

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Detección de anomalías cardíacas mediante algoritmos de inteligencia artificial.

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE SISTEMAS DE
TELECOMUNICACIÓN

Autor: Isaac Nicolás Marrero

Director: José Fernando Cerdán Cartagena

Codirectores: Mario Pérez Gomariz

Daniel Carreres Prieto

Cartagena, 04/04/2022

SUMARIO.

En la actualidad, en el campo de la medicina, los médicos emplean diferentes herramientas para poder detectar diversas anomalías en los latidos de un paciente, la frecuencia o si hay un aumento de las cavidades y de esa manera localizar arritmias. Estas arritmias son causantes de un gran número de muertes en todo el mundo cada año. Se utilizan técnicas como los electrocardiogramas (ECG), extraídos de los pacientes, para poder visualizar y analizar la actividad eléctrica del corazón.

Sin embargo, la interpretación se vuelve un tanto tediosa a veces y el médico puede equivocarse en determinadas ocasiones. Es por ello por lo que se ha optado por la creación de un sistema automatizado de detección y clasificación de latidos del corazón, empleando técnicas de inteligencia artificial y herramientas proporcionadas por *PhysioNet*, como son las librerías de WFDB, todas ellas implementadas dentro del lenguaje de programación Python. Para poder elaborar dicho sistema, se empezó creando un algoritmo, empleando la base de datos de arritmias MI-BIH, que permitía la extracción de diferentes características presentes en un latido, concretamente 28. Como resultado se obtuvo una nueva base de datos con dichas características, que se usaría para el entrenamiento de una red neuronal, permitiéndonos así, poder clasificar los diferentes tipos de latidos de un electrocardiograma, en 5 clases diferentes. En los siguientes capítulos se documentará todo el procedimiento seguido durante la elaboración del proyecto, así como los diferentes resultados. Para concluir cabe destacar que tras la creación de la red neuronal los resultados que se obtuvieron presentaron una tasa de error bastante aceptable, situando el proyecto como una herramienta útil y fiable para el análisis, detección y clasificación de latidos, aunque con diversas excepciones que se comentarán más adelante.

Índice.

SUMARIO.....	2
CAPÍTULO 1	9
Introducción.....	9
1.1 El corazón.....	9
1.1.1 ¿Cómo se genera un latido?.....	9
1.1.2 Electrocardiograma.....	10
1.1.2.1 Derivaciones de un electrocardiograma.....	11
1.1.2.2 Características presentes en un latido.....	13
CAPÍTULO 2	16
Recursos empleados durante el proyecto.....	16
2.1. Base de datos de arritmias MIT-BIH.....	16
2.2. Agrupación de latidos según AMMI.....	18
2.3. Visual Studio Code y Python.....	19
2.3.1 Keras.....	20
2.3.2. Herramienta WFDB.....	20
CAPÍTULO 3	21
Elaboración de la base de datos.....	21
3.1. Dataset.....	22
3.1.1. Extracción de las características usadas de cada registro.....	23
3.1.2. Creación del Dataset.....	26
CAPÍTULO 4	29
Redes Neuronales.....	29
4.1. Computación neuronal.....	29

4.2. Elementos básicos de una red neuronal.....	30
4.3. Arquitecturas de redes neuronales.....	31
4.4. Tipos de aprendizaje automático.....	32
4.5. Estructura de la red neuronal.....	34
4.5.1. Descripción del modelo.....	36
4.5.2. División del Dataset.....	37
CAPÍTULO 5	39
Resultados y Discusión.....	39
5.1. Introducción:.....	39
5.2. Análisis de los modelos:.....	39
5.2.1. Modelo 3:.....	41
5.2.1.1. Resultados del testeo:	42
5.3. Modificación:	43
5.3.1. Análisis de los modelos:	44
5.3.1.1. Modelo 3:.....	45
5.3.1.2. Resultados del testeo:	47
5.4. Comparación de los resultados:.....	48
5.4.1. Comparación del antes y después de las modificaciones.	48
CAPÍTULO 6	50
Conclusiones.....	50
6.1. Conclusión.....	50
Bibliografía.....	51

Índice de tablas.

Tabla 1: Clasificación de latidos según AAMI.	19
Tabla 2: Relación entre los latidos empleados en el proyecto y los latidos de la base de datos MIT-BIH.	22
Tabla 3: Asignación de los datos a los diferentes conjuntos.	38
Tabla 4: Clasificación de los latidos según su tipo.	39
Tabla 5: Análisis de los modelos desbalanceados.	40
Tabla 6: Resultados del testeo del modelo con desbalanceo.	42
Tabla 7: Análisis de los modelos modificados.	45
Tabla 8: Resultados del testeo del modelo modificado.	47
Tabla 9: Comparación del modelo sin modificar y el modelo modificado.	48



Índice de figuras.

Figura 1: El corazón [4].	10
Figura 2: Electrocardiograma [6].	11
Figura 3: Derivaciones periféricas dentro de un electrocardiograma [8].	12
Figura 4: Derivaciones precordiales dentro de un electrocardiograma [7].	13
Figura 5: Intervalos dentro de un latido [12].	15
Figura 6: Electrocardiograma de las dos derivaciones de un registro de MIT-BIH [13].	17
Figura 7: Etiquetado de la base de datos MIT-BIH [13].	17
Figura 8: Tabla de anotaciones para el registro 200 de MIT-BIH [13].	18
Figura 9: Diagrama de elaboración del dataset.	21
Figura 10: Representación de los intervalos de un latido [10].	25
Figura 11: Latido original antes del muestreo donde el eje de ordenadas representa los (mV) y el eje de abscisas representa las muestras.	26
Figura 12: Latido después de volver a muestrear donde el eje de ordenadas representa los (mV) y el eje de abscisas representa las muestras.	26
Figura 13: Dataset de 53373 filas y 28 columnas.	27
Figura 14: Dataset de 51643 filas y 28 columnas con valores NaN eliminados.	28
Figura 15: Funcionamiento de una neurona artificial [26].	29
Figura 16: Ejemplos de funciones de activación [30].	31
Figura 17: Estructura de una red neuronal [31].	32
Figura 18: Estructura del modelo de aprendizaje supervisado [27].	33
Figura 19: Estructura del modelo de aprendizaje no supervisado [27].	34
Figura 20: Diagrama de la red neuronal.	35
Figura 21: Diagrama del sistema de detección y clasificación de latidos.	35

Figura 22: Representación de la evolución de las pérdidas y la exactitud.	41
Figura 23: Matriz de confusión para el modelo con desbalanceo.	43
Figura 24: Representación de la evolución de las pérdidas y la exactitud.	46
Figura 25: Matriz de confusión para el modelo modificado.	48



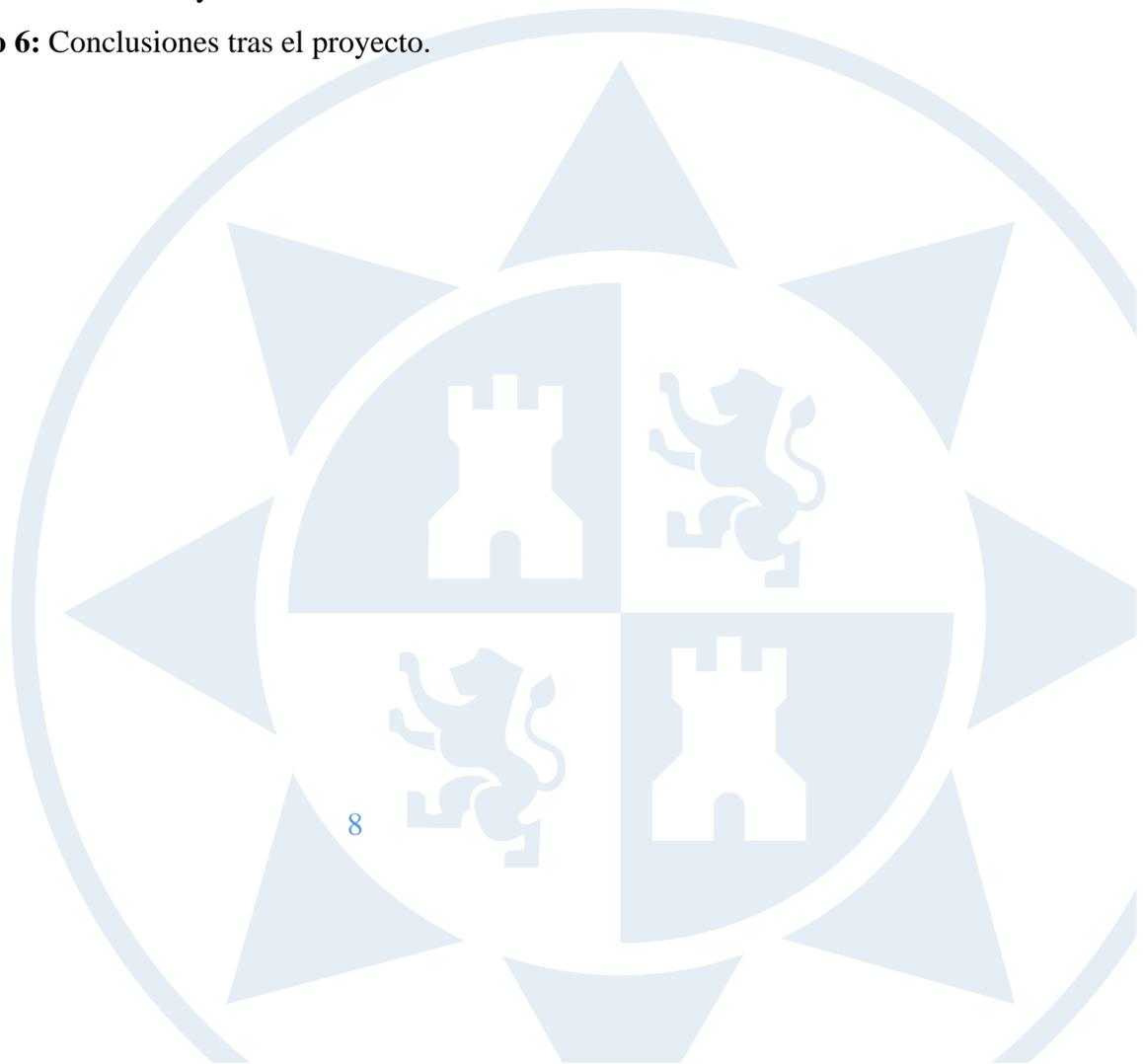
Objetivos del proyecto y estructura.

Los objetivos del proyecto han sido:

- Familiarizarse con la fisiología que hay detrás del corazón, así como la identificación de latidos y su clasificación.
- Elaborar una base datos simplificada en la que se recopilen las diferentes características que componen los latidos.
- Diseñar y optimizar una red neuronal capaz de clasificar de forma optimizada los diferentes latidos de los que se compone un ECG.

Se ha estructurado de la siguiente manera:

- **Capítulo 1:** Introducción.
- **Capítulo 2:** Recursos empleados para la elaboración del proyecto.
- **Capítulo 3:** Elaboración de la base de datos.
- **Capítulo 4:** Creación de una red neuronal.
- **Capítulo 5:** Resultados y discusiones.
- **Capítulo 6:** Conclusiones tras el proyecto.



CAPÍTULO 1

Introducción.

1.1 El corazón.

En el ámbito de la medicina, el corazón siempre ha sido un órgano complejo de analizar. Consiste en una bomba que, gracias a su acción impulsora, proporciona el empuje necesario para permitir que tanto la sangre como las sustancias que esta transporta, circulen fácilmente a través de venas y arterias, llegando a todas las partes del cuerpo.

El corazón presenta un sistema eléctrico que permite generar dicho impulso y mantener una frecuencia y un ritmo cardiaco adecuado. Este sistema consta de un centro de generador del latido, llamado nódulo sinusal (sinoauricular), situado en la aurícula derecha, que genera señales eléctricas. Estas señales viajan a través de células conductoras que llevan la señal desde las cámaras superiores (aurículas) a las cámaras inferiores (ventrículos) del corazón, provocando una contracción del músculo [1].

Puesto que este sistema se encarga de mantener un ritmo y frecuencia estables del corazón, en el caso de existir una anomalía en su funcionamiento podría derivar en problemas como arritmias, provocando un desequilibrio en el latido y haciendo que el corazón lata con mayor rapidez, lo cual estaríamos ante una taquicardia o que lo haga más lentamente en cuyo caso significaría la presencia de una bradicardia. Normalmente su tratamiento suele ayudar a controlar o eliminar por completo los latidos irregulares [2].

De presentar un funcionamiento correcto, este sistema mantendría al corazón proporcionando sangre tanto a los pulmones como al resto del cuerpo.

1.1.1 ¿Cómo se genera un latido?

Dentro del corazón podemos localizar 4 cavidades [3]: 2 aurículas situadas en la parte superior y dos ventrículos situados en la parte inferior. El ritmo está controlado por el nódulo sinusal, situado como hemos dicho anteriormente en la aurícula derecha. Este generador de impulsos eléctricos provoca una contracción de los músculos de las aurículas y en consecuencia un bombeo de la sangre hacia los ventrículos. A continuación,

dichos impulsos eléctricos llegan a unas células llamadas nódulo auriculoventricular cuya función es ralentizar la señal eléctrica para que los ventrículos se llenen de sangre. Una vez que los impulsos eléctricos han llegado a los ventrículos, estos llegan a contraerse de forma que bombean la sangre a las diferentes partes del cuerpo.

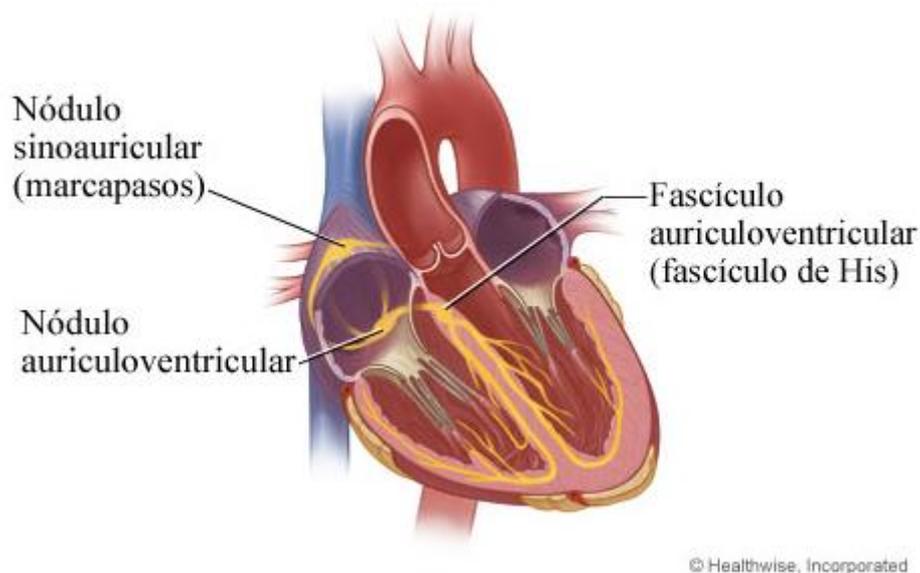


Figura 1: El corazón [4].

1.1.2 Electrocardiograma.

Un electrocardiograma [5] es una prueba que evalúa la actividad eléctrica del corazón, permitiendo medir el ritmo y la regularidad de los latidos. Esta actividad se registra desde la superficie corporal del paciente mediante electrodos colocados en los tobillos, muñecas y pecho, obteniéndose así el impulso eléctrico desde diferentes posiciones. Estos electrodos detectan los cambios eléctricos que son resultado de la despolarización del musculo cardiaco seguida de una repolarización durante cada latido. Aunque realizarle a un paciente una electrocardiografía (proceso de producir un ECG) es sencillo, la interpretación de estos suele ocasionar errores, sobre todo en aquellos en los que se ha hecho un registro a largo plazo. Es por ello por lo que la inteligencia artificial coge fuerza en este campo de la medicina. Aportando una ayuda adicional a los médicos para poder

detectar con más facilidad y rapidez diferentes enfermedades cardiacas siendo esta la motivación del proyecto.

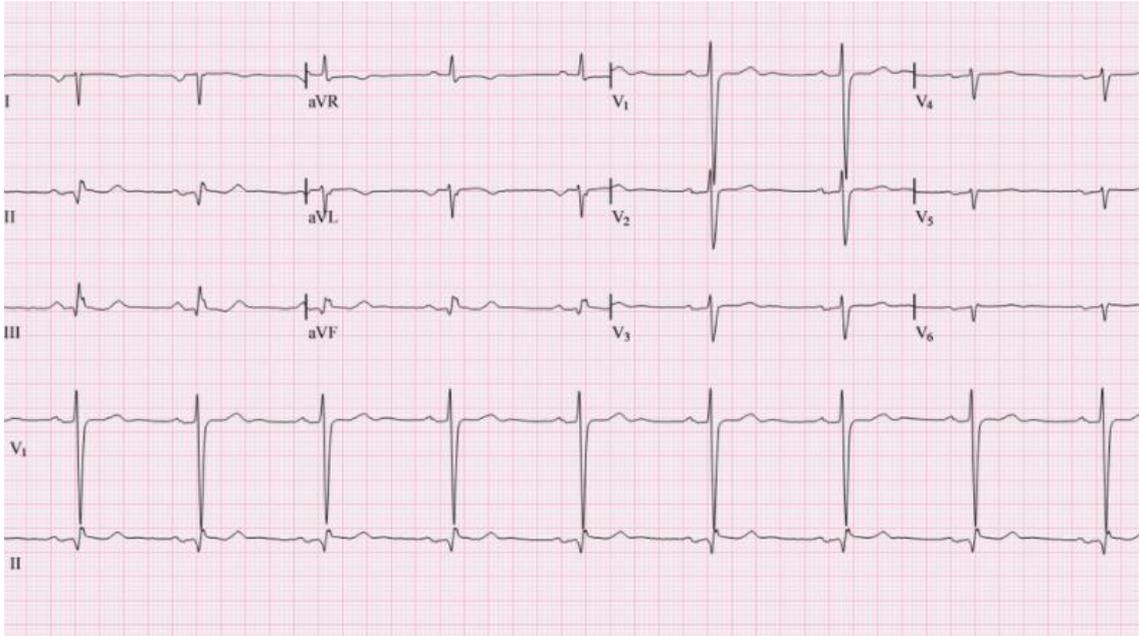


Figura 2: Electrocardiograma [6].

1.1.2.1 Derivaciones de un electrocardiograma.

Como hemos mencionado antes, los electrodos colocados en la piel recogen la actividad eléctrica del corazón. Normalmente se emplean un total de 10, aunque este número puede variar. Para poder evaluar de forma tridimensional la actividad del corazón, con el fin de identificar con facilidad anomalías, se registran 12 derivaciones.

La colocación de los electrodos se denomina [9]:

- **RL:** pierna derecha.
- **LL:** pierna izquierda.
- **RA:** brazo derecho.
- **LA:** brazo izquierdo
- Y por último los 6 que se encuentran colocados en el pecho: **V1- V6.**

Cada par de estos electrodos se usan para extraer una derivación, las cuales se pueden clasificar de la siguiente manera [7]:

Derivaciones periféricas: los electrodos son colocados en las extremidades del paciente. Estas derivaciones se llaman bipolares (I,II,y III) y unipolares-aumentadas (aVL, aVR y aVF).

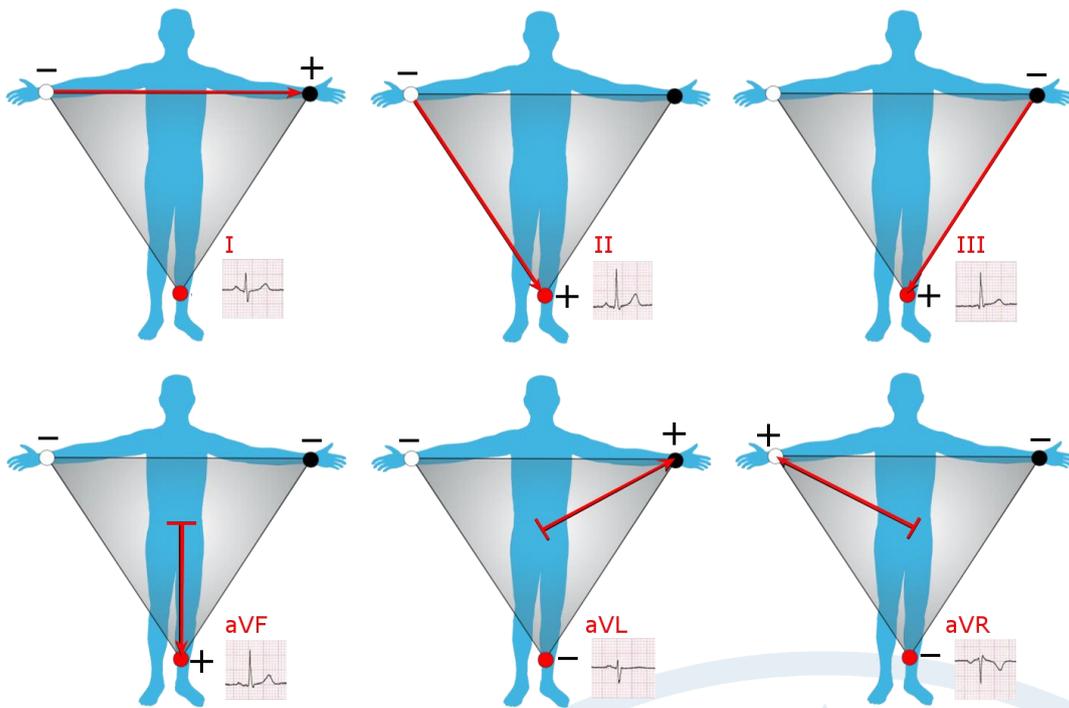


Figura 3: Derivaciones periféricas dentro de un electrocardiograma [8].

- **La derivación I:** registra la diferencia de potencia que existe entre el brazo derecho y el izquierdo.
- **La derivación II:** registra la diferencia de potencia que existe entre el brazo derecho y la pierna izquierda.
- **La derivación III:** registra la diferencia de potencia que existe entre el brazo y la pierna izquierdos.
- **La derivación aVL:** presenta un electrodo positivo en el brazo izquierdo y un electrodo negativo que representa la combinación del brazo derecho y la pierna izquierda.

- **La derivación aVR:** presenta un electrodo positivo en el brazo derecho y un electrodo negativo que representa la combinación del brazo izquierdo y la pierna izquierda
- **La derivación aVF:** presenta un electrodo positivo en la pierna izquierda y un electrodo negativo que representa la combinación del brazo derecho y el brazo izquierdo

Derivaciones precordiales: los electrodos son colocados en el pecho. Son derivaciones unipolares que registran la actividad eléctrica en el plano horizontal. Estas son: V1, V2, V3, V4, V5 y V6. Al estar tan cerca del corazón no es necesario aumentarlas para su evaluación.

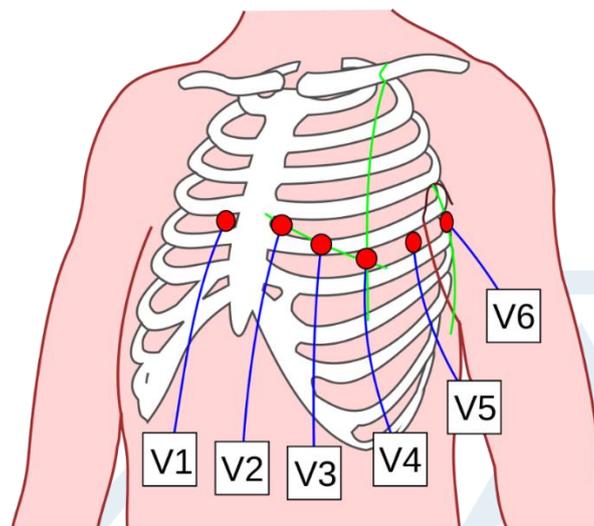


Figura 4: Derivaciones precordiales dentro de un electrocardiograma [7].

1.1.2.2 Características presentes en un latido.

Hay que tener en cuenta de que existen un gran número de componentes presentes en un latido. A continuación, se explicarán las diferentes partes que contienen, que podemos visualizar dentro de las ondas de un electrocardiograma. Estas partes nos ayudarán a identificar y determinar la naturaleza de los diferentes latidos. Cabe resaltar que estas ondas presentan unas características de amplitud y morfología determinadas por lo que cualquier variación de estas supondría un caso importante de estudio clínico [10].

La onda P: representa la despolarización de las aurículas dando como resultado una contracción o sístole auriculares. Presenta una duración normalmente de unos 80ms y en la mayoría de las derivaciones es ascendente. Y precede al complejo QRS.

El complejo QRS: muestra la despolarización de los ventrículos. Este complejo se puede visualizar en el ECG con una amplitud mayor que la que tiene la onda P. Suele tener una duración aproximada de unos 80 a 100 ms. Puede aportar información de diversas anomalías como taquicardias ventriculares entre otras.

El segmento ST: es la conexión entre el complejo QRS mencionado anteriormente con la onda T. Este segmento representa la despolarización completa del miocardio ventricular. Con una duración aproximada de 5 a 150 ms.

El segmento PR: muestra el periodo que existe entre el final de la onda P y el comienzo del complejo QRS.

El intervalo PR: muestra el periodo que tarda el impulso eléctrico en recorrer la distancia que hay entre el nódulo sinusal hasta el nódulo auriculoventricular. Este se mide desde el inicio de la onda P hasta el comienzo del complejo QRS.

La onda T: surge de la repolarización de los ventrículos, presentando una duración de unos 160 ms.

El intervalo QT: es el periodo entre el comienzo que abarca la despolarización ventricular y el final de la repolarización ventricular.

La onda U: aparece por la repolarización del tabique interventricular. Presentando una amplitud baja, aunque normalmente su presencia es nula.

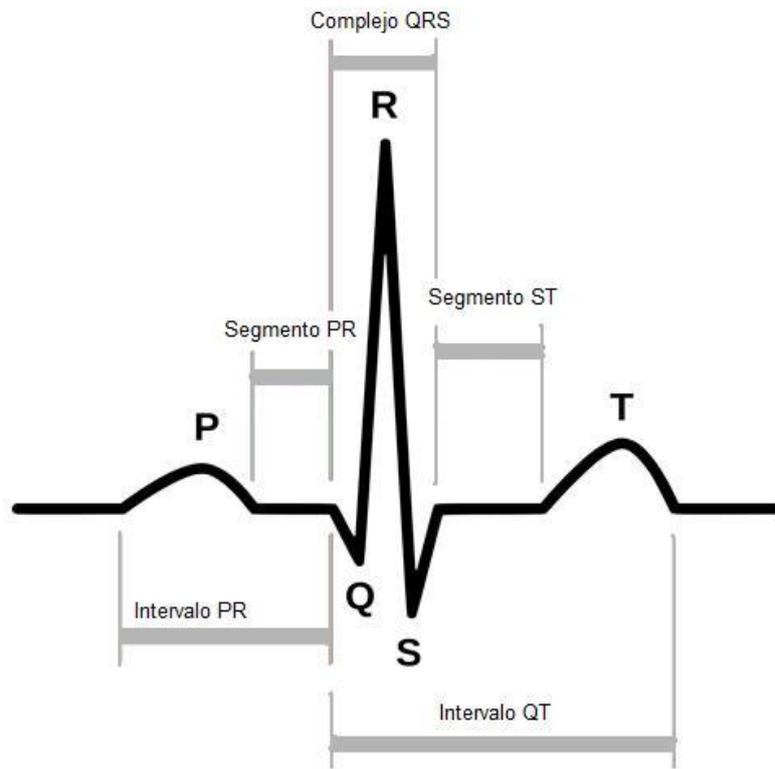


Figura 5: Intervalos dentro de un latido [12].

La onda U no aparece en la figura anterior ya que no suele ser visible en un electrocardiograma. Mas adelante se detallará que características se extrajeron de los latidos para la elaboración del proyecto [11].

CAPÍTULO 2

Recursos empleados durante el proyecto.

2.1. Base de datos de arritmias MIT-BIH.

La base de datos de arritmias *MIT-BIH* [13] ha sido seleccionada para la realización de este proyecto debido a un buen procesado de los datos y su facilidad para extraerlos. Esta base de datos se compone de 48 extractos de media hora de registros de ECG de dos canales (un total de 650 000 muestras por cada registro), extraídos de 47 sujetos obtenidos por el Laboratorio de Arritmias del Hospital Beth Israel entre 1975 y 1979. Los primeros 23 registros se escogieron al azar mientras que los 25 restantes fueron seleccionados para proporcionar una variedad de fenómenos raros, pero clínicamente relevantes, que no estarían bien representados por una pequeña muestra aleatoria de registros holter. El primer grupo aporta una variedad de formas de ondas y artefactos que un detector de arritmias se puede topar. Y los del segundo grupo se seleccionaron por presentar arritmias supraventriculares, de la unión, ventriculares y anomalías de la conducción. Se seleccionaron mujeres de entre 23 y 89 años, concretamente 22 mujeres y 25 hombres de entre 32 y 89 años. En casi todos los registros la señal superior es una derivación de extremidad modificada II (MLII), que ha sido obtenida mediante la colocación de los electrodos en el pecho. La señal más baja con normalidad es una derivación modifica V1, sin embargo, existen registros cuya señal inferior es la V2, V5 y en un caso V4. Además, hay que añadir que en la señal superior podemos ver complejos QRS prominentes a diferencia de la señal inferior que presentan una mayor dificultad para distinguir estos complejos. Todas las grabaciones presentes en esta base de datos se encuentran digitalizadas con una resolución de unos 11 bits en un rango de 10mV y han sido muestreadas con una frecuencia de 360 muestras por segundo. Además, la base de datos contiene anotaciones que aportan información desde el tipo de latido a comentarios y frecuencia cardiaca [13]. Por ejemplo, existen unas 109 000 etiquetas, que muestran el tipo de latido. Estas etiquetas han sido revisadas a lo largo de los años.



Figura 6: Electrocardiograma de las dos derivaciones de un registro de MIT-BIH [13].

Como podemos ver en la figura 6, se han representado durante 10 segundos el ECG del registro 200, donde se puede ver las diferentes anotaciones (atr), indicando el tipo de latido. La señal superior, como se comentó antes, es la MLII mientras que la señal inferior es la V1. Estas etiquetas vienen acompañadas de unas tablas que proporcionan el significado de cada etiqueta usada.

<i>Símbolo</i>	<i>Sentido</i>
o N	ritmo normal
L	Latido de bloqueo de rama izquierda
R	Latido de bloqueo de rama derecha
A	Latido auricular prematuro
a	Latido prematuro auricular aberrado
j	Latido prematuro nodal (de la unión)
S	Latido prematuro supraventricular
V	Contracción ventricular prematura
F	Fusión de latido ventricular y normal
[Inicio de aleteo/fibrilación ventricular
!	Onda de aleteo ventricular
]	Fin del aleteo/fibrilación ventricular
mi	Latido de escape auricular
j	Latido de escape nodal (de la unión)
mi	Latido de escape ventricular
/	ritmo de ritmo
F	Fusión de ritmo normal y ritmo
X	Onda P no conducida (APB bloqueada)
q	Ritmo inclasificable
	Artefacto tipo QRS aislado

Figura 7: Etiquetado de la base de datos MIT-BIH [13].

En esta imagen (figura 7) veríamos un pequeño ejemplo de dichas tablas.

Además, la base de datos del MIT-BIH aporta información adicional en cuanto a características de edad, sexo, incluso la medicación empleada en cada paciente. Cada registro viene acompañado además del número de anotaciones presentes en su ECG mostrando las cantidades de cada tipo como se puede observar en la siguiente figura.

Summary

Record length 00:30:05
Clock frequency 360 ticks per second
Annotator: atr (2797 annotations)
N 1749
V 825
(B 71
(N 70
~ 43
A 30
(VT 7
F 2

Signal: MLII 1 tick per sample; 200 adu/mV; 11-bit ADC, zero at 1024; baseline is 1024
Signal: V1 1 tick per sample; 200 adu/mV; 11-bit ADC, zero at 1024; baseline is 1024

Notes:

```
64 M 1953 3655 x1
Digoxin, Quinidine
The PVCs are multiform. There are occasional bursts of high-frequency
noise in the upper channel, and severe noise and artifact in the lower
channel.
```

Figura 8: Tabla de anotaciones para el registro 200 de MIT-BIH [13].

2.2. Agrupación de latidos según AMMI.

Para poder identificar anomalías del funcionamiento del corazón, como son las arritmias, es necesario poder identificar los diferentes latidos que hay. Las arritmias aparecen cuando los impulsos eléctricos que se encargan de la coordinación de los latidos no funcionan correctamente provocando que el corazón lata demasiado rápido o lento o de manera irregular. Dentro de las arritmias podemos ir desde las más inocuas hasta aquellas que pueden hacer peligrar la salud del paciente. Cuando las arritmias provocan un mal funcionamiento del corazón, este no es capaz de bombear la sangre de manera correcta

provocando una menor capacidad para hacer ejercicio, mareos incluso la imposibilidad de respirar correctamente [14]. Por ello se han empleado diferentes métodos para determinar y clasificar los diferentes latidos que hay. Para poder simplificar este proyecto se seguirá el criterio seleccionado por la *Association of Medical Instrumentation* (AMMI) [15] que clasifica los latidos en tan solo 5 clases. Teniendo en cuenta de que nuestra base de datos de arritmias MIT-BIH realiza una clasificación de unos 15 tipos de latidos, en la siguiente tabla se mostrará la distribución elegida para agrupar los latidos de MIT-BIH con la clasificación empleada por AMMI y de esa forma nuestro sistema final será capaz de diferenciar entre 5 tipos de latidos.

Clasificación de latido por AAMI:	Latido de Fusión(F)	Latido Normal(N)	Latido desconocido(Q)	Latido ectópico supraventricular(S)	Latido ectópico ventricular(V)
Clasificación de latido por MIT-BIH:	Latido ventricular y normal de fusión(F)	Latido normal(N) Bloqueo de la rama izquierda(L) Bloqueo de la rama derecha(R) Latido de escape auricular(e) Latido prematuro nodal(j)	Latido no clasificado(U) Latido normal y de ritmo de fusión(f) Latido de ritmo(P)	Latido auricular prematuro(A) Latido auricular prematuro aberrado(a) Latido prematuro nodal(J) Latido supraventricular prematuro(S)	Latido ventricular prematuro(V) Latido de escape ventricular(E)

Tabla 1: Clasificación de latidos según AAMI.

2.3. Visual Studio Code y Python.

Python es un lenguaje de alto nivel [18], utilizado para el desarrollo de aplicaciones de todo tipo. Se trata de un lenguaje multiplataforma de código abierto y gratuito que permite desarrollar software sin límites. En los últimos años está cogiendo una importante notoriedad debido a su facilidad para trabajar con inteligencia artificial, big data, machine learning, y datacience. Las principales librerías para el análisis de datos son las de: numpy, pandas, matplotlib, scipy entre otras. Este lenguaje de programación se usará para

implementar el código de nuestras redes neuronales, así como la elaboración de nuestra base de datos. La versión usada durante el proyecto fue la Python 3.9.7 con diferentes librerías para el análisis, el procesamiento de los datos y la elaboración de nuestra red neuronal. Visual Studio Code [19] tiene como objetivo satisfacer las necesidades de cualquier usuario, por tanto, fue diseñado para poder operar en los sistemas operativos más conocidos: Linux, Windows y Mac OS. Es un editor de código fuente que nos permite trabajar con diferentes lenguajes de programación como Python, C++, JavaScript, entre otros. Además, un editor gratuito que otorga un gran número de opciones como etiquetas o recomendaciones de autocompletado. También ofrece un gran número de extensiones que ayudan con el lenguaje que estemos usando en ese momento. En este proyecto Visual Studio Code servirá de editor para nuestro código de Python.

2.3.1 Keras.

Keras [17] es una API (Application Programming Interfaces, interfaz de programación de aplicaciones) de aprendizaje profundo elaborada en Python. Ha sido desarrollada para poder conseguir una experimentación rápida gracias a una sintaxis homogénea y a una interfaz sencilla. Permite al usuario centrarse en las partes importantes a resolver quitándole carga de trabajo debido a su simplicidad. Está diseñado para poder construir por bloques la arquitectura de nuestras redes neuronales.

2.3.2. Herramienta WFDB.

El paquete empleado para la extracción, análisis y procesamiento de las señales ha sido WFDB [16]. Los componentes de este paquete son la biblioteca WFDB con unas 75 aplicaciones para el análisis de señales y un software para la visualización de datos de forma de onda llamado WAVE. El paquete está escrito en el lenguaje de programación C permitiendo su uso en plataformas como MacOS, Windows y Linux. Nosotros emplearemos de este paquete, la biblioteca WFDB que nos da la opción de leer los datos directamente desde los servidores web sin necesidad de ser descargarlos. Los archivos se encuentran en la base de datos PysioBank, que es una recopilación de todas las grabaciones de señales fisiológicas de la comunidad médica. Mientras que PysioNet es el servidor que nos permite el acceso a ese banco de datos.

CAPÍTULO 3

Elaboración de la base de datos.

Durante este capítulo se expondrán las diferentes técnicas empleadas para la extracción de los recursos que formarán parte de nuestra nueva base de datos, que servirá como dato de entrada para la red neuronal y de esta manera clasificar los latidos en 5 tipos diferentes. La extracción e identificación de los latidos de un ECG siempre ha sido una tarea complicada en el ámbito de la medicina. A lo largo de los años se han desarrollado diferentes sistemas que permiten la detección de latidos. Normalmente, estos se centran en la extracción del complejo QRS, que como se explicó anteriormente es la representación gráfica de la despolarización de los ventrículos del corazón formando una simple estructura puntiaguda en el electrocardiograma. Sin embargo, este complejo presenta ciertos problemas ya que en ocasiones puede estar distorsionado debido al ruido muscular o movimiento de los electrodos entre otros. Estos sistemas son de gran utilidad pues permiten a los médicos diagnosticar con mayor facilidad y rapidez los latidos sin la necesidad de analizar todo el ECG. En la siguiente figura podemos visualizar el diagrama que muestra el proceso hasta la elaboración del dataset.

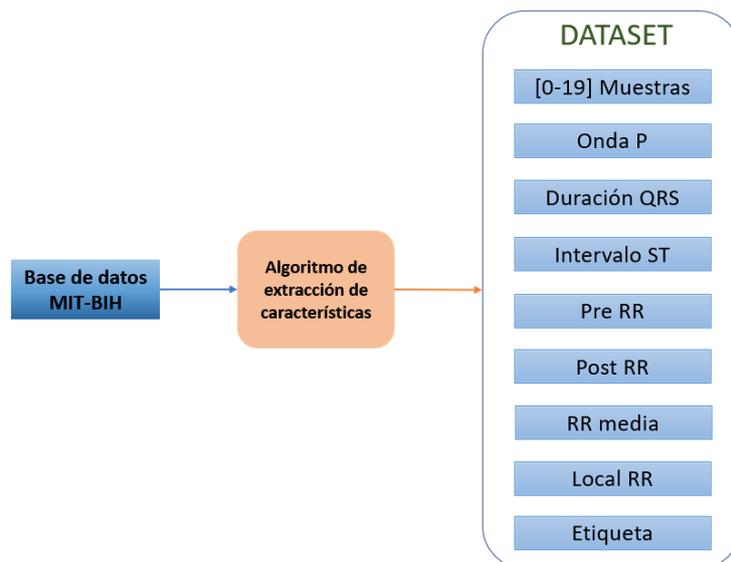


Figura 9: Diagrama de elaboración del dataset.

A continuación, expondremos que características se llegaron a usar para la elaboración de nuestra base de datos.

3.1. Dataset

Antes de nada, hay que recordar que la base de datos empleada fue la de arritmias MIT-BIH [20]. Que nos proporciona una cantidad de muestras y registros distintos y suficientes para la elaboración del proyecto, apareciendo en ella desde latidos normales hasta señales clínicamente anormales. Esta base de datos presenta un total de 109.000 latidos de los cuales se emplearon 51.643 latidos. La señal utilizada fue la MLII de cada uno de los ECG. Basándonos en otras investigaciones académicas, las cuales recomendaron el descarte de ciertos registros como 102, 104, 217 y 107 [21], [22], [23], por tener un ritmo acelerado, se emplearon para la elaboración de nuestra base de datos los siguientes: 100, 101, 103, 105, 106, 109, 119, 200, 202, 203, 205, 207, 208, 209, 210, 213, 214, 222, 223, 232 y 233. Estos representan un total de 21 registros.

Tipo de latido:	Latido de Fusión(F)	Latido Normal(N)	Latido desconocido(Q)	Latido ectópico supraventricular(S)	Latido ectópico ventricular(V)	Total:
Latidos de la base de datos(dataset):	788	42917	15	2341	5582	51.643
Latidos de MIT-BIH del proyecto:	803	90632	33	2779	7235	101.482
Porcentaje de latidos usados:	98.13%	47.35%	45.45%	84.23%	77.15%	50.88%

Tabla 2: Relación entre los latidos empleados en el proyecto y los latidos de la base de datos MIT-BIH.

Analizando la tabla anterior, hemos realizado una comparación de los latidos empleados para entrenar nuestra red neuronal y los latidos totales empleados en el proyecto, extraídos de la base de datos MIT-BIH. En primer lugar, de los latidos de fusión(F), los cuales representan el 0.79% de latidos totales, se usaron 788 de 803 latidos, lo que supone el 98.13%. De los latidos normales(N), los cuales representan el 89.30% de latidos totales, se han usado 42917 de 90632 latidos, lo que representa el 47.35%. En tercer lugar, los latidos desconocidos cuyo porcentaje es el más bajo de toda la base datos con un 0.03%,

se usaron 15 de 33 latidos, lo que representa el 45.45%. Después nos encontramos con los latidos ectópicos supraventriculares(S), los cuales representan el 2.73% de latidos totales y cuya presencia en nuestro dataset ha sido de 2341 de 2779 latidos, lo que supone un 84.23%. Finalmente, de los latidos ectópicos ventriculares, con un 7.12% de los latidos totales, se usaron 5582 de 7235 latidos, lo que representa el 77.15%.

3.1.1. Extracción de las características usadas de cada registro.

Para poder elaborar nuestra base de datos e implementarla, como veremos más adelante, en una red neuronal, antes tenemos que conocer la posición de los diferentes latidos que hay en cada uno de los registros y extraer sus características. Recordemos que la base de datos tenía un tamaño aproximado [13] de unos 109.000 latidos muestreados a una frecuencia de 360 Hz, con un total de 650.000 muestras por registro. Se decidió emplear al final, para reducir el tiempo de desarrollo del proyecto, un total de 51.643 latidos. Los registros utilizados fueron: 100, 101, 103, 105, 106, 109, 119, 200, 202, 203, 205, 207, 208, 209, 210, 213, 214, 222, 223, 232 y 233. Siguiendo otras investigaciones y artículos [21], [22], [23], [24] y [25] que exponen las técnicas de clasificación y detección de latidos, se decidió extraer 28 características para cada latido [21], [22], las cuales aportarían información suficiente para la entrada de nuestra red neuronal. Estas características las explicaremos con más detalle a continuación.

- Intervalo Pre RR: lo podemos definir como el tiempo que hay entre dos picos R de dos latidos contiguos. Es decir, el intervalo Pre RR es el tiempo que transcurre entre un pico R del latido actual y el pico R del latido anterior. Para la realización de este cálculo se empleó la herramienta de WFDB para Python. Esta herramienta nos proporcionaba la información de la posición de los picos R dentro de cada registro.
- Intervalo Post RR: lo podemos definir como el tiempo que existe entre un pico R del latido actual y el pico R del latido posterior. Se obtuvo empleando la herramienta WFDB para Python.
- Complejo QRS: la duración del complejo se define como el tiempo que existe entre el inicio del complejo hasta el final de este. Como hemos mencionado anteriormente, es la representación gráfica de la despolarización de los ventrículos

derecho e izquierdo, dando como resultado una estructura picuda en nuestro ECG. La onda Q es la deflexión descendente que aparece tras la onda P. La onda R es la deflexión ascendente que aparece tras la onda Q. Mientras que la onda S es la segunda deflexión descendente que aparece. Para poder realizar el cálculo de esta distancia se empleó, usando la biblioteca de *Neurokit2* una función para delimitar el complejo QRS: *ecg_delineate()* aportando las muestras de los diferentes picos. De aquí extraíamos los valores de inicio y final del complejo. A esto le aplicamos la diferencia de ambos.

- EL valor medio de los intervalos RR: además de los valores post RR y pre RR se obtuvo la media de todos los pre-intervalos de cada registro aplicando la función de media en Python. El intervalo RR solo nos aporta información si se compara con el valor normal de esa persona, es decir su media.
- El valor medio local de los intervalos RR: este valor medio local nos da información sobre el ritmo cardiaco de una persona promediando los *pre_intervalos* correspondientes a los 5 latidos posteriores y 5 latidos anteriores al latido que estamos analizando.
- Presencia de la onda P: esta onda no se necesita para determinar enfermedades, pero su presencia o no, nos permite determinar un latido u otro. Esta presencia la fijamos como 1 si está o 0 si no está presente. Para obtenerla empleamos la misma función de antes implementada por la biblioteca *Neurokit2*, esta vez nos indicaba la presencia o no de la onda P en forma de muestras que después transformábamos a 1 o 0.
- Intervalo ST: este intervalo se establece como el final del complejo QRS y el final de la onda T. También lo podemos definir y calcular como la suma que existe entre la duración de la onda T y el segmento ST. Estos dos intervalos nos aportan información muy útil para clasificar latidos y para el diagnóstico de enfermedades. Finalmente el intervalo ST fue calculado empleando de la biblioteca *NeuroKit2*, la función *ec_delineate()*, de donde se extrajo el inicio de la onda T y el final del complejo QRS para determinar el segmento ST mediante

una resta, después se obtuvo el inicio y final de la onda T y ambos resultados se sumaron para determinar la duración del intervalo ST.

- Etiquetas: que muestran el tipo de latido.

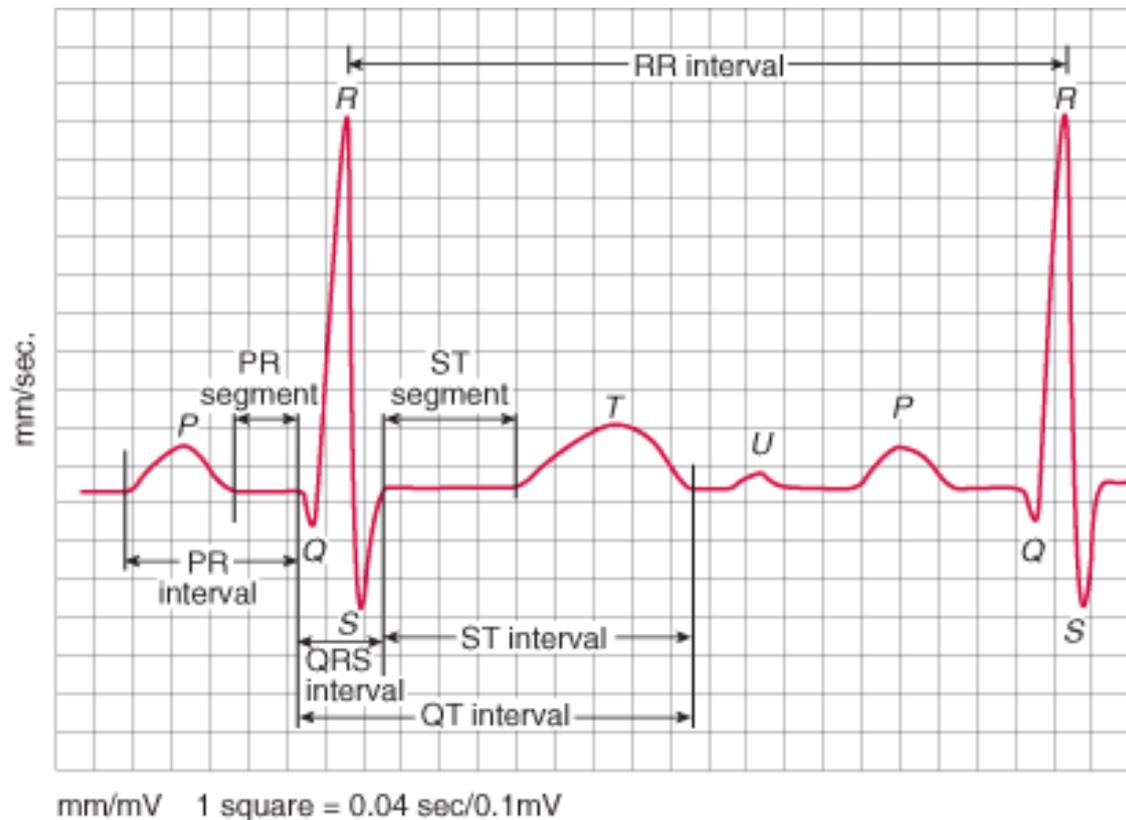


Figura 10: Representación de los intervalos de un latido [10].

- Características (morfológicas) del latido: se aplicó una ventana de muestreo sobre cada latido. Esta abarcaba desde el inicio del complejo QRS hasta el final de la onda T. Esto nos permitió separar cada latido del corazón. Esto se hizo para reducir el número de muestras de entrada para nuestra red neuronal. Se realizó empleando la función *epochs_create()* que extraía las épocas donde ocurrían eventos estableciendo un inicio y final como 100ms antes del pico R y unos 500ms después. Una vez hecho esto se volvió a muestrear la señal con *resample_sig()*. Finalmente se obtuvo unas 20 muestras por latido. Este proceso lo podemos ver representado en las siguientes figuras.

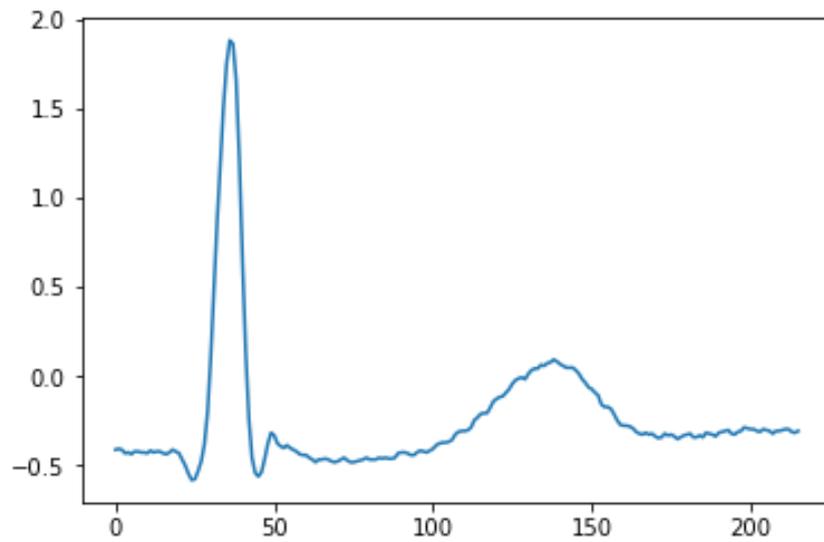


Figura 11: Latido original antes del muestreo donde el eje de ordenadas representa los (mV) y el eje de abscisas representa las muestras.

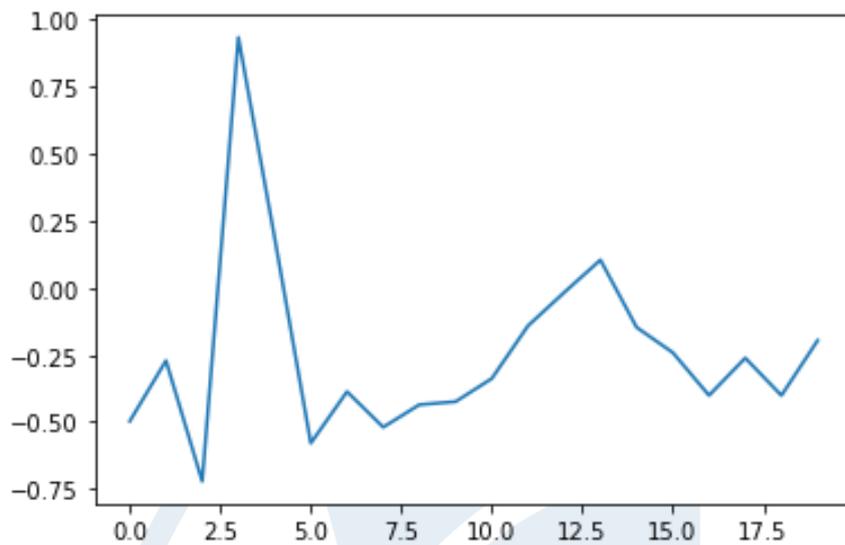


Figura 12: Latido después de volver a muestrear donde el eje de ordenadas representa los (mV) y el eje de abscisas representa las muestras.

3.1.2. Creación del Dataset.

Una vez extraídas todas las características como hemos mostrado en los apartados anteriores conseguimos un total de 28 incluyendo las etiquetas que indican el tipo de latido. Hay que matizar de que existen intervalos que no se han podido calcular, como lo

observamos al principio y final de los intervalos pre RR y post RR. Es decir, como no hay un latido anterior al primer latido este no se puede calcular, y lo mismo ocurre para el último latido en el cálculo de post RR. Para ello se aplicó la función *SimpleImputer()* para poder completar los valores faltantes. Además, se realizó una estandarización de los datos, que eliminaba la media y un escalado para conseguir que su varianza fuera igual a 1. Una vez que se había aplicado dicha estandarización se concateno con las etiquetas que indicaban el tipo de latido. Esta función era *StandardScaler()*. En un principio se elaboró un dataset con 53373 filas con 28 columnas. Cada fila corresponde a un latido y las columnas a las diferentes características que presenta. De la columna 0 hasta la 19 se ven reflejadas las características morfológicas extraídas tras el muestreo, la columna 20 muestra la presencia de la onda P, la 21 la duración del complejo QRS, la 22 el intervalo ST, la 23 el intervalo pre_RR, la 24 el intervalo post_RR, la 25 el valor medio de los intervalos pre_RR, la 26 corresponde al valor medio local de los intervalos pre_RR y la columna 27 muestra el tipo de latido.

	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21
0	0.611552	1.068155	0.521122	-1.019462	-0.555289	0.517548	0.415449	0.215524	0.527142	1.663539	...	-0.249448	-0.415882	-10.772467	0.0
1	-0.414872	-0.272866	-0.755218	-0.541414	-0.362811	-0.130512	0.034324	-0.095059	-0.091717	-0.22383	...	-0.213721	-0.228226	0.092829	0.0
2	-0.4627	-0.296238	-0.759037	-0.640389	-0.634333	-0.012101	-0.054479	-0.011868	-0.117159	-0.143057	...	-0.33553	-0.3836	0.092829	0.267918
3	-0.129611	0.177557	-0.699216	-0.393318	-0.142661	0.072858	0.344183	0.1997	0.138467	0.013048	...	0.005964	0.010164	0.092829	-0.446609
4	-0.106866	0.210389	-0.483101	-0.363139	-0.236071	0.162341	0.333985	0.254212	0.147979	0.054125	...	-0.04449	-0.034109	0.092829	0.125013
...
53368	0.098601	0.408499	0.090163	-0.47712	-0.588641	-0.098493	0.350126	0.615906	0.602888	0.440621	...	0.070278	-0.084323	0.092829	0.410824
53369	0.096082	0.392447	0.125092	-0.45147	-0.512214	-0.076558	0.316645	0.631596	0.62969	0.453686	...	0.044312	-0.152607	0.092829	-0.160798
53370	0.048538	0.402094	0.006438	-0.325089	-0.435892	-0.132901	0.251493	0.499378	0.586418	0.400888	...	-0.02894	-0.115118	0.092829	0.125013
53371	0.034652	0.272317	-0.047811	-0.4543	-0.588927	-0.240165	0.222673	0.475097	0.474725	0.33659	...	-0.050091	-0.111547	0.092829	-0.73242
53372	-0.130261	-0.067225	-0.231191	-0.448836	-0.546359	-0.518766	-0.169503	0.108256	0.165157	0.056388	...	-0.134553	-0.138483	0.092829	-0.303704

53373 rows x 28 columns

Figura 13: Dataset de 53373 filas y 28 columnas.

Sin embargo, se vio como algunos latidos presentaban valores nulos (NaN), ya que habíamos configurado que solo se etiquetaran aquellos que pertenecieran a nuestras clases a clasificar estableciendo los demás como NaN, por lo que se optó por eliminarlos ya que no se pueden introducir valores de este tipo en una red neuronal. Finalmente, se consiguió un dataset de 51643 filas y 28 columnas como se ve en la siguiente figura que

posteriormente se usaría para entrenar la red neuronal y por tanto conseguir una clasificación de latidos automática.

	0	1	2	3	4	5	6	7	8	9	...	18	19	20	21
1	-0.414872	-0.272866	-0.755218	-0.541414	-0.362811	-0.130512	0.034324	-0.095059	-0.091717	-0.223830	...	-0.213721	-0.228226	0.092829	5.077023e-16
2	-0.462700	-0.296238	-0.759037	-0.640389	-0.634333	-0.012101	-0.054479	-0.011868	-0.117159	-0.143057	...	-0.335530	-0.383600	0.092829	2.679184e-01
3	-0.129611	0.177557	-0.699216	-0.393318	-0.142661	0.072858	0.344183	0.199700	0.138467	0.013048	...	0.005964	0.010164	0.092829	-4.466091e-01
4	-0.106866	0.210389	-0.483101	-0.363139	-0.236071	0.162341	0.333985	0.254212	0.147979	0.054125	...	-0.044490	-0.034109	0.092829	1.250129e-01
5	-0.293370	-0.071871	-0.612923	-0.483704	-0.442968	0.011601	0.108431	0.065370	-0.080478	-0.179162	...	-0.162213	-0.158804	0.092829	-4.466091e-01
...
53368	0.098601	0.408499	0.090163	-0.477120	-0.588641	-0.098493	0.350126	0.615906	0.602888	0.440621	...	0.070278	-0.084323	0.092829	4.108239e-01
53369	0.096082	0.392447	0.125092	-0.451470	-0.512214	-0.076558	0.316645	0.631596	0.629690	0.453686	...	0.044312	-0.152607	0.092829	-1.607981e-01
53370	0.048538	0.402094	0.006438	-0.325089	-0.435892	-0.132901	0.251493	0.499378	0.586418	0.400888	...	-0.028940	-0.115118	0.092829	1.250129e-01
53371	0.034652	0.272317	-0.047811	-0.454300	-0.588927	-0.240165	0.222673	0.475097	0.474725	0.336590	...	-0.050091	-0.111547	0.092829	-7.324200e-01
53372	-0.130261	-0.067225	-0.231191	-0.448836	-0.546359	-0.518766	-0.169503	0.108256	0.165157	0.056388	...	-0.134553	-0.138483	0.092829	-3.037036e-01

51643 rows x 28 columns

Figura 14: Dataset de 51643 filas y 28 columnas con valores NaN eliminados.



CAPÍTULO 4

Redes Neuronales

4.1. Computación neuronal.

A lo largo de los años hemos visto como se ha intentado imitar el cerebro humano computacionalmente. Sabemos que las redes neuronales se han inspirado en la estructura que presentan las neuronas del cerebro humano, así como su estructura. Estas redes están formadas por un conjunto de unidades de procesamiento cuya función es transformar datos de entrada en datos de salida.

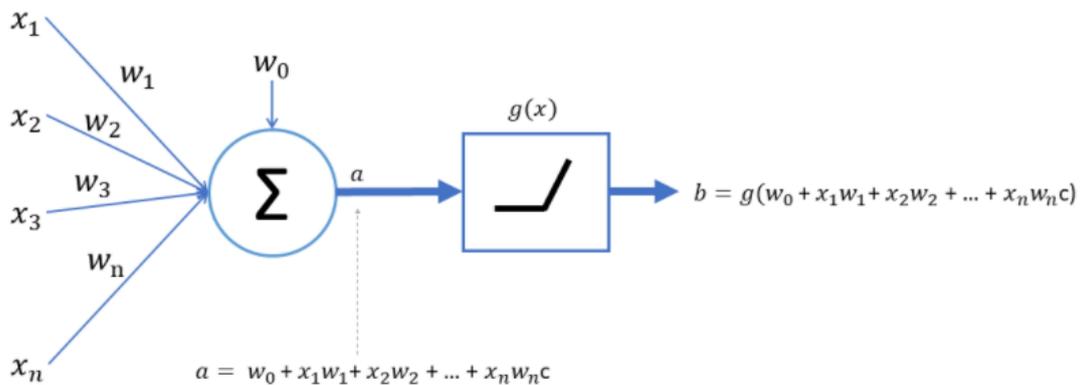


Figura 15: Funcionamiento de una neurona artificial [26].

El funcionamiento de una neurona simple [26] sería el siguiente:

- A cada dato de entrada x se le asigna un peso w , como podemos observar en figura. La neurona recibe estos datos, que son una multiplicación de cada dato por su peso y realiza una suma ponderada. A esta suma ponderada se le aplica un valor adicional llamado bias (w_0), que es un parámetro independiente. El resultado es la función lineal que podemos ver en la figura 15.
- Al resultado que se ha obtenido tras la operación anterior se le aplicará finalmente una función de activación (no lineal, normalmente), que distorsionará nuestro valor de salida añadiéndole deformaciones no lineales para poder encadenar de forma efectiva la computación de varias neuronas.

4.2. Elementos básicos de una red neuronal.

Como sabemos las redes neuronales, intentan imitar el proceso que se genera en el cerebro. Estas se componen de una:

- **Función de entrada:** las neuronas combinan todos los valores que reciben como uno solo. Como hemos mencionado anteriormente, los valores de entrada se multiplican por aquellos pesos que se han asignado previamente a las neuronas. Hay que decir que estos pesos permiten aportar mayor o menor influencia a esos valores de entrada. Si estos son pequeños, su influencia será pequeña, y lo mismo ocurre, al contrario [28].
- **Función de activación:** determina el valor de la salida dependiendo de si el valor de entrada está por encima o por debajo de un umbral [28]. Podemos encontrar diferentes funciones de activación como son la RELU, empleada en este proyecto, la sigmoide, la tangente hiperbólica etc.

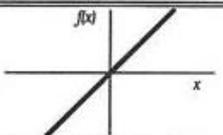
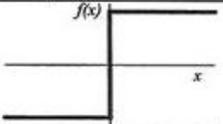
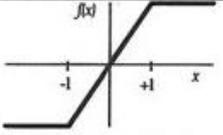
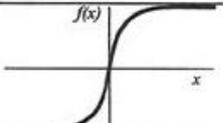
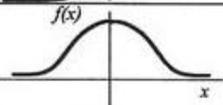
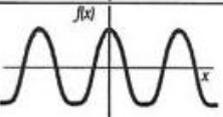
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & \text{si } x < -1 \\ x, & \text{si } -1 \leq x \leq +1 \\ +1, & \text{si } x > +1 \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0, +1]$	
Sinusoidal	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

Figura 16: Ejemplos de funciones de activación [30].

- **Función de pérdidas:** la función de pérdidas o función de coste nos va a permitir medir el error que se ha producido. El objetivo principal del entrenamiento es poder reducir dicho error, en pocas palabras obtener el mejor modelo [29].
- **Optimizador:** es el método que se empleara para conseguir reducir la función de pérdidas [29].

4.3. Arquitecturas de redes neuronales.

Definimos la arquitectura [32] de una red neuronal, como la forma en la que se conectan todas las neuronas entre sí. La colocación de estas neuronas se establece mediante capas, las cuales están formadas por un número establecido de neuronas. Podemos distinguir varios tipos:

- **Capa de entrada:** se compone de neuronas cuya función es recibir los datos que se introducen a la red neuronal.
- **Capa oculta:** esta capa, se compone de neuronas que no tiene ninguna conexión con el exterior, es decir las entradas a esta capa representan las salidas de la capa anterior mientras que sus salidas representan las entradas de la siguiente capa. En el caso de estar en una red con una sola capa, esta puede estar formada por un número de neuronas que varía según el diseño de nuestra red neuronal. Hay que añadir, que esto también ocurre en redes neuronales multicapa.
- **Capa de salida:** esta última capa está representada por neuronas que muestran el resultado generado por la red neuronal [32].

La siguiente figura 17, muestra la distribución de estas capas en una red neuronal.

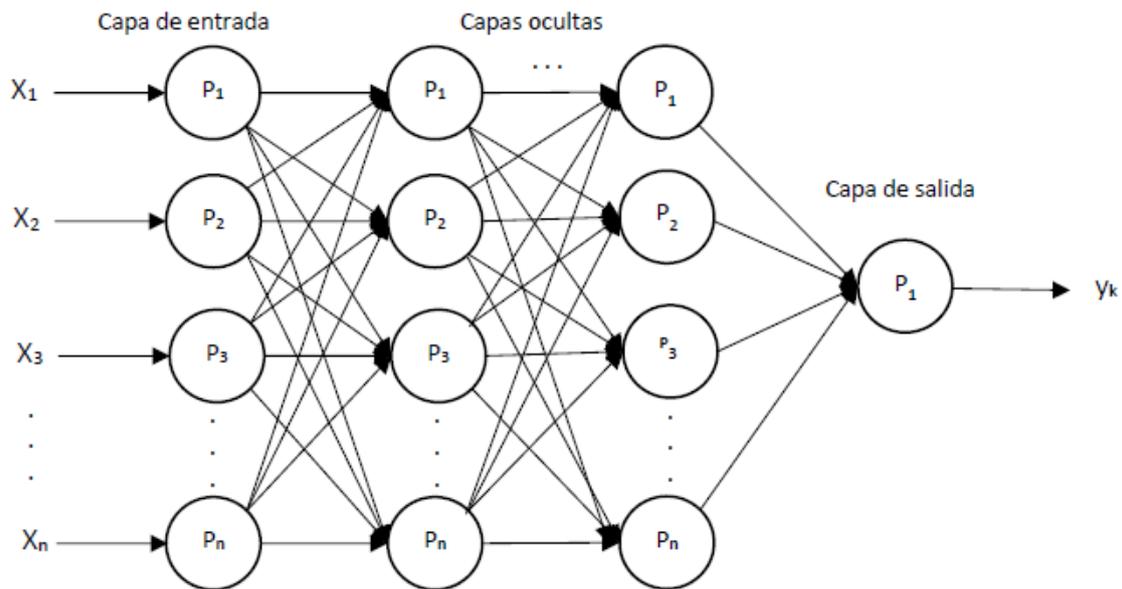


Figura 17: Estructura de una red neuronal [31].

Analizado dicha arquitectura, se puede hacer una distinción entre redes monocapa y redes multicapa. Además, dependiendo de cómo circule la información se distinguen redes unidireccionales como son las feedforward, cuya información se transmite en un único sentido, y las redes recurrentes feedback, cuya información circula en cualquier dirección entre las capas. En nuestro proyecto se emplearán las feedforward. Hay que añadir que el diseño de una red neuronal, tanto el número de capas, como de neuronas usado no sigue unas normas preestablecidas. Alcanzar el diseño óptimo para una red es mediante el ensayo y error.

4.4. Tipos de aprendizaje automático.

A diferencia de los sistemas clásicos, las redes neuronales emplean mecanismos de aprendizaje que les permiten aprender y evolucionar para finalmente realizar predicciones a partir de unos datos de entrada. Este proceso de aprendizaje permite a la red neuronal ajustar los diferentes pesos que hay en sus conexiones para poder dar una respuesta.

Podemos destacar dos grandes tipos [33] de aprendizaje automático:

- Aprendizaje supervisado: se trata de un modelo en el cual, partiendo de unos datos de entrada que se encuentran etiquetados, es decir que conocemos además sus

salidas, es capaz de hacer predicciones sobre datos que no se han visto antes. En pocas palabras, el supervisor es el encargado de vigilar si la salida de la red coincide con la que estamos buscando. En el caso de que no coincida se modificaran los pesos, para intentar de que el resultado se aproxime lo máximo posible al resultado deseado.

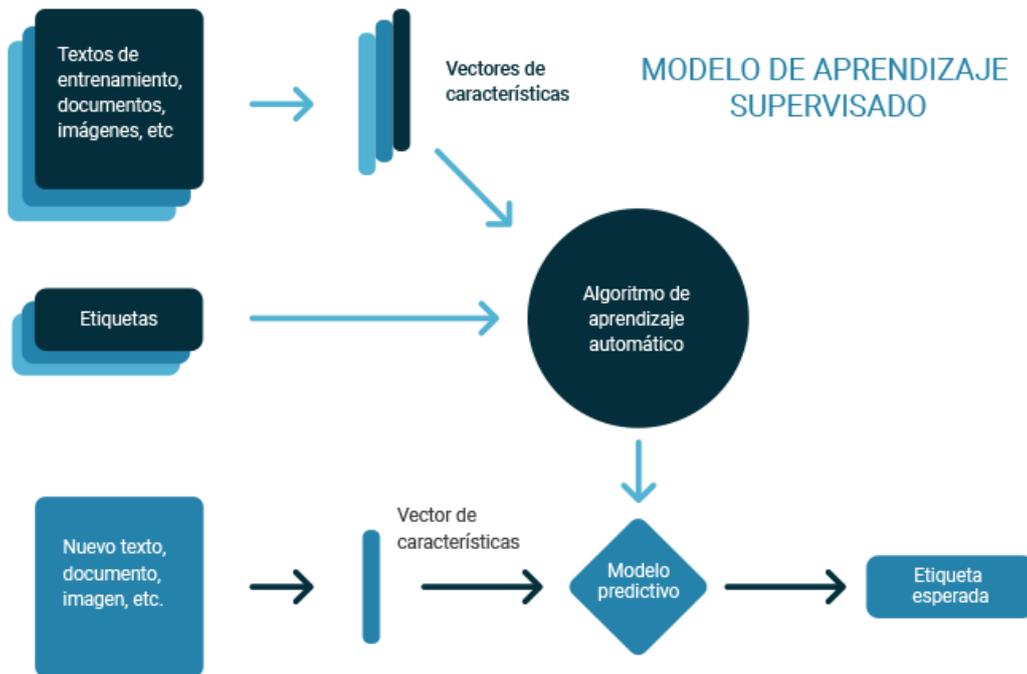


Figura 18: Estructura del modelo de aprendizaje supervisado [27].

- Aprendizaje sin supervisión: se trata de un modelo en el cual, se parte de unos datos que no están etiquetados, es decir que no conocemos sus salidas, y a partir de estos somos capaces de realizar predicciones. No existe para este modelo ningún agente externo que indique si la respuesta dada es correcta o no. En su lugar, usa técnicas de agrupamiento, alrededor de características o categorías que existan entre los datos entrada que introducimos, agrupándolos en diferentes tipos (clúster) [28].

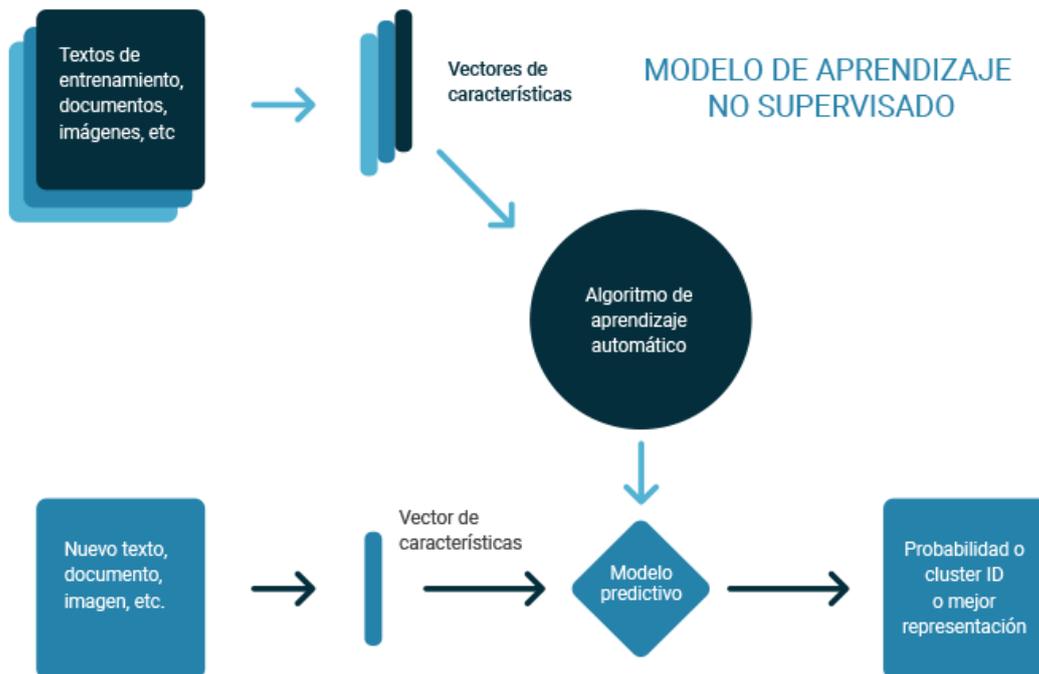


Figura 19: Estructura del modelo de aprendizaje no supervisado [27].

4.5. Estructura de la red neuronal.

El estructurar una Red Neuronal, en cuanto a diseño y número de neuronas presentes en las capas ocultas, es uno de los principales problemas que una persona puede afrontar en el ámbito del desarrollo de una red neuronal, puesto que no hay una regla exacta que indique con seguridad que pasos hay que seguir de manera generalizada. Es por ello por lo que el usuario deberá hacer uso de técnicas de prueba y error para dar finalmente con la red más óptima que ofrezca los mejores resultados.

En nuestro caso, como se explicó con anterioridad, nuestros datos de entrada de la red neuronal serán las primeras 27 características extraídas de cada latido. Y las salidas corresponden a la última característica (etiquetas).

RED NEURONAL



Figura 20: Diagrama de la red neuronal.

La figura (20) muestra la estructura de la red neuronal creada para este proyecto.

Sistema de detección y clasificación de latidos

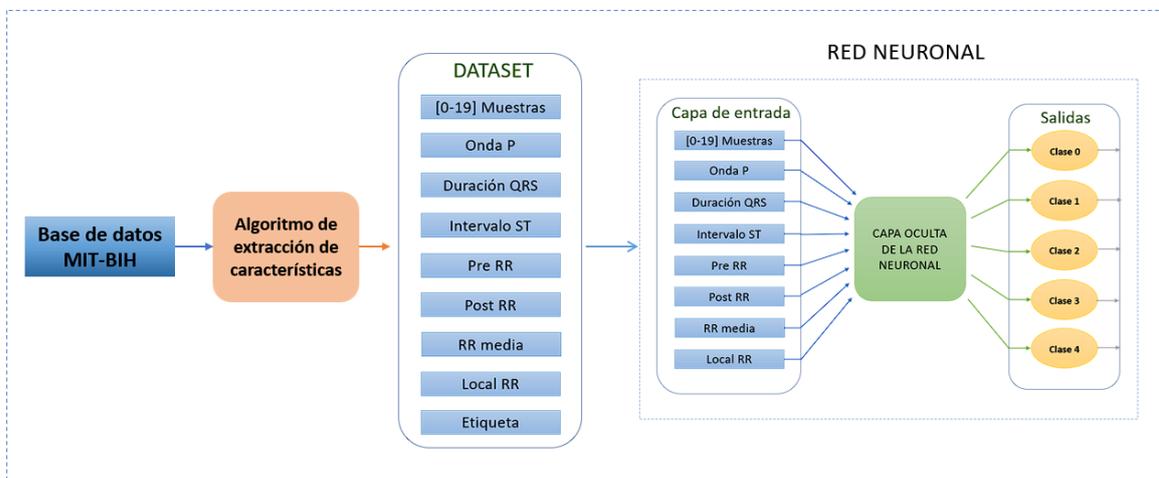


Figura 21: Diagrama del sistema de detección y clasificación de latidos.

En la última figura (21) puede verse el sistema de detección y clasificación de latidos desarrollado en el proyecto, partiendo de la base de datos MIT-BIH y de la extracción de sus características, hasta la elaboración de la red neuronal que nos permitirá clasificar los latidos en 5 tipos diferentes.

4.5.1. Descripción del modelo.

En primer lugar, se realizó un análisis para obtener el mejor modelo que pudiera resolver de forma más eficaz el objetivo del proyecto. Para ello se eligió una red neuronal de clasificación multiclase que se adaptaba mejor al problema en cuestión. Esta red consistía en una capa de entrada que contiene neuronas y estas representan los datos que se usarán en la red neuronal para entrenar (el número de neuronas depende del número de características que presentan los latidos). En nuestro caso se emplearon las 27 primeras características como datos de entrada. A continuación, se creó una capa oculta y finalmente una capa de salida con tantas neuronas como clases a resolver, que se encargará de entregar los resultados. En nuestro caso las salidas corresponden a la última característica de nuestro dataset (etiquetas). Este tipo de redes presentan conexiones unidireccionales con aprendizaje supervisado. Se estableció dos tipos de funciones de activación. La primera para la capa oculta fue la función ReLU, mientras que para la capa de salida se empleó una función Softmax. Esta última transforma las salidas a una representación en forma de probabilidades, de tal manera que el sumatorio de todas las probabilidades de las salidas de 1, permitiendo el reconocimiento de clases. Además, hay que añadir que se obtendrán datos de salida en forma de vectores indicando con 1 aquellas clases a las que se pertenezca y con 0 a las que no.

Una vez que se ha establecido el diseño de nuestra red neuronal se procederá a su entrenamiento, empleando diferentes números de neuronas en la capa oculta.

Finalmente se dedujo que la mejor configuración con los mejores resultados era empleando una función de pérdidas: Entropía Cruzada Categórica, que es una medida de precisión usada para variables categóricas [34]. Un optimizador: Adam, que permite actualizar los pesos para reducir la función de coste y por ello, reducir el error que existe entre los datos predichos y los datos reales. Este optimizador es uno de los más rápidos en minimizar el error. Se empleo además una codificación One-Hot-Encoding cuyo objetivo era crear una columna para cada valor distinto que exista en la característica que estamos codificando. Y finalmente se analizó para este modelo, diferentes situaciones en las cambiábamos el número de neuronas de la clase intermedia. Después de su análisis y de obtener que el modelo que mejor resultados ofrecía era el que presentaba 45 neuronas

en la capa oculta, se dio con un problema de desequilibrio de los datos que en un principio se pensaba que se podría resolver sin necesidad de modificar nada en nuestro modelo. Al ver que los resultados que se nos brindaban no compensaban esta desigualdad, se pusieron en marcha diferentes técnicas.

Al final se dedujo que la mejor forma de abordar dicho problema era aplicando una técnica de penalización para compensar. Esta se encargaba de equilibrar a las clases minoritarias durante el entrenamiento. Esto se consigue empleando el parámetro *class_weight* en modo balanceado que aportaba una mayor penalización a las predicciones erróneas en las clases minoritarias, de esa manera aportamos un mayor peso a esas clases con un menor número de muestras y equilibramos las clases [35], [36].

Para terminar los latidos fueron codificados para leer la respuesta de la red neuronal de la siguiente forma: Latido de fusión (F) = 0, Latido normal (N)= 1, Latido desconocido (Q)= 2, Latido ectópico supraventricular (S)= 3, Latido ectópico ventricular (V)= 4.

4.5.2. División del Dataset.

Una vez que hemos definido y estructurado nuestra red, se empleó una biblioteca de código abierto escrita en Python llamada Keras que proporciona bloques complejos modulares sobre los que se pueden desarrollar modelos complejos de aprendizaje profundo. A partir de ahora el siguiente paso es dividir los datos en tres conjuntos: entrenamiento, validación y prueba.

- **Datos de entrenamiento:** son aquellos empleados para realizar un ajuste de todos los pesos y para entrenar nuestra red con el fin de dar una respuesta correcta.
- **Datos de validación:** son aquellos asignados para realizar la tarea de comprobación del proceso de entrenamiento.
- **Datos de prueba:** son aquellos utilizados para comprobar y validar el correcto funcionamiento de la red neuronal una vez finalizado el entrenamiento, en el momento en el que los pesos se encuentran ya fijados.

No hay que olvidar de que la técnica empleada para la detención del entrenamiento en nuestro modelo ha sido la función *EarlyStopping()*, cuya utilidad es la de monitorear la métrica perdidas y comprobar si ha dejado de disminuir o no. Una vez que detecta que no disminuye más (después de 3 épocas), el entrenamiento se detiene.

La proporción en la que se han dividido nuestros datos ha sido: 80% para los datos de entrenamiento, donde una vez introducidos todos los datos al modelo, se irán actualizando los pesos en cada época, y un 20% para los datos de testeo, este conjunto de datos no los ha visto nunca modelo. Posteriormente de los datos de entrenamiento se ha retirado un 20% que nos servirá para la validación y comprobar si se ha disminuido la función de coste o no. Además, se ha dividido los diferentes conjuntos de datos de manera aleatoria para conseguir un mejor resultado. Los datos entrada pertenecen a las 27 características seleccionadas y obtenidas en apartados anteriores, mientras que los datos salida pertenecen a la última columna de nuestro dataset que nos proporciona el etiquetado de cada latido.

Finalmente, al final del proceso, sabremos con seguridad si la separación de los datos ha sido la adecuada.

La asignación del conjunto de datos se muestra en la siguiente tabla.

Dataset	Número de latidos
Conjunto de entrenamiento	33052
Conjunto de validación	8262
Conjunto de testeo	10329
Total:	51643

Tabla 3: Asignación de los datos a los diferentes conjuntos.

CAPÍTULO 5

Resultados y Discusión.

5.1. Introducción:

En este capítulo nos centraremos en analizar los resultados que hemos ido obteniendo a partir de la clasificación anterior realizada, empleando la base de datos de arritmias MIT-BIH siguiendo el criterio establecido por AAMI. Todo esto con herramientas de machine learning.

Para comenzar determinaremos antes del proceso de clasificación el número de latidos de cada tipo que tenemos para cada clase. En la siguiente tabla podemos visualizar dichos latidos.

Clase:	Número de latidos
Latidos normales	42917
Latido ectópico supraventricular	2341
Latido ectópico ventricular	5582
Latidos de fusión	788
Ritmo desconocido	15

Tabla 4: Clasificación de los latidos según su tipo.

5.2. Análisis de los modelos:

Modelo 1: 55 neuronas	Clase	Precisión %	Recall %
Accuracy: 93.81%	0	45	9
	1	96	98
	2	0	0
	3	93	83
	4	83	78

Modelo 2: 50 neuronas	Clase	Precisión %	Recall %
Accuracy: 93.61%	0	50	13
	1	95	98
	2	0	00
	3	92	82
	4	86	76
Modelo 3: 45 neuronas	Clase	Precisión %	Recall %
Accuracy: 94.08%	0	40	8
	1	95	99
	2	0	00
	3	92	83
	4	86	77
Modelo 4: 40 neuronas	Clase	Precisión %	Recall %
Accuracy: 92.85%	0	49	15
	1	95	97
	2	0	0
	3	94	83
	4	85	76

Tabla 5: Análisis de los modelos desbalanceados.

En la tabla anterior podemos identificar tres componentes importantes, dos para cada clase y un parámetro global. En primer lugar, la precisión nos muestra la relación que existe entre los latidos detectados correctamente (verdaderos positivos) y los resultados positivos, que es la suma de los verdaderos positivos y falsos positivos es decir es el porcentaje de casos positivos detectados. Estos falsos positivos son los latidos que se detectan como que pertenecen a una clase, pero en realidad no es así. En segundo lugar, podemos encontrar el recall que es la sensibilidad de nuestro modelo. Esta muestra la relación existente entre los latidos detectados correctamente con los verdaderos positivos y los falsos negativos. Estos falsos negativos son los latidos que se detectan como que no pertenecen a una clase, pero en realidad sí que pertenecen. Este parámetro en el campo de la medicina presenta una gran relevancia y muestra el porcentaje de casos positivos detectados correctamente. Y por último encontramos el accuracy, que es la exactitud de

nuestro modelo. Esta exactitud relaciona tanto valores del recall como la precisión y los verdaderos negativos (latidos que se detectan como que no pertenecen a una clase y no pertenecen a esa clase) permitiendo conocer la cantidad de predicciones positivas que fueron correctas (lo cerca que esta el resultado del valor verdadero).

Tras un análisis variando el número de neuronas en la capa oculta el modelo que nos aporta una mejor exactitud es la red neuronal con una capa de 45 neuronas. Esta exactitud es de 94.08%. Estamos ante un modelo con una alta precisión y un bajo recall, lo que se traduce como un modelo que no capaz de detectar tan bien las clases como nos gustaría, pero cuando lo hace, lo hace con una alta fiabilidad. Por tanto, nos disponemos a analizarlo con más detenimiento.

5.2.1. Modelo 3:

Durante nuestro entrenamiento se generó un histórico con todos los valores que ha ido tomando la función de coste y además la exactitud tanto del conjunto de datos de entrenamiento como el de validación.

De aquí obtenemos dos representaciones que nos muestran cómo se comporta nuestra red.

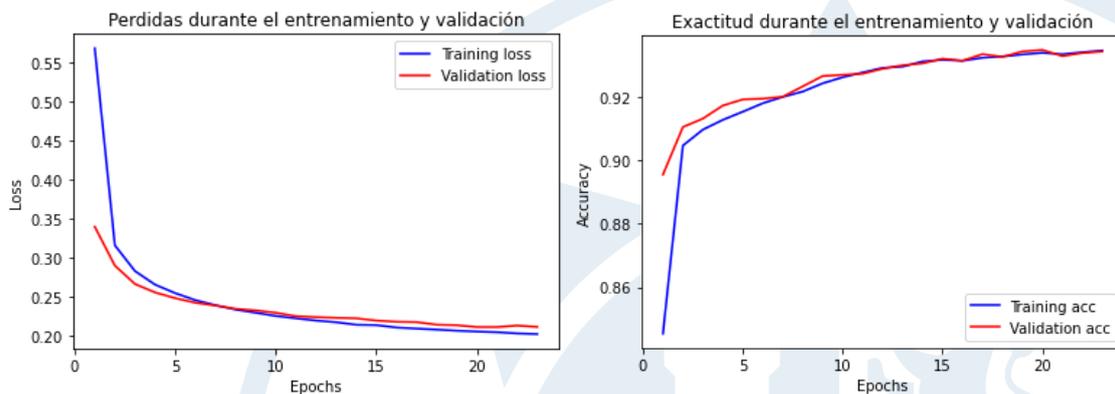


Figura 22: Representación de la evolución de las pérdidas y la exactitud.

Como observamos, en la figura 22, la de la función de costes, donde el eje y son el número de pérdidas y el eje x nos informa del número de iteraciones (épocas), la red aprende rápidamente los parámetros (pesos y bias) que le permiten disminuir el coste, por eso

vemos como decrece con facilidad, tanto la función de coste sobre el conjunto de validación como la misma sobre el conjunto de entrenamiento, donde ambas curvas se ajustan generalizando los datos aparentemente bien. La segunda gráfica nos muestra cómo evoluciona la exactitud del modelo con respecto a las épocas, empleando el conjunto de datos de entrenamiento y el de validación. Cabe añadir que ambas graficas se detuvieron justo antes de que empeorara el coste, tal y como se configuro en nuestra red neuronal por lo que no se observa una reaccionan negativa en las mismas, es decir evitamos un sobreajuste de los datos (un sobre entrenamiento), que haría que aumentara el error.

5.2.1.1. Resultados del testeo:

Después del entrenamiento y la validación de nuestros datos se obtuvo unos resultados del testeo.

Número de muestras = 10329	
Número de latidos por clase.	Latidos clasificados correctamente por clase.
Clase 0 = 150	8.00%
Clase 1 = 8585	98.50%
Clase 2 = 4	0.00%
Clase 3 = 463	83.15%
Clase 4 = 1127	76.66%

Tabla 6: Resultados del testeo del modelo con desbalanceo.

Analizando la matriz de confusión vemos como en la primera clase, se clasificaron mal 138 latidos de 150 latidos, es decir 12 latidos clasificados correctamente, lo que significó el 8.00%. En la segunda clase se clasificaron erróneamente 128 latidos de 8585 latidos, es decir 8457 latidos clasificados correctamente, lo que significó el 98.50%. En la tercera clase se clasificaron mal 4 latidos de 4 latidos lo que significó un 0.00% con ningún latido clasificado correctamente. En la 4 clase, se clasificaron mal 78 latidos de 463 latidos, es decir 385 latidos clasificados correctamente que supuso el 83.15% y en la última clase, se clasificaron mal 263 latidos de 1127 latidos es decir 864 latidos clasificados correctamente, que supuso un 76.66%.

Del número total de latidos clasificados correctamente conseguimos el global de 94.08% para el conjunto de datos de testeo.

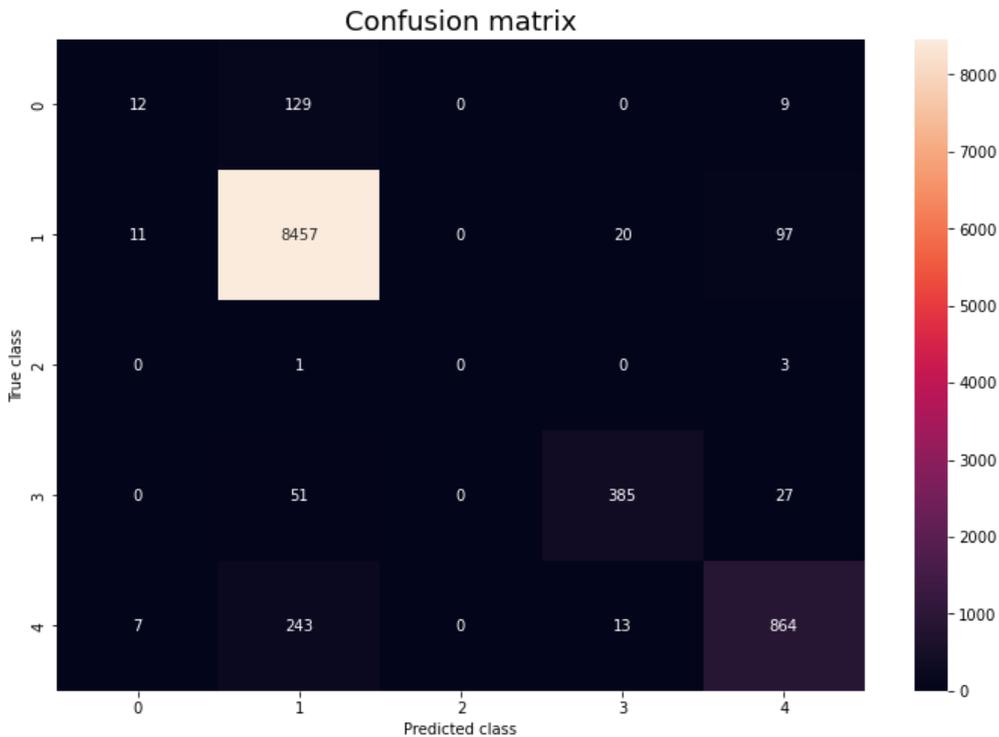


Figura 23: Matriz de confusión para el modelo con desbalanceo.

La matriz de confusión está compuesta por tres parámetros fundamentales. En primer lugar, los elementos de la diagonal muestran los latidos detectados correctamente. En segundo lugar, para cada clase, los latidos del eje de ordenadas representan los falsos positivos, mientras que los latidos del eje de abscisas representan los falsos negativos. La suma total de latidos del eje horizontal representa la suma total de latidos para una clase (valores reales), mientras que el eje vertical muestra los valores predichos para cada clase.

5.3. Modificación:

Como ya hemos mencionado anteriormente, el campo del aprendizaje automatizado es un campo de ensayo y error, hasta conseguir finalmente el mejor resultado. En este apartado se comentarán todas aquellas modificaciones realizadas y se expondrán y explicarán los resultados obtenidos.

5.3.1. Análisis de los modelos:

Aunque en el modelo anterior se obtuviera una exactitud elevada de unos 94.08%, este no funciona correctamente. Esto ocurre por el sobre entrenamiento del conjunto de datos de la clase mayoritaria (clase 1), provocado por el desequilibrio de los conjuntos de datos existentes, que hace que se detecten latidos de otras clases, asignándolos a la clase mayoritaria cuando en realidad no pertenecen a ella, es decir un aumento de los falsos negativos del resto de clases. Este parámetro necesitamos reducirlo lo máximo posible en el campo de la medicina, para poder detectar a tiempo las diferentes enfermedades. Estaríamos por tanto ante un modelo que no detecta tan bien las clases, pero cuando lo hace, lo hace con una alta fiabilidad. Esto ocurre porque los modelos siempre optimizan las predicciones basándose en la clase con un mayor número de muestras, consiguiendo de esta forma minimizar mucho más la función de coste, dando un falso resultado de exactitud elevada.

Por ello, nuestro objetivo es aumentar nuestra sensibilidad. Se plantearon técnicas como reducir las muestras de las clases mayoritarias para compensar. Sin embargo, se vio que esto podría ocasionar un mal funcionamiento de la red neuronal, pues se podrían eliminar muestras con características relevantes que ayudarían a la identificación de los latidos. De esta forma, se descartó esa técnica y se optó por introducir un nuevo parámetro, *class_weight*, para poder compensar la desigualdad entre clases. Un problema bastante común en el ámbito de la medicina que afecta a los algoritmos en su proceso de generalización de la información y perjudicando a las clases minoritarias.

Una vez ajustado el parámetro, que proporcionaba un mayor peso a los datos de las clases minoritarias, se volvió a entrenar la red con diferentes números de neuronas en la capa oculta. En la siguiente tabla se muestran dichos resultados.

Modelo 1: 55 neuronas	Clase	Precisión %	Recall %
Accuracy: 82.64%	0	12	93
	1	99	82
	2	0	0
	3	62	91

	4	65	83
Modelo 2: 50 neuronas	Clase	Precisión %	Recall %
Accuracy: 82.49%	0	12	89
	1	99	82
	2	0	0
	3	61	89
	4	65	83
Modelo 3: 45 neuronas	Clase	Precisión %	Recall %
Accuracy: 83.70%	0	13	93
	1	99	83
	2	00	0
	3	62	91
	4	71	83
Modelo 4: 40 neuronas	Clase	Precisión %	Recall %
Accuracy: 79.63%	0	11	93
	1	99	79
	2	2	25
	3	56	89
	4	63	79

Tabla 7: Análisis de los modelos modificados.

Tras el análisis variando el número de neuronas extraemos que el modelo 4 consigue en cierto modo, disminuir los falsos negativos en la clase 2 donde antes no obteníamos ningún resultado. Aspecto muy importante en el campo de la medicina. Sin embargo, puesto que buscamos la mejor relación entre los valores de la precisión, el recall y los verdaderos negativos, al ver que el incremento en la clase 2 mencionada antes no presenta una mejora relevante, seleccionamos el modelo 3 que nos aporta la mejor exactitud con un 83.70%. Por ello nos disponemos a analizarlo.

5.3.1.1. Modelo 3:

Nuevamente el histórico generado con todos los valores que ha ido tomando la función de coste y además la precisión tanto del conjunto de datos de entrenamiento como el de validación es el siguiente.

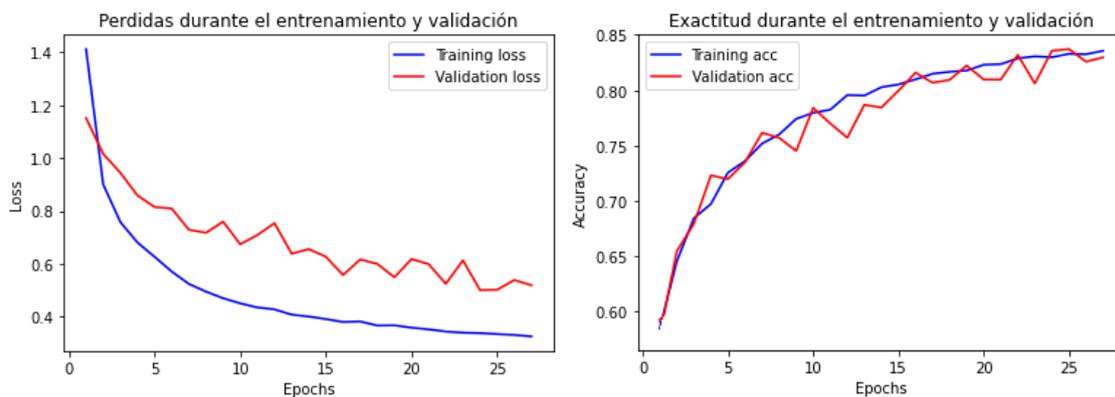


Figura 24: Representación de la evolución de las pérdidas y la exactitud.

Como observamos, en la figura 24, la de la función de costes, donde el eje y son el número de pérdidas y el eje x nos informa del número de iteraciones (épocas), la red no aprende tan rápido, como ocurría en el modelo anterior, los parámetros que le permiten disminuir el coste, tardando más épocas en conseguirlo. En el modelo anterior (sin compensación) veíamos las curvas juntas lo que aparentemente significaba que el modelo era capaz de generalizar realizando nuevas predicciones correctamente. Pero esto no era así ya que el modelo al tener los datos descompensados y por tanto una presencia mayor de la clase de latidos normales, tanto en el conjunto de datos de entrenamiento como el de validación, se optimizaba las predicciones basándose en esa clase con mayor número de muestras. De esta forma se conseguía reducir la función de perdidas, pero sin tener en cuenta las clases minoritarias. En cambio, en este modelo, aunque las curvas no están lo suficientemente juntas como nos gustaría, sí que es verdad que ofrece resultados más realistas y mejores ya que hemos compensado la desigualdad entre clases, permitiendo generalizar de forma correcta y detectar un mayor número de latidos de otras clases. La segunda gráfica nos muestra cómo evoluciona la exactitud del modelo con respecto a las épocas, empleando el conjunto de datos de entrenamiento y el de validación. Cabe añadir que ambas graficas se detuvieron justo antes de que empeorara el coste, tal y como se configuro en nuestra red neuronal por lo que no se observa una reaccionan negativa en las mismas.

5.3.1.2. Resultados del testeo:

Después del entrenamiento y la validación de nuestros datos se obtuvo unos resultados del testeo.

Número de muestras = 10329	
Número de latidos por clase.	Latidos clasificados correctamente por clase.
Clase 0 = 150	92.66%
Clase 1 = 8585	83.29%
Clase 2 = 4	0.00%
Clase 3 = 463	91.14%
Clase 4 = 1127	82.87%

Tabla 8: Resultados del testeo del modelo modificado.

Analizando la matriz de confusión vemos como en la primera clase, se clasificaron mal 11 latidos de 150 latidos, es decir, 139 latidos clasificados correctamente, lo que significó el 92.66%. En la segunda clase se clasificaron erróneamente 1434 latidos de 8585 latidos, es decir, 7151 latidos clasificados correctamente, lo que supuso un 83.29%. En la tercera clase se clasificaron mal 4 latidos de 4 latidos lo que significó un 0.00% con ningún latido clasificado correctamente. En la 4 clase, se clasificaron mal 41 latidos de 463 latidos, es decir, 422 latidos clasificados correctamente, lo que supuso el 91.14% y en la última clase, se clasificaron mal 193 latidos de 1127 latidos, es decir, 934 latidos clasificados correctamente, lo supuso un 82.87%.

Del número total de latidos clasificados correctamente conseguimos el global de 83.70% para el conjunto de datos de testeo.

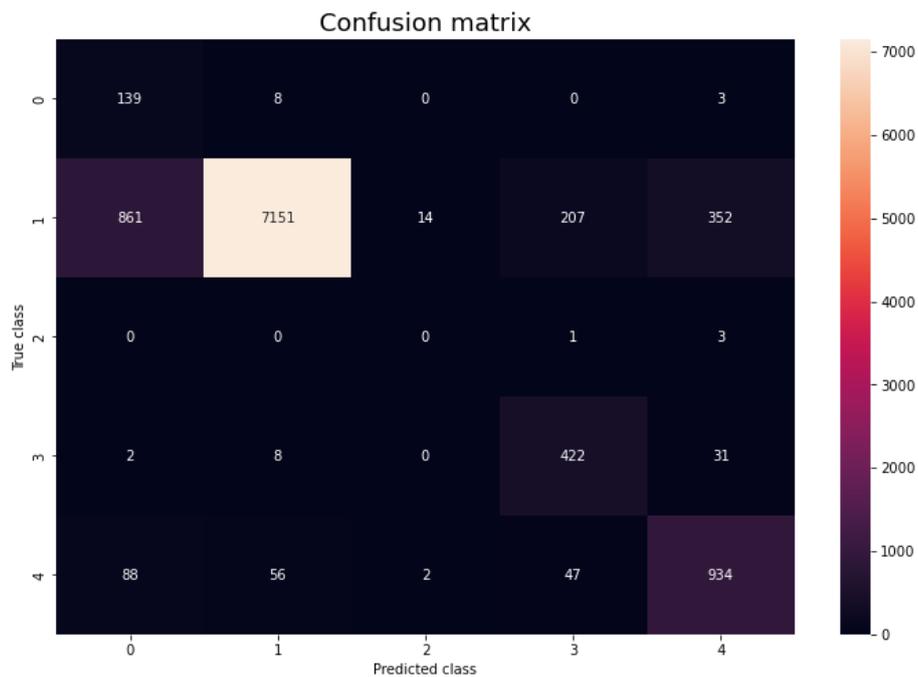


Figura 25: Matriz de confusión para el modelo modificado.

5.4. Comparación de los resultados:

En este apartado realizaremos una comparación de los resultados antes y después de las modificaciones.

5.4.1. Comparación del antes y después de las modificaciones.

Se realizará una comparación de los dos modelos seleccionados para este proyecto para comprender cual nos ofrece mejores características, visualizando el porcentaje de latidos detectados correctamente.

Antes de la modificación Modelo 3		Después de la modificación Modelo 3	
F	8.00%	F	92.66%
N	98.50%	N	83.29%
Q	0.00%	Q	0.00%
S	83.15%	S	91.14%
V	76.66%	V	82.87%
Total:	94.08%	Total:	83.70%

Tabla 9: Comparación del modelo sin modificar y el modelo modificado.

Se ha podido observar cómo tras la compensación de las diferentes clases minoritarias respecto de las clases con un mayor número de muestras, existe un incremento de sus latidos clasificados correctamente permitiendo así reducir los falsos negativos. Esto nos ayudará en la tarea de averiguar y detectar un mayor número de latidos de diferentes clases, a diferencia de lo que ocurría por ejemplo en el modelo sin compensación, donde latidos como los de fusión no se llegaban a detectar prácticamente.

A pesar de una mejoría en la detección y de una reducción de los falsos negativos, es decir latidos que se han detectado como si no pertenecieran a una clase, pero si pertenecen, nuestro porcentaje de latidos detectados correctamente desciende en el modelo 3 (con compensación) para la clase 1 de latidos normales, debido a que para conseguir equilibrar los datos se les ha dado una menor influencia a los pesos asociados a los datos pertenecientes a la clase mayoritaria. De esta forma conseguimos balancear los datos y una detección mayor de otras clases. Como consecuencia al reducir los falsos negativos del resto de clases hay un aumento de los falsos positivos, es decir, latidos que se han detectado como que pertenecen a una clase, pero en realidad no pertenecen a ella, sino que son de la clase de latidos normales. Estos falsos positivos no nos importan tanto ya que podrán reducirse empleando otras técnicas complementarias, como una segunda verificación por parte de un médico. Cabe añadir que el total de latidos detectados correctamente, en términos de proporción, es menor para el modelo con compensación. Esto ocurre debido únicamente a que la clase 1 de latidos normales presenta un gran número de muestras, lo que provoca, en el total, que un descenso del 15.21%, disminuya dicha proporción.

En resumen, detectamos un mayor número de latidos correctamente en las clases minoritarias a costa de reducir el número de latidos detectados correctamente en las clases con mayor número de muestras (latidos normales), todo este causado por la penalización.

Para finalizar, en la clase 2 (latidos desconocidos) no se han podido detectar latidos debido a una insuficiente recopilación de muestras.

CAPÍTULO 6

Conclusiones.

6.1. Conclusión.

Este proyecto ha permitido adquirir diversos conocimientos dentro del campo de la medicina, tanto el funcionamiento que presenta el corazón, la clasificación de sus latidos, adquiriendo un total de 28 características extraídas, como además el aprendizaje de un nuevo lenguaje de programación (Python), la creación de una base de datos y la implementación de redes neuronales.

Como hemos visto anteriormente, se ha podido resolver el problema presente en la base de datos de arritmias MIT-BIH, compensando la desigualdad entre clases y obteniendo una mejora en la detección de latidos en las clases minoritarias, a diferencia de otras investigaciones mencionadas en este proyecto. Hemos conseguido, a pesar de las limitaciones que supone la falta de muestras en diferentes clases, reducir el número de falsos negativos, lo que demuestra que estamos ante un modelo que ofrece datos positivos y que puede ser la base para futuras investigaciones.

Es por ello, que necesita tanto la recopilación de un mayor número de muestras, para completar mejor las diferentes clases, lo que nos arrojaría a nuestro modelo una mayor exactitud. Permittiéndonos implementar como segundo paso en futuras líneas de trabajo, el poder desarrollar un algoritmo que emplee esta red neuronal, como herramienta para la detección de patrones dentro de un ECG y de esa forma detectar diferentes arritmias, incluso anticiparse a ellas, para mejorar la calidad de vida de las personas.

Bibliografía

- [1] Myhealth.ucsd.edu. (2022). *Anatomía y funciones del sistema eléctrico del corazón* / UC San Diego Health. [online]. Obtenido de: <<https://myhealth.ucsd.edu/Spanish/RelatedItems/85,P03337>> [Accedido el 15 de agosto 2021].
- [2] Uofmhealth.org. (2022). *Electrical System of the Heart* / Michigan Medicine. [online] Obtenido de: <[https://www.uofmhealth.org/health-library/te7147abc#:~:text=The%20electrical%20signal%20starts%20in,the%20sinoatrial%20\(SA\)%20node.&text=The%20SA%20node%20\(called%20the,an%20impulse%20into%20the%20ventricles.](https://www.uofmhealth.org/health-library/te7147abc#:~:text=The%20electrical%20signal%20starts%20in,the%20sinoatrial%20(SA)%20node.&text=The%20SA%20node%20(called%20the,an%20impulse%20into%20the%20ventricles.)> [Accedido el 15 de agosto 2021].
- [3] Cardiosalud.org. (2022). *Como funciona el corazón* / Comisión Honoraria para la salud cardiovascular. [online]. Obtenido de: <<http://cardiosalud.org/corazon-y-salud/como-funciona-tu-corazon/#:~:text=%C2%BFC%C3%B3mo%20funciona%20el%20coraz%C3%B3n%3F,circuito%20cerrado%2C%20nada%20se%20pierde.>> [Accedido el 17 de agosto 2021].
- [4] Cigna.com. (2022). *El corazón y su sistema eléctrico*. [online]. Obtenido de: <<https://www.cigna.com/es-us/individuals-families/health-wellness/hw/el-corazn-y-su-sistema-elctrico-zm2272>> [Accedido el 17 de agosto 2021].
- [5] Southsudanmedicaljournal.com. (2022). *Como leer un electrocardiograma (ECG). Primera parte: Principios básicos del EC. El electrocardiograma normal*. [online] Obtenido de: <<http://www.southsudanmedicaljournal.com/archive/may-2010/how-to-read-an-electrocardiogram-ecg.-part-one-basic-principles-of-the-ecg.-the-normal-ecg.html>> [Accedido el 17 de agosto 2021].
- [6] Fisiosaludable.com. (2022). *Electrocardiograma*. [online]. Obtenido de: <<https://fisiosaludable.com/publicaciones/conceptos/247-electrocardiograma>> [Accedido el 27 de agosto 2021].
- [7] Wikipedia.org. (2022). *Electrocardiograma* / *Derivaciones periféricas y precordiales*. [online]. Obtenido de: <<https://es.wikipedia.org/wiki/Electrocardiograma#:~:text=El%20ECG%20se%20estructura%20en,brazo%20derecho%20y%20el%20izquierdo.>> [Accedido el 1 de septiembre 2021].
- [8] Pngwing.com. (2022). *Electrocardiograma extremidad Eindhoven triangula de 12 derivaciones ecg*. [online]. Obtenido de: <<https://www.pngwing.com/es/free-png-pxqra>> [Accedido el 1 de septiembre 2021].

- [9] Enfermeriabuenosaires.com. (2022). *¿Cómo se colocan los electrodos del electrocardiograma?* [online]. Obtenido de: <<https://enfermeriabuenosaires.com/como-se-colocan-los-electrodos-del-electrocardiograma/>>[Accedido el 1 de septiembre 2021].
- [10] Msdmanuals.com. (2022). *Electrocardiografia*. [online]. Obtenido de:<<https://www.msdmanuals.com/es-es/professional/trastornos-cardiovasculares/pruebas-y-procedimientos-cardiovasculares/electrocardiograf%C3%ADa>>[Accedido el 10 de septiembre 2021].
- [11] Wikipedia.org. (2022). *Electrocardiography*. [online] Obtenido de: <<https://en.wikipedia.org/wiki/Electrocardiography#:~:text=There%20thre%20main%20components,the%20repolarization%20of%20the%20ventricles.>>> [Accedido el 10 de septiembre 2021].
- [12] Girodmedical.es. (2022). *Como interpretar un electrocardiograma*. [online]. Obtenido de: <https://www.girodmedical.es/blog_es/como-interpretar-un-electrocardiograma/>[Accedido el 12 de septiembre 2021].
- [13] Physionet.org. (2022). *MIT-BIH Arrhythmia Database v1.0.0*. [online]. Obtenido de: <<https://physionet.org/content/mitdb/1.0.0/>> [Accedido el 27 de septiembre 2021].
- [14] Msdmanuals.com. (2022). Introducción a las arritmias. [online]. Obtenido de:<<https://www.msdmanuals.com/es-es/hogar/trastornos-del-coraz%C3%B3n-y-los-vasos-sangu%C3%ADneos/arritmias/introducci%C3%B3n-a-las-arritmias>>[Accedido el 15 de septiembre 2021].
- [15] Aami.org. (2022). *ANSI / AAMI EC57: 2012*. [online]. Obtenido de: <<https://www.aami.org/>>[Accedido el 15 de septiembre 2021].
- [16] Archive.physionet.org. (2022). *The WFDB Software Package*. [online] Obtenido de: <<https://archive.physionet.org/physiotools/wfdb.shtml>>[Accedido el 20 de septiembre 2021].
- [17] Keras.io. (2022). *About Keras*. [online]. Obtenido de: <<https://keras.io/about/>>[Accedido el 20 de septiembre 2021].
- [18] Cursosaula21.com. (2022) *Python* [online]. Obtenido de: <<https://www.cursosaula21.com/que-es-python/>> [Accedido el 20 de septiembre 2021].
- [19] Code.visualstudio.com. (2022). *Visual Studio*. [online]. Obtenido de: <<https://code.visualstudio.com/learn>>[Accedido el 20 de septiembre 2021].

- [20] Moody GB, Mark RG. The impact of the MIT-BIH arrhythmia database. *IEEE Eng Med Biol Mag.* 2001 May-Jun;20(3):45-50. doi: 10.1109/51.932724. PMID: 11446209.
- [21] De Chazal P, O'Dwyer M, Reilly RB. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Trans Biomed Eng.* 2004 Jul;51(7):1196-206. doi: 10.1109/TBME.2004.827359. PMID: 15248536.
- [22] C. Ye, B. V. K. Vijaya Kumar and M. T. Coimbra, "Heartbeat Classification Using Morphological and Dynamic Features of ECG Signals," in *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 10, pp. 2930-2941, Oct. 2012, doi: 10.1109/TBME.2012.2213253.
- [23] P. de Chazal and R. B. Reilly, "A Patient-Adapting Heartbeat Classifier Using ECG Morphology and Heartbeat Interval Features," in *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2535-2543, Dec. 2006, doi: 10.1109/TBME.2006.883802.
- [24] Ravier, P., Leclerc, F., Dumez-Viou, C., & Lamarque, G. (0AD). Redefining Performance Evaluation Tools for Real-Time QRS Complex Classification Systems. *IEEE Transactions on Biomedical Engineering*, 54(9), 1706–1710. <https://doi.org/10.1109/TBME.2007.902594>
- [25] T. H. Yeap, F. Johnson and M. Rachniowski, "ECG Beat Classification By A Neural Network," [1990] Proceedings of the Twelfth Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1990, pp. 1457-1458, doi: 10.1109/IEMBS.1990.691837.
- [26] Interactivechaos. (2022). *Redes Neuronales*. [online]. Obtenido de:<<https://interactivechaos.com/es/manual/tutorial-de-machine-learning/redes-neuronales>>[Accedido el 10 de octubre 2021].
- [27] Medium.com. (2022). *Tipos de aprendizaje automático*. [online] Obtenido de:<<https://medium.com/soldai/tipos-de-aprendizaje-autom%C3%A1tico-6413e3c615e2>>[Accedido el 10 de octubre 2021].
- [28] Damián Jorge Matich, "Redes Neuronales: Conceptos Básicos y aplicaciones," Universidad Tecnológica Nacional de Rosario, Rosario, Argentina, 2001.
- [29] Interactivechaos. (2022). *Optimización de un modelo*. [online]. Obtenido de:<<https://interactivechaos.com/es/manual/tutorial-de-machine-learning/optimizacion-de-un-modelo>>[Accedido el 10 de octubre 2021].

- [30] Ibiblio.org. (2022). *La neurona artificial*. [online]. Obtenido de:<https://www.ibiblio.org/pub/linux/docs/LuCaS/Presentaciones/200304curso-glisa/redes_neuronales/curso-glisa-redes_neuronales-html/x38.html>[Accedido el 10 de octubre 2021].
- [31] Scielo.cl. (2022). *Redes Neuronales Artificiales*. [online]. Obtenido de:<https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642020000100273>[Accedido el 10 de octubre 2021].
- [32] Pedro Larranaga, Iñaki Inza, Abdelmalik Moujahid. "*Tema 8. Redes Neuronales*". Departamento de ciencias de la computación e inteligencia artificial. Universidad del País Vasco–Euskal Herriko Unibertsitatea, 2007.
- [33] Machinelearningparatodos.com, (2022). *Tipos de aprendizaje automático*. [online]. Obtenido de:< <https://machinelearningparatodos.com/tipos-de-aprendizaje-automatiko/#:~:text=Los%20algoritmos%20de%20aprendizaje%20autom%C3%A1tico,o%20con%20aprendizaje%20por%20refuerzo>>[Accedido el 11 de octubre 2021].
- [34] DiegoCalvo. (2022). *Entropía cruzada categórica*. [online]. Obtenido de:<<https://www.diegocalvo.es/funcion-de-coste-redes-neuronales/#:~:text=La%20entrop%C3%ADa%20cruzada%20categ%C3%B3rica%20es,de%20precisi%C3%B3n%20para%20variables%20categ%C3%B3ricas.&text=Caracter%C3%ADsticas%20del%20Entrop%C3%ADa%20cruzada%20categ%C3%B3rica,Escala%20univariante.>> [Accedido el 20 de octubre 2021].
- [35] Aprendemachinelearning.com. (2022). *Clasificación con datos desbalanceados*. [online]. Obtenido de:< <https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/>> [Accedido el 20 de octubre 2021].
- [36] Sebastian Raschka, Vahid Mirjalili, "Python Machine Learning Aprendizaje automatico y aprendizaje profundo con Python, scikit-learn y TensorFlow " 2017, Marcombo, S.A.