



Global Knowledge
EDITORIAL

MODELOS COMPUTACIONALES PARA LA DETECCIÓN Y ANÁLISIS DE ARRITMIAS CARDÍACAS, APLICANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO



Alfonso Aníbal Guijarro Rodríguez
Jorge Alberto Medina Avelino
María Antonieta Touriz Bonifaz
José Luis Rodríguez Matías
Gladys Cristina Jácome Morales



Universidad de
Guayaquil



Carrera de Ingeniería en
Sistemas Computacionales

MODELOS

COMPUTACIONALES PARA LA DETECCIÓN
Y ANÁLISIS DE ARRITMIAS CARDÍACAS,
APLICANDO ALGORITMOS DE APRENDIZAJE
AUTOMÁTICO



Alfonso Aníbal Guijarro Rodríguez
Jorge Alberto Medina Avelino
María Antonieta Touriz Bonifaz
José Luis Rodríguez Matías
Gladys Cristina Jácome Morales

Copyright © 2023

Global Knowledge - Editorial CCGECON USA LLC. 

848 BRICKELL AVE STE 950 MIAMI, FLORIDA, US 33131

Móvil - (WhatsApp): (+1) 786 977 9421 (+593)99 211 8124

Página Web: <https://egk.ccgecon.us>

E-mail: egk@ccgecon.us

Modelos Computacionales para la detección y análisis de arritmias
cardíacas, aplicando algoritmos de aprendizaje automático.

Guayaquil - Ecuador, 2023

1ª edición

© Alfonso Aníbal Guijarro Rodríguez

Jorge Alberto Medina Avelino

María Antonieta Touriz Bonifaz

José Luis Rodríguez Matías

Gladys Cristina Jácome Morales

15,24 x 22,86 cm

Todos los derechos reservados.

ISBN: 978-1-957271-12-5

ÍNDICE GENERAL

AGRADECIMIENTOS.....	1
PRÓLOGO	3
INTRODUCCIÓN	5
CAPÍTULO 1:	9
EL CORAZÓN: FUNCIONAMIENTOS, TRASTORNOS Y HERRAMIENTAS TECNOLÓGICAS PARA LA DETECCIÓN DE PROBLEMAS.....	9
1.1. Anatomía.....	9
1.1.1. Esquema general de la circulación de la sangre.....	10
1.1.2. Esquema de circulación en las cavidades cardíacas.....	11
1.1.3. Las caras del corazón	12
1.1.4. Morfología externa	14
1.1.5. Morfología interna.....	14
1.2 Fisiología. Funcionamiento del Corazón.....	18
1.2.1. Funcionamiento eléctrico	19
1.2.2. El flujo de sangre a través del corazón.....	20
1.3 Trastornos cardiovasculares.....	21
1.3.1. Arritmias cardíacas	22
1.3.2. Enfermedad de las arterias coronarias	28
1.3.3. Defectos cardíacos congénitos	29
1.3.4. Presión arterial alta (hipertensión).....	29
1.3.5. Cardiopatía congénita.....	30
1.3.6. Arteriopatía periférica	30
1.3.7. Cardiopatía coronaria	32
1.4. Herramientas tecnológicas para la detección de problemas.....	32
1.4.1. Electrocardiograma	32
1.4.2. Monitor Holter.....	39
1.4.3. Monitor de Eventos	40

1.4.4. Alivecorkardia.....	41
1.4.5. Registrador de bucle implantable.....	42
1.5. Resumen	44
CAPÍTULO 2:	45
PRESENCIA DE LA INTELIGENCIA ARTIFICIAL EN ÁREAS DE LA INGENIERÍA Y SALUD MÉDICA.....	45
2.1. Inteligencia Artificial	45
2.1.1. Diferencia entre la Inteligencia Artificial y Machine Learning.....	46
2.1.2. Relación entre la Inteligencia Artificial y machine Learning .	47
2.1.3 Beneficios de aplicar inteligencia artificial con pacientes clínicos.....	47
2.1.4. La Inteligencia Artificial en desarrollo de fármacos	49
2.1.5 Evolución de la Inteligencia Artificial	49
2.2. Tipos de Algoritmos de Machine Learning	50
2.2.1. Aprendizaje Supervisado	52
2.2.2. Aprendizaje No Supervisado.....	57
2.2.3. Aprendizaje Profundo.....	60
2.2.4. Aprendizaje por refuerzo.....	60
2.2.5. Algoritmos de transferencia de aprendizaje.....	63
2.3. Los de datos (dataset)	67
2.3.1. Hiperparámetro.....	67
2.3.2. Datos de entrenamiento	68
2.3.3. Set de validación	68
2.4. Resumen	69
CAPÍTULO 3:	71
RECOPIACIÓN DE ALGORITMOS DE MACHINE LEARNING ENFOCADOS A LA DETECCIÓN DE ARRITMIAS CARDIACAS.....	71
3.1. Metodología de la investigación	71
3.2 Plataformas de Data Science.....	75

3.2.1. Que es Data Science	75
3.2.2. Kaggle	76
3.2.3. Mendeley Data	79
3.3. Redes neuronales	80
3.4. Algoritmo detección de arritmias usando ECG	86
3.5. Algoritmo categorización de subtipos de arritmia del ECG	87
3.6. Algoritmo RestNet.....	88
3.7. Resumen	88
CAPÍTULO 4:	91
Optimización de algoritmos de Machine Learning para la detección de arritmias cardiacas.	91
4.1. Configuración de entorno para optimización de algoritmos	91
4.2. Evaluación de los algoritmos	97
4.3. Algoritmo detección de arritmias usando ECG	98
4.4. Algoritmo categorización de subtipos de arritmia del ECG	104
4.5 Algoritmo RestNet.....	110
4.6. Resumen	118
CAPÍTULO 5:	119
EVALUACIÓN DEL RESULTADO DE ENTRENAMIENTO DE LOS ALGORITMOS MEDIANTE DATOS DE PRUEBA	119
5.1. Resultados obtenidos del algoritmo de detección de arritmias usando ECG	119
5.2. Resultados obtenidos del algoritmo categorización de subtipos de arritmia del ECG	123
5.3. Resultados obtenidos del algoritmo RestNet	124
5.4. Resumen.....	128
CONCLUSIÓN	129
REFERENCIAS BIBLIOGRÁFICAS.....	131

ÍNDICE TABLA

Tabla 1. Proyectos de optimización sobre enfermedades cardiovasculares.....	7
Tabla 2. Matriz de causas y consecuencias del problema.	22
Tabla 3. Tipos de arritmias cardíacas	25
Tabla 4. Valores de Tiempo y Voltaje de las ondas, intervalos, segmentos complejos electrocardiográficos.	34
Tabla 5. Tipos de monitor de eventos.....	40
Tabla 6. Habilidades requeridas en Inteligencia Artificial y Machine Learning..	46

ÍNDICE DE FIGURA

FIG 1.	Información general sobre el corazón. (Cardiología Madrid, 2023).....	10
FIG 2.	Esquema de circulación de la sangre en el corazón. (Latarget & Ruiz, 2019).....	11
FIG 3.	Esquema de la circulación intracardíaca. (Latarget & Ruiz, 2019).....	12
FIG 4.	Vista anterior del corazón y de los grandes vasos. (Latarget & Ruiz, 2019).....	12
FIG 5.	Morfología interna del corazón. (profesorenlinea, 2022).....	15
FIG 6.	Corte transversal que muestra diferencias en el espesor. (Gerard, 2018).....	17
FIG 7.	Funcionamiento del Corazón.	18
FIG 8.	El sistema eléctrico del corazón (myhealth, 2021)	20
FIG 9.	El flujo de sangre a través del corazón. (National Heart Lung and Blood Institute, 2022).....	21
FIG 10.	Arritmia cardíaca. Foto Tomada de (Castelloninformacion, 2020).....	23
FIG 11.	Ritmo cardíaco normal.	24
FIG 12.	Latidos anormales del corazón	24
FIG 13.	Arteriopatía periférica en las piernas. (Ortega et al., 2019)...	31
FIG 14.	Señal de Electrocardiograma. (Pozas, 2017).....	33
FIG 15.	Papel del Electrocardiograma	34
FIG 16.	Procesamiento de Señales	34
FIG 17.	Derivaciones bipolares y monopolares	38
FIG 18.	Mediblu. (Mediblu, 2021).....	39
FIG 19.	Monitor Holter. (Medineplus, 2022).....	39

FIG 20.	Toma de ECG con Kardia. Foto tomada de (Wegner et al., 2020).....	42
FIG 21.	Registrador de bucle implantable. (Solbiati et al., 2016).....	43
FIG 22.	Inteligencia artificial en medicina. Foto tomada de (IBM, 2022).....	45
FIG 23.	Beneficios de integrar IA en la medicina. (APD, 2021).....	48
FIG 24.	Evolución de la IA (Lazcano, 2020)	50
FIG 25.	Se explican los algoritmos de Machine Learning	50
FIG 26.	En esta figura podemos visualizar el procesamiento del Aprendizaje supervisado. (Aristega & Mendoza, 2022)	52
FIG 27.	Representación de Regresión Lineal.....	53
FIG 28.	Representación de curva sigmoïdal en Regresión Logística....	54
FIG 29.	Representación de Árbol de decisión y sus respectivos nodos.....	55
FIG 30.	Un bosque aleatorio se conforma por varios árboles de decisiones	56
FIG 31.	La frontera o el hiperplano separa las clases.....	56
FIG 32.	Se detalla la separación en distintos grupos bajo el clasificador Naive Bayes	57
FIG 33.	Algoritmo de Aprendizaje No Supervisado: K-Medias.(Caparrini, 2020).....	58
FIG 34.	Técnicas del aprendizaje no supervisado	58
FIG 35.	Alpha Zero es una inteligencia Artificial que juega al ajedrez haciendo uso de este aprendizaje	60
FIG 36.	Red Neuronal de convolución	61
FIG 37.	Diagrama que describe los pasos de la herramienta de clasificación. (Attallah, 2022)	64
FIG 38.	Arquitectura InceptionV3. (Google Cloud, 2022)	65

FIG 39.	Arquitectura de ResNetV1	66
FIG 40.	Arquitectura de MobileNetV2.	67
FIG 41.	Descripción de la clase de ECG mediante la norma AAMI.	77
FIG 42.	Estructura Semántica de Búsqueda	80
FIG 43.	Arquitectura de búsqueda de patrones de una red neurona convolucional. (Rahman et al., 2022)	80
FIG 44.	Aplicación de convolución en imagen. (Leivi, 2019)	81
FIG 45.	Función de activación Relu (SAN MARTÍN, 2020)	82
FIG 46.	Capa de Agrupación (Levi, 2019)	82
FIG 47.	Diagrama de flujo del Algoritmo detección de arritmias usando ECG. Elaborado por Steven Aristega y Carlos León.....	86
FIG 48.	Diagrama de flujo del algoritmo categorización de subtipos de arritmia del ECG	87
FIG 49.	Flujograma del algoritmo RestNet	88
FIG 50.	Cuaderno de Google Colab	91
FIG 51.	En la imagen se puede observar las versiones principales de Python con su respectiva publicación. (Marco, 2022).	94
FIG 52.	Se detalla la representación de las iteraciones con sus datos de prueba y entrenamiento. (Kucheryavskiy et al., 2020)	97
FIG 53.	FIG 53. Importación de Librerías.....	98
FIG 54.	FIG 54. Importación de los Datasets.	98
FIG 55.	Conjunto de Datos	98
FIG 56.	Datos del MIT Instituto de Tecnología de Massachusetts.....	99
FIG 57.	Datos del PTB (PHYSIKALISCH-TECHNISCHE BUNDESANSTALT).....	99
FIG 58.	Clasificación del tipo de latido del corazón.....	99
FIG 59.	Función que nos permite mostrar del conjunto de datos de entrenamiento del MIT con su respectiva clasificación.....	100

FIG 60. Importamos librerías y paquetes adicionales para su correcto funcionamiento.....	100
FIG 61. Preparación para el entrenamiento.	101
FIG 62. Diccionarios en python.....	101
FIG 63. Función para construir redes convulsionales el cual es multicapa.	102
FIG 64. Creación de un modelo basado en una serie de capas Conv1D con 2 bloques residuales que están conectados a otra serie de capas densas conectadas por completo.....	102
FIG 65. Procederemos con el entrenamiento definimos el número de épocas a considerar	103
FIG 66. Cálculo de una función para hallar la probabilidad más alta de detección.....	103
FIG 67. Procederemos con el entrenamiento definimos el número de épocas a considerar	103
FIG 68. Procederemos a importar algunas librerías y paquetes adicionales para la ejecución del algoritmo	104
FIG 69. Importamos los datos de arritmia MIT-BIH	105
FIG 70. Cantidad de datos	105
FIG 71. Muestra el número de registro de cada categoría	105
FIG 72. Transformar el dataset en conjunto de datos de variables x y variable y.....	106
FIG 73. Trazar el número de registros en cada categoría	106
FIG 74. Información estadística del dataset	107
FIG 75. Tomamos muestra de los datos	107
FIG 76. Se grafica los datos de ECG pertenecientes a cada etiqueta de clase.....	108
FIG 77. Visualizamos los datos porcentuales de subtipos de arritmia.....	108

FIG 78.	Nos muestra la cantidad de valores por cada clase	109
FIG 79.	proceso durante entrenamiento	109
FIG 80.	Se conforma los valores para luego establecer nuestra matriz de resultados.....	109
FIG 81.	FIG 81. Importación de librerías	110
FIG 82.	FIG 82. Conjunto de datos de entrenamiento	110
FIG 83.	FIG 83. Estos datos de entrenamiento son los que obtuvimos desde el data set previamente importado, los cuales contienen en la presentación 10 filas, las cuales van de 0 a las 9 y 188 columnas las cuales comprenden desde el 0 al 187.	111
FIG 84.	Con el método describe () obtenemos una información estadística de los datos de cada columna de la data set importado ...	111
FIG 85.	Conjunto de datos de prueba	111
FIG 86.	FIG 86. Como se puede observar estos son los datos obtenidos de la data set para ejecutar las pruebas respectivas en 10 filas, las cuales van de 0 a las 9 y 188 columnas las cuales comprenden desde el 0 al 187.....	112
FIG 87.	Con esta función Vamos a obtener un resumen conciso del marco de los datos, en este caso va a ser de la variable df, en otras palabras, una descripción general rápida del conjunto de datos.	112
FIG 88.	Conjunto de datos desequilibrados	112
FIG 89.	Observamos el gráfico previamente codificado, el cual nos muestra la distribución de los datos de entrenamiento entre 5 clases	113
FIG 90.	En este gráfico podemos observar las 5 clases definidas visualizadas en un gráfico en el cual podemos observar lo que son los ritmos normales, o latidos normales, latido prematuro supraventricular, contracción ventricular prematura, fusión de latido ventricular y normal, y los ritmos cardíacos inclasificables, todos tienden a una variación dependiendo de los datos obtenidos desde el Data set	114
FIG 91.	Preparación de los datos para la evaluación del modelo.....	114

FIG 92. Importamos la librería tensorflow que es la que nos permite construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones análogas del aprendizaje y el razonamiento del humano.....	115
FIG 93. Modelado de datos	115
FIG 94. Creación del bloque de identidad.....	116
FIG 95. Creación del bloque de convulsión.....	116
FIG 96. Bloque residual de identidad.....	117
FIG 97. Aquí podemos observar cómo se van realizando los cambios antes mencionados en cada capa sucesiva, la columna “Output Shape” nos va a mostrar el tamaño de los mapas de características en cada una de las capas	118
FIG 98. Importación de librería keras	118
FIG 99. Cálculo de una función para hallar la probabilidad más alta de detección.	119
FIG 100. Elaboramos nuestra matriz de confusión para presentar los resultados obtenidos	120
FIG 101. Gráfico estadístico de entrenamiento y validación	120
FIG 102. Matriz de resultados.....	120
FIG 103. Gráfico estadístico de entrenamiento y validación	121
FIG 104. Conjunto de datos del MIT	121
FIG 105. Procederemos con el entrenamiento definimos el número de épocas a considerar	122
FIG 106. Se reforma el marco de datos en una matriz Numpy y se convierte el resultado en una matriz categórica para cada clase.....	122
FIG 107. Cálculo de ponderaciones de cada clase para alimentar el modelo.....	122
FIG 108. Exactitud del algoritmo	123
FIG 109. Muestra de resultados	123

FIG 110. Épocas del algoritmo	124
FIG 111. Representación gráfica de las Accuracy y del puntaje de pérdidas versus el número de épocas	124
FIG 112. Nos permite pasar los lotes de prueba y emitir los puntajes para la validación realizada.	125
FIG 113. Nos permite pasar los lotes de prueba y emitir los puntajes de la prueba realizada	125
FIG 114. Validación y Prueba de la data set.	126
FIG 115. Validación y matriz de confusión de la data set.....	127
FIG 116. Prueba y matriz de confusión de la data set.	127
FIG 117. Resultados de la validación y de las pruebas	127
FIG 118. Resultados de las pruebas en cuanto a los porcentajes de precisión.....	127

AGRADECIMIENTOS

A nuestros colegas docentes y estudiantes de la Universidad de Guayaquil Jorge Medina, Alfonso Guijarro, Cristina Jácome, Jorge Medina, María Touriz, José Rodríguez, Erick Limón, Angie Salazar, Harold Frías, Cecibel Mite, Steven Aristega y Carlos León cuyos trabajos de investigación brindaron los insumos necesarios para cristalizar esta obra que sirve de base para afianzar criterios en áreas de la investigación aplicada entre el campo de la medicina y la informática.

PRÓLOGO

El trabajo que lleva como nombre Sistema Ubicuo para la detección y análisis de arritmias cardíacas, aplicando algoritmos de aprendizaje automático asumió el reto de optimizar los algoritmos de Machine Learning para ser utilizados como parte de los objetivos de este trabajo que pretende prevenir problemas de salud provocados por arritmias cardíacas.

Para las pruebas iniciales se trabajó con los dataset de la comunidad de Kaggle fue uno de los puntos de partida para llegar a cumplir las metas propuestas de un diagnóstico preliminar, eficaz y certero para tener un oportuno tratamiento. El libro puede ser tomado como un manual para la recolección y selección de datos con herramientas disponibles en internet y motivar al constante crecimiento en el área médica con el manejo de Inteligencia Artificial como apoyo a la prevención de problemas cardiacos y salud en general.

Este libro es una muestra del trabajo de investigación que realizan la Facultad de ciencias Matemáticas y Físicas y la Facultad de Ciencias Médicas que lleva la Universidad de Guayaquil por medio de proyectos de fondo concursable interno que mediante dispositivos inteligentes (dispositivos ubicuos), proporcionados por la facultad de medicina se recopilará datos para generar la información sobre arritmias cardíacas mediante un conjunto de datos con los que se puede aportar al entrenamiento de los modelos de Machine Learning (Aristega y León, 2022). Sin embargo, como se ha expresado antes, debido a el uso de Kaggle como herramienta para obtener los datos se nutrió de información relevante para favorecer la evolución de la investigación llevada a cabo sobre el problema cardíaco mencionado dentro de la Universidad de Guayaquil y cuyos avances se ven reflejados en un estilo de vida ayudado por los avances tecnológicos provocados por la convergencia entre la Inteligencia Artificial y la medicina.

En todo momento se contó con la asesoría de los médicos e ingenieros para representar los algoritmos según las especificaciones de las arritmias estudiadas y delimitadas.

INTRODUCCIÓN

La enfermedad cardiovascular (ECV) se han convertido en una de las causas de muerte más comunes en el mundo, especialmente para países en vía desarrollo o países de ingresos bajos y medianos que alcanza más del 75% de la estimación global cada año. En comparación con otros países que son más desarrollados. A nivel mundial conforme a los datos que brinda la Organización Mundial de la Salud (OMS) se considera que en el año 2015 murieron 17,7 millones de personas debido a enfermedades cardiovasculares que representa un 31% de todas las muertes registradas en el mundo, de estas muertes, 7,4 millones se debieron a cardiopatías coronarias (CHD), de los 17 millones de muertes de personas, la mayor parte pertenece a la población menor de 70 años que atribuyen a las enfermedades no transmisibles, un 82% corresponden a los países bajos y un 37% se deben a las Enfermedades Cardiovasculares (ECV) (OMS, 2017).

Debido a lo que sucede en el mundo, los médicos que elaboran en la Facultad de Medicina de la Universidad de Guayaquil decidieron contribuir a la sociedad o al mundo con pequeños estudios de investigación que ayuden a detectar problemas cardiovasculares, en los Estados Unidos las enfermedades del corazón no son consideradas como una de las causas principales de defunción en niños y adolescentes, pero si como la primera causa de muerte en adultos. Es conocido que existen variables que desempeñan un papel importante en las probabilidades de que una persona sufra de enfermedades del corazón. Estas variables son conocidas o llamadas factores de Riesgo (Texasheart, 2020).

La mayor parte de los factores de riesgos se pueden prevenir o controlar desde temprana edad es decir, desde la infancia, reduciendo así la probabilidad de sufrir enfermedades del corazón en la adultez. En marzo de 2016, la Organización Panamericana de la Salud (OPS), realizo un estudio en Ecuador perteneciente a la población considerada como riesgo de padecer enfermedades cardiovasculares, determinando que del 100%, el 30% de la población adulta entre 40 y 69 años corre el riesgo de sufrir de

alguna enfermedad que este asociado con el mal funcionamiento del sistema cardíaco. (Caycedo-Casas et al., 2017).

Lo que nos da a entender que las enfermedades cardiovasculares no discriminan la edad, todos somos susceptibles a sufrir alguna de ellas. Estas cifras poseen una tendencia al incremento al no llevar un estricto control y prevención de las cardiopatías en base a estos aspectos surge la idea de desarrollar este proyecto. Considerando que el diagnóstico oportuno de las afecciones cardíacas puede reducir de manera considerable la mortalidad de la población y en busca de mejorar los métodos y procedimientos. Según los datos tomados por el Instituto Nacional de Estadísticas y Censos del Ecuador (INEC) por sus siglas en inglés, se conoce que en el año 2014 se llegaron a reportar un total de 4.430 muertes por enfermedades isquémicas del corazón, por otro lado, los fallecimientos llegaron a 1.316 por insuficiencia cardíaca, las arritmias cardíacas sumaron un total de 168 muertes, mientras tanto que por paro cardíacos sumaron un total de 106 (INEC, 2018).

Lo que nos permite hacer una proposición que permita involucrar el Machine Learning para automatizar el diagnóstico preliminar de las enfermedades relacionadas al corazón, en el mes de septiembre del 2021, en los hospitales Luis Vernaza y de los Ceibos ubicados en la Ciudad de Guayaquil se realizaron charlas de prevención de enfermedades, debido a la incidencia del Covid-19, ya que el corazón es uno de los órganos vitales que es afectado gravemente por el virus.

Cada 29 de septiembre las organizaciones como lo son la Organización Panamericana de la Salud y la Organización Mundial de la Salud recuerdan el día del Corazón, para fomentar o recordar actividades sobre la prevención de patologías que puedan afectar al corazón. Entre las enfermedades cardiovasculares que padecen aquellas personas que han superado el Covid-19 son: Pancarditis, Taquicardias, Arritmias y enfermedades mucho más complejas como pericarditis, miocarditis, trombosis, etc (El Universo, 2021). A través de todo esto se busca concientizar a la población de tomar precauciones para evitar sufrir enfermedades como las mencionadas anteriormente.

Tabla 1. *Proyectos de optimización sobre enfermedades cardiovasculares*

Nombre del proyecto	Utilidad	¿Qué algoritmo se usó?	Año
Diagnóstico de enfermedades cardíacas con los algoritmos supervisados Naives Bayesian.	A través del uso de algoritmos poder determinar que pacientes están propensos a sufrir enfermedades cardíacas.	Algoritmos supervisados Naives Bayesian y SemiNaive Bayesian.	2019
Algoritmos genéticos aplicados a la optimización de características en la clasificación de arritmias cardíacas utilizando los clasificadores KNN y Naive Bayes	Asegurar y establecer que la clasificación de los electrocardiogramas sea exacta, con ayuda de un sistema.	Algoritmos genéticos y clasificadores KNN y Naive Bayes.	2017
Análisis de técnicas de ML en diagnóstico de enfermedades cardiovasculares.	Identificar una técnica o método que permita obtener mayores resultados de precisión al momento de identificar enfermedades cardiovasculares	Algoritmos supervisados Arboles de decisión, máquinas de soporte vectorial, Naive Bayes y redes neuronales.	S.f
Modelo basado en aprendizaje de máquina supervisado, para el análisis de datos cardiovasculares y su aplicación en el diagnóstico y pronóstico médico mediante el uso de una página web dinámica.	Mediante técnicas de aprendizaje automático identificar la presencia o ausencia de una enfermedad cardiovascular.	Algoritmos de aprendizaje automático.	2019

Nota: Se observa en la tabla los proyectos de optimización. Elaboración: Steven Aristega y Carlos León

Esta propuesta pretende seleccionar los algoritmos que ayudan a detectar las arritmias cardíacas de manera preliminar, de tal manera que los resultados tengan buena precisión y se reduzca el margen de error.



CAPÍTULO 1:

EL CORAZÓN: FUNCIONAMIENTOS, TRASTORNOS Y HERRAMIENTAS TECNOLÓGICAS PARA LA DETECCIÓN DE PROBLEMAS

1.1. Anatomía.

Localización

El corazón es un órgano que pertenece al sistema cardiovascular y se encuentra alojado en el tórax, protegido por detrás del esternón y al frente del esófago, la aorta que es la arteria más grande del cuerpo y la columna vertebral. Así mismo, reposa en el diafragma y el músculo que separa las cavidades torácicas y abdominales. Entre su principal función está la de bombear sangre para hacerla circular por los vasos sanguíneos (Luque Bernal, 2021)

Se encuentra dentro de una membrana o saco denominado pericardio, tiene dos hojas: una interna en ubicada en la superficie cardíaca y otra externa que está en los vasos que salen del corazón. Por ello, para mantener una correcta circulación de sangre, se cuenta con 3 vasos sanguíneos principales correspondientes a: Arterías, venas y capilares:

- Arterías: Distribuyen la sangre a todo el cuerpo.
- Venas: Conducen hacia el corazón la sangre proveniente de los diversos órganos.

Capilares: Permite el intercambio de sangre y los tejidos. El resultado se refleja en la transformación de la sangre arterial rica en oxígeno en sangre venosa cargada de dióxido de carbono. La unión entre las arterias y capilares, por una parte, entre los capilares y las venas, por otra se denominan zonas intermedias arteriola capilar y capilar venular.

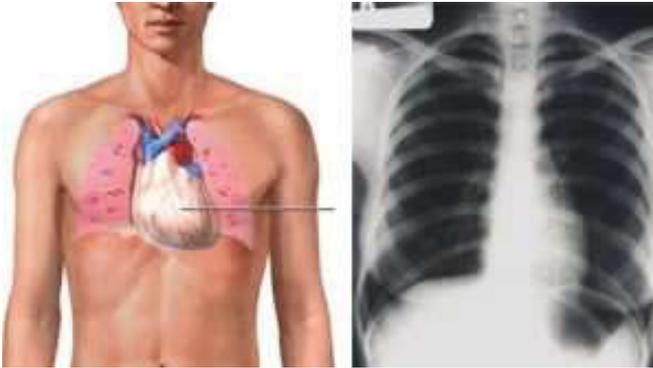


FIG 1. Información general sobre el corazón. (Cardiología Madrid, 2023).

Como se observa en la figura anterior, el corazón se apoya sobre el diafragma, acercándose en la línea media de la cavidad torácica encontrándose en el mediastino, una masa de tejido que se expande desde el esternón hasta la columna vertebral. Así mismo, cabe mencionar que casi dos tercios del del corazón se encuentra en la línea izquierda del cuerpo y es protegido por el pericardio al ser una membrana permite mantenerlo fijo en su posición sin dejar a un lado la libertad para las contracciones rápidas y vigorosas que bombean sangre al cuerpo (Gerard, 2018).

1.1.1. Esquema general de la circulación de la sangre

La sangre circula a partir del ventrículo izquierdo, su contracción impulsa la sangre arterial a la aorta y se reparte a todo el cuerpo. En los órganos y capilares se establecen intercambios fisicoquímicos que permiten la vida de los diferentes tejidos. El resultado del intercambio es transportado por los capilares y una vez recogido por las venas que la conducen a la aurícula derecha por intermedio de las venas cavas superior e inferior.

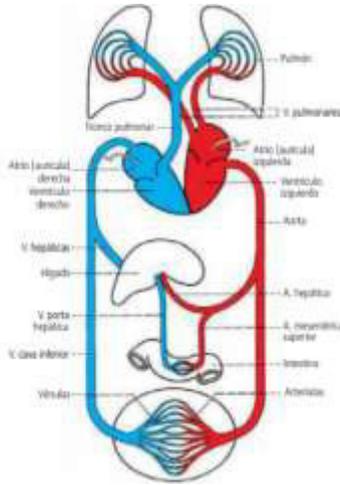


FIG 2. Esquema de circulación de la sangre en el corazón. (Latarget & Ruiz, 2019)

Tal y como se observa en la figura anterior, del atrio derecho la sangre pasa al ventrículo derecho que impulsa la sangre venosa al tronco pulmonar y de allí a los pulmones lo que permite a la sangre venosa eliminar al exterior el dióxido de carbono enriqueciéndose de oxígeno a su paso. Bajo dicho proceso, la sangre oxigenada regresa al corazón por las venas pulmonares que comienzas por el atrio izquierdo y pasa al ventrículo izquierdo para cerrar el circuito sanguíneo señalado en la figura anterior.

1.1.2. Esquema de circulación en las cavidades cardíacas

La circulación de la sangre en las cavidades cardiacas se distingue por la gran circulación que lleva la sangre oxigenada del corazón a todas las regiones del cuerpo y circulación pulmonar que moviliza entre el corazón y los pulmones:

- La gran circulación, general o sistémica: En dicha circulación desembocan los vasos linfáticos que son el conducto torácico a la izquierda y el conducto linfático derecho a la derecha.
- La pequeña circulación o circulación pulmonar: Las arterias contiene sangre carboxigenada y las venas, sangre oxigenada que es lo contrario de lo que sucede en la circulación sistémica.

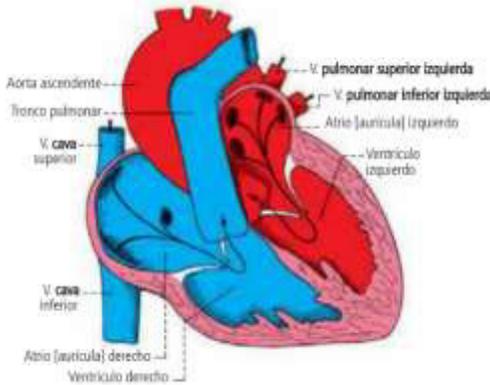


FIG 3. Esquema de la circulación intracardiaca. (Latarget & Ruiz, 2019)

Tal y como se observa en la figura anterior, las cuatro cavidades (dos aurículas y dos cavidades). Presenta un tabique entre dos aurículas y otro entre los ventrículos. Adicionalmente, se suma que las arterias llevan la sangre hacia fuera del corazón y las venas hacia adentro, por lo que, el flujo de sangre a través de los vasos y las cavidades es controlado por válvulas.

1.1.3. Las caras del corazón

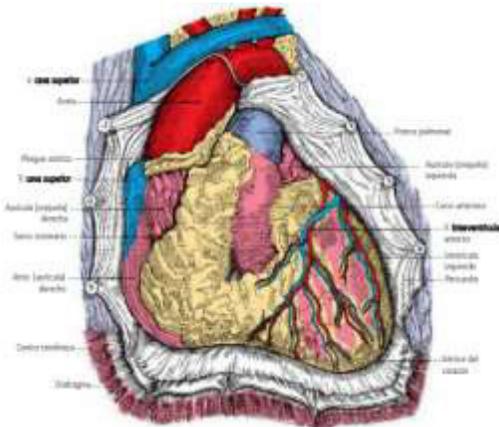


FIG 4. Vista anterior del corazón y de los grandes vasos. (Latarget & Ruiz, 2019)

Las caras del corazón son la anterior, inferior, pulmonar izquierda y derecha en donde:

Cara anterior

Un surco coronario separa los atrios de los ventrículos que limita dos sectores correspondientes a el sector atrial y el sector ventricular tal y como se observa en la figura anterior:

- Sector atrial: Se encuentra por encima de del sector ventricular del tronco pulmonar y de la arteria aorta. El conjunto de atrios y aurículas se aplica a dos grandes vasos cuya cara anterior está prácticamente oculta por la aurícula de los atrios. Por lo tanto, las aurículas son prolongaciones de los atrios y la ubicada en el lado derecho es cónica con base en el atrio derecho. Por otro lado, la ubicada en el lado izquierdo tiene bordes irregulares (Latarget & Ruiz, 2019).
- Sector ventricular: Se encuentra por debajo y a la izquierda el surco coronario. Desciende hasta el borde anteroinferior del corazón 1 o 2 cm por detrás del vértice. Dicho surco contiene la arteria interventricular anterior.

Cara inferior

Se aplica sobre el diafragma y tiene forma de triángulo, dividida por el surco coronario en las siguientes partes: una parte ventricular anterior, una parte atrial derecha que representa el quinto restante de la superficie de la cara. El segmento ventricular se divide en sentido longitudinal por el surco interventricular posterior que se origina algo a la derecha del vértice del corazón.

Cara pulmonar izquierda y derecha

La cara pulmonar derecha comprende una cara inferior y anterior que corresponde el ventrículo izquierdo y una parte superior que pertenece al atrio izquierdo. Por otro lado, la cara pulmonar derecha está formada por la porción convexa de la pared lateral del atrio que tiene relación con la cara medial del pulmón derecho. Los orificios de las dos venas cavas están reunidas en la cara lateral por el surco terminal (Latarget & Ruiz, 2019)

1.1.4. Morfología externa

Se visualiza la forma de un cono invertido con el vértice o ápex dirigido a la izquierda. En la estructura del corazón se encuentran los vasos sanguíneos que permiten la circulación de la sangre dentro que corresponden a las venas cavas superior e inferior junto con las venas pulmonares y la circulación fuera está encargado por los vasos de la arteria pulmonar y aorta. Así mismo, las venas cavas, encargadas de recoger la sangre venosa de todo el cuerpo, desembocan en la aurícula derecha, y las venas pulmonares, y permiten llevar sangre oxigenada desde los pulmones.

Tiene un borde derecho e izquierdo junto con una cara posterior y anterior. Así mismo, en la superficie tiene grasa donde se encuentran arterias y las venas que cubren al corazón correspondientes a las arterias coronarias que trasladan la sangre al musculo y las venas que sacan la sangre.

En cuanto a valores numéricos, en el caso de un hombre, el peso del corazón con respecto a todo el cuerpo seria en un 0.45% porcentaje que disminuye en las mujeres con un 0.40%. Así mismo, tomando en cuenta una estatura media se habla que oscila entre 250 – 350 gramos en el caso de los hombres y 200 – 300 gramos en las mujeres (Flórez & Augusto, 2022). Bajo dicho contexto, se tiene una variación del peso del corazón debido a factores como la edad, sexo, tamaño y peso de la persona.

1.1.5. Morfología interna

Internamente se observan cuatro cavidades correspondientes a lado izquierdo y derecho que constituyen lo que se denomina ventrículo izquierdo y ventrículo derecho que en un corazón con características normales no presentan comunicación entre dichos lados mencionados debido a la barrera puesta por un tabique muscular que lleva como nombre tabique interauricular, que separa ambas aurículas.

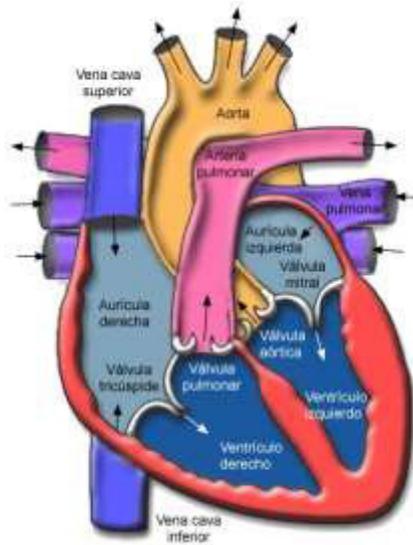


FIG 5. Morfología interna del corazón. (profesorenlinea, 2022)

Ventrículo derecho

El corazón derecho consta de una aurícula en la parte superior y un ventrículo en la inferior. A la aurícula derecha llega la sangre venosa (no oxigenada) de todo el cuerpo a través de las venas cavas, que desembocan en ella. Ambas se encuentran en la pared posterior, próximas al tabique: la superior, en la zona más alta, y la inferior, en la baja. También desemboca en la aurícula derecha el seno venoso, conducto que recoge la sangre venosa del corazón. En la cara anterior se ubica la orejuela derecha, de forma triangular.

La aurícula se comunica con el ventrículo derecho a través de una válvula, la tricúspide. Esta válvula permite el paso de sangre de la aurícula al ventrículo, pero no en sentido contrario. Cuando el corazón se contrae (sístole), la sangre sale del corazón a través de la válvula pulmonar, pasa a la arteria pulmonar y ésta la lleva a los pulmones para que se oxigene. Las válvulas tricúspide y pulmonar están separadas por una cresta muscular. El ventrículo derecho tiene forma triangular y su superficie muestra músculos, denominados papilares, que sobresalen de ella y sirven de anclaje para la válvula tricúspide.

Ventrículo izquierdo

En la parte superior del corazón izquierdo, como sucede en el derecho, se encuentra la aurícula izquierda, en la que desembocan cuatro venas pulmonares, responsables de llevar la sangre oxigenada desde los pulmones hasta el corazón. Muestra una orejuela larga y estrecha.

La aurícula se comunica con el ventrículo a través de una válvula, la mitral, que permite el paso de la sangre desde la primera hasta el segundo, pero no en sentido contrario. Cuando se produce la sístole, la sangre pasa del ventrículo a la arteria aorta a través de la válvula aórtica y es distribuida por todo el organismo. El ventrículo izquierdo es más largo y estrecho que el derecho, de tal forma que la punta del corazón está formada por ese ventrículo. Se observan dos grupos musculares papilares bien definidos: anterior y posterior, que sirven de anclaje a la válvula mitral.

Aurículas y ventrículos

Las aurículas tienen las paredes finas y están constituidas, de fuera hacia dentro, por el pericardio, la hoja interna o miocardio y una capa muy fina o endocardio. Esta última reviste toda la superficie interna del corazón, incluidas las válvulas, y está formada por una capa de células endoteliales, semejantes a las de los vasos sanguíneos, y fibras de colágeno y elásticas. La estructura de los ventrículos es semejante. La diferencia estriba en el grosor de la capa muscular. Mientras que el ventrículo derecho tiene un espesor de 3-4 mm, el izquierdo alcanza aproximadamente los 10 mm. Esta diferencia se debe a que, al expulsar la sangre durante la sístole, el ventrículo izquierdo se encuentra con una resistencia mayor: la presión arterial.

Las válvulas

Las válvulas situadas en los orificios que comunican las aurículas y los ventrículos, llamadas tricúspide y mitral, tienen una morfología diferente de las válvulas que se encuentran entre los ventrículos y las arterias pulmonar y aorta, es decir, las válvulas

pulmonar y aórtica. Todas tienen la misma función: se abren y dejan pasar la sangre, para después cerrarse e impedir que la sangre retroceda.

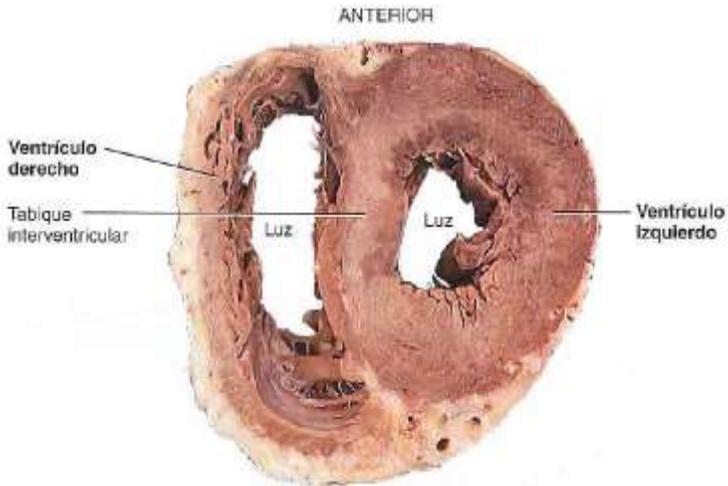


FIG 6. Corte transversal que muestra diferencias en el espesor. (Gerard, 2018)

Espesor miocárdico y función

El espesor de las cuatro cámaras varía según la función porque las aurículas de las paredes finas permiten entregar sangre a los ventrículos que, a su vez, ya presentan paredes gruesas debido a que bombean sangre a mayores distancias.

El ventrículo izquierdo es considerablemente más grueso que la del derecho debido a el trabajo que debe realizar para mantener la misma velocidad del flujo sanguíneo. Además, el espacio que lleva por nombre luz en la figura anterior, del ventrículo izquierdo es más o menos circular mientras que la del ventrículo derecho es semilunar.

Así mismo, las paredes cardiacas también contienen tejido conectivo denso que forma el esqueleto fibroso del corazón y dicha estructura está formado por cuatro anillos de tejidos conectivos densos que rodean las válvulas cardíacas. Al mismo tiempo que forman la base estructural de las válvulas cardiacas el esqueleto

fibroso también evita la sobredistensión de las válvulas al pasar la sangre a través de ellas.

1.2. Fisiología. Funcionamiento del Corazón

El corazón funciona como una bomba, sus latidos están entre 60 y 100 veces por minuto, en cada latido el corazón despacha sangre a todo el cuerpo, transportando oxígeno a cada una de sus células. Después de haber distribuido todo el oxígeno a las células, la sangre regresa al corazón. Desde allí la sangre bombea hacia los pulmones. Donde se vuelve a cargar el oxígeno (Larissa Hirsch, 2018).

En el trabajo publicado acerca de la Fisiología Cardíaca de (García, 2022) señala que: El corazón consta de dos aurículas y dos ventrículos a través de los cuales la sangre entra al corazón y sale de los ventrículos. El corazón y los vasos sanguíneos transportan sangre a todas las células del cuerpo para que puedan recibir oxígeno, nutrientes y otras sustancias esenciales. Forman un perfecto sistema de irrigación donde la sangre es rica en oxígeno y recoge la sangre deficiente y llena de impurezas.

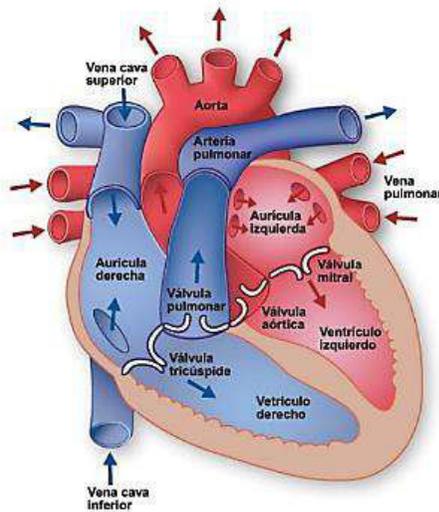


FIG 7. *Funcionamiento del Corazón.*

El corazón suele dar entre 60 y 100 latidos por minuto, pero esto suele variar acorde a la salud, sexo, edad, entre otros factores que pueda tener una persona. Según (Vallejo Zambrano et al., 2019) menciona en su artículo que cuando los impulsos eléctricos llegan a los músculos de los ventrículos, estos se contraen, de esta manera el corazón bombea sangre hacia distintas partes del cuerpo y venas, llevando consigo oxígeno entre otros nutrientes que brinda la sangre.

La sangre que el corazón envía al cuerpo se produce a través de impulsos eléctricos internos (Ramos et al., 2018) indica que el bombeo de sangre permite que la aurícula derecha reciba la sangre que ha pasado por el cuerpo, con dióxido de carbono, de allí al ventrículo derecho, que a su vez la lleva a la arteria pulmonar. También explica que la aurícula izquierda recibe sangre oxigenada de los pulmones y la envía al ventrículo izquierdo, que la envía a la aorta, que la redistribuye por todo el cuerpo (Ramos et al., 2018). El sistema eléctrico del corazón tiene un sistema un poco complejo, expertos en el área denominan al área que se encarga de distribuir la energía eléctrica del corazón como nodo sinusal.

Las arritmias cardiacas son ocasionadas por latidos irregulares del corazón estas suelen generar la sensación de aceleración o aleteo del corazón, en ciertos casos suelen ser inofensivas, pero en otros puede significar alguna enfermedad cardiaca. Aunque es normal que una persona posea un ritmo cardíaco lento o rápido, en algunos casos, por ejemplo, el ritmo cardíaco aumenta durante el ejercicio o disminuye durante el (Aristega & León, 2022).

1.2.1. Funcionamiento eléctrico

El ritmo cardiaco es provocado por la contracción de las cuatro cavidades del músculo cardiaco. Dichas cavidades, como se mencionó anteriormente, corresponden a aurícula derecha, ventrículo derecho, aurícula y ventrículo izquierdos. Bajo dicho contexto, el músculo cardiaco tiene un sistema que crea y envía señales eléctricas para estimular las cuatro cavidades.

Dicho proceso comienza con una señal en la cavidad superior derecha del corazón que comunica a las demás cavidades superiores que se contraigan haciendo que la sangre se traslade a las dos cavidades inferiores para terminar con las señales indicando a los ventrículos que se contraigan para bombear sangre a los pulmones, cerebro y al corazón.

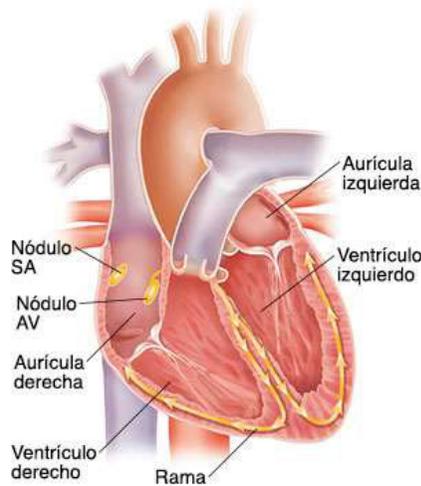


FIG 8. El sistema eléctrico del corazón (myhealth, 2021)

Hay grupos de células especializadas de enviar señales eléctricas del corazón que se llaman nodos recorriendo caminos en los ventrículos que tienen como nombre ramas del haz de His y por lo general, el nódulo SA marca el ritmo de los latidos y que oscilan (en reposo) entre 60 y 100 pulsaciones por minuto. Una vez finalizada la contracción, las aurículas se relajan para llenarse de sangre.

1.2.2. El flujo de sangre a través del corazón

El corazón bombea sangre a todo el cuerpo junto con oxígeno y nutrientes eliminando en el camino a el dióxido de carbono y elementos residuales. A medida que a la sangre recorre el cuerpo, se consume el oxígeno de la sangre y el proceso se detalla a continuación:

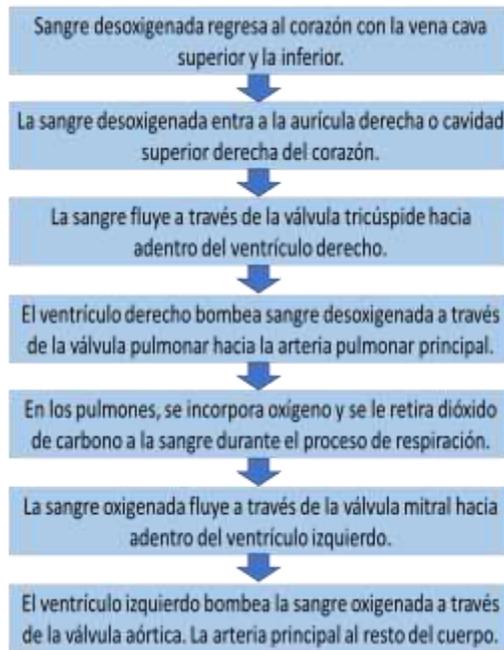


FIG 9. *El flujo de sangre a través del corazón. (National Heart Lung and Blood Institute, 2022)*

Tal y como se observa en la FIG 3, el flujo de la sangre a través de corazón comienza con la aurícula derecha hasta el ventrículo derecho para luego transportar la sangre con bajo contenido de oxígeno por la arteria pulmonar que mediante los pulmones se agrega oxígeno a la sangre.

1.3. Trastornos cardiovasculares

La enfermedad cardiaca hace referencia a cualquier infección que afecte al corazón. Existen muchos tipos de los cuales algunos se pueden prevenir. A diferencia de las enfermedades cardiovasculares, que afecta a todo el sistema. La enfermedad cardiaca solo afecta al corazón. Según el Centro para el Control y prevención de enfermedades, la enfermedad cardiaca es él es principal motivo de defunción en los Estados Unidos. Cerca de 1 de cada 4 muertes en los Estados Unidos sucede debido a una enfermedad cardiaca y afecta a todos los géneros en general (Felman, 2019).

Las malas decisiones de los médicos no permiten un tratamiento eficaz de los pacientes; la falta de un diagnóstico anticipado aumenta la posibilidad de enfermedades cardiovasculares, ya sean estas cerebrales o cardiológicas, entre otras. Estas son las principales causas y consecuencias del problema, que se muestran a continuación en la siguiente tabla:

Tabla 2. *Matriz de causas y consecuencias del problema.*

Causas	Consecuencias
Problemas cardíacos.	Mal funcionamiento de válvulas e insuficiencia cardíaca
Persona adulta mayor	Mayor probabilidad de mortalidad
La ausencia de información de protocolo a seguir	Negligencia en la atención al paciente.
Inexistencia de un prototipo que permita controlar la arritmia cardíaca en un tratamiento cardiológico	Mala gestión del proceso para controlar debidamente al paciente.
Déficit de diagnósticos anticipados	Incrementa la probabilidad de enfermedades cardiovasculares.
Perdida repentina de la función cardíaca.	Provoca un paro cardíaco.
Variación de los impulsos eléctricos del corazón	Arritmia Cardíaca
Carencia de aptitud preventiva ante enfermedades del corazón	Ocasiona un aumento de personas que sufran de enfermedades del corazón

Nota: Se observa las causas y consecuencias del tema abordado.

1.3.1. Arritmias cardíacas

Las arritmias cardíacas pueden ocasionar una sensación de aleteo o de corazón acelerado, pueden llegar a ser inofensivas para el ser humano. Sin embargo, existen arritmias cardíacas que pueden originar signos y síntomas muy molestos, y que muchas veces pueden llegar a poner en riesgo la vida. Aunque en ciertas situaciones es muy normal que una persona tenga el ritmo cardíaco lento o acelerado por ejemplo la frecuencia cardíaca puede aumentar con el ejercicio o disminuir durante el sueño (Mayoclinic, 2021).

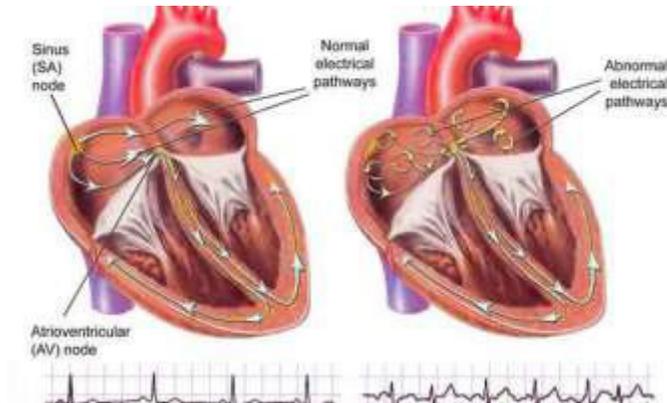


FIG 10. Arritmia cardíaca. Foto Tomada de (Castelloninformacion, 2020)

Causas

La causa más usual de una arritmia es un trastorno cardíaco, en simples palabras una arritmia es provocada por la coronariopatía, las valvulopatías y la insuficiencia cardíaca. Algunos fármacos ya sean con o sin receta médica, incluyendo los que son especiales para tratar dolencias cardíacas también pueden ocasionar arritmia. Otra causa son las anomalías anatómicas, es decir, son defectos congénitos (Mitchell, 2019). También puede darse por un fallo en el sistema eléctrico del ritmo cardíaco, llamado sistema de excitación y conducción. Este sistema puede fallar por alguno de estos 3 motivos que se mencionan a continuación:

1. Uno de los mecanismos eléctricos deja de funcionar por la falta de generación de impulso eléctrico.
2. El impulso eléctrico se crea en el sitio equivocado.
3. Se origina un cortocircuito en el sistema eléctrico.

Síntomas de la Arritmia.

Los síntomas dependerán del tipo y de la gravedad de la arritmia. Pueden ocurrir todos los días o con poca frecuencia. Con algunos tipos de arritmia pueden incluso no presentar ningún síntoma. Generalmente las palpitaciones y síncope son los principales síntomas de la arritmia.

Los síntomas de arritmia incluyen:

- Palpitaciones o golpes en el pecho. Mareos.
- Desmayos.
- Cansancio en el cuerpo.
- Problemas para respirar.

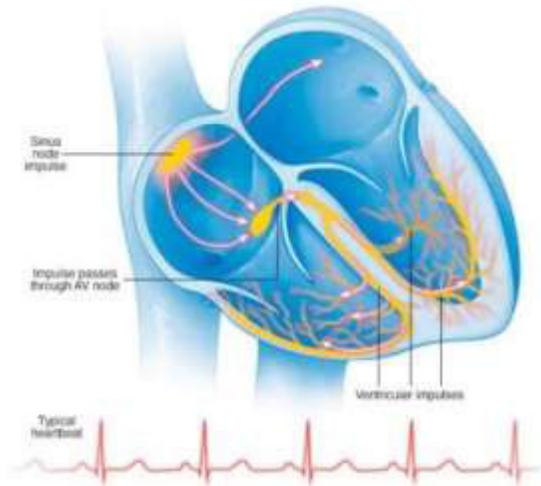


FIG 11. Ritmo cardíaco normal.

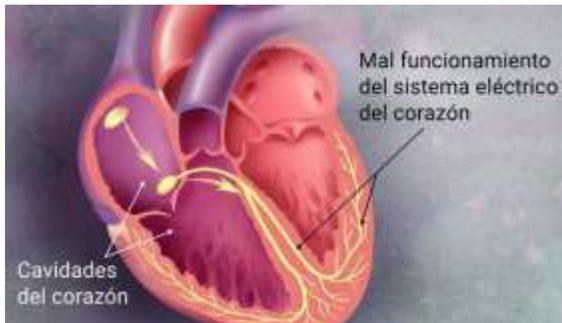


FIG 12. Latidos anormales del corazón

Tipos de arritmias cardiacas

Tabla 3. *Tipos de arritmias cardiacas*

Arritmia	Causas	Síntomas
<p>Bradiarritmias Se definen por la presencia de una frecuencia cardiaca (FC) < 60 latidos por minuto (lpm) motivada por distintas causas según el subtipo, pudiendo asociarse a alteraciones del tono vegetativo o electrolíticas, si bien las que aparecen en pacientes con un infarto de miocardio son debidas a la isquemia y necrosis irreversible del sistema de conducción.</p>	<p>Edad, infartos, infecciones, hipertensión, consumir drogas como el tabaco, el alcohol, o fármacos que pueden desencadenar alteraciones del impulso eléctrico del corazón, otro factor puede ser realizar ejercicio físico de manera constante e intensa.</p>	<p>Mareos, confusión, fatiga, somnolencia, incapacidad mental, disnea, insuficiencia cardiaca congestiva, angina de pecho, palpitaciones, presíncope o síncope (pérdida de conocimiento).</p>
<p>Bradicardia sinusal (BS) Se trata de una observación recurrente, comúnmente transitoria, que se asocia a un incremento del tono vagal. En personas aparentemente sanas y ancianos, se observan en ocasiones frecuencias de < 50 lpm, que tienen la posibilidad de inclusive descargar de los 30 lpm en deportistas y adultos adolescentes sanos en reposo o por la noche (cuando se destaca el tono vagal).</p>	<p>Es muy frecuente en personas que llevan a cabo mucho ejercicio físico o muy intenso, debido a la activación del tono vagal.</p>	<p>Vómitos, tos, defecación o síncope vasovagal. Diversas enfermedades (acidosis, apnea del sueño, brucelosis, hipotiroidismo, hipertensión intracraneal, etc.)</p>
<p>Bloqueos de la conducción auriculoventricular En este tipo de arritmia el impulso eléctrico auricular tiene un retraso en su conducción o no se conduce a los ventrículos, por un bloqueo de la conducción de los impulsos que puede producirse en el nódulo AV (bloqueo suprahisiano) y en el haz de His o en sus ramas (bloqueo infrahisiano).</p>	<p>Medicamentos (p. ej., betabloqueantes, bloqueantes de los canales de calcio, digoxina, amiodarona) Aumento del tono vagal Valvulopatía Cardiopatías congénitas, trastornos genéticos u otros.</p>	<p>Mareos, presíncope, síncope, angina e insuficiencia cardiaca.</p>
<p>Bradiarritmias asociadas a infarto agudo de miocardio. La BS es una de las arritmias más frecuentes en pacientes que han sufrido un infarto, sobre todo si existe una lesión de la arteria coronaria derecha.</p>	<p>En estos pacientes pueden aparecer alteraciones de la conducción auriculoventricular e intraventriculares (transitorias en un 18% de los pacientes y persistentes en un 5%); especialmente estas últimas se asocian a un</p>	<p>Mareos, lipotimia, sensación de mareo/desmayo inminente (presíncope) o síncope.</p>

Arritmia	Causas	Síntomas
	riesgo elevado de mortalidad a corto plazo	
<p>Anomalías de la conducción intraventricular</p> <p>Los procesos isquémicos o degenerativos pueden deprimir o interrumpir la conducción a nivel de las ramas del haz de His, produciendo un retraso de la despolarización ventricular que se traduce en una prolongación del complejo QRS del ECG (hasta > 120 ms); si el bloqueo es incompleto, el complejo QRS varía entre 100-120 ms (en condiciones normales tiene una duración de 0,08-0,12 s). También se alterará la repolarización ventricular, que se suele traducir como una desviación de la onda T en dirección opuesta a la desviación principal del complejo QRS (Figura 3C).</p> <p>Este tipo de anomalías se conocen clásicamente como bloqueos de rama y engloban los bloqueos de la rama derecha (BRD) o de la rama izquierda (BRI) del haz de His, el bloqueo fascicular o sus combinaciones; cada tipo presenta características fisiopatológicas diferenciales.</p>	<p>Son frecuentes en ancianos, incluso en individuos sanos (en especial el RBD), y en pacientes con miocardiopatía dilatada o cardiopatía isquémica tras la cirugía de sustitución de la válvula aórtica o un implante percutáneo de la misma.</p>	<p>Mareos, lipotimia, sensación de mareo/desmayo inminente (presíncope) o síncope.</p>
<p>Taquiarritmias</p> <p>Su importancia clínica viene definida por su capacidad para producir un fallo de la función de bomba cardiaca, resultando en la potencial aparición de una fibrilación ventricular que puede producir la muerte súbita del paciente.</p>	<p>Por encima de cierta FC, la reducción del periodo de llenado de sangre de los ventrículos durante la diástole conduce a una disminución del volumen-minuto cardiaco.</p>	<p>Fatiga, astenia, disnea, disminución de la tolerancia al ejercicio, oliguria, etc..</p>
<p>Taquicardias supraventriculares</p> <p>Se trata de ritmos rápidos taquicardias generadas en aurículas o nódulo AV (por encima de la bifurcación del haz de His) que se presentan en el ECG como un QRS estrecho (<120 ms) e intervalos RR regulares.</p>	<p>La dilatación, hipertrofia o fibrosis cardiaca, los estados inflamatorios crónicos, algunos fármacos (simpaticomiméticos, metilxantinas) y el hipertiroidismo.</p>	<p>Palpitaciones, ansiedad, mareo, sudoración, disnea, opresión precordial, hipotensión arterial, presíncope y, en ocasiones, síncope.</p>
<p>Taquicardias auriculares</p> <p>Pueden ubicarse en cualquier parte de la aurícula y se suelen reconocer</p>	<p>Entre ellas se incluyen, por ejemplo, las extrasístoles, un latido</p>	<p>Aunque pueden producir palpitaciones y ser el desencadenante</p>

Arritmia	Causas	Síntomas
<p>en el ECG por la aparición de una onda P prematura habitualmente seguida de un complejo QRS con un intervalo PR \geq 120 ms.</p>	<p>que va seguido de una pausa posterior compensatoria, y aparecen hasta en el 60% de los pacientes adultos sanos que ingieren café, tabaco, alcohol u otros estimulantes, y en ancianos.</p>	<p>de otras taquiarritmias supraventriculares, por lo general son asintomáticas y no suelen requerir tratamiento.</p>
<p>Taquicardias de la unión auriculoventricular</p> <p>Se generan una vez que el nódulo AV forma parte del circuito de reentrada, y se caracterizan por la existencia de un ritmo veloz (160-240 lpm) y un QRS estrecho. Se distingue entre:</p> <p>a) taquicardias por reentrada intranodal: una vez que hay 2 vías de conducción diferentes por medio del nódulo AV.</p> <p>b) taquicardias ectópicas de la alianza auriculoventricular: son debidas a un incremento del automatismo de un foco localizado en el nódulo AV que estimula los ventrículos, y cuya morfología del QRS es parecido a la de los latidos sinusales.</p> <p>c) taquicardias relacionadas a una vía accesoria auriculoventricular: son debidas a la reentrada del fomento cardiaco por medio de una vía accesoria de conexión entre aurículas y ventrículos, por lo cual se despolarizan los ventrículos más precozmente de lo esperable.</p>		
<p>Taquicardias ventriculares</p> <p>Incluirían a todas las que se originan por debajo de la bifurcación del haz de His y en cuyo mantenimiento solo participa el miocardio ventricular. Dado que los ventrículos se contraen de forma independiente de las aurículas, no se aprecia una relación constante entre la onda P y el complejo QRS, siendo estos complejos anchos e irregulares (los impulsos no usan el sistema de conducción habitual).</p>		
	<p>Aparecen asociadas a un aumento del tono simpático, infarto de miocardio, fiebre reumática, intoxicación digitalica o a cirugía valvular.</p>	<p>Generalmente no producen síntomas, pero en ocasiones son percibidas como una pausa en el latido cardiaco seguida de un latido más fuerte.</p>
	<p>Son más frecuentes en pacientes con Cardiopatía isquémica o cardiopatía estructural.</p>	<p>Se puede manifestar a través de síntomas como palpitaciones, mareo, dolor torácico.</p>

Nota: Tipos de arritmias cardiacas.

Tal y como se observa en la tabla anterior relacionado a los tipos de arritmias cardiacas. En primer lugar, encontramos los tipos de Bradicardia y en segundo lugar tenemos los tipos de Taquiarritmias, incluidas las definiciones, las causas y los síntomas que producen. La información para la realización de este cuadro fue tomada del trabajo de De Souza (2021).

¿Cómo se diagnostica la arritmia cardíaca?

Para el correcto diagnóstico de la arritmia cardíaca, el cardiólogo debe revisar y analizar los síntomas que el paciente presenta, antecedentes médicos y realizarle un examen físico. Incluso puede efectuar unas pruebas de monitoreo cardíaco que son esenciales para detectar las arritmias. Estas pueden ser:

- Electrocardiograma.
- Monitor Holter.
- Monitor de Eventos.
- Registrador de bucle implantable.

1.3.2. Enfermedad de las arterias coronarias

La enfermedad cardíaca hace referencia a cualquier infección que afecte al corazón. Existen muchos tipos de los cuales algunos se pueden prevenir. A diferencia de las enfermedades cardiovasculares, que afecta a todo el sistema. La enfermedad cardíaca solo afecta al corazón. Según el Centro para el Control y prevención de enfermedades, la enfermedad cardíaca es el principal motivo de defunción en los Estados Unidos. Cerca de 1 de cada 4 muertes en los Estados Unidos sucede debido a una enfermedad cardíaca y afecta a todos los géneros en general (Felman, 2019).

Defectos cardíacos congénitos: Una persona que padezca de esta enfermedad nace con un problema en el corazón. Los diferentes tipos de defectos cardíacos congénitos que existen son:

Válvulas cardíacas anormales.

Es muy probable que las válvulas no se abran de la manera correcta o que goteen sangre.

Defectos septales.

Son unos agujeros que se encuentran en la pared entre las cámaras inferiores o superiores del corazón.

Atresia.

No se encuentra ni una de las válvulas cardíacas.

El padecimiento cardíaco congénito puede implicar algunos problemas estructurales importantes, como la ausencia de un ventrículo además de traer problemas con las arterias principales que salen del corazón.

Arritmia.

Se sabe que una arritmia o también conocida fibrilación auricular es una enfermedad cardíaca que se origina cuando el pulso o el ritmo del corazón cambian.

1.3.3. Defectos cardíacos congénitos

Una persona que padezca de esta enfermedad nace con un problema en el corazón. Los diferentes tipos de defectos cardíacos congénitos que existen son:

- Válvulas cardíacas anormales. Es muy probable que las válvulas no se abran de la manera correcta o que goteen sangre.
- Defectos septales. Son unos agujeros que se encuentran en la pared entre las cámaras inferiores o superiores del corazón.
- Atresia. No se encuentra ni una de las válvulas cardíacas. El padecimiento cardíaco congénito puede implicar algunos problemas

1.3.4. Presión arterial alta (hipertensión)

La presión arterial alta es una afección que debido a diversos factores como la edad, raza, antecedentes familiares o consumo de alcohol y tabaco pueden afectar al paciente y se identifica como la fuerza que ejerce la sangre contra las paredes de las arterias con el

transcurso del tiempo y es lo suficientemente alta para causar un problema de salud cardíaco.

La presión arterial depende de la cantidad de sangre que bombea el corazón y lo estrecho de las arterias. Se mide en milímetros de mercurio (mm Hg) teniendo presión sistólica (cuando el corazón late) y presión diastólica (presión de arterias entre latidos) (Mayo Clinic, 2022). Dado que, se puede tener dicha afección durante varios años sin presentar síntoma, se recomienda consultar al médico cada 2 años para evitar problemas como ataques cardíacos y accidentes cerebrovasculares.

1.3.5. Cardiopatía congénita

Una persona que padezca de esta enfermedad nace con un problema en el corazón. Los diferentes tipos de defectos cardíacos congénitos que existen son:

Válvulas cardíacas anormales: Es muy probable que las válvulas no se abran de la manera correcta o que goteen sangre.

Defectos septales: Son unos agujeros que se encuentran en la pared entre las cámaras inferiores o superiores del corazón.

Atresia:

- No se encuentra ni una de las válvulas cardíacas.
- El padecimiento cardíaco congénito puede implicar algunos problemas estructurales importantes, como la ausencia de un ventrículo además de traer problemas con las arterias principales que salen del corazón.

1.3.6. Arteriopatía periférica

Son enfermedades de los vasos sanguíneos que irrigan las extremidades superiores e inferiores. Es una afección de los vasos sanguíneos que irrigan las piernas y los pies. Esta se presenta a causa del estrechamiento de las arterias en las piernas. Ocasiona una disminución del flujo de sangre, provoca que surjan lesiones en nervios y otros tejidos.

Cuando se provoca la enfermedad, mayormente será por la aterosclerosis. El problema ocurre cuando el material graso conocido como placa se acumula sobre y dentro de las paredes de las arterias y las vuelve más estrechas. Así mismo, tomando en cuenta las paredes de las arterias también se vuelven más rígidas y no pueden ensancharse que provoca dilatación para permitir el abundante y normal flujo de sangre que se necesita.

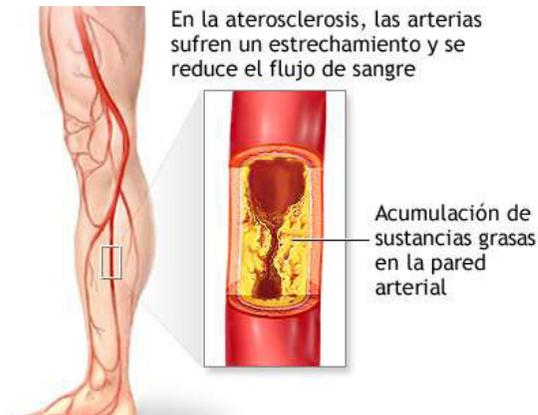


FIG 13. Arteriopatía periférica en las piernas. (Ortega et al., 2019)

Tal y como se observa en la figura anterior, como resultado de la arteriopatía, cuando los músculos de las piernas están trabajando más intensamente que puede ser como ejemplo correr o caminar no pueden recibir suficiente oxígeno y sangre. Si esta enfermedad aumenta en su gravedad, es posible que no quede suficiente sangre y oxígeno, incluso los músculos cuando está en reposo.

La arteriopatía periférica es un trastorno común. Casi siempre afecta a los hombres mayores de 50 años, pero las mujeres también pueden sufrirla. Las personas están en mayor riesgo si tienen antecedentes de:

- Colesterol anormal.
- Diabetes.
- Cardiopatía (arteriopatía coronaria).
- Presión arterial alta (hipertensión).
- Enfermedad renal que involucra hemodiálisis.

- Tabaquismo.
- Accidente cerebrovascular (enfermedad cerebrovascular).

1.3.7. Cardiopatía coronaria

También conocida como enfermedad coronaria, se desenvuelve cuando las arterias que suministran la sangre al corazón se tapan con placa. Esto provoca que se endurezcan y estrechen. Dando como resultado, que el suministro de sangre se reduzca y el corazón reciba menos oxígeno y por consiguiente recibe menos cantidad de nutrientes (Felman, 2019).

1.4. Herramientas tecnológicas para la detección de problemas

Para el correcto diagnóstico de la arritmia cardiaca, el cardiólogo debe revisar y analizar los síntomas que el paciente presenta, antecedentes médicos y realizarle un examen físico. Incluso puede efectuar unas pruebas de monitoreo cardíaco que son esenciales para detectar las arritmias.

1.4.1. Electrocardiograma

El electrocardiograma (ECG) representa gráficamente la actividad eléctrica del corazón, mediante el uso de electrodos, es decir, almohadillas conductoras, en la superficie del cuerpo para medir los potenciales cardíacos del corazón. Un cable electrocardiográfico es un electrodo de registro o un par de electrodos de registro en una ubicación específica (Christov et al., 2006).

El (ECG) es una representación gráfica de la actividad generada por el corazón, por medio del uso de electrodos colocados en regiones concretas corporal, capta los impulsos eléctricos causados por procesos de despolarización o contracción y repolarización o relajación (Llodrá, 2018). El electrocardiograma lo consideran una de las técnicas de detección de ritmos cardiacos, para poder realizar un diagnóstico sobre enfermedades cardiovasculares como arritmias, latidos anormales, etc.

Una definición del electrocardiograma dada por (Rivera et al., 2020) define que es un gráfico en el cual se registra las variaciones de voltaje relacionadas con la época y en el cual se la corriente eléctrica del corazón a lo largo de una época definida. La detección de enfermedades cardiovasculares a través de los electrocardiogramas es una de las maneras más rápidas, económicas y fiable de captar y leer los latidos del corazón.

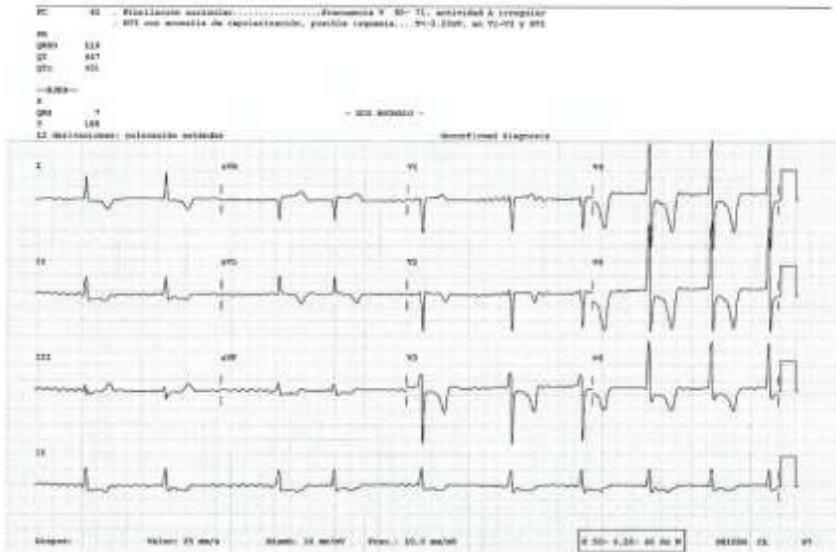


FIG 14. Señal de Electrocardiograma. (Pozas, 2017)

Según (Yanez et al., 2021), nos sugiere que para obtener un ECG y de muchas señales médicas es de importante trascendencia la implementación del electrodo, el cual es un material conductor que al entrar en contacto con la dermis capta los potenciales eléctricos, que con ello producen la señal resultante de electrocardiografía, empiezan en la región preeminente por medio de vías especializadas hasta los ventrículos, llevando la señal. (Olvera et al., 2020). Estas ondas se registran en un papel de medición. en donde se hace una alusión en que del lado horizontal se mide la era o frecuencia cardiaca, en esta línea hay 5 cuadrados de 1 milímetros cada uno, midiendo la amplitud de corriente eléctrica captadas en voltios.

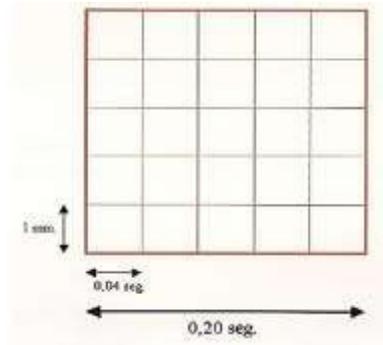


FIG 15. *Papel del Electrocardiograma*

Dentro de los cuadros que respecta al electrocardiograma, el posicionamiento de las ondas se representa de forma gráfica en el electrocardiograma y en forma numérica en un contenedor rotulado con la letra de la onda, donde se muestra el segundo en el cual ocurrió (Rivera et al., 2020). En otras palabras, de lo citado anteriormente, el electrocardiograma muestra de manera grafica a través de una onda senoidal, el pulso del corazón, mediante unos cuadros de 5mm formado de cuadros de 1mm respectivamente.

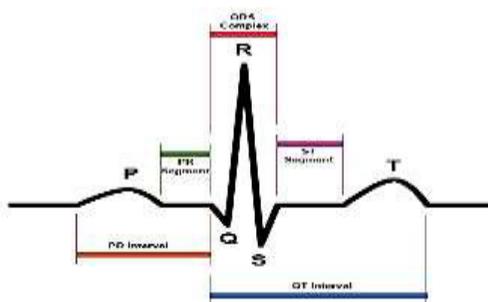


FIG 16. *Procesamiento de Señales*

Tabla 4. *Valores de Tiempo y Voltaje de las ondas, intervalos, segmentos complejos electrocardiográficos.*

	Valores de Tiempo	Valores de Voltaje
Onda P	<0.10 seg.	<0.25mV
Intervalo PR	0.12-0.20 seg.	

	Valores de Tiempo	Valores de Voltaje
Onda Q	<0.4 seg.	>1/3 QRS
Complejo QRS	0.6-0.8 seg.	
Segmento ST		
Onda T	0.10-0.25 seg.	2-6 mm
Intervalo QT	0.32-0.40 seg.	
Onda U	0.16-0.24 seg.	2 mm

Nota: Valores de tiempo en voltajes. Fuente y elaboración propia.

En la siguiente tabla podemos observar los diferentes segmentos en valores los cuales son las líneas que une el final de cada onda P con el inicio del QRS y los intervalos que son los que comienzan de la onda P hasta en inicio del QRS, incluyendo a la onda P, según tiempo y voltaje en el que el electrocardiógrafo. La siguiente tabla fue tomada del Autor Zavala (2018)

El electrocardiograma está representado por fenómenos eléctricos internos, registrados en un ECG, los cuales permanecen representados por todas sus ondas, intervalos y segmentos (Salatino, 2021).

ONDAS

Onda P Representa una activación auricular inferior a 0,12 segundos y un voltaje inferior a 0,25 mV. Siempre DI, DII y aVF positivos, normalmente aVR negativos, con cualquier polaridad DIII y aVL (Zavala, 2018). En definitiva, la onda p es una de las primeras ondas en ser escritas en un electromiograma.

La Onda Q, es toda onda negativa y está ubicada al inicio del complejo QRS (Zavala, 2018). Esta procede de una onda R, producida por una despolarización ventricular de una deflexión negativa.

La Onda R es aquella la cual representa la recuperación de los pulsos eléctricos en el área de las aurículas. Esta onda la podemos observar que se encuentra representada en un complejo QRS,

durante la deflexión positiva (Zavala, 2018). Mientras se realiza la despolarización ventricular, podemos observar que es la primera deflexión que podemos encontrar en el ECG.

La Onda S, siendo una onda negativa seguida de una onda R (Zavala, 2018). Esta se produce durante la despolarización ventricular, durante una primera deflexión positiva.

La Onda T está representada por la repolarización de los ventrículos en la onda del ciclo cardiaco, debido a esta causa una fuerza del lado opuesto a la despolarización, debido a la presión desempeñada por la sangre se repolarizan rápidamente, esto ocurre en la sístole mecánica ventricular (Zavala, 2018). La Onda T está representada por la repolarización ventricular, en una deflexión lenta.

La Onda U, esta onda se la reconoce por el bajo voltaje que encontramos después de la onda T y está ubicada antes de la onda P, esta suele ser inconsistente, siempre y cuando sea de menor voltaje y tenga la misma polaridad que la onda T es considerada normal (Zavala, 2018). Esta la podemos ver entre la Onda T y P y está representada por una deflexión positiva.

Intervalos

El Intervalo PR es aquella que se puede observarla al inicio de la Onda P hasta que empieza el complejo QRS, cuando este se encuentra acortado se deduce que hay una aceleración de la conducción atrio ventricular. Este intervalo se lo puede visualizar exista o no la Onda Q, su tamaño es al inicio de la onda P hasta el comienzo de la Onda Q. En el tiempo de conducción AV., su valor normal va de 0.12 a 0.20 segundos o 120 a 200 milisegundos (Zavala, 2018). La distancia que comprende este intervalo está entre el inicio de la onda P y el inicio del complejo QRS.

El intervalo QT siendo aquella que está representada por la sístole eléctrica ventricular, se visualiza al inicio del complejo QRS hasta el final de la Onda T. (Zavala, 2018). Este intervalo se precede por la actividad ventricular, además de la diferencia de su

frecuencia cardiaca y debido a esto su duración debe mantenerse entre 320 a 440ms aproximadamente.

El complejo QRS representa la activación de los ventrículos, aparece normalmente como una (monofásico), dos (bifásico) o tres (trifásico) ondas individuales, antes del complejo QRS, toda onda negativa se encuentra anterior a esta (Zavala, 2018). Tiene una duración que se encuentra entre 80 a 100ms. Además, este espacio comprende desde el inicio de la Onda Q hasta el final de la Onda S.

Segmentos

El segmento P-R, este se da cuando una activación ventricular empieza, y cuando hay una recuperación ventricular al final, esta corresponde a la sístole y diástole eléctricas, se mide antes de iniciar el complejo QRS, estando existente o no la onda Q hasta el finalizar la Onda T, teniendo un tiempo de duración de 0.32 y 0.40 mm/seg. (Zavala, 2018). Esta está comprendida entre el final de la onda P y el inicio del complejo QRS.

El segmento S-T es el transcurso del tiempo donde existe una pausa que divide la despolarización ventricular de la repolarización ventricular, suele ser isoeléctrico (Zavala, 2018). La distancia de este se encuentra desde el final del QRS, hasta el inicio de la Onda T, también llamado punto J.

Derivaciones electrocardiográficas

Según (Salatino, 2021) nos indica que un ECG estándar consta de 12 derivaciones, que consisten en medir la diferencia de potencial que hay entre 2 puntos de vista para decidir su polaridad. Se dividen en:

- a) estándar o bipolares (D1, D2, y D3): que conectan brazo derecho-brazo izq.; brazo derecho-pierna izquierda; y, brazo izq.-pierna izq., respectivamente.
- b) uni o monopolares: destinadas a medir el potencial de un determinado lugar con interacción a un punto distante o cero (electrodo neutro), que se sitúa en la pierna derecha.

Entre estas poseemos aVR, aVL, y aVF (o aumentadas de las extremidades derecha, izquierda y pierna izquierda respectivamente), y las precordiales V1, V2, V3, V4, V5, y V6, que rastrean las diferentes superficies del músculo cardíaco desde el muro anterior del tórax.

Instrumentos de medición de frecuencias cardiacas

Según (Pino et al., 2019) nos da a conocer mediante su artículo que los científicos del deporte como de ciencias del movimiento humano se encuentra relacionado con el ritmo cardiaco mediante la actividad eléctrica que manda el corazón por medio de la interpretación de las señales. Por medio de estas ondas proporcionadas por los instrumentos de medición, podemos obtener la información elemental de acuerdo con lo relacionado con el ritmo cardiaco.

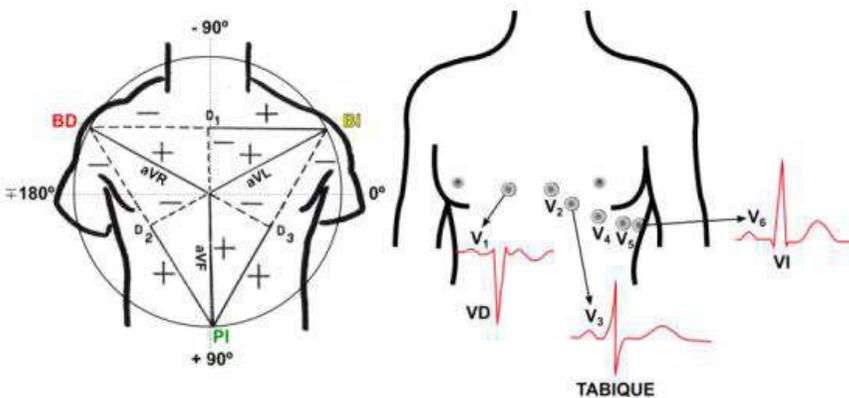


FIG 17. Derivaciones bipolares y monopares

En la imagen podemos observar las derivaciones con relación a los electrodos colocados en las distintas áreas del músculo cardíaco desde la pared anterior del tórax (Salatino, 2021)

Mediblu

Este es un nuevo ECG de 12 derivaciones que mejora drásticamente las tecnologías clave de ECG en términos de velocidad, precisión y estabilidad, y mejora el flujo de trabajo.

Adopta una nueva plataforma de adquisición de señales de ECG, que puede funcionar bien con reducción de interfaz, visualización de forma de onda rápida y nivel de filtrado adecuado (Mediblu, 2021).



FIG 18. Mediblu. (Mediblu, 2021)

1.4.2. Monitor Holter

Es un dispositivo portátil de electrocardiograma que se encarga de apuntar la actividad del corazón mientras realiza sus actividades diarias. Es una máquina que registra los ritmos cardíacos en forma continua.

Entre las formas que trabaja el monitor es mediante pequeños parches conductores que se pegan en el pecho y se conectan por medio de alambres a un pequeño monitor registrado. Se carga el monitor con Holter en un bolsillo o en un lugar seguro que se lleve alrededor del cuello o cintura mediante baterías. Bajo dicho contexto, las actividades que se registra las actividades eléctricas del corazón.

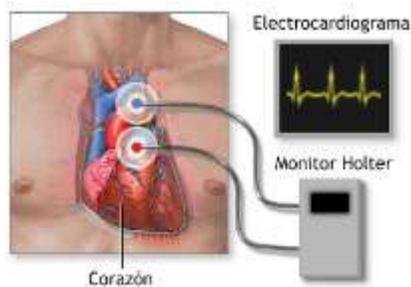


FIG 19. Monitor Holter. (Medineplus, 2022)

Tal y como se observa en la FIG 5, mientras se lleva puesto el

monitor, está registrando las actividades del corazón. Después de 24 a 48 horas, tiene que regresar el monitor al consultorio del médico de la atención. El proveedor observará los registros y mirará si ha habido algún ritmo cardíaco anormal.

Entre las recomendaciones que se hacen al usar el dispositivo es evitar sábanas eléctricas, áreas de alto voltaje, imanes y detectores de metales. Bajo dicho contexto, se debe continuar con las actividades normales mientras se usa el monitor. Es recomendable que realice ejercicio mientras se esté haciendo el monitoreo si los síntomas desfavorables han ocurrido en el pasado mientras se estaba ejercitando.

1.4.3. Monitor de Eventos

Para la arritmia esporádica, se debe mantener este dispositivo portátil encendido y pegado al cuerpo, y solamente activarlo cuando se experimenten los síntomas. El dispositivo permite mantener controlado el ritmo cardíaco al momento de sentir los síntomas.

Para realizar los exámenes, cada tipo de monitor es ligeramente diferente, pero todos tienen sensores (llamados electrodos) para registrar su ECG. En algunos modelos, estos se adhieren a la piel de su pecho usando parches adhesivos. Los sensores necesitan buen contacto con su piel. El poco contacto puede causar resultados deficientes.

Tabla 5. Tipos de monitor de eventos

Monitor de memoria continua	Monitor de eventos sintomáticos	Parches de registro	Monitor de implantable
Los electrodos permanecen adheridos a su pecho, y el monitor registra constantemente, pero no guardan su ECG. Cuando usted presenta síntomas, presiona un botón para activar el dispositivo. Luego este guarda el ECG poco antes, durante y después durante un	Este dispositivo registra su ECG solamente cuando los síntomas se presentan, no antes. Usted lleva el dispositivo en un bolsillo o lo usa en su muñeca	Este monitor no usa cables ni electrodos, monitorea continuamente la actividad de su ECG durante 14 días, utiliza un parche adhesivo que se adhiere a su	Es un pequeño monitor que se implanta bajo la piel del pecho. Puede dejarse

Monitor continua	de memoria	Monitor de sintomáticos	de eventos	Parches registro	de	Monitor implantabl e
tiempo después de que sus síntomas inicien. Algunos monitores de eventos se activan por sí solos cuando detectan ritmos cardíacos anormales				pecho		para monitorear el ritmo cardíaco por 3 o más años.

Nota: Se observa los tipos de monitores de eventos con los que se pueden contar. Elaboración y fuente propia.

Entre los consejos al usar el dispositivo se indican que las actividades se deben desarrollar de forma normal y en ocasiones es importante hacer ejercicio. Llevar un diario de las actividades que realice mientras usa el monitor, cómo se siente y cualquier síntoma que presente. Esto le ayudará a su proveedor de atención médica a relacionar los síntomas con los hallazgos de su monitor. El personal de la estación de monitoreo le dirá cómo transferir la información desde el teléfono.

Mientras usa el dispositivo, posiblemente una de las recomendaciones será que evite ciertas cosas que pueden alterar la señal entre los sensores y el monitor que pueden incluir:

- Teléfonos celulares.
- Mantas eléctricas.
- Cepillos de dientes eléctricos.
- Zonas de alto voltaje.
- Imanes.
- Detectores de metales.

1.4.4. Alivecorkardia

El monitor de ECG AliveCor Kardia permite una detección muy precisa de la fibrilación auricular por parte de un electrofisiólogo intérprete tanto en la derivación estándar I como en una nueva derivación paraesternal. El algoritmo de diagnóstico que ofrece el sistema puede ser útil en la detección de grabaciones para una revisión adicional. Desafíos diagnósticos presentes en el aleteo auricular y la estimulación del marcapasos ventricular (Wegner et

al., 2020).

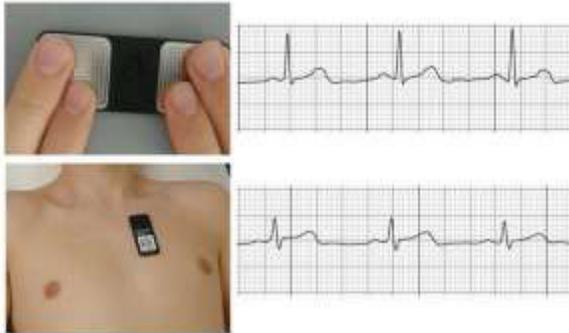


FIG 20. Toma de ECG con Kardia. Foto tomada de (Wegner et al., 2020).

1.4.5. Registrador de bucle implantable

Este dispositivo este encargado de detectar los ritmos cardíacos anormales y se implanta debajo de la piel en la zona del pecho. Es un monitoreo para el sistema cardíaco que permite registrar ritmo del corazón de forma constante durante hasta tres años. Permite, además, llevar un control de latidos cardíacos de manera remota mientras el paciente se desenvuelve en las actividades diarias.

La manera en la que se implanta el dispositivo que se llama registrador de eventos cardíacos, se coloca justo debajo de la piel en el pecho en donde se debe realizar porque el médico tiene que registrar latidos del corazón anormales como arritmias, accidentes cerebrovasculares sin causa aparente y desmayo que puede ser provocado por síncope. Bajo dicho contexto, puede captar información que puede ser difícil para un electrocardiograma o monitor Holter especialmente en arritmias breves o poco frecuentes.

Un hecho a destacar es que para realizar el procedimiento del registrador de bucle implantable no se necesita un requisito previo. Luego, se realiza la colocación del bucle por debajo de la piel, en un centro médico donde es importante que el paciente está despierto durante el procedimiento, pero puede ser que te den un

medicamento para que te relajés (sedante). La piel en el área del pecho debe estar adormecida para que el médico hará una incisión pequeña, insertará el dispositivo y cerrará la incisión para tomar las medidas hasta tres años.

Luego de realizar la intervención y esté implantado el Registrador de bucle rara vez se presentan complicaciones. Sin embargo, como es una cirugía menor deberá estar atento a cualquier signo de infección cuyas señales pueden ser enrojecimiento o hinchazón. El médico deberá informar que medicamentos o como controlar los posibles problemas que se presenten. Una recomendación que se suele decir es limitar las actividades que lleven esfuerzo hasta que la herida esté cicatrizada.

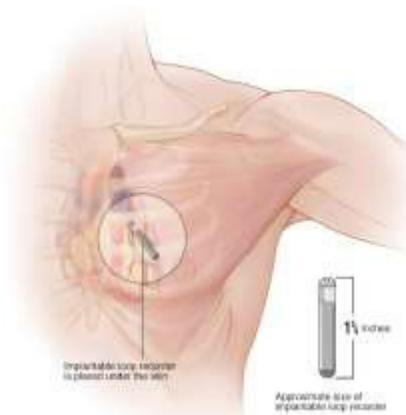


FIG 21. Registrador de bucle implantable. (Solbiati et al., 2016)

Tal y como se observa en la FIG 6 el registrador de bucle implantable, una vez instalado, se entregará un dispositivo transmisor que deberás colocar junto a tu cama. Mientras el paciente está descansando, el transmisor envía información automáticamente desde el implante a tu médico. También puedes presionar un botón para enviar datos de inmediato cuando tienes síntomas.

1.5. Resumen

El corazón al estar alojado en el tórax cuyas dimensiones y peso dependen del género de la persona ya que, con respecto a dicho parámetro varía entre 0.45% y 0.40% del peso en el caso de los hombres y mujeres respectivamente, siendo en forma de cono invertido y el funcionamiento se reparte entre un corazón derecho e izquierdo que permiten complementar las funciones de bombeo de sangre a todo el cuerpo y órganos permitiendo la estabilidad.

Bajo dicho contexto, hay trastornos provocados por efecto de la edad, herencia familiar y estilos de vida que afectan al comportamiento cardíaco que usual que debe presentar un paciente como la presión arterial alta y arritmias cardíacas que se relacionan al funcionamiento eléctrico y por ello, se han desarrollado técnicas a través de instrumentos tecnológicos para la detección de problemas como lo son los electrocardiogramas, monitor de Holter, monitor de eventos, Alivecorkardia y el registrado de bucle implantable que presentan de forma gráfica las anomalías en el funcionamiento del órgano que permiten la detección y diagnóstico oportuno.



CAPÍTULO 2:

PRESENCIA DE LA INTELIGENCIA ARTIFICIAL EN ÁREAS DE LA INGENIERÍA Y SALUD MÉDICA.

2.1. Inteligencia Artificial

La inteligencia artificial (IA) es la ciencia que permite la imitación de procesos inteligentes a través de una máquina (Duarte & Uribe, 2019). A través de algoritmos esta, es entrenada para que cumpla una tarea, capaz de manipular y aprender a través de una cantidad masiva de datos, en los cuales la IA busca patrones que le permitan resolver el problema (Aristega & Mendoza, 2022). Esto permite a un modelo de inteligencia artificial resolver problemas y realizar distintas tareas de manera automática con un bajo margen de error. Actualmente se han desarrollado diferentes tipos de algoritmos que permiten solventar problemas de diferentes áreas como lo es la medicina.



FIG 22. *Inteligencia artificial en medicina. Foto tomada de (IBM, 2022)*

Una inteligencia artificial realiza una predicción en cuestión de segundos, ayuda a disminuir el tiempo de interpretación de la información. La clasificación y la interpretación de imágenes es una de las tareas que tiene un especialista en medicina, el cual a través de distintos tipos de tecnología como el electrocardiograma, rayos x, ultrasonidos etc., establece un diagnóstico hacia un paciente (Leivi, 2019). El tiempo que normalmente es usado para realizar una ecografía o electrocardiograma, pueden ser reducidas al ser realizadas por una IA que realice estos análisis con solo presentarle la imagen.

2.1.1. Diferencia entre la Inteligencia Artificial y Machine Learning.

Tenemos que la Inteligencia Artificial es la capacidad que tienen las computadoras de mostrar un comportamiento inteligente mientras que la Machine Learning es una técnica que se usa para crear y perfeccionar dicho comportamiento mediante entrenamientos de forma automática con exposición de datos.

Tabla 6. *Habilidades requeridas en Inteligencia Artificial y Machine Learning.*

Habilidades requeridas en la Inteligencia Artificial	Habilidades requeridas en el Machine Learning
Algoritmos y técnicas para analizarlos.	Matemáticas aplicadas.
Aprendizaje automático y cómo aplicar técnicas para extraer inferencias a partir de datos.	Arquitecturas de redes neuronales.
Cuestiones éticas en el desarrollo de tecnologías de IA responsables.	Física.
Data Science.	Modelado y evaluación de datos.
Robótica.	Procesamiento natural del lenguaje.
Programación.	Lenguajes de programación.
Diseño de programación.	Probabilidades y estadísticas.
Procesamiento de datos.	Algoritmos.
Resolución de problemas.	
Machine learning.	

Nota: Se observa las habilidades requeridas. Elaboración propia

Tal y como se observa en la tabla anterior, hay habilidades que se exige al momento de desarrollar soluciones basadas en Inteligencia Artificial y Machine Learning, por ejemplo, en el

primer caso Data Science y en el segundo caso no.

2.1.2. Relación entre la Inteligencia Artificial y machine Learning

En un comienzo antes del año 80, la inteligencia artificial era una capacidad que se conseguía al alcanzar una serie de reglas que les decían a las computadoras que debían hacer, cómo y en qué momento. De esta manera, un sistema inteligente solo obedecía a la regla con la cual estaba configurada. Por ejemplo:

Al jugar al ajedrez, una computadora debía tener configurada reglas como: “Si encuentras un caballo en tu línea de ataque, debes comértelo” De ese modo, al crear reglas de este tipo, muchas computadoras ya podían jugar ajedrez.

Al surgir el ML esa técnica basada en reglas con inteligencia artificial se abandonó. Debido a que el Machine Learning aportaba a las computadoras una capacidad de aprendizaje mucho más aplicada al concepto de Inteligencia artificial.

Este proceso de aprendizaje está basado en exponer a la computadora a muchos datos para que pueda procesarlos, analizarlos y aprender de ellos.

Siguiendo con el ejemplo del ajedrez, ya haciendo uso del ML significa dotar a la computadora con una gran base de datos de partidas de ajedrez. De esta manera, el sistema inteligente estudia cada una de las partidas, identifica cuales son los movimientos que ayudaron a conseguir la victoria, y de esta manera aprende a jugar ajedrez (Lazcano, 2020).

2.1.3. Beneficios de aplicar inteligencia artificial con pacientes clínicos

Las herramientas que incorporan Inteligencia Artificial en su sistema pueden identificar relaciones significativas mediante los datos proporcionados que pueden aplicarse en diversos campos siempre y cuando se obtenga información para construir un

modelo fiable. Tiene el potencial de aplicarse en diversos campos como la construcción de fármacos, tratamientos a pacientes y seguridad financiera en el ámbito médico.

La Inteligencia Artificial llegó para mejorar diversas áreas de la salud como, por ejemplo, el seguimiento del paciente en problemas de salud en los que es indispensable el constante monitoreo con el apoyo de dispositivos electrónicos y tecnología vestible.

En el caso de la elaboración de fármacos, reduce el tiempo que toma identificar el comportamiento de las células de interés en el estudio para evaluar el comportamiento y llegar a el resultado esperado.

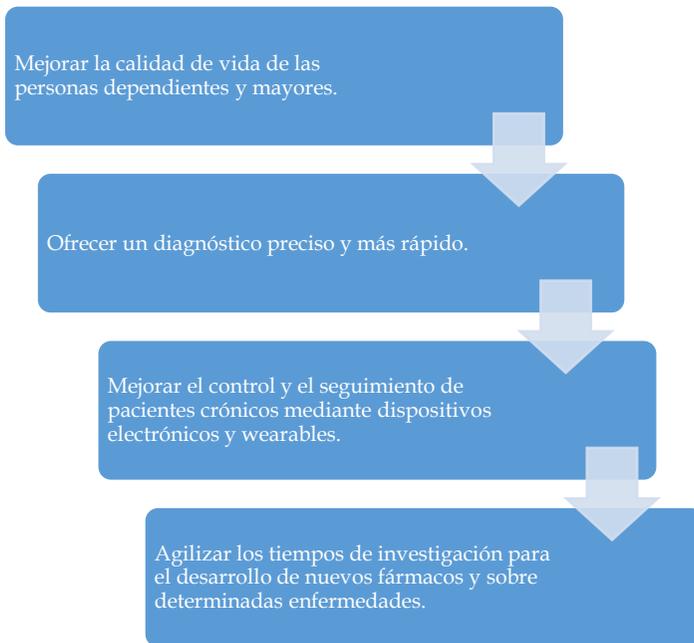


FIG 23. Beneficios de integrar IA en la medicina. (APD, 2021).

Permite, además, acelerar el diagnóstico médico de manera precoz en enfermedades agresivas y degenerativas como lo puede ser el cáncer, generación de vacunas, análisis médico con diversos datos en forma de fotos de rasgos faciales o aspecto físico del paciente. Así mismo, para mantener estable la salud en caso de embarazo y asegurar un buen desarrollo fetal y en el caso de

pacientes que necesiten de prótesis, se benefician con la personalización y adaptación que se construyen (Regalado & Medina, 2022).

2.1.4. La Inteligencia Artificial en desarrollo de fármacos

Aunque el tiempo tradicional estimado de desarrollo de una vacuna dura 10 años, con el uso de soluciones con Inteligencia Artificial la fabricación se reduce a la mitad, pero dependiendo del tipo de medicamento o vacuna a desarrollar varía porque se debe tener en cuenta de qué papel se le va a pedir al sistema de IA para analizar los datos. Por lo tanto, ya hay empresas como Pfizer que trabaja con IBM Watson en un sistema de Machine Learning que logra aumentar sus avances en fármacos para inmunologías oncológicas (Tobalina, 2020).

Uno de los objetivos es predecir la forma en que se comportan determinadas moléculas. Entre las principales razones que llevan a empresas a utilizar IA en el desarrollo de fármacos tenemos:

- Acelerar y disminuir el fracaso en los ensayos clínicos de la propuesta a desarrollar.
- Aumentar el margen de búsqueda de nuevos medicamentos.
- Disminuir los costos que se generan al investigar y desarrollar.

2.1.5. Evolución de la Inteligencia Artificial

Para entender este panorama existen 2 aspectos claves que son:

- Una cosa es establecer un sistema para que realice una determinada tarea.
- Y otra cosa es configurar el sistema para que aprenda a realizar por su propia cuenta la tarea asignada (Esto es lo que hace ML).

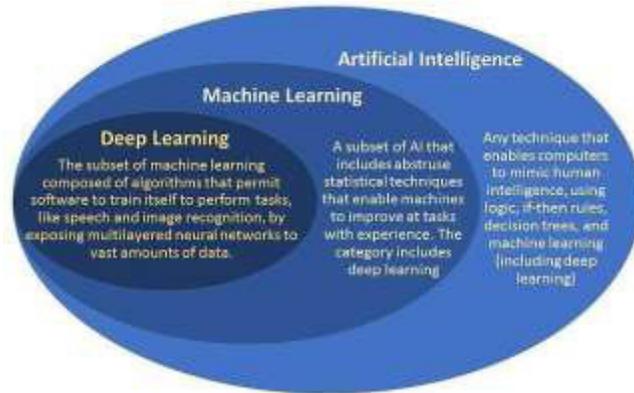


FIG 24. Evolución de la IA (Lazcano, 2020)

2.2. Tipos de Algoritmos de Machine Learning

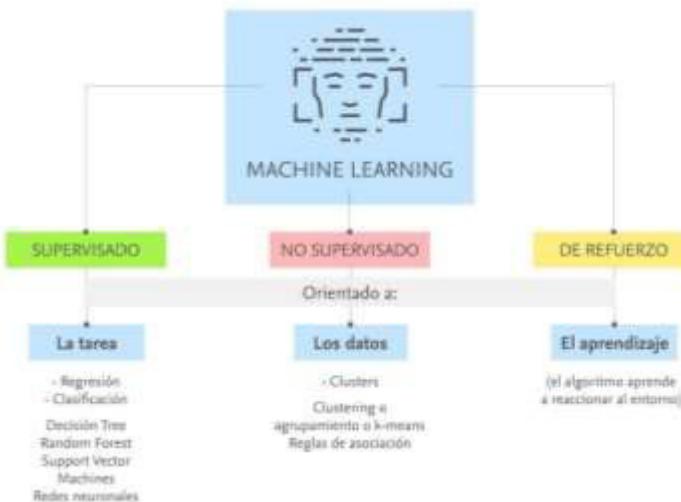


FIG 25. Se explican los algoritmos de Machine Learning

Los algoritmos de Machine Learning constituyen el subcampo de inteligencia artificial que permiten recibir y organizar datos de entrada que producen valores acordes a el modelo que se entrenó dentro de un rango aceptable. Bajo dicho contexto, es importante indicar que a medida que se ingresan más datos, el algoritmo mejora la calidad de respuesta dependiendo de la calidad de información ingresada.

Aplicaciones prácticas del Machine Learning

- **Recomendaciones:** El machine Learning permite realizar sugerencias personalizadas en plataformas de compras online. En su manera más natural analiza el historial de compras y lo compara con compras hechas por otros usuarios con tendencias o gastos parecidos.
- **Vehículos inteligentes:** Según el estudio realizado por Automotive 2025: industry without borders de IBM, en el año 2025 ya podremos observar vehículos en inteligentes en la carretera, gracias al aprendizaje automático, a estos vehículos el usuario podrá ajustar la configuración interna dependiendo de su gusto.
- **Redes Sociales:** Como ejemplo presentamos a Twitter, hace uso de algoritmos de Machine Learning para reducir el spam publicado en esta red Social. Mientras tanto Facebook lo usa para detectar noticias falsas o contenido impropio que rompa las reglas impuestas por el mismo.
- **Procesamiento de Lenguaje Natural:** Gracias a la comprensión del Lenguaje humano, los asistentes virtuales como Alexa o Siri pueden traducir de manera inmediata de un idioma a otro, reconocer la voz del usuario, inclusive examinar sus sentimientos.
- **Búsqueda:** Los motores de búsqueda a través del aprendizaje automático busca seleccionar sus resultados en función de su eficacia, midiéndolo por medio de los clics del usuario.
- **Medicina:** Los investigadores del instituto de Tecnología de Massachusetts, ya trabajan con el Machine Learning para detectar con mayor precisión el cáncer de mama o cualquier otra enfermedad, algo que es de vital importancia ya que su detección temprana significa mayor probabilidad de curación.
- **Ciberseguridad:** En la actualidad los nuevos antivirus y motores de detección de programa maligno, se basan en el aprendizaje automático para impulsar el escaneado, acelerar la detección y mejorar mucho más la habilidad de reconocer anomalías.

2.2.1. Aprendizaje Supervisado

El aprendizaje supervisado etiqueta cada grupo de datos en un espacio de acuerdo con criterios específicos, este aprendizaje generaliza el conocimiento a partir de ciertas reglas que clasifican la salida del sistema como correcta o incorrecta según el contexto (Duarte & Uribe, 2019). Por ejemplo, en la FIG 4, se puede observar como un algoritmo se le envía un conjunto de datos en primera instancia, luego el modelo es entrenado en base a ciertos datos previamente clasificados, para finalmente estas, pueden realizar análisis o predicciones de otros datos.

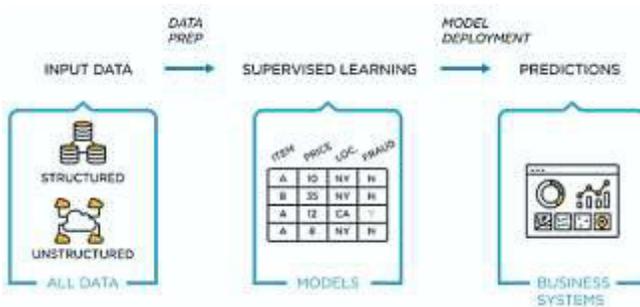


FIG 26. En esta figura podemos visualizar el procesamiento del Aprendizaje supervisado. (Aristega & Mendoza, 2022)

Tal y como se observa en la figura anterior, el aprendizaje supervisado resuelve problemas conocidos y utiliza un conjunto de datos etiquetados para entrenar un algoritmo para realizar tareas específicas. Utiliza modelos para predecir resultados conocidos como "¿Cuál es el color de la imagen?" "¿Cuántas personas hay en la imagen?" "¿Cuáles son los factores determinantes para el fraude o los defectos del producto?" etc. Por ejemplo, un proceso de aprendizaje supervisado podría consistir en clasificar vehículos de dos y cuatro ruedas a partir de sus imágenes.

Los datos de entrenamiento tendrían que estar correctamente etiquetados para identificar si un vehículo es de dos o cuatro ruedas.

El aprendizaje supervisado permite que los algoritmos 'aprendan' de datos históricos/de entrenamiento y los apliquen a

entradas desconocidas para obtener la salida correcta. Para funcionar, el aprendizaje supervisado utiliza árboles de decisión, bosques aleatorios y Gradient Boosting Machine.

Existen dos tipos principales de aprendizaje supervisado; clasificación y regresión. La clasificación es el lugar donde se entrena a un algoritmo para clasificar los datos de entrada en variables discretas. Durante el entrenamiento, los algoritmos reciben datos de entrada de entrenamiento con una etiqueta de 'clasificación'. Por ejemplo, los datos de entrenamiento pueden consistir en las últimas facturas de tarjetas de crédito de un conjunto de clientes, con la etiqueta de si realizaron una compra futura o no fue así.

Cuando el saldo de la tarjeta de un nuevo cliente se presenta al algoritmo, este clasificará al cliente en el grupo de "comprará" o "no comprará".

2.2.1.1. Regresión Lineal

La regresión es una técnica para predecir un valor objetivo utilizando factores independientes. Este enfoque se utiliza sobre todo para predecir y determinar la conexión causa-efecto de las variables. Las técnicas de regresión en su mayoría difieren según el número de variables independientes y el tipo de relación entre las variables independientes y dependientes (Gandhi, 2018).

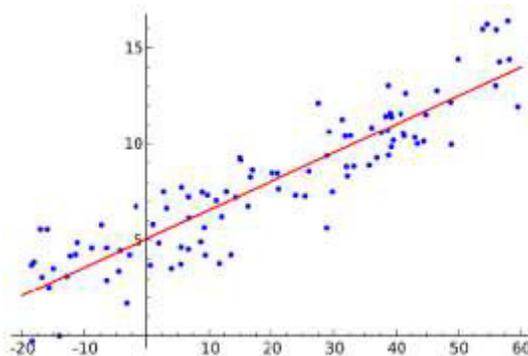


FIG 27. Representación de Regresión Lineal

La regresión lineal es un modo de análisis de regresión en el que sólo hay una variable independiente y existe una conexión lineal entre ambas variables. La línea roja en el gráfico anterior se conoce como la línea recta de mejor ajuste (Gandhi, 2018).

2.2.1.2. Regresión Logística

Es una técnica de machine learning que se usa para clasificar. Es una herramienta de predicción que se origina en la probabilidad. Las variables independientes se analizan para determinar el resultado binario y los resultados caen en una de las dos categorías. Las variables independientes pueden ser categóricas o numéricas, pero la variable dependiente es siempre categórica (Hojas, 2021).

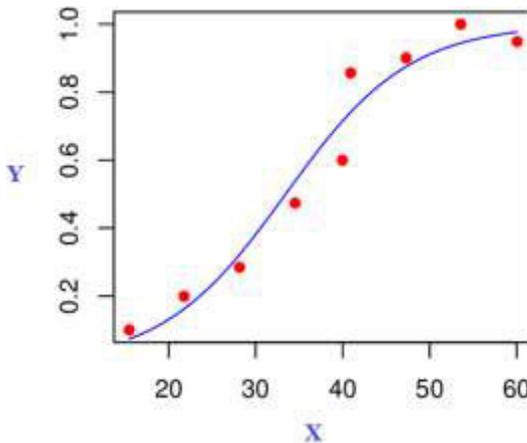


FIG 28. Representación de curva sigmoide en Regresión Logística

2.2.1.3. Árbol de decisiones

Es una forma de algoritmo de aprendizaje supervisado que se emplea sobre todo para cuestiones de clasificación. Es aplicable tanto a variables dependientes categóricas como continuas. Con este algoritmo se separa la población en dos o más conjuntos homogéneos. Esto se hace en función a las cualidades más importantes o variables independientes para establecer grupos tan diferentes como sea factible (España, 2021).

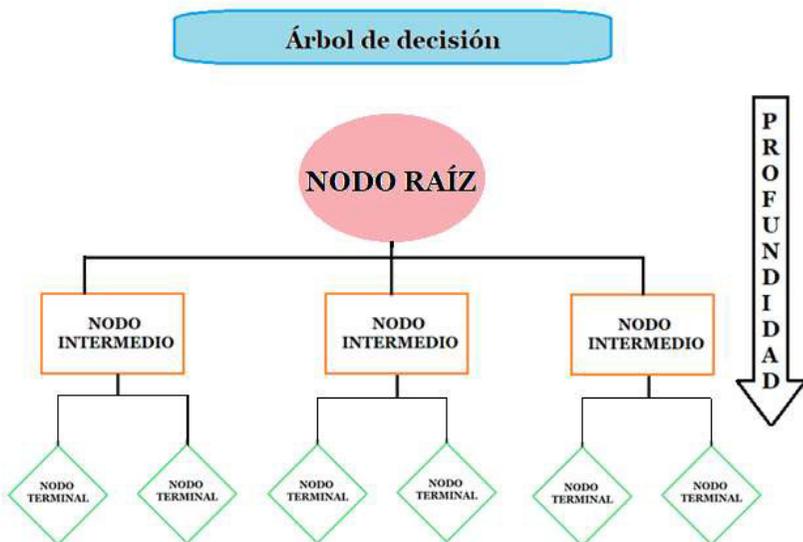


FIG 29. Representación de Árbol de decisión y sus respectivos nodos

2.2.1.4. Bosque Aleatorio

Son clasificadores con una especie de enfoque de aprendizaje conjunto utilizado para la clasificación, la regresión y otras labores que pueden desempeñar los árboles de decisión. Estos árboles de decisión se pueden construir en el momento del entrenamiento y el resultado de la clase puede ser clasificación o regresión. Con la ayuda de estos bosques aleatorios, se puede corregir el hábito de sobre ajustar al conjunto de entrenamiento (Silipo, 2019). Algunas de las ventajas y desventajas de los clasificadores de bosques aleatorios son las siguientes:

- **Ventajas:** Los clasificadores de bosque aleatorio facilitan la reducción del ajuste excesivo del modelo y estos clasificadores son más precisos que los árboles de decisión en varios casos.
- **Desventajas:** Los bosques aleatorios exhiben predicción en tiempo real, pero eso es lento por naturaleza. Además, son complejos de implementar.

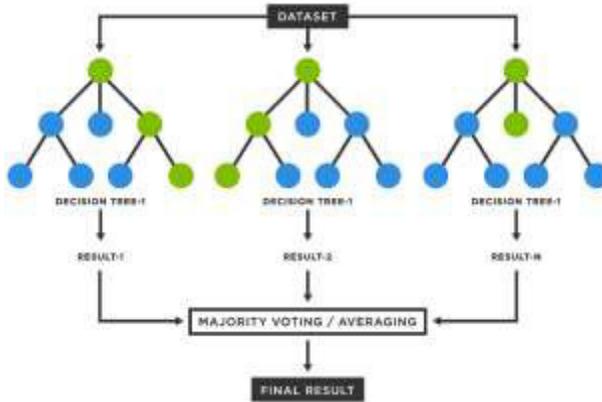


FIG 30. Un bosque aleatorio se conforma por varios árboles de decisiones

2.2.1.5. Máquina Vector de Soportes (SVM)

Cada elemento de datos en este enfoque se muestra como un punto en un espacio n-dimensional, y el valor de cada elemento es el valor de una coordenada determinada.

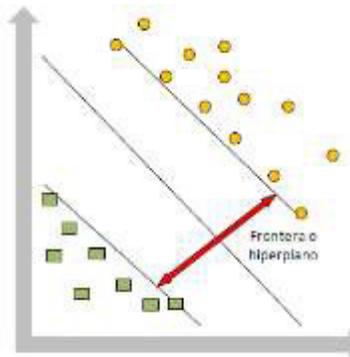


FIG 31. La frontera o el hiperplano separa las clases

- Ventajas: funciona relativamente bien cuando hay un claro margen de separación entre clases. Es más efectivo y relativamente eficiente en memoria.
- Desventajas: No es adecuado para grandes conjuntos de datos. No funciona muy bien cuando el conjunto de datos tiene más ruido, es decir, las clases objetivo se superponen.

2.2.1.6. Clasificador Naive Bayes

Es empleado principalmente en la clasificación de texto que incluye un grupo de datos de entrenamiento de alta dimensión. Es uno de los algoritmos de clasificación más simples y efectivos que ayuda a construir modelos de aprendizaje automático que pueden hacer predicciones rápidas. Es un clasificador probabilístico, lo que significa que predice sobre la base de la probabilidad de un objeto (Camacho, 2021).

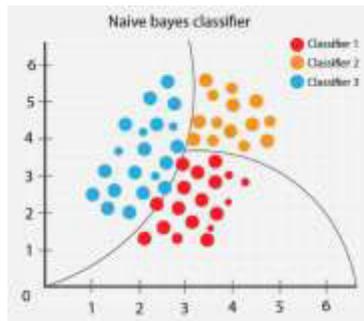


FIG 32. Se detalla la separación en distintos grupos bajo el clasificador Naive Bayes

2.2.1.7. K-Nearest Network

Es aplicable tanto a las problemáticas de clasificación como de regresión. No obstante, se utiliza más comúnmente en las dificultades de categorización industrial. El método sirve para calcular el número de personas que vivan más cerca unas de otras es un algoritmo sencillo que mantiene todas las instancias conocidas y clasifica los nuevos asuntos basándose en el voto mayoritario de sus k vecinos. La instancia asignada es más frecuente entre sus K vecinos más cercanos, según una función de distancia (García Cambroner, 2019).

2.2.2. Aprendizaje No Supervisado

La característica principal de este tipo de aprendizaje es que los datos en el espacio no están clasificados, a diferencia del aprendizaje supervisado, cada dato tiene un valor en función de las variables, pero aparte de ocupar espacio, no tiene una clasificación

o permitir que tenga un parámetro asociado (Duarte & Uribe, 2019). Dado que el algoritmo no realiza recibe una clasificación de entrada, este intenta buscar un patrón en los datos. En la FIG 5 podemos ver que los datos al no tener una clasificación como tal, se agrupan de distintas formas intentando crear un patrón.

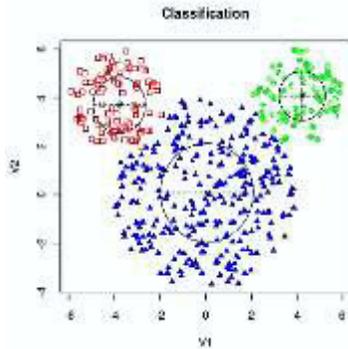


FIG 33. Algoritmo de Aprendizaje No Supervisado: K-Medias.(Caparrini, 2020)

Para este tipo de aprendizaje se lleva a cabo las siguientes técnicas:

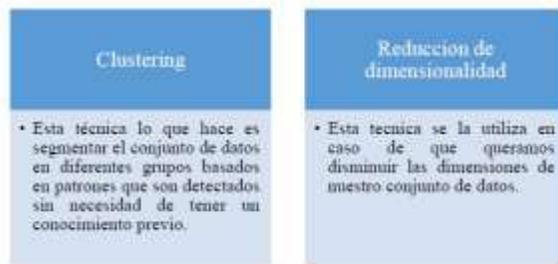


FIG 34. Técnicas del aprendizaje no supervisado

Los algoritmos más conocidos del Aprendizaje no Supervisado son:

- K-means Clustering.
- t-SNE.
- PCA.
- SVD.

El aprendizaje no supervisado es un tipo de Machine Learning que se utiliza para identificar nuevos patrones y detectar anomalías. Los datos que se introducen en los algoritmos de aprendizaje no supervisados no están etiquetados.

El algoritmo (o modelos) intentan dar sentido a los datos por sí mismos mediante la búsqueda de características y patrones. Una pregunta de muestra que Machine Learning sin supervisión podría responder es "¿Están surgiendo nuevos clústeres de fraude o patrones de compra o modos de falla?" El aprendizaje no supervisado utiliza el agrupamiento, los componentes principales, las redes neuronales y las máquinas de vectores de soporte.

2.2.2.1. K-Means

Se trata de un método no supervisado que se utiliza para abordar la problemática de la agrupación. Su proceso emplea un enfoque sencillo y directo para clasificar un grupo de datos determinados en un número predeterminado de agrupaciones (suponga k conglomerados). Los puntos de datos dentro de un clúster son homogéneos y heterogéneos para los grupos de pares (García Cambroner, 2019).

El algoritmo consta de los siguientes pasos:

- **Inicialización:** una vez escogido el número de grupos, k , se establecen k centroides en el espacio de los datos, por ejemplo, escogiéndolos aleatoriamente.
- **Asignación objetos a los centroides:** cada objeto de los datos es asignado a su centroide más cercano.
- **Actualización centroides:** se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

Se repiten los pasos 2 y 3 hasta que los centroides no se mueven, o se mueven por debajo de una distancia umbral en cada paso. El algoritmo k -means resuelve un problema de optimización, siendo la función para optimizar (minimizar) la suma de las distancias cuadráticas de cada objeto al centroide del cluster.

2.2.3. Aprendizaje Profundo

El objetivo de este algoritmo es de que aprenda a partir de su propia experiencia. Que sea capaz de tomar las mejores decisiones ante diferentes situaciones de acuerdo con un proceso de prueba y error donde se recompensa las decisiones correctas. El modelo se entrena hasta tal punto que cada reacción que realice recibirá recompensa o una penalización. Comenzando con prueba y error localiza el resultado que maximiza las recompensas.

2.2.4. Aprendizaje por refuerzo

El objetivo de este algoritmo es de que aprenda a partir de su propia experiencia. Que sea capaz de tomar las mejores decisiones ante diferentes situaciones de acuerdo con un proceso de prueba y error donde se recompensa las decisiones correctas. El modelo se entrena hasta tal punto que cada reacción que realice recibirá recompensa o una penalización. Comenzando con prueba y error localiza el resultado que maximiza las recompensas.

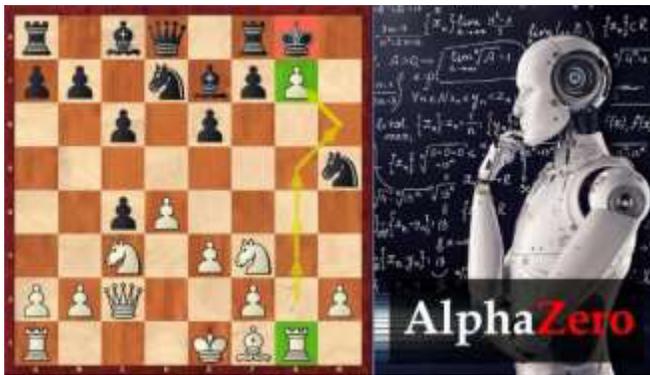


FIG 35. Alpha Zero es una inteligencia Artificial que juega al ajedrez haciendo uso de este aprendizaje

El machine Learning hace uso de los algoritmos tanto supervisados como no supervisados para descubrir el comportamiento y la tendencia que tiene los datos gracias, a un proceso de entrenamiento similar al del cerebro humano.

2.2.4.1. Redes Neuronales Recurrentes

Es una forma de red neuronal artificial que trabaja con series temporales o datos secuenciales. Estos algoritmos de aprendizaje profundo se usan comúnmente para problemas ordinales o temporales, como traducción de idiomas, reconocimiento de voz y subtítulos de imágenes; se incorporan a aplicaciones populares como Siri, búsqueda por voz y Google Translate (Education, 2021).

Son distinguidas por su memoria, ya que toman información de entradas anteriores para influir en la entrada y salida actual. Mientras que las redes neuronales profundas tradicionales asumen que las entradas y salidas son independientes entre sí, la salida de las redes neuronales recurrentes depende de los elementos previos dentro de la secuencia (Education, 2021).

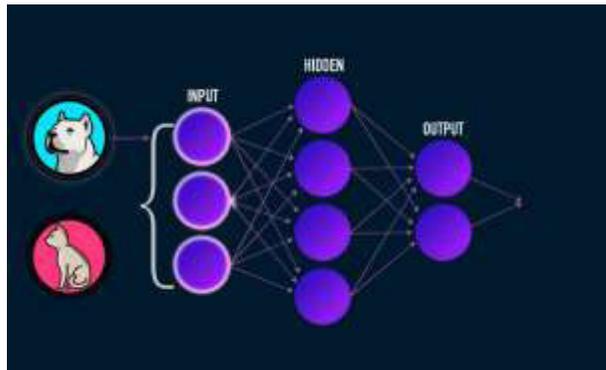


FIG 36. Red Neuronal de convolución

Una red neuronal convolucional es un algoritmo de aprendizaje que puede apropiarse de una imagen de entrada, asignar características a varios objetos en la imagen y ser apto para diferenciar uno del otro.

Los pasos más importantes para el algoritmo CNN son:

- La Capa de convolución considerada como la capa más importante en el CNN.

- La Función de activación tiene como característica potencia de refuerzo, especialmente para la capa ReLu.
- Agrupación.
- Aplanamiento significa conversión de forma de matriz a una sola columna grande.
- La Capa de activación.
- Conexión completa.

Tensorflow hub proporciona una gran cantidad de ejemplos y algoritmos que nos ayuda a la clasificación de imágenes, en donde se encuentran muestras de algoritmos convolucionales clasificadores de imágenes, desde el más básico al más avanzado.

También cuenta con algoritmos de transferencia de aprendizaje; que son en pocas palabras algoritmos preentrenados los cuales cuentan con varias capas de convolución ajustadas de tal manera que puedan extraer una gran cantidad de características.

En el presente trabajo se realizaron pruebas de 3 algoritmos convolucionales de transferencia de aprendizaje que son ResNet, InceptionV3 y MobileNetV2 con el fin de encontrar el algoritmo más adecuado en nuestro trabajo.

MobileNets son redes neuronales convolucionales eficientes para aplicaciones de visión móvil, se basa en una arquitectura simplificada que utiliza circunvoluciones separables en profundidad para construir redes neuronales profundas y livianas (Howard et al., 2017) Este modelo permite realizar detección de objetos, clasificación de grano fino, detección de atributos faciales y la geolocalización a gran escala.

MobileNetV2 mejora el rendimiento de última generación de los modelos móviles en múltiples tareas y puntos de referencia, así como en un espectro de diferentes tamaños de modelos (Sandler et al., 2018). Usa convoluciones profundas livianas para filtrar características en la capa de expansión intermedia.

Un algoritmo por transferencia de aprendizaje es un modelo que se entrenó previamente con un gran conjunto de datos, permitiendo

aprovechar estas capas sin tener que empezar un algoritmo desde cero, ahorrándonos trabajo en lo que respecta a las capas convolucionales.

Una de las características principales de este tipo de algoritmo es que permite extraer una gran cantidad de características con una cantidad de datos muy baja, siéndonos de gran utilidad en este trabajo, permitiéndonos clasificar la cantidad de imágenes que disponemos en este trabajo.

Se escogieron los modelos MobileNetV2, Inception, Resnet debido a que tiene compatibilidad con la biblioteca de Tensorflow y su facilidad de importación usando Google colab.

2.2.5. Algoritmos de transferencia de aprendizaje

Para este trabajo se usó la técnica de transferencia de aprendizaje en inglés Transfer learning. Según (Bazan Yaranga, 2018) la transferencia de aprendizaje es un método de aprendizaje automático, que se apoya en practicar una construcción neuronal para una tarea específica o similar a esta.

Lo que nos da a explicar el autor es que la técnica de transferencia de aprendizaje es un modelo en el cual podemos realizar varias tareas a la vez a partir de un proceso de construcción de datos.

Comúnmente se usa transferencia de aprendizaje de dos formas:

Modelo Pre-entrenado: Se fundamenta en utilizar un modelo Pre-entrenado, y reemplazar sus últimas capas por otras, para que de esa forma se extraigan las características del nuevo conjunto de datos.

Sintonización de la red convolucional: Esta estrategia se basa en sintonizar los pesos de las capas superiores usando propagación hacia detrás (en inglés “backpropagation”).

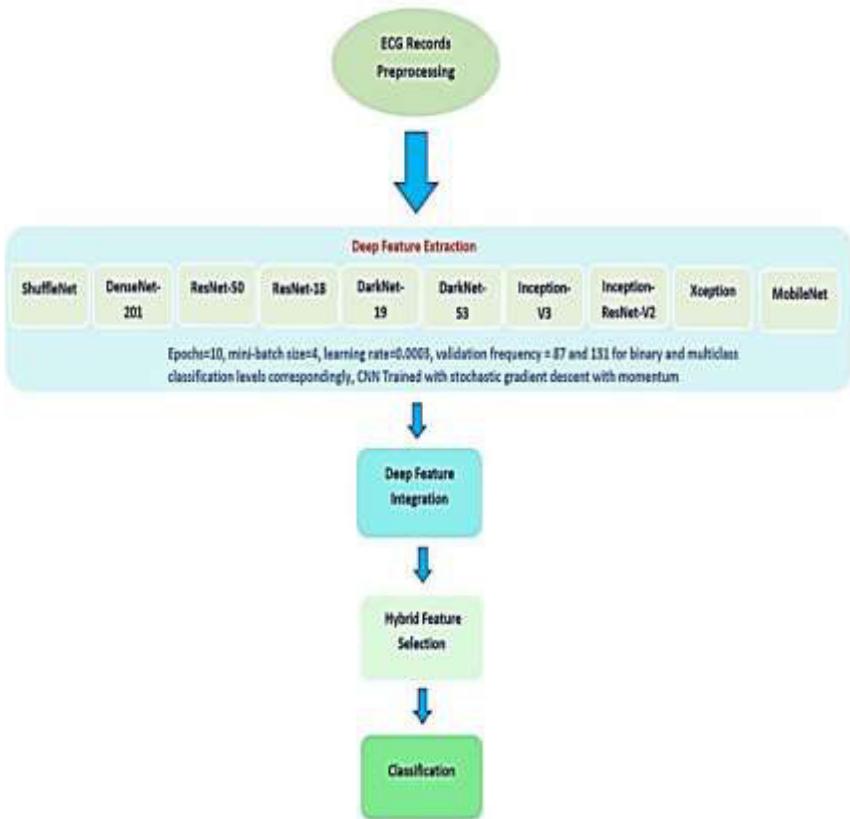


FIG 37. Diagrama que describe los pasos de la herramienta de clasificación. (Attallah, 2022)

Existen en la actualidad una gran cantidad de algoritmos de transferencia de aprendizaje, entre los más populares se encuentran: ResNet, ShuffleNet, Inception, MobileNet, Xception, DarkNet.

Inception v3

Google propuso la arquitectura Inception CNN en 2016. Es una versión actualizada de GoogleNet con algunos cambios. Se introdujo por primera vez para reducir los requisitos de memoria y el costo computacional.

Su componente principal es la unidad de inicio, que combina varios filtros en una nueva estructura de filtros, lo que reduce el número de parámetros (Attallah, 2022).

En comparación con arquitecturas más simples y monolíticas, este modelo puede proporcionar redes de visión de alto rendimiento con un costo computacional relativamente bajo (Szegedy et al., 2016).

Esta arquitectura utiliza menos cantidad de cálculos en comparación con otros modelos de transferencia de aprendizaje, en la (figura 18), se muestra la arquitectura de InceptionV3, en donde para ser utilizado se debe desactivar la última capa e implementar nuestro dataset clasificado.

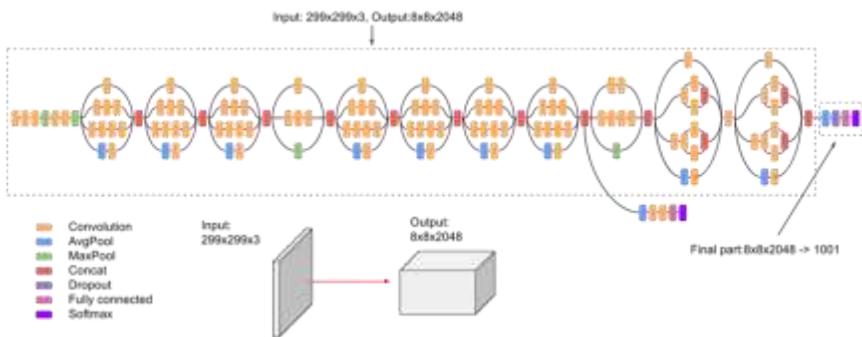


FIG 38. Arquitectura InceptionV3. (Google Cloud, 2022)

Resnet

Es una de las CNN que ahorran tiempo y ha ganado popularidad debido a su estructura novedosa. ResNet tiene bloques residuales que incorporan intersecciones en capas estándar de CNN para atravesar múltiples capas convolucionales, lo que hace que el proceso de convergencia de CNN sea más rápido y fácil a pesar de la gran cantidad de capas convolucionales (Attallah, 2022).

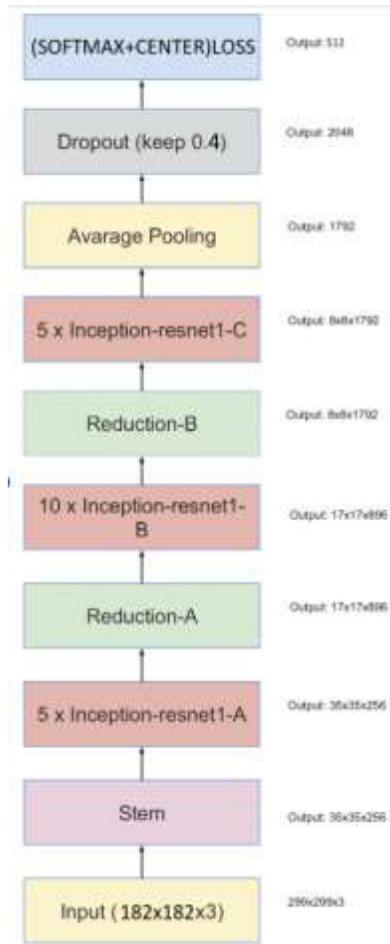


FIG 39. Arquitectura de ResNetV1

MobileNet

Es una arquitectura DL fina y eficiente en el tiempo que se diseñó originalmente. Puede disminuir la complejidad del modelo de entrenamiento al reducir la cantidad de parámetros mientras mantiene un rendimiento aceptable. Estas son capas convolucionales de dimensiones 3×3 y 1×1 , respectivamente. MobileNet tiene 53 capas profundas. (Attallah, 2022). Al ser puesto el algoritmo a prueba se pudo corroborar la efectividad de este modelo a un 79.4% de precisión (Howard et al., 2017).

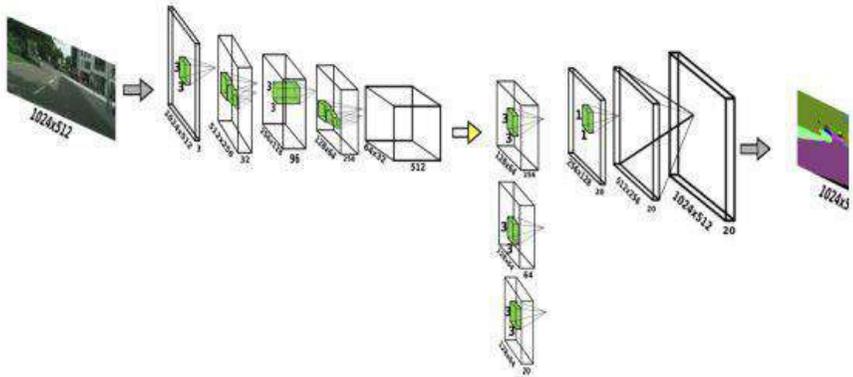


FIG 40. Arquitectura de MobileNetV2.

2.3. Los de datos (dataset)

Los sets de datos o dataset son un grupo de datos usados en entrenamiento fuera de línea se puede dividir en tres subconjuntos: el conjunto de entrenamiento, el conjunto de validación y el conjunto de prueba (Leivi, 2019). Esta distribución de datos permite realizar ajustes a la red neuronal, permitiéndonos entrenar el modelo de manera adecuada.

Un conjunto de datos en el aprendizaje automático es una colección de piezas de datos que pueden ser tratadas por un ordenador como una sola unidad con fines analíticos y de predicción. Esto significa que los datos recogidos deben ser uniformes y comprensibles para una máquina que no ve los datos de la misma manera que los humanos. Para ello, después de recoger los datos, es importante preprocesarlos limpiándolos y completándolos, así como anotarlos añadiendo etiquetas significativas legibles por un ordenador (Sydorenko, 2021).

2.3.1. Hiperparámetro

Los hiperparámetros son parámetros cuyos valores rigen el proceso de aprendizaje y definen los parámetros del modelo que finalmente obtiene un algoritmo de aprendizaje. Son valores de la configuración que se utilizan durante el entrenamiento y dicho

valor no se puede conocer a priori para un problema planteado por lo que, se tiene que utilizar valores genéricos y reglas estándares siendo una buena práctica mediante prueba y error.

2.3.2. Datos de entrenamiento

El set de entrenamiento o data training es usado para realizar el entrenamiento y optimizar los parámetros que son analizadas por las redes neuronales. El entrenamiento implica usar los errores generados para reutilizar el conjunto de datos para ajustar los pesos neuronales. (Leivi, 2019). Para realizar un entrenamiento correcto de nuestro algoritmo, los datos deben pasar por una etapa de depuración o limpieza.

Existen algunas páginas donde almacenan distintos tipos de datasets de entrenamiento para redes neuronales convolucionales orientadas a la clasificación de ECG, en este trabajo nos centraremos en la página de Mendeley Data (<https://data.mendeley.com/>), donde se encontraron imágenes previamente clasificadas por 4 diferentes tipos de electrocardiogramas. Para realizar la selección del dataset, se tuvieron en consideraciones los siguientes aspectos:

- Los exámenes de ECG no fueran tomados menos de 5 años atrás.
- Los datos proporcionados sean en formato Imagen para poder entrenar el modelo.
- Exista una clasificación de tipos de arritmias cardiacas para su debido análisis.

2.3.3. Set de validación

Los datos de validación o testing data, después de pasar la fase de entrenamiento, se ejecuta con el único propósito de adaptar los parámetros de la red. Se recalca que el algoritmo no aprende en esta etapa (Leivi, 2019).

La validación cruzada es una técnica de uso común en el aprendizaje automático para evaluar la diversidad de un conjunto

de datos y la confiabilidad de cualquier modelo que se esté entrenando. Se utilizar generalmente para la detección de sobreajustes en los modelos entrenados.

Esta técnica consiste en dividir nuestros datos al menos 4 grupos, de los cuales se utilizará una parte solo como validación. En la siguiente figura observaremos como es el proceso de iteración utilizando 5 iteraciones.

2.4. Resumen

La inteligencia artificial permite la interpretación de anomalías de la naturaleza a través de patrones y procesos inteligentes utilizando una máquina para resolver problemas que llevan de un análisis especial y profundo que sirven en diversas áreas como la medicina. Entonces, cuando se busca perfeccionar el comportamiento del proceso se usa la técnica del Machine Learning mediante aprendizaje supervisado, no supervisado, aprendizaje por refuerzo o transferencia de aprendizaje.

Los datos utilizados para la construcción de los modelos de Machine Learning dentro del campo de la salud médica, al tratarse de pacientes, se debe tomar en cuenta registros de comportamientos normales y aquellos que no califican como situaciones comunes para identificar patrones y que el proceso de inteligencia artificial pueda predecir trastornos de salud. Para dicho proceso, se debe tomar en cuenta reservar datos para validar el modelo entrenado.



CAPÍTULO 3:

RECOPIACIÓN DE ALGORITMOS DE MACHINE LEARNING ENFOCADOS A LA DETECCIÓN DE ARRITMIAS CARDIACAS.

3.1. Metodología de la investigación

La metodología empleada en el proyecto es de naturaleza descriptiva y también tiene un poco de la metodología bibliográfica ya que nos basamos en artículos o experiencias de otros investigadores que le dan validez a nuestro estudio. Aparte utilizamos el tema de Machine Learning porque mediante el procesamiento de cómputo de una computadora podríamos reducir o simular la experiencia de un experto.

La Metodología descriptiva analiza o estudia las características de una población o fenómeno sin entrar a conocer las relaciones entre ellas. En otras palabras, la metodología descriptiva lo que hace es definir, clasificar, dividir o resumir (Arias, 2021). Tiene como único objetivo caracterizar a la población estudiada. Sus características más destacadas son:

- Es considerado un proceso importante tanto en la investigación cuantitativa y cualitativa.
- Sus conceptos son parte de la estadística básica.
- Lo primero que se debe realizar es ordenar, resumir y dividir los datos.

Tipos de investigación descriptiva.

Método de observación: Es considerado como el más eficaz para llevar a cabo la metodología descriptiva. Se emplea en la observación cuantitativa y en la observación cualitativa. En nuestro proyecto.

Método de Estudio de Caso: Este método implica un análisis profundo y el estudio a individuos o grupos. Suele conducir a una hipótesis y amplía el alcance de investigación de un fenómeno.

La investigación o método descriptivo de investigación

El corazón tiene un sistema en específico de creación de estímulos cardíacos, además posee un sistema que conduce dichos estímulos eléctricos. En un corazón normal el marcapasos es una zona de la aurícula derecha llamada nódulo sinusal. El ritmo normal perteneciente a los latidos cardíacos, también llamado ritmo sinusal por su origen está entre las 60 y 80 contracciones por minuto. Cuando este ritmo normal se altera, se provocan las llamadas arritmias cardíacas, trastorno muy frecuente que puede suceder en presencia o ausencia de una enfermedad cardíaca. Para valorar una arritmia cardíaca se considera un conjunto de características que decidirán la gravedad de esta y son las siguientes:

1. Elementos o factores que pueden desencadenar la arritmia cardíaca como el ejercicio, ansiedad, etc.
2. La existencia de una regularidad o irregularidad de los latidos.
3. Frecuencia cardíaca ya sea rápida o lenta.
4. Características que presenten la circulación durante la arritmia.
5. Tolerancia del paciente dependiendo de los factores como: la duración de la crisis, la frecuencia cardíaca, la posible existencia de la cardiopatía y el tipo de taquicardia.

Comprobación de hipótesis

¿Los algoritmos de Machine Learning serán capaces de detectar las arritmias cardíacas y a su vez ayudar a las personas?

Investigación de diagnóstico

La arritmia Cardíaca puede provocar o causar una sensación de aleteo en el paciente, usualmente estos síntomas suelen ser inofensivos, mas, sin embargo, existen arritmias cardíacas que pueden provocar síntomas mucho más molestos y mortales. La diabetes, el consumo excesivo de alcohol, drogas, tabaco, insuficiencia cardíaca, estrés físico como por ejemplo exceso de trabajo de o ejercicio, son algunos de los factores que promueven a la aparición de la arritmia cardíaca, por ello es muy importante conocer que estilo de vida lleva el paciente.

Investigación causal

La investigación es causal, ya que nos permite conocer si un individuo o paciente tiene arritmia o no, ya que para llegar a este punto del resultado se tiene que pasar por algunos factores a evaluar entre ellos y el más importante son los pulsos eléctricos que coordinan los latidos del corazón, cuando no funcionan adecuadamente o de una forma defectuosa puede que el corazón lata muy rápido (taquicardia) o demasiado lento (bradicardia), en la taquicardia la frecuencia cardíaca en reposo es superior a 100 latidos por minuto, mientras que en la bradicardia la frecuencia cardíaca en reposo es inferior a 60 latidos por minutos con ambos resultados el individuo tendría un resultado de positivo a una arritmia cardíaca, mientras que si el valor de los pulsos eléctricos están entre 60 y 100 latidos por minutos daría negativo a la arritmia cardíaca.

Investigación cuasi experimental

En nuestro estudio ya tenemos un grupo formado por datos los cuales son públicos y obtenidos de la página Kaggle, el proyecto recién comienza por lo cual en la actualidad no se cuentan con los

relojes, mientras se están levantando datos reales, se van haciendo pruebas con los algoritmos que se van a utilizar más adelante desde el sitio web.

Investigación correlacional

En nuestro estudio se usarán algoritmos los cuales podrán ser evaluados entre sí para llegar a definir cuál de ellos es el mejor y el que nos muestre los mejores resultados y el mejor rendimiento en su respectiva función.

Criterios para optimización de los algoritmos

Al diseñar una red neuronal, es importante seguir una secuencia de pasos bien definida. Estos pasos podrían verse así:

- Decidir qué datos utilizar.
- Especificar variables de entrada.
- Dividir los datos en conjuntos de entrenamiento y prueba.
- Definir la arquitectura de la red.
- Elija un algoritmo de aprendizaje.
- Convertir variables a entrada de red.
- Aprendizaje (aprendizaje repetitivo hasta que el error de red cae por debajo del valor permitido)
- Prueba (verifica la generalidad de la red sobre la muestra retenida).

El diseñador de redes neuronales debe definir un conjunto de criterios de diseño óptimos de la siguiente manera:

- Variables de entrada relevantes (independientes).
- La mejor manera de aprender.
- Arquitectura adecuada: el número de capas ocultas depende del método de aprendizaje elegido, el número de métodos de procesamiento (nodos) en cada capa oculta.

3.2. Plataformas de Data Science

3.2.1. Que es Data Science

Data Science es el término que se le da a el efecto de extraer grandes cantidades de datos para detectar patrones que llevan a obtener información y predecir ciertos resultados. Por ello, permite identificar y controlar aspectos variables de una organización que influye en la competencia y en el mercado. Así mismo, se encarga de estudiar el origen de la información, la representación y las formas que existen para obtener los datos.

Entre las áreas que se benefician de la ciencia de datos están los enfocados en la salud, recursos humanos, educación y empresariales. Por ejemplo, al aplicarlo a la salud, permite obtener avances de los procesos ayudando a impulsar la productividad en el diagnóstico, tratamiento y mejora de las etapas en el trabajo con optimización como las siguientes:

- Facilitar el flujo de trabajo del sistema de salud.
- Reducir el riesgo de fracaso del tratamiento de un paciente.
- Proporcionar un tratamiento adecuado a tiempo.
- Evitar las urgencias innecesarias debidas a la falta de disponibilidad de médicos.
- Reducir el tiempo de espera de los pacientes en un centro de salud.

Dado que el descubrimiento de medicinas se considera una disciplina muy compleja, las distintas industrias farmacéuticas confían en gran medida en el impacto de la Data Science para resolver sus problemas y crear mejores medicamentos para las personas.

Pero, el descubrimiento de nuevas medicinas conlleva un gran gasto económico y la realización de numerosas pruebas. Por esta razón, es un proceso que requiere mucho tiempo. Es por ello que la Data Science, junto con los algoritmos de Machine Learning, están acelerando estos procesos. Además, está proporcionando una

perspectiva mayor para optimizar la tasa de éxito de las predicciones que se pueden hacer.

Las diferentes compañías farmacéuticas utilizan la información de los pacientes, como los perfiles de mutación y los metadatos. Esta información ayuda realmente a los investigadores a desarrollar modelos y encontrar relaciones estadísticas entre los múltiples atributos que se presentan.

Del mismo modo, las organizaciones pueden diseñar fármacos que aborden las mutaciones clave en las secuencias genéticas. Así, los algoritmos de Deep Learning pueden encontrar la probabilidad de desarrollo de enfermedades en el cuerpo humano.

Los algoritmos de la Data Science pueden ayudar a simular cómo actuarán los fármacos en el cuerpo humano que elimina las largas experimentaciones de laboratorio.

3.2.2. Kaggle

Es la comunidad de ciencia de datos en línea más grande del mundo, con más de 536 000 miembros activos en 194 países y más de 150 000 publicaciones por mes, que brinda todas las herramientas y recursos importantes para ayudarlo a lograr los mayores avances en ciencia de datos. Kaggle, como Data Scientist, tiene una interfaz personalizable y no configurable para cuaderno de Jupyter. Permite el acceso gratuito a la GPU, así como a una gran cantidad de datos y códigos publicados por la comunidad. En Kaggle encontrarás el código y los datos que necesitas para realizar proyectos de análisis de datos.

La competencia y el desafío son la columna vertebral de la plataforma. Los usuarios se desafían por medio de las clasificaciones como punto de referencia para medir el desempeño y usándolas como motivación. Si bien se fomenta la competitividad, la comunidad siempre está interesada y las preguntas son bienvenidas con el objetivo de fomentar la innovación en la ciencia de datos. Hay 4 tipos de concursos:

- **Patrocinadores:** Personas que tienen empresas, asociaciones e incluso organismos gubernamentales como patrocinadores.
- **Investigación:** Esto está orientado a la investigación.
- **Reclutamiento:** Son los concursos más escasos y se premian con ofertas de trabajo de las empresas que los patrocinan.
- **Principiantes:** Estos concursos no tienen premios, son para nuevos usuarios con temas sencillos y datos fáciles de entender. Puedes unirte en cualquier momento.

Banco de datos de arritmias del MIT-BIH

Esta incluye 48 extractos de registros de ECG ambulatorios de dos canales de 47 pacientes examinados por el Laboratorio de Arritmias del BIH. Veintitrés grabaciones se eligieron al azar de un conjunto de 4.000 grabaciones de ECG ambulatorias de 24 horas recogidas de una población mixta de pacientes internos y externos en el Hospital Beth Israel de Boston (Moody & Mark, 2001).

Estándar AAMI

La AAMI (2021) agrupan a las arritmias cardíacas en cinco superclases: normal (N), latido ectópico supraventricular (S), latido ectópico ventricular (V), latido de fusión (F) y latido desconocido (Q).

Clasificación de latidos según el AAMI	Tipos de latidos del MIT-BIH				
Latido normal (N)	Latido normal (N)	Bloqueo de rama izquierda (L)	Bloqueo de rama derecha (R)	Latido de escape auricular (e)	Ritmo de escape nodal (j)
Latido ectópico supraventricular (S)	Latido prematuro auricular (A)	Latido prematuro aberrado (a)	Latido prematuro nodal (J)	Latido prematuro supraventricular (S)	
Latido ectópico ventricular (V)	Contracción ventricular prematura (V)	Latido de escape ventricular (E)			
Latido de fusión (F)	Fusión de latidos ventriculares y normales (F)				
Latido desconocido (Q)	Latido estimulado (I)	Fusión de latido normal y estimulado (f)	Latido no clasificado (Q)		

FIG 41. Descripción de la clase de ECG mediante la norma AAMI.

Latido normal (N)

Una categoría de ritmo normal genera los latidos del corazón desde el nodo sinoauricular situado en la aurícula derecha. Esta categoría incluye varios tipos de latidos como el bloqueo de rama izquierda/derecha y latidos de escape y nodales auriculares (AAMI, 2021).

Latidos ectópicos supraventriculares (S)

Esta esta categoría corresponde a los latidos prematuros tanto auriculares como supraventriculares, y sus variantes incluyen latido prematuro auricular, latido prematuro aberrado, latido prematuro nodal y latido prematuro supraventricular. El latido auricular prematuro es un latido cardíaco adicional que se origina en un sitio anormal en las aurículas justo antes de que ocurra un latido cardíaco normal. Son inofensivos y, a menudo, se perciben como un latido interrumpido o una sacudida en el pecho (AAMI, 2021).

Latidos ectópicos ventriculares (V)

Esta categoría incluye la contracción ventricular prematura y los latidos de escape ventricular. La contracción ventricular prematura es un latido cardíaco adicional que se origina en los ventrículos, lo que resulta en una contracción caótica de los ventrículos. Esta arritmia puede ser causada por estrés físico o emocional y exceso de cafeína en el torrente sanguíneo. Los latidos de escape ventriculares son el resultado de una descarga eléctrica autogenerada que generalmente ocurre después de una larga pausa en el ventrículo (AAMI, 2021).

Latidos de fusión (F)

Estos pueden ocurrir cuando las aurículas o los ventrículos son activados por diferentes fuentes al mismo tiempo, por ejemplo, desde el nódulo sinoatrial del corazón y el latido ectópico auricular o desde un marcapasos externo (AAMI, 2021).

Latidos sin clasificar (Q)

Son latidos atípicos que deben eliminarse de los datos de entrenamiento. Los ejemplos de latidos cardíacos no clasificados incluyen artefactos, malas conexiones de electrodos o latidos cardíacos originados por marcapasos (AAMI, 2021).

3.2.3. Mendeley Data

Es la comunidad de ciencia de datos en línea, brinda todas las herramientas y recursos, así como Kaggle, como Data Scientist, tiene una interfaz personalizable y no configurable para Google Colaboratory. Permite el acceso gratuito a la GPU, así como grandes grupos de datos y códigos publicados por la comunidad. En TensorFlow Hub podemos encontrar el código y los datos necesarios.

BÚSQUEDA SISTEMÁTICA PARA LA APORTACIÓN DE ESTE TRABAJO.

La búsqueda se organiza en tres niveles: ECVIA (L1), CNN (L2) y la aplicación de cuatro Protocolos de Revisión (L3). Existe un nivel intermedio en el cual se aplica la unión de los resultados del L1 y L2, obteniendo en la intersección de estos dos grandes conjuntos, los artículos que refieren a ECVIA y CNN (n=).

L1	Ecvia	Enfermedades Cardiovasculares	(ecv ((cardiovasculares* o arritmia o aceleración o arterias o angina o arteoclorosis o cardiomegalia o cardiopatía o cardiomiopatía o prolapso) w/2 (padecimiento* o pericardio)) w/2 pericardite)	45,565
		Inteligencia artificial	(IA O AI O ((artificial* O robot O machine O simulato O agente O predicción* O algoritmo O sistema O sensores O computarizado) w/2 (inteligencia* O inteligente* O habilidad O intelectual O percepción)) O ((machine O profundo) w/2 minería)	1,089,667
L2	Cnn	Redes neuronales convolucionales	(cnn o ((neuronal* o redes o simulador o pseudoneuronas o tecnología o cerebro o capas o reconocimiento o pixeles) w/2 (redes o neuronales o convolucionales o patrones) w/2 multicapa)	19,019

[L1 y L2]	[iaecv y cnn]			166
L3		Protocolos		
		Año	2018-2022	
		Área temática	Biomedicina, Tecnología y Salud	90
		Tipo de documento	Artículos	50
		Lenguaje	Español	80
	Criterio de calidad	Revisión de Experto en lectura detallada de cada artículo y su relación con el área de investigación		10

FIG 42. Estructura Semántica de Búsqueda

3.3. Redes neuronales

Redes neuronales convolucionales

Las (RNC) o CNN en sus siglas en ingles Estos se refieren específicamente a datos de una topología de cuadrícula conocida, los time-series data, se consideran una cuadrícula unidimensional, la cual toma muestras a intervalos de tiempo regulares y los image data se consideran una cuadrícula bidimensional de píxeles (Jalomo, 2019). Este tipo de red realiza una operación de convolución a la señal de entrada respecto a una propagación que se encuentra en la salida de la neuronal

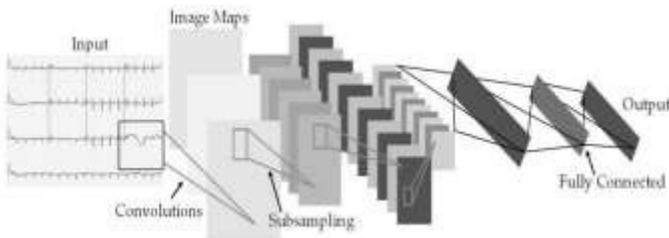


FIG 43. Arquitectura de búsqueda de patrones de una red neurona convolucional. (Rahman et al., 2022)

Arquitectura

Las distintas capas que componente a una RNC son la capa convolucional, capa de activación, capa de agrupación o Pooling y capa de salida.

Capa convolucional

La operación de convolución toma la imagen clasificada como entrada. Luego le aplica un filtro conocido como kernel para devolver un mapa funcional de la imagen real como salida. De esta manera puede disminuir el tamaño de los factores y disminuir la complejidad de la imagen (Leivi, 2019). En definitiva, esta capa permite extraer rasgos de cada imagen de entrada como vemos en la (figura 14), el kernel realiza una interpretación de ciertas partes de la imagen, cambiándole la intensidad del brillo o invirtiendo el color.

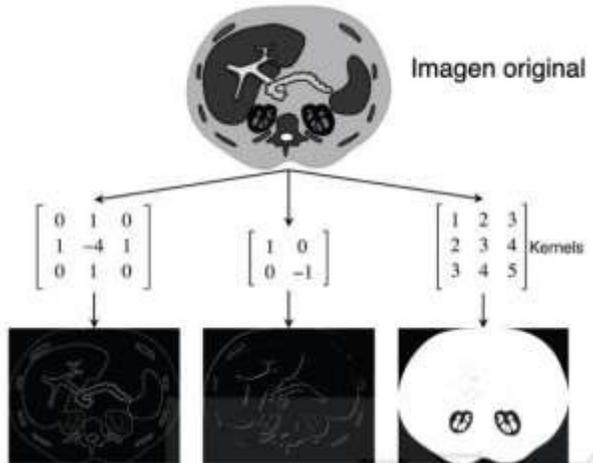


FIG 44. Aplicación de convolución en imagen. (Leivi, 2019)

Capa de activación

Incluye una función de activación encargada de proporcionar datos de salida a partir de datos de entrada. La función más utilizada en la literatura se llama ReLU. La razón particular para usar esta característica es que los valores negativos son inútiles en el procesamiento de imágenes, y si se usa un tipo diferente, la entrada cambiará a un rango muy estrecho (SAN MARTÍN, 2020). La función de activación devuelve una salida la cual ha sido generada por una o un grupo de entradas.

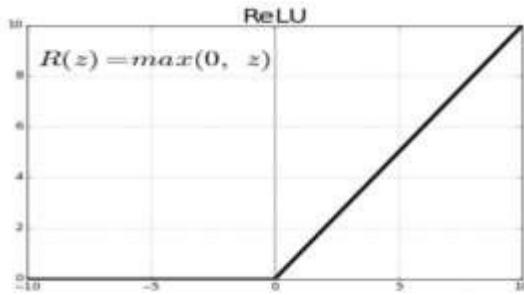


FIG 45. Función de activación Relu (SAN MARTÍN, 2020)

Capa de agrupación

Su propósito es reducir la dimensión espacial (ancho y alto) del volumen de datos de entrada para proporcionarlo como entrada para la siguiente capa de convolución. Esta capa también se conoce como downampling porque la disminución del tamaño permite el extravío de datos (Levi, 2019). Uno de los beneficios que ofrece esta capa es la reducción del tamaño de la imagen como podemos observar en la FIG 11.

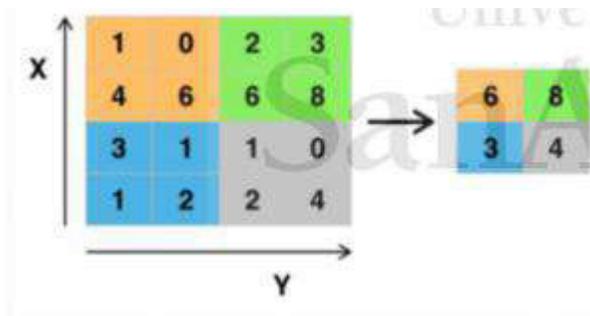


FIG 46. Capa de Agrupación (Levi, 2019)

Capa de salida

Es en la cual ingresa todo el volumen de datos, clasificando los datos de entrada. El número de neuronas de salida es igual al número de clases en las que se pueden clasificar los datos de entrada (San Martín, 2020). Esta capa permite observar el porcentaje de éxito de clasificación que ha logrado el algoritmo.

Un enfoque común para que la IA estudie datos es extraer características de procesos tradicionales y usarlas para entrenar clasificadores de aprendizaje automático. Muchos estudios utilizan el método convencional para realizar este tipo de predicciones, sin embargo se requiere de un arduo trabajo para tener un equilibrio, precisión y carga computacional para evitar caer en errores (Attallah, 2022). En la actualidad en Deep Learning, se han implementado modelos de transferencia de aprendizaje los cuales evitan alguno de los errores que se daban en una red neuronal convencional.

Métricas para verificar el modelo

Exactitud (Accuracy)

La exactitud es la medida de rendimiento más intuitiva y no es más que una relación entre la observación correctamente predicha y el total de observaciones. Se puede pensar que, si se tiene una alta exactitud, el modelo es el mejor, aunque no siempre es lo correcto, ya que, la precisión puede ser una gran medida, pero sólo cuando sus datos son simétricos. Por lo tanto, hay que tener en cuenta otros parámetros para evaluar el rendimiento del modelo (Gad, 2021).

Precisión (Precision)

La precisión se define como la relación entre las observaciones positivas previstas con exactitud y el número total de observaciones positivas previstas correctamente (Brownlee, 2020).

Exhaustividad (Recall)

La relación entre las observaciones positivas predichas con precisión y la totalidad de las observaciones de la clase real se define como exhaustividad (Brownlee, 2020).

F1

La puntuación F1 se basa en la media ponderada de Precisión y Exhaustividad. Intuitivamente no es tan fácil de entender como la

exactitud, pero F1 suele ser más útil que la exactitud, especialmente si se tiene una distribución de clases desigual (Korstanje, 2022).

Validación cruzada

La validación cruzada es un enfoque estadístico para evaluar la capacidad de los modelos de aprendizaje automático. Se emplea a menudo en el aprendizaje automático aplicado para contrastar y escoger un modelo para un determinado problema de modelado predictivo, ya que es fácil de entender, sencillo de aplicar y produce estimaciones de habilidad con un sesgo menor que otros enfoques (Rodrigo, 2019).

Los datos suelen estratificarse antes de ser divididos en k pliegues. La estratificación es el proceso de reorganización de los datos para garantizar que cada pliegue sea un buen representante del del conjunto. Por ejemplo, en un problema de clasificación binaria en el que cada clase comprende el 50% de los datos, es de los datos, lo mejor es ordenar los datos de forma que en cada pliegue, cada clase comprende alrededor de la mitad de las instancias (Elizalde et al., 2009).

Técnica para balancear datos SMOTE

La clasificación desequilibrada se utiliza para anticipar el crecimiento de un modelo de clasificación basado en un conjunto de datos con mucho desequilibrio. Si no se aborda el problema, el rendimiento de predicción del modelo se verá afectado. El método SMOTE se puede utilizar en pequeñas muestras para generar muchas muestras sintéticas. La recomendación en el documento es usar bajas aleatorias para eliminar un cierto número de muestras en la clase grande, luego, luego usar SMOTE (Brownlee, 2021).

Curva ROC

Una curva ROC se crea graficando un recuento de verdaderos positivos contra un recuento de falsos positivos, donde el recuento de verdaderos positivos es la proporción de observaciones positivas que fueron predichas correctamente ($TP/(TP + FN)$). Del

mismo modo, el de falsos positivos es el porcentaje de observaciones que se predijeron incorrectamente como positivas entre todas las observaciones negativas ($FP / (TN + FP)$) (Cerdea & Cifuentes, 2012).

Por ejemplo, en las pruebas médicas, la tasa de verdaderos positivos es la tasa en la que se identifica correctamente a las personas que dan positivo a la enfermedad en cuestión (Cerdea & Cifuentes, 2012).

Matriz de confusión

Es una forma de describir el rendimiento de un algoritmo de clasificación. Cuando se tiene un número desequilibrado del cumplimiento en una clase o más de 2 clases en la colección de datos, la precisión sola puede ser engañosa. La matriz puede ayudar a entender lo que un modelo de clasificación está obteniendo correctamente y verificar qué tipo de errores se están cometiendo (Arce, 2021).

Validación

Validación cruzada

La validación cruzada es un enfoque estadístico para evaluar la capacidad de los modelos de aprendizaje automático. Se emplea a menudo en el aprendizaje automático aplicado para contrastar y escoger un modelo para un determinado problema de modelado predictivo, ya que es fácil de entender, sencillo de aplicar y produce estimaciones de habilidad con un sesgo menor que otros enfoques (Rodrigo, 2019).

Los datos suelen estratificarse antes de ser divididos en k pliegues. La estratificación es el proceso de reorganización de los datos para garantizar que cada pliegue sea un buen representante del del conjunto. Por ejemplo, en un problema de clasificación binaria en el que cada clase comprende el 50% de los datos, es de los datos, lo mejor es ordenar los datos de forma que en cada pliegue, cada clase comprende alrededor de la mitad de las instancias (Elizalde et al., 2009).

3.4. Algoritmo detección de arritmias usando ECG

Nuestro primer algoritmo nos indica que mediante un electrocardiograma previamente cargado de la base de datos puede detectar si el paciente sufre o esta propenso a sufrir una arritmia para poner a prueba nuestro algoritmo procedemos a importar algunas librerías correspondientes para utilizar en nuestro primer algoritmo como lo es numpy, panda y maplolib que son las librerías más comunes para este tipo de proyectos.

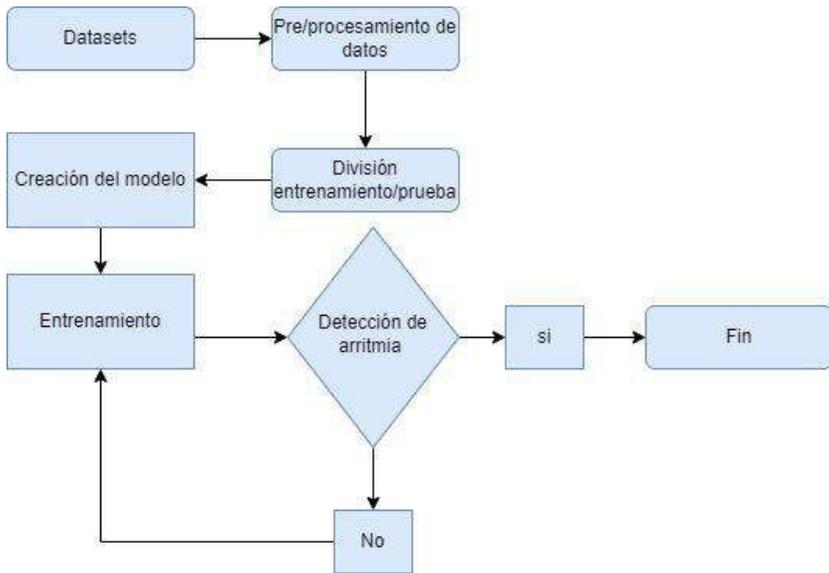


FIG 47. Diagrama de flujo del Algoritmo detección de arritmias usando ECG. Elaborado por Steven Aristega y Carlos León

Nuestro primer algoritmo nos indica que mediante un electrocardiograma previamente cargado de la base de datos puede detectar si el paciente sufre o esta propenso a sufrir una arritmia para poner a prueba nuestro algoritmo procedemos a importar algunas librerías correspondientes para utilizar en nuestro primer algoritmo como lo es numpy, panda y maplolib que son las librerías más comunes para este tipo de proyectos.

3.5. Algoritmo categorización de subtipos de arritmia del ECG

Para el siguiente algoritmo comenzaremos importando algunas librerías el cual nos ayudara para hacer la clasificación de los subtipos y luego categorizarla de acuerdo con el modelo establecido.

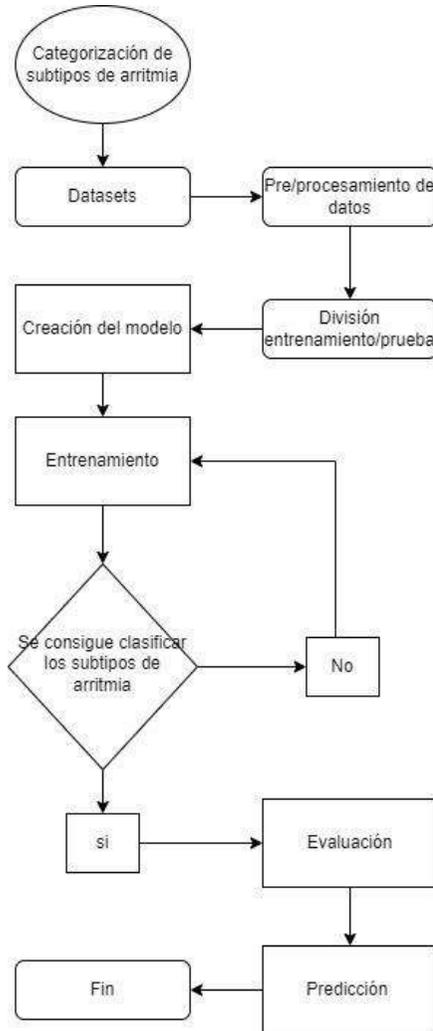


FIG 48. Diagrama de flujo del algoritmo categorización de subtipos de arritmia del ECG

3.6. Algoritmo ResNet

La idea en sí de ResNet, se basa en aumentar el número de capas introduciendo una conexión residual (con una capa de identidad). De hecho, aparecen problemas con el desvanecimiento del gradiente (vanishing gradiente) y de la dimensionalidad (curse of dimensionality) y la red deja de aprender. Con la profundidad de las capas la precisión llega a un momento en el que se estanca y comienza en si a degradarse, con lo que se puede llegar a deducir que las redes convolucionales menos profundas aprenden mejor que las más profundas. Y es aquí donde nace la idea de los bloques residuales, ya que lo que hacen es saltar unas cuantas capas. Aumentan el número de capas introduciendo una conexión residual (con una capa identidad) y con ello se mejora el proceso de aprendizaje, ya que esta capa pasa directamente a la siguiente.

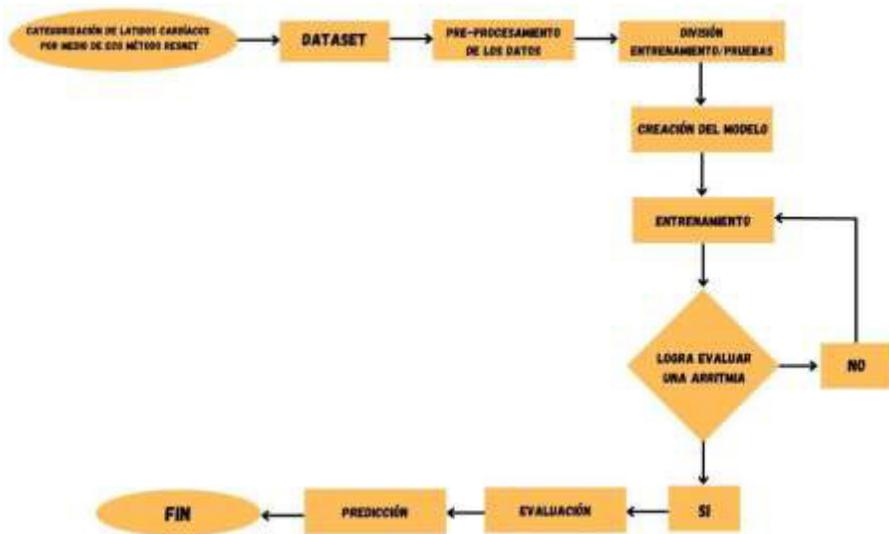


FIG 49. Flujograma del algoritmo ResNet

3.7. Resumen

En el capítulo se detalla la metodología utilizada fue la descriptiva que permitió definir, clasificar, dividir o resumir en donde adicionalmente, se hace uso de la metodología bibliográfica

porque la investigación está basada en artículos o experiencias de otros investigadores referente al corazón y el sistema para la creación de estímulos eléctricos en donde se identifica el ritmo normal que está comprendido entre las 60 y 80 contracciones por minuto que permite seleccionar los algoritmos Machine Learning enfocados a la detección de arritmias cardiacas.

Adicionalmente, se hace la explicación de las plataformas para Data Science que se toman en cuenta en investigaciones sobre datos y recursos de inteligencia artificial como Kaggle y Mendeley Data que permiten, a través de una comunidad, contar con opciones para la detección de arritmias. Se detalla el diagrama de flujo correspondiente a cada uno de los algoritmos seleccionados que se abordarán en la siguiente sección.



CAPÍTULO 4:

Optimización de algoritmos de Machine Learning para la detección de arritmias cardiacas.

4.1. Configuración de entorno para optimización de algoritmos

Google Colab

Para realizar el entrenamiento y ajuste del algoritmo convolucional se optó por utilizar Google Colab debido a los recursos que brinda Google para ejecutar algoritmos de manera rápida. Para acceder a Google colab, nos dirigimos a la página.

<https://colab.research.google.com>.



FIG 50. Cuaderno de Google Colab

Tal y como se observa en la figura 18, se crea un cuaderno en Google Colab donde se irán instalando las librerías necesarias para

ejecutar los algoritmos de Machine Learning y proceder con el entrenamiento y mejoramiento. Se cargan los datos, para cargar los datos al modelo, se recomienda almacenar los datos comprimidos en un zip en Google drive, es más práctico a la hora de realizar el entrenamiento en Google Colaboratory, debido a que si cerramos la pantalla los datos se eliminan y se tendrían que cargar los datos nuevamente, haciendo el proceso de carga muy lento.

Algunas de las características más interesantes se enumeran a continuación:

- Escritura y ejecución de código Python sin tener una configuración local.
- Ejecución de comandos de terminal desde el portátil.
- Importes conjuntos de datos de fuentes externas como Kaggle.
- Almacenamiento y gestión desde Google Drive.
- Servicio gratuito en la nube, GPU y TPU.
- Importar o publicar directamente desde/hacia GitHub.

Google Drive

Es un servicio gratis de almacenamiento de archivos en línea que consiente a los usuarios guardar y acceder a archivos. El sistema se encarga de sincronizar documentos, imágenes y otros datos en los dispositivos de los usuarios, incluidos los teléfonos móviles, las tabletas y las computadoras (Mixon & Wigmore, 2018).

Lenguaje de programación Python

Python admite una gran variedad de paradigmas de programación, como el orientado a objetos, el procedimental y el reactivo. Incorpora un gestor de tipos dinámico, memoria principal automatizada y una amplia y variada biblioteca estándar (Lubanovic, 2019).

Python es un lenguaje de programación sencillo de leer y escribir debido a su gran parecido con el lenguaje humano, además de que es código abierto (Open Source). Python no solo permite desarrollar

aplicaciones, sino también facilita trabajar con inteligencia artificial, Big data, machine Learning y data science. Tomado de (Santander-Becas, 2021).

Jupyter Notebook

Es una herramienta de software gratuito en la nube que permite a los usuarios crear y compartir documentos con código en directo, ecuaciones, gráficos y texto. Los cuadernos Jupyter son un proyecto derivado del proyecto IPython, que a su vez tenía un proyecto de cuadernos IPython. El nombre, Jupyter, viene de los lenguajes de programación principales que soporta: Julia, Python y R (Driscoll, 2021).

Origen

El origen de Python remota a los años 90, cuando el señor Guido Van Rossum, un trabajador del Centrum Wiskunde & Informatica, perteneciente al centro de investigación holandés, tuvo la idea de desarrollar un nuevo lenguaje apoyándose de un proyecto anterior llamado Lenguaje de programación "ABC". Desde el primer momento tuvo la ideología de crear un lenguaje sencillo fácil de leer y escribir, sin que eso detuviera su potencial para desarrollar todo tipo de aplicaciones.

Versiones de Python

Guido Van Rossum empezó el desarrollo del lenguaje de programación en el año 1989 y empezando su implementación en febrero de 1991, fue en ese momento que se publicó la primera versión pública la 0.9.0. Hasta el año 2018 el desarrollo de las versiones de Python estaba bajo el mando de Guido Van Rossum. En julio de ese mismo año Guido Van Rossum anuncio su retiro de seguir dirigiendo el desarrollo de las versiones de Python, a partir del año 2019 el desarrollo de Python quedo en manos de un consejo que está conformado por 5 miembros y que se renuevan anualmente (Marco, 2022).

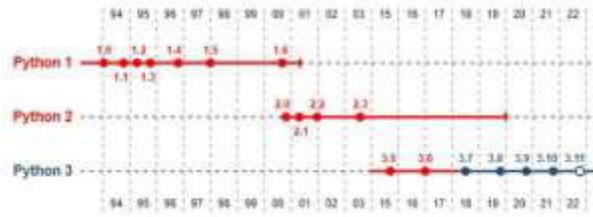


FIG 51. En la imagen se puede observar las versiones principales de Python con su respectiva publicación. (Marco, 2022).

En la figura anterior se observa las versiones de Python con su respectiva fecha de publicación. Así mismo las versiones de color rojo significan que ya están obsoletas, las versiones con el punto blanco pertenecen a futuras versiones con fecha ya predicha.

Entre las principales librerías para la elaboración y entrenamiento de los modelos de Machine Learning tenemos:

NumPy

NumPy conforma una biblioteca de Python para realizar cálculos científicos. El paquete se utiliza principalmente para realizar operaciones matemáticas y estructurar cualquier dato dado de forma que permita a los desarrolladores trabajar con él de forma más eficiente. NumPy presenta capacidades de generación de números aleatorios, así como herramientas de álgebra lineal, la transformada de Fourier y un amplio conjunto de herramientas de manipulación de matrices multidimensionales (Ramonyai, 2022).

Pandas

Mientras que NumPy se puede utilizar para ordenar y manipular datos, pandas es uno de los paquetes de Python de aprendizaje automático que se utiliza para analizarlos. Es particularmente útil en el desarrollo de una aplicación de aprendizaje automático, debido a que la calidad de la entrada inicial es tan importante Pandas es conocido por su rápido y eficiente objeto DataFrame, las capacidades de escritura y lectura de datos en diferentes formatos, incluyendo, bases de datos SQL, Microsoft Excel, y estructuras de datos en memoria (Ramonyai, 2022).

Scikit-learn

Scikit-learn es una biblioteca que contiene los principales algoritmos de aprendizaje automático de Python que utilizan los programadores. Permite a los desarrolladores utilizar algoritmos de regresión, clasificación y clustering para realizar la operación o predicción requerida en unas pocas líneas de código. La biblioteca cuenta con algoritmos de ML Python como k-means, bosques aleatorios, gradient boosting, máquinas de vectores de soporte, regresión lineal y varios otros (Ramonyai, 2022).

Keras

Keras es una API que se creó para permitir a los desarrolladores experimentar con redes neuronales e implementar las ideas de los programadores más rápidamente. Tiene facilidad de uso y mejora el rendimiento de los desarrolladores simplificando el proceso de escritura del código. La API admite redes convolucionales y redes recurrentes, se ejecuta tanto en la CPU como en la GPU y, lo que es más importante, permite la creación rápida de prototipos (Ramonyai, 2022).

TensorFlow

TensorFlow no es sólo una librería, sino toda una plataforma compuesta por paquetes de Python para el aprendizaje automático. TensorFlow reúne todos los recursos necesarios para construir una aplicación de ML de forma rápida y eficiente. El conjunto de herramientas es frecuentemente comparado con Keras y scikit-learn, aunque su funcionalidad objetivo es ligeramente diferente a la de ambos. TensorFlow funciona mejor cuando se aplica a la investigación de redes neuronales profundas y al aprendizaje automático (Ramonyai, 2022).

Matplotlib

Matplotlib es una de las bibliotecas de Python de aprendizaje automático para la visualización de datos. Puede utilizarse para crear gráficos estáticos, dinámicos e interactivos, diagramas y otras

formas de representación visual de datos. En general, Matplotlib es altamente personalizable. En el ámbito científico, la investigación y el procesamiento de datos con Python no son posibles sin la visualización de los mismos. Por lo tanto, las empresas pueden utilizarlo para presentaciones y para hacer legibles los resultados de los algoritmos de aprendizaje automático (Ramonyai, 2022).

Seaborn

Seaborn es otra biblioteca de visualización de datos. Está basada en Matplotlib, y presenta una versión mejorada de esta última. Seaborn tiene más temas por defecto y puede aplicarse para un rango más amplio de objetivos que Matplotlib.

OpenCV

OpenCV es una biblioteca que contiene aproximadamente 2500 algoritmos optimizados para la programación de IA en Python, la visión por ordenador y otros tipos de aprendizaje automático. Se distribuye bajo licencia BSD, lo que significa que las empresas casi no tienen restricciones para utilizarla. Las empresas pueden utilizar OpenCV para detectar e identificar rostros y objetos, seguir los movimientos de la cámara, clasificar acciones en los vídeos, mejorar la resolución de una imagen y encontrar imágenes similares de cualquier base de datos. Este paquete de aprendizaje automático de Python es increíblemente útil en la creación de aplicaciones de realidad aumentada (Ramonyai, 2022).

spaCy

Procesa lenguaje natural, es un paquete de IA y aprendizaje automático escrito en Python. Fue creado con la intención de ser utilizado en artículos comerciales. Cuenta con modelos estadísticos pre-entrenados, así como el apoyo a más de cincuenta idiomas, redes neuronales convolucionales, velocidad admirable, y la integración de aprendizaje profundo (Ramonyai, 2022).

SciPy

SciPy es un ecosistema basado en diversas herramientas de Python para el cálculo científico y el aprendizaje automático. Uno

de sus elementos esenciales es la biblioteca SciPy que contiene módulos y algoritmos para álgebra lineal, estadística, optimización, procesamiento de imágenes e interpolación. La pila SciPy contiene el paquete NumPy, así como Matplotlib, pandas y SymPy, además de muchos otros (Ramonyai, 2022).

4.2. Evaluación de los algoritmos

El método que se utilizará en este proyecto es la matriz de validación cruzada, realizando la separación de datos en grupos evaluándolos y validando los resultados. Para ello, se accederá por medio de la página de Google Colaboratory a la validación de los resultados, debido que su funcionalidad es realmente rápida a la hora de efectuar los distintos procedimientos a través de los algoritmos.

Los datos suelen estratificarse antes de ser divididos en k pliegues. La estratificación es el proceso de reorganización de los datos para garantizar que cada pliegue sea un buen representante del del conjunto. Por ejemplo, en un problema de clasificación binaria en el que cada clase comprende el 50% de los datos, es de los datos, lo mejor es ordenar los datos de forma que en cada pliegue, cada clase comprende alrededor de la mitad de las instancias (Elizalde et al., 2009).

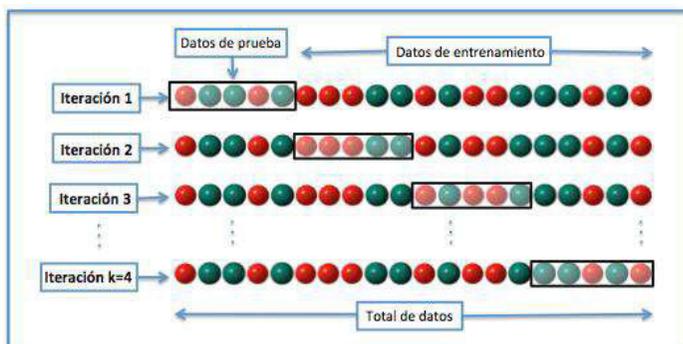


FIG 52. Se detalla la representación de las iteraciones con sus datos de prueba y entrenamiento. (Kucheryavskiy et al., 2020)

4.3. Algoritmo detección de arritmias usando ECG

```
import numpy as np # linear algebra
import pandas as pd
from sklearn.preprocessing import StandardScaler #to normalize the data preprocessing

import warnings
import matplotlib.pyplot as plt
import cv2
warnings.filterwarnings('ignore') #to suppress the warnings

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

/kaggle/input/heartbeat/ptbdb_abnormal.csv
/kaggle/input/heartbeat/ptbdb_normal.csv
/kaggle/input/heartbeat/mitbih_test.csv
/kaggle/input/heartbeat/mitbih_train.csv
```

FIG 53. FIG 53. Importación de Librerías

Procederemos a hacer la importación de los datasets el cual están ubicados dentro del mismo repositorio de kaggle para fines investigativos ya que en la Universidad de Guayaquil aún no cuenta con esa información.

```
mit_test = pd.read_csv('/kaggle/input/heartbeat/mitbih_test.csv', header=None)
mit_train = pd.read_csv('/kaggle/input/heartbeat/mitbih_train.csv', header=None)
ptb_abnormal = pd.read_csv('/kaggle/input/heartbeat/ptbdb_abnormal.csv', header=None)
ptb_normal = pd.read_csv('/kaggle/input/heartbeat/ptbdb_normal.csv', header=None)
```

FIG 54. FIG 54. Importación de los Datasets.

Se procede a crear los diccionarios de datos el cual nos sirve para guardar una serie de mapeos entre dos conjuntos de datos para el entrenamiento.

```
mit_test.rename(columns={187:"Class"}, inplace=True)
mit_train.rename(columns={187:"Class"}, inplace=True)
ptb_abnormal.rename(columns={187:"Class"}, inplace=True)|
ptb_normal.rename(columns={187:"Class"}, inplace=True)
```

FIG 55. Conjunto de Datos

Procedemos a leer los datos y separarlos y contarlos por clases del cual el MIT tiene 5 clases y el PTB 2 clases el cual aquí nos muestra la cantidad de datos que hay cada clase.

```
print ("MIT Datos para entrenamiento : \n ", mit_train["Class"].value_counts())  
print ("\n MIT Datos de prueba: \n", mit_test["Class"].value_counts())
```

```
MIT Datos para entrenamiento :  
0.0    72471  
4.0     6431  
2.0     5788  
1.0     2223  
3.0      641  
Name: Class, dtype: int64  
  
MIT Datos de prueba:  
0.0    18118  
4.0    1608  
2.0    1448  
1.0     556  
3.0     162  
Name: Class, dtype: int64
```

FIG 56. Datos del MIT Instituto de Tecnología de Massachusetts

```
print ("\nPTB Abnormal classes: \n", ptb_abnormal["Class"].value_counts())  
print ("\nPTB Normal classes: \n", ptb_normal["Class"].value_counts())
```

```
PTB A normal classes:  
1.0    10506  
Name: Class, dtype: int64  
  
PTB Normal classes:  
0.0    4046  
Name: Class, dtype: int64
```

FIG 57. Datos del PTB (PHYSIKALISCH-TECHNISCHE BUNDESANSTALT)

Procedemos a darle nombre a cada clase del respectivo datasets como lo es MIT el cual sería: 'Latido normal', 'Latido prematuro supraventricular', 'Contracción ventricular prematura', 'Fusión de latidos ventriculares y normales', 'Ritmo no clasificable' y del PTB que puede ser de ritmo normal o ritmo anormal.

```
MIT_Outcome = {0. : 'Latido normal',  
               1. : 'Latido prematuro supraventricular',  
               2. : 'Contracción ventricular prematura',  
               3. : 'Fusión de latidos ventriculares y normales',  
               4. : 'Ritmo no clasificable'}  
  
PTB_Outcome = {0. : 'Normal',  
               1. : 'A normal'}
```

FIG 58. Clasificación del tipo de latido del corazón

Mediante la siguiente función generamos gráficos de algunas muestras que están dentro del dataset del MIT el cual nos muestra algunos ejemplos de electrocardiogramas que tiene un latido normal o latido prematuro depende cual sea el caso.

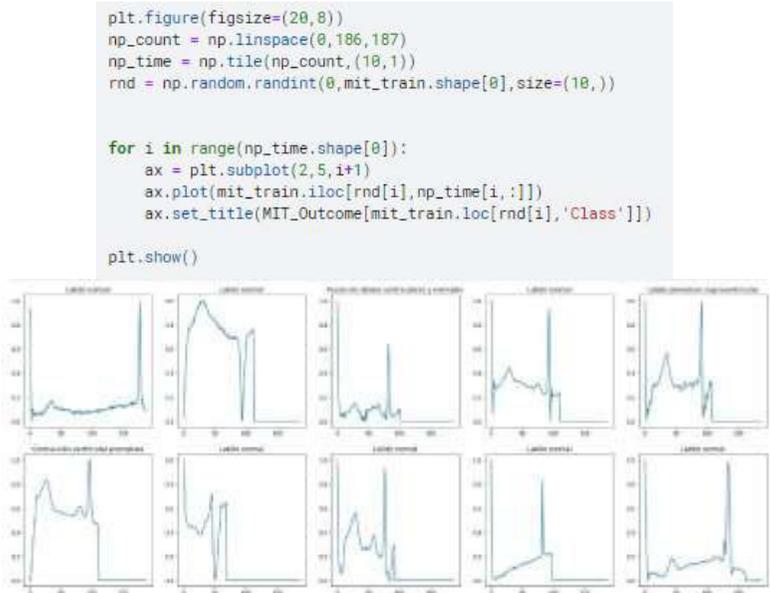


FIG 59. Función que nos permite mostrar del conjunto de datos de entrenamiento del MIT con su respectiva clasificación

Procedemos a importar librerías para crear nuestro modelo de machine Learning a través de la librería de sklearn, también utilizamos tensorflow para crear y entrenar las redes neuronales y keras el cual nos ayuda a entrenar nuestro modelo de clasificación de imágenes.

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import normalize
from sklearn.metrics import classification_report, plot_confusion_matrix, confusion_matrix

import tensorflow as tf
import tensorflow.keras as tfk
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, Flatten, Dropout, InputLayer, LSTM, GRU, BatchNormalization, Bidirectional, Conv2D
from tensorflow.keras.models import Model
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensorflow.keras.preprocessing import sequence
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.utils import to_categorical
```

FIG 60. Importamos librerías y paquetes adicionales para su correcto funcionamiento.

Preparación de los conjuntos de entrenamiento, validación y prueba para el conjunto de datos PTB.

```
ptb_full = pd.concat([ptb_normal, ptb_abnormal], axis=0, sort_index=True)
ptb_full.index.names = ['time', 'patient_id']
ptb_full = ptb_full.sample(frac=1, random_state=42)
train_ptb, test_ptb, val_train_ptb, val_test_ptb = train_test_split(ptb_full.iloc[:, :187], ptb_full.iloc[:, 187], test_size=0.3, random_state=42)
train_ptb, val_train_ptb, val_test_ptb, test_ptb = train_test_split(train_ptb, val_train_ptb, test_size=0.2, random_state=42)
```

FIG 61. Preparación para el entrenamiento.

Cree un diccionario de cada peso de clase para alimentar el modelo ya que los datos están desequilibrados, el cual nos muestra que hay 9895 registros para entrenamiento, 2474 para validación y 2183 para hacer la prueba correspondiente.

```
normal, abnormal = np.bincount(ptb_full.loc[:, 'Class'])
norm_weight = (1/normal) * ((normal+abnormal)/2)
abnorm_weight = (1/abnormal) * ((normal+abnormal)/2)
class_weight = {0: norm_weight, 1: abnorm_weight}

print("Tamaño del conjunto de datos de entrenamiento: ", train_ptb.shape)
print("Tamaño del conjunto de datos de validación: ", valid_ptb.shape)
print("Tamaño del conjunto de datos de prueba: ", test_ptb.shape)

Tamaño del conjunto de datos de entrenamiento: (9895, 187)
Tamaño del conjunto de datos de validación: (2474, 187)
Tamaño del conjunto de datos de prueba: (2183, 187)
```

FIG 62. Diccionarios en python

Función para construir el modelo de redes convolucionales el cual está dividido por capas que nos ayudan a identificar distintas características en las entradas de datos que en definitiva hacen que pueda identificar objetos y “ver” y con las funciones de activación relu y same nos ayuda con la comparación de las imágenes.

```
def build_conv1d_model(input_shape=(x_train_ptb.shape[1], 1)):
    model = keras.models.Sequential()

    model.add(Conv1D(32, 7, padding='same', input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(tf.keras.layers.ReLU())
    model.add(MaxPool1D(5, padding='same'))

    model.add(Conv1D(64, 7, padding='same'))
    model.add(BatchNormalization())
    model.add(tf.keras.layers.ReLU())
    model.add(MaxPool1D(5, padding='same'))

    model.add(Conv1D(128, 7, padding='same', input_shape=input_shape))
    model.add(BatchNormalization())
    model.add(tf.keras.layers.ReLU())
    model.add(MaxPool1D(5, padding='same'))
    model.add(Conv1D(256, 7, padding='same'))
```

```
model.add(BatchNormalization())
model.add(tf.keras.layers.ReLU())
model.add(MaxPool1D(5, padding='same'))
model.add(Conv1D(512,7, padding='same', input_shape=input_shape))
model.add(BatchNormalization())
model.add(tf.keras.layers.ReLU())
model.add(MaxPool1D(5, padding='same'))
model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(2, activation='relu'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=[tf.keras.metrics.F1Score(2, "micro")])
```

FIG 63. Función para construir redes convulsionales el cual es multicapa.

Serie de capas Conv1D con 2 bloques residuales

Creación de un modelo basado en una serie de capas Conv1D con 2 bloques residuales que están conectados a otra serie de capas densas conectadas por completo.

```
def build_conv1d_res_model(input_shape=(x_train_ptb.shape[1],1)):
    model = keras.models.Sequential()

    input_ = tf.keras.layers.Input(shape=input_shape)

    conv1_1 = Conv1D(64,7, padding='same', input_shape=input_shape)(input_)
    conv1_1 = BatchNormalization()(conv1_1)
    conv1_1 = tf.keras.layers.ReLU()(conv1_1)

    conv1_2 = Conv1D(64,7, padding='same')(conv1_1)
    conv1_2 = BatchNormalization()(conv1_2)
    conv1_2 = tf.keras.layers.ReLU()(conv1_2)

    conv1_3 = Conv1D(64,7, padding='same')(conv1_2)
    conv1_3 = BatchNormalization()(conv1_3)
    conv1_3 = tf.keras.layers.ReLU()(conv1_3)

    concat_1 = Concatenate()([conv1_1, conv1_3])
    max_1 = MaxPool1D(5, padding="same")(concat_1)

    conv1_4 = Conv1D(128,7, padding='same')(max_1)
    conv1_4 = BatchNormalization()(conv1_4)
    conv1_4 = tf.keras.layers.ReLU()(conv1_4)

    flat = Flatten()(max_2)
    dense_1 = Dense(400, activation='relu')(flat)
    drop_1 = Dropout(0.5)(dense_1)
    dense_2 = Dense(256, activation='relu')(drop_1)
    drop_2 = Dropout(0.5)(dense_2)
    dense_3 = Dense(128, activation='relu')(drop_2)
    dense_4 = Dense(64, activation='relu')(dense_3)
    dense_5 = Dense(32, activation='relu')(dense_4)
    dense_6 = Dense(2, activation="softmax")(dense_5)
```

FIG 64. Creación de un modelo basado en una serie de capas Conv1D con 2 bloques residuales que están conectados a otra serie de capas densas conectadas por completo

```
checkpoint_cb = ModelCheckpoint("conv1d_res_ptb.h5", save_best_only=True)

earlystop_cb = EarlyStopping(patience=5, restore_best_weights=True)

inp_shape = (x_train_ptb.shape[1], x_train_ptb.shape[2])
model_conv1d_res_ptb= build_conv1d_res_model(input_shape=(x_train_ptb.shape[1], x_train_ptb.shape[2]))

history_conv1d_res_ptb = model_conv1d_res_ptb.fit(x_train_ptb, y_train_ptb, epochs=10, batch_size=30,
                                                class_weight=class_weight, validation_data=(x_valid_ptb, y_valid_ptb),
                                                callbacks=[checkpoint_cb, earlystop_cb])
```

FIG 65. Procederemos con el entrenamiento definimos el número de épocas a considerar

```
model_conv1d_res_ptb.load_weights("conv1d_res_ptb.h5")
model_conv1d_res_ptb.evaluate(x_test_ptb, y_test_ptb)
```

```
69/69 [=====] - 2s 23ms/step - loss: 0.0678 - f1_score: 0.9794
[0.06777141243219376, 0.9793861508369446]
```

FIG 66. Cálculo de una función para hallar la probabilidad más alta de detección

Una vez realizada la función de la red convolucional procedemos a indicar el número de épocas (epoch) del y el tamaño del lote (batch size) el cual nos indica la cantidad de muestras procesadas en el entrenamiento.

```
checkpoint_cb = ModelCheckpoint("conv1d_ptb.h5", save_best_only=True)

earlystop_cb = EarlyStopping(patience=5, restore_best_weights=True)

model_conv1d_ptb= build_conv1d_model(input_shape=(x_train_ptb.shape[1], x_train_ptb.shape[2]))
model_conv1d_ptb.summary()

history_conv1d_ptb = model_conv1d_ptb.fit(x_train_ptb, y_train_ptb, epochs=50, batch_size=32,
                                         class_weight=class_weight, validation_data=(x_valid_ptb, y_valid_ptb),
                                         callbacks=[checkpoint_cb, earlystop_cb])
```

```
Epoch 1/50
310/310 [=====] - 31s 97ms/step - loss: 0.4750 - f1_score: 0.7482 - val_loss: 0.5983 - val_f1_score: 0.7175
Epoch 2/50
310/310 [=====] - 29s 94ms/step - loss: 0.2508 - f1_score: 0.8934 - val_loss: 0.2289 - val_f1_score: 0.8793
Epoch 3/50
310/310 [=====] - 29s 94ms/step - loss: 0.1546 - f1_score: 0.9388 - val_loss: 0.5888 - val_f1_score: 0.8293
Epoch 4/50
310/310 [=====] - 29s 95ms/step - loss: 0.1300 - f1_score: 0.9432 - val_loss: 0.5870 - val_f1_score: 0.9054
Epoch 5/50
310/310 [=====] - 29s 94ms/step - loss: 0.1814 - f1_score: 0.9638 - val_loss: 1.9897 - val_f1_score: 0.9538
Epoch 6/50
310/310 [=====] - 29s 94ms/step - loss: 0.1864 - f1_score: 0.9629 - val_loss: 0.5076 - val_f1_score: 0.8994
Epoch 7/50
310/310 [=====] - 29s 94ms/step - loss: 0.8883 - f1_score: 0.9679 - val_loss: 1.1688 - val_f1_score: 0.7352
Epoch 8/50
310/310 [=====] - 29s 94ms/step - loss: 0.8827 - f1_score: 0.9693 - val_loss: 0.5420 - val_f1_score: 0.8896
Epoch 9/50
310/310 [=====] - 29s 93ms/step - loss: 0.8783 - f1_score: 0.9738 - val_loss: 0.4845 - val_f1_score: 0.8892
```

FIG 67. Procederemos con el entrenamiento definimos el número de épocas a considerar

4.4. Algoritmo categorización de subtipos de arritmia del ECG

```
import os
import pywt
import time
import random
import itertools
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.svm import SVC
from tensorflow import keras
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.utils import resample
from keras.models import Sequential
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE

for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

FIG 68. Procederemos a importar algunas librerías y paquetes adicionales para la ejecución del algoritmo

Visualizamos información del dataset que lo hemos obtenido del mismo repositorio de kaggle ya que la universidad de guayaquil aún no cuenta con los datasets correspondiente y se asigna el número de clase que el algoritmo clasificara las imágenes del electrocardiograma.

- Número de muestras: 109446.
- Número de categorías: 5.
- Frecuencia de muestreo: 125Hz.
- Fuente de datos: Conjunto de datos de arritmia MIT-BIH de Physionet.

Clases:

- Latidos no ecóticos (latido normal) ('N'): 0.
- Latidos ectópicos supraventriculares ('S'): 1.
- Latidos ectópicos ventriculares ('V'): 2.
- Latidos de fusión ('F'): 3.
- Latidos desconocidos ('Q'): 4.

Procederemos con la importación de los datasets y vamos a leer la información a través del método read_csv.

ejemplos de la clase baja en el conjunto de datos de entrenamiento antes de ajustar un modelo. Esto puede equilibrar la distribución de clases, pero no proporciona ninguna información adicional al modelo. Una mejora en la duplicación de ejemplos de la clase baja es resumir nuevos ejemplos de la clase baja. Este es un tipo de aumento de datos para la clase minoritaria y se conoce como Técnica de sobre muestreo de minorías sintéticas (SMOTE).

```
steps = [('o', SMOTE(sampling_strategy=0.5)), ('u', RandomUnderSampler(sampling_strategy=0.7))]
pipeline = Pipeline(steps=steps)

def balance(train_df_0):
    data, label = [0] * len(set(list(train[train.columns[-1]])), [0] * len(set(list(train[train.columns[-1]])))
    for i in set(list(train[train.columns[-1]])):
        if i == 0.0:
            continue
        else:
            df = train[train[train.columns[-1]] == i]

            train_df = pd.concat([df_0, df])
            X = train_df[train_df.columns[:-1]]
            y = train_df[train_df.columns[-1]]
            # transform the dataset
            X, y = pipeline.fit_resample(X, y)

            data[int(i)] = X[y==1]
            label[int(i)] = y[y==1]
    return data, label, X, y

df_0 = data_train[data_train[data_train.columns[-1]] == 0].sample(n=30000, random_state=42)
temp, temp_1, X, y = balance(data_train, df_0)
```

FIG 72. Transformar el dataset en conjunto de datos de variables x y variable y

```
plt.figure(figsize=(15, 7.5))
plt.bar(y_train.value_counts(), labels=['Ritmo no ecotópico', 'Latidos suprainfrecuentes ecotópicos', 'Latidos entrafrecuentes ecotópicos',
                                         'Ritmos de fusión', 'Latidos desconocidos'],
        colors=['yellowgreen', 'dark', 'paletorainbow', 'slategray', 'dimgrey'], autopct='%2.1f%%')
p=plt.gca()
p.patches[0].artist.set(facecolor='white')
plt.show()
```

FIG 73. Trazar el número de registros en cada categoría

Con el método describe () obtenemos una información estadística de los datos de cada columna de la dataset importado.

```
data_train.describe()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
mean	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
std	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
90%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
95%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
99%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

MODELOS COMPUTACIONALES PARA LA DETECCIÓN Y ANÁLISIS DE ARRITMIAS CARDÍACAS,
 APLICANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO.

	0	1	2	3	4	5	6	7	8	9	...	178	179	180	181
mean	2.081200000	2.081200000	2.081200000	2.081200000	2.081200000	2.081200000	2.081200000	2.081200000	2.081200000	2.081200000	...	2.081200000	2.081200000	2.081200000	2.081200000
std	0.004410	0.004410	0.004410	0.004410	0.004410	0.004410	0.004410	0.004410	0.004410	0.004410	...	0.004410	0.004410	0.004410	0.004410
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000
max	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000
5%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000
95%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000
max	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000

FIG 74. Información estadística del dataset

Podemos visualizar los electrocardiogramas previamente seleccionados que están en los datasets.

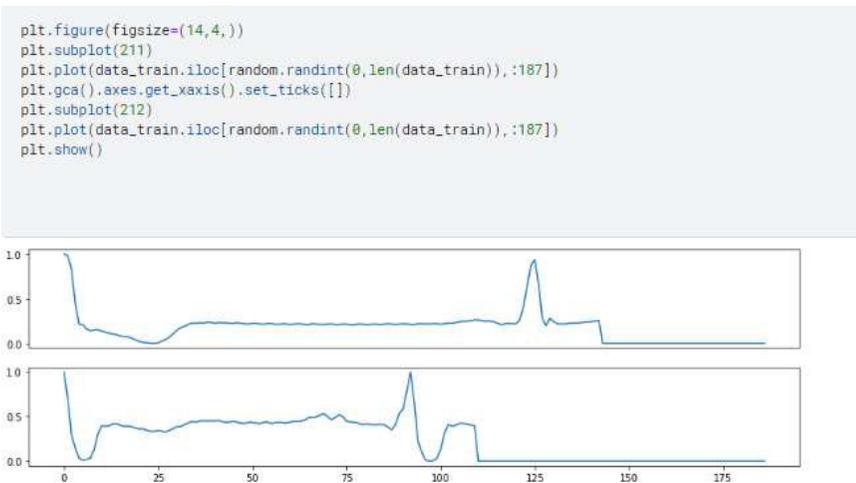


FIG 75. Tomamos muestra de los datos

En la siguiente función podemos categorizar las diferentes imágenes del dataset el cual se dividen en 5 grupos con un color diferente cada grupo para su mejor visualización.

```
label = ["Ritmos no ecoticos","Latidos supraventriculares ectopicos","Latidos ventriculares ectopicos",
        "Ritmos de fusión","Latidos desconocidos"]
color = ['blue','red','green','orange','purple','black']
fig, ax = plt.subplots(5, 1, sharex=True, sharey=True,figsize=(20,12))
for i, row in enumerate(ax):
    row.plot((data_train[data_train[187] == i].iloc[0]:-1), label=label[i], color=color[i])
    row.legend(fontsize=12)
fig.text(0.5, 0.04, 'Amplitude', va='center', ha='center', fontsize=15)
fig.text(0.04, 0.5, 'Time (ms)', va='center', ha='center', rotation='vertical', fontsize=15)
```

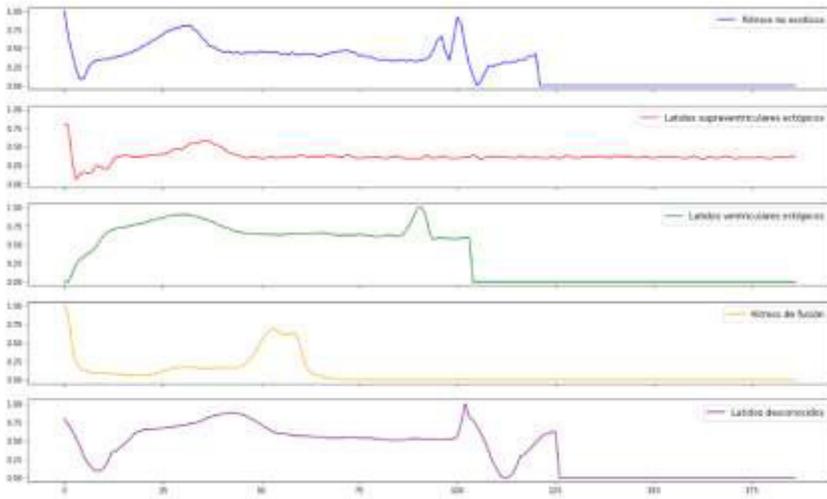


FIG 76. Se grafica los datos de ECG pertenecientes a cada etiqueta de clase

Se muestra la cantidad de registro totales en cada categoría de subtipo de arritmia el cual nos da como resultado porcentual que el 82.8% son latidos no ecóticos, el 0,7% latidos desconocidos para el algoritmo, el 2.5 % son latidos de ritmo de fusión, el 6.6% son latidos ventriculares ectópicos, y el 7.3 latidos supraventriculares ectópicos.

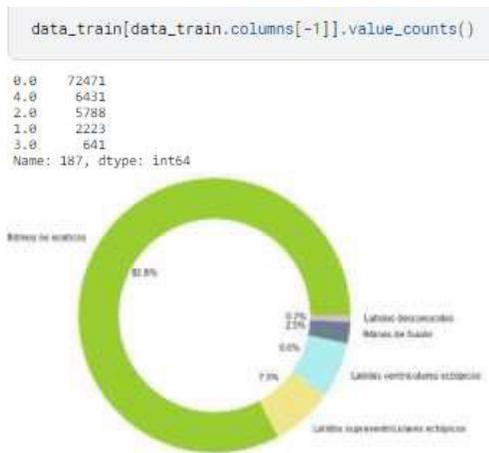


FIG 77. Visualizamos los datos porcentuales de subtipos de arritmia

El método. shape muestra el total de filas y columnas y con el método value_counts contamos esos registros utilizados en el entrenamiento.

```
X_train.shape, y_train.shape
((81428, 187), (81428,))

y_train.value_counts()
0.0    21428
1.0    15000
3.0    15000
4.0    15000
2.0    15000
dtype: int64
```

FIG 78. Nos muestra la cantidad de valores por cada clase

Procedemos con el entrenamiento el cual hicimos con 30 número de épocas el cual se demora 8 segundos por cada época el valor de datos perdidos que varía, y con un margen de exactitud que se va elevando a medida que aumentan las épocas.

```
Epoch 1/30
2545/2545 [=====] - 8s 3ms/step - loss: 0.2820 - accuracy: 0.8987
Epoch 2/30
2545/2545 [=====] - 8s 3ms/step - loss: 0.1494 - accuracy: 0.9471
Epoch 3/30
2545/2545 [=====] - 8s 3ms/step - loss: 0.1143 - accuracy: 0.9605
Epoch 4/30
2545/2545 [=====] - 8s 3ms/step - loss: 0.0965 - accuracy: 0.9662
Epoch 5/30
Epoch 27/30
2545/2545 [=====] - 8s 3ms/step - loss: 0.0253 - accuracy: 0.9924
Epoch 28/30
2545/2545 [=====] - 7s 3ms/step - loss: 0.0269 - accuracy: 0.9921
Epoch 29/30
2545/2545 [=====] - 7s 3ms/step - loss: 0.0241 - accuracy: 0.9925
Epoch 30/30
2545/2545 [=====] - 7s 3ms/step - loss: 0.0249 - accuracy: 0.9924
```

FIG 79. proceso durante entrenamiento

```
print("X_train", X_train.shape)
print("y_train", y_train.shape)
print("X_test", X_test.shape)
print("y_test", y_test.shape)

X_train (56285, 187, 1)
y_train (56285, 5, 2, 2)
X_test (23002, 187, 1)
y_test (23002, 5, 2, 2)
```

FIG 80. Se conforma los valores para luego establecer nuestra matriz de resultados

4.5 Algoritmo RestNet

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('whitegrid')
plt.style.use('seaborn-deep')
plt.style.use('fivethirtyeight')
plt.rcParams['font-size'] = 14
plt.rcParams['axes.labelsize'] = 12
plt.rcParams['axes.titlesize'] = 12
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
plt.rcParams['legend.fontsize'] = 12
plt.rcParams['figure.titlesize'] = 14
plt.rcParams['figure.figsize'] = (20,8)
plt.rcParams['ps.useafm'] = True
import random
random.seed(21)
import warnings
warnings.filterwarnings('ignore')
pd.options.mode.chained_assignment = None
pd.options.display.float_format = '{:.2f}'.format
```

FIG 81. FIG 81. Importación de librerías

```
df = pd.read_csv('../input/heartbeat/mitbih_train.csv', header=None)
print("Conjunto de datos de entrenamiento : ",df.shape)
df.head(10)
```

FIG 82. FIG 82. Conjunto de datos de entrenamiento

En esta parte nos dedicamos en si al entrenamiento del algoritmo, por el cual en la variable df le asignamos la lectura como tal del data set en formato .csv y posteriormente hacemos una presentación del contenido de los datos, el df.head(10) nos indicará cuantas filas presentará en este caso presentará 10 filas .A continuación se presenta la ejecución y la presentación del data set, como se lo presentará a continuación.

Conjunto de datos de entrenamiento (37564, 488)																																					
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30						
0	0.96	0.99	0.98	0.29	0.19	0.18	0.19	0.20	0.20	0.21	0.22	0.20	0.27	0.28	0.25	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10			
1	0.98	0.98	0.98	0.27	0.19	0.19	0.20	0.20	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10		
2	1.00	0.98	0.98	0.27	0.19	0.19	0.20	0.20	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
3	0.99	0.97	0.94	0.28	0.20	0.20	0.21	0.21	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
4	0.97	1.00	0.93	0.28	0.20	0.20	0.21	0.21	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
5	0.93	1.00	0.93	0.28	0.20	0.20	0.21	0.21	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
6	0.42	0.78	1.00	0.20	0.20	0.20	0.21	0.21	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
7	0.73	0.74	0.78	0.19	0.19	0.19	0.20	0.20	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
8	0.87	0.85	0.40	0.20	0.20	0.20	0.21	0.21	0.21	0.21	0.21	0.20	0.27	0.28	0.27	0.28	0.28	0.27	0.27	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10
9	0.90	1.00	0.95	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.25	0.24	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.10	0.08	0.09	0.11	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10	0.12	0.10

MODELOS COMPUTACIONALES PARA LA DETECCIÓN Y ANÁLISIS DE ARRITMIAS CARDÍACAS, APLICANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO.

188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------

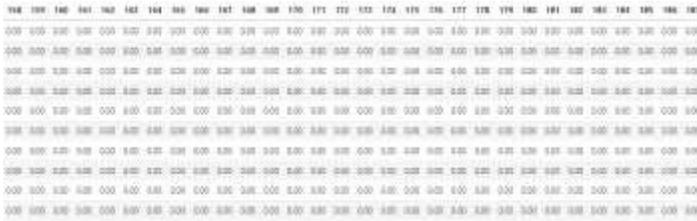


FIG 86. *FIG 86. Como se puede observar estos son los datos obtenidos de la data set para ejecutar las pruebas respectivas en 10 filas, las cuales van de 0 a las 9 y 188 columnas las cuales comprenden desde el 0 al 187*

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 87554 entries, 0 to 87553
Columns: 188 entries, 0 to 187
dtypes: float64(188)
memory usage: 125.6 MB
```

FIG 87. *Con esta función Vamos a obtener un resumen conciso del marco de los datos, en este caso va a ser de la variable df, en otras palabras, una descripción general rápida del conjunto de datos.*

En las cuales nos indica que de la librería importada pandas obtenemos la siguiente información de la variable antes mencionada:

- RangeIndex:** Índice de rango en el cual hemos tenido 87554 de 0 a 87553.
- Columns:** Las columnas son 188 desde 0 a 187.
- Dtypes:** El tipo de dato de cada dato es flotante.
- Memory:** La memoria utilizada en la ejecución de esta parte ha sido de 125.6 MB.

```
pd.Series(np.bincount(target_col), index=labels)
```

FIG 88. *Conjunto de datos desequilibrados*

Pd.series es una estructura de pandas, la cual es similar a los arrays de una dimensión en las cuales vamos a tener elementos del mismo tipo, por lo cual su tamaño no se puede cambiar pero si su contenido.

Np.bincount es una estructura de numpy la cual nos permite contar el número de ocurrencias de cada valor en el conjunto de datos int que no sean negativos.

Por ende, la ejecución de dicha estructura y conteo para así obtener una clasificación de los datos previamente cargados de los data sets y visualizados con su respectiva clasificación dan como resultado un ritmo normal de 72471, latido prematuro supraventricular de 2223 y contracción prematura de 5788.

Plt.figure es un método de la librería matplotlib la cual nos permite generar con uno o más ejes en nuestro caso va a ser con 2 ejes X, Y.

Sns.countplot es un método la cual pertenece a la librería seaborn, la cual nos permite visualizar gráficos estadísticos, este método se usa para mostrar los conteos de observaciones en cada uno de los contenedores o barras estadísticas.

Plt.title este método es por el cual nos permite ponerle nombre al gráfico estadístico



FIG 89. Observamos el gráfico previamente codificado, el cual nos muestra la distribución de los datos de entrenamiento entre 5 clases

Con el método **Plt.figure** de la librería matplotlib generamos un gráfico con uno o más ejes en nuestro caso va a ser con 2 ejes el X, Y.

Plt.plot con la siguiente función vamos a recibir un conjunto de valor en X e Y los cuales se van a mostrar en un plano definido por los ejes como puntos unidos por líneas.

Plt.legend la usamos para ponerle una leyenda o un nombre a los ejes.

Plt.title Nos permite ingresar un nombre al gráfico.

Plt.ylabel Nombre del eje Y.

Plt.xlabel Nombre del eje X.

Plt.show() Nos permite mostrar el gráfico.

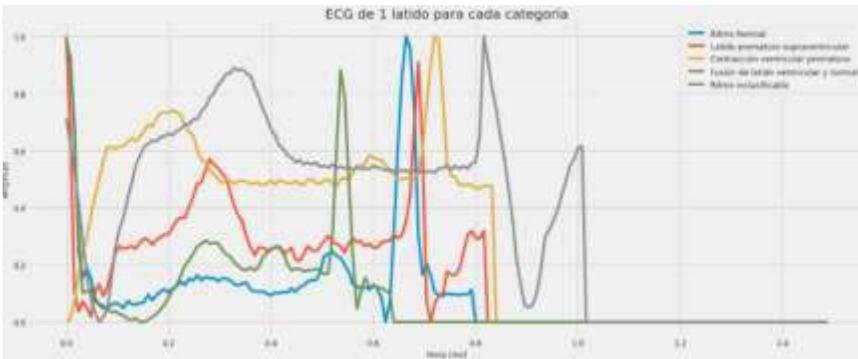


FIG 90. En este gráfico podemos observar las 5 clases definidas visualizadas en un gráfico en el cual podemos observar lo que son los ritmos normales, o latidos normales, latido prematuro supraventricular, contracción ventricular prematura, fusión de latido ventricular y normal, y los ritmos cardíacos inclasificables, todos tienden a una variación dependiendo de los datos obtenidos desde el Data set

```
from sklearn.model_selection import train_test_split
x_train,x_val,y_train,y_val = train_test_split(X,y,values,random_state=42,stratify=y,shuffle=True,train_size=0.8)
print("Datos de entrenamiento : ",x_train.shape,y_train.shape)
print("Datos de validación : ",x_val.shape,y_val.shape)
```

FIG 91. Preparación de los datos para la evaluación del modelo

Creamos la preparación para el modelo, y con la función `sklearn.model_selection` dividimos el dataset en 2 bloques los cuales son destinados para el entrenamiento y la validación del modelo, en nuestro caso son del 80% para entrenamiento y 20%

para las pruebas. Una vez generados los bloques de entrenamiento y pruebas, se puede obtener la variable objetivo para así crear de esta manera la estructura que utilizará el algoritmo, el parámetro `random_state` fija una semilla para el generador de número aleatorios, lo que permite ejecutar la función en este caso son 42 números que generará. El parámetro `Shuffle`, (el cual toma el valor de `True` por defecto) especifica si los registros deberán ser desordenados previamente o no.

```
import tensorflow as tf
y_train = tf.keras.utils.to_categorical(y_train)
y_val = tf.keras.utils.to_categorical(y_val)
y_test = tf.keras.utils.to_categorical(y_test)
x_train = np.expand_dims(x_train, 2)
x_val = np.expand_dims(x_val, 2)
x_test = np.expand_dims(x_test, 2)
print("Datos de entrenamiento : ", x_train.shape, y_train.shape)
print("Datos de validación : ", x_val.shape, y_val.shape)
print("Datos de prueba : ", x_test.shape, y_test.shape)
```

FIG 92. Importamos la librería `tensorflow` que es la que nos permite construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones análogos del aprendizaje y el razonamiento del humano

```
from keras.models import Model
from keras.layers import Input, Dense, Conv1D, MaxPooling1D, Softmax, Add, Flatten,
Activation, Dropout, BatchNormalization, AveragePooling1D, ZeroPadding1D
from keras import backend as K
from keras import Sequential
from keras.initializers import glorot_uniform
```

FIG 93. Modelado de datos

Importamos la librería `keras` que viene con la librería `TensorFlow` y es aquella que nos proporcionará bloques modulares sobre los que se van a desarrollar modelos complejos de aprendizaje profundo, además de `keras` nos permite crear `ResNet` el cual va a ser nuestro modelo seleccionado.

```
def identity_block(X, f, filters, stage, block):
    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'
    F1, F2, F3 = filters
    X_shortcut = X
```

```
X = Conv1D(filters=F1, kernel_size=1, strides=1, padding='valid', name=conv_name_base +
    '2a', kernel_initializer=glorot_uniform(seed=0))(X)
X = BatchNormalization(axis=2, name=bn_name_base + '2a')(X)
X = Activation('relu')(X)

X = Conv1D(filters=F2, kernel_size=f, strides=1, padding='same', name=conv_name_base +
    '2b', kernel_initializer=glorot_uniform(seed=0))(X)
X = BatchNormalization(axis=2, name=bn_name_base + '2b')(X)
X = Activation('relu')(X)

X = Conv1D(filters=F3, kernel_size=1, strides=1, padding='valid', name=conv_name_base +
    '2c', kernel_initializer=glorot_uniform(seed=0))(X)
X = BatchNormalization(axis=2, name=bn_name_base + '2c')(X)

X = Add()([X, X_shortcut])# SKIP Connection
X = Activation('relu')(X)

return X
```

FIG 94. Creación del bloque de identidad

Para este tipo de modelo primeramente debemos de crear un bloque de identidad, los parámetros strides y padding han sido seleccionados para que el tensor de salida tenga las mismas dimensiones seleccionadas que el modelo de entrada.

```
def convolutionsI_block(X, f, filters, stage, block, s=2):

    conv_name_base = 'res' + str(stage) + block + '_branch'
    bn_name_base = 'bn' + str(stage) + block + '_branch'

    F1, F2, F3 = filters

    X_shortcut = X

    X = Conv1D(filters=F1, kernel_size=1, strides=s, padding='valid', name=conv_name_base +
        '2a', kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=2, name=bn_name_base + '2a')(X)
    X = Activation('relu')(X)

    X = Conv1D(filters=F2, kernel_size=f, strides=1, padding='same', name=conv_name_base +
        '2b', kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=2, name=bn_name_base + '2b')(X)
    X = Activation('relu')(X)

    X = Conv1D(filters=F3, kernel_size=1, strides=1, padding='valid', name=conv_name_base +
        '2c', kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=2, name=bn_name_base + '2c')(X)

    X_shortcut = Conv1D(filters=F3, kernel_size=1, strides=s, padding='valid', name=conv_name_base + '1',
        kernel_initializer=glorot_uniform(seed=0))(X_shortcut)
    X_shortcut = BatchNormalization(axis=2, name=bn_name_base + '1')(X_shortcut)

    X = Add()([X, X_shortcut])
    X = Activation('relu')(X)

    return X
```

FIG 95. Creación del bloque de convulsión

Seguido de la creación del bloque de identidad se tiene que crear el bloque de convolución o convolucional, los parámetros en la capa

convolucional strides y padding han sido seleccionadas para que el tensor de salida tenga las mismas dimensiones seleccionadas que el modelo de entrada.

```
def ResNet(input_shape=(feature,depth)):
    X_input = Input(input_shape)

    X = ZeroPadding1D(3)(X_input)

    X = Conv1D(64, 7, strides=2, name='conv1', kernel_initializer=glorot_uniform(seed=8))(X)
    X = BatchNormalization(axis=2, name='bn_conv1')(X)
    X = Activation('relu')(X)
    X = MaxPooling1D(3, strides= 2)(X)

    X = convolutional_block(X, f=3, filters=[128, 128, 256], stage=2, block='a', s=1)
    X = identity_block(X, 3, [128, 128, 256], stage=2, block='b')
    X = identity_block(X, 3, [128, 128, 256], stage=2, block='c')

    X = convolutional_block(X, f=3, filters=[128, 128, 512], stage=3, block='a', s=2)
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='b')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='c')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='d')

    X = convolutional_block(X, f=3, filters=[256, 256, 1024], stage=4, block='a', s=2)
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='b')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='c')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='d')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='e')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='f')

    X = AveragePooling1D(pool_size= 2, padding='same')(X)

    model = Model(inputs=X_input, outputs=X, name='ResNet')

    return model
```

FIG 96. Bloque residual de identidad

En este tercer modelo o también llamado bloque residual identidad, con este modelo los datos que son puenteados no son modificados, esto es porque el tensor que resulta de las capas convolucionales anteriores tiene las mismas dimensiones que los datos originales, por lo que ya no es necesario ajustarlos.

layer (type)	Output Shape	Param #	Connected to
input_3 (InputLayer)	[(None, 107, 1)]	0	
zero_padding1d_2 (ZeroPadding1D)	(None, 103, 1)	0	input_3[0][0]
conv1 (Conv1D)	(None, 94, 64)	512	zero_padding1d_2[0][0]
bn_conv1 (BatchNormalization)	(None, 94, 64)	256	conv1[0][0]
activation_98 (Activation)	(None, 94, 64)	0	bn_conv1[0][0]
max_pooling1d_2 (MaxPooling1D)	(None, 48, 64)	0	activation_98[0][0]
res2a_branch2a (Conv1D)	(None, 46, 128)	8320	max_pooling1d_2[0][0]

activation_146 (Activation)	(None, 6, 2848)	0	add_47[0][0]
average_pooling1d_2 (AveragePoo	(None, 3, 2848)	0	activation_146[0][0]
flatten_2 (Flatten)	(None, 6144)	0	average_pooling1d_2[0][0]
fc1 (Dense)	(None, 256)	1573120	flatten_2[0][0]
fc2 (Dense)	(None, 128)	32896	fc1[0][0]
fc3 (Dense)	(None, 5)	645	fc2[0][0]

Total params: 17,839,109			
Trainable params: 17,785,221			
Non-trainable params: 53,888			

FIG 97. Aquí podemos observar cómo se van realizando los cambios antes mencionados en cada capa sucesiva, la columna "Output Shape" nos va a mostrar el tamaño de los mapas de características en cada una de las capas

```
import keras
h = model.fit(
    x_train, y_train, validation_data=(x_val, y_val),
    epochs=80, batch_size=batch_size,
    callbacks = [
        keras.callbacks.EarlyStopping(monitor='val_acc', patience=2, mode='auto'),
        keras.callbacks.ModelCheckpoint('model_val_acc_{f}.h5',
            save_best_only = True, save_weights_only=False,
            monitor='val_acc')
    ])

```

FIG 98. Importación de librería keras

Para este siguiente paso nuevamente importamos la librería keras la cual nos va a permitir realizar el entrenamiento del modelo, y la cantidad de iteraciones de aprendizaje (épocas) de entrenamiento, en nuestro caso hemos selecciona 80 épocas, las cuales nos van a permitir que nuestro modelo aprenda y saque los mejores resultados del entrenamiento.

4.6. Resumen

En esta sección se procedió a realizar la optimización de los algoritmos de Machine Learning seleccionados por lo que, se configuró el entorno de desarrollo en Google Colaboratory debido a los beneficios en cuanto a ejecución de los modelos que se ofrece. Los algoritmos tomados en cuenta son la detección de arritmias usando ECG, categorización de subtipos de arritmia del ECG y Algoritmo RestNet y para llevar a cabo el proceso mencionado, se instalaron las librerías de Python y cargaron los datos en formato csv necesarios, reservando campos para realizar la evaluación del entrenamiento y aportar en la optimización.



CAPÍTULO 5:

EVALUACIÓN DEL RESULTADO DE ENTRENAMIENTO DE LOS ALGORITMOS MEDIANTE DATOS DE PRUEBA

5.1. Resultados obtenidos del algoritmo de detección de arritmias usando ECG

Se crea la función para evaluar el entrenamiento el cual nos muestra un resultado de los datos perdidos (los) del 0,10% y el valor de precisión es 0,95% mientras más bajo sea el valor de los más eficiente es la red neuronal y mientras más alto sea el valor de f1 más alto mayor eficacia tendrá nuestra red.

```
model_conv1d_ptb.load_weights("conv1d_ptb.h5")
model_conv1d_ptb.evaluate(x_test_ptb,y_test_ptb)

conv1d_pred_proba_ptb = model_conv1d_ptb.predict (x_test_ptb)
conv1d_pred_ptb = np.argmax(conv1d_pred_proba_ptb, axis=1)

model_conv1d_ptb.load_weights("conv1d_ptb.h5")
model_conv1d_ptb.evaluate(x_test_ptb,y_test_ptb)

69/69 [=====] - 1s 7ms/step - loss: 0.1066 - f1_score: 0.9560
[0.10659676045179367, 0.9560238122940065]
```

FIG 99. Cálculo de una función para hallar la probabilidad más alta de detección

```
plt.figure(figsize=(20,8))
plt.plot(history_conv1d_ptb.epoch, history_conv1d_ptb.history['loss'],
         color='r', label='Entrenamiento perdido')
plt.plot(history_conv1d_ptb.epoch, history_conv1d_ptb.history['val_loss'],
         color='b', label='Validacion perdida' , linestyle="--")
plt.xlabel('Epoca')
plt.ylabel('Pérdida')
plt.plot(history_conv1d_ptb.epoch, history_conv1d_ptb.history['f1_score'],
         color='g', label='Entrenamiento F1')
plt.plot(history_conv1d_ptb.epoch, history_conv1d_ptb.history['val_f1_score'],
         color='c', label='Validacion F1' , linestyle="--")
plt.xlabel('Epoocas')
plt.ylabel('Pérdida')
plt.legend()
plt.show()
```

FIG 100. Elaboramos nuestra matriz de confusión para presentar los resultados obtenidos

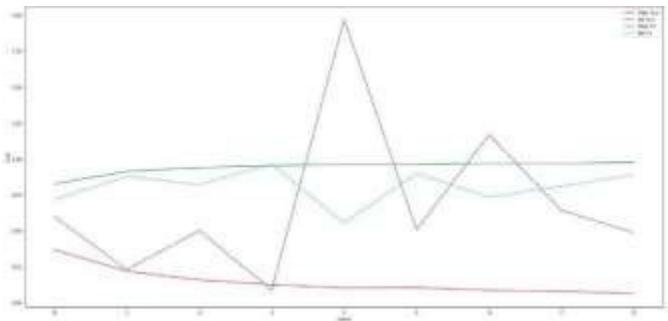


FIG 101. Gráfico estadístico de entrenamiento y validación

Serie de capas Conv1D con 2 bloques residuales

Reporte de los resultados de entrenamiento el cual la clase normal nos da un 0,99% de precisión, la exhaustividad del 0,94% y el valor de F1 del 0,96%. Mientras que en la clase anormal la precisión nos da un 0,98%, la exhaustividad del 0,1.00 y el valor del F1 0,99%

```
print(classification_report(out_test_ptb, conv1d_res_pred_ptb > 0.5, target_names=PTB_Outcome))
```

	precision	recall	f1-score	support
Normal	0.99	0.94	0.96	414
Abnormal	0.98	1.00	0.99	207
accuracy			0.98	2107
macro avg	0.98	0.97	0.97	2107
weighted avg	0.98	0.98	0.98	2107

FIG 102. Matriz de resultados

Mostrar los resultados de entrenamiento el cual podemos observar que los datos perdidos (train loss) a medida que avanza el entrenamiento van disminuyendo en la gráfica, mientras que la curva de (train fl) nos indica que se mantiene constante es decir que tiene un nivel alto de predicción y bajo de exhaustividad.



FIG 103. Gráfico estadístico de entrenamiento y validación

Dado que el conjunto de datos del MIT ya viene como un conjunto de entrenamiento y un conjunto de prueba, solo dividimos el 20% del conjunto de entrenamiento para la validación.

```

train_mit, valid_mit, test_mit, test_mit = train_test_split(train_data, test_size=0.2, random_state=42)

test_mit, test_test_mit = test_test_split(test_data, test_size=0.5)

train_mit = normalizer(train_mit, axis=0, norm='max')
valid_mit = normalizer(valid_mit, axis=0, norm='max')
test_mit = normalizer(test_mit, axis=0, norm='max')
    
```

FIG 104. Conjunto de datos del MIT

Una vez ya equilibrado los datos y puesta el método de evaluación se procede con el entrenamiento del algoritmo el cual se asigna el número de épocas para su aprendizaje.

```
checkpoint_cb = ModelCheckpoint("conv1d_mit.h5", save_best_only=True)

earlystop_cb = EarlyStopping(patience=5, restore_best_weights=True)

model_conv1d_mit = build_conv1d_model_mit(input_shape=(x_train_mit.shape[1], x_train_mit.shape[2]))
model_conv1d_mit.summary()

history_conv1d_mit = model_conv1d_mit.fit(x_train_mit, y_train_mit, epochs=2, batch_size=256,
                                         class_weight=class_weight_mit, validation_data=(x_valid_mit, y_valid_mit),
                                         callbacks=[checkpoint_cb, earlystop_cb])
```

FIG 105. Procederemos con el entrenamiento definimos el número de épocas a considerar

```
x_train_mit = train_mit.reshape(len(train_mit), train_mit.shape[1], 1)
x_valid_mit = valid_mit.reshape(len(valid_mit), valid_mit.shape[1], 1)
x_test_mit = test_mit.reshape(len(test_mit), test_mit.shape[1], 1)

y_train_mit = to_categorical(out_train_mit)
y_valid_mit = to_categorical(out_valid_mit)
y_test_mit = to_categorical(out_test_mit)
```

FIG 106. Se reforma el marco de datos en una matriz Numpy y se convierte el resultado en una matriz categórica para cada clase

Se crea las variables para clasificar los datos del dataset mit_train el cual una vez echo el entrenamiento los datos entraran en cada una de las categorías normalmit, superpre, prevent, fusión, unclass.

```
normalmit, superpre, prevent, fusion, unclass = np.bincount(mit_train.loc[:, 'Class'])
total_rec = len(mit_train)
normmit_weight = (1/normalmit) * (total_rec/5)
superpre_weight = (1/superpre) * (total_rec/5)
prevent_weight = (1/prevent) * (total_rec/5)
fusion_weight = (1/fusion) * (total_rec/5)
unclass_weight = (1/unclass) * (total_rec/5)
class_weight_mit = {0: normmit_weight, 1: superpre_weight, 2: prevent_weight,
                    3: fusion_weight, 4: unclass_weight}
```

```
print("Training dataset size: ", x_train_mit.shape, " -- Y size: ", y_train_mit.shape)
print("Validation dataset size: ", x_valid_mit.shape, " -- Y size: ", y_valid_mit.shape)
print("Test dataset size: ", x_test_mit.shape, " -- Y size: ", y_test_mit.shape)

Training dataset size: (70043, 187, 1) -- Y size: (70043, 5)
Validation dataset size: (17511, 187, 1) -- Y size: (17511, 5)
Test dataset size: (21892, 187, 1) -- Y size: (21892, 5)
```

FIG 107. Cálculo de ponderaciones de cada clase para alimentar el modelo

5.2. Resultados obtenidos del algoritmo categorización de subtipos de arritmia del ECG

Nivel de precisión del 92.6 % al momento de detectar algún subtipo de arritmia y tiempo de ejecución el cual se dio en 6 minutos y 54 segundos

```

y_pred = model.predict(X_test)
print("Accuracy: ", sum(y_test == y_pred)/len(y_test)*100, '%')
print("Execution time: ", (time.time() - start_time)/60, 'minutes')

```

Accuracy: 92.60468442170655 %
 Execution time: 6.545802521705627 minutes

FIG 108. Exactitud del algoritmo

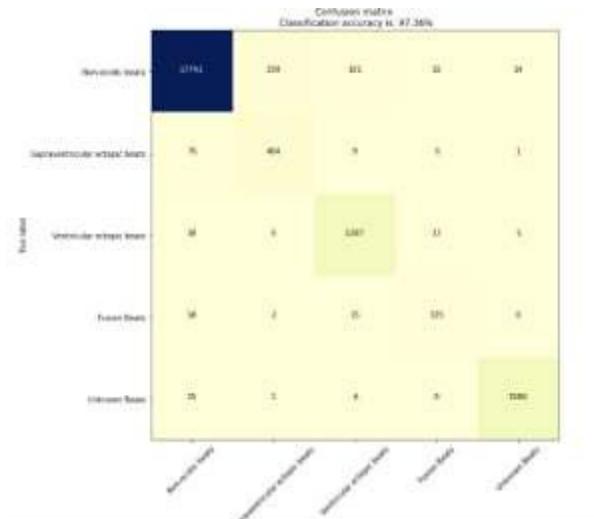


FIG 109. Muestra de resultados

Resultados de la figura 77 de nuestro entrenamiento el cual podemos ver que los ritmos no ecóticos serían 17741, Latidos desconocidos 1586, Latidos ectópicos ventriculares 1387, Latidos ectópicos supraventriculares 464, Latidos de fusión 135 el cual los algoritmos si nos detectó con una precisión de 92.6% los demás resultados serían falsos positivos el cual los electrocardiogramas serian falso para arritmia, pero el algoritmo los detecto como positivo.

5.3. Resultados obtenidos del algoritmo RestNet

```
Epoch 1/50  
137/137 [*****] - 500s 5s/step - loss: 0.7501 - auc: 0.9859 - val_loss: 0.7959 - val_auc: 0.9948  
Epoch 2/50  
137/137 [*****] - 500s 5s/step - loss: 0.6083 - auc: 0.9975 - val_loss: 0.6961 - val_auc: 0.9443
```

FIG 110. Épocas del algoritmo

En este caso podemos observar 2 épocas o 2 entrenamientos de 80, la cual nos da como resultados los siguientes:

- Loss:** Valores perdidos en el entrenamiento.
- Auc:** Puntuación que evalúa el rendimiento del algoritmo en el entrenamiento.
- Val_loss:** Valores perdidos en la validación.
- Val_auc:** Valores de precisión en la validación.

Estas épocas nos muestran tanto la perdida como la precisión, en el conjunto de datos original como en el conjunto de datos de validación.

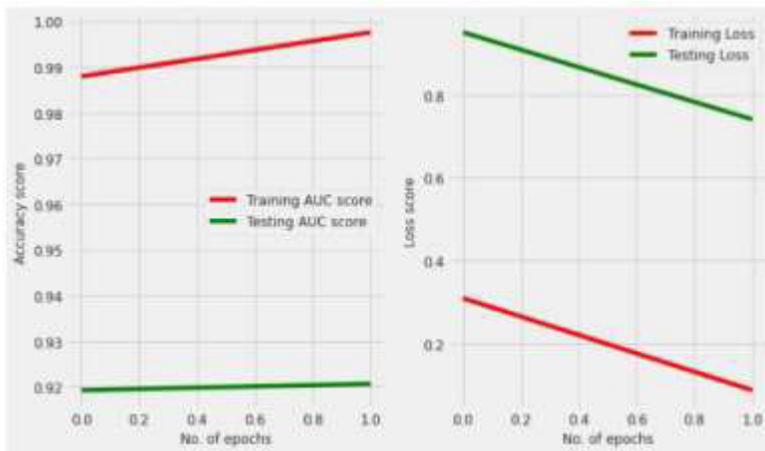


FIG 111. Representación gráfica de las Acurracy y del puntaje de pérdidas versus el número de épocas

Una vez finalizado el entrenamiento obtenemos estas 2 gráficas las cuales en la primera grafica identificamos las Acurracy score (puntuación en la precisión) versus No. Of epochs (el número de

épocas), en la cual nos indica que mientras más entrenamiento se realice vamos a tener una precisión mayor del algoritmo, la línea roja significa el puntaje de entrenamiento AUC, mientras que la línea verde significa el puntaje de prueba AUC.

En la siguiente gráfica obtenemos el puntaje de perdidas versus el número de épocas, en la cual nos da como un resultado que, a mayor número de épocas, menor número de perdidas tenemos, la línea roja en la gráfica significa las perdidas en el entrenamiento y la verde las perdidas en las pruebas como tales ambas van disminuyendo según vayan aumentando las épocas.

```
model.evaluate(x_val, y_val)
```

```
548/548 [=====] - 75s 137ms/step - loss: 0.0619 - auc: 0.9981  
[0.06193983182311058, 0.9981079697608948]
```

FIG 112. Nos permite pasar los lotes de prueba y emitir los puntajes para la validación realizada.

Finalmente obtenemos los resultados de nuestras pruebas realizadas al algoritmo y obtuvimos un 6% perdida y un 99% de precisión

```
model.evaluate(x_test, y_test)
```

```
685/685 [=====] - 93s 136ms/step - loss: 0.0716 - auc: 0.9975  
[0.07155568897724152, 0.9975442886352539]
```

FIG 113. Nos permite pasar los lotes de prueba y emitir los puntajes de la prueba realizada

Una vez finalizada la prueba a la cual fue sometido nuestro algoritmo, obtenemos un resultado que tiene una efectividad del 99%.

A Continuación, tenemos una tabla donde podemos observar las siguientes métricas:

Precision (Precisión): Nos permite medir la calidad del modelo de machine learning.

Recall (Exhaustividad): Nos ayuda a tener información con lo referente a la cantidad que el algoritmo es capaz de identificar.

F1-Score: Es utilizado para combinar las medidas de precisión y recall en un solo valor (ambos valores sumados y divididos para 2).

La interpretación de estos resultados por ejemplo es que en nuestra clase 4 tenemos una alta precisión del 99% y una baja exhaustividad del 97% por lo tanto nuestro modelo no detecta la clase muy bien pero cuando lo hace si es altamente confiable, a diferencia de la clase 0 que tiene un porcentaje del 99% tanto para la precisión, exhaustividad y el score por lo tanto nuestro modelo detecta muy bien la clase y es altamente confiable.

En nuestro caso nuestra matriz de confusión nos indica que tenemos 5 columnas que vienen a ser las clases y las filas pertenecen los valores verdaderos de cada clase como tal, procedemos a graficar nuestra matriz por medio de la función heatmap de la biblioteca de seaborn que es la que nos permite visualizar gráficamente la matriz, para poder visualizar la matriz nos ayudamos de la función plt.show(), en nuestro caso ya normalizamos los datos de nuestra matriz, es por ello que para hacerlo utilizamos la función normalize=true la cual nos permite normalizar por los datos verdaderos es decir vamos a dividir sobre el total de casos existentes reales por cada clase, las filas como tal deben de sumar 1.

```
fig = plt.figure(figsize=(8, 8))
ax = plt.subplot()
cm = confusion_matrix(y_true=np.argmax(y_val,axis=1),y_pred=np.argmax(y_pred_val,axis=1),normalize="true")
sns.heatmap(cm, annot=True, ax = ax, fmt = '.1f',)
ax.set_xlabel('Etiquetas predicha')
ax.set_ylabel('Etiquetas Real')
plt.show()
```

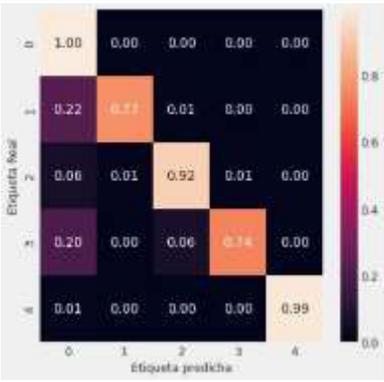


FIG 114. Validación y Prueba de la data set.

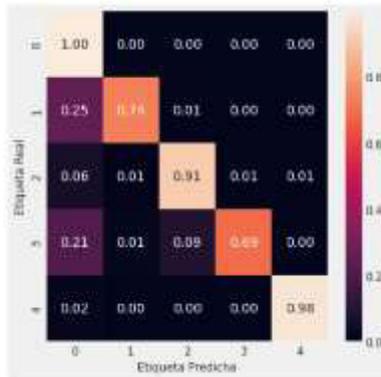


FIG 115. Validación y matriz de confusión de la data set

```
fig = plt.figure(figsize=(6, 6))
ax = plt.subplot()
cm = confusion_matrix(y_true=np.argmax(y_test,axis=1),y_pred=np.argmax(y_pred,axis=1),normalize="true")
sns.heatmap(cm, annot=True, ax = ax, fmt = '.2f')
ax.set_xlabel('Etiqueta Predicha')
ax.set_ylabel('Etiqueta Real')
plt.show()
```

FIG 116. Prueba y matriz de confusión de la data set.

```
El puntaje f1 para la clase 0 (Ritmo Normal ) : 0.990941966592044
El puntaje f1 para la clase 1 (Latido prematuro supraventricular) : 0.8163265306122448
El puntaje f1 para la clase 2 (Contracción ventricular prematura) : 0.9600679694137637
El puntaje f1 para la clase 3 (Fusión de latido ventricular y normal) : 0.7642276422764227
El puntaje f1 para la clase 4 (Ritmo inclasificable) : 0.9859044635865308
```

FIG 117. Resultados de la validación y de las pruebas

Una vez interpretada la matriz de confusión obtenemos los valores finales de precisión de la validación por lo cual nos da los siguientes resultados:

- Ritmo Normal: 99%.
- Latido prematuro supraventricular: 81%.
- Contracción ventricular prematura: 96%.
- Fusión de latido ventricular y normal: 76%.
- Ritmo inclasificable: 98%.

```
El puntaje f1 para la clases 0 (Ritmo Normal ) : 0.9897117077464788
El puntaje f1 para la clases 1 (Latido prematuro supraventricular) : 0.7907425265188042
El puntaje f1 para la clases 2 (Contracción ventricular prematura) : 0.954639175257732
El puntaje f1 para la clases 3 (Fusión de latido ventricular y normal) : 0.7557003257328991
El puntaje f1 para la clases 4 (Ritmo inclasificable) : 0.9830081812460668
```

FIG 118. Resultados de las pruebas en cuanto a los porcentajes de precisión

- Ritmo Normal: 98%.
- Latido prematuro supraventricular: 79%.
- Contracción ventricular prematura: 95%.
- Fusión de latido ventricular y normal: 75%.
- Ritmo inclasificable: 98%

5.4. Resumen

En la sección se detalla los resultados obtenidos por algoritmos optimizados en donde, se recalca la importancia de datos reservados para la evaluación, y tomando en cuenta esto, el modelo que corresponde a detección de arritmias usando ECG indica que la precisión es de 0,95% y al realizar la serie de capas Conv1D con 2 bloques residuales reporta un 0,99% de precisión, la exhaustividad del 0,94%. Así mismo, el algoritmo de categorización de subtipos de arritmia del ECG tiene un nivel de precisión del 92.6 % al momento de detectar algún subtipo de arritmia y tiempo de ejecución el cual se dio en 6 minutos y 54 segundos. Por último, el algoritmo RestNet demostró un 6% perdida y un 99% de precisión sobre datos entregados.

CONCLUSIÓN

Se ha realizado una búsqueda sistemática de algoritmos de Machine Learning, que utilicen un modelo de aprendizaje supervisado para disminuir los errores ante la presencia de arritmias cardíacas en un tratamiento cardiológico. Los problemas del corazón son una de las principales causas de muerte en el mundo, es por ello que se han estudiado algoritmos, que se centran en controlar a una temprana edad falencias en este órgano, reduciendo así la tasa de mortandad en especial de personas en la edad adulta, o en pacientes que tengan alguna anomalía llegar a tener un mejor diagnóstico, más rápido, eficaz y certero para tener un oportuno tratamiento.

Para la realización de la búsqueda sistemática de los algoritmos de Machine Learning aplicados a enfermedades del corazón en repositorios y revistas de alto impacto nos basamos en el estudio de arritmias cardíacas, debido a que en las mismas existen algoritmos que tratan otros tipos de anomalía.

Con la ayuda de Google Colab y librerías de TensorFlow facilito la implementación y pruebas del algoritmo, haciendo de una manera mucho más ágil el trabajo y el entrenamiento de la red neuronal convolucional. En la consideración del dataset de características cardiológicas para la detección y diagnósticos de arritmias cardíacas, se eligió un data set público del repositorio de "Kaggle" ya que al momento aún no contamos con los datasets de la facultad de medicina, en los mismos que se iba a almacenar la información obtenida a través de relojes inteligentes (smartwatch)

en los cuales se toma los pulsos eléctricos del corazón y a su vez se almacena en los datasets para posteriormente ser utilizarlos en los algoritmos estudiados.

En la implementación del ambiente de desarrollo, pruebas y validación de los algoritmos de machine Learning para el diagnóstico de arritmias cardíacas, se lo hizo en el mismo repositorio de “Kaggle” ya que posee su propio editor, y nos permitió realizar esta parte, creando así nuestra cuenta y con ello poder crear un cuadernillo en el cual pudimos probar, validar y a su vez poder realizar las pruebas respectivas. La validación de los resultados que indiquen los algoritmos de machine Learning se lo hizo por medio de una validación cruzada que es una técnica en la cual nos permite tomar el dataset y dividirlo en subconjuntos de información y luego utilizar cada subconjunto en el entrenamiento.

REFERENCIAS BIBLIOGRÁFICAS

- APD. (2021). Aplicaciones reales de la inteligencia artificial en la medicina | APD. <https://www.apd.es/aplicaciones-inteligencia-artificial-en-medicina/>
- Arias, E. (2021). Investigación descriptiva - Qué es, definición y concepto | 2022 | Economipedia. <https://economipedia.com/definiciones/investigacion-descriptiva.html>
- Aristega, S., & León, C. (2022). Optimización de algoritmos de machine learning aplicados a problemas de arritmias cardíacas.
- Attallah, O. (2022). An Intelligent ECG-Based Tool for Diagnosing COVID-19 via Ensemble Deep Learning Techniques. *Biosensors*, 12(5), 299. <https://doi.org/10.3390/BIOS12050299/S1>
- Bazan Yaranga, B. C. (2018). Transferencia de Aprendizaje Mediante Redes Neuronales Convolucionales para el Reconocimiento de Conductores Distráidos. *TECNIA*, 28(2). <https://doi.org/10.21754/TECNIA.V28I2.549>
- Caparrini, F. (2020). Aprendizaje Supervisado y No Supervisado. <http://www.cs.us.es/~fsancho/?e=77>
- Cardiología Madrid. (2023). Información general sobre el corazón. <https://cardioquiron.com/informacion-general-sobre-el-corazon/>
- Castelloninformacion. (2020). La fibrilación auricular, la arritmia cardíaca más frecuente, no controlada multiplica por cinco el riesgo de ictus - Castellon Información. <https://www.castelloninformacion.com/fibrilacion->

auricular-arritmia-cardiaca-frecuente-multiplica-cinco-ictus/

- Caycedo-Casas, X., Sancan-Moreira, M. T., & Looor-Mera I, L. R. (2017). Factores de riesgo cardiovascular en una población laboral. *Polo Del Conocimiento*, 2(5), 1115–1127. <https://polodelconocimiento.com/ojs/index.php/es/article/view/182>
- Christov, I., Dotsinsky, I., Simova, I., Prokopova, R., Trendafilova, E., & Naydenov, S. (2006). Dataset of manually measured QT intervals in the electrocardiogram. *BioMedical Engineering Online*, 5(Table 4), 1–8. <https://doi.org/10.1186/1475-925X-5-31>
- Duarte, J. & Uribe, N. (2019). ESTUDIO E IMPLEMENTACIÓN DE MACHINE LEARNING EN EL DESARROLLO DE VIDEOJUEGOS. https://repository.unab.edu.co/bitstream/handle/20.500.12749/6971/2019_Tesis_Duarte_Antolinez_Juan_Diego.pdf?sequence=1&isAllowed=y
- El Universo. (2021). Arritmias cardíacas. <https://www.eluniverso.com/guayaquil/comunidad/taquicardias-arritmias-e/>
- Felman, A. (2019). Enfermedad cardíaca: Tipos, causas y tratamientos. <https://www.medicalnewstoday.com/articles/es/327293#tipos>
- Flórez, E., & Augusto, C. (2022). Estudio del metodo de Schneiders para la generación de mallas cuadrangulares no estructuradas y su aplicación en la simulación de la electrofisiología del corazón [Universidad del Quindío]. <https://bdigital.uniquindio.edu.co/handle/001/6225>
- Gerard, T. (2018). PRINCIPIOS DE ANATOMÍA Y FISIOLOGÍA (15th ed.). Editorial Medica Panamericana.
- Google Cloud. (2022). Guía avanzada de Inception v3. <https://cloud.google.com/tpu/docs/inception-v3-advanced>
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

- <https://doi.org/10.48550/arxiv.1704.04861>
- IBM. (2022). Inteligencia Artificial en la Medicina | IBM. <https://www.ibm.com/ar-es/topics/artificial-intelligence-medicine>
- INEC. (2018). Instituto Nacional de Estadística y Censos. <https://www.ecuadorencifras.gob.ec/institucional/home/>
- Jalomo, J. (2019). Diseño de una red neuronal convolucional para la clasificación de señales biomédicas Design of a convolutional neural network for classification of biomedical signals.
- Kucheryavskiy, S., Zhilin, S., Rodionova, O., & Pomerantsev, A. (2020). Procrustes Cross-Validation - A Bridge between Cross-Validation and Independent Validation Sets. *Analytical Chemistry*, 92(17), 11842-11850. <https://doi.org/10.1021/ACS.ANALCHEM.0C02175>
- Latarget, M., & Ruiz, L. (2019). LATARJET: Anatomía Humana II. In *Editorial Médica Panamericana* (Vol. 2, Issue 9).
- Leivi, A. E. (2019). Análisis de la implementación de Machine Learning en el diagnóstico por imágenes.
- Lubanovic, B. (2019). Introducing Python. https://books.google.com.ec/books/about/Introducing_Python.html?id=i8m5nQAACAAJ&redir_esc=y
- Luque Bernal, R. M. (2021). Anatomía del corazón y vasos sanguíneos.
- Mayo Clinic. (2022). Presión arterial alta (hipertensión) - Síntomas y causas - Mayo Clinic. <https://www.mayoclinic.org/es-es/diseases-conditions/high-blood-pressure/symptoms-causes/syc-20373410>
- Mayoclinic. (2021). Electrocardiogram (ECG or EKG) - Mayo Clinic. <https://www.mayoclinic.org/es-es/tests-procedures/ekg/about/pac-20384983>
- Medlineplus. (2022). Monitor Holter (24 horas). <https://medlineplus.gov/spanish/ency/article/003877.htm>
- Mixon, E., & Wigmore, I. (2018). What is Google Drive? <https://www.techtarget.com/searchmobilecomputing/definition/Google-Drive>
- National Heart Lung and Blood Institute. (2022). Cómo fluye la

sangre por el corazón.
<https://www.cdc.gov/ncbddd/spanish/heartdefects/howtheheartworks.html>

- OMS. (2017). Enfermedad pulmonar obstructiva crónica (EPOC). [https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
- Ortega, J. P., Castillo, A. B., Lozano, J. M. O., Valverde, D. R., Soto, P. R., & Carmona, C. D. G. (2019). Comparación de dos métodos de detección para el registro de la frecuencia cardíaca: banda torácica vs camiseta técnica (Comparison of two methods for recording heart rate telemetry: chest band vs technical shirt). *Retos*, 36(36), 469-473. <https://doi.org/10.47197/RETOS.V36I36.68676>
- Rahman, T., Akinbi, A., Chowdhury, M. E. H., Rashid, T. A., Şengür, A., Khandakar, A., Islam, K. R., & Ismael, A. M. (2022). COV-ECGNET: COVID-19 detection using ECG trace images with deep convolutional neural network. *Health Information Science and Systems*, 10(1), 1-16. <https://doi.org/10.1007/S13755-021-00169-1/FIGURES/12>
- Regalado, M., & Medina, A. (2022). La inteligencia artificial al servicio de la medicina. *Atención Primaria Práctica*, 4(3), 100143. <https://doi.org/10.1016/J.APPR.2022.100143>
- SAN MARTÍN, A. G. (2020). Diseño de un modelo compuesto de arquitecturas de red neuronal convolucional y recurrente para descripción de vídeo en entornos outdoor/indoor. [https://riunet.upv.es/bitstream/handle/10251/165340/Golfe - Dseño de un modelo compuesto de arquitecturas de red neuronal convoluconal y recurrente pa....pdf?sequence=1](https://riunet.upv.es/bitstream/handle/10251/165340/Golfe-Diseño%20de%20un%20modelo%20compuesto%20de%20arquitecturas%20de%20red%20neuronal%20convoluconal%20y%20recurrente%20pa....pdf?sequence=1)
- Solbiati, M., Costantino, G., Casazza, G., Dipaola, F., Galli, A., Furlan, R., Montano, N., & Sheldon, R. (2016). Implantable loop recorder versus conventional diagnostic workup for unexplained recurrent syncope. *Cochrane Database of Systematic Reviews*, 2016(4). <https://doi.org/10.1002/14651858.CD011637.PUB2/ABSTRACT>
- Texasheart. (2020). Factores de riesgo cardiovascular para niños y adolescentes | Texas Heart Institute.

<https://www.texasheart.org/heart-health/heart-information-center/topics/factores-de-riesgo-cardiovascular-para-ninos-y-adolescentes/>

- Tobalina, B. (2020). La Inteligencia Artificial permite crear fármacos en la mitad de tiempo. <https://www.madrimasd.org/notiweb/noticias/inteligencia-artificial-permite-crear-farmacos-en-mitad-tiempo>
- Wegner, Ellermann, Lange, Frommeyer, Leitz, Eckardt, & Dechering. (2020). Revista Europea de Medicina Interna Evaluación prospectiva ciega del monitor de ECG AliveCor Kardia basado en un teléfono inteligente para la detección de fibrilación auricular : el estudio PEAK-AF.
- De Souza, A. D. (1991). ARRITMIAS CARDIACAS. In Revista Brasileira de Medicina (Vol. 48, Issue 5, pp. 219-234).



Alfonso Aníbal Guijarro Rodríguez

Profesor de la Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, Ecuatoriano, Ingeniero en Computación, Máster en Docencia y Gerencia en Educación Superior, Máster Universitario en Modelado Computacional en Ingeniería. Profesor investigador, posee varios artículos publicados con indexación regional y de alto impacto, ha tutorado tesis de grado y posgrado.



Jorge Alberto Medina Avelino

Máster Universitario En Desarrollo de Software, Magister en Docencia y Gerencia en Educación Superior, Estudiante del Programa Doctoral en Tecnologías de la Información y Comunicación, TICS de la Universidad de Granada, España. Profesor de la Universidad de Guayaquil. Docente Investigador Facultad de Ciencias Médicas, Director de Proyectos de Investigación en área de Informática Médica.



María Antonieta Touríz Bonifáz

Docente de la Universidad de Guayaquil, Docente de la Universidad de Católica Santiago de Guayaquil, Medico, Magister en Epidemiología, Magister en Salud Publica, Doctorante en Ciencias de la Salud, Docente Investigadora acreditada por la SENESCYT, ha participado en proyectos de investigación, ha tutorado varias tesis de grado de Educación Superior.



José Luis Rodríguez Matías

Profesor de la Universidad de Guayaquil y de la Universidad Católica Santiago de Guayaquil. Médico. Especialista en Neurocirugía. Especialista en Cirugía General. Docente Investigador con publicaciones regionales y de alto impacto. Director en proyecto de Investigación. Médico tratante del Hospital de especialidades Guayaquil.



Gladys Cristina Jácome Morales

Docente de la Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Físicas, Ecuatoriana, Ingeniera en Sistemas Computacionales, Magister en Sistemas de Información con mención en Inteligencia de Negocios, Profesora investigadora, posee varios artículos publicados con indexación regional y de alto impacto, ha tutorado varias tesis de grado en Instituciones de Educación Superior.

