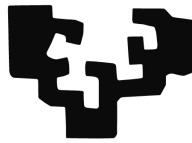


eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Grado en Ingeniería Informática
Computación

Memoria del Proyecto Fin de Grado

Un electrocardiógrafo inteligente de bajo coste

Autor

Alexander Mariel Mendiguren

Directores

Iñaki Inza Cano

Carlos Amuchastegui Uriarte



©2014 por Alexander Mariel. “Un electrocardiógrafo inteligente de bajo coste”.

Este documento está sujeto a la licencia *Creative Commons Attribution-ShareAlike 4.0 International*. Puede consultarse un resumen de la licencia y su texto legal en <http://creativecommons.org/licenses/by-sa/4.0/>.

En primer lugar quisiera dar las gracias a mis directores, Iñaki Inza y Carlos Amuchastegui por confiar en este proyecto y ayudarme a llevarlo adelante.

También quisiera agradecer a la empresa Mirakonta el apoyo en el diseño y fabricación de uno de los componentes hardware de este proyecto.

Gracias a mis padres, por apoyarme a lo largo de mi carrera y permitirme pasar dos años de ella en San Sebastián, para poder estudiar aquello que más deseaba.

Agradezco el apoyo mis más cercanos amigos, a los que he torturado con complicados problemas surgidos durante el proyecto, Jorge, Iñaki, Carlos y Mikel. Quisiera agradecer especialmente a Mikel su compromiso en este proyecto y todo el tiempo que ha pasado transmitiéndome una pequeña parte de sus vastos conocimientos médicos.

No me olvido de todos aquellos y aquellas que he tenido el placer de conocer a lo largo de mi vida universitaria, en Bilbao y en San Sebastián, estudiantes y profesores.

Vosotros habéis dado vida a mis cuatro años de Ingeniería Informática.

Quisiera mencionar a tres profesores en particular, que me han prestado consejo cuando lo he necesitado y me han ayudado a llevar a cabo tareas que habrían sido imposibles sin ellos. Aitziber Atutxa, Jose Miguel Blanco y Txelo Ruiz. Gracias.

También deseo recordar a todas aquellas personas que he conocido a través de ITSAS. Jamás pensé encontrar tanta gente increíble en el mundo de las tecnologías libres.

Especialmente Aitor, Luis, Iker y Olatz.

Extiendo mis agradecimientos internacionalmente a Jiří Fajmon. La ayuda proporcionada para hacer funcionar las pruebas con amplificadores, cuando este proyecto aún era un concepto, es incalculable. Y las explicaciones sobre electrónica prestadas fueron impecables.

Por último te agradezco a ti, lector o lectora, que consultes esta memoria. Probablemente no la estudiarás en su totalidad, pero ya has demostrado tu interés por un proyecto el cual significa mucho para mi.

Resumen

A lo largo de este proyecto se ha desarrollado un sistema de bajo coste para la toma de electrocardiogramas y posterior visualización de los mismos en un dispositivo Android. Además se ha creado un módulo inteligente capaz de realizar un diagnóstico de manera automática y razonada sobre los datos recogidos. El proyecto se ha realizado principalmente sobre tecnologías abiertas: Arduino como componente central del sistema electrónico, Android para visualizar datos en una plataforma móvil y CLIPS como motor sobre el cual se ha desarrollado el sistema experto que realiza el diagnóstico.

Abstract

In this project a low-cost system has been developed for electrocardiogram reading and visualization in an Android app. An artificial intelligence module was also created to generate an automated and reasoned diagnostic on the received data. The project has been developed around open source technologies: Arduino as the base for the electronic system, Android to access the information on a portable device and CLIPS as the engine for the expert system which generates the diagnostic.

Laburpena

Proiektu honetan zehar, elektrokardiogramak lortzeko eta ostean Android gailu batean bistartzeko, kostu baxuko sistema bat garatu da. Horrez gain, jasotako datuetatik, diagnostiko automatikoa eta arrazoitua egiteko gai den modulu adimentsu bat sortu da. Nagusiki, proiektu hau teknologia irekiekin diseinatu eta garatu da: Sistema elektronikoaren osagai nagusi bezala Arduinoa erabili da, gailu mugikor batean datuak bistartzeko Android sistema erabili da eta, diagnostikoak burutzeko, CLIPS motorean oinarritutako sistema-aditua sortu da.

Índice

Índice	XI
Listado de figuras	XVII
Listado de tablas	XIX
1. Introducción	1
2. Conocimientos del entorno	5
2.1. Conocimientos biológicos básicos sobre el corazón humano	5
2.1.1. Función del corazón	5
2.1.2. Anatomía del corazón	6
2.1.3. Importancia del corazón	7
2.2. Principios de la electrocardiografía	7
2.2.1. Actividad eléctrica del corazón	7
2.2.2. Electrocardiografía	8
2.2.3. Electrocardiógrafos	11
2.3. Diagnóstico de patologías cardiovasculares basado en electrocardiogramas .	12
2.3.1. Interpretación de electrocardiogramas	12
2.3.2. Diagnóstico basado en ECGs	13
I Desarrollo del proyecto	15
3. Electrónica	17
3.1. Arduino	17
3.1.1. Open hardware	18
3.1.2. Conversor analógico-digital	18

3.1.3.	Comunicación	19
3.1.4.	Alimentación	21
3.2.	Placa de adquisición de datos	22
3.2.1.	Ruido eléctrico	22
3.2.2.	Amplificación	24
3.2.3.	Placas de adquisición de datos	26
3.2.4.	Filtrado del ruido	30
3.2.5.	Elección final	32
4.	Software, app	35
4.1.	Elección de la plataforma	35
4.1.1.	Hardware	35
4.1.2.	Software	36
4.2.	Flujo de ejecución de la aplicación	36
4.3.	Aspectos destacados de la implementación	38
4.3.1.	Filtrado en tiempo real	38
4.3.2.	Comunicación Bluetooth	39
4.3.3.	Flujo de trabajo con AsyncTask	40
4.3.4.	Interfaz gráfica	40
5.	Software, extracción de características del ECG	43
5.1.	Algoritmos desarrollados	44
5.1.1.	Codificación de los datos	44
5.1.2.	Detección de picos R	45
5.1.3.	Detección de complejos QRS	45
5.1.4.	Detección de ondas P	46
5.1.5.	Detección de ondas T	47
5.1.6.	Detección de otros parámetros del ECG	48
5.1.7.	Integración de los algoritmos	50
5.2.	Aseguramiento de la calidad	51
5.2.1.	Ejecución sobre electrocardiogramas de Physiobank	51
5.2.2.	Ejecución sobre electrocardiogramas de la placa Olimex	52
5.2.3.	Ejecución sobre electrocardiogramas sintéticos	54
5.3.	Selección manual de regiones	54

6. Sistema experto	57
6.1. Motores para sistemas expertos en Android	57
6.1.1. CLIPS	58
6.1.2. Jess	58
6.1.3. Elección final	59
6.2. Integración de CLIPS en Android	59
6.2.1. Limitaciones de CLIPS bajo Android	59
6.3. Estructura del sistema experto implementado	60
6.3.1. Plantillas	62
6.3.2. Hechos	63
6.3.3. Módulos	64
6.3.4. Reglas	64
6.3.5. Características especiales	66
II Expansión y mejoras	67
7. Electrónica	69
7.1. Mejora del filtrado por software	69
7.2. Revaluación de las placas	69
7.3. Ampliación a seis derivaciones	70
7.3.1. Usando placas Olimex ECG Shield	71
7.3.2. Mediante placas de fabricación propia usando el chip AD8232	71
7.3.3. Nuevos problemas al hallar seis derivaciones	71
7.4. Mejora del protocolo de comunicación	72
7.5. Abaratamiento de costes de adquisición	72
8. Software, app	75
8.1. Ampliación a seis derivaciones	75
8.2. Interacción con el ECG	76
8.2.1. Pausa del electrocardiograma	76
8.2.2. Modificación del tamaño del electrocardiograma	77
8.2.3. Visualización 3D del diagnóstico	77
8.3. Adaptación a estándares médicos	78
8.4. Exportación del electrocardiograma	78
8.5. Integración con variantes del hardware	79

8.6. Refactorización del código	80
9. Software, extracción de características del ECG	81
9.1. Mejor detección de ondas P y T	81
9.1.1. Comienzo y final de la onda P	81
9.1.2. Ondas P y T negativas y difásicas	82
9.2. Análisis del ECG a alto nivel	82
9.3. Extracción de características de ECGs de personas que sufren patologías cardiovasculares	83
9.4. Ampliación a seis derivaciones	83
9.4.1. Nueva información accesible	83
9.4.2. Mejora de resultados por combinación de derivaciones	84
9.5. Creación de un plan de calidad para evaluar el sistema	85
9.5.1. Etiquetado de ECGs	85
9.5.2. Lanzamiento del sistema de extracción de características	85
9.5.3. Formas de representar los resultados	86
10. Sistema experto	87
10.1. Un sistema basado en lógica difusa	87
10.1.1. Motores de lógica difusa existentes	87
10.1.2. Proceso de adaptación del conocimiento al nuevo sistema	88
10.2. Combinación con otros sistemas de diagnóstico	88
III Otros aspectos del proyecto	91
11. Conclusiones	93
11.1. Conclusiones generales del proyecto	93
11.2. Manual de usuario	95
11.3. Coste del prototipo final	95
11.4. Licencia	97
11.5. Código del proyecto	97
12. Gestión del proyecto	99
12.1. Gestión de tareas	99
12.2. Gestión del alcance	101
12.3. Gestión del tiempo	102

Índice	xv
12.4. Gestión de costes	102
Bibliografía y referencias	105
Anexo A. Diagramas eléctricos de las placas de adquisición de datos	109
Anexo B. Manual de usuario	115

Listado de figuras

2.1. Diagrama del corazón humano	6
2.2. Sistema nervioso del corazón	8
2.3. Camino del impulso eléctrico	8
2.4. Posiciones de los electrodos en el paciente	9
2.5. Electrocardiograma etiquetado en la derivación I	10
2.6. Electrocardiograma profesional completo	11
2.7. ECG en derivación I completamente etiquetado	12
3.1. ECG sin conexión a la red eléctrica con ruido de 50Hz	24
3.2. ECG con ruido de origen AC	24
3.3. ECG con ruido causado por respiración	25
3.4. Etapa de amplificación diferencial	25
3.5. Eliminación del ruido por filtrado diferencial	26
3.6. Diagrama de un amplificador de instrumentación	27
3.7. ECG de la placa AD8232Eval	29
3.8. ECG de la placa Olimex con pulseras	30
3.9. ECG de la placa Olimex con pinzas	30
4.1. Flujo de ejecución principal de la aplicación	37
4.2. Flujos de ejecución parciales de la aplicación	37
5.1. Onda P ampliada con vectores	48
5.2. Múltiples electrocardiogramas con puntos característicos	53
5.3. Captura de pantalla de Region-Select	55
9.1. Representación del sistema hexaxial	84
12.1. Esquema de descomposición de trabajo	100

A.1. Circuito interno del chip AD8232	111
A.2. Diagrama de la placa de evaluación del chip AD8232 creada por Analog Devices	112
A.3. Diagrama de la placa de diseño propio basado en el chip AD8232	113
A.4. Diagrama de la placa Olimex ECG Shield	114

Listado de tablas

3.1. Ventajas y desventajas de la alimentación por red eléctrica	21
3.2. Ventajas y desventajas de la alimentación por baterías	22
6.1. Módulos creados para el sistema experto	64
9.1. Ejemplo de matriz de confusión para la evaluación del sistema de extracción de características	86
11.1. Coste de los componentes primarios del prototipo en distribuidores varios .	96
12.1. Coste humano del proyecto	103

Capítulo 1

Introducción

El objetivo de este proyecto consiste en crear un electrocardiógrafo para tomar electrocardiogramas, grabaciones de la actividad eléctrica del corazón, de manera sencilla. Los electrocardiogramas podrán visualizarse en una aplicación para dispositivos Android. La app será capaz de realizar un diagnóstico, de manera automática, sobre el estado del corazón del sujeto de quien se tome el electrocardiograma. En caso de que el corazón sufra alguna alteración, se listarán las posibles patologías cardiovasculares que el paciente pueda sufrir.

El coste del electrocardiógrafo deberá ser reducido, para ser asequible a los colectivos de estudiantes de medicina o aficionados a la electrónica y biología. Los electrocardiogramas serán tomados a partir de cuatro electrodos contando así con las derivaciones I, II, III, aVR, aVL y aVF. El diagnóstico orientativo automatizado se realizará mediante técnicas de minería de datos o de sistemas expertos.

Motivación

Para los estudiantes del campo de la informática no resulta complicado practicar y mejorar conocimientos sobre un área, e incluso crear nuevos productos o servicios a partir de ellos. Esto es posible gracias a unos requisitos reducidos: un ordenador, acceso a la red eléctrica y acceso a Internet. Los estudiantes de otros campos no tienen esta oportunidad, pero la necesidad de llevar a la práctica los conocimientos que adquieren a lo largo de su carrera es igualmente crítica.

Uno de estos casos es el de los futuros licenciados en medicina, en especial, aquellos interesados en cardiología. Los conocimientos de lectura e interpretación de electrocardiogramas tienen un fuerte componente heurístico únicamente adquirible y mejorable a través

de la lectura de éstos. Lamentablemente, las máquinas usadas para tomar electrocardiogramas no son asequibles económicamente para un estudiante, y en el ámbito universitario no se les permite usarlos libremente. Por estas razones resulta muy difícil para un futuro licenciado en medicina mejorar su formación en electrocardiografía por cuenta propia.

Este proyecto pretende acabar con este problema.

Alcance

El alcance del proyecto se divide en tres grandes sistemas que trabajarán conjuntamente para generar el resultado final. Estos sistemas podrán ser desarrollados en serie, según se presentan, o de manera paralela.

- Un sistema de toma de bioseñales cardíacas que cumpla los siguientes requisitos:
 - Toma de datos de cuatro electrodos
 - Conversión de datos a formato digital
 - Codificación y envío inalámbrico de datos en tiempo real

- Una app para dispositivos Android que cumpla los siguientes requisitos:
 - Recepción de datos de forma inalámbrica y decodificación de los mismos en tiempo real
 - Visualización de los datos en tiempo real
 - Realización de un diagnóstico orientativo a partir de los datos recibidos
 - Compatibilidad garantizada con la tablet Nexus 7, modelo 2012

- Un sistema de diagnóstico automatizado que cumpla los siguientes requisitos:
 - Integrado en una app para dispositivos Android
 - Capacidad de realizar un diagnóstico en menos de un minuto
 - La máxima precisión posible

Además el sistema final deberá poder ser adquirido y ensamblado por el usuario por un **precio menor o igual a 50 euros**, excluyendo el valor de los componentes desechables.

Estructura de la memoria

Este documento está estructurado en cuatro grandes apartados.

Al comienzo de la memoria pueden encontrarse el índice, listado de figuras y listado de tablas. Los dos capítulos contenidos en el primer apartado presentan una breve introducción al proyecto y los conocimientos del área de la electrocardiografía necesarios para comprender correctamente esta memoria.

El segundo apartado recoge la información correspondiente al desarrollo del proyecto. Éste se ha dividido en cuatro capítulos, y cada uno presenta una de las cuatro grandes fases del proyecto.

En el tercer apartado se reúnen todos los aspectos de mejora y expansión del proyecto. Algunos de estos aspectos se encontraban en la planificación inicial y no pudieron ser implementados. Otros son posibilidades que se prestan a ser incluidas en futuras iteraciones del proyecto de manera natural.

El último apartado contiene dos capítulos: el primero describe las conclusiones del proyecto y el segundo algunos aspectos de la gestión del mismo. Al final del documento pueden encontrarse la bibliografía y los anexos.

Capítulo 2

Conocimientos del entorno

En este capítulo se resumen los conocimientos básicos sobre el entorno de este proyecto que deben conocerse para comprender esta memoria y el funcionamiento del prototipo creado. Los temas aquí descritos han sido ya ampliamente estudiados y existe gran cantidad de información accesible si se desea una formación más detallada. La información contenida a lo largo de los siguientes apartados se encuentra fuertemente resumida.

2.1. Conocimientos biológicos básicos sobre el corazón humano

A lo largo de las próximas subsecciones se presentan los conocimientos más básicos para comprender la funcionalidad del corazón humano, su estructura y la importancia de dicho órgano.

2.1.1. Función del corazón

Los tejidos del cuerpo humano precisan oxígeno y otros elementos esenciales para funcionar correctamente. El oxígeno es introducido en el organismo humano a través de los pulmones, pero debe ser transportado desde dicho lugar a los receptores del oxígeno. Para ello existen los eritrocitos o glóbulos rojos. Estos elementos, compuestos principalmente por hemoglobina, son capaces de anexionarse las moléculas de oxígeno existentes en el aire disponible en los pulmones y liberar dichas moléculas en los tejidos. Los eritrocitos se encuentran en la sangre.

El corazón es un órgano muscular cuya función es el bombeo de sangre, y por tanto eritrocitos, a través de los vasos sanguíneos del cuerpo humano. A través de estos vasos y

capilares los eritrocitos llegan a todos los lugares del cuerpo, alimentando con oxígeno las células que forman los tejidos.

2.1.2. Anatomía del corazón

El corazón es hueco en su interior y se divide en cuatro cámaras: dos superiores, las aurículas, y dos inferiores, los ventrículos. Las aurículas tienen la función de recepción de sangre del sistema sanguíneo y los ventrículos la función de expulsión. El corazón también se divide en dos mitades verticalmente: la mitad izquierda y la mitad derecha, cada una compuesta de una aurícula y un ventrículo. Según esta división vertical, el corazón puede ser interpretado como un mecanismo de dos bombas paralelas que trabajan conjuntamente.

La mitad derecha recibe sangre desoxigenada que ha pasado por los tejidos del cuerpo humano y la bombea hacia los pulmones: este circuito se conoce como circulación pulmonar. La mitad izquierda recibe la sangre oxigenada proveniente de los pulmones y la bombea hacia los tejidos del cuerpo humano a través del sistema sanguíneo.

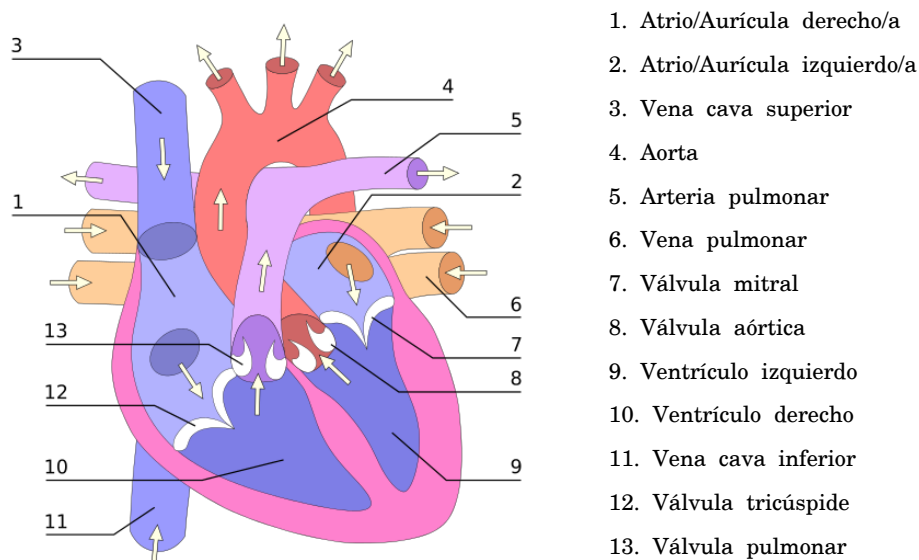


Fig. 2.1 Diagrama del corazón humano

Fuente: [commons.wikimedia.org, Ss181292, Indolences.](https://commons.wikimedia.org/wiki/File:Ss181292)

El bombeo sucede mediante la contracción (sístole) y relajación (diástole) asíncrona entre las aurículas y ventrículos. Las aurículas, al relajarse, succionan la sangre de los conductos que las alimentan: las venas cava superior e inferior para la aurícula derecha y las venas pulmonares para la aurícula izquierda. Después las aurículas se contraen y desplazan su contenido a los ventrículos a través de las válvulas mitral y tricúspide. Al contraerse los

ventrículos, estos impulsan la sangre a las arterias pulmonares (ventrículo derecho) y a la arteria aorta (ventrículo izquierdo).

2.1.3. Importancia del corazón

Un fallo en este órgano puede tener consecuencias fatales. Sin oxígeno, los tejidos del cuerpo humano sufrirán una isquemia, en esta situación puede que las células dejen de cumplir sus funciones debido a la incapacidad de mantener su metabolismo. Si la situación se prolonga, las células podrían morir dando lugar a una necrosis. Las consecuencias de las isquemias y necrosis son amplias y los efectos de estas varían según la zona en la que se den.

Como puede verse en [22] las patologías cardiovasculares representaron la segunda causa de muerte en España. Dicha tendencia continua manteniendo su puesto hoy en día en los países del primer mundo.

2.2. Principios de la electrocardiografía

La electrocardiografía es una técnica usada para tomar información sobre el corazón de un sujeto. Sus aplicaciones en la medicina son de extrema importancia, ya que permite conocer la salud cardiovascular de un paciente.

En esta sección se expone cual es la base física que soporta la electrocardiografía. También se indica cómo se obtiene un electrocardiograma o ECG y cómo se interpretan las ondas leídas. Por último se presenta información relativa a los electrocardiógrafos, las máquinas usadas para tomar electrocardiogramas.

2.2.1. Actividad eléctrica del corazón

El corazón, como ya se ha indicado anteriormente, es un músculo. Como tal, utiliza complejos procesos químicos para moverse. Estos procesos químicos conllevan la liberación de iones, partículas con cargas eléctricas no neutras. Los iones causan la despolarización del tejido cardíaco, desencadenando la contracción de partes del mismo. Más adelante los músculos se repolarizan y relajan volviendo a su situación inicial. El movimiento de los iones causa unas débiles señales eléctricas: éstas pueden verse como vectores de potencial que recorren el corazón.

La despolarización y repolarización del corazón anteriormente descritos no suceden de manera aleatoria, sino que están claramente dirigidos por unas vías nerviosas y un mar-

capasos natural propio del corazón. Los caminos que siguen los impulsos eléctricos en el corazón se pueden observar en las Figuras 2.2 y 2.3.

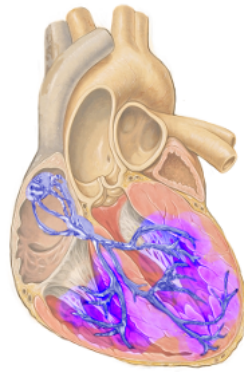


Fig. 2.2 Sistema nervioso del corazón, vías nerviosas en azul

Fuente: *commons.wikimedia.org, Trlkly.*

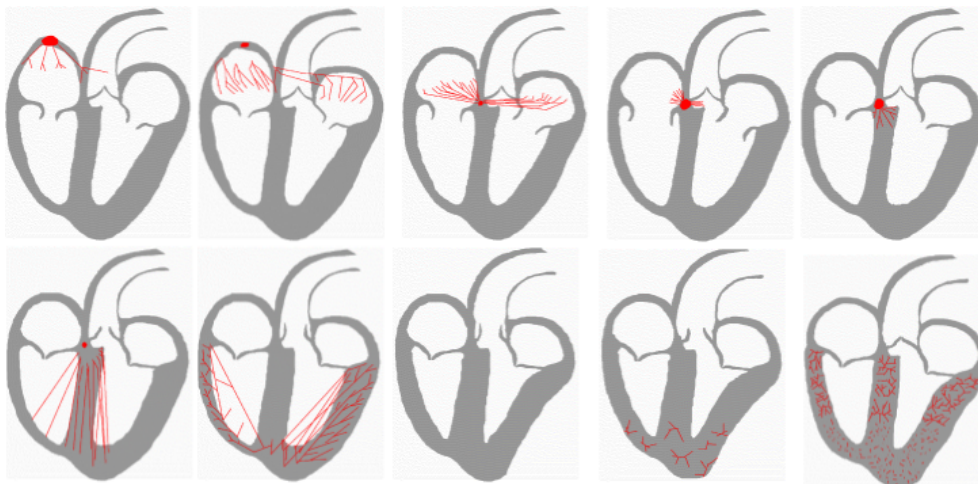


Fig. 2.3 Camino del impulso eléctrico. En las primeras cuatro imágenes se observa la descarga del nodo sinusal y la consecuente contracción de las aurículas. A partir de la quinta imagen se observa la descarga del nodo atrioventricular, contracción de los ventrículos y relajación posterior de los mismos.

Fuente: *commons.wikimedia.org, Kalumet.*

2.2.2. Electrocardiografía

La electrocardiografía se basa en la lectura no invasiva de los vectores de potencial que genera el corazón. Para ello se adhieren unos electrodos a la piel del sujeto del cuál se quiera

tomar un electrocardiograma en lugares específicos. Cada combinación de lugares genera un *punto de vista* diferente desde el cual se pueden observar los vectores de potencial. Cuantos más electrodos se sitúan sobre el sujeto, mayor será la información que se pueda tomar sobre su corazón.

Los cuatro electrodos principales son nombrados mediante los siguientes acrónimos:

- LA: Left Arm, brazo izquierdo
- RA: Right Arm, brazo derecho
- LL: Left Leg, pierna izquierda
- RL: Right Leg, pierna derecha

Los electrodos LA y RA se suelen situar en las muñecas u hombros. Los electrodos LL y RL se sitúan en los tobillos o bajo la cintura. Los electrocardiogramas más completos se generan a partir de estos y otros seis electrodos, llamados V1 a V6. Éstos se sitúan en el pecho. En la Figura 2.4 puede verse un ejemplo de colocación de electrodos. Además de por nombre, los electrodos también suelen estar definidos por un código de colores.

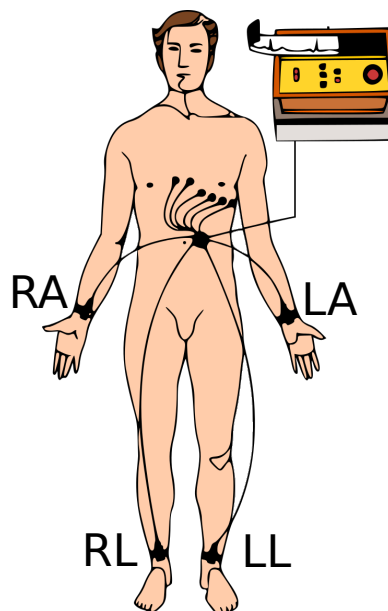


Fig. 2.4 Posiciones de los electrodos en el paciente. Junto a cada electrodo se sitúa su nombre según el sistema estadounidense. Los electrodos del pecho se nombran V1 a V6 de derecha a izquierda desde la perspectiva del sujeto.

Cada *punto de vista* generado por la combinación de electrodos es llamado derivación. La derivación más simple se halla restando la lectura del potencial que se obtiene en el brazo derecho a la lectura del potencial hallado en el brazo izquierdo. Esta derivación se representa con el número romano I. Si se pudiera visualizar el vector, esta derivación sería el *punto de vista* equivalente a un observador situado en la mano izquierda de un sujeto con los brazos en cruz, mirando a través de él hacia la mano derecha. En la Figura 2.5 se muestran los valores de potencial que visualizaría el observador ficticio a lo largo de un pulso del corazón:

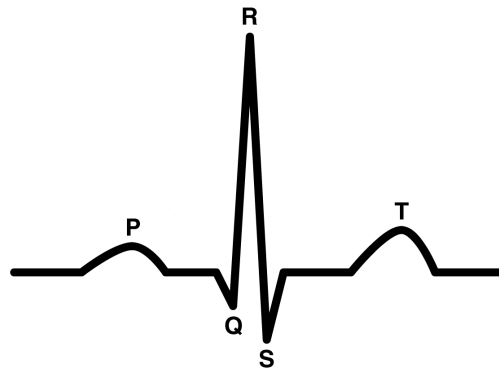


Fig. 2.5 Electrocardiograma etiquetado en la derivación I

Fuente: mublog.marymount.edu

Como ya se ha indicado antes, si el número de electrodos aumenta, también lo hará el número de derivaciones que se puedan calcular, y junto a ellas la información que se obtiene del corazón. Cada nueva derivación aporta un nuevo *punto de vista* sobre el corazón, pudiendo ver éste desde debajo hacia arriba (eje coronal), desde atrás hacia delante (eje axial), etcétera. Como referencia se muestra en la Figura 2.6 un electrocardiograma completo de un sujeto.

Los electrocardiogramas se presentan en un papel con una cuadrícula milimétrica. Sobre esta cuadrícula se superpone otra con cuadrados de cinco por cinco milímetros. Cada cuadrado de 1 milímetro representa 40 milisegundos en el eje de las abscisas (eje horizontal) y 1 milivoltio en el eje de las ordenadas (eje vertical). En la bibliografía y campo médico se hace un uso extensivo de las dimensiones geométricas, dejando de lado las magnitudes físicas que representan.

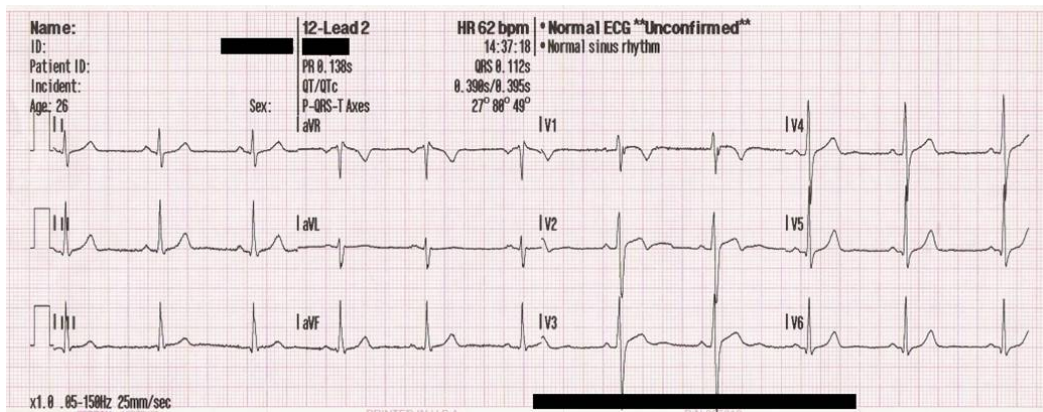


Fig. 2.6 Electrocardiograma profesional completo.

Pueden observarse las derivaciones I, II, III, aVL, aVR, aVF y V1 a V6.

Fuente: commons.wikimedia.org

2.2.3. Electrocardiógrafos

Las máquinas que capturan electrocardiogramas son llamadas electrocardiógrafos. Dichos aparatos consisten en un aparato principal, del cual provienen unos cables para conectar los electrodos. Dicho aparato suele contener una pantalla para visualizar el electrocardiograma o un dispositivo de impresión, el cual imprime el electrocardiograma en tiempo real o en diferido sobre el ya mencionado papel estandarizado. Otras funcionalidades adicionales de los electrocardiógrafos pueden ser el envío de datos a un computador en diferentes formatos o la realización de un diagnóstico preliminar de forma automática.

Los cables que provienen del electrocardiógrafo se conectan a unos electrodos. Existen múltiples tipos de electrodos, pero sólo se mencionarán aquí los más comunes. Los electrodos desechables son unas almohadillas autoadhesivas que contienen un gel conductor y un borne. Éstos se adhieren a la piel del paciente en una serie de posiciones estandarizadas. A través del gel conductor y el borne las señales eléctricas que surgen del corazón pueden llegar a los cables del electrocardiógrafo. Estos cables suelen estar codificados mediante un código basado en colores para saber inmediatamente a qué electrodo deben de ser conectados.

Las señales eléctricas que toman los electrodos en la piel del paciente suelen ser del orden de decenas o centenares de milivoltios. Las características exactas de la señal en el electrocardiógrafo dependen del tipo de electrodo usado durante la medición.

Las empresas fabricantes de electrocardiógrafos apenas liberan información técnica sobre sus productos públicamente, de manera que es difícil conocer cómo funcionan estas

máquinas internamente. La mayoría de ellas toman datos de hasta 16 derivaciones simultáneamente a 500 muestras por segundo (500Hz). Algunas de las máquinas de mayor nivel incluyen funcionalidades adicionales, como almacenamiento de datos, la posibilidad de transmisión de datos o realización de un *registro Holter*.

2.3. Diagnóstico de patologías cardiovasculares basado en electrocardiogramas

A continuación se presentan las dos fases básicas aplicadas por los expertos para el uso de electrocardiogramas: Primero se realiza una interpretación del mismo y después se procede a realizar un diagnóstico basado en la morfología de la onda.

2.3.1. Interpretación de electrocardiogramas

Como se puede ver en la Figura 2.5 en la página 10, las señales eléctricas del corazón pueden ser etiquetadas. Para etiquetar los electrocardiogramas se usa un algoritmo estándar y la nomenclatura propuesta por *Willem Einthoven*. Esta metodología aplicable a la derivación I es extrapolada al resto de derivaciones para realizar el etiquetado de las mismas. En la Figura 2.7 se muestra un pulso en la derivación I completamente etiquetado.

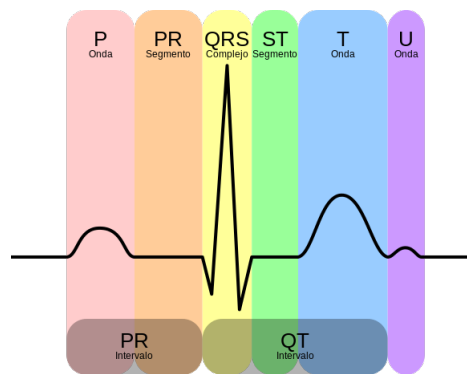


Fig. 2.7 ECG en derivación I completamente etiquetado

Fuente: commons.wikimedia.org, Hazmat2.

Lógicamente en un corazón no sano la onda no tendrá la misma morfología y puede que presente formaciones no descritas en la anterior figura. A la hora de interpretar y etiquetar el ECG, los especialistas usan una componente heurística e intuitiva adquirida y mejorada a lo largo de su experiencia profesional.

2.3.2. Diagnóstico basado en ECGs

El electrocardiograma es usado por especialistas médicos para detectar posibles alteraciones en el corazón como arritmias, infartos, bloqueos de rama, alteraciones electrolíticas o fallos en el estado físico del corazón, entre otros. Existen múltiples recursos, como [10] que contienen diagramas de bloques o algoritmos para realizar un diagnóstico sobre el electrocardiograma recibido de un paciente.

A modo de ejemplo se muestra a continuación una cita adaptada del libro referenciado en el párrafo anterior:

Para detectar una Q de necrosis:

- Onda Q significativa es un milímetro de ancho (un cuadro pequeño), el cual es .04 seg. de duración
- O es una onda Q 1/3 de la amplitud (o más) del complejo QRS

Otro ejemplo similar:

Para detectar una hipertrofia ventricular izquierda:

1. Sumar los valores de R en la derivación V5 y S en la derivación V1
2. Si la suma es superior a 35 milímetros (equivalente a 3,5 milivoltios) entonces se da la alteración

Dada la naturaleza de este tipo de diagnósticos es posible automatizar dicho proceso. Existen algunas empresas fabricantes de electrocardiógrafos como *GE Healthcare* o *Philips* que han invertido en investigación para crear algoritmos con este cometido. *GE Healthcare* posee el algoritmo *Marquette 12 SL* y *Philips* posee el algoritmo *DXL*. Lógicamente, debido al valor y secreto industrial de estos algoritmos, la información accesible públicamente sobre su funcionamiento es nula.

Parte I

Desarrollo del proyecto

Capítulo 3

Electrónica

La creación de un electrocardiograma comienza por la lectura de pequeñas señales eléctricas de un sujeto. Dichas señales son amplificadas y filtradas para poder ser recibidas correctamente. Después se convierten a formato digital, para poder procesarlas fácilmente en software. Por último serán enviadas por vía inalámbrica a la app Android que se ejecutará en una tablet.

El sistema electrónico se ha dividido en la *placa de adquisición de datos* y la *base*. La primera toma las señales del paciente, las amplifica y filtra, después las envía al sistema base. Éste último convierte la señal a formato digital y realiza el envío de la señal.

3.1. Arduino

Como base del sistema electrónico se ha utilizado una placa Arduino (o Arduino a solas). Los creadores de Arduino definen sus placas como sistemas de prototipado electrónico de bajo coste y fáciles de usar, dirigidas a aficionados de la electrónica, artistas y diseñadores, entre otros. La mayoría de las placas Arduino están basados en la familia de chips *ATmega* de *Atmel Corporation*.

En este proyecto, la placa Arduino se encarga de la recepción de las señales biológicas en formato analógico, conversión a formato digital y por último envío de los datos a la app complementaria que estará siendo ejecutada en una tablet. A lo largo de las próximas subsecciones se describe como se ha llevado a cabo la implementación de estos sistemas y la alimentación de la placa Arduino.

El modelo de placa Arduino utilizado en el proyecto ha sido el Arduino UNO R3. Esta placa está basada en el chip ATmega328 de Atmel Corporation. Se puede consultar más información sobre dicha placa en su página web [3].

3.1.1. Open hardware

Resulta interesante destacar que las placas Arduino son uno de los ejemplos más prominentes de hardware libre. Esto significa que todos sus diseños son públicamente accesibles, lo cual permite que cualquier fabricante de componentes electrónicos, con suficiente capacidad, pueda crear sus propias placas basadas en Arduino y posteriormente, comercializarlas. En la sección 7.5 en la página 72 se detallan las ventajas de este sistema de cara al abaratamiento de este proyecto.

3.1.2. Conversor analógico-digital

El conversor analógico-digital o ADC es un sistema electrónico que convierte valores de voltaje analógicos a valores digitales, lo cual permite enviar y tratar los datos de forma cómoda mediante software. El chip ATmega328 que incorpora la placa Arduino UNO incluye un ADC interno.

El conversor está limitado por una serie de parámetros físicos. Los más importantes a destacar de cara a este proyecto son la resolución y la frecuencia.

Resolución

La resolución se define como la menor diferencia entre dos valores sucesivos que puede detectar el ADC. El conversor del Arduino cuenta con una resolución de 10 bits, lo cual limita el rango de salida a 1024 valores. Teniendo en cuenta que el conversor es lineal, si el rango de entrada varía de 0 a 5 voltios, entonces la menor diferencia entre dos valores sucesivos de voltaje es 0,00488 voltios.

Resulta interesante ajustar la sensibilidad lo mejor posible al rango de entrada, para sacar el mayor provecho a la resolución del ADC. Si los valores que recibe el ADC varían entre 0 y 1 voltios es absurdo que el rango de entrada alcance los 5 voltios. Reduciendo el rango de entrada a 1,1 voltios se puede alcanzar una resolución de 0,00107 voltios, ya que los 1024 valores posibles tienen un rango menor que cubrir.

En la subsección 2.2.3 en la página 11 se menciona que el rango de voltaje de las señales cardíacas varía dentro de las decenas o centenas de milivoltios. Aun usando el menor rango posible en el ADC del Arduino no es posible detectar estas señales. Por esta razón es necesaria una o varias etapas de amplificación, lo cual se describe más adelante en la subsección 3.2.2 en la página 24. En el prototipo que aquí se presenta se ha delegado el establecimiento del rango de valores del ADC a la placa de adquisición de datos usada.

Frecuencia

La frecuencia indica el número de conversiones que puede realizar el ADC por unidad de tiempo. El conversor del chip ATmega328 necesita 13 ciclos para realizar una conversión normal y utiliza la frecuencia de oscilación del cristal oscilante de la placa Arduino reducida a través de un premultiplicador.

Por defecto el valor de frecuencia del cristal oscilante es de 16MHz y el valor del premultiplicador es “división entre 128”. Esto quiere decir que el ADC funciona a $16.000\text{KHz}/128 = 125\text{KHz}$. A 13 ciclos por conversión el conversor puede convertir 9600 valores al segundo: $125\text{KHz}/13 = 9,6\text{KHz}$. Debe tenerse en cuenta que estos resultados se obtienen con un uso correcto del conversor analógico-digital. Es posible reducir el valor del divisor en el premultiplicador para aumentar el rendimiento del conversor, pero a cambio se reduce la precisión del mismo.

Para establecer la frecuencia de muestreo para la obtención de un electrocardiograma se estudian cuales son los valores máximo y mínimo más adecuados para la tarea presente. El valor máximo viene dado por las limitaciones físicas del ADC. De cara al ECG es indiferente que la frecuencia sea demasiado elevada, la única consecuencia de este hecho es un superávit de información. Para hallar el valor mínimo de muestreo debe estudiarse que complejo, intervalo o segmento del ECG es el más rápido y cuantos puntos hay que tomar sobre él para evitar la pérdida de información.

La parte más breve y a la vez más crítica de un ECG son los complejos QRS, éstos tienen una longitud de 100 a 200 milisegundos. Aplicar una frecuencia de muestreo de 500Hz, al igual que los electrocardiógrafos profesionales, resulta suficiente para obtener la información deseada sin exceder los límites del ADC. Por estas razones el valor de frecuencia de muestreo utilizado será 500Hz o 500 muestras por segundo.

Actualmente el prototipo toma y convierte mediciones de un solo electrodo, pero en el futuro podría hacerlo de más, tres electrodos según el alcance original propuesto. En este caso el ADC convertiría 500 muestras por segundo por electrodo, luego 1500 muestras en total. Este valor sigue estando por debajo de los 9600 máximos, luego sería posible ampliar la información recogida sin modificar los parámetros del ADC del Arduino.

3.1.3. Comunicación

El chip ATmega328 usado por el Arduino cuenta con funcionalidad de comunicación serie basada en USART, *Universal Synchronous and Asynchronous serial Receiver and Transmitter*. Las librerías de Arduino simplifican extremadamente su funcionalidad, haciendo el

envío de datos tan sencillo como llamar a una función.

La placa Arduino cuenta con dos pines, TX y RX, que sirven como punto de transmisión y recepción de datos respectivamente. La conexión se realiza a una velocidad dada, denominada *tasa de baudios* cuyos valores están estandarizados. Tanto el emisor como el receptor deben usar la misma tasa de baudios para realizar una comunicación de datos correcta, en caso contrario el receptor recibirá valores inútiles.

En las siguientes subsecciones se estudia la forma en la que se codificarán los datos para ser enviados y el valor de baudios que se debe usar para evitar problemas en la comunicación.

Codificación

Los datos que se envíen del Arduino a la app deben de ser codificados de alguna manera para poder ser entendidos por ésta última al recibirlos. La codificación usada en la versión del proyecto que aquí se presenta es muy simple: cada lectura se envía seguida del carácter U+0058, este carácter representa la equis mayúscula.

Así es un segmento de electrocardiograma enviado desde el Arduino:

```
675X640X604X554X643X641X611X613X713X644X616X      (3.1)
```

Velocidad de comunicación

Debe tenerse en cuenta que la velocidad de transmisión de los datos en una aplicación como esta es extremadamente importante. Todos los datos recogidos deben enviarse en tiempo real del Arduino a la app complementaria. Si los datos no pueden ser enviados a suficiente velocidad, entonces los buffers del chip ATmega328 acabarán desbordándose, conllevando una pérdida de información.

Como ya se ha mencionado en la subsección 2.2.3 en la página 11, un electrocardiógrafo profesional puede tomar del orden de 500 muestras por segundo sobre varios electrodos. En la implementación actual de este proyecto se toman 500 muestras por segundo sobre una derivación cardíaca, y se envían de la manera indicada en la anterior subsección.

Cada una de las cifras de cada valor tomado se codifica como un carácter, en la amplia mayoría de los casos el valor leído contendrá tres cifras. Al sumar el carácter de control, el total de caracteres enviados por cada lectura asciende a cuatro.

Cada carácter se codifica como un byte, es decir ocho bits. 500 lecturas por segundo generan 2000 bytes de información, o 16.000 bits. El primer valor estándar de *tasa de baudios* superior a 16.000 es 19.200 baudios. De todas formas, debido al overhead añadido por la

placa Arduino a través de las funciones de transmisión de datos, la tasa de baudios utilizada en el prototipo final del proyecto es 57.600 baudios.

Bluetooth

En este proyecto se ha conectado un módulo Bluetooth genérico (JY-MCU, basado en el chip HC-06) directamente a la placa Arduino a través de los pines TX y RX para dotar a la placa de comunicación inalámbrica. El módulo Bluetooth ha sido configurado para usar la misma tasa de baudios que la placa Arduino a través del *Conjunto de comandos Hayes*. El módulo Bluetooth actúa de forma transparente, de manera que toda la programación necesaria para enviar y recibir datos es implementada en la placa Arduino.

En la subsección 4.3.2 en la página 39 se indica como la app complementaria recibe los datos de este módulo Bluetooth.

3.1.4. Alimentación

La placa Arduino debe de ser alimentada mediante electricidad para poder funcionar. Hay múltiples formas de alcanzar este objetivo, como el suministro de electricidad a partir de una conexión a la red eléctrica (usando el conversor apropiado) o el uso de baterías.

En las Tablas 3.1 y 3.2 se muestran las ventajas y desventajas de cada uno de los métodos candidatos.

Tabla 3.1 Ventajas y desventajas de la alimentación por red eléctrica

<i>Red eléctrica</i>	
<i>Ventajas</i>	<i>Desventajas</i>
No se agota con el uso del aparato	Precisa un circuito de protección para el paciente
Disponible en la mayoría de los lugares en los que se tomaría un ECG	La electricidad de corriente alterna debe de ser convertida a continua para poder ser usada aquí

Es necesario destacar un factor altamente relevante en la toma de datos que no ha sido mostrado en las Tablas: el ruido eléctrico. Este fenómeno se describe en la subsección 3.2.1 en la página 22. El ruido eléctrico afecta gravemente a los aparatos de medición sensibles, echando a perder sus lecturas. El prototipo que aquí se presenta no es una excepción, por tanto se puede deducir que es estrictamente necesario que el aparato sea independiente de la red eléctrica o que cuente con un buen sistema para eliminar el ruido eléctrico.

Tabla 3.2 Ventajas y desventajas de la alimentación por baterías

<i>Baterías</i>	
<i>Ventajas</i>	<i>Desventajas</i>
Producto autocontenido en una sola unidad: mayor portabilidad y sencillez	Necesita recambios cada cierto tiempo
Se evita riesgo de shock eléctrico	
Disponible globalmente bajo valores estándar (Baterías AA, AAA, 9V, etcétera)	
Las baterías no precisan una transformación o regulación de su salida	

Dadas las claras ventajas del uso de baterías sobre el uso de la red eléctrica como alimentación del electrocardiógrafo, son éstas las elegidas como fuente de alimentación en el prototipo final. Se ha usado un pack de baterías compuesto por 5 pilas AA, para obtener un voltaje total de 7,5 voltios. Éstas se conectan al Arduino UNO, el cual cuenta con los circuitos de regulación necesarios para adaptar los valores de voltaje a los valores adecuados para el resto de componentes electrónicos.

3.2. Placa de adquisición de datos

El cometido de la placa de adquisición de datos es tomar las bioseñales del paciente y transmitir las a la placa Arduino. La creación de este elemento del proyecto no es trivial, ya que las bioseñales, como se indica en la sección 2.2.3 en la página 11, son muy débiles. Para que la placa Arduino pueda recibir las ondas en su conversor analógico digital, éstas serán previamente amplificadas. Además también se les aplicarán técnicas de filtrado, ya que el ruido electromagnético interfiere en la toma de datos.

A lo largo de las próximas subsecciones se exponen los problemas encontrados, investigación realizada y soluciones halladas para la amplificación de la señal, placas de datos usadas, interferencia del ruido y filtrados por software y hardware.

3.2.1. Ruido eléctrico

El ruido eléctrico es un fenómeno que se da en todos los circuitos electrónicos, se define como una fluctuación aleatoria en la señal. Normalmente el ruido eléctrico resulta desfavorable en situaciones en las cuales se desea medir con precisión el potencial de una señal. Las fuentes del ruido eléctrico pueden ser naturales o artificiales, entre estas últimas podemos

contabilizar la red eléctrica, aparatos que precisen alto voltaje (lámparas fluorescentes) o emisores de radiofrecuencia, como móviles o puntos de acceso WiFi.

La fuente más predominante de ruido suele ser la red eléctrica, presente en la mayoría de los edificios. Las fases contenidas en dicha red oscilan a una frecuencia de 50Hz. Existen múltiples métodos para prevenir o eliminar la influencia del ruido eléctrico, aunque rara vez son totalmente efectivos.

El método de prevención más extendido es denominado *apantallamiento*. Este método consiste en recubrir el cable que transporta la señal que se desea leer con una malla o tubo metálico conectado a masa. Dicha malla o tubo actúa como una *Jaula de Faraday*, previniendo la intromisión de ruidos.

El único método de eliminación de ruido, una vez que éste se ha introducido en la señal, es llamado filtrado. Hoy en día existe una gran extensión de técnicas de filtrado, aplicables en hardware, software, mediante métodos matemáticos o simulación de circuitos.

Ruido en la electrocardiografía

Dado que el potencial eléctrico del ruido es muy pequeño éste no afecta a la mayoría de los circuitos. En este caso no es así, ya que el potencial de un ECG se sitúa en el mismo rango de valores que el de las fluctuaciones ruidosas. En los electrocardiogramas brutos, aquellos a los cuales no se les ha aplicado ninguna técnica de filtrado, pueden darse múltiples tipos de ruido, algunos de ellos bien conocidos y eliminables y otros que deben de ser evitados en primer lugar.

Por una parte está el ruido en los 50Hz insertado en la onda por la red eléctrica. No hay nada que se pueda hacer para evitar este fenómeno, más que usar unos cables apantallados de gran calidad para conectar los electrodos. Un prototipo alimentado por baterías generado a lo largo de este proyecto tomaba la señal mostrada en la Figura 3.1. En dicha Figura se observa la existencia de una gran cantidad de *picos* consecutivos, pero si nos fijamos en la parte central de la onda entre se puede vislumbrar un ECG.

En la Figura 3.2 se muestra la interferencia introducida por un sistema de conversión AC-DC de baja calidad. En este caso el prototipo estaba conectado a la red eléctrica a través de un puerto USB en un portátil enchufado a la red eléctrica. En este caso el ECG es irreparable, debido a que el ruido no es uniforme.

Más allá del ruido introducido por la red eléctrica, de manera directa o indirecta, existe el ruido generado por aparatos de radiofrecuencia. Para mitigar lo máximo posible la aparición de interferencia causada por estos aparatos es recomendable apartarlos de la zona donde se fuera a tomar el ECG antes de llevar a cabo la medición.

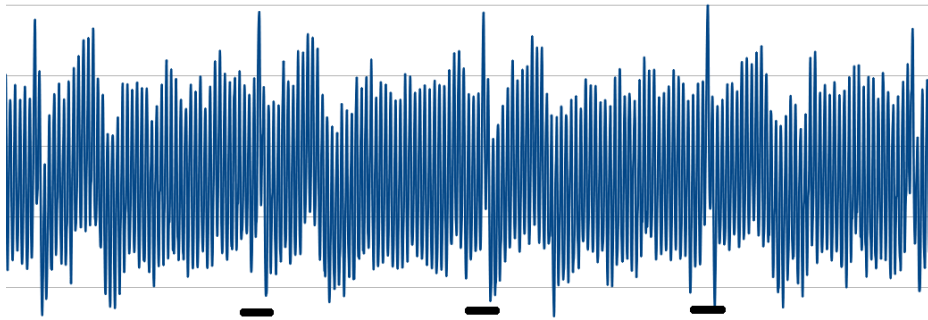


Fig. 3.1 ECG sin conexión a la red eléctrica con ruido de 50Hz con 3 complejos QRS subrayados en negro



Fig. 3.2 ECG con ruido de origen AC con 3 complejos QRS subrayados en negro

El movimiento de los músculos del sujeto al cual se le está tomando un electrocardiograma también introduce ruido en el mismo. Téngase en cuenta que los demás músculos, al igual que el corazón, también utilizan reacciones químicas que generan señales eléctricas para moverse. La única forma de evitar esta componente ruidosa es que el paciente esté en un estado de reposo y evite cualquier movimiento durante la toma del ECG.

Por último debe considerarse un movimiento muscular inevitable: el generado por la respiración. El ruido causado por este fenómeno no se puede evitar, ya que conllevaría consecuencias negativas para el sujeto del cual se esté tomando el electrocardiograma. Afortunadamente se sabe que la respiración introduce ruido en forma de onda sinusoidal a muy baja frecuencia, en torno a los 0,5Hz o inferior. Esto simplifica la eliminación de la componente respiratoria. El ruido eléctrico introducido por la respiración puede verse en la Figura 3.3, su efecto es un ligero desplazamiento del ECG en el eje de las ordenadas.

3.2.2. Amplificación

Las señales que se toman de los electrodos son demasiado pequeñas para la sensibilidad del módulo ADC de la placa Arduino, como ya se ha indicado en la sección 3.1.2 en la página 18.

La única solución al problema consiste en aumentar la magnitud del voltaje de las bioseñales. Para ello se debe usar un circuito de amplificación. Los circuitos de amplificación

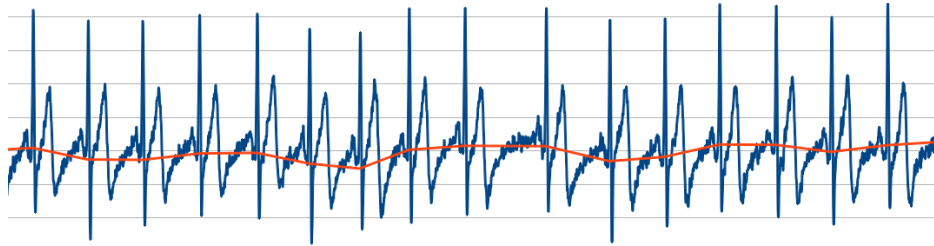


Fig. 3.3 ECG con ruido causado por respiración, la línea naranja representa el desplazamiento de la onda

cuentan con un amplificador y otros componentes electrónicos para conformar una etapa de amplificación. Existen diferentes tipos de etapas de amplificación: sumadoras, diferenciales, inversoras, no inversoras, etcétera.

Como ya se ha mencionado en la subsección 3.2.1 en la página 23, tanto las bioseñales como el ruido, se encuentran en el mismo rango de valores de potencial. Esto hace que la señal no pueda ser amplificada en su totalidad, ya que el ruido también sería amplificado, lo cual es negativo. Para conseguir amplificar únicamente la señal de datos se deberá usar un tipo de etapa de amplificación específica.

Una etapa de amplificación configurada como *amplificador diferencial* ofrece buenos resultados. Esta etapa aumenta la diferencia entre las dos señales de entrada. Si se conectan el electrodo del brazo izquierdo y el electrodo del brazo derecho al amplificador, el resultado es la señal correspondiente a la derivación I.

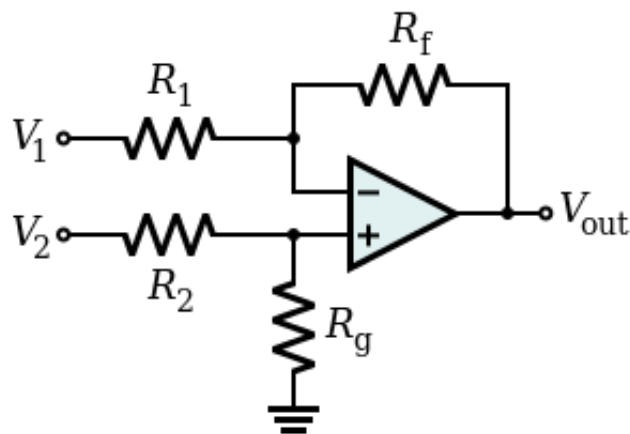


Fig. 3.4 Etapa de amplificación diferencial

Fuente: commons.wikimedia.org

El hecho de amplificar la diferencia entre dos señales ayuda con la eliminación del ruido, el razonamiento tras esta aserción es el siguiente: Las señales de ambos electrodos contenen-

drán las mismas fluctuaciones causadas por el ruido a la entrada del amplificador. Si sólo se amplifica la diferencia, las fluctuaciones del ruido quedarán descartadas, eliminándolas en la onda presente a la salida del amplificador. El efecto aquí descrito puede verse en la Figura 3.5.

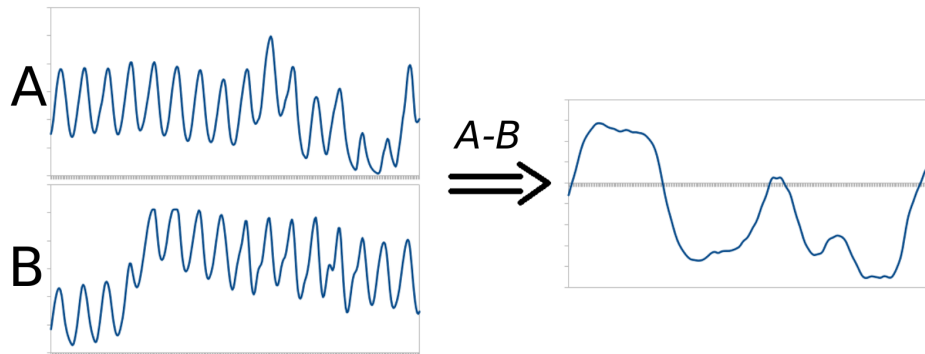


Fig. 3.5 Eliminación del ruido por filtrado diferencial. La onda en la mitad derecha se genera a partir de la resta de la onda A menos la onda B. Nótese la ausencia de ruido en el resultado. Aunque aquí no se muestran las gráficas a escala, la onda resultante es de mayor magnitud que A y B, ya que ha sido amplificada.

Fuente: commons.wikimedia.org

Conectar los electrodos del paciente a las entradas del amplificador diferencial directamente no es correcto. La impedancia de entrada sería demasiado baja, causando el movimiento de corriente por los cables. Este movimiento de corriente deterioraría la señal que se desea tomar. Para aumentar la impedancia de entrada, manteniendo el mismo comportamiento de amplificación de la diferencia entre dos ondas, se usa una combinación de amplificadores operacionales que forman una etapa llamada *amplificador de instrumentación*.

En el amplificador de instrumentación la impedancia de entrada es infinita, lo cual evita la aparición de corrientes en los cables que se conectan a los electrodos. Esto hace que las señales que forman el electrocardiograma queden intactas durante la fase de amplificación.

3.2.3. Placas de adquisición de datos

A lo largo de este proyecto se han probado tres diferentes placas de adquisición de datos. Estas placas han sido diseñadas con el propósito de tomar electrocardiogramas sencillos y cuentan con la circuitería de amplificación y filtrado necesaria para obtener un ECG útil.

Todas estas placas tienen como objetivo leer la señal correspondiente a la derivación I respecto a la referencia de voltaje del sujeto sobre el cual se esté obteniendo el ECG, para

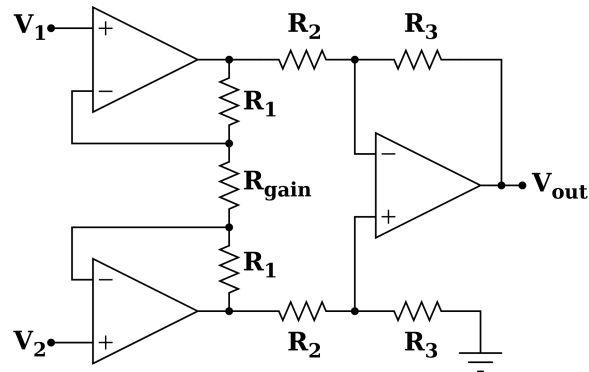


Fig. 3.6 Diagrama de un amplificador de instrumentación. V_1 y V_2 representan las entradas de la etapa, donde se conectarán los cables que llevan a los electrodos. Del conector V_{out} saldrá la señal amplificada y libre de la mayoría del ruido eléctrico.

ello se hace uso tres electrodos. Estos electrodos se sitúan en las muñecas de ambos brazos y en el tobillo derecho del sujeto de pruebas.

Los diagramas de conexiones de las tres placas que se presentan a continuación se encuentran presentes en el Anexo A.

Placa de diseño propio con el chip AD8232

La primera placa sobre la cual comenzó el proceso de pruebas fue una adaptación de uno de los circuitos de referencia propuestos por *Analog Devices* en la ficha técnica de su chip AD8232 [2]. Este chip ha sido diseñado para amplificar y filtrar electrocardiogramas sencillos adquiridos en aparatos deportivos comerciales. La placa personalizada fue diseñada y creada con el apoyo de la empresa *Mirakonta*.

Sobre la circuitería propuesta se añadieron los componentes necesarios para crear dos vías de alimentación mediante baterías de litio de 3,6 voltios. Una vía sencilla para transmitir la energía directamente al chip AD8232 y una vía con un regulador de voltaje para reducir el voltaje de la batería de litio al valor teórico máximo admitido por el chip. El resto de adiciones sobre el diseño de *Analog Devices* consistieron en facilidades para compatibilizar los demás componentes que fueron usados en la placa, como los cables apantallados.

Debido un error no identificado en el proceso de fabricación de la placa, ésta resultó quedar inutilizada. Las señales producidas por la misma no parecían tener ningún sentido, estando compuestas puramente de ruido eléctrico. Es por ello que esta solución fue descartada.

Placa de evaluación para el chip AD8232

Otra de las placas usadas para tomar electrocardiogramas en este proyecto fue la placa de evaluación del chip AD8232, creada por Analog Devices. Esta placa se encuentra disponible comercialmente para especialistas en electrónica que deseen conocer el funcionamiento del chip antes de crear sus propios diseños. La placa es altamente configurable, debido a su naturaleza como plataforma de pruebas.

De ahora en adelante la placa será referenciada como AD8232Eval.

En este proyecto los valores de configuración utilizados han sido estos:

- FR a FR_EN
- AC/DC a DC
- SDN_N a EN
- P1, P2 y P4 abiertos
- P3 cerrado

Los terminales con conexiones fueron los siguientes:

- Salida de voltaje regulada a 3,3 voltios del Arduino a TP9 en AD8232Eval
- GND de la placa Arduino a TP10 en AD8232Eval
- TP15 en AD8232Eval a A0 en la placa Arduino
- TP7 en AD8232Eval a GND en la placa Arduino
- Electrodo del brazo izquierdo a TP1 en AD8232Eval
- Electrodo del brazo derecho a TP2 en AD8232Eval
- Electrodo de la pierna derecha a TP3 en AD8232Eval

Toda conexión en esta placa se realizó con cables **sin** apantallar.

Se presenta un ECG tomado con esta placa en la Figura 3.7. La placa AD8232Eval genera unas señales de morfología extraña, no comparables con la onda esperada para un ECG de la derivación I. Además incluyen una pequeña componente ruidosa difícil de eliminar, debido a lo poco uniforme que es.

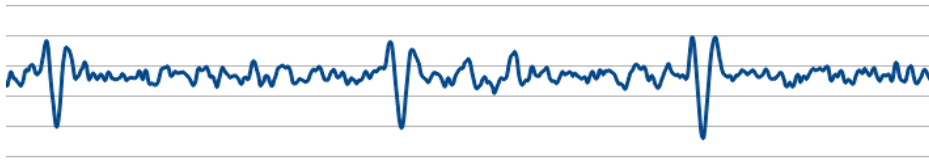


Fig. 3.7 ECG de la placa AD8232Eval (Derivación I)

Olimex ECG Shield

La placa Olimex ECG Shield es, como su nombre indica, un shield (extensión) para Arduino. Esta placa se monta directamente sobre el Arduino y únicamente requiere una configuración correcta basada en *jumpers* y la conexión de los electrodos a través de unos cables especiales. Las placas Olimex son de diseño abierto, luego cualquiera puede fabricarlas con las maquinaria necesaria. La empresa que creó el diseño original y comercializa las placas actualmente es *Olimex*. Olimex también provee unos cables con electrodos de pulsera para tomar los electrocardiogramas a través de su placa.

La configuración usada a lo largo del proyecto en esta placa ha sido:

- REF_E: cerrado
- CAL: abierto
- PWR: 3.3V
- D4/D9: D9
- AIN_SEL: 1
- Potenciómetro TR1: valor por defecto

Los electrodos de pulsera recomendados para esta placa no ofrecían unos resultados aceptables, de manera que las pulseras fueron sustituidas por pinzas, las cuales pueden ser conectadas a electrodos autoadhesivos desechables con borne. Debido al gel electroconductor de los electrodos desechables la calidad de los electrocardiogramas aumentó considerablemente. Los resultados del cambio pueden verse en las Figuras 3.8 y 3.9. En ambas figuras se ha eliminado el ruido a 50Hz mediante postprocesamiento digital para poder visualizar correctamente las ondas.

Aunque la calidad de la onda sea bastante aceptable, la onda es morfológicamente incorrecta. Las anomalías se deben probablemente a un fallo electrónico. Estas sospechas se basan en que las anomalías se han dado en electrocardiogramas de diferentes sujetos con

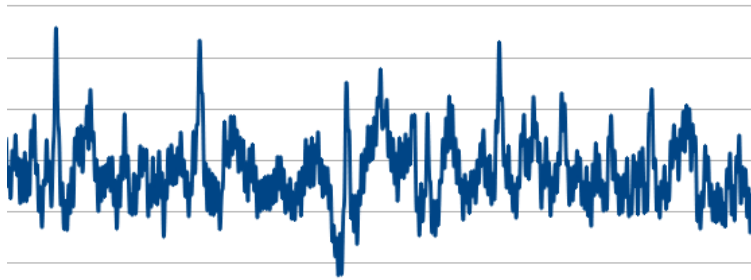


Fig. 3.8 ECG de la placa Olimex con electrodos de pulsera (Derivación I)

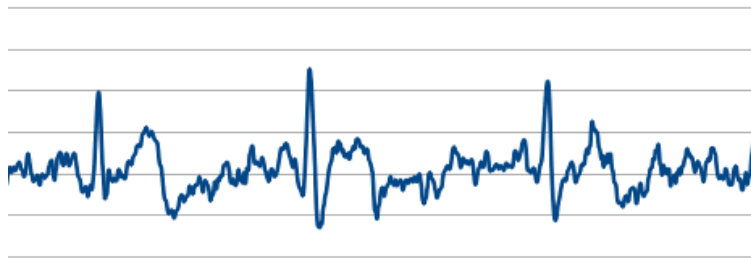


Fig. 3.9 ECG de la placa Olimex con pinzas y electrodos desechables (Derivación I)

corazones que no debieran presentar los problemas asociados a la onda anómala y que éstas se rigen por un patrón previsible. De todas formas, dada la complejidad de la placa Olimex no se ha podido identificar el circuito causante de dichas anomalías en la onda.

3.2.4. Filtrado del ruido

Un electrocardiograma puede interpretarse como una onda con componentes en diferentes frecuencias, este es el mejor enfoque de cara a eliminar el ruido, ya que éste último también puede ser interpretado como componentes a ciertas frecuencias que forman parte de una onda. Si se puede distinguir que frecuencias representan el electrocardiograma y cuales el ruido, éstas últimas podrían eliminarse para obtener únicamente el electrocardiograma.

El filtrado de las componentes indeseadas puede realizarse por hardware o software.

El **filtrado por hardware** no ha sido estudiado de cara a este proyecto, más allá de lo expuesto en la subsección 3.2.2. Las placas presentadas en la anterior subsección utilizan una gran cantidad de métodos internamente para evitar el ruido y poseen filtros por hardware, aunque estas técnicas no han resultado totalmente efectivas.

El **filtrado por software** se basa en el procesamiento posterior a la captación de la señal, una vez que ésta ha sido digitalizada. Éste ha sido el tipo de filtrado trabajado durante el proyecto de cara a la mejora de la señal, partiendo de los datos imperfectos generados por

las placas de adquisición.

Frecuencias a eliminar

Si el ruido es representable como una o varias componentes a lo largo de diferentes frecuencias y se desea eliminar, entonces se deberá conocer cuáles son los valores de estas frecuencias. Los artículos académicos en los que se estudia la instrumentación de los electrocardiógrafos, es decir, los circuitos de amplificación y filtrado, poseen opiniones divergentes. Aun así, parece haber un consenso general de que las frecuencias útiles se encuentran en el rango de los 0,5Hz a 100Hz, de manera que todas las componentes en estas frecuencias deberían ser conservadas y las demás atenuadas o eliminadas.

Filtrado por simulación de circuitos RC

Los circuitos RC son circuitos formados por una resistencia y un condensador, en serie o paralelo. Si estos dos componentes se ajustan correctamente pueden eliminar las frecuencias indeseadas de una onda. Un circuito RC puede ser simulado a través de un algoritmo que opera sobre un array. Para eliminar el ruido de alta frecuencia (filtro paso bajo) se partió del pseudocódigo disponible en [27] y se realizaron unas modificaciones para adaptarlo al lenguaje de programación Java y mejorar su comprensibilidad.

Los resultados de las ejecuciones de este código fueron infructuosos, aunque muy variables. Éstos se extienden desde un fracaso total en el filtrado de la onda, dejándola al igual que su estado anterior, a resultados más positivos. En las mejores pruebas la onda quedaba libre de altas frecuencias, a costa de perder parte de la propia onda, haciendo ésta inútil de cara a un futuro diagnóstico.

También se realizaron pruebas con otra variante de la simulación RC destinada a eliminar ruido de baja frecuencia (filtro paso alto). El objetivo de estas pruebas fue eliminar el ruido causado por la respiración. Al igual que con el anterior tipo de simulación RC se partió del código disponible en la Wikipedia [26] y se realizaron una serie de modificaciones.

En este caso los resultados no fueron más esperanzadores que los anteriores. Todas las ondas filtradas sufrían una anomalía mediante la cual tomaban el valor cero como asíntota, haciendo que estas perdieran toda su información útil.

Filtrado por la transformada Fourier

Se realizaron algunos intentos de filtrar la señal aplicando Fourier. A través de la transformada de Fourier una onda puede trasladarse del dominio del tiempo al dominio de las

frecuencias. En este segundo dominio pueden verse que frecuencias forman la onda y si se poseen suficientes conocimientos matemáticos, entonces se puede transformar la onda.

En teoría, a través de esta metodología se pueden eliminar las frecuencias molestas en el electrocardiograma y reconstruir la onda original sin ellas. La práctica resulta más complicada y no se obtuvieron los resultados esperados. Dada la complejidad de este tipo de filtrado y el poco tiempo disponible para estudiarlo se optó por descartarlo e investigar otros métodos en su lugar.

Otros tipos de filtrado

Para eliminar el ruido eléctrico introducido por la respiración se consideró la aplicación de un filtro *Butterworth*. Este filtro ha sido elegido debido a un interesante estudio que trata sobre la aplicación de filtros paso bajo que dañen lo menos posible la onda P de los ECGs [6]. Del citado artículo se extrae la conclusión de que un filtro Butterworth bidireccional de cuarto orden con una frecuencia de corte de 0,5Hz es el más adecuado para esta tarea. A partir de una herramienta online de creación de filtros [11] se obtuvo una implementación en código C, la cual fue adaptada a Java.

Los resultados fueron idénticos al de la simulación de un filtro RC paso alto. La onda tomaba el valor cero como asíntota, con las negativas consecuencias que ello conlleva.

Filtrado por media móvil

El filtro más efectivo aplicado a lo largo del desarrollo del proyecto ha sido un filtrado realizado mediante la aplicación de una variante de la media móvil a los valores leídos. Cada valor tomado era transformado a la media de si mismo y los 19 valores anteriores (bajo una frecuencia de muestreo de 1KHz). La cantidad de valores tomada para realizar la media se corresponde con 20 milisegundos, exactamente el valor en el cual fluctúa el ruido eléctrico de 50Hz. Los resultados de este tipo de filtro resultaron ser excepcionales, ya que, aunque no se eliminan todas las frecuencias de las componentes ruidosas, sí se elimina la más molesta.

Se intentó aplicar este tipo de filtrado al ruido de baja frecuencia, pero no fue posible debido a que la frecuencia que conforma dicho ruido se encuentra por debajo de 1Hz y es variable entre diferentes sujetos o incluso en un mismo sujeto en diferentes situaciones.

3.2.5. Elección final

La razón de haber estudiado todas estas posibilidades y las múltiples combinaciones entre placas de datos ha consistido en encontrar la mejor combinación de placa de adquisición

de datos y filtro software para obtener los electrocardiogramas más claros.

Para el prototipo presentado en este proyecto se ha usado la placa **Olimex ECG Shield** con cables de tipo pinza conectados a electrodos desechables. Esta placa se monta sobre un Arduino UNO Rev.3, que convierte y envía los datos a través de un módulo Bluetooth genérico una app en una tablet. En dicha app se aplica una variante del **filtro por media móvil** que neutraliza el ruido en la frecuencia de 50Hz.

El ruido de baja frecuencia causado por la respiración sigue presente, no se ha podido continuar la investigación sobre su eliminación debido a las limitaciones temporales del proyecto.

Capítulo 4

Software, app

Una vez definido el sistema de adquisición de datos puede intuirse que en alguna parte deberán poder visualizarse o tratarse dichos datos. Para resolver este problema se ha desarrollado una aplicación para dispositivos Android. La app creada permite visualizar un electrocardiograma en tiempo real. Además incluye un sistema de diagnóstico automatizado, implementado mediante un motor de inferencia. El motor de inferencia se estudia en el Capítulo 6.

Durante el desarrollo de la app se ha prestado especial atención a las directivas de diseño y programación de Google y se han investigado intensivamente todas las formas disponibles para crear cada característica de la misma, de esta manera siempre se ofrece al usuario la mejor experiencia posible.

4.1. Elección de la plataforma

Hoy en día existen múltiples soluciones en el mercado capaces de cumplir los objetivos que se acaban de establecer. A lo largo de las próximas subsecciones se estudian las plataformas disponibles para soportar la app desde los puntos de vista de hardware y software en los que están basados.

4.1.1. Hardware

Para poder mostrar los datos recogidos por la placa de adquisición de datos y ejecutar un diagnóstico sobre el ECG se precisa una plataforma con capacidad computacional suficiente. Un computador personal podría ser la plataforma óptima, pero en la actualidad se presenta un competidor capaz de igualarlo en cuanto a rendimiento y superarlo en cuanto a

portabilidad y manejabilidad: los *Tablet PC*.

Los Tablet PC o simplemente tablet (de ahora en adelante) cuentan con algunas ventajas más sobre los computadores personales, como su estructura autocontenida y extensas capacidades de comunicación inalámbrica. El bajo coste de una tablet, con capacidad suficiente para llevar a cabo las tareas aquí descritas, hace esta tecnología asequible para el público objetivo del producto final que se ha procurado alcanzar en este proyecto.

4.1.2. Software

Dentro del mercado de las tablets existen dos grandes ecosistemas disjuntos, el ecosistema *iOS* creado por la empresa *Apple* y el ecosistema *Android* creado por *Google*. Bajo *iOS* se encuentran los diferentes modelos de la tablet *iPad* de *Apple*. Por otra parte, dentro del ecosistema *Android* existen múltiples fabricantes de tablets, ya que *Google* no controla el desarrollo de nuevo hardware. El hecho de que existan múltiples fabricantes genera un sistema competitivo en el cual los dispositivos avanzan rápidamente y el coste de compra descende año tras año.

Hoy en día es prácticamente imposible desarrollar una aplicación de alta calidad para dispositivos *iOS* sin realizar un desembolso económico considerable, lo cual no es asequible en este contexto.

Desarrollar para *Android* resulta extremadamente sencillo y barato, tan solo es necesario descargar y poner a punto el *Android Software development Kit*(*Android SDK*), pública y gratuitamente accesible. Además desde la página web para desarrolladores también está disponible una versión del entorno de desarrollo (IDE) *Eclipse* preparada para desarrollar para dispositivos *Android*.

Además, al disponerse de una tablet *Android* es posible depurar la aplicación en un dispositivo real. Al usar un entorno real, a diferencia de uno simulado, se ha podido probar la integración con la placa *Arduino*.

4.2. Flujo de ejecución de la aplicación

La Figura 4.1 representa el flujo de ejecución principal de la aplicación. No existe un punto de salida específico para la aplicación, el usuario podrá abandonarla en cualquier momento usando los mecanismos existentes en el sistema operativo *Android*.

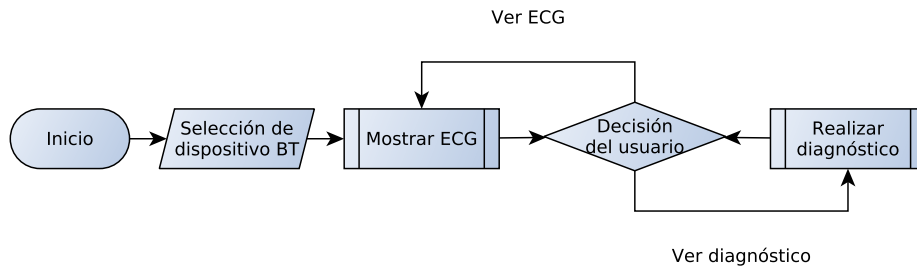


Fig. 4.1 Flujo de ejecución principal de la aplicación

Los diagramas de la Figura 4.2 muestran en detalle los algoritmos que forman la visualización del ECG y la realización del diagnóstico.

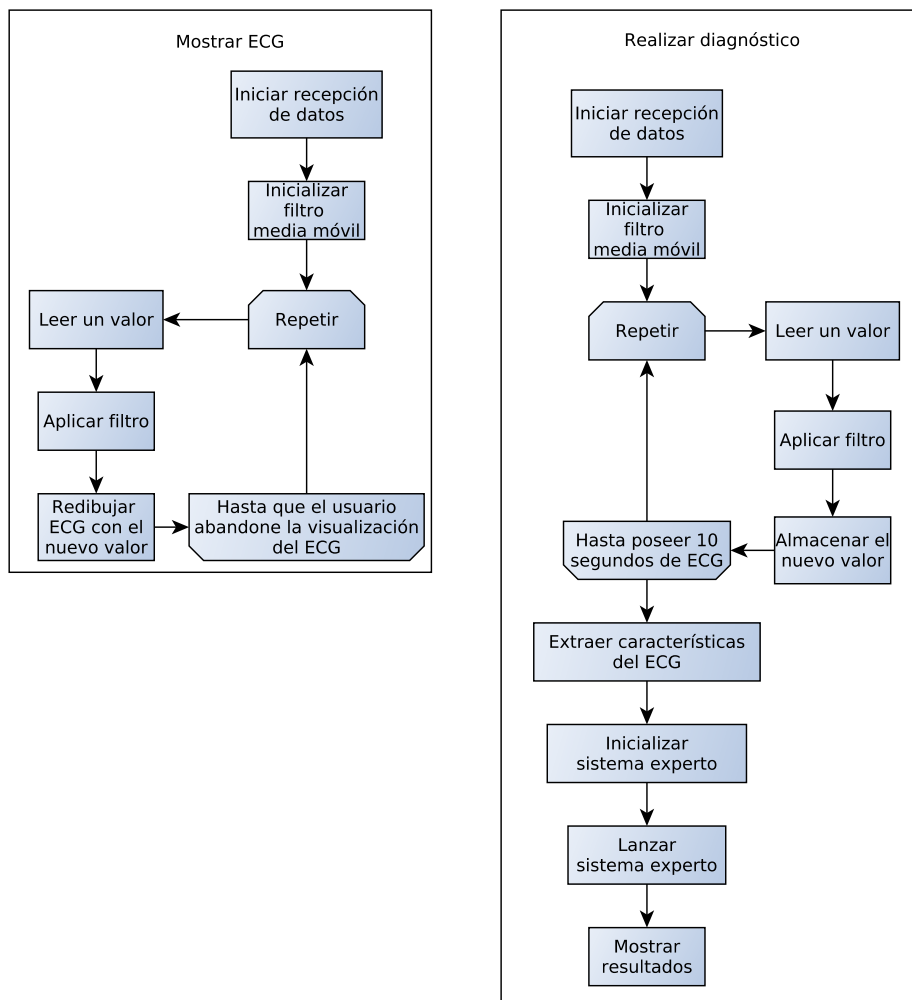


Fig. 4.2 Flujos de ejecución parciales de la aplicación

La primera parte en ambos flujos de ejecución es compartida por ambas tareas, única-

mente se diferencian en las acciones que realizan tras tomar y filtrar un dato y las acciones posteriores a la adquisición de datos. En la implementación esta compartición de código se ha realizado mediante una superclase.

4.3. Aspectos destacados de la implementación

La creación de la app ha sido un proceso extenso, aunque no complicado. Aun así existen algunos aspectos de la implementación que deben recibir especial atención de cara a su documentación, debido a lo específicos que resultan. Entre ellos se encuentran un mecanismo de filtrado en tiempo real para poder eliminar de los datos las interferencias existentes, un sistema Bluetooth sólido y una interfaz creada mediante técnicas punteras.

4.3.1. Filtrado en tiempo real

Los datos provenientes del Arduino contienen una componente ruidosa, como ya se ha explicado a lo largo del capítulo anterior. Para eliminar el ruido se aplica un filtro basado en una variante de la media móvil, ya que ha demostrado ser el filtro más efectivo de todos los estudiados.

La técnica aplicada para realizar el filtrado se basa en el siguiente pseudocódigo:

```
integer valorFinal = obtenerValorFinal(datos);  
media = media * CAPACIDAD_ARRAY_CIRCULAR;  
media = media - valorFinal;  
media = media + ultimo_valor_recibido;  
media = media / CAPACIDAD_ARRAY_CIRCULAR;  
insertarAlComienzo(ultimo_valor_recibido, datos);
```

La estructura de datos usada es un array circular capaz de almacenar 20 milisegundos de ECG, de esta manera se usa la mínima cantidad de memoria necesaria y se disponen de unos accesos a la estructura altamente eficientes.

Para poder ser usado correctamente, el filtro precisa una inicialización. Para ello los primeros 20 milisegundos de datos son almacenados en el array circular directamente, sin aplicar el código anterior, y no se transmiten al resto de sistemas de la app. Tras almacenar los datos se calcula su media para inicializar la variable `media`.

4.3.2. Comunicación Bluetooth

Como ya se ha indicado anteriormente, en la subsección 3.1.3 en la página 21, la placa Arduino envía los datos digitalizados a través de un módulo Bluetooth. En la app que se ha creado para este proyecto se implementa la recepción de dichos datos.

La implementación se ha realizado haciendo uso de las APIs estándar de Java y Android. Para facilitar la comprensión del resto de esta sección se describirán brevemente cuales han sido las partes usadas de dichas APIs.

La clase `BluetoothDevice` representa un dispositivo con capacidad de comunicación vía Bluetooth, en este proyecto dicho dispositivo es el módulo conectado a la placa Arduino. La aplicación móvil puede intentar conectarse a dicho dispositivo, haciendo uso de los métodos que brinda la clase anterior. Si la conexión es posible y se establece, entonces se obtiene un objeto de la clase `BluetoothSocket`. A través del `BluetoothSocket` se pueden extraer dos flujos para realizar el envío de datos al dispositivo y otro para la recepción, ambos a nivel de byte. Estos flujos son objetos instanciados a partir de las clases `InputStream` y `OutputStream` del paquete `java.io`.

La tasa de baudios a la que funciona el módulo Bluetooth no resulta relevante en este contexto, ya que Android utiliza el protocolo *RFCOMM* [5] para la comunicación Bluetooth.

El desarrollo de este módulo software ha partido de un proyecto personal: *Android_BT_Connection* [17]. Algunas de sus características más relevantes aprovechadas en este proyecto son:

- Una interfaz gráfica estándar y comprensible para la visualización y selección de dispositivos Bluetooth
- Un flujo de ejecución sencillo
- Control de errores (por ejemplo, evitar que el usuario se conecte a un dispositivo no disponible)

Por otra parte, se ha añadido una nueva función a *Android_BT_Connection*: la simulación de un `BluetoothSocket`. La razón por la que se ha añadido esta funcionalidad es la necesidad de poder hacer pruebas sobre la app sin tener el resto del hardware del proyecto disponible. El `BluetoothSocket` falso toma los datos que ofrece a través de su `InputStream` a la app, de un array predefinido. Dicho array contiene un electrocardiograma tomado anteriormente.

4.3.3. Flujo de trabajo con AsyncTask

Las aplicaciones Android trabajan con un único hilo de ejecución principal. Las interfaces gráficas se ejecutan sobre este hilo, junto al resto de tareas que ejecuta la aplicación. Mientras las tareas a ejecutar sean ligeras y no bloqueantes, la app será capaz de responder a la interacción del usuario, en caso contrario se bloqueará.

La creación de puertos (sockets) Bluetooth y la recepción continua de datos son tareas pesadas y probablemente bloqueantes, de manera que deben de ser trasladadas fuera del hilo de ejecución principal para garantizar al usuario una experiencia fluida. Existen múltiples maneras de llevar esto a cabo en Android, como el uso de servicios, creación de hilos manual o uso de objetos de la clase AsyncTask. La clase AsyncTask gestiona la creación y ejecución de un hilo de trabajo secundario y es extendible para crear clases derivadas con código personalizado. En la aplicación que aquí se estudia se han aplicado AsyncTasks para gestionar la lectura de datos del módulo Bluetooth, debido a su facilidad de uso.

Realizar las operaciones pesadas en un hilo secundario ha permitido mantener una interfaz gráfica capaz de reaccionar frente al usuario en todo momento y dibujar un electrocardiograma en la pantalla en tiempo real.

4.3.4. Interfaz gráfica

Durante el diseño e implementación de la interfaz gráfica se han seguido las guías de diseño de Android y consejos de programación de Google para crear una aplicación moderna y funcional. Para la creación de interfaces modulares se han aplicado las clases pertenecientes al estado del arte actual de la plataforma Android, los Fragment. La creación de menús interactivos se ha realizado mediante un objeto de la clase NavigationDrawer. Por último, para representar el electrocardiograma en tiempo real se ha creado una implementación personalizada a partir de la clase View.

Las interfaces gráficas en Android pueden ser definidas en un archivo XML y mostradas por la aplicación en ejecución, o creadas dinámicamente mediante código. En este proyecto se ha hecho uso de ambos enfoques.

Fragments

Fragment es una clase específica de las librerías de desarrollo de Android. Representa una sección de interfaz gráfica que puede ser reutilizada a lo largo de diferentes *Activities* (*pantallas*) de la aplicación. Los Fragment son útiles cuando se desea crear una aplicación dirigida a dispositivos con paneles de diferentes tamaños.

La razón principal por la que se han usado objetos `Fragment` en esta aplicación ha sido la integración de un `NavigationDrawer`, presentado en la próxima subsección, ya que este sólo funciona con ellos. La utilización de estas clases haría sencillo portar la aplicación a diferentes dispositivos en un momento futuro.

NavigationDrawer

El `NavigationDrawer` es un menú oculto que permite navegar entre las diferentes funcionalidades de la aplicación. Para poder interactuar con el menú este se debe hacer visible mediante un gesto, arrastrar un dedo desde el borde izquierdo de la pantalla hacia el centro. En dicho menú se presentan las diferentes funciones que presenta la aplicación, en este caso, visualizar un ECG en tiempo real o realizar un diagnóstico.

Una vez que el usuario visualiza el menú podrá seleccionar una de las opciones mediante un toque sobre la que desee. En ese momento el menú se ocultará y se mostrará la interfaz correspondiente a la acción escogida.

ElectroWaveView

Para dibujar el electrocardiograma en tiempo real se ha extendido la clase `View`. Esta clase de las librerías de Android representa el componente más básico de las interfaces gráficas. En la extensión personalizada de `View` se ha usado el área del componente gráfico como lienzo sobre el cual se dibujan los pixels deseados para representar en un ECG. Los pixels que se deseen representar en pantalla deben de ser listados de manera explícita, para ello se introducen en un array pares de coordenadas X/Y. Más adelante se transfiere el array a la función encargada de mostrarlos en pantalla.

A este objeto se le ha añadido un simple código que aplica una relación de linealidad entre los valores involucrados, para redimensionar los valores de tensión del ECG al tamaño vertical del componente gráfico. Esto evita que se pueda dibujar un valor que quede fuera de los límites del componente.

Una vez que el electrocardiograma llega al valor horizontal máximo del componente, un contador interno se reinicia para continuar el trazado desde el comienzo. De esta manera se obtiene una traza continua del ECG, a costa de perder los anteriores segundos de electrocardiograma. Una línea vertical indica cual es el valor más reciente que se ha dibujado, para poder interpretar correctamente los datos.

Interfaz dinámica

Para la impresión de los resultados del módulo de diagnóstico se ha creado una interfaz dinámica, ya que no se puede saber cual será el número de patologías cardíacas que detectará el sistema en cada ejecución.

Los resultados del módulo de diagnóstico se componen de una lista de tuplas de dos elementos. Cada tupla representa una patología o alteración cardíaca y el razonamiento que ha llevado a diagnosticar al sistema experto porque el paciente sufre esa patología (más información sobre esto en la sección 6.3.5 en la página 66). Un bucle recorre las tuplas y las descompone, dibujando en cada línea el nombre de la alteración cardíaca, un espacio en blanco y por último la razón que ha llevado al sistema a realizar el diagnóstico de dicho problema cardiovascular.

Capítulo 5

Software, extracción de características del ECG

La extracción de características de un ECG resulta necesaria de cara a la automatización del diagnóstico de un sujeto, ya que este último proceso necesita datos sobre los que trabajar. Para poder definir mejor el alcance del sistema descrito en este capítulo se fija la siguiente categorización de características de un ECG:

- Características de bajo nivel: estudio de ondas, segmentos o intervalos de manera individual, es decir, sin relacionarlos con el resto de elementos que aparecen a lo largo del ECG
- Características de alto nivel: estudio del ECG como un *todo*, sin fijarse en cada pulso, segmento o elemento de la onda individualmente

Los expertos realizan la extracción de características de manera intuitiva y disimilar a la de un algoritmo, fijándose en ambos tipos de características al mismo tiempo. Este comportamiento no resulta sencillo de expresar en un algoritmo. Por esta razón los estudios académicos realizados hasta el momento se centran únicamente en una de las dos categorías.

Existe una gran cantidad de artículos de investigación reunidos en el artículo [15], donde se realiza una revisión de la literatura existente. De todas formas la implementación de cualquier método de extracción de características, independientemente de la categoría de estos últimos, es complejo. Esto es debido a la poca o inexistente información técnica sobre la implementación que ofrecen la mayoría de los artículos y la elevada complejidad de parte de las técnicas aplicadas.

5.1. Algoritmos desarrollados

Debido a las razones citadas en el párrafo anterior se ha optado por desarrollar una serie de algoritmos sencillos de extracción de características. Estos algoritmos no pueden competir con los académicos o profesionales, pero sirven para sustituir su función con el propósito de mostrar un prototipo funcional.

Los algoritmos que aquí se presentan trabajan con el ECG a bajo nivel, es decir, analizan por cada pulso del corazón las características del mismo, pero no estudian el ECG total.

Téngase presente la Figura 2.7 de la página 12 para comprender qué secciones de la onda se desean detectar en cada una de las siguientes subsecciones.

5.1.1. Codificación de los datos

A lo largo del proceso de extracción de características los algoritmos trabajan con tres grandes estructuras de datos:

- Un array con el ECG en bruto, cada posición del array representa un momento en el tiempo y el contenido de dicha posición el valor de voltaje leído en ese momento
- Una lista de pulsos
- Un pulso, el pulso es una estructura que contiene las siguientes características correspondientes a un pulso del corazón:
 - La posición del comienzo de la onda P
 - La posición del final de la onda P
 - Si la P es positiva o no (si la onda no es positiva puede ser negativa o bifásica)
 - La posición del comienzo de la caída del pico Q
 - La posición del pico Q o comienzo del complejo QRS
 - La posición del pico R
 - La posición del pico S
 - La posición del final de la subida tras el pico S o final del complejo QRS
 - La posición del pico T
 - El final de la onda T

5.1.2. Detección de picos R

Los picos R son aquellos que más se elevan en un ECG. Partiendo de esta información se creó el siguiente algoritmo:

```

detectar todos los máximos de la onda;
ordenar los máximos según su magnitud (de mayor a menor);
for cada pico do
    if pico  $\geq$  valor_arbitrario then
        | marcar pico;
    else
        | romper bucle;
ordenar picos marcados en base a cronología (de más antiguo a más reciente);

```

El *valor_arbitrario* fue hallado mediante la ejecución de múltiples pruebas, el que mejor resultados ofreció fue $maximo_absoluto - maximo_absoluto * 0,22$.

Posteriormente se añadió una verificación de la inclinación del pico, ya que algunas ondas T especialmente altas interferían con el proceso de detección de picos R. El algoritmo de esta verificación posterior es el siguiente:

```

generar un vector desde el posible pico R al punto del ECG 30 ms a su izquierda;
generar un vector desde el posible pico R al punto del ECG 30 ms a su derecha;
hallar las inclinaciones de ambos vectores;
if inclinación del vector izquierdo es inferior a un valor arbitrario (A) O la inclinación del vector derecho es superior a otro valor arbitrario diferente (B) then
    | el pico no es una R;

```

Para hallar los valores arbitrarios de la condición **if** se ejecutó el algoritmo sobre una serie de electrocardiogramas y se registraron los valores de inclinación de los vectores. Tras estudiar estos valores y relacionarlos con picos R y ondas T, se definieron los valores. El valor arbitrario (A) para los vectores izquierdos es 4,0 y para los vectores derechos (B), -7,0.

5.1.3. Detección de complejos QRS

Tras detectar los picos R se procede a detectar el resto de los puntos característicos del complejo QRS. Estos puntos son, los picos Q y S, el comienzo del complejo a la izquierda

de la Q y el final del complejo a la derecha de la S. El algoritmo base para la detección del complejo es el siguiente:

```
descartar el primer pico R;  
while haya picos R do  
    encontrar pico Q;  
    encontrar pico S;  
    encontrar comienzo del QRS;  
    encontrar final del QRS;  
    pulso = crear_pulso(R, S, Q, comienzo QRS, final QRS);  
    añadirALista(pulso, lista_de_pulsos);  
    seleccionar siguiente pico R;
```

Dada la estructura del algoritmo, no se hallan las correspondientes características del complejo QRS para el primer y último pico R. Estos dos picos se descartan para evitar futuros problemas con condiciones limítrofes.

Los algoritmos para hallar los picos Q y S son extremadamente sencillos, tan sólo buscan el primer mínimo a la izquierda y derecha, respectivamente, del pico R. Los algoritmos para encontrar el comienzo y el final del complejo QRS tampoco son complejos. A continuación se explicará el algoritmo para encontrar el comienzo del complejo QRS, el algoritmo para hallar el final es análogo.

Para encontrar el comienzo del complejo QRS se recorre el ECG hacia la izquierda desde el pico Q. Tan pronto como la onda deja de ascender se sitúa el punto característico buscado.

A partir de este punto ya está formada la lista de pulsos mencionada en la anterior subsección 5.1.1, aunque las estructuras de datos que representan pulsos no poseen toda la información que pueden contener. De ahora en adelante se trabajará con la lista, junto al array de datos brutos para hallar el resto de características.

5.1.4. Detección de ondas P

La onda P resulta extremadamente difícil de detectar sin usar técnicas avanzadas como las que describen algunos artículos referenciados al comienzo de este capítulo. Los algoritmos aquí desarrollados no son sencillos, ni sólidos, debido al enfoque utilizado. Por estas razones el contenido de esta subsección se debe interpretar como prueba de concepto y no como una versión final.

La detección de las ondas P no consiste únicamente en hallar el comienzo y final de la onda, sino también su forma (si la onda es puntiaguda o suave) y si es inversa o no. Los algoritmos de detección de ondas P aquí descritos son incapaces de detectar ondas P difásicas.

Debido a que la onda puede ser positiva o negativa, muchas de las comprobaciones que se hacen posteriormente deberían “cambiar de signo”. Para aplicar este cambio dinámicamente sin reescribir el código para cada caso se ha hecho uso el patrón de diseño *Strategy*.

La detección de las características de la onda se ha dividido en diferentes pasos, cada uno con su propio algoritmo:

1. Detectar comienzo de la onda
2. Detectar orientación de la onda (si es positiva o negativa)
3. Vectorizar la onda P
4. Refinar el comienzo y el final de la onda P
5. Estudiar de la forma de la onda

La vectorización de la onda se basa en visualizarla como un conjunto de diferencias entre valores del ECG, y no como múltiples valores individuales consecutivos. La distancia entre valores tomada para la creación de los vectores ha sido hallada experimentalmente, siendo el mejor resultado una distancia de 10 valores. Es decir, una onda P compuesta por 80 valores será convertida a 8 vectores que definen la morfología de la onda, como puede verse en la Figura 5.1.

Si observamos los valores de los vectores podríamos deducir que la onda tiene un crecimiento fuerte al comienzo, un pico suave y una dura caída con una ligera cola.

5.1.5. Detección de ondas T

La detección de la onda T se realiza en base a los datos obtenidos del ECG hasta este momento, es decir, información del complejo QRS y de la onda P (de esta última interesa únicamente su comienzo).

Primero se detecta el valor máximo de la onda T. Para ello se toma un valor arbitrario entre el final del complejo QRS y el comienzo de la onda P del siguiente pulso y se recorre la onda hacia la izquierda en busca de un máximo. El el valor arbitrario ha sido hallado experimentalmente y se ha definido como *posición del final del complejo QRS + un tercio*

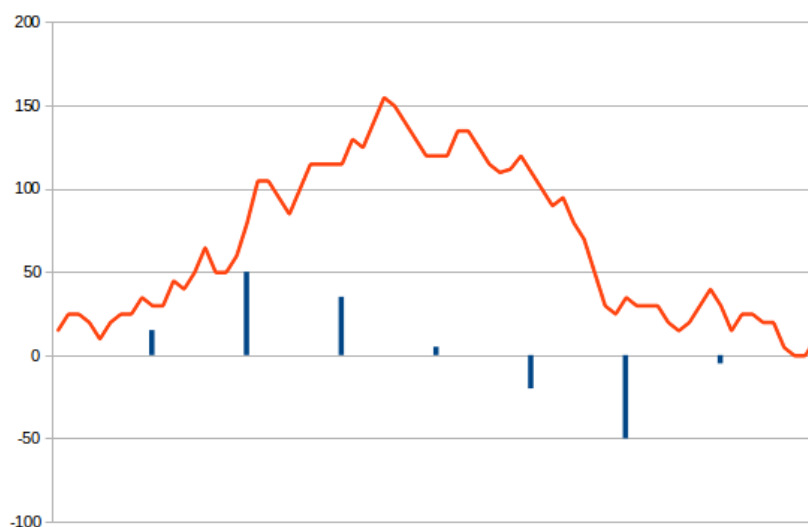


Fig. 5.1 Onda P ampliada (naranja) con vectores (en azul)

de la distancia entre el final del complejo QRS y el comienzo de la onda P del siguiente pulso. Mediante experimentación se ha comprobado que este valor es optimal.

Una vez conocido el pico de la onda T se procede a hallar su final. Para ello se sitúan dos puntos distanciados por 20 milisegundos a la derecha del pico T y se recorre el ECG hacia la derecha hasta que se detecta que los puntos ya no están descendiendo por la onda T. Este algoritmo lleva una pequeña comprobación adicional para asegurar que no se ha detectado un final de onda T falso, causado por una pequeña deformación de la onda debida al ruido.

Conocer el comienzo de la onda T resulta de poca importancia, ya que este punto se encuentra siempre muy cercano al final del complejo QRS, el cual ya conocemos. Debido a la naturaleza de prototipo de este proyecto se optará por usar el final del complejo QRS como comienzo de la onda T para reducir la complejidad de la extracción de ondas.

5.1.6. Detección de otros parámetros del ECG

Además de la extracción de puntos de interés en la onda, el sistema de extracción de características también detecta otros parámetros del ECG, como el pulso (*beats per minute* o BPM en inglés), longitud RR, longitud del complejo QRS, número de pulsos irregulares, cantidad de ondas P anormales y negativas, longitud del intervalo PR, longitud del intervalo QT corregido y la longitud de la onda P.

A continuación se explican como se han hallado estas características, aquellas cuya obtención es trivial han sido omitidas. La obtención de la cantidad de ondas P anormales y negativas es puramente experimental, por esta razón la explicación de su funcionamiento

también será omitida.

BPM y longitud RR

El pulso del paciente es relativamente sencillo de obtener, junto a la longitud RR. Este es el algoritmo utilizado:

```
picoAnterior = tomarPosicionSiguietePicoR();
while haya picos R do
    picoActual = tomarPosicionSiguietePicoR();
    longitudRR = picoActual - picoAnterior;
    longRRAcumulada = longRRAcumulada + longitudRR;
    picoAnterior = picoActual;
longRRMedia = longRRAcumulada / NUM_PICOS_R;
BPM = (60 * 1000) / longRRMedia;
```

Comprobación de ritmo regular

Un ritmo cardíaco regular se define como aquel en el que las distancias entre picos R es estable y apenas varía. En un ECG de ritmo irregular la distancia entre pulsos es extremadamente variable. Para definir un ECG como “de pulso irregular” se cuenta el número de pulsos irregulares que contiene.

Para hallar el número de pulsos irregulares únicamente se recorren las longitudes RR correspondientes a cada espacio entre pulsos y se comprueba si se encuentran dentro de un rango de valores. Dicho rango de valores se define según la Fórmula 5.1.

$$\textit{longitud RR media} \pm \textit{desviacion tipica.} \quad (5.1)$$

Por cada pulso irregular se aumenta en una unidad un contador.

Longitud del intervalo QT corregido

Hallar la longitud del intervalo QT es trivial. Aun así, este valor debe ser corregido, ya que cuanto mayor sea el pulso del paciente menor será la longitud de este intervalo, dificultando un diagnóstico posterior. Existen múltiples formas de corregir la longitud del intervalo QT. En este proyecto se ha escogido la corrección según la *fórmula de Bazett* [4], ya que es el método más extendido clínicamente.

La *fórmula de Bazett* es la siguiente:

$$QT_B = \frac{QT}{\sqrt{RR}} \quad (5.2)$$

La fórmula mostrada es aplicada a la longitud QT media formada entre todos los pulsos estudiados, obteniendo tras su aplicación la longitud QT corregida correspondiente al electrocardiograma.

5.1.7. Integración de los algoritmos

Todos los algoritmos estudiados a lo largo de las anteriores secciones han sido integrados en un único sistema monolítico de extracción de características. El enfoque de programación de este sistema ha sido principalmente procedural, debido a la naturaleza del análisis, el cual se realiza por pasos claramente definidos.

El sistema final aplica los algoritmos en el siguiente orden:

1. Extracción de picos R e inicialización de la lista de pulsos
2. Extracción de picos Q y S, extracción de comienzos y finales de complejos QRS
3. Cómputo del pulso, longitud RR, longitud QRS, comprobación de ritmo regular
4. Extracción de ondas P
5. Vectorización de ondas P
6. Análisis de ondas P y cómputo de la longitud del intervalo PR
7. Extracción de ondas T
8. Cómputo de la longitud del intervalo QT corregido

Este sistema es alimentado con un electrocardiograma bruto en forma de un array, el cual guarda en cada posición el valor de voltaje leído. Su salida ofrece una serie de variables que contienen los aspectos del ECG más relevantes.

Por último, resulta interesante destacar que los algoritmos de estas secciones son independientes de la frecuencia de muestreo del ECG. Es decir, siempre que se aplican valores temporales en milisegundos, estos son transformados a la cantidad de posiciones equivalente en el array de datos original.

Mediante un ejemplo: Si el array ha sido generado tomando 500 muestras por segundo (es decir entre muestra y muestra pasan 2 milisegundos) y se desea avanzar 10 milisegundos los algoritmos avanzan 5 posiciones. Si la frecuencia de muestreo es 1000 muestras por segundo, entonces los algoritmos avanzan 10 posiciones.

5.2. Aseguramiento de la calidad

Sobre el sistema aquí propuesto se ha realizado una extensa serie de pruebas usando electrocardiogramas de diferentes orígenes y con diferentes alteraciones cardiovasculares para comprobar la estabilidad y corrección del mismo. Estas pruebas han sido puramente cualitativas, lo cual significa que el aseguramiento de la calidad podría mejorarse mediante pruebas sistemáticas, siguiendo un plan de calidad previamente diseñado. Existe más información al respecto en el apartado correspondiente de posibles mejoras y trabajo futuro (subsección 9.5, página 85).

A continuación se presentan los mecanismos usados para probar el sistema.

5.2.1. Ejecución sobre electrocardiogramas de Physiobank

Physiobank [13] es una importante colección base de datos de señales fisiológicas aportadas por la comunidad científica. En dicha colección existen múltiples grabaciones de electrocardiogramas, las cuales suelen estar etiquetadas en base a las patologías cardíacas detectadas en ellas.

Para probar el sistema aquí descrito se han tomado grabaciones de las bases de datos de las categorías 1 y 2, las más utilizadas y citadas por la comunidad. Las grabaciones usadas han sido mitdb-100, mitdb-203, mitdb-210 y mitdb-230 de MIT-BIH Arrhythmia Database [19]. Edb-e0107, edb-e0108 y edb-e0111 de European ST-T Database [24]. Ecgiddb-p01-r1, ecgiddb-p02-r16 y ecgiddb-p09-r1 de The ECG-ID Database [16].

El mecanismo de aseguramiento de la calidad siguió los siguientes pasos: los electrocardiogramas fueron estudiados por un experto previamente para extraer de manera manual las características que más adelante el sistema extraerá automáticamente. Después se procedió a aplicar un preproceso a los datos para adaptarlos al rango con el que trabaja el sistema de extracción. Por último se realizaron las ejecuciones necesarias sobre una variante del sistema de extracción con funciones de depuración, y se comprobó que los valores hallados por el sistema se correspondían a los localizados manualmente.

Con esta metodología se detectaron múltiples pequeños problemas relativos a condicio-

nes limítrofes, bucles con condiciones de salida incorrectas, inconsistencias entre unidades o errores en cálculos matemáticos.

5.2.2. Ejecución sobre electrocardiogramas de la placa Olimex

Además de usar electrocardiogramas profesionales en el sistema de extracción de características, éste también se probó mediante ECGs tomados por las placas de adquisición de datos disponibles, principalmente la placa Olimex, la cual ya se ha estudiado en el apartado 3.2.3 en la página 29. La metodología aplicada fue idéntica a la de la subsección anterior, a diferencia del preprocesado para adaptar los datos, el cual era innecesario ya que en este caso se estaba usando la fuente de datos original.

A continuación, en la Figura 5.2, se muestran tres ECGs de sujetos diferentes sobre los cuales el sistema de extracción de características ha hallado los puntos relevantes. Puede observarse como la forma del ECG, diferente entre los sujetos, afecta fuertemente a la detección de las características de la onda.

Los complejos QRS y sus correspondientes picos suelen ser identificados correctamente en la mayoría de las ocasiones. Las ondas P no suelen ser detectadas si la onda no es de gran calidad o no existe un segmento PR suficientemente claro. Las ondas T suelen ser identificadas correctamente, pero algunos problemas con la definición del final del complejo QRS causan inconsistencias.

Si la onda generada por la placa Olimex fuera más limpia y el ruido pudiera eliminarse por completo, el sistema de extracción de características debería ser capaz de extraer correctamente todos los puntos importantes de la onda.

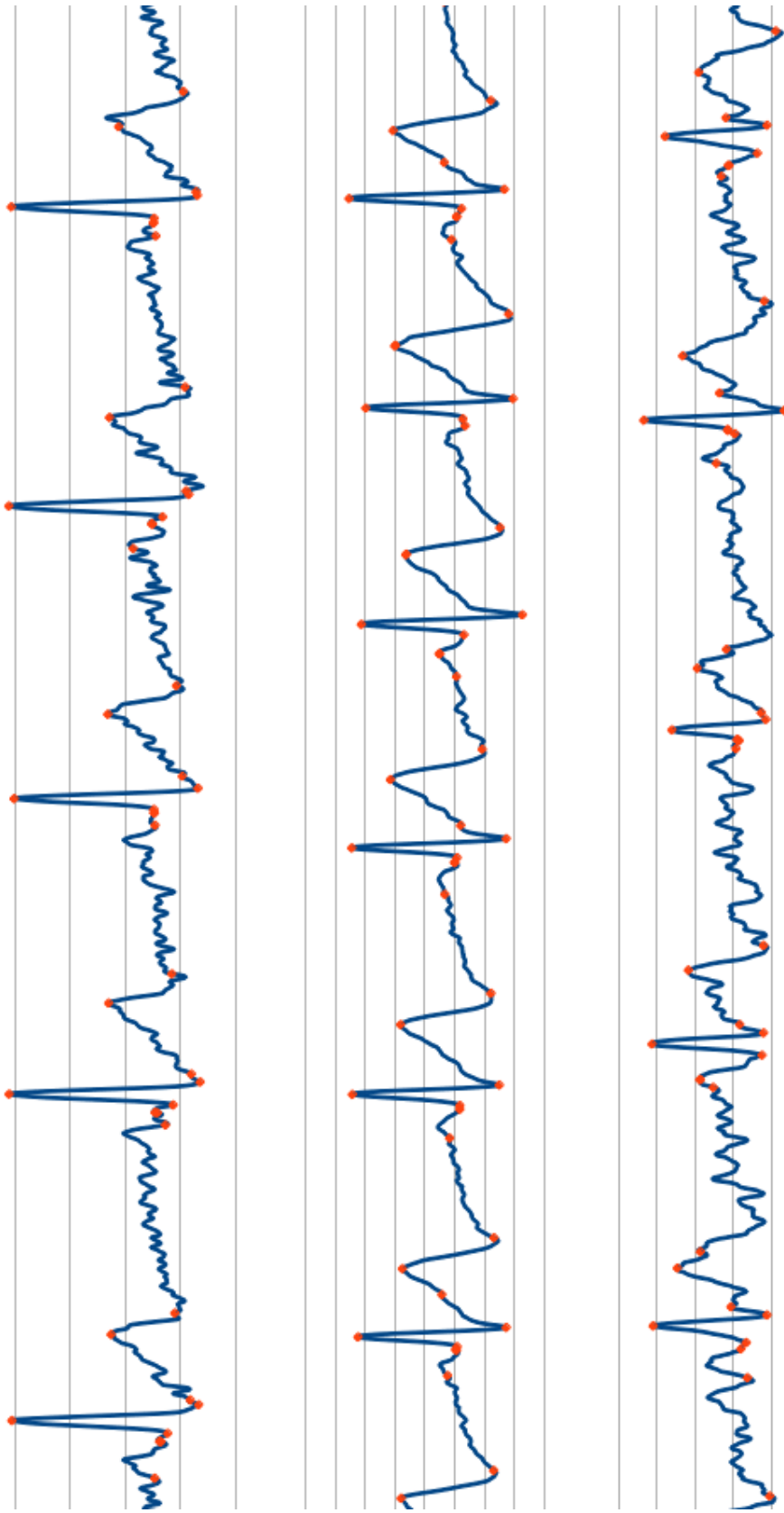


Fig. 5.2 Múltiples electrocardiogramas con puntos característicos, éstos están representados en naranja. En una situación correcta los puntos representan (el orden es relevante): comienzo de onda P, pico Q, pico R, pico S, final del complejo QRS, pico T, final de la onda T

5.2.3. Ejecución sobre electrocardiogramas sintéticos

Existe una herramienta llamada ECGSYN [18] desarrollada por un grupo de investigadores y contribuida a Physionet la cual es capaz de crear electrocardiogramas sintéticos realistas. Se han realizado pruebas con esta herramienta para generar electrocardiogramas con características de especial interés para probar alguna parte específica del sistema de extracción de características.

Aunque la herramienta ha demostrado ser de gran utilidad, a través de su aplicación en artículos académicos como [25], los electrocardiogramas generados no resultaron útiles para este proyecto. Las razones son dos: por una parte, si se genera un electrocardiograma sintético sin ruido, la onda es demasiado poco realista, haciendo que el sistema de extracción de características falle, ya que se aprovecha de las leves deformidades que el ruido añade a la onda. Si se añade ruido, este es generado aleatoriamente y no puede ser atenuado por el sistema de filtrado desarrollado en este proyecto, haciendo que la onda sea ilegible para el sistema de extracción.

5.3. Selección manual de regiones

Como alternativa a la extracción de características automática se desarrolló un sistema mediante el cual una persona, con conocimientos suficientes para leer un electrocardiograma, pudiera ayudar en el proceso de extracción de características. Este sistema fue desarrollado como una aplicación Android para facilitar una posible integración posterior en la app principal creada a lo largo del proyecto.

La aplicación de apoyo a la extracción, *Region-Select*, de ahora en adelante, muestra un segmento de electrocardiograma que contiene de uno a unos pocos pulsos. *Region-Select* guía al usuario a través de un proceso mediante el cual, éste debe seleccionar las regiones en las que se encuentren una onda P, un complejo QRS y una onda T. Tras etiquetar las regiones, *Region-Select* permite al usuario confirmar sus selecciones o corregirlas.

Si un usuario competente en cardiología selecciona las regiones indicadas, una gran parte de la complejidad del sistema de extracción de características es eliminada. La calidad de las características calculadas y del diagnóstico posterior aumentará considerablemente. Las desventajas de este sistema es que no resulta tan rápido o sencillo de usar como un sistema automatizado y precisa alguien con conocimientos de electrocardiografía para seleccionar las regiones.

A continuación, en la Figura 5.3 se muestra una captura de pantalla de la app. En ella

se pueden observar las regiones correspondientes a la onda P y complejo QRS, ya seleccionadas. En el momento en el que se ha tomado la captura el usuario está seleccionando la región correspondiente a la onda T y posteriormente pulsará el botón “Aceptar” para finalizar el proceso.

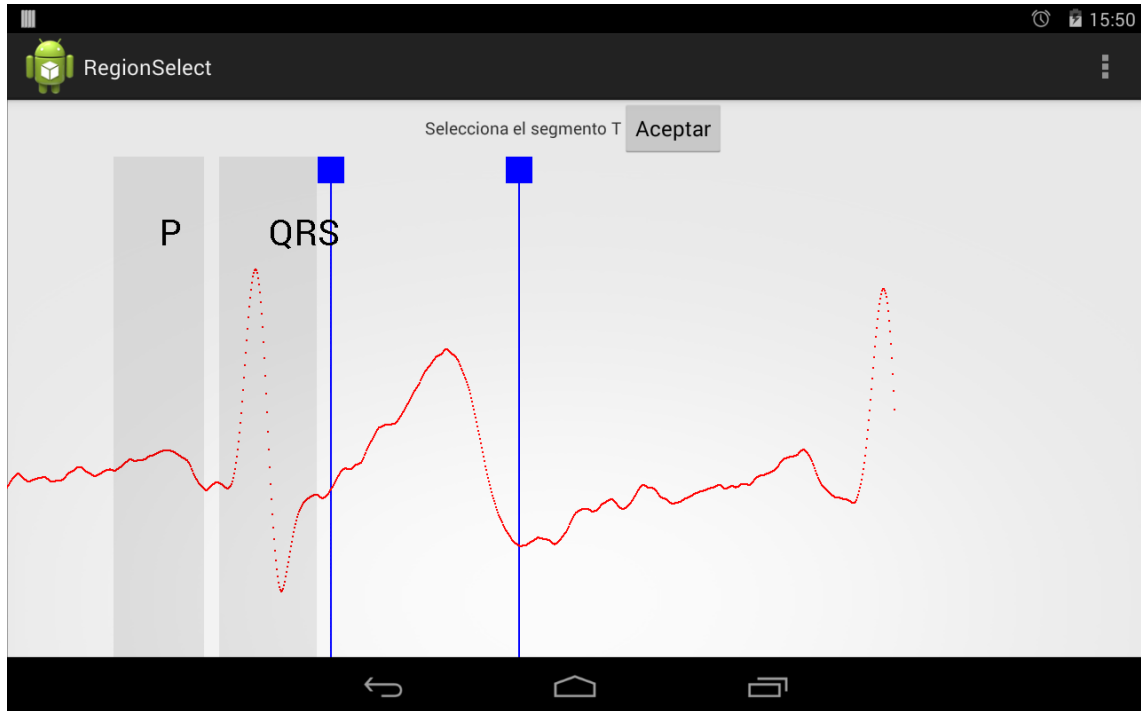


Fig. 5.3 Captura de pantalla de Region-Select

Capítulo 6

Sistema experto

La inclusión de un módulo que realice un diagnóstico sobre el electrocardiograma de un sujeto requiere la existencia de algún tipo de inteligencia. Existen múltiples métodos basados en diferentes técnicas para asociar un ECG a una serie de patologías cardiovasculares, como técnicas de minería de datos, redes neuronales o inteligencia artificial.

Gran cantidad de recursos en el mundo de la medicina presentan técnicas de diagnóstico a través de diagramas de bloques o conjuntos de reglas. Este formato en el que se codifica la información médica se presta perfectamente para su adaptación a un sistema experto. La existencia de libros médicos donde se recoge esta información hace que el conocimiento sea fácilmente accesible a lo largo del desarrollo del proyecto.

6.1. Motores para sistemas expertos en Android

Crear un sistema experto completo desde la base resulta imposible dadas las restricciones temporales de este proyecto, además de presentar una gran complejidad que no se desea incluir en el proyecto. La única alternativa a crear un motor para un sistema experto es usar uno ya existente.

Existen múltiples motores de inferencia para crear sistemas expertos, como *CLIPS*, *Jess* o *Drools* entre otros. La integración de estos sistemas en Android está muy poco madura y es imposible en la mayoría de los casos. Únicamente *CLIPS* y *Jess* parecen ser candidatos aceptables para ser utilizados en Android.

6.1.1. CLIPS

CLIPS [7] está programado en el lenguaje C y ha sido publicado como aplicación independiente o librería. La *Java Native Interface* (JNI de ahora en adelante) es una librería de Java que permite a la máquina virtual (JVM) interactuar con código nativo a un dispositivo y librerías escritas en C, C++ y ensamblador. A lo largo de los últimos años se han realizado múltiples esfuerzos por diferentes comunidades de desarrolladores con el objetivo de portar CLIPS a Java utilizando la JNI. El resultado más destacado se encuentra en la página del proyecto CLIPS [8].

Aitor Gómez Goiri, un investigador de la universidad de Deusto, ha tomado el proyecto CLIPS JNI y ha realizado una adaptación a Android usando la librería *Native Development Kit* (Android NDK de ahora en adelante). Su adaptación está disponible públicamente como proyecto en github bajo el nombre CLIPS4Android [14].

CLIPS4Android ofrece la integración de código Android con CLIPS a través de unas clases que actúan de puente entre ambos sistemas. Estas clases representan hechos, valores multcampo o monocampo, direcciones de memoria e incluso el propio entorno CLIPS que se esté ejecutando. La interacción con estas clases se realiza mediante una pequeña API que se ofrece públicamente.

6.1.2. Jess

Jess [23] es un motor de inferencia programado en su totalidad en código Java. Parece lógico usar esta implementación de un motor de inferencia, ya que las aplicaciones Android también se programan en código Java y la combinación debería ser sencilla.

En realidad, esto no es así. La utilización de Jess en Android es imposible debido a que Jess hace uso de las clases del paquete *java.beans*, no disponible en la plataforma Android. Llegado este punto se desecharon todas las ideas que consideraran el uso de Jess en este proyecto.

Durante la realización de esta memoria se ha visitado de nuevo la página principal de Jess. Según sus últimas noticias, la versión más reciente de Jess (8.0) es compatible con Android. Este hecho hace que se deba considerar portar el sistema experto de este prototipo a Jess si el proyecto se continua desarrollando más adelante.

6.1.3. Elección final

Dadas las circunstancias se ha tomado el proyecto **CLIPS4Android** como base para el motor de inferencia del sistema experto a desarrollar. El flujo de trabajo y programación de este sistema es idéntico al ya utilizado para programar CLIPS a lo largo del Grado en Ingeniería Informática, de manera que no se han dado complicaciones en su desarrollo.

El código de CLIPS4Android se encuentra en dominio público, eliminando así cualquier posible conflicto con la propiedad intelectual del mismo y permitiendo su uso en este proyecto.

6.2. Integración de CLIPS en Android

La integración de CLIPS4Android en el resto de la aplicación fue complicada, aunque este hecho pudo ser mitigado a través de la documentación accesible. Dicha documentación toma la forma de sencillos programas de ejemplo comentados que demuestran diferentes aspectos del funcionamiento del sistema experto en el entorno Android. La carga de reglas a partir de ficheros de texto, inicialización del motor, ejecución del sistema y la posterior extracción del conocimiento generado fueron algunos de ellos.

A la hora de programar, CLIPS4Android se puede usar de dos maneras: como una librería autocontenida o como un proyecto al que se enlaza desde una app. La primera variante resulta complicada y precisa la compilación de parte del código de CLIPS JNI, la cual puede ser problemática. La segunda variante resulta extremadamente sencilla usando un IDE como Eclipse, de manera que ha sido la elegida. A la hora de exportar la aplicación a un fichero instalable en dispositivos Android el entorno se encarga de resolver las dependencias e incluir en el fichero Android el código necesario automáticamente.

6.2.1. Limitaciones de CLIPS bajo Android

Al usar una migración a Android de una adaptación Java de CLIPS las limitaciones del sistema experto, respecto a una instalación de su versión nativa en un ordenador personal, son múltiples.

La limitación más destacable es la obligación de acceder al sistema experto a bajo nivel. Los comandos más avanzados que agrupan múltiples pequeñas tareas, no están disponibles. Un ejemplo de ello es el comando (`batch . . .`), usado para cargar múltiples ficheros al mismo tiempo. El acceso a los hechos devuelve direcciones de memoria a los objetos que maneja

el sistema y no sus contenidos, los cuales deben de ser accedidos mediante la única función existente para ello.

Como ya se ha mencionado en el párrafo anterior la carga simultánea de múltiples ficheros es imposible. Por esta razón se deben crear los objetos necesarios para alimentar al sistema experto correctamente y lanzar la carga del fichero por cada uno que se desee cargar. Esto resulta tedioso y multiplica las posibilidades de fallo.

La inclusión de los ficheros de datos en una aplicación instalable en Android también resulta complicada, ya que el motor CLIPS no acepta las rutas a las carpetas estándar que usa Android en las aplicaciones. Para resolver este problema se incluyeron los ficheros de datos en la carpeta *assets* y antes de cargarlos eran copiados a la carpeta interna del sistema operativo destinada a la aplicación.

El acceso a los hechos residentes en el sistema experto también es problemático. Los hechos ordenados pueden ser extraídos del sistema, pero sus campos no pueden ser fácilmente accedidos. Por esta razón es preferible usar hechos no ordenados.

La extracción de campos de los hechos se realiza mediante una función en la cual se indica el nombre del campo cuyos datos se desean extraer. Esta función devuelve un objeto que extiende la clase `PrimitiveObject`, pero a priori no se sabe a qué subclase de la anterior pertenece el objeto en cuestión. El programador debe conocer estas limitaciones y mantener un registro manual de los tipos de los campos que se manejan en el sistema experto para poder extraerlos correctamente.

Por último resulta interesante destacar que el módulo principal, *MAIN*, no puede ser redefinido en esta versión de CLIPS salvo que no esté seleccionado. Desafortunadamente éste es el módulo seleccionado mientras se cargan los ficheros. Esto evita que se puedan añadir en él sentencias *import* o *export* salvo que se aplique algún enrevesado truco.

6.3. Estructura del sistema experto implementado

El sistema experto creado hace uso de características aún no implementadas en anteriores módulos. Las plantillas cuentan con campos para contener información que aún no se extrae en la fase anterior del proyecto y existen reglas que actúan sobre esta información que jamás podrán ser lanzadas. A pesar de lo absurdo que resulta este hecho, el razonamiento que lo soporta es relativamente simple: crear un sistema experto lo más rico posible para aprovechar las ventajas que brinda el desarrollo del mismo, como una formación más completa sobre el funcionamiento interno de CLIPS y mayor pericia en el desarrollo de sistemas expertos.

La programación del sistema experto se ha dividido en seis ficheros, uno de ellos existe por conveniencia. El resto de ficheros incluyen sendos módulos. De esta manera, al actualizar el sistema experto únicamente se modifica la mínima cantidad de ficheros necesaria.

Los módulos y ficheros generados han sido los siguientes:

- Definición de plantillas (*electro_templates.clp*)
- Creación de hechos iniciales (*electro_facts.clp*)
- Módulo de inicio del sistema (*electro_rules.clp*)
- Módulo de preprocesamiento de datos (*electro_rules_data.clp*)
- Módulo de diagnóstico (*electro_rules_diagnostic.clp*)

El flujo del programa ha resultado complicado de establecer, debido a la necesidad de hacerlo genérico en cuanto a pasos de ejecución y la imposibilidad, a priori, de redefinir el módulo principal impuesta por la adaptación utilizada de CLIPS. Para evitar estas limitaciones, antes de cargar el fichero que redefine el módulo MAIN se carga un fichero con un módulo vacío y establece como módulo actual mediante el comando (`set-current-module . . .`). A partir de este punto se carga el resto de ficheros con sus respectivos módulos. Durante el proceso se puede redefinir el módulo MAIN.

Tras realizar esta maniobra, el flujo de ejecución es relativamente sencillo:

1. Añadir un hecho que representa el electrocardiograma a partir de los datos obtenidos por el sistema de extracción de características
2. Ejecutar el sistema
 - a) Realizar preproceso de datos
 - b) Realizar diagnóstico
3. Extraer resultados del sistema desde la app
4. Mostrar los resultados del sistema

A continuación se destacan los aspectos más relevantes de cada faceta del sistema experto creado.

6.3.1. Plantillas

Se han usado dos plantillas (*templates*) en el sistema experto. Una de ellas es una estructura que contiene las características que forman un electrocardiograma. Estas mismas características son aquellas que detecta el sistema descrito en el Capítulo 5. La segunda plantilla creada representa la estructura de un diagnóstico.

A continuación se extiende la información sobre estas plantillas.

Electrocardiograma

La estructura que representa la información extraída del electrocardiograma cuenta con los siguientes campos:

- Longitud del complejo QRS
- Amplitud del complejo QRS
- BPM
- Longitud del intervalo RR
- Amplitud de la onda P
- Longitud de la onda P
- Longitud del intervalo PR
- Longitud QT corregida
- Pulsos irregulares
- Amplitud de la onda Q
- Amplitud de la onda R
- P difásica
- Amplitud de la onda U

Las unidades de voltaje utilizadas son milivoltios, y las unidades de tiempo milisegundos.

Los campos *pulsos irregulares* y *P difásica* codifican diferentes situaciones mediante un número entero. A continuación se indica su significado.

Pulsos irregulares:

- 0 = Se dan 2 o menos pulsos irregulares
- 1 = Existen 3 o más pulsos irregulares, precedidos de onda P
- 2 = Existen 3 o más pulsos irregulares, no precedidos de onda P

P difásica:

- 0 = Las ondas P no son difásicas
- 1 = Las ondas P son difásicas, primero positivas y luego negativas
- 2 = Las ondas P son difásicas, primero negativas y luego positivas

Estos dos campos se podrían haber definido como campos de tipo *SYMBOL* y especificado sus valores de forma clara mediante la sentencia *allowed-values*.

Diagnóstico

La estructura diagnóstico representa la forma en la que se guardarán los resultados de la ejecución del sistema experto. Cuenta con dos campos, ambos multivalor: *patología* y *razonamiento*. Es evidente que contendrá cada uno de ellos. Los campos son multivalor, ya que un sujeto puede sufrir más de una patología o alteración cardíaca en un mismo momento.

6.3.2. Hechos

Durante la ejecución del sistema experto hechos ordenados y no-ordenados conviven en la memoria de trabajo del motor de inferencia.

Hecho no ordenados

Los hechos no ordenados son generados al comienzo de la ejecución del sistema.

El hecho que materializa la estructura del electrocardiograma es insertado en el sistema a través de la ejecución del comando (`assert . . .`). Esta estructura se mantiene intacta hasta el final de la ejecución del sistema. Únicamente es accedida por algunas reglas para extraer la información necesaria del ECG.

El hecho no ordenado restante es aquel en el que se guardan las patologías cardíacas, junto a la razón que ha llevado al sistema experto a realizar el diagnóstico de dicha patología. El orden de los elementos en este hecho es importante, ya que cada razonamiento se asocia en la app Android a la patología con el mismo índice. Es decir, la aplicación asume que el primer razonamiento justifica la primera patología, el segundo razonamiento la segunda patología, etcétera.

A continuación se muestra un ejemplo de un posible estado del hecho que aplica la estructura *diagnóstico* al final de la ejecución del sistema experto:

```
(diagnostico
  (patologias nil bloqueo-de-rama taquicardia)
  (razonamientos nil QRS-largo pulso-elevado)
)
```

Hechos ordenados

Los hechos ordenados se utilizan como pequeños extractos de información del ECG y transmisores de éstos a través de los módulos de preprocesado de datos y diagnóstico. Se añaden a la memoria de trabajo a lo largo de la ejecución del sistema a través de reglas

que hacen uso del comando (`assert ...`), y son eliminados por otras reglas a través del comando (`retract ...`).

6.3.3. Módulos

El sistema ha sido dividido en una serie de módulos, cada uno contenido en un fichero independiente, para facilitar el control de la ejecución del sistema.

A continuación se presentan los módulos existentes, junto a los módulos que importan y su cometido:

Tabla 6.1 Módulos creados para el sistema experto

Nombre	Importaciones	Descripción
TMP	Ninguna	Permite redefinir el módulo MAIN
TEMPLATES	Ninguna	Almacena las plantillas para los hechos no ordenados del sistema experto
FACTS	TEMPLATES	Añade a memoria el hecho no ordenado que representa el diagnóstico generado por el sistema
MAIN	FACTS, TEMPLATES	Punto de parada del sistema para introducir el hecho no ordenado que representa el ECG. También indica el orden de ejecución de los siguientes módulos
INTERPRET-DATA	MAIN, TEMPLATES	Genera hechos no ordenados que representan pequeños segmentos de información del ECG
GET-DIAGNOSTIC	TEMPLATES, FACTS, INTEPRET-DATA	Realiza un diagnóstico en base a los hechos generados por INTERPRET-DATA

El módulo MAIN inicializa la pila de *focus*, añadiendo a ella el módulo de preprocesamiento de datos, seguido del de diagnóstico. Las plantillas y hechos también han sido definidos en módulos propios para ofrecer la posibilidad de ocultar información entre módulos, si es necesario, en una implementación futura.

6.3.4. Reglas

Los módulos INTERPRET-DATA y GET-DIANGOSTIC son los únicos que contienen reglas. Las del primero preprocesan los datos obtenidos del ECG y las del segundo realizan

el diagnóstico automatizado. Las reglas de GET-DIAGNOSTIC están íntegramente programadas según el paradigma de programación por reglas y programación funcional, mientras que en las reglas del módulo INTERPRET-DATA se han añadido algunas líneas de código más afines a la programación imperativa. El uso de sentencias imperativas se materializa principalmente en estructuras de control *si... entonces... sino...* las cuales permiten una gestión más fina de algunos aspectos y evitan las repeticiones de código que conlleva reescribir la misma regla con ligeras modificaciones.

La necesidad de implementar una serie de reglas que preprocesen los datos del electrocardiograma se basa en evitar bucles infinitos en la realización del diagnóstico. Las reglas de realización de diagnóstico *matchean* una característica del ECG y el hecho en el que se guarda el diagnóstico. Es necesario capturar una característica del ECG para tener realizar un diagnóstico y es necesario conseguir una forma de acceder a la estructura de diagnóstico para guardar el resultado de ejecución de la regla. Si la regla modifica el diagnóstico y la característica del ECG sigue presente en el hecho que la contiene, la regla se reactivará, causando un bucle infinito de activaciones.

El preproceso de los datos genera pequeños segmentos de información sobre el ECG en forma de hechos ordenados. Estos hechos ordenados pueden ser *matcheados* por las reglas de diagnóstico y después eliminados, previniendo una reactivación de la regla.

A continuación se muestran dos de las reglas del sistema experto a modo de ejemplo. La primera pertenece al módulo de preproceso de datos, la segunda al módulo de diagnóstico:

```
(defrule tratar-pulso "Comprueba si el pulso esta fuera de los
limites saludables"
  (electro (BPM ?frec &(> ?frec 100)))
=>
  (if (> ?frec 100)
    then (assert (BPM demasiado-elevada))
    else (if (< ?frec 40)
            then (assert (BPM demasiado-baja))
          )
    )
)

(defrule diangosticar-taquicardia
  ?BPM-fact <-(BPM demasiado-elevada)
  ?diag <- (diagnostico (patologias $?pat)(razonamientos $?raz))
```

```
=>
  (modify ?diag (patologias ?pat taquicardia)
           (razonamientos ?raz pulso-elevado)
  )
  (retract ?BPM-fact)
)
```

6.3.5. Características especiales

Como ya se ha indicado en múltiples ocasiones, el sistema experto aquí desarrollado incluye la capacidad de justificar sus decisiones. Los beneficios de este comportamiento son múltiples.

Por una parte se facilita la comprensión de la ejecución del sistema, desde el punto de vista de desarrollo y depuración. Por otra parte permite a un experto tener una segunda opinión válida sobre el estado del paciente, ya que las decisiones del sistema estarán apoyadas en una base científica. Por último, en caso de que un usuario inexperto en cardiología esté usando el sistema, éste podrá comprender cuales son las razones por las cuales el sistema razona de una forma y no de otra, pudiendo aprender del diagnóstico realizado.

Parte II

Expansión y mejoras

Capítulo 7

Electrónica

La fase de electrónica del proyecto puede mejorarse trabajando con mayor detalle los filtros por software aplicados y realizando nuevas pruebas con las placas de adquisición bajo condiciones diferentes. Las oportunidades de expansión son extensas, comenzando por uno de los puntos del alcance que fue eliminado a lo largo del proyecto: la toma de datos de 4 electrodos.

También se estudiará como mejorar el proyecto de cara a abaratar los costes de adquisición de los componentes que conforman la parte electrónica.

7.1. Mejora del filtrado por software

Las técnicas de filtrado por software pueden mejorarse en unos cuantos aspectos. Las técnicas de filtrado descartadas deberían ser revaluadas. Además la investigación de más técnicas de filtrado paso alto para eliminar el ruido causado por la respiración también deberán ser retomada.

Una interesante prueba a realizar en un futura reevaluación de las técnicas de filtrado consiste en aplicar los filtros a pequeños subconjuntos del total de datos, de manera análoga al filtro basado en la media móvil presentado en al subsección 3.2.4 en la página 32 de esta memoria.

7.2. Revaluación de las placas

La placa AD8232Eval fue utilizada sin cables apantallados. Usar este tipo de cables para conectar los electrodos y la salida de datos quizás pueda hacer que la placa ofrezca un mejor

rendimiento. Para ello únicamente habría que modificar los conectores en la superficie de la placa.

Otra posibilidad de mejora para las placas consiste en tomar un electrocardiograma con una de ellas a la vez que con un electrocardiógrafo profesional, de esta manera se podrían contrastar las lecturas de ambos dispositivos. La comparación entre lecturas permitiría calibrar de manera precisa las placas de adquisición de datos aquí presentadas.

7.3. Ampliación a seis derivaciones

Esta expansión se aplica a todas las fases del proyecto. En la parte de electrónica consiste en ampliar el número de electrodos de los cuales se toman los datos y tomar dichos datos de manera diferente. Mediante cuatro electrodos, uno en cada muñeca y uno en cada tobillo del sujeto, pueden obtenerse tres fuentes de información y una referencia.

A partir de los datos tomados de los tres electrodos de información es posible hallar seis derivaciones sobre el paciente: I, II, III, aVL, aVR y aVF. Las ventajas de contar con seis derivaciones son claras: se obtiene más información sobre el paciente. Esta información permite: que el módulo de diagnóstico ofrezca unos resultados más fiables, que el mismo módulo ofrezca la identificación de un mayor número de alteraciones cardíacas y por último que un experto posea más información al usar la función de visualización del ECG de la app Android.

La forma de hallar las derivaciones en base a las señales de los electrodos es sencilla, tan sólo consiste en realizar una serie de operaciones aritméticas sobre dichas señales. Por ejemplo, para hallar la derivación III se deberá aplicar la fórmula 7.1, y para hallar la derivación aVR, la fórmula 7.2. LL representa la señal del electrodo del tobillo izquierdo, LA la señal del electrodo de la muñeca izquierda y los números romanos las derivaciones correspondientes.

$$\text{Derivación III} = LL - LA \quad (7.1)$$

$$\text{Derivación aVR} = -\frac{I + II}{2} \quad (7.2)$$

Para tomar datos de tres fuentes de información independientemente, y no de una formada por la combinación de dos electrodos como en el prototipo presentado, las placas de adquisición de datos deberán tener un funcionamiento diferente.

7.3.1. Usando placas Olimex ECG Shield

Las placas Olimex ECG Shield permiten su uso conjunto para realizar una toma de datos independiente de varios electrodos. Estas placas pueden montarse encima de otras idénticas, pudiendo crear una torre de tres Olimex ECG Shield sobre un Arduino, como se muestra en la sección 1.2 del manual de la placa Olimex [21]. Para evitar que las placas interfieran entre si se han de configurar mediante el jumper AIN_SEL.

7.3.2. Mediante placas de fabricación propia usando el chip AD8232

Como alternativa al uso de múltiples placas Olimex ECG Shield podría diseñarse una placa basada en el chip AD8232. Dicha placa contaría con un conector para cada uno de los cuatro electrodos y tres chips AD8232 para amplificar independientemente las señales de cada electrodo de datos.

Para obtener una señal independiente por cada electrodo usando la etapa de amplificación diferencial del chip AD8232 hay que introducir en el amplificador la señal correspondiente a un electrodo y la señal de potencial de referencia del sujeto. De esta manera el ruido se anulará y la señal del electrodo es la única amplificada.

7.3.3. Nuevos problemas al hallar seis derivaciones

Debe tenerse en cuenta que la velocidad de muestreo bajo estas nuevas condiciones sea suficiente, ya que si no lo es, puede que se forme un desfase entre las 3 ondas que se tomen de los electrodos. Si las ondas están desfasadas entonces no se podrán aplicar las fórmulas para hallar las derivaciones más adelante (en la app), ya que no se estará operando sobre las ondas en el mismo momento del tiempo.

Existen dos formas de evitar este problema: la primera es capturar y convertir las tres ondas en una extensión de tiempo tan pequeña que el desfase entre ellas sea despreciable. La extensión de tiempo que recomendaría es 1 milisegundo. En el caso de aplicar esta solución tan sólo se hace uso de tres conversiones por milisegundo. Es factible usar el ADC del chip ATmega328 de la placa Arduino, ya que éste soporta 9600 conversiones por segundo o, lo que es lo mismo, 9,6 conversiones por milisegundo, como se indica en la subsección 3.1.2 en la página 19.

La segunda forma de evitar el problema del desfase de ondas es convertir todas de formato analógico a digital al mismo tiempo, para ello habría que hacer uso de 3 conversores analógico digital que operen en paralelo. Para llevar esta idea a cabo puede tomarse como

punto de partida el diagrama de la figura 68 de la hoja de especificaciones del chip AD8232 [2]. En dicho diagrama se usa un ADC ADuCM360 para convertir la señal del chip AD8232 y enviarla posteriormente a una plataforma para su transmisión.

7.4. Mejora del protocolo de comunicación

El protocolo de comunicación estudiado en 3.1.3 en la página 20 es muy ineficiente. Actualmente se envían 4 bytes de datos por cada lectura.

El ADC del Arduino genera valores entre 0 y 1023, de manera que con dos bytes ya se puede codificar el valor. No es necesario un carácter de control si cada medición tiene un tamaño en bytes fijo, de manera que el número de máximo de unidades mínimas de información a enviar será 16 bits, o dos bytes.

Las formas de codificar y enviar los datos a bajo nivel son múltiples. Una posible propuesta es: codificar la medición como un `int` (2 bytes en el Arduino UNO) y enviarlo byte por byte a través de la función `Serial.write(...)`. Debe tenerse en cuenta que el receptor quizás deba ser reprogramado para adaptarse a esta nueva codificación.

Con el envío de la información correspondiente a 3 electrodos la cantidad de bytes a mandar ascenderá a seis. Puede que el envío de seis bytes simultáneos pueda realizarse de manera más eficiente a la descrita en el anterior párrafo, de manera que se recomienda realizar una breve investigación al respecto.

7.5. Abaratamiento de costes de adquisición

Los costes de adquisición pueden ser reducidos si en vez de usarse múltiples sistemas independientes conectados entre sí se usa un sistema integrado en una sola unidad.

Un sistema integrado podrá componerse de una fase de adquisición de datos, con tres chips AD8232 como se ha indicado anteriormente en la subsección 7.3.2, por ejemplo. También habrá que añadir un microcontrolador programable como el ATmega328 que convierta los datos a formato digital y los transmita al módulo Bluetooth, éste último también podrá integrarse en la placa.

Pueden darse múltiples problemas con una integración como esta y se debería realizar un estudio preliminar de su viabilidad. Un problema claro existente en el diseño anterior es la cercanía física entre un módulo de radiofrecuencia y los circuitos de toma de datos. Como ya se indica en la subsección 3.2.1 en la página 22 el módulo Bluetooth probablemente introduciría ruido en las señales que se procuran leer.

Una vez diseñada la placa podrá enviarse a una fábrica para su producción en masa. Existen empresas dedicadas a crear placas bajo demanda a partir de diseños. Algunos ejemplos son *Fritzing Fab* [12] o *Advanced Circuits* [1].

Capítulo 8

Software, app

Las mejoras de la aplicación Android tratarán principalmente sobre la experiencia de usuario, ya que la app es el punto de contacto entre el usuario y el producto. Las posibilidades son múltiples y pueden orientarse a diferentes colectivos, desde los usuarios más profesionales a principiantes.

Existen también otras mejoras posibles, como la continuación de la expansión a seis derivaciones del capítulo anterior (sección 7.3, página 70), integración con variantes del sistema de adquisición de datos o aumento de la legibilidad y disminución de la complejidad del código mediante refactorización.

8.1. Ampliación a seis derivaciones

En el anterior capítulo se ha propuesto y estudiado la ampliación del hardware a seis derivaciones. Si el sistema electrónico toma y envía los datos necesarios para calcular seis derivaciones, la app deberá ser capaz de tratarlos y representarlos correctamente.

Suponiendo que se recibe la información bruta de los tres electrodos de información de un ECG de cuatro electrodos, entonces el primer paso a realizar es el cálculo de las derivaciones. Como se ha mostrado en las fórmulas 7.1 y 7.2 en el anterior capítulo, el cálculo de derivaciones no es más que la aplicación de una serie de fórmulas matemáticas a los datos. Incluir este mecanismo en la app es trivial, aunque quizás programarlo de manera que sea eficiente pueda complicar la implementación.

Tras hallar las derivaciones, éstas deberán ser visualizadas en la pantalla del dispositivo. Se proponen dos modalidades de hacer esto. Una es representar todas las derivaciones seguidas, una bajo la otra, y ofrecer al usuario la posibilidad de hacer *scrolling* para elegir cuales ver. La alternativa es dividir la pantalla actual de visualización de ECGs en dos mitades

horizontalmente. En una mitad se representan tres derivaciones, una debajo de otra, como en el prototipo actual. En la otra mitad se aplica la misma técnica con las tres derivaciones restantes.

La primera variante no permite ver las seis derivaciones al mismo tiempo y relacionar la información relativa a cada onda entre el total de todas las ondas, en cambio, permite ver un segmento considerable de tres derivaciones. En la segunda variante sucede al revés, pueden verse todas las derivaciones a la vez, pero sólo se puede ver un pequeño segmento de onda. Para evitar estos problemas podría hacerse uso de la segunda variante combinada con la propuesta de zoom descrita más adelante en la sección 8.2.2.

8.2. Interacción con el ECG

En este apartado se estudian las posibles mejoras o expansiones que se pueden realizar sobre la app para mejorar la interacción con el usuario. La posibilidad de pausar la toma del ECG para estudiar un segmento del mismo detenidamente, hacer zoom para observar en detalle partes de la onda y quizás una visualización gráfica de los resultados del diagnóstico pueden mejorar significativamente la experiencia de usuario.

8.2.1. Pausa del electrocardiograma

Actualmente el electrocardiograma es mostrado en tiempo real en el dispositivo Android. La aparición de una anomalía en el mismo no puede ser estudiada si su aparición no es regular, ya que la información presente en la pantalla desaparece al cabo de unos segundos para dar paso a los nuevos valores del ECG.

Añadir una función de pausa puede ser beneficioso para estudiar anomalías puntuales en el ECG. Su implementación no debería ser muy complicada, ya que únicamente hay que detener la entrada de datos al objeto que dibuja la onda en pantalla. Los datos pueden seguir siendo recogidos durante el período en el que el ECG esté pausado. Queda a discreción de quien implemente esta funcionalidad decidir si cuando el ECG debe ser resumido éste muestra los datos almacenados mientras el ECG estaba pausado, creando un desfase temporal variable, o si se vacía la pantalla y se muestra la información del ECG en tiempo real.

8.2.2. Modificación del tamaño del electrocardiograma

Otra funcionalidad interesante es la posibilidad de aumentar o reducir el tamaño del ECG, o bien, redimensionar el eje temporal. De esta manera se puede mostrar más información en pantalla que la perteneciente a dos o tres pulsos. Es posible realizar la implementación mediante unos botones que sean pulsados por el usuario o gestos, como el famoso *pinch to zoom*.

La complicación técnica asociada a incluir esta mejora no es excesiva, pero tampoco menospreciable. Deberán realizarse cambios en el bucle de renderizado del objeto que dibuja el electrocardiograma para que este comprima o expanda la cantidad de valores a representar en la pantalla. Actualmente se usa la función `drawPoints(...)` de la clase `Canvas` de Android para dibujar el ECG. Esta función o método toma como entrada un array de enteros, cada par de enteros representa un punto a dibujar. Este array deberá ser recreado cada vez que el usuario cambie el nivel de zoom, lo cual será computacionalmente caro.

En caso de que no existiera forma de implementar la funcionalidad de zoom de manera eficiente puede optarse por portar la representación del ECG a OpenGL. Mediante una combinación de configuración entre la cámara, viewport y matrices de escalado puede implementarse la ampliación del ECG de manera eficiente y rápida.

8.2.3. Visualización 3D del diagnóstico

De cara a usuarios aficionados sin profundos conocimientos o estudiantes del mundo de la biología o medicina, puede añadirse la funcionalidad de mostrar el diagnóstico realizado por el sistema experto en un diagrama tridimensional del corazón humano. El diagnóstico podrá representarse como una zona roja en la parte del corazón afectada por la patología cardiovascular que sufra el paciente. Otra posibilidad es crear una animación del funcionamiento de un corazón afectado por la patología diagnosticada.

Para implementar esta funcionalidad deberá hacerse uso de la librería OpenGL de manera directa o indirecta, estando ambas formas soportadas en Android. Dentro de la documentación oficial existen guías para dibujar gráficos mediante OpenGL a bajo nivel. En caso de no querer interactuar con OpenGL directamente, puede optarse por usar un motor 3D ya existente, como *LibGdx*.

Se proponen estos pasos para implementar esta mejora: primero, tomar un modelo tridimensional del corazón, en Internet hay múltiples fuentes con licencias permisivas. A partir de ahí, crear un objeto `View` que sea integrado en el `Fragment` que muestra actualmente el diagnóstico. Éste puede ser situado justo debajo del listado de patologías. Los conoci-

mientos básicos de OpenGL obtenidos a lo largo del Grado en Ingeniería Informática y la documentación oficial de Android son suficientes para crear una visualización del modelo básica con unas pocas luces y una cámara. No es necesario el uso de *shaders* ni otras técnicas avanzadas de gráficos.

8.3. Adaptación a estándares médicos

Aunque la aplicación en su estado actual muestra el ECG en tiempo real correctamente, no lo hace acorde a los estándares médicos. La adaptación a éstos es esencial para que el ECG pueda ser interpretado correctamente por expertos, pero conlleva importantes problemas de implementación.

Como se menciona en la subsección 2.2.2 en la página 8, los electrocardiógrafos profesionales presentan el ECG sobre una cuadrícula. Esta cuadrícula permite a los médicos interpretar el electrocardiograma en las dimensiones de voltaje y tiempo.

Añadir la cuadrícula requiere un estudio preliminar de como dibujarla en la pantalla del dispositivo Android, es decir que funciones de las APIs de Android llamar. Probablemente el estudio de la familia de funciones `drawLine(...)` de la clase `Canvas` de Android es un buen punto de partida. La función para dibujar las líneas deberá llamarse antes de que la que dibuja el ECG para generar una superposición de gráficos correcta.

Hay dos factores críticos a tener en cuenta en la implementación de esta característica: la densidad de pixels (DPI) de la pantalla del dispositivo en el que se ejecute la app y el nivel de zoom con el que se esté visualizando el ECG. La variedad en pantallas de dispositivos Android es muy amplia, pero en todas se deberá visualizar la cuadrícula de la misma manera. El estudio de la estructura `DisplayMetrics` de las APIs de Android es recomendable para crear una buena adaptación a diferentes pantallas.

8.4. Exportación del electrocardiograma

Hoy en día los electrocardiógrafos más avanzados son capaces de exportar los electrocardiogramas que toman. La creación y posterior integración de un sistema de exportación de ECGs para la aplicación desarrollada es una ampliación interesante. Las primeras modalidades de exportación pueden tomar la forma de un fichero CSV o XML en la memoria externa del dispositivo Android. La implementación es muy sencilla y ya estuvo presente en una iteración del prototipo.

Otros métodos de exportación más complejos resultarían muy útiles; por ejemplo, exportación a servicios de almacenamiento en la nube o impresión del ECG a través de una red WiFi. La primera técnica es posible a través de las APIs de los servicios de almacenamiento en la nube. Un buen sitio para comenzar a investigar como añadir esta funcionalidad es la documentación de la API de Google Drive en Android. La impresión en la nube es posible a través de las APIs de Android destinadas a este propósito.

Lógicamente exportar un ECG sin información adicional, como información sobre el sujeto es, probablemente, inútil. Por esta razón deberá añadirse un mecanismo para asociar alguna información adicional al electrocardiograma. Esto puede hacerse de manera sencilla mediante un formulario.

Ha de tenerse en cuenta que si el ECG está asociado a datos de carácter personal y se exporta a un servicio de almacenamiento en la nube o se imprime, deberá respetarse la *Ley Orgánica de Protección de Datos* para el tratamiento de ficheros. Los datos relativos a la salud de una persona, como es aquí el ECG, son categorizados como de *Nivel Alto* de manera que deberá prestarse especial atención al cumplimiento de dicha ley.

8.5. Integración con variantes del hardware

Actualmente la app Android sólo funciona con el hardware desarrollado en este proyecto. Ya que los diseños del prototipo son abiertos, puede que en el futuro aparezcan variantes del sistema de adquisición y transmisión de datos. La app cuenta con la información de los parámetros sobre los que opera el sistema electrónico, a través de ellos puede interpretar correctamente los datos que recibe. Esta información deberá ser modificada si se usa un hardware diferente al aquí presentado.

La adaptación a variantes del sistema electrónico puede consistir en cambios sencillos que el usuario pudiera realizar a través de menús. Unos ejemplos son: la alteración de los valores mínimo y máximo que representan el voltaje o la variación de la frecuencia de muestreo. Actualmente el código de la aplicación ya está parametrizado según estos valores, pero no existe manera de modificarlos sin reescribir una línea del código fuente y recompilar la aplicación.

Además pueden considerarse modalidades de adaptación más profundas y complejas, en caso de estar usando un sistema de adquisición de datos muy diferente al aquí prototipado. Una de estas podría ser la modificación de la especificación de la codificación de datos. Para abrir estas posibilidades de modificación debe crearse una API.

8.6. Refactorización del código

El desarrollo del proyecto ha tenido lugar a lo largo de varios meses y en todo momento la disponibilidad temporal para esta fase ha sido reducida. Por esta razón se ha invertido muy poco tiempo en mejorar la legibilidad y complejidad del código. Estos aspectos deberán ser trabajados en el futuro para asegurar un desarrollo rápido y sencillo de nuevas funcionalidades.

Unos conocimientos avanzados de patrones de programación y conocimientos de programación en Android, especialmente en flujos de trabajo multihilo, resultarán de gran ayuda al programador.

Capítulo 9

Software, extracción de características del ECG

El software de extracción de características debería recibir mejoras en lo relativo a la detección de la onda P en todas sus formas, ondas T negativas o bifásicas y onda U. La detección de complejos QRS y ondas T normales ya se realiza con buenos resultados. También resulta útil e interesante, aunque complejo, añadir la funcionalidad de detección de electrocardiogramas de personas no sanas de forma consistente y fiable.

9.1. Mejor detección de ondas P y T

Hay múltiples puntos de mejora en lo relativo a las ondas P y T. La onda P ofrece una gran cantidad de información sobre el estado del corazón y por ello su correcta detección resulta crítica. A continuación se indican cuales son las mejoras aplicables y el enfoque que se usaría para llevarlas a cabo de manera efectiva.

9.1.1. Comienzo y final de la onda P

Sería interesante conseguir una detección más fiable del comienzo y final de la onda P. Los esfuerzos realizados para hallarlos a través de algoritmos simples no han sido suficientes y la vectorización de la onda P no proporciona beneficios si el ECG contiene un nivel de ruido leve o superior.

Las acciones a seguir para paliar este problema serían la realización de una búsqueda y lectura de artículos científicos, el estudio de los diferentes métodos encontrados y por último la implementación del mejor en cuanto a simplicidad y calidad de los resultados.

9.1.2. Ondas P y T negativas y difásicas

La detección de ondas P y T negativas es otro punto de mejora del sistema de extracción actual. Como primera aproximación quizás podría adaptarse el sistema de detección de ondas P negativas a las ondas T, aunque es recomendable una reescritura del algoritmo, debido a que el sistema no es suficientemente sólido.

Al igual que la metodología propuesta en la subsección anterior, en este caso también debería realizarse un estudio detallado de artículos académicos y del estado del arte para recoger la mayor cantidad de información posible sobre algoritmos de detección de ondas P y T anormales. A partir de entonces se podría escribir una buena implementación para resolver el problema.

En cuanto a la posible aparición de ondas bifásicas: este problema resulta más complicado que un estudio de la positividad o negatividad de la onda.

Representar la onda mediante aproximación de funciones podría dar buenos resultados, ya que la positividad, negatividad y el hecho de que la onda sea o no bifásica pueden ser calculadas mediante herramientas matemáticas, principalmente las derivadas de una función. Mediante la segunda derivada se puede saber si la onda es cóncava (P o T positiva) o convexa (P o T negativa); y mediante la tercera derivada si hay un punto de inflexión, es decir, si la onda es bifásica.

9.2. Análisis del ECG a alto nivel

Como se ha mencionado en la sección 5.1 en la página 44, el sistema de extracción de características creado sólo estudia características individuales del ECG y no el ECG como un conjunto de pulsos. Un análisis global del ECG abre las puertas a la posibilidad de diagnosticar una mayor cantidad de patologías cardiovasculares. Un ejemplo de patología cardiovascular detectable únicamente si se visualiza un electrocardiograma completo es la fibrilación auricular.

Una forma de afrontar esta mejora es crear un modelo mediante técnicas de aprendizaje automático. Dicho modelo podrá ser entrenado a partir de ECGs completos, etiquetados con una patología detectable según un análisis global del ECG. La principal desventaja de este método es la necesidad de igualar los parámetros del ECG tomado del sistema de adquisición a los de los ECGs de entrenamiento. Por otra parte, una gran ventaja de este enfoque es que no resulta necesario extraer características del ECG previamente.

9.3. Extracción de características de ECGs de personas que sufren patologías cardiovasculares

Los algoritmos desarrollados en este proyecto parten de la premisa de que el ECG del cual se deben extraer las características será similar a un ECG de una persona sana en cuanto a la morfología general de un pulso: un complejo QRS, precedido de una onda P y seguido de una onda T. En caso de que una alteración cardíaca elimine algún pico, altere el rango de valores habitual de una onda o modifique de otra manera el ECG, éste no podrá ser tratado por los algoritmos actuales.

Algunos ejemplos de pulsos que escapan de lo habitual pueden ser los recogidos en un ECG de una persona que sufra infarto de miocardio, representado como una elevación ST. Otro caso podría ser el de una persona que sufra un bloqueo de conducción de tipo Mobitz I, manifestado como ondas P que no preceden a un complejo QRS.

Para poder tratar todo tipo de electrocardiogramas, incluidos los de personas que sufran grandes modificaciones en el mismo debido a patologías cardiovasculares, los algoritmos deberán ser reescritos. Se recomienda buscar e investigar los artículos académicos referentes a esta temática.

9.4. Ampliación a seis derivaciones

La existencia de seis derivaciones podría dotar al sistema de extracción de características de una mayor cantidad de información con la que trabajar. Además podrían combinarse las diferentes formas de una misma característica en todas las derivaciones disponibles para aumentar la solidez de los algoritmos.

9.4.1. Nueva información accesible

Contar con nuevas derivaciones es similar a obtener más *puntos de vista* sobre el corazón. Lógicamente, esto deriva en una mayor cantidad de información accesible sobre el estado de este importante órgano. Para comenzar, la disponibilidad de varias derivaciones permite hallar el eje eléctrico del corazón o eje cardíaco.

Debido a lo sencillo que es hallar el eje cardíaco, éste suele ser usado por los expertos como comprobación preliminar para saber si existe alguna patología cardiovascular en el paciente sin tener que realizar un análisis exhaustivo del ECG. En caso de sospechar que el paciente sufre alguna alteración cardíaca los expertos recurren a otros métodos posterior-

mente para confirmar sus sospechas. El eje cardíaco puede verse afectado, no sólo por el estado del corazón, sino también por el peso del paciente.

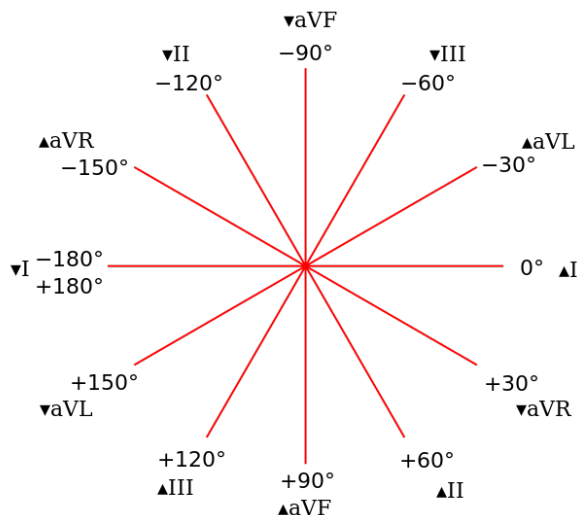


Fig. 9.1 Representación del sistema hexaxial

Fuente: commons.wikimedia.org

Para hallar el eje cardíaco en un sistema automatizado deben tomarse dos derivaciones a 90 grados entre sí según el sistema hexaxial, por ejemplo I y aVF. Tras elegir las dos derivaciones se aplica el producto vectorial sobre los vectores de potencial presentes en cada derivación. La dirección del vector resultante es equivalente a la del eje cardíaco. Los valores del ángulo del vector son contrastados con una serie de intervalos establecidos y a partir de dicha relación se deduce si el paciente puede estar sufriendo alguna patología.

Otros aspectos del corazón que no pueden ser estudiados con una sola derivación son las anomalías causadas por una alteración cardiovascular que sólo son visibles en derivaciones diferentes a I. Por ejemplo: las consecuencias de un infarto en la zona inferior del corazón se observan en los picos Q de las derivaciones II, III y aVF. El embolismo pulmonar únicamente se puede detectar mediante la combinación de la información que ofrecen las ondas en las derivaciones I y III.

9.4.2. Mejora de resultados por combinación de derivaciones

Las múltiples características relevantes de un ECG toman diferentes formas en cada una de las derivaciones. Para cada característica que se desee extraer de forma fiable puede estudiarse en qué derivación se detecta mejor y hallar la característica correspondiente en ella. Si la información que provee una sola derivación no es suficiente, entonces puede

estudiarse la misma característica en diferentes derivaciones simultáneamente y combinar los resultados obtenidos.

Para decidir en qué derivación extraer cada característica o qué derivaciones combinar para hallarla de forma fiable debe consultarse a un experto con experiencia en electrocardiografía. La información correspondiente al *conocimiento estratégico* deberá ser entonces convertida en algoritmo.

9.5. Creación de un plan de calidad para evaluar el sistema

Para poder asegurar que el sistema es sólido y fiable es necesario un plan de calidad. El plan de calidad deberá ser sencillo y rápido de ejecutar o en el mejor caso será totalmente automatizado. Esto permite modificar el sistema de extracción de características continuamente, comprobando cuales son los resultados, incluso de los más pequeños cambios. Además los cambios podrán ser aplicados sin miedo a dañar el sistema, ya que cualquier error será descubierto al momento.

En esta sección no se especifica cómo deberá ser el plan de calidad, pero se dan algunas recomendaciones que pueden simplificar el proceso o ayudar a determinar los pasos que lo compongan.

9.5.1. Etiquetado de ECGs

Los ECGs usados para evaluar el sistema deberán ser etiquetados manualmente por un experto. Las etiquetas deberán representar cada unidad de información que el sistema pueda extraer, ya sean las patologías que se manifiestan en el ECG, los puntos relevantes de la onda o las características globales de la onda.

Algunos ECGs de la base de datos Physiobank ya están parcial o totalmente etiquetados. Partir de electrocardiogramas ya etiquetados acelerará el proceso. Probablemente una de las mayores dificultades inherentes a tomar ECGs de múltiples bases de datos externas sea la normalización de sus etiquetas, y de los rangos y unidades de los valores que contengan.

9.5.2. Lanzamiento del sistema de extracción de características

Como se ha mencionado en la introducción de esta sección, en el caso óptimo, el sistema de aseguramiento de la calidad estaría automatizado. Esto conlleva el lanzamiento por software del sistema de extracción de características, el cual tomaría los ECGs ya etiquetados y realizaría su propio proceso de etiquetación.

Para lanzar el proceso automáticamente, recomiendo el aislamiento del sistema de extracción a una unidad autocontenida. Esto es posible en el estado actual del sistema debido a la estrategia que se ha aplicado para programarlo. Sobre dicha unidad autocontenida se pueden lanzar tests automatizados mediante *JUnit*, un sistema ya estudiado a lo largo del Grado en Ingeniería Informática.

Los resultados de las ejecuciones podrán ser almacenados en ficheros para su posterior procesamiento.

9.5.3. Formas de representar los resultados

Los resultados de la ejecución de las pruebas de calidad podrían ser representados mediante matrices de confusión. Este método tiene claras ventajas, como una visualización rápida del rendimiento global del sistema. Además, si se aplica la matriz de confusión a cada característica individual que detecta el sistema de extracción de características, puede obtenerse información muy granular.

La matriz de confusión podría tener la siguiente forma:

Tabla 9.1 Ejemplo de matriz de confusión para la evaluación del sistema de extracción de características

	<i>Picos R</i>	<i>Picos Q</i>	<i>Picos T</i>	<i>No identificado</i>	...
Picos R	53	0	2	4	...
Picos S	0	3	0	12	...
Picos T	1	0	60	1	...
Comienzo QRS	0	0	0	23	...
Final T	0	2	0	4	...
...

Capítulo 10

Sistema experto

El funcionamiento del sistema experto actual es relativamente simple, de manera que resulta difícil definir mejoras. El sistema experto únicamente puede ser expandido a través de su conversión a un sistema de lógica difusa o contrastando sus resultados con otro sistema capaz de realizar un diagnóstico basado en electrocardiogramas. Este sistema de diagnóstico complementario podría estar basado en técnicas de aprendizaje automático.

10.1. Un sistema basado en lógica difusa

Los expertos médicos no realizan los diagnósticos a partir de electrocardiogramas de manera binaria, ya que siempre existe incertidumbre en los datos o su interpretación. Para conseguir un comportamiento similar al de la realidad el sistema experto creado debería utilizar lógica difusa a la hora de diagnosticar al paciente.

A modo de ejemplo: en un paciente, el eje cardíaco se sitúa en torno a los 30 grados, en el límite de la zona en la que se considera que existe una desviación izquierda del eje. Esto causa incertidumbre en la interpretación que realiza el experto. Éste podrá dictaminar que existe una posibilidad del 40 % de que el paciente sufra una hipertrofia ventricular izquierda.

La incertidumbre existente en la interpretación de los datos evita que el experto pueda dar un diagnóstico certero.

10.1.1. Motores de lógica difusa existentes

Existe una variante de CLIPS que añade la capacidad de razonamiento difuso a dicho sistema. Esta versión es conocida como *FuzzyCLIPS*, actualmente el proyecto se puede encontrar en github [20]. *FuzzyCLIPS* apenas cuenta con información que lo soporte, aparte del

manual de usuario creado junto al sistema. Actualmente no parece existir ningún esfuerzo público de adaptar FuzzyCLIPS a la plataforma Android. La adaptación puede considerarse complicada.

Si se desea considerar una opción similar a FuzzyCLIPS, pero mejor documentada y moderna, entonces se recomienda investigar en Internet que sistemas de lógica difusa existen actualmente que a la vez sean compatibles con Android. Recientemente *jFuzzyLite*, la adaptación Java de *FuzzyLite* ascendió a la versión 1.0. Parece interesante investigar más sobre este sistema.

10.1.2. Proceso de adaptación del conocimiento al nuevo sistema

El uso de un motor de lógica difusa obliga a modificar algunos elementos del sistema actual. Por una parte deberán reescribirse las reglas existentes para asociar a cada una los factores de certeza correctos. Será necesario realizar entrevistas con cardiólogos expertos para definir lo mejor posible dichos valores.

Por otra parte el sistema de extracción de características deberá ser capaz de asociar a cada característica un nivel de certeza. Esto podría hacerse aprovechando los resultados de las pruebas de calidad mencionados en la sección 9.5.3 en la página 86.

Incluir incertidumbre o factores de certeza a las características extraídas ofrece una ventaja clave al sistema de extracción: los resultados obtenidos no tendrán porque ser extremadamente buenos. Queda a discreción de quién o quiénes implementen esta funcionalidad decidir si el sistema de extracción de características merece un esfuerzo de desarrollo significativo para ser mejorado o no.

10.2. Combinación con otros sistemas de diagnóstico

En la concepción original de este proyecto se consideraba únicamente el uso de técnicas de minería de datos para realizar el diagnóstico del ECG. Tras descubrir los algoritmos usados por los expertos se optó por usar un sistema basado en reglas (véase la subsección 2.3.2 en la página 13), pero eso no significa que la minería deba de ser olvidada.

El uso de técnicas de *machine learning* ha sido aplicado anteriormente al tratamiento de ECGs [9] [28], de manera que ya ha sido probado que aplicar esta estrategia es posible. Los sistemas de clasificación automática suelen clasificar pulsos, extraídos manual o automáticamente del ECG, entre un pequeño conjunto de clases (patologías cardíacas).

En el estado actual del proyecto, la aplicación de técnicas de aprendizaje automático

puede tomar dos enfoques. Enfoque 1: tanto el sistema experto como el sistema de aprendizaje automático diagnostican las mismas patologías y sus resultados son contrastados. Enfoque 2: cada uno de los sistemas diagnostica aquellas alteraciones cardíacas que mejor detecte y los resultados son combinados para ofrecer un diagnóstico completo.

En el primer caso deberá estudiarse cómo realizar el contraste de resultados. Esto puede realizarse mediante la asignación de pesos a cada veredicto, la elección de aquel con mayor fiabilidad, etcétera. En el segundo caso, el problema reside en conocer qué alteraciones son detectadas de manera más fiable por cada sistema y configurar un programa global que realice el reparto del trabajo. Lógicamente, se asignará a cada sistema la detección de aquellas patologías que mejor detecte.

Independientemente del enfoque utilizado, al usar técnicas de machine learning, se precisa la creación y evaluación de un modelo. Éste puede ser entrenado a través de los ECGs disponibles en multitud de bases de datos online. Por ejemplo, en la ya mencionada y utilizada en este proyecto, Physiobank.

Parte III

Otros aspectos del proyecto

Capítulo 11

Conclusiones

A lo largo de esta sección se presentan las conclusiones formadas tras el desarrollo del proyecto. Además también se incluye información sobre el manual de usuario creado para el prototipo y se presenta cuál es el coste de adquisición del prototipo generado.

11.1. Conclusiones generales del proyecto

Durante el proyecto aquí presentado se ha creado un electrocardiógrafo sencillo de coste reducido. El desarrollo del mismo ha sido dividido en cuatro fases, las cuales se han desarrollado en paralelo.

En la fase de **electrónica** se ha trabajado con conocimientos y sistemas no adquiridos durante el Grado de Ingeniería Informática, haciendo dura su realización. Los puntos fuertes en cuanto a formación han sido comprendidos por la electrónica analógica y el procesamiento digital de señales para realizar el filtrado de los electrocardiogramas.

Los desarrollos en el sistema electrónico han partido de sistemas ya existentes y se han basado en su modificación para adaptar o mejorar su rendimiento de cara a los objetivos del proyecto. Mediante este método de trabajo se ha evitado invertir tiempo en crear un sistema desde cero, de manera análoga al uso de librerías en el desarrollo de software. En el prototipo final todas las partes del sistema electrónico son *Open Hardware*, aplicando así exitosamente los principios de una filosofía abierta en el campo de este proyecto que más complicaciones podría presentar al respecto.

La segunda fase está compuesta por el desarrollo de la **aplicación Android**. Los conocimientos previos de este entorno han ayudado en la programación del esqueleto de la aplicación y los aspectos más básicos, pero se han desarrollado componentes más complejos para dotar a la aplicación de otras funcionalidades necesarias para este proyecto. Algunos

de estos componentes complejos son la comunicación Bluetooth, la visualización del ECG en tiempo real o la interfaz modular.

La **extracción de características** ha sido la parte más compleja y dura de este proyecto. Se ha alcanzado un sistema que detecta de forma consistente los complejos QRS en la mayoría de los electrocardiogramas y las ondas T en ECGs de buena calidad. La detección de ondas P resulta poco fiable. Esto ha resultado decepcionante, ya que debido a la importancia de esta onda se ha invertido una gran cantidad de tiempo en crear un sistema para su detección. Además de detectar puntos significativos sobre el ECG, este sistema también es capaz de calcular algunas características adicionales a partir de los datos extraídos.

Todos los algoritmos implementados han sido creados desde la base, ya que la literatura existente no ha resultado útil de cara a implementar cada parte del sistema de extracción de características. Para reducir las consecuencias negativas de una extracción de características deficiente también se implementó, aunque no se incorporó al sistema final, una app para seleccionar manualmente algunas características de la onda.

La fase final del proyecto ha estado conformada por un **sistema experto**, capaz de realizar un diagnóstico sobre el electrocardiograma. El sistema experto tiene como entrada los datos extraídos del ECG por el sistema anteriormente descrito. Como salida ofrece una lista de patologías cardiovasculares detectadas junto a las razones que han llevado al motor de inferencia a realizar el diagnóstico presentado.

El sistema experto ha sido incorporado en la aplicación Android a través de una adaptación del motor CLIPS. Su estructura ha sido dividida de manera modular a través de las herramientas que CLIPS ofrece para ello y a través de ficheros. La principal ventaja de este enfoque es la posibilidad de actualizar el sistema sin tener que modificar éste en su totalidad. Además se han utilizado múltiples paradigmas de programación para sacar el mayor provecho al lenguaje CLIPS.

Aunque el prototipo aquí presentado ofrece una funcionalidad similar a la descrita en el alcance original, no deja de ser un prototipo. Esto significa que aún queda recorrido de cara a crear un producto final. Para facilitar dicho recorrido se ha presentado un amplio conjunto de mejoras y oportunidades de expansión del prototipo. La licencia permisiva (más información en la sección 11.4 en la página 97) asociada al trabajo realizado permite un desarrollo futuro del proyecto a otros estudiantes, comunidades o empresas.

11.2. Manual de usuario

Con el propósito de ofrecer un producto lo más cercano a su versión final se ha redactado un breve manual de usuario para el prototipo desarrollado. Dicho manual indica cómo realizar todos los pasos necesarios para hacer funcionar el prototipo, desde el ensamblado de los componentes electrónicos, hasta la utilización de la aplicación Android.

El manual puede consultarse en el Anexo B.

11.3. Coste del prototipo final

Como se ha indicado en múltiples partes de esta memoria, uno de los objetivos principales del proyecto es que el sistema generado tenga un coste de adquisición reducido. En la Tabla 11.1 puede observarse cuál es el desembolso económico que conlleva a día de hoy (12 de junio de 2014) adquirir los componentes primarios que forman el prototipo.

Por componentes primarios se entienden aquellos necesarios exclusivamente para crear este prototipo. Debe tenerse en cuenta que la placa Arduino y el módulo Bluetooth, pueden ser reutilizados en otros proyectos o sistemas.

Tabla 11.1 Coste de los componentes primarios del prototipo en distribuidores varios

<i>Componente</i>	<i>Procedencia</i>	<i>Precio (euros)</i>
Olimex ECG Shield	Página web de Olimex	19,95
Cables de conexión para Olimex ECG Shield	Página web de Olimex	9,95
Arduino UNO (incluye cable USB e IVA)	Amazon.es	16,99
Módulo Bluetooth (incluye IVA)	Amazon.es	8,45
IVA	Página web de Olimex	6,97
Costes de envío	Página web de Olimex	4,96
Costes de envío	Amazon.es	2,50
Coste total		69,77

Debido a la categoría del transportista elegido en la página web de Olimex, éste podría tardar hasta 3 semanas en transportar el pedido.

Los costes de la placa Arduino y el módulo Bluetooth podrían ser fuertemente reducidos si se adquieren versiones modificadas de éstos en el continente asiático. Recuérdese que los diagramas de la placa Arduino son públicos y por tanto cualquier fabricante puede modificar los diseños para crear placas similares a menor precio.

Aparte de los componentes primarios también son necesarios otros componentes reutilizables o desechables para mejorar el prototipo y llevarlo al estado que se presenta en esta memoria. Estos componentes pueden encontrarse en entornos de desarrollo electrónico y múltiples páginas web. A continuación se lista cuáles son estos componentes:

- Cable USB A-B: permite la conexión de la placa Arduino a un ordenador para modificar su programa
- Estación de soldadura, estaño y pinzas de crocodilo: necesarios para la mejora de los electrodos para la placa Olimex descrita en la subsección 3.2.3 en la página 29
- Electrodo desechables: a través de ellos y la mejora anterior se consiguen mejores lecturas de la placa Olimex
- Baterías AA: alimentan el sistema electrónico
- Soporte (*holder*) de baterías AA: hace más cómoda la conexión de las baterías

11.4. Licencia

El código de este proyecto será liberado bajo la licencia *GNU Lesser General Public License Version 3*. Entre otras características, esta licencia permite el uso del código en sistemas propietarios y Open Source siempre y cuando los componentes de este proyecto que se usen sean liberados con las modificaciones realizadas. Este es un enfoque aceptable para permitir el uso del código generado a cualquier empresa o particular, siempre y cuando este proyecto y sus derivados queden expuestos al público.

El código interno de CLIPS se encuentra en dominio público. El código de las APIs Android utilizadas se encuentra protegido bajo los términos y condiciones impuestas por Google para el Android SDK. El código Arduino utilizado a través de su API se encuentra protegido bajo la licencia GNU GPL v2.

El código original de esta plantilla \LaTeX está sujeto a la licencia MIT y fue desarrollado por Krishna Kumar, Universidad de Cambridge.

11.5. Código del proyecto

El código correspondiente a este proyecto puede encontrarse en <https://github.com/Inversebit/LCIECG>. Éste es un repositorio público alojado en Github. En el repositorio puede encontrarse un fichero `README.md` que indica cómo está estructurado el repositorio y cómo trabajar con el código.

Capítulo 12

Gestión del proyecto

Este proyecto se ha desarrollado mediante una filosofía de desarrollo iterativo, a excepción del sistema electrónico. Se ha trabajado en todo momento en mantener un prototipo funcional de cada una de las fases a desarrollar, ampliando éstas mediante pequeñas adiciones hasta alcanzar los puntos indicados en el alcance del proyecto. Este comportamiento ha sido posible gracias a un continuo ciclo de seguimiento y control seguido de replanificación.

12.1. Gestión de tareas

Al comienzo del desarrollo del proyecto se definió el alcance como la lista de puntos recogida en la sección 1 de esta memoria, en la página 2. Para llevar a cabo los objetivos previstos, el proyecto fue dividido en una serie de subtareas representadas en el *Esquema de descomposición de trabajo*, Figura 12.1 en la página siguiente. Aquellas tareas que se prestaron a ser realizadas en paralelo fueron llevadas a cabo de esta manera.

A lo largo del proyecto se crearon dos grandes tareas no planificadas. La primera de ellas consistió en el estudio e implementación de técnicas de filtrado, correspondiente a la categoría *Electrónica*. La segunda tarea no planificada fue la extracción de características del ECG. Ésta obtuvo su propia categoría dada su envergadura, y se dividió en dos subtareas: investigación sobre el estado del arte e implementación.

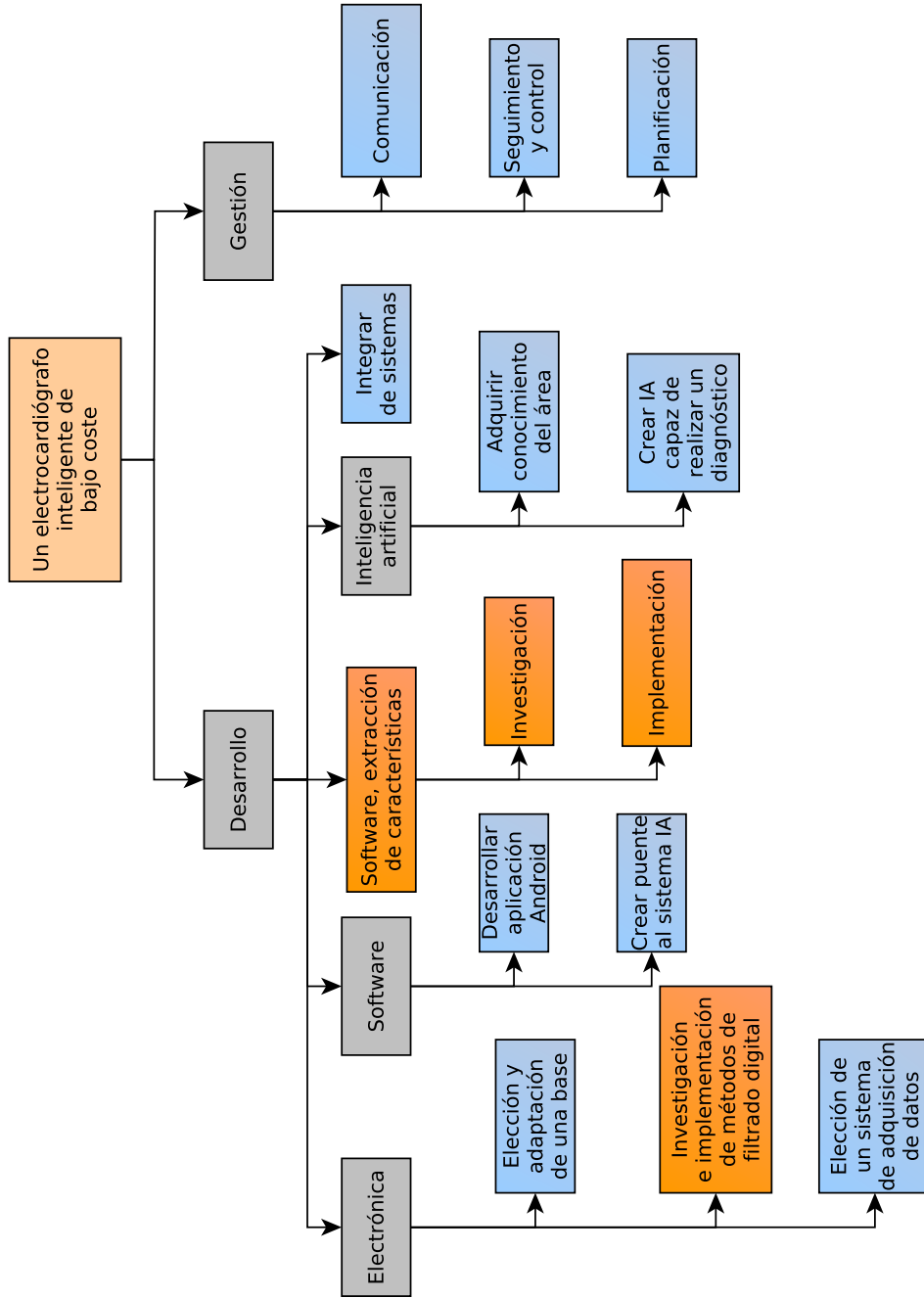


Fig. 12.1 Esquema de descomposición de trabajo. Las categorías están representadas en gris. Las tareas en azul. En naranja se muestran las tareas añadidas en un momento posterior a la planificación inicial.

12.2. Gestión del alcance

A lo largo del desarrollo del proyecto se dieron una serie de problemas con la gestión del mismo. A su vez surgieron nuevas tareas no previstas, como ya se ha explicado en la sección anterior. Estas tareas resultaron ser mucho más duras de lo previsto. La combinación de estos factores causó la decisión de reducir el alcance original del proyecto presentado en la sección 1 al alcance que aquí se presenta.

El desarrollo del proyecto ha culminado con la creación de un prototipo funcional. Los objetivos alcanzados a través de este prototipo son:

- Un sistema de toma de bioseñales cardíacas que cumple los siguientes requisitos:
 - Conversión de datos a formato digital
 - Codificación y envío de datos inalámbrico en tiempo real
- Una app para dispositivos Android que cumple los siguientes requisitos:
 - Recepción de datos de forma inalámbrica y decodificación de los mismos en tiempo real
 - Filtrado de los datos en tiempo real
 - Visualización de los datos en tiempo real
 - Realización de un diagnóstico orientativo a partir de los datos recibidos
 - Compatibilidad con la tablet Nexus 7, modelo 2012
- Un sistema de diagnóstico automatizado que cumple los siguientes requisitos:
 - Integrado en una app para dispositivos Android
 - Capacidad de realizar un diagnóstico en menos de un minuto

Respecto a las especificaciones de alcance iniciales, las diferencias son:

- Coste objetivo de adquisición del producto no alcanzado
- Toma de datos de tres en vez de cuatro electrodos, reduciendo así la lectura de datos a únicamente una de las tres derivaciones principales: I, II o III. En el prototipo final se usa la derivación I. Las consecuencias de esta reducción se extienden a lo largo de todas las fases del proyecto. Puede verse cuales han sido las ventajas perdidas a través de la lectura de las secciones “Ampliación a seis derivaciones” existentes en los Capítulos 7, 8 y 9

- El diagnóstico no tiene una precisión tan alta como la esperada. Las causas de este suceso se pueden atribuir a la calidad de los algoritmos desarrollados y no contar con más que con una derivación
- No se han aplicado técnicas de machine learning para realizar el diagnóstico sobre el corazón
- La app cuenta con la capacidad de filtrar la señal en tiempo real para su correcta visualización por el usuario y correcta interpretación por el sistema de extracción de características

12.3. Gestión del tiempo

El proyecto dio comienzo el día 20 de Enero de 2014 y finalizó el 13 de Junio de 2014. A continuación se recogen los hitos más relevantes acontecidos a lo largo del desarrollo del proyecto:

17 de marzo Lanzamiento de la primera iteración del prototipo

26 de marzo Elección de la placa Olimex como sistema de adquisición de datos definitivo para el prototipo

8 de mayo Fin del desarrollo del sistema de extracción de características

12 de mayo Finalización de la última iteración del prototipo

13 de junio Entrega de la documentación del proyecto

12.4. Gestión de costes

En la Tabla 12.1 se representa en horas el coste de dedicación que este proyecto ha requerido para llevarse a cabo. Además del coste humano, se han dado una serie de costes económicos. Estos costes han sido generados mayoritariamente por la fase de electrónica. Consisten en la compra de las placas de adquisición de datos AD8232Eval y Olimex ECG Shield, además de los costes de diseño y fabricación de la placa basada en el chip AD8232.

Existen otros costes atribuidos a elementos desechables, como los electrodos, y componentes eléctricos, por ejemplo, cables. Aun así su valor es despreciable frente a los otros costes generados por el desarrollo del proyecto.

Tabla 12.1 Coste humano del proyecto

<i>Categoría</i>	<i>Tarea</i>	<i>Dedicación (h)</i>
Electrónica	Desarrollo del sistema base	12
	Trabajo sobre placas de adquisición de datos y toma de electrocardiogramas	23
	Desarrollo de métodos de filtrado	22
Software, app	Desarrollo de la app	19
	Integración de sistemas en la app	13
Software, extracción de características	Investigación	14
	Implementación y pruebas	35
Inteligencia artificial	Investigación	4
	Implementación	13
	Reuniones con el experto	6
Otros	Formación sobre el corazón humano y electrocardiografía	5
	Aprendizaje de L ^A T _E X y adaptación de plantilla	12
	Redacción de la memoria	100
	Redacción del manual de usuario	6
	Preparación de la defensa (Estimación)	20

Bibliografía y referencias

- [1] Advanced Circuits, . 4pcb homepage. <http://www.4pcb.com/>. Último acceso: 01/06/2014.
- [2] Analog Devices, . *AD8232. Single-Lead, Heart Rate Monitor Front End*, 2012. Rev. A.
- [3] Arduino, . Arduino UNO. <http://arduino.cc/en/Main/ArduinoBoardUno>. Último acceso: 28/05/2014.
- [4] Bazett, H.C. An analysis of the time-relations of electrocardiograms. *Heart*, 3:353–370, 1920.
- [5] Bluetooth SIG, . *RFCOMM WITH TS 07.10*, 2011. Rev. V12.
- [6] Censi, F., Calcagnini, G., Triventi, M., Mattei, E., Bartolini, P., Corazza, I., and Boriani, G. Effect of high-pass filtering on ecg signal on the analysis of patients prone to atrial fibrillation. *Ann Ist Super Sanita*, 45(4):427–431, 2009.
- [7] CLIPS project, . Clips homepage. <http://clipsrules.sourceforge.net/>, . Último acceso: 28/05/2014.
- [8] CLIPS project, . Clipsjni version 0.3 beta. <http://clipsrules.sourceforge.net/CLIPSJNIBeta.html>, . Último acceso: 28/05/2014.
- [9] De Chazal, P., O’Dwyer, M., and Reilly, R.B. Automatic classification of heartbeats using ecg morphology and heartbeat interval features. *Biomedical Engineering, IEEE Transactions on*, 51(7):1196–1206, 2004.
- [10] Dubin, D.B. *Dubin: Interpretacion de ECG: Metodo Clasico del Dr. Dubin para Entender los Mensajes Electricos del Corazon*. Cover Publishing Company, illustrated edition, 2007.
- [11] Fisher, T. Butterworth / bessell / chebyshev filters. <http://www-users.cs.york.ac.uk/~fisher/mkfilter/trad.html>. Último acceso: 26/05/2014.
- [12] Fritzing, . Fritzing fab homepage. <http://fab.fritzing.org/fritzing-fab>. Último acceso: 01/06/2014.

- [13] Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. Ch., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000 (June 13). Circulation Electronic Pages: <http://circ.ahajournals.org/cgi/content/full/101/23/e215> PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.
- [14] Gómez-Goiri, A. Clips4android. <https://github.com/gomezgoiri/CLIPS4Android>. Último acceso: 20/05/2014.
- [15] Karpagachelvi, S., Arthanari, M., and Sivakumar, M. Ecg feature extraction techniques - a survey approach. *International Journal of Computer Science and Information Security*, 8(1), 2010.
- [16] Lugovaya, T.S. Biometric human identification based on electrocardiogram. Master's thesis, Faculty of Computing Technologies and Informatics, Electrotechnical University "LETI", 2005.
- [17] Mariel, A. Android_bt_connection. https://github.com/Inversebit/useful-templates/tree/master/Android_BT_Connection, 2014. Último acceso: 21/05/2014.
- [18] McSharry, P.E., Clifford, G.D., Tarassenko, L., and Smith, L.A. A dynamical model for generating synthetic electrocardiogram signals. *Biomedical Engineering, IEEE Transactions on*, 50(3):289–294, 2003.
- [19] Moody, G.B. and Mark, Roger G. The impact of the mit-bih arrhythmia database. *Engineering in Medicine and Biology Magazine, IEEE*, 20(3):45–50, 2001.
- [20] National Research Council of Canada, . Fuzzyclips project on github. <https://github.com/rochard/FuzzyCLIPS>. Último acceso: 03/06/2014.
- [21] Olimex LTD, . Manual de olimex ECG shield. <https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/resources/SHIELD-EKG-EMG.pdf>. Último acceso: 29/05/2014.
- [22] Regidor, E. and Gutiérrez-Fisac, J.L. Patrones de mortalidad en españa, 2010. *Ministerio de Sanidad, Servicios Sociales e Igualdad*, 2013.
- [23] Sandia National Laboratories, . Jess homepage. <http://http://herzberg.ca.sandia.gov/>. Último acceso: 01/06/2014.
- [24] Taddei, A., Distanto, G., Emdin, M., Pisani, P., Moody, G.B., Zeelenberg, C., and Marchesi, C. The european st-t database: standard for evaluating systems for the analysis of st-t changes in ambulatory electrocardiography. *European heart journal*, 13(9): 1164–1172, 1992.
- [25] Tuzcu, V. and Nas, S. Dynamic time warping as a novel tool in pattern recognition of ecg changes in heart rhythm disturbances. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 182–186. IEEE, 2005.

-
- [26] Wikipedia, . High-pass filter — wikipedia, the free encyclopedia, 2014. URL http://en.wikipedia.org/w/index.php?title=High-pass_filter&oldid=608146718. [Online; accessed 25-May-2014].
- [27] Wikipedia, . Low-pass filter — wikipedia, the free encyclopedia, 2014. URL http://en.wikipedia.org/w/index.php?title=Low-pass_filter&oldid=609901792. [Online; accessed 26-May-2014].
- [28] Zhang, Z., Dong, J., Luo, X., Choi, K., and Wu, X. Heartbeat classification using disease-specific feature selection. *Computers in Biology and Medicine*, 2013.

Anexo A

Diagramas eléctricos de las placas de adquisición de datos

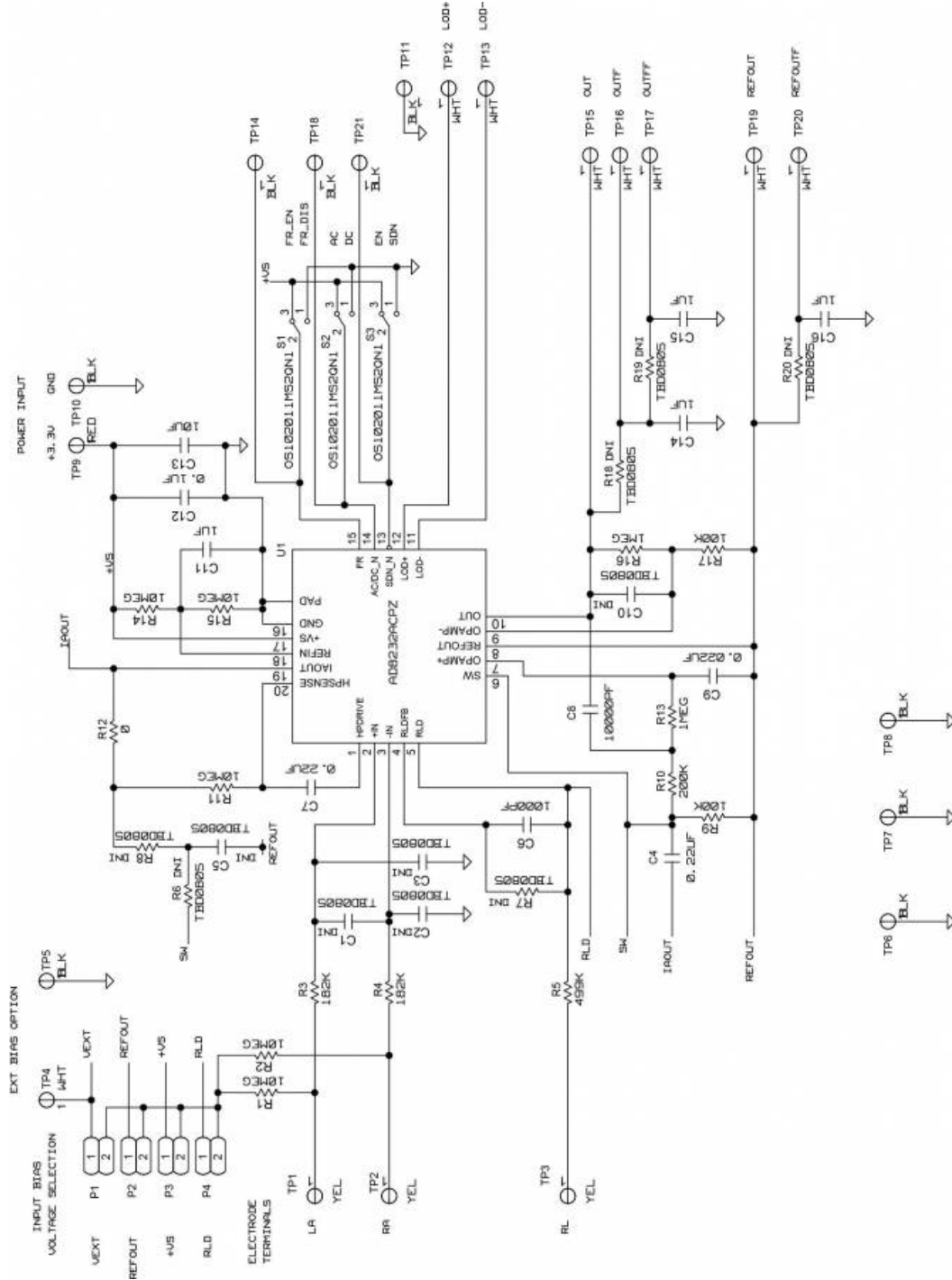


Fig. A.2 Diagrama de la placa de evaluación del chip AD8232 creada por Analog Devices

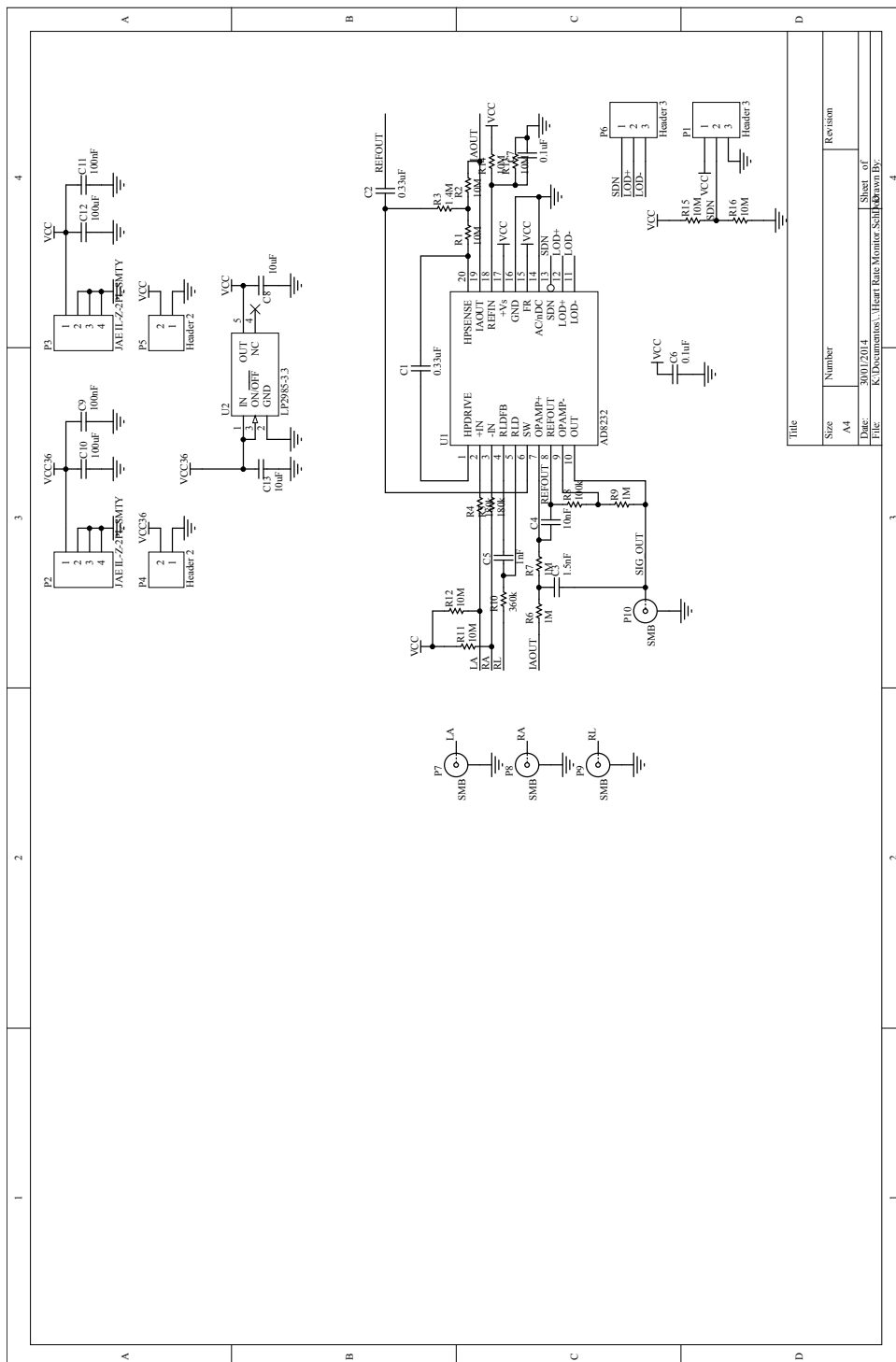


Fig. A.3 Diagrama de la placa de diseño propio basado en el chip AD8232

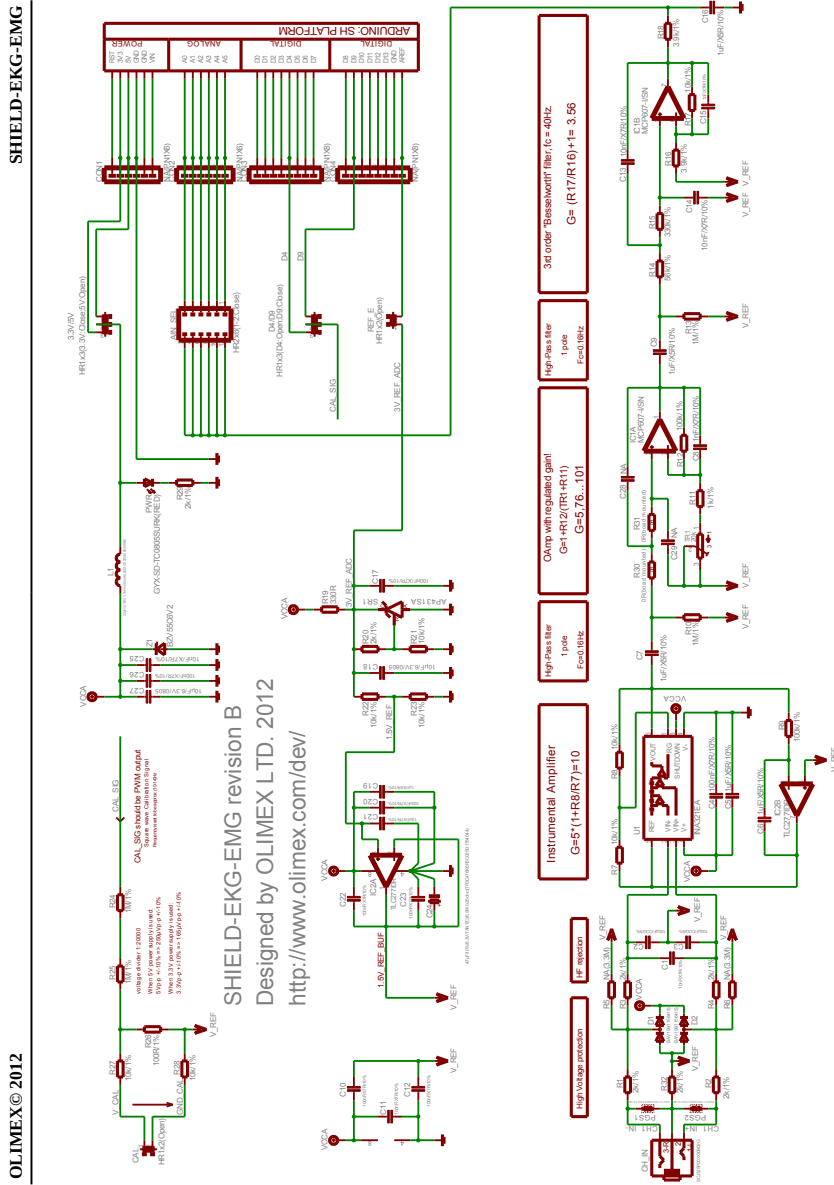


Fig. A.4 Diagrama de la placa Olimex ECG Shield. En la mitad inferior del diagrama se encuentran las etapas de amplificación y filtrado claramente etiquetadas.

Anexo B

Manual de usuario

Manual de usuario del prototipo del proyecto “Un electrocardiógrafo inteligente de bajo coste”

Alexander Mariel

Revisión E

11 de junio de 2014

Resumen

En este documento se describen los pasos a seguir para realizar un uso correcto del prototipo generado en el Proyecto de Fin de Grado “Un electrocardiógrafo inteligente de bajo coste” por Alexander Mariel. Cada sección de este documento representa un paso del proceso. Al final de este manual se encuentra un resumen de las acciones a seguir en caso de no necesitar instrucciones detalladas.

Advertencia médica

La información generada por el sistema aquí descrito y su manual no sustituyen de ninguna manera las recomendaciones médicas, diagnósticos o tratamientos profesionales. La información generada por el sistema y su manual únicamente debe ser utilizada con propósitos informativos.

Alexander Mariel no asume ninguna responsabilidad sobre las consecuencias directas o indirectas que se deriven del uso de la información generada por este sistema y su manual.

El usuario jamás debe descartar consejo médico profesional debido a las razones que haya podido generar a partir del uso de esta aplicación.

Advertencia general

Alexander Mariel no asume ninguna responsabilidad sobre las consecuencias directas o indirectas que se deriven del uso del sistema electrónico.

El usuario será responsable de comprender y aplicar las medidas de seguridad requeridas por las regulaciones vigentes en materia de dispositivos médicos. El usuario también velará por comprender y aplicar todas las medidas de seguridad necesarias para el correcto funcionamiento y uso del sistema electrónico.

Las guías aquí descritas únicamente deben de ser utilizadas con propósitos informativos.

Requisitos

Para poder crear el electrocardiógrafo y utilizarlo son necesarios los siguientes componentes:

- Placa Arduino UNO
- Placa Olimex ECG Shield
- Electrodo de pulsera para la placa Olimex ECG Shield o similar
- Dispositivo Android con la versión 4.2.X o superior del sistema operativo

1. Colocación y conexión de los electrodos

Los electrodos deben colocarse en ambas muñecas y en el tobillo de la pierna derecha del sujeto de pruebas. Ha de comprobarse que la adherencia de los electrodos es correcta, sobre todo es importante que la parte central, la cual contiene el gel conductor, esté en contacto total con la superficie de la piel del sujeto.

A continuación deberán conectarse los cables de la placa Olimex a los electrodos de la siguiente manera:

- Cable etiquetado como **L** conectado al electrodo de la muñeca izquierda
- Cable etiquetado como **R** conectado al electrodo de la muñeca derecha
- Cable etiquetado como **D** conectado al electrodo situado en el tobillo

2. Preparación y activación del sistema electrónico

El primer paso para ensamblar el sistema electrónico es montar la placa *Olimex ECG Shield* sobre la placa *Arduino*. Para ello deben alinearse los conectores de la placa Olimex con los de la placa Arduino y unir ambas placas. Ya que el número de conectores es diferente en ambas placas, éstas deben alinearse en la zona opuesta a donde están los conectores para los electrodos (en la placa Olimex) y el conector USB (en la placa Arduino). En la Figura 1 se representa esta indicación.

A continuación deberá conectarse el módulo Bluetooth según el diagrama que se muestra en la Figura 2. Por último deberá conectarse la alimentación, es decir las baterías. El pack de baterías contendrá 5 pilas AA. Éstas serán conectadas en serie y los polos negativo y positivo resultantes serán conectados como se indica en la Figura 2.

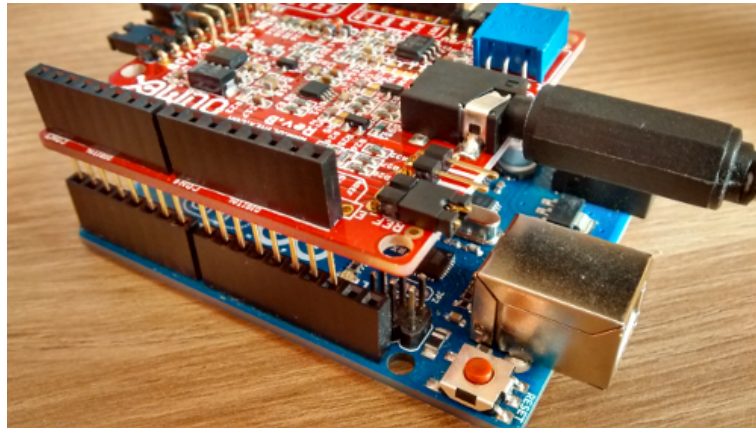


Figura 1: Conexión entre la placa arduino y la placa Olimex. Nótese como los dos pines más cercanos a la cámara en la placa arduino no están poblados y el pin más lejano sí lo está.

Llegado este momento compruébese que el LED *ON* de la placa Arduino UNO y el LED *PWR* de la placa Olimex estén encendidos. En caso contrario comprobar las conexiones del pack de baterías.

3. Acoplamiento Bluetooth

Si es la primera vez que se usa el electrocardiógrafo con el dispositivo Android elegido, entonces los dispositivos deben de ser emparejados. Si desconoce el procedimiento, consúltese el Anexo A de este manual. En caso de no ser necesario el emparejamiento, continúese la lectura de esta sección.

El primer paso para establecer una conexión con el electrocardiógrafo ensamblado en la sección anterior es establecer una conexión entre dicho dispositivo y la aplicación. Para ello ejecútese la aplicación Android generada en el proyecto “Un electrocardiógrafo inteligente de bajo coste”. Aparecerá un listado de dispositivos Bluetooth emparejados con el smartphone o tablet Android, selecciónese el correspondiente al módulo Bluetooth conectado al Arduino.

4. Aplicación Android

Si se han cumplido todas las indicaciones correctamente hasta el momento deberá visualizarse en el dispositivo Android una imagen como la de la captura de la Figura 3.

Si el electrocardiograma no es correcto síganse los siguientes pasos:

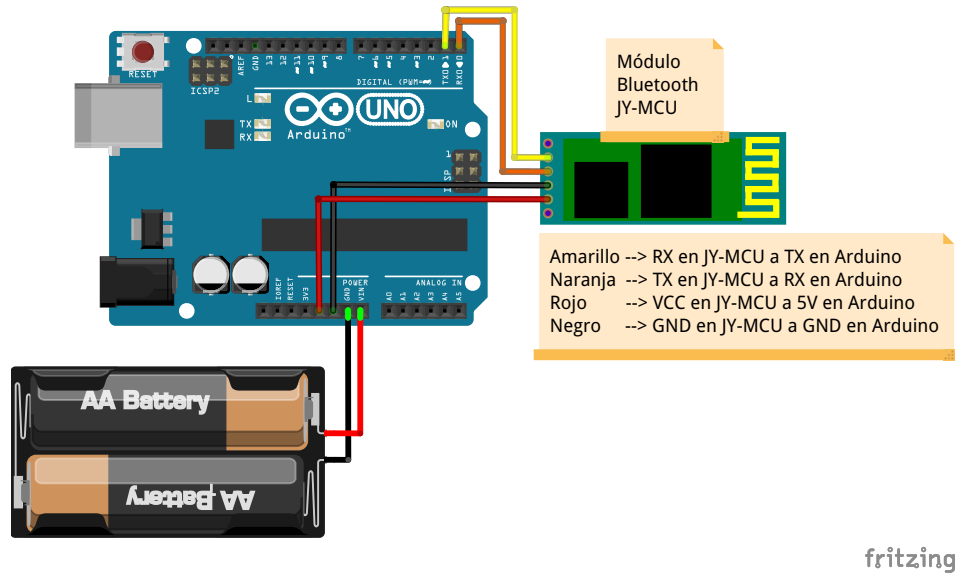


Figura 2: Diagrama de conexión del módulo Bluetooth y las baterías

1. Asegurar que el sujeto de pruebas se encuentra en reposo, el sujeto debe evitar cualquier movimiento o el habla
2. Comprobar que los cables están en buen estado y que no se han desconectado de los electrodos
3. Comprobar que los cables están correctamente conectados a los electrodos correspondientes

En caso de que siguiendo las anteriores indicaciones el electrocardiograma siga siendo defectuoso repítanse los pasos de la sección 2. Si aún así el ECG sigue siendo defectuoso entonces contactar con el creador de este manual.

5. Diagnóstico automatizado

Para realizar el diagnóstico debe desplegarse el menú lateral desde la izquierda. Para ello debe arrastrarse el dedo desde el borde izquierdo del dispositivo hacia el centro de la pantalla. En dicho menú seleccionar la opción *Diagnóstico*. Explicar al sujeto de pruebas que debe mantenerse inmóvil durante los siguientes diez (10) segundos y pulsar el botón *Start*.



Figura 3: Captura de pantalla del proceso de visualización de ECG

Tras unos momentos aparecerán en pantalla una serie de datos relativos al ECG del sujeto y un breve diagnóstico de las alteraciones cardíacas que pudiera estar sufriendo. Si la lista está vacía, entonces el sujeto no debería sufrir ninguna patología cardíaca.

Resumen

1. Adherir electrodos al sujeto de pruebas
2. Conectar los cables del Olimex a los electrodos correctamente
3. Ensamblar el sistema electrónico: Unir placas Arduino y Olimex, conectar módulo Bluetooth, conectar baterías. Seguir el diagrama 2
4. Emparejar módulo Bluetooth con el dispositivo Android (únicamente necesario la primera vez)
5. Iniciar app
6. Seleccionar módulo Bluetooth correspondiente al arduino en la app

Para iniciar el diagnóstico automatizado síganse los pasos de la sección 5.

A. Emparejamiento Bluetooth en Android

Como requisito para este anexo deben haberse completado previamente los pasos de la sección 2.

Estos son los pasos para emparejar el dispositivo Android con el dispositivo Bluetooth conectado a la placa Arduino. Estos pasos han sido escritos para la versión **4.4.2** del sistema operativo Android en una versión *no modificada* por el fabricante del dispositivo.

1. Acceder al menú de ajustes
2. Navegar a la categoría Bluetooth y seleccionar ésta
3. Activar la comunicación Bluetooth
4. Esperar unos momentos hasta que aparezca el nombre del dispositivo Bluetooth conectado al Arduino, este nombre es variable. En caso de no conocerlo consúltese el manual del módulo Bluetooth o al fabricante del mismo
5. Seleccionar el módulo Bluetooth
6. Introducir el código de emparejamiento. En caso de no conocerlo consúltese el manual del módulo Bluetooth o al fabricante del mismo
7. Finalizar emparejamiento