

UNIVERSIDAD DE MÁLAGA  
ESCUELA TÉCNICA SUPERIOR DE  
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DISEÑO, IMPLEMENTACIÓN Y PRUEBAS  
DE UN SISTEMA DE ANÁLISIS ECG  
PARA DISPOSITIVOS PORTÁTILES

GRADO EN INGENIERÍA DE  
SISTEMAS ELECTRÓNICOS

DIONISIO ROMERO DÍAZ  
MÁLAGA, 2020



## **DISEÑO, IMPLEMENTACIÓN Y PRUEBAS DE UN SISTEMA DE ANÁLISIS ECG PARA DISPOSITIVOS PORTÁTILES**

**Autor:** Dionisio Romero Díaz

**Tutor:** Carmen García Berdonés

**Cotutor:** <Nombre del cotutor> (elimina esta línea si no hay cotutor)

**Departamento:** <Nombre de departamento>

**Titulación:** Grado en Ingeniería de sistemas electrónicos

**Palabras clave:** SEC, ECG, Detección QRS en tiempo real

### **Resumen**

El resumen debe ser una breve descripción del contexto del proyecto, sus objetivos y los resultados obtenidos. Se recomienda que no exceda esta página.



## **DESIGN, IMPLEMENTATION AND TEST OF AN ECG ANALYSIS SYSTEM FOR PORTABLE DEVICES**

**Author:** Dionisio Romero Díaz

**Supervisor:** Carmen García Berdonés

**Co-supervisor:** <Nombre del cotutor> (elimina esta línea si no hay cotutor)

**Department:** <Department name>

**Degree:** Grado en Ingeniería de sistemas electrónicos

**Keywords:** CES, ECG, Real-time QRS detection

### **Abstract**

The abstract should briefly describe the project context, goals and obtained results. It should not exceed this page.



En caso de dedicatoria,  
se realiza con esta página.  
No es obligatoria, si bien es recomendable.

*El autor*





# Agradecimientos

Este apartado es opcional. En él se incluirían los agradecimientos personales y profesionales. Si no los hubiere, debe eliminarse esta página y la siguiente (para ello puedes comentar la línea 48 de A0.MiTFG.tex).



# Acrónimos

<b>ETSIT</b>	Escuela Técnica Superior de Ingeniería de Telecomunicación
<b>PFC</b>	Proyecto Fin de Carrera
<b>TFG</b>	Trabajo Fin de Grado
<b>TFM</b>	Trabajo Fin de Máster
<b>UMA</b>	Universidad de Málaga
<b>ECG</b>	Electrocardiografía



# Índice

<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>V</b>
<b>Agradecimientos</b>	<b>IX</b>
<b>Acrónimos</b>	<b>XI</b>
<b>1 Introducción y visión general</b>	<b>1</b>
1.1 Estado del arte . . . . .	1
1.1.1 Monitores portátiles de ECG . . . . .	2
1.1.2 La señal ECG . . . . .	3
1.1.3 Cálculo de la tasa cardíaca basada en ECG . . . . .	5
1.1.4 Breves conclusiones . . . . .	6
1.2 Objetivo . . . . .	7
1.3 Estructura del documento . . . . .	7
1.4 Ámbito de aplicación . . . . .	7
<b>2 Especificaciones</b>	<b>9</b>
2.1 Diagramas . . . . .	10
2.1.1 Contexto . . . . .	10
2.1.2 Casos de uso . . . . .	10
2.1.3 Flujo . . . . .	11
2.2 Requisitos generales . . . . .	13

2.3 Metodología . . . . .	15
<b>3 Implementación</b>	<b>17</b>
3.1 Iteración inicial: Planteamiento básico . . . . .	18
3.1.1 Resumen . . . . .	18
3.1.2 Desarrollo . . . . .	18
3.1.3 Diagramas . . . . .	18
3.1.4 Conclusiones . . . . .	19
3.2 Primera iteración: El núcleo . . . . .	20
3.2.1 Resumen . . . . .	20
3.2.2 Requisitos . . . . .	20
3.2.3 Desarrollo . . . . .	20
3.2.4 Pruebas . . . . .	24
3.2.5 Conclusiones . . . . .	25
<b>Conclusiones y líneas futuras</b>	<b>27</b>
<b>A Manual de uso</b>	<b>29</b>
A.1 Primera sección . . . . .	29
<b>Bibliografía</b>	<b>31</b>

# Índice de figuras

1.1	Principales causas de muerte en 2016 . . . . .	2
1.2	Elementos fundamentales ECG . . . . .	4
1.3	Ilustración derivaciones ECG . . . . .	5
1.4	Representación intervalo R-R . . . . .	5
2.1	Diagrama de contexto . . . . .	10
2.2	Diagrama de casos de uso . . . . .	11
2.3	Diagrama de bloques . . . . .	12
3.1	Diagrama de bloques general . . . . .	19
3.2	Función de Matlab para extraer los datos del repositorio y formatearlos como texto . . . . .	21
3.3	Diagrama de lectura de ficheros simplificado . . . . .	21
3.4	Diagrama de lectura de ficheros en QT . . . . .	22
3.5	Selector de frecuencias y lectura del fichero . . . . .	23
3.6	Proceso construcción de trama . . . . .	24





# Índice de Tablas

2.1	Requisitos iniciales . . . . .	14
-----	--------------------------------	----



# Capítulo 1

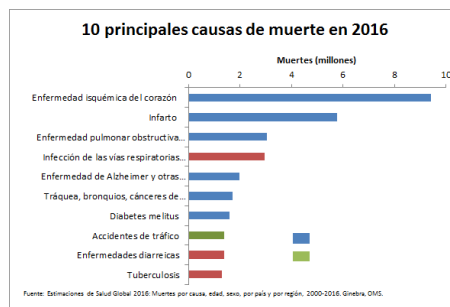
## Introducción y visión general

### Contenido

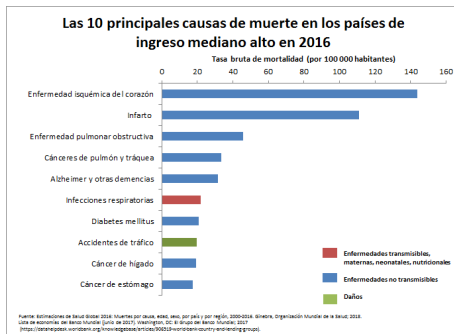
<b>1.1 Estado del arte</b>	<b>1</b>
1.1.1 Monitores portátiles de ECG	2
1.1.2 La señal ECG	3
1.1.3 Cálculo de la tasa cardíaca basada en ECG	5
1.1.4 Breves conclusiones	6
<b>1.2 Objetivo</b>	<b>7</b>
<b>1.3 Estructura del documento</b>	<b>7</b>
<b>1.4 Ámbito de aplicación</b>	<b>7</b>

### 1.1. Estado del arte

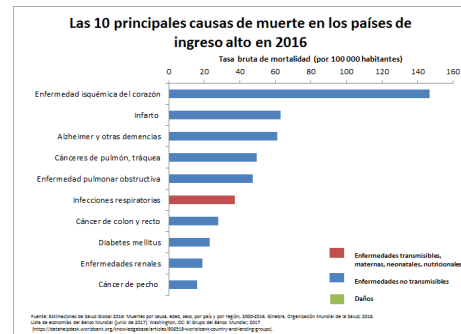
Como puede verse en la figura 1.1, según la organización mundial de la salud (OMS) la principal causa de muertes en el mundo el año 2016 fueron ocasionadas por una enfermedad isquémica del corazón y la segunda más relevante los infartos. Además cabe destacar que más del 75 % de las defunciones provocadas por enfermedades cardiovasculares se producen en países con bajos o medianos ingresos y que el 80 % de los infartos de miocardio y de los AVC (Accidente Vascular Cerebral) prematuros son prevenibles, suena razonable cualquier intento por desarrollar sistemas para mejorar la diagnosis y prevenir dichas enfermedades con mayor efectividad.



(a) Fallecimientos globales



(b) Fallecimientos en países con ingresos bajos-medios



(c) Fallecimientos en países con ingresos altos

Figura 1.1: Principales causas de muerte en 2016

### 1.1.1. Monitores portátiles de ECG

Desde la década de los cuarenta se llevan investigando y mejorando dispositivos portátiles capaces de registrar las señales de ECG con el fin de obtener datos durante periodos extrahospitalarios. Concretamente el más antiguo de estos dispositivos es conocido como Holter, en honor a Norman J. Holter, quien desarrolló el primer prototipo en 1947, el cual pesaba alrededor de 38 Kg. No sería hasta el año 1962 cuando gracias a “Del Mar Engineering Laboratories” bajo el nombre de “Avionics Research Products Corporation” se consiguiera producir el primer dispositivo comercial, más ligero y capaz de almacenar el ECG de las últimas 24 horas. [2]

Hasta hoy el paradigma del registro de ECG portátil ha cambiado ligeramente gracias al desarrollo de la técnica y la utilidad de los monitores portátiles en la diagnosis y tratamiento de enfermedades cardiovasculares. En la actualidad hay dos tipos de dispositivos ampliamente utilizados, los ya mencionados monitores Holters y los monitores de eventos cardíacos. La principal diferencia entre ambos es la forma de almacenar los datos, mientras que los holters registran los datos

constantemente desde que es colocado por el médico hasta la consulta de vuelta, los monitores de eventos cardíacos son activados por los pacientes cuando padecen algún síntoma cardiovascular.

Teóricamente los monitores de eventos cardíacos son menos eficientes a la hora de la diagnosis de enfermedades que los Holters debido a que pierden los datos referentes a eventos que sean asintomáticos o cuando el paciente no los perciba. Sin embargo encuentran su espacio en el mercado ya que típicamente son más pequeños y cómodos de usar que los Holters.

No obstante esto podría cambiar gracias al desarrollo de un nuevo tipo de monitores de eventos cardíacos, los supervisores de eventos cardíacos autodisparados. La teoría detrás de estos dispositivos es sencilla, si se puede analizar en tiempo real la señal adquirida se puede detectar en tiempo real el evento deseado y emplearlo para almacenar ese fragmento del ECG del paciente e incluso transmitirla. Evitando así el enorme gasto de batería que conlleva la transmisión constante del registro y gozando de todas la ventajas de los supervisores cardíacos sin su mayor contra. La práctica, por otro lado, no es tan sencilla como se expondrá después.

La base del funcionamiento de un supervisor de eventos cardíaco autodisparados es la detección automática de los latidos presentes en la señal ECG. (TODO: Satija Reference) Localizados estos latidos, el supervisor debe extraer diversas características de ellos que permitan clasificarlos como anormales. La ocurrencia de estas anomalías constituyen los eventos susceptibles de ser guardados o transmitidos. Antes de continuar hablando de este tipo de monitores cardíacos es necesario comprender moderadamente los diferentes elementos de un Electrocardiograma y qué datos se pueden obtener de ellos.

### 1.1.2. La señal ECG

Como se puede ver en la figura 1.2 pueden apreciarse cinco elementos principales, la onda P que representa la despolarización auricular, el complejo QRS, fruto de la despolarización ventricular y la onda T que representa la repolarización ventricular. Además de las ondas en sí mismas se puede extraer mucha información de los intervalos entre las ondas, aportando información sobre las duraciones de los procesos biológicos del corazón.

En función de las posiciones donde se coloquen los electrodos a la hora de tomar las muestras las señales se recibirán con una polaridad u otra. Esto se conoce como una derivación y el ECG clásico posee doce derivaciones agrupadas en d grupos de seis, seis derivaciones de las extremidades y seis precordiales. Las de las extremidades registran los potenciales que se transmiten al plano

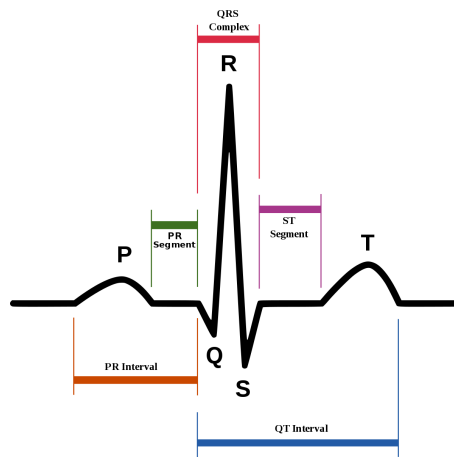


Figura 1.2: Elementos fundamentales ECG

frontal y las precordiales recogen los potenciales del plano horizontal como puede verse en la figura 1.2. [5]

Podría decirse que cada derivación hace las veces de un observador en un punto diferente, por lo que la señal resultante, que procede de diferencias potenciales con dirección y sentido, variará de polarización en según qué derivación, como puede “variar” el sentido de rotación de disco en función de si es observado desde arriba o desde abajo.

Desde que Willem Einthoven inventara el primer electrocardiograma hasta hoy la técnica ha mejorado enormemente y la precisión en las medidas hace posible observar fenómenos muy sutiles con un método de pruebas muy poco invasivo. La cantidad de patologías que pueden diagnosticarse mediante un electrocardiograma en el momento apropiado es enorme. [1] Por ejemplificar esto tan solo mediante la morfología de la onda P se pueden diagnosticar más de 15 patologías diferentes, entre ellas fibrilación auricular cuando se da una ausencia total de la onda o varios tipos de taquicardia cuando se observan ausencias parciales. También se puede detectar un crecimiento auricular anormal si se observa que la onda es bifásica. Y un largo etcétera de patologías.

En muchas ocasiones la detección de estas patologías requiere como paso previo la detección del latido cardíaco. Una vez detectado se pueden realizar procesos complicados basados en la morfología de la onda u otros más sencillos como el cálculo de la tasa cardíaca, que pueden aportar mucha información.

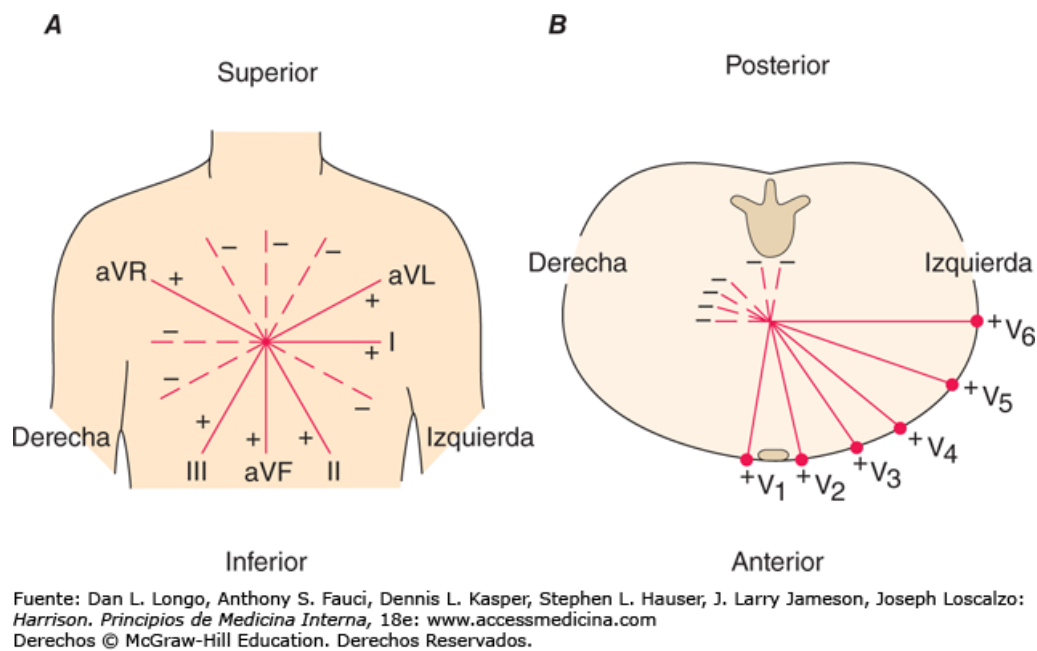


Figura 1.3: Ilustración derivaciones ECG

### 1.1.3. Cálculo de la tasa cardíaca basada en ECG

Una vez reconocidos los latidos cardíacos se puede medir la diferencia de tiempo entre dos latidos, esto se conoce como intervalo RR (Ver figura 1.4) y es equivalente al período de los latidos, de donde se obtiene la frecuencia cardíaca instantánea, parámetro típico que es empleado para activar numerosos supervisores de eventos cardíacos autodisparados, y es por esto que el proyecto se centra en este parámetro.

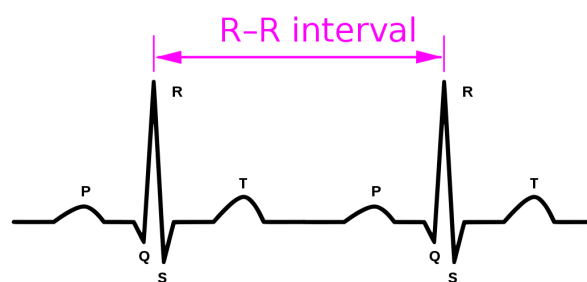


Figura 1.4: Representación intervalo R-R

En la literatura actual existen numerosos algoritmos muy testeados y eficaces a la hora de detectar el complejo QRS y, en base a esa detección, calcular la tasa cardíaca. Tradicionalmente se ha desarrollado para ser ejecutados sobre unos datos extraídos previamente y almacenados en memoria. Pero esto no sirve para ser aplicados en tiempo real por diseño, así que en los últimos años se ha estado invirtiendo mucho esfuerzo en desarrollar algoritmos de tiempo real. Siendo el más conocido y uno de los más antiguos el de “Pan Tompkins” basado en el análisis de la pendiente, la intensidad y la duración de los complejos QRS.[6] A pesar de ello, a día de hoy se pueden encontrar numerosos estudios sobre nuevas metodologías para llevar a cabo esta tarea como algoritmos especializados en muestras con mucho ruido[3] o métodos con diferentes aproximaciones como máquinas de estado finitas[4] o empleando coeficientes de wavelet.[7]

Sin embargo no resulta sencillo encontrar la forma de testar dichos métodos en entornos reales. Para realizar las pruebas pertinentes se hace uso de bases de datos estandarizadas como la MIT-BIH (TODO: Poner esto good con referencia y toh.), evaluando principalmente la tasa de falsos positivos y negativos. (TODO: Referencia Pootja 2017) Un ejemplo de algoritmo para determinar la fiabilidad de un algoritmo se puede encontrar en la norma para marcado CE de los Holter de ECG (TODO: referencia- AENOR, te la mando por correo)

#### **1.1.4. Breves conclusiones**

Como se ha mencionado anteriormente las muertes ocasionadas por patologías cardíacas son una de las mayores causas de mortalidad en todo el mundo, por ende los supervisores de eventos cardiacos autodisparador podrían tener un efecto directo en la detección y seguimiento de estas patologías. La base principal de estos supervisores radica en la correcta detección de los complejos QRS, por lo que su fiabilidad es de suma importancia.

Por otro lado, la tendencia actual es emplear dispositivos cada vez de menor consumo que suelen ir acompañados de menores recursos y baterías más pequeñas. Por ello cada vez se requieren métodos de detección con menor complejidad computacional pero con una robustez suficiente para ser caracterizados como fiables. Minimizando así las transmisiones de datos a lo estrictamente necesario, ahorrando batería para poder tomar medidas durante un periodo más largo. En este contexto resulta complejo evaluar cuando un algoritmo es suficientemente válido o no debido únicamente a su fiabilidad de detección, influyen también parámetros como el consumo de batería en ejecución, la necesidad de memoria y la portabilidad de este a lenguajes de bajo nivel.



## 1.2. Objetivo

El objetivo de este proyecto es el diseño y la implementación de un entorno de pruebas genérico para testear diferentes algoritmos de detección de tasa cardíaca instantánea basados en ECG en un entorno portátil de tiempo real con recursos limitados. Siendo validado con un ejemplo concreto de algoritmo aplicado sobre señales ECG tomadas de la MIT-BIH.

Como objetivos personales de este proyecto se propusieron la profundización en los conocimientos sobre el procesamiento de datos, concretamente los datos de electrocardiogramas. Además se propuso emplear dicho proyecto para investigar y aprender a realizar documentación con  $\text{\LaTeX}$

## 1.3. Estructura del documento

En esta sección, se explican los posteriores capítulos u otra información adicional que el proyecto contenga.

## 1.4. Ámbito de aplicación

Por último, completando los apartados anteriores, se explican las áreas de las que se compone el proyecto.



# Capítulo 2

## Especificaciones

### Contenido

<b>2.1 Diagramas</b>	<b>10</b>
2.1.1 Contexto	10
2.1.2 Casos de uso	10
2.1.3 Flujo	11
<b>2.2 Requisitos generales</b>	<b>13</b>
<b>2.3 Metodología</b>	<b>15</b>

## 2.1. Diagramas

En este apartado se engloban todos los diagramas realizados a la hora de definir el proyecto y su alcance.

### 2.1.1. Contexto

En la figura 2.1 se puede observar el entorno circundante al proyecto, representado por un diagrama de contexto. El bloque correspondiente al proyecto en sí es el denominado como “Entorno pruebas de algoritmos ECG”, los otros bloques representan elementos relacionados con los que se interacciona de una forma u otra.

Este diagrama resulta de bastante ayuda a la hora de establecer los casos de uso que serán detallados más adelante, pues al visualizar de un solo vistazo todo el entorno circundante al producto se pueden descubrir nuevas interacciones que habrían sido ignoradas de otra forma.

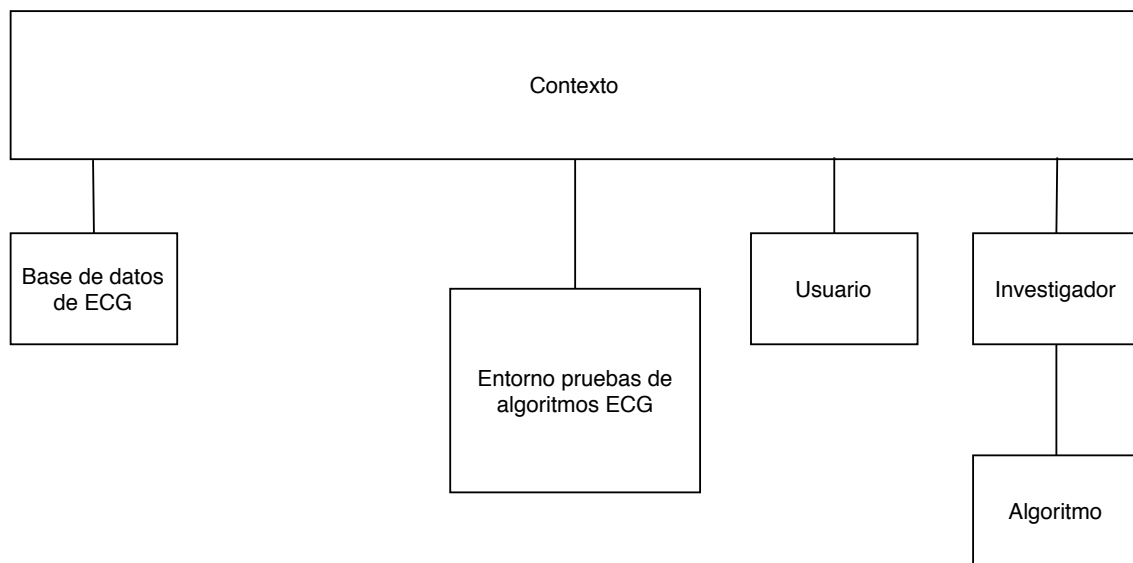


Figura 2.1: Diagrama de contexto

### 2.1.2. Casos de uso

Una vez establecido el contexto en el que se desenvuelve el proyecto se pueden establecer los casos de uso. De nuevo se realiza un esquema para poder

visualizarlos como conjunto. Este esquema puede verse en la figura (TODO: Cita figura), en él se muestran las acciones que podría llevar a cabo el usuario una vez el proyecto esté finalizado, aunque será tarea futura determinar cuales gozan de mayor prioridad, pues no todas son iguales de fundamentales a la hora de llevar a cabo el objetivo principal del proyecto.

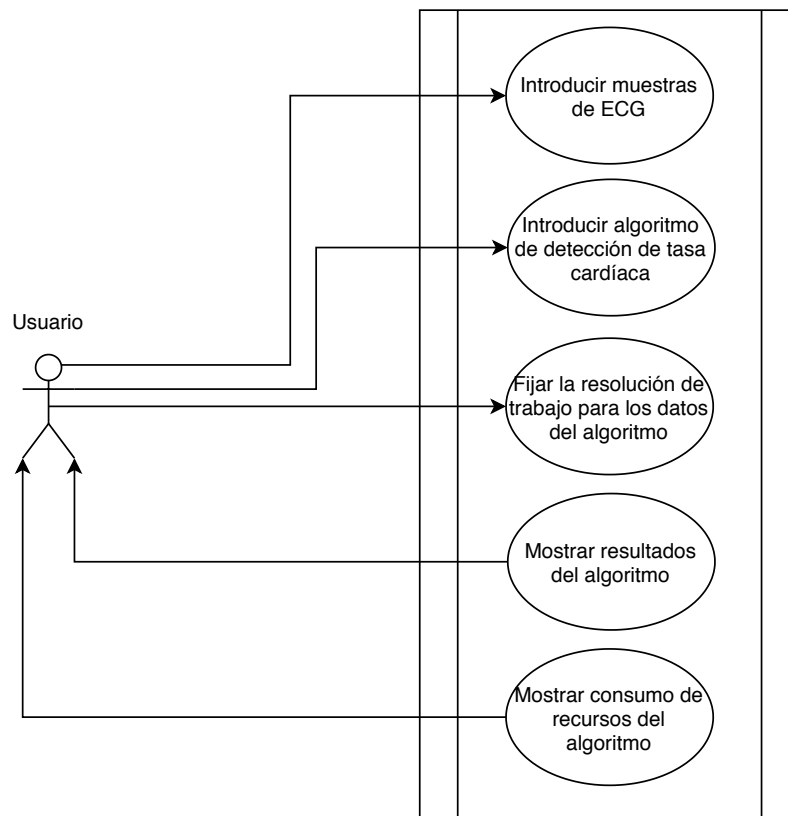


Figura 2.2: Diagrama de casos de uso

Estando aclaradas estas interacciones ya resulta posible comenzar a establecer los requisitos del proyecto, sin embargo primero se debería establecer conceptualmente el flujo de trabajo que se llevará a cabo para el testeo de los algoritmos.

### 2.1.3. Flujo

De forma muy simplificada se puede observar en la figura (TODO: Citar figura) el flujo de trabajo del entorno de pruebas. Este diagrama es útil para tener en mente en todo momento el objetivo del proyecto en su versión más simplificada.

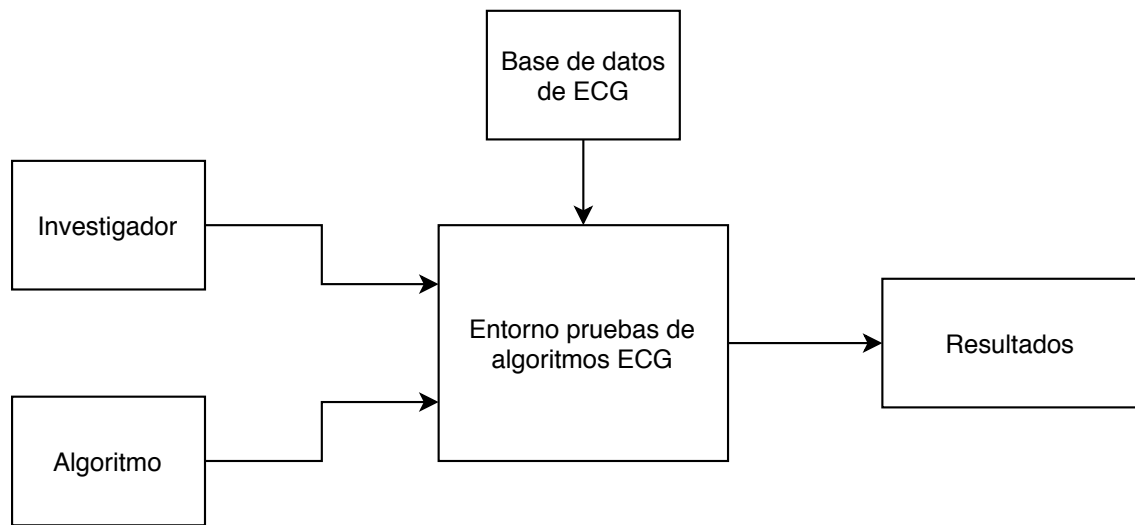


Figura 2.3: Diagrama de bloques

## 2.2. Requisitos generales

Con la ayuda de los diagramas realizados se establecen los siguientes requisitos para cumplir con el objetivo establecido:

ID	Requisito	Descripción	Prueba	Prio
1	Panel de control	El proyecto debe disponer de un panel de usuario intuitivo para llevar a cabo las pruebas		1
1.1	Introducir señal de prueba	Al panel de control se le debe proveer una señal de ECG formateada específicamente para poder probar el algoritmo deseado	Introducir un ECG extraído de alguna base de datos y comprobar que los valores de las muestras enviados se corresponden	1
1.2	Observar el comportamiento del algoritmo	Al panel de control se le debe proveer de gráficas de salida que muestren el estado de la señal procesada.	Observar las gráficas de salida y comprobar con un algoritmo predecible que el resultado obtenido es el deseado	2
1.3	Resultados	El panel de control debe ser capaz de mostrar al usuario los siguientes elementos:		
1.3.1	Frecuencia cardíaca instantánea	El panel deberá mostrar la frecuencia cardíaca instantánea medida por el algoritmo a testear.	Comprobar que el dato enviado por la aplicación de pruebas y el recibido por el panel son el mismo. La precisión de este no es un requisito pues dependerá del algoritmo con el que se esté trabajando.	1
1.3.2	Picos detectados	El panel deberá mostrar información sobre los picos R detectados y sus posiciones, para facilitar la evaluación del algoritmo	Al igual que la prueba anterior, se debe garantizar la coherencia de los datos entre la aplicación y el panel de usuario. La validez de estos dependerá del algoritmo.	1
1.3.3	Utilización del procesador instantánea	El panel deberá mostrar de forma instantánea el uso de CPU que corresponde al proceso del algoritmo de la forma más aproximada posible		2
1.3.4	Utilización de memoria instantánea	El panel deberá mostrar el uso de memoria en cada momento de la forma más aproximada posible		2
1.3.5	Fiabilidad del algoritmo	El panel deberá mostrar la fiabilidad (bien visual o bien en forma de tasas de falsos positivos y negativos) del algoritmo.	Contrastar con las anotaciones de la base de datos.	2
1.4	Resolución muestras	El panel deberá poder configurar la resolución empleada para las muestras en la aplicación, ofreciendo al usuario cierto control sobre la cantidad de muestras simultáneas con las que su algoritmo podrá trabajar.		2
2	Aplicación de pruebas	El proyecto debe disponer de una aplicación encargada de simular el comportamiento de un supervisor de eventos cardíacos introduciendo el algoritmo de detección de tasa cardíaca deseado.		1
2.1	Protocolo de comunicación	La aplicación debe ser capaz de recibir los datos desde el panel de usuarios y devolver los resultados procesados.	El correcto funcionamiento del resto de requisitos es prueba suficiente para este.	1

2.2	Entrada de datos en tiempo real	La aplicación debe enviar y recibir los datos en tiempo real para emular las condiciones de funcionamiento de un SEC.	Comprobar que la aplicación es capaz de recibir datos paulatinamente y devolverlos.	1
2.2.1	Simular la entrada de datos por el ADC	La aplicación debe simular la adquisición de datos mediante un conversor analógico digital. Emulando el funcionamiento de un supervisor real.	Comprobar que el acceso a esos datos está controlado con un Timer, como si del ADC se tratase.	2
2.3	Interfaz para el algoritmo	La aplicación debe disponer de una interfaz, dentro de la cual se pueda encapsular el algoritmo de detección de tasa cardíaca que el usuario desee probar.	Implementar dos algoritmos diferentes sin necesidad de modificar nada fuera de la interfaz.	1
2.4	Resolución muestras	La aplicación debe ser capaz de modificar la resolución empleada para las muestras en la aplicación, ofreciendo al usuario cierto control sobre la cantidad de muestras simultáneas con las que su algoritmo podrá trabajar.	Comprobar que el máximo de muestras almacenadas en el dispositivo varían en función de la calidad seleccionada en el panel de control.	2

Tabla 2.1: Requisitos iniciales



## 2.3. Metodología

Para la realización de este proyecto se ha decidido emplear una metodología de desarrollo ágil, en lugar de una metodología más tradicional como podría ser un desarrollo en cascada. El carácter iterativo e incremental de este tipo de diseño permite reaccionar más rápidamente a los cambios de diseño o nuevas necesidades que se encuentren durante el desarrollo.

Uno de los marcos de trabajo ágiles más empleados en el desarrollo de software es el Scrum, que será empleado de una forma simplificada para el desarrollo de este proyecto. Dentro del Scrum nos encontramos tres figuras (Product owner, Scrum master y Equipo) sin embargo dado el carácter individual de este proyecto solo se emplearán dos. El product owner, representado por los tutores, que revisará el proyecto tras cada iteración y el desarrollador, encargado de organizar y llevar a cabo las iteraciones.

(TODO: Buscar figura de metodología ágil)

Una iteración es la unidad fundamental de trabajo en una metodología ágil, y consiste en la adición de un nuevo fragmento de código que añade o mejora alguna función del proyecto, tratando de generar valor en cada iteración. Cada una de estas unidades de trabajo consta de su propia planificación, análisis de requisitos, diseño, codificación y su propia documentación.

Dada la necesidad de agrupar toda la documentación generada del proyecto en un solo documento, en lugar de generar un fichero de documentación específico para cada iteración se clasificará cada iteración como un apartado del documento. El lector deberá tener en cuenta pues que los apartados referentes a las iteraciones del proyecto gozarán de una temporalidad y orden explícitos.



# Capítulo 3

## Implementación

### Contenido

<b>3.1 Iteración inicial: Planteamiento básico</b>	<b>18</b>
3.1.1 Resumen	18
3.1.2 Desarrollo	18
3.1.3 Diagramas	18
3.1.4 Conclusiones	19
<b>3.2 Primera iteración: El núcleo</b>	<b>20</b>
3.2.1 Resumen	20
3.2.2 Requisitos	20
3.2.3 Desarrollo	20
3.2.4 Pruebas	24
3.2.5 Conclusiones	25

## **3.1. Iteración inicial: Planteamiento básico**

### **3.1.1. Resumen**

Esta primera iteración consiste en la selección justificada del hardware y software en los que se basará todo el proyecto.

### **3.1.2. Desarrollo**

El proyecto consta de dos partes fundamentales. Primeramente un panel de control desde el cual un usuario podrá modificar los datos de entrada y observar los resultados del análisis de efectividad y eficiencia de los algoritmos. En segundo lugar debe existir un dispositivo el cual lleve a cabo las evaluaciones de los algoritmos de tiempo real.

Para el panel de control se ha decidido emplear QT 4.9.0 gracias al cual se evita tener que gestionar los componentes gráficos de la interfaz, pudiendo centrar el esfuerzo mayoritariamente en el tratamiento de los datos y las interacciones de usuario.

Para el hardware del dispositivo en tiempo real se ha escogido una Tiva C Series (TM4C123GXL) por sus numerosos modos de bajo consumo y familiaridad con el dispositivo.

### **3.1.3. Diagramas**

Una vez especificados el hardware y software a emplear se vuelve posible, y necesario, profundizar en los diagramas realizados anteriormente añadiendo cuestiones ya específicas de la solución escogida para el desarrollo del proyecto. Teniendo en cuenta las características de la tecnología escogida se ha realizado el siguiente diagrama (TODO: Figura x) detallando un concepto general del flujo de trabajo del proyecto.

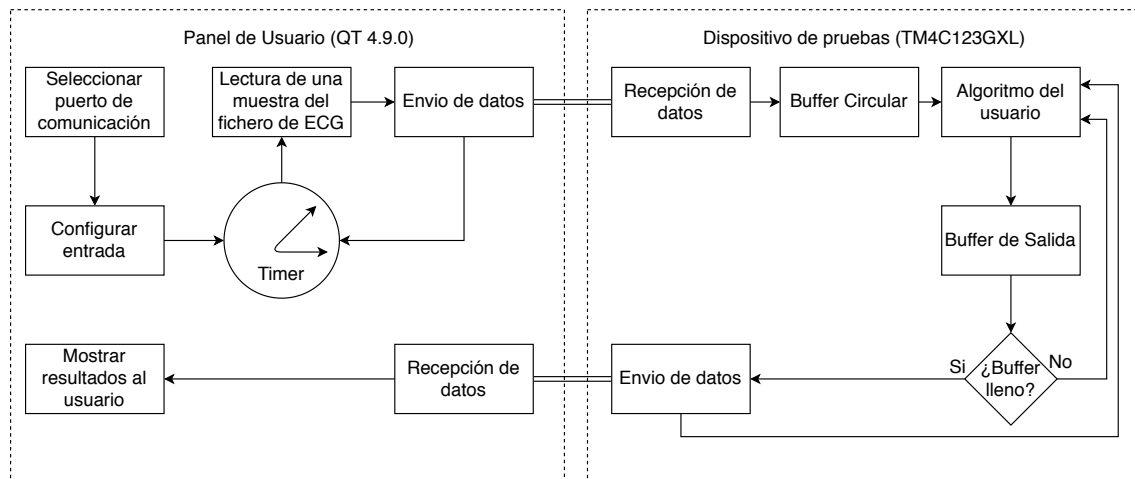


Figura 3.1: Diagrama de bloques general

La parte izquierda representa la funcionalidad mínima necesaria para el panel de control y la parte derecha el flujo de trabajo del hardware de testeo. En un comienzo el factor "Tiempo Real" vendrá determinado por un timer en el panel de control que enviará muestras del ECG de forma cíclica a la frecuencia de muestreo que este se encuentre registrado, simulando así los tiempos biológicos reales de dichas señales.

Los datos son almacenados en un buffer circular, el cual una vez lleno comenzará a sobrescribir los datos más antiguos. El algoritmo introducido deberá emplear únicamente los datos almacenados en el buffer circular para llevar a cabo el análisis. Como en la vuelta de los datos no es tan importante simular el tiempo real se ha decidido devolverlos al panel en forma de paquetes, saturando menos la vía de comunicaciones.

Una vez devuelta la información al panel de control este deberá mostrar de alguna forma los datos al usuario que lo controla.

### 3.1.4. Conclusiones

Decidida la estructura general del proyecto, puede comenzar la primera iteración de desarrollo, en la que se implementará la funcionalidad básica, sentando las bases para todo el desarrollo posterior.

## 3.2. Primera iteración: El núcleo

### 3.2.1. Resumen

Esta primera iteración está centrada en la codificación de la estructura básica de la aplicación, creando durante ella la infraestructura que servirá de apoyo para todas las funcionalidades que se implementen en las sucesivas iteraciones, llamado comúnmente como “Core application”.

Para dar esta iteración por finalizada, la aplicación del panel deberá ser capaz de enviar las muestras de ECG hacia la plataforma de muestreo y está a su vez ser capaz de recibirlas y enviarlas de vuelta al panel, verificando así la comunicación en el doble sentido.

### 3.2.2. Requisitos

Los requisitos a realizar en esta iteración son el 1.1, el 2.2 y el 2.1 de la tabla de requisitos iniciales. (Tabla 2.1) Estos requisitos dan lugar al siguiente desglose de tareas y subtareas:

1. Introducir señales de pruebas.
  - a) Lectura de un fichero de datos con muestras de ECG.
  - b) Envío de las lecturas a la plataforma de testeo a la frecuencia original de muestreo.
  - c) Interfaz de usuario básica para gestionar la lectura del fichero.
2. Entrada de datos en tiempo real.
  - a) Buscar la manera de realizar el análisis de manera continuada sin dejar de recibir nuevos datos en el proceso.
3. Protocolo de comunicación.
  - a) Definir el protocolo de comunicación a emplear.

### 3.2.3. Desarrollo

El primer paso fue la investigación de las bases de datos fisiológicos, concretamente bases de datos con muestras de ECG estandarizadas y ampliamente extendidas. Destacando la MIT-BIH Arrhythmia Database (TODO: citar <https://www.physionet.org/con>

```

function GetSampleAsText(sampleName,time)
%GETSAMPLEASTEXT(SampleName,SamplesAmount)
% sampleName == 'mitdb/1xx'
% time in minutes
[signal,Fs,tm]=rdsamp(sampleName,[],time * 21601);
output = [tm,signal].';

fileID = strrep(sampleName,'/','_');
file = fopen(strcat(fileID,'_',num2str(time),'.txt'),'wt');
fprintf(file,'%5.3f\n',Fs);
fprintf(file,'%5.4f,%3.4f,%3.4f\n',output);
fclose(file);
end

```

360
0.0000,-0.1450,-0.0650
0.0028,-0.1450,-0.0650
0.0056,-0.1450,-0.0650
0.0083,-0.1450,-0.0650
0.0111,-0.1450,-0.0650
0.0139,-0.1450,-0.0650
0.0167,-0.1450,-0.0650
0.0194,-0.1450,-0.0650
0.0222,-0.1200,-0.0800
0.0250,-0.1350,-0.0800
0.0278,-0.1450,-0.0850
0.0306,-0.1500,-0.0850
0.0333,-0.1600,-0.0750
0.0361,-0.1550,-0.0700
0.0389,-0.1600,-0.0700

Figura 3.2: Función de Matlab para extraer los datos del repositorio y formatearlos como texto

Accediendo a los datos a través del repositorio de Physionet, haciendo uso de su propia API para descargar los datos directamente desde matlab. (TODO: Citas) De este modo, con un simple script, es posible convertir cualquier muestra de datos procedente de dicho repositorio en un fichero de texto estandarizado para la entrada de datos del entorno de pruebas.

El siguiente paso consiste en leer dichos ficheros de texto desde el panel de usuario y preparar los datos para enviarlos al dispositivo de pruebas. Este proceso tiene dos puntos claves, la lectura debe ser periódica y su frecuencia debe ser igual que la frecuencia de muestreo de las muestras. Para lograr este cometido se ha desarrollado un script siguiendo la estructura del diagrama 3.3.

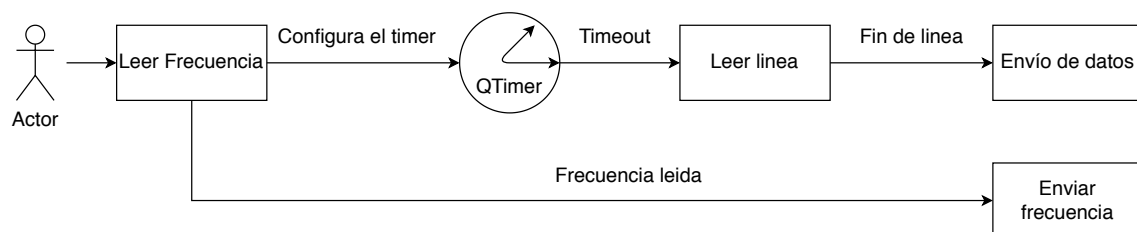


Figura 3.3: Diagrama de lectura de ficheros simplificado

Para sincronizar las partes del script y garantizar la compatibilidad y sincronismo con futuras funcionalidades del panel de control se hace uso de los SLOT y las SIGNAL de Qt. Estos elementos son el pilar fundamental de las aplicaciones desarrolladas con Qt ya que actúan como canales de mensajes, permitiendo

la sincronización de todas las funciones de la aplicación sin caer en fuertes restricciones por acoplamiento de las diferentes funcionalidades.

Como se puede ver en el diagrama 3.4 se han añadido también diferentes entradas para que el usuario interactúe. Estas son una rueda selectora para configurar la frecuencia de forma manual, un botón para realizar la lectura de la frecuencia de muestreo directamente desde el fichero y un tercero que daría comienzo a la prueba. En la figura 3.5 se puede ver la disposición de estos elementos.

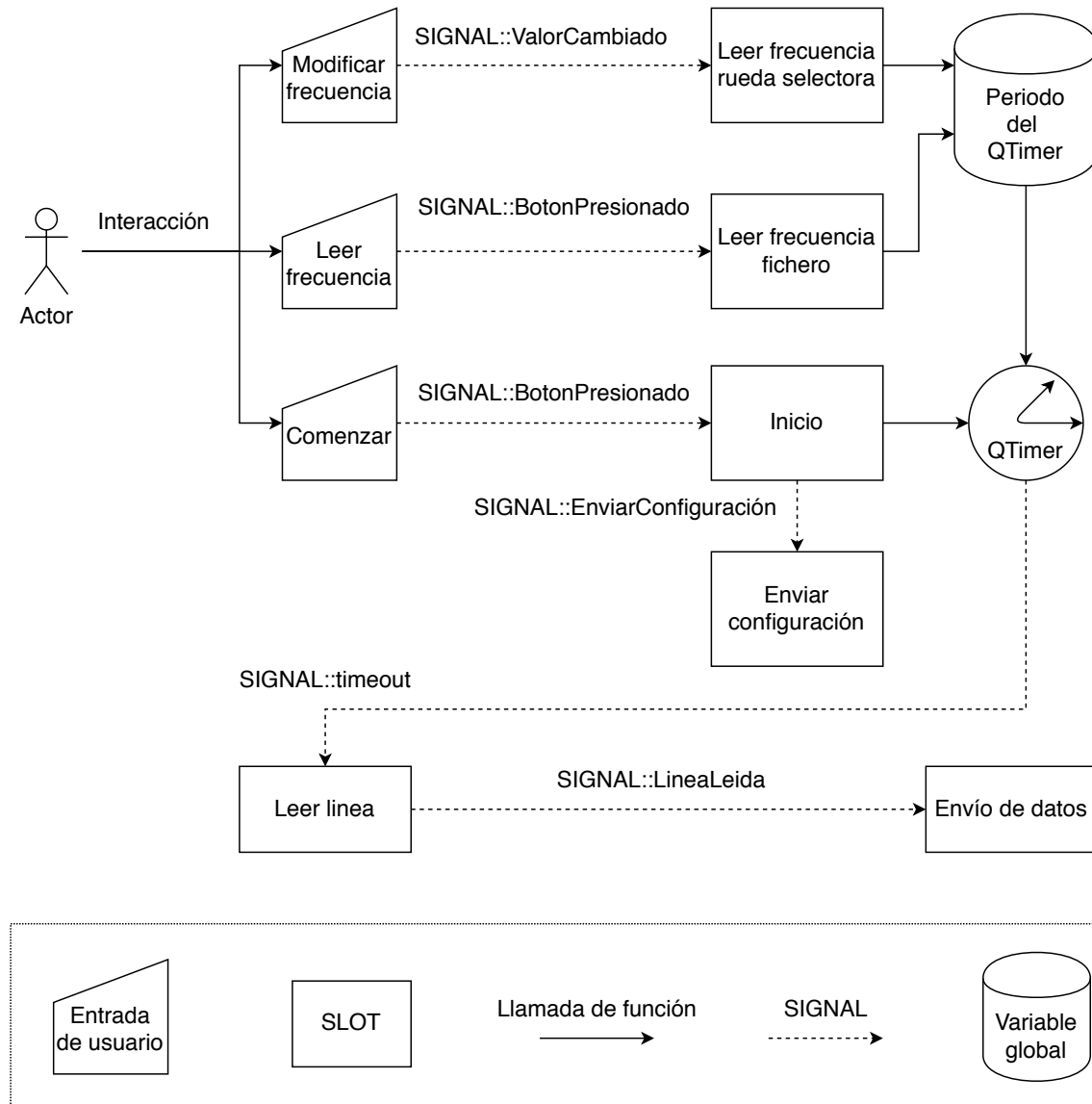


Figura 3.4: Diagrama de lectura de ficheros en QT



Antes de continuar con la información sobre el envío de los datos, es necesario detenerse en la gestión del tiempo real y el sincronismo dentro del dispositivo de pruebas, concretamente la Tiva (TM4C123GXL). Para esta finalidad se ha hecho uso de un sistema operativo de tiempo real de software libre llamado FREERTOS (TODO: Cita). Gracias a él se pueden gestionar y sincronizar tareas asignándoles diferentes prioridades, solucionando el problema de tener que gestionar las comunicaciones a la vez que se realizan las pruebas de rendimiento, ganando en el proceso además mayor independencia entre las tareas reduciendo acoplamientos.

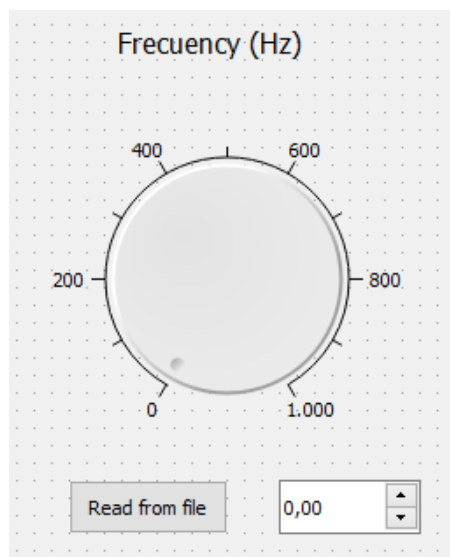


Figura 3.5: Selector de frecuencias y lectura del fichero

Para la comunicación entre el dispositivo de pruebas y el panel de usuario se hace uso de un puerto serie. Para garantizar la correcta transmisión de los datos se emplean dos mecanismos muy extendidos, checksum y byte stuffing. Pudiendo representar el protocolo en el diagrama 3.6. Cada salto de línea representa cada nueva etapa del protocolo de envío. El ejemplo expuesto contiene un byte de datos que coincide con el byte de final de trama, representado como “END”, a fin de ejemplificar el funcionamiento del método de stuffing escogido.

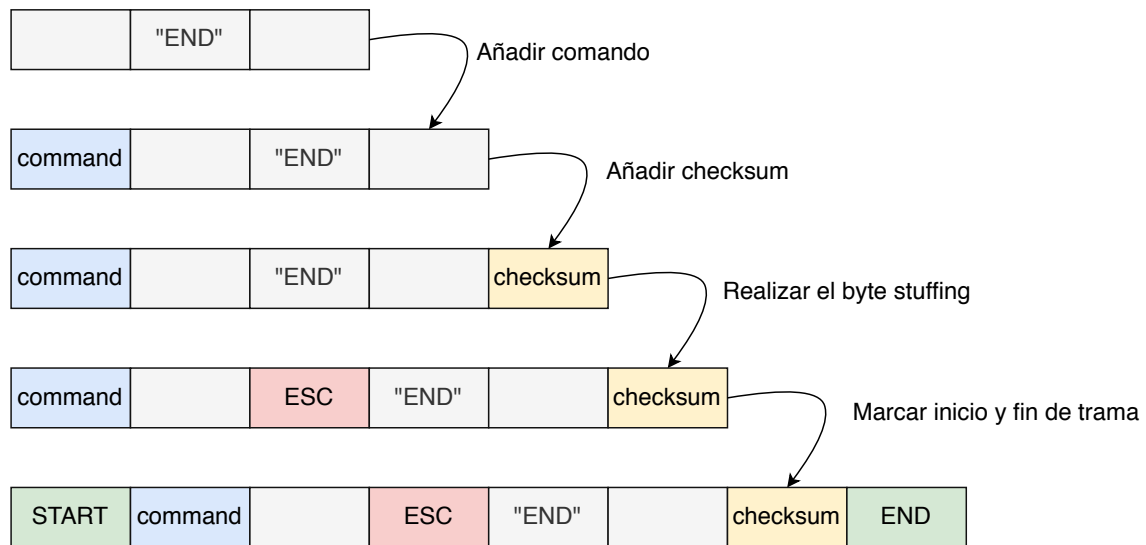


Figura 3.6: Proceso construcción de trama

### 3.2.4. Pruebas

Para testear la lectura de ficheros en Qt se han implementado una serie de tests unitarios con el objetivo de verificar las lecturas, pero además, de verificar que ante entradas de datos erróneas como ficheros mal formateados el sistema reacciona de forma correcta.

Los casos sometidos a test han sido:

- Lectura correcta de la frecuencia desde el fichero
- Lectura incorrecta de la frecuencia desde el fichero
- Lectura correcta de una línea de datos
- Lectura incorrecta de una línea de datos

Para el protocolo el proceso ha sido similar, si es capaces de codificar y decodificar un mensaje, donde entre en juego la necesidad de emplear el bit Stuffing, entonces debe ser capaz de codificar y decodificar cualquier mensaje.

Para comprobar la transmisión de los datos entre el panel y el dispositivo de prueba se ha optado por un test de carácter más empírico. Si se puede enviar en tiempo real y recibir de vuelta la señal deseada sin que los datos se distorsionen, se puede concluir que el sistema de transmisión funciona correctamente.

(TODO: Imagen de los dos ECG)

### 3.2.5. Conclusiones

Tras la realización de esta iteración quedan sentadas las bases para el correcto desarrollo del proyecto. A su vez se hizo presente la necesidad de implementar una nueva funcionalidad.

La inclusión de un mecanismo para que el usuario pueda seleccionar de manera fácil e intuitiva la muestra de señal que desea emplear para las pruebas. Esta funcionalidad ha sido incluida en la tabla de requisitos finales con el código 1.5. de la sección de conclusiones finales. (TODO: Referencia a la tabla final)

De la realización de los test también se observó la necesidad de incluir algún tipo de mecanismo de mensajes para notificar a los usuarios los errores o advertencias que se produzcan en el panel de control. Esta funcionalidad se ha incluido en la tabla de requisitos finales con el código 1.6. de la sección de conclusiones finales (TODO: Referencia a la tabla)



# Conclusiones y líneas futuras

Después de todo el desarrollo del proyecto, es pertinente hacer una valoración final del mismo, respecto a los resultados obtenidos, las expectativas o el resultado de la experiencia acumulada.

Esta sección es indispensable y en ella se ha de reflejar, lo más claramente posible, las aportaciones del trabajo con unas conclusiones finales.

Además, considerando también el estado de la técnica, se deben indicar las posibles líneas futuras de trabajo, proponer otros puntos de vista o cualquier otra sugerencia como postámbulo del presente trabajo, para ser considerada por el lector o el tribunal evaluador.



# Apéndice A

## Manual de uso

Contenido	
A.1 Primera sección . . . . .	29

### A.1. Primera sección





# Bibliografía

- [1] Adriana Mas Casals and José M.<sup>a</sup> Lobos Bejarano. De las ondas del ecg a la patología ¿a qué puede corresponder cada alteración de una onda, complejo o intervalo? *AMF*, 10:597–600, 11 2012.
- [2] Joanna Gawłowska and Jerzy Krzysztof Wranicz. Norman j. “jeff” holter (1914-1983). *Cardiology Journal*, 16(4):386–387, 2009.
- [3] Jan Gierałtowski, Kamil Ciuchciński, Iga Grzegorczyk, Katarzyna Stępień, Mateusz Soliński, and Piotr Podziemski. Rs slope detection algorithm for extraction of heart rate from noisy, multimodal recordings. *Physiological measurement*, 36:1743–1761, 08 2015.
- [4] Raquel Gutiérrez Rivas, Juan García, William Marnane, and Álvaro Hernández. Novel real-time low-complexity qrs complex detector based on adaptive thresholding. *Sensors Journal, IEEE*, 15:6036–6043, 10 2015.
- [5] Dennis Kasper, Anthony Fauci, Stephen Hauser, Dan Longo, J. Larry Jameson, and Joseph Loscalzo. *Harrison. Principios de Medicina Interna*. McGraw-Hill Education, 2018.
- [6] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230236, 1985.
- [7] Zahia Zidelmal, Ahmed Amirou, Mourad Adnane, and Adel Belouchrani. Qrs detection based on wavelet coefficients. *Computer methods and programs in biomedicine*, 107:490–6, 01 2012.



