UAEM

Universidad Autónoma del Estado de México

Centro Universitario UAEM Atlacomulco

Proyecto de Investigación:

Sistema de adquisición de señales electrocardiográficas para su clasificación utilizando Redes Neuronales Recurrentes

Presentan:

Alejandro Casiano Angeles, Esmeralda Contreras Piña, Adrián Correa Ramos, María del Carmen Huitrón Domingo

Programa Educativo:

Licenciatura en Ingeniería en Computación

Unidad de Aprendizaje:

Gestión de Proyectos de Investigación

Docente:

Dr. Everardo Efrén Granda Gutiérrez

Mayo de 2023

Resumen

El proyecto de adquisición y clasificación de señales electrocardiográficas utilizando un sensor AD8232 y una red neuronal recurrente es una idea original que se enfoca en la obtención de señales electrocardiográficas mediante sensores para su clasificación en Normal y Anormal. Para garantizar la validez y confiabilidad del proyecto, se han establecido métricas de evaluación como la **exactitud**, **sensibilidad**, **especificidad** y **clasificación errónea** de la red neuronal. Estas métricas son esenciales para evaluar la eficacia del modelo de predicción y asegurar que las decisiones cuentan con precisión y confiabilidad. La aportación del proyecto a la disciplina de Ingeniería en Computación se centra en el sistema embebido de obtención de señales y su tratamiento, lo que permite una integración natural con la red neuronal.

Además, el uso de herramientas especializadas como MATLAB para el desarrollo de la red neuronal representa una innovación en la aplicación de la inteligencia artificial y el machine learning en la detección temprana de enfermedades cardíacas.

El proyecto se enfoca en áreas como sistemas embebidos e inteligencia artificial, lo que representa una oportunidad para explorar nuevas aplicaciones de la tecnología en el campo de la ingeniería. En general, el proyecto aspira poder mejorar la obtención de señales para su posterior clasificación, lo que podría tener un impacto significativo en la contribución al conocimiento.

Palabras clave: Señales electrocardiográficas, Aprendizaje Automático, Sistema Embebido, Clasificación, Inteligencia Artificial.

Abstract

The project for the acquisition and classification of electrocardiographic signals using an AD8232 sensor and a recurrent neural network is an original idea that focuses on obtaining electrocardiographic signals through sensors for their classification into Normal and Abnormal. To guarantee the validity and reliability of the project, evaluation metrics have been established such as accuracy, sensitivity, specificity and neural network misclassification.

These metrics are essential for evaluating the effectiveness of the forecasting model and ensuring that decisions are accurate and reliable. The contribution of the project to the discipline of Computer Engineering focuses on the embedded system for obtaining signals and their treatment, which allows a natural integration with the neural network. In addition, the use of specialized tools such as MATLAB for the development of the neural network represents an innovation in the application of artificial intelligence and machine learning in the early detection of cardiac diseases. The project focuses on areas such as embedded systems and artificial intelligence, which represents an opportunity to explore new applications of the technology in the field of engineering. In general, the project aspires to be able to improve the obtaining of signals for their subsequent classification, which could have a significant impact on the contribution to knowledge.

Keywords: Electrocardiographic signals, Machine Learning, Embedded System, Predictions, Artificial Inteligence

Índice general

Capítulo 1

Introducción

El presente capítulo tiene como propósito abordar la problemática a resolver, se desarrollan los antecedentes, planteamiento del problema mediante la formulación de preguntas, objetivos e hipótesis, así mismo, se añade la relevancia de dicho enfoque que se pretende luego de realizar una investigación en diversas fuentes confiables (artículos, tesis, patentes y libros) abordando la justificación y sus impactos, finalmente se presenta el alcance y limitaciones, declarando los requerimientos, restricciones y magnitud del proyecto.

1.1. Antecedentes y contexto del problema

Las enfermedades cardiovasculares son perturbaciones del corazón y de los vasos sanguíneos(Gonzélez Medina, 2022). Desde hace un tiempo dichas enfermedades han mostrado un crecimiento en México radical provocando grandes cantidades de defunciones por tal motivo se pretende la creación de un sistema embebido mediante el cual se obtengan señales cardíacas y posteriormente estas puedan ser analizadas con algún algoritmo de IA previamente entrenado para poder predecir este tipo de enfermedades y contribuir a la prevención y un tratamiento efectivo.

Son escasos los proyectos desarrollados con este enfoque, creación de sistema embebido para analizar señales ECG. Uno de estos, es un microsistema electrónico económico que permite adquirir señales ECG nombrado CARDIOCEL, de forma remota para transmitirlas mediante línea telefónica y ser almacenadas, procesadas y proyectadas en un computador personal. Se tiene como propuesta de mejora de visualización más de una señal y cambiar el tipo de comunicación inalámbrica (Rodríguez et al., 2023). Otra propuesta destacada es el diseño e implementación de un sistema de monitoreo de electrocardiograma (ECG) de bajo costo utilizando Arduino y smartphone (Prabu y Ravikumar, 2016). A su vez (Gonzélez Medina, 2022) hace uso de modelos predictivos para evaluar las enfermedades cardiovasculares más frecuentes en el país.

En cuanto a la creación de software con la misma finalidad se encuentran trabajos precedentes, como el desarrollo de una aplicación Android con conexión inalámbrica, con la necesidad de implementar algún algoritmo de inteligencia artificial para analizar las señales biomédicas (Smith et al., 2022). Así mismo se encuentra la adquisición de variables físicas en el software Soft V8.3 empleando un generador de señales enviadas a un Controlador Lógico Programable y un sensor PT100 (Ortega Ordoñez et al., 2023).

Cabe mencionar que ya se han implementado modelos orientados únicamente en la aplicación de métodos de inteligencia artificial o en la adquisición de señales cardíacas, uno de estos es el análisis de las técnicas más populares de machine learning aplicadas a la clasificación y predicción de enfermedades cardiovasculares (Jennifer S. et al., 2021), también se ha analizado el impacto de la inteligencia artificial en el área de cardiología específicamente en la identificación de patrones del ser humano posibles de detectar (Dorado Díaz et al., 2019). Además ya se han aplicado algunas métricas para su evaluación utilizando herramientas como WEKA y SPSS (Abuhaija et al., 2023). De igual forma se destaca la aplicación de una red neuronal convolucional profunda (CNN) para clasificar cuatro tipos de arritmias cardíacas en señales de ECG, y comparar el rendimiento del modelo propuesto con otros enfoques de clasificación de arritmias (Rajkumar et al., 2019).

De manera similar tanto (Palma et al., 2020) como (Vargas-Cañas et al., 2021) puntualizan el uso de sensores ya sea inalámbricos o no para la obtención de señales para su posterior filtrado, algunas de las técnicas de filtrado empleadas son filtros pasa bajas (FPB) o Transformada de Wavelet.

Finalmente, (Ahamed et al., 2015) muestra la capacidad de las placas Arduino para la obtención se señales para su posterior envío a una aplicación móvil en donde se realiza el filtrado de estas, para entonces visualizarlas en tiempo real.

1.2. Planteamiento del problema

1.2.1. Definición del problema

La problemática de este proyecto consiste en desarrollar un sistema que permita capturar señales de electrocardiograma (ECG) utilizando electrodos y Arduino, así mismo analizarlas para detectar posibles anomalías mediante el uso de técnicas de machine learning. La detección temprana de anormalidades en el ECG es importante para el diagnóstico y tratamiento de enfermedades cardíacas, lo que puede mejorar la calidad de vida de los pacientes y salvar vidas. En la actualidad, existen varias técnicas para la detección de irregularidades en el ECG, como el análisis de la frecuencia cardíaca y la detección de arritmias. Sin embargo, estos métodos pueden tener limitaciones en términos de precisión y eficacia. La utilización de técnicas de machine learning, específicamente redes neuronales recurrentes(RNN) puede permitir la identificación de patrones complejos en las señales de ECG, lo que puede mejorar la precisión y la eficacia de la detección de dichas anomalías.

La brecha entre el conocimiento actual y el nuevo se encuentra en la implementación de un sistema integrado que permita la captura de señales de ECG mediante electrodos y Arduino, y el posterior análisis de estas señales mediante técnicas de machine learning para detectar afecciones cardíacas. Además, se busca la optimización de los algoritmos de machine learning para lograr una detección más precisa y eficaz. La situación actual es que existen sistemas comerciales para la detección de anormalidades en el ECG, pero estos pueden ser costosos y no siempre están disponibles en todas las regiones. Por lo tanto, la implementación de un sistema accesible y de bajo costo que pueda detectar irregularidades en el ECG es un desafío importante. La solución propuesta busca mejorar la disponibilidad y accesibilidad de la localización de anomalías en el ECG mediante la utilización de técnicas de ingeniería en computación y machine learning. Se han realizado varios estudios en el ámbito de la detección de posibles patrones irregulares en señales electrocardiográficas utilizando técnicas

de machine learning, como el uso de redes neuronales artificiales y algoritmos de clasificación. Sin embargo, aún existen desafíos en cuanto a la optimización de los algoritmos para mejorar la precisión y eficacia de la detección. Este proyecto busca contribuir a la investigación en este campo y desarrollar soluciones innovadoras para detectar anomalías en el ECG utilizando técnicas de ingeniería en computación y machine learning.

1.2.2. Preguntas de investigación

Pregunta central

¿Puede el uso de sensores ECG y la placa de desarrollo Arduino Uno combinados con técnicas de aprendizaje automático mejorar la precisión en la clasificación de patrones en enfermedades cardíacas?

- ¿Cómo puede la integración de sensores ECG y la placa de desarrollo Arduino uno, en combinación con técnicas de aprendizaje automático, mejorar la precisión en la detección de patrones de anomalías cardíacas?
- ¿Cuál es el mejor enfoque de análisis de señales ECG para maximizar la precisión de la clasificación señales electrocardiográficas?
- ¿Cómo se pueden optimizar los parámetros de las técnicas de aprendizaje automático para mejorar la sensibilidad de la detección de patrones anormales en las señales ECG capturadas?
- ¿Como se pueden procesar señales electrocardiográficas adquiridas con el sistema propuesto para encontrar similitudes con las adquiridas de fuentes como repositorios medicos?

1.2.3. Objetivos

Objetivo general

Desarrollar un sistema de adquisición de señales ECG usando sensores para electrocardiograma con Arduino para clasificar los patrones que presenten dichas señales obtenidas utilizando técnicas de Machine Learning considerando como variables la frecuencia de muestreo, la duración del registro de la señal y la ubicación de los sensores.

Objetivos específicos

- Implementar métodos en el microcontrolador Arduino para adquirir y almacenar las señales ECG obtenidas.
- Desarrollar algoritmos de procesamiento de señales ECG para extraer características relevantes.
- Aplicar técnicas de machine learning para clasificar las señales ECG según patrones de normalidad.
- Analizar registros de señales ECG adquiridas de fuentes como repositorios médicos.
- Validar la precisión y confiabilidad del sistema desarrollado mediante pruebas con datos de señales ECG reales.

1.2.4. Meta de ingeniería

- Meta de Ingeniería
  Diseñar y desarrollar un sistema de adquisición de señales ECG utilizando sensores y la placa de desarrollo Arduino Uno, que sea capaz de capturar señales eléctricas del corazón de alta calidad y transferirlas a un sistema de procesamiento y análisis de datos. Posteriormente, utilizar técnicas de aprendizaje automático para analizar las señales electrocardiográficas y mejorar la precisión en la detección de patrones anormales en comparación con los métodos tradicionales. Finalmente, optimizar los parámetros del sistema de aprendizaje automático para mejorar aún más la precisión y velocidad de detección de patrones anormales en las señales.

1.3. Relevancia de la investigación

1.3.1. Justificación

La aplicación de los sensores de ECG son una herramienta no invasiva y relativamente económica para obtener información sobre la actividad eléctrica del corazón, lo que permite identificar patrones y anomalías en dichas señales. Por otro lado, los algoritmos de Machine Learning(ML) son una herramienta cada vez más potente en el campo de la medicina, debido a su capacidad para analizar grandes cantidades de datos, encontrar patrones y relaciones que pueden ser difíciles de detectar empleando métodos tradicionales. Definitivamente la implementación de algoritmos a las señales ECG adquiridas puede ayudar a identificar anomalías presentes en la señal de forma automatizada, teniendo como potencial ventaja el diagnóstico más temprano.

1.3.2. Impactos

- Tecnológico.
  Adquisición de señales ECG con sensores de monitoreo para su posterior clasificación mediante la aplicación de algoritmos de inteligencia artificial, por lo que este proyecto pretende impulsar el desarrollo de tecnología innovadora en el campo de la ingeniería en computación. Además, la investigación en este ámbito podría tener implicaciones en otras áreas, como el internet de las cosas.
- Social.
  Las enfermedades cardiovasculares son una de las principales causas de mortalidad en todo el mundo, afectando a personas de todas las edades y orígenes. Un proyecto que tenga como objetivo mejorar la prevención y el tratamiento de estas enfermedades tendría un impacto social significativo al mejorar la calidad de vida de las personas y reducir la carga en los sistemas de salud.
- Económico.
  La creación del sistema embebido pretender ser una herramienta tecnológica de bajo costo.

1.4. Alcance y limitaciones

1.4.1. Declaración del alcance del proyecto

Desarrollar un sistema embebido automatizado de monitoreo y análisis de señales cardiovasculares utilizando sensores de electrocardiograma y un microcontrolador Arduino, estos componentes facilitarán las capturas de las señales electrónicas del corazón, para posteriormente procesarlas y aplicar un análisis automático implementando el uso de las técnicas de machine learning. Dicho proyecto se pretende elaborar en un periodo comprendido de un mes.

1.4.2. Requerimientos del proyecto

La tabla 1.1 muestra los requerimientos con los que se cuenta para realizar el proyecto.

Tabla 1.1: Requerimientos Disponibles

Requerimiento	Descripción
Arduino IDE	Es una aplicación de software que le permite escribir, cargar y depurar el código que se ejecutara en la placa Arduino. Debido a su fácil implementación de algoritmos, se pretende utilizar esta herramienta para desarrollar algoritmos de procesamiento de señales.
MATLAB	Es un software de cálculo numérico y análisis de datos desarrollado por la empresa MathWorks. MathWorks ofrece diferentes opciones de licencia para MATLAB, en este caso en particular se cuenta con la licencia Académica. Gracias a sus múltiples herramientas en el área de procesamiento de señales, así como el área de Inteligencia Artificial, se pretende usar MATLABpara realizar el entrenamiento y validación de la red neuronal.
Equipo de cómputo	Se encargará de almacenar tanto las señales adquiridas por el sensor, como los algoritmos necesarios para el desarrollo de la red neuronal, dicho equipo consta de las siguientes características: • PROCESADOR: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz MEMORIA RAM: 12.0 GB SISTEMA OPERATIVO: WINDOWS 11
Datasets	Tanto para entrenamiento como validación, se utilizará el dataset de tipo ECG Heartbeat Categorization Dataset que está compuesto de dos colecciones de señales de latidos cardíacos derivados de dos datasets famosos, el primer dataset se enfoca en Arritmia MIT-BIH y el segundo dataset de diagnóstico de ECG de PTB.
Tiempo	El proyecto se pretende concluir en un periodo comprendido de un mes.

Financiamiento	Al momento de plantear los requerimientos del proyecto, se estableció un presupuesto aproximado de $1200.00 pesos, el cual se obtuvo de la sumatoria de los precios el desglose de esta estimación se muestra en la tabla 1.0.

De la misma forma la tabla 1.2 muestra los requerimientos por adquirir necesarios para elaborar el proyecto.

Tabla 1.2: Requerimientos por adquirir para el desarrollo del proyecto.

Requerimiento	Descripción	Cantidad	Costo
Amplificador AD8232	Es un amplificador de instrumentación utilizado en la medición de señales electrónicas del corazón.	1	$ 200.00
Arduino UNO	Es una placa de desarrollo basada en el microcontrolador ATmega 328P, que cuenta con un procesador AVR de 8 bits y una velocidad de reloj de 16 MHz. la placa UNO cuenta con una arquitectura AVR	1	$500.00
Electrodos Meditrace serie 200	Son electrodos de grado médico utilizados en la medición de actividad eléctrica del corazón, sobre todo en pruebas de electrocardiograma. Algunas de sus caracteristicas son: - La facilidad de adherirse a la piel del paciente. - Proporciona una señal electrica clara y precisa para el monitoreo de la actividad cardiaca.	1	$500.00
	TOTAL		**$1200.00**

1.4.3. Limitantes

Existen diversas limitaciones que pueden presentarse en el desarrollo del proyecto. En primera instancia los datos, tanto de entrenamiento como de validación, pues su disponibilidad y relevancia constituye un factor critico al momento de desarrollar el proyecto. Adicionalmente, el tratamiento efectivo de las señales adquiridas puede representar un factor limitante, pues para filtrar el ruido de la señal captada es necesario emplear técnicas de filtrado de señales. En caso de presentarse alguna limitante con relación a la relevancia de los datos se pretende indagar en bases de datos en clínicas internacionales. En el caso de presentarse una limitante con respecto a las señales y el ruido que estas puedan presentar, se optaría por implementar otro tipo de filtros para su limpieza correcta.

Plan de Riesgos

En caso de presentarse las limitantes antes mencionadas, se pretende poner en marcha el siguiente plan de riesgos, el cual contempla dos criterios.

- Aplicación del 70 % de total de los datos del dataset a entrenamiento y 30 % a la validación.
- Aplicar técnicas de filtrado para señales ECG, como Filtro Pasa Banda o Transformada Rápida de Fourier.

Capítulo 2

Revisión de la Literatura

2.1. Propósito

El propósito de la siguiente revisión de la literatura es identificar y examinar las técnicas, métodos, así como las herramientas empleadas en la adquisición y procesamiento de señales ECG. De la misma forma reconocer tanto los avances, como las limitaciones que las técnicas de machine learning pueden presentar en la detección de patrones en señales electrocardiográficas, pero sobre todo hacer énfasis en las oportunidades e inconvenientes al desarrollar el sistema haciendo uso de tarjetas de adquisición de datos como lo es la placa Arduino DUE y el módulo amplificador AD8232.

2.2. Selección de la Literatura

Los criterios para la identificación de la literatura están basados en la jerarquía de fuentes de información en donde se encuentran las primarias, estas comprenden artículos de investigación. Secundarias, son aquellas que recopilan información de las fuentes anteriores en estas podemos encontrar libros de texto, informes y artículos de revista. Finalmente, las terciarias son principalmente compilaciones, en estas se encuentran enciclopedias y diccionarios.

Para la elaboración de este proyecto solo se consideraron fuentes de información primarias y secundarias estableciendo la siguiente jerarquía. En primer lugar, libros, artículos de investigación y patentes, los cuales deben contar con las siguientes características:

1. Actual, es decir con una longevidad no mayor a 5 años.

2. En el caso de los artículos de investigación deben de haber pasado por el proceso de aceptación mismo que comprende la revisión por pares, haber sido corregido y finalmente ser publicado.

2.3. Análisis de la Literatura

2.3.1. El corazón

De acuerdo con Mangne Machaca (2009, p. 36) El corazón es el músculo más importante del cuerpo humano, el cual esta formado por **Aurículas**, tanto izquierda como derecha, **Ventrículos** izquierda y derecha y **fibras musculares** que de especializan en excitación y conducción.

La imagen 2.1 muestra las partes antes mencionadas.

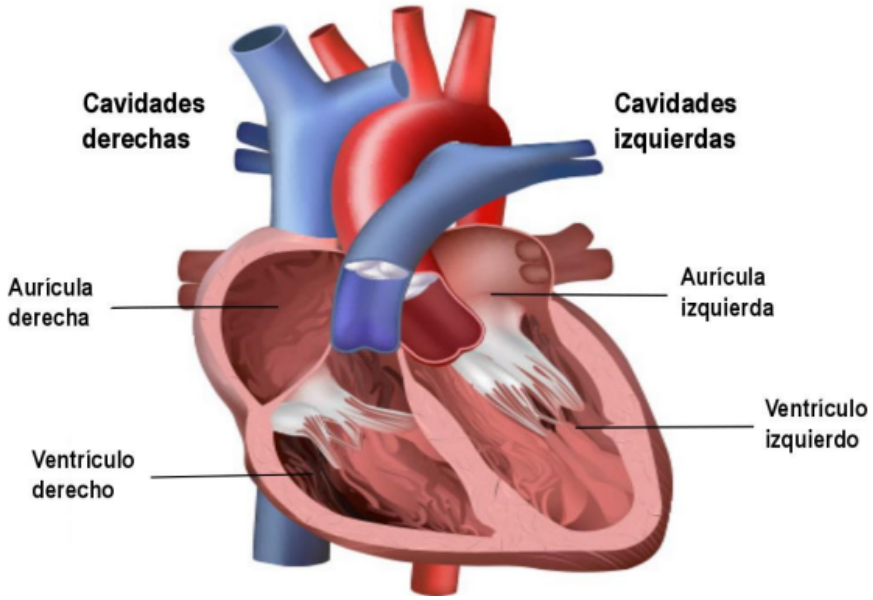


Figura 2.1: Anatomía del Corazón (González Cervantes et al., 2016)

2.3.2. Electrocardiograma

Existen diversas formas de monitorear la actividad física del corazón, por ejemplo, Rodríguez et al. (2023, p. 1) menciona que "El ECG es una representación gráfica de la actividad bioeléctrica del músculo cardíaco, por lo que un equipo de registro de ECG es prácticamente un voltímetro que realiza una función de registrador."
En otras palabras un Electrocardiograma o ECG es un tipo de prueba médica empleada para evaluar el ritmo cardíaco y la actividad eléctrica del corazón.

Otra técnica similar al ECG es el Balistocardiograma, Palma et al. (2020, p. 1) lo define como "Un balistocardiograma (BCG) consiste en el registro del movimiento mecánico del corazón mediante la monitorización de la fuerza o aceleración desde el pecho o alternativamente tomando medidas remotas de la actividad de bombeo sanguíneo debido al latido cardíaco."
Como se puede apreciar, ambas formas comparten algunas características, por ejemplo, ambas son consideradas como pruebas **no invasivas**, sin embargo, existe una diferencia sustancial entre estas.

- El **ECG** mide la **Actividad eléctrica** del corazón.

- EL **BCG** mide las **vibraciones** producidas por la circulación sanguínea.

De acuerdo con Rajkumar et al. (2019, p. 365). Es posible diagnosticar varias anomalías en el corazón, ya que cualquier variación en la forma de onda, como aumento o disminución en los valores de de amplitud y periodo de tiempo, sumado a la ausencia de ondas características permite diagnosticar la presencia de alguna anomalía en el corazón.

La imagen 2.2 muestra una onda de electrocardiograma normal.

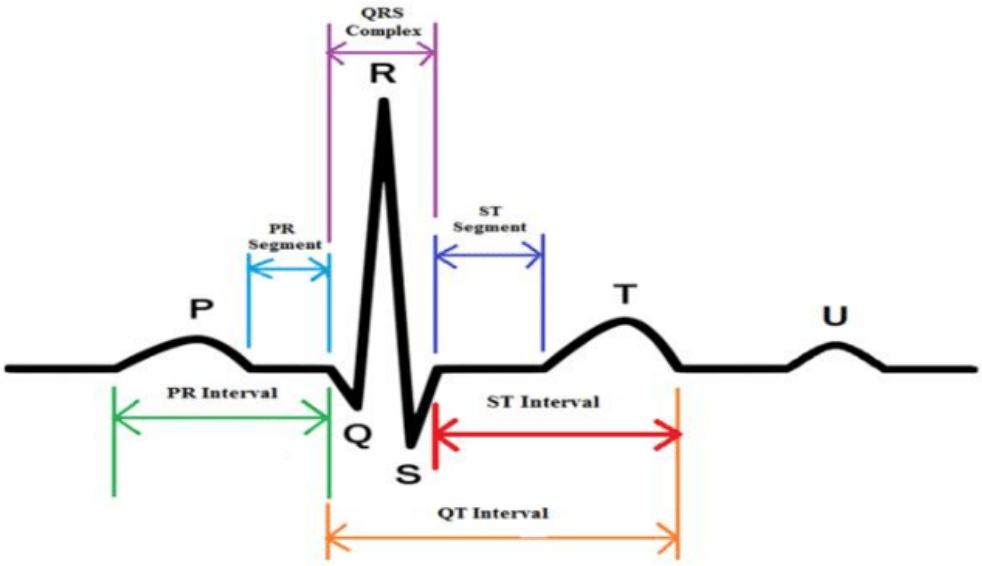


Figura 2.2: Forma de onda de ECG normal (Rajkumar et al., 2019)

Otro aporte relacionado con un electrocardiograma normal es mostrado por Mangne Machaca (2009, p. 38) el cual menciona que

> "Un electrocardiograma normal esta formado por una onda **P** un complejo **QRS** y una onda **T**. Con frecuencia, pero no siempre, el complejo **QRS** está formado por tres ondas separadas: La onda **Q**, la onda **R** y la onda **S**.
> La onda **P** está producida por los potenciales eléctricos que se generan cuando se **despolarizan** las aurículas antes del comienzo de la contracción auricular. El complejo **QRS** está formado por los potenciales que se generan cuando de despolarizan los **ventrículos** antes de su contracción, es decir, a medida que la onda de despolarización se propaga por lo ventrículos.
> Por lo tanto, tanto la onda **P** como los componentes del complejo **QRS** son las ondas de despolarización.
> La onda **T** está producida por los potenciales que se generan cuando los ventrículos se recuperan del estado de despolarización. Este proceso normalmente aparece en el músculo ventricular entre 0.25 y 0.35 segundos después de la despolarización y la onda **T** se conoce como onda de **repolarización.**"

2.3.3. Inteligencia Artificial

Hoy en dia este termino es ampliamente usado, sin embargo, tal y como menciona Dorado Díaz et al. (2019, p. 1066), "A día de hoy es difícil encontrar una definición universal de lo que se conoce como IA. A menudo se refiere al campo de las ciencias de la computación que trata de imitar los procesos cognitivos humanos, la capacidad de aprendizaje y el almacenamiento de conocimiento".

En otras palabras la Inteligencia Artificial (IA) consiste en algoritmos y modelos matemáticos que permiten a los sistemas y maquinas imitar elementos cognitivos como la memoria, el aprendizaje, el reconocimiento de patrones, etc. con el objetivo de tomar decisiones y resolver problemas complejos.

2.3.4. Machine Learning

También conocido como **Aprendizaje Automático** es un subconjunto de la inteligencia artificial en el que se diseñan sistemas capaces de aprender, en otras palabras, tiene la capacidad de reconocer patrones complejos para los humano, sin intervención de estos.

La imagen 2.3 muestra un diagrama de Venn donde se expone al Aprendizaje Automático (Machine Learning) como un subconjunto tanto de la Inteligencia Artificial, como de la ciencia de datos (Data Science).

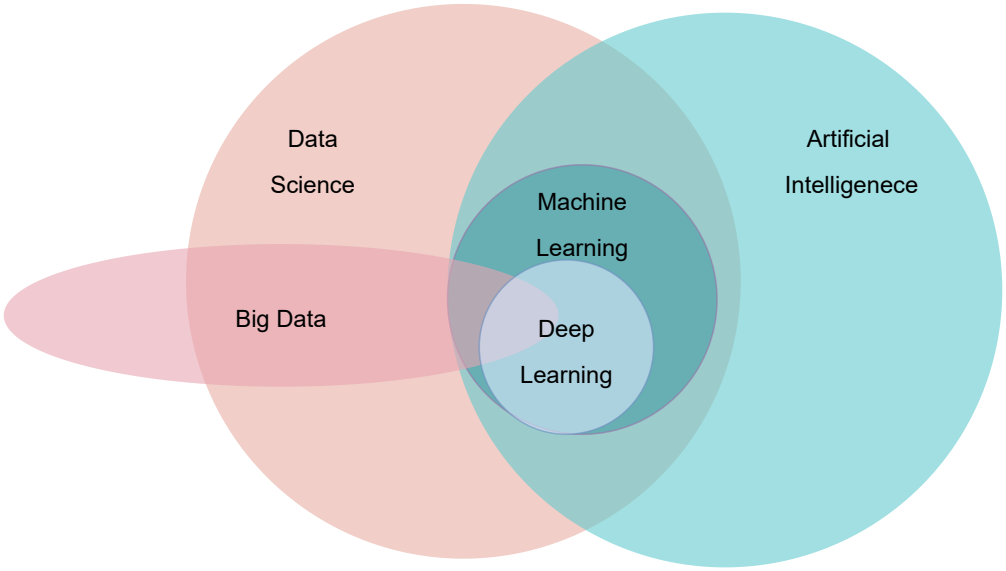


Figura 2.3: Diagrama de Venn con algunas tendencias actuales de la información

Tanto (Dorado Díaz et al., 2019) como (Abuhaija et al., 2023) y (Rajkumar et al., 2019) concuerdan en que las técnicas de machine learning se pueden clasificar en tres categorías; aprendizaje **supervisado**, el aprendizaje **no supervisado** y el aprendizaje por **refuerzo**, la tabla 2.1 expone las principales características de algunos tipos de aprendizaje.

Tabla 2.1: Cuadro comparativo de los tipos de aprendizaje

Tipo de Aprendizaje	Principales Cracteristicas
Aprendizaje Supervisado	- Es el tipo de aprendizaje más empleado, ademas de obtener los mejores resultados. - Se cuenta con un conjunto de datos etiquetados para entrenar el modelo. - Se emplea en problemas de clasificación y regresión. - Requiere de una gran cantidad de datos etiquetados.
Aprendizaje No Supervisado	- Se enfoca en encontrar patrones y estructuras en los datos de entrada. - Es ampliamente usado en problemas de clustering. - Requiere un menor cantidad de datos. - Suele ser más difícil de evaluar que el supervisado.
Aprendizaje por Refuerzo	- Intenta aprender a traves de la retroalimentación de su entorno. - Toma decisiones a parti de su estado actual. - Recibe un puntaje en función de la precisión de sus decisiones.

2.3.5. Técnicas de Machine Learning

Existen tantas técnicas de aprendizaje automático, que seria muy extenso explicar cada una, sin embargo (Dorado Díaz et al., 2019), (Abuhaija et al., 2023), (Jennifer S. et al., 2021) y (Rajkumar et al., 2019) concuerdan en que estas técnicas pueden ser empleadas en la detección de tanto arritmias como enfermedades cardiovasculares.

La tabla 2.2 esta basada en la mostrada por Dorado Díaz et al. (2019, p. 1068), expone las técnicas más populares de machine learning que se han usado en numerosas investigaciones.

Tabla 2.2: Descripción de las técnicas de machine learning más empleadas en cardiología.

Algoritmo	Descripción
Random Forest (RM)	Combinación de árboles predictores no correlacionados.
Gradient boosting	Combinación de árboles predictores escalonados.
Regresión logística (LR)	Análisis de regresión utilizado para predecir el resultado de una variable categórica.
Máquinas de vectores de soporte (SVM)	Clasificador a través de construcción de hiperplanos disociadores.
K vecinos más próximos (k-NN)	Estimación de la función de densidad de las variables predictoras en función de las clases
Análisis discriminate lineal	Generalización del discriminate lineal de Fisher
Clasificador bayesiano ingenuo	Clasificador probabilístico fundamentado en el teorema de Bayes

2.3.6. Redes Neuronales

Una definición de este concepto es proporcionado por Mangne Machaca (2009, p. 10) el cual menciona que son un modelo computacional que intenta reproducir el comportamiento del cerebro.

En otras palabras una red neuronal es un modelo matemático que toma como base el funcionamiento del cerebro humano. Está compuesta por múltiples unidades llamadas *neuronas*, las cuales están interconectadas y trabajan en conjunto para procesar información.

Neurona Artificial

Como ya se mencionó, la neurona artificial se puede considerar como la piedra angular de las redes neuronales, una definición más exacta es proporcionada por Mangne Machaca (2009, p. 13), quien menciona que

> "La neurona artificial pretende simular las características más importantes de la neurona biológica, partiendo de un elemento simple de cálculo (vector de entrada) procedente del exterior o de otras neuronas conectadas, posee un estado interno, llamado nivel de activación que le permite cambiar de estado y proporcionar una única respuesta o salida."

La figura 2.4 muestra el modelo matemático de una neurona artificial.

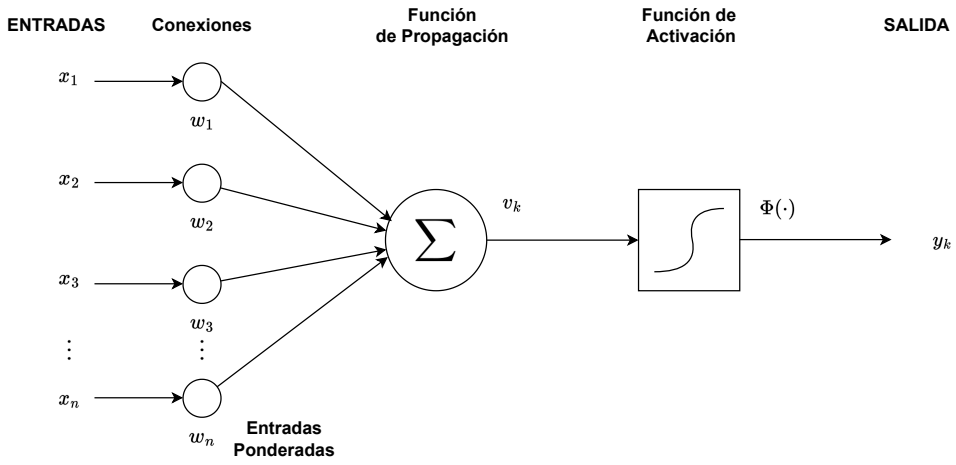


Figura 2.4: Modelo matemático de una neurona artificial

Como se puede apreciar en la figura 2.4 las neuronas de entrada están definidas por la expresión 2.1.

$$X = [x_1, x_2, x_3, \cdots, x_n] \tag{2.1}$$

Donde
X es el vector de entradas que corresponden a las señales sinápticas de una neurona biológica.

Los pesos sinápticos o entradas ponderadas son descritos en la expresión 2.2.

$$W = [w_1, w_2, w_3, \cdots, w_n] \tag{2.2}$$

Donde

W es un vector cuyos elementos representan la fuerza de una conexión sináptica.

La función de red o de **propagación** es el equivalente al cuerpo de la neurona, en esta se suman las **entradas** que han sido multiplicadas por los **pesos sinápticos**, dando como resultado una salida, misma que sera multiplicada por una función de **activación**, la cual produce una señal.

La expresión 2.3 muestra el modelo matemático descrito anteriormente

$$v_k = \sum_{j=1}^{n} w_j x_j \tag{2.3}$$

Donde
x_i son las entradas de la red.
w_j son los pesos sinápticos

Ademas de la definición, el autor menciona la existencia de tres tipos de neuronas de acurdo con su ubicación.

- **Neuronas de Entrada** las cuales reciben información del exterior, ademas de conformar la primera capa de la red o capa de **entrada**.
- **Neuronas Ocultas** estas conforman un capa intermedia que procesa la información que fluye a través de la red.
- **Neuronas de Salida** estas producen una respuesta después de que la información ha sido procesada por las neuronas ocultas.

Función de Propagación

Consiste en en transmitir la señal de entrada hacia adelante a través de las capas de la red hasta producir la salida. Mangne Machaca (2009, p. 12), la define como "La función de propagación o función de red convierte los valores de entrada en uno solo llamado típicamente el potencial que en la neurona biológica equivaldrá al total de las señales que le llegan a la neurona por sus dendritas." Esta se encuentra denotada como Net_j y es expresada en la expresión 2.4, la cual representa la suma ponderada de las entradas por los pesos sinápticos.

$$v_k = \sum (w_{ji} \times x_i(t)) \tag{2.4}$$

Donde
v_k es la función de propagación.
w_{ji} son los pesos sinápticos.
x_i son las entradas de la red.

Funciones de Activación

A diferencia de la función de propagación, la **función de activación** se utiliza para determinar la salida de una neurona en función de sus entradas. Aggarwal (2018, p. 25) menciona que "La elección de la función de activación es una parte fundamental del diseño de redes neuronales."

Tanto (Mangne Machaca, 2009, p. 12) como (Aggarwal, 2018, p. 26) concuerdan en que existen diversos tipos de funciones de activación los cuales serán descritos de forma individual.

- Función Lineal.

También conocida como función **identidad** es la forma más básica, la cual es utilizada comúnmente en los nodos de salida sobre todo cuando el objetivo es un valor real. La expresión 2.5 muestra una función de activación lineal.

$$\Phi(v) = v \tag{2.5}$$

Donde
$\Phi(v)$ es la función de activación.
v es la salida de la neurona sin modificaciones.

- Función Sigmoidal.

Este tipo de función es usada comúnmente en las capas ocultas de un red neuronal profunda.
La expresión 2.6 muestra la función de activación sigmoidal.

$$\Phi(v) = \frac{1}{1 + e^{-v}} \tag{2.6}$$

Donde
$\Phi(v)$ es la función de activación.
(v) es la entrada de la neurona.

- Función Gaussiana

Este tipo de funciones son utilizadas en redes donde las entradas y salidas son continuas y tienen una distribución normal.
La expresión 2.7 muestra la definición matemática de la función gaussiana.

$$\Phi(v) = e^{\left(-\frac{(v-c)^2}{2\sigma^2}\right)} \tag{2.7}$$

Donde
$\Phi(v)$ es la función de activación
v es la entrada de la neurona.
c es el centro de la curva.

σ controla el ancho de la curva.

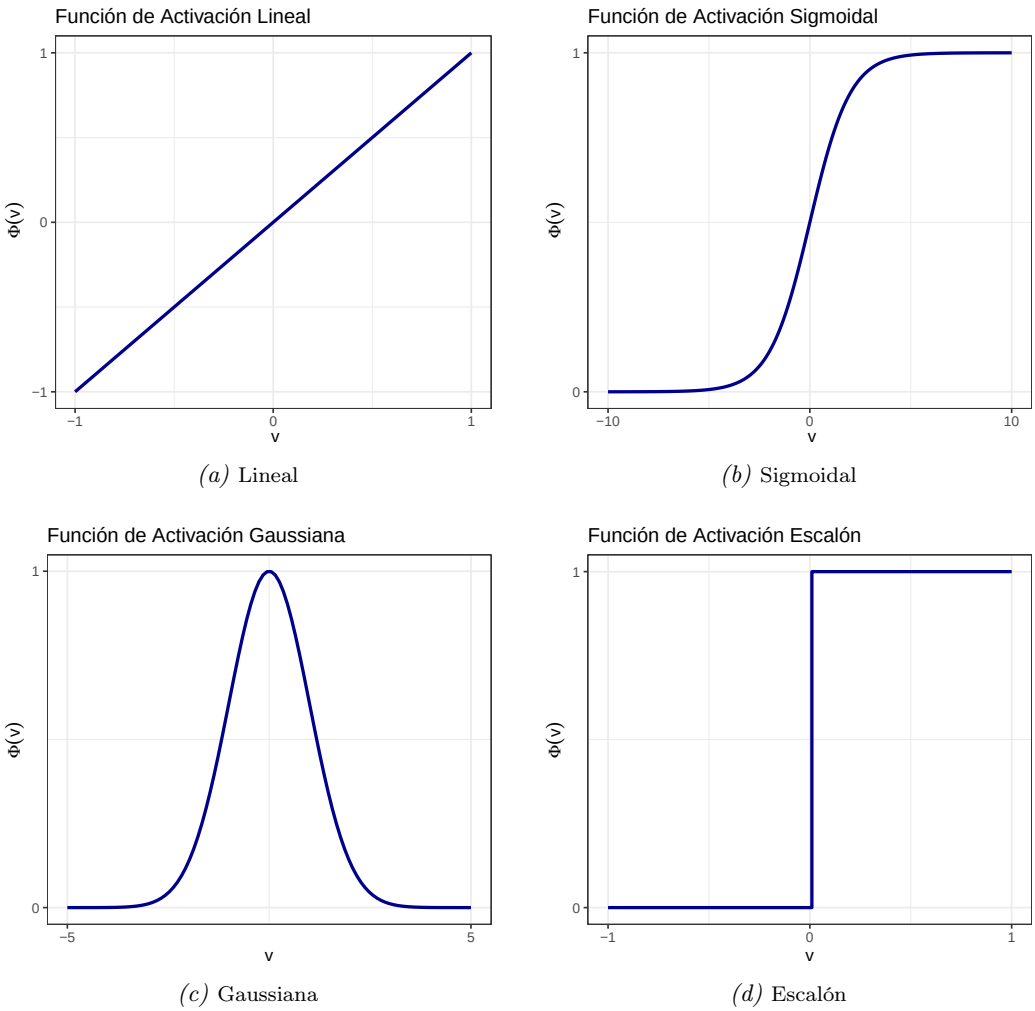
- Función Escalón

Este tipo de funciones son ampliamente utilizadas en redes neuronales binarias gracias a su simplicidad y a su no linealidad. La expresión 2.8 muestra el modelo matemático de una función escalón.

$$\Phi(v) = \begin{cases} 0, & \text{si } v < 0 \\ 1, & \text{si } v \geq 0 \end{cases} \tag{2.8}$$

Donde
$\Phi(v)$ es la función de activación.
v es la entrada de la neurona.

La figura 2.5 muestra las funciones de activación antes detalladas.



(a) Lineal *(b)* Sigmoidal *(c)* Gaussiana *(d)* Escalón

Figura 2.5: Funciones de activación más comunes

Red Neuronal Monocapa

Como su nombre sugiere, son redes neuronales de una sola capa, Mangne Machaca (2009, p.16), las define como "Son redes de una sola capa, establecen conexiones laterales entre las neuronas que pertenecen a la única que constituye a la red, también pueden existir las conexiones autorrecurrentes (salida de una neurona conectada a su propia entrada)".

La figura 2.6 muestra el modelo matemático de una red neuronal monocapa.

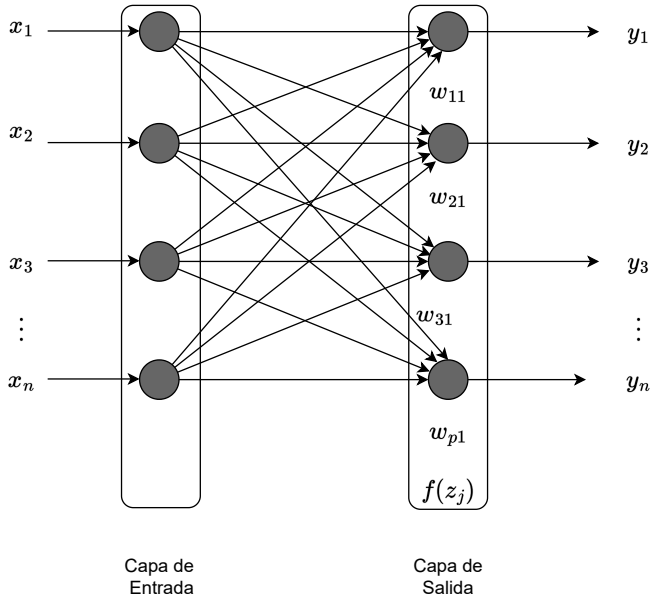


Figura 2.6: Modelo matemático de una red neuronal monocapa

La salida de la red monocapa esta expresada en la ecuación

$$y_j = f(\sum_{i=0}^{n} w_{ji} x_i + b) \tag{2.9}$$

Donde

- x_i es el elemento i-ésimo del vector de entrada de la red neuronal,
- w_{ji} es la matriz de pesos de la red neuronal,
- b es el bias de la neurona,
- f es una función de activación neuronal.

Red Neuronal Profunda

Una Red Neuronal Profunda o Deep Neural Network (DNN) es un tipo de red neuronal diseñada para aprender patrones complejos en los datos de entrada. Adicionalmente, Jennifer S. et al. (2021, p. 731) menciona que las redes profundas tienes más de una capa de cálculo, las cuales son comúnmente conocidas como **capas ocultas**.

La figura 2.7 muestra la topología de una red profunda.

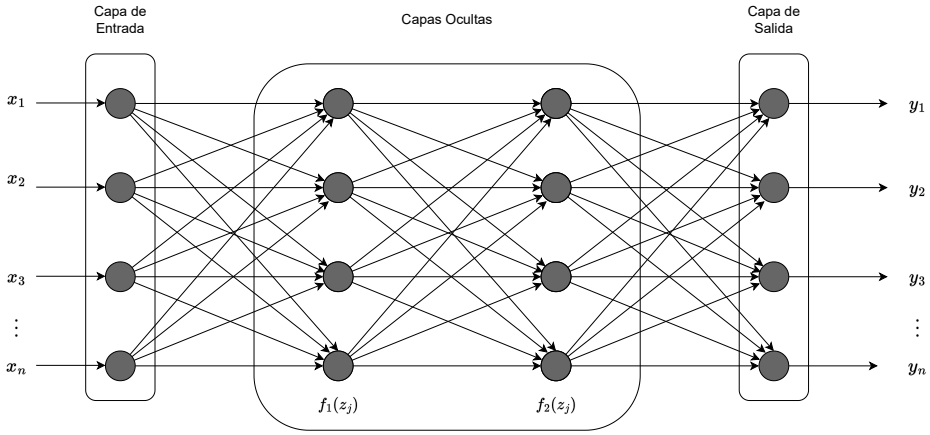


Figura 2.7: Topología de una Red Neuronal Profunda

Para comprender este tipo de red neuronal se plantea una instancia de entrenamiento de la forma (X, y), donde $X = [x_1, x_2, x_3, \cdots, x_d]$ que contiene n variables de características y $y \in [0, 1]$, donde y es el valor observado. El IL *Input Layer* o *capa de entrada* contiene d neuronas con pesos $W = [w_1, w_2, w_3, \cdots, w_d]$. El valor de salida de la neurona es calculado usando la función lineal descrita en la expresión 2.10.

$$y = h(z) \tag{2.10}$$

Donde
h es la **función de activación**.

La salida de cada neurona se describe en la expresión 2.11.

$$z = \sum_{j=1}^{d} w_j x_j + b \tag{2.11}$$

En función de un **umbral**, según la **función de activación** utilizada. Cada instancia de los datos se introduce en la red neuronal y los pesos W se actualizan en función de error $(y - \hat{y})$.

La expresión 2.12 se utiliza para calcular los pesos sinápticos de las siguientes neuronas.

$$W = W + \alpha(y - \hat{y})X \tag{2.12}$$

Donde

- y es la predicción
- α es la taza de aprendizaje de la red neuronal.

Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNN por sus siglas en inglés, Recurrent Neural Network), son un tipos de red neuronal artificial especializada en procesas secuencias de datos. (Pérez Guerrero, 2020, p. 19) menciona que las Redes Neuronales Recurrentes conforman una familia de modelos diseñados para procesar datos de estructura secuencial. Ademas de resaltar su potencia por tener un estado oculto con una dinámica no lineal que permite a la red recordar y procesar información.

Adicionalmente, Bonet Cruz et al. (2007, p. 51) aclara que "Existen tres tipos de tareas esenciales que se pueden realizar con este tipo de redes:

- Reconocimiento de secuencias: Se produce un patrón de salida particular cuando se especifica una secuencia de entrada.
- Reproducción de secuencias: La red debe ser capaz de generar el resto de una secuencia cuando ve parte de ella.
- Asociación temporal: En este caso una secuencia de salida particular se debe producir en respuesta a una secuencia de entrada especifica."

Existen dos categorías para las redes neuronales recurrentes; **parcial** y/o **totalmente** recurrente.
La primera consiste en aquellas en donde cada neurona puede estar conectada a cualquier otra y sus conexiones recurrentes son variables. La segunda categoría es aquella en donde sus conexiones recurrentes son fijas, siendo esta ultima la forma más usual para reconocer o reproducir secuencias.

Adicionalmente, (Bonet Cruz et al., 2007, p. 51), aclara que existen 5 clases de algoritmos de aprendizaje, los cuales son:

1. Backpropagation Through Time (BPTT)
2. Forward Propagation o Rea/ Time Recurrent Learning (RTRL)
3. Fast Forward Propagation
4. Funciones de Green
5. Actualizaci6n en Bloque (Block Update)

La figura 2.8 muestra la estructura básica de una red neuronal recurrente

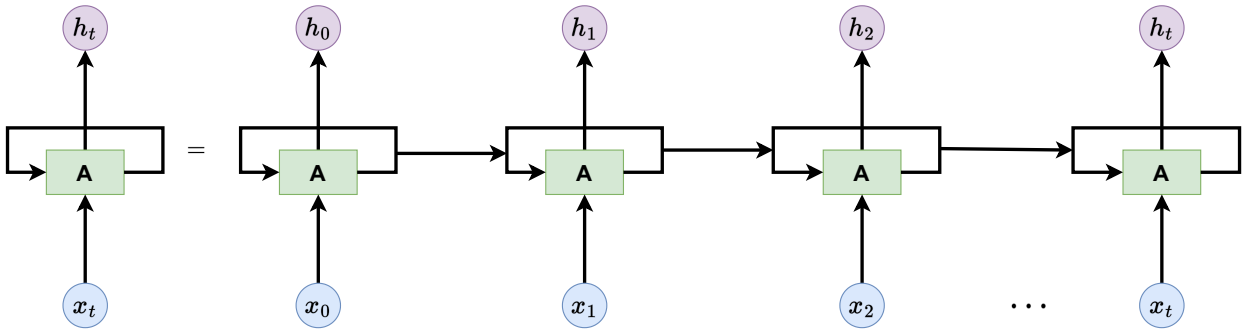


Figura 2.8: Estructura básica de una RNN

La ecuación 2.13 muestra la forma generalizada para las relaciones de recurrencia.

$$h(t) = f(h^{t-1}, x^t) \tag{2.13}$$

Donde:
$x^{(t)}$ representa la entrada de una instancia de tiempo en particular.
$h^{(t-1)}$ es una representación de los aspectos relevantes de la secuencia pasada de entradas hasta t

Unidad Recurrente con Compuertas

Las redes **GRU** (Gated Recurrent Unit) son una variación de las RNN, las cuales utilizan una denominada *puerta de actualización* y una *puerta de reinicio*, de acuerdo con (Matrínez Sastre, 2021, p. 20), menciona que estas puertas son dos vectores que deciden que información debe pasar a la salida.

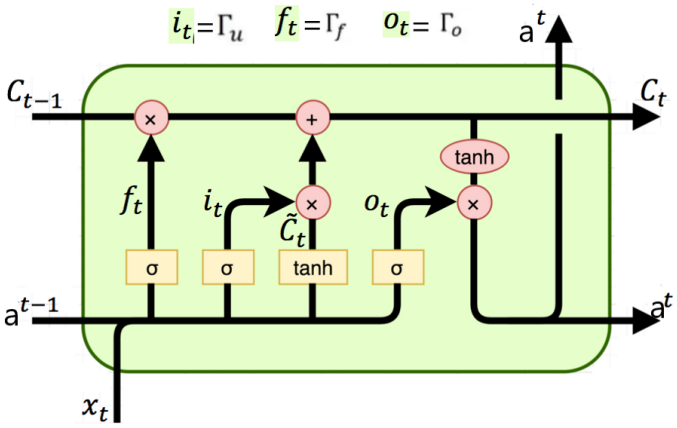


Figura 2.9: Estructura de una Unidad Recurrente con Compuertas (Matrínez Sastre, 2021, p. 20)

Dada una secuencia de entrada en un instante de tiempo t, denominada x_t y el estado oculto anterior $a^t - 1$.

La ecuación 2.14 muestra la función de la compuerta de reinicio

$$r(t) = \sigma(W_{rx}x(x_t)) + U_{rx}h(t-1) + b_r \tag{2.14}$$

Donde
W_r y U_r son matrices de pesos.
x_t es la entrada en el instante de tiempo t
h_t es es estado oculto anterior.
σ es la función de activación sigmoide.

La ecuación 2.15 muestra la función de la compuerta de actualización

$$C_{t-1} = \sigma(W_{zx}x(x_t)) + U_{zx}h(t-1) + b_z \tag{2.15}$$

Donde
W_z y U_z son matrices de pesos.
x_t es la entrada en el instante de tiempo t
h_t es es estado oculto anterior.
σ es la función de activación sigmoide.

La ecuación 2.16 muestra la función del estado candidato

$$C_t = tanh(W_{sx}x(t) + r(t) \cdot (U_{sx}h(t-1)) + b_s) \tag{2.16}$$

Donde
W_s y U_s son matrices de pesos.
x_t es la entrada en el instante de tiempo t
h_t es es estado oculto anterior.

2.3.7. Métricas de Evaluación

En el trabajo realizado por Jennifer S. et al. (2021, p. 732), en donde el modelo empleado fue una red neuronal profunda, se menciona el uso de métricas de evaluación tales como **Precisión**, **Sensibilidad**, **Especificidad**, **Clasificación Errónea** y **Puntuación F1**.

- **Precisión**
  Se define como la capacidad de un modelo para diferenciar correctamente los casos sanos de los no sanos.

  La expresión 2.17 es usada para calcular la precisión del modelo.

$$P = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.17}$$

  En donde
  - P es la precisión a calcular.
  - TP son los valores *verdaderos positivos* **True Positive**
  - TN son los valores *verdaderos negativos* **True Negative**
  - FP son los valores *falsos positivos* **False Positive**
  - FN son los valores *falsos negativos* **False Negative**
- **Sensibilidad**
  Es la capacidad de un modelo para identificar correctamente los casos no sanos.

  La expresión 2.18 es utilizada para calcular la sensibilidad del modelo.

$$S = \frac{TP}{TP + FN} \tag{2.18}$$

  Donde

S es la sensibilidad del modelo.

- **Especificidad** Es la capacidad de d un modelo para identificar correctamente los casos sanos.

  La expresión 2.19 es empleada para determinar la especificidad del modelo.

$$E = \frac{TN}{FP + FN} \tag{2.19}$$

  Donde E es la especificidad del modelo

- **Clasificación Errónea** Es el indice de errores en la clasificación de los datos.
  La expresión 2.20 es utilizada para calcular la clasificación errónea.

$$CE = \frac{FP + FN}{TP + TN + FP + FN} \tag{2.20}$$

- **Puntaje F1** La precisión puede no ser siempre la mejor métrica, ya que falsos negativos y falsos positivos son considerados. El puntaje F1 es útil cuando la distribución es uniforme.
  La expresión 2.21 es usada para determinar el puntaje F1 del modelo.

$$F1 = \frac{TP}{TP + 0.5(FP + FN)} \tag{2.21}$$

2.3.8. Procesamiento de Señales

El primer paso para la detección de características presentes en un electrocardiograma, inicia con el filtrado del ruide presente en estas. En trabajos como (Ahamed et al., 2015, 3) y (Rajkumar et al., 2019, p. 366) se emplean algunos filtros, dentro de los que destacan **filtro pasa bajas**, **filtro pasa altas** y **filtro de Butterwort**.

- **Filtro pasa bajas**
  Es un filtro que permite el paso de frecuencias por debajo de cierto rango de corte, con el objetivo de atenuar o bloquear las frecuencias por encima de esta frecuencia.
  La expresión 2.22 muestra la función de transferencia de de un filtro pasa bajas de primer orden.

$$H(s) = \frac{1}{1 + \frac{s}{\omega_c}} \tag{2.22}$$

  Donde s es la variable compleja de Laplace.
  ω_c es la frecuencia de corte del filtro.
  $H(s)$ es la función de transferencia del filtro.

- **Filtro pasa altas**
  Este tipo de filtros permite el paso de frecuencias por encima de cierto rango de corte, de igual forma atenúa las frecuencias por debajo de esta frecuencia.
  La expresión 2.23 describe la función de transferencia de un filtro pasa altas.

$$H(s) = \frac{s}{s + \omega_c} \tag{2.23}$$

  Donde

s es la variable compleja de Laplace.
ω_c es la frecuencia de corte de filtro.
$H(s)$ es la función de transferencia del filtro.

- **Filtro de Butterwort**
  Es un tipo de filtro pasa bajas el cual emplea una frecuencia plana en la banda de paso y atenúa suavemente fuera de la banda de paso.
  La expresión 2.24 muestra la función de transferencia de un filtro de Butterwort.

$$H(s) = \frac{1}{\left[1 + \left(\frac{s}{\omega_c}\right)^{2n}\right]^{\frac{1}{2}}} \tag{2.24}$$

  Donde
  s es la variable compleja de Laplace.
  ω_c es la frecuencia de corte del filtro.
  n es orden del filtro.

A diferencia del caso anterior, la obtención de las características presentes en la señal se empleo una técnica diferente la cual es la **transformada de Wavelet**, De acuerdo con Vargas-Cañas et al. (2021, p. 271) la transformada de Wavelet consiste en la convolución de un señal $f(t)$ con una función Wavelet madre $\varphi(t)$ desplazada en el tiempo por un parámetro de traslación b y dilatada por un parámetro de escala a.

La expresión 2.25 muestra la implementación de la transformada de Wavelet.

$$Wf(a,b) = \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt[2]{a}}\varphi \times \left(\frac{t-b}{a}\right)dt : a, b \in \mathbb{R}, a \neq 0 \tag{2.25}$$

Donde
$\varphi * (t)$ es el complejo conjugado de la función de wavelet $\varphi(t)$ y se comprime o expande dependiendo del parámetro de escala a.

2.3.9. Placas de Desarrollo

FPAA (Field Programmable Analog Array)

Una FPAA, en español, "Matriz Analógica Programable en Campo", se trata de un circuito integrado el cual consta de bloques analógicos interconectados. Ademas de su definición, en el trabajo realizado por Rodríguez et al. (2023, p. 3) se menciona que se empleó una FPAA para el procesamiento de se señales ECG, y algunas otras tareas relacionadas con este propósito, dentro de las que se encuentran:

- Amplificación de la señal.
- Filtrado de la señal.
- Digitalización de la señal.

Todo esto con la finalidad de crear un microsistema electrónico capaz de adquirir y registra señales

cardíacas.

La figura 2.10 muestra un ejemplo de una FPAA (Field Programmable Analog Array)

Figura 2.10: El FPAA AN221E04 de Anadigm (Anadigm. Inc, 2012)

Arduino

Esta es quizás la plataforma de desarrollo de desarrollo más popular para el desarrollo de sus sistemas de adquisición de señales electrocardiográficas, ya que tanto (Vargas-Cañas et al., 2021) como (Ahamed et al., 2015) y (Prabu y Ravikumar, 2016) emplearon placas Arduino Uno en alguna de las fases de adquisición de señales de sus proyectos.
Prabu y Ravikumar (2016, p. 3) menciona que "Arduino es una plataforma de creación de prototipos de código abierto basada en hardware y software fáciles de usar.". Adicionalmente, esta son capaces de leer entradas y transformarlas en una salida.

Dentro de las etapas del proceso de adquisición de señales ECG, Prabu y Ravikumar (2016, p. 273) declara que se implemento un programa en la tarjeta Arduino Uno para la lectura y envío de las señales almacenadas en memoria cada $2ms$.

Finalmente Ahamed et al. (2015, p. 10) expresa que empleo esta tarjeta para la conversión analógica a digital y la transmisión en serie de la señal ECG.

La figura 2.11 muestra una placa Arduino Uno.

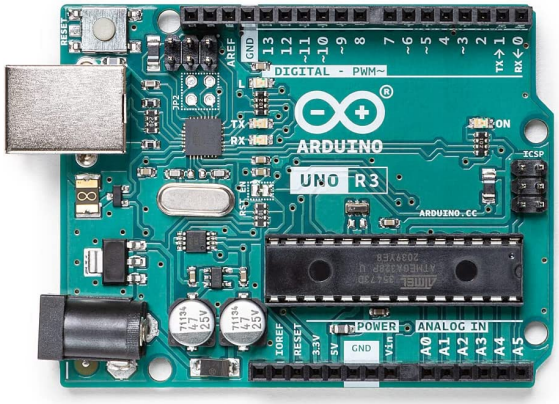


Figura 2.11: Placa Arduino uno con el microcontrolador ATmega328P

2.4. Áreas de Oportunidad

Una vez analizada la literatura se determinó que el proyecto propuesto presenta algunas ventajas y/o mejoras con respecto a los trabajos previamente consultados, sin embargo en esta sección se explicaran a detalle cada una de las areas de oportunidad encontradas al momento de realizar la revisión de la literatura.

1. Del trajo realizado por (Smith et al., 2022) se determinó que carece de funciones relacionadas al análisis y procesamiento de señales, en este aspecto, una opción es implementar algoritmos de Inteligencia Artificial, más específicamente de Machine Learning para realizar la clasificación de las señales ECG normales y anormales.
2. En el caso del articulo escrito por (Rodríguez et al., 2023) se puede apreciar que la visualización de la forma de onda depende del envío inalámbrico de datos entre el sistema de adquisición y el equipo de computo, en donde una limitante es el tiempo de envió entre estos. Por lo tanto realizar una transferencia serial y un rediseño de la interfaz que permita mostrar dos o más derivaciones de electrocardiograma mejorarían el desempeño.
3. En cuanto a procedimiento mostrado en la patente de (Palma et al., 2020), probar con diferentes técnicas de filtrado podría mejorar el resultado de la fase de análisis de señales cardíacas.
4. En cuanto a la propuesta mostrada por (Abuhaija et al., 2023), analizar con técnicas exploratorias las señales ECG para resaltar características implícitas en la señal, para posteriormente clasificarlas.
5. En el caso de (Jennifer S. et al., 2021), implementar una red neuronal para clasificar las señales ECG en normales y anormales, evaluando los resultados obtenidos para obtener un modelo eficiente.
6. En el caso de (González Medina, 2022), conocer las enfermedades cardiovasculares más comunes para considerarlas en el entrenamiento del modelo, lo que permitirá una mejor precisión al momento de clasificar.
7. En relación con (Ortega Ordoñez et al., 2023), implementar herramientas especializadas como MATLAB para aplicar filtros, con el objetivo de atenuar la señales.
8. Del articulo realizado por (Vargas-Cañas et al., 2021) se encontró que realizar análisis en tiempo real a la forma de onda generada por el electrocardiograma puede mejorar la capacidad de diagnostico.
9. En cuanto al trabajo realizado por (Ahamed et al., 2015), realizar evaluaciones a los modelos implementados permitiría escoger aquellos con los mejores resultados.
10. En cuanto a (Rajkumar et al., 2019), se determinó que enfocar el estudio a un numero concreto de arritmias permitirá una respuesta más certera. En un futuro, sera posible incrementar el numero de arritmias que el modelo clasificará.
11. En relación con (Prabu y Ravikumar, 2016), una opción sería implementar algoritmos que permitan la clasificación se señales aún con ruido implícito en la señal.
12. Finalmente, en el trabajo realizado por (Dorado Díaz et al., 2019) una area de oportunidad es eliminar la dependencia en cuanto a la cantidad de los datos de entrenamiento, enfocándose mayormente en la calidad de estos.

Adicionalmente. la tabla 2.3 muestra la matriz de referencias, en donde se exponen los trabajos que conforma la literatura y sobre todo los métodos y areas de oportunidad.

Tabla 2.3: Matriz de Referencias

Fuente	Resumen	Método	Áreas de Oportunidad
Aplicación Android para la adquisición inalámbrica y visualización de señales biomédicas. Smith, R.E., Socarrás, B. N., & Vázquez, C. R.(2022).	En la actualidad se han implementado diversas aplicaciones (software) móviles con afines a la monitorización inalámbrica de señales para adquirir, visualizar y almacenar variables biomédicas.	Para la construcción de dicha aplicación se emplea el lenguaje de programación Java, por lo que se basa de recursos como botones, listas, imágenes, etc. Y para montarla en el sistema operativo Android se emplea Android Studio versión 4.3.1 del IDE Intellij.	Incluir más funcionalidades relacionadas al procesamiento y análisis de los bioseñales: Implementación de algún algoritmo de inteligencia artificial para realizar clasificaciones o predicciones de enfermedades
Diseño de un Microsistema para Adquisición de Señales Cardíacas Usando FPAAs. Rodriguez, P., Castro, H., Pinedo, C. & Velasco, J. (2023).	En el corazón se producen variaciones de voltaje, gracias a la implementación de microelectrodos se pueden adquirir dichos datos como señales bioeléctricas y posteriormente representarlos gráficamente como señales ECG, las cuales presentan distintas regiones (P, Q, R, S y T).	CARDIOCEL se concibe como un microsistema electrónico económico que permite adquirir señales ECG. Consta de dos partes importantes: - Hardware (adquirir, procesar señales analógicas, digitalizarlas y transmitirlas). - Software (procesamiento y visualización de las ECG).	Rediseño del sistema para visualizar simultáneamente dos o más derivaciones de ECG. Implementar otro tipo de comunicación inalámbrica para que el microsistema CARDIOCEL se pueda integrar a una red de telemetría hospitalaria.
Procedimiento para la detección de señales balistocardiográficas y sistema que lo implementa OMPI. Palma, A. et al. (2020)	La implementación de balistocardiograma (BCG) tiene la ventaja de no usar electrodos u otros elementos que se dieran al cuerpo para monitorear los latidos del corazón, pero si emplean sensores capacitivos con la finalidad de ofrecer más flexibilidad y comodidad.	Para adquirir dicha información existen diversos métodos: - Filtro paso baja o FPB. - Transformada de Wavelet. - Técnicas de aprendizaje no supervisado.	Implementación de otras técnicas (diferentes a FPB.) para analizar señales cardíacas.

A comprehensive study of machine learning for predicting cardiovascular disease using Weka and SPSS tools Abuhaija, B. et. al. (2023)	La clasificación es considerada como un sistema de aprendizaje supervisado que usa datasets etiquetados representando predicciones, ANN y MLP son una de las mejores técnicas para clasificar datos organizados en forma tabular.	Los métodos de machine learning más comunes para predecir la presencia de ataques al corazón incluyen k-Nearest Neighbor con poca precisión y Random Forest con la mayor precisión con un 90 %. Otras como CNN (Convolutional Neural Network) con un 82 % de precisión.	Analizar con técnicas exploratorias las señales ECG para identificar enfermedades cardíacas por medio de clusters o redes neuronales artificiales.
Intelligent Sustainable Systems: Proceedings of ICISS 2021 (Lecture Notes in Networks and Systems, 213) Jennifer S. Raj. et. al. (2021)	Una Red Neuronal Profunda (DNN) permite robustez para variaciones de datos, así como la generalización para múltiples aplicaciones además de ser escalable para más datos.	Los dataset más completos para la predicción de enfermedades cardiovasculares son Statlog y Cleveland. Para su preprocesado cada uno fue dividido en un radio de 70:30, es decir, el 70 % está destinado a entrenamiento y el 30 % restante para probar el modelo.	Emplear de señales electrocardiográficas para la clasificación y predicción de enfermedades cardiovasculares
Análisis Espacial De Las Enfermedades Cardiovasculares En México: Modelos Predictivo González Medina, L. E., (2022).	Dentro del modelo predictivo se considera el manejo y desarrollo identificado como campo aleatorio que es un proceso estocástico que toma valores en un espacio euclidiano, ya que dentro de su proceso está definido sobre un espacio parametral de dimensión uno.	La estadística de los datos permite la manipulación y clasificación, las diversas predicciones se manejan por medio de áreas las cuales por ejemplo permiten la identificación del riesgo relativo de padecer una enfermedad cardiovascular, número de casos, casos esperados.	Identificar y conocer el manejo espacial de datos dentro de un sistema de modelos predictivos de enfermedades cardiovasculares.

Adquisición de Señales analógicas de instrumentación con logo! soft V8.3 Mediante Generador de Señales y el sensor PT100, Ortega Ordoñez, R.C. et al. (2023)	Por medio del análisis de señales permiten la visualización de los diversos datos de un manejo de datos altos y bajos para la adquisición de la predicción, ya que este mecanismo permite la identificación de enfermedades cardiovasculares obteniendo así las variables identificadas para tener nuevos resultados.	La adquisición de datos se obtiene por medio de las señales o de igual manera se puede utilizar el software LabVIEW que permite manejar un entorno virtual para el manejo de análisis y predicción de personas con enfermedades cardiovasculares.	Aplicar filtros pasa bajas, altas, o pasa bandas para determinar el filtro más adecuado en este ámbito.
Construcción de un sistema electrocardiográfico con conexión inalámbrica a teléfonos inteligentes Andrés , R.M. Willian Andrés,C.C. et al. (2021)	Para el diagnóstico y tratamiento primeramente es indispensable la realización de un examen médico electrocardiograma, el cual dentro de su proceso registra la actividad eléctrica del corazón, dicho proceso permite el análisis para la predicción en la persona si tiene alguna enfermedad cardiovascular.	Red de sensores inalámbricos ECG o WSN son un conjunto de dispositivos de tamaño pequeño interconectados entre sí, su principal funcionamiento es que monitorea variables físicas para enviar información del entono de manera inalámbrica hasta el dispositivo en donde llegara la señal o servidor.	Aplicar análisis de de señales ECG dentro del sistema electrocardiográfico
Design and implementation of low cost ECG monitoring system for the patient using smartphone. Ahamed, Md Asif, Hasan, Md. Kamrul, & Alam, Md. (2015).	El dispositivo se conecta a los electrodos de ECG colocados en el cuerpo humano y envía los datos de ECG adquiridos a través de Bluetooth al smartphone.	La aplicación utiliza algoritmos de procesamiento de señales para filtrar la señal de ECG y visualizarla en tiempo real.	Realizar evaluaciones clínicas para determinar la precisión y la eficacia del sistema de monitoreo de ECG en comparación con sistemas convencionales.

Classification of ECG Arrhythmia Using Deep Learning. Mohammadpour, A., & Sadabadi, F. (2019).	Se describe un enfoque basado en una red neuronal convolucional profunda (CNN) para la clasificación de arritmias cardíacas. La base de datos utilizada en el estudio es la base de datos de arritmia del MIT-BIH, que contiene 48 registros de señales de ECG de pacientes con diferentes tipos de arritmias. El conjunto de datos se divide en conjuntos de entrenamiento y prueba en una proporción de 70:30.	Los resultados del estudio muestran que la CNN propuesta obtuvo una tasa de precisión del 98,4 % en la clasificación de las arritmias, lo que indica que el enfoque propuesto es altamente efectivo en la detección y clasificación de arritmias cardíacas.	Enfocar el estudio en la clasificación de cuatro tipos de arritmias, pero existen muchos otros tipos de arritmias que podrían ser abordados en futuros estudios.
Real-Time ECG Monitoring System Using Raspberry Pi and Arduino for Healthcare Applications. Zolkipli, M. Z., Yusof, N. M., & Ali, N. M. (2018).	El microcontrolador Arduino es el encargado de procesar la señal de ECG y de enviar los datos a través del puerto serial al Raspberry Pi. En el Raspberry Pi, los datos de ECG son procesados y analizados por medio de un programa en Python.	El sistema desarrollado fue evaluado utilizando señales de ECG de voluntarios y se encontró que el sistema es capaz de adquirir y procesar las señales de ECG en tiempo real con una alta precisión.	Investigar cómo mejorar el rendimiento del sistema en ambientes ruidosos, como en presencia de otros dispositivos médicos o en entornos hospitalarios.
Aplicaciones de la inteligencia artificial en cardiología: el futuro ya está aquí. Dorado-Díaz, P.I., Sampedro-Gómez, J., Vicente-Palacios, V., Sánchez, P.L., 2019.	Existen dos tipos de técnicas de aprendizaje AA: supervisado y no supervisado. En el aprendizaje supervisado, se cuenta con un conjunto de datos etiquetados para predecir una variable de respuesta específica, mientras que en el no supervisado, no se tiene información sobre la variable de respuesta. Los algoritmos de clasificación y regresión se utilizan para la predicción en el aprendizaje supervisado.	El proceso de construcción de un modelo de Aprendizaje Automático (AA) no se limita a aplicar un algoritmo de aprendizaje a una base de datos, sino que es un proceso más complejo que involucra varios pasos. En primer lugar, se parte de los datos brutos y se realiza un preprocesado para convertirlos en datos estructurados.	Identificar patrones que los seres humanos no serían capaces de detectar. Sin embargo existe una fuerte dependencia de la calidad y cantidad de los datos utilizados para entrenar el modelo.

2.5. Conclusión de la Revisión

Una vez realizada la revisión de la literatura y la exposición de las areas de oportunidad es posible determinar que el proyecto presenta algunas varios hallazgos trascendentes.
En primer lugar, el procedimiento de adquisición de señales por medio de la placa Arduino DUE y el amplificador AD8232, tiene una alta probabilidad de éxito, tanto por las especificaciones de los componentes, como los algoritmos empleados para realizar esta tarea. En cuanto al procedimiento de filtrado existe la incognita acerca de que herramienta emplear, ya que algunas fuentes sugieren usar la plataforma de desarrollo Arduino IDE, mientras que otras reconocen que MATLAB es una alternativa superior. Adicionalmente, ua brecha brecha importante en a literatura es la clasificación binaria de las señales ECG, es decir, clasificar una señal de acuerdo a su forma de onda y determinar si esta es normal o anormal, dicha brecha se pretende cubrir en el desarrollo de esta investigación.

En conclusión la revisión de la literatura a demostrado que los componentes físicos (hardware) compuestos por la placa Arduino DUE y el amplificador AD8232, tienen la capacidad de adquirir señales ECG con calidad y precisión, de la misma forma, los componentes de software poseen una alta probabilidad de éxito en cuanto a la adquisición de características presentes en la señal y su posterior clasificación.

Capítulo 3

Método

3.1. Ciclo de Vida Iterativo

Las metodologías ágiles se caracterizan por su flexibilidad, dado que pueden modificar los requerimientos y entregas al cliente para ajustar a la realidad percibida. (Navarro et al., 2013).

Dentro de la clasificación de dichas metodologías se destaca la nombrada SCRUM, la cual es un marco de trabajo diseñado para lograr una colaboración eficaz entre los equipos que integran el proyecto, se destaca por definir roles para formar una estructura y así lograr un correcto funcionamiento, además de implementar procesos iterativos (Navarro et al., 2013).

Tanto J.J (2020, p. 16) , como (Cho, 2010) menciona que los roles de la metodología SCRUM son tres, los cuales son:

- Propietario del producto (cliente)
- Equipo de desarrollo (SCRUM Team)
- Scrum Master (Líder del proyecto)

La imagen 3.1 muestra las fases de la metodología SCRUM de forma general.

La adaptación de la metodología SCRUM para el proyecto propuesto se muestra en la siguiente sección, donde se especifican las actividades a realizar por cada una de ellas.

- Sprint de Planificación
  Esta fase esta compuesta por:
  - Identificación de los objetivos de proyecto
  - Identificación de los requerimientos del sistema.
  - Identificación de los recursos necesarios.
  - Diseño del diagrama de bloques del sistema.
  - Selección de los componentes del sistema.

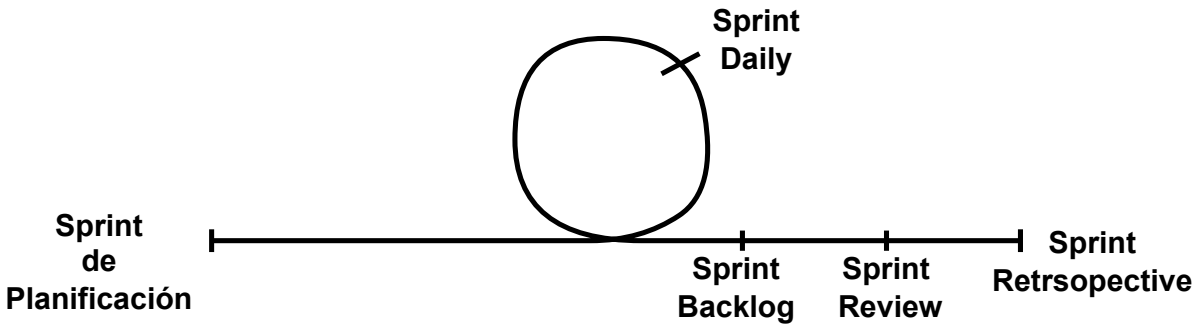


Figura 3.1: Fases de la metodología SCRUM (Ryan Ripley, 2020)

- Diseño de la arquitectura del software.

- Sprint Daily

  El cual esta compuesto por:
  - Reuniones Cliente - Equipo
- Sprint Backlog

  Sus componentes son:
  - Instalación y configuración del software necesario.
  - Implementación del algoritmo de procesamiento de señales ECG.
  - Implementación de la técnicas de Machine Learning.
  - Verificación del software implementado
- Sprint Review

  El cual esta compuesto por:
  - Integración del Hardware y Software.
  - Pruebas unitarias del Hardware y Software.
  - Pruebas de integración del sistema completo.
- Sprint Retrospective

  Compuesto por:
  - Documentación del proyecto.
  - Implementación en escenarios reales.

3.2. Requerimientos

3.2.1. Requerimientos Específicos

- Interfaz Gráfica

  La interfaz gráfica del sistema para el usuario necesita ser intuitiva, buscando que sin instrucciones previas y a primera vista el usuario pueda identificar los componentes y secciones del sistema. La interfaz deberá contar con una paleta de color agradable a la vista, el usuario debería poder usar la interfaz por un largo tiempo sin problemas.

- Información del paciente
  - En la interfaz principal, en la parte superior izquierda habrá un pequeño formulario que permitirá al usuario (médico o investigador) ingresar los datos de identificación del paciente, tales como nombre y edad, el registro deberá hacerse con caracteres alfanuméricos, se debe registrar a cada paciente por cada sesión del sistema.

- Pantalla de bioseñales
  - Esta sección muestra las bioseñales capturadas del corazón en tiempo real.
  - Su objetivo es mostrar tanto al usuario como al paciente el electrocardiograma
  - Rango de este campo estará en función del tiempo que dure la sesión o hasta que el cliente lo indique.
- Pantalla de bioseñales
  - Esta sección muestra los resultados del análisis de la señal, como frecuencia cardíaca.
  - El rango de este campo depende del tiempo de la sesión.
  - La frecuencia cardíaca deberá medirse en pulsaciones por minuto.
  - El formato de los datos deberá ser numérico.
- Pantalla clasificación por redes neuronales
  - Esta sección muestra la clasificación de la señal.
  - Deberá mostrar si la señal es normal o no, indicando el porcentaje de normalidad o anormalidad.
- Pantalla exportar datos
  - Esta sección permite generar un archivo pdf con los datos generados en las pantallas de bioseñales, análisis de señales, y la clasificación por redes neuronales.

- Funciones
  Las funciones del sistema "Cardio Scope" deben definir las acciones fundamentales que tiene lugar en el software.
  - Visualización de Datos
    - El sistema mostrará la frecuencia cardíaca y la forma de onda del ECG, así como otros datos pertinentes.
  - Análisis de la señal
    - El sistema proporcionará los pulsos por minuto de la forma de onda.
  - Exportación de datos
    - El sistema permitirá al usuario exportar los datos generados en la sesión en un archivo en formato pdf.
- Rendimiento
  Los requisitos de rendimiento del sistema "Cardio Scope" constan de tres elementos fundamentales: velocidad, capacidad y eficiencia.
  - Capacidad
    - El sistema debe albergar los datos tanto de entrenamiento como de validación.
  - Velocidad
    - El sistema debe mostrar tanto la forma de onde del ECG, así como valores numéricos de este y la clasificación en tiempo real.

- Eficiencia
  - El sistema debe realizar el proceso de adquisición, análisis y clasificación sin interrupciones.

3.2.2. Requerimientos del Hardware

El hardware con el que cuente el sistema para la adquisición de señales ECG con Arduino para su clasificación con redes neuronales deberá ser capaz de adquirir señales ECG de alta calidad, amplificar y filtrar la señal para reducir el ruido y mejorar la calidad. Un hardware adecuado y bien diseñado es crucial para la precisión y fiabilidad del sistema de adquisición y clasificación de señales ECG, esto puede influir en la calidad de la atención proporcionada al paciente, En este sentido se deben considerar cuidadosamente los requerimientos funcionales del hardware y la compatibilidad con los componentes del sistema en general.

- Adquisición, amplificación y filtrado de señales ECG
  - El hardware debe ser capaz de adquirir señales ECG con suficiente calidad para permitir una clasificación precisa.
  - Para la amplificación el hardware debe ser capaz de amplificar y filtrar la señal ECG con el fin de reducir el ruido mejorar la calidad de la señal adquirida.
- Conversión analógico-digital
  - El hardware deberá ser capaz de convertir la señal ECG analógica a digital para que pueda ser procesada por el sistema de clasificación de redes neuronales.
- Conectividad
  - Debe haber compatibilidad entre Arduino y los dispositivos como el sensor y la computadora, para lograr transmitir la señal ECG y los resultados de clasificación.
- Alimentación
  - El hardware para la adquisición de a señal ECG debe ser alimentado de manera confiable y segura, preferiblemente por una fuente de alimentación externa.

3.2.3. Otros Requerimientos

La norma ISO 14971:2019 se enfoca en la gestión de riesgos de los dispositivos médicos, y puede ser aplicable al proyecto, ya que involucra la adquisición y clasificación de señales ECG. Esta norma establece los requisitos para la identificación, análisis, evaluación y control de los riesgos asociados con los dispositivos médicos.

- Identificación de riesgos
  - Es necesario identificar los posibles riesgos que puedan resultar del uso del dispositivo medico en el proceso de adquisición de las señales ECG.
- Análisis de Riesgos
  - Se debe evaluar la probabilidad y posible impacto de cada riesgo identificado, se recomienda el uso de herramientas para análisis como FMEA (Análisis de modo y efecto de Falla) para identificar causas y consecuencias de los riesgos.
- Evaluación de riesgos

- El nivel de riesgo identificado, puede realizarse por medio de una matriz de riesgos.

- Control de Riesgos
  - Aplicar medidas de control para reducir los riesgos hasta un nivel aceptable,tales como, implementar controles de calidad, capacitación personal, entre otras.
- Monitoreo y Revisión
  - Se deberá realizar de forma periódica la actualización y mantenimiento de los dispositivos de obtención de señales.

3.2.4. Diagrama de Bloques

Los componentes del sistema se muestra en la figura 3.2, como se puede apreciar este esta compuesto por tres bloques (sistemas) principales: **Adquisición de Señales ECG**, **Procesamiento de señales ECG** y **Interfaz de Usuario**.

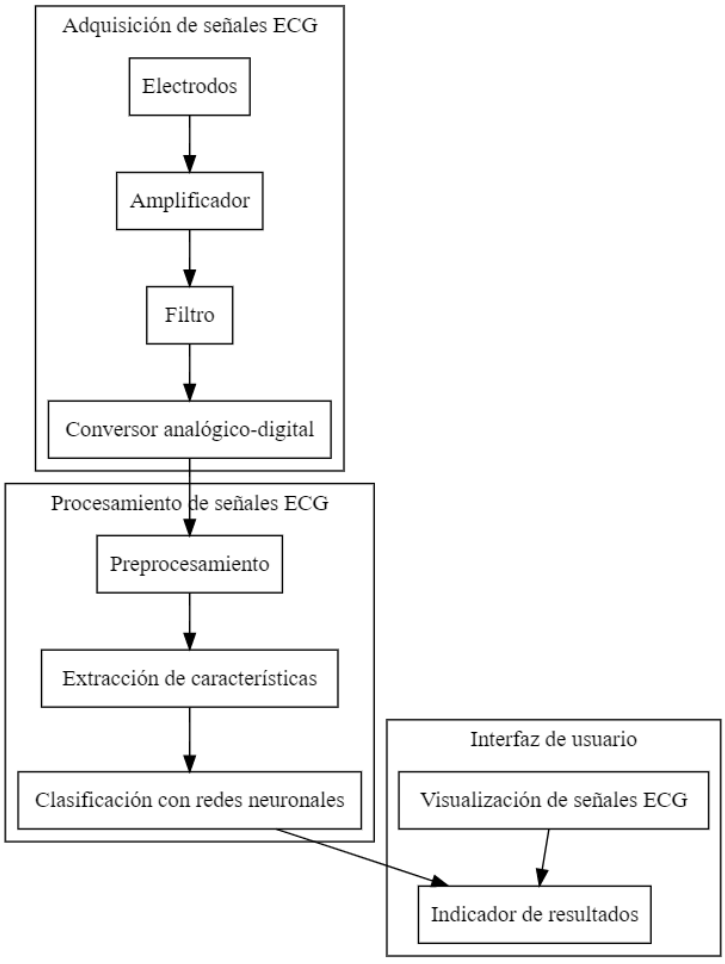


Figura 3.2: Diagrama de bloques con los componentes del sistema

3.2.5. Diagrama de Casos de Uso

La figura 3.3 muestra el diagrama de casos de uso del proyecto, como se puede apreciar el usuario solo interactúa con el sistema y este provee de las respuestas para las que esta diseñado, todo esto por medio de una interfaz gráfica.

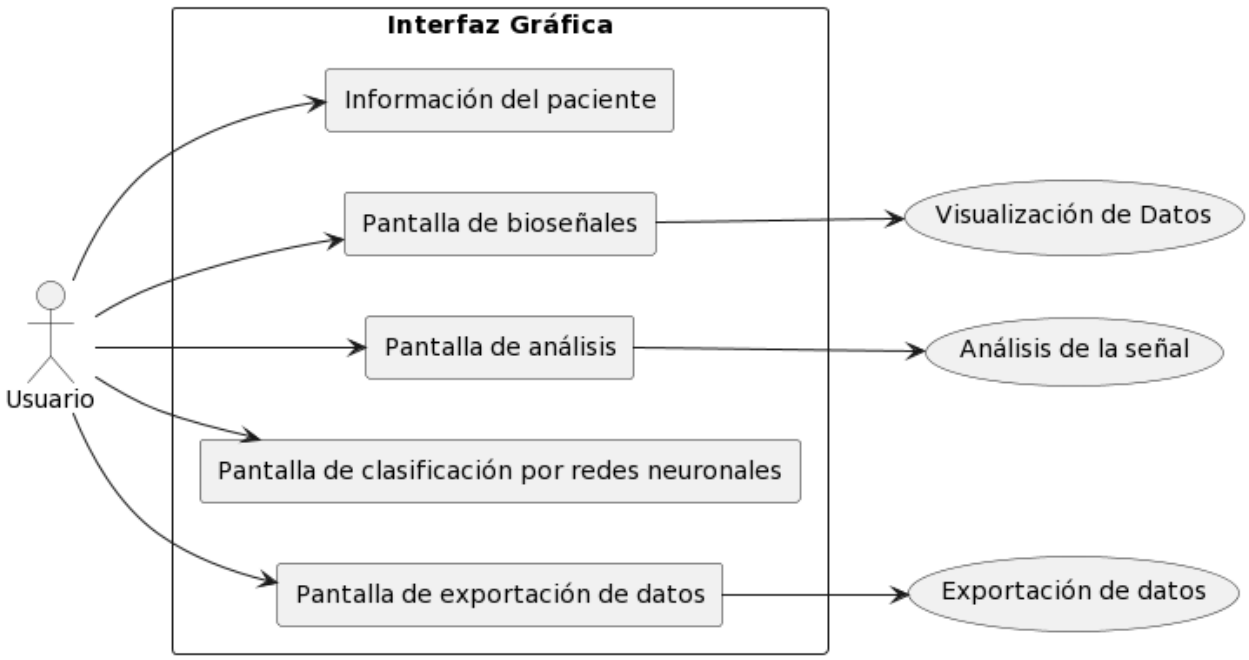


Figura 3.3: Diagrama de Casos de Uso del sistema.

3.3. Propuesta de Solución

En la figura 3.4 imagen se describe el proceso propuesto para la construcción del sistema capaz de analizar señales ECG y poder clasificarlas.

De acuerdo con el contenido de la literatura se determina lo siguiente:

1. Arduino DUE es la placa de desarrollo de microcontroladores óptimo para el desarrollo del sistema porque cuenta con un procesador ARM Cortex-M3 de 32 bits y una velocidad de reloj de 84 MHz, a diferencia de otros modelos como Arduino Uno y Mega, además tiene pines de entrada/salida digital (de los cuales 12 pueden ser usados como salidas PWM), entradas analógicas, puertos serie de hardware (UART), puertos de comunicación I2C, puertos SPI, un puerto USB nativo, una conexión Ethernet RJ45 y una conexión para tarjetas microSD. Asimismo, cuenta con una conexión de programación mediante el uso de un programador externo.es importante mencionar que es compatible con la mayoría de los shields de Arduino y puede programarse utilizando el IDE de Arduino.
2. El amplificador AD8232 es una opción popular para adquirir señales electrocardiográficas (ECG) debido a su alta precisión y capacidad para amplificar señales de baja amplitud, además se caracteriza por su bajo consumo de energía y su pequeño tamaño, lo que lo hace ideal para aplicaciones portátiles y de

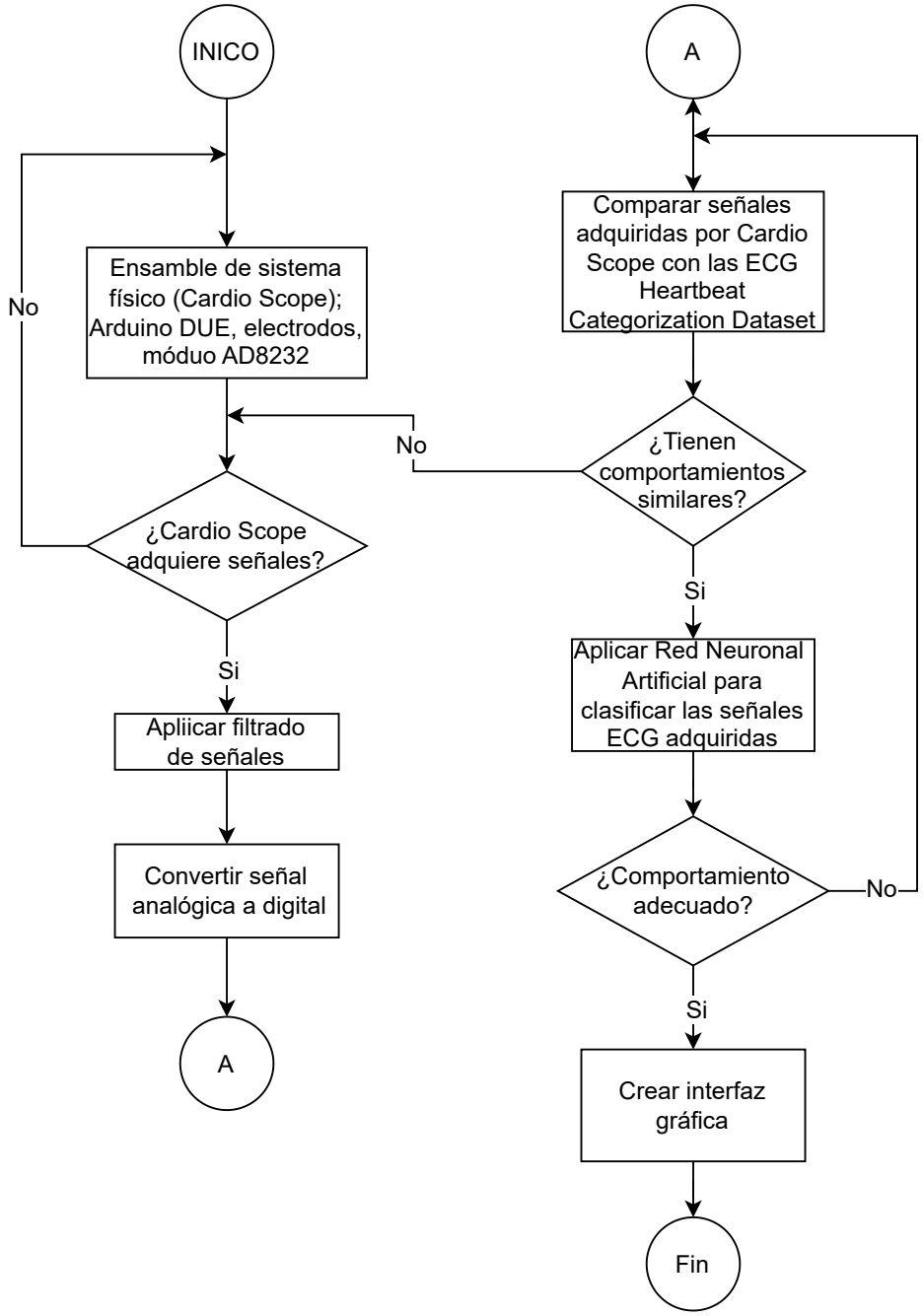


Figura 3.4: Diagrama de flujo que describe el proceso de construcción del sistema.

bajo consumo, como monitores de salud y dispositivos de diagnóstico, de igual forma permite aplicar un filtro de paso bajo integrado para eliminar el ruido de alta frecuencia y un circuito de protección contra interferencias electromagnéticas.

3. Las redes neuronales pueden ser entrenadas con grandes conjuntos de datos de ECG y pueden aprender a identificar patrones específicos de señales que corresponden a diferentes afecciones cardíacas. Una vez entrenadas, las redes neuronales pueden clasificar nuevas señales ECG con una alta precisión, lo que las hace útiles en aplicaciones clínicas y de diagnóstico.

3.4. Cronograma de Actividades

Finalmente la figura 3.5 muestra el cronograma de las actividades planteadas para el desarrollo e implementación del sistema.

ACTIVIDAD	ABRIL				MAYO			
	1	2	3	4	1	2	3	4
1. Sprint de planificación.								
1.1. Identificación de objetivos del proyecto.								
1.2. Identificación de los requerimientos del sistema.								
1.3. Identificación de los recursos necesarios.								
1.4. Diseño del diagrama de bloques del sistema.								
1.5. Selección de los componentes del sistema.								
1.6. Diseño de la arquitectura del software.								
2. Sprint daily.								
2.1. Reuniones cliente - equipo.								
3. Sprint backlog.								
3.1. Instalación y configuración del software necesario.								
3.2. Implementación del algoritmo de procesamiento de señales ECG.								
3.3. Implementación de las técnicas de machine learning.								
3.4. Verificación del software implementado.								
4. Sprint review.								
4.1. Integración del hardware y software.								
4.1.1. Pruebas unitarias del hardware y software.								
4.1.2. Pruebas de integración del sistema completo.								
5. Sprint retrospective.								
5.1. Documentación del proyecto.								
5.2. Implementación del sistema en escenarios reales.								

Figura 3.5: Cronograma de Actividades

3.5. Desarrollo e Implementación del Proyecto

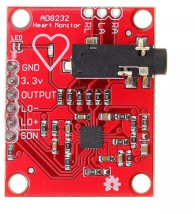
En esta sección se describe el proceso de desarrollo del sistema propuesto, el cual tiene como objetivo la obtención de señales electrocardiográficas utilizando Arduino, para posteriormente enviar estas señales al entorno de Matlab, en donde se filtraran por medio de un filtro de suavizado. Una vez filtrada la señal, estas serán clasificadas haciendo uso de una red neuronal recurrente, para finalmente integrar las funciones de adquisición, filtrado y clasificación dentro de una interface de usuario que facilite esta tarea. La fases para el desarrollo sen encuentran divididas de tal forma que se aborda en detalle procurando mostrar todas la partes que conforman este sistema.

3.5.1. Configuración de Hardware: Conexiones Arduino Uno y Módulo AD8232

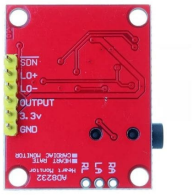
El primer paso para la construcción de este sistema involucra la conexión de los elementos físicos (hardware) el cual consta de dos dispositivos. El primero es el módulo amplificador **AD8232**, el cual está diseñado específicamente para para amplificar y filtrar las señales débiles provenientes del corazón. Dentro de los filtros

que emplea para se encuentran filtros **pasa altas** y **pasa bajas**, el objetivo de estos es eliminar el ruido y las interferencias no deseadas. A su vez, el módulo AD8232 analiza la señal obtenida para calcular la frecuencia cardíaca.

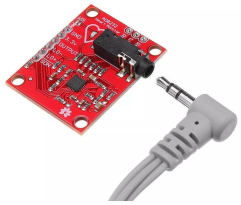
La figura 3.6 muestra un módulo AD8232, así como las partes que lo conforman.



(a) Vista Frontal



(b) Vista Trasera



(c) Conector jack 3.5mm

(d) Electrodos

(e) Parches adhesivos

(f) Módulo AD8232 completo

Figura 3.6: Componentes de un módulo amplificador AD8232

El segundo y ultimo componente de hardware es la placa Arduino uno, la figura 3.7 muestra el diagrama de conexiones entre el módulo AD8232 y la placa Arduino UNO, en donde se pueden destacra algunas de las conexiones:

- Conexión de Alimentación

Se conecta el pin **3.3V** del AD8232 al pin **3.3V** de la placa.
Se conecta el pin **GND** del módulo al pin **GND** de la placa.

- Conexiones de Señal

Se conecta el pin **OUTPUT** del módulo AD8232 al pin analógico de entrada de la placa, en este caso se usó el pin **A0**. Se conecta el pin **LO-** (referencia negativa) del módulo al pin **11** de la placa. Se conecta el pin **LO+** (amplificador derecho) del módulo AD8232 al pin **10** de la placa.

3.5.2. Transferencia de Señales ECG desde Arduino a Matlab

Una vez realizadas la conexiones físicas entre los componentes (hardware), es momento de adquirir algunas señales para comprobar su funcionamiento, para esta primera adquisición se utilizará Arduino IDE, para esto se escribió el código de adquisición que a grandes rasgo realiza lo siguiente:

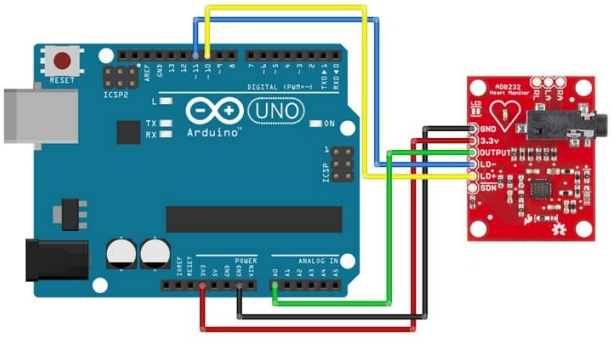


Figura 3.7: Diagrama de conexiones módulo AD8232 y Arduino UNO

- Inicializar comunicación en serie, con el objetivo de enviar datos por medio del puerto serial.
- Se configuran los pies **10** y **11** como entradas digitales, estos serán utilizados para detectar señales.
- Se realiza la lectura del valor analógico **A0**, mismo que será enviado a traves del puerto serial.

La figura 3.8 muestra una primera adquisición por medio del Serial Monitor integrado en Arduino IDE.

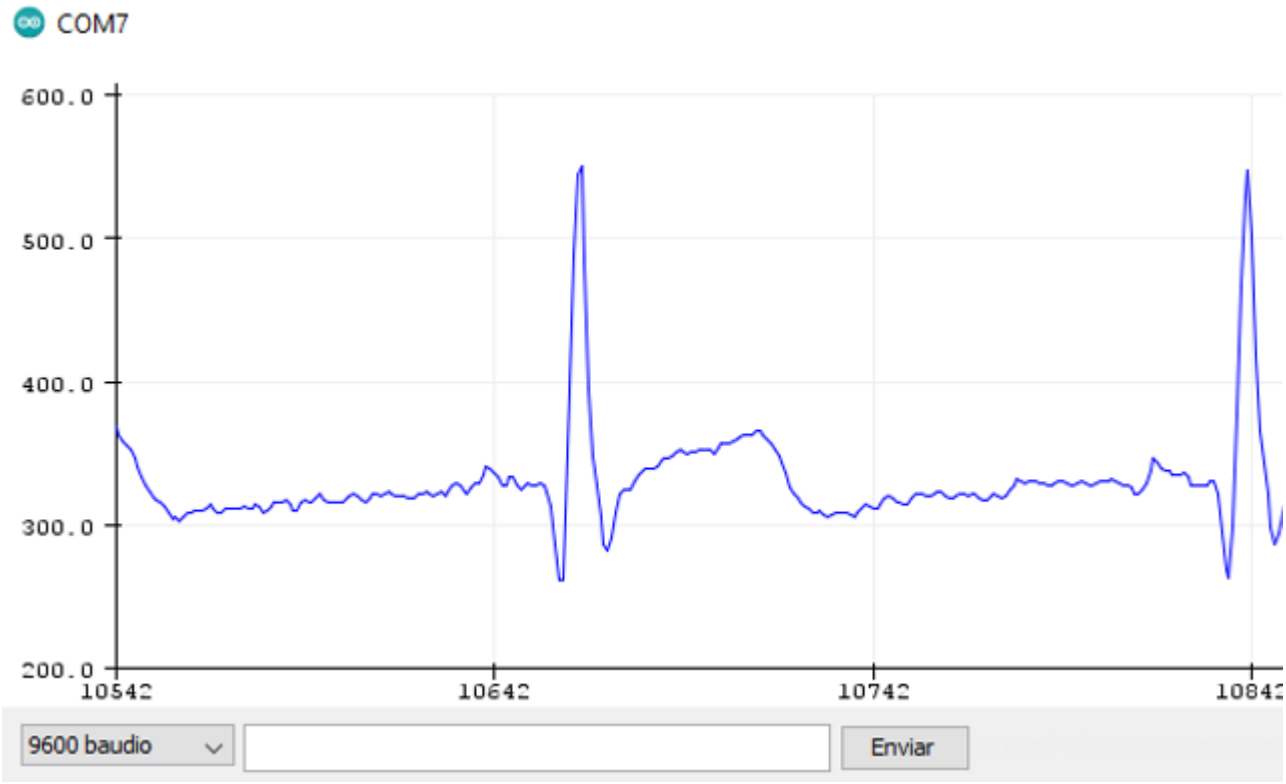


Figura 3.8: Señal ECG mostrada desde Serial Monitor de Arduino IDE

Una vez comprobado el funcionamiento de la adquisición de señales por medio de Arduino IDE, es momento de enviar estas señales a Matlab, para esto se desarrolló el código para el envió de datos el cual realiza lo siguiente:

- Configuración inicial:

- Se utilizan los comandos `clear all` y `clc` para limpiar el espacio de trabajo y la ventana de comandos respectivamente.
- Se establece la conexión con la placa Arduino Uno a través del puerto serial utilizando el objeto `serialport`. En este caso, se utiliza el puerto COM1 y una velocidad de transmisión de 9600 baudios.

- Configuración de la adquisición de datos:
  - Se definen parámetros como el tiempo de muestreo (`sampleTime`), el número de muestras (`numSamples`), el número de muestras por marco (`samplesPerFrame`), entre otros.
  - Se crea una matriz para almacenar los datos adquiridos.
- Inicialización y bucle de lectura de datos:
  - Se inicializan los datos del búfer y la variable `i` para contar el número de muestras adquiridas.
  - Dentro del bucle, se lee el valor del sensor a través del puerto serial y se almacena en la variable `sensorValue`.
  - Se guarda el tiempo y el valor del sensor en la matriz.
  - Se verifica si se ha alcanzado el número de muestras por marco y se importan los datos al espacio de trabajo.
  - Se agrega un retraso (`pause(sampleTime)`) para evitar un desbordamiento de datos.
- Cierre y visualización de datos:
  - Se cierra el puerto serial.
  - Se asigna la matriz `x` al espacio de trabajo para poder acceder a los datos adquiridos.
  - Se gráfica la matriz `x`, mostrando el tiempo en el eje x y el valor del sensor en el eje y.

La imagen 3.9 muestra el resultado del envió de datos desde Arduino a Matlab.

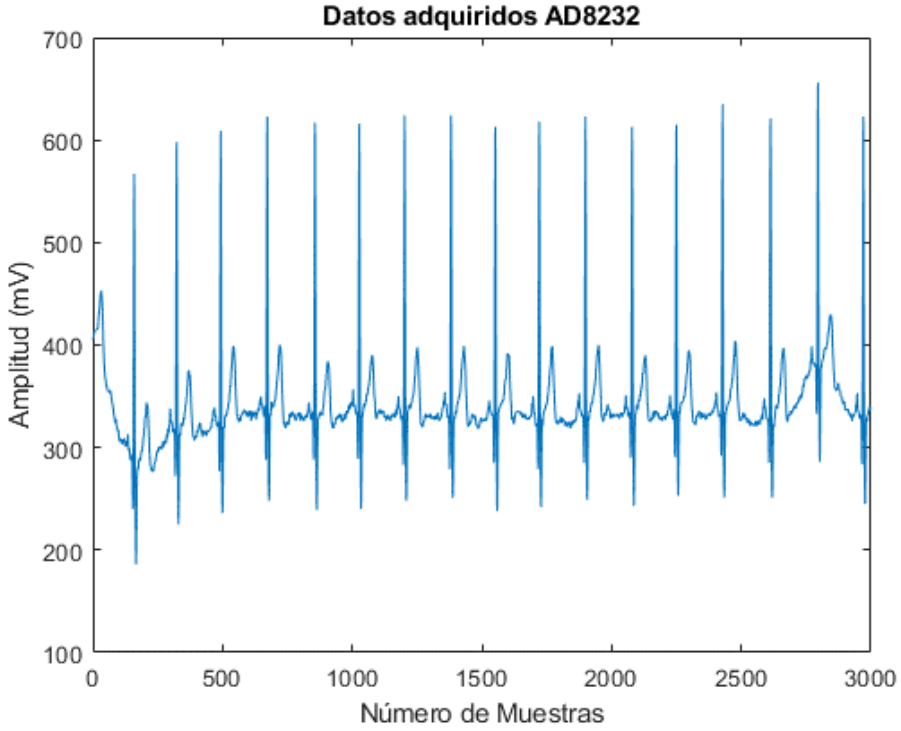


Figura 3.9: Envió de datos y almacenamiento en Matlab

Como se puede observar en la imagen anterior se aprecian los componentes de una ECG, por lo tanto es momento de implementar el filtro y diseñar la red neuronal recurrente para la clasificación, estos últimos serán desarrollados en Matlab, por lo tanto ya no se usará Arduino IDE.

3.5.3. Realce de Características de la Señal ECG mediante Filtro

Una vez que se cuentan con las señales en el entorno de trabajo de Matlab, se pueden aplicar el filtro cuyo objetivo es suavizar el ruido presente en la señal. A grandes rasgos, el filtro de media móvil es una técnica de filtrado ampliamente utilizada en areas como ciencia de datos en donde se busca obtener una representación más clara de las frecuencias subyacentes de la señal. Tradicionalmente, se establece un tañamo de ventana el cual permite calcular el promedio de un vector de muestras y remplaza el valor original por este promedio.

Para la aplicación del filtro se empleó la función `smoothdata` en conjunto con la el método `movmean` esto para permitir a Matlab escoger el tamaño de ventana optimo para el filtro.

La figura 3.10 muestra el resultado de la aplicación del filtro de media móvil para suavizar el ruido de la señal.
Se puede observar que antes del filtro algunas ondas de la señal ECG cuentan con ruido, una vez aplicado el filtro se observan de manera más clara los complejos y ondas de la señal.

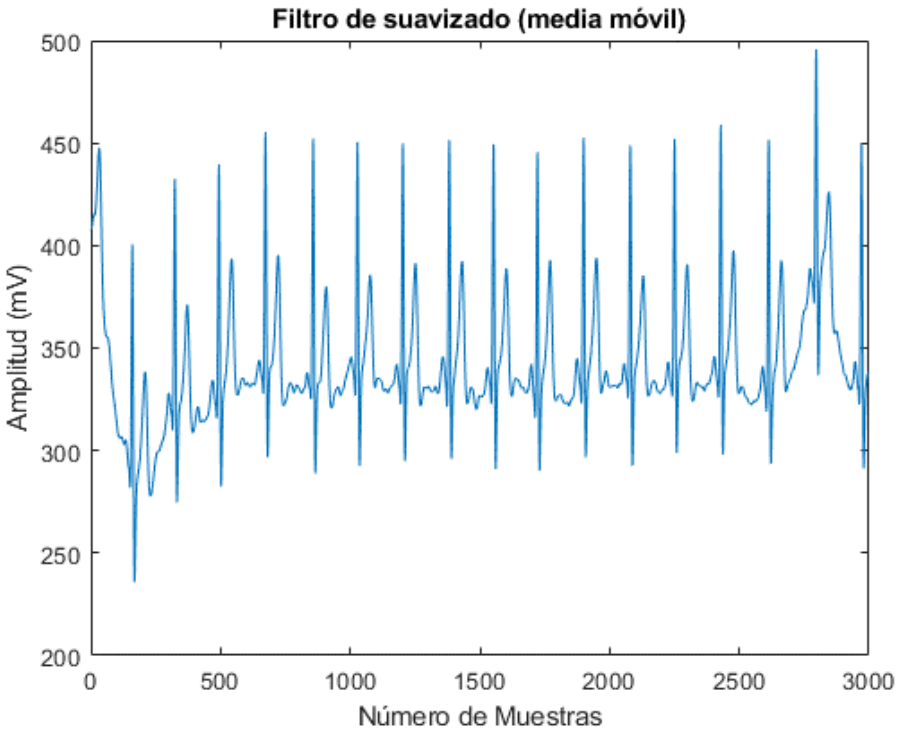


Figura 3.10: Resultados del filtro de media móvil

3.5.4. Diseño de la Red Neuronal Recurrente

Una vez comprobado el funcionamiento del filtro de media móvil, el siguiente paso es crear el modelo de clasificación, para esto se optó por implementar una red neuronal recurrente agregando **Unidades Recurrentes con Compuertas**, las cuales, como ya se mencionó en el análisis de la literatura, por su simplicidad y eficiencia con respecto a otras como las **LSTM** son la mejor opción para cumplir los objetivos del proyecto, dentro de los cuales de determinó clasificar las señales en **Normal** y **Anormal**.

Descarga del dataset

El primer paso para la construcción del modelo de clasificación es contar con un conjunto de datos que permitan a la red neuronal entender las relaciones entre los registros **Normales** y **Anormales**. El dataset elegido es parte del repositorio de Physionet, el cual lleva por nombre **AF Classification from a Short Single Lead ECG Recording: The PhysioNet/Computing in Cardiology Challenge 2017**, el cual consta de una serie de registros de electrocardiograma de **una derivación**, la razón por la cual se eligió este dataset es por su compatibilidad con el modulo amplificador AD8232, ya que ambos cuentan con una derivación.

Preprocesamiento

Una vez descargado el dataset, se puede observar que este cuenta con un total de 5788 registros, catalogados como **Normal** y **Anormal**.

La tabla 3.1 muestra la estructura del dataset, contemplando el total de registros, así como sus categorías.

Tabla 3.1: Estructura del dataset

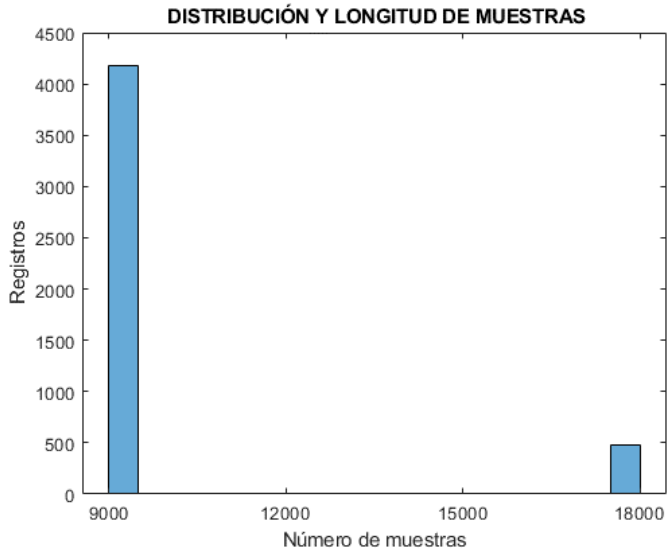
TOTAL DE REGISTROS	5788
NORMAL	5050
ANORMAL	738

Una vez explorado el dataset, se puede observar que la distribución de muestras se encuentra divida en dos categorías, aquellas con 9000 muestras y aquellas con 18000 muestras, es decir no se tiene una distribución uniforme en cuanto al numero de muestras presentes en el dataset. Para forzar un distribución uniforme, se optó por realizar una segmentación de señales, esta técnica consiste en dividir las señales con un mayor número de muestras tantas veces sea necesario hasta lograr señales con el mismo número de muestras. en este caso las señales con un mayor número de muestras son de 18000, mientras que las más pequeñas de 9000, por lo tanto el factor de segmentación es 2, por lo tanto las señales con 18000 muestras son divididas entre dos y agregadas al conjunto mayoritario, dando como resultado una distribución uniforme en cuanto al numero de muestras presentes en cada señal. Todas aquellas señales menores a 9000 o 18000 son descartadas.

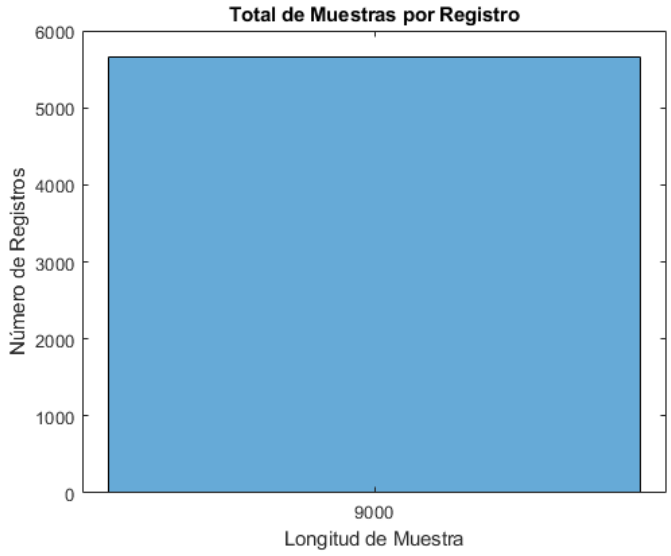
La figura 3.11 *(a)* muestra la distribución inicial de muestras por señal, en donde se puede observar las dos categorías de muestras.

La figura 3.11 *(b)* muestra el resultado de la segmentación de señales, se puede apreciar que ahora se

cuenta con una distribución uniforme en cuanto al número de señales por registro.



(a) Distribución de muestras por registro



(b) Resultado de la segmentación de señales.

Figura 3.11: Segmenatación de señales

Una vez que se tiene una distribución uniforme, se pueden aplicar análisis descriptivo para conocer las categorías que conforman este nuevo dataset.

la tabla 3.2 muestra el total de señales, así como el número de señales catalogadas como **Normales** y **Anormales**.

Tabla 3.2: Total de registros y categorias

TOTAL DE REGISTROS	5655
NORMAL	4937
ANORMAL	718

Balanceo de clases

Una vez que se cuenta con una distribución uniforme es necesario crear los conjuntos de entrenamiento y validación, este paso es muy importante, ya que de no hacerlo se corre el riesgo de generalizar las respuestas del modelo, en otras palabras, si el modelo es entrenado con clases desbalanceadas, este aprenderá a clasificar todas las instancias como pertenecientes a la clase dominante.

El balanceo propuesto en este caso esta dividido en tres categorías.

1. 70 % entrenamiento y 30 % validación
2. 80 % entrenamiento y 20 % validación
3. 90 % entrenamiento y 10 % validación

Una vez obtenido el numero de registros por clase es necesario balancear las clases, esto se consigue aplicando **sobremuestreo**, esta es una técnica que consiste en multiplicar tantas veces sea necesario una la case minoritaria con el objetivo de igualar el numero de instancias en las clase mayoritaria. En este caso especifico, primero se dividió el numero de instancias de la clase minoritaria entre las instancias de la clase mayoritaria, dando un factor de multiplicación de $6.7 \simeq 7$, el cual es usado para multiplicar las instancias minoritarias, dando como resultado el balanceo de clases.

La tabla 3.3 Muestra el resultado de los conjuntos de entrenamiento, así como del balanceo de clases.

Tabla 3.3: Estimación de conjuntos de entrenamiento/validación y balanceo de clases

70 % Entrenamiento		80 % Entrenamiento		90 % Entrenamiento	
Normal	3456	Normal	3949	Normal	4443
Anormal	3456	Anormal	3949	Anormal	4443
30 % Validación		20 % Validación		10 % Validación	
Normal	1481	Normal	987	Normal	494
Anormal	1481	Anormal	987	Anormal	494

Patron a Reconocer

Una vez establecidos los conjuntos de entrenamiento y validación es necesario identificar el patrón que la red neuronal deberá reconocer para determinar si una señal es **Normal** o **Anormal**.

La figura 3.12 muestra la forma de onda de un electrocardiograma con un ritmo normal, en este se pueden apreciar con claridad las ondas **P**, ondas **T** y el complejo **QRS**.

Por otro lado, la figura 3.13 muestra la forma de onda de una electrocardiograma en donde no se aprecia la onda **P**, esto indica un comportamiento anormal, el cual puede indicar algún tipo de anomalía en el corazón.

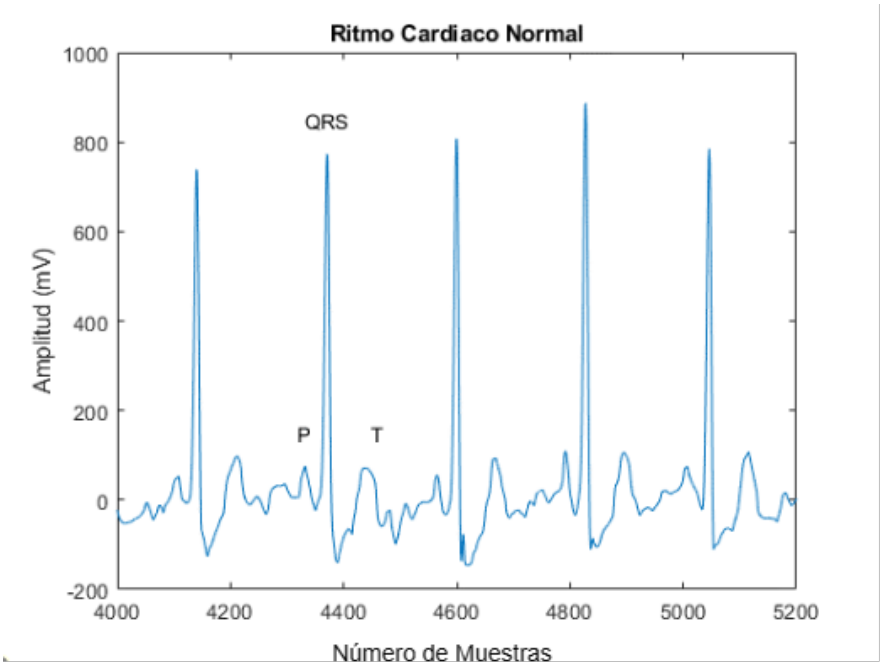


Figura 3.12: Ritmo cardíaco normal

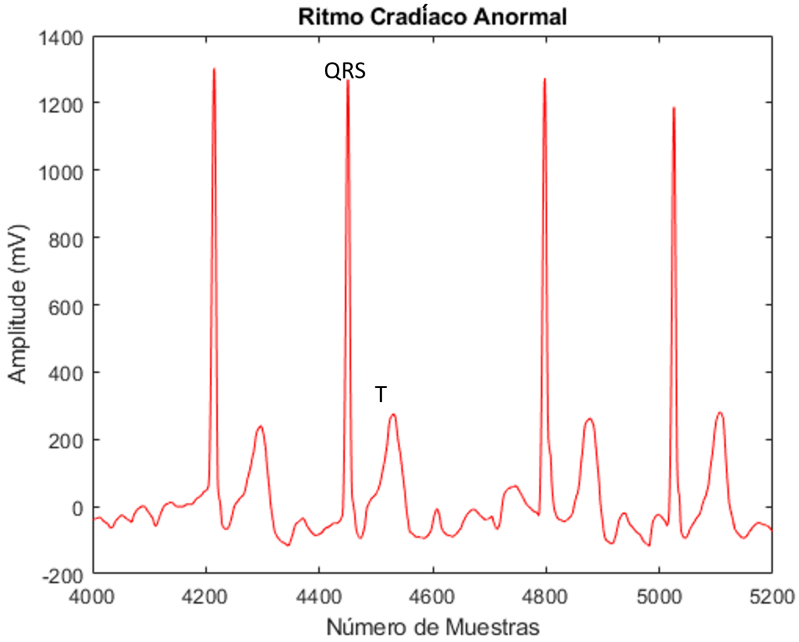


*Figura 3.13: Ausencia de la onda **P** en el electrocardiograma*

Topología de la Red Neuronal Recurrente

Una vez definidos los aspectos anteriores, se debe especificar la topología de la red neuronal que se utilizará en el proceso de clasificación.

La figura 3.14 muestra la topología de la red neuronal recurrente empleada en este caso. En la cual se pueden apreciar los siguiente elementos:

1. Capa de Entrada:
   Capa de entrada de tipo secuencial, permite la entrada de una secuencia de datos, en este caso una señal ECG.
2. Capa GRU
   Contiene 16 Unidades Recurrentes con Compuertas. Además utilizan la función de transferencia Tangente Hiperbólica
3. Primera capa oculta
   Cuenta con 16 neuronas y utiliza la función de transferencia lineal.
4. Segunda capa oculta
   Cuanta con dos neuronas, ademas emplea la función lineal como función de transferencia.
5. Capa Softmax
   Normaliza las salidas de la capa anterior y las interpreta como probabilidades.
6. Capa de Clasificación
   Es la capa de clasificación final, es usada para calcular la perdida y el rendimiento del modelo.

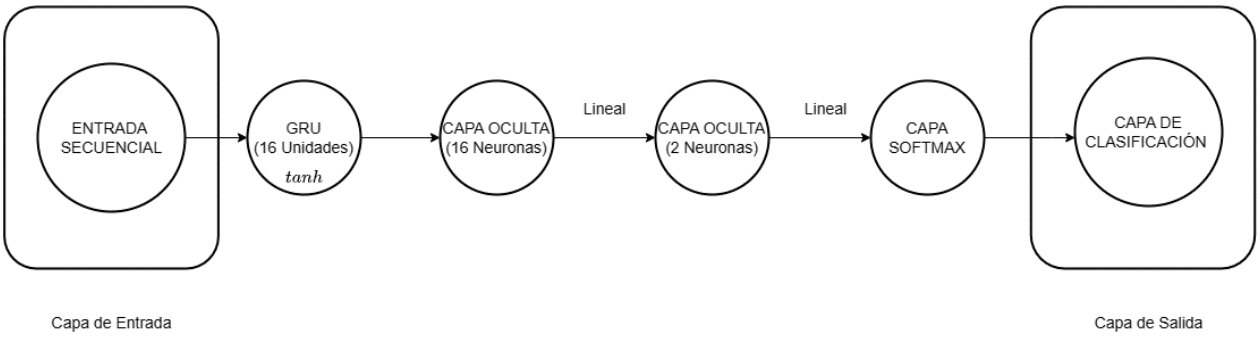


Figura 3.14: Topología de la Red Neuronal Recurrente

Interface Gráfica

Para facilitar y en cierta forma acelerar el proceso de adquisición de señales se desarrollo una interface de usuario por medio de Matlab App Designer. La figura 3.15 muestra la interface de usuario creada.

La interface esta compuesta por tres secciones principales.

1. Datos del paciente
2. Visualización de la forma de onda
3. Estado o clasificación

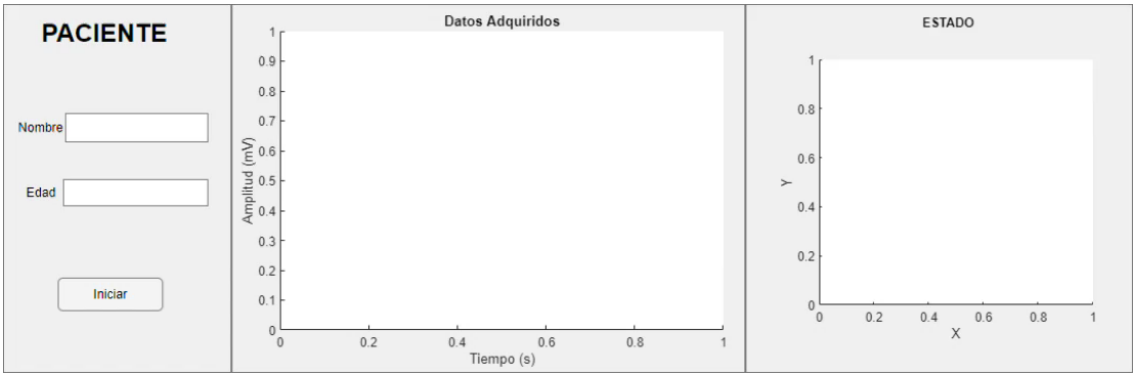


Figura 3.15: Interface de usuario

3.5.5. Experimentación

Una vez implementada la Red Neuronal Recurrente, se procede a la fase de experimentación en donde se emplearon los conjuntos de entrenamiento y validación antes definidos. En esta fase se realizaron iteraciones de entrenamiento, asi como el ajuste de los parámetros de la red para mejorar su precisión.

La tabla 3.4 muestra los resultados para el entrenamiento y validación del conjunto 70 % entrenamiento y 30 % validación

- 70 % Entrenamiento - 30 % Validación

Tabla 3.4: Experimentación conjuntos (70-30)

Intento	Epocas	Iteraciones	GRU	1 C_O	2 C_O	softmax	clasificación	VP	FP	FN	VN	Exactitud
1	1	414	16	16	2	Si	Si	2954	567	2866	590	50.79
2	10	4140	16	16	2	Si	Si	2429	1092	1869	1587	57.56
3	30	12420	16	16	2	Si	Si	2170	791	2161	1295	57.37
4	10	4140	32	32	2	Si	Si	2730	791	2161	1295	57.68
5	20	8280	64	64	2	Si	Si	1680	1841	1216	2240	56.18

La tabla 3.5 muestra los resultados para el entrenamiento y validación del conjunto 80 % entrenamiento y 20 % validación

- 80 % Entrenamiento - 20 % Validación

Tabla 3.5: Experimentación conjuntos (70-30)

INTENTO	Epocas	Iteraciones	GRU	1 C_O	2 C_O	softmax	Clas_Layer	VP	FS	FN	VN	Exactitud
1	5	2385	16	16	2	SI	SI	3367	651	3024	926	53.87
2	10	4470	16	16	2	SI	SI	3535	438	3115	835	55.15
3	1	477	16	16	2	SI	SI	98	3920	115	3835	49.35
4	20	9540	16	16	2	SI	SI	1498	2520	957	2993	56.36
5	30	14310	16	16	2	SI	SI	3507	511	3129	821	54.31
6	30	14310	16	16	2	SI	SI	2793	1225	1901	2049	60.76
7	30	14310	20	20	2	SI	SI	2933	1085	2192	1758	58.87
8	30	14310	16	16	2	SI	SI	1813	2205	956	2994	60.32
9	40	19080	16	16	2	SI	SI	1631	2387	1058	2892	56.76
10	30	14310	32	32	2	SI	SI	2275	1743	1793	2157	55.62
11	10	4770	16	32	2	SI	SI	3311	707	2767	1183	56.40
12	20	9540	16	32	2	SI	SI	1911	2107	1256	2694	57.79

La tabla 3.6 muestra los resultados para el entrenamiento y validación del conjunto 90 % entrenamiento

y 10 % validación

- 90 % Entrenamiento - 10 % Validación

Tabla 3.6: Experimentación conjuntos (90-10)

Intento	Epocas	Iteraciones	GRU	1 C_O	2 C_O	softmax	Capa_clasificación	VP	FP	FN	VN	EXACTITUD
1	1	531	256	256	2	Si	Si	378	4060	272	4166	51.19423164
2	10	5310	256	256	2	Si	Si	189	4249	126	4312	50.70977918
7	10	5310	16	16	2	Si	Si	3374	1064	2883	1555	55.53177107
8	20	10620	16	16	2	Si	Si	3535	903	2923	1515	56.89499775
9	30	15930	16	16	2	Si	Si	2737	1701	2130	2308	56.83866607
10	10	5310	32	32	2	Si	Si	1470	2968	1223	3215	52.78278504
11	100	53100	32	32	2	Si	Si	3416	1022	2940	1498	55.36277603
12	10	79200	16	16	2	Si	Si	2065	2373	787	3651	64.39837765
13	10	79200	16	16	2	Si	Si	392	4046	74	4364	53.58269491

Como se puede apreciar el conjunto de entrenamiento que obtuvo una mejor precision fue el conjunto de 90 % Entrenamiento y 10 % Validación, ya que se obtuvo un 64.39 % de precisión.

Capítulo 4

Resultados

Como ya se mencionó la **Exactitud** alcanzada por el modelo en la fase de entrenamiento es del 64.39 %, en esta sección se abordara a detalle el resultado obtenido por el modelo.

La figura 4.1 muestra la matriz de confusión del modelo de clasificación.

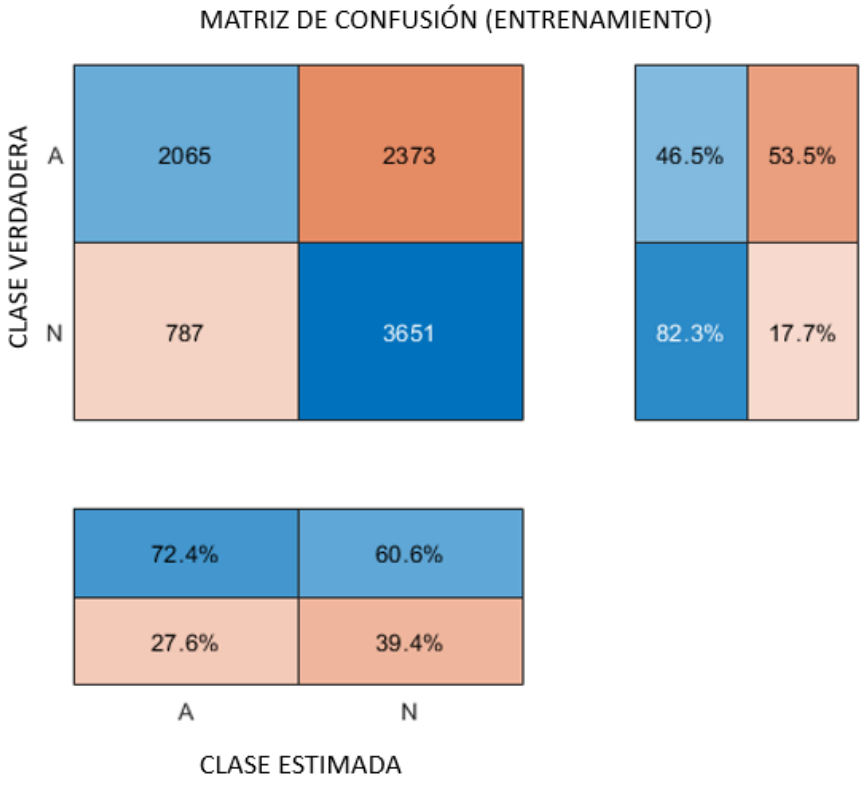


Figura 4.1: Matriz de confusión del modelo (Entrenamiento)

La Exactitud del modelo esta dada por la formula 2.17

$$E = \frac{2065 + 3651}{2065 + 3651 + 787 + 2373} \times 100 = 64.3983 \tag{4.1}$$

Es decir, el modelo fue capaz de clasificar un 64.39 % de las señales ECG en la categoría correcta en el conjunto de datos de entrenamiento.

La Sensibilidad del modelo esta dada por 2.18

$$S = \frac{2065}{2065 + 787} \times 100 = 34.2795 \tag{4.2}$$

Es decir, el modelo ha sido capaz de clasificar 34.27 % de las muestras positivas en relación con el total de muestras positivas del conjunto.

La Especificidad del modelo esta dada por 2.19

$$Es = \frac{3651}{3651 + 2373} \times 100 = 60.6075 \tag{4.3}$$

En otras palabras, el modelo clasificó un 60.6075 % de muestras negativas en relación con el total.

En cuanto a los resultados de la **Validación** alcanzada por el modelo es del 61.93 %, en esta sección se abordara a detalle el resultado obtenido por el modelo.

La figura 4.2 muestra la matriz de confusión del modelo de clasificación.

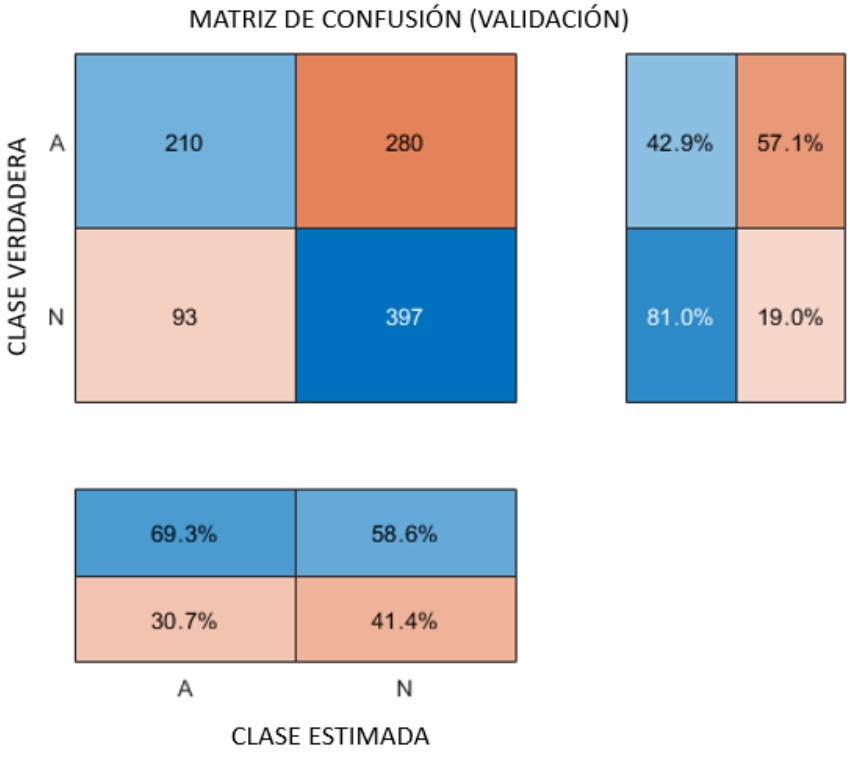


Figura 4.2: Matriz de confusión del modelo (Validación)

- Exactitud

$$E = \frac{210 + 397}{210 + 397 + 93 + 280} \times 100 = 61.9387 \tag{4.4}$$

Es decir, el modelo fue capaz de clasificar un 61.93 % de las señales ECG en la categoría correcta en el conjunto de datos de entrenamiento.

- Sensibilidad

$$S = \frac{210}{210 + 93} \times 100 = 69.3096 \tag{4.5}$$

Es decir, el modelo ha sido capaz de clasificar 69.30 % de las muestras positivas en relación con el total de muestras positivas del conjunto en la fase de validación.

- Especificidad

$$Es = \frac{397}{397 + 280} \times 100 = 58.6410 \tag{4.6}$$

En otras palabras, el modelo clasificó un 58.64 % de muestras negativas en relación con el total en la fase de validación.

Capítulo 5

Discusión

Tal y como se estableció en los objetivos, se utilizó un filtro media móvil para atenuar el ruido y resaltar las características de la señal, una vez finalizada la fase de experimentación y obtenidos los resultados se determinó que este filtro no es suficiente para extraer características relevantes en la señal, debido a que las señales obtenidas por el módulo AD8232, por ende se plantea aplicar transformada de Fourier o de Wavelet para analizar las señales del conjunto de datos tanto de entrenamiento como de validación, dado que estas dos técnicas de filtrado ofrecen ventajas como la preservación de información de alta frecuencia, adaptabilidad a diferentes escalas, eficiencia computacional y supresión selectiva de ruido en comparación con el filtro aplicado.

En cuanto a la evaluación del modelo de clasificación para señales ECG normales y anormales ha mostrado un porcentaje bajo en cuanto a exactitud, es decir, la proporción de todas las predicciones correctas realizadas, ya que solo alcanzó un 64.39 % al realizar esta tarea, pero se obtuvo un mayor porcentaje ("69.3069") al evaluar la sensibilidad, métrica que mide la capacidad del modelo para identificar correctamente los casos positivos que clasificó la red neuronal recurrente.

Como trabajo a futuro se plantea aplicar ya sea transformada de Fourier o de Wavelet para analizar los datos adquiridos por el módulo AD8232 y las señales del conjunto de datos tanto de entrenamiento como de validación, ya que de acuerdo con algunos autores al convertir una señal en función del tiempo en una representación en función de la frecuencia permitirá identificar patrones característicos. Por lo tanto, utilizar este tipo de transformadas tiene potencial para obtener información relevante para futuros proyectos.

Capítulo 6

Anexos

- Código para adquisición de datos desde Arduino UNO

```
void setup(){
  Serial.begin(9600);
  pinMode(10, INPUT); +
  pinMode(11, INPUT);
}
void loop() {
if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
Serial.println('!');
}
else{
// Imprimir la lectura del puerto A0
Serial.println(analogRead(A0));
}
//Espere un poco para evitar que los datos en serie se saturen
delay(1);
}
```

- Código para envío de datos desde Arduino IDE a MATLAB

```
clear all
clc

s = serialport("COM1", 9600);

% Muestra de tiempo
sampleTime = 0.1; % segundos
samplesPerFrame = 1;
numSamples = 3000;
headerRow = {'Tiempo (s)', 'Valor'};
dataFormat = {'%f', '%f'};
delimiter = ',';


configureTerminator(s, "LF");
flush(s);

```

```
ACR = zeros(numSamples, 2);

i = 0;


while s.Status == "open" && i < numSamples
    sensorValue = readline(s);
    sensorValue = str2double(sensorValue);

    i = i + 1;
    ACR(i,:) = [i*sampleTime, sensorValue];

    if mod(i, samplesPerFrame) == 0 && i < numSamples
        assignin('base', 'data', ACR(1:i,:));
    end

    % Delay para evitar desbordamiento
    pause(sampleTime);
end

fclose(s);

assignin('base', 'data', ACR);

plot(ACR(:,1), ACR(:,2));

title('AD8232 Sensor Data');
xlabel('Time (s)');
ylabel('Sensor Value');
```

- Filtro de Suavizado (Media Móvil)

```
function y = preprocess(x)

y = smoothdata(x,'movmean');

y = smoothdata(y,'movmean');
```

- Preprocesamiento y Red Neural Recurrente

```
if ~isfile('PhysionetData.mat')
    ReadPhysionetData
end
load PhysionetData

Signals(1:10)
Labels(1:10)
summary(Labels)
L = cellfun(@length,Signals);
h = histogram(L);

xticks(0:3000:18000);
xticklabels(0:3000:18000);
title('')
xlabel('Longitud de Muestra')
```

```
ylabel('Cantidad')


normal = Signals{1};
aFib = Signals{4};

subplot(2,1,1)
plot(normal)
title('Ritmo Cardiaco Normal')
xlim([4000,5200])
ylabel('Amplitud (mV)')
text(4330,150,'P','HorizontalAlignment','center')
text(4370,850,'QRS','HorizontalAlignment','center')

subplot(2,1,2)
plot(aFib)
title('Ritmo Cardicaco Anormal')
xlim([4000,5200])
xlabel('Muestras')
ylabel('Amplitud (mV)')
hold off
%text(4330,150,'P','HorizontalAlignment','center')
%text(4370,850,'QRS','HorizontalAlignment','center')

[Signals,Labels] = segmentSignals(Signals,Labels);
Signals(1:5)

L = cellfun(@length,Signals);
h = histogram(L);
hold off
xticks(0:3000:18000);
xticklabels(0:3000:18000);
title('')
xlabel('Longitud de Muestra')
ylabel('Cantidad')

summary(Labels)

afibX = Signals(Labels=='A');
afibY = Labels(Labels=='A');

normalX = Signals(Labels=='N');
normalY = Labels(Labels=='N');


[trainIndA,~,testIndA] = dividerand(718,0.9,0.0,0.1);
[trainIndN,~,testIndN] = dividerand(4937,0.9,0.0,0.1);

XTrainA = afibX(trainIndA);
YTrainA = afibY(trainIndA);

XTrainN = normalX(trainIndN);
YTrainN = normalY(trainIndN);

XTestA = afibX(testIndA);
```

```
YTestA = afibY(testIndA);

XTestN = normalX(testIndN);
YTestN = normalY(testIndN);

XTrain = [repmat(XTrainA(1:634),7,1); XTrainN(1:4438)];
YTrain = [repmat(YTrainA(1:634),7,1); YTrainN(1:4438)];

XTest = [repmat(XTestA(1:70),7,1); XTestN(1:490)];
YTest = [repmat(YTestA(1:70),7,1); YTestN(1:490);];

layers = [ ...
 sequenceInputLayer(1)
 gruLayer(256, 'OutputMode', 'last')
 fullyConnectedLayer(256)
 fullyConnectedLayer(2)
 softmaxLayer
 classificationLayer
];

options = trainingOptions('adam', ...
 'MaxEpochs',1, ...
 'MiniBatchSize', 150, ...
 'InitialLearnRate', 0.01, ...
 'SequenceLength', 1000, ...
 'GradientThreshold', 1, ...
 'ExecutionEnvironment',"auto",...
 'plots','training-progress', ...
 'Verbose',false);


net = trainNetwork(XTrain,YTrain,layers,options);

trainPred = classify(net,XTrain,'SequenceLength',1000);


GRUAccuracy = sum(trainPred == YTrain)/numel(YTrain)*100

figure
confusionchart(YTrain,trainPred,'ColumnSummary','column-normalized',...
 'RowSummary','row-normalized','Title','Matriz de Confusion');

testPred = classify(net,XTest,'SequenceLength',1000);


GRUAccuracy = sum(testPred == YTest)/numel(YTest)*100
figure
confusionchart(YTest,testPred,'ColumnSummary','column-normalized',...
              'RowSummary','row-normalized','Title','Matriz de Confusion');
```

- Interface de Usuario

```
classdef app2 < matlab.apps.AppBase

    % Properties that correspond to app components
```

```
    properties (Access = public)
        UIFigure              matlab.ui.Figure
        GridLayout            matlab.ui.container.GridLayout
        LeftPanel             matlab.ui.container.Panel
        NombreEditField       matlab.ui.control.EditField
        NombreEditFieldLabel  matlab.ui.control.Label
        EdadEditField         matlab.ui.control.EditField
        EdadEditFieldLabel    matlab.ui.control.Label
        PACIENTELabel         matlab.ui.control.Label
        IniciarButton         matlab.ui.control.Button
        CenterPanel           matlab.ui.container.Panel
        UIAxes                matlab.ui.control.UIAxes
        RightPanel            matlab.ui.container.Panel
        ESTADOLabel           matlab.ui.control.Label
        Label                 matlab.ui.control.Label
        UIAxes2               matlab.ui.control.UIAxes
    end

    % Properties that correspond to apps with auto-reflow
    properties (Access = private)
        onePanelWidth = 576;
        twoPanelWidth = 768;
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: IniciarButton
        function IniciarButtonPushed(app, event)
               load Datos_Red_GRU.mat
               plot(app.UIAxes, s_ACA.mV)
               title('Datos adquiridos')
               ylabel('Amplitud (mV)')
               xlabel('Numero de Muestras')

               category = classify(net, CellEntrada)
               pie(app.UIAxes2, category)
               legend(app.UIAxes2, {'A = Anormal', 'N = Normal'},Location="southoutside" )



        end

        % Button down function: UIAxes
        function UIAxesButtonDown(app, event)

        end

        % Button down function: UIAxes2
        function UIAxes2ButtonDown(app, event)

        end

        % Changes arrangement of the app based on UIFigure width
        function updateAppLayout(app, event)
```

```
59            currentFigureWidth = app.UIFigure.Position(3);
60            if(currentFigureWidth <= app.onePanelWidth)
61                % Change to a 3x1 grid
62                app.GridLayout.RowHeight = {352, 352, 352};
63                app.GridLayout.ColumnWidth = {'1x'};
64                app.CenterPanel.Layout.Row = 1;
65                app.CenterPanel.Layout.Column = 1;
66                app.LeftPanel.Layout.Row = 2;
67                app.LeftPanel.Layout.Column = 1;
68                app.RightPanel.Layout.Row = 3;
69                app.RightPanel.Layout.Column = 1;
70            elseif (currentFigureWidth > app.onePanelWidth && currentFigureWidth <= app.
    twoPanelWidth)
71                % Change to a 2x2 grid
72                app.GridLayout.RowHeight = {352, 352};
73                app.GridLayout.ColumnWidth = {'1x', '1x'};
74                app.CenterPanel.Layout.Row = 1;
75                app.CenterPanel.Layout.Column = [1,2];
76                app.LeftPanel.Layout.Row = 2;
77                app.LeftPanel.Layout.Column = 1;
78                app.RightPanel.Layout.Row = 2;
79                app.RightPanel.Layout.Column = 2;
80            else
81                % Change to a 1x3 grid
82                app.GridLayout.RowHeight = {'1x'};
83                app.GridLayout.ColumnWidth = {217, '1x', 368};
84                app.LeftPanel.Layout.Row = 1;
85                app.LeftPanel.Layout.Column = 1;
86                app.CenterPanel.Layout.Row = 1;
87                app.CenterPanel.Layout.Column = 2;
88                app.RightPanel.Layout.Row = 1;
89                app.RightPanel.Layout.Column = 3;
90            end
91        end
92    end
93
94    % Component initialization
95    methods (Access = private)
96
97        % Create UIFigure and components
98        function createComponents(app)
99
100           % Create UIFigure and hide until all components are created
101           app.UIFigure = uifigure('Visible', 'off');
102           app.UIFigure.AutoResizeChildren = 'off';
103           app.UIFigure.Position = [100 100 1073 352];
104           app.UIFigure.Name = 'MATLAB App';
105           app.UIFigure.SizeChangedFcn = createCallbackFcn(app, @updateAppLayout, true);
106
107           % Create GridLayout
108           app.GridLayout = uigridlayout(app.UIFigure);
109           app.GridLayout.ColumnWidth = {217, '1x', 368};
110           app.GridLayout.RowHeight = {'1x'};
111           app.GridLayout.ColumnSpacing = 0;
112           app.GridLayout.RowSpacing = 0;
```

```
113             app.GridLayout.Padding = [0 0 0 0];
114             app.GridLayout.Scrollable = 'on';
115 
116             % Create LeftPanel
117             app.LeftPanel = uipanel(app.GridLayout);
118             app.LeftPanel.Layout.Row = 1;
119             app.LeftPanel.Layout.Column = 1;
120 
121             % Create IniciarButton
122             app.IniciarButton = uibutton(app.LeftPanel, 'push');
123             app.IniciarButton.ButtonPushedFcn = createCallbackFcn(app, @IniciarButtonPushed,
    true);
124             app.IniciarButton.Position = [66 55 100 32];
125             app.IniciarButton.Text = 'Iniciar';
126 
127             % Create PACIENTELabel
128             app.PACIENTELabel = uilabel(app.LeftPanel);
129             app.PACIENTELabel.FontSize = 24;
130             app.PACIENTELabel.FontWeight = 'bold';
131             app.PACIENTELabel.Position = [49 305 125 32];
132             app.PACIENTELabel.Text = 'PACIENTE';
133 
134             % Create EdadEditFieldLabel
135             app.EdadEditFieldLabel = uilabel(app.LeftPanel);
136             app.EdadEditFieldLabel.HorizontalAlignment = 'right';
137             app.EdadEditFieldLabel.Position = [17 157 33 22];
138             app.EdadEditFieldLabel.Text = 'Edad';
139 
140             % Create EdadEditField
141             app.EdadEditField = uieditfield(app.LeftPanel, 'text');
142             app.EdadEditField.Position = [57 155 155 26];
143 
144             % Create NombreEditFieldLabel
145             app.NombreEditFieldLabel = uilabel(app.LeftPanel);
146             app.NombreEditFieldLabel.HorizontalAlignment = 'right';
147             app.NombreEditFieldLabel.Position = [9 219 48 22];
148             app.NombreEditFieldLabel.Text = 'Nombre';
149 
150             % Create NombreEditField
151             app.NombreEditField = uieditfield(app.LeftPanel, 'text');
152             app.NombreEditField.Position = [59 216 153 28];
153 
154             % Create CenterPanel
155             app.CenterPanel = uipanel(app.GridLayout);
156             app.CenterPanel.Layout.Row = 1;
157             app.CenterPanel.Layout.Column = 2;
158 
159             % Create UIAxes
160             app.UIAxes = uiaxes(app.CenterPanel);
161             title(app.UIAxes, 'Datos Adquiridos')
162             xlabel(app.UIAxes, 'Tiempo (s)')
163             ylabel(app.UIAxes, 'Amplitud (mV)')
164             zlabel(app.UIAxes, 'Z')
165             app.UIAxes.ButtonDownFcn = createCallbackFcn(app, @UIAxesButtonDown, true);
166             app.UIAxes.Position = [7 6 468 339];
```

```
167
168             % Create RightPanel
169             app.RightPanel = uipanel(app.GridLayout);
170             app.RightPanel.Layout.Row = 1;
171             app.RightPanel.Layout.Column = 3;
172
173             % Create UIAxes2
174             app.UIAxes2 = uiaxes(app.RightPanel);
175             xlabel(app.UIAxes2, 'X')
176             ylabel(app.UIAxes2, 'Y')
177             zlabel(app.UIAxes2, 'Z')
178             app.UIAxes2.ButtonDownFcn = createCallbackFcn(app, @UIAxes2ButtonDown, true);
179             app.UIAxes2.Position = [31 30 307 276];
180
181             % Create Label
182             app.Label = uilabel(app.RightPanel);
183             app.Label.FontSize = 24;
184             app.Label.FontWeight = 'bold';
185             app.Label.Position = [67 292 25 32];
186             app.Label.Text = '';
187
188             % Create ESTADOLabel
189             app.ESTADOLabel = uilabel(app.RightPanel);
190             app.ESTADOLabel.FontWeight = 'bold';
191             app.ESTADOLabel.Position = [168 323 54 22];
192             app.ESTADOLabel.Text = 'ESTADO';
193
194             % Show the figure after all components are created
195             app.UIFigure.Visible = 'on';
196         end
197     end
198
199     % App creation and deletion
200     methods (Access = public)
201
202         % Construct app
203         function app = app2
204
205             % Create UIFigure and components
206             createComponents(app)
207
208             % Register the app with App Designer
209             registerApp(app, app.UIFigure)
210
211             if nargout == 0
212                 clear app
213             end
214         end
215
216         % Code that executes before app deletion
217         function delete(app)
218
219             % Delete UIFigure when app is deleted
220             delete(app.UIFigure)
221         end
```

```
222     end
223 end
```

Referencias

B. Abuhaija, A. Alloubani, M. Almatari, G. M. Jaradat, H. B. Abdalla, A. M. Abualkishik, M. K. Alsmadi. A comprehensive study of machine learning for predicting cardiovascular disease using Weka and SPSS tools. *International Journal of Electrical and Computer Engineering*, 13(2):1891 – 1902, 2023. ISSN 20888708. DOI http://doi.org/10.11591/ijece.v13i2.pp1891-1902. URL https://bit.ly/3JH0J36. Publisher: Institute of Advanced Engineering and Science Type: Article.

C. C. Aggarwal. *Neural Networks and Deep Learning. A Textbook*. Springer, 2018. ISBN 978-3-319-94463-0. URL http://gen.lib.rus.ec/book/index.php?md5=2f4d0e05eb1218c27e23558b60c48735.

M. A. Ahamed, M. K. Hasan, M. S. Alam. Design and implementation of low cost ecg monitoring system for the patient using smartphone. *International Conference on Electrical & Electronic Engineering (ICEEE)*, pp 261–264, 2015. DOI https://doi.org/10.1109/CEEE.2015.7428272. URL https://ieeexplore.ieee.org/document/7428272.

Anadigm. Inc. An221e04 datasheet dynamically reconfigurable fpaa with enhanced i/o. https://www.anadigm.com/documents/AN221E04-1.pdf, 2012.

I. Bonet Cruz, S. Salazar Martínez, A. Rodríguez Abed, R. Grau Ábalo, M. M. García Lorenzo. Redes neuronales recurrentes para el análisis de secuencias. *Revista Cubana de Ciencias Informáticas*, 2007. ISSN 1994-1536. URL https://www.redalyc.org/articulo.oa?id=378343634004.

J. J. Cho. An exploratory study on issues and challenges of agile software development with scrum. *All Graduate theses and dissertations*, p 599, 2010. DOI https://doi.org/10.26076/4055-b875. URL https://digitalcommons.usu.edu/etd/599.

P. I. Dorado Díaz, J. Sampedro Gómez, V. Vicente Palacios, P. L. Sánchez. Aplicaciones de la inteligencia artificial en cardiología: el futuro ya está aquí. *Revista Española de Cardiología*, 72(12):1065–1075, 2019. ISSN 03008932. DOI https://doi.org/10.1016/j.recesp.2019.05.016. URL https://www.revespcardiol.org/es-aplicaciones-inteligencia-artificial-cardiologia-el-articulo-S0300893219302507.

N. González Cervantes, A. Espinoza Valdez, R. Salido Ruiz. Potencial eléctrico en el corazón: Representación mediante un grafo. *ReCIBE. Revista electrónica de Computación, Informática, Biomédica y Electrónica*, 2016. URL https://www.redalyc.org/articulo.oa?id=512253114012.

L. E. Gonzélez Medina. Análisis espacial de las enfermedades cardiovasculares en méxico: Modelos predictivo. Bachelor's thesis, UNAM, 2022. URL http://132.248.9.195/ptd2022/enero/0821498/Index.html.

R. Jennifer S., P. Ram, P. Isidoros, S. Yong. *Intelligent Sustainable Systems: Proceedings of ICISS 2021 (Lecture Notes in Networks and Systems, 213)*. Springer, 1st ed. 2022 edición, 2021. ISBN 9811624216,9789811624216. URL http://gen.lib.rus.ec/book/index.php?md5=D3466E299E8201699E17D0E8E1EE4C2C.

S. J.J. *SCRUM: MANUAL DE CAMPO. Una clase magistral para acelerar el desempeño, obtener resultados y planear para el futuro*. OCEANO, 1 ed edición, 2020. ISBN 978-607-557-191-1.

D. O. Mangne Machaca. Detección de arritmias cardiacas con redes neuronales artificiales. Bachelor's thesis, UNIVERSIDAD MAYOR DE SAN ANDRÉS FACULTAD DE CIENCIAS PURAS Y NATURALES, 2009. URL https://repositorio.umsa.bo/bitstream/handle/123456789/1485/T-1900.pdf?sequence=1&isAllowed=y.

R. Matrínez Sastre. Redes neuronales recurrentes para la predicción de energia eólica y fotovoltaica, 2021. URL https://repositorio.uam.es/bitstream/handle/10486/700222/martinez_sastre_ruben_tfm.pdf?sequence=1&isAllowed=y.

A. Navarro, J. D. Fernández, J. Morales. Revisión de metodologías ágiles para el desarrollo de software. *PROSPECTIVA*, 11:30–39, 2013. ISSN 1692-8261. URL https://www.redalyc.org/articulo.oa?id=496250736004.

R. C. Ortega Ordoñez, L. F. Cobos Recalde, G. P. Segura Núñez, E. M. Sandoval Sandoval. Adquisición de señales analógicas de instrumentación con logo! soft v8.3 mediante generador de señales y el sensor pt100. *Ciencia Latina Revista Científica Multidisciplinar*, 7(1):7865–7880, mar. 2023. DOI https://doi.org/10.37811/cl_rcm.v7i1.5017. URL https://ciencialatina.org/index.php/cienciala/article/view/5017.

A. Palma, C. Martìnez, E. Castillo, A. García, L. Capitán, A. Martínez, M. Carvajal, P. Escobedo. Procedimiento para la detección de señales balistocardiográficas y sistema que lo implementa, 2020. URL https://patentimages.storage.googleapis.com/77/b5/9b/0fceec4eb8a2dc/WO2020136302A1.pdf. Accessed: 2023-03-18.

K. Prabu, T. Ravikumar. Arduino and raspberry pi based efficient patient monitoring system. *International Journal of Engineering Research & Technology (IJERT) ICETET*, Volume 4, 2016. ISSN 2278-0181. URL https://www.ijert.org/arduino-and-raspberry-pi-based-efficient-patient-monitoring-system.

J. Pérez Guerrero. Redes recurrentes, 2020. URL https://idus.us.es/bitstream/handle/11441/115230/TFG%20DGMyE%20P%c3%a9rez%20Guerrero%2c%20Jes%c3%bas.pdf?sequence=1&isAllowed=y.

A. Rajkumar, M. Ganesan, R. Lavanya. Arrhythmia classification on ecg using deep learning. *International Conference on Advanced Computing & Communication Systems (ICACCS)*, 5:365–369, 2019. DOI https://doi.org/10.1109/ICACCS.2019.8728362. URL https://ieeexplore.ieee.org/document/8728362.

P. Rodríguez, H. Castro, C. Pinedo, J. Velasco. Diseño de un microsistema para adquisición de señales cardiacas usando fpaas. *Grupo de Bio-nanoelectrónica EIEE. Universidad del Valle*, 2023. URL http://161.111.232.132/iberchip2006/ponencias/77.pdf.

T. M. Ryan Ripley. *Fixing Your Scrum: Practical Solutions to Common Scrum Problems*. Pragmatic Bookshelf, 1 edición, 2020. ISBN 1680506978,9781680506976. URL http://gen.lib.rus.ec/book/index.php?md5=DA023A375B81251731C887801B7D1A02.

R. E. Smith, B. N. Socarrás, C. R. Vázquez. Aplicación android para la adquisición inalámbrica y visualización de señales biomédicas. *Revista de Información Científica*, 101(3), 2022. URL https://www.redalyc.org/articulo.oa?id=551771993001.

R. Vargas-Cañas, A. R. Muñoz, W. A. Campo Cuaran. Construcción de un sistema Electrocardiográfico con conexión inalámbrica a teléfonos inteligentes. *Scientia et Technica*, 26(03):269–277, Sept. 2021. DOI https://doi.org/10.22517/23447214.23711. URL https://revistas.utp.edu.co/index.php/revistaciencia/article/view/23711. Section: Mecánica.