

APLICACIÓN DE ESCRITORIO BASADA EN DEEP LEARNING PARA WINDOWS
QUE CLASIFICA SEÑALES ECG EN SEÑAL CON ARRITMIA FIBRILACIÓN
AURICULAR, RUIDO O NORMAL

ALEJANDRO PULIDO SARAVIA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍA FÍSICO-MECÁNICA
ESCUELA DE INGENIERÍA ELÉCTRICA,
ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA

2023

APLICACIÓN DE ESCRITORIO BASADA EN DEEP LEARNING PARA WINDOWS
QUE CLASIFICA SEÑALES ECG EN SEÑAL CON ARRITMIA FIBRILACIÓN
AURICULAR, RUIDO O NORMAL

ALEJANDRO PULIDO SARA VIA

Tesis de grado para optar al título de Ingeniero Electrónico

Director

Carlos Augusto Fajardo Ariza,
Doctor en Ingeniería.

Codirector

Jaime Guillermo Barrero Pérez,
Magíster en Potencia Eléctrica.

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍA FÍSICO-MECÁNICA
ESCUELA DE INGENIERÍA ELÉCTRICA,
ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA

2023

Agradecimientos

Deseo expresar mi más profundo agradecimiento a todas las personas y seres que creyeron en mí y en mis capacidades para lograr mis objetivos. También agradezco a aquellos que me brindaron apoyo y oportunidades para desarrollarme y adquirir las habilidades que poseo. No puedo dejar de mencionar a todas aquellas personas que me brindaron su enseñanza y conocimientos para mi formación. Asimismo, me agradezco a mí mismo por nunca darme por vencido y perseverar en la consecución de mis metas.

CONTENIDO

	pág.
INTRODUCCIÓN	10
1. OBJETIVOS	13
1.1. OBJETIVO GENERAL	13
1.2. OBJETIVOS ESPECÍFICOS	13
2. MODELO DE DEEP LEARNING PARA LA DETECCIÓN DE ARRITMIAS CARDIACA	14
3. DESARROLLO DE APLICACIÓN	18
3.1. FLUTTER: EL FRAMEWORK PARA EL DESARROLLO DE LA APLICACIÓN.	18
3.2. BLOC: LA ARQUITECTURA DE LA APLICACIÓN.	18
3.3. DESARROLLO DE LA INTERFAZ GRÁFICA PRINCIPAL	19
3.4. INTERFACES NECESARIAS PARA LA GESTIÓN DE PROYECTOS	20
3.5. CONTROLADOR DE LA APLICACIÓN	20
3.5.1. Comunicación con el microcontrolador	21
3.5.2. Gestión de la Información del Proyecto	22
3.5.3. Clasificación con la red neuronal	24
3.5.4. Cálculo de frecuencia cardiaca	24
3.5.5. Más detalles	26
4. RESULTADOS	27
4.1. APLICACIÓN PARA WINDOWS	27
4.1.1. Pantalla Principal	27

4.1.2. Crear un proyecto	29
4.1.3. Captura de la señal	31
4.1.4. Información del profesional médico	32
4.1.5. Abrir Proyecto	32
4.1.6. Mediciones en la gráfica	33
4.1.7. Exportar como PDF	34
4.1.8. BLE configuracion de conexion	36
4.2. APLICACIÓN PARA ANDROID	37
4.2.1. Permisos	37
4.2.2. Compatibilidad	37
5. CONCLUSIONES	39
BIBLIOGRAFIA	40

LISTA DE FIGURAS

	pág.
Figura 2.1. Modelo de Deep Learning para clasificar señales ECG ⁵	15
Figura 3.1. Ilustración de la arquitectura BLoC ¹³	19
Figura 3.2. GATT estructura ¹⁹	21
Figura 3.3. Estructura de archivos del proyecto almacenado	23
Figura 3.4. Señal original y resultado de aplicar el filtro IIR	25
Figura 3.5. Comparación entre la señal original y la señal con los pulsos identificados correspondientes al complejo QRS	26
Figura 4.1. Pantalla principal de la versión para Windows	28
Figura 4.2. Proyecto nuevo	29
Figura 4.3. Formulario información del paciente	30
Figura 4.4. Selección de dispositivo Bluetooth	30
Figura 4.5. Pantalla capturando una señal ECG	31
Figura 4.6. Información del profesional médico	32
Figura 4.7. Pantalla proyecto cargado	33
Figura 4.8. Herramienta de medición	34
Figura 4.9. Formato página 1 del PDF generado	35
Figura 4.10. Formato de las páginas restantes del PDF generado	36
Figura 4.11. Pantalla principal de la versión para Android	37
Figura 4.12. Permiso de ubicación	38
Figura 4.13. Permiso de almacenamiento	38

LISTA DE TABLAS

	pág.
Tabla 2.1. Parámetros de entrada de la red neuronal	17

RESUMEN

TÍTULO: APLICACIÓN DE ESCRITORIO BASADA EN DEEP LEARNING PARA WINDOWS QUE CLASIFICA SEÑALES ECG EN SEÑAL CON ARRITMIA FIBRILACIÓN AURICULAR, RUIDO O NORMAL *

AUTOR: ALEJANDRO PULIDO SARAVIA **

PALABRAS CLAVE: ARRITMIA CARDIACA, ANDROID, ECG, DEEP LEARNING, FLUTTER, APLICACIÓN, FIBRILACIÓN AURICULAR, WINDOWS, TENSORFLOW.

DESCRIPCIÓN:

La fibrilación auricular es una de las arritmias más comunes y con alta mortalidad y morbilidad. Esta arritmia puede presentar una atención tardía y costosa. En este trabajo se desarrolló una aplicación que brinda la posibilidad de capturar, visualizar y guardar una señal electrocardiográfica (ECG), así mismo auto evalúa dicha señal con el fin de clasificar en señal con arritmia fibrilación auricular, ruido o normal para una acción preventiva. Se desarrolló usando el SDK Flutter con una arquitectura BLOC donde implementamos una red neuronal que evalúa la señal capturada. Por medio de un microcontrolador ESP32 y un AD8232 logramos capturar la señal y enviarla al dispositivo de cómputo usando una conexión Bluetooth Low-Energy (BLE). Adicionalmente, la aplicación permite leer las señales ECG para una interpretación manual donde se puede cambiar los parámetros de visualización y una regla que facilita hacer mediciones de la señal. Adicionalmente, la aplicación cuenta con información del paciente y del profesional que realizó el examen de captura e incluso. Toda la información es factible de ser exportada a un documento en formato PDF para imprimir o archivar en el historial clínico. La aplicación es compatible para equipos con sistema operativo Windows y dispositivos con sistema operativo Android. Se espera que esta aplicación contribuya con la detección de arritmias cardiacas de forma temprana y de esta forma ayudar a su tratamiento.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Carlos Augusto Fajardo Ariza, Doctor en Ingeniería. Codirector: Jaime Guillermo Barrero Pérez, Magíster en Potencia Eléctrica.

ABSTRACT

TITLE: DEEP LEARNING BASED DESKTOP APPLICATION FOR WINDOWS THAT CLASSIFIES EKG SIGNALS INTO ATRIAL FIBRILLATION ARRHYTHMIA, NOISE OR NORMAL SIGNAL *

AUTHOR: ALEJANDRO PULIDO SARAIVA **

KEYWORDS: CARDIAC ARRHYTHMIA, ANDROID, EKG, DEEP LEARNING, FLUTTER, APP, ATRIAL FIBRILLATION, WINDOWS, TENSORFLOW.

DESCRIPTION:

Atrial fibrillation is one of the most common arrhythmias with high mortality and morbidity. This arrhythmia can present late and costly care. In this work, an application that provides the possibility of capturing, visualizing, and saving an electrocardiographic (ECG) signal, was developed, as well as auto-evaluate this signal in order to classify it as an atrial fibrillation arrhythmia, noise or normal signal for preventive action. The application was developed using the Flutter SDK with a BLOC architecture where we implemented a neural network that evaluates the captured signal. By means of an ESP32 microcontroller and an AD8232 we were able to capture the signal and send it to the computing device using a Bluetooth Low-Energy (BLE) connection. Additionally, the application allows reading the ECG signals for manual interpretation where you can change the display parameters and a ruler that facilitates signal measurements. In addition, the application has information about the patient and the professional who performed the capture and even the exam. All the information can be exported to a PDF document for printing or archiving in the clinical history. The application is compatible with Windows operating system computers and Android operating system devices. It is expected that this application will contribute to the early detection of cardiac arrhythmias and thus help in their treatment.

* Bachelor Thesis

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Carlos Augusto Fajardo Ariza, Doctor en Ingeniería. Codirector: Jaime Guillermo Barrero Pérez, Magíster en Potencia Eléctrica.

INTRODUCCIÓN

El sistema de salud colombiano presenta largas esperas en los centros de atención de las EPS y centros de urgencias. Esto puede llegar a representar un peligro para la vida de los usuarios especialmente cuando se trata de enfermedades cardiovasculares, que son una de las principales causas de muerte en el mundo ^{1,2}. Además, la falta de acceso a la medicina especializada y la alta tasa de diagnóstico tardío en Colombia hace que el tratamiento sea aún más difícil ³.

A nivel mundial las enfermedades cardiovasculares son la primera causa de muerte. Requiriendo estas enfermedades de medicina especializada en Cardiología y puesto que en Colombia se presenta actualmente una cobertura limitada en dicha especialidad, adicionando a eso las largas esperas para poder acceder al servicio por su elevado costo, reduce la posibilidad de recibir un diagnóstico de manera temprana ⁴. Así la detección y diagnóstico de arritmias cardiacas es fundamental para

-
- ¹ Ferran Pons et al. "Mortalidad y Causas de muerte en pacientes con insuficiencia cardiaca: Experiencia De Una Unidad especializada multidisciplinaria". En: *Revista Española de Cardiología* 63.3 (2010), 303–314. DOI: 10.1016/s0300-8932(10)70089-0.
 - ² Valetín Fuster. "Aproximación terapéutica a la epidemia de fibrilación auricular". En: *Revista Española de Cardiología* 55 (2002), págs. 27-32.
 - ³ Martín Romero y Diana Chávez. "Carga de Enfermedad atribuible a fibrilación auricular en Colombia (2000-2009)". En: *Revista Colombiana de Cardiología* 21.6 (2014), 374–381. DOI: 10.1016/j.rccar.2014.08.006.
 - ⁴ Fernando Worner, Alberto San Román, Pedro Luis Sánchez, Ana Viana Tejedor y José Ramón González-Juanatey. "Atención a los pacientes con Enfermedades Cardiacas Agudas y Críticas. Posición de la sociedad española de cardiología". En: *Revista Española de Cardiología* 69.3 (2016), 239–242. DOI: 10.1016/j.recesp.2015.07.018.

realizar un tratamiento oportuno y la prescripción médica ⁵.

Ante la problemática, hace un par de décadas la comunidad científica ha mostrado interés en el uso de estrategias clásicas de Machine Learning para la detección de arritmias cardíacas ⁶. Por lo tanto, recientemente, el Deep Learning ha logrado igualar la precisión de los cardiólogos ⁷.

Puesto que el desarrollo de equipos médicos basados en Deep Learning es una tecnología reciente y que aún está en fase de desarrollo ⁸, una aplicación de bajo costo que utilice Deep Learning para facilitar la generación de alertas tempranas con relación a arritmias cardíacas, que una vez pueda ser instalada en un equipo de cómputo podría facilitarle a un profesional de la salud, no especialista, la remisión de un paciente a centros o especialistas en cardiología.

Así, en pro de mejorar las oportunidades para la salud de las personas, se busca desarrollar un modelo funcional de un sistema de detección automática, para que trabajos futuros puedan llevarlo a un siguiente nivel de desarrollo tecnológico, brindando un bien para todos lo que lo puedan llegar a necesitar.

⁵ Karen Fonseca, Sergio Osorio, Jeyson Castillo y Carlos Fajardo. “Contrastive Learning for atrial fibrillation detection in challenging scenarios”. En: *2022 30th European Signal Processing Conference (EUSIPCO)* (2022). DOI: 10.23919/eusipco55093.2022.9909842.

⁶ D.A. Coast, R.M. Stern, G.G. Cano y S.A. Briller. “An approach to cardiac arrhythmia analysis using Hidden Markov models”. En: *IEEE Transactions on Biomedical Engineering* 37.9 (1990), 826–836. DOI: 10.1109/10.58593.

⁷ Awni Y. Hannun et al. “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network”. En: *Nature Medicine* 25.1 (2019), 65–69. DOI: 10.1038/s41591-018-0268-3.

⁸ Lorenzo Gutierrez. *Beyond the hype: Medical Device Artificial Intelligence (AI)*. 2022.

El resto del documento se organiza de la siguiente manera: En la sección 2 se presenta el modelo de Deep Learning utilizado. La sección 3 detalla el proceso de desarrollo de la aplicación. En la sección 4 se presentan los resultados obtenidos y en la sección V se incluyen las conclusiones finales del presente trabajo.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

- Desarrollar una aplicación de escritorio basada en Deep Learning para la clasificación de señales ECG en señales con arritmia fibrilación auricular, Ruido o Normal.

1.2. OBJETIVOS ESPECÍFICOS

- Implementar en tensorflow lite el modelo pre-entrenado [3], el cual será utilizado para la clasificación.
- Diseñar y montar en el SDK Flutter Back-End de la aplicación.
- Diseñar y montar en el SDK Flutter Front-End de la aplicación.

2. MODELO DE DEEP LEARNING PARA LA DETECCIÓN DE ARRITMIAS CARDIACA

En este proyecto se utilizó un modelo de Deep Learning para detectar arritmias cardíacas, como se ilustra en la figura 2.1. Este modelo fue propuesto por Fajardo et al. en ⁵.

Este modelo utiliza capas convolucionales y pooling. Adicionalmente utiliza bloques residuales y conexiones de salto para mejorar la propagación de la información en redes neuronales profundas. También utiliza capas de Batch Normalization para mantener los valores dentro de los límites y evitar la saturación, y Dropout para evitar el sobreajuste durante el entrenamiento ⁵. La función de activación utilizada es la ReLU, que ha demostrado mejorar tanto la precisión como el coste computacional ⁵. La cantidad de bloques residuales (N) es un hiper parámetro que permite variar la profundidad del modelo. Las primeras y últimas capas del modelo son casos especiales debido a una estructura de pre-activación. Para mayor detalles acerca del modelo se refiere el lector a las referencias ⁵ y ⁹.

Este modelo utiliza capas convolucionales y pooling. Adicionalmente utiliza bloques residuales y conexiones de salto para mejorar la propagación de la información en redes neuronales profundas. También utiliza capas de Batch Normalization para mantener los valores dentro de los límites y evitar la saturación, y Dropout para evitar el sobreajuste durante el entrenamiento ⁵. La función de activación utilizada es la ReLU, que ha demostrado mejorar tanto la precisión como el coste computacional

⁹ Carlos A. Fajardo. *Cafajar-uis/ECG_Compressed_Model*.

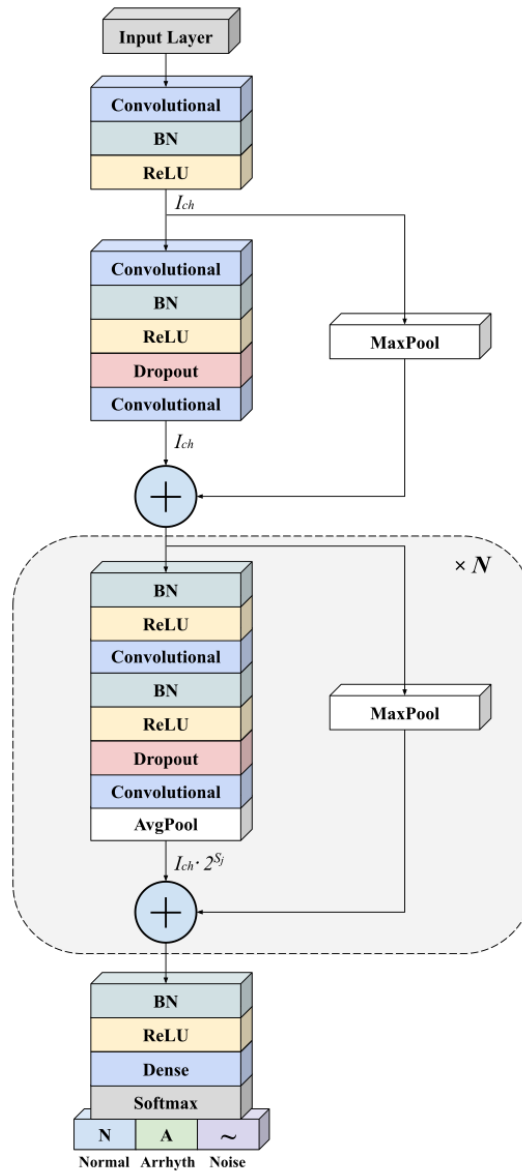


Figura 2.1. Modelo de Deep Learning para clasificar señales ECG ⁵

⁵. La cantidad de bloques residuales (N) es un hiper parámetro que permite variar la profundidad del modelo. Las primeras y últimas capas del modelo son casos especiales debido a una estructura de pre-activación. Para mayor detalles acerca del modelo se refiere el lector a las referencias ⁵ y ⁹.

El modelo implementado en este proyecto tiene un total de 13 bloques residuales (N=13, en la Figura 2.1), cuatro canales Ich y 73343 parámetros. Este modelo es equivalente al modelo CNN5 mencionado en ⁵. Para este proyecto, se utilizó un modelo previamente entrenado en TensorFlow ¹⁰ que cumple con estas características, obtenido de ⁹.

Para la implementación del modelo en el dispositivo final utilizamos TensorFlow Lite.

TensorFlow Lite es un conjunto de herramientas que ayuda a los desarrolladores a ejecutar sus modelos en dispositivos incorporados, móviles o de IoT, y les permite implementar el aprendizaje automático integrado en el dispositivo ¹⁰.

Para utilizar correctamente esta red neuronal de clasificación es necesario adaptar los datos de entrada. La red neuronal requiere procesar señales ECG de 2049 muestras, las cuales son obtenidas mediante un microcontrolador. Estos datos deben ser recolectados a una frecuencia de 250 Hz. Cada valor de los datos de entrada representa los milivoltios de la señal ECG en su correspondiente momento temporal. Este capítulo busca sentar las bases necesarias para facilitar la lectura de los demás capítulos del libro. En este sentido resulta importante comenzar por comprender las necesidades de comunicación que pueden llegar a tener los primeros satélites hechos en Colombia. Describir los conceptos teóricos en que se apoya el proyecto,

¹⁰ Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

así como los conceptos esenciales es el siguiente paso. Pero en este capítulo también se realiza un avance al diseño al identificar restricciones necesarias como las bandas de frecuencias a usar, los protocolos a respetar entre otros requerimientos.

Parámetros de entrada de la red neuronal	Valor
Tamaño de entrada	2049 muestras
Frecuencia de muestreo	250 Hz
Unidad de cada dato	milivolts

Tabla 2.1. Parámetros de entrada de la red neuronal

3. DESARROLLO DE APLICACIÓN

A continuación se detallan el procedimiento y herramientas empleadas en el desarrollo de esta aplicación.

3.1. FLUTTER: EL FRAMEWORK PARA EL DESARROLLO DE LA APLICACIÓN.

Para el desarrollo de aplicación utilizamos Flutter, que es un framework desarrollado por Google con el fin de facilitar el desarrollo de aplicaciones móviles con la posibilidad de exportarlo a otras plataformas de forma nativa, haciendo uso del mismo código de base ¹¹. Su lenguaje de programación es Dart que es muy versátil, fácil de aprender y de alto nivel. Flutter cuenta con una gran variedad de características, herramientas y paquetes que permiten el uso de diferentes funciones. Para la creación del entorno gráfico, Flutter usa Widgets ¹² los cuales cuentan con diferentes configuraciones para alterar o cambiar su visualización.

3.2. BLOC: LA ARQUITECTURA DE LA APLICACIÓN.

Por otro lado, se utilizó BLoC ¹³ con el fin de mantener el código organizado, mejorar el rendimiento de la aplicación y facilitar el mantenimiento y/o revisión. BLoC es una arquitectura de desarrollo de software que se enfoca en la división de la aplicación por áreas de trabajo. Es una variación de la arquitectura MVP (Modelo-

¹¹ Flutter. *Build apps for any screen.*

¹² *Introduction to widgets.*

¹³ Felix Angelov.

Vista-Controlador)¹⁴ y MVVM (Modelo-Vista-VistaModelo)¹⁴ que se puede utilizar en Flutter. Este se basa en el patrón Reactive Programming¹⁵, que permite gestionar el flujo de la aplicación de forma asíncrona manejando estados y eventos.

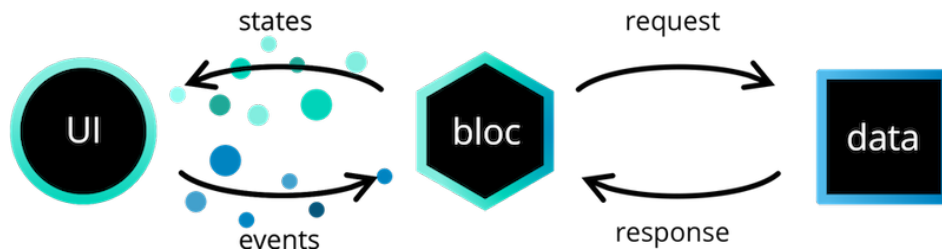


Figura 3.1. Ilustración de la arquitectura BLoC¹³

La arquitectura BLoC se divide en tres componentes principales:

- La vista (UI): Es la interfaz gráfica que el usuario ve e interactúa.
- El BLoC: Es el componente encargado de la lógica de negocio y de la gestión de los estados de la aplicación.
- Los datos: Son la fuente de información que se utiliza en la aplicación.

3.3. DESARROLLO DE LA INTERFAZ GRÁFICA PRINCIPAL

Se diseñó la pantalla principal de la aplicación considerando las necesidades esenciales. Se utilizó el paquete `fl_chart`¹⁶ que permite crear gráficos y estadísticas de manera sencilla y personalizada, para crear una gráfica de la señal ECG, configurada como una hoja de resultados ECG con una cuadrícula que muestra la amplitud

¹⁴ Kevin Morales. *Arquitecturas de software en Android: MVC, MVP Y MVVM*. 2019.

¹⁵ André Medeiros. *The introduction to reactive programming you've been missing*. 2014.

¹⁶ Kael'thas. *Fl_chart: Flutter Package*. 2023.

de la señal en milivoltios y el tiempo transcurrido en segundos. Se incluyó una sección similar a un osciloscopio para mostrar en tiempo real la frecuencia cardíaca, los resultados de la clasificación de la red neuronal y opciones para cambiar la escala y acercar/alejar la gráfica. Para facilitar la gestión de los datos, se agregó un menú vertical con botones para administrar proyectos y crear/cargar archivos existentes.

3.4. INTERFACES NECESARIAS PARA LA GESTIÓN DE PROYECTOS

Para la gestión de los proyectos, se diseñaron varias pantallas que brindan las acciones y la información necesarias, estas pantallas se muestran sobre la pantalla principal en forma de diálogo flotante. La creación de proyectos requiere tres pantallas: la primera con un formulario para definir el nombre y la ubicación del proyecto (usando el paquete `file_picker`¹⁷ que proporciona una interfaz sencilla para seleccionar y trabajar con archivos y carpetas desde el sistema de archivos de un dispositivo), la segunda con un formulario para la información básica del paciente, y la tercera para mostrar los dispositivos Bluetooth disponibles para la conexión. Para cargar un proyecto, solo se requiere una pantalla con la ubicación del proyecto (también con `file_picker`¹⁷). Además, se incluyeron dos pantallas más: una para ver la información del paciente y otra para ver la información del profesional médico.

3.5. CONTROLADOR DE LA APLICACIÓN

Se implementó un solo controlador para toda la aplicación, ya que solo tiene una pantalla principal. El controlador gestiona la interacción con esta pantalla y las pantallas secundarias.

¹⁷ Miguel Ruivo. *File_picker: Flutter Package*. 2023.

3.5.1. Comunicación con el microcontrolador El microcontrolador es el responsable de muestrear la señal ECG y enviarla al dispositivo de cómputo mediante una conexión Bluetooth Low-Energy (BLE) ¹⁸. Para la comunicación entre dispositivos BLE, es necesario utilizar un Generic Attribute Profile (GATT) ¹⁹, que se compone de servicios y características. Los dispositivos BLE pueden ser Maestros o Esclavos, donde el Maestro proporciona los servicios y el Esclavo accede a ellos. Solo se puede establecer una conexión entre dos dispositivos, por lo que el Maestro anuncia que está disponible para conectarse hasta que otro dispositivo se conecta con él. En este proyecto, el microcontrolador es el maestro y la aplicación es el esclavo.

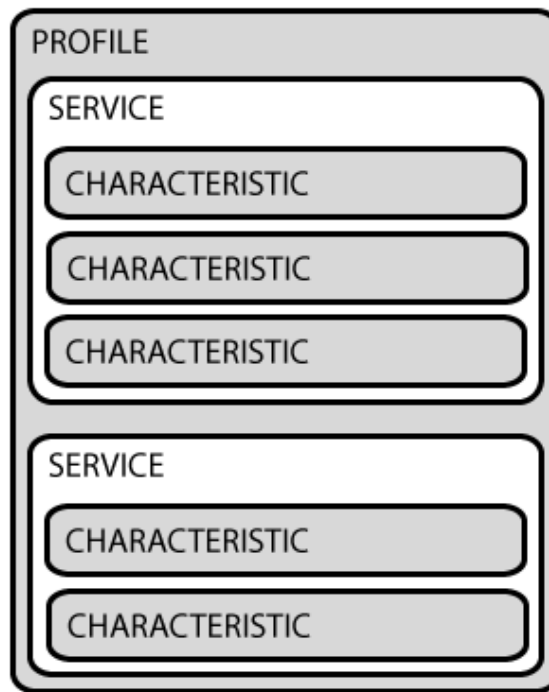


Figura 3.2. GATT estructura ¹⁹

¹⁸ *Bluetooth technology overview.*

¹⁹ Kevin Townsend. *Introduction to bluetooth low energy.* 2014.

Cada dispositivo BLE como maestro, puede tener diferentes servicios, los cuales contienen características específicas. Cada característica puede tener varios descriptores, los cuales establecen qué acciones se pueden realizar con dicha característica. Algunos de estos descriptores son el de lectura, que permite al dispositivo esclavo leer la característica, el de escritura, que permite al dispositivo esclavo escribir en la característica, y el notificable, que notifica al dispositivo esclavo cuando se cambia el valor de la característica. Tanto los servicios, características y descriptores son identificados mediante un Identificador Único Universal (UUID).

Al momento de establecer la conexión bluetooth, se revisan primero los permisos de Bluetooth y ubicación con el paquete `permission_handler`²⁰ que facilita la gestión de permisos del sistema en aplicaciones. Luego, con el paquete `quick_blue`¹² que proporciona una interfaz simple e intuitiva para trabajar con tecnología Bluetooth Low-Energy (BLE) en aplicaciones desarrolladas con Flutter, se escanean los dispositivos disponibles y se muestran en la pantalla de dispositivos Bluetooth disponibles. Después de que el usuario seleccione el dispositivo, se realiza la conexión con `quick_blue`. Tras una conexión exitosa, se verifica si el dispositivo tiene el servicio y característica necesarios para la captura de la señal ECG y se establece una conexión de notificación para empezar a capturar la señal ECG. La señal capturada se envía a la pantalla principal para ser visualizada en la gráfica ECG.

3.5.2. Gestión de la Información del Proyecto El manejo de datos del proyecto implica cómo se guarda y accede a la información relacionada con el proyecto. Al crear un proyecto, se registran los detalles de ubicación, nombre y paciente. Luego, tras una conexión exitosa con el microcontrolador, se crea una carpeta en la

²⁰ Baseflow. *Permission_handler: Flutter Package*. 2022.

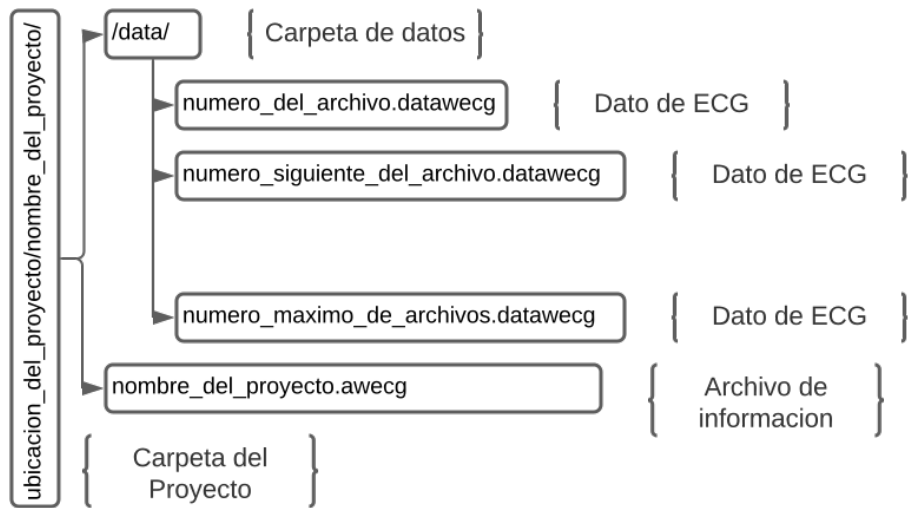


Figura 3.3. Estructura de archivos del proyecto almacenado

ubicación especificada con el nombre del proyecto y dentro de ella, un archivo con extensión “.awecg” que incluye información detallada sobre el proyecto, incluyendo el número de archivos capturados, información del paciente, fecha de creación, información del profesional médico y los nombres de los archivos capturados.

La información del médico se obtiene y almacena mediante el paquete `shared_preferences`¹⁴ que facilita la gestión de permisos del sistema en aplicaciones. Si no hay información registrada, se establecerá como vacía. Para agregar la información del médico, el usuario debe hacerlo en la pantalla de información del médico y guardarla usando `shared_preferences`¹⁴.

Al registrar los datos, cada vez que se completan 2049 valores, se crea un archivo en una carpeta llamada "data", con el nombre "número_del_archivo.dataawecg". Luego, se actualiza el archivo “nombre_del_proyecto.awecg” con la información adicional del proyecto. El número del archivo va incrementando a medida que llegan más datos.

Durante la carga de un proyecto, se accede a su carpeta y se verifica la existencia del archivo con el mismo nombre que la carpeta y con extensión ".awecg". Se lee el archivo y se confirma que todos los archivos de datos descritos en él existan. Luego, se cargan los datos y luego se grafica el primer registro ECG para su análisis. Si los datos no son válidos, se rechaza la apertura del proyecto y se informa al usuario sobre su daño.

3.5.3. Clasificación con la red neuronal En el momento en que se completan 2049 datos capturados o cuando se carga un proyecto y se muestra la señal ECG, esta es evaluada con el uso del paquete `tflite_flutter`²¹ que permite cargar el modelo de red neuronal en formato TensorFlow Lite¹⁰ para crear un intérprete, y así permitir obtener su clasificación y enviarla a la pantalla principal para ser mostrado el resultado junto con su probabilidad.

3.5.4. Cálculo de frecuencia cardíaca Para determinar la frecuencia cardíaca, en el momento en que se completan 2049 datos capturados o cuando se carga un proyecto y se muestra la señal ECG. Primero, se aplica un filtro IIR a la información siguiendo la ecuación 1. Luego, se identifican los puntos superiores al umbral de 0.04 utilizando un detector de umbral, marcándose con un 1. Posteriormente, se realiza una operación de convolución con un kernel de 30 unos para localizar cada complejo QRS. Finalmente, se calcula el tiempo promedio entre cada complejo y se utiliza como una estimación aproximada de la frecuencia cardíaca por medio de la ecuación 2.

²¹ Amish Garg. *Tflite_flutter: Flutter Package*. 2021.

$$y[n] = 0.05833987x[n] - 0.05833987x[n - 2] + 1.72253523y[n - 1] - 0.88332027y[n - 2] \quad (1)$$

$$HR = \frac{60}{RR} \quad (2)$$

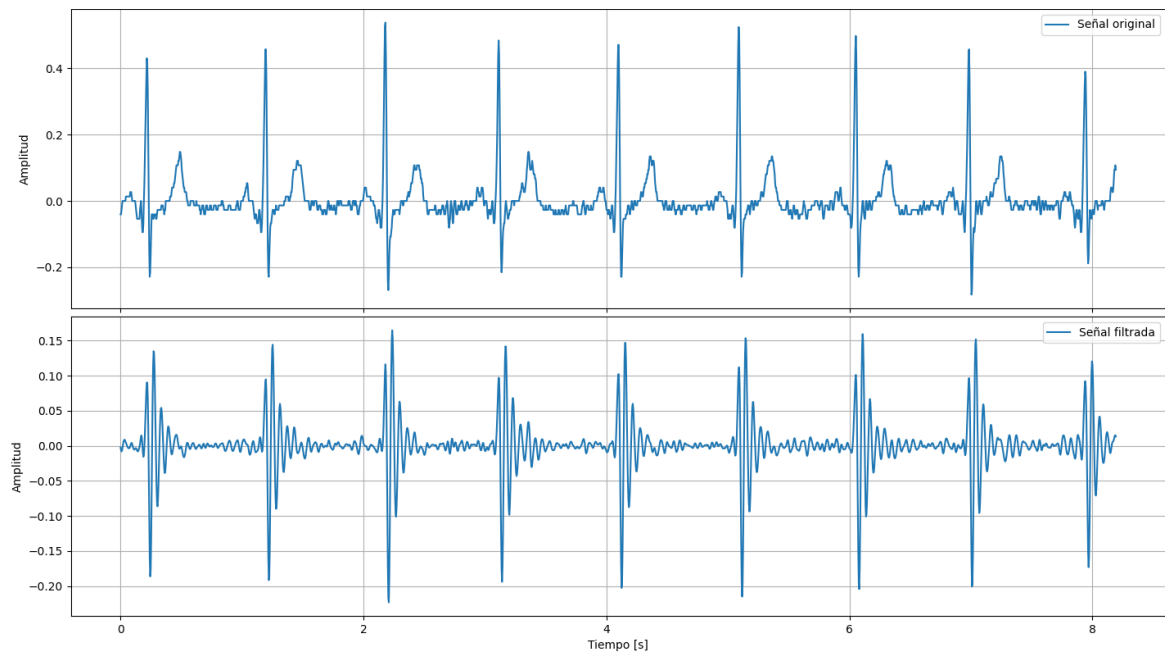


Figura 3.4. Señal original y resultado de aplicar el filtro IIR

La ecuación 1 es un filtro pasa banda Butterworth de primer orden que es la discretización del filtro propuesto por Thakor et. al.²². Este filtra la señal de ECG y extraer exclusivamente el complejo QRS. Con este filtrado, se garantiza una mayor precisión en la identificación del complejo QRS.

En la ecuación 2, HR representa la frecuencia cardíaca mientras que RR representa

²² Nitish V. Thakor, John G. Webster y Willis J. Tompkins. "Estimation of QRS Complex Power Spectra for design of a QRS filter". En: *IEEE Transactions on Biomedical Engineering* BME-31.11 (1984), 702–706. DOI: 10.1109/tbme.1984.325393.

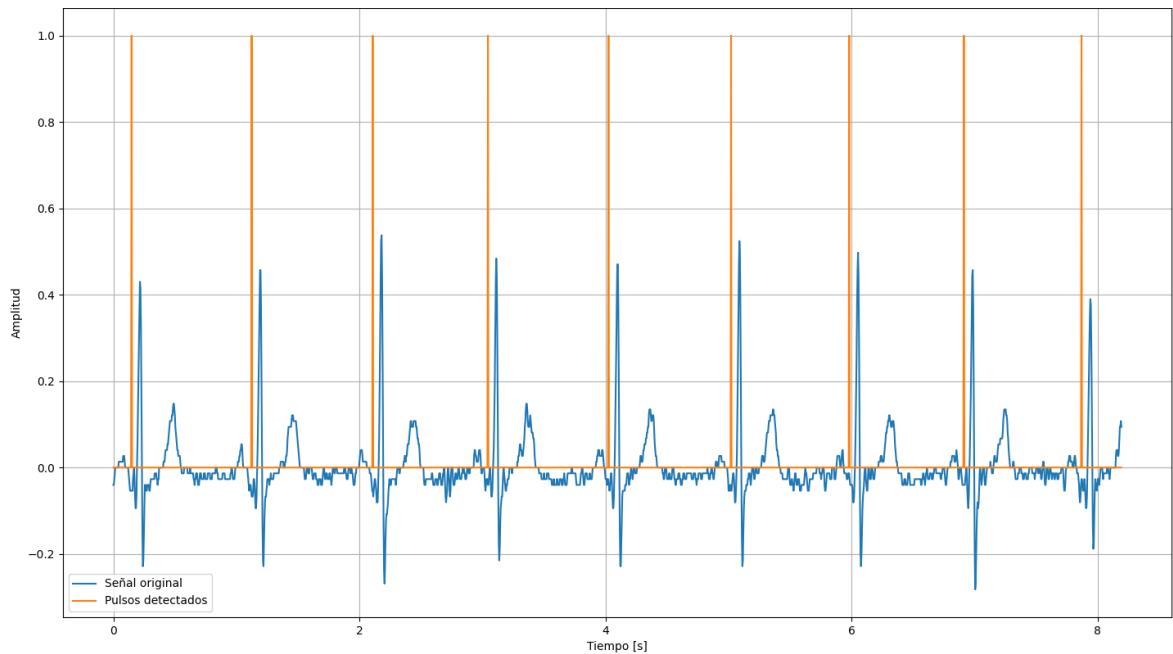


Figura 3.5. Comparación entre la señal original y la señal con los pulsos identificados correspondientes al complejo QRS

el tiempo estimado entre cada complejo QRS.

3.5.5. Más detalles El código fuente de la aplicación desarrollada en este proyecto se encuentra disponible en ²³.

²³ Alejandro Pulido. *Alpusa/AWECEG: Flutter ECG application to Windows and Android*. 2023.

4. RESULTADOS

4.1. APLICACIÓN PARA WINDOWS

Se desarrolló una aplicación de escritorio para Windows que tiene como función principal clasificar una señal electrocardiográfica (ECG) capturada a través de un microcontrolador ESP32 ²⁴ y un circuito integrado AD8232 ²⁵. La comunicación entre el dispositivo donde se ejecuta la aplicación y el circuito se realiza mediante Bluetooth Low Energy (BLE) ¹⁸. Para el desarrollo de la aplicación se utilizó el SDK de Flutter ¹¹, implementando una arquitectura basada en BLoC con el objetivo de optimizar su rendimiento y mantener un código ordenado y estructurado.

4.1.1. Pantalla Principal Se realizó con una interfaz amigable (Figura 4.1) y fácil de usar, donde al sobreponer el cursor sobre cada botón, este indicará su nombre o funcionamiento.

Los principales componentes de la aplicación son:

1. **Menú Principal:** Se visualiza en la parte izquierda una barra de botones que nos permite crear o cargar un proyecto, ver la información del paciente, exportar como pdf la señal capturada y editar (si es cargado el proyecto) o visualizar la información del profesional médico.
2. **Sección Gráfica:** En el centro de la aplicación se presenta una cuadrícula en la que se verá graficada la señal de ECG. También cuenta con dos espacios

²⁴ espressif. *ESP32WROOM32E ESP32WROOM32UE*. Espressif Systems. 2020.

²⁵ Analog Devices. *AD8232*. 2019.



Figura 4.1. Pantalla principal de la versión para Windows

de texto en la parte inferior, en uno de ellos indica el nombre del proyecto si este ha sido cargado y en el otro nos indica la escala y velocidad de la gráfica de la señal ECG.

- Herramientas:** Por la derecha tenemos tres secciones con diferentes opciones: en la primer sección indica la frecuencia cardiaca medida digitalmente con la señal ECG; la segunda sección cuenta con 6 botones que sirven para cambiar la escala, la velocidad, acercar, alejar, restablecer la escala y velocidad, restablecer el zoom, adicionalmente tiene un Slider que permite mover la gráfica verticalmente; la tercera sección muestra el resultado de la clasificación usando la red neuronal ⁵, así indicará si la señal es arritmia, ruido o normal consecutivamente también el porcentaje de probabilidad que este resultado presenta.



Figura 4.2. Proyecto nuevo

4.1.2. Crear un proyecto Para la creación de un proyecto nuevo es necesario seleccionar la carpeta donde se guardará el proyecto usando el botón de búsqueda, también colocar un nombre o generar uno automáticamente con el botón “Generar” (Figura 4.2); seguido aparecerá un formulario donde se deberá completar con la información del paciente (Figura 4.3).

Información del Paciente

Complete la información del paciente. Todos los campos son requeridos

Nombre _____

Edad _____

Teléfono _____

Dirección _____

Email _____

Identificación _____

[Siguinte](#) [Cancelar](#)

Figura 4.3. Formulario información del paciente

Seleccione el dispositivo Bluetooth

AWECG Monitor
RSSI: -54

[Cancelar](#)

[🔍](#)

Figura 4.4. Selección de dispositivo Bluetooth

Después de completar el formulario se debe seleccionar el dispositivo bluetooth correspondiente (el microcontrolador que debe tener el paciente), este se identifica por la MAC del dispositivo o por el nombre, también por su RSSI y ya empezar la toma de los datos.

4.1.3. Captura de la señal En la toma de la señal ECG se puede ver en vivo cómo se va graficando a medida que va obteniendo datos y cada que completa los datos necesarios para la clasificación mediante el uso de la red neuronal ⁵ en señal con arritmia fibrilación auricular, ruido o normal, mostrando también el porcentaje de probabilidad del resultado y también en pantalla se muestra la frecuencia cardiaca de dicha trama de datos. Aparece un botón en la parte superior indicando que está conectado y al dar click finaliza la conexión y así mismo la toma de datos.

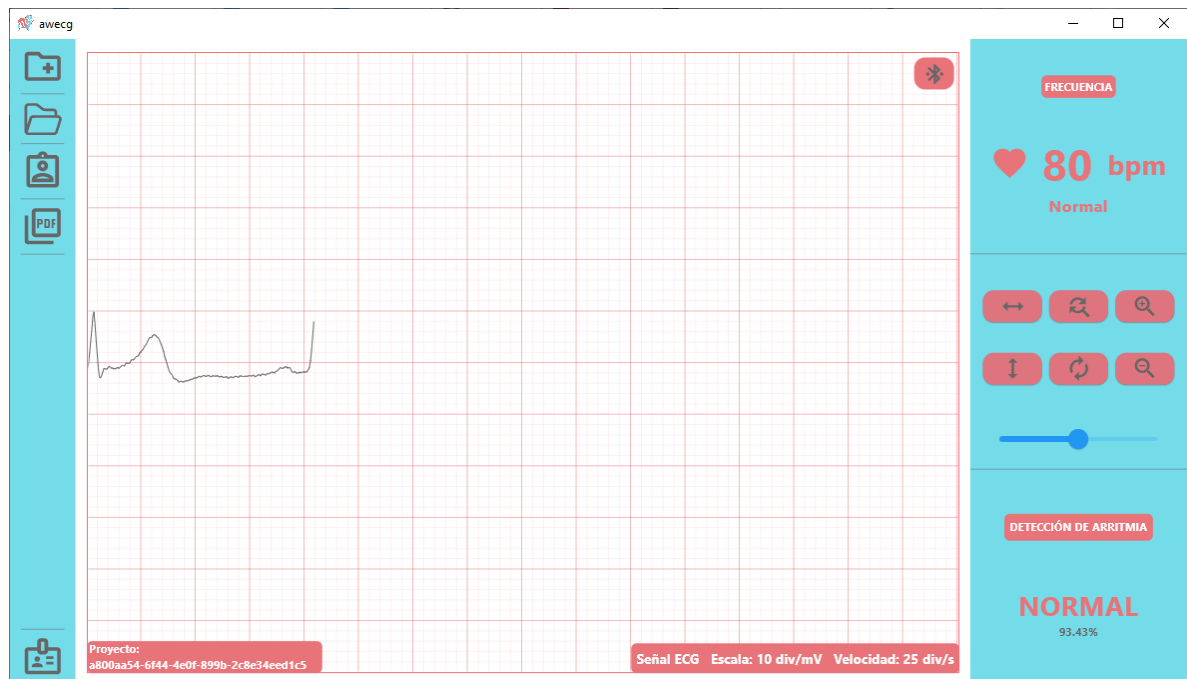


Figura 4.5. Pantalla capturando una señal ECG

4.1.4. Información del profesional médico Cada vez que creamos un proyecto, este se crea con la información que esté establecida en su momento en la aplicación y cuando se carga un proyecto, este mostrará la información del médico que realizó el examen. Para ver o editar la información del médico profesional se realiza en el último botón de la barra lateral izquierda y editar el formulario que aparece (Figura 4.6).



Formulario de información del profesional médico con los siguientes campos:

Nombre	Alejandro Pulido Saravia
Teléfono	3508977722
Dirección	Calle 17 # 26-65 apto 201
Email	pulido_saravia@hotmail.com
Identificación	cc 1098798329
Especialidad	Estudiante de Ingeniería Electro
Lugar	Casa

Botones: Guardar, Limpiar, Cancelar

Figura 4.6. Información del profesional médico

4.1.5. Abrir Proyecto Para abrir o cargar un proyecto debe presionar en el segundo botón de la barra lateral izquierda donde mostrará un formulario para seleccionar la carpeta del proyecto y abrirlo. Posteriormente se cargará la señal ECG y mostrará las evaluaciones por trama de datos, apareciendo un Slider (Figura 4.7) superior con el que se puede desplazar por toda la señal y al final o inicio de la trama si hay otra trama disponible, anterior o siguiente aparece un botón para continuar a esa trama. Al pasar el cursor por la señal, se mostrará el voltaje y tiempo en la trama.

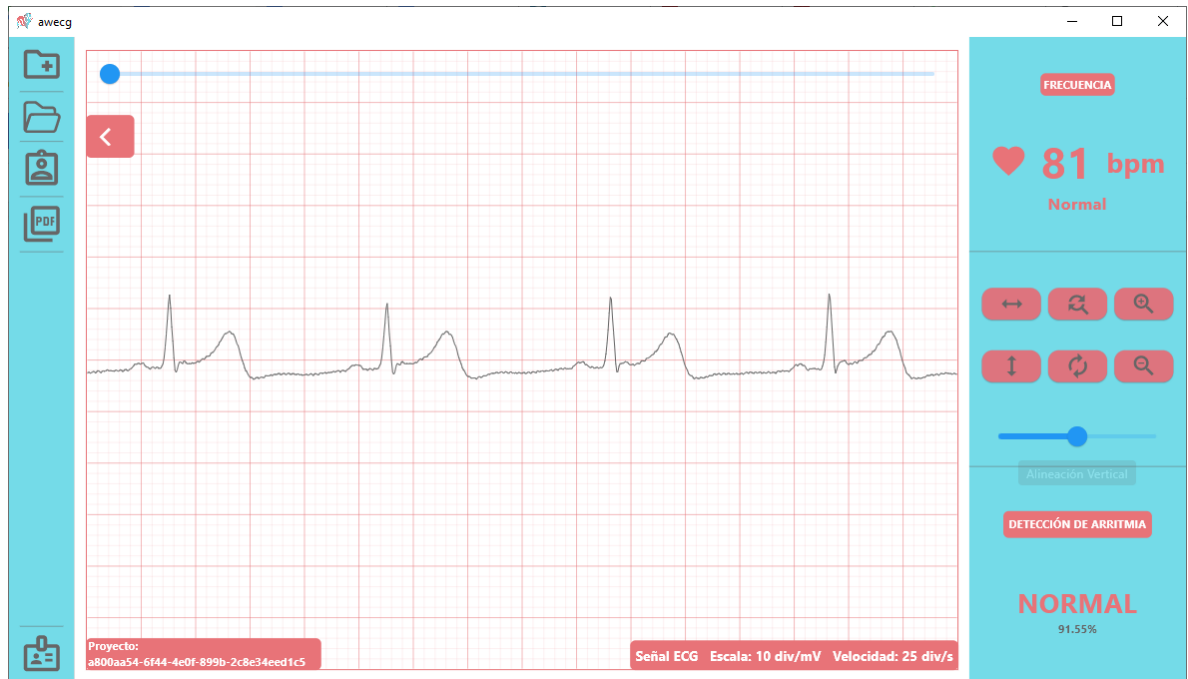


Figura 4.7. Pantalla proyecto cargado

4.1.6. Mediciones en la gráfica En la gráfica existe una herramienta (Figura 4.8) para realizar mediciones, debe clickear en algún punto de la gráfica para establecer el primer punto de medición, seguido, se clickea el segundo, así volviendo a pasar el cursor por alguno de esos dos puntos, tendremos la diferencia de voltaje y tiempo que hay entre ellos. Para reiniciar la herramienta de medición se debe clickear en alguna otra parte de la gráfica.

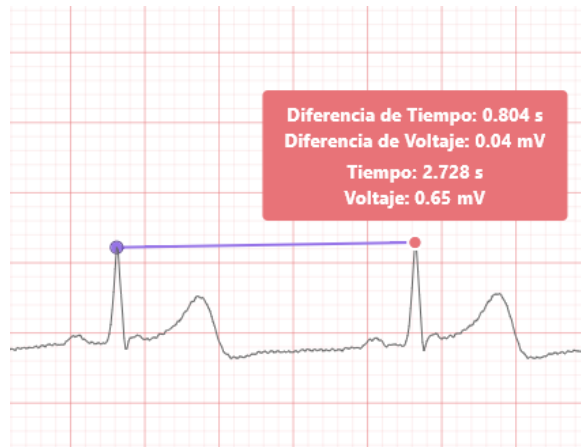


Figura 4.8. Herramienta de medición

4.1.7. Exportar como PDF Después de haber cargado un proyecto, es posible exportar dicho proyecto como pdf donde tendrá como primera página un formato con la información del paciente y del profesional médico que realizó el examen, junto con un espacio para la firma del profesional médico. En las páginas restantes, se tendrá la señal ECG capturada. El PDF generado se guardará en la carpeta del proyecto con el nombre del proyecto, si ya existe el archivo, este será sobrescrito.



		AWECG Información Médica		Fecha: 25/1/2023 4:4
Información del Paciente				
Nombre:	Alejandro Pulido Saravia	Edad: 25	Teléfono: 3508977722	
Dirección:	Calle 17 # 26-65	Identificación:	cc 1098798329	
Email:	pulido_saravia@hotmail.com			
Información del Profesional Médico				
Nombre:	Alejandro Pulido Saravia	Especialidad:	Estudiante de Ingeniería Electronica	
Email:	pulido_saravia@hotmail.com	Teléfono:	3508977722	
Dirección:	Calle 17 # 26-65 apto 201	Identificación:	cc 1098798329	
Lugar:	Casa			
<hr/> Alejandro Pulido Saravia Estudiante de Ingeniería Electronica				
				

Figura 4.9. Formato página 1 del PDF generado

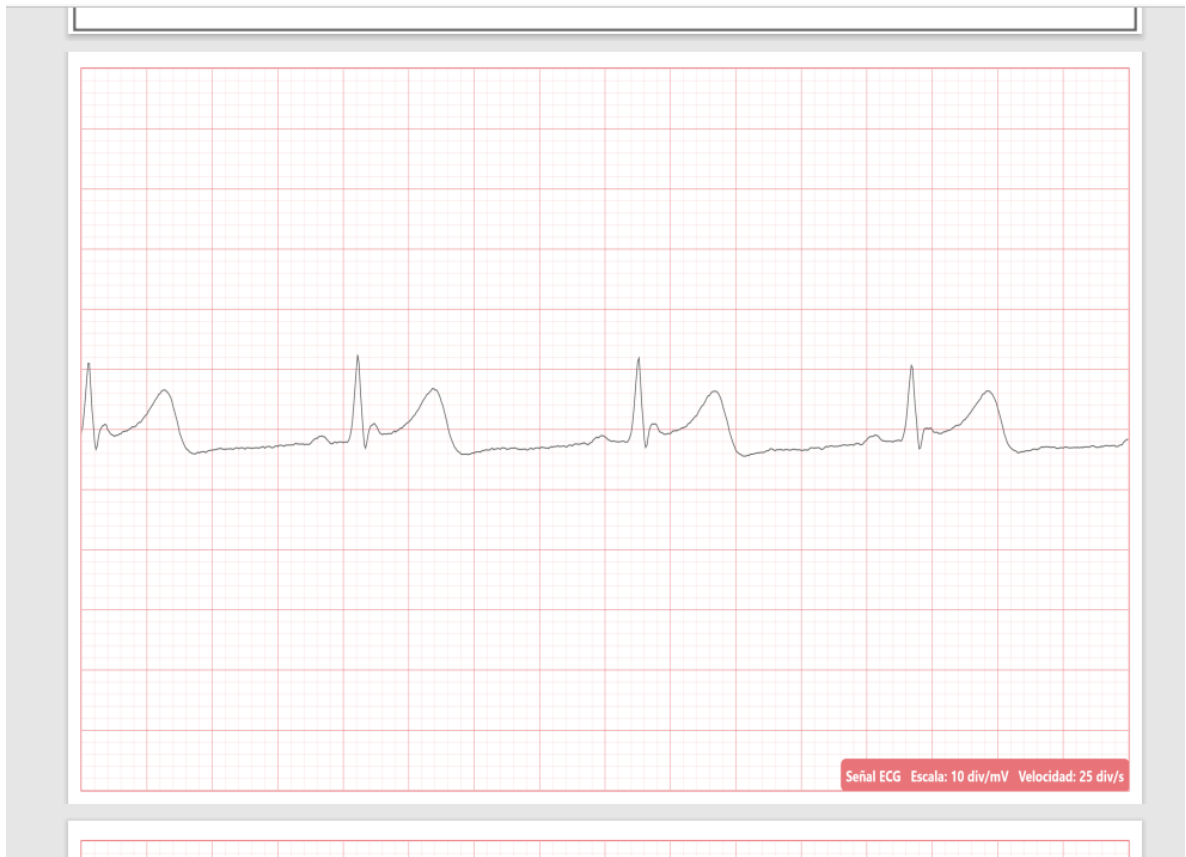


Figura 4.10. Formato de las páginas restantes del PDF generado

4.1.8. BLE configuración de conexión Para crear la conexión con el microcontrolador y el dispositivo, el dispositivo (Windows o Android) es el esclavo. El maestro (el microcontrolador) debe tener un GATT de la siguiente forma:

1. Servicio UUID = 832a0638-67db-11ed-9022-0242ac120002
2. Característica UUID = 00002b18-0000-1000-8000-00805f9b34fb .
3. Un descriptor 0x2902 . Debe ser notificable

4.2. APLICACIÓN PARA ANDROID

También se desarrolló una aplicación para Android donde se mantuvo todo igual que la versión para Windows aunque cuenta con algunas pantallas adicionales.

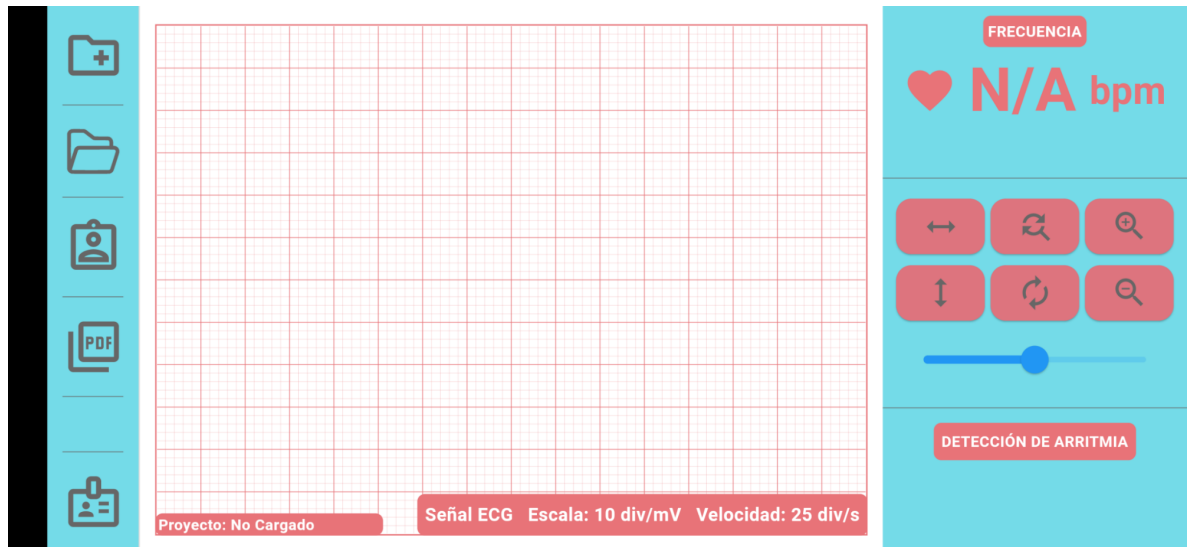


Figura 4.11. Pantalla principal de la versión para Android

4.2.1. Permisos Al crear un proyecto nuevo, se pedirá el permiso de Ubicación y/o Bluetooth dependiendo de la versión de Android; también al crear o abrir un proyecto, se pedirá permisos de almacenamiento y acceso al almacenamiento para poder gestionar los archivos del proyecto.

4.2.2. Compatibilidad Con el objetivo de asegurar la compatibilidad de la aplicación con la amplia gama de dispositivos Android existentes en el mercado, se desarrollaron dos versiones distintas: una para aquellos dispositivos con sistemas operativos desde Android 5.0 hasta Android 11, y otra específica para dispositivos con sistemas operativos a partir de Android 12.

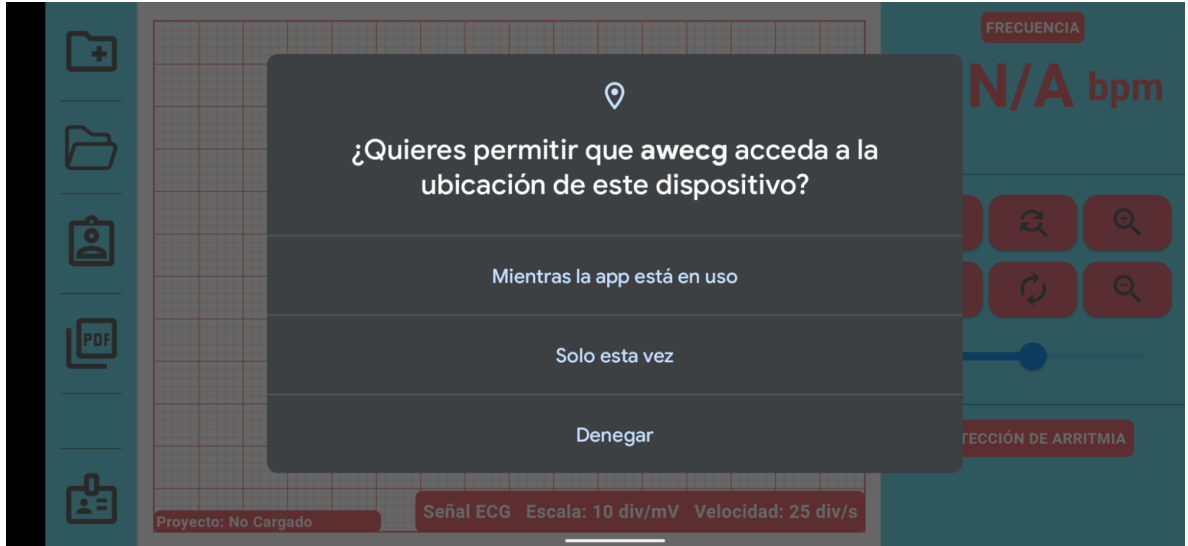


Figura 4.12. Permiso de ubicación

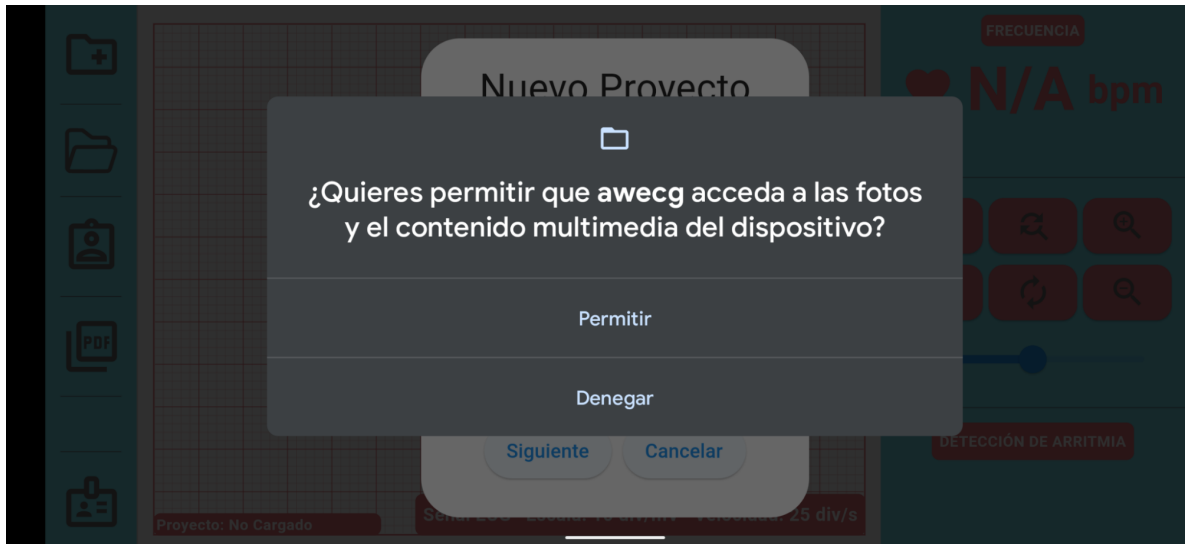


Figura 4.13. Permiso de almacenamiento

5. CONCLUSIONES

Se ha desarrollado una aplicación que permite detectar la arritmia fibrilación auricular al analizar señales ECG, obtenidas mediante la ayuda de un microcontrolador y conexión Bluetooth.

La aplicación se encuentra disponible para dispositivos Windows y Android, gracias a esto, la aplicación puede llegar a ser accesible desde cualquier lugar.

Tras llevar a cabo pruebas y testeos con usuarios, se ha comprobado que la herramienta es fácil de usar y comprender.

La compatibilidad con microcontroladores y Bluetooth Low-Energy (BLE) abre la posibilidad de desarrollar en el futuro un hardware portátil de monitoreo ECG que optimice aún más las capacidades de detección de la arritmia.

BIBLIOGRAFIA

Abadi, Martín et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015 (vid. págs. 16, 24).

Angelov, Felix (vid. pág. 18).

Baseflow. *Permission_handler: Flutter Package*. 2022 (vid. pág. 22).

Bluetooth technology overview (vid. págs. 21, 27).

Coast, D.A., R.M. Stern, G.G. Cano y S.A. Briller. “An approach to cardiac arrhythmia analysis using Hidden Markov models”. En: *IEEE Transactions on Biomedical Engineering* 37.9 (1990), 826–836. DOI: 10.1109/10.58593 (vid. pág. 11).

Devices, Analog. *AD8232*. 2019 (vid. pág. 27).

espressif. *ESP32WROOM32E ESP32WROOM32UE*. Espressif Systems. 2020 (vid. pág. 27).

Fajardo, Carlos A. *Cafajar-uis/ECG_Companded_Model* (vid. págs. 14, 16).

Flutter. *Build apps for any screen* (vid. págs. 18, 27).

Fonseca, Karen, Sergio Osorio, Jeyson Castillo y Carlos Fajardo. “Contrastive Learning for atrial fibrillation detection in challenging scenarios”. En: *2022 30th European Signal Processing Conference (EUSIPCO)* (2022). DOI: 10.23919/eusipco55093.2022.9909842 (vid. págs. 11, 14, 16, 28, 31).

Fuster, Valetín. “Aproximación terapéutica a la epidemia de fibrilación auricular”. En: *Revista Española de Cardiología* 55 (2002), págs. 27-32 (vid. pág. 10).

Garg, Amish. *Tflite_flutter: Flutter Package*. 2021 (vid. pág. 24).

Gutierrez, Lorenzo. *Beyond the hype: Medical Device Artificial Intelligence (AI)*. 2022 (vid. pág. 11).

Hannun, Awni Y. et al. “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network”. En: *Nature Medicine* 25.1 (2019), 65–69. DOI: 10.1038/s41591-018-0268-3 (vid. pág. 11).

Introduction to widgets (vid. págs. 18, 22).

Kael'thas. *Fl_chart: Flutter Package*. 2023 (vid. pág. 19).

Medeiros, André. *The introduction to reactive programming you've been missing*. 2014 (vid. pág. 19).

Morales, Kevin. *Arquitecturas de software en Android: MVC, MVP Y MVVM*. 2019 (vid. págs. 19, 23).

Pons, Ferran et al. “Mortalidad y Causas de muerte en pacientes con insuficiencia cardiaca: Experiencia De Una Unidad especializada multidisciplinaria”. En: *Revista Española de Cardiología* 63.3 (2010), 303–314. DOI: 10.1016/s0300-8932(10)70089-0 (vid. pág. 10).

Pulido, Alejandro. *Alpusa/AWEKG: Flutter ECG application to Windows and Android*. 2023 (vid. pág. 26).

Romero, Martín y Diana Chávez. “Carga de Enfermedad atribuible a fibrilación auricular en Colombia (2000-2009)”. En: *Revista Colombiana de Cardiología* 21.6 (2014), 374–381. DOI: 10.1016/j.rccar.2014.08.006 (vid. pág. 10).

Ruivo, Miguel. *File_picker: Flutter Package*. 2023 (vid. pág. 20).

Thakor, Nitish V., John G. Webster y Willis J. Tompkins. “Estimation of QRS Complex Power Spectra for design of a QRS filter”. En: *IEEE Transactions on Biomedical Engineering* BME-31.11 (1984), 702–706. DOI: 10.1109/tbme.1984.325393 (vid. pág. 25).

Townsend, Kevin. *Introduction to bluetooth low energy*. 2014 (vid. pág. 21).

Worner, Fernando, Alberto San Román, Pedro Luis Sánchez, Ana Viana Tejedor y José Ramón González-Juanatey. “Atención a los pacientes con Enfermedades Cardíacas Agudas y Críticas. Posición de la sociedad española de cardiología”. En: *Revista Española de Cardiología* 69.3 (2016), 239–242. DOI: 10.1016/j.recesp.2015.07.018 (vid. pág. 10).