

3.1 Caracterización de ADC de 24 bits

Para realizar la caracterización del módulo ADC de 24 bits del microcontrolador C8051F350 se realizaron distintas pruebas.

3.1.1 Medición del ruido en modo común

Se utilizó el ejemplo incluido en el ambiente de programación Simplicity Studio 5 llamado F35x_ADC0_Buffered.

Para realizar la medición del ruido en modo común, se realizó un barrido conectando el canal 1 del microcontrolador y la terminal de referencia a cero, utilizando la alimentación de 3.3V como valor de referencia máximo, y la terminal de referencia como valor de referencia mínimo.

El barrido se realizó utilizando distintas frecuencias de muestro desde 10 Hz hasta 1000 Hz.

Se realizaron 25 series de 128 lecturas del valor del ADC y se promediaron los valores de desviación estándar y valor promedio en unidades digitales de cada serie.

Las frecuencias utilizadas para realizar el barrido se muestran en la tabla 3.1.

Tabla 3.1. Frecuencias utilizadas para realizar barrido para caracterización del ADC.

10 Hz
20 Hz
40 Hz
60 Hz
80 Hz
120 Hz
360 Hz
500 Hz
1000 Hz

Se observó un aumento significativo en el ruido en modo común a partir de 360 Hz, como en la figura 3.1 se observa.

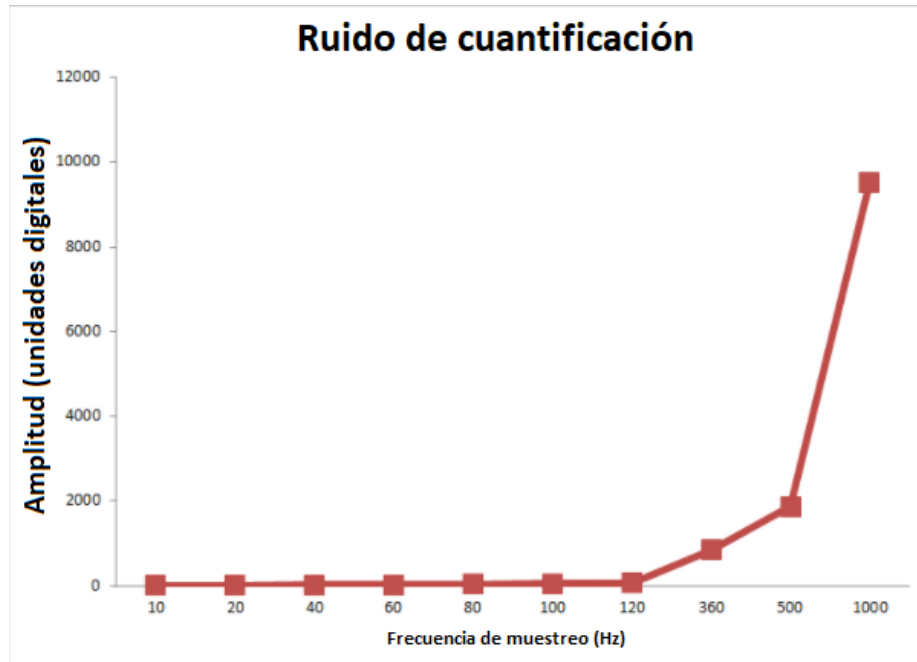


Figura 3.1 Resultado de barrido de frecuencias para medición de ruido de cuantificación en modo común de ADC.

3.1.2 Pruebas de captura de ECG y selección de frecuencia de muestreo

Se realizaron distintas lecturas de una señal de ECG del paciente A a 120 Hz y a 360 Hz. Observando la morfología de la señal, se determinó una frecuencia de muestreo a utilizar para realizar lecturas de ECG.

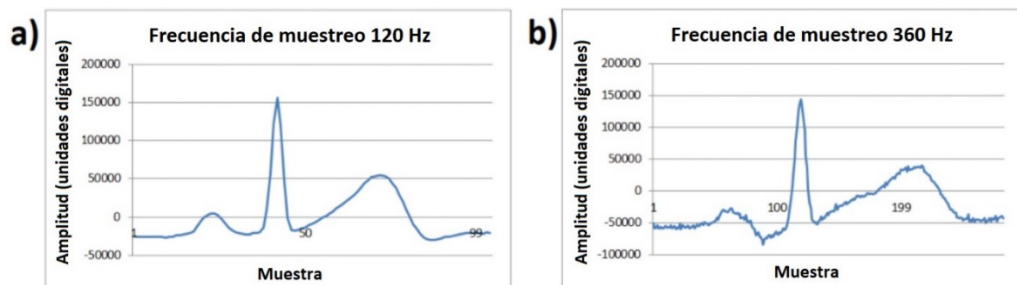


Figura 3.2 Señales de ECG muestreadas con ADC de microcontrolador C8051F350. a) Señal muestreada con ganancia 100 a 120 Hz. b) Señal muestreada con ganancia 100 a 360 Hz [3].

La frecuencia seleccionada fue 180 Hz.

3.2 Modificación de tablero Sparkfun AD8232

El tablero Sparkfun AD8232 fue modificado en ganancia y se habilitó la terminal REFOUT. También se deshabilitó el LED utilizado para indicar pulsos cardíacos abriendo el circuito.

3.2.1 Modificación de ganancia

El arreglo de resistencias del tablero Sparkfun AD8232 da al circuito una ganancia de 1100 respecto al valor de voltaje en la entrada.

Debido a que la aplicación del dispositivo es para la realización de lecturas mientras el paciente se encuentra en movimiento, la presencia de artefactos de movimiento, cuya amplitud es mayor que la señal ECG ciertos casos, provoca que el sistema alcance la saturación de voltaje y se pierda información de la señal. Para prevenir esto, se modificó la ganancia del tablero Sparkfun AD8232 de 1100 a 100, aprovechando la capacidad del microcontrolador de realizar lecturas de alta resolución (24 bits). De esta manera, el artefacto de movimiento no es capaz de elevar el voltaje hasta el punto de saturación.

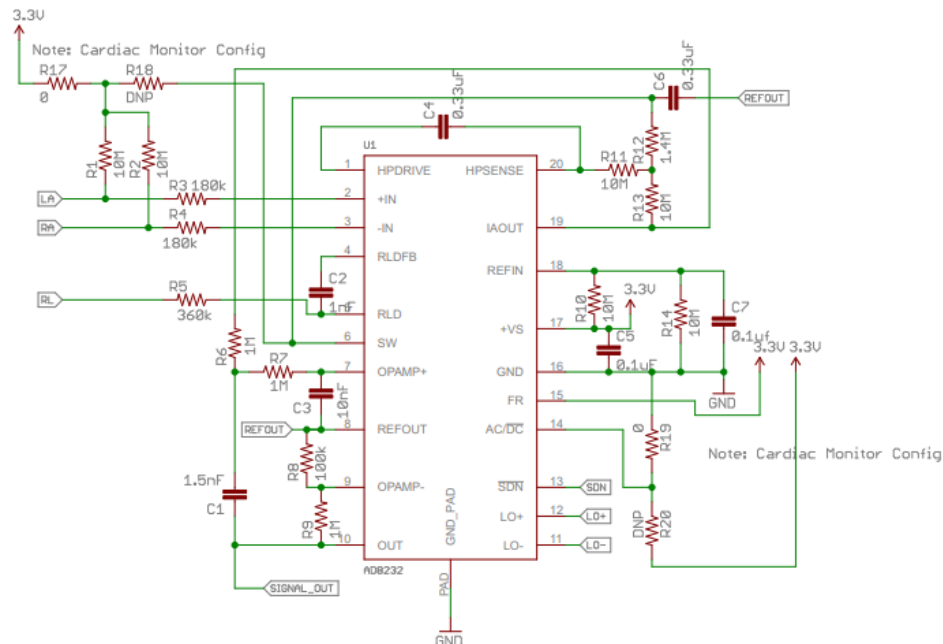


Figura 3.3 Diagrama electrónico de tablero Sparkfun AD8232 con configuración "cardiac monitor" [29].

Debido a que la ganancia fija del circuito integrado AD8232 es de 100, la fórmula para calcular la ganancia total del circuito es la siguiente [27]:

$$G = (100) \left(1 + \frac{R9}{R8} \right) \quad (3.1)$$

El arreglo de resistencias del tablero proporciona una ganancia de 11, la cual multiplica a la ganancia base del circuito integrado, dando como resultado 1100. Para volver 100 la ganancia total, fue necesario volver 0 el valor de la resistencia R9, por lo que se realizó un corto circuito entre ambos extremos de la resistencia.

A continuación, se muestra la señal capturada utilizando la ganancia de 1100 saturando el sistema.



Figura 3.4 Señal de ECG muestreada a 180 Hz con ganancia 1100. Los límites superior e inferior de saturación se muestran con franjas de color rojo [3].

3.2.2 Habilidad de terminal REFOUT

La terminal REFOUT del circuito integrado AD8232 no se encuentra conectada a una terminal del tablero fabricado por Sparkfun. Debido a que la medición sería realizada utilizando el ADC en modo diferencial, fue necesario utilizar la terminal REFOUT del circuito integrado AD8232 para realizar la medición de voltaje entre OUT y REFOUT, y no entre OUT y GND.

Para habilitar la terminal REFOUT, se conectó un cable en el respectivo nodo del tablero para conectar la terminal REFOUT al canal analógico del ADC del microcontrolador.

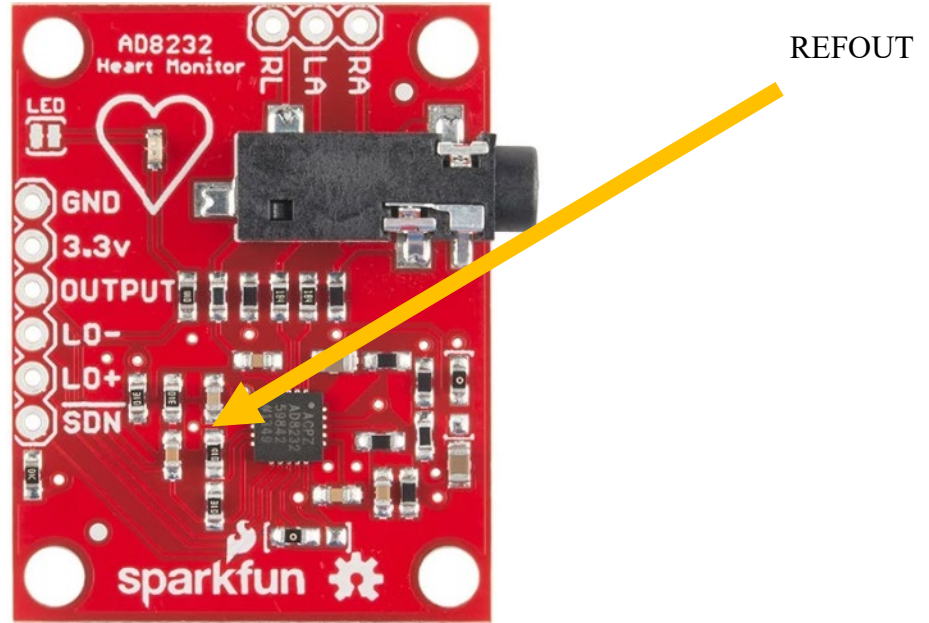


Figura 3.5 Localización de terminal REFOUT indicada en tablero Sparkfun AD8232.

3.3 Módulo Bluetooth HC-10 y configuración del puerto UART

Para la transmisión de datos de forma inalámbrica, se implementó el módulo HC-10. El módulo fue conectado al puerto UART en las terminales P.04 y P.05 del microcontrolador, permitiendo transmitir los valores de lectura del ADC de forma serial hacia un monitor serial de una computadora o teléfono inteligente [51].

El puerto UART fue configurado como en la tabla 3.2 se muestra.

Tabla 3.2 Configuración de puerto UART de microcontrolador C8051F350 para transmisión serial de datos.

Tasa de baud	115200
Bits de datos	8
Bits de paridad	0
Bits de paro	1

Para la recepción de datos en una computadora, se utilizó el monitor serial de la plataforma Arduino UNO.

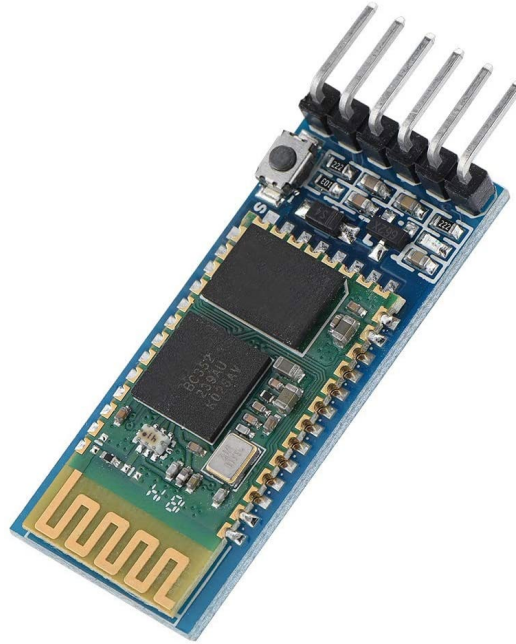


Figura 3.6 Módulo bluetooth HC-10 [51].

3.4 Integración de dispositivo portátil

Se integró un dispositivo portátil conteniendo los siguientes módulos: Tablero Sparkfun AD8232 modificado, el tablero de pruebas con el microcontrolador C8051F350, y el módulo bluetooth HC-10.

Como sensores, se utilizan electrodos de cloruro de plata, los cuales no forman parte del dispositivo. El monitor tampoco forma parte del dispositivo integrado.

Se muestra en la figura 3.7 el esquema de bloques del dispositivo integrado.

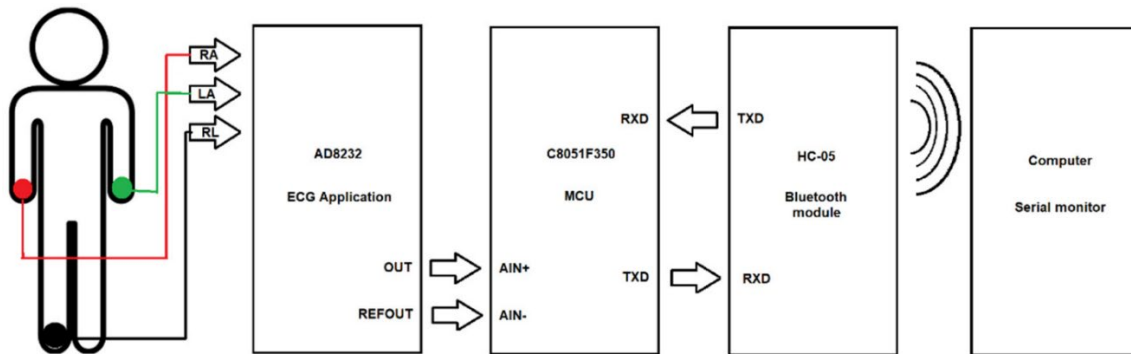


Figura 3.7 Esquema de bloques de etapas del dispositivo portátil integrado [1].

3.5 Captura de señal ECG

3.5.1 Características del paciente

Las características del paciente se muestran en la tabla 3.3.

Tabla 3.3 Características del paciente de cual se adquirió la señal de ECG

Sexo	Masculino
Edad	23 años
Peso	61 kg
Estatura	177 cm
Eje eléctrico cardiaco	120 grados

Se realizó la medición conectando al paciente en estado de reposo al dispositivo utilizando la configuración de la derivación DII, colocando el electrodo positivo en el brazo derecho, el electrodo negativo en la pierna derecha, y el electrodo de referencia en el brazo izquierdo.



Figura 3.8 Paciente en estado de reposo durante adquisición de señal ECG.

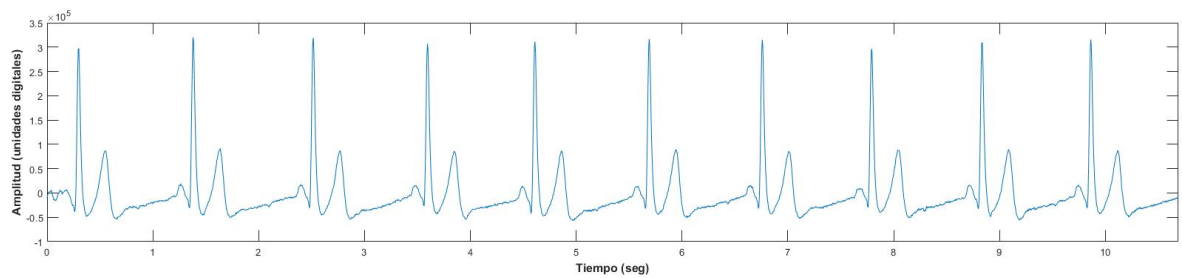


Figura 3.9 Señal de ECG adquirida de paciente en reposo a 180 Hz y ganancia 100×.

Posteriormente se realizó la medición de una señal de ECG del mismo paciente, pero realizando movimiento repentino rápido estando de pie, el cual consistió en pasar de posición de firme a posición de marcha y volver a posición de firme [1].



a)

b)

Figura 3.10 Paciente durante adquisición de señal en posición firme y movimiento repentino. a) Paciente en posición firme. b) Paciente tras realizar movimiento repentino colocándose en posición de marcha.

Este movimiento se realizó en repetidas ocasiones mientras la señal era capturada. El resultado de la captura se muestra en la siguiente figura, y fue llamada “señal original”.

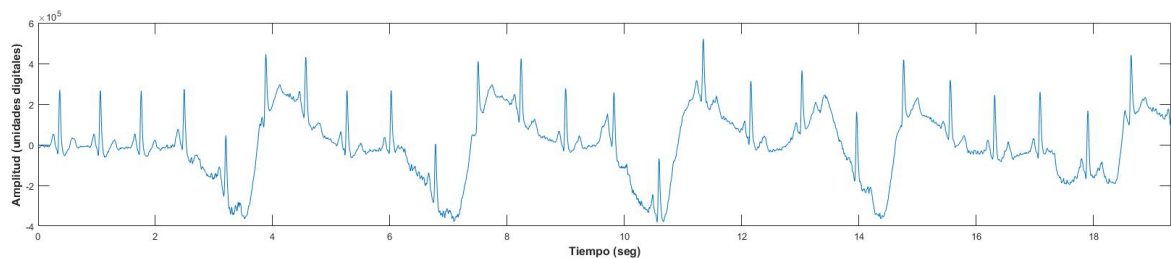
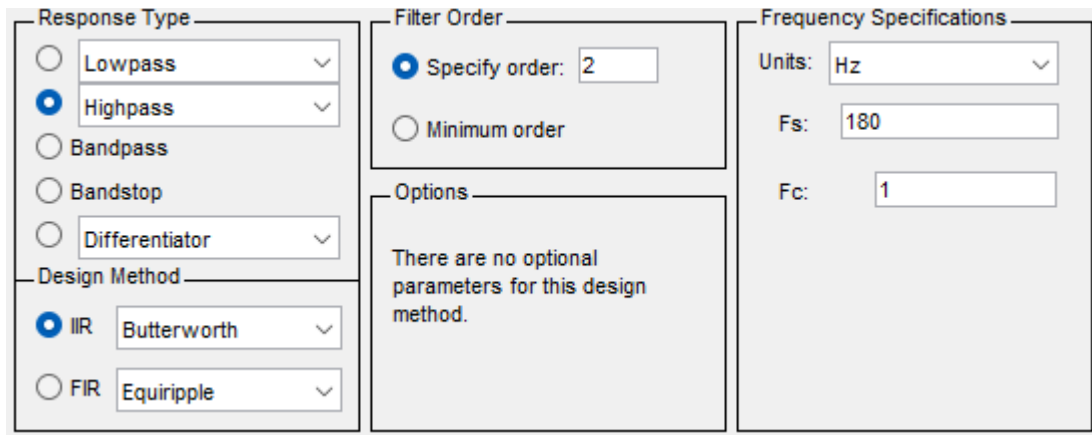


Figura 3.11 Señal original adquirida.

3.6 Reducción de DLB y artefactos de movimiento

3.6.1 Filtro digital

Mediante la herramienta “Filter designer” del software MATLAB, se diseñó un filtro de Butterworth de segundo orden pasa-altas con frecuencia de corte de 1 Hz.



The screenshot shows the MATLAB Filter Designer interface with the following settings:

- Response Type:** Highpass (selected), Lowpass, Bandpass, Bandstop, Differentiator.
- Filter Order:** Specify order: 2 (selected), Minimum order.
- Frequency Specifications:** Units: Hz, Fs: 180, Fc: 1.
- Design Method:** IIR Butterworth (selected), FIR Equiripple.
- Options:** There are no optional parameters for this design method.

Figura 3.12 Parámetros especificados en interfaz gráfica de herramienta “Filter designer” para diseño de filtro digital pasa-altas [41].

La respuesta en frecuencia del filtro diseñado se observa en la figura 3.13.

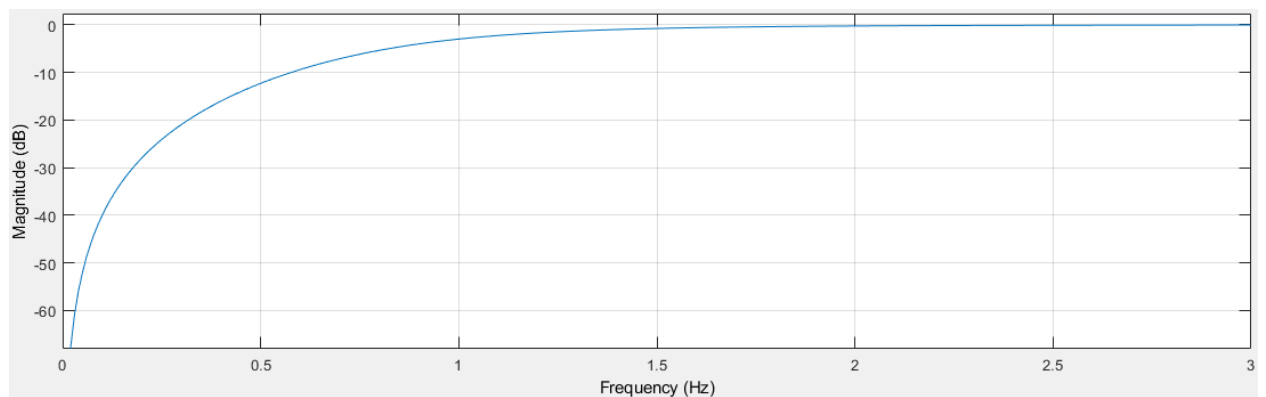


Figura 3.13 Respuesta en frecuencia de filtro pasa-altas digital generado con herramienta “Filter designer” (figura 3.12) visualizado con la función fvtool de MATLAB.

Se obtuvo, como resultado del diseño del filtro, la siguiente ecuación de diferencias:

$$y[n] = x[n] - 2x[n - 1] + 1.91y[n - 1] + x[n - 2] - 0.95y[n - 2] \quad (3.2)$$

Utilizando la función “filter” de MATLAB, se filtró la señal original.

3.6.2 Interpolación spline

Para la atenuación de la DLB, se propuso también crear un vector que replique la tendencia de la DLB en la señal original. Para esto, se realizó una interpolación de tipo spline entre distintos puntos seleccionados de forma arbitraria de la señal original, en donde el valor de la señal debería de ser cero.

Los puntos seleccionados de la señal original se muestran en la figura 3.14.

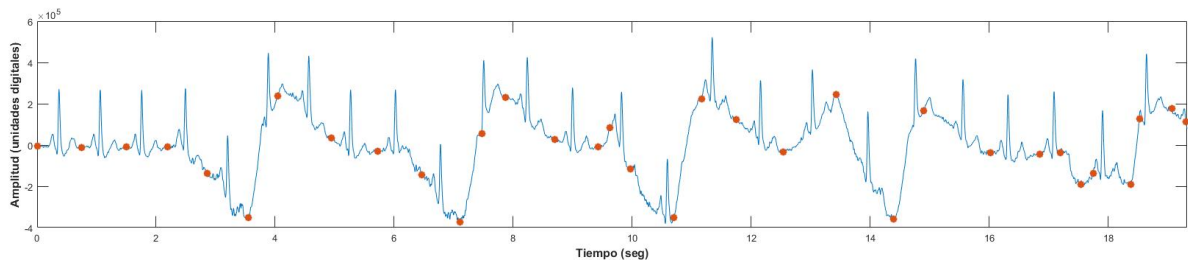


Figura 3.14 Puntos de señal original seleccionados para interpolar por interpolación spline.

Las muestras de la señal original seleccionadas para realizar la interpolación se muestran en la tabla 3.4.

Tabla 3.4 Números de muestra de puntos de señal original seleccionados para interpolar por interpolación spline.

1
143
286
417
544
675
769
939
1088

1228
1351
1420
1495
1652
1790
1828
1893
2032
2121
2231
2381
2551
2734
2829
3044
3200
3266
3333
3371
3491
3519
3623
3668

Utilizando la función “spline” de Matlab, se generó un vector, cuya longitud igualaba a la de la señal original, y tenía una tendencia similar a la de la DLB [52].

En la siguiente figura se puede comparar visualmente la señal original y el vector generado por la función “spline”.

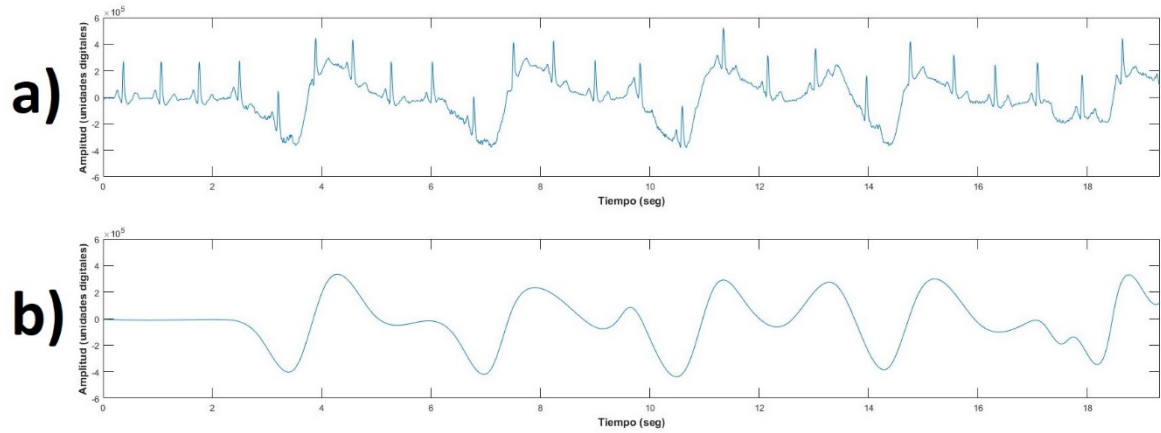


Figura 3.15 a) Señal original. b) Señal generada tras interpolar puntos seleccionados de señal original.

La atenuación de la DLB se realizó al restar de la señal original el vector generado mediante la función “spline” de MATLAB.

3.6.3 Generación de señal esperada

De la señal original, se seleccionó el primer ciclo de ECG que fue capturado mientras el paciente permanecía en estado de reposo, el cual representa un fragmento de la señal sin contener artefactos de movimiento, así como no existe una DLB.

El ciclo de ECG seleccionado fue utilizado como plantilla para generar una señal que sería llamada “señal esperada”. La plantilla fue seleccionada de forma arbitraria.

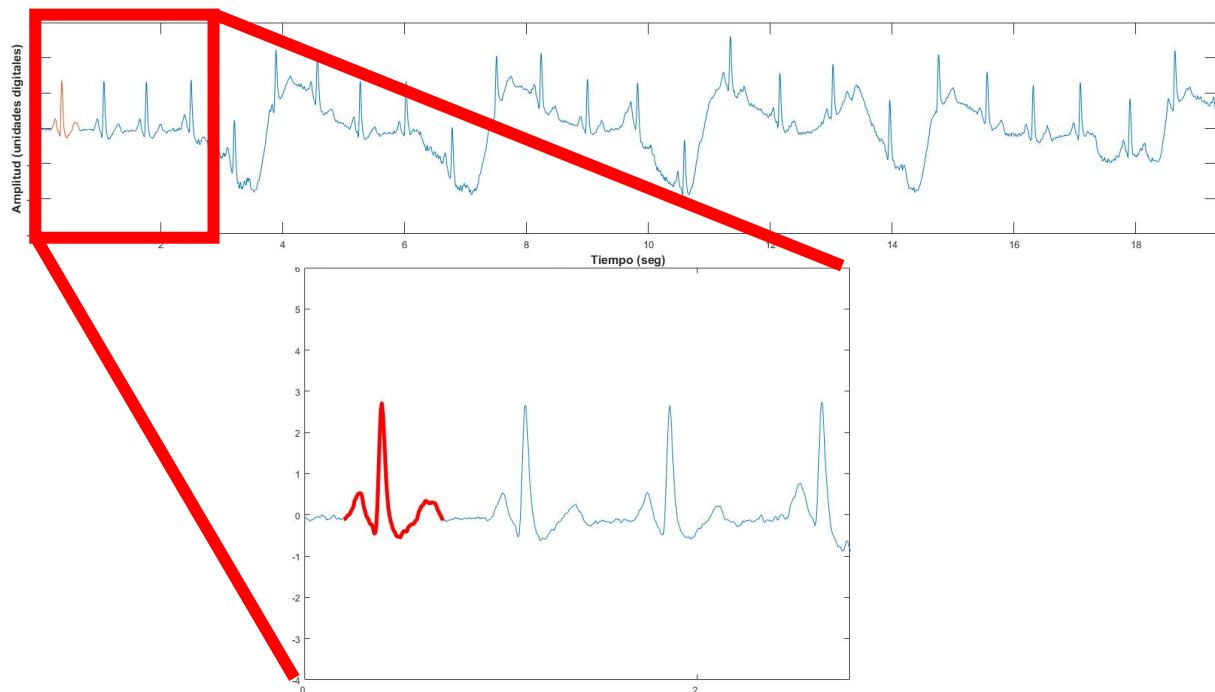


Figura 3.16 Selección de primera onda de ECG presentada en la señal original en estado de reposo para utilizar como plantilla.

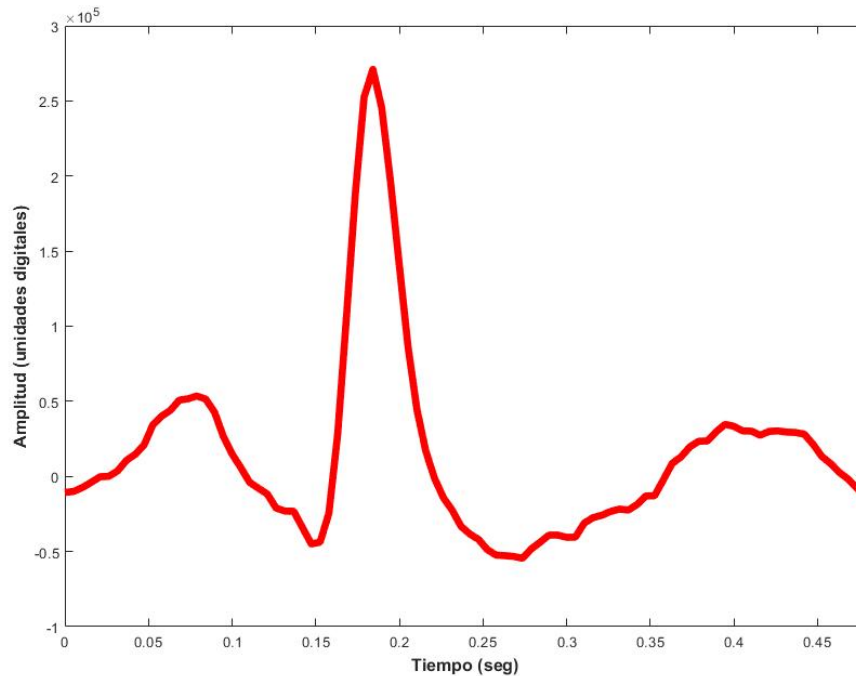


Figura 3.17 Plantilla de ECG en reposo seleccionada.

Asumiendo que todos los ciclos de ECG de la señal esperada tendrían una morfología idéntica, se creó el vector llamado “señal deseada”, el cual es un tren de plantillas colocadas en las mismas posiciones que cada ciclo ECG se encuentra en la señal original, centrados en los picos R de la señal, es decir, el pico R de la plantilla en el tren de plantilla fue colocado en la misma posición que el pico R del ciclo de ECG correspondiente en la señal original.

Las posiciones de los picos R a lo largo de la señal fueron seleccionadas de forma manual a lo largo de la señal.

El tren de plantillas llamado “señal deseada” resultó como se muestra en la figura 3.18.

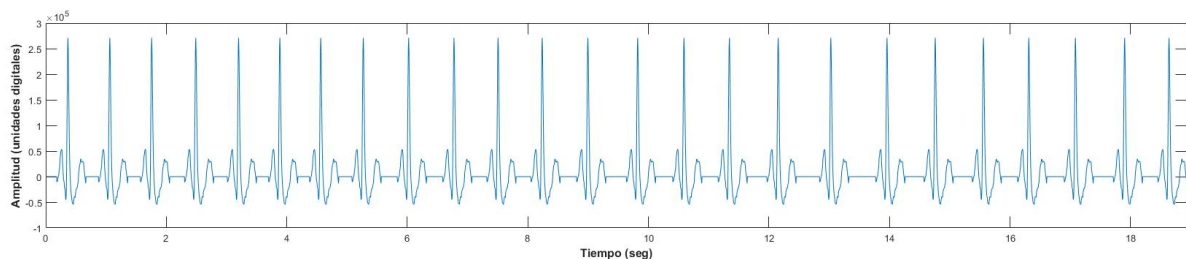


Figura 3.18 Tren de plantillas generado.

El tren de plantillas fue sustraído de la señal original para obtener un nuevo vector llamado “artefacto de movimiento y DLB”. Dicho vector representa la contaminación por ruido en la señal original asumiendo que la señal “limpia” es idéntica al tren de plantillas generado.

El resultado de sustraer el tren de plantillas de la señal original se muestra en la siguiente figura.

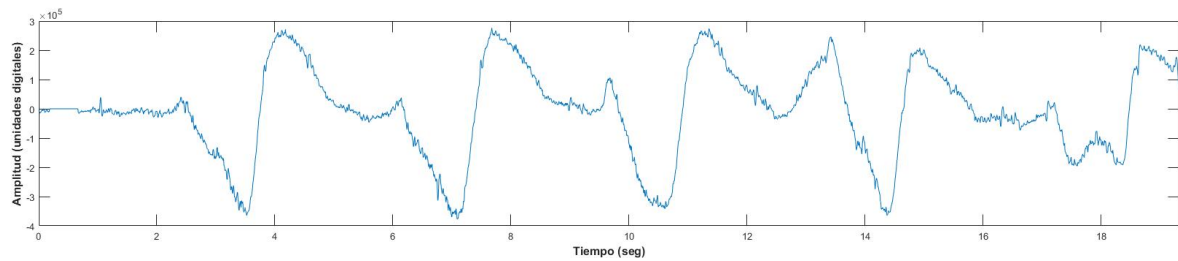


Figura 3.19 Señal resultante de sustraer tren de plantillas (figura 3.18) a señal original (figura 3.11).

Para obtener un vector que replique únicamente la tendencia de la DLB en la señal original, se diseñó un filtro pasa-bajas para atenuar el componente de alta frecuencia de la señal.

El filtro fue diseñado haciendo uso de la aplicación “Filter Designer” de MATLAB. Las características del filtro fueron: orden 2, tipo: Butterworth, frecuencia de corte: 1 Hz.

Response Type	Filter Order	Frequency Specifications
☒ Lowpass ☐ Highpass ☐ Bandpass ☐ Bandstop ☐ Differentiator	☒ Specify order: 2 ☐ Minimum order	Units: Hz Fs: 180 Fc: 1
Design Method: ☒ IIR: Butterworth ☐ FIR: Equiripple	Options: There are no optional parameters for this design method.	

Figura 3.20 Parámetros especificados en interfaz gráfica de herramienta “Filter designer” para diseño de filtro digital pasa-bajas.

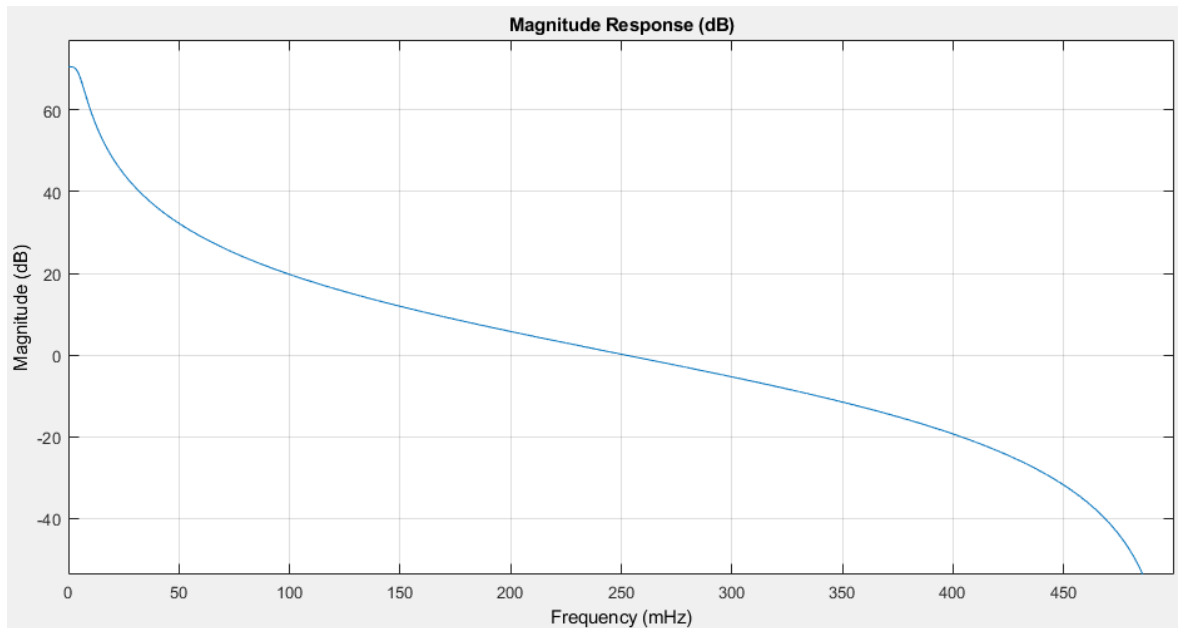


Figura 3.21 Respuesta en frecuencia de filtro pasa-bajas digital generado con herramienta “Filter designer” (figura 3.20) visualizado con la función fvtool de MATLAB.

Se aplicó el filtro y se obtuvo la señal que replica a la DLB en la señal original.

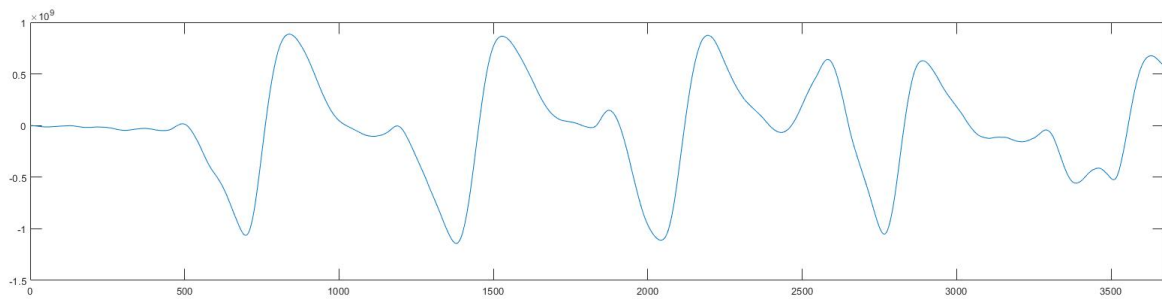


Figura 3.22 Línea de tendencia de DLB generada al filtrar la señal obtenida tras sustraer el tren de plantillas de la señal original (figura 3.19).

La señal obtenida tras aplicar el filtro fue desplazada 47 muestras para compensar el desfaseamiento provocado por el filtro.

Se ajustó la amplitud de la señal filtrada. Para esto, se calculó el cociente entre el valor máximo de la señal de entrada y la señal de salida del filtro, y se multiplicó por el vector de la señal de salida del filtro.

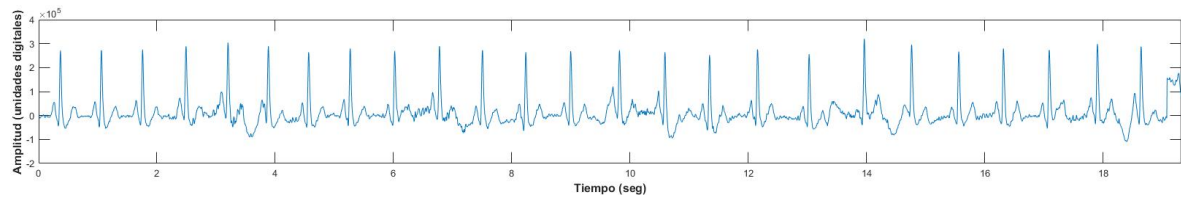


Figura 3.23 Señal generada tras sustraer la línea de tendencia de DLB de la señal original.

Utilizando transformada de Fourier en la señal obtenida con la función `fft` de MATLAB, se revisó el contenido en frecuencia de la señal y se determinó que existía un componente de baja amplitud a 58.73 Hz provocada por la interferencia del campo magnético generado por la toma eléctrica de la instalación.

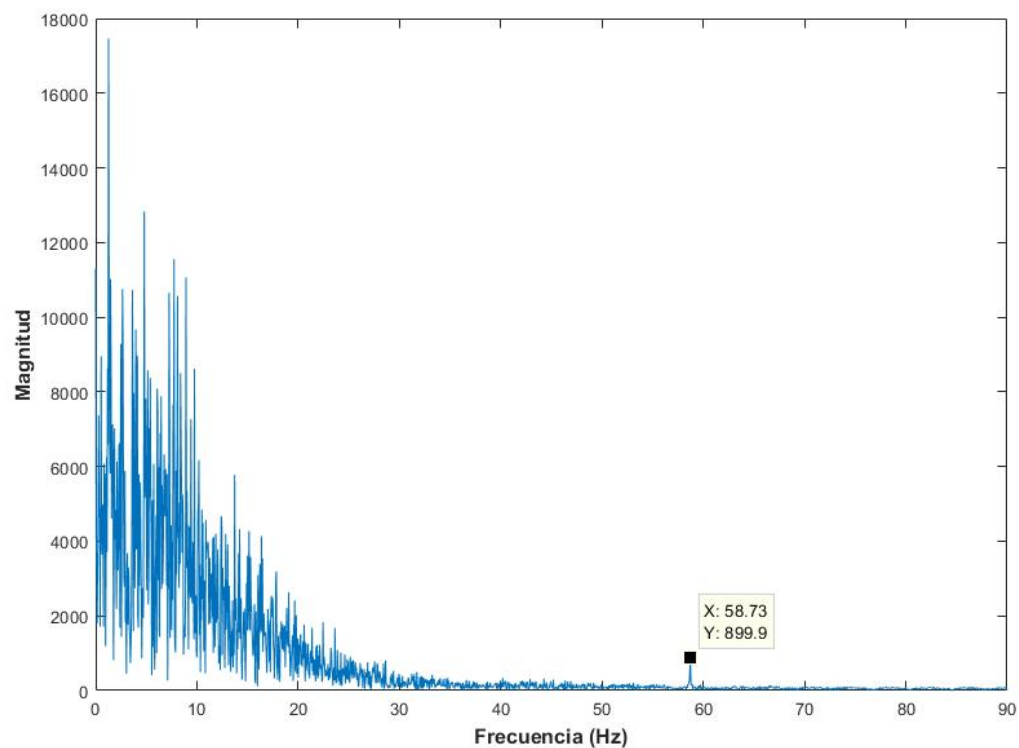


Figura 3.24 Espectro de frecuencias de señal generada al sustraer DLB de señal original (figura 3.23).

Para atenuar dicho componente de la señal, se diseñó un filtro notch con la herramienta “Filter Designer” de MATLAB.

The screenshot displays the MATLAB Filter Designer interface, which is divided into three main sections: Response Type, Design Method, and Frequency/Magnitude Specifications.

- Response Type:** This section contains four radio buttons and four dropdown menus. The radio buttons are for Lowpass, Highpass, Bandpass, and Bandstop. The dropdown menus are for Notching, Single Notch, and Equiripple. The 'Notching' radio button is selected.
- Design Method:** This section contains two radio buttons and two dropdown menus. The radio buttons are for IIR and FIR. The dropdown menus are for Single Notch and Equiripple. The 'IIR' radio button is selected.
- Frequency Specifications:** This section contains a Units dropdown menu (set to Hz), a Fs input field (set to 180), a Fnotch input field (set to 59), and two radio buttons for Bandwidth and Q. The 'Bandwidth' radio button is selected, and its value is set to 5.
- Magnitude Specifications:** This section contains a Units dropdown menu (set to dB) and an Apass input field (set to 1).

Figura 3.25 Parámetros especificados en interfaz gráfica de herramienta “Filter designer” para diseño de filtro digital notch para atenuar componente de frecuencia de 59 Hz.

El filtro fue aplicado utilizando la función “filter” de MATLAB posteriormente se ajustó la amplitud de la señal de acuerdo con la amplitud de la señal de entrada del filtro.

3.7 Comparación de métodos de atenuación de ruido

Para comparar la efectividad de los tres métodos probados de atenuación de ruido, se calculó el valor de desviación estándar (σ) y valor cuadrático medio (RMS) de cada señal para comparar la atenuación del mismo valor respecto al de la señal original.

EL cálculo de RMS y σ de cada señal fue realizado con las funciones “rms” y “std” de MATLAB respectivamente.

La comparación se realizó calculando el cociente entre el RMS y σ de cada señal entre el respectivo valor de la señal original. La atenuación fue calculada en decibeles (dB).

3.8 Generación de señal sintética de ECG por medio de generador de funciones

Se generó una señal de ECG sintética haciendo uso del software “ArbExpress Application” de la compañía Tektronix, así como el generador de funciones modelo AFG3021B de la misma compañía.

La forma de onda se creó manualmente dentro del software mencionado para generar un archivo tipo TFW, el cuál mediante un dispositivo de almacenamiento externo puede ser utilizado por el generador de funciones.

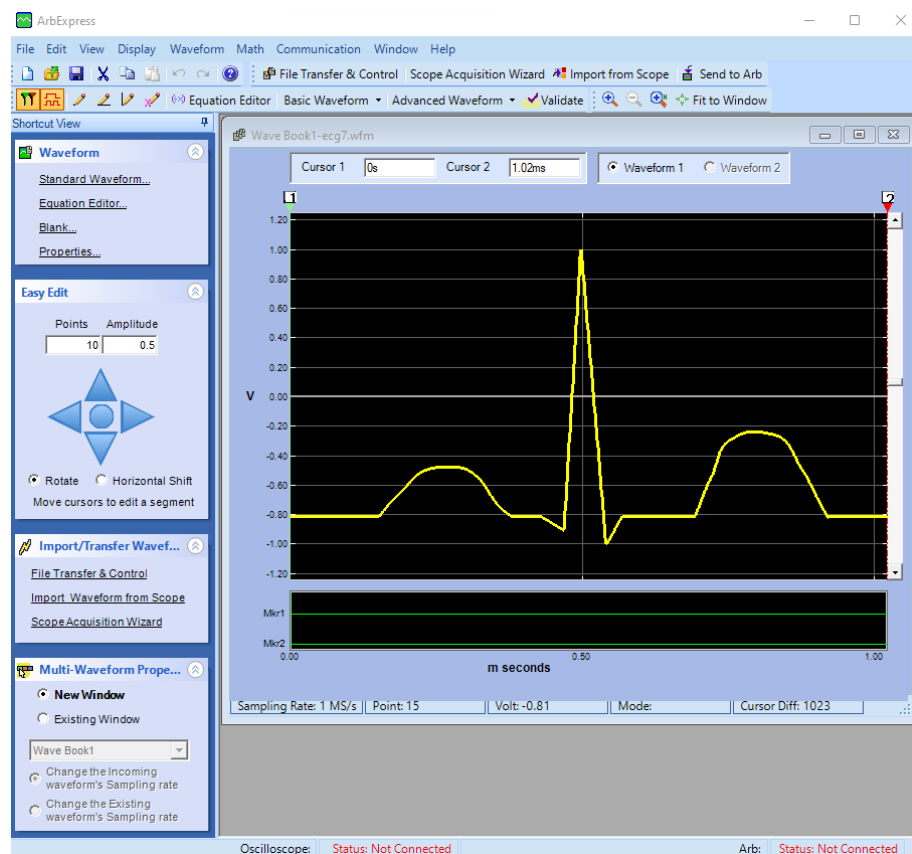


Figura 3.26 Onda ECG diseñada en software ArbExpress para generar en generador de funciones de ondas arbitrarias [53].

El archivo generado fue almacenado en una memoria USB y abierto en el generador de funciones utilizando la función “Arbitrary Function”.

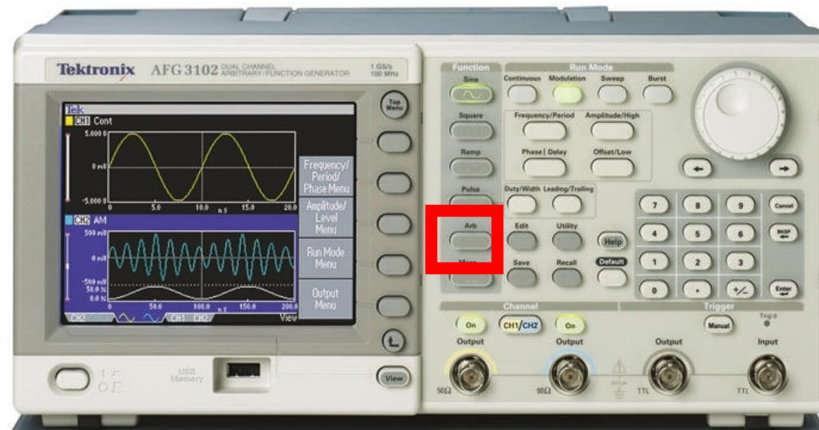


Figura 3.27 Generador de funciones Tektronix AFG3021B con habilidad de generar funciones arbitrarias.

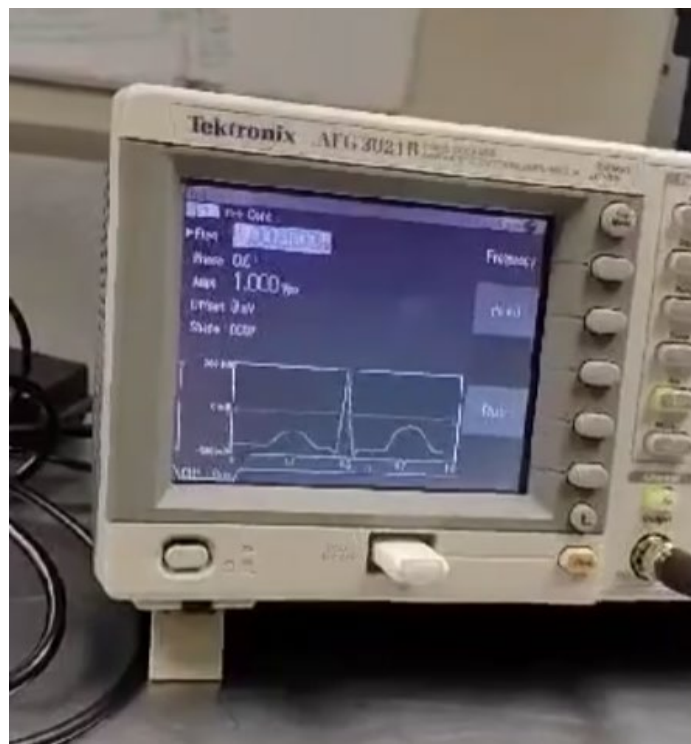


Figura 3.28 Generador de funciones Tektronix AFG3021B generando onda ECG diseñada.

La señal generada por el generador de funciones fue monitoreada con ayuda de un osciloscopio DS1102D de la compañía RIGOL.

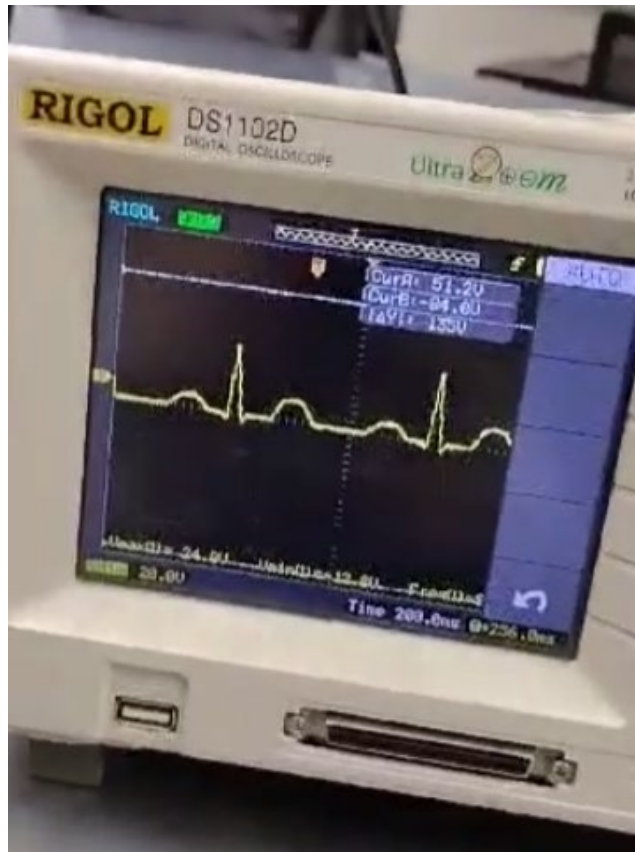


Figura 3.29 Osciloscopio RIGOL SD1102D realizando monitoreo de onda ECG generada por generador de funciones.

La señal fue capturada con el ADC del microcontrolador C8051F350 mediante transmisión serial con el módulo bluetooth HC-10.

Se agregó a la señal capturada un componente senoidal para replicar el efecto de la DLB en la señal original capturada de un paciente. El componente senoidal que fue agregado a la señal se creó con la función “chirp” de MATLAB que permite generar una señal cosenoidal en un barrido de frecuencias. Se generó la señal cosenoidal con frecuencias desde 0.5 Hz hasta 1.2 Hz. Además, se estableció una fase de inicio de -90 grados, para que la señal inicie con valor cero como una señal senoidal.

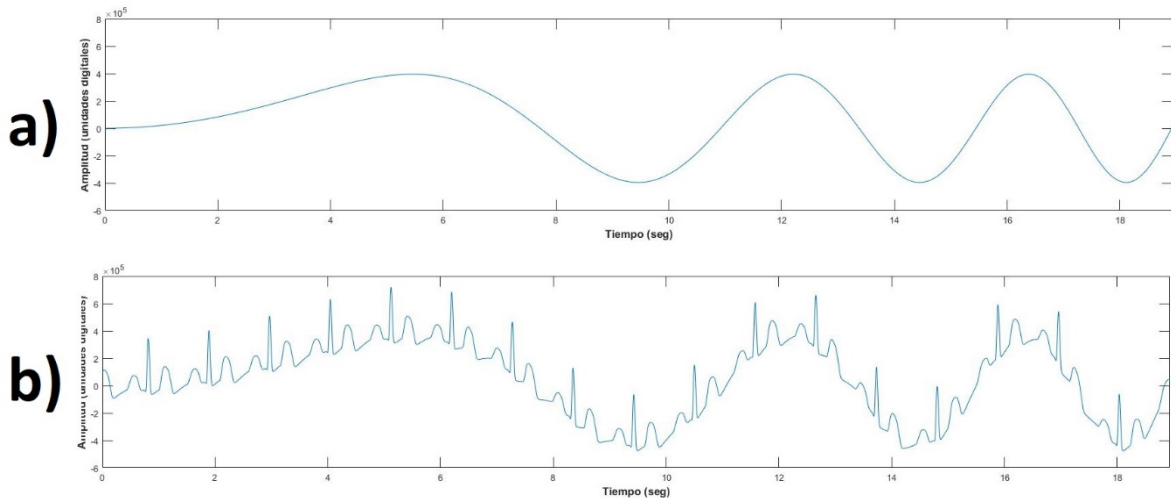


Figura 3.30 a) Señal generada por función chirp. b) Señal generada tras añadir la señal generada por generador de funciones a la señal generada por función chirp.

3.9 Generación de señal sintética con tren de plantillas

Se generó una señal sintética sin ser capturada por el ADC del microcontrolador, pues se utilizó la plantilla seleccionada de la señal original (ver sección 3.6.3).

Al tren de plantillas generado al colocar el punto R de la plantilla en la misma posición que cada punto R de la señal original se le agregó un componente senoidal para simular una DLB. La señal senoidal se generó mediante la función “chirp” de MATLAB. El rango del barrido de frecuencias de la señal senoidal generada fue de 0.5 Hz a 1.2 Hz, con una fase de -90 grados, y fue agregada al tren de plantillas.

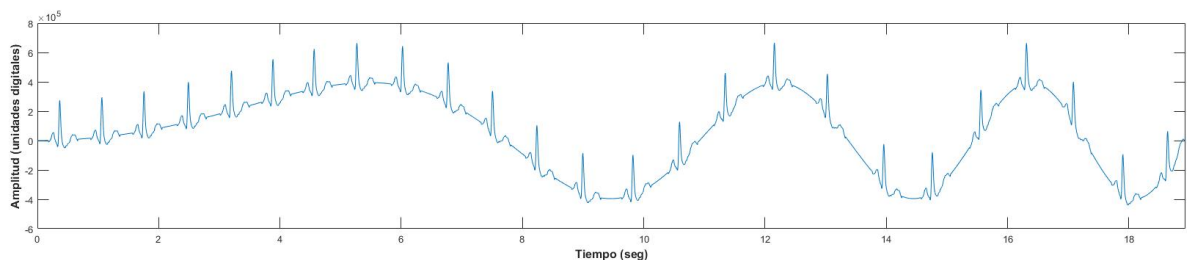


Figura 3.31 Señal generada tras añadir la señal creada por la función chirp (figura 3.30a) al tren de plantillas (figura 3.18).

3.10 Diseño de detector de complejos QRS

Para la automatización en la colocación de plantillas en las posiciones donde un punto R se encuentre en la señal original, fue necesario detectar automáticamente la ubicación de cada punto R a lo largo de una señal ECG.

Se diseñó un detector de complejos QRS que detectara los picos R de una señal ECG, para esto, se creó un algoritmo que compara los valores de muestras consecutivas de una señal ECG y calcula la diferencia entre ambos, detectando así cambios rápidos de valor a lo largo de la señal, lo cual caracteriza a la fase de despolarización de los ventrículos del corazón y produce el complejo QRS.

Para la detección de cambios rápidos de valor en las muestras comparadas, se definió un valor de umbral para

El algoritmo se describe a continuación:

1. Se introduce la señal.
2. Número de muestra inicial = 1.
3. Se compara el valor de la muestra del valor del contador actual y la muestra siguiente.
4. Según la diferencia entre los valores comparados, se realiza una acción.
 - a. Si la diferencia entre ambos valores es igual o mayor al valor de umbral, extrae la muestra actual y las siguientes 15 muestras y determina la posición (número de muestra) en que se encuentra el valor más alto entre las 16 muestras y la almacena en un vector B, y se avanza 15 muestras.
 - b. Si la diferencia entre ambos valores es menor al valor de umbral, avanza a la siguiente muestra.
5. Si el número de muestra actual es menor al número total de muestras en la señal, se regresa al paso 1.
6. Se extrae el vector B.

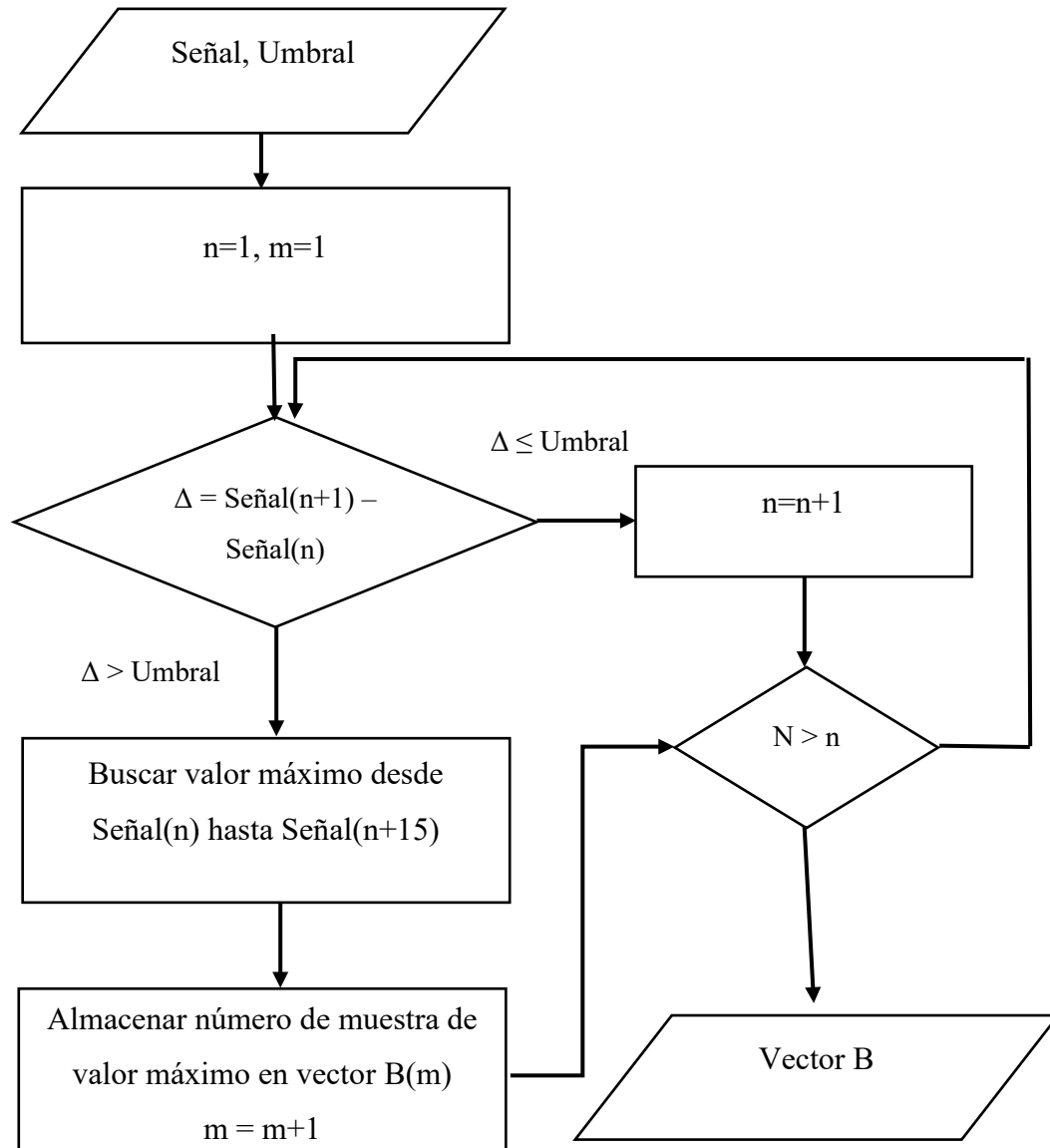


Figura 3.32 Diagrama de flujo de funcionamiento de algoritmo creado para detección de complejos QRS.

El detector se implementó en MATLAB en un script.

El valor de umbral para identificar una rápida variación en el voltaje entre dos muestras consecutivas de la señal fue establecido de manera arbitraria.

3.11 Pruebas de detectores de complejos QRS

Para realizar pruebas de funcionamiento a los detectores de complejos QRS, se utilizaron 3 señales: La señal original capturada del paciente (señal 1), la señal sintética creada en MATLAB (señal 2), y la señal sintética creada con un generador de funciones y MATLAB (señal 3).

Las posiciones en donde se encontraban los picos R de los complejos QRS se determinaron primeramente de forma arbitraria, para así compararlas con las posiciones que cada detector determinó.

Las posiciones de los picos R de cada señal se muestran en la tabla 3.5.

Tabla 3.5 Números de muestra de cada señal en donde se encontraron picos R manualmente.

Señal original	Señal sintética	Señal generada por generador de funciones
71	71	156
203	203	360
335	335	565
474	474	769
609	609	674
739	739	1178
868	868	1383
1002	1002	1587
1145	1145	1792
1288	1288	1996
1427	1427	2201
1566	1566	2405
1710	1710	2609
1867	1867	2814
2013	2013	3019
2156	2156	3223
2310	2310	3428
2476	2476	
2653	2653	
2805	2805	
2957	2957	
3100	3100	
3247	3247	
3402	3402	
3542	3542	

Se utilizaron 3 detectores de complejos QRS para realizar pruebas. Los detectores de complejos QRS utilizados fueron: Algoritmo Pan-Tompkins en MATLAB, algoritmo propio en MATLAB (ver sección 3.10), y algoritmo Simple RTS en MATLAB [54].

3.11.1 Algoritmo propio en MATLAB

Señal 1:

Se introdujo el vector correspondiente a la señal 1 como variable de entrada al algoritmo y se graficó la ubicación del pico R de los complejos detectados.

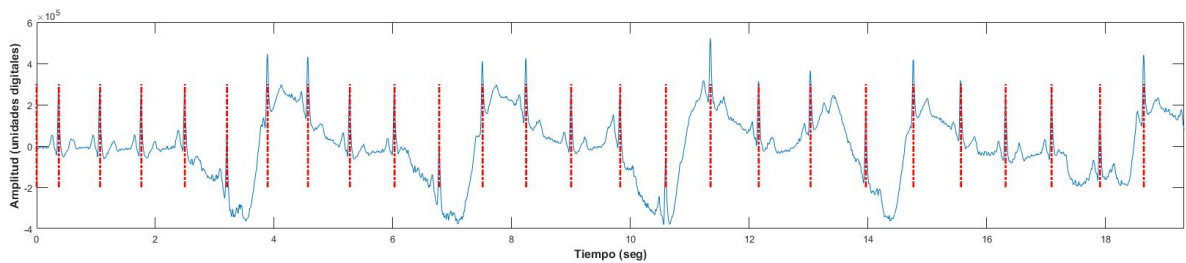


Figura 3.33 Ubicación de picos detectados de la señal 1 por algoritmo propio.

Señal 2:

Se introdujo la señal 2 en la entrada del algoritmo y se graficó la ubicación en que se detectaron los complejos QRS de la señal.

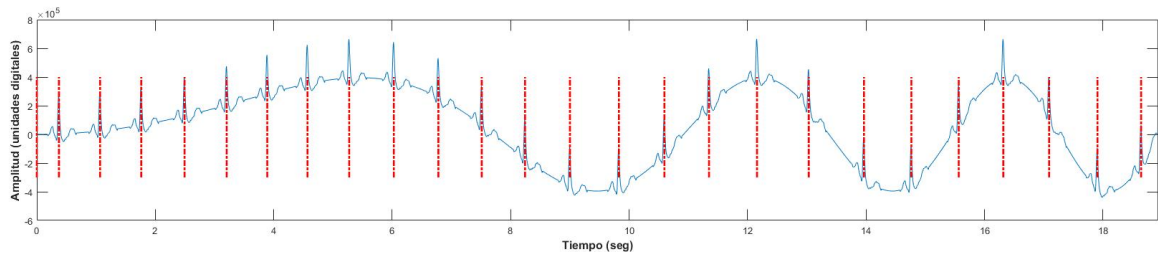


Figura 3.34 Ubicación de picos detectados de la señal 2 por algoritmo propio.

Señal 3:

Se introdujo la señal 3 en la entrada del algoritmo y se graficó la ubicación en que se detectaron los complejos QRS de la señal.

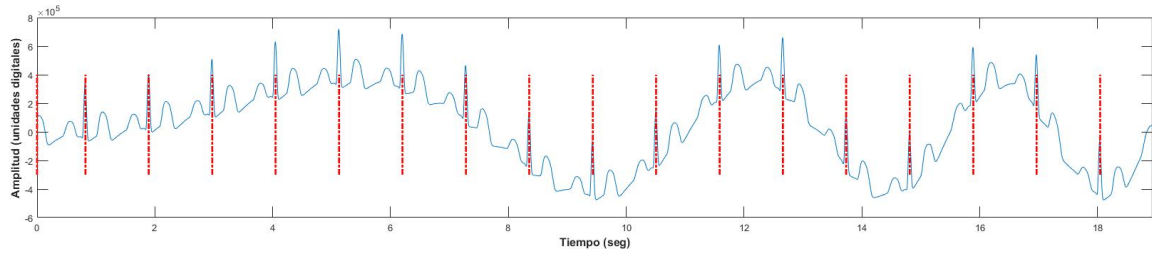


Figura 3.35 Ubicación de picos detectados de la señal 3 por algoritmo propio.

3.11.2 Algoritmo Pan-Tompkins en MATLAB

Señal 1:

Se introdujo la señal 1 como entrada del algoritmo, y se especificó una frecuencia de muestreo de 180 Hz.

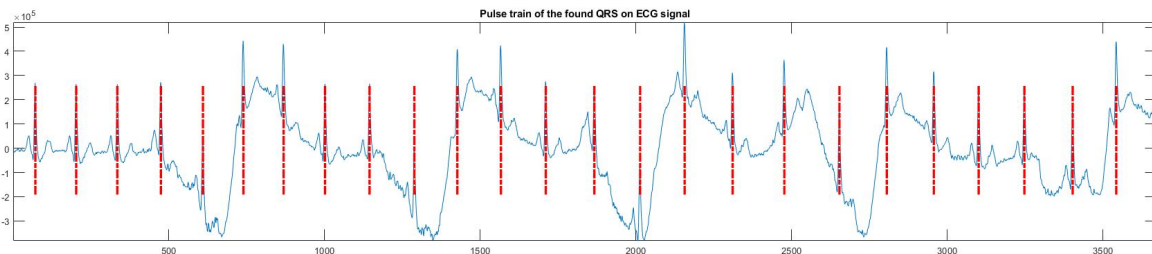


Figura 3.36 Ubicación de picos detectados de la señal 1 por algoritmo Pan-Tompkins.

Señal 2:

La señal 2 se introdujo como entrada del algoritmo especificando una frecuencia de muestreo de 180 Hz.

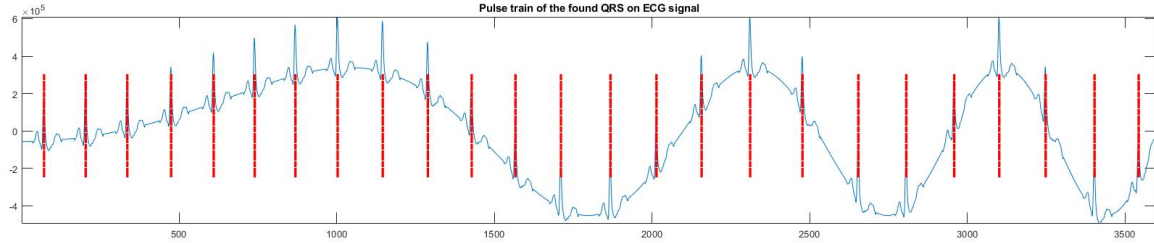


Figura 3.37 Ubicación de picos detectados de la señal 2 por algoritmo Pan-Tompkins.

Señal 3:

La señal 3 se introdujo como entrada del algoritmo especificando una frecuencia de muestreo de 180 Hz.

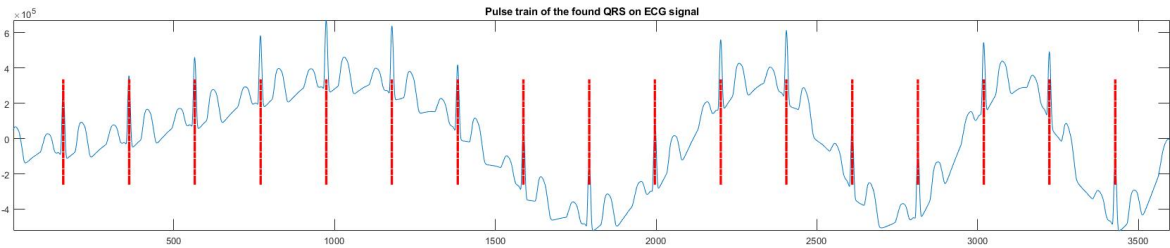


Figura 3.38 Ubicación de picos detectados de la señal 3 por algoritmo Pan-Tompkins.

3.11.3 Algoritmo SimpleRST

Señal 1:

Se introdujo como entrada del algoritmo la señal 1. Se especificó una frecuencia de muestreo de 180 Hz.

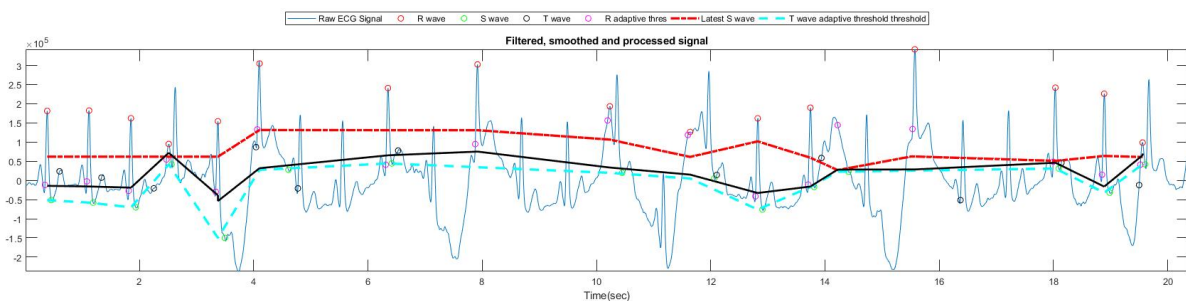


Figura 3.39 Ubicación de picos detectados de la señal 1 por algoritmo SimpleRST.

Señal 2:

Se introdujo la señal 2 como entrada del algoritmo. Se indicó que la frecuencia de muestreo fue de 180 Hz.

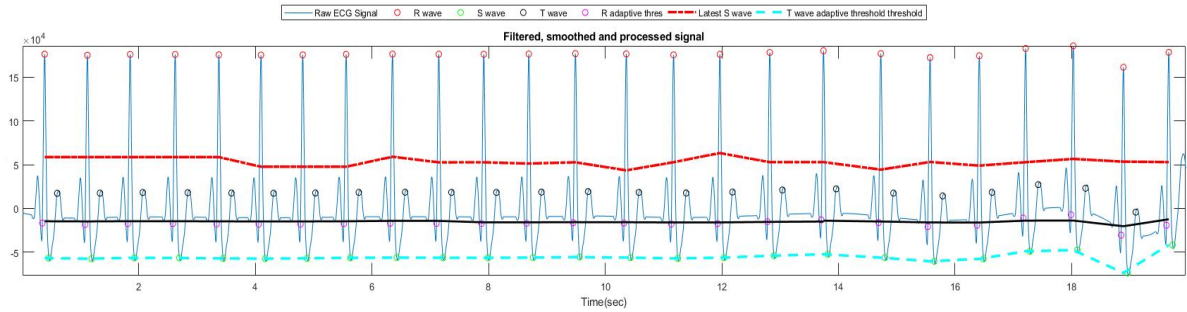


Figura 3.40 Ubicación de picos detectados de la señal 2 por algoritmo SimpleRST.

Señal 3:

Como entrada del algoritmo de introdujo la señal 3, especificando que la frecuencia de muestreo fue de 180 Hz.

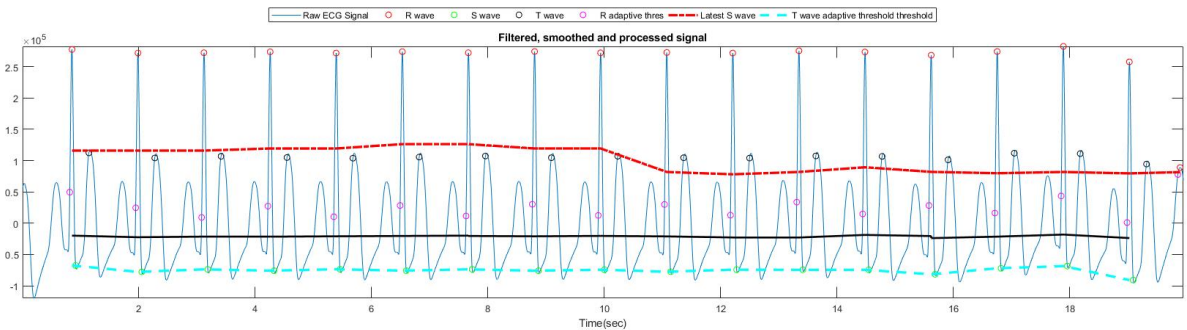


Figura 3.41 Ubicación de picos detectados de la señal 3 por algoritmo SimpleRST.

CAPÍTULO 4. RESULTADOS

4.1 Caracterización de ADC de 24 bits

El barrido de frecuencias realizado para seleccionar la frecuencia óptima de muestreo arrojó los resultados que en la tabla 4.1 se observan.

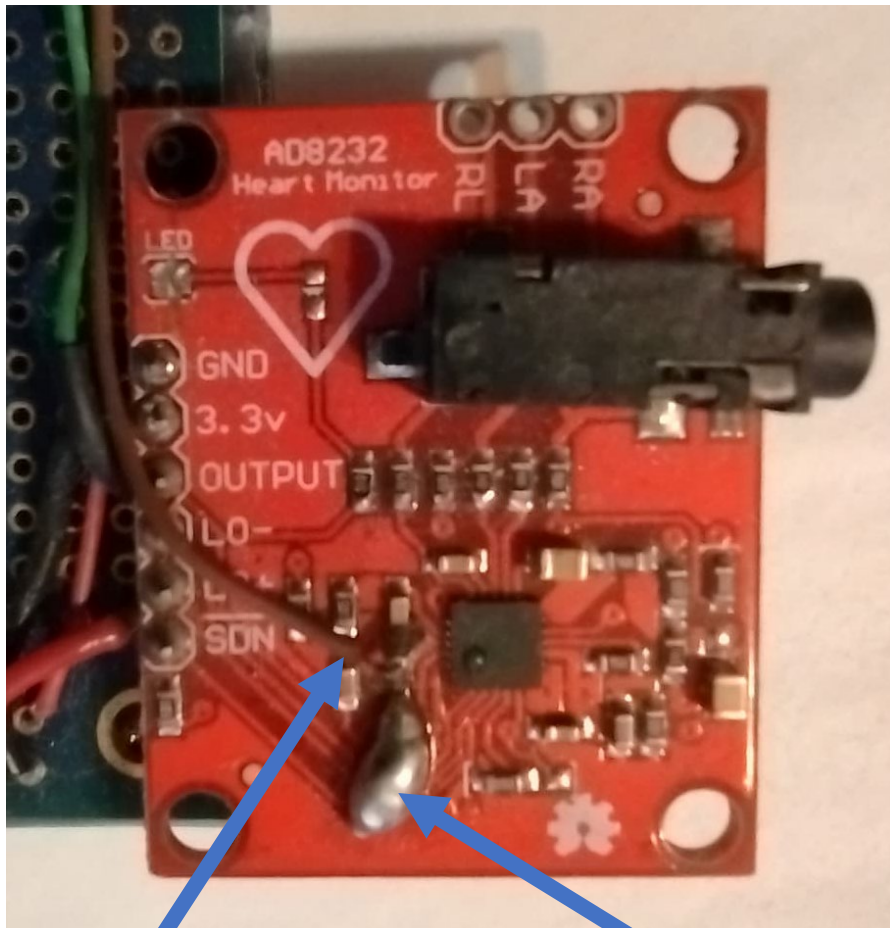
Tabla 4.1 Valores de desviación estándar, SNR, y bits efectivos medidos tras realización de barrido de frecuencias durante caracterización de ADC.

Frecuencia de muestreo (Hz)	Desviación estándar (σ) (unidades digitales)	Relación señal-ruido (SNR)	Bits efectivos
10	10.87	117.78	19.63
60	28.07	109.51	18.25
120	64.55	102.45	17.08
180	149.53	94.97	15.83
360	851.06	79.87	13.31
500	1882.77	72.98	12.16
1000	9527.46	58.89	9.82

4.2 Modificación de tablero Sparkfun AD8232

La modificación del tablero habilitó la terminal REFOUT del circuito integrado AD8232, permitiendo así la medición de voltaje en modo diferencial con el ADC del microcontrolador.

El corto circuito realizado en la resistencia de ganancia volvió 1 el multiplicador de la ganancia interna del circuito integrado AD8232, volviendo 100 la ganancia total del sistema.



REFOUT

Corto Circuito

Figura 4.1 Tablero Sparkfun AD8232 con terminal REFOUT habilitada y ganancia 100.

4.3 Integración de dispositivo portátil

El dispositivo integrado tras unir los componentes que conforman los bloques del esquema presentado en la sección 3.4 constó de una superficie de 100.5 cm^2 y un peso de 77 gramos.

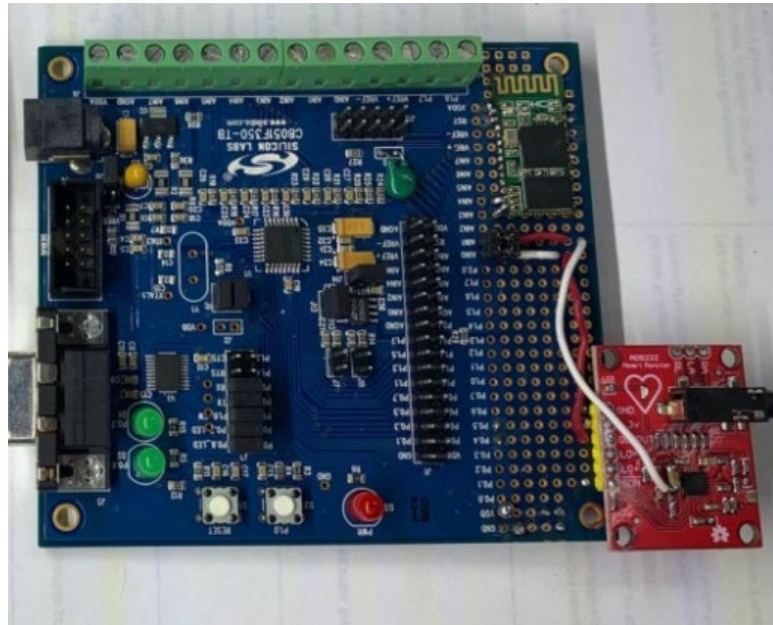


Figura 4.2 Dispositivo portátil integrado para adquisición de ECG.

4.4 Filtrado digital

Después de aplicar el filtro digital diseñado con la aplicación Filter Designer de MATLAB a la señal original se obtuvo una señal cuyo RMS y desviación estándar (σ) disminuyeron con respecto a la señal original.

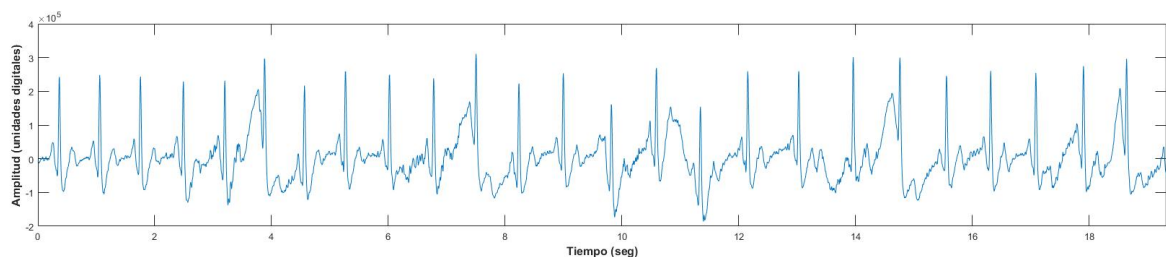


Figura 4.3 Señal generada tras utilizar con la señal original el filtro diseñado.

El valor RMS y desviación estándar (σ) de la señal fueron respectivamente 81722 y 71643 unidades digitales.

La atenuación de los valores RMS y desviación estándar (σ) de la señal resultante respecto a la señal original fueron -2.92 dB y -3.50 dB respectivamente.

4.5 Interpolación spline

Tras restar la señal de tendencia de la DLB generada por medio de interpolación spline de la señal original se obtuvo una señal resultante cuyos valores de RMS y desviación estándar (σ) resultaron menores respecto a los medidos de la señal original.

Los valores medidos de RMS y desviación estándar (σ) para la señal resultante fueron 87154 y 84682 unidades digitales respectivamente.

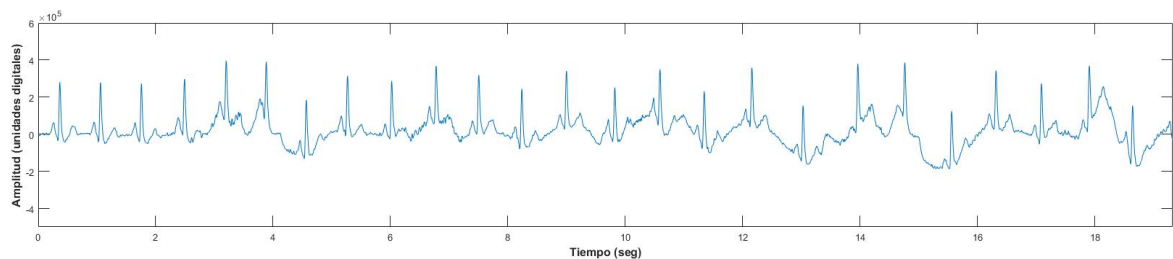


Figura 4.4 Señal generada tras sustraer a la señal original la línea de tendencia generada por interpolación spline.

La atenuación de los valores RMS y desviación estándar (σ) respecto a la señal original fueron respectivamente: -2.64 dB y -2.77 dB.

4.6 Atenuación de DLB con plantilla de ECG en reposo

El resultado de restar la señal de tendencia de la DLB generada utilizando una plantilla de ECG en reposo fue una señal de ECG cuyos valores medidos de RMS y desviación (σ) estándar fueron 59670 y 58629 unidades digitales respectivamente.

La atenuación de los valores de RMS y desviación estándar (σ) de la señal resultante respecto a la señal original fue de -4.29 dB y -4.37 dB respectivamente.

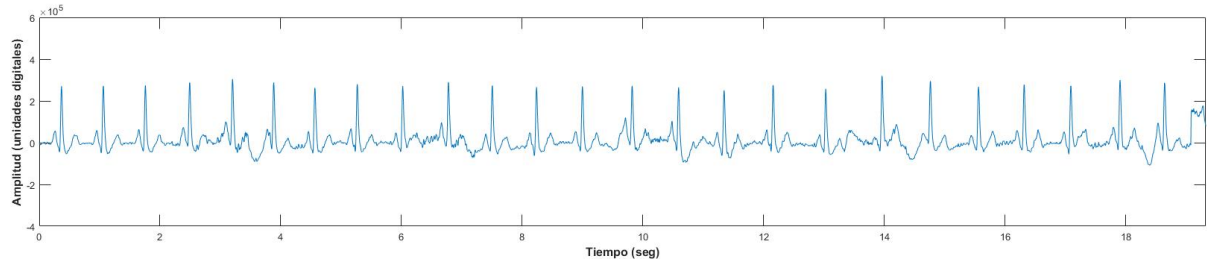


Figura 4.5 Señal generada tras sustraer a la señal original la línea de tendencia generada en sección 3.6.3.

4.7 Comparación de métodos de reducción de ruido

La comparación de la atenuación de RMS y desviación estándar (σ) por cada método se muestra en la tabla 4.2.

Tabla 4.2 Comparación entre cada método descrito en la investigación. Se comparan los valores de RMS y desviación estándar de cada señal procesada respecto a los valores de RMS y desviación estándar de la señal original. Se muestra la atenuación.

Método	RMS	σ	Relación RMS	Relación σ	Atenuación RMS (dB)	Atenuación σ (dB)
Señal original	160230	160238	1.00	1.00	0.00	1.00
Método 1	81722	71643	0.51	0.45	-2.92	-3.50
Método 2	87154	84682	0.54	0.53	-2.64	-2.77
Método 3	59670	58629	0.37	0.37	-4.29	-4.37

4.8 Efectividad de algoritmo propio para detección de picos R de ECG

4.8.1 Señal 1

Al realizar la detección automática de picos R de la señal de ECG por medio del algoritmo propio creado en MATLAB se obtuvieron las posiciones que se muestran en la tabla 4.3, donde son comparadas con las seleccionadas de manera arbitraria.

Tabla 4.3 Comparación de ubicación de picos R detectados en señal 1 por algoritmo propio respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R seleccionados arbitrariamente	Posiciones de picos R detectados por algoritmo
71	71
203	203
335	335
474	474
609	609
739	739
868	868
1002	1002
1145	1145
1288	1288
1427	1427
1566	1566
1710	1710
1867	1867
2013	2013
2156	2156
2310	2310
2476	2476
2653	2653
2805	2805
2957	2957
3100	3100
3247	3247
3402	3402
3542	3542

Se utilizó la función “isequal” de MATLAB para comparar ambos vectores. El valor arrojado por la función al utilizar ambos vectores como entrada fue un 1 lógico, indicando que ambos valores son equivalentes [55].

4.8.2 Señal 2

Debido a que, para generar la señal sintética, se ubicó manualmente el pico R de la plantilla a lo largo de la señal en las mismas posiciones en que los picos R se encontraban en la señal original, no fue necesaria la detección manual de picos para la señal sintética.

La tabla 4.4 muestra la comparación entre vector con las posiciones de los picos R de la señal y el vector con las posiciones de los picos R detectados automáticamente por el algoritmo.

Tabla 4.4 Comparación de ubicación de picos R detectados en señal 2 por algoritmo propio respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R colocados manualmente	Posiciones de picos R detectados por algoritmo
71	71
203	203
335	335
474	474
609	609
739	739
868	868
1002	1002
1145	1145
1288	1288
1427	1427
1566	1566
1710	1710
1867	1867
2013	2013
2156	2156
2310	2310
2476	2476
2653	2653
2805	2805
2957	2957
3100	3100
3247	3247
3402	3402
3542	3542

4.8.3 Señal 3

Se generó manualmente un vector que contenía las posiciones de los picos R a lo largo de la señal seleccionados de forma arbitraria. Se comparó el vector con el vector generado por el algoritmo propio creado para automatizar la selección. La comparación se muestra en la tabla 4.5.

Tabla 4.5 Comparación de ubicación de picos R detectados en señal 3 por algoritmo propio respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R seleccionados arbitrariamente	Posiciones de picos R detectados por algoritmo
156	156
360	360
565	565
769	769
674	674
1178	1178
1383	1383
1587	1587
1792	1792
1996	1996
2201	2201
2405	2405
2609	2609
2814	2814
3019	3019
3223	3223
3428	3428

Haciendo uso de la función “isequal” de Matlab se comprobó que ambos vectores son idénticos.

4.9 Efectividad de algoritmo Pan-Tompkins para detección de picos R de ECG

4.9.1 Señal 1

Tras utilizar el algoritmo Pan-Tompkins implementado en MATLAB se creó un vector con las posiciones de los picos R a lo largo de la señal. La comparación de las posiciones en que se detectaron picos por el algoritmo y las posiciones seleccionadas manualmente se muestra en la tabla 4.6.

Tabla 4.6 Comparación de ubicación de picos R detectados en señal 1 por algoritmo Pan-Tompkins respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R seleccionados arbitrariamente	Posiciones de picos R detectados por algoritmo Pan-Tompkins
71	71
203	203
335	335
474	474
609	609
739	739
868	868
1002	1002
1145	1145
1288	1288
1427	1427
1566	1566
1710	1710
1867	1867
2013	2013
2156	2156
2310	2310
2476	2476
2653	2653
2805	2805
2957	2957
3100	3100
3247	3247
3402	3402
3542	3542

Con la función de MATLAB “isequal” se comprobó la equivalencia de ambos vectores.

4.9.2 Señal 2

Los picos R detectados en la señal sintética por el algoritmo Pan-Tompkins fueron los que se muestran en la tabla 4.7 comparados con las posiciones en donde se colocaron manualmente.

Tabla 4.7 Comparación de ubicación de picos R detectados en señal 2 por algoritmo Pan-Tompkins respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R colocados manualmente	Posiciones de picos R detectados por algoritmo Pan-Tompkins
71	71
203	203
335	335
474	474
609	609
739	739
868	868
1002	1002
1145	1145
1288	1288
1427	1427
1566	1566
1710	1710
1867	1867
2013	2013
2156	2156
2310	2310
2476	2476
2653	2653
2805	2805
2957	2957
3100	3100
3247	3247
3402	3402
3542	3542

4.9.3 Señal 3

Se compararon las posiciones de los picos R detectadas en la señal por el algoritmo, y las seleccionadas manualmente, cuya comparación se muestra en la tabla 4.8.

Tabla 4.8 Comparación de ubicación de picos R detectados en señal 3 por algoritmo Pan-Tompkins respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R seleccionados arbitrariamente	Posiciones de picos R detectados por algoritmo
156	156
360	360
565	565
769	769
674	674
1178	1178
1383	1383
1587	1587
1792	1792
1996	1996
2201	2201
2405	2405
2609	2609
2814	2814
3019	3019
3223	3223
3428	3428

4.10 Efectividad de algoritmo SimpleRST para detección de picos R de ECG

4.10.1 Señal 1

La comparación entre las posiciones en donde se detectaron picos R por el algoritmo y los detectados manualmente se muestran en la tabla 4.9.

Tabla 4.9 Comparación de ubicación de picos R detectados en señal 1 por algoritmo SimpleRST respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R seleccionados arbitrariamente	Posiciones de picos R detectados por algoritmo Simple RST
71	71
203	203
335	335
474	454
609	609
739	740

868	1145
1002	1427
1145	1843
1288	2097
1427	2310
1566	2156
1710	2476
1867	2805
2013	3247
2156	3402
2310	3522
2476	
2653	
2805	
2957	
3100	
3247	
3402	
3542	

Utilizando la función “isequal” de MATLAB se comprobó que ambos vectores no resultaron equivalentes debido a que la función arrojó como resultado un cero lógico.

4.10.2 Señal 2

Tras utilizar la señal como entrada de la función del algoritmo SimpleRST se creó un vector con las posiciones de los picos R detectados a lo largo de la señal. Dicho vector fue comparado con las posiciones en que se colocaron manualmente los picos R en la señal original.

La tabla 4.10 muestra la comparación de los vectores que contenían las posiciones de los picos R.

Tabla 4.10 Comparación de ubicación de picos R detectados en señal 2 por algoritmo SimpleRST respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R colocados manualmente	Posiciones de picos R detectados por algoritmo SimpleRST
71	71
203	203
335	335
474	474
609	609
739	739
868	868
1002	1002
1145	1145
1288	1288
1427	1427
1566	1566
1710	1710
1867	1867
2013	2013
2156	2156
2310	2310
2476	2476
2653	2653
2805	2805
2957	2957
3100	3100
3247	3247
3402	3402
3542	3542

La función “isequal” de MATLAB fue utilizada para comprobar la equivalencia de ambos vectores, arrojando un 1 lógico en su salida.

4.10.3 Señal 3

El vector generado tras detectar los picos R de la señal con la función del algoritmo en MATLAB se comparó con el vector que contenía las posiciones de los picos R de la misma señal seleccionados manualmente. La tabla 4.11 muestra la comparación.

Tabla 4.11 Comparación de ubicación de picos R detectados en señal 3 por algoritmo SimpleRST respecto a la ubicación de picos R detectados manualmente.

Posiciones de picos R seleccionados arbitrariamente	Posiciones de picos R detectados por algoritmo
156	156
360	360
565	565
769	769
674	674
1178	1178
1383	1383
1587	1587
1792	1792
1996	1996
2201	2201
2405	2405
2609	2609
2814	2814
3019	3019
3223	3223
3428	3428

Se comprobó la equivalencia de ambos vectores al obtener un uno lógico en la salida de la función “isequal”.

4.11 Comparación de efectividad de detectores de picos R

La tabla 4.12 muestra cuáles métodos fueron efectivos con cada señal utilizada para realizar las pruebas.

Tabla 4.12 Efectividad de detectores QRS probados con cada señal. Señal 1: Señal original del paciente. Señal 2: Señal sintética generada en MATLAB. Señal 3: Señal generada por generador de funciones.

Señal	Algoritmo propio	Pan-Tompkins	SimpleRST
Señal 1	✓	✓	X
Señal 2	✓	✓	✓
Señal 3	✓	✓	✓

CAPÍTULO 5. DISCUSIÓN DE RESULTADOS

5.1 Caracterización de ADC de 24 bits

A pesar de que el fabricante del microcontrolador C8051F350 especifica que el módulo de ADC es capaz de realizar mediciones a una frecuencia máxima de 1000 Hz, el ruido de cuantificación se incrementa al incrementar la frecuencia de muestreo debido a la arquitectura del convertidor, pues la comparación realizada por el modulador tendría que ser realizada antes de que termine de retroalimentarse, comparando el valor de entrada del ADC con un valor incorrecto, provocando falsos positivos en la comparación. Entre mayor sea la frecuencia de muestreo, mayor será la probabilidad de realizar comparaciones con valores incorrectos de retroalimentación.

El convertidor funciona adecuadamente al realizar conversiones a una baja tasa de muestreo, pero a tasas de muestreo mayores a los 180 Hz, el ruido impide la captura de una señal útil para la aplicación que se le da al convertidor.

Una tasa de muestreo adecuada para la medición de señales biofisiológicas es alcanzable con un convertidor de la arquitectura delta-sigma.

5.2 Modificación de tablero Sparkfun AD8232

El tablero Sparkfun funciona adecuadamente para realizar prototipos de medición de ECG utilizando la configuración de monitor cardíaco recomendada por el fabricante del chip AD8232. Disminuir la ganancia que proporciona el circuito analógico ha permitido la captura de una señal de ECG por un ADC de alta resolución (24 bits) evitando la saturación del sistema por presencia de artefactos de movimiento, sin provocar la pérdida de información de la señal de interés.

Se logró realizar una medición diferencial con el ADC al habilitar la terminal REFOUT del tablero, la cual permite utilizar como referencia un valor de voltaje mayor al valor de referencia de la alimentación del circuito, obteniendo valores positivos y negativos en las lecturas del ADC.

5.3 Integración de dispositivo portátil

Uniendo los bloques que conforman las etapas del sistema de medición de ECG se logró integrar un dispositivo portátil de bajo peso y volumen que es capaz de transmitir de forma inalámbrica a través de bluetooth los valores de voltaje que conforman una señal de ECG desde el puerto UART del microcontrolador, medidos con el módulo de ADC de 24 bits de este tras ser filtrados y amplificados analógicamente, para ser visualizados en un monitor serial de una computadora o teléfono inteligente.

La utilización de baterías para alimentar el dispositivo evitó que las mediciones que este realiza se vieran afectadas por interferencia provocada por la alimentación desde un tomacorriente.

5.4 Filtrado digital

Haciendo un uso adecuado de la aplicación filter designer de MATLAB se logró diseñar un filtro digital con las características necesarias con el objetivo de atenuar el componente de baja frecuencia contenido en la DLB en la señal 1.

El filtro de 2do orden no logró atenuar completamente la DLB, especialmente cuando la señal comienza a oscilar.

Se observó una atenuación tanto del valor RMS como de la desviación estándar en la señal tras ser filtrada respecto a la señal original.

5.5 Interpolación spline

La técnica de interpolación spline utilizada para replicar la tendencia de la DLB produjo una señal aún contaminada por el componente de baja frecuencia contenido en la DLB, pues la línea de tendencia generada no fue demasiado similar a la que se buscó replicar utilizando los puntos seleccionados de forma arbitraria de la señal original.

La amplitud del componente de baja frecuencia que contaminaba la señal después de aplicar la técnica disminuyó con respecto a la señal original.

Una línea de tendencia de mayor utilidad podría ser generada al utilizar más puntos de la señal original para interpolar.

5.6 Atenuación de DLB con plantilla de ECG en reposo

Con una plantilla de ECG del mismo paciente mientras se encuentra en estado de reposo, asumiendo que los ciclos de cada latido serían iguales a lo largo de la señal, se logró generar, tras un filtrado adecuado, una señal que replicara a la DLB provocada por movimiento del paciente que contaminaba la señal original.

La atenuación del componente de baja frecuencia en la señal se pudo demostrar al medir valores de RMS y desviación estándar, pues ambos disminuyeron respecto a la señal original.

La atenuación lograda por este método fue mayor que la lograda por los otros dos métodos incluidos en esta investigación.

5.7 Efectividad de algoritmo propio para detección de picos R de ECG

El algoritmo creado para la detectar picos R de una señal de ECG mostró ser efectivo al detectar de manera correcta los picos R de cada una de las señales de prueba.

El algoritmo utiliza funciones simples del sistema de desarrollo, lo cual permitiría replicarlo en otros ambientes de desarrollo con otros lenguajes de programación, lo cual vuelve factible que sea empotrado en el mismo dispositivo portátil que realice las mediciones para automatizar el trabajo de detección de picos.

La selección arbitraria del valor de umbral adecuado es un factor importante que condiciona la correcta funcionalidad del algoritmo. Esto podría presentar una limitación del algoritmo, pues al trabajar con señales con una DLB de mayor amplitud sería capaz de presentar falsos positivos.

5.8 Efectividad de algoritmo Pan-Tompkins para detección de picos R de ECG

El algoritmo Pan-Tompkins fue capaz de detectar cada uno de los picos en las mismas posiciones en que se identificaron previamente de manera manual en las tres señales de prueba.

Se utiliza un código extenso para ejecutar el algoritmo en el entorno de MATLAB, así como se utilizan muchas funciones y variables. Esto podría ser una limitación al traducir el código a un lenguaje de programación utilizado para programar un dispositivo capaz de realizar las mediciones de ECG, si se desea que la detección de picos la realice el mismo dispositivo.

5.9 Efectividad de algoritmo SimpleRST para detección de picos R de ECG

El algoritmo SimpleRST ejecutado en el ambiente de MATLAB solamente mostró efectividad en las señales 2 y 3 al detectar las posiciones de los picos en las mismas en que se detectaron de forma manual. El algoritmo no logró identificar todos los picos que a lo largo de la señal 1 se encontraban presentes. Debido a que la señal de interés fue la única en la que el detector no funcionó de manera adecuada, el algoritmo no es adecuado para ser utilizado en un dispositivo en el que se desee medir la señal y realizar la detección de picos de esta.

5.10 Comparación de efectividad de detectores de picos R de ECG

De acuerdo con los resultados obtenidos tras realizar pruebas con cada una de las 3 señales en cada detector, se ordenan en orden descendiente de viabilidad para ser empotrados en un dispositivo programable de la siguiente manera: Algoritmo propio, algoritmo Pan-Tompkins, y algoritmo SimpleRST.

CONCLUSIONES

Al concluir este proyecto de investigación se obtuvo un método que permite obtener atenuar la deriva de la línea basal de una señal electrocardiográfica con una mayor efectividad que utilizando un filtro digital y que realizando una interpolación spline para replicar la tendencia de la DLB para posteriormente sustraerla de la señal ruidosa.

El método propuesto requiere que el paciente sea capaz de permanecer en reposo por un breve periodo de tiempo para lograr la obtención de una plantilla de ECG no contaminada por el ruido que el movimiento del paciente provoque. Por lo tanto, no sería posible realizar la atenuación de la DLB de una señal de un paciente que se encuentre en movimiento involuntario como al tener convulsiones, pero sí podría ser utilizado en señales capturadas de un paciente cuyo movimiento sí puede controlar y puede elegir cuándo permanecer en estado de reposo, como un paciente al realizar ejercicio.

Para evaluar la viabilidad del método se valoró la posibilidad de implementarlo en un microcontrolador cuyo convertidor analógico-digital muestree la señal de un paciente en movimiento, para lo cual se validaron distintos detectores de complejos QRS, incluyendo uno creado durante este proyecto de investigación, el cual resultó más sencillo y, por lo tanto, con mayor viabilidad para ser implementado en tiempo real como trabajo futuro.

El entorno de desarrollo y programación MATLAB ha permitido crear e implementar algoritmos de detección de complejos QRS. Es una herramienta que facilita el diseño de filtros digitales de distintos tipos y permite su implementación en señales capturadas externamente. Todo esto vuelve a MATLAB un software útil para realizar un completo y adecuado postprocesamiento de señales biofisiológicas cuya información útil quiera ser extraída. Utiliza un lenguaje de programación demasiado versátil para realizar operaciones matemáticas con matrices y vectores, para lo cual presentan ciertas limitaciones otros lenguajes de programación como C.

La implementación de este método en otros lenguajes de programación de bajo nivel requerirá ajustes al código utilizado en MATLAB.

El convertido analógico-digital de 24 bits de arquitectura Delta-Sigma funciona de manera adecuada para la medición de señales biofisiológicas debido a la baja tasa de muestreo

requerida para la captura de estas. Aunque el fabricante indica que la tasa de muestreo máxima es de 1000 Hz, el incremento del ruido de cuantificación y la disminución de la relación señal-ruido hace que señales de baja amplitud como una señal de ECG se vean afectadas por dicho ruido, sin embargo, caracterizando el convertidor fue posible establecer una tasa de muestreo adecuada para la medición de una señal de ECG sin que el ruido de cuantificación le afecte.

Los resultados de esta investigación indican que la utilización de dispositivos de alta resolución para realizar mediciones biofisiológicas durante el ejercicio permite obtener señales que tras un adecuado postprocesamiento presenten información útil sobre el estado del paciente. Lo cual abre pie para la producción de dispositivos capaces de realizar el procesamiento de la señal con procesadores y controladores cuyas capacidades lo permitan, implementando algoritmos que automaticen el proceso presentado en esta investigación.

TRABAJO FUTURO

En este proyecto se propuso un método para procesar una señal de ECG que fue capturada por un dispositivo de alta resolución mientras el paciente se encontraba en movimiento, lo cual aporta un punto de partida para la implementación del método en tiempo real.

Algunas acciones que contribuirían y que deberían ser realizadas durante la continuación de esta investigación son:

- Investigar sobre antecedentes de detectores de complejos QRS ya empotrados en dispositivos portátiles.
- Traducción de códigos de algoritmos de detección de complejos QRS de MATLAB a lenguaje C.
- Escritura desde cero de algoritmos en lenguaje C
- Creación de algoritmo que permita identificar una onda de ECG en reposo si el paciente permanece en estado de reposo por un instante y poder obtener una plantilla.
- Validación de aplicación MATLAB coder para traducir algoritmos a lenguaje C.

También resultará adecuado el diseño de una carcasa para la integración del dispositivo portátil para la captura de señales ECG.

REFERENCIAS

- [1] E. Butkeviciute *et al.*, “Removal of Movement Artefact for Mobile EEG Analysis in Sports Exercises,” *IEEE Access*, vol. 7, pp. 7206–7217, 2019, doi: 10.1109/ACCESS.2018.2890335.
- [2] P. Bing, W. Liu, Z. Wang, and Z. Zhang, “Noise Reduction in ECG Signal Using an Effective Hybrid Scheme,” *IEEE Access*, vol. 8, pp. 160790–160801, 2020, doi: 10.1109/ACCESS.2020.3021068.
- [3] D. Cuevas-González, M. Bravo-Zanoguera, E. Altamira-Colado, R. L. Avitia, J. P. García-Vázquez, and M. A. Reyna, “Coping with motion artifacts by analog front-end ECG microchips under a low voltage supply, and variable digital resolution and gain,” *Engineering Proceedings*, p. 8247, 2020, doi: 10.3390/ecsa-7-08247.
- [4] M. Bravo-Zanoguera, D. Cuevas-González, M. A. Reyna, J. P. García-Vázquez, and R. L. Avitia, “Fabricating a portable ECG device using AD823X analog front-end microchips and open-source development validation,” *Sensors (Switzerland)*, vol. 20, no. 20, pp. 1–30, Oct. 2020, doi: 10.3390/s20205962.
- [5] N. Arora and B. Mishra, “Origins of ECG and Evolution of Automated DSP Techniques: A Review,” *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 140853–140880, 2021. doi: 10.1109/ACCESS.2021.3119630.
- [6] J. Li, G. Deng, W. Wei, H. Wang, and Z. Ming, “Design of a Real-Time ECG Filter for Portable Mobile Medical Systems,” *IEEE Access*, vol. 5, pp. 696–704, 2017, doi: 10.1109/ACCESS.2016.2612222.
- [7] J. J. Bailey *et al.*, “Special Report Recommendations for Standardization and Specifications in Automated Electrocardiography,” 1989. [En línea]. Disponible en: <http://ahajournals.org>
- [8] B. Chen, Y. Li, and N. Zeng, “Centralized Wavelet Multiresolution for Exact Translation Invariant Processing of ECG Signals,” *IEEE Access*, vol. 7, pp. 42322–42330, 2019, doi: 10.1109/ACCESS.2019.2907249.
- [9] World Health Organization, “Cardiovascular diseases (CVDs),” Jun. 21, 2021. [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) (Consultado el 13 de abril, 2022).
- [10] Tresguerres, “Capítulo 34: Bases electrofisiológicas del electrocardiograma,” en *Fisiología humana (3a. ed.)*, 2005.
- [11] ElsevierConnect, “6 claves para aprender a interpretar el electrocardiograma,” 2016. <https://www.elsevier.com/es-es/connect/medicina/6-claves-para-aprender-a-interpretar-el-electrocardiograma> (Consultado el 13 de abril, 2022).
- [12] J. Guerreiro *et al.*, “A Biosignal Embedded System for Physiological Computing,” Instituto Superior de Engenharia de Lisboa, 2013.

- [13] J. E. Hall and A. C. Guyton, *Fisiología Médica*, 13th ed. 2016. [En línea]. Disponible en: www.meddics.com
- [14] S. Maddio, “The Electrostatic Foundation of the Electrocardiogram.”
- [15] G. Chen, M. Chen, J. Zhang, L. Zhang, and C. Pang, “A Crucial Wave Detection and Delineation Method for Twelve-Lead ECG Signals,” *IEEE Access*, vol. 8, pp. 10707–10717, 2020, doi: 10.1109/ACCESS.2020.2965334.
- [16] John L. Semmlow and Benjamin Griffel, “Chapter 2: Biosignal Measurements, Noise, and Analysis,” en *BIOSIGNAL and MEDICAL IMAGE PROCESSING*, 3rd ed., 2014.
- [17] R. M. Rangayyan, “Chapter 1: Introduction to Biomedical Signals,” en *Biomedical signal analysis : a case-study approach*, 2002.
- [18] Z. F. Apandi, R. Ikeura, S. Hayakawa, and S. Tsutsumi, “An analysis of the effects of noisy electrocardiogram signal on heartbeat detection performance,” *Bioengineering*, vol. 7, no. 2, pp. 1–15, Jun. 2020, doi: 10.3390/bioengineering7020053.
- [19] J. C. Huhta and J. G. Webster, “60-Hz Interference in Electrocardiography,” *IEEE Transactions on Biomedical Engineering*, vol. BME-20, no. 2, pp. 91–101, 1973, doi: 10.1109/TBME.1973.324169.
- [20] “Smallies Med CVL8 - ECG Flashcards | Chegg.com.” <https://www.chegg.com/flashcards/smallies-med-cvl8-ecg-4d6ccb8c-4f32-4f6d-b65f-316dc79e75d8/deck> (Consultado el 20 de mayo, 2022).
- [21] A. C. Vinzio Maggio, M. Paula, E. Laciari, and P. David, “Quantification of Ventricular Repolarization Dispersion Using Digital Processing of the Surface ECG,” en *Advances in Electrocardiograms - Methods and Analysis*, InTech, 2012. doi: 10.5772/23050.
- [22] V. de Pinto, “Filters for the reduction of baseline wander and muscle artifact in the ECG,” *Journal of Electrocardiology*, vol. 25, pp. 40–48, 1992.
- [23] C. Stolojescu, “Estimation of noise in ECG signals using wavelets,” en *11th International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2008*, 2008, pp. 113–118. doi: 10.1109/OPTIM.2008.4602509.
- [24] U. Satija, B. Ramkumar, and M. S. Manikandan, “Automated ECG noise detection and classification system for unsupervised healthcare monitoring,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 3, pp. 722–732, 2018, doi: 10.1109/JBHI.2017.2686436.
- [25] M. Fontanals Fernández, “7. Holter,” en *Manual de Enfermería*, 2013, pp. 123–136.
- [26] S. Kwon *et al.*, “Validation of adhesive single-lead ecg device compared with holter monitoring among non-atrial fibrillation patients,” *Sensors*, vol. 21, no. 9, May 2021, doi: 10.3390/s21093122.

- [27] Analog Devices, “AD8232.” 2020. Accessed: Aug. 09, 2020. [En línea]. Disponible en: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>
- [28] “SparkFun Single Lead Heart Rate Monitor - AD8232 - SEN-12650 - SparkFun Electronics.” https://www.sparkfun.com/products/12650?_ga=2.163507196.1106084270.1653593674-1098657528.1652855862&_gac=1.149103044.1653593674.CjwKCAjwryUBhBSEiwAGN5OCACdEpu4QhmfBwYb3tMgOI_itLN3mda_-2wKxV99IaeNivUNqtsyKhoCaCMQAvD_BwE (Consultado el 20 de mayo, 2022).
- [29] C. Kuhns, “AD8232_Heart_Rate_Monitor_v10.” 2014.
- [30] N. v. Thakor, J. G. Webster, and W. J. Tompkins, “Optimal QRS detector,” *Medical & Biological Engineering & Computing*, vol. 21, no. 3, pp. 343–350, 1983, doi: 10.1007/BF02478504.
- [31] W. J. Tompkins, “A Real-Time QRS Detection Algorithm,” 1985.
- [32] T. Tegoshi *et al.*, “Implementation of algorithms for identification of QRS-complexes in real-time ECG systems,” *Biomedical Engineering*, vol. 35, no. 6, pp. 304–309, 2001, doi: 10.1023/A:1014626021370.
- [33] M. Jia, F. Li, J. Wu, Z. Chen, and Y. Pu, “Robust QRS Detection Using High-Resolution Wavelet Packet Decomposition and Time-Attention Convolutional Neural Network,” *IEEE Access*, vol. 8, pp. 16979–16988, 2020, doi: 10.1109/ACCESS.2020.2967775.
- [34] Hooman Sedghamiz, “Complete Pan Tompkins Implementation ECG QRS detector - File Exchange - MATLAB Central,” 2018. <https://la.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector> (Consultado el 16 de octubre, 2021).
- [35] L. Tan, “Chapter 3: Digital Signals and Systems,” en *Digital Signal Processing*, 2008, pp. 57–69.
- [36] A. Ambardar, *Analog and Digital Signal Processing*, 2nd ed. 2004.
- [37] R. M. Rangayyan, “Chapter 3: Filtering for Removal of Artifates,” en *Biomedical signal analysis : a case-study approach*, 2002.
- [38] R. M. Rangayyan, *Biomedical signal analysis : a case-study approach*. 2002.
- [39] Youngchun Kim and Ahmed H. Tewfik, “A Novel QRS Complex Detection On ECG With Motion Artifact During Exercise,” *IEEE Xplore*, 2015.
- [40] “Introducción a MATLAB - MathWorks América Latina.” <https://la.mathworks.com/help/matlab/getting-started-with-matlab.html> (Consultado el 13 de abril, 2022).

- [41] “Introduction to Filter Designer - MATLAB & Simulink Example - MathWorks América Latina.” <https://la.mathworks.com/help/signal/ug/introduction-to-filter-designer.html> (Consultado el 13 de abril, 2022).
- [42] “ADC $\Delta\Sigma$ elimina los circuitos AFE complejos | DigiKey.” <https://www.digikey.com.mx/es/articles/analog-basics-part-2-delta-sigma-adcs-how-to-exploit-digital-capabilities> (Consultado el 13 de abril, 2022).
- [43] D. de Un, F. Javier, D. Pino, S. D. Jesús, and R. Pulido Medina, “Diseño de un modulador Sigma-Delta en tiempo continuo para un PLL N-Fraccional en tecnología UMC 90nm,” 2013.
- [44] B. Baker, “How delta-sigma ADCs work, Part 1,” 2011. [En línea]. Disponible en: www.ti.com/aaj
- [45] “Sigma-Delta A/D Conversion - MATLAB & Simulink - MathWorks América Latina.” <https://la.mathworks.com/help/dsp/ug/sigma-delta-a-d-conversion.html> (Consultado el 13 de abril, 2022).
- [46] SiliconLabs, “C8051F350/1/2/3 8 k ISP Flash MCU Family.” Silicon Labs. Accessed: May 26, 2022. [En línea]. Disponible en: <https://www.silabs.com/documents/public/data-sheets/C8051F35x.pdf>
- [47] “C8051F350DK C8051 8-bit MCU Development Kit - Silicon Labs.” <https://www.silabs.com/development-tools/mcu/8-bit/c8051f350-development-kit> (Consultado el 13 de abril, 2022).
- [48] “C8051F35x-DK C8051F35X-DK DEVELOPMENT KIT USER ’S GUIDE,” 2014. [En línea]. Disponible en: www.silabs.com/simplicity-studio.
- [49] “Simplicity Studio - Silicon Labs.” <https://www.silabs.com/developers/simplicity-studio> (Consultado el 13 de abril, 2022).
- [50] “Simplicity Studio® 5 User’s Guide - v5.3.0 - Simplicity Studio 5 Users Guide Silicon Labs.” <https://docs.silabs.com/simplicity-studio-5-users-guide/5.3.0/ss-5-users-guide-overview/> (Consultado el 13 de abril, 2022).
- [51] e-Gizmo Mechatronix Central, “Bluetooth Modules Wireless UART Cable Replacement Hardware Manual & AT Commands Reference Manual Rev. 1r0.”
- [52] “Cubic spline data interpolation - MATLAB spline - MathWorks América Latina.” <https://la.mathworks.com/help/matlab/ref/spline.html> (Consultado el 18 de octubre, 2021).
- [53] “Replicating Real World Signals with an Arbitrary/Function Generator.” [En línea]. Disponible en: www.tektronix.com/oscilloscopes
- [54] “SimpleRST.” <https://in.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/45404/versions/13/previews/SimpleRST.m/index.html> (Consultado el 21 de noviembre, 2021).

- [55] “Determine array equality - MATLAB isequal - MathWorks América Latina.”
<https://la.mathworks.com/help/matlab/ref/isequal.html> (Consultado el 13 de abril, 2022).

APÉNDICE A. CÓDIGOS DE MATLAB

A1. Algoritmo propio para detección de complejos QRS

```
function [expected,picos]=expect(original)
original=original(:);
% declarar ecg

ecg=original(36:127)';
[m,n]=size(original);
i=1;
idx=1;

% crear vector de picos

picos=zeros(1,100);

% ciclo while
while i<m
    if original(i+1)-original(i)>50000
        %obtener tamaño de vector
        [o,p]=max(original(i:i+15));
        %añadir pico al vector de picos
        picos(idx)=p+i-1;
        i=i+15;
        idx=idx+1;
    else
        i=i+1;
    end
end
d=0;
c=1;
expected=int32([]);
[q,r]=max(ecg);
for j=1:length(picos)
    if picos(j)>0 && d==0
        expected=[zeros(1,picos(j)-r),ecg];
        c=j;
        d=d+1;
    elseif picos(j)>0
        expected=[expected,zeros(1,picos(j)-length(ecg)-picos(c)),ecg];
        c=j;
    end
end
end
```

A2. Filtro digital para atenuar DLB de señal original

```
b=[1,-2,1];
a=[1,-1.950644296288739,0.951833300398571];
filtered_signal=filter(b,a,original);
```

A3. Código para método por interpolación spline

```
% pq is the vector containing the original signal's indexes used to
%generate a trend curve using a spline interpolation
pq=[1 143 286 417 544 675 769 939 1088 1228 1351 1420 1495 ...
    1652 1790 1828 1893 2032 2121 2231 2381 2551 2734 2829 ...
    3044 3200 3266 3333 3371 3491 3519 3623 3668];

% knots is the vector containing the points to interpolate
knots=(original(pq));

% A trend curve of the same length as the original signal is
% created
xq=1:1:size(original);
baseline=spline(pq,knots,xq)';

% The trend curve is subtracted from the original signal
filtrada=original-baseline;

subplot(3,1,1), plot(original), ylim([-5E+05 6E+05]),
xlim([0 3620]), xlabel('samples'), title('Original signal');
subplot(3,1,2), plot(baseline), ylim([-5E+05 6E+05]),
xlim([0 3620]), xlabel('samples'), title('Trend curve');
subplot(3,1,3), plot(filtrada), ylim([-5E+05 6E+05]),
xlim([0 3620]), xlabel('samples'), title('Processed signal');
```

A4. Atenuación de DLB por método propuesto con plantilla

```
% ECG template is created
ecg=original(36:127)';

% The desired signal is created
ceros=[zeros(1,35),ecg,zeros(1,40),ecg,zeros(1,40),ecg, ...
    zeros(1,47),ecg,zeros(1,43),ecg,zeros(1,38),ecg, ...
    zeros(1,37),ecg,zeros(1,42),ecg,zeros(1,51),ecg, ...
    zeros(1,51),ecg,zeros(1,47),ecg,zeros(1,47),ecg, ...
    zeros(1,52),ecg,zeros(1,65),ecg,zeros(1,54),ecg, ...
    zeros(1,51),ecg,zeros(1,62),ecg,zeros(1,74),ecg, ...
    zeros(1,85),ecg,zeros(1,60),ecg,zeros(1,60),ecg, ...
    zeros(1,51),ecg,zeros(1,55),ecg,zeros(1,63),ecg, ...
    zeros(1,48),ecg,zeros(1,74)]';

% The desired signal is subtracted to the original signal to obtain
% the signal's motion artifact
artefacto=original-ceros;

% The motion artifact is filtered with a 2nd order low-pass filter
% to replicate the baseline wander
a=[1 -1.950644296288738921418826066656038165092 ...
    0.951833300398571169331773944577435031533];
b=[1 2 1];
mafilt=filter(b,a,artefacto);
```

```

% The filtered motion artifact is adjusted back on phase
mafilt4=[mafilt(48:3672)',zeros(1,47)'];

% The amplitude is adjusted
relacion=max(mafilt4)/max(artefacto);
mafilt5=mafilt4/relacion;
% A 60 Hz notch filter is applied and the amplitude is adjusted
filtrada4=original-mafilt5;
b2=[0.957379058065359611617850532638840377331 ...
    0.89892448513685641042769702835357747972 ...
    0.957379058065359611617850532638840377331];
a2=[1 0.89892448513685641042769702835357747972 ...
    0.914758116130719223235701065277680754662];
filtrada5=filter(b2,a2,filtrada4);
relacion2=max(filtrada5)/max(filtrada4);
filtrada5=filtrada5/relacion2;
subplot(3,1,1), plot(original), ylim([-4E+05 6E+05]),
xlim([0 3620]), xlabel('samples'), title('Original signal');
subplot(3,1,2), plot(mafilt5), ylim([-4E+05 6E+05]),
xlim([0 3620]), xlabel('samples'), title('Baseline Wander');
subplot(3,1,3), plot(filtrada5), ylim([-4E+05 6E+05]),
xlim([0 3620]), xlabel('samples'), title('Processed signal');

```

A5. Generación de espectro de frecuencias

```

Fs = 180;                % Sampling frequency
T = 1/Fs;                % Sampling period
L = 3672;                % Length of signal
t = (0:L-1)*T;           % Time vector

Y = fft(filtrada4);
P2 = abs(Y/L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(L/2))/L;
plot(f,P1)
xlabel('Frecuencia (Hz)', 'FontSize',12, 'FontWeight', 'bold')
ylabel('Magnitud', 'FontSize',12, 'FontWeight', 'bold')

```

A6. Código para graficar señales en un mismo formato

```

function grafcualq(x)
t=0:1/190:(length(x)-1)/190;
figure
plot(t,x)
set(gcf, 'Position', [-100, 400, 2020, 350])
xlim([0 (length(x)-1)/190])
ylim([-400000 600000])
xlabel('Tiempo (seg)', 'FontSize',12, 'FontWeight', 'bold')
ylabel('Amplitud (unidades digitales)', 'FontSize',12, 'FontWeight', 'bold')

```