

Santiago de Cali, 31 de agosto de 2021.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Luis Eduardo Tobon Llano

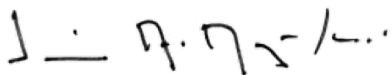
Director Carrera de Ingeniería Electrónica

Cali.

Cordial Saludo.

Por medio de la presente nos permitimos informarle que hemos revisado el trabajo de grado de Ingeniería Electrónica titulado **“Diseño de una aplicación como ayuda para predecir el riesgo de padecer una isquemia cardíaca”** del estudiante Alex Camayo Chagüendo y lo consideramos apto para ser presentado y sometido a consideración del jurado.

Atentamente,



---

Dr. Jaime Aguilar Zambrano  
Director



---

Dr. Helberg Asencio Santofimio  
Codirector

Santiago de Cali, 31 de agosto de 2021.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Luis Eduardo Tobon Llano

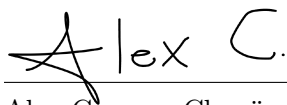
Director Carrera de Ingeniería Electrónica.

Cali.

Cordial Saludo.

Me permito presentar a su consideración el trabajo de grado titulado **“Diseño de una aplicación como ayuda para predecir el riesgo de padecer una isquemia cardíaca”** con el fin de optar por el título de Ingeniero Electrónico. De esa manera, espero que este trabajo de grado reúna todos los requisitos necesarios para su aprobación.

Atentamente,

Handwritten signature of Alex Camayo Chagüendo, consisting of a stylized star-like symbol followed by the text "Alex C.".

---

Alex Camayo Chagüendo

Código: 8923546



## **Acta de Correcciones al Proyecto de Grado de Ingeniería Electrónica**

**Fecha:** 20 de octubre de 2021

**Autor:** Alex Camayo Chaguendo

**Nombre del Proyecto de Grado:** Diseño de una aplicación como ayuda para predecir el riesgo de padecer una isquemia cardiaca.

**Director:** Jaime Aguilar Zambrano

**Codirector:** Helberg Asencio Santofimio.

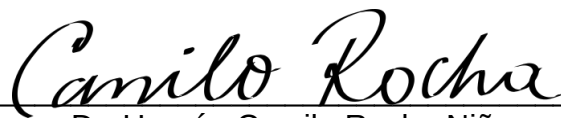
Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, hemos verificado que el estudiante indicado arriba ha implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de Director del Proyecto de Grado

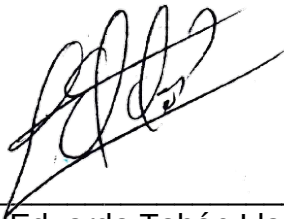
Firma de Codirector del Proyecto de Grado

Nota de Aceptación

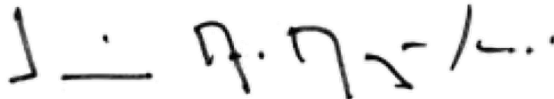
Aprobado por el Comité de Trabajo de Grado  
en cumplimiento de los requisitos exigidos por la  
Pontificia Universidad Javeriana para optar el  
título de Ingeniero Electrónico.



Dr. Hernán Camilo Rocha Niño  
Decano de la Facultad de Ingeniería



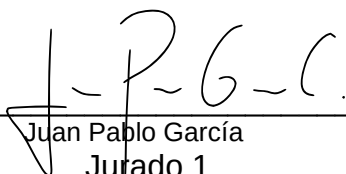
Dr. Luis Eduardo Tobón Llano  
Director Carrera Ingeniería Electronica.



Dr. Jaime Aguilar Zambrano  
Director(a) Trabajo



Dr. Helberg Asencio Santofimio  
Codirector(a) Trabajo

  
Juan Pablo García  
Jurado 1

  
Manuel Vicente Valencia  
Jurado 2



Pontificia Universidad  
**JAVERIANA**  
Cali

# Diseño de una aplicación como ayuda para predecir el riesgo de padecer una isquemia cardiaca

Autor:

Alex Camayo Chagüendo

# Diseño de una aplicación como ayuda para predecir el riesgo de padecer una isquemia cardiaca

Autor:

Alex Camayo Chagüendo

Trabajo de grado presentado como requisito parcial para optar por el título de:

**Ingeniero Electrónico**

Director:

Dr. Jaime Aguilar Zambrano

Codirector:

Dr. Helberg Asencio Santofimio

# Índice general

<b>Resumen</b>	I
<b>Agradecimientos</b>	II
<b>1. Introducción</b>	1
<b>2. Objetivos</b>	2
2.1. Objetivo general	2
2.2. Objetivos específicos	2
<b>3. Marco de referencia</b>	3
3.1. TRIZ (Teoría de problemas inventivos)	3
3.2. Sistema técnico	4
3.3. Sistema cardiovascular	5
3.4. Enfermedad isquémica del corazón (EIC)	8
3.4.1. Tipos de EIC	9
3.4.2. Causas y factores de riesgo	10
3.4.3. Signos, síntomas y complicaciones	11
3.4.4. Detección y diagnóstico	11
3.5. Métodos computacionales para predicción	12
3.5.1. Regresión Logística	13
3.5.2. Redes Neuronales Artificiales	13
3.5.3. Evaluación de los algoritmos para la predicción	14
3.6. Sensores	15
3.7. Microcontrolador	16
3.8. Software de aplicación	16
<b>4. Análisis</b>	18
4.1. Planteamiento del problema asociado con las enfermedades isquémicas del corazón	18
4.2. Formulación del problema	20
4.3. Justificación	20
4.4. Sistema técnico	21
4.5. Análisis de las nueve ventanas	22
4.5.1. Ventana 1 Sistema (Presente)	22
4.5.2. Ventana 2 Supersistema (Presente)	29
4.5.3. Ventana 3 Subsistema (Presente)	32
4.5.4. Ventana 4 Sistema (Pasado)	33
4.5.5. Ventana 5 Supersistema (Pasado)	34
4.5.6. Ventana 6 Subsistema (Pasado)	35

4.5.7. Ventana 7 Supersistema (Futuro)	36
4.5.8. Ventana 8 Subsistema (Futuro)	37
4.5.9. Ventana 9 Sistema (Futuro)	38
<b>5. Diseño de la aplicación para predecir el riesgo de padecer isquemia cardiaca</b>	<b>39</b>
5.1. Matriz de contradicciones	39
5.1.1. Contradicción 1	39
5.1.2. Contradicción 2	40
5.1.3. Contradicción 3	40
5.2. Requerimientos del producto	41
5.3. Parámetros de diseño para cada requerimiento funcional	42
5.3.1. Parámetros de diseño para el Requerimiento Funcional N°1	42
5.3.2. Parámetros de diseño para el Requerimiento Funcional N°2	44
5.3.3. Parámetros de diseño para el Requerimiento Funcional N°3	45
5.4. Descripción de los parámetros de diseño	47
5.4.1. Medición de la frecuencia cardiaca (DP 3)	47
5.4.2. Trasmisión de los datos de frecuencia cardiaca (DP 3)	49
5.4.3. Plataforma de desarrollo del sistema principal (DP 1 Y DP 2)	51
5.4.4. Plataforma para guardar la información del usuario (DP 1, DP 2 y DP 3)	52
5.4.5. Base de datos del algoritmo Machine Learning (DP 1)	53
5.4.6. Plataforma de desarrollo del algoritmo Machine Learning (DP 1)	55
5.4.7. Desarrollo del modelo Machine Learning en la plataforma seleccionada (FR1)	58
<b>6. Implementación de la aplicación como ayuda para predecir el riesgo de padecer una isquemia cardiaca</b>	<b>65</b>
6.1. Diseño del subsistema de medición de la frecuencia cardiaca	65
6.1.1. Diseño del esquemático	65
6.1.2. Diseño del contenedor de los elementos hardware	67
6.2. Diseño del sistema principal	68
6.2.1. Programación de las interfaces de la aplicación	69
<b>7. Evaluación del Hardware y Software</b>	<b>78</b>
7.1. Evaluación del hardware para la medición de FC	78
7.1.1. Análisis de demanda energética del dispositivo de medición de FC	79
7.2. Evaluación del sistema completo	80
7.2.1. Iniciar sesión y registrar un nuevo usuario	80
7.2.2. Predicción del riesgo de padecer una EIC	80
7.2.3. Recordatorios y recomendaciones	80
7.2.4. Registro de frecuencia cardiaca	81
7.2.5. Históricos y comunicación con otros usuarios	82
7.3. Resultados de la evaluación del sistema completo	82
7.4. Comparación entre el sistema actual y la aplicación realizada	83
7.5. Trabajo futuro	84
<b>8. Conclusiones</b>	<b>87</b>

<b>9. Anexos</b>	<b>94</b>
9.1. Anexo A: Código en Arduino IDE del ESP32	94
9.2. Anexo B: Android Studio	97
9.3. Anexo C: Repositorio en GitHub de la aplicación en Android Studio	99
9.4. Anexo D: SQLite	100
9.5. Anexo E: Principales rutinas en Firebase	102
9.6. Anexo F: Rutina para verificar estado Bluetooth y listar los dispositivos vinculados	105
9.7. Anexo G: Rutina para recibir y presentar la información que llega a través de Bluetooth	107
9.8. Anexo H: Algoritmo de prueba Machine Learning en Python	110

# Resumen

Hoy en día, las enfermedades cardíacas son las principales causas de muerte alrededor del mundo. Por esta razón, en este proyecto se realiza una aplicación que, no solamente pretende predecir el riesgo de padecer una enfermedad isquémica del corazón como prevención secundaria, sino que también, busca acompañar a los pacientes en su tratamiento de la enfermedad. Para lograr lo anterior, se realizó un proceso de diseño con el uso de la Teoría de Solución de Problemas Inventivos TRIZ, que es un conjunto de herramientas basadas en el conocimiento de la ciencia y la tecnología para el diseño de productos. De esa manera, se realizó una búsqueda de literatura científica para caracterizar los sistemas actuales que pretenden atacar la problemática de las enfermedades cardiovasculares, dentro de una línea de evolución tecnológica de manera temporal y espacial. Además, se investigaron los métodos que existen para la predicción de un nuevo episodio de la enfermedad. Así, se encontró que Machine Learning es una herramienta de gran uso y con buenos resultados para la predicción en términos de clasificación del riesgo de padecer una enfermedad isquémica del corazón. Sin embargo, se encontró una ausencia de involucrar estos algoritmos en productos de apoyo donde el paciente sea el centro de atención y no sólo la enfermedad. Así, la aplicación de este proyecto lleva el modelo de los estudios a un prototipo inicial que puede ser usado por las personas como un indicador en el tratamiento del padecimiento. Por otro lado, gracias al acompañamiento de potenciales usuarios, se plantearon requerimientos que debe tener la aplicación para contribuir a la adherencia de los pacientes al tratamiento. Como resultado, se obtuvo una aplicación en la plataforma Android que utiliza Tensorflow para el desarrollo del modelo Machine Learning, y Firebase junto con Android Studio para crear las interfaces que permitan crear adherencia hacia el tratamiento.

# Agradecimientos

*“Me volví y vi debajo del sol, que ni es de los ligeros la carrera, ni la guerra de los fuertes, ni aun de los sabios el pan, ni de los prudentes las riquezas, ni de los elocuentes el favor; sino que tiempo y ocasión acontecen a todos”*

---

*Eclesiastés*

Agradezco primero a Dios por permitirme llegar hasta esta instancia de mi vida. Igualmente, mi familia tiene todo el agradecimiento de mi parte por estar constantemente apoyándome durante los años que estuve lejos de casa. También, doy gracias a esa amiga que me acompañó y ayudó durante todo el tiempo que estuvimos en la carrera; tiene todo mi aprecio. Además, agradezco a todos los profesores de la Pontificia Universidad Javeriana de Cali que con sus palabras y ejemplo me enseñaron cosas importantes de la profesión y de la vida. En especial, agradezco a los profesores Jaime Aguilar y Helberg Asencio por destinar parte de su tiempo para el acompañamiento de este proyecto. Finalmente, doy reconocimiento a los autores, las guías y los creadores de contenido educativo en internet (videos, tutoriales, foros, etc.) que me permitieron realizar el proyecto de este documento.

# Capítulo 1

## Introducción

Las enfermedades relacionadas con el corazón junto con las cerebrovasculares y la respiratorias son las principales causas de muerte a nivel mundial [24]. Dentro del primer grupo de enfermedades se tiene a la isquemia cardiaca, la cual se presenta en las personas cuando el flujo sanguíneo al corazón se reduce, lo que impide que el músculo cardiaco reciba suficiente oxígeno [24]. Dicha reducción del flujo sanguíneo generalmente se produce debido a una obstrucción parcial o total de las arterias del corazón. Debido a lo anterior, el tratamiento para la isquemia cardiaca implica lograr mejorar el flujo sanguíneo al corazón. Tal tratamiento puede incluir medicamentos, un procedimiento para abrir las arterias bloqueadas (angioplastia) o cirugía de baipás [37]. Sin embargo, la detección a tiempo de la isquemia cardiaca juega un papel crucial en la salud de las personas, además de ser un factor importante que permite reducir los costos del tratamiento.

Actualmente, varios grupos de académicos y científicos han emprendido una búsqueda de patrones relevantes en los conjuntos de datos de personas que ya padecen o han padecido una isquemia cardiaca (datos como la edad, el género, la frecuencia cardiaca, etc.). Para lograr dicho objetivo, se utilizan conceptos como Machine Learning y redes neuronales artificiales [28]. De esta manera, este trabajo de grado busca diseñar un producto novedoso, a partir del uso y enriquecimiento de sistemas, algoritmos o aplicaciones actuales que utilizan los conceptos presentados anteriormente, que le permita a los especialistas y pacientes conocer con anticipación si hay riesgo de padecer una isquemia cardiaca en ámbitos de prevención secundaria (evitar que un episodio grave de la enfermedad vuelva a ocurrir). Para lograr el diseño de este producto, se utiliza la metodología TRIZ, la cual plantea principios universales de invención para productos, los cuales hacen que los procesos inventivos sean más predecibles, además de servir como base para obtener innovaciones a favor de la tecnología, y en este caso, de la salud [1]. Así, se pretende ofrecer una aplicación móvil que ayude al cuidado de la población con isquemia cardiaca, cumpliendo con las necesidades del usuario.

# Capítulo 2

## Objetivos

### 2.1. Objetivo general

Diseñar un producto de apoyo que sea capaz de determinar la predicción del riesgo de padecer una isquemia cardiaca en personas, utilizando la metodología TRIZ, para la prevención secundaria de la enfermedad.

### 2.2. Objetivos específicos

- Caracterizar el procedimiento y la funcionalidad de algunas aplicaciones, métodos o sistemas actuales que ayuden a la predicción del riesgo de padecer una isquemia cardiaca.
- Definir y establecer los requerimientos del producto propuesto, basándose en las necesidades del usuario, ya sea paciente o especialista.
- Diseñar, construir y evaluar el prototipo del producto basado en Machine Learning que ayude a mejorar la predicción del riesgo de padecer una isquemia cardiaca y que sea accesible por las personas a quien va dirigido.

# Capítulo 3

## Marco de referencia

### 3.1. TRIZ (Teoría de problemas inventivos)

TRIZ es una teoría desarrollada por Genrich Altshuller en 1946 [1], la cual consiste en un conjunto de herramientas basadas en el conocimiento de la ciencia y la tecnología. TRIZ plantea principios universales de invención para productos, los cuales hacen que los procesos inventivos sean más predecibles, además de servir como base para obtener innovaciones a favor de la tecnología teniendo en cuenta aspectos de **idealidad**, **contradicción** y **evolución de los sistemas técnicos** [2].

Para el proceso de diseño se utilizan principalmente dos técnicas, la **técnica de las nueve ventanas** y la **matriz de contradicciones**. La primera (figura 3.1) permite realizar un análisis del problema de forma sistémica, a partir de un sistema técnico que cumpla una función principal; este se proyecta al futuro de manera que en el límite alcance la idealidad [1]. En la aplicación de la técnica se considera el sistema, su entorno (súper-sistema) y sus componentes (sub-sistema) tanto en el presente como en el pasado, lo cual resulta en nueve componentes de análisis. El futuro es concebido por el equipo de diseño en la búsqueda de la idealidad del sistema. De esta manera, se obtienen patrones de evolución del producto esperado, y se detectan ciertas contradicciones técnicas que desde el presente le impiden llegar a cumplir esa idealidad hipotética del futuro.



Figura 3.1: Técnica de las nueve ventanas

Luego de resolver la técnica de las nueve ventanas, se deben detectar contradicciones (estas se encuentran en el análisis de las ventanas 1 y 9 de las nueve ventanas) en el sistema cuando dos parámetros técnicos sean opuestos, es decir, cuando se dé el caso de que al mejorar una característica del sistema se desmejore otra. Así, para resolver este dilema, TRIZ plantea 40 soluciones o principios de invención que se aplican en general a cualquier problema. Para ello se usa la segunda técnica del proceso de diseño, la matriz de contradicciones (figura 3.2), en la cual se muestran los 40 principios inventivos para una solución de los parámetros en conflicto [3]. Luego, el diseñador evalúa los principios sugeridos en la matriz de contradicciones y utiliza el principio o los principios que den la mejor solución a la problemática planteada.

Parametros		1	2	3	4	35
1	Peso del Objeto Móvil,	+		15, 8, 29, 34		29, 5, 15, 8
2	Peso del Objeto Estacionario,		+		10, 1, 29, 35	19, 15, 29
3	Longitud del Objeto Móvil,	1, 15, 29, 34		+		14, 15, 1, 16
4	Longitud del Objeto Estacionario,		35, 28, 40, 29		+	1, 35
	...					

Figura 3.2: Matriz de contradicciones [1]

### 3.2. Sistema técnico

Para la utilización de TRIZ se debe ver la problemática a partir de un sistema técnico, que se define como la representación de un producto (sistema) que toma la energía del exterior para realizar una función principal sobre un objeto. El sistema se modela con cuatro componentes (figura 3.3):

- **Motor:** Parte del sistema que convierte la energía en otra para asegurar la función principal.
- **Transmisión:** Conduce la energía desde el motor hacia la herramienta.
- **Herramienta:** Asegura el contacto entre el sistema y el objeto donde se realiza la función principal.
- **Control:** Reacciona a los cambios del sistema adaptándose automáticamente a su forma, estructura o salida.

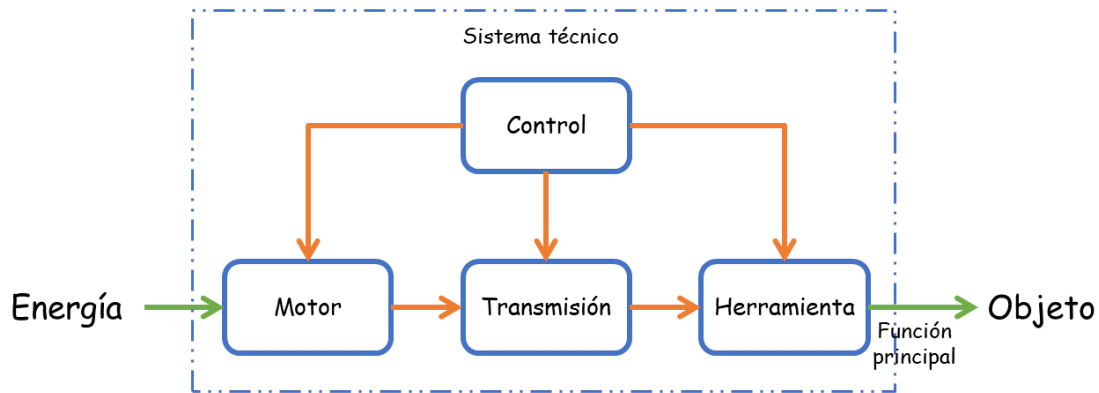


Figura 3.3: Sistema técnico

### 3.3. Sistema cardiovascular

El sistema cardiovascular está compuesto por el corazón y una red de arterias, venas y capilares que llevan la sangre por todo el cuerpo. El hombre adulto en promedio tiene entre 5 y 6 litros de sangre, mientras que la mujer tiene entre 4 y 5 litros. Esta sangre lleva oxígeno y nutrientes esenciales a todas las células vivas del cuerpo, y también lleva desechos de los tejidos a los sistemas del organismo a través de los cuales son eliminados. Igualmente, el sistema cardiovascular se encarga de transportar células sanguíneas y sistemas de defensa por todo el organismo [4]. A continuación, se describen los 3 elementos más importantes del sistema cardiovascular:

- El corazón:** Este es una bomba (aspirante-impelente) muscular que proporciona la energía para mover la sangre por los vasos sanguíneos. El corazón pesa entre 200 a 425 gramos y es un poco más grande que una mano cerrada. Cada día, en promedio el corazón late 100.000 veces, bombeando aproximadamente 7.571 litros de sangre [5]. En la figura 3.4 se observa la composición interna del corazón, donde generalmente las venas se ilustran con color azul y las arterias con color rojo. Además, el corazón tiene paredes, hechas de músculo, que se comprimen (contraen) para bombear la sangre cardiovascular hacia los vasos sanguíneos y por todo el cuerpo. Así, las venas transportan la sangre desoxigenada a través del sistema cardiovascular hacia el lado derecho del corazón. Mientras que las arterias transportan sangre oxigenada hacia todos los órganos del cuerpo humano. [6].

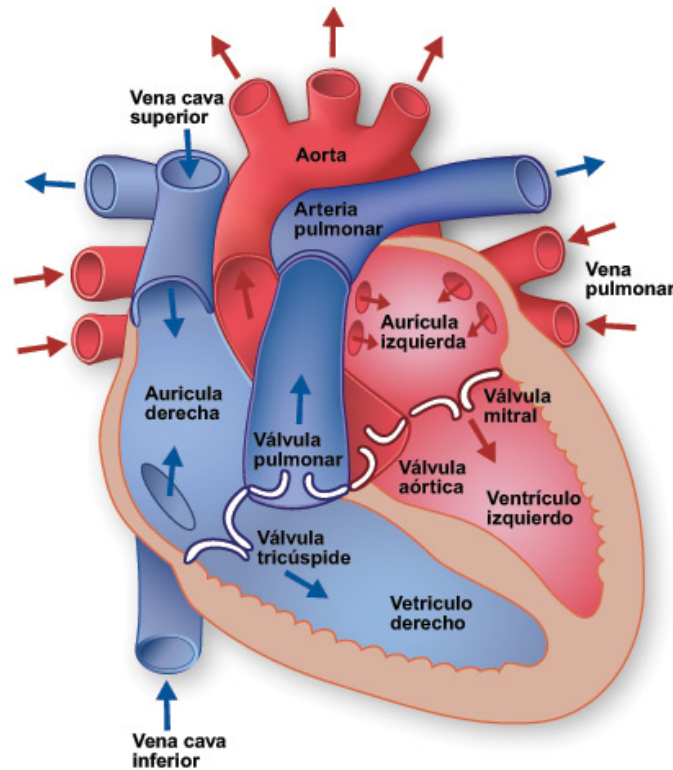


Figura 3.4: El corazón humano [5]

- **La sangre:** Es el contenido o tejido líquido que circula por los vasos, venas y arterias, consistiendo en un 80 % de agua y un 20 % de sustancias sólidas [8]. Los componentes principales de la sangre son el oxígeno y nutrientes, que son transportados a los tejidos, además de los desechos que ya no necesita el cuerpo y que se transportan también a través del sistema vascular [6]. De igual forma, en la sangre se encuentran otro tipo de células como los glóbulos rojos, glóbulos blancos y las plaquetas. A continuación, se explica cada una de ellas.

  - **Glóbulos rojos:** Son encargados del transporte de oxígeno. De los 3 tipos de células sanguíneas, los glóbulos rojos son las más numerosas. Los glóbulos rojos también se denominan eritrocitos [7].
  - **Glóbulos blancos:** Son los elementos que aportan en el proceso de combatir infecciones. Cuando el organismo combate una infección, aumenta su producción de estas células. Aun así, comparado con el número de glóbulos rojos, el número de glóbulos blancos es bajo. Los glóbulos blancos también se denominan leucocitos [7].
  - **Plaquetas:** Son los elementos que intervienen en el proceso de coagulación sanguínea. La coagulación detiene el flujo de sangre fuera del cuerpo cuando se rompe una vena o una arteria. Si el número de plaquetas es demasiado bajo, puede ocasionar una hemorragia excesiva. Por otra parte si el número de plaquetas es demasiado alto, pueden formarse coágulos sanguíneos y ocasionar trombosis. Las plaquetas también se denominan trombocitos [7].
- **Vasos sanguíneos:** son las arterias, las venas y los capilares (vasos sanguíneos pequeños) que conforman el sistema de tubos elásticos del cuerpo humano por donde circula la sangre. Las

arterias son vasos que distribuyen la sangre del corazón a los tejidos. Las arterias se ramifican y progresivamente en cada ramificación disminuye su calibre y se forman las arteriolas. En el interior de los tejidos las arteriolas se ramifican en múltiples vasos microscópicos, los capilares, que se distribuyen entre las células. Ahora bien, los capilares se unen en grupos formando venas pequeñas, llamadas vénulas, que se fusionan para dar lugar a venas de mayor calibre [9]. Por último, las venas retornan la sangre menos oxigenada al corazón.

Es importante mencionar que el corazón, como cualquier otro órgano o tejido necesita sangre para poder funcionar. Dicha sangre la recibe de su propio aparato vascular. Lo anterior es denominado “circulación coronaria”. La aorta (el principal conducto de suministro de sangre del organismo) se ramifica en dos vasos sanguíneos coronarios principales (también denominados arterias). Estas arterias coronarias se ramifican a su vez en arterias más pequeñas que suministran sangre rica en oxígeno a todo el músculo cardíaco. La arteria coronaria izquierda, que se ramifica en la arteria descendente anterior izquierda y la arteria circunfleja, suministra sangre al lado izquierdo del corazón. En la figura 3.5 se ilustra la aorta y algunas arterias que rodean al corazón [10].

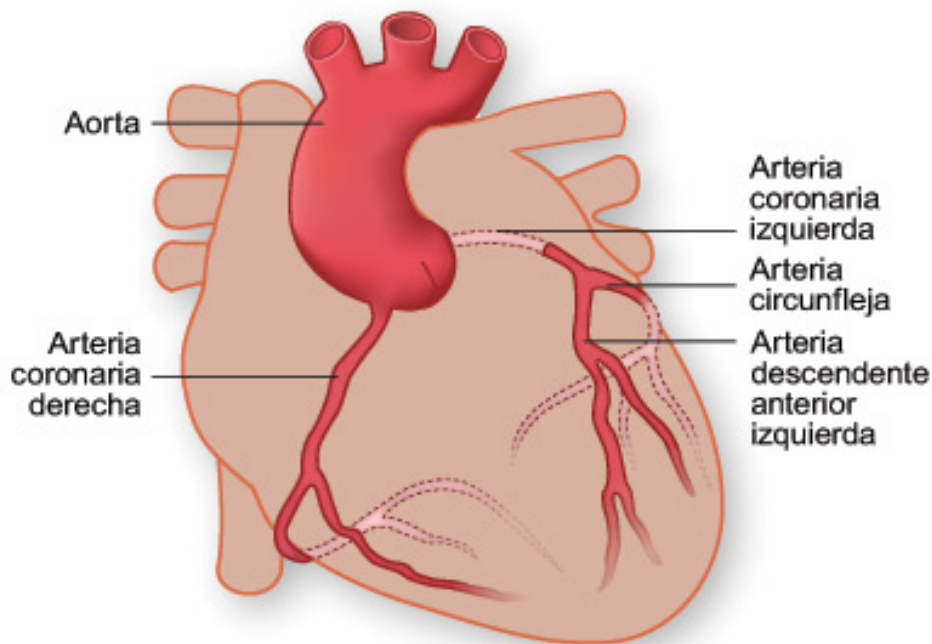


Figura 3.5: Arterias coronarias [10]

El funcionamiento del sistema cardiovascular es el siguiente: todos los seres humanos respiran aire por la boca y la nariz para que llegue a los pulmones (donde los alvéolos, que son sacos microscópicos revestidos de vasos sanguíneos, realizan el intercambio de los gases de oxígeno y dióxido de carbono) y de ahí al sistema cardíaco. Luego, el oxígeno del aire es absorbido por el torrente sanguíneo a través de los pulmones. Cuando llega a los pulmones empieza la función del aparato cardiovascular, ya que es ahí donde la sangre no oxigenada se oxigena y regresa al corazón. Es entonces cuando en el sistema cardiovascular, el corazón bombea la sangre oxigenada, y en cada latido la hace fluir a

través de la red de vasos sanguíneos (las arterias) y ramas hasta llegar a los tejidos, incluyendo los órganos, músculos y nervios de todo el cuerpo [6].

Ahora bien, cuando en el sistema cardiovascular la sangre llega a los capilares en los tejidos se libera el oxígeno, que utilizan las células para producir energía. Estas células liberan los productos de desecho, como el dióxido de carbono y agua, que son absorbidos y transportados por la sangre. La sangre usada (o "desoxigenada") viaja entonces el sistema vascular a través de las venas y de regreso hacia el corazón. El corazón bombea entonces la sangre desoxigenada de nuevo a los pulmones, desde donde se deshecha y se absorbe el oxígeno fresco, haciendo que el ciclo del sistema cardiovascular comience nuevamente [6]. En la figura 3.6 se observa una ilustración del sistema cardiovascular. En dicha imagen se puede observar dos de los componentes principales: el corazón y el nombre de los vasos sanguíneos importantes.

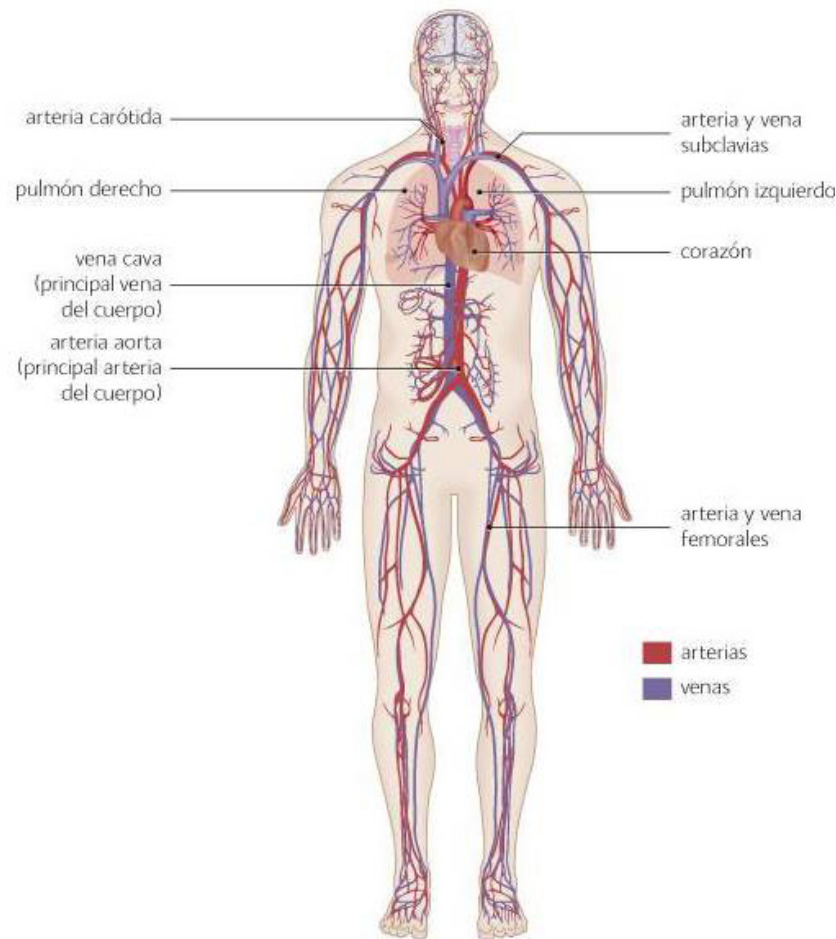


Figura 3.6: Sistema cardiovascular [6]

### 3.4. Enfermedad isquémica del corazón (EIC)

La enfermedad del corazón es un término muy común para una variedad de condiciones que afectan la estructura y función del corazón que a su vez afecta al sistema cardiovascular. La enfermedad isquémica del corazón (EIC) ocurre cuando las arterias del corazón no pueden suministrar suficiente

sangre con oxígeno al corazón [24]. La EIC (también conocida como **enfermedad coronaria microvascular, enfermedad de la arteria coronaria, síndrome coronario X, enfermedad coronaria, enfermedad coronaria no obstructiva, enfermedad obstructiva de la arteria coronaria**), es la principal causa de muerte en los Estados Unidos [11] y en Colombia [21], con la mayoría de las muertes por enfermedad coronaria, también conocida como enfermedad arterial coronaria.

La enfermedad coronaria es causada por la acumulación de placa, una sustancia cerosa, dentro de las arterias coronarias. Dicha acumulación puede bloquear de forma parcial o total el flujo sanguíneo en las grandes arterias del corazón (arteria aorta o sus derivaciones, por ejemplo) (figura 3.7). Algunos tipos de esta afección pueden ser causados por una enfermedad o lesión que afecta el funcionamiento de las arterias en el corazón. La enfermedad coronaria microvascular es otro tipo de EIC. Esta última ocurre cuando las diminutas arterias del corazón no funcionan normalmente. Luego, los síntomas de la EIC pueden ser diferentes de una persona a otra, incluso si tienen el mismo tipo de EIC. Sin embargo, debido a que muchas personas no presentan síntomas, no se percatan que tienen una EIC hasta que experimentan complicaciones como un ataque al corazón o un paro cardíaco súbito [11].

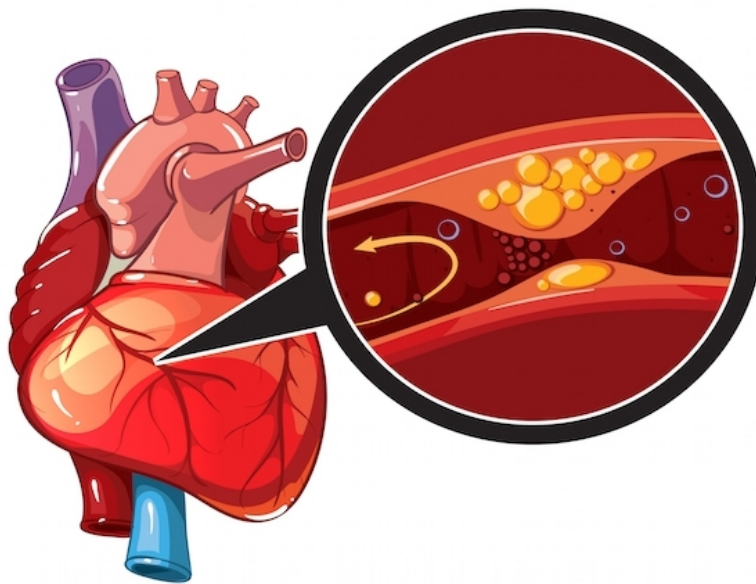


Figura 3.7: Ilustración de una EIC [18]

### 3.4.1. Tipos de EIC

Según *National Heart, Lung and Blood Institute* [11] hay tres tipos principales de enfermedad isquémica del corazón:

- **Enfermedad arterial coronaria obstructiva:** Esta ocurre cuando la placa se acumula en las arterias grandes, lo que hace que se estrechen gradualmente. Lo anterior hace que se reduzca el suministro de sangre enriquecida en oxígeno al corazón. La enfermedad arterial coronaria obstructiva significa que las arterias del corazón están bloqueadas por encima del 50%.
- **Enfermedad arterial coronaria NO obstructiva:** las arterias grandes se estrechan por

la placa, pero no tanto como en la enfermedad obstructiva. Se diagnostica si los estudios de imágenes muestran menos del 50 % de obstrucción de las grandes arterias del corazón. Entre el 20 % y el 50 % de las personas que se someten a estudios de imágenes del corazón padecen de la enfermedad arterial coronaria no obstructiva.

- **Enfermedad coronaria microvascular:** Esta afecta a las arterias más pequeñas del corazón. La enfermedad coronaria microvascular puede ocurrir sola o con la enfermedad arterial coronaria obstructiva o no obstructiva, u otras enfermedades del corazón. Este tipo de enfermedad del corazón ocurre con mayor frecuencia cuando los cambios moleculares en el sistema microvascular (los diminutos vasos sanguíneos) del corazón impiden el flujo normal de sangre a través de las arterias pequeñas [11].

### 3.4.2. Causas y factores de riesgo

La EIC puede tener más de una causa, incluida la acumulación de placa o cambios moleculares en los vasos sanguíneos. En el primer caso, la acumulación de placa en las arterias se llama **aterosclerosis**. Si dicha acumulación perdura en las arterias por varios años, tales arterias se vuelven más estrechas y se endurecen, reduciendo el flujo de sangre al corazón. Lo anterior da como resultado la enfermedad arterial coronaria. También, se pueden adherir pequeños trozos de placa o coágulos sanguíneos a las arterias pequeñas, lo que causa la enfermedad coronaria microvascular. Esto puede suceder cuando los pedazos de placa o coágulos de sangre se desprenden durante un procedimiento médico o quirúrgico que luego viajan a las arterias pequeñas [11].

Ahora bien, no están completamente claras las causas de la enfermedad coronaria microvascular, pero pueden implicar cambios moleculares en los pequeños vasos del corazón que forman parte del proceso normal de envejecimiento. También, dicha enfermedad se puede presentar debido a daños en los pequeños vasos sanguíneos causados por inflamación crónica, presión arterial alta o diabetes. Igualmente, los cambios moleculares afectan la forma en que los genes y las proteínas se controlan dentro de las células. Estos cambios significan que las pequeñas arterias del corazón pueden no responder correctamente cuando reciben señales que indican que el corazón necesita más sangre rica en oxígeno (al hacer actividad física, por ejemplo). Así que, en lugar de expandirse para permitir un mayor flujo de sangre, el tamaño de estas arterias no cambia o incluso puede disminuir [11]. En la siguiente tabla, se resumen los factores de riesgos de una EIC más destacados según *National Heart, Lung and Blood Institute*:

FACTORES DE RIESGO	IMPLICACIONES
Hábitos de estilo de vida poco saludables	Provocan la acumulación de placa en los vasos sanguíneos del corazón. <ul style="list-style-type: none"> <li>• Estar físicamente inactivo</li> <li>• Fumar o estar expuesto prolongadamente al humo directo</li> <li>• Estrés</li> <li>• Alimentación no saludable</li> </ul>
Edad y sexo	La enfermedad arterial coronaria obstructiva es más común entre los hombres mientras que en las mujeres es más común la NO obstructiva. En los hombres, el riesgo de enfermedad isquémica del corazón comienza a aumentar alrededor de los 45 años. Después de alrededor de los 55 años, el riesgo de una EIC aumenta al mismo ritmo tanto en mujeres como en hombres.
Medio ambiente y profesión	La contaminación del aire puede causar o empeorar otras condiciones, como la aterosclerosis y la presión arterial alta que a su vez aumenta el riesgo de una EIC. Trabajar más de 55 horas por semana también se asocia con un mayor riesgo de EIC.
Genética	Un antecedente familiar de enfermedad cardiovascular temprana es un factor de riesgo para la EIC. Esto es especialmente cierto si el padre o hermano fue diagnosticado antes de los 55 años, o si la madre o la hermana fue diagnosticada antes de los 65 años. Investigaciones muestran que algunos genes están relacionados con un mayor riesgo de enfermedad isquémica del corazón.
Raza o etnia	La EIC es la principal causa de muerte para las personas de la mayoría de los grupos raciales y étnicos en los Estados Unidos. Para los estadounidenses de origen asiático, isleños del Pacífico, indios americanos o nativos de Alaska, las enfermedades del corazón son superadas por el cáncer.
Otros	Otras afecciones médicas que pueden aumentar el riesgo de desarrollar una EIC incluyen: aterosclerosis, enfermedades inflamatorias y autoinmunes, enfermedad renal crónica, defectos congénitos del corazón, diabetes, nivel alto de colesterol LDL, tensión arterial alta, triglicéridos elevados en la sangre, VIH, síndrome metabólico, sobrepeso y obesidad, e incluso, trastornos de sueño.

Figura 3.8: Factores de riesgo [\[1\]](#)

### 3.4.3. Signos, síntomas y complicaciones

Los signos variarán según el tipo de EIC que la persona tenga. Algunos pacientes informan síntomas severos de EIC, pero otros no presentan ningún signo o síntoma. Si existe una EIC “silenciosa”, es posible que la persona no experimente ningún síntoma hasta que tenga complicaciones, como eventos coronarios agudos, incluido un ataque al corazón. Ahora bien, un evento coronario agudo, como un ataque al corazón, puede causar síntomas como:

- Angina, que puede sentirse como presión, compresión, ardor o tensión durante la actividad física. También, se siente como el dolor o la molestia generalmente comienza detrás del esternón, pero también puede ocurrir en los brazos, los hombros, la mandíbula, la garganta o la espalda.
- Sudores fríos, mareos, aturdimiento, náuseas o sensación de indigestión, dolor de cuello, falta de aliento, especialmente con la actividad, trastornos del sueño y debilidad.

### 3.4.4. Detección y diagnóstico

Las pruebas de detección y las evaluaciones de riesgo para el desarrollo de la EIC deben comenzar alrededor de los 20 años para las personas que no tienen ningún factor de riesgo de EIC. Para evaluar la EIC, el médico determinará si se tiene algún factor de riesgo conocido, evaluará los niveles de colesterol y de azúcar en la sangre, y la presión arterial. Luego, el médico puede usar un *sistema de*

*cálculo de riesgo* para determinar si está en un grupo de riesgo bajo, intermedio o alto. Así mismo, el médico puede recomendar tratamientos como cambios para alcanzar un estilo de vida saludable para el corazón con el fin de ayudar a prevenir la EIC [11]. Así, el médico diagnosticará la EIC según los síntomas, la historia médica y familiar, los factores de riesgo y los resultados de las pruebas y procedimientos. Debido a que las mujeres y los médicos pueden no reconocer los síntomas de la EIC, que son diferentes a los de los hombres, es posible que estas no obtengan un diagnóstico o tratamiento tan rápido como los hombres [11]. Luego, para diagnosticar una EIC, el médico puede ordenar las siguientes pruebas [11]:

- Exámenes de sangre: Para verificar los niveles de colesterol, triglicéridos, azúcar, lipoproteínas o proteínas que son marcadores de inflamación.
- Electrocardiograma (ECG): Este registra las señales eléctricas del corazón de manera NO invasiva. Es una prueba frecuente que se utiliza para detectar problemas cardiacos y controlar el estado del corazón en muchas situaciones.
- Pruebas de esfuerzo: Para verificar cómo funciona el corazón durante el esfuerzo físico. Durante las pruebas de esfuerzo, el paciente camina o corre en una caminadora o pedalea una bicicleta estacionaria para hacer que el corazón trabaje con fuerza y palpite rápidamente.
- Imágenes de resonancia magnética del corazón (IRM): Para detectar daños en los tejidos o problemas con el flujo de sangre en el corazón o las arterias coronarias. Puede ayudar al médico a diagnosticar una enfermedad coronaria microvascular o una enfermedad arterial coronaria obstructiva o no obstructiva
- Angiografía coronaria: Es un procedimiento en el que se utiliza un tinte especial (material de contraste) y rayos X para observar la forma en que fluye la sangre a través de las arterias en el corazón. La angiografía estándar no se puede usar para diagnosticar la enfermedad coronaria microvascular, porque los diminutos vasos sanguíneos son demasiado pequeños para visualizarlos.

### 3.5. Métodos computacionales para predicción

El avance de la computación en las últimas décadas ha permitido el desarrollo de software para la predicción de eventos futuros a partir de información actual. La ciencia de datos, por ejemplo, combina múltiples campos, como las estadísticas, los métodos científicos, la inteligencia artificial (IA) y el análisis de información para extraer el valor de los datos. Además, la ciencia de datos abarca la preparación de los datos para el análisis, incluida la limpieza, la agregación y la manipulación de los datos para realizar análisis avanzados. Las aplicaciones analíticas y los científicos de datos pueden revisar los resultados para descubrir patrones y permitir que las empresas, hospitales, universidades, etc., obtengan información fundamentada de eventos futuros [12]. Además, por el lado de la inteligencia artificial (IA), en términos sencillos, se refiere a los sistemas o las máquinas que imitan la inteligencia humana para realizar tareas y que tienen la capacidad de mejorar iterativamente a partir de la información que recopilan. Su objetivo es mejorar significativamente las capacidades y contribuciones humanas. Eso la convierte en un activo comercial muy valioso [13].

### 3.5.1. Regresión Logística

La regresión logística es un método de regresión que permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. Una de las principales aplicaciones de la regresión logística es la de clasificación binaria, en el que las observaciones se clasifican en un grupo u otro dependiendo del valor que tome la variable empleada como predictor. Por ejemplo, clasificar a un individuo desconocido como hombre o mujer en función del tamaño de la mandíbula. Ahora bien, si una variable cualitativa con dos niveles se codifica como 1 y 0, matemáticamente es posible ajustar un modelo de regresión lineal por mínimos cuadrados  $Y = ax + b$ . El problema de esta aproximación es que, al tratarse de una recta, para valores extremos del predictor, se obtienen valores de  $Y$  menores que 0 o mayores que 1, lo que entra en contradicción con el hecho de que las probabilidades siempre están dentro del rango  $[0,1]$ . Para evitar estos problemas, la regresión logística transforma el valor devuelto por la regresión lineal  $ax + b$  empleando una función cuyo resultado está siempre comprendido entre 0 y 1 [14]. Existen varias funciones que cumplen esta descripción, una de las más utilizadas es la función logística (también conocida como función sigmoide):

$$\textit{sigmoid} = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.1)$$

En la ecuación anterior para valores de  $x$  muy grandes positivos, el valor de  $e^{-x}$  es aproximadamente 0 por lo que el valor de la función sigmoide es 1. Para valores de  $x$  muy grandes negativos, el valor  $e^{-x}$  tiende a infinito por lo que el valor de la función sigmoide es 0 [15].

### 3.5.2. Redes Neuronales Artificiales

Las redes neuronales artificiales son un modelo computacional que consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal produciendo unos valores de salida. Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral (activación), que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona [16].

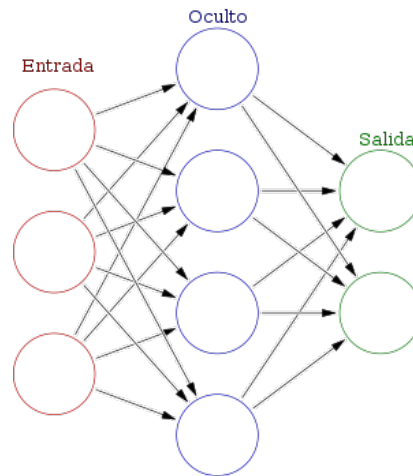


Figura 3.9: Red Neuronal Artificial

Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional. Para realizar este aprendizaje automático, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su total. Los valores de los pesos de las neuronas se van actualizando, buscando reducir el valor de la función de pérdida. Este proceso se realiza mediante la propagación hacia atrás. El objetivo de la red neuronal es resolver los problemas de la misma manera que el cerebro humano, aunque las redes neuronales son más abstractas. Las redes neuronales actuales suelen contener desde unos miles a unos pocos millones de unidades neuronales. Además, las redes neuronales se han utilizado para resolver una amplia variedad de tareas, como la visión computacional y el reconocimiento de voz, que son difíciles de resolver usando la ordinaria programación basado en reglas [16].

### 3.5.3. Evaluación de los algoritmos para la predicción

En términos de clasificación, se puede evaluar la cantidad de desaciertos y aciertos de un algoritmo a través de una **matriz de confusión**. Una matriz de confusión es una herramienta que permite visualizar el desempeño de un algoritmo para predecir cierto valor a partir de cierta información (está nublado, hace frío y hay mucha humedad, entonces va a llover). Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. En términos prácticos, esta matriz nos permite ver qué tipos de aciertos y errores está teniendo un modelo de predicción. Por ejemplo, se hace la prueba de un modelo supervisado de clasificación binaria ML y se obtuvo, que para 10 muestras, el modelo predijo que las 7 primeras de ellas son positivas y las 3 restantes son negativas. Pero en realidad, las primeras 4 de ellas son positivas y las 6 restantes son negativas. Así, para el ejemplo anterior, según la matriz de la figura 3.10, se tienen 4 verdaderos positivos (VP), 3 falsos positivos (FP) y 3 verdaderos negativos (VN) y 0 falsos negativos (FN).

		Predicción	
		Positivos	Negativos
Reales	Positivos	Verdaderos Positivos	Falsos Negativos
	Negativos	Falsos Positivos	Verdaderos Negativos

Figura 3.10: Matriz de confusión

Además, la matriz de confusión permite obtener métricas importantes para evaluar el desempeño del modelo, teniendo en cuenta los valores reales y los valores predichos modelo de predicción. Dichas métricas son la precisión, la sensibilidad, la especificidad y el error de clasificación:

- **Precisión:** Es la cantidad de aciertos verdaderos del modelo. Su fórmula está dada por:

$$Precision = \frac{VP + VN}{VP + FP + FN + VN} \quad (3.2)$$

- **Sensibilidad:** Es el cociente entre el total de verdaderos positivos sobre el total de muestras positivas reales:

$$Sensibilidad = \frac{VP}{VP + FN} \quad (3.3)$$

- **Especificidad:** Esta es similar a la sensibilidad. Sin embargo, la especificidad se define como la relación entre la cantidad verdaderos negativos y la cantidad total de muestras reales negativas:

$$Especificidad = \frac{VN}{FP + VN} \quad (3.4)$$

- **Error de clasificación:** Es el error general de la clasificación hecha por el modelo:

$$Error = \frac{FP + FN}{VP + FP + FN + VN} \quad (3.5)$$

### 3.6. Sensores

Los sensores son dispositivos electrónicos que permiten captar información del medio físico, es decir, son capaces de detectar acciones o estímulos externos. Estos dispositivos convierten magnitudes físicas (luz, presión, temperatura, movimiento, etc) en valores medibles de dicha magnitud. El proceso que lleva a cabo internamente un sensor se puede resumir en:

- Recepción de datos.
- Procesamiento de la magnitud física y para convertirla en una variable eléctrica.
- Salida de señales por medio cableado o inalámbrico hacia un sistema computacional.

Actualmente, con el avance de la tecnología, se encuentra todo tipo de sensores para muchas variedades de magnitudes físicas. Entre ellos, se encuentran los ultrasónicos, humedad, velocidad, presión, temperatura, sonido, contacto, movimiento, entre otros.

### 3.7. Microcontrolador

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida. Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia. Por lo general, este elemento tendrá la capacidad de mantenerse a la espera de un evento como pulsar un botón o de otra interrupción. De esa manera, el consumo de energía durante el estado de reposo (reloj de la CPU y los periféricos de la mayoría) puede ser solo de nanowatts, lo que hace que muchos de ellos sean muy adecuados para aplicaciones con batería de larga duración. Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal, con velocidades de reloj y consumo de energía más altos.

Cuando es fabricado el microcontrolador, no contiene datos en la memoria ROM. Para que pueda controlar algún proceso es necesario generar o crear y luego grabar en la EEPROM o equivalente del microcontrolador algún programa, el cual puede ser escrito en lenguaje ensamblador u otro lenguaje para microcontroladores. Sin embargo, para que el programa pueda ser grabado en la memoria del microcontrolador, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando este es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento [19].

### 3.8. Software de aplicación

Una aplicación es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas. Generalmente, son diseñadas para facilitar ciertas tareas complejas y hacer más sencilla la experiencia informática de las personas. Las aplicaciones son programas totalmente distintos a los sistemas operativos o lenguajes de programación, ya que estas cumplen una función puntual y es pensada para que sea de uso común como por ejemplo manejar finanzas o contabilidad desde hojas de cálculo. Los equipos tecnológicos (celular, computador, tableta, relojes inteligentes, entre otros) tienen la facilidad de traer aplicaciones pre-instaladas o la posibilidad de instalarlas. Una de las formas de conseguir una aplicación es por medio de las tiendas de aplicaciones.

Según el dispositivo, se encontrará una tienda para descargarlas e instalarlas. Por ejemplo, existe la Tienda Microsoft, para equipos con sistema operativo de esa compañía; la AppStore para equipos con sistema operativo diseñado por Apple; o PlayStore, si el dispositivo usa el sistema operativo Android, entre otros. Además, existen cientos de empresas dedicadas a crear aplicaciones y venderlas en forma de paquete, también conocido como suites, como por ejemplo Microsoft Office o Adobe CS. Con el avance de la tecnología, es común escuchar el término “aplicaciones móviles”. Estas hacen referencia a las aplicaciones diseñadas para dispositivos como el celular o la tableta. A las aplicaciones de la computadora se les conoce como aplicaciones de escritorio [20].

## Capítulo 4

# Análisis

### 4.1. Planteamiento del problema asociado con las enfermedades isquémicas del corazón

En Colombia existen tres enfermedades que son las principales causas de muerte en la nación. Según el DANE, desde enero de 2016 hasta agosto de 2017 se registraron en el país alrededor de 355.582 defunciones no fetales, es decir, los fallecimientos ocurridos después de nacer. Dichas enfermedades son: las isquémicas del corazón, con 60.944 casos (17.1 %); las cerebrovasculares, con 24.548 casos (6.9 %), y las enfermedades crónicas de las vías respiratorias inferiores, con 22.709 casos (6.3 %). El Valle del Cauca (7488 casos), Córdoba (4083 casos), Antioquia (7810 casos), Cundinamarca (3151 casos) y Tolima (3196 casos) son los departamentos con más porcentaje de defunción debido a las enfermedades isquémicas del corazón [21].

De igual forma, según la Organización Mundial de la salud, se prevé que para el 2030, 18 millones de muertes alrededor del mundo serán por enfermedades cardiovasculares [22]. Debido a la relevancia de este tema, se consideró necesario trabajar en este problema desde la ingeniería con el apoyo de las ciencias de la salud. Además, en un estudio de la Fundación Santa Fe de Bogotá, en el que se analizaron los costos médicos directos registrados entre 2017 y 2018, se encontró que enfermedades como un infarto, podrían costarle al sistema de salud entre 6 y 12 millones de pesos, por paciente [23]; lo anterior perjudica tanto las finanzas públicas como la calidad de vida de las personas. Además, los estilos de vida actuales, en los que se camina poco, se trabaja más, se duerme menos y se manejan mayores niveles de estrés son factores de riesgo determinantes para contraer estas enfermedades y otras de tipo cardiovascular. Esto muestra que últimamente las personas no se interesan por la salud de su corazón, lo cual aumenta la probabilidad de contraer enfermedades isquémicas del corazón.

Las actividades físicas y buena alimentación son aspectos importantes para contrarrestar la propagación a nivel nacional de este tipo de enfermedades no transmisibles. En ese mismo sentido, es importante mencionar que el mismo estilo de vida que puede ayudar a tratar la isquemia cardiaca también puede ayudar a evitar que se presente por primera vez. Las arterias pueden conservarse fuertes, elásticas y uniformes y permitir un flujo sanguíneo máximo si se lleva un estilo de vida saludable para el sistema cardiaco. Sin embargo, cuando se padece alguna isquemia en el corazón, el tratamiento puede incluir medicamentos, un procedimiento para abrir las arterias bloqueadas (angioplastia) o

cirugía de bypass. Mientras tanto, se debe llevar un control adecuado de las funciones del corazón mediante el monitoreo de variables que componen el sistema cardiovascular.

Ahora bien, durante el tratamiento del paciente con una EIC se toman unas medidas o se recopilan ciertos datos relacionados con su enfermedad (sexo, edad, nivel de colesterol, ritmo cardiaco, presión arterial, etc.). Dichos datos son almacenados física o digitalmente para llevar históricos de los pacientes. Luego, dentro de esa gran cantidad de datos se pueden encontrar patrones que permitan a otras personas conocer si ciertos síntomas o factores de riesgo pueden desencadenar una EIC. Pero, tales patrones no son fácilmente encontrados (o no son lo suficientemente confiables y precisos) con el tratamiento de datos tradicional (creación del algoritmo, codificación, utilización del algoritmo planteado en código para obtener ciertos resultados). De esta manera, la herramienta Machine Learning juega un papel importante en la búsqueda de información en las bases de datos.

Machine learning (ML), es un método de análisis de datos que automatiza la construcción de modelos analíticos, el cual está basado en la idea de que los sistemas pueden aprender de datos, identificar patrones y tomar decisiones con mínima intervención humana [27]. Así, el problema desde ingeniería radica en crear el algoritmo que permita determinar la predicción del riesgo de una EIC. Además, la revisión de la literatura científica da cuenta de trabajos de optimización de algoritmos ([24], [25], [27], [31], [29], [30]), sin embargo, muy pocos tienen una orientación de servicio al paciente, lo cual se interpreta como una ausencia importante que requiere de la producción de alternativas para cuidado de los pacientes y la disminución de la mortalidad (a excepción de P.Amirtharaj y otros [30], quienes en su investigación, constantemente consultaban a un especialista sobre que atributos eran importantes o para comprobar la efectividad del método), y tampoco, se muestra algún tipo de producto, prototipo, o aplicación de sus proyectos (excepto Hassan Kahtan y otros [25], quienes presentan una aplicación en JAVA básica basada en lógica difusa (que no es Machine Learning) para entregar el nivel de riesgo).

Así, desde la ingeniería se pueden considerar los anteriores criterios para presentar un producto que tenga en cuenta las necesidades del usuario. Pero, para llegar a lo anterior, generalmente sucede que una vez se tenga el problema, rápidamente se quiere llegar a la solución, dejando a un lado aspectos importantes, e incluso lo que el usuario necesita. De esta manera, se usará la metodología TRIZ [1] (para romper la inercia mental de crear soluciones apresuradamente), la cual plantea principios universales de invención para productos, los cuales hacen que los procesos inventivos sean más predecibles, además de servir como base para obtener innovaciones a favor de la **tecnología** teniendo en cuenta aspectos de idealidad, contradicción y evolución de los sistemas técnicos. En definitiva, el diseño y la implementación del producto, de cierta manera, beneficia a un gran número de la población Colombiana que está afrontando alguna EIC, con el fin de minimizar sus efectos, en especial, la muerte. Además, este sistema podría ser utilizado por personas del común (como una continuación de este trabajo) que no tengan indicios de estas enfermedades, pero quieran llevar un control o registro de su actividad cardiaca, como un complemento de prevención primaria para este tipo de afecciones.

## 4.2. Formulación del problema

- Durante la prevención secundaria de las enfermedades isquémicas del corazón hace falta un producto de apoyo que de métricas precisas para conocer el nivel de riesgo de presentar nuevamente un episodio grave de la enfermedad.

## 4.3. Justificación

Las enfermedades isquémicas del corazón (EIC) han sido catalogadas en las principales enfermedades humanas más complejas y mortales del mundo. En estas afecciones, generalmente el corazón no puede empujar la cantidad de sangre requerida a otras partes del sistema cardiovascular para cumplir con las funciones normales del cuerpo, y debido a esto, en última instancia, ocurre la insuficiencia cardiaca [24]. El diagnóstico y el tratamiento de la enfermedad cardiaca es muy complejo, especialmente en los países en desarrollo, debido a la escasa disponibilidad de aparatos de diagnóstico y la escasez de especialistas y otros recursos que afectan la predicción y el tratamiento adecuado de los pacientes cardiacos. Además, según una encuesta realizada por la OMS, los profesionales médicos pueden predecir correctamente la enfermedad cardiaca con solo un 67 % de precisión [27]. Una buena solución a este problema es poder predecir cómo será el estado de salud de un paciente, para que los médicos puedan comenzar el tratamiento con anticipación y obtener mejores resultados. Esto ayudará a los pacientes a tomar medidas de precaución, como seguir una dieta equilibrada y medicamentos, lo que a su vez puede mejorar la calidad de vida del paciente y aumentar su esperanza de vida.

Con el objetivo de contrarrestar el avance de la enfermedad, de forma preventiva, las personas pueden tomar medidas como dejar de fumar, seguir una dieta saludable, hacer ejercicio, disminuir el estrés, entre otras. Tales medidas pueden ser tomadas con más aceptación si se da a conocer a los individuos que tienen riesgo de padecer una EIC. Sin embargo, la mayoría de las personas no acuden al médico si no hasta padecer un episodio de infarto o angina de pecho. Por esta razón, es necesario contar con un sistema que indique si se corre el riesgo de padecer una isquemia cardiaca, con el fin de tomar las medidas necesarias a tiempo. También, cuando se está en tratamiento, es importante la prevención secundaria, es decir, evitar que episodios graves de la enfermedad vuelvan a ocurrir.

Por otra parte, en un contexto más amplio, se tiene que los estados miembros de ONU para el 2015 adoptaron los Objetivos de Desarrollo Sostenible (17 ODS), como un llamado universal para poner fin a la pobreza, proteger el planeta y garantizar que todas las personas gocen de paz y prosperidad para 2030 [41]. Uno de esos objetivos es “Bienestar y salud”, que pretende reducir en un tercio la mortalidad prematura por enfermedades no transmisibles mediante la prevención y el tratamiento y promover la salud mental y el bienestar [42]. También está el objetivo “Industria, Innovación e Infraestructura”, el cual busca apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas [43]. De esta manera, el desarrollo del producto propuesto está alineado con los anteriores dos objetivos que plantea la ONU.

Por otro lado, el uso de la metodología TRIZ, en diferentes trabajos de grado como los de J.

Aguilar-Zambrano y otros, F. Torres, D. Lopez, N. Polo y D. Erira, y D. Bolaños y V. Mazuera, ha mostrado diversos productos, en su mayoría en el área médica, que son novedosos, y algunos están en proceso de patentes debido a que TRIZ permite crear nuevos productos a partir de la consideración de los existentes, sus limitantes y cuál es la idealidad del producto en el futuro. Para llegar a eso, el proceso de diseño seguido por los autores anteriores comprendió primero el análisis: que es la identificación de las necesidades de los usuarios; análisis sistémico del problema; y determinación de requerimiento del producto, proceso o sistema. Luego, se hizo la síntesis: proposición de alternativas de solución, modelamiento, cálculos, producción y operación. Por último, se realizó la evaluación: verificación de la satisfacción de necesidades e identificación de nuevos problemas. Así, a partir de lo anterior, se considera que TRIZ es la herramienta ideal para el desarrollo de este trabajo de grado si se le quiere dar el enfoque del diseño de un producto que tenga en cuenta la necesidad del usuario y sea novedoso.

En síntesis, es muy importante tener a disposición un sistema que apoye el proceso de diagnóstico de una EIC (ya que este es un problema grave en la actualidad), basándose en información personal (datos) y en el historial clínico de pacientes que ya padecen la enfermedad. Además, la predicción del riesgo de una EIC para tomar medidas preventivas es mucho mejor que actuar en el último minuto donde el paciente ya tiene la enfermedad bastante desarrollada y las recomendaciones clínicas ya nos son tan efectivas [25]. Por lo tanto, el producto a diseñar es pertinente y viable gracias al conocimiento adquirido en el transcurso de la carrera de ingeniería electrónica, los recursos y el apoyo de las ciencias de la salud.

#### 4.4. Sistema técnico

Para la utilización de TRIZ se debe ver la problemática como un sistema técnico, el cual se define como la representación de un producto (sistema) que toma la energía del exterior para realizar una función principal sobre un objeto. En este caso, donde se tiene un enfoque en diseñar un sistema para predecir el riesgo de padecer una isquemia cardiaca, el sistema técnico tiene como fuente de energía “la alimentación” y este se define como:

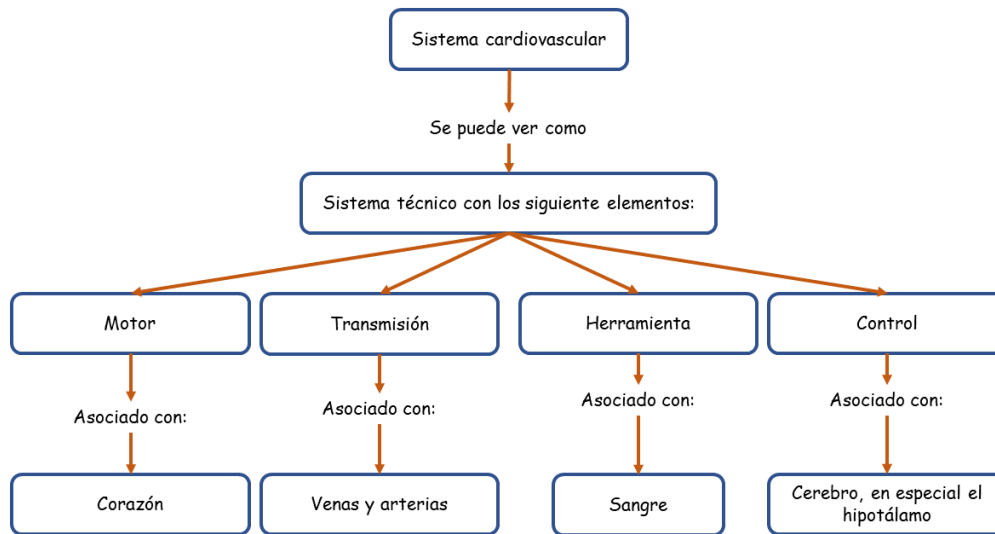


Figura 4.1: Modelo para el análisis del problema

Donde el sistema cardiovascular tiene como función principal **sumir las necesidades de los tejidos corporales, transportando hasta ellos los nutrientes como el oxígeno, eliminando los productos del metabolismo, llevando las hormonas de una parte del cuerpo a otra y, manteniendo los líquidos en el organismo para que las células funcionen de forma adecuada.** Además de esto, el sistema cardiovascular también cumple otra función importante, la cual es llevar la sangre baja en oxígeno hasta los pulmones para realizar el intercambio gaseoso y oxigenarse nuevamente [54]. De esta manera, el problema planteado en este trabajo de grado radica en que algunas de las partes del sistema cardiovascular como el corazón, venas y arterias sufren una afección, lo cual hace que la función principal NO se lleve a cabo correctamente. Por el motivo anterior, se pretende desarrollar una herramienta que permita informar al paciente o especialista de una anomalía. Lo anterior, junto con un grado de responsabilidad alto por parte de los interesados y un tratamiento pertinente, permite lograr un adecuado funcionamiento del sistema cardiovascular.

## 4.5. Análisis de las nueve ventanas

La técnica de las nueve ventanas permite analizar en forma sistémica un producto, desde una mirada funcional, de composición y de contexto en tres escenarios de tiempo, dos conocidos (el presente y el pasado), y uno de hipótesis (el futuro). Esta técnica permite dar cuenta de la evolución de los productos técnicos para poder proponer una solución de un objeto novedoso que realice la función principal por la que se ha concebido.

### 4.5.1. Ventana 1 Sistema (Presente)

- **Estudios recientes para la predicción del riesgo de una EIC:**

Hasan Kahtan y otros [25] presentan *Heart Disease Diagnosis System Using Fuzzy Logic*. Ellos dan a conocer cuatro etapas y siete pasos que están involucrados en la implementación del

enfoque de lógica difusa para el diagnóstico de una EIC. Dichas etapas son: fuzzificación, base de reglas difusas, motor de inferencia difusa y defuzzificación. Este estudio presenta su predicción con cuatro atributos (edad, presión arterial, colesterol, porcentaje de azúcar en la sangre). Además, el sistema propuesto, mencionan los autores, tiene un 98 % de precisión cuando se evalúa utilizando *UCI Machine Learning Repository: Heart Disease Data Set* [26]. No obstante, no se muestra la forma en que se calcula ese valor. Además, ellos presentan los resultados con 3 porcentajes para saludable, punto medio y enfermo. Finalmente, proponen un trabajo futuro, donde se puede implementar una aplicación móvil con reconocimiento de patrones y herramientas que puedan generar datos en tiempo real utilizando las huellas digitales de los usuarios para permitirles obtener fácilmente datos para las variables de entrada. También, cabe decir que Hasan Kahtan [25] y otros ofrecen un tipo de aplicación en Java en donde se introducen los 4 atributos (edad, presión arterial, colesterol, porcentaje de azúcar en la sangre) para el diagnóstico de una EIC utilizando la lógica difusa. Sin embargo, el uso de tal cantidad de atributos puede dar cabida a errores de precisión, como lo presentan otros investigadores [29].

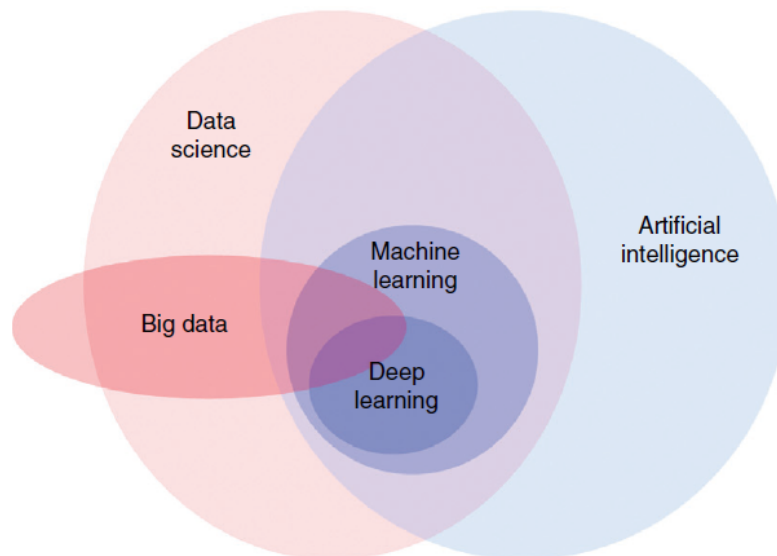


Figura 4.2: Diagrama de Venn de los términos más utilizados en las disciplinas de la ciencia de datos [28].

También, actualmente existe la inteligencia artificial (IA), la cual es clasificada por P. Ignacio Dorado-Díaz y otros [28] como aparece en la figura 4.2. El término en sí, a menudo se aplica al campo de la informática que intenta imitar los procesos cognitivos humanos, la capacidad de aprendizaje y el almacenamiento de conocimiento. Ellos mencionan que la inteligencia artificial no es una idea futurística, ya que esta tecnología se está implementando en diversas áreas como la cardiología. Su trabajo consiste en presentar las diversas alternativas que presenta la ciencia de datos para el análisis de Big Data con el objetivo de proporcionar excelentes oportunidades para la automatización de tareas, la aplicación de medicina de precisión o el progreso de la investigación a través de la detección de patrones complejos en bases de datos médicas. También, mencionan limitaciones, aspectos éticos y legales que puede acarrear la implementación de la inteligencia artificial. Al final, ellos concluyen que la aplicación de la IA en el área de medicina

es el presente y futuro de la cardiología.

Amanda H. Gonsalves y otros [27], presentan en *Prediction of Coronary Heart Disease using Machine Learning: An Experimental Analysis* una propuesta de diferentes métodos ML que se han venido utilizando para la predicción de EIC. Los autores mencionan que las redes neuronales artificiales (ANN), los árboles de decisión (DT) y el algoritmo de Naïve Bayes (NB) son los más usados en el campo para este tipo de propósito. Sin embargo, en su estudio utilizan NB, DT y Support Vector Machine (SVM), ya que la razón para elegir estas técnicas es su aplicabilidad en diferentes dominios, así como las diferentes metodologías de aprendizaje que adoptan. Ahora bien, el objetivo de su estudio fue encontrar el modelo de ML más efectivo para predecir la presencia de una EIC utilizando la base de datos de enfermedades cardiacas de Sudáfrica [36]. Al final, los autores concluyen que el algoritmo Naïve Bayes resultó ser el mejor clasificador para el conjunto de datos considerado, ya que su tasa de predicción es mejor que la de los algoritmos DT y SVM, al menos en el conjunto de datos considerado.

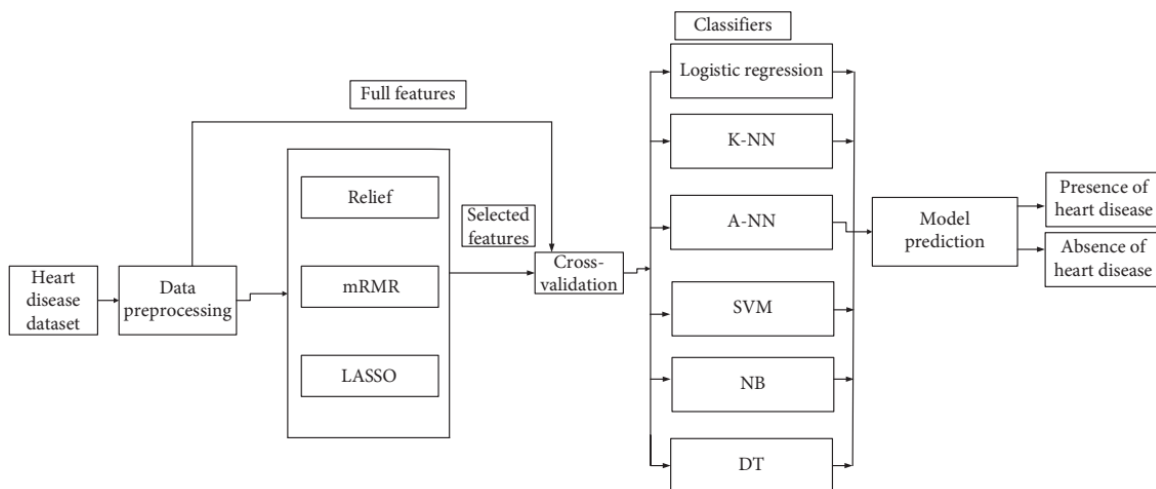


Figura 4.3: Diagrama de un sistema inteligente que predice EIC. [24]

Amin Ul Haq y otros [24] en su estudio *A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms* realizan varias pruebas con distintos algoritmos de selección y clasificadores [4.3], con el fin de conocer cuál es el mejor método que se debe utilizar en cuanto a la predicción del riesgo de EIC. Para lograr lo anterior, toman referencia la base de datos (dataset) del hospital de Cleveland la cual es utilizada por varios investigadores, y se puede acceder a ella desde el repositorio de minería de datos en línea de la Universidad de California, USA. Luego, utilizan 3 algoritmos que les permitan obtener de la base de datos los atributos más importantes, ya que el dataset que utilizan tiene 76 atributos las cuales, entre ellos pueden existir algunos que sean redundantes o de poca información para predecir una EIC (la descripción de estos atributos se encuentra en el **Anexo C** [9.3]). Después, utilizan el método *k-fold* para validar si se hizo una correcta elección de atributos.

Luego, el paso a seguir es la implementación de los 6 métodos ML (los cuales aparecen en la figura [4.3] junto con los demás pasos utilizados por los autores) para clasificar el riesgo de padecer una EIC. Después, se hace una validación de los resultados de tales clasificadores. Para

lograr eso, se utiliza una matriz de confusión para evaluar el desempeño de los clasificadores. Luego, con el cálculo de los valores de dicha matriz, utilizan cuantificadores de rendición como la precisión, la tasa de error, la sensibilidad y la especificidad. Con las anteriores medidas, Amin Ul Haq y otros [24] identifican que **las redes neuronales artificiales junto con el algoritmo de selección Relief, clasifica correctamente la enfermedad cardiaca en las personas**. Sin embargo, ellos mencionan que este no es el mejor método, ya que la regresión logística tiene mejor precisión cuando se utiliza con Relief. En resumen, los autores presentan que el uso de redes neuronales es muy viable para la predicción de EIC.

Similarmente, Alexandre Vallée y otros [29] realizan en su trabajo de investigación *Coronary heart disease diagnosis by artificial neural networks including aortic pulse wave velocity index and clinical parameters* la implementación de una red neuronal artificial para el diagnóstico de una enfermedad coronaria. Para ello, toman en cuenta el índice de velocidad de onda de pulso aórtico (PWV) y un conjunto de datos clínicos. En el caso de PWV, este se define como la velocidad del pulso arterial que se mueve a lo largo de la pared arterial; y, debido a numerosos estudios, mencionan los autores, PWV resulta ser un *predictor* de los eventos cardiovasculares, en especial una enfermedad coronaria. Por otra parte, el conjunto de datos clínicos fue tomado de 530 pacientes en el *Paris Hôtel-Dieu University Hospital*, donde un total de 93 pacientes libres de hipertensión, diabetes y EIC fueron usados para el cálculo del PWV teórico. Después, la cuestión a resolver es clasificar con ‘1’ a individuos con una enfermedad coronaria o con ‘0’ a sujetos con ausencia de esta. Además, la red neuronal opera en 2 modos: entrenamiento y predicción utilizando dos conjuntos de datos. Luego, cuando se implementa dicha red neuronal con un perceptrón multicapa, los autores utilizan 10 modelos que utilizan un número diferente de características (visibles en la figura 4.4) para validar cada modelo.

Model	Factors	Accuracy	Se (%)	Sp (%)
1	Age, diabetes, hypertension, obesity, smoking, family anamnesis of CHD	0.63	53	79
2	Age, sex, diabetes, hypertension, obesity, smoking, family anamnesis of CHD	0.76	56	81
3	Age, sex, diabetes, hypertension, obesity, smoking, family anamnesis of CHD, glycaemia, cholesterol total	0.77	56	83
4	Age, sex, diabetes, hypertension, obesity, smoking, family anamnesis of CHD, TG, cholesterol total, HDL, LDL, glycaemia	0.81	69	79
5	Age, sex, diabetes, hypertension, obesity, smoking, family anamnesis of CHD, carotid plaque	0.83	64	85
6	Diabetes, hypertension, obesity, smoking, family anamnesis of CHD, PWV index	0.87	70	83
7	Diabetes, hypertension, obesity, smoking, family anamnesis of CHD, carotid plaque, PWV index	0.91	80	91
8	Diabetes, hypertension, obesity, smoking, family anamnesis of CHD, TG, cholesterol total, HDL, LDL, glycaemia, carotid plaque, PWV index	0.93	80	92
9	Age, sex, diabetes, hypertension, obesity, smoking, family anamnesis of CHD, glycaemia, cholesterol total, cGFR	0.77	53	87
10	Age, sex, diabetes, hypertension, obesity, smoking, family anamnesis of CHD, glycaemia, cholesterol total, left ventricular hypertrophy	0.77	60	80

All cases, the multilayer perceptron neuronal network topology with three hidden neuron layers was used. cGFR, calculated glomerular filtration rate; CHD, coronary heart disease; PWV, pulse wave velocity; Se: sensitivity; Sp: specificity; TG, triglycerides.

Figura 4.4: Precisión, sensibilidad y especificidad de la red neuronal artificial [29]

En la figura 4.4 se aprecia que el modelo 8 es el que tiene mejor precisión (93 %). Este modelo solo utiliza 12 atributos en los que se incluye el PWV. Después, los autores hacen varias pruebas con diferentes tipos de red neuronal donde se utilizan más capas. Al final, los distintos tipos de pruebas sugieren que **el tipo de red neuronal, el número y la selección específica de los atributos de los pacientes son muy determinantes en la precisión del diagnósti-**

**co de una enfermedad coronaria.** Es así como Alexandre Vallée y otros concluyen que la implementación de una red neuronal puede llevar a una alta precisión para la predicción de una enfermedad coronaria. Además, mencionan que esta precisión puede ser mejorada no solo incrementando el número de atributos, sino también una correcta selección de estos. Sin embargo, ellos presentan que el uso de todos los atributos complica el modelo, el cual, también está limitado por los métodos instrumentales (hacer la medición de los atributos que necesita el modelo).

Igualmente, P.Amirtharaj y otros [30] plantean en su trabajo *A novel risk level classification of ischemic heart disease using artificial neural network technique* el diseño de una red neuronal artificial de propagación hacia atrás para clasificar el nivel de riesgo de una isquemia miocárdica. Ellos utilizan un conjunto de datos de 236 personas de la India. Para su estudio, utilizan 17 atributos de estos pacientes para hacer la predicción del riesgo. Su idea es desarrollar un sistema inteligente, que apoye a una persona, ya sea enfermera o voluntaria, en un área rural o urbana para medir el riesgo de EIC. Esto reducirá el tiempo que un médico tiene que utilizar para analizar el riesgo de una persona. Al final, **los autores presentan que su sistema tiene un 90% de precisión cuando se hace las pruebas correspondientes.** También, mencionan que para un trabajo futuro se pretende reducir el número de atributos usando técnicas de optimización como los algoritmos genéticos.

Por otra parte, Aman Mahajani y otros [32] en su trabajo *Transformation in Healthcare by-Wearable Devices for Diagnostics and Guidance of Treatment* presentan que la monitorización portátil permite medir la condición clínica de los sujetos mientras se encuentran en sus entornos ambulatorios regulares, incluidos los entornos residenciales y laborales, en comparación con el entorno hospitalario. Esto tiene una ventaja adicional de reducir los costos de atención médica y mejorar la conveniencia del paciente al promover la atención médica remota o “telemédica” y reducir la necesidad de que el paciente sea trasladado a los hospitales. En general, su trabajo está destinado a proporcionar orientación a los investigadores en *computación para la atención médica* que deseen adoptar la misión convincente del desarrollo de dispositivos portátiles para el diagnóstico y la orientación para el cuidado de la salud. Dicha orientación se presenta en forma de una guía, en la cual se presentan una serie de pasos importantes que se deben tomar en cuenta a la hora de realizar los diseños.

La guía anterior se aplica en un ejemplo de investigación para desarrollo de un dispositivo portátil para el diagnóstico de enfermedades coronarias. En ese ejemplo, se muestran los beneficios, métodos de diseño, implementación y evaluación del sistema que propone la investigación. Sin embargo, aunque existen algunas limitaciones para los dispositivos portátiles, las ventajas son mayores que las desventajas cuando se tienen en cuenta en el diseño general del sistema. Además, los autores dan un respaldo al uso de dispositivos portátiles para el diagnóstico de enfermedades como las EIC, ya que ellos concluyen que la tecnología portátil tiene una promesa increíble de poder doblar la curva de costos en atención médica. También, la investigación en dispositivos portátiles puede liderar la transformación clínica en la forma en que se brinda la atención médica en el futuro.

En síntesis, los trabajos consultados presentan propuestas interesantes para contribuir al cuidado y mejoramiento de la salud de las personas utilizando métodos de ML. En el caso de Amanda H. Gonsalves [27], Amin Ul Haq [24], P.Amirtharaj [30] y Alexandre Vallée [29], presentan métodos ML interesantes para el diagnóstico o predicción de una enfermedad del corazón. Los dos últimos autores utilizan redes neuronales para el desarrollo de sus sistemas, mientras que Amanda H. Gonsalves y Amin Ul Haq presentan diferentes métodos de ML para realizar una comparación de estos y conocer cuál es el que mejor en términos de precisión, sensibilidad y especificidad.

Igualmente, es importante resaltar el trabajo de P. Ignacio Dorado-Díaz y otros [28], quienes presentan las grandes oportunidades de la inteligencia artificial en la cardiología. Ellos muestran el desarrollo de distintos términos de la ciencia de datos, y como la integración y aplicación de estos en los últimos años han sido de gran ayuda para el cuidado de la salud. Estos últimos investigadores presentan una lista de otras investigaciones en el área de cardiología para soportar la relevancia de su trabajo, pero, P. Ignacio Dorado-Díaz y otros no desarrollan un método para predecir el riesgo de una EIC. Para finalizar, Aman Mahajan y otros [32] se encargan de presentar una guía de pasos para el uso y la implementación de sistemas que utilicen dispositivos portátiles para el cuidado de la salud. Sin embargo, ellos no hacen un desarrollo de un sistema para el diagnóstico de una EIC, más bien, citan a otros autores con el fin de mostrar cómo se implementa su guía en dichas investigaciones.

- **Aplicaciones digitales para la salud del corazón:** Actualmente existen diversas aplicaciones digitales que acompañan a las personas diariamente para el cuidado de su salud. Muchas de ellas varían entre temas como actividad física, alimentación saludable, rápido contacto con un médico, registro de variables importantes para la salud, entre otros. Una de esas aplicaciones es *Instant Heart Rate*, la cual está disponible en dispositivos como Android y Apple. Esta aplicación, según sus creadores “es la aplicación monitor de pulso más precisa para cualquier teléfono inteligente y no necesita ningún hardware externo” [45]. De igual manera, para dispositivos Apple está *HeartWatch*. Dicha aplicación es una herramienta muy útil para recibir alertas sobre cualquier posible problema de salud en un formato conciso que se pueda mostrarle al médico. También, *HeartWatch* registra las pulsaciones cuando el usuario camina, descansa, realiza ejercicio físico o duerme. Además de medir y mostrar la frecuencia cardiaca durante el día, también emite notificaciones cuando esta excede o cae por debajo del valor adecuado para el usuario de lpm (latidos por minuto). Otra ventaja de esta aplicación es que permite tomar notas sobre el día y hacer un seguimiento del sueño.

HeartWatch ofrece una visión exhaustiva de la información que la aplicación recoge gracias a la información del Apple Watch. La aplicación se divide en cuatro secciones: Constantes, Dashboard, Actividad y Más. Por el lado de Constantes, se tiene toda la información de las diferentes lecturas representadas con un sistema de anillos de colores. Con un toque en cualquiera de los diferentes gráficos se accede a la gráfica detallada. En la sección Dashboard se puede consultar varios datos. Todos ellos se muestran más y más detallados a medida que se pulsa en las diferentes métricas y se accede a gráficos e históricos. Los datos de esta sección son:

- Actividad física: Movimiento, Ejercicio y De pie.



Figura 4.5: HeartWatch

- Lecturas de ritmo cardiaco: divididos en ritmo durante el sueño, al despertar y durante los entrenos.
- Pasos y movimiento: donde se ven los pasos del día, las vueltas y las calorías durante el movimiento.
- Los de Timogochi, o Tim para abreviar, que muestra una representación visual del día en forma de un pequeño muñequito más o menos en forma.

Por último, la sección Actividad da acceso a lo que la aplicación considera un sustituto de los anillos de actividad del Apple Watch. En la introducción, la misma aplicación explica que, a veces, dado que cada día es distinto, los anillos de actividad del Apple Watch pueden ser complejos de cerrar. HeartWatch pasa a un sistema dinámico que varía los objetivos del día en función de los datos históricos. La idea es acompañar al usuario a estar activos sin que haya demasiada presión [46]. En resumen, HeartWatch no solo se encarga del registro de la frecuencia cardiaca, sino que se ocupa de otros aspectos fundamentales en el cuidado de la salud, además de, contar con una interfaz agradable e intuitiva para el usuario.



Figura 4.6: Otras características de HeartWatch

Por otra parte, están aplicaciones similares a *Adidas Runtastic Balance*, la cual sirve para el registro de comidas, centrándose en la salud y el bienestar, poniendo el acento en la nutrición. Con una gran base de datos de comidas, información nutricional detallada y planes de nutrición optimizados para cada objetivo de peso, Balance pretende hacer que sus usuarios tomen conciencia sobre sus hábitos alimenticios y, a cambio, animarlos a inclinarse por opciones más saludables. Ahora bien, en un caso más drástico está *Ariadna*, la cual se basa en un mapa colaborativo de desfibriladores. Sus creadores mencionan que la muerte súbita cardíaca es una de las causas más frecuentes de muerte en los países desarrollados y se estima que las posibilidades de sobrevivir sin secuelas a un paro cardíaco extrahospitalaria oscilan del 5 al 11 % [47]. El usuario de *Ariadna* puede registrarse como “rastreador”, o como “colaborador”. Los “rastreadores” pueden indicar la ubicación de desfibriladores, tanto en espacios públicos como privados. Y pueden, incluso, completar la geolocalización con fotografías del lugar para que sea más fácil ubicar el dispositivo. Del mismo modo, también podrán validar los DEA (Desfibrilador Externo Automático) que hayan sido dados de alta previamente en la aplicación. *Ariadna* está “gamificada”, por lo que los usuarios que más contribuyan al desarrollo del mapa de desfibriladores recibirán medallas simbólicas como premio.

#### 4.5.2. Ventana 2 Supersistema (Presente)

En esta parte se muestra información del contexto en el presente para una visión amplia del problema en el análisis.

- Especialistas médicos en cardiología:** El observatorio laboral para la educación (OLE) presenta la vinculación laboral de los recién graduados en el país. De esta manera, OLE tiene datos de los profesionales relacionados con la salud cardiovascular desde 2001 hasta el 2018. Lo anterior se resume en la tabla de la figura (4.7) donde en la segunda columna se recopila el total de graduados desde 2001 hasta 2018; y en la columna 2 y 3 se resume el IBC (Ingreso Base de Cotización) y la tasa de cotizantes en el 2015 de los recién graduados en las profesiones listadas. En la 4.7 se puede evidenciar que la cantidad de especialistas es reducida (574 según la tabla) en un país con una población cercana a los 50 millones de personas que mueren principalmente por enfermedades del corazón. Además, dichos especialistas se concentran en las grandes urbes, haciendo que en zonas rurales sea escasa la atención respecto a enfermedades del corazón.

Programa	Graduados	IBC COP	Tasas cotizantes
Especialización en enfermería cardiovascular	80	\$ 3.788.615	100,0 %
Especialización en ecocardiografía	83	\$ 2.884.881	100,0 %
<b>especialización en cardiología</b>	<b>172</b>	<b>\$ 7.785.143</b>	<b>90,0 %</b>
especialización en cardiología de adultos	125	\$ 4.200.699	100,0 %
especialización en enfermería cardiorrespiratoria	64	\$ 3.783.571	88,9 %
especialización en cardiología pediátrica	50	\$ 1.262.000	100,0 %

Figura 4.7: Información de la cantidad de especialistas relacionados con la salud cardiovascular y su vinculación laboral [53]

- Dieta poco saludable en los jóvenes:** La Encuesta Nacional de Salud Escolar elaborada por la Universidad del Valle para el 2018 consultó a 79.640 estudiantes de 13 a 17 años de todo el país sobre la prevalencia de factores relacionados con la salud [38]. Dicha encuesta encontró que nueve de cada diez (86 %) estudiantes escolares en Colombia no consumen frutas ni verduras, según las recomendaciones de la OMS (al menos 400 gramos al día). Además, tres de cada cuatro escolares (74 %) consumen una o más bebidas azucaradas al día. También, el 47,3 % consume fritos una o más veces al día; 14,8 % consume comidas rápidas tres días a la semana y el 83 % come paquetes de alimentos ultraprocesados una o más veces al día. En definitiva, los datos anteriores muestran que no hay un interés elevado en la juventud Colombiana por cuidar de su salud, haciendo que se eleve el número de personas con una enfermedad de corazón en el país y en el mundo.
- Absentismo laboral por enfermedades relacionadas con el sistema cardiovascular:** Desde al año 2012, el Ministerio de Salud y Protección Social lleva registro de las incapacidades por enfermedades generales que afectan, particularmente, a quienes cotizan al régimen contributivo de seguridad social en salud. De esta manera, las enfermedades del sistema circulatorio de clasificación I por el CIE (Clasificación Internacional de Enfermedades) para el 2017 hicieron que 91.482 personas (de 3'755.945 de casos) faltaran al trabajo, sumando un total de 1'174.158 días de incapacidad, donde cada persona presentaba 12,8 días en promedio de incapacidad. Sin

embargo, cabe decir que las enfermedades del sistema osteomuscular y del tejido conectivo son las que presentan más casos de absentismo laboral con un total de 733.430 [39].

- **No consulta al médico :** En un artículo del eltiempo.com [40], la experta bioética Lydia Feito, habla del miedo al diagnóstico o al tratamiento. Ella menciona que las personas tienen miedo a ser examinado físicamente; al centro hospitalario; a las investigaciones y pruebas médicas; al tratamiento; al estigma o discriminación que pueda suscitar la enfermedad; a ser presionado; a tener que cambiar el estilo de vida; a parecer débil o a no tener el control de la vida; a las consecuencias que pueda tener en la pareja o la vida laboral. Lo que dicen los estudios, según esta docente, es que el temor se da con mayor frecuencia en los varones de mediana edad. “Curiosamente, es una etapa donde aumenta el riesgo de que aparezcan enfermedades como la diabetes o los infartos”, precisa ella.
- **Objetivos de desarrollo sostenible:** También conocidos como Objetivos Mundiales, se adoptaron por todos los Estados Miembros en 2015 como un llamado universal para poner fin a la pobreza, proteger el planeta y garantizar que todas las personas gocen de paz y prosperidad para 2030 [41]. Dentro de esa lista de objetivos se encuentran Salud y Bienestar (Objetivo 3), e Industria, Innovación e Infraestructura (Objetivo 9). El primero, pretende reducir en un tercio la mortalidad prematura por enfermedades no transmisibles mediante la prevención y el tratamiento y promover la salud mental y el bienestar [42]. Mientras que el segundo, busca apoyar el desarrollo de tecnologías, la investigación y la innovación nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas [43]. Todo lo anterior tiene como fecha objetivo el año 2030.
- **Crecimiento del mercado de celulares:** Según la firma de estudios de Growth from Knowledge (GFK), el mercado de telecomunicaciones representa un incremento de más de COP 809.000 millones para el 2019, dicho incremento es debido al fortalecimiento de SIM free (la red celular del teléfono puede funcionar con todos los operadores disponibles), de las estrategias de promoción y publicidad de los operadores, así como la decisión de los fabricantes de ofrecer teléfonos con características más avanzadas a mejores precios. Igualmente, GFK ha encontrado que cerca del 31 % de la población investigada para su estudio declara la intención de invertir menos de COP 750.000 en su próximo equipo, un 25 % entre COP 750.000 y COP 999.000, y un 44 % más de COP 1'000.000. De igual forma, el estudio anterior muestra que el comprador Colombiano en el 2019 valoró de manera más importante el rendimiento del equipo, seguido de la capacidad de almacenamiento y la batería (tamaño y duración) [44]. Así, se logra identificar que la mayoría de Colombianos cada vez más se interesan por conseguir un celular con mayores prestaciones (“alta gama”) con un precio alrededor de COP 1'000.000.
- **Colesterol, grasas, sal y conservantes:** Las grasas de la dieta influyen sobremanera en el riesgo de EIC como la cardiopatía coronaria y el accidente cerebrovascular, debido a sus efectos en los lípidos sanguíneos, la formación de trombos, la tensión arterial, la función arterial y la inflamación [50]. Además, la ingesta alimentaria de sodio de todas las fuentes influye en los

niveles de tensión arterial de las poblaciones y debe limitarse para reducir el riesgo de cardiopatía coronaria y de los dos tipos de accidente cerebrovascular. Por otro lado, los efectos de los conservantes a largo plazo pueden ser perjudiciales para la salud. Algunos conservantes pueden alterar la composición y el funcionamiento intestinal que luego se traduce en la inflamación intestinal que promueve el desarrollo de la enfermedad inflamatoria intestinal y el síndrome metabólico [51].

- **Telemedicina:** Este concepto aparece como una forma de luchar contra las barreras geográficas aumentando la accesibilidad a los cuidados de salud, especialmente en zonas rurales y países en desarrollo. De igual manera, telemedicina significa “medicina a distancia” (diagnóstico, tratamiento, etc.), mediante recursos tecnológicos que optimizan la atención, ahorrando tiempo y dinero, además de aumentar la accesibilidad. Por otra parte, para una implementación de telemedicina se necesitan considerar factores como: conocimiento tecnológico por parte de los pacientes y médicos, existencia de protocolos que se adapten al contexto, visión de futuro e innovadora, compromiso de las personas involucradas, desarrollo tecnológico, apoyo económico e institucional y la aceptación por parte de clínicos y pacientes [49].

#### 4.5.3. Ventana 3 Subsistema (Presente)

Aquí, se hace un análisis del estado actual de la tecnología para soportar los productos de apoyo al cuidado de la salud cardiovascular. De esa manera, en esta ventana se muestran los diferentes elementos que componen a los métodos, algoritmos o sistemas para la predicción del riesgo de una EIC. Esto se resume en la tabla de la figura (4.8), donde en la segunda columna se hace el análisis de los estudios de Alexandre Vallée [29] y P.Amirtharaj [30]. De igual forma, en la tercera columna se analizan los trabajos de Amin Ul Haq [24] y Amanda H. Gonsalves [27]. También, en la cuarta columna solo se hace referencia al trabajo de Hasan Kahtan [25]. Por último, en la quinta columna se hace el análisis de la aplicación Heart Rate Monitor [45].

Elementos	Diagnóstico de una EIC utilizando una ANN	Predicción de una enfermedad coronaria utilizando métodos ML	Diagnóstico de una enfermedad del corazón con Lógica Difusa	Heart Rate Monitor
¿Cómo se utiliza?	Los datos médicos del paciente se introducen en el método para que luego se le determine si tiene el riesgo o no de padecer una EIC.	Se toman las características más relevantes del conjunto de datos. Después, dichas características sirven de entrada para los diferentes métodos ML que al final de los procesos, entregan la predicción.	El usuario dispone de una aplicación en JAVA en la cual se introducen la edad, la presión, el nivel de azúcar y el colesterol. Después, la salida es la presencia o ausencia de una EIC.	El usuario coloca su dedo en la cámara del celular. Después de un tiempo, se le informa que frecuencia cardiaca (FC) tiene.
¿Qué base de datos utiliza?	Datos de 530 residentes del Paris Hôtel-Dieu University Hospital y datos de 125 personas que habitan en la India.	La base de datos de Cleveland y la base de datos de Sudáfrica.	UCI Machine Learning Repository: Heart Disease Dataset	No se utiliza ninguna base de datos. El usuario puede crear un registro de la FC que puede compartirse con un profesional médico.
¿Cómo funciona?	Una ANN cuenta con 3 capas (input, hidden, output). Esta consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.	Se hace un preprocesamiento de la base de datos. Luego, se eligen las características más importantes con diversos "seleccionadores". Después, con dichas características se prueban los distintos métodos ML. Al final, se crea un modelo que entrega la predicción, ya sea presencia o ausencia de una EIC.	La lógica difusa cuenta con 4 etapas en donde los datos de entrada se les asigna un rango máximo y mínimo numérico (las variables se les asignan conjuntos difusos a través de funciones de membresía). Después, se crean las reglas de tipo if-then. Finalmente, en la última etapa se le asigna un valor numérico (escala) a la salida difusa.	La fuerte luz del flash LED es capaz de que la cámara de fotos pueda detectar los movimientos de las venas, y así calcular cuál el ritmo cardiaco
¿Qué método utiliza?	Utiliza una red Neuronal de perceptrón multicapa (MLP) y una de propagación hacia atrás.	En los estudios se utilizan diversos métodos con DT, NBA, SVM, Regresión logística y K-NN para conocer cuál es el más adecuado para la predicción.	Utiliza la lógica difusa (fuzzification, fuzzy rule base, fuzzy inference engine and defuzzification) para todo el proceso.	No se menciona que método utiliza
¿Qué hardware y software utiliza?	SAS: Analítica, Inteligencia Artificial y Gestión de Datos. No se menciona el hardware utilizado	WEKA, open source data analysis software version 3.8.3, y Python en un Intel(R) Core™ i5 -2400CPU @3.10GHz PC.	Se utiliza NetBeans IDE 8.2 en JAVA para la construcción y el desarrollo de la aplicación. No se menciona que hardware se utiliza.	Se necesita de un celular con cámara y flash, además de una plataforma Android o IOS
¿Cómo da la predicción?	1 para la presencia de una EIC o 0 para la ausencia de esta. También, en el otro estudio se da la predicción en escala de No riesgo, Bajo, Medio o Alto.	Presencia o ausencia de una EIC.	La aplicación entrega la predicción del riesgo de una EIC en 3 escalas, Alto, Medio o Bajo.	Con los diferentes registros de la FC, la aplicación da consejos para mejorar la FC.
¿Qué ventajas tiene?	Se presenta un 90% y 93% de precisión en los estudios. En uno de ellos se crean diversos modelos para saber cual es más adecuado para predecir una EIC.	Presentan diferentes métodos, sus resultados y la conclusión acerca de cual de ellos es el que mejor predice una EIC a partir de métricas como la precisión, sensibilidad y especificidad.	Dispone de una interfaz sencilla y entendible. Además, el usuario necesita solo 4 características para conocer el riesgo. Tiene un 98% de precisión.	No necesita ninguna base de datos, es sencillo de utilizar, es fácil de acceder y es gratis.

Figura 4.8: Elementos del subsistema presente.

#### 4.5.4. Ventana 4 Sistema (Pasado)

En esta ventana se realiza el análisis de los sistemas pasados asociados con el problema para evidenciar tendencias de evolución con los actuales y futuros. También para evitar errores en el sistema futuro. Ahora bien, desde 1948, el *Framingham Heart Study*, bajo la dirección del *National Heart, Lung, and Blood Institute* (NHLBI), se ha comprometido a identificar los factores o características comunes que contribuyen al desarrollo de las EIC, y se ha mantenido hasta el presente con tres generaciones de participantes [35]. El estudio empezó con el reclutamiento de una cohorte original

de 5,209 hombres y mujeres de entre 30 y 62 años de la ciudad de Framingham, Massachusetts, que aún no habían desarrollado síntomas evidentes de una enfermedad del corazón o habían sufrido un ataque cardíaco o un derrame cerebral. Desde entonces, el estudio ha agregado una Cohorte de Hijos en 1971, la Cohorte Omni en 1994, una Cohorte de Tercera Generación en 2002, una Cohorte de Cónyuges de Nuevos Hijos en 2003 y una Cohorte de Omni de Segunda Generación en 2003. A lo largo de los años, el seguimiento cuidadoso de la población del estudio Framingham ha permitido identificar los principales factores de riesgo de las EIC, así como información valiosa sobre los efectos de estos factores, como **la presión arterial, los niveles de triglicéridos y colesterol en sangre, la edad, el sexo y problemas psicosociales** que acarrearán las EIC. Se han investigado y se siguen investigando los factores de riesgo de otras afecciones fisiológicas como la demencia. Además, se están estudiando las relaciones entre los rasgos físicos y los patrones genéticos.

Ahora bien, para el año de 1998 Peter W.F. Wilson y otros [34] proponen en su artículo denominado *Prediction of Coronary Heart Disease Using Risk Factor Categories* un modelo simplificado de predicción de enfermedad coronaria, basándose en las categorías de presión arterial, colesterol y LDL-C (colesterol malo), las cuales fueron propuestas por diversos paneles de expertos en el tema de las EIC para la época. La muestra poblacional utilizada para este artículo incluyó 2489 hombres y 2856 mujeres de 30 a 74 años en el momento de su examen del Framingham Heart Study en 1971 a 1974. El resultado de su proyecto consiste en que una persona común y corriente se le hace una consulta de factores como la edad, presión arterial, si es diabética, si fuma y el nivel de colesterol en el organismo, para luego, darle una puntuación a cada factor, donde la suma de cada puntuación, a través de unas tablas que ellos desarrollan, indican el nivel de riesgo de una EIC en 10 años. En síntesis, los autores mencionan que durante las décadas de 1980 y 1990 ha sido posible **estimar el riesgo de cardiopatía coronaria mediante el uso de ecuaciones de regresión derivadas de estudios observacionales** donde, precisamente su estudio demuestra resultados similares, prediciendo una cardiopatía coronaria futura en una muestra de población blanca de mediana edad. Los modelos de predicción se han basado típicamente en la función logística (es una función matemática que aparece en diversos modelos de crecimiento de poblaciones, propagación de enfermedades epidémicas y difusión en redes sociales, la cual constituye un refinamiento del modelo exponencial para el crecimiento de una magnitud), aunque también se ha utilizado la distribución de Weibull. En síntesis, el estudio de Peter W.F. Wilson y otros presenta un modelo de hace 20 años para predecir una EIC basado en la información del estudio Framingham. Cabe destacar que para el año de 1998 no había el auge de los dispositivos móviles como hoy en día (apenas estaban en su desarrollo) por lo que el modelo, o fue probado en las computadoras de ese entonces o se hicieron los cálculos de manera manual.

#### 4.5.5. Ventana 5 Supersistema (Pasado)

Las enfermedades relacionadas con el corazón siempre han afectado a la raza humana y no es solamente nuestro estilo de vida moderno el culpable de su aparición y auge. Por ejemplo, en la reunión de la American Heart Association de 2009 en Florida, investigadores presentaron los resultados de un estudio que mostraban que las momias egipcias, de unos 3500 años de antigüedad, tenían evidencia

de enfermedad cardiovascular, específicamente aterosclerosis en diferentes arterias del cuerpo [52]. Lo anterior, según dichos investigadores, pudo ser posible debido a que la clase élite de Egipto, para ese entonces, consumía muchas carnes grasas de ganado, patos y gansos. Sin embargo, es difícil decir exactamente cuándo la civilización se dio cuenta por primera vez de la enfermedad relacionada con el corazón. Pero, se sabe que Leonardo da Vinci, alrededor de sus 50 años, investigó las arterias coronarias [52]. Luego, 500 años después, la enfermedad de las arterias coronarias es una de las causas más comunes de muerte en el mundo occidental.

La década de 1900 marca un período de mayor interés, estudio y comprensión de las enfermedades cardíacas. En 1915, un grupo de médicos y trabajadores sociales formó una organización llamada *Asociación para la Prevención y el Alivio de las Enfermedades Cardíacas* en la ciudad de Nueva York. Luego, en 1924, varios grupos de asociaciones del corazón se convirtieron en la *Asociación Estadounidense del Corazón*. Estos médicos estaban preocupados por la enfermedad porque sabían poco sobre ella. Los pacientes que normalmente asistían a citas médicas tenían pocas esperanzas de recibir tratamiento o una vida plena. Después, en 1948, investigadores bajo la dirección del Instituto Nacional del Corazón (ahora llamado *National Heart, Lung, and Blood Institute*) iniciaron el Framingham Heart Study, el primer estudio importante que ayuda a comprender las enfermedades cardíacas, según un artículo de la revista *The Framingham Heart Study*. También, en la década de 1950, el científico estadounidense Ancel Keys descubrió en sus viajes que las enfermedades cardíacas eran poco frecuentes en algunas poblaciones mediterráneas y japonesas donde las personas consumían una dieta baja en grasas. Es así como, las grasas saturadas eran una causa de enfermedades cardíacas. Luego, en 1958, F. Mason Sones, cardiólogo pediátrico de la Clínica Cleveland, desarrolló la técnica para producir imágenes de diagnóstico de alta calidad de las arterias coronarias. **La nueva prueba hizo posible un diagnóstico preciso de la enfermedad de las arterias coronarias por primera vez [52].**

La breve historia anterior muestra que la humanidad siempre ha tenido problemas relacionados con las enfermedades del corazón. Sin embargo, es en el siglo pasado que hubo un interés por asociarse y dedicar tiempo en investigaciones para encontrar alguna solución al problema de las EIC. Lo anterior muestra que las fundaciones, asociaciones, institutos, colegios, entre otros entes educativos o sin ánimo de lucro, aparecieron solamente en el siglo pasado (al igual que la ONU, donde apenas en el 2015 proponen los ODS). Esto debido al consumismo de un mundo globalizado y el poco interés por una vida saludable (esta tendencia se sigue conservando hoy en día al igual que el consumo de grasas y sal; sin embargo, ha crecido el interés por el consumo de frutas y verduras debido a sus variados beneficios sustentados por investigaciones). Por último, los especialistas cardiólogos en el país a inicios del siglo XX hacían sus estudios en el exterior, luego, a finales del siglo XX es que empieza a haber en las universidades del país las especializaciones de cardiología.

#### 4.5.6. Ventana 6 Subsistema (Pasado)

Como apenas a mediados del siglo XX empezó a existir un interés por el cuidado, la predicción y el tratamiento de las EIC, entonces en el pasado no existían sistemas o aplicaciones para la predicción del riesgo de una EIC. Sin embargo, se puede encontrar el estudio de Peter W.F. Wilson y otros [34],

cuyas características se resumen a continuación:

- **¿Cómo se utilizaba?:** El usuario introducía 6 atributos que posteriormente se le asignaban valores numéricos según fuera la información; después, se hacían unos cálculos para entregar un porcentaje del nivel de riesgo de EIC.
- **¿Qué base de datos utilizaba?:** Toda la información para diseñar el modelo fue extraída del estudio Framingham.
- **¿Cómo funcionaba?:** El modelo fue diseñado a partir de métodos estadísticos que a partir de la información de la base de datos podía entregar los valores numéricos que se le debían asignar a los atributos de entrada del usuario.
- **¿Qué método utilizaba?:** Estadística como regresión lineal y regresión logística.
- **¿Qué software o hardware utilizaba?:** No se hace mención del software o hardware utilizado.
- **¿Cómo daba la predicción?:** Niveles de riesgo muy bajo, bajo, moderado, alto y muy alto.
- **¿Qué ventajas tenía?:** Hasta la fecha recopilaba los anteriores estudios relacionados con la predicción. Además, era sencillo de utilizar.

#### 4.5.7. Ventana 7 Supersistema (Futuro)

Cuando se piensa en el futuro se realiza un plano imaginario donde se enfatiza el contexto en el que se encontrarían los productos, sistemas o aplicaciones para predecir el riesgo de una EIC.

- **Alimentación:** La alimentación saludable es otro aspecto que se debe mejorar en el futuro, porque con la tendencia actual del crecimiento de la industria de alimentos con altos contenidos en sales, grasas trans y conservantes, se tiene que, de igual forma las EIC aumentarán en el futuro. Por tal motivo, hace falta más conciencia social por el consumo de dichos alimentos, además de programas preventivos y educativos efectivos para tener una buena salud cardiovascular. Por otra parte, cabe destacar que los alimentos sanos generalmente son más costosos, así, se debe encontrar un balance entre precio y calidad para que toda la población Colombiana opte por alimentos saludables [33].
- **Especialistas médicos en cardiología:** El número de especialistas en cardiología debe aumentar de la misma forma que lo hacen las EIC. De esa manera, se debe apoyar más a los jóvenes para que estudien una carrera profesional en el área de salud y que se le den incentivos para que hagan una especialización relacionada con la cardiología.
- **Absentismo laboral:** Si las EIC siguen aumentando y se mantienen en la posición de ser la principal causa de muerte a nivel mundial, entonces, las incapacidades médicas en las empresas aumentarán por este motivo, ubicando a las EIC como una de las principales causas del absentismo laboral, que a su vez se traduce en pérdida de dinero y disminuye la productividad. Por tal razón, se espera que en el futuro diversas empresas de todos los ámbitos realicen programas de concientización y prevención de las EIC.

- **ODS:** Hasta el momento (2020), han pasado 5 años desde que la ONU propuso a los estados miembros los ODS. Se espera que para el 2030 las metas propuestas (como las del objetivo 3: salud y bienestar) por el organismo internacional sean cumplidas por todos para mejorar la calidad de vida de la población mundial.
- **Mercado de celulares con infraestructura de comunicación** El avance tecnológico exponencial del siglo XXI que se ha visto hasta ahora, muestra que en el futuro la tecnología de consumo como los celulares, tabletas, laptops, etc., estarán más desarrollados y con más funcionalidades. Sin embargo, hace falta una adecuada gestión por las empresas y el gobierno para que toda la población Colombiana puedan acceder a dichos aparatos tecnológicos que se pueden traducir en progreso, educación y comunicación con el mundo (donde también hace falta una buena y amplia infraestructura de telecomunicaciones). Se espera que la cobertura a nivel nacional, permitiendo a toda la población acceder a redes de alta velocidad.
- **Telemedicina:** En nuestro país existen nichos poblacionales que aún no tienen acceso a la tecnología, ni mucho menos a internet. Por tal razón, un aspecto a mejorar es el desarrollo tecnológico en diversos campos de la medicina y la sociedad para que se pueda implementar la telemedicina. Lo anterior debería ser apoyado principalmente por el gobierno, ya que, con la telemedicina se puede hacer seguimiento y prevención de enfermedades que, en el peor de los casos, pueden costarle al sistema de salud numerosos recursos.

#### 4.5.8. Ventana 8 Subsistema (Futuro)

La aplicación ideal en el futuro debería brindar información con alta precisión de la posibilidad de riesgo de padecer una EIC. De igual forma, debe mantener la información actualizada de los atributos que se utilizan en la predicción. Por otro lado, la aplicación debe conocer y recomendar hábitos saludables. A su vez, debería usar datos de personas de diversas poblaciones para considerar casos extremos o aislados. También, el método que utilice para dar la predicción debería ser muy rápido y que se pueda instalar en cualquier dispositivo. De igual forma, la aplicación debería ser capaz de tomar medidas al instante en el cuerpo humano a través de múltiples sensores inmersivos para el ritmo cardíaco, la presión arterial, el nivel de colesterol, sodio, y azúcar en la sangre, el peso, para poder hacer la predicción sin tener que desplazarse al centro médico. Dichos sensores pueden ser invasivos o no, pero el paciente debería tener la certeza de que son seguros y que dan medias precisas y exactas de las variables medidas. Por otra parte, cuando se dé la predicción, la aplicación dependiendo de la clasificación del riesgo, a través de un protocolo de comunicación, debería recomendarle un especialista cercano o unas recomendaciones para mantener una buena salud. Además, los datos del paciente se pueden guardar en una base de datos para hacer históricos y ver el progreso de la salud cardíaca más adelante. Por último, la aplicación debería ser de una interfaz agradable que se adapte a todo tipo de personas (personalizable), donde se pueda comunicar constantemente con el especialista, pueda acceder a contenido multimedia para alimentarse sanamente, hacer ejercicio o dormir adecuadamente, por ejemplo.

#### **4.5.9. Ventana 9 Sistema (Futuro)**

Finalmente, teniendo en cuenta la información de las ventanas anteriores y sobre todo los sistemas actuales para reducir el riesgo de padecer una EIC, el sistema futuro deberá dar la predicción de padecer una EIC de forma precisa (cercano al 100%) y registrar las predicciones para su posterior análisis. Además, dicho sistema deberá ofrecer recomendaciones de las actividades del tratamiento, como consumir los medicamentos o hacer ejercicio. También, la predicción deberá hacerse con métodos computacionales avanzados como ML y alguna de sus variantes, con el fin de garantizar al paciente una predicción confiable. Por otro lado, el software del sistema debe ser intuitivo y agradable para generar adherencia al tratamiento. De igual forma, dicho sistema deberá estar acompañado por un dispositivo que permita tomar medidas de frecuencia cardiaca para ser utilizadas en la predicción. Finalmente, el sistema futuro deberá dar a conocer al especialista el progreso del paciente, para que entre los dos tomen las medidas adecuadas respecto a la enfermedad.

## Capítulo 5

# Diseño de la aplicación para predecir el riesgo de padecer isquemia cardiaca

En este capítulo se propone una alternativa de solución a la problemática planteada, teniendo en cuenta la información y el análisis recolectado por la técnica TRIZ. Adicionalmente, se utiliza la segunda herramienta que ofrece TRIZ, es decir, la matriz de contradicciones, que ayuda a encontrar los principios que dan la mejor solución a las contradicciones halladas. Se toma como referencia actual para la evolución del producto a **HeartWatch** del desarrollador Tantsissa, la cual es una aplicación utilizada junto a un reloj de pulso inteligente **Watch de Apple**, para hacer mediciones del pulso cardiaco, notificar irregularidades en este y hacer seguimiento del sueño.

### 5.1. Matriz de contradicciones

La matriz de contradicciones es una matriz establecida por el seguimiento del método TRIZ, utilizada cuando dos parámetros técnicos son opuestos. Para emplear dicha matriz, se debe partir del requerimiento base y generar un enunciado donde se plantee una contradicción. De esa manera, de dicho enunciado se derivan dos características, una de ellas es el parámetro que se mejora y la otra, el parámetro que se empeora cuando se aplica una solución preliminar. Adicional a ello, esta matriz proporciona principios inventivos, que sirven para solucionar la contradicción generada por los parámetros. Además, este análisis se hace teniendo en cuenta la información suministrada en la técnica de las nueve ventanas. De esta manera, se detectaron las siguientes contradicciones:

#### 5.1.1. Contradicción 1

Se desea que el sistema pueda hacer una predicción del riesgo de padecer una EIC, pero, no se puede hacer con el sistema actual (HeartWatch) porque este no tiene la información suficiente de pacientes con EIC para hacer la predicción.

- Parámetro que se mejora: Confiabilidad (27)
- Parámetro que se empeora: Pérdida de información (24)

Con la matriz de contradicciones se buscan los principios que dan solución a la contradicción planteada:

Parámetro que mejora → Parámetro que empeora ↓	Confiabilidad
Pérdida de información	10, 28

Figura 5.1: Principios de solución para la contradicción N° 1

Al leer cada una de las definiciones y “tips” de solución de cada principio, se podría realizar un subsistema híbrido que utilice un sistema que necesite de información previa de pacientes con EIC y otro sistema que no necesite de dicha información, sino de los valores o parámetros que permiten determinar el riesgo de la EIC. Para el primer método, se puede utilizar algunas de las variantes de Machine Learning; para el segundo método, se podría utilizar lógica difusa (estos dos métodos tienen buenos resultados según el análisis hecho en las nueve ventanas). De esta manera, consecuentemente se podría hacer una integración de ambos métodos para mejorar la predicción.

### 5.1.2. Contradicción 2

Se desea favorecer la adherencia tratamiento médico (medicamentos y recomendaciones saludables) de una EIC, pero, no se puede hacer con el sistema actual (HeartWatch) porque no está diseñado para integrar dicho tratamiento.

- Parámetro que se mejora: Conveniencia de uso (33)
- Parámetro que se empeora: Pérdida de sustancia (23)

Con la matriz de contradicciones se buscan los principios que dan solución a la contradicción planteada:

Parámetro que mejora → Parámetro que empeora ↓	Conveniencia de uso
Pérdida de sustancia	28, 32, 2, 24

Figura 5.2: Principios de solución para la contradicción N° 2

Para favorecer la adherencia al tratamiento, según los principios TRIZ, se puede implementar una serie de alarmas visuales, sonoras o físicas (vibración) que influya al usuario a realizar las actividades de su tratamiento. Dichas alarmas deben ser personalizables (para que no resulten tediosas o poco “atractivas” para el usuario). Por otro lado, se puede presentar una serie de recomendaciones que se muestren en forma de noticias, links, conferencias, videos, páginas de redes sociales, etc., con temáticas de alimentación saludable, ejercicio o lecturas relacionadas con el tratamiento.

### 5.1.3. Contradicción 3

Se desea que el sistema integre la medición de la frecuencia cardiaca a través de un dispositivo inalámbrico, sin embargo, el sistema actual (HeartWatch) requiere de un dispositivo de medición

costoso y de una manufacturabilidad muy compleja.

- Parámetro que se mejora: Complejidad de control (37)
- Parámetro que se empeora: Manufacturabilidad (32)

Con la matriz de contradicciones se buscan los principios que dan solución a la contradicción planteada:

Parámetro que mejora →	Complejidad de control
Parámetro que empeora ↓	
Manufacturabilidad	5, 28, 11, 29

Figura 5.3: Principios de solución para la contradicción N° 3

Con la ayuda del principio 5 se puede idear un dispositivo que el usuario utilice con frecuencia, como un reloj, una pulsera o algún tipo de oxímetro. Dicho artefacto debe ser menos costoso que el Apple Watch (sistema de referencia), barato, confiable y de fácil manipulación de las variables medidas.

## 5.2. Requerimientos del producto

Gracias al resultado de las contradicciones anteriores y las recomendaciones y sugerencias de potenciales usuarios, se obtienen los siguientes requerimientos funcionales (RF) del producto:

1. Predecir el riesgo de una EIC a través de un método computacional.
2. Presentar una serie de recordatorios informativos sobre las actividades del tratamiento.
3. Adquirir los valores de la frecuencia cardiaca para ser usados en la predicción de la EIC.

Para el primer requerimiento, *Predecir el riesgo de una EIC a través de un método computacional*, es necesario contar con la plataforma líder en este tipo de trabajos computacionales. De igual forma, la implementación del método se debe hacer en tecnología y sistema operativo de bajo costo (como un celular Smartphone con Android) para que muchas personas puedan hacer uso de él.

Ahora bien, para el segundo requerimiento, *Presentar una serie de recordatorios informativos sobre las actividades del tratamiento*, se debe contar con un sistema que el usuario use constantemente y que permita el correcto seguimiento del tratamiento. Así, se debe contar con tecnología de gran uso cotidiano como un Smartphone, Tablet o Computador que cuente con un sistema operativo de mayor demanda como Android, iOS o Windows. Por otro lado, dichos recordatorios deben ser personalizados tanto por el usuario como por el especialista, para que se pueda generar adherencia al tratamiento.

Finalmente, para el tercer requerimiento, *Adquirir los valores de frecuencia cardiaca para ser usados en la predicción de la EIC* se debe contar con un sistema hardware que pueda hacer mediciones con bastante precisión de la frecuencia cardiaca, además de ser de bajo costo para que muchas

personas puedan acceder a él. Dichas mediciones se pueden enviar al sistema principal a través de un sistema inalámbrico como WiFi o Bluetooth para mayor comodidad y autonomía [32].

### 5.3. Parámetros de diseño para cada requerimiento funcional

Las necesidades del usuario/cliente se traducen como *atributos del cliente*. En el dominio funcional, estos atributos se convierten en un conjunto de requerimientos funcionales como los que se describieron anteriormente. Luego, en el dominio físico se escoge un conjunto de Parámetros de Diseño que son soluciones técnicas para satisfacer dichos Requerimientos Funcionales. Dichos parámetros de diseño vienen determinados por unos Rangos de Diseño, por fuera de los cuales un determinado parámetro de diseño no se considera satisfactorio. A medida que se establecen los parámetros de diseño, surgen unos sub-requerimientos funcionales, que a vez necesitan de sub-parámetros de diseño, llevando una orientación en zigzag para que se pueda cumplir con el requerimiento funcional principal. De esta manera, se procede a la descripción de los parámetros de diseño para los requerimientos ya planteados.

#### 5.3.1. Parámetros de diseño para el Requerimiento Funcional N°1

De acuerdo con el análisis hecho en las nueve ventanas, hay dos métodos para inferir un posible padecimiento de una EIC a partir de los atributos de un paciente. Dichos métodos son el Machine Learning y la Lógica difusa. Por el lado de Machine Learning, en cuanto a entrenamiento supervisado, este necesita una base de datos con los atributos más relevantes respecto a la salud cardiovascular de una persona, junto con un valor verdadero (target) que diga si aquel paciente descrito con esos atributos padece o no de una EIC. Ahora bien, con la lógica difusa, no se necesita de una base de datos, solo se requiere un conjunto de reglas que tengan que ver con los atributos de un paciente. De esa manera, atributos como la edad, la frecuencia cardiaca o la presión arterial se subdividen en etiquetas lingüísticas (edad: joven, adulto, viejo; FC: baja, media, alta; Presión: baja, media, alta) que luego se combinan (reglas difusas) y se les asigna un valor de salida. Por ejemplo, si un paciente posee una edad de 20 años (joven), con una frecuencia cardiaca de 80 lpm (media) y una presión de 110 mm/Hg (media), entonces esta persona no tiene riesgo de padecer una EIC. Sin embargo, a medida que aumentan los atributos y la subdivisión en etiquetas lingüísticas, con la lógica difusa aumenta enormemente la cantidad de reglas difusas. Por ejemplo, según Alexandre Vallée [29] y otros autores, un buen modelo para predecir una EIC, debe contar con 8 o más atributos para que este sea confiable. De esa manera, si se subdividen esos 8 atributos en 3 etiquetas lingüísticas, se tendrían en total 6561 reglas difusas al combinar cada una de las etiquetas lingüísticas. Ahora bien, el problema con el resultado anterior es que se debe conocer el valor de salida de cada una de las reglas difusas, y para ello, se debe contar con la opinión de un experto. Lo anterior se hace tedioso y difícil para este trabajo de grado, ya que no se cuenta con la disposición de un médico o especialista para preguntarle por los 6561 resultados de las reglas difusas. De esa manera, **a pesar de que, en el análisis de las nueve ventanas, la lógica difusa obtuvo una alta precisión [25], no es posible implementar un modelo que cuente con al menos 8 atributos.** El anterior análisis

permite justificar la elección de Machine Learning como parámetro de diseño para predecir la EIC, el cual tiene un rango de diseño donde se entrega la predicción en forma de clasificación binaria: 1 o 0 para presencia o ausencia de EIC respectivamente.

Ahora, un sub-requerimiento es elegir en que lenguaje de programación y plataforma se va a implementar el modelo ML. En la actualidad existen varias opciones como MatLab, Python, C++. Sin embargo, el lenguaje líder en Machine Learning es Python [56], además de ser abierto y con una amplia documentación. Por ello, se elige a Python como el lenguaje de programación para el modelo. Luego, otro sub-requerimiento es elegir la base de datos de entrenamiento del modelo; de esa manera, se elige como parámetro de diseño la información contenida en *UCI Repository* [26] ya que, la mayoría de los autores en el análisis de las nueve ventanas acuden a esta base de datos para realizar sus trabajos. Después, la elección del modelo toma en cuenta principalmente la precisión. De esa manera, se implementa un modelo supervisado de clasificación como una red neuronal artificial, regresión logística o Random Forest, los cuales, en la ventana 1 tuvieron una alta precisión. Ahora, se procede a decidir en qué plataforma el usuario va a interactuar con el modelo, para ello, se toma en cuenta plataformas abiertas y de gran uso como Windows y Android. Además, para implementar el modelo ML en Android o Windows, se puede elegir entre BeeWare, Chaquopi, Pytorch o Tensorflow. Finalmente, se requiere que el resultado de las predicciones se pueda guardar y compartir con personas como el especialista. Para lo anterior, se crea una interfaz en la aplicación que utilice una base de datos local o en internet para guardar la información de cada predicción. También, se crea una “opción” dentro de la interfaz que de la posibilidad al paciente de compartir su predicción con otros usuarios de la aplicación. El resumen de los parámetros de diseño para el requerimiento 1 se presenta en la figura 5.4.

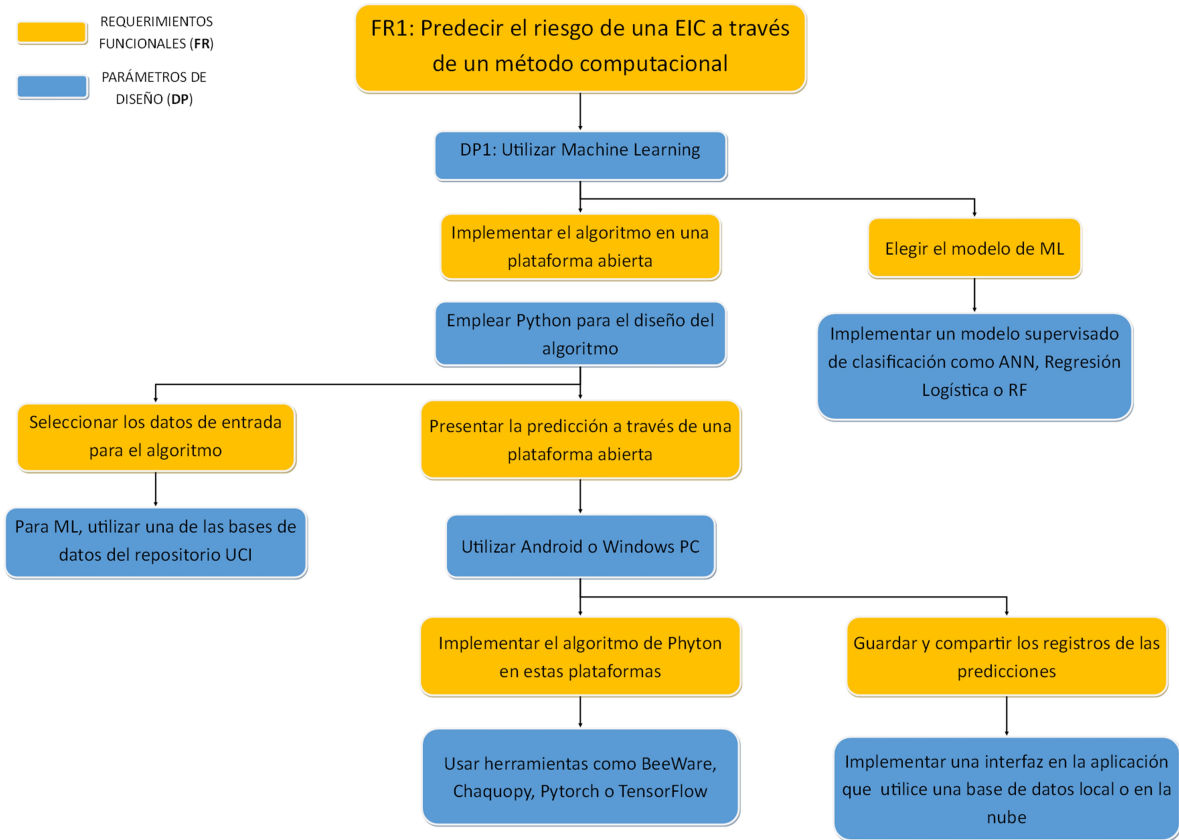


Figura 5.4: Diagrama del requerimiento funcional N°1

### 5.3.2. Parámetros de diseño para el Requerimiento Funcional N°2

Para este requerimiento, se toma en cuenta que en el tratamiento de la EIC se plantean actividades como el ejercicio, la comida y el consumo de medicamentos. Las anteriores actividades, a menudo no son realizadas por las personas, ya sea por falta de motivación u olvido. Por esta razón, se plantea como parámetro de diseño utilizar medios digitales como un Smartphone o Tablet para que el paciente pueda estar atento y realizar las actividades del tratamiento. El rango de diseño para este parámetro es que los medios digitales deben ser de gran uso y que puedan ser de código abierto para tener un acceso universal. Luego, un sub-requerimiento es poder alertar al paciente sobre los recordatorios en un determinado tiempo y con una cierta repetición (por ejemplo: tomar medicamentos cada 7 horas), para ello, se plantea utilizar el parlante o vibrador de un Smartphone o Tablet. Después, otro sub-requerimiento es poder controlar el tipo, el horario y el contenido de cada alarma. Para ello, se debe plantear una interfaz dentro del sistema principal para poder personalizar los recordatorios. Ahora bien, un sub-requerimiento adicional es proveer al paciente recomendaciones sobre hábitos saludables, para ello, en el sistema principal se plantea integrar una sección en donde se listen las recomendaciones y se presente información de fuentes confiables. Además, un sub-requerimiento adicional, es poder elegir qué tipo de información de hábitos saludables se debe presentar al paciente, para ello, el especialista o médico contará con una interfaz que le permitirá compartir recomendaciones de acuerdo con el caso de cada paciente. Finalmente, se requiere que los recordatorios se conserven y se puedan compartir otras personas. Para lo anterior, se crea una interfaz en la aplicación que utilice

una base de datos local o en la nube para guardar la información de los recordatorios. También, se crea una “opción” dentro de la aplicación que permita al paciente de compartir los recordatorios con otras personas. El resumen de los parámetros de diseño para el requerimiento 2 se presenta en la figura 5.5.

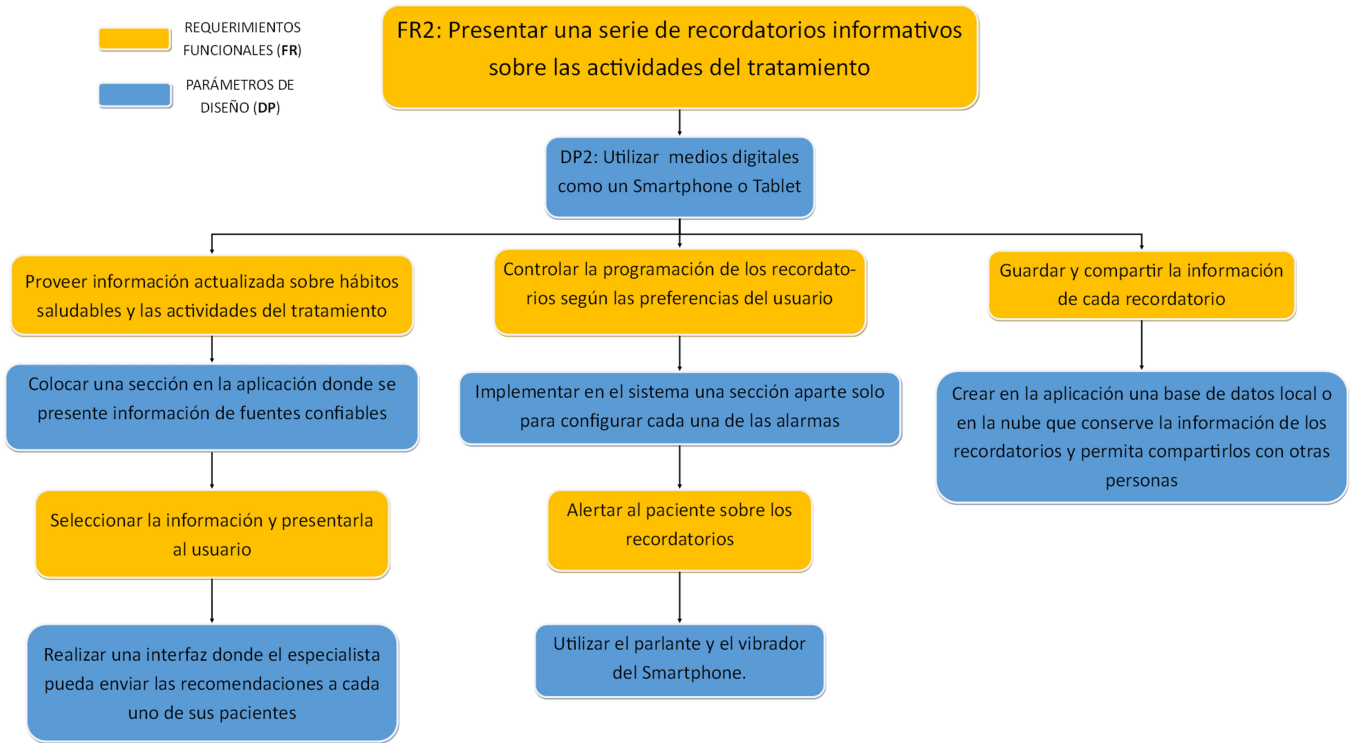


Figura 5.5: Diagrama del requerimiento funcional N°2

### 5.3.3. Parámetros de diseño para el Requerimiento Funcional N°3

Para el último requerimiento, se considera utilizar un sensor de bajo costo y que sea capaz de entregar datos confiables y en tiempo real de la frecuencia cardiaca. Por esta razón se tiene en cuenta al sensor MAX30102 como parámetro de diseño para la medición de la frecuencia cardiaca. El rango de diseño para este parámetro es que la medición de FC esté entre 30 lpm y 200 lpm. Estos valores se toman de lo mínimo y lo máximo que una persona puede tener de frecuencia cardiaca. Después, aparece un sub-requerimiento, el cual indica que se debe crear la conexión entre el sensor y el sistema principal. Para lograr lo anterior, se propone utilizar el microcontrolador ESP32, el cual cuenta con WiFi y Bluetooth para la transmisión y recepción de datos. De esa manera, se conecta el MAX30102 de forma cableada al ESP32 y se envían los registros de FC a través de WiFi o Bluetooth. Sin embargo, para lograr la autonomía del sensor y el módulo ESP32, se plantea otro sub-requerimiento, el cual consiste en alimentar energéticamente al sensor, de esa manera, se piensa utilizar baterías. Luego, surge el sub-requerimiento de alojar el sensor, el módulo y las baterías a una estructura de protección; para ello, se plantea simular un modelo 3D que integre todos los elementos anteriores y que permita una medición cómoda de la FC. Dicho modelo (contenedor de los dispositivos hardware con condiciones de ergonomía y presentación estética) se debe materializar físicamente a través de una impresora 3D. Además, entra en escena el sub-requerimiento de enviar los datos de FC al sistema

principal. Para lograr lo anterior, se plantea diseñar un algoritmo junto con una interfaz que permita la lectura, interpretación y presentación de la frecuencia cardiaca. Finalmente, se requiere que las mediciones de FC se conserven para contar con unos históricos y se puedan compartir otras personas. Para lo anterior, se crea una interfaz en la aplicación que utilice una base de datos local o en la nube para guardar la información de la frecuencia cardiaca. También, se crea una “opción” dentro de la aplicación que permita al paciente compartir los históricos de FC con otros usuarios de la aplicación. El resumen de los parámetros de diseño para el requerimiento 3 se presenta en la figura 5.6.

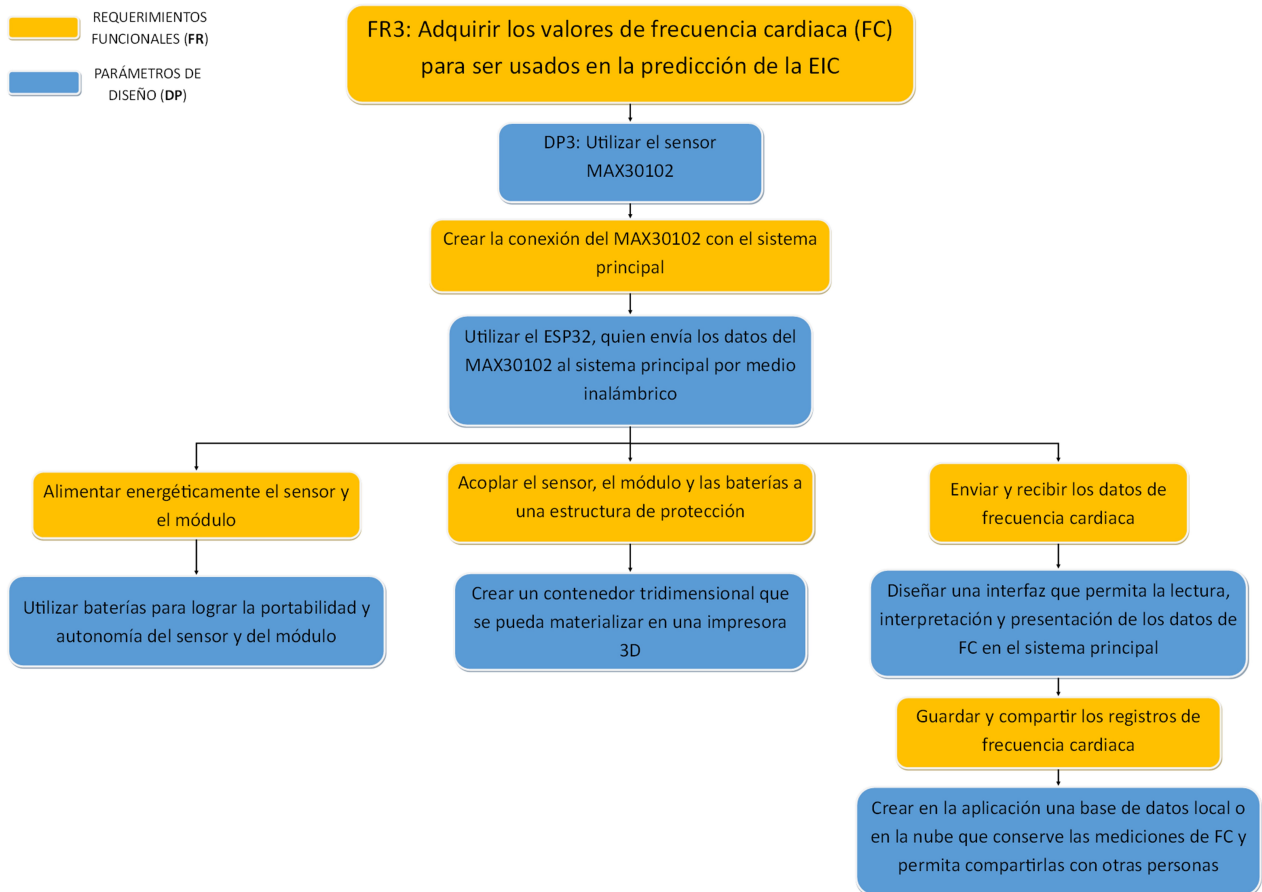


Figura 5.6: Diagrama del requerimiento funcional N°3

En la figura 5.7 se resumen los 3 requerimientos que surgieron en el análisis de las nueve ventanas y la matriz de contradicciones, junto con los parámetros de diseño.

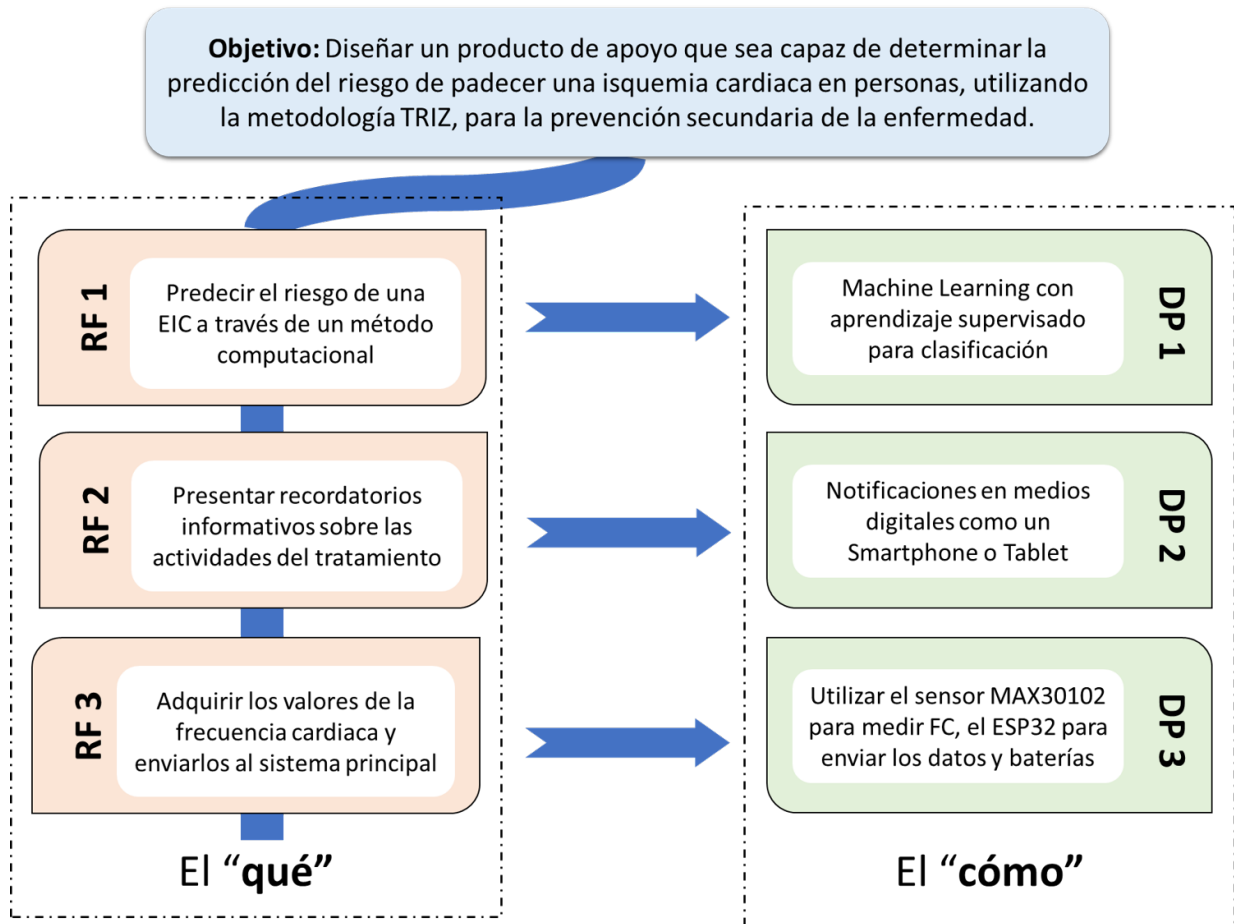


Figura 5.7: Resumen de los requerimientos del sistema

## 5.4. Descripción de los parámetros de diseño

Ahora, se procede a la descripción de cada uno de los elementos que hacen parte de los parámetros de diseño descritos en la sección anterior. Al inicio de cada descripción, se indicará, en el título, los requerimientos relacionados con el parámetro de diseño.

### 5.4.1. Medición de la frecuencia cardiaca (DP 3)

La medición de la frecuencia cardiaca se realiza en latidos por minuto (lpm) y como su nombre lo indica, es la cantidad de pulsaciones que el corazón realiza en un minuto. Esta medida es necesaria a la hora de hacer la predicción de la EIC, además de ser un indicador importante en la salud cardiovascular. Para conocer dicha medida, se decide utilizar el sensor MAX30102 de frecuencia cardiaca y oxígeno en sangre, el cual es un módulo biodetector integrado de oxímetro de pulso y monitor de frecuencia cardiaca [57]. Ahora bien, respecto a la elección de ese sensor, se observaron diferentes formas en las que se podía medir la FC; entre ellas se encontraba el uso de una manilla para la muñeca (o un reloj de pulsera) que tenía un sensor que, al entrar en contacto con el brazo, hacía las mediciones de FC. Sin embargo, este tipo de elementos resulta costoso, ya que la construcción de dicha manilla exigía un sensor robusto y sensible al movimiento ajetreado de la muñeca humana. Por otro lado, estaba la opción de utilizar el flash y la cámara de un Smartphone para realizar la

medición. Sin embargo, dichas mediciones no eran muy fiables y se prestaban para errores, además de que no se encuentra mucha documentación sobre la programación de una rutina para medir la FC con el Smartphone. Por ello, se eligió el MAX30102 para el proyecto, ya que era de bajo precio y sus componentes dedicados específicamente para medir la FC permitían tener mediciones correctas de esta variable.

### Sensor MAX30102:

Este dispositivo integra un LED rojo y un LED infrarrojo, fotodetector, componentes ópticos y circuitos electrónicos de bajo ruido con supresión de luz ambiental. El también denominado pulsioxímetro MAX30102 necesita de dos voltajes para funcionar: 1.8V para alimentar la circuitería y un voltaje entre 3.3V y 5V para los leds rojo e infrarrojo. Este módulo incluye reguladores para los voltajes anteriores en la placa, por lo que solo se necesita una fuente de 5V para la alimentación. Su consumo de corriente es mínimo, por lo que es ideal para aplicaciones portátiles (cuenta con dimensiones de 20.6 mm x 15.5 mm). Cabe decir que el MAX30102 se utiliza para la frecuencia cardiaca y la adquisición de oxígeno en la sangre principalmente en dispositivos portátiles, equipos de monitoreo médico, asistentes de estado físico y “wearables” en general (smartwatch) que se usan en los dedos, el lóbulo de la oreja y la muñeca, además de ser de bajo precio (alrededor de los COP 22000) [58]. La interfaz de comunicación es estándar, compatible con I2C. Además, puede transmitir los valores recopilados a Arduino y otros microcontroladores para el cálculo del ritmo cardiaco y el oxígeno en la sangre integrando un módulo ADC de 16 Bits que realiza hasta 1000 muestras por segundo para entregar una óptima medición. Debido a su excelente rendimiento, el MAX30102 se usa ampliamente en los teléfonos móviles de la serie Samsung Galaxy S. La forma física del sensor se presenta en la figura 5.8.

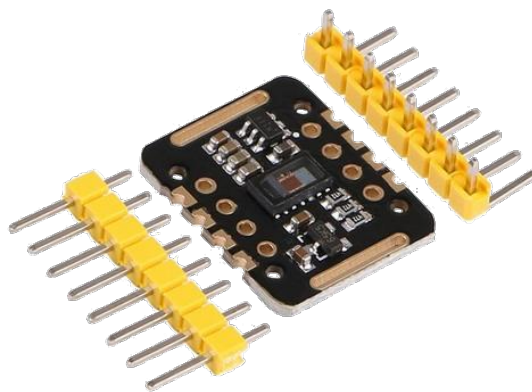


Figura 5.8: Pulsioxímetro MAX30102

### Principales Características del sensor MAX30102:

- Voltaje de alimentación DC de 5V

- Regulador de voltaje de 1.8V integrado en la placa para el circuito
- Regulador de voltaje de 3.3V integrado en la placa para los Leds internos
- Protocolo de comunicación I2C (compatible con 3.3V 5V)
- Longitud de onda del Led rojo: 680nm
- Longitud de onda del Led IR: 880nm
- Filtro de luz entre 50 Hz y 60 Hz
- Integra conversor análogo a digital ADC de 16 Bits

#### **Pines del sensor MAX30102:**

- SCL: Reloj del bus I2C
- SDA: Datos del bus I2C
- INT: Pin de interrupción del chip MAX30102
- RD: Terminal cátodo del Led rojo del chip MAX30102. Generalmente no se conecta
- IRD: Terminal cátodo del Led infrarrojo del chip MAX3010. Generalmente no se conecta
- GND: Cable a tierra.

#### **5.4.2. Trasmisión de los datos de frecuencia cardiaca (DP 3)**

Los datos que entrega el sensor MAX30102 deben ser enviados al sistema principal. Para ello, se elige el microcontrolador ESP32, el cual cuenta interfaces de Bluetooth y WiFi, que permiten el envío de información por medio inalámbrico. Lo anterior, permite que la cantidad de elementos hardware se reduzca, ya que otros microcontroladores, como Arduino Nano, necesitan un módulo externo Bluetooth o Wifi para la transmisión inalámbrica de información. Además, cuando se realice la integración de los elementos hardware en un contenedor, por ejemplo, no se necesitará mucho espacio gracias al diseño compacto del ESP32.

#### **Microcontrolador ESP32:**

El ESP32 es un SoC (System on Chip) que integra el procesador de 32 bits LX6, de doble núcleo que trabaja en modo normal a 160MHz (pudiendo alcanzar los 240MHz) [59]. Este microcontrolador está diseñado para dispositivos móviles, dispositivos electrónicos portátiles y aplicaciones de IoT. Además, el ESP32 logra un consumo de energía ultra bajo con una combinación de varios tipos de software patentado. El ESP32 también incluye características de vanguardia, como la sincronización de reloj de grano fino, varios modos de energía y escalado dinámico de energía. Para un correcto funcionamiento del ESP32, es necesario suministrar un voltaje de entre 2,8V y 3,6V. Cabe resaltar que el ESP32 es de bajo precio, rondado su valor los COP 25000. Para este proyecto, se pretende

utilizar el módulo Bluetooth integrado en el ESP32 para el envío de datos hacia la plataforma principal, ya que permite que el usuario no esté dependiendo de una conexión a internet a través de una red WiFi para la medición de la frecuencia cardiaca. En la figura 5.9 se presenta una imagen del microcontrolador en cuestión.

**Conectividad:**

- El módulo ESP32 dispone de todas las variantes de WiFi:-802.11 b/g/n/e/i/n.
- WiFi Direct (P2P), P2P Discovery, P2P Group Owner mode and P2P Power Management.
- Bluetooth v4.2 BR/EDR and BLE
- BLE Beacon.
- Además, puede comunicarse mediante los protocolos SPI, I2C, UART, MAC Ethernet, Host SD.

**El ESP32 dispone de 48 pines:**

- 18 ADC de 12 bits
- 2 DAC de 8 bits
- 10 pines sensores de contacto
- 16 PWM
- 20 Entradas/salidas digitales

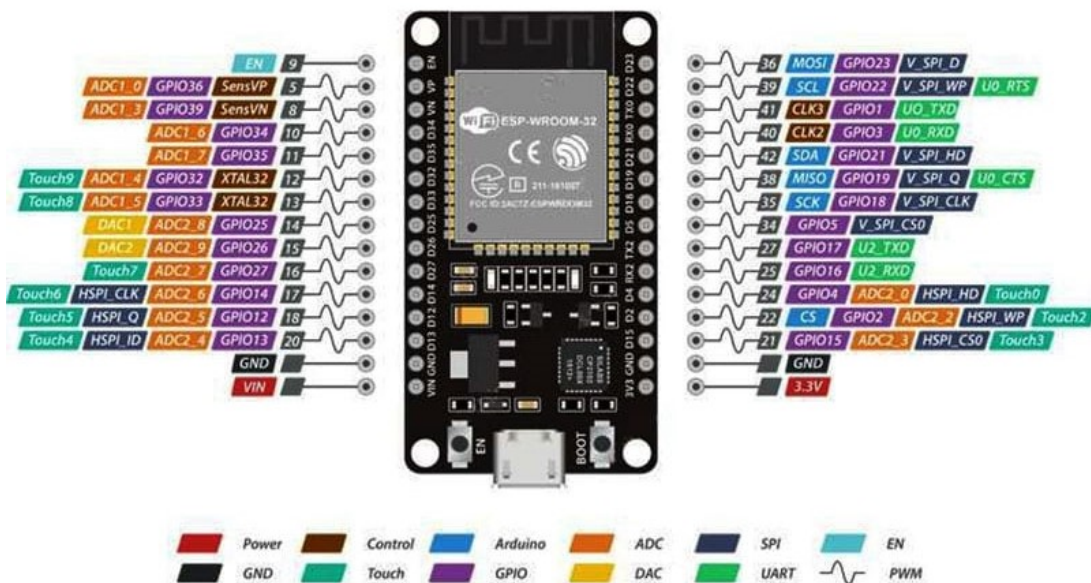


Figura 5.9: PinOut del ESP32

**Arduino IDE:**

El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores como la ESP32. El IDE de Arduino admite los lenguajes C y C++ utilizando reglas especiales de estructuración de códigos. El IDE de Arduino suministra una biblioteca de software del proyecto Wiring, que proporciona muchos procedimientos comunes de I/O. El código escrito por el usuario solo requiere dos funciones básicas, para iniciar el código (`void setup()`) y el ciclo principal del programa, que se compilan y vinculan con un apéndice de programa `main()` en un ciclo con el GNU toolchain, que también se incluye. El IDE de Arduino emplea el programa *avrdude* para convertir el código ejecutable en un archivo de texto en codificación hexadecimal que se carga en la placa Arduino, ESP32 u otra mediante un programa de carga en el firmware de la placa [72]. De esa manera, Arduino IDE será la herramienta que permitirá la programación del código que se implementará en el ESP32 para medir del MAX30102 la frecuencia cardíaca.

#### 5.4.3. Plataforma de desarrollo del sistema principal (DP 1 Y DP 2)

Existe una diversidad de herramientas para el desarrollo de aplicaciones que utilicen un modelo ML para el sistema operativo Android (que se eligió por ser un SO de gran uso a nivel global, permitiendo llegar a más personas). Dentro de ellas, está la herramienta oficial de Android para el desarrollo de sus aplicaciones denominada Android Studio, la cual, permite el despliegue de múltiples modelos ML provenientes de diversas plataformas o entornos de desarrollo Python, en aplicaciones de alto nivel. Además, Android Studio cuenta con una documentación bastante amplia, ya sea, de manera oficial o en foros, videos, tutoriales de los usuarios de esta plataforma. De igual forma, su interfaz gráfica y procesador de código hacen que el diseño de aplicaciones sea intuitivo, rápido y fácil.

##### **Android Studio:**

Las aplicaciones móviles nativas son las que se desarrollan específicamente para cada sistema operativo, iOS o Android, adaptando a cada uno el lenguaje con el que se desarrolla: lenguaje Objective-C para iOS y Java para Android. Es decir, hay que realizar el desarrollo de aplicaciones móviles para cada plataforma. Entre las ventajas de este tipo de aplicaciones destacan que aprovechan las funcionalidades del dispositivo y que pueden funcionar sin conexión a Internet. De esa manera, Android Studio permite la programación y el diseño de aplicaciones de forma nativa.

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android [65]. Ha sido diseñado específicamente para el desarrollo de Android. Desde el 7 de mayo de 2019, Kotlin es el lenguaje preferido de Google para el desarrollo de aplicaciones de Android. Aun así, Android Studio admite otros lenguajes de programación, como Java y C++. Android Studio ofrece más funciones que aumentan la productividad cuando se desarrolla aplicaciones para Android, como las siguientes:

- Un sistema de compilación flexible basado en Gradle (ver **Anexo B** [9.2]).
- Un emulador rápido y cargado de funciones, junto con un entorno unificado donde se puede

desarrollar para todos los dispositivos Android.

- Integración con GitHub y plantillas de código para compilar funciones de aplicaciones comunes y también importar código de muestra.
- Variedad de marcos de trabajo y herramientas de prueba.
- Compatibilidad integrada con Google Cloud Platform, que facilita la integración con Google Cloud Messaging y App Engine.

La lógica de las aplicaciones en Android Studio usa Java o Kotlin como lenguaje de programación. De esa manera, Java será la herramienta que permitirá la programación del sistema principal junto con los demás subsistemas e interfaces que se han mencionado en secciones anteriores (conectividad con el ESP32, creación de recordatorios y la implementación de Machine Learning en la aplicación Android). Por otro lado, la interfaz gráfica se hará en el lenguaje xml, el cual ya está integrado en Android Studio. En el **Anexo B 9.2** se da una explicación de cómo realizar una interfaz de aplicación en Android Studio.

#### 5.4.4. Plataforma para guardar la información del usuario (DP 1, DP 2 y DP 3)

Entre los sub-requerimientos de los tres requerimientos principales, está el de guardar la información de las recomendaciones, los recordatorios, las predicciones y los registros de frecuencia cardiaca. Además, la aplicación de este proyecto parte de la descripción del sistema actual que se describió en la ventana 1 del análisis de nueve ventanas. De esa manera, pensando en un producto de software, se busca diseñar una aplicación que permita al usuario guardar datos y llevar registro de sus actividades si por alguna razón el usuario cambia de dispositivo o desinstala la aplicación. Así, se necesita una base de datos en la nube que permita guardar la información del usuario y entregarla a este cuando la necesite. Por ello, entre las diversas opciones se elige a **Firestore**, que tiene la capacidad de guardar datos, archivos y de implementar funciones interesantes como la comunicación con el médico o especialista que trata el caso de cada paciente. Además, Firestore se integra muy bien para el desarrollo de aplicaciones en Android Studio, es fácil de implementar y es gratis (solo se necesita de una cuenta en Google para empezar a utilizar Firestore).

##### **Firestore:**

Firestore de Google es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. Está disponible para distintas plataformas (iOS, Android y web), con lo que es más rápido trabajar en el desarrollo. Además, la función esencial de Firestore es hacer más sencilla la creación de tanto aplicaciones webs como móviles y su desarrollo, procurando que el trabajo sea más rápido, pero sin renunciar a la calidad requerida. Sus herramientas son variadas y de fácil uso, considerando que su agrupación simplifica las tareas de gestión a una misma plataforma. También, Firestore es especialmente interesante para que los desarrolladores no necesiten dedicarle tanto tiempo al *backend*, tanto en cuestiones de desarrollo como de mantenimiento.

### 5.4.5. Base de datos del algoritmo Machine Learning (DP 1)

La base de datos que se utilizará en el algoritmo ML es la que se encuentra en el repositorio UCI [26]. En dicho repositorio existe una sección aparte donde se listan diferentes bases de datos relacionadas con las enfermedades del corazón. Esta base de datos contiene 76 atributos, pero todos los experimentos publicados se refieren al uso de un subconjunto de 14 de ellos. En particular, la base de datos de Cleveland es la única que han utilizado los investigadores de ML para esta fecha. En la figura 5.10 se observan 5 de las 297 muestras de la base de datos de Cleveland. De igual forma, en la figura 5.11 se listan las principales estadísticas de las muestras de la base de datos de Cleveland.

	atr1	atr2	atr3	atr4	atr5	atr6	atr7	atr8	atr9	atr10	atr11	atr12	atr13	target
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0

Figura 5.10: Base de datos de Cleveland

El campo “target” se refiere a la presencia de enfermedad cardiaca en el paciente. Tiene un valor entero de 0 (sin presencia) a 4. Los experimentos con la base de datos de Cleveland se han concentrado en simplemente intentar distinguir la presencia (valores 1, 2, 3, 4) de la ausencia (valor 0). Las demás bases de datos (hungarian, long-beach y switzerland), están incompletas, respecto a que faltan valores en alguno de los 13 atributos de varias muestras. De esa manera, se utilizará la base de datos de Cleveland para el entrenamiento e implementación del algoritmo ML del proyecto en cuestión. La descripción de los atributos se detalla en Anexo C 9.3. Ahora bien, los 13 atributos que se utilizan en los experimentos se listan a continuación:

1. Edad: En años.
2. Género: Masculino=1, Femenino=0.
3. Tipo de dolor de pecho: Atípica angina=1, Típica angina=2, Asintomático=3, No hay dolor de pecho=4. La angina es un tipo de dolor de pecho causado por la reducción del flujo de sangre al corazón. Algunas personas con síntomas de angina dicen que se siente como si una prensa les apretara el pecho o tuvieran un peso grande sobre el pecho [60].
4. Presión sanguínea en reposo: En mm Hg.
5. Colesterol sérico: En mg/dl.
6. Azúcar en ayunas mayor a 120 mg/dl: Verdadero=1, Falso=0.
7. Resultados de electrocardiograma en reposo: Normal=0, Anomalía en la onda ST=1, Hipertrofia ventricular izquierda probable o definitiva=2. Al realizar un electrocardiograma, el segmento

ST, en condiciones normales, es plano o isoelectrico, aunque puede presentar pequeñas variaciones menores de 0.5 mm. El descenso del segmento ST de forma aguda, es un signo de daño miocárdico, al igual que la elevación [61]. Por otro lado, la hipertrofia ventricular izquierda es el agrandamiento y engrosamiento (hipertrofia) de las paredes de la cavidad de bombeo principal del corazón (ventrículo izquierdo). La pared del corazón engrosada pierde elasticidad, lo que lleva a un aumento de la presión para permitir que el corazón llene su cavidad de bombeo para poder enviar la sangre al resto del cuerpo [62].

8. Máximo ritmo cardiaco alcanzado: En lpm.
9. Hacer ejercicio produce dolor de pecho: Sí=1, No=0.
10. Depresión del ST inducida por ejercicio relacionada con el reposo físico
11. Pendiente del segmento ST en el pico del ejercicio (actividad física): Ascendiendo=1, Constante=2, Descendiendo=3.
12. Número de vasos sanguíneos principales coloreados por fluoroscopia: 0 a 3. La fluoroscopia es un tipo de radiografía que muestra en tiempo real órganos, tejidos y otras estructuras internas en movimiento. Además, la fluoroscopia es como una película donde se muestra los aparatos y sistemas del cuerpo en acción. Esto incluye los aparatos cardiovasculares (corazón y vasos sanguíneos), digestivo y reproductivo. El procedimiento ayuda a los profesionales de la salud a evaluar y diagnosticar una variedad de afecciones [63].
13. Thallium scan: Normal=3, Defecto fijo=6, Defecto reversible=7. El thallium scan es un método para examinar el corazón con el fin de obtener información sobre el suministro de sangre al músculo cardíaco. Para ello, se utilizan cámaras especiales que toman una serie de imágenes del corazón empleando una sustancia radiactiva que se inyecta en el torrente sanguíneo para servir como trazador [64].

	count	mean	std	min	25%	50%	75%	max
atr1	297.0	54.542088	9.049736	29.0	48.0	56.0	61.0	77.0
atr2	297.0	0.676768	0.468500	0.0	0.0	1.0	1.0	1.0
atr3	297.0	3.158249	0.964859	1.0	3.0	3.0	4.0	4.0
atr4	297.0	131.693603	17.762806	94.0	120.0	130.0	140.0	200.0
atr5	297.0	247.350168	51.997583	126.0	211.0	243.0	276.0	564.0
atr6	297.0	0.144781	0.352474	0.0	0.0	0.0	0.0	1.0
atr7	297.0	0.996633	0.994914	0.0	0.0	1.0	2.0	2.0
atr8	297.0	149.599327	22.941562	71.0	133.0	153.0	166.0	202.0
atr9	297.0	0.326599	0.469761	0.0	0.0	0.0	1.0	1.0
atr10	297.0	1.055556	1.166123	0.0	0.0	0.8	1.6	6.2
atr11	297.0	1.602694	0.618187	1.0	1.0	2.0	2.0	3.0
atr12	297.0	0.676768	0.938965	0.0	0.0	0.0	1.0	3.0
atr13	297.0	4.730640	1.938629	3.0	3.0	3.0	7.0	7.0

Figura 5.11: Estadísticas de la base de datos de Cleveland

#### 5.4.6. Plataforma de desarrollo del algoritmo Machine Learning (DP 1)

El aprendizaje supervisado de clasificación es una de las áreas de la inteligencia artificial (IA) con más auge en la actualidad. Por lo tanto, existe una variedad de plataformas enfocadas a la construcción de sistemas con esta tecnología, cada una con características que la hacen adecuada para cierto tipo de aplicaciones. Aquí se presentarán aquellas plataformas que se ajustan a las necesidades del proyecto, permiten implementaciones rápidas y requieren de un tiempo reducido de capacitación para su uso.

##### **Tensorflow:**

Tensorflow es una plataforma de código abierto de Google de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que permite que los investigadores innoven con el aprendizaje automático y los desarrolladores creen e implementen aplicaciones con tecnología de IA fácilmente. Tensorflow permite la creación y entrenamiento de modelos de IA con facilidad mediante API intuitivas y de alto nivel como Keras, con ejecución inmediata, que permite una iteración de modelos al instante y una depuración fácil. De igual forma, Tensorflow permite entrenar e implementar con facilidad modelos en la nube, locales, en el navegador o en el dispositivo, sin importar el lenguaje de programación. Además, Tensorflow posee una arquitectura simple y flexible para llevar las ideas nuevas del concepto al código, a modelos innovadores y a publicaciones con mayor rapidez. También, en su repositorio se pueden encontrar diversos modelos de ML para ser estudiados, modificados e implementados, tales como Redes Neuronales para la clasificación de imágenes, reconocimiento de texto, detección de rostros, detección y seguimiento de objetos, detección de idioma, respuesta inteligente entre otros [66].

##### **Principales características de Tensorflow:**

- Código abierto.
- API de alto nivel para redes neuronales: Keras.
- Modelos previamente creados, algunos de ellos ya entrenados.
- Soporte para dispositivos móviles: con herramientas como Tensorflow Mobile o Tensorflow Lite, este marco de trabajo cuenta con posibilidades de optimización de los modelos para posibilitar su ejecución de manera directa en hardware móvil.

Tensorflow cuenta con una herramienta llamada Tensorflow Lite, que sirve para el despliegue de modelos de Tensorflow en dispositivos móviles y embebidos, que aplica técnicas de optimización con el fin de cumplir con las restricciones que estos dispositivos presentan en cuanto a rendimiento, tamaño y portabilidad de aquellos modelos. Tensorflow Lite permite convertir modelos previamente construidos y entrenados utilizando Tensorflow en versiones comprimidas de los mismos en un archivo de extensión *.tflite*. Este archivo puede ser posteriormente ejecutado directamente en dispositivos con limitaciones de recursos.

##### **Pytorch:**

Es una plataforma de desarrollo para aprendizaje profundo, de código abierto y mantenida principalmente por ingenieros del departamento de inteligencia artificial de Facebook, que busca brindar una experiencia de transición sencilla entre entornos de prototipado investigativo y despliegue para producción. Consiste en una biblioteca para Python que permite la realización de cálculos utilizando tensores, tanto en procesadores tipo CPU como GPU con CUDA. Destaca por sus funciones para la construcción de modelos de manera rápida y flexible, combinando distintos modos de programación y permitiendo la modificación de las redes en tiempo de ejecución, ya que la plataforma se encarga de recalculer los parámetros usados en el entrenamiento, con base en los cambios que se hagan al grafo dinámico asociado [68].

#### **Principales características de Pytorch:**

- Código abierto.
- Construido para estar altamente integrado con Python.
- Contiene definiciones para arquitecturas populares en visión computacional con aprendizaje profundo (biblioteca TorchVision).
- Permite evaluar el modelo en cualquier momento, debido a su manejo dinámico del mismo.

Los modelos desarrollados en Pytorch pueden ser ejecutados directamente en Python, por lo cual aplicaciones móviles desarrolladas en este lenguaje podrían hacer uso de ellos. De esa manera, PyTorch cuenta con una herramienta similar a Tensorflow Lite para el despliegue de sus modelos tanto en Android como iOS. Así, se crea un archivo de extensión *.plt* que se puede cargar como un módulo en Android Studio para realizar la operación de ML.

#### **Scikit-learn:**

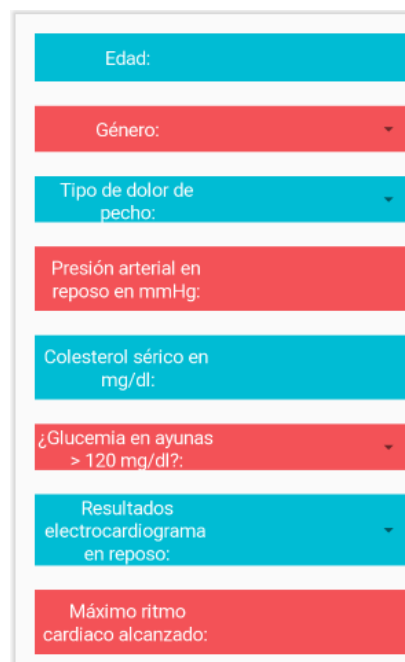
Scikit-learn es una biblioteca para aprendizaje automático de software libre para el lenguaje de programación Python. Incluye varios algoritmos de clasificación, regresión y análisis de grupos entre los cuales están máquinas de vectores de soporte, bosques aleatorios, Gradient boosting, K-means y DBSCAN. Scikit-learn está diseñada para interoperar con las bibliotecas numéricas y científicas NumPy y SciPy. Scikit-learn está escrita principalmente en Python, con algunos algoritmos de núcleo escritos en Cython para conseguir mayor rendimiento [69]. Además, cabe resaltar que Scikit-learn es la herramienta que utilizó Amin UI Haq [24] para su proyecto, obteniendo una precisión de 89 % con regresión logística. De esa manera, esta herramienta puede ser la candidata principal para implementar en la aplicación móvil. Para ello, se hizo una prueba del algoritmo de regresión logística utilizando Scikit-learn, teniendo como base de datos la que se encuentra en el repositorio UCI [26], más específicamente, los datos del Hospital de Cleveland. Dicha prueba se realiza con el código del **Anexo H 9.8**.

Al ejecutar la rutina, se obtiene una precisión del 83 %, cercana al 84 % que obtuvieron Amin UI Haq y otros [24] en su trabajo cuando utilizaron los 13 atributos con 297 muestras. La rutina del **Anexo H** funciona adecuadamente, pero, hace falta implementarla en un dispositivo móvil Android.

Para ello, se realizaron pruebas con **Chaquopy**, que es un SDK Python para Android [70]. Este SDK es la forma más sencilla de usar Python en una aplicación de Android (cabe recordar que se está usando Python como lenguaje de programación para el algoritmo ML). Chaquopy proporciona todo lo que necesita para incluir componentes de Python en una aplicación de Android, que incluye:

- Integración completa con el sistema de compilación Gradle estándar de Android Studio.
- API simples para llamar a código Python desde Java / Kotlin y viceversa.
- Una amplia gama de paquetes Python de terceros, incluidos SciPy, OpenCV, Tensorflow y muchos más.

De esa manera, se realiza una aplicación de prueba desarrollada en Android Studio, en la cual se introducen los 13 atributos del algoritmo ML y la aplicación muestra la predicción de EIC con esa nueva entrada. De esa manera, se creó la interfaz de la figura 5.12.



Edad:
Género:
Tipo de dolor de pecho:
Presión arterial en reposo en mmHg:
Colesterol sérico en mg/dl:
¿Glucemia en ayunas > 120 mg/dl?:
Resultados electrocardiograma en reposo:
Máximo ritmo cardíaco alcanzado:

Figura 5.12: Aplicación prototipo para la prueba del algoritmo con Chaquopy

### **Evaluación de las alternativas de la plataforma de desarrollo del algoritmo ML:**

Respecto a Chaquopy, a pesar de que el algoritmo en la aplicación prototipo Android entregó el resultado de la predicción, el rendimiento no fue adecuado, ya que la pantalla de la aplicación quedó congelada mientras importaba todas las librerías de Python y el SDK de Chaquopy y procesaba la predicción con los atributos que se ingresaron. De esa manera, la aplicación quedó congelada alrededor de 25 segundos. Es importante mencionar que dicha prueba se realizó en un Smartphone de gama media/baja como lo es el Nokia 1.3. En síntesis, Chaquopy abre muchas posibilidades de implementar cualquier código de Python en Android. Sin embargo, cosas como el congelamiento de pantalla y el tamaño de la aplicación (280 MB del archivo apk, ya que se importan muchas librerías de Python para el correcto funcionamiento del algoritmo ML) hacen que esta gran opción se descarte para el diseño de la aplicación para la predicción de una EIC.

Por el lado de Pytorch y Tensorflow, ambas tienen una herramienta de despliegue de sus modelos en dispositivos móviles. Si bien, tanto Tensorflow como PyTorch son de código abierto, ambas han sido creadas por dos asistentes diferentes. Por el lado de Tensorflow, este se basa en Theano y ha sido desarrollado por Google, mientras que PyTorch está basado en Torch y ha sido desarrollado por Facebook. La diferencia más importante entre los dos es la forma en que estos definen los gráficos computacionales. Mientras que Tensorflow crea un gráfico estático, PyTorch crea un gráfico dinámico. Esto significa que en Tensorflow, primero se debe definir todo el gráfico de cálculo del modelo y luego ejecutar el modelo ML. Pero, en PyTorch, se puede definir/manipular el gráfico sobre la marcha. Además, Tensorflow tiene una curva de aprendizaje más pronunciada que PyTorch. PyTorch es más orientado a Python y la construcción de modelos ML se siente más intuitiva.

Por otro lado, Tensorflow tiene una comunidad de desarrollo y aprendizaje mucho más grande que PyTorch. Entonces, en términos de recursos, se encontrará mucho más contenido sobre Tensorflow que PyTorch. Sin embargo, esta comparación estaría incompleta sin mencionar TensorBoard, la cual es una herramienta brillante que permite visualizar los modelos ML directamente en el navegador. PyTorch no tiene una herramienta de este tipo. Sin embargo, existen integraciones que le permiten usar Tensorboard con PyTorch, pero no es compatible de forma nativa. Finalmente, Tensorflow es mucho mejor para modelos de producción y escalabilidad, mientras que PyTorch es más fácil de aprender y más liviano para trabajar y, por lo tanto, es relativamente mejor para proyectos apasionantes y para la construcción de prototipos rápidos [71]. De esa manera, **el análisis anterior permite decidirse por Tensorflow para el desarrollo del algoritmo ML para el proyecto en cuestión**, además de contar con un soporte dentro de Android Studio para facilitar el despliegue del modelo ML.

#### 5.4.7. Desarrollo del modelo Machine Learning en la plataforma seleccionada (FR1)

Google Colab es la plataforma recomendada por Tensorflow para el desarrollo de sus modelos. Esta plataforma en línea permite la creación de “Notebooks” Python, facilitando el desarrollo de código, ya que no hay que instalar localmente (en el PC) librerías como Pandas, Numpy, Sklearn o Tensorflow. De esa manera, se crea un nuevo Cuaderno en Colab que comienza con la importación de las librerías que utilizará el modelo ML:

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import lite
from tensorflow import keras
from tensorflow import feature_column
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import pathlib
```

Después, se lee la información de la base de datos de Cleveland. Para ello, se eliminaron las filas con atributos faltantes. De esa manera, quedan 297 muestras para el modelo Machine Learning. Esta base de datos se encuentra en el repositorio del **Anexo C 9.3**.

```
csv_file = 'databasefinal.csv'
dataframe = pd.read_csv(csv_file)
```

Luego, para verificar la efectividad del modelo, se hace una división de la base de datos en entrenamiento y prueba. De esa manera, el 80% (238 muestras) de la información es para entrenamiento y el 20% (59 muestras) queda para una evaluación final. Lo anterior se realiza de forma aleatoria colocando el término *randomstate* del siguiente código en 0:

```
train_dataset = dataframe.sample(frac=0.8,random_state=0)
test_dataset = dataframe.drop(train_dataset.index)
train_labels = train_dataset.pop('target')
test_labels = test_dataset.pop('target')
```

Para que funcionen mejor los algoritmos de Machine Learning usados en Data Science, hay que normalizar las variables de entrada al algoritmo. “Normalizar” significa, en este caso, comprimir o extender los valores de la variable para que estén en un rango definido. Lo anterior es sugerido por varios autores en la ventana 1 del análisis de las nueve ventanas del capítulo antepasado. De esa manera, se utiliza el “Escalado estándar” para normalizar las columnas de atributos de la base de datos.

```
train_stats = dataframe.describe() //Estadísticas de cada atributo
train_stats.pop("target")
train_stats = train_stats.transpose()
def norm(x):
    return (x - train_stats['mean']) / train_stats['std']
normed_train_data = norm(train_dataset)
normed_test_data = norm(test_dataset)
```

Después, se procede con la construcción del modelo con Tensorflow. Para ello, se utiliza la librería de Keras, la cual es un framework de alto nivel para el aprendizaje automático (ML), escrito en Python y capaz de correr sobre los frameworks Tensorflow, CNTK, o Theano. Fue desarrollado con el objetivo de facilitar un proceso de experimentación rápida. Ahora bien, según la información presentada en el análisis de las nueve ventanas (ventana 1), se observó que la Regresión Logística y las Redes Neuronales Artificiales (ver sección 3.5) son la mejor opción para predecir el riesgo de una EIC con la base de datos que se está trabajando. De esa manera, se hicieron pruebas con Tensorflow y Keras para conocer cuál de las dos opciones se desempeñaba mejor. De esa manera, se plantean los siguientes modelos en Tensorflow con Keras:

```
#Regresión logística
def build_model():
    model = keras.Sequential([
        layers.Dense(1,
            activation = 'sigmoid',
            input_shape=[len(train_dataset.keys())],
        ])
    model.compile(optimizer='sgd', loss='binary_crossentropy',
        metrics=['mse', 'accuracy'])
    return model
#Red neuronal
```

```

def build_model():
    model = keras.Sequential([
        layers.Dense(13,
            activation='sigmoid',
            input_shape=[len(train_dataset.keys())]),
        layers.Dense(32, activation='sigmoid'),
        layers.Dense(1, activation='sigmoid')
    ])
    optimizer = tf.keras.optimizers.RMSprop(0.001)
    model.compile(loss='mse', optimizer=optimizer,
        metrics=['mse', 'accuracy'])
    return model

```

De la definición de la documentación de Keras, el modelo secuencial (Sequential) es una pila lineal de capas. La capa *Dense* es la capa regular de la red neuronal profundamente conectada. Es la capa más común y de uso frecuente. En Keras, se puede definir el modelo de regresión logística utilizando una única capa de red neuronal con activación *sigmoid* [15]. Ahora, para el modelo de regresión logística se elige que el tamaño de entrada de la capa sea de 13 (la cantidad de atributos) y que la salida sea de 1 (la predicción). Por otro lado, hay que recordar que la regresión logística utiliza la función sigmoide para limitar el valor de salida, por lo que en la rutina se coloca el valor de *activation* en *sigmoid*. Además, como parte del algoritmo de optimización, el error para el estado actual del modelo debe estimarse repetidamente. Esto requiere la elección de una función de error, llamada convencionalmente función *loss*, que se puede utilizar para estimar la pérdida del modelo de modo que los pesos se puedan actualizar para reducir la pérdida en la siguiente evaluación. Por otro lado, los optimizadores son algoritmos o métodos que se utilizan para cambiar los atributos de la red neuronal, como los pesos y la tasa de aprendizaje, para reducir las pérdidas. Los optimizadores se utilizan para resolver problemas de optimización minimizando la función [17]. Así, la función *compile* en las rutinas anteriores se encarga de definir el optimizador del algoritmo, la función *loss* y las métricas del algoritmo. Por otro lado, la red neuronal tiene una definición parecida a la regresión logística, solo que se agregan 2 capas adicionales y el compilado es diferente respecto a *loss* y el optimizador.

Ahora, se procede con el entrenamiento de cada uno de los modelos. Para ello, se utiliza el conjunto de datos normalizado para entrenamiento y los valores reales padecimiento de EIC *targets* de cada muestra.

```

model = build_model()
EPOCHS = 200
history= model.fit(normed_train_data,
    train_labels,
    epochs=EPOCHS,
    validation_split = 0.1,
    verbose=1)

```

En la rutina anterior, un *epoch* o época consiste en un ciclo de entrenamiento completo en el conjunto de entrenamiento. Una vez que se ven todas las muestras del conjunto, comienza de nuevo,

marcando el comienzo de la segunda época. No se puede estar seguro de si 5 épocas o 500 son suficientes para el modelo, ya que variará de un dato a otro. De esa manera, se puede colocar un valor alto para *EPOCH* y detener el entrenamiento cuando el error converja o sea inferior a un determinado umbral. Por otro lado, el termino *verbose* es simplemente un indicador para imprimir en la consola de Colab el proceso de entrenamiento. Así, si *verbose* está en 0, entonces no se verá nada en la consola, y si es 1 0 2, se verá el resultado de entrenamiento en cada época.

Ahora bien, la finalidad última del modelo es predecir el riesgo de padecer una EIC en observaciones futuras o en observaciones que el modelo no ha “visto” antes. El error mostrado por defecto tras entrenar un modelo suele ser el error de entrenamiento, que es el error que comete el modelo al predecir las observaciones que ya ha “visto”. Si bien estos errores son útiles para entender cómo está aprendiendo el modelo, no es una estimación realista de cómo se comporta el modelo ante nuevas observaciones (el error de entrenamiento suele ser demasiado optimista). Para conseguir una estimación más certera, se utilizan métodos de validación, también conocidos como *resampling*, que son estrategias para estimar la capacidad predictiva de los modelos cuando se aplican a nuevas observaciones, haciendo uso únicamente de los datos de entrenamiento. La idea en la que se basan todos ellos es la siguiente: el modelo se ajusta empleando un subconjunto de observaciones del conjunto de entrenamiento y se evalúa (calcular una métrica que mida como de bueno es el modelo, por ejemplo, la precisión) con las observaciones restantes. Este proceso se repite múltiples veces y los resultados se agregan y promedian. Gracias a las repeticiones, se compensan las posibles desviaciones que puedan surgir por el reparto aleatorio de las observaciones [75].

Keras permite realizar un *resampling* del conjunto de entrenamiento para hacer la validación del modelo. Esto se puede hacer configurando la opción argumento de *validation\_split* en la función *fit()* de la rutina anterior a un porcentaje ente 0 y 1. De esa manera, Keras puede separar una parte de los datos de entrenamiento (*normedtraindataset*) en un conjunto de datos de validación y evaluar el rendimiento del modelo en ese conjunto de datos de validación en cada época. Después de hacer el entrenamiento de los modelos con un 10% para validación, se realiza la gráfica de la figura 5.13 de la precisión del conjunto de entrenamiento y el conjunto de validación para el modelo de regresión logística y el de la red neuronal artificial.

```
#Rutina para imprimir la gráfica de precisiones
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Precisión del modelo')
plt.ylabel('Precisión')
plt.xlabel('epoch')
plt.legend(['entrenamiento', 'validación'], loc='lower right')
plt.show()
```

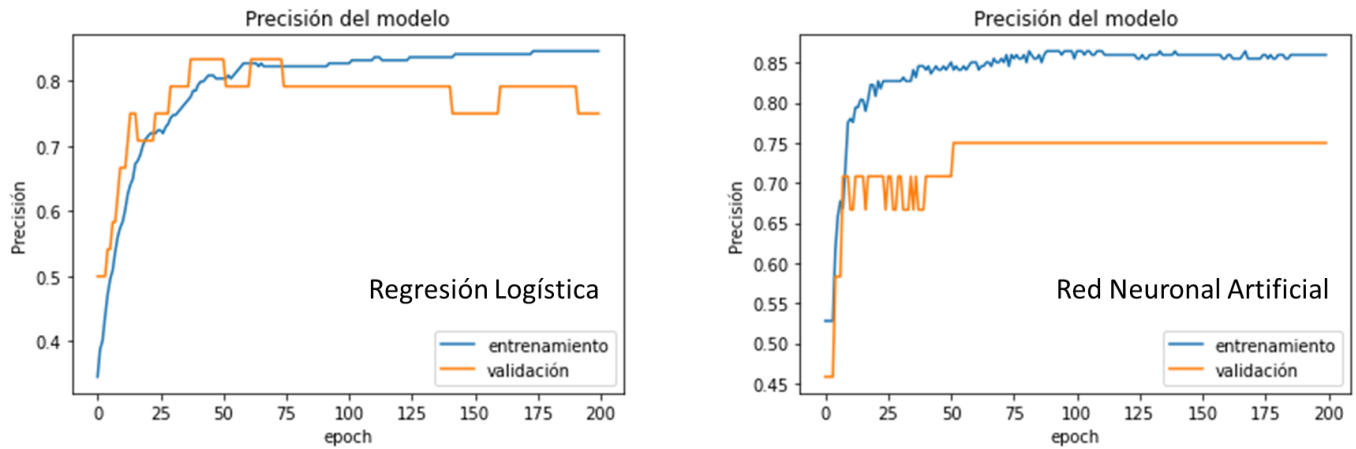


Figura 5.13: Gráfica de la precisión (entrenamiento y validación) para el modelo de Regresión Logística y la Red Neuronal Artificial

Después de entrenar los modelos con 200 épocas, se prueba con información que no ha visto antes (*normedtestdata*). Así, se conocerá cual es el desempeño de cada modelo en cuanto a precisión y error para un conjunto de datos de 59 muestras. La siguiente rutina permite evaluar el modelo entrenado y entrega las métricas de precisión y error cuadrático medio.

```
loss, mse, accuracy = model.evaluate(normed_test_data, test_labels, verbose=1)
print("Precisión con el conjunto de prueba: ", accuracy)
print("Error cuadrático medio con el conjunto de prueba: ", mse)
```

La precisión y el error cuadrático medio de esta rutina son **91.54 %** y **0.08754** respectivamente para el modelo de regresión logística. De igual forma, la precisión y el error cuadrático medio de esta rutina son **91.52 %** y **0.08598** respectivamente para el modelo de la red neuronal artificial. Por el lado de la precisión, se debe recordar que este indicador muestra el cociente de la cantidad de aciertos del modelo (verdaderos positivos y verdaderos negativos) entre el total de pruebas que es 59 (ver sección [3.5.3](#)). Por otro lado, el error cuadrático medio (ECM) mide el promedio de los errores al cuadrado. Estos dos resultados nos muestran que los 2 modelos se comportan muy bien cuando se enfrentan a información que no han conocido antes. De esa manera, al hacer una comparación de los modelos, se observa que ambos tienen un valor similar para la precisión y para el ECM.

Sin embargo, en la gráfica [5.13](#) se observa que para la regresión logística la precisión de validación es errante y no se mantiene constante. Así, propuso aumentar la cantidad de épocas de entrenamiento del modelo de regresión logística para ver su comportamiento. Luego, se obtuvo que alrededor de las 250 épocas se mantenía estable los valores de precisión con el modelo de la regresión logística, pero, el valor de precisión para la evaluación del modelo con *normedtestdata* disminuyó al 88 %. Por otro lado, el modelo de la red neuronal es adecuado con 100 épocas manteniendo una alta precisión respecto a la regresión logística cuando se evalúa con la misma cantidad de épocas. Así, **se elige a la red neuronal artificial para construir el modelo final** a pesar de que la regresión tuvo un buen resultado también. Además, para corroborar el valor de precisión de la red neuronal artificial (ya que el valor presentado anteriormente fue entregado por una rutina interna de las librerías de TensorFlow), se utiliza la matriz de confusión definida en la sección [3.5.3](#). Para ello, se evalúan las 59 muestras de prueba y se obtienen los valores predichos y se comparan con los valores reales ('target'). Así, se

obtienen 34 Verdaderos Positivos, 2 Falsos Negativos, 3 Falsos Positivos y 20 Verdaderos Negativos. De esa manera, al hacer el cálculo se obtiene un 91.52% de precisión. Finalmente, se entrena el algoritmo con todo el conjunto de datos (es decir, las 297 muestras) para crear el modelo final. Dicho modelo, se exporta mediante el conversor de Tensorflow lite [74]:

```
#Convertir el modelo
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
#Guardar el modelo
with open('modelo.tflite', 'wb') as f:
    f.write(tflite_model)
```

**Exportación del modelo e implementación en Android Studio:** El conversor de Tensorflow Lite toma un modelo de Tensorflow y genera un modelo de Tensorflow Lite (un formato FlatBuffer optimizado que se puede identificar mediante la extensión de archivo `.tflite`). De esa manera, el modelo diseñado anteriormente se convierte a un archivo de extensión `.tflite`. Dicho archivo se importa en Android Studio para ser usado en la aplicación. Ahora bien, para usar el modelo, se realiza la siguiente rutina en Java:

```
private MappedByteBuffer loadModelFile() throws IOException {
    AssetFileDescriptor assetFileDescriptor =
        this.getAssets()
            .openFd("modelo.tflite");
    FileInputStream fileInputStream = new FileInputStream(assetFileDescriptor
        .getFileDescriptor());
    FileChannel fileChannel = fileInputStream.getChannel();
    long startOffset=assetFileDescriptor.getStartOffset();
    long length = assetFileDescriptor.getLength();
    return fileChannel.map(FileChannel.MapMode.READ_ONLY,startOffset,length);}

public String doInference(){
    try {interpreter = new Interpreter(loadModelFile(),null);
    } catch (IOException e) {e.printStackTrace();}
    float[] mean = new float[13]; //El promedio de cada atributo
    float[] std = new float[13]; //La desviación estandar de cada atributo
    float[] muestra = new float[13]; //Los datos que introduce el usuario
    float[] inputs=new float[0]; //La información que entra al modelo
    for (int i=0;i<muestra.length;i++){
        inputs[i]=(muestra[i] - mean[i])/(std[i]); //Normalizar los datos del usuario
    }
    float[][] output = new float[1][1];
    interpreter.run(inputs,output); //Hacer la predicción
    float val_ML=output[0][0];
    String pred="";
    if (val_ML>=0.5){pred="SÍ";}else{pred="NO";}
    return pred;}
}
```

En la rutina anterior, primero se crea un `MappedByteBuffer` que permite cargar y utilizar el archivo `model.tflite` en Android Studio. Luego, en la siguiente función (`doInference`) se carga el intérprete de Tensorflow para hacer la inferencia (predicción). Después, se procesa la información que el usuario proporciona para hacer la predicción, ya que, en el modelo de Tensorflow se hizo una normalización

de la base de datos para obtener media de 0 y varianza de 1. De esa manera, en la rutina se debe colocar los datos de promedio y varianza de cada atributo (revisar figura 5.11). Luego de hacer la normalización, el arreglo *inputs* se convierte en la información de entrada del modelo. Así, se utiliza el comando *interpreter.run* para conocer el resultado que entrega el modelo. Finalmente, la variable *output* contiene el valor de la predicción, pero, cabe recordar que, se está utilizando una función de tipo *Sigmoid* como salida del modelo. Por tal razón, si el valor dentro de *output* es mayor o igual a 0.5, entonces, **SÍ** existe un alto riesgo de padecer una EIC, de lo contrario, hay una alta probabilidad de **NO** correr el riesgo de padecer una EIC.

En la tabla de la figura 5.14 se presenta el resumen de los elementos escogidos para el diseño de la aplicación. Cabe recordar que la aplicación se hará para dispositivos móviles Android como se mencionó en la sección 5.4.3.

Requerimientos	Parámetros de diseño	Costo
N°1	<ul style="list-style-type: none"> <li>- Python</li> <li>- Base de datos de Cleveland (UCI repository)</li> <li>- TensorFlow</li> <li>- Red neuronal artificial</li> <li>- Android Studio</li> <li>- Firebase</li> </ul>	COP \$0
N°2	<ul style="list-style-type: none"> <li>- Android Studio</li> <li>- Firebase</li> </ul>	COP \$0
N°3	<ul style="list-style-type: none"> <li>- Sensor MAX30102</li> <li>- Microcontrolador ESP32</li> <li>- Baterías triple A</li> <li>- Arduino IDE</li> <li>- Firebase</li> </ul>	COP \$61800

Figura 5.14: Resumen de los parámetros de diseño

## Capítulo 6

# Implementación de la aplicación como ayuda para predecir el riesgo de padecer una isquemia cardiaca

En este capítulo se realiza el diseño, la implementación y la integración de los elementos dados a conocer en el capítulo anterior para crear el sistema principal junto con los demás subsistemas o interfaces.

### 6.1. Diseño del subsistema de medición de la frecuencia cardiaca

De acuerdo con el tercer requerimiento, se busca diseñar un dispositivo que permita la medición de la frecuencia cardiaca. Para ello, se plantea el siguiente diseño, donde se muestran las conexiones de los dispositivos escogidos y la integración dentro un contenedor:

#### 6.1.1. Diseño del esquemático

Con la ayuda del software Fritzing se crea una ilustración de la interacción entre el sensor MAX30102, el microcontrolador ESP32 y fuente de alimentación. En la figura [6.1](#) se presenta el esquemático resultante que incluye los 3 elementos mencionados anteriormente.

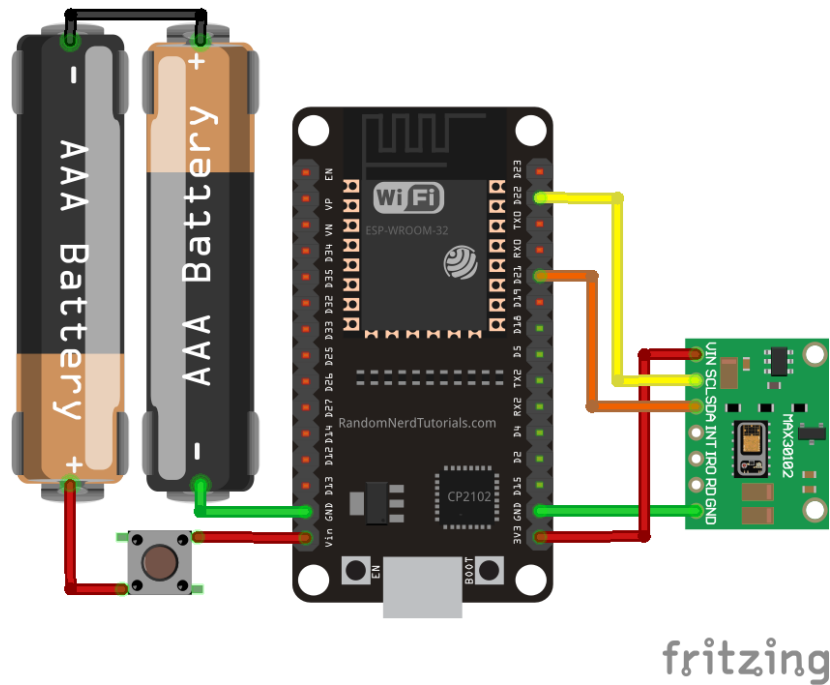


Figura 6.1: Esquemático de las conexiones entre los módulos

La explicación del esquemático de la figura [6.1](#) es sencilla: Primero, se alimenta el módulo ESP32 con un voltaje de 3V proveniente de las baterías AAA. Sin embargo, entre estos dos elementos existe un interruptor para controlar el encendido y apagado de este subsistema medidor de FC. Luego, el ESP32 es capaz de proporcionar energía al sensor MAX30102 a través de los pines 3V3 y GND. Después, para recuperar los datos del sensor, se utilizan dos conexiones cableadas que se conectan al pin SCL y SDA del ESP32 (ver figura [5.9](#)).

Ahora bien, en el **Anexo A** [9.1](#) se presenta el código implementado en Arduino IDE para la programación del microcontrolador. Además, la parte del código de medición de la frecuencia cardiaca fue extraída del repositorio de desarrolladores del MAX30102 [\[73\]](#). De esa manera, el código del **Anexo A** empieza con el uso de las librerías *wire.h*, *MAX30105.h*, *hearRate.h* y *BluetoothSerial.h*. Estas librerías básicamente permiten la lectura de la información del sensor MAX30102 y el envío de información a través de Bluetooth. Luego, se inicializan variables importantes como la cantidad de latidos por minuto (*beatsPerMinute*), la media de latidos por minutos (*beatAvg*), la información de salida al sistema principal (*dataout*), entre otras. Seguidamente, en el *void setup()* se configura el puerto serial de Arduino IDE para poder visualizar la información del sensor y se inicializa el sensor MAX30102 con una configuración por defecto. Después, en el *void loop()* se revisa si el sistema principal ha dado la orden de empezar a medir la FC. De esa manera, si la variable *DataBluetooth* tiene un valor de *N*, entonces el sensor permanecerá apagado, mientras que, si *DataBluetooth* tiene un valor de *S*, entonces, el usuario ha comenzado a hacer la medición de FC. Así, cuando se esté registrando la FC, el ESP32 revisa si se ha detectado un pulso en el sensor. Una vez suceda lo anterior, se agrega el valor de la medida de FC a un arreglo (*rates[]*). En dicho arreglo se van guardando las mediciones de FC (*beatsPerMinute*) a medida que se detectan más pulsos en el sensor. Finalmente, se promedian los valores del arreglo y se envía esa información por Bluetooth al sistema principal.

### 6.1.2. Diseño del contenedor de los elementos hardware

Luego de conectar los diferentes elementos para la medición de FC, se procede con el diseño de un contenedor que integre las baterías, el sensor MAX30102 y el ESP32. Para lo anterior, se utiliza el software SolidWorks. Ahora bien, la forma del contenedor se diseña teniendo en cuenta la apariencia de los pulsioxímetros del mercado, los cuales vienen en forma de gancho o dedal. En la figura [6.2](#) se pueden apreciar las perspectivas del diseño del contenedor. Cabe mencionar que la parte superior posee una tapa con tornillos para cubrir el ESP32. Mientras que la parte inferior cuenta con una tapa removible para colocar las baterías.

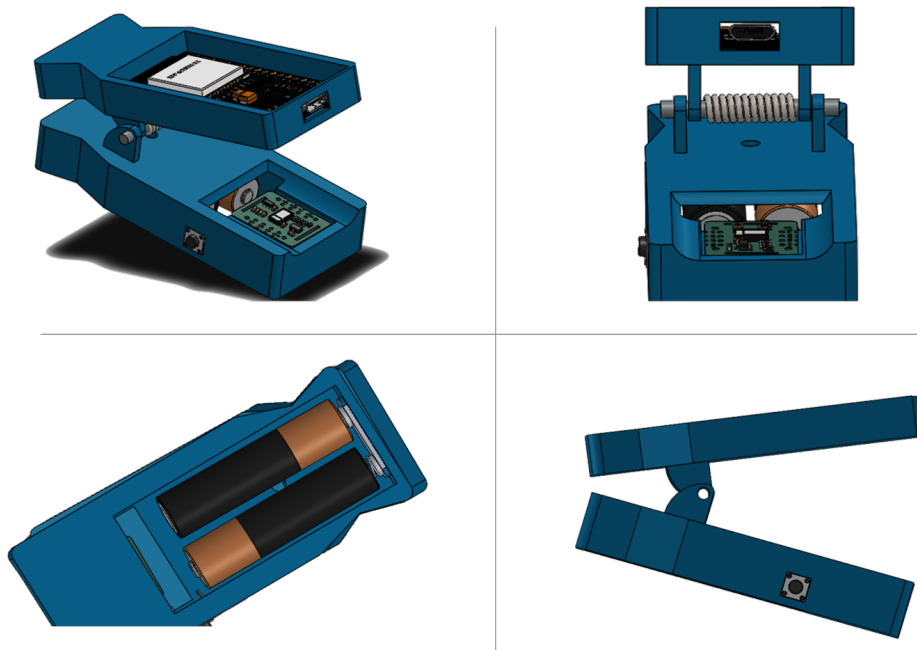


Figura 6.2: Contenedor desde diferentes perspectivas

Por el lado de las dimensiones del contenedor, se consideró que este fuera lo más pequeño posible, pero, que tuviera la capacidad de contener adecuadamente el ESP32, el MAX30102, las dos baterías triple A, el cableado y el interruptor de encendido. De esa manera, de acuerdo con las dimensiones de los elementos hardware, se plantea el diseño espacial de la figura [6.3](#). Ahora, respecto la parte superior del contenedor, esta posee un espesor de 10 mm para alojar el ESP32 y el cableado adicional. De igual forma, la parte inferior del contenedor tiene un espesor de 14 mm, lo cual facilita contener las baterías y el sensor.

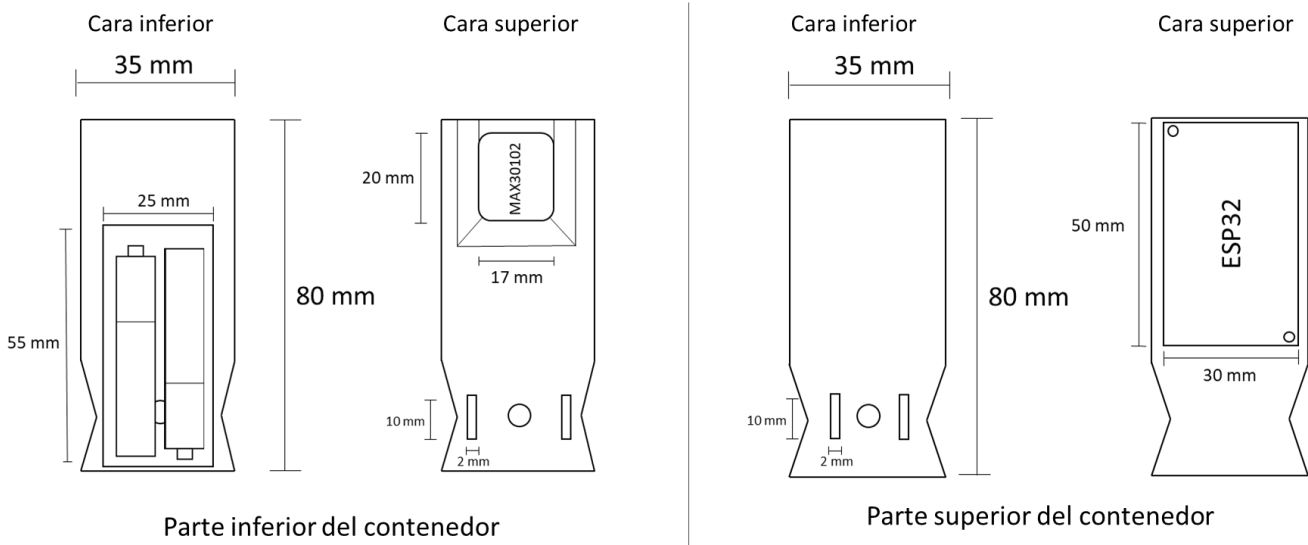


Figura 6.3: Diseño del contenedor

## 6.2. Diseño del sistema principal

Para realizar el diseño de esta parte, se utilizarán las herramientas Android Studio y Firebase. De esa manera, según los parámetros de diseño planteados en el capítulo anterior, se plantea el “diagrama de flujo” de la aplicación que se presenta en la figura 6.4. En dicha figura se aprecian las diferentes pantallas o interfaces del sistema principal. De esa manera, cuando el usuario inicie la aplicación, aparecerá una pantalla de bienvenida, que después de unos segundos, pasa a iniciar sesión, donde puede ir a registrarse como un usuario nuevo. Después, se presenta la interfaz de inicio, donde el usuario tiene la capacidad de ir a las diferentes interfaces que se presenta en la figura.

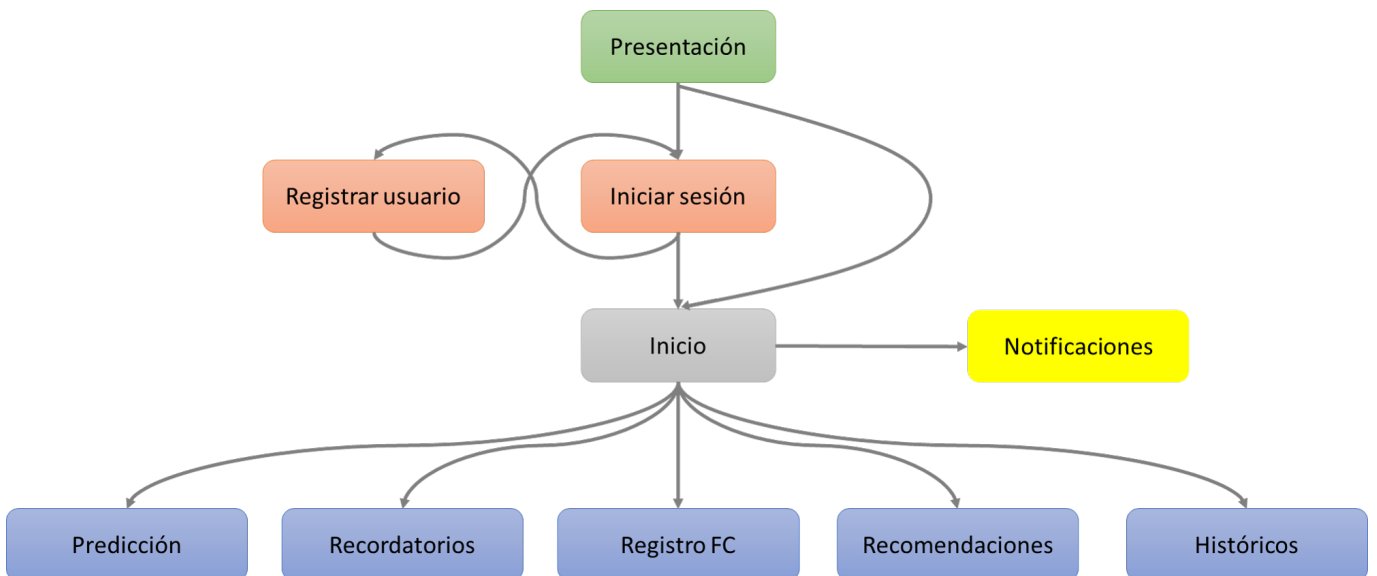


Figura 6.4: Diagrama de flujo de la aplicación

### 6.2.1. Programación de las interfaces de la aplicación

Ahora, se procede con la programación (Java) y el diseño (xml) de las pantallas de la aplicación (sistema principal). Para ello, se toma en cuenta el flujo que aparece en la figura 6.4. Cabe destacar que, por cuestiones de estética y extensión del documento, no se puede presentar todo el código que se utiliza en Android Studio para programar la aplicación; por lo anterior, se creó un repositorio en GitHub donde está contenido todo el proyecto de Android Studio (ver Anexo C 9.3). Por otro lado, se definió el nombre de **IscodeApp** para la aplicación. Dicho lo anterior, se comienza por la pantalla de bienvenida.

#### Pantalla de Bienvenida, Iniciar Sesión y Registrar

En la pantalla de Bienvenida simplemente se coloca el logo de la aplicación, con una animación de carga. Para ello, se emplea una imagen y una animación *Lottie* (que es un archivo de extensión .json) que se implementa utilizando el complemento *Lottie for Android* en el archivo Gradle (ver Anexo B 9.2). Además, esta pantalla tiene un tiempo de duración de 5 segundos, que al terminar, pasa a la siguiente pantalla (iniciar sesión, o inicio). En el primer ítem de la figura 6.5 se observa el resultado visual de esta pantalla.

Ahora bien, según la explicación del capítulo anterior, es importante que el usuario, si por algún motivo desinstala la aplicación o cambia de dispositivo, tenga la forma de recuperar los registros de FC y predicciones. Por ello, la aplicación está pensada para que tenga una parte de “iniciar sesión” donde se recuperen los datos de la persona desde Firebase. Así, la información se descarga y se guarda localmente en un archivo de base de datos gracias a la herramienta **SQLite** (ver Anexo D 9.4). En el segundo ítem de la figura 6.5 se observa la interfaz de la pantalla en cuestión.

Por otro lado, en el tercer ítem de la figura 6.5 se observa el diseño de la pantalla Registrar. En dicha pantalla, el usuario tiene la posibilidad de registrarse como un nuevo usuario de la aplicación. Para ello, se le pide un nombre, un correo, una contraseña y la fecha de nacimiento. Cuando se presiona el botón de “Registrar”, se revisa si todos los datos están completos. Después de esto, entra en escena Firebase, que permite la creación de este usuario con el correo y la contraseña proporcionados. Por otro lado, *Real Time Database* de Firebase, permite la creación de un banco de datos relacionados con el nuevo usuario. Una vez se haya hecho todo lo anterior, la aplicación pasa a la siguiente pantalla, o sea, Inicio.

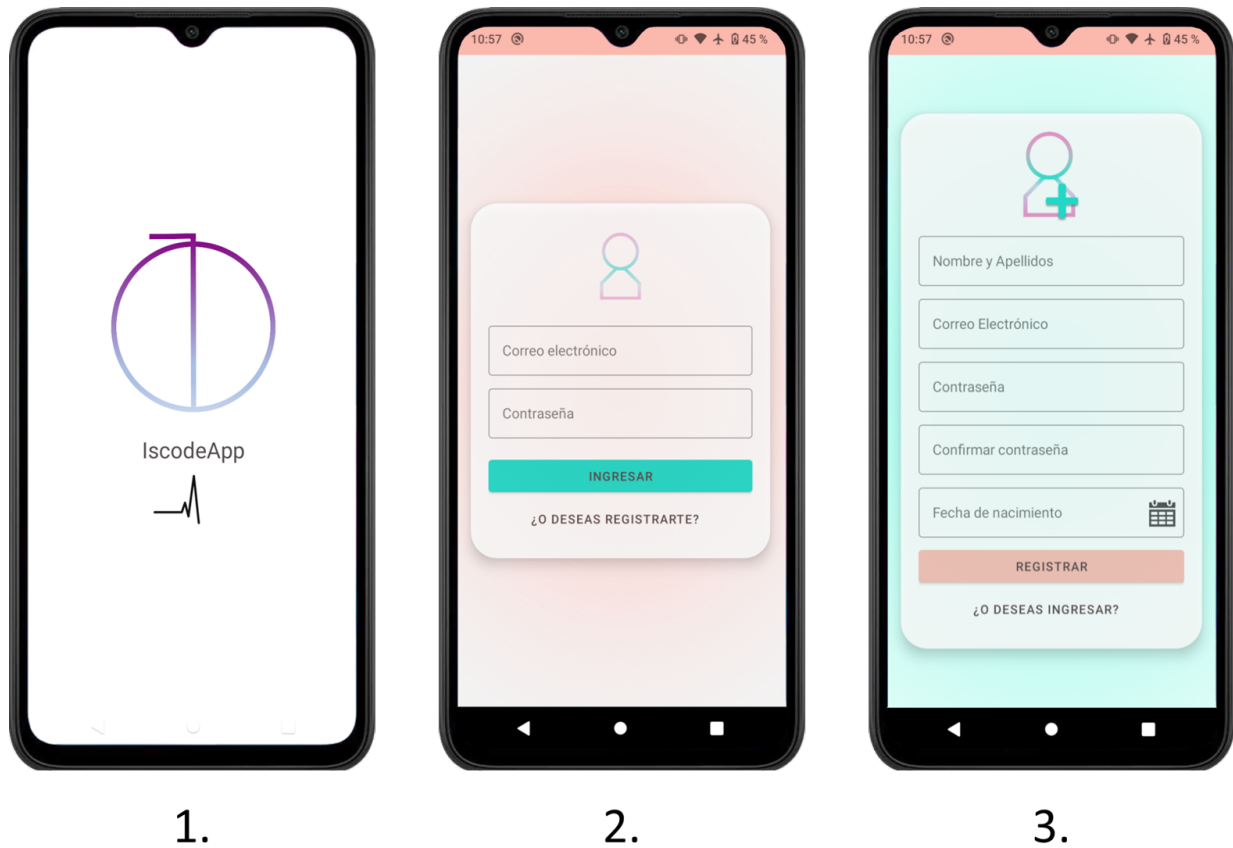


Figura 6.5: Pantalla de Bienvenida

### Pantalla de Inicio

En la figura [6.6](#) se presenta la interfaz de la pantalla inicio (parte izquierda). En dicha pantalla, se presentan los “vínculos” a las principales interfaces del sistema principal (Predicción, Recordatorios, Notificaciones, etc.). Por otro lado, si la persona lo desea, se puede cerrar sesión en el dispositivo actual (lado derecho de la figura [6.6](#)). Esto hace que toda la información del usuario se elimine y se cambie a la pantalla de Iniciar Sesión.

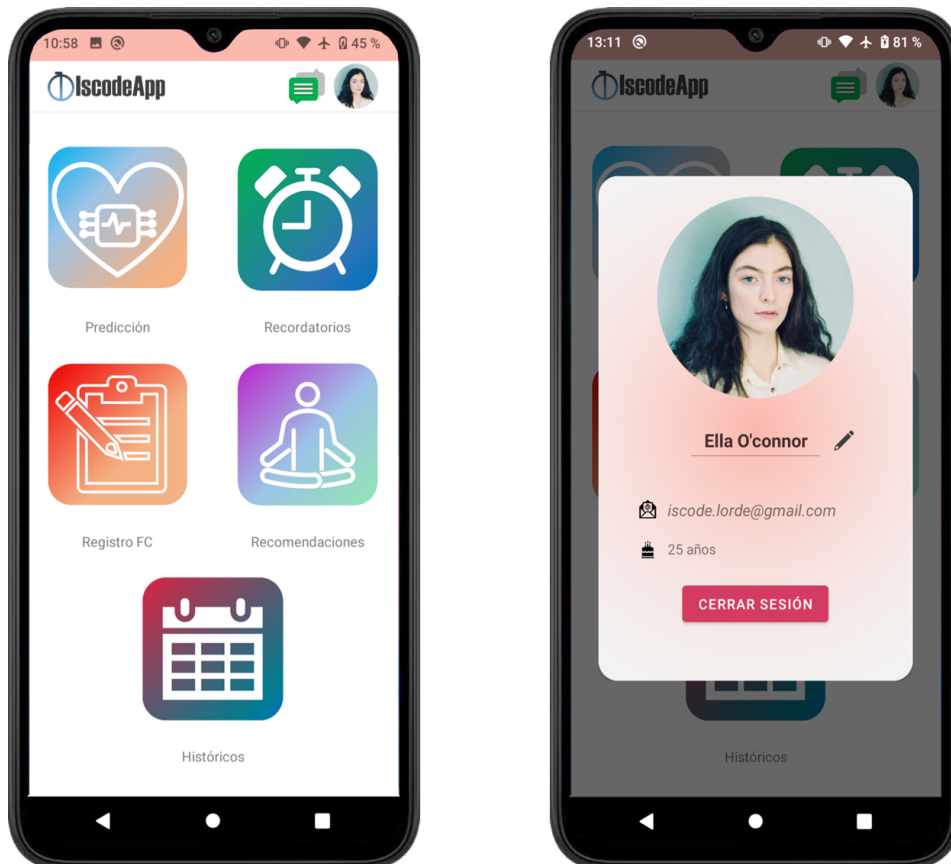


Figura 6.6: Pantalla de Inicio

### Pantalla de Predicción

Después de importar el modelo de Machine Learning a Android Studio, se procede a realizar el diseño de esta pantalla. Para eso, se crean varios elementos de diseño que hacen referencia a cada atributo de la predicción (edad, género, frecuencia cardíaca, etc.). Así, el usuario puede escribir el valor de la medición (si es numérico) o seleccionar el valor del atributo al desplegar una lista (por ejemplo, en género hay dos opciones: Masculino o Femenino). Después de completar todos los datos, el usuario puede presionar el botón de “Predecir”, que básicamente, hace que se procesen los datos proporcionados por el usuario para ser usados como entrada del modelo. En el primer ítem de la figura [6.7](#) se observa este diseño.

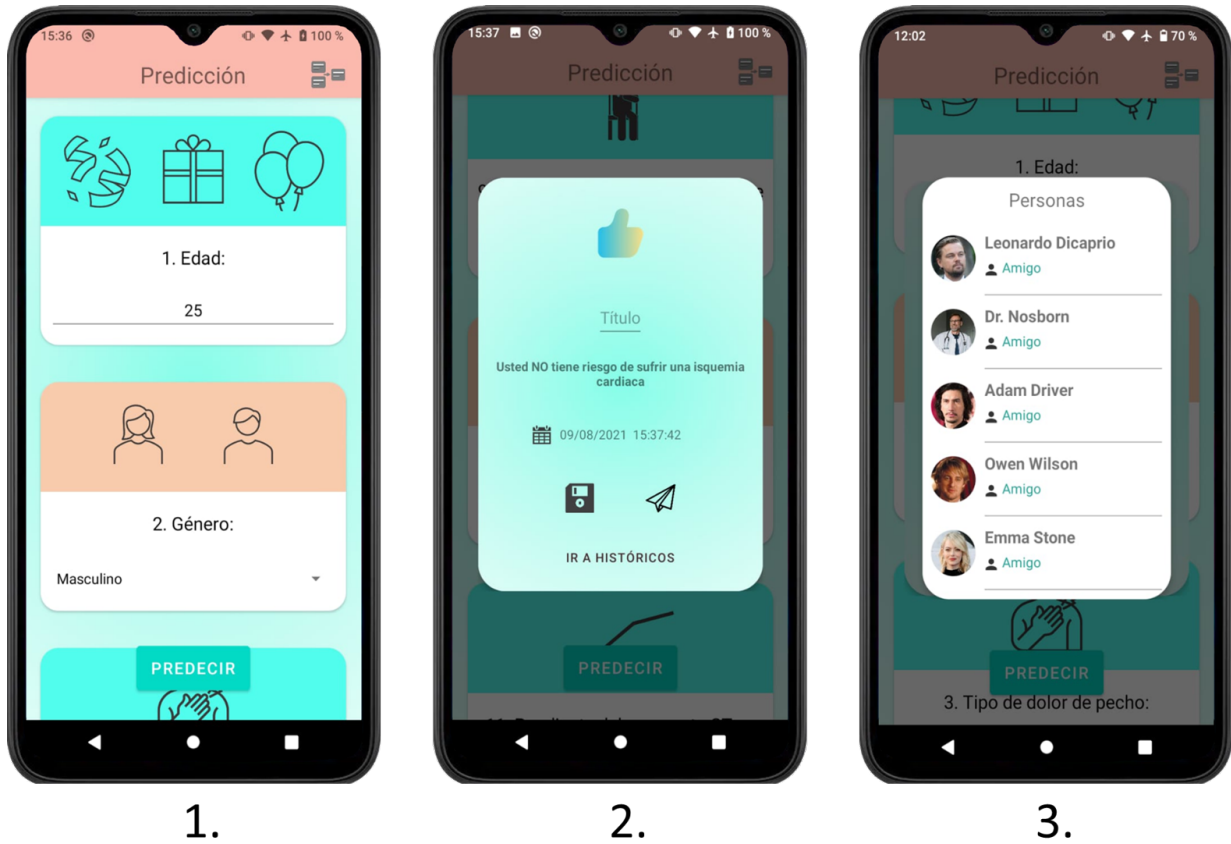


Figura 6.7: Pantalla de Predicción del riesgo de padecer una EIC

Cuando termine de procesar la solicitud de predicción, la aplicación presenta un dialogo con el resultado. Dicho resultado se puede guardar o compartir con un contacto. Para ello, es necesario proporcionar un título para la predicción. En el segundo ítem de la figura 6.7 se observa ese dialogo. Ahora bien, si la persona presiona el botón de “Guardar predicción”, el registro se guarda localmente en la base de datos **SQLite**. Además, cuando la base de datos se modifica, esta se sube a *Storage* de Firebase para mantener actualizada la información. En el **Anexo E 9.5** se lista la rutina en código Java para subir cualquier archivo a Firebase. Por otro lado, cuando se presione el botón de “Compartir predicción”, se despliega una lista de contactos como la que aparece en el tercer ítem de la figura 6.7. Luego, el usuario selecciona la persona a la que desea enviar la predicción y la aplicación cambia a la pantalla de Notificaciones para llevar a cabo esta operación (ver la sección de la pantalla Notificaciones).

### Pantalla de Recordatorios

La figura 6.8 (lado izquierdo) se logra ver la vista con la pantalla de Recordatorios. En dicha interfaz se presenta una lista de cada recordatorio. Cada ítem muestra el título, la fecha de programación del recordatorio y una bandera para denotar la importancia del recordatorio. Cuando el usuario presione un ítem de la lista, se abre otra interfaz donde se puede observar con más detalle el recordatorio, además de editarlo. Ahora bien, esta última pantalla se presenta en la figura 6.8 (lado derecho), donde el usuario tiene la capacidad de colocar un título, un contenido, una importancia al recordatorio, un color, la fecha y enviar el recordatorio a otras personas. Cabe mencionar, que

la información de cada recordatorio se guarda localmente (**SQLite**) y se sincroniza con Firebase. Además, el recordatorio se puede repetir. Por ejemplo, si se necesita consumir algún medicamento cada 12 horas, el recordatorio aparecerá cada 12 horas como una notificación.

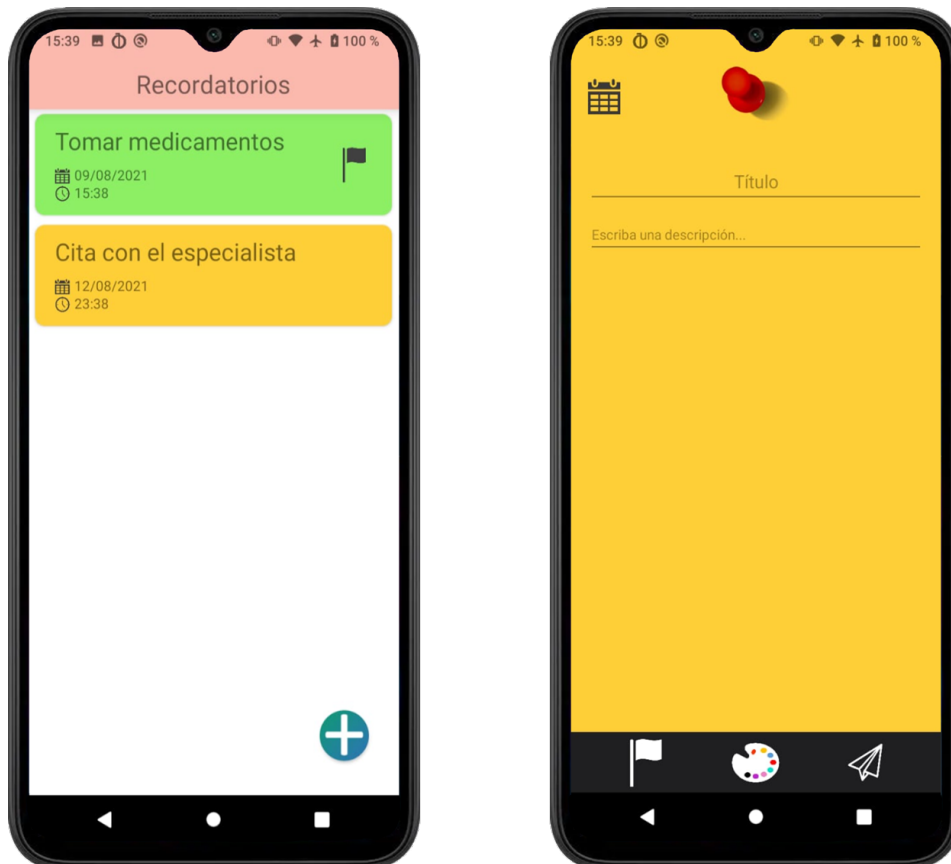


Figura 6.8: Pantalla de Recordatorios

### Pantalla de Registro FC

Para la medición de la frecuencia cardiaca se hace uso del dispositivo diseñado en la sección [6.1](#). En dicha sección se mencionó que se utilizará la conexión Bluetooth para la transmisión de los datos de frecuencia cardiaca. De esa manera, la pantalla de Registro FC comienza con el listado de los dispositivos Bluetooth vinculados al Smartphone (primer ítem de la figura [6.9](#)). Así, el usuario debe antes vincular el Sensor de frecuencia cardiaca con el Smartphone para que aparezca en la lista. Para lo anterior, se utiliza, en la programación de Java, *BluetoothAdapter*, que básicamente es la forma de manejar los eventos Bluetooth dentro de la aplicación Android. En el **Anexo F** [9.6](#) se observa la rutina para verificar el estado Bluetooth y listar los dispositivos vinculados. Después, el usuario selecciona el ítem de lista Bluetooth y se procede con la verificación de la conexión (para saber si el ESP32 está encendido). Si la conexión falla, no se puede realizar la medición de FC, de lo contrario, en la pantalla de Registro FC aparece el botón de “nueva medición” (ver segundo ítem de la figura [6.9](#) lado derecho).

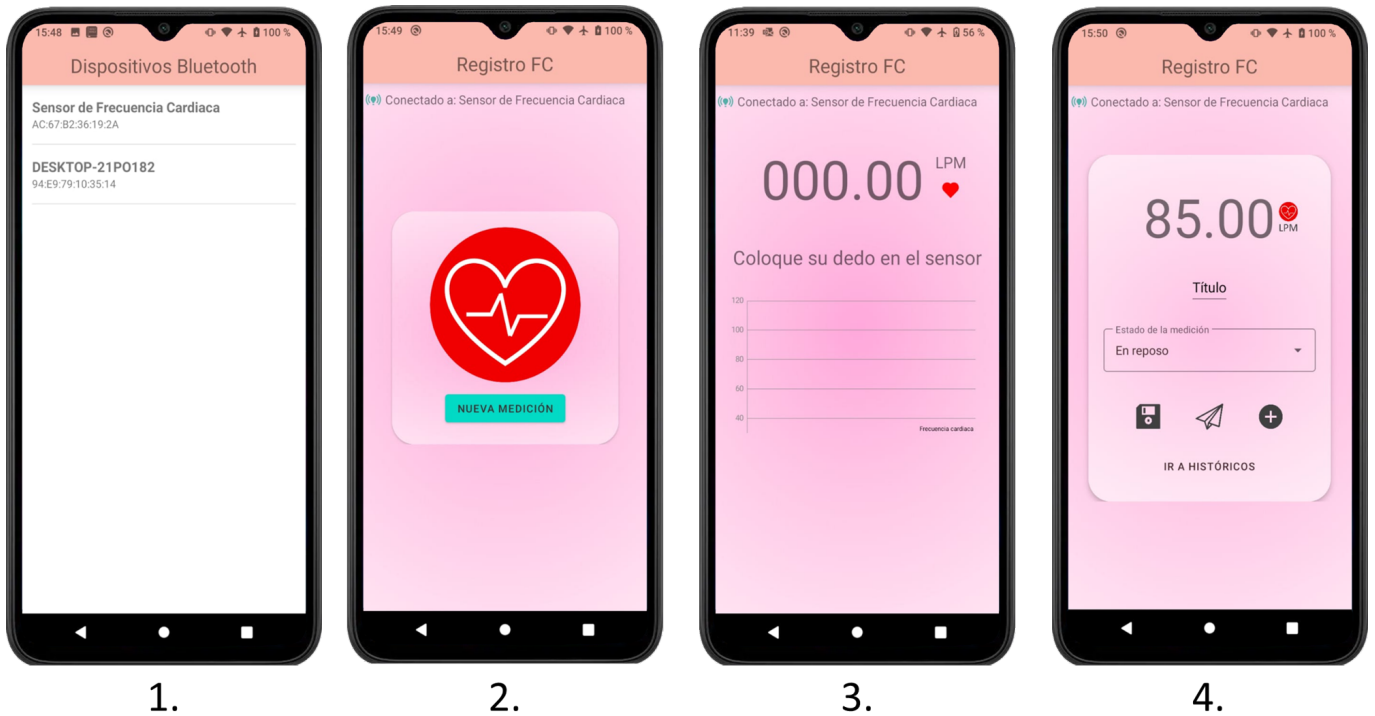


Figura 6.9: Pantalla de Registro de Frecuencia Cardíaca

Luego, aparece en pantalla un mensaje en forma de texto que verifica si se ha colocado el dedo en el sensor. Una vez lo coloque, el mensaje pasa a estado de “leyendo” (ver tercer ítem de la figura 6.9), mientras se recopilan los datos del sensor cada vez que se detecte un pulso. Luego de 10 mediciones, se presenta como resultado final el promedio de las pulsaciones por minuto que el sensor detecto (ver Anexo A 9.1). Después de terminada la medición, se presenta un dialogo en el que el usuario tiene la posibilidad de guardar, enviar o realizar una nueva medición de FC. En dicho dialogo el usuario puede colocar un título y establecer el estado físico en el que hizo la medición (cuarto ítem de la figura 6.9). Además, cuando el usuario presiona el botón de “guardar registro FC”, la información del registro se conserva en la base de datos SQL que se usó para los recordatorios y las predicciones, pero en otra tabla. Luego, si el usuario lo desea, puede ir a la pantalla de Históricos para conocer los registros de FC. Por otro lado, si se presiona el botón de “enviar registro FC”, se presenta la lista de contactos, para que el usuario decida a quien enviar la medición. Finalmente, cuando se da click en el botón de “nueva medición FC” (icono con el símbolo +), se ignora la medición actual y se realiza otra.

### Pantalla de Recomendaciones

La pantalla de recomendaciones es como aparece en la figura 6.10. En esta interfaz el usuario podrá visualizar las recomendaciones que le envía el especialista. Así, cada ítem (recomendación) contiene un título, un contenido, un tipo y un enlace de página web donde el usuario podrá encontrar más información sobre el tema de la recomendación. Ahora bien, la información que se presenta en las recomendaciones se extrae de *Real Time Database* de Firebase. Entonces, cuando el especialista envía una recomendación, la información se guarda en Firebase y la pantalla de Recomendaciones se encarga de descargar y presentar tal información.



Figura 6.10: Pantalla de Recomendaciones

### Pantalla de Históricos

En la pantalla de históricos se presenta el resumen de las predicciones y los registros de frecuencia cardiaca. Así, en el primer ítem de la figura [6.11](#) se presentan los registros de FC y una gráfica donde se resumen los datos en latidos por minuto (lpm). Igualmente, en el segundo ítem de la figura [6.11](#) se presentan las predicciones. Cabe mencionar que la información presentada en esta pantalla se extrae de la base de datos local (**SQLite**) que se explicó en las anteriores pantallas.

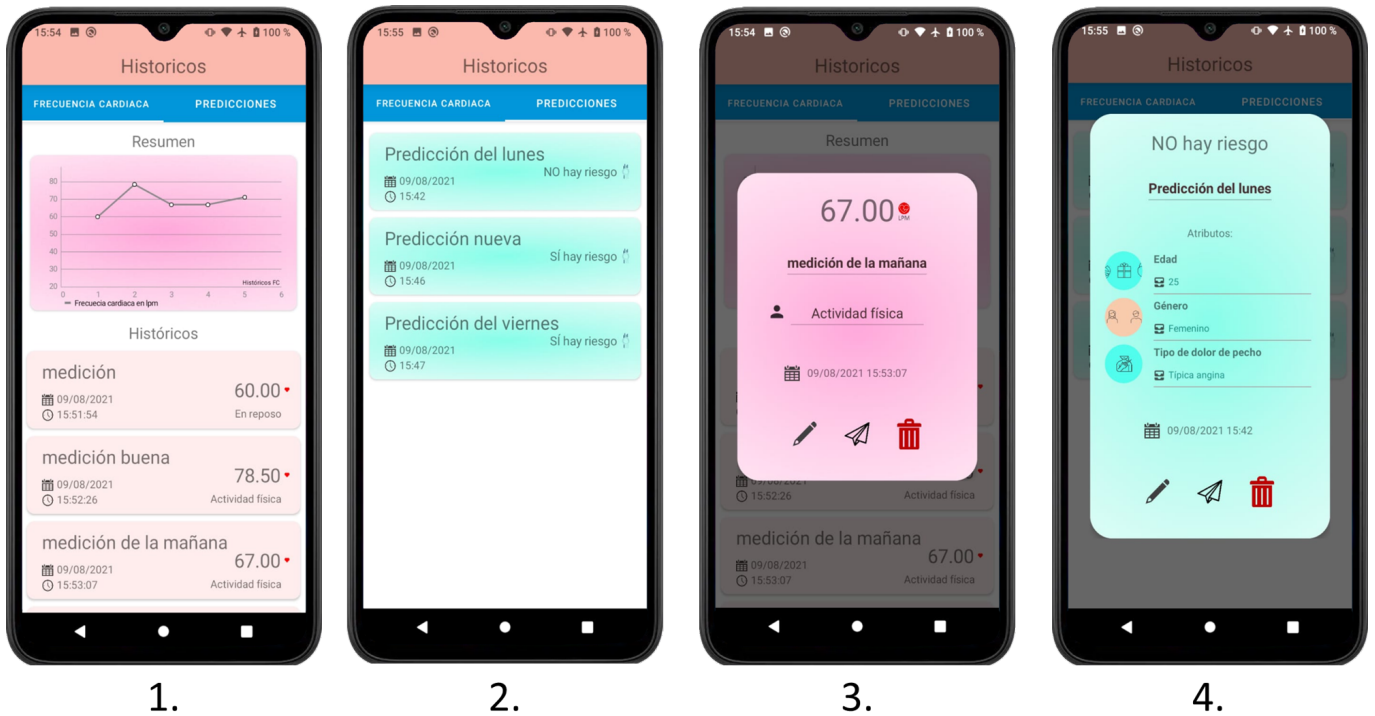


Figura 6.11: Pantalla de Históricos

Si el usuario desea editar, enviar o eliminar cada histórico, entonces se diseñan dos diálogos (ventanas emergentes) para dicho fin. En el caso de los registros de FC y las predicciones, el usuario puede editar el título de estos. También, si desea enviar algún registro a sus contactos, existe el botón de “enviar”, el cual presenta la lista de contactos para que el usuario elija a quien compartir el histórico. Este diseño se muestra en el tercer ítem de la figura 6.11. Además, por el lado de las predicciones, cada histórico presenta los atributos (edad, género, frecuencia cardiaca, presión arterial, etc.) que condujeron al resultado del histórico (ver el cuarto ítem de la figura 6.11). Igualmente, cada histórico, ya sea de FC o riesgo de padecer una EIC, tiene una fecha que hace referencia al momento en el tiempo en que se guardó dicho histórico.

### Pantalla de Notificaciones

La pantalla de notificaciones permite la interacción del usuario con el especialista o con otros usuarios. Para ello, la aplicación descarga la información de los usuarios contenida en *Real Time Database* (nombre, la fecha de nacimiento y el correo) y la guarda localmente en otra base de datos. En dicha base de datos se crea una tabla donde se guarda la información descargada. De esa manera, mientras se descargan los datos, se muestra un diálogo como el que aparece en el primer ítem de la figura 6.12. Después, la aplicación presenta la interfaz del segundo ítem de la figura 6.12, donde se da a conocer el listado de todos los usuarios y se muestra el nombre junto con la foto de perfil.

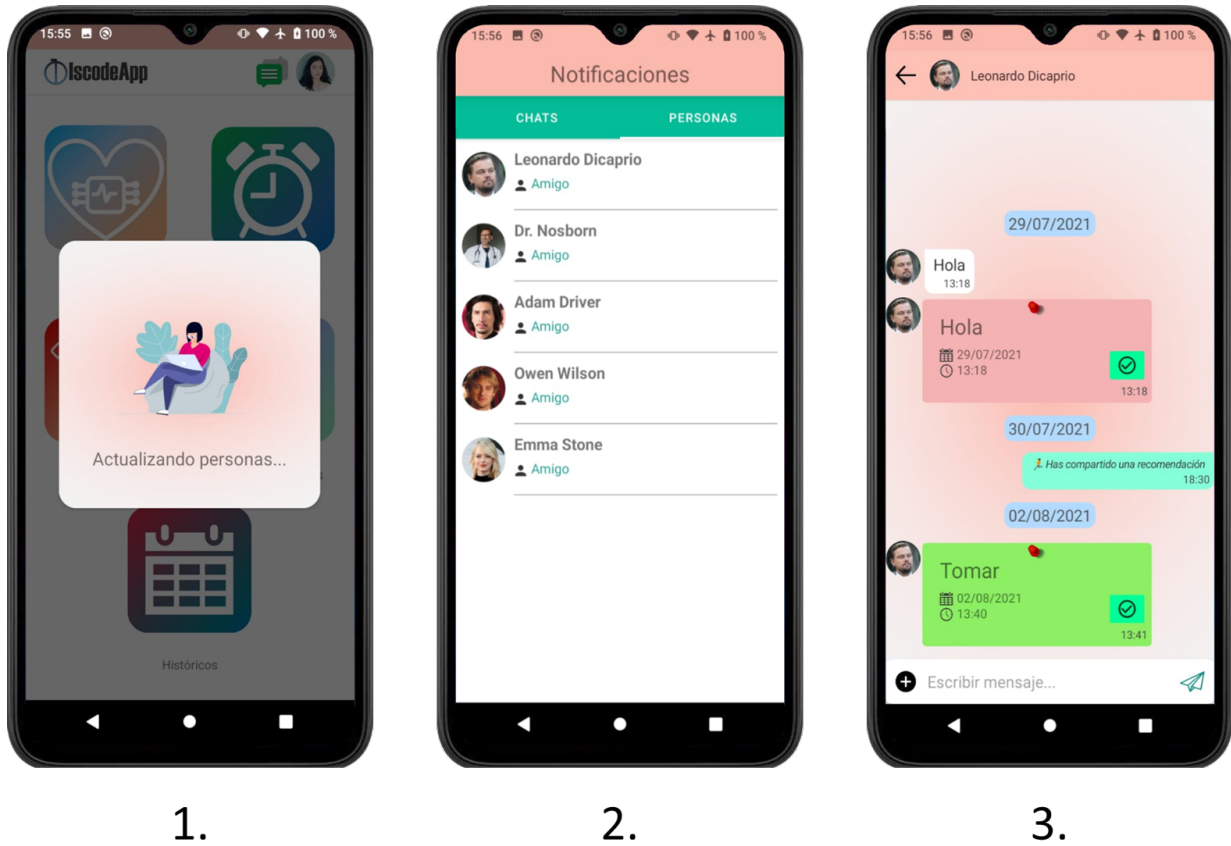


Figura 6.12: Pantalla de Notificaciones

Cuando el usuario presiona sobre algún contacto, se abre una interfaz para “mensajes”. En dicha interfaz, cada persona tiene la posibilidad de enviar recordatorios, recomendaciones, registros FC, predicciones o mensajes de texto. Ahora bien, cuando se envía un mensaje, este se aloja en *Real Time Database* de Firebase. De esa manera, cuando se requiera leer los mensajes con cierto contacto, la rutina en Java descarga los mensajes y los presenta en la interfaz de “mensajes” (ver el tercer ítem de la figura [6.12](#)). Además, cuando se envía el mensaje, la aplicación tiene la capacidad de mostrar una notificación, en la barra de notificaciones del Smartphone, con el contenido del mensaje y la persona quien lo envía.

## Capítulo 7

# Evaluación del Hardware y Software

En este capítulo se evalúa el funcionamiento del software y hardware de la aplicación desarrollada con el fin de cumplir con los objetivos propuestos en este documento.

### 7.1. Evaluación del hardware para la medición de FC



Figura 7.1: Hardware para la medición de frecuencia cardiaca

En la figura [7.1](#) se observa la materialización del dispositivo de medición de frecuencia cardiaca. En dicho dispositivo se integra el sensor MAX30102, el ESP32, las baterías triple A, el cableado y el interruptor de encendido. Ahora bien, para revisar el funcionamiento de este dispositivo, se utiliza el puerto serial (*Monitor Serie*) de Arduino IDE, el cual, permite observar las mediciones de frecuencia cardiaca, y los dos distintos estados del dispositivo mientras está encendido: “colocar el dedo” y “leyendo”. Después, con la rutina del *Anexo A* [9.1](#), se puede enviar a través de Bluetooth la información que aparece en la figura [7.2](#).

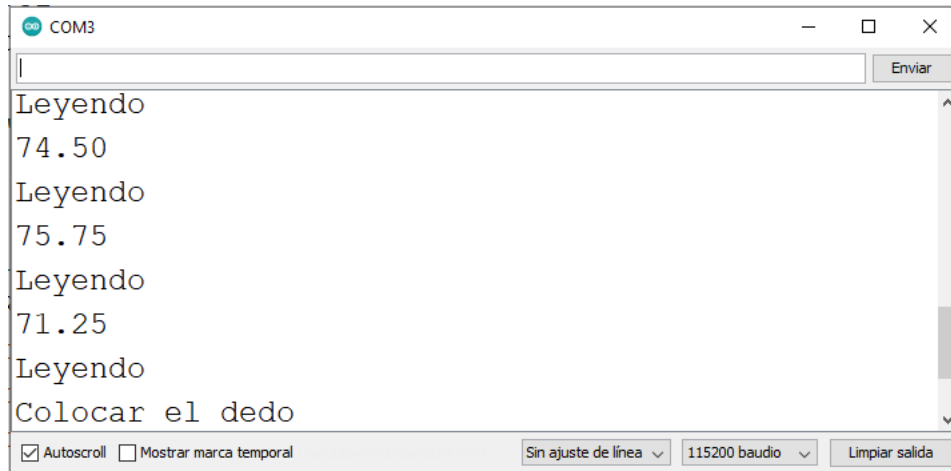


Figura 7.2: Datos obtenidos del dispositivo de medición de frecuencia cardiaca

Por otro lado, en Android Studio, se implementa un código de prueba para recibir y presentar la información que llega desde el dispositivo de medición de FC. Dicha rutina se encuentra en el **Anexo G 9.7**. En la figura **7.3** se presenta la interfaz en Android Studio donde se muestran los datos que llegan desde el dispositivo de medición de FC. En dicha figura se logra percibir que efectivamente está llegando la información de frecuencia cardiaca a la aplicación.



Figura 7.3: Datos obtenidos del dispositivo de medición de frecuencia cardiaca

### 7.1.1. Análisis de demanda energética del dispositivo de medición de FC

Todo el hardware es alimentado por un par de baterías triple A. Para hacer el cálculo de demanda energética, se tomará como referencia un par de baterías AAA alcalinas de la marca Duracell. Dichas baterías, en serie, entregan un voltaje de 3V. Además, cada una de ellas tiene una capacidad de 1200 mAh. Por otro lado, el ESP32 consume en modo “Activo” 80 mA **[76]**. Además, el sensor MAX30102 demanda una corriente de 0.6 mA **[77]** para su funcionamiento. Así, con estos datos, se logra hacer el siguiente cálculo del tiempo de autonomía que puede tener el dispositivo de medición de FC:

$$Tiempo = \frac{1200mAh * 2}{80mA + 0,6mA} = 29,78h$$

De esta manera, en teoría se puede mantener todo el hardware funcionando por más de 2 días continuamente con dos baterías triple A. Ahora bien, en un caso hipotético donde el usuario use el dispositivo 3 veces al día durante 5 minutos todos los días de la semana, se puede tener una autonomía cercana a los 119 días o 4 meses.

## 7.2. Evaluación del sistema completo

Para la evaluación del sistema, se consultó a un profesor de la Universidad Javeriana de Cali, que ha sufrido una enfermedad relacionada con el corazón. Se le informó a esta persona a través de una presentación como es el funcionamiento de **IscodeApp** (nombre de la aplicación) y cuáles son las distintas interfaces dentro de ella. Por otro lado, la aplicación en formato *apk* tiene un tamaño de 31 MB. Así, se hizo el recorrido de cada pantalla de la aplicación con el potencial usuario, teniendo en cuenta el diagrama de flujo que se presentó en la figura [6.4](#).

### 7.2.1. Iniciar sesión y registrar un nuevo usuario

Para esta parte, se le explica al usuario que **IscodeApp** permite guardar información en internet (Firebase) para tenerla disponible posteriormente por si algún motivo desinstala la aplicación o cambia de dispositivo. Lo anterior hace que los recordatorios, los históricos y los mensajes del usuario se conserven. De esa manera, se le explica los pasos para registrarse como un nuevo usuario, al ingresar su nombre, su correo electrónico, una contraseña y su fecha de nacimiento. Además, se le indica que puede cerrar sesión cuando lo desee y puede ingresar simplemente con el correo y la contraseña a **IscodeApp**. También, se le informa al usuario que puede colocar una foto de perfil para tener más personalización dentro de la aplicación, además de, poder editar el nombre con el que los demás usuarios lo denotan.

### 7.2.2. Predicción del riesgo de padecer una EIC

Aquí, se le indica al usuario que debe ingresar los atributos para conocer el riesgo de padecer una EIC (edad, género, presión arterial, etc.). Una vez complete todos los atributos, el usuario podrá hacer la predicción, de lo contrario, **IscodeApp** notifica que hacen falta ciertos atributos. Después, de hacer la predicción, se le explica al usuario que tiene la capacidad de compartir el resultado, descartarlo o guardarlo (en un archivo de base de datos que constantemente se actualiza en Firebase) para posteriormente visualizarlo en “Históricos”.

### 7.2.3. Recordatorios y recomendaciones

Para los recordatorios, se le menciona al usuario que **IscodeApp** tiene la capacidad de mostrar una notificación en el Smartphone con sonido y vibración en cierta hora el día para recordarle alguna actividad del tratamiento, el consumo de medicamentos o una cita con el especialista (en la figura

7.4 se presenta el diseño de la notificación). Dicho recordatorio se puede personalizar con un título, un contenido, un color, una importancia y un periodo de repetición. Además, se le explica al usuario que puede aceptar recordatorios que el especialista envía a través de la sección de “Notificaciones”. También, se le explica al usuario que la información de cada recordatorio se guarda en archivo de base de datos que se sube a internet (Firebase) para recuperar los recordatorios en un nuevo inicio de sesión. Por otro lado, el especialista tiene la capacidad de enviar recomendaciones desde la aplicación a cada usuario. En dicha recomendación se puede colocar un título, un contenido, un vínculo web y un tema. Luego, el usuario puede visualizar cada ítem en la sección “Recomendaciones” y se le muestra a través de una notificación cuando ha llegado una nueva recomendación.

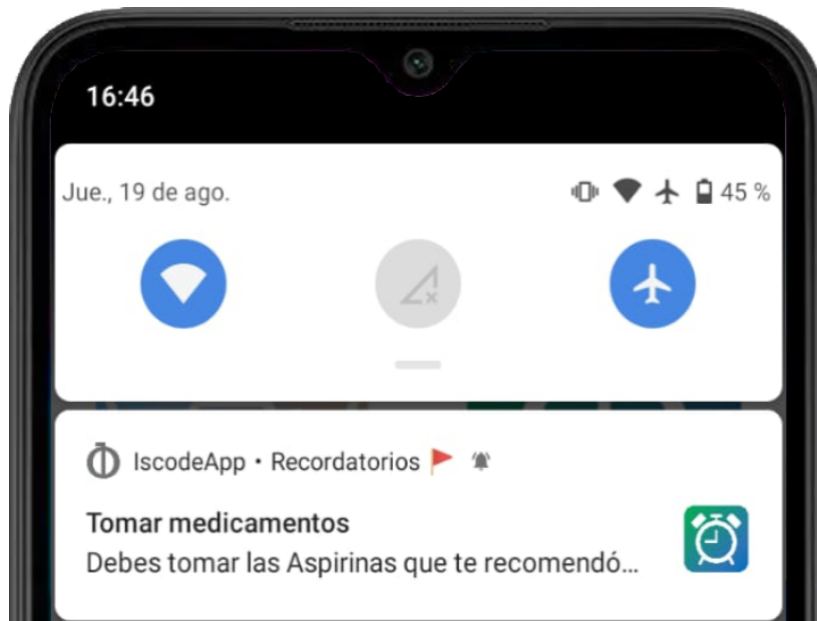


Figura 7.4: Recordatorio en la barra de notificaciones Android

#### 7.2.4. Registro de frecuencia cardiaca

Ahora, en esta parte de la aplicación, se le indica al usuario que debe encender el Bluetooth del Smartphone. Luego, debe emparejarse con el dispositivo de medición de frecuencia cardiaca. Para ello, debe acceder a la sección de “Ajustes” del Smartphone y seleccionar la opción de “Agregar un nuevo dispositivo Bluetooth”. Después, dentro de **IscodeApp** en la pantalla “Registro FC”, con el Bluetooth encendido, se listan todos los dispositivos emparejados con el Smartphone. Así, se le explica al usuario que debe seleccionar de ese listado el ítem que diga “Sensor de Frecuencia Cardiaca”. Seguidamente, **IscodeApp** revisa si el sensor está encendido para poder hacer la medición de FC. Luego, se le muestra al usuario que debe colocar el dedo sobre el sensor MAX30102 que se encuentra dentro del dedal. De esa manera, después de 10 mediciones, se presenta el resultado final de la medición de FC, donde se le indica al usuario que tiene la capacidad de guardar, descartar o compartir el registro con sus contactos.

### 7.2.5. Históricos y comunicación con otros usuarios

Finalmente, se procede con la explicación al usuario de la capacidad que tiene **IscodeApp** de presentar un resumen gráfico de la frecuencia cardiaca en la pantalla de “Históricos”. Además, puede editar el nombre, enviar o eliminar cada registro de FC. De igual forma, se pueden listar los resultados de las predicciones guardadas por el usuario. Así, se le indica a la persona que puede revisar el valor de los atributos que condujeron al resultado de la predicción. También, existe la opción de editar el nombre, compartir o eliminar el registro de predicción del riesgo de EIC. Por otro lado, se le enseña al usuario que **IscodeApp** tiene la capacidad de comunicarse con otros usuarios de la aplicación. Para lo anterior, se hace uso de la pantalla “Notificaciones”, en donde se puede enviar un texto, un recordatorio, un registro FC y una predicción al contacto que el usuario desee. De esa manera, al receptor del mensaje se le mostrará una notificación en el Smartphone con la información que envía el usuario.

## 7.3. Resultados de la evaluación del sistema completo

En siguientes tablas, se muestran los resultados de la evaluación del sistema respecto a los requerimientos funcionales que se plantearon en el capítulo 5:

Evaluación	Descripción
<b>Lo esperado</b>	Predecir el riesgo de una EIC a través de un método computacional.
<b>Lo obtenido</b>	Con la ayuda de Python y Tensorflow se logró crear un modelo entrenado con la base de datos de <i>UCI Repository (Cleveland)</i> , con la capacidad de predecir el riesgo de padecer una EIC. Además, dicho modelo se implementó dentro de una aplicación Android, que toma alrededor de 1 segundo en hacer una predicción cuando se ingresan los atributos y se le da el botón de “Predecir”. También, al evaluar el modelo con información que no había conocido antes (muestras) se obtuvo una precisión del 91 %.
<b>Por mejorar</b>	El profesor que dio su opinión sobre la aplicación mencionó que el diseño de la pantalla “Predicción” estaba muy bien. Sin embargo, él describió que cada atributo debería tener una opción de “Más información” donde se desplegara la explicación de cada atributo por si los usuarios no entienden de que se trata cada uno de ellos. Además, cuando se da el resultado de la predicción, hace falta colocar una explicación más profunda sobre lo que se puede hacer en caso de un <b>NO</b> o un <b>SÍ</b> . Igualmente, el profesor recomendó aumentar la escala de predicción, es decir, “No hay riesgo”, “Riesgo bajo”, “Riesgo medio” y “Riesgo alto”.

Cuadro 7.1: Resultados de la evaluación de **IscodeApp** respecto al requerimiento funcional N°1

Evaluación	Descripción
<b>Lo esperado</b>	Presentar una serie de recordatorios informativos sobre las actividades del tratamiento.
<b>Lo obtenido</b>	Los recordatorios son personalizables (información) dentro de la aplicación y se muestra en la hora exacta (con un desfase de unos 20 segundo) en que el usuario los programa. Dichos recordatorios pueden ser de cualquier tipo (medicamentos, citas, ejercicio, etc.), de acuerdo con las necesidades del paciente. Además, el usuario tiene la capacidad de compartir sus recordatorios con otros contactos dentro de la aplicación. Por otro lado, el especialista puede enviar recomendaciones a sus pacientes sobre alimentación, entretenimiento, lectura o actividad física.
<b>Por mejorar</b>	Se puede mejorar, en cuanto a interfaz, que el usuario tenga la capacidad de organizar o agrupar recordatorios. De igual forma, se espera que los recordatorios no solamente se presenten en la barra de notificaciones del Smartphone, sino que también, se presenten en una notificación de escritorio si el usuario usa una computadora constantemente.

Cuadro 7.2: Resultados de la evaluación de **IscodApp** respecto al requerimiento funcional N°2

Evaluación	Descripción
<b>Lo esperado</b>	Adquirir los valores de la frecuencia cardiaca para ser usados en la predicción de la EIC.
<b>Lo obtenido</b>	Se logró crear un dispositivo para la medición de la frecuencia cardiaca. Dicho dispositivo es portátil y funciona con un par de baterías triple A lo que permite su autonomía. Además, los valores de FC se envían a la aplicación a través de Bluetooth. Ya en la aplicación, las medidas de FC se pueden usar para hacer la predicción del riesgo de padecer una EIC, guardarlas en una base de datos o compartirlas con otros usuarios o el especialista.
<b>Por mejorar</b>	El profesor mencionó que el dispositivo de medición debería ser más pequeño, e incluso, fuera capaz de alertar a la persona si la frecuencia cardiaca supera cierto umbral cuando se hacen mediciones constantes. Además, se espera que desde el mismo Smartphone se puedan hacer las mediciones sin necesidad de un dispositivo exterior. Por otro lado, en cuanto al diseño de la interfaz, se podría ser más explícito a la hora de presentar las mediciones y los mensajes informativos dentro de la pantalla de "Registro FC".

Cuadro 7.3: Resultados de la evaluación de **IscodApp** respecto al requerimiento funcional N°3

## 7.4. Comparación entre el sistema actual y la aplicación realizada

En esta parte es importante mencionar que TRIZ es una herramienta que permite el diseño de productos a partir de un sistema actual que intente resolver el problema que se está tratando. De esa manera, se realiza una tabla (ver figura [7.5](#)) donde se listan diferentes características del sistema actual (**HeartWatch**) en comparación con la aplicación desarrollada (**IscodApp**). Lo anterior, permite conocer cuál ha sido la "evolución" del sistema actual hacia un sistema "ideal" en el futuro. Cabe reconocer que **HeartWatch** es una aplicación bastante funcional y útil con diversas características que la hacen interesante. Así, lo que se espera con TRIZ, es poder diseñar un nuevo sistema a partir de ciertas características que posea la aplicación actual. De esa manera, **IscodApp** es un acercamiento a la aplicación "ideal" para acompañar a las personas en el cuidado de su salud cardiovascular.

	<b>IscodeApp</b>	<b>HeartWatch</b>
Uso del sistema	El usuario puede instalar la aplicación a través del archivo apk. La aplicación es intuitiva. Se puede hacer la predicción del riesgo de padecer una isquemia cardiaca a partir de la información que recopila (registros FC) y los datos ingresados por el usuario.	La aplicación es intuitiva y posee tutoriales internos donde le explica al usuario como usar cada herramienta de la aplicación. No cuenta con una opción para predecir información futura a partir de la información que recolecta.
Visualización	El acceso a las diferentes pantallas de la aplicación se presenta en el menú principal. El avance por cada pantalla es fluido. Las predicciones, los registros de FC y los recordatorios, se presentan adecuadamente en la pantalla.	La aplicación cuenta con diferentes colores y temáticas personalizables. La visualización de los datos se hace a través de gráficas, mapas o indicadores. La interfaz es fluida y el diseño de la aplicación es agradable.
Registro de datos	Se utiliza un dispositivo de bajo costo y de gran uso como lo es el sensor MAX30102 para la medición de la frecuencia cardiaca. Los datos se envían a través de Bluetooth a la aplicación. Dichos datos se pueden listar, enviar, editar, eliminar o graficar.	Para la medición de frecuencia cardiaca, el usuario debe contar con el reloj inteligente de Apple. Dicho dispositivo tiene un costo cercano a COP 1'000.000. Los datos se miden constantemente. La información se presenta en gráficas donde se tiene la posibilidad de ordenar por tiempo y lugar.
Tecnología	La aplicación se implementa utilizando el sistema operativo Android que es de gran demanda. Además, se puede utilizar en un Smartphone de bajo precio. Por otro lado, el dispositivo de medición de FC cuenta con elementos no tan costosos y accesibles en el mercado.	Se debe contar con un dispositivo Smartphone Apple (iPhone, iPad, iPod o iMac). Además, tiene un costo de COP 3.500 en la tienda oficial de Apple.
Usabilidad	La aplicación se puede usar por cualquier persona que cuente con un dispositivo Android con una versión superior o igual a Oreo.	Se debe contar con un dispositivo con sistema operativo iOS en conjunto con el reloj inteligente Apple Watch.

Figura 7.5: Resumen comparativo entre **HeartWatch** e **IscodeApp**

## 7.5. Trabajo futuro

Como se ha mencionado hasta ahora, las enfermedades relacionadas con el corazón son predominantes hoy en día. De esa manera, socialmente existe un gran interés por reducir las cifras de personas

que padecen dichas enfermedades. Además, económicamente se pretende reducir costos médicos en tratamientos y el ausentismo laboral por las incapacidades. De esa manera, **IscodeApp** debe seguir evolucionando y mejorando la capacidad que tiene de acompañar el tratamiento, o incluso, la prevención de las enfermedades cardiovasculares. Por ello un aspecto importante que se puede trabajar en el futuro es el de agregar más funciones como predicciones para personas que nunca han tenido algún episodio de enfermedad cardiovascular. Por otro lado, se puede aumentar la escala de clasificación del modelo actual, como lo mencionó el potencial usuario de la sección anterior. Sin embargo, para aumentar dicha escala se necesita de una gran cantidad de muestras en la base de datos. Es así como, **haciendo las pruebas de un modelo multclasificación (4 posibles predicciones) con las 297 muestras de la base de datos de *Cleveland*, se observó que la precisión era del 50 %.** Además, **conseguir más muestras requiere la información de muchas personas, al igual que, el acompañamiento de un especialista en el área para validar la información de cada muestra.** Esto último se vuelve difícil, ya que la cantidad de especialistas en esta área es reducida, por lo que hay poca disponibilidad de estos para realizar un trabajo conjunto para aumentar el tamaño de la base de datos. De esa manera, se debe crear un proyecto de investigación con médicos e ingenieros para trabajar el problema.

También, en el futuro se puede hacer un plan de pruebas con el especialista para validar la predicción que arroja la aplicación (comparar el valor real de riesgo de padecer EIC con el que arroja la aplicación). De igual forma, se le puede pedir su opinión sobre el diseño de **IsocdeApp** y el modelo de predicción. Para esto último, se espera que el especialista mencione si los atributos son los adecuados y si hay alguno de ellos que sea más importante que el otro para tenerlo en cuenta en el desarrollo del modelo. Igualmente, hace falta realizar un trabajo en conjunto con otras personas para evaluar todos los aspectos de la aplicación (en la evaluación solo se consultó a una persona), ya que por cuestiones de tiempo y restricciones por la pandemia COVID-19 no se lograron hacer más pruebas de la aplicación.

Otro aspecto importante para **IscodeApp** es que se encuentre disponible en la mayor cantidad de plataformas posibles. De esa manera, en un futuro, se puede desarrollar la aplicación para iOS o Windows de forma nativa para que alcance a más personas. Igualmente, se debe contar con la ayuda de algún diseñador gráfico que permita el desarrollo de una aplicación más llamativa y de interfaz agradable. Además, se espera que **IscodeApp** esté disponible en internet en forma de página web, donde el especialista pueda manejar y conocer la información de sus pacientes desde cualquier dispositivo. Para esto último, Firebase tiene la capacidad de desplegar una página web gracias a la opción de *Hosting*. De esa manera, en la figura [7.6](#) se observa una aproximación a dicha página web. Cabe recordar que el desarrollo de dicha página web está por fuera de los alcances de este trabajo de grado, por lo que se plantea como un proyecto futuro como se ha mencionado.

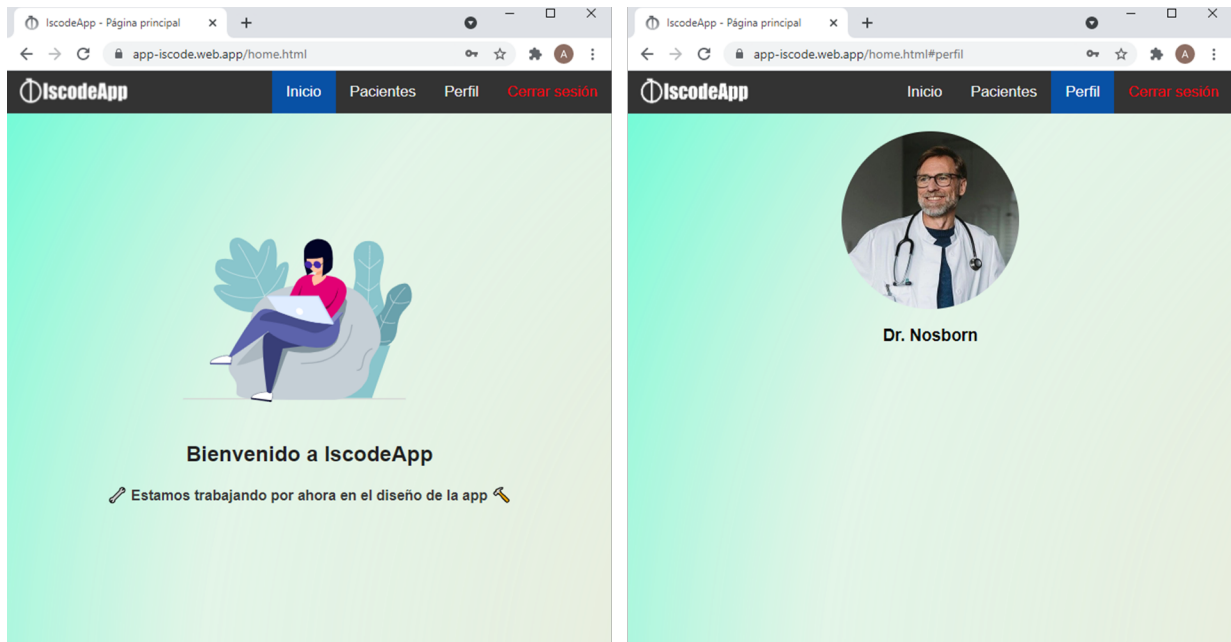


Figura 7.6: “IscodeWeb”

En cuanto al dispositivo de medición de frecuencia cardiaca, se espera que pueda utilizar baterías recargables, pueda ser más pequeño y pueda ser usado constantemente como una pulsera. Además, se pueden añadir más elementos para que pueda medir o recolectar información adicional sobre la salud del paciente. También, hace falta realizar un plan de pruebas junto con el especialista para validar la información que entrega el dispositivo de medición de frecuencia cardiaca. Finalmente, con un desarrollo de **IscodeApp** más avanzado, se espera colocar la aplicación y el dispositivo de medición en el mercado. Además, se desea poder participar en convocatorias relacionadas con la salud cardiovascular para darle más visibilidad al proyecto y conseguir apoyo adicional.

## Capítulo 8

# Conclusiones

Se desarrolló una aplicación para un dispositivo móvil con la funcionalidad de predicción de riesgo de padecer una enfermedad cardíaca a partir del uso de Machine Learning tomando como referencia una base de datos pública con un buen resultado de predicción. También, la aplicación posee otras funcionalidades que pretenden ayudar al paciente en el tratamiento de la enfermedad.

La técnica de las nueve ventanas permitió caracterizar el procedimiento y la funcionalidad de algunas aplicaciones, métodos o sistemas actuales que ayudan a la predicción del riesgo de padecer una isquemia cardíaca.

La revisión de la literatura científica da cuenta de trabajos de optimización de algoritmos y presentan propuestas interesantes para contribuir al cuidado y mejoramiento de la salud de las personas utilizando métodos de ML. Sin embargo, muy pocos tienen una orientación de servicio al paciente, lo cual se interpreta como una ausencia importante que requiere de la producción de alternativas para cuidado de los pacientes.

La matriz de contradicciones fue esencial a la hora de determinar de manera estructurada un conjunto de requerimientos a partir del uso de las herramientas TRIZ dirigidas a la búsqueda de una alternativa de solución teniendo en cuenta la opinión de potenciales usuarios de la aplicación.

La participación de distintas disciplinas, favorecen la observación de diferentes aspectos de un mismo producto y junto con el trabajo multidisciplinar, se amplía la visión de una problemática y permite encontrar mejores soluciones de diseño. Sin embargo, en cuanto a enfermedades del corazón, es escaso el personal humano capacitado para trata dichas enfermedades, por lo que se debe hacer un esfuerzo adicional para poder concretar un trabajo en conjunto con este tipo de especialistas.

Para los pacientes que padecen alguna enfermedad cardiovascular, las ayudas técnicas son útiles, pero necesitan alguna forma de motivación para que sigan adecuadamente el tratamiento. Así, este tipo de aplicaciones, como **IscodeApp**, debe estar mejorando continuamente para cubrir ese aspecto y mantener el paciente “enganchado” a su tratamiento.

Para tener una multclasificación del riesgo de padecer una isquemia cardíaca (riesgo bajo, medio, alto o ausente), el modelo ML se necesita agrandar la base de datos actual, por lo que se requiere un esfuerzo adicional para diseñar la logística de recolección y validación de nueva información. Con lo anterior, también es importante formular un proyecto de investigación para utilizar estrategias de cómputo y el análisis médico, para realizar un Proceso de Análisis Jerárquico (**Saaty**) de los

atributos del modelo con el fin de conocer cuáles de ellos son más importantes y mejorar el modelo.

Tensorflow, Firebase y Android Studio son plataformas amigables para el prototipado rápido de modelos Machine Learning en una aplicación móvil. Actualmente existen condiciones y potencial para el desarrollo de soluciones tecnológicas valiosas utilizando ML en dispositivos con restricciones de hardware.

Las tecnologías de placas reducidas permiten a los desarrolladores crear nuevos dispositivos de manera fácil y económica, lo cual beneficia a la población menos favorecida pues pueden tener acceso a soluciones tecnológicas de bajo costo y buena calidad. De esa manera, el microcontrolador ESP32 permite la integración hardware/software de manera amigable con el desarrollador, pues, a través de sus múltiples módulos, logran reducir la cantidad de materiales usados en el desarrollo de prototipos.

# Bibliografía

- [1] S. Savransky, *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving* (Primera edición), *CRC Press*, 2000.
- [2] Y. Salamatov, “The right solution at the right time”, 1999.
- [3] SolidCreativity, TRIZ40, [Online] Disponible en: [http://www.triz40.com/aff\\_Matrix\\_TRIZ.php](http://www.triz40.com/aff_Matrix_TRIZ.php)
- [4] MedlinePlus, “Sistema cardiovascular”. Disponible en: <https://medlineplus.gov/spanish/ency/anatomyvideos/000023.htm>, 2019. Consultado el 12 de septiembre de 2020.
- [5] Texas Heart Institute, “Anatomía del corazón”. Disponible en: <https://www.texasheart.org/heart-health/heart-information-center/topics/anatomia-del-corazon/>. Consultado el 12 de septiembre de 2020.
- [6] Bupa Global Latinoamérica, “Sistema cardiovascular”. Disponible en: <https://www.bupasalud.com/salud/sistema-cardiovascular>. Consultado el 12 de septiembre de 2020.
- [7] Texas Heart Institute, “La sangre”. Disponible en <https://www.texasheart.org/heart-health/heart-information-center/topics/la-sangre/>. Consultado el 12 de septiembre de 2020.
- [8] K. E. Barret, S. M. Barman, S. Boitano, H. L . Brooks, “Fisiología Médica Ganong”, *Mc Graw Hill LANGE*, ed. 23, p.515, 2010.
- [9] Genomasur, “Capítulo 13: Sistema Circulatorio”. Disponible en: [http://www.genomasur.com/BCH/BCH\\_libro/capitulo\\_13.htm](http://www.genomasur.com/BCH/BCH_libro/capitulo_13.htm). Consultado el 12 de septiembre de 2020.
- [10] Texas Heart Institute, “Las arterias coronarias”. Disponible en: <https://www.texasheart.org/heart-health/heart-information-center/topics/las-arterias-coronarias/>. Consultado el 12 de septiembre de 2020.
- [11] National Heart, Lung and Blood institute, *Enfermedad coronaria*, 2019. Disponible en: <https://www.nhlbi.nih.gov/health-topics/espanol/enfermedad-coronaria-0>. Consultado el 12 de septiembre de 2020.
- [12] Oracle, “¿Qué es la ciencia de datos?”. Disponible en: <https://www.oracle.com/co/data-science/what-is-data-science/>. Consultado el 12 de agosto de 2021.
- [13] Oracle, “¿Qué es la inteligencia artificial (IA)?”. Disponible en: <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>. Consultado el 12 de agosto de 2021.
- [14] J. A. Rodrigo, “Regresión logística simple y múltiple”. Disponible en: [https://www.cienciadedatos.net/documentos/27\\_regresion\\_logistica\\_simple\\_y\\_multiple](https://www.cienciadedatos.net/documentos/27_regresion_logistica_simple_y_multiple). Consultado el 13 de agosto de 2021.
- [15] A. Goyal, “Logistic Regression with Keras”, (2021). Disponible en: <https://www.marktechpost.com/2021/04/08/logistic-regression-with-keras/>. Consultado el 13 de agosto de 2021.
- [16] IBM, “El modelo de redes neuronales”. Disponible en: <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model>. Consultado el 13 de agosto de 2021.

- [17] KDNuggets, “Optimization Algorithms in Neural Networks”. Disponible en: <https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html>. Consultado el 13 de agosto de 2021.
- [18] L. Lima, “Cardiopatía isquémica: qué es, principales síntomas y tratamiento”. [Figura]. 2020. Disponible en <https://www.tuasaude.com/es/isquemia-cardiaca/>. Consultado el 12 de febrero de 2021.
- [19] G. Gridling, B. Weiss, “Introduction To Microcontrollers”, *Vienna University of Technology* 26 de febrero de 2007. Consultado el 12 de agosto de 2021.
- [20] GCFGlobal, “¿Qué son las aplicaciones?”. Disponible en: <https://edu.gcfglobal.org/es/cultura-tecnologica/que-son-las-aplicaciones-o-programas/1/>. Consultado el 12 de agosto de 2021.
- [21] C. Hernández M., “Las tres enfermedades que matan más que la violencia”, *El Tiempo*, 2018 [Online]. Disponible en: <https://www.eltiempo.com/colombia/otras-ciudades/enfermedades-que-son-las-principales-causas-de-muerte-de-los-colombianos-175988>. Consultado: el 7 de abril de 2020.
- [22] J. Castellano, J. Narula, J. Castillo, V. Fuster, “Promoción de la salud cardiovascular global: estrategias, retos y oportunidades”, *Revista española de cardiología*, vol. 67, no. 9, p. 724-730, 2014.
- [23] D. Cabrera, “Los Infartos Son La Primera Causa De Muerte De Los Colombianos: Expertos”, *RCN Radio*, 2019[online]. Disponible en: <https://www.rcnradio.com/salud/los-infartos-son-la-primera-causa-de-muerte-de-los-colombianos-expertos>. Consultado el 14 marzo de 2020.
- [24] A. Ul Haq , J. Ping Li ,M. Hammad Memon ,S. Nazir ,R. Sun, “A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms”, *Hindawi Mobile Information Systems*, vol. 2018, p.1, 2018.
- [25] H. Kahtan, K. Z. Zamli, W. N. Ashikin W. A. Fatthi, A. Abdullah, M. Abdulleteef, N. S. Kamarulzaman, “Heart Disease Diagnosis System Using Fuzzy Logic”, *ICSCA 2018: Proceedings of the 2018 7th International Conference on Software and Computer Applications*, p. 300, 2018.
- [26] K. Bache and M. Lichman, “UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science, Irvine, CA, 2013.
- [27] A. H. Gonsalves, F. Thabtah, R. M. A. Mohammad, G. Singh, “Prediction of Coronary Heart Disease using Machine Learning: An Experimental Analysis, *ICDLT 2019: Proceedings of the 2019 3rd International Conference on Deep Learning Technologies*, p. 51-56, 2019.
- [28] P. I. Dorado Diaz, J. Sampedro-Gómez, V. V. Palacios, P. L. Sánchez, “Applications of Artificial Intelligence in Cardiology. The Future is Already Here”, *Revista Española de Cardiología*, vol. 72, no. 12, p. 2, 2019.
- [29] A. Vallée, A. Cinaud, V. Blachier, H. Lelong, M. E. Safara, J. Blachera, “Coronary heart disease diagnosis by artificial neural networks including aortic pulse wave velocity index and clinical parameters”, *Journal of hypertension*, vol. 37, no. 8, p. 5, 2019.
- [30] P. Amirtharaj, K.Rajeswari, V.Vaithiyanathan, “A novel risk level classification of ischemic heart disease using artificial neural network technique”, *University Journal of Surgery and Surgical Specialities*, vol. 4, no. 4, p. 1-4, 2018.
- [31] CM. Wu, M. Badshah, V. Bhagwat, “Heart Disease Prediction Using Data Mining Techniques”, *DSIT 2019: Proceedings of the 2019 2nd International Conference on Data Science and Information Technology*, p. 7-11, 2019

- [32] A. Mahajan, G. Pottie, W. Kaiser, “Transformation in Healthcare by Wearable Devices for Diagnostics and Guidance of Treatment”, *ACM Transactions on Computing for Healthcare*, vol. 1, no. 1, p. 1-10, 2020.
- [33] Greenpeace, “10 datos preocupantes sobre los efectos del actual nivel de consumo de carne en nuestra salud”. Disponible en <https://es.greenpeace.org/es/noticias/10-datos-preocupantes-sobre-los-efectos-de-la-carne-en-nuestra-salud/>. Consultado el 9 de agosto de 2021.
- [34] P. W. F. Wilson, R. B. D’Agostino, D. Levy, A. M. Belanger, H. Silbershatz, W. B. Kannel, “Prediction of Coronary Heart Disease Using Risk Factor Categories”, *American Heart Association Journal*, vol. 97, n. 18. p.1837-1847. Consultado el 10 de septiembre de 2020.
- [35] Framingham Heart Study, “About the Framingham Heart Study”. Disponible en: <https://framinghamheartstudy.org/fhs-about/>. Consultado el 10 de septiembre de 2020.
- [36] Knowledge Extraction based on Evolutionary Learning. (2004-2018). South African Heart data set. Retrieved from KEEL (Knowledge Extraction based on Evolutionary Learning): <https://sci2s.ugr.es/keel/dataset.php?cod=184>. Consultado el 10 de septiembre de 2020.
- [37] Mayo Clinic, Isquemia miocárdica, 2019. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/myocardial-ischemia/symptoms-causes/syc-20375417>. Consultado el 19 de septiembre de 2020.
- [38] Portafolio, “Aumenta el consumo de comida ‘chatarra’ en los jóvenes colombianos”. Disponible en <https://www.portafolio.co/tendencias/aumenta-el-consumo-de-comida-chatarra-en-los-jovenes-colombianos-524158>, 2018. Consultado el 19 de marzo de 2021.
- [39] ANDI, “Segundo informe de seguimiento sobre salud y estabilidad en el empleo”. Disponible en: <http://www.andi.com.co/Uploads/II%20Informe%20de%20Seguimiento%20ausentismo%20laboral%20y%20ELR-2.pdf>, 2017. Consultado el 16 de abril de 2021.
- [40] P. González Moreno, “Lo que hay detrás del miedo de ir al médico”, *El tiempo*, Disponible en: <https://www.eltiempo.com/salud/por-que-las-personas-sienten-miedo-de-ir-al-medico-281046>, 2018. Consultado el 10 de septiembre de 2019
- [41] PNUD, “Objetivos de Desarrollo Sostenible”. Disponible en: <https://www.undp.org/content/undp/es/home/sustainable-development-goals.html>, 2015.
- [42] PNUD, “Objetivo 3: Salud y bienestar”. Disponible en: <https://www.undp.org/content/undp/es/home/sustainable-development-goals/goal-3-good-health-and-well-being.html#targets>, 2015.
- [43] PNUD, “Objetivo 9: Industria, innovación e infraestructura”. Disponible en: <https://www.undp.org/content/undp/es/home/sustainable-development-goals/goal-9-industry-innovation-and-infrastructure.html#targets>, 2015.
- [44] Revista Dinero, “¿Los colombianos están dispuestos a pagar más de \$1’000.000 por un celular?”. Disponible en: <https://www.dinero.com/tecnologia/articulo/mercado-de-los-celulares-en-colombia-2019/280290>, 2019. Consultado el 25 de marzo de 2021.
- [45] Google Play, “Ritmo Cardíaco - Monitor Pulso”. Disponible en: <https://play.google.com/store/apps/details?id=si.modula.android.instantheartate>. Consultado el 22 de marzo de 2021.
- [46] D. Bernal, “HeartWatch, la app con más información sobre nuestro ritmo cardíaco, se actualiza con nuevas funciones”. *Applesfera*. Disponible en: <https://www.applesfera.com/aplicaciones-ios-1/>
-

- [heartwatch-app-informacion-nuestro-ritmo-cardiaco-se-actualiza-nuevas-funciones](#), 2020. Consultado el 25 de marzo de 2021.
- [47] Fundación Española del Corazón, “ARIADNA”. Disponible en: <https://fundaciondelcorazon.com/corazon-facil/apps/3259-ariadna.html>. Consultado el 1 de febrero de 2021.
- [48] Organización Mundial de la Salud, “Aumentar el consumo de frutas y verduras para reducir el riesgo de enfermedades no transmisibles”. Disponible en: [https://www.who.int/elena/titles/fruit\\_vegetables\\_ncds/es/](https://www.who.int/elena/titles/fruit_vegetables_ncds/es/), 2019.
- [49] J. A. Prados, “Telemedicina, una herramienta también para el médico de familia”. *Atención Primaria*, vol. 45, n. 3, p.129, 2013. Consultado el 1 de junio de 2021.
- [50] Organización Mundial de Salud, “DIETA, NUTRICIÓN Y PREVENCIÓN DE ENFERMEDADES CRÓNICAS”, *Serie de Informes Técnicos*, n. 916, p.93-95, 2003. Disponible en: [https://apps.who.int/iris/bitstream/handle/10665/42755/WHO\\_TRS\\_916\\_spa.pdf?ua=1](https://apps.who.int/iris/bitstream/handle/10665/42755/WHO_TRS_916_spa.pdf?ua=1).
- [51] A. Wong García, “Riesgos de los conservadores de los alimentos”, *El universal de México*. Disponible en: <https://www.eluniversal.com.mx/menu/los-conservadores-de-los-alimentos-hacen-dano>, 2019. Consultado el 5 de agosto de 2020.
- [52] C. M. Story, “The History of Heart Disease”, *Healthline*. Disponible en: <https://www.healthline.com/health/heart-disease/history>, 2018.
- [53] Observatorio Laboral para la Educación, “Programas académicos”. Disponible en: <http://bi.mineducacion.gov.co:8380/eportal/web/men-observatorio-laboral/programas-academicos>
- [54] A. Lima, Sistema cardiovascular: Qué es, función, anatomía y enfermedades, *TUASA ÚDE*. Disponible en: <https://www.tuasaude.com/es/sistema-cardiovascular/>. Consultado el 21 de mayo de 2021.
- [55] Experto Portátil, “Portátil para Machine Learning en 2020”. Disponible en: <https://expertoportatil.com/portatiles/portatil-para-machine-learning/>. Consultado el 10 de agosto de 2021.
- [56] SpringBoard, “Best language for Machine Learning: Which Programming Language to Learn”. Disponible en: <https://in.springboard.com/blog/best-language-for-machine-learning/>. Consultado el 10 de agosto de 2021.
- [57] MaximIntegrated, “High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health”. *Hoja técnica*. Disponible en: [https://cdn.shopify.com/s/files/1/2267/1961/files/Sensor\\_MAX30102\\_de\\_Frecuencia\\_Cardiaca\\_y\\_Oxigeno\\_en\\_la\\_Sangre.pdf?240](https://cdn.shopify.com/s/files/1/2267/1961/files/Sensor_MAX30102_de_Frecuencia_Cardiaca_y_Oxigeno_en_la_Sangre.pdf?240). Consultado el 19 de agosto de 2021.
- [58] Ferretrónica, “Sensor MAX30102 Frecuencia Cardíaca y Oxígeno en Sangre”. Disponible en: <https://ferretronica.com/products/sensor-max30102-frecuencia-cardiaca-y-oxigeno-en-sangre>. Consultado el 27 de agosto de 2021.
- [59] Espressif, “A feature-rich MCU with integrated Wi-Fi and Bluetooth connectivity for a wide-range of applications”. Disponible en: <https://www.espressif.com/en/products/socs/esp32>. Consultado el 19 de agosto de 2021.
- [60] Mayo Clinic, “Angina de pecho”. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/angina/symptoms-causes/syc-20369373>. Consultado el 30 de agosto de 2021.
- [61] My EKG, “Valoración del Segmento ST”. Disponible en: <https://www.my-ekg.com/como-leer-ekg/segmento-st.html>. Consultado el 30 de agosto de 2021.

- [62] Mayo Clinic, “Hipertrofia ventricular izquierda”. Disponible en: <https://www.mayoclinic.org/es-es/diseases-conditions/left-ventricular-hypertrophy/symptoms-causes/syc-20374314>. Consultado el 30 de agosto de 2021.
- [63] MedlinePlus, “Fluoroscopia”. Disponible en: <https://medlineplus.gov/spanish/pruebas-de-laboratorio/fluoroscopia/>. Consultado el 30 de agosto de 2021.
- [64] MedicineNet, “Medical Definition of Thallium scan”. Disponible en: [https://www.medicinenet.com/thallium\\_scan/definition.htm](https://www.medicinenet.com/thallium_scan/definition.htm). Consultado el 30 de agosto de 2021.
- [65] Developers, “Introducción a Android Studio”. Disponible en: <https://developer.android.com/studio/intro?hl=es-419>. Consultado el 30 de agosto de 2021.
- [66] Tensorflow, “Plataforma de extremo a extremo de código abierto para el aprendizaje automático”. Disponible en: <https://www.tensorflow.org/>. Consultado el 30 de agosto de 2021.
- [67] Firebase, “Firebase helps you build and run successful apps”. Disponible en: <https://firebase.google.com/>. Consultado el 10 de agosto de 2021.
- [68] PyTorch, “PYTORCH DOCUMENTATION”. Disponible en: <https://pytorch.org/docs/stable/index.html>. Consultado el 30 de agosto de 2021.
- [69] Scikit-learn, “About us”. Disponible en: <https://scikit-learn.org/stable/about.html#citing-scikit-learn>. Consultado el 10 de agosto de 2021.
- [70] Chaquopy, “FAQ”. Disponible en: <https://chaquo.com/chaquopy/doc/current/faq.html>. Consultado el 10 de agosto de 2021.
- [71] Udacity, “Tensorflow or PyTorch : The force is strong with which one?”. Disponible en: <https://medium.com/@UdacityINDIA/tensorflow-or-pytorch-the-force-is-strong-with-which-one-68226bb7dab4>. Consultado el 4 de agosto de 2021.
- [72] Arduino, “What is Arduino?”. Disponible en: <https://www.arduino.cc/en/Guide/Introduction>. Consultado el 4 de agosto de 2021.
- [73] GitHub, “SparkFun MAX3010x Sensor Library”. Disponible en: [https://github.com/sparkfun/SparkFun\\_MAX3010x\\_Sensor\\_Library/tree/master/examples/Example5\\_HeartRate](https://github.com/sparkfun/SparkFun_MAX3010x_Sensor_Library/tree/master/examples/Example5_HeartRate). Consultado el 5 de agosto de 2021.
- [74] TensorFlow, “Convertor de TensorFlow Lite”. Disponible en <https://www.tensorflow.org/lite/convert?hl=es-419>. Consultado el 12 de agosto de 2021.
- [75] J. A. Rodrigo, “Validación de modelos predictivos: Cross-validation, OneLeaveOut, Bootstrapping”, (2020). Disponible en [https://www.cienciadatos.net/documentos/30\\_cross-validation\\_oneleaveout\\_bootstrap](https://www.cienciadatos.net/documentos/30_cross-validation_oneleaveout_bootstrap). Consultado el 12 de agosto de 2021.
- [76] Last Minute Engineers, “Insight Into ESP32 Sleep Modes & Their Power Consumption”. Disponible en <https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/>. Consultado el 19 de agosto de 2021.
- [77] Maxim integrated, “MAX30102, High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health”. Disponible en <https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX30102.html>. Consultado el 19 de agosto de 2021.

# Capítulo 9

## Anexos

### 9.1. Anexo A: Código en Arduino IDE del ESP32

```
#include <Wire.h>
#include "MAX30105.h"
#define LED_INTEGRADO 2
#include "heartRate.h"
MAX30105 particleSensor;
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;

const byte RATE_SIZE = 4;
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;
float beatsPerMinute;
float beatAvg;

char DataBluetooth = 'N';
String dataout = "";
char buf[8];

int statusSensor=0;
int statusA=0;
int statusB=0;
int statusC=0;

void setup(){
  Serial.begin(115200);
  SerialBT.begin("Sensor de Frecuencia Cardiaca");
  Serial.println("Initializing...");
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)){
    Serial.println("MAX30105 no encontrado. Revisar conexiones o energía ");
    while (1);
  }
  Serial.println("Colocar dedo indice con una presión adecuada sobre el sensor.");
  pinMode(LED_INTEGRADO, OUTPUT);
```

```

particleSensor.setup();
particleSensor.setPulseAmplitudeRed(0x0A);
particleSensor.setPulseAmplitudeGreen(0);
}

void loop(){

    if (SerialBT.available()) {
        DataBluetooth = SerialBT.read();
        Serial.print(DataBluetooth);
        Serial.print("\n");
    }

    //Apagar sensor
    if (DataBluetooth=='N'){
        if (statusC==0){
            rates[RATE_SIZE]={};
            rateSpot = 0;
            lastBeat = 0;
            beatsPerMinute=0;
            beatAvg=0;
            particleSensor.shutdown();
            SerialBT.write((const uint8_t*)"@APAGADO",8);
            digitalWrite(LED_INTEGRADO,LOW);
            statusSensor=0;
            statusC=1;
        }
    }

    //Despertar el sensor
    else if(DataBluetooth=='S'){

        statusC=0;
        if (statusSensor==0){
            particleSensor.wakeUp();
            statusSensor=1;
        }

        long irValue = particleSensor.getIR();
        if (checkForBeat(irValue) == true){
            //Se detectó una pulsación
            long delta = millis() - lastBeat;
            lastBeat = millis();
            beatsPerMinute = 60 / (delta / 1000.0);
            if (beatsPerMinute < 255 && beatsPerMinute > 20){
                rates[rateSpot++] = (byte)beatsPerMinute;
                rateSpot %= RATE_SIZE;

                //Realizar el promedio de las medidas
                beatAvg = 0;
                for (byte x = 0 ; x < RATE_SIZE ; x++)
                    beatAvg += rates[x];
            }
        }
    }
}

```



## 9.2. Anexo B: Android Studio

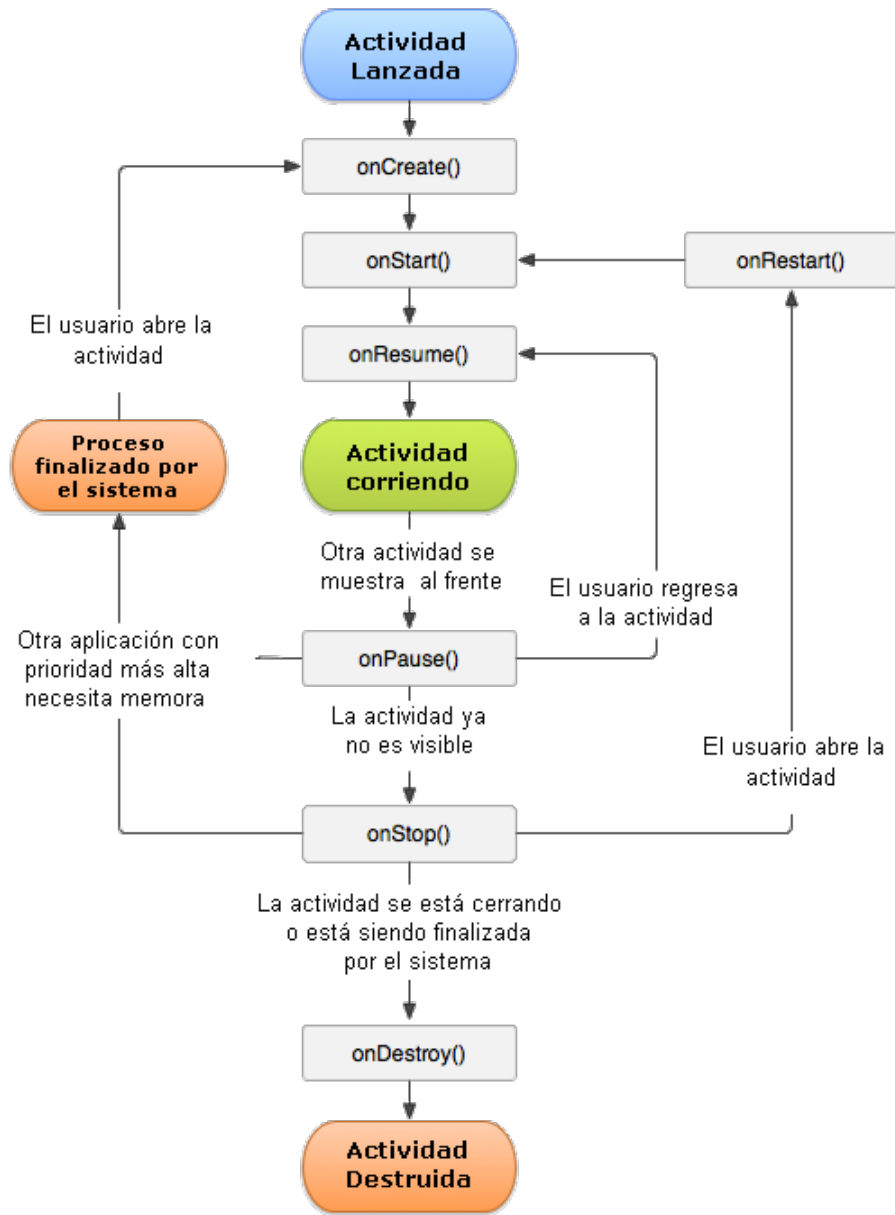


Figura 9.1: Ciclo de vida de una Actividad

### Programación en Android Studio:

Las interfaces en Android Studio se declaran como una **Activity**. La clase Activity es un componente clave de una aplicación para Android, y la forma en que se inician y se crean las actividades es una parte fundamental del modelo de aplicación de la plataforma. A diferencia de los paradigmas de programación en los que las aplicaciones se inician con un método *main()*, el sistema Android inicia el código en una instancia de Activity, invocando métodos de devolución de llamada específicos que corresponden a etapas específicas de su ciclo de vida. Ahora bien, la experiencia con la aplicación para dispositivos móviles difiere de la versión de escritorio, ya que la interacción del usuario con la aplicación no siempre comienza en el mismo lugar. En este caso, no hay un lugar específico desde donde el usuario comienza su actividad. Por ejemplo, si se abre una aplicación de correo electrónico desde el menú de aplicaciones, es

posible que se vea una lista de correos electrónicos en la bandeja de entrada. Por el contrario, si se usa una aplicación de redes sociales que luego inicia la aplicación de correo electrónico, es posible que acceda directamente a la pantalla de la aplicación de correo electrónico para redactar uno.

Cuando una aplicación invoca a otra, la aplicación que realiza la llamada invoca una actividad en la otra, en lugar de toda la aplicación en sí. De esta manera, la actividad sirve como el punto de entrada para la interacción de una aplicación con el usuario. En Android Studio se implementa una actividad como una subclase de la clase Activity. Una actividad proporciona la ventana en la que la aplicación dibuja su interfaz con el usuario. Por lo general, esta ventana llena la pantalla por completo, pero puede ser más pequeña y flotar sobre otras ventanas. Generalmente, una actividad implementa una pantalla en una aplicación. Por ejemplo, una actividad de una aplicación puede implementar una pantalla Preferencias mientras otra implementa una pantalla Seleccionar foto.

Información tomada desde: <https://developer.android.com/guide/components/activities/intro-activities>?hl=es



Código XML

Interfaz con el usuario

Figura 9.2: Interfaz con el usuario en Android

En la figura 9.2 se observa la programación de una interfaz sencilla en Android Studio. Para lo anterior se emplea código xml. En dicha figura se presentan 3 elementos comunes en una aplicación android: una imagen, un texto y un botón. De esa manera, en la parte izquierda de la imagen aparece el código xml y en la parte derecha está el resultado visual que genera dicho código. Lo anterior permite dar una idea de como se harán todas las interfaces de la aplicación.

```

1  package com.proyecto.iscodeapp.Funciones;
2  import ...
10
11 public class prueba extends AppCompatActivity {
12     ImageView imagen;
13     TextView texto;
14     Button boton;
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.prueba);
19         imagen=findViewById(R.id.imagen);
20         texto=findViewById(R.id.texto);
21         boton=findViewById(R.id.boton);
22     }
23 }

```

Figura 9.3: Programación con Java en Android Studio

Por otro lado, el funcionamiento de la aplicación se realiza con el lenguaje de programación Java. De esa manera, si se quiere controlar algún elemento de la interfaz con el usuario, se debe “llamar” el elemento xml dentro del código en Java (los elementos en la figura 9.2 se les asigna un id para poder diferenciarlos). Lo anterior se puede apreciar en la figura 9.3. En dicha figura, se observa que primero se importan las librerías que necesita cada elemento, luego, se crea la clase “prueba” que es una extensión de una actividad. Después, se inicializan los tipos de variables junto con sus nombres. Después en el método `onCreate` se “invoca” la vista xml (`setContentView()`) y los elementos dentro de esta vista.

### Gradle en Android Studio

Android Studio usa Gradle, que es un paquete de herramientas de compilación avanzadas, para automatizar y administrar el proceso de compilación, y al mismo tiempo definir configuraciones de compilación personalizadas y flexibles. Cada configuración de compilación puede definir su propio conjunto de código y recursos, y reutilizar las partes comunes a todas las versiones de la aplicación. El complemento de Android para Gradle funciona con el paquete de herramientas de compilación a fin de proporcionar procesos y ajustes configurables específicos para la compilación y prueba de aplicaciones para Android. El sistema de compilación de Android compila recursos y código fuente de la aplicación, y los empaqueta en APK que se puede probar, implementar, firmar y distribuir. Además, dentro de Android Studio está un archivo Gradle que básicamente define toda la configuración, versión y bibliotecas que utiliza la aplicación. De esa manera, si se quiere utilizar un complemento que no se encuentra actualmente instalado en Android Studio (complemento para mostrar gráficas, animaciones o incluso Firebase), se puede editar el archivo Gradle para instalar el complemento requerido.

Información obtenida de: <https://developer.android.com/studio/build?hl=es-419>

## 9.3. Anexo C: Repositorio en GitHub de la aplicación en Android Studio

En la siguiente dirección se encuentra el repositorio del proyecto de la aplicación. <https://github.com/alexcamayo/IscodeApp>

En la siguiente dirección se encuentra un repositorio donde se alojan la base de datos de Cleveland utilizada para el proyecto, el código C de Arduino para el ESP32 y el Notebook Python del modelo ML:

<https://github.com/alexcamayo/otrasRutinasIscodeApp>

La descripción de los atributos de la base de datos *UCI Machine Learning Repository - Heart Disease Data Set* se encuentra en la siguiente dirección <https://archive.ics.uci.edu/ml/datasets/Heart+Disease> o en el repositorio de Github <https://github.com/alexcamayo/otrasRutinasIscodeApp> en el archivo *descripcioncaracteristicas.txt*.

## 9.4. Anexo D: SQLite

SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones. Además, es el motor de base de datos más utilizado del mundo. De igual forma, SQLite está integrado en todos los teléfonos móviles y la mayoría de las computadoras y viene incluido dentro de innumerables otras aplicaciones que la gente usa todos los días.

Más información en: <https://www.sqlite.org/index.html>

**Código en Java para bases de datos SQL en Android Studio:**

```
import...
public class MyDataBaseHelper extends SQLiteOpenHelper {
    private Context context;
    private static final String DATABASE_NAME="DATOS_USUARIO.db";
    private static final int DATABASE_VERSION=1;
    private static final String TABLE_NAME="recordatorios";

    //Columnas Recordatorios
    private static final String COLUMN_ID="ID";
    private static final String TITULO_REC="TÍTULO";
    private static final String CONTENIDO_REC="CONTENIDO";
    private static final String COLOR_REC="COLOR";
    private static final String FECHA_REC="FECHA";
    private static final String HORA_REC="HORA";
    private static final String IMPORTANCIA_REC="IMPORTANCIA";
    private static final String SEREPITE_REC="SE_REPITE";
    private static final String TIEMPOREP_REC="TIEMPO_REPETICION";
    private static final String INTERVALOREP_REC="INTERVALO_REPETICION";

    public MyDataBaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.context=context;
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String query=
            "CREATE TABLE " + TABLE_NAME +
            " (" + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            TITULO_REC + " TEXT, " +
            CONTENIDO_REC + " TEXT, "+
            COLOR_REC + " TEXT, "+
            FECHA_REC + " TEXT, "+
            HORA_REC + " TEXT, "+
            IMPORTANCIA_REC + " TEXT, "+
            SEREPITE_REC + " TEXT, "+
            TIEMPOREP_REC + " TEXT, "+
            INTERVALOREP_REC + " TEXT);";
        db.execSQL(query);
    }
}
```

```

}

@Override
public void onUpgrade(SQLiteDatabase db,
int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS "+TABLE_NAME);
    onCreate(db);
}

public long agregar_recordatorio(String titulo,
String contenido,String color,
String fecha,String hora,String importancia,
String serepite,String tiemporep,String intervalorep){
    long result;
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues cv = new ContentValues();
    cv.put(TITULO_REC,titulo);
    cv.put(CONTENIDO_REC,contenido);
    cv.put(COLOR_REC,color);
    cv.put(FECHA_REC,fecha);
    cv.put(HORA_REC,hora);
    cv.put(IMPORTANCIA_REC,importancia);
    cv.put(SEREPITE_REC,serepite);
    cv.put(TIEMPOREP_REC,tiemporep);
    cv.put(INTERVALOREP_REC,intervalorep);
    result = db.insert(TABLE_NAME,null,cv);
    if(result!=-1){
        Toast.makeText(context,"No se pudo guardar su recordatorio",Toast.LENGTH_SHORT).show();
    }
    else{
        Toast.makeText(context,"Recordatorio
guardado con éxito: "+result,Toast.LENGTH_SHORT)
        .show();
    }
    return result;
}

public Cursor readAllData(){
    String query="SELECT * FROM "+TABLE_NAME;
    SQLiteDatabase db=this.getReadableDatabase();
    Cursor cursor=null;
    if(db != null){
        cursor= db.rawQuery(query,null);
    }
    return cursor;
}

public Cursor readOneRow(String row_id){
    String query="SELECT * FROM "+TABLE_NAME+" WHERE "+
    COLUMNA_ID+"=" + row_id;
    SQLiteDatabase db=this.getWritableDatabase();
    Cursor cursor=null;

```



```

.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            //Éxito
        }
        else {
            if (task.getException() instanceof
                FirebaseAuthUserCollisionException){
                //Ya existe este usuario
            }
            else{
                //Error
            }
        }
    }
});

//Iniciar sesión con Authorization de Firebase
FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
firebaseAuth.signInWithEmailAndPassword("perez@gmail.com", "perez12345")
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if (task.isSuccessful()) {
            //Éxito
        }
        else {
            //Error
        }
    }
});

//Obtener información desde Real Time Database de Firebase
FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
FirebaseDatabase database = FirebaseDatabase.getInstance().getReference();
String id_usuario= firebaseAuth.getCurrentUser().getUid();
database.child("Usuarios").child(id_usuario)
.child("InfoUsuario").get().addOnCompleteListener(new OnCompleteListener<DataSnapshot>() {
    @Override
    public void onComplete(@NonNull Task<DataSnapshot> task) {
        if (!task.isSuccessful()) {
            //Error
        }
        else {
            String dato=task.getResult().child("Nombre").getValue().toString();
        }
    }
});

//Agregar información a Real Time Database de Firebase

```

```

FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
FirebaseDatabase database = FirebaseDatabase
    .getInstance().getReference();
String id_usuario= firebaseAuth.getCurrentUser().getUid();

Map<String,Object> map= new HashMap<>();
map.put("Nombre", "Juan Perez");
map.put("Correo", "perez@gmail.com");
map.put("Fecha", "10/09/1987");
map.put("ID", id_usuario);

database.child("Usuarios").child(id_usuario)
    .child("InfoUsuario").setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {
        @Override
        public void onComplete(@NonNull @NotNull Task<Void> task) {
            //Éxito
        }
    }).addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull @NotNull Exception e) {
            //Error
        }
    });

//Agregar archivo a Storage de Firebase
FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
String id_usuario= firebaseAuth.getCurrentUser().getUid();
Uri uri=Uri.parse("/data/data/foto.jpeg");
File imagen=new File(uri.getPath());
StorageReference storageReference= FirebaseStorage
    .getInstance().getReference();
StorageReference filepath=storageReference.child(id_usuario)
    .child("FotoUsuario/fotoperfil.jpeg");

filepath.putFile(uri).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
        //Éxito
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull Exception e) {
        //Error
    }
});

//Descargar archivo a Storage de Firebase
FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
String id_usuario= firebaseAuth.getCurrentUser().getUid();
FirebaseStorage storage = FirebaseStorage.getInstance();
StorageReference gsReference = storage

```

```

.getReferenceFromUrl("gs://aplicacion.com/"+id_usuario+"/FotoUsuario/fotoperfil.jpeg");
File fotoimagen=new File("/data/data/foto.jpeg");

gsReference.getFile(fotoimagen).addOnSuccessListener(new
OnSuccessListener<FileDownloadTask.TaskSnapshot>() {
    @Override
    public void onSuccess(FileDownloadTask.TaskSnapshot taskSnapshot) {
        //Éxito
    }
}).addOnFailureListener(new OnFailureListener() {
    @Override
    public void onFailure(@NonNull @NotNull Exception e) {
        //Error
    }
});

```

## 9.6. Anexo F: Rutina para verificar estado Bluetooth y listar los dispositivos vinculados

```

import ...
public class BluetoothActivity extends AppCompatActivity {
    BluetoothAdapter bluetoothAdapter;
    RecyclerView recyclerView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bluetooth);
        recyclerView = findViewById(R.id.recyclerViewDevice);
        VerificarEstadoBT();
    }

    private void VerificarEstadoBT() {
        // Comprueba que el dispositivo tiene Bluetooth y que está encendido.
        bluetoothAdapter= BluetoothAdapter.getDefaultAdapter();
        if(bluetoothAdapter==null) {
            //El Smartphone no soporta Bluetooth
        }
        else {
            if (bluetoothAdapter.isEnabled()) {
                //Bluetooth activado
                mostrarDispositivos();
            }
            else {
                //Solicita al usuario que active Bluetooth
                Intent enableBtIntent = new
                Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
                startActivityForResult(enableBtIntent, 1);
            }
        }
    }
}

```

```
@Override
public void onActivityResult(int requestCode,
int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==1 && resultCode==0){
        //Rechazo la activación de bluetooth
        Intent go_home=new Intent(BluetoothActivity
        .this,MainActivity.class);
        startActivity(go_home);
        finish();
    }
    else if(requestCode==1 && resultCode==1){
        mostrarDispositivos();
    }
}

private void mostrarDispositivos(){
    // Bluetooth configuración
    bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

    // Obtener la lista de dispositivos vinculados
    Set<BluetoothDevice> pairedDevices = bluetoothAdapter
    .getBondedDevices();
    List<Object> deviceList = new ArrayList<>();
    if (pairedDevices.size() > 0) {
        //Obtener los nombres y las direcciones físicas de los dispositivos
        for (BluetoothDevice device : pairedDevices) {
            String deviceName = device.getName();
            String deviceHardwareAddress = device.getAddress(); // MAC address
            DeviceInfoModel deviceInfoModel = new
            DeviceInfoModel(deviceName,
            deviceHardwareAddress);
            deviceList.add(deviceInfoModel);
        }
        // Mostrar los dispositivos utilizando un recyclerView
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        DeviceListAdapter deviceListAdapter = new
        DeviceListAdapter(this,deviceList);
        recyclerView.setAdapter(deviceListAdapter);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
    }
}
}
```

## 9.7. Anexo G: Rutina para recibir y presentar la información que llega a través de Bluetooth

```
import...
public class PruebaBluetoothActivity extends AppCompatActivity {
    private String deviceName = null;
    private String deviceAddress;
    public static Handler handler;
    public static BluetoothSocket mmSocket;
    public static ConnectedThread connectedThread;
    public static CreateConnectThread createConnectThread;
    private final static int CONNECTING_STATUS = 1;
    private final static int MESSAGE_READ = 2;
    private final static int handlerState = 0;
    TextView informacion,datos;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fcregistrar);

        informacion=findViewById(R.id.informacion);
        datos=findViewById(R.id.datos); //frecuencia cardiaca

        deviceName = getIntent().getStringExtra("deviceName");
        if (deviceName != null){
            deviceAddress = getIntent().getStringExtra("deviceAddress");
            BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();
            createConnectThread = new CreateConnectThread(bluetoothAdapter,deviceAddress);
            createConnectThread.start();
        }

        handler = new Handler(Looper.getMainLooper()) {
            @Override
            public void handleMessage(Message msg){
                switch (msg.what){
                    case CONNECTING_STATUS:
                        switch(msg.arg1){
                            case 1:
                                informacion.setText("Conectado a: "+deviceName);
                                break;
                            case -1:
                                informacion.setText("Falló al conectar con el sensor");
                                break;
                        }
                        break;
                    case MESSAGE_READ:
                        String MyDataIn= (String) msg.obj;
                        if(MyDataIn.length()<=8) {
                            PresentarInfo(MyDataIn);
                        }
                }
            }
        }
    }
}
```

```

        else{
            PresentarInfo("");
        }
        break;
    }
}
};

}

public void PresentarInfo(String info){
    int val1=info.indexOf('+');
    int val2=info.indexOf('Y');
    int val3=info.indexOf('T');
    if (val1!=-1 && val2!=-1 && val3!=-1){
        informacion.setText("Leyendo...");
        String datofc=info.substring(0,val1);
        datos.setText(datofc);
    }
    else if (val1==-1 && val2!=-1 && val3!=-1){
        informacion.setText("Leyendo...");
    }
    else if (val1==-1 && val2==-1 && val3!=-1){
        informacion.setText("Coloque su dedo en el sensor");
    }
}

}

//Hilo para crear la conexión Bluetooth
public static class CreateConnectThread extends Thread {
    public CreateConnectThread(
        BluetoothAdapter bluetoothAdapter, String address) {
        BluetoothDevice bluetoothDevice =
            bluetoothAdapter.getRemoteDevice(address);
        BluetoothSocket tmp = null;
        UUID uuid = bluetoothDevice.getUuids()[0].getUuid();

        try {
            tmp = bluetoothDevice.
                createInsecureRfcommSocketToServiceRecord(uuid);

        } catch (IOException e) {
            //Falló la creación del socket bluetooth
        }
        mmSocket = tmp;
    }

    public void run() {
        BluetoothAdapter bluetoothAdapter =
            BluetoothAdapter.getDefaultAdapter();
        bluetoothAdapter.cancelDiscovery();
        try {
            mmSocket.connect();

```

```

        //Dispositivo conectado
        handler.obtainMessage(CONNECTING_STATUS, 1, -1).sendToTarget();
    } catch (IOException connectException) {
        // No se puede conectar
        try {
            mmSocket.close();
            handler.obtainMessage(CONNECTING_STATUS, -1, -1).sendToTarget();
        } catch (IOException closeException) {
        }
        return;
    }
    connectedThread = new ConnectedThread(mmSocket);
    connectedThread.run();
}

// Cerrar el cliente bluetooth.
public void cancel() {
    try {
        mmSocket.close();
    } catch (IOException e) {
        //No se pudo cerrar el cliente
    }
}

}

// Hilo para la transferencia de datos
public static class ConnectedThread extends Thread {
    private final BluetoothSocket mmSocket;
    private final InputStream mmInStream;
    private final OutputStream mmOutStream;
    StringBuffer sb;

    public ConnectedThread(BluetoothSocket socket) {
        mmSocket = socket;
        InputStream tmpIn = null;
        OutputStream tmpOut = null;
        try {
            tmpIn = socket.getInputStream();
            tmpOut = socket.getOutputStream();
        }
        catch (IOException e) { }

        mmInStream = tmpIn;
        mmOutStream = tmpOut;
    }

    public void run() {
        byte[] byte_in = new byte[1];
        sb = new StringBuffer();
        byte[] buffer = new byte[256]; //buffer para guardar los datos recibidos
        int bytes; // bytes returned from read()
        // Se mantiene en modo escucha
    }
}

```



```
y=dataset.iloc[:,13]

#Normalización de los atributos
train_stats = dataset.describe()
train_stats.pop("target")
train_stats = train_stats.transpose()
def norm(x):
    return (x - train_stats['mean']) / train_stats['std']
X = norm(X)

#Validación cruzada de 10 conjuntos
scores=cross_val_score(LogisticRegression(),X,y,cv=10)
print(scores.mean())
```