



eHeart-BP: prototipo de Internet de las Cosas para monitorear la presión arterial utilizando algoritmos de Machine Learning

Lic. Néstor Germán Bolívar Pulgarín

Universidad Nacional de Colombia

Facultad Ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2019

eHeart-BP: prototipo de Internet de las Cosas para monitorear la presión arterial utilizando algoritmos de Machine Learning

Lic. Néstor Germán Bolívar Pulgarín

Trabajo final de profundización presentado como requisito parcial para optar al título
de:

Magister en Ingeniería - Telecomunicaciones

Director (a):

(Ingeniero) Octavio José Salcedo Parra

Codirector (a):

(Ingeniera) Libia Denise Cangrejo Aljure

Línea de Investigación:

Señales e información

Universidad Nacional de Colombia

Facultad Ingeniería, Departamento de Ingeniería de Sistemas e Industrial

Bogotá, Colombia

2019

*La disciplina, humildad y respeto forjan el horizonte hacia
un camino prospero, carente de desolación
y enfocado al ideal de futuro propio.*

Néstor Bolívar

Agradecimientos

Agradezco a la Universidad por las posibilidades de estudio y crecimiento personal que me fueron concedidas a lo largo de la carrera. De modo especial, también reconozco la colaboración de los compañeros de clase y profesores que con sus consejos y conocimientos aportaron en la culminación y construcción de esta investigación.

A los docentes

En las múltiples adversidades encontradas a lo largo de esta senda, destaco los consejos e impulsos del profesor Jesús Tovar cuyas pláticas cimentaron una esfera de experiencias aplicables en la academia.

A la profesora Denise Cangrejo un especial reconocimiento por su disposición frente al respaldo en la divulgación de este proyecto, esas sugerencias y soportes posibilitaron la culminación del mismo.

Al profesor Octavio Salcedo por su disposición en las inquietudes y observaciones donde insistentemente buscaba orientación.

A los amigos

Pretendo mostrar un humilde reconocimiento a Francy cuyos consejos respetuosos, oportunos y sincera amistad permitieron aliviar los desdenes en los momentos de duda.

A Daniel Zúñiga, por su persistencia, aportes y acompañamiento en la aventura de Perú cuyos logros permanecerán en la andanza.

Resumen

Este trabajo se da en respuesta a las altas tasas de muerte en Colombia y en el mundo por hipertensión arterial, problemática que se puede abordar mediante la promoción de los hábitos saludables, el autocuidado y monitoreo sistemático. Hoy en día, se utilizan diversos dispositivos de medición para la supervisión de esta variable biométrica, sin embargo, es notoria la ausencia de un dispositivo práctico que satisfaga los requerimientos de un sistema de monitoreo inteligente y cómodo de la presión arterial. El objetivo de esta investigación es el diseño y desarrollo de un prototipo de comunicación inalámbrica, sustentado en la utilización de un tensiómetro arterial de brazo. Este dispositivo fue modificado para posibilitar el envío de los datos de presión arterial y ritmo cardíaco a un servidor en la web.

Si bien, en la búsqueda de literatura realizada se resaltan el desarrollo de múltiples sistemas de monitoreo de la presión arterial que involucran Tecnologías de la Información y Comunicación. Esta investigación enfatiza la utilización de herramientas del ecosistema digital actual que favorezcan una lectura cómoda y completa sobre los aspectos relevantes entorno a la presión arterial del usuario. También se destaca el diseño de algoritmos de aprendizaje automático que posibiliten el tratamiento de los datos proporcionados por el tensiómetro, de modo que se pueda obtener una probabilidad de riesgo de padecer un episodio de hipertensión mediante un análisis derivado de las medidas de presión arterial del usuario.

En este documento se presenta el prototipo de monitoreo, la descripción y diseño de las capas estructurales que lo componen, tecnologías asociadas y los atributos técnicos que posibilitan su funcionamiento. Finalmente, se incluye un análisis comparativo de un sistema similar, donde se destaca en los resultados obtenidos, la predicción del modelo de aprendizaje de máquina en cerca del 67%. Adicional, es de resaltar la disponibilidad de las mediciones de PA en tiempo cercano al real, posibilitando notificaciones o alertas a eventualidades críticas.

Palabras clave: (*Aprendizaje automático, Internet of Things (IoT), Presión arterial, Tele-medicina*).

Abstract

This work is given in response to the high death rates in Colombia and in the world due to arterial hypertension, problem that can be addressed by promoting healthy habits, self-care and systematic monitoring. Today, various measuring devices are used for the monitoring of this biometric variable, however, the absence of a practical device that meets the requirements of a smart and comfortable blood pressure monitoring system is noticeable. The objective of this research is the design and development of a wireless communication prototype, based on the use of an arm blood pressure monitor. This device was modified to enable the sending of blood pressure and heart rate data to a server on the web.

Although, in the search for literature carried out, the development of multiple blood pressure monitoring systems that involve Information and Communication Technologies are highlighted, this research emphasizes the use of current digital ecosystem tools that favor a comfortable and complete reading about the relevant aspects around the user's blood pressure. It is also highlighted the design of machine learning algorithms that enable the treatment of the data provided by the tensiometer, so that a probability of risk of hypertension can be obtained through an analysis derived from the user's blood pressure measurements.

This document presents the monitoring prototype, the description and design of the structural layers that compose it, associated technologies and the technical and design attributes that enable its operation. Finally, a comparative analysis of a similar system is included, where the prediction of the machine learning model stands out in about 67%. In addition, it is worth highlighting the availability of BP measurements in close to real time, allowing notifications or alerts to critical eventualities.

Keywords: (*Blood pressure, Internet of Things (IoT), Machine learning, Tele-medicine*).

Contenido

	Pág.
Resumen	IX
Lista de figuras	XIII
Lista de tablas	XIV
Lista de Símbolos y abreviaturas	XV
Introducción	1
1. Antecedentes y objetivos	3
1.1 Incidencia de las enfermedades cardiovasculares	3
1.2 Clasificación de los valores de la presión sanguínea	4
1.3 Problemática.....	6
1.4 Propósito del Estudio.....	6
1.5 Pregunta de Investigación.....	7
1.6 Objetivos de la investigación.....	7
1.6.1 Objetivo general.....	8
1.6.2 Objetivos específicos.....	8
2. Estado del arte de los sistemas de monitoreo de ECV sustentados en IoT y tecnologías asociadas	9
2.1 Telecardiología y el monitoreo a distancia.....	9
2.2 Bases de datos utilizadas en desarrollos de ECV	10
2.3 Técnicas de ML orientadas a la predicción de anomalías cardíacas.....	12
2.3.1 Aprendizaje Automático	13
2.3.2 Algoritmos de ML.....	14
2.4 Ecosistema IoT	16
2.4.1 Economías emergentes	16
2.4.2 Retos en IoT	17
2.4.3 Privacidad y seguridad de la información	18
2.5 Contribuciones en el campo de la salud de los sistemas sustentados en IoT.	18
2.5.1 Convergencia tecnológica	19
2.5.2 Limitaciones de los dispositivos de medición	20
3. Metodología de diseño del prototipo	23
3.1 Métodos para responder las preguntas de investigación	24
3.1.1 Enfoque metodológico.....	24
3.1.2 Estrategia de Investigación	25

3.2	Diseño del prototipo.....	26
3.2.1	Adhesión de características en Hardware	27
3.2.2	Diseño de la base de datos en tiempo real.....	29
3.2.3	Diseño del App en Android Studio	31
3.2.4	Diseño de los algoritmos basados en ML	34
3.2.5	Implementación de los algoritmos.....	39
4.	Resultados	45
4.1	Discusión de resultados	47
4.2	Productos realizados	48
5.	Conclusiones y recomendaciones	51
5.1	Conclusiones	51
5.2	Recomendaciones.....	52
A.	Anexo: Código de captura de datos de la tarjeta ESP8266.....	53
B.	Anexo: Modelo para RNA desarrollado en Google Colab	57
C.	Anexo: Manual de uso del prototipo de medición de la PA	62
	Bibliografía	65

Lista de figuras

	Pág.
Figura 1-1: Cifras sobre afecciones cardíacas en Colombia [2].	4
Figura 2-2: Esquema conceptual para del Aprendizaje Automático [14].	13
Figura 2-3: Esquema IoT en un modelo de monitoreo de la salud [20].	20
Figura 3-4: Esquema del prototipo IoT para el monitoreo de la PA, (Fuente autor).	26
Figura 3-5: Tensiómetro arterial seleccionado, (Panasonic EW3106) [25].	27
Figura 3-6: Módulo ESP8266 y placa interna del tensiómetro, (Fuente autor).	27
Figura 3-7: Bytes en formato hexadecimal de las mediciones de la PA [27].	28
Figura 3-8: Ejemplo de un envío de la PA a la DB en tiempo real de Firebase, (Fuente autor).	29
Figura 3-9: Interacción de los usuarios en Firebase, (Fuente autor).	31
Figura 3-10: Diagrama de clases del prototipo, (Fuente autor).	32
Figura 3-11: Interfaces del aplicativo para la visualización de los registros, (Fuente autor).	33
Figura 3-12: Índice porcentual de la herramienta de predicción, (Fuente autor).	33
Figura 3-13: Problema de la hipótesis para el algoritmo regresión lineal [31].	35
Figura 3-14: Función sigmoide utilizada en el algoritmo de regresión logística [31].	36
Figura 3-15: Valor de convergencia de la función de coste [32].	37
Figura 3-16: Distribución de datos en la DB seleccionado, (Fuente autor).	39
Figura 3-17: Distribución de muestras para los atributos PAS y PAD, (Fuente autor).	41
Figura 3-18: Frontera de decisión obtenida del aprendizaje multivariable, (Fuente autor).	41
Figura 3-19: Distribución de muestras del dataset, (Fuente autor).	43
Figura 3-20: Métricas de exactitud obtenidas por el modelo realizado en TensorFlow, (Fuente autor).	44
Figura 3-21: Esquema de comunicaciones del prototipo de monitoreo de la PA, (Fuente autor).	45
Figura 3-22: Comparativa de los resultados en el entrenamiento y la validación del modelo, (Fuente autor).	47

Lista de tablas

Pág.

Tabla 1-1: Rango de valores de presión arterial sistólica y diastólica [3].	4
Tabla 2-2: Métodos para tomar la presión sanguínea [3].	6
Tabla 2-3: Base de datos utilizadas en técnicas de aprendizaje automático [11].	11
Tabla 3-4: Matriz de confusión para la predicción de la hipertensión arterial [12].	38

Lista de Símbolos y abreviaturas

Abreviaturas

Abreviatura	Abreviatura en Ingles	Término
ECV	CVD	Enfermedad Cardiovascular
PA	BP	Presión Arterial
PAS	SBP	Presión Arterial Sistólica
PAD	DBP	Presión Arterial Diastólica
RNA	ANN	Redes Neuronales Artificial
MSV	SVM	Máquinas de Vectores Soporte
ECG	ECG	Electrocardiograma
BD	DB	Base de Datos

Introducción

El paradigma de Internet de las cosas, concibe escenarios diversos de pequeñas “cosas”, inteligentes e interconectadas que capturan información de distinta índole y esencialmente la comunican. Quizás uno de los escenarios más interesantes es el de sensores, los cuales posibilitan la medición en forma permanente y continua de parámetros biométricos, orientados a la detección oportuna y autónoma de la condición de salud de las personas. En este escenario de monitoreo, los parámetros biométricos y otras variables del contexto del paciente pueden ser realizadas bajo los estándares de calidad suficientes para soportar decisiones relevantes de salud y bienestar de los individuos. Adicionalmente, dichas mediciones pueden ser reportadas en tiempo real a expertos del área de la salud, de esta forma se promueve la inclusión de TIC's en el ecosistema de salud y prevención.

Recientemente, la convergencia tecnológica en el cuidado de la salud se ha vuelto relevante porque es una alternativa prometedora para el tratamiento de pacientes con enfermedades crónicas. Puntualmente, la Internet de las cosas (IoT) logra esquematizar un panorama de tecnologías, que en este trabajo se enfocan al autocuidado y promoción de los hábitos saludables. Algunos de los campos en relación a IoT se relacionan a la sensibilidad al contexto, la computación ubicua o las redes de sensores inalámbricos, todos estos ayudan a construir una amplia gama de protocolos y modelos de atención médica. Algunos de esos modelos de monitoreo de la salud están orientados a la planificación de políticas y otros al autocuidado; estos últimos han planteado desafíos tecnológicos asociados con la atención y monitoreo continuo en el hogar, lo que implica la participación de diferentes actores como pacientes, familiares y terceros.

El objetivo principal de esta investigación es diseñar y desarrollar un prototipo de comunicación inalámbrica utilizando un monitor de presión arterial existente e implementando algoritmos de Machine Learning. Estos últimos, permitirán clasificar los registros de presión arterial (PA) del usuario, proporcionando un porcentaje sobre la probabilidad de padecer un episodio de hipertensión arterial. Para lograr esto, es

imprescindible recopilar las principales investigaciones en el área, de modo que se puedan analizar el funcionamiento y relevancia de las tecnologías utilizadas. A continuación, se muestran las secciones en que esta dividido el documento:

El capítulo 1 presenta los antecedentes, el propósito del estudio y el abordaje del problema de investigación.

En el capítulo 2 se menciona el estado de arte referente a las tecnologías, procesos y sistemas de monitoreo orientados a la supervisión de Enfermedades Cardiovasculares (ECV).

Posteriormente, en el capítulo 3 se muestra la metodología de desarrollo y diseño en cada una de las capas del prototipo (Percepción, Red y aplicación).

En el capítulo 4, se describen los resultados obtenidos. Finalmente, se realiza un análisis comparativo respecto a uno de los modelos de monitoreo de salud consultados, mencionando las principales conclusiones de la investigación y las recomendaciones o trabajos futuros asociados.

1. Antecedentes y objetivos

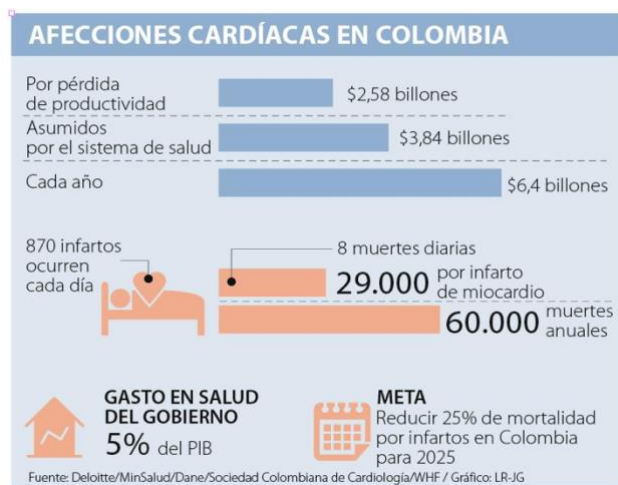
Diversas investigaciones descritas a lo largo de este documento han demostrado que la implementación de las tecnologías habilitantes en IoT podrían mejorar no solo la calidad de vida de las personas, sino también en la prevención de presentar nuevas condiciones de salud e incluso modernizar el monitoreo en la atención médica. En particular, esta sección tiene como objetivo mostrar la importancia de supervisar las Enfermedades Cardiovasculares (ECV) a partir de la contextualización de las cifras evidenciadas en cuanto a morbilidad y mortalidad. Adicionalmente, se proporciona un punto de vista sobre el diagnóstico comúnmente realizado de la Presión Arterial (PA) en los centros médicos. Por otra parte, desde una perspectiva de hardware y software, se mencionan los aspectos de portabilidad y precisión a tener en cuenta al diseñar el prototipo.

1.1 Incidencia de las enfermedades cardiovasculares

Según la Federación Mundial del Corazón [1] las ECV causan 17.5 millones de muertes cada año y además es la principal causa de muerte en todo el mundo. En comparación con el cáncer, la proporción es de hasta veinte a uno, lo que significa que para una persona que muere de cualquier tipo de cáncer, veinte de ellos mueren a causa de enfermedades cardiovasculares. Refiriéndose a Colombia, la **Figura 1-1** muestra estadísticas preocupantes donde casi el 30% del total de muertes en el territorio corresponden a ECV, según un informe sobre la promoción de la salud cardiovascular [2].

En el mismo informe, también se enfatiza sobre la relevancia de las investigaciones que utilizan las herramientas del ecosistema de tecnologías aplicadas al cuidado de la salud. El Ministerio de Salud y Protección Social de Colombia, indica la importancia de que las industrias del sector de la salud estén constantemente innovando con aplicaciones o sistemas de información que identifiquen previamente el riesgo de cualquier ECV y que mejoren la calidad del servicio médico [2].

Figura 1-1: Cifras sobre afecciones cardíacas en Colombia [2].



1.2 Clasificación de los valores de la presión sanguínea

La PA se define como la fuerza ejercida por la sangre hacia las paredes de las arterias. Cuando el corazón bombea sangre, se llama presión arterial sistólica (PAS), que tiene un valor cercano a 120 mmHg; luego, cuando el corazón permanece en reposo entre latidos, se detecta la presión arterial diastólica (PAD), cuyo valor ideal corresponde a 80 mmHg. Los valores de referencia establecidos para el nivel de PA se enumeran en la **Tabla 1-1**:

Tabla 1-1: Rango de valores de presión arterial sistólica y diastólica [3].

Grado de presión sanguínea	Rango de presión sanguínea
Normal	PAS >120 y PAD 80 mmHG
Pre hipertensión	PAS 120-139 o PAD 80-89 mmHG
Grado 1 de hipertensión	PAS 140-159 o PAD 90-99 mmHG
Grado 2 de hipertensión	PAS >= o PAD >= 100 mmHG

Según la **Tabla 1-1**, los registros por encima de 120 mmHG en PAS necesitan ser monitoreados para prevenir futuros problemas cardíacos. Este procedimiento de supervisión de las variables mencionadas está consignado en la "*Guía de atención de la hipertensión arterial*" propuesta por el Ministro de Salud de Colombia [4]. El protocolo de supervisión de PA se realiza en la mayoría de casos a través del siguiente proceso:

El usuario prepara la solicitud de una cita de medicina general; en la cita, el médico general procede a verificar los signos vitales y la medición de la PA. Este proceso es realizado para el chequeo y supervisión de la variable de interés, el cual también es mencionado por Arindam Ghosh [6], Hijazi, Page, Kantarci y Soyata [7] y Restrepo [4], cuyos autores describen algunas desventajas descritas a continuación:

- Es clara la ausencia de un seguimiento continuo y automático de las variables cardíacas del paciente.
- La fiabilidad del diagnóstico es sujeta a la experiencia del especialista.
- Imposibilidad del servicio médico para reaccionar ante cualquier evento de salud crítico del usuario.
- Un pronóstico de rutina hace imposible identificar complejidades relacionadas.

Los puntos anteriores muestran una descripción general sobre algunos de los puntos a considerar en esta investigación. Esto permitirá crear un marco de posibilidades para la preparación de un sistema para la medición de la PA que captura, almacena, procesa y permite la visualización de las variables de interés. Sin embargo, el concepto innovador que se resalta concierne a la accesibilidad e interacción de las medidas de PA y a los algoritmos de aprendizaje automático (ML) implementados para el procesamiento de los registros del paciente, los cuales permitirán visualizarán una probabilidad de riesgo si la persona puede sufrir un caso de hipertensión arterial [5].

Finalizando con esta sección, es relevante mencionar los métodos que normalmente se utilizan en los establecimientos que brindan el servicio médico para obtener los valores de PA. Los aspectos descritos en la **Tabla 2-2** deben tenerse en cuenta cuando se pretende diseñar un sistema o prototipo como lo es en este caso. Los siguientes son los aspectos tenidos en cuenta para la escogencia del tensiómetro: La precisión muestra el nivel de proximidad de la medición del sensor en comparación con el valor real, mientras que el tipo de medición muestra indirectamente qué tan cómodo es el sistema de captura.

Tabla 2-2: Métodos para tomar la presión sanguínea [3].

Método	Invasivo	Continuo	Tipo de medición	Precisión
Palpación	No	No	Indirecta	Baja
Auscultatorio	No	No	Indirecta	Media
Oscilo métrico	No	No	Indirecta	Media-Alta
Penas	No	Si	Indirecta	Media-Alta
Tonometría	No	Si	Indirecta	Media-Alta
Tiempo de Transito de Pulso	No	Si	Indirecta	Media-Alta
Sensor Intravascular	Si	Si	Directa	Alta
Sensor Extravascular	Si	Si	Directa	Alta

1.3 Problemática

En Colombia la hipertensión arterial ha liderado como una de las primeras causas de muerte, sin embargo, no se ha planteado un modelo o sistema de monitoreo, cuyo objetivo primario sea la prevención de este tipo de condición cardiaca, mediante el uso y acceso a las herramientas y tecnologías actualmente disponibles [6]. Por otra parte, el valor de abordar este problema se puede visualizar desde la importancia en que la industria del sector de la salud esté innovando constantemente con aplicaciones o sistemas de información (TIC) que prevengan el riesgo cardiovascular [2].

1.4 Propósito del Estudio

Las múltiples investigaciones esfuerzos que se han realizado alrededor del mundo en el área del monitoreo y la atención médica, se destaca el marco para monitorear pacientes con enfermedades crónicas, como el modelo basado en ontologías y la red de atención para pacientes con enfermedades crónicas [7], o la "Iniciativa de Investigación de cuidado en el Hogar" (AHRI por sus siglas en ingles) del Instituto de Tecnología de Georgia. Este último, enfocado en abordar los desafíos técnicos, de diseño y sociales fundamentales para el cuidado de las personas en el hogar.

El prototipo que se presenta en este documento, pretende trascender las tecnologías y la intercomunicabilidad entre dispositivos enfocados a la medición de variables biométricas. Este proyecto hace parte de un proceso secuencial que involucra la captura de datos y almacenamiento de los mismos en una plataforma web. Adicionalmente, mediante el diseño y creación de un aplicativo móvil, es posible la visualización cómoda e interactiva de los registros de PA del usuario. Un sistema como el descrito permite una visualización cómoda de los datos del usuario de forma asíncrona. Adicionalmente, el aplicativo provee la opción de procesar dichos registros y proporcionar una probabilidad al usuario sobre un posible episodio futuro de hipertensión arterial.

Por otra parte, desde una perspectiva técnica, un conjunto de datos de medidas continuas de PA permitiría encontrar información cardiovascular relacionada al paciente. Estos son aspectos relevantes que se tendrán en cuenta para el futuro desarrollo e interconexión de sistemas vigentes que permitan proporcionar información significativa al usuario final. Como resultados de este trabajo, se considera el diseño y construcción de algoritmos de aprendizaje automático, atendiendo las posibles situaciones de overfitting o underfitting que se puedan presentar, en la predicción de episodios de hipertensión. Adicionalmente, con registros de PA disponibles en tiempo cercano real en la web, se posibilita el envío de notificaciones o alertas a eventualidades críticas. Esto contribuye de manera coherente a un sistema de apoyo médico y autocuidado en la salud, sustentado en TICs.

1.5 Pregunta de Investigación

¿Qué aspectos se deben tener en cuenta para el diseño y realización de un prototipo de monitoreo y prevención de la presión arterial, apoyado en algoritmos de Machine Learning y tecnologías IoT?

1.6 Objetivos de la investigación

El impresionante ritmo de innovación y desarrollo evidenciado en el campo de la tecnología aplicado a múltiples áreas del conocimiento en el mundo real. Entre estos el campo de la salud, el cual se ha potenciado significativamente con el paradigma de interconexión

ubicua e Internet de las Cosas (IoT). La presente investigación pretende realizar una contribución puntual en este campo, limitada por los siguientes objetivos:

1.6.1 Objetivo general

Diseñar un prototipo de Internet de las Cosas para el monitoreo de la presión arterial, utilizando algoritmos de Machine Learning.

1.6.2 Objetivos específicos

- Agregar la funcionalidad de interconexión al diseño electrónico del tensiómetro que caracteriza la presión arterial, considerando la presión sistólica, diastólica y pulso.
- Desarrollar el aplicativo móvil que permita la visualización de los datos de medida proporcionados por el tensiómetro arterial.
- Diseñar e implementar algoritmos de Machine Learning para el procesamiento de los datos alojados en la base de datos, relacionados a la presión arterial.
- Validar el prototipo de Internet de las Cosas para monitorear la presión arterial, mediante simulación y/o mediciones reales.

2. Estado del arte de los sistemas de monitoreo de ECV sustentados en IoT y tecnologías asociadas

La conclusión de múltiples artículos e investigaciones muestran la relevancia de las posibilidades de uso disponibles en dispositivos IoT en escenarios de captura de datos e interconexión entre los mismos. Esta importancia trasciende y es evidente en el sinnúmero de aplicaciones en los múltiples campos del conocimiento y en la cantidad de artefactos que crece día tras día. En los siguientes apartados se expone la caracterización de la PA, los modelos y sistemas de monitoreo de ECV basados en IoT, conjuntos de datos y atributos disponibles para el entrenamiento de algoritmos y técnicas de análisis esos datos.

2.1 Telecardiología y el monitoreo a distancia

Un concepto de gran interés expuesto en la investigación de Unnikrishnan et al. [8] es telecardiología. Dicho termino está relacionado con el cuidado y monitoreo de la salud cardíaca a distancia, donde se combina la intercomunicación entre los dispositivos de adquisición de datos y el servidor que recolecta las medidas cardíacas. En este orden de ideas, una investigación relacionada con el monitoreo de enfermedades cardíacas y la supervisión a larga distancia [9] propone un modelo de monitoreo inalámbrico disponible en complicaciones específicas en el área de la cardiología. Esta investigación menciona factores que van desde evitar una re-hospitalización innecesaria y una atención médica más personalizada, hasta una comunicación efectiva entre dispositivos y la gestión apropiada de datos desde la plataforma.

Adicional a la adquisición de datos crudos y la gestión de los mismos, es importante enfatizar en los atributos que contribuyan a la detección temprana de una ECV y como se administran los servicios de mensajería y de alertas. Hsu, Hsieh, Li y Yang [10] diseñaron

un sistema de recopilación de datos de frecuencia cardíaca basado en el Equivalente Metabólico de Esfuerzo (MET). El prototipo envía datos por medio de un controlador electrónico a un teléfono móvil, luego se alojan dichos datos en un servidor en la nube. Este proceso se lleva a cabo gracias al protocolo de mensajes MQTT disponible en los estándares de comunicación IoMT. De esta manera, los usuarios reciben información o mensajes de advertencia, según sea el caso, sobre los esfuerzos físicos que implica el proceso de recuperación cardíaca postoperatoria del paciente.

Por otro lado, la evolución de los dispositivos que imponen una vanguardia tecnológica en el campo de la salud, en términos de generación de información a partir de bases de datos, se expone en un estudio de caso realizado en la Universidad de Rochester [11] el cual tiene como objetivo profundizar en la información generada por estos dispositivos para uso médico, de modo que sea posible la distinción entre variables no relevantes o información de menor importancia. En este orden de ideas, un conjunto de datos de medidas continuas de la PA podría permitir encontrar información cardiovascular relacionada del paciente. Estos son aspectos fundamentales para el desarrollo de un dispositivo que proporcione información significativa sobre los aspectos en prevención a tenerse en cuenta en su condición de salud.

En la siguiente sección se esbozarán algunos de las bases de datos utilizadas para el entrenamiento y análisis de los algoritmos utilizados en el procesamiento de las medidas de PA. También se enfatizará en parámetros específicos como lo es el número de atributos, las entradas seleccionadas y la calidad de los conjuntos de datos.

2.2 Bases de datos utilizadas en desarrollos de ECV

Actualmente, el desarrollo e implementación de algoritmos matemáticos en diferentes áreas del conocimiento ha tenido un despliegue exitoso. Esto debido a las posibilidades de descubrimiento de información, clasificación de muestras o inclusive relaciones entre datos, solo por mencionar algunos usos de los algoritmos computacionales. Específicamente, para tareas de clasificación, se referencia el trabajo de Acharya [12], donde se especifica en la **Tabla 2-3** algunos de los atributos de conjuntos de datos disponibles para la clasificación y predicción de ECV, el cual está disponible en el

repositorio de aprendizaje automático UCI¹. En el proyecto, se realiza una comparación entre diferentes algoritmos de clasificación, donde se relacionan las condiciones que contribuyen a sufrir una ECV.

Los resultados destacan la efectividad de las Redes Neuronales Artificiales (ANN), el Clasificador Bayesiano (NB) y la Máquina de Vector de Soporte (SVM). Dichos resultados se evaluarán para su implementación en el prototipo, según las especificaciones y requerimientos de los mismo y el volumen de datos a trabajar.

Tabla 2-3: Base de datos utilizadas en técnicas de aprendizaje automático [12].

ATRIBUTOS	DESCRIPCIÓN DEL CONJUNTO DE DATOS
Edad	Edad en años
Sexo	Instancia de género (0=femenino, 1=masculino)
Trestbps, PSA	Descanso de la presión sanguínea en mmHG
Chol	Suero de colesterol en mg/dl
Restecg	Resultados de ECG (0: normal, st-t anormalidad en onda, 2:lv hipertrofia)
Thalach, HR	Alcance máximo del ritmo cardíaco
PDA	Presión Diastólica Arterial
Num	<i>Diagnóstico de enfermedad del corazón (0: saludable, 1:no saludable)</i>

Para los propósitos de la presente investigación, en la **Tabla 2-3** se describen los siguientes atributos que serán capturados mediante el uso del monitor de PA:

- TRESTBPS
- THALACH
- DBP

Los anteriores, se refieren a la presión arterial sistólica, la frecuencia cardíaca (FC) y la presión arterial diastólica respectivamente. Sin embargo, en el diseño realizado para el prototipo el conjunto de datos se complementa con los atributos AGE, SEX, TIMESTAMP y la variable objetivo NUM. En cuanto al análisis y manejo de estos datos, se utilizará la

¹ Center for Machine Learning and Intelligent Systems

herramienta Tensorflow, debido a su practicidad en la construcción de modelos de regresión lineal y ANN, además de las facilidades de accesibilidad que proporciona desde Firebase, gestor de base de datos en tiempo real Firebase. Esta sección de diseño y desarrollo se expone en detalle en la Metodología.

2.3 Técnicas de ML orientadas a la predicción de anomalías cardíacas

Con respecto a la detección temprana de problemas cardíacos mediante el uso de técnicas computacionales, Wang, Xia, Li, An y Chen [13], muestran cómo se utilizan algoritmos de ANN como técnica principal, basada en una regresión logística. Los resultados del estudio demuestran que el procesamiento de variables estáticas por ANN (ejercicio, peso, edad, sexo, entre otros) resulta en un índice de alta precisión para pronosticar, si la persona puede sufrir un episodio futuro de hipertensión. Según los resultados, alrededor del 72% de asertividad está asociada con capas neuronales ocultas. Majed, Al-Mutairi, Sangaiah y Samuel encontraron resultados similares, demostrando cómo SVM, se implementó para el diseño de un modelo fundamentado en una base de datos de pacientes con riesgo cardíaco potencial. Los resultados en este diseño ofrecen una precisión de predicción del 87,9%.

Por otra parte, en el contexto computacional, existen ecuaciones para determinar cierto nivel de riesgo de sufrir alguna ECV. Un estudio publicado por Unnikrishnan et al. [14] describe el nivel de sensibilidad y especificidad de la ecuación de Framingham, aludiendo a las variables demográficas de una población dada. Los resultados predictivos del estudio muestran el riesgo de contraer alguna ECV durante los siguientes diez años. Si bien, este tipo de análisis matemático aborda múltiples circunstancias externas que una persona pueda tener en relación a su salud cardiovascular, el análisis de las variables del prototipo presentado se limita a las mediciones obtenidas por el tensiómetro arterial. No obstante, este tipo de abordaje en los algoritmos de predicción proporcionan un alto índice de eficiencia en los algoritmos de aprendizaje automático orientados a la predicción de estados sintomatológicos.

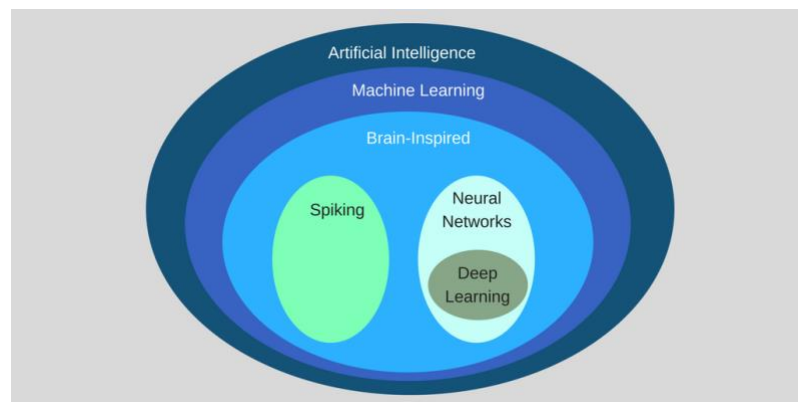
2.3.1 Aprendizaje Automático

Múltiples vistas para la interpretación de información mediante el procesamiento computacional se han venido desarrollando en la presente década. No obstante, la proliferación de dispositivos electrónicos, sensores, sistemas embebidos, entre otros, han generado la necesidad de implementar e innovar en las técnicas y algoritmos que permiten el análisis de múltiples fuentes de información con el fin de descubrir relaciones entre esos datos y las variables objetivo seleccionadas. En la presente sección se pretende esbozar los conceptos y técnicas relevantes relacionados al aprendizaje automático, los cuales servirán como soporte conceptual al diseño y requerimientos planteados en esta investigación.

Inteligencia artificial y Aprendizaje automático

En relación con el tema de predicciones en el campo de la salud, diferentes establecimientos médicos e industrias del campo, almacenan grandes cantidades de registros de pacientes e información relacionada, que en la mayoría de casos es de difícil procesamiento o simplemente no se cuenta con las herramientas adecuadas. En este punto, el *Aprendizaje de Automático* o *Machine Learning* (ML), ha jugado un papel clave en el descubrimiento de información, clasificación y relaciones entre datos, a partir del análisis de múltiples DB y los registros de pacientes. Sin embargo, es importante identificar y delimitar el marco conceptual en el cual se identifican los algoritmos utilizados en la clasificación o predicción de información. La siguiente figura esboza la jerarquía conceptual detrás del ML.

Figura 2-2: Esquema conceptual para del Aprendizaje Automático [15].



De manera resumida, en la **Figura 2-2: Esquema conceptual para del Aprendizaje Automático** [15]. se muestra la Inteligencia Artificial (AI) *como el campo de ingeniería y de la ciencia que posibilita a las maquinas realizar acciones como los humanos*, definición dada por John McCarthy, autor ampliamente reconocido en el campo. No obstante, *Machine Learning* es un sub-campo de estudio de la AI, relacionado con el aprendizaje de las máquinas, las cuales no están explícitamente programadas para ese objetivo. De ese modo se puede hacer una analogía sobre cómo se produce el aprendizaje de los humanos.

Siguiendo dentro de esta categorización de campos, se resaltan el funcionamiento en una de las grandes máquinas para la solución de problemas, el cerebro. Estas múltiples conexiones internas de neuronas, las cuales proveen decisiones basadas en información recepcionada, han sido replicadas computacionalmente, a lo cual se le atribuye el nombre de *Redes neuronales Artificiales (ANN)*. En más detalle, dentro del campo de ML y las ANN se encuentra la técnica de *Aprendizaje Profundo* o *Deep Learning*, la cual se diferencia específicamente por el uso de algoritmos de ANN de múltiples capas. A continuación, se describirán los campos de la AI mencionados previamente.

2.3.2 Algoritmos de ML

Dentro del capitulo de estudio del *Aprendizaje Profundo*, se suele referir a un concepto denominado *Minado de Datos*, cuyo proceso se refiere a la extracción automática de información proveniente de grandes cantidades de datos [16]. Dentro de las posibles técnicas utilizadas en ML, existen diferentes enfoques que determinan el funcionamiento y modo de operación en las tareas de clasificación, aprendizaje o predicción. Seguidamente se mencionarán los principales algoritmos utilizados en este campo de estudio.

Arboles de decisión (J48)

Esta técnica utiliza el concepto de entropía de la información para la construcción de árboles condicionados a los atributos de una DB [17]. De este modo, se separan las muestras de la DB dentro de subcategorías o subclases. Para lograr esta discriminación se suele seleccionar un atributo para la probabilidad de ocurrencia de valores en una clase particular. También existe un método para lograr esta clasificación, denominado índice Gini, el cual expresa la desigualdad entre los valores de una distribución.

Naïve Bayes

Esta técnica se basa en el teorema de Bayes, donde se enfatiza en la independencia entre los atributos de entrada, la cual se diseña a partir de una combinación de clasificación estadística y aprendizaje supervisado [17]. En tareas de clasificación, el objetivo es predecir correctamente el valor de una variable de clase dado un vector de predictores o atributos. En particular, el clasificador Naïve Bayes es una red bayesiana donde la clase no tiene padres y cada atributo tiene a la clase como su único padre [16].

Support Vector Machine

Este enfoque se refiere a la clasificación binaria, dada por la obtención de un hiperplano entre dos clases obtenido por la maximización de los vectores de soporte. SVM tiene la ventaja de que no se requiere relaciones lineales o independencia entre los atributos de entrada y, por lo tanto, es más adecuado para la clasificación de datos clínicos [14].

K-Nearest Neighbour

Este clasificador se considera un algoritmo de aprendizaje estadístico y es extremadamente simple de implementar. En su funcionamiento, cuando se requiere una predicción sobre un punto desconocido, el clasificador encuentra el punto de entrenamiento más cercano al punto en cuestión. Posteriormente, señala y predice la categoría de ese punto de acuerdo con alguna métrica de distancia. Esta métrica para atributos numéricos puede ser la Distancia Simple Euclidiana [16].

Multilayer Neural Network

Las ANN son métodos auto adaptativos, que ajustarse a los datos sin una previa especificación sobre la distribución de la DB. Por otra parte, esta técnica se caracteriza por su flexibilidad y efectividad para modelar las complejas relaciones entre entradas y salidas de un sistema real [13].

Logistic Regression Model

Este tipo de regresión estadística, provee la relación entre una variable dependiente u objetivo, y un grupo de variables no dependientes. Según el número de valores de la variable dependiente, el modelo de regresión logística se clasifica en análisis de regresión

binomial y multinomial [13]. Para el caso de binomial, la variable objetivo se suele etiquetar con un "1" o "0", (enfermo, o no enfermo).

2.4 Ecosistema IoT

El vasto panorama de posibilidades emergente de posibilitar el procesamiento y la interconexión entre diferentes dispositivos de uso cotidiano, han configurado un paradigma sobre el impacto que este tema tendrá en la sociedad, incluyendo las esferas políticas, económicas, jurídicas y reglamentarias. Si bien, IoT promete un mundo conectado, es importante resaltar los múltiples desafíos técnicos, relacionados a la seguridad y privacidad de los datos, e incluso, los hábitos sociales derivados de la dependencia de dispositivos tecnológicos. Este capítulo promete mostrar los aspectos relevantes cuando se referencia la IoT, abordando las tecnologías que la sustentan, hasta los cambios culturales en razón del uso de dispositivos y Tecnologías de la Información y Comunicación.

2.4.1 Economías emergentes

La computación ubicua, ha traído consigo la implementación de poderosos dispositivos de procesamiento que generan enormes cantidades de información. Estos datos a su vez, pueden ser procesados y de ese modo obtener patrones de similitud, que permitan realizar tareas con mayor nivel de complejidad [18]. La realización de las tareas involucradas en el anterior proceso, convergen en las industrias del diseño de software, dispositivos de almacenamiento local y remoto, sin mencionar la construcción de los gadgets y sus periféricos. La convergencia de las tecnologías vanguardistas en las diversas áreas del saber, aumentan de manera exponencial la necesidad de apropiación y entendimiento de la IoT en el contexto contemporáneo.

Frente a la realidad de la computación pervasiva, resulta trascendente entender el propósito de cada proceso de innovación y el área de conocimiento o mercado a la que pretende acceder. De ese modo, se puede concebir un plan de contextualización y apropiación tecnológica, del cual destaque el concepto de productividad sobre el de consumo. En esta transición, es factible encontrar barreras culturales, técnicas e incluso económicas. No obstante, los factores relacionados a la eficiencia, precisión y versatilidad,

en la mayoría de las situaciones terminan posicionándose como los pilares en la innovación de los procesos industriales.

Una economía emergente basada en las tecnologías de computo contemporáneas, exige un entendimiento que va mas allá del manejo de dispositivos electrónicos. Este proceso encapsula la apropiación del contenido y el aprovechamiento e implementación de esos saberes en mercados que aun se desarrollan, en ideas sustentadas en la visión de Cosa, Internet y Semántica en IoT. La visión orientada a la Cosa, se abstrae en la característica de proveer comunicación entre los dispositivos electrónicos, cuya técnica se extiende al uso de sensores y sistemas embebidos [18]. En la visión de internet, impera el uso de una IP, y así posibilitar la gestión del objeto. Finalmente, en la visión de semántica prima el entendimiento de los datos generados por las Cosas.

2.4.2 Retos en IoT

Se han hablado sobre las múltiples bondades y posibilidades de un mundo interconectado, sin embargo, vale la pena resaltar aquellos factores que hoy en día presentan ciertas barreras en la implementación y despliegue del potencial en la IoT. La automatización y computación ubicuidad supone un consumo constante de recursos los cuales posibiliten la ejecución de tareas. En sistemas embebidos o sensores, en los cuales se evidencia un flujo constante de información, los requerimientos sobre abastecimiento de energía y consumo de recursos deben estar diseñados con base en los factores clave de portabilidad, eficiencia y conectividad. La miniaturización de sensores, placas de procesamiento y demás periféricos de los dispositivos inteligentes, que se ha evidenciado en las ultimas dos décadas, han perforado la barrera mencionada. Aunque, la creciente demanda de servicios y dispositivos *todo en uno*, prevalecerá simultáneamente con las innovaciones orientadas a portabilidad y autonomía.

Por otra parte, el flujo de tramas e información por la web aumenta de manera significativa debido a la inclusión de millones de dispositivos en ese mundo virtual. No solo se pueden encontrar dificultades en los dispositivos, si no en la infraestructura que los soporta. Algunos aspectos relevantes a señalar, responden a la necesidad de estandarizar protocolos y medios de comunicación, de este modo se aseguran altos índices de compatibilidad en los productos y servicios de diferentes compañías. Esto a su vez

repercute en la disponibilidad del servicio y escalabilidad en ciertos segmentos del mercado. En relación a la creciente demanda de un sistema escalable, sobresalen las diversas brechas en cuanto a seguridad de la información. A continuación, se describen las generalidades a considerar relacionadas a la confiabilidad en un ambiente interconectado.

2.4.3 Privacidad y seguridad de la información

La seguridad y confiabilidad de la información proporcionada por una red de dispositivos interconectados, se puede analizar desde 3 perspectivas: Legal o jurídica, social y desde el punto de vista cultural [19]. Desde el primer punto de vista, las responsabilidades legales varían de acuerdo a las políticas socio-culturales de una región, no obstante, en la mayoría de casos esta normatividad se orienta hacia un enfoque de responsabilidades y derechos del consumidor para que abarquen las externalidades negativas relacionadas con la Internet de las Cosas [20].

Ampliando el panorama a la perspectiva social, otro problema se relaciona con la normatividad interna de las compañías que retienen grandes bases de datos. La destrucción de esta información resulta vital en la disminución en los fallos de privacidad, sin embargo, el constante aumento de los dispositivos inteligentes en el entorno, somete al usuario a un proceso de concientización y reflexión sobre los datos que esta suministrando a la red. Esta revolución de la industria ha esquematizado nuevos patrones de comportamiento en la sociedad, para bien o para mal, un aspecto a prevalecer se asocia a la apropiación tecnológica en beneficio del bien común.

2.5 Contribuciones en el campo de la salud de los sistemas sustentados en IoT.

Como se menciona en la anterior sección, las posibilidades de mejorar la eficiencia en la predicción de diferentes condiciones cardíacas aumentan considerablemente con el uso de técnicas de aprendizaje de máquina. Por lo tanto, es esencial conocer los parámetros y variables objetivo, de modo que sea viable decidir cuál es el algoritmo que mejor se adapta, según los requerimientos del proyecto. Sin embargo, es importante describir las herramientas y procesos tecnológicos utilizadas en diversas investigaciones que dan

cuenta de las posibilidades de acceso e interconexión entre plataformas. A continuación, se describen los sistemas y prototipos en los cuales se ha desarrollado algún tipo de innovación que mejora las prestaciones en cuanto a la predicción y el monitoreo de ECV.

2.5.1 Convergencia tecnológica

Uno de los puntos que se deben tener en mente se relaciona a los desafíos fundamentados en el tratamiento de datos y transmisión de los mismos. Tomando el caso de [21], se muestra un problema relacionado con el proceso de adquisición de datos en dispositivos portátiles: los desafíos que enfrentan los equipos de electrocardiografía en la medición; en dicho artículo, se concluye que los dispositivos interconectados basados en IoT mejoran significativamente los aspectos asociados con la portabilidad, el rendimiento, monitoreo de datos y la asequibilidad. El marco general de ese estudio destaca las múltiples innovaciones que sean realizado para el modelo de capas y arquitectura de comunicación que deben manejar estos dispositivos médicos (percepción, red y aplicación).

Una de las aplicaciones de la arquitectura mencionada, la expone Ghosh en [22] donde se desarrollan un sistema de pronóstico y monitoreo de hipertensión, en el cual se utiliza una manilla electrónica (Wristband Empatica) y una plataforma de procesamiento (HEAL). La investigación expone los beneficios de la cohesión entre la practicidad de obtener datos con sensores electrónicos y el procesamiento inteligente de una plataforma en la nube. Continuando con esa misma línea, otro estudio relacionado con el seguimiento y la supervisión de enfermedades cardíacas a grandes distancias [23], propone un modelo de monitoreo inalámbrico aplicable a complicaciones puntuales del área de cardiología. Esta investigación menciona factores que van desde evitar la re-hospitalización innecesaria y un cuidado más personalizado de la salud, hasta una comunicación efectiva entre dispositivos y manejo de datos en plataforma.

De las anteriores investigaciones se destaca el concepto de “Telecardiología”, la cual se ha referido previamente como el monitoreo o diagnóstico de condiciones cardíacas mediante el uso de tecnologías de la comunicación [24]. No obstante, en ese contexto existen retos relacionados a las señales inalámbricas utilizadas por estos dispositivos que capturan los datos de interés, cobertura de señales para la transmisión de información y limitaciones de energía, aspectos que se abordarán en la siguiente sección.

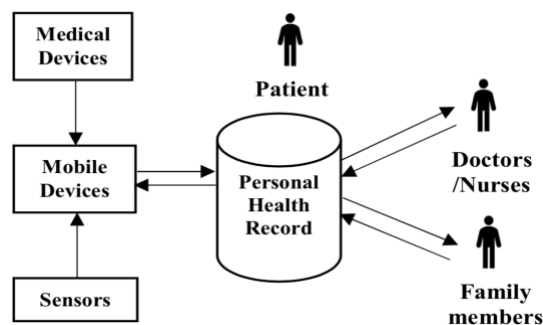
2.5.2 Limitaciones de los dispositivos de medición

Un reto constante que se pueden encontrar en la implementación de dispositivos orientados al monitoreo y adquisición de señales biométricas, se suele relacionar con las limitaciones físicas de estos. Sean esos problemas relacionados con interferencia en la señal de captura, disponibilidad de medios para el envío de información, en inclusive autonomía del dispositivo, por mencionar algunas. Es importante destacar que la viabilidad de un sistema o prototipo orientado al cuidado de la salud humana, debe destacar en su alta disponibilidad y disminuir en lo posible las variables o aspectos que puedan perjudicar a los usuarios involucrados.

En [21] se expone la comparativa entre los tradicionales sistemas de medición de electrocardiogramas y Los dispositivos contemporáneos. En la investigación de Khairuddin, se menciona las limitaciones de comodidad que posee un holter. Este dispositivo funciona mediante la colocación de un revestimiento de gel y electrodos que transmiten las señales del corazón. En ocasiones este gel produce irritamiento en la piel, así como degradación en la calidad de la señal. Otra característica de este dispositivo obliga al paciente a realizar los procedimientos necesarios para el almacenamiento y captura de señales eléctricas necesarios en la onda del ECG.

Adicional a los problemas referenciados, en el desarrollo del prototipo presentado se evidencia que en numerosas ocasiones esos dispositivos de captura de datos no poseen conectividad inalámbrica para el envío de los mismos, por lo que se imposibilita un tratamiento externo de dicha información. Por otra parte, la visualización histórica de las muestras es incómoda o inexistente.

Figura 2-3: Esquema IoT en un modelo de monitoreo de la salud [21].



Entonces, solventando los anteriores inconvenientes, se posibilita la intercomunicación entre los dispositivos electrónicos contemporáneos lo cual genera un marco de posibilidades concernientes a la miniaturización de los dispositivos y la descentralización de la información. En la **Figura 2-3: Esquema IoT en un modelo de monitoreo de la salud** [21]. se esquematiza una vista general del panorama IoT en el cuidado la salud. Las anteriores y otras razones hicieron parte de la motivación para la culminación del prototipo de medición de la PA, cuyo diseño y desarrollo se expone a continuación.

3. Metodología de diseño del prototipo

Teniendo en cuenta las capas del modelo general de arquitectura IoT, el prototipo presentado en ese documento, se diseña desde tres perspectivas generales: *Aplicación*, *Captura de Datos* y *Red*. En referencia a la capa de *Aplicación* e Interfaz gráfica, se ha diseñado y construido un aplicativo móvil, intuitivo y cómodo, programado en *Android Studio*. Este tipo de entorno de desarrollo posibilita la utilización de las librerías y herramientas de JAVA proveyendo un ambiente idóneo en el diseño y test de las versiones del App.

Correspondiente a la capa de *Captura de Datos*, se ha contemplado modificar un monitor de presión arterial el cuál sea capaz de enviar datos a través del protocolo Wifi, gracias a la interconexión del módulo ESP8266. Por último, en la capa de *Red*, se ha utilizado Firebase como DB para el almacenamiento de las medidas de PA. Adicionalmente en este segmento se relaciona la plataforma de Tensor Flow, utilizada para el análisis de los datos capturados.

La **Figura 2-3:** Esquema IoT en un modelo de monitoreo de la salud [21]. muestra un esquema general las operaciones de medición de las variables de interés, transmisión y almacenamiento de esa información, y la generación de una predicción porcentual sobre un posible padecimiento de un episodio de hipertensión arterial, el cual está dirigido al usuario mediante el aplicativo móvil y destinado a proveer la información necesaria para mejorar los hábitos saludables y control de la PA. En las siguientes secciones se expone la metodología utilizada para el cumplimiento de los objetivos planteados en esta investigación.

3.1 Métodos para responder las preguntas de investigación

La metodología implementada para dar respuesta a la principal pregunta en esta investigación, se fundamentó en la contextualización académica y en las técnicas o procesos tecnológicos utilizados en investigaciones relacionadas, en los cuales se evalúa la practicidad y eficiencia de los modelos propuestos alicientes a la utilización de modelos matemáticos u algoritmos de Machine Learning, destacando los resultados logrados.

Lo anterior fue posible mediante la indagación y profundización en el paradigma IoT y su convergencia del cuidado de la salud, término que se acuña en el acrónimo IoMT (Internet de las cosas Médicas). Este enfoque es trascendental en los sistemas contemporáneos de información e intercomunicación de dispositivos. La base conceptual de la terminología en mención, así como las técnicas de la investigación, exploración y abstracción, se explican siguientemente.

3.1.1 Enfoque metodológico

En este punto del diseño metodológico de la investigación, se ha planteado realizar una vigilancia tecnológica de los sistemas vigentes e investigaciones realizadas concernientes al monitoreo de la hipertensión arterial. En primera instancia se realiza una búsqueda sistemática de información en los principales motores de referencias bibliográficas, los cuales permitan soportar los objetivos y los requerimientos del diseño del prototipo.

Con básicas pero determinantes preguntas orientadoras fundamentadas en la problemática abordada, se busca conformar un marco de conceptos relacionado a los sistemas vigentes de monitoreo en salud, las técnicas de procesamiento de información aplicadas en bases de datos y las técnicas de captura de variables biométricas, entre otros temas. Esta primera muestra información permite visualizar las tecnologías, conceptos y sistemas basados en IoT, de los cuales se pueden tomar elementos para el proceso de diseño y desarrollo del prototipo planteado [25].

Por otra parte, es importante resaltar para quien se está realizando esta investigación. La finalidad que se pretende cumplir, responde a la necesidad de vincular las herramientas y

procesos disponibles en TIC, que puedan trascender y contribuir a disminuir los números de morbilidad relacionados a la hipertensión arterial. Este desarrollo se orienta al autocuidado de la PA, mediante el uso de los dispositivos y tecnologías disponibles en el ecosistema digital contemporáneo. Desde este modo se contribuye a una interrelación multidisciplinaria, que recoge las bondades de la tecnología y las dispone en pro del bienestar humano.

Concluyendo con esta sección, es relevante mencionar los métodos que validen el cumplimiento de los objetivos de esta investigación, así como la eficiencia, funcionamiento del prototipo presentado. En cuando los objetivos técnicos relacionados a la creación del aplicativo, modificaciones físicas del tensiómetro arterial y el diseño de los algoritmos de ML; estos se pueden evaluar en cuanto a apariencia, funcionalidad y porcentajes de precisión. Sin embargo, es importante atender el concepto del profesional médico el cual pueda identificar y contribuir en factores clave para la mejora del prototipo. Esa triangulación de enfoques, base conceptual literaria, conceptos y experiencia del personal médico, y el levantamiento de información en esta investigación, se exponen en detalle las posteriores secciones.

3.1.2 Estrategia de Investigación

Mencionado previamente, para acuñar los conceptos necesarios que se trabajarán a lo largo esta exploración, es fundamental realizar una revisión bibliográfica en donde resalten las principales investigaciones relacionadas a la aplicabilidad de las tecnologías habilitantes en IoT concernientes al monitoreo o pronóstico de condiciones de salud en personas, especialmente las alienantes a la presión arterial. Luego, se categorizan y priorizan los elementos que puedan contribuir en un contexto real a la investigación. Con lo anterior, se pretende enfatizar en las necesidades de comodidad, gestión de la información y propósitos de prevención que debe sustentar el prototipo.

Con un primer diseño del aplicativo y la posibilidad de enviar las mediciones de PA a la base de datos en tiempo real de Firebase, el siguiente paso corresponde al entrenamiento de los algoritmos que procesarán los registros históricos del usuario. Posteriormente, resulta imprescindible evaluar la usabilidad del prototipo. Para lograr lo anterior, se realizaron múltiples indagaciones sobre los aspectos a tener en cuenta desde el punto de

vista médico, para ello se conto con la colaboración de la doctora Marta Lucia Lancheros especialista en psicología medica y de la salud, actualmente labora en área de medicina general del hospital de la Universidad Nacional de Colombia.

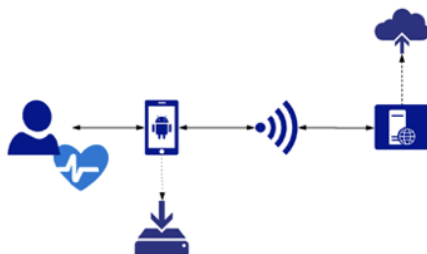
Gracias a estas recomendaciones y puntos de vista, es posible evaluar el funcionamiento del prototipo y en su debido caso, mejorar aspectos o algunos casos de uso detallados, tales como inicio de sesión, despliegue del menú principal o incluso la visualización de los registros históricos en Firebase. Adicionalmente, teniendo presente ese concepto médico, se consolida y soporta la información acá expuesta, posibilitando la continuidad de esta investigación y diversos trabajos relacionados. En seguida, se expone en detalle el proceso realizado en cada una de las fases del prototipo.

3.2 Diseño del prototipo

Concerniente a las modificaciones y tareas realizadas entorno al monitoreo de la PA, se destacan en el prototipo las características de portabilidad y el procesamiento inteligente de los datos recopilados, donde se puede pronosticar e informar cierta información del usuario mediante una aplicación. Para lograr esta funcionalidad, en la presente sección se describen en detalle las modificaciones realizadas al tensiómetro, las cuales hacer referencia a la capa de *Captura de datos*; luego se detallan los elementos considerados en la capa de *Aplicación* que permitan una visualización cómoda desde dispositivos móviles.

A continuación, se especifican los aspectos involucrados en cada una de las capas mencionadas, las cuales a su vez se muestran en la **Figura 3-4** en un esquema ordinario referente a las tareas involucradas en el proceso de monitoreo.

Figura 3-4: Esquema del prototipo IoT para el monitoreo de la PA, (Fuente autor).



3.2.1 Adhesión de características en Hardware

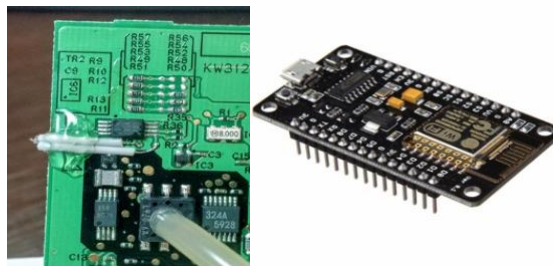
El rediseño del tensiómetro comienza desde la capa de percepción. Se utiliza un monitor de presión arterial (Panasonic EW3106, ver **Figura 3-5**). Este monitor puede mostrar datos de frecuencia cardíaca, presión sistólica y diastólica. Además, cuenta con una memoria de almacenamiento EEPROM con conectividad I2C, en la cual se almacenan las variables de PA del usuario que posteriormente serán capturadas con la ESP8266 (ver **Figura 3-6**).

Figura 3-5: Tensiómetro arterial seleccionado, (Panasonic EW3106) [26].



Aunque el dispositivo o monitor de PA es portátil, este carece de algún tipo de interconectividad para transferir los datos del usuario, lo cual se requiere para el procesamiento de los registros. Por lo tanto, es prioritario agregar un componente de conexión inalámbrica el cual posibilite dicha exportación de datos. La opción se oriento al uso del módulo de interconexión Wi-Fi ESP8266, cuyas características y utilización se describen en [27]. La **Figura 3-6** exhibe de modo general la placa interna del tensiómetro y las conexiones realizadas para escuchar los datos de la memoria EEPROM. Además, se presenta la placa que captura los datos de PA y realiza el envío de los mismos a la DB en Firebase, a través del protocolo WIFI.

Figura 3-6: Módulo ESP8266 y placa interna del tensiómetro, (Fuente autor).



Para la ejecución de estas modificaciones, se requirió el acceso al dispositivo desde la placa electrónica principal para capturar la trama de datos proveniente de la memoria EEPROM, la cual almacena cada una de las medidas del usuario. Esta captura de información es posible cuando el dispositivo realiza un envío de datos hacia la memoria. Acto seguido, a través de los pines físicos del módulo ESP8266, es posible capturar la señal de información. Luego, esta trama se procesa mediante un análisis del protocolo de conexión serial el cual se describe en profundidad en [28]. Después de analizar los valores hexadecimales enviados a la memoria EEPROM (**Figura 3-6**) el módulo obtiene los valores de interés HR, SBP y DBP. El código y la captura de datos, procesos mencionados previamente, realizados por la ESP8266, se encuentra en detalle en el Anexo B de este documento.

Figura 3-7: Bytes en formato hexadecimal de las mediciones de la PA [28].

START	ADDRESS	DATA	STOP
	WRITE TO: 0x50	0x1C 0x62 0x53 0x80 0x44	

En la **Figura 3-7** Se pueden apreciar los bytes capturados y la condición de inicio. El primer byte (x1C) es la dirección en la memoria donde el microcontrolador (MCU) escribe los datos. El segundo byte (x62) es el pulso, el cual traduce 98 HR en decimal. El tercer byte (x53) es la SBP: 83 mmHg. El cuarto y quinto bytes juntos forman la DBP. El esquema utilizado fue almacenar la mitad de la presión sistólica en el quinto byte y usar el cuarto byte como indicador si el número es impar o par. El cuarto byte (x80) significa que la bandera está activada, 0 significa que no lo está. Entonces, en el este caso, (x44) se traduce en 68 decimales. Multiplicando el quinto byte por dos se obtiene 136. Dado que la bandera está configurada, se agrega 1, por lo que el número final es 137 mmHg [28].

Después de obtenidos los valores de Ritmo Cardíaco (HR), Presión Sistólica y Diastólica (SBP y DBP, respectivamente), el módulo procede al envío de dichos valores a Firebase, mediante sentencias Push las cuales generaran un Hash o llave aleatoria para cada grupo de medidas. A continuación, se describe un fragmento del código utilizado en la interfaz de Arduino, la cual habilita el envío de las peticiones HTTP para el envío de datos.

```
root["diastolic"] = diastolic; (3.1)
```

```
root["pulse"] = hr; (3.2)
```

```
root["systolic"] = systolic; (3.3)
```

```
root.printTo(jsonStr); (3.4)
```

```
Firebase.pushJSON(firebaseData,path,jsonStr) (3.5)
```

(Fuente autor)

En las líneas (3.1) a (3.3) se crean los espacios o ramificaciones del archivo JSON que contendrá los valores capturados de la PA del usuario. Posteriormente en (3.4) se crea el archivo JSON y en (3.5) se procede a realizar una petición HTTP que envía los datos a Firebase. Previamente se deben configurar los permisos de escritura y la autenticación para escribir en la DB.

Figura 3-8: Ejemplo de un envío de la PA a la DB en tiempo real de Firebase, (Fuente autor).



En la **Figura 3-8** se aprecia el registro de las variables de media consignadas en la base datos de tiempo real, de la cual se describirá detalladamente en la siguiente sección. Es importante destacar las tres variables relacionadas a la PA y una cuarta que garantiza un registro en línea tiempo (timestamp).

3.2.2 Diseño de la base de datos en tiempo real

Con un tensiómetro que puede exportar información a un servidor web, la selección de una base de datos capaz de almacenar en orden temporal las mediciones de la PA y de fácil integración con Android Studio y el IDE de Arduino, son las razones para emplear Firebase. Adicionalmente, la utilización de archivos JSON, los cuales son ligeros y fáciles de construir

para el módulo (ESP8266), brindan un índice confiable de escritura en la DB, además de soporte en la escalabilidad para la adquisición de datos proveniente de múltiples dispositivos. Por otra parte, el diseño de una aplicación móvil utilizando el API de Firebase, facilita las tareas de interconexión entre plataformas y protocolos de comunicación [23]. En cuanto a la organización y el almacenamiento, para cada grupo de mediciones se obtendrá una llave aleatoria.

Por otra parte, es importante referirse al tratamiento de la información consignada en Firebase y el marco legal bajo el cual se rigen en Colombia. La ley 1266 del 2008 regula el manejo de la información contenida en bases de datos personales, incluidos los servicios de salud y especifica que los datos personales no pueden ser transmitidos ni compartidos sin consentimiento del cliente [29]. Firebase posee un sistema de autenticación mediante la creación de una cuenta personal de Google, la creación de reglas sobre el control de escritura y lectura de los datos almacenados y la opción de creación de backups para evitar pérdidas en los registros. Estas características de la DB, proveen una opción idónea para los requerimientos de la investigación planteada.

En otro contexto, resulta interesante visualizar las posibilidades de trabajo con las opciones de inteligencia artificial y análisis de datos que ofrece la plataforma de Firebase. Como ejemplo de lo anterior, en la **Figura 3-9** se aprecia la interacción de los usuarios con la DB, es decir la frecuencia con que los métodos del App escriben los registros de PA en la base de datos de tiempo real. Esta posibilidad análisis llega a hacer propicia para el control y gestión de múltiples dispositivos con la opción de registro de información sensible.

Para este caso en donde el sistema puede ser escalable, Google-Firebase exige según el Reglamento General de Protección de Datos (RGPD), que los programadores que recojan o traten datos de usuarios a gran escala o determinados tipos de datos sensibles, así como aquellos que constituyan una autoridad u organismo público, deben designar a un delegado de protección de datos.

Figura 3-9: Interacción de los usuarios en Firebase, (Fuente autor).

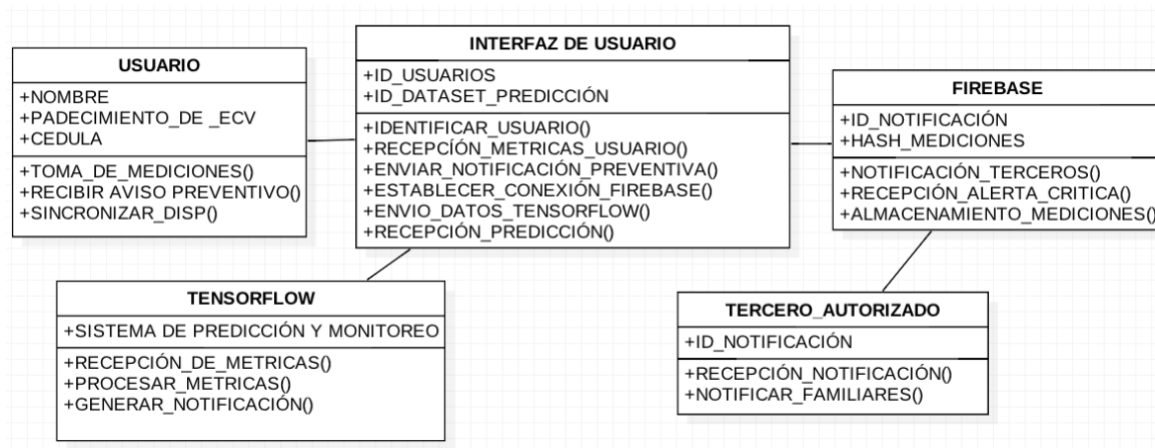
3.2.3 Diseño del App en Android Studio

En cuanto al desarrollo realizado en la capa de aplicación e interfaz de usuario, se propone una App mediante la cual el usuario dispone de algunas características para la visualización de su presión arterial, tales como la posibilidad de visualizar en primer plano la última medición de PA, además de una tabla con todos los registros organizados temporalmente, los cuales también son visibles en un plano bidimensional para una mejor interacción.

Para el logro de esta etapa, se optó por implementar la metodología de diseño de Software RUP (Rational Unified Process). Este marco de desarrollo de procesos en software se presenta como una referencia útil, dado el amplio conjunto de roles propuestos para estructurar equipos de desarrollo de software. RUP es un enfoque para la asignación de tareas y responsabilidades dentro de una organización, con el objetivo de garantizar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios. Lo anterior supeditado a un estricto cumplimiento de un calendario y presupuesto [30].

Una de las características esenciales en el diseño del App, acorde a los parámetros de RUP, son los casos de uso. Estos especifican los requisitos de sistema, además de guiar el proceso en las 4 etapas principales: Requerimientos, análisis y diseño, implementación y pruebas. Para la realización de las tareas en cada una de las anteriores etapas se utilizó el lenguaje de modelamiento visual UML, el cual permite una adecuada gestión de los modelos y consistencia entre los diagramas. La siguiente figura muestra el diagrama de clases e identificación de actores principales en el prototipo de medición de la PA.

Figura 3-10: Diagrama de clases del prototipo, (Fuente autor).

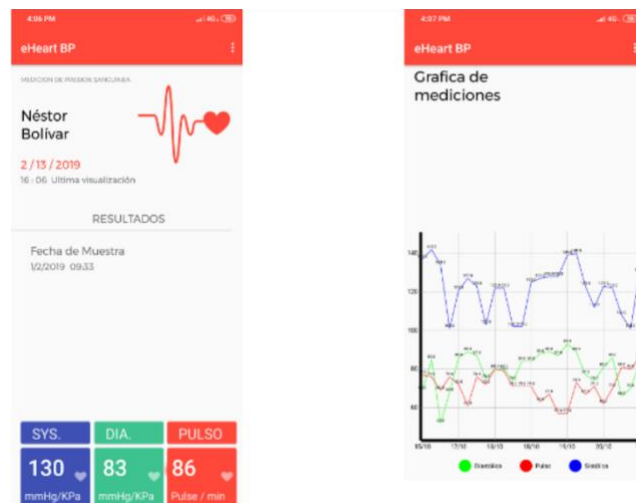


Adicionalmente, se realizaron los principales diagramas de secuencia que involucran la tarea de medición de la PA por parte del usuario, el proceso del almacenamiento y visualización de los registros, además de la recepción de la notificación de predicción proveniente del servidor donde se ejecuta el modelo de ML.

Por otra parte, la opción de software para desarrollar el App se orientó a la utilización de Android Studio, debido a las herramientas que este provee, la fácil sincronización con Firebase y la flexibilidad que sólo el IDE nativo de Android proporciona. Además, este IDE cuenta con una herramienta de virtualización, en la cual se pueden realizar pruebas y corrección de errores de manera ágil y confiable, siempre que se tenga los recursos de cómputo suficientes.

En cuanto al diseño de la interfaz de la aplicación, en un principio se realizaron Mockups o plantillas de prueba que permitieron recolectar la información necesaria sobre la mejor disposición y visualización de los elementos. A continuación, en la **Figura 3-11** se muestran dos de las interfaces del App.

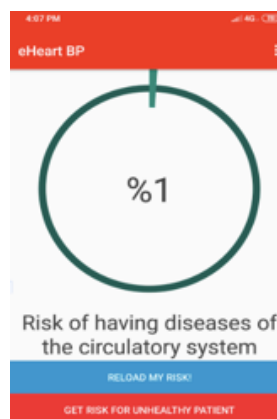
Figura 3-11: Interfaces del aplicativo para la visualización de los registros, (Fuente autor).



Finalmente, en las opciones de menú aparece una opción de predicción la cual permite la ejecución del algoritmo implementado en TensorFlow, que permite generar una probabilidad sobre la eventual aparición de un episodio de hipertensión arterial.

Este análisis es posible mediante la ejecución de un método en el App, que organiza los registros alojados en Firebase y los envía al servidor de ejecución de TensorFlow, de modo que el algoritmo de ML recibe de información, la procesa y devuelve el índice porcentual de probabilidad que se muestra en el aplicativo (ver **Figura 3-12**). En la siguiente sección se escribe en profundidad el diseño de los algoritmos de procesamiento.

Figura 3-12: Índice porcentual de la herramienta de predicción, (Fuente autor).



Adicionalmente, en esta ultima interfaz se muestran las recomendaciones principales orientadas a la promoción del autocuidado y hábitos saludables, para ese usuario con factores de riesgo de hipertensión arterial. Algunos de esos factores se relacionan con el uso de tabaco, una dieta poco saludable e inactividad física (que en conjunto resultan en obesidad), presión sanguínea elevada (hipertensión), lípidos sanguíneos anormales (dislipidemia) y glucosa en sangre elevada (diabetes) [1].

3.2.4 Diseño de los algoritmos basados en ML

Para el procesamiento de los datos y la predicción que se desea obtener de estos, se optó por la utilización de las librerías TensorFlow y Keras. Estas proveen código abierto para la utilización de redes neuronales que pueden implementarse en tareas de clasificación en una base de datos determinada. Por otra parte, en el software de Matlab también se han diseñado los algoritmos de regresión lineal y logística, de modo que se puedan realizar pruebas de comparación entre la precisión de uno u otro.

Regresión lineal multivariada

Este tipo de regresión es un método estadístico el cual permite estimar la relación entre una variable dependiente y dos o más variables independientes. La regresión lineal multivariada produce un modelo que identifica la mejor combinación ponderada de variables independientes para predecir la variable dependiente [31]. Para lograr un diseño apropiado del algoritmo se deben tener en cuenta los siguientes parámetros para la formulación de la función hipótesis:

$x_j^{(i)}$ = valor del atributo j en el i^{th} ejemplo de entrenamiento

x^i = Los atributos de entrada en el i^{th} ejemplo de entrenamiento

m = Numero de ejemplos de entrenamiento

n = Numero de atributos o características

(Fuente, [32])

La ecuación resultante de la hipótesis con múltiples atributos es la siguiente:

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n \quad (3.6)$$

Para los objetivos de la presente investigación se requiere vectorizar la anterior hipótesis de modo que la multiplicación entre matrices se pueda llevar a cabo. Esto se puede representar de modo general en la siguiente ecuación.

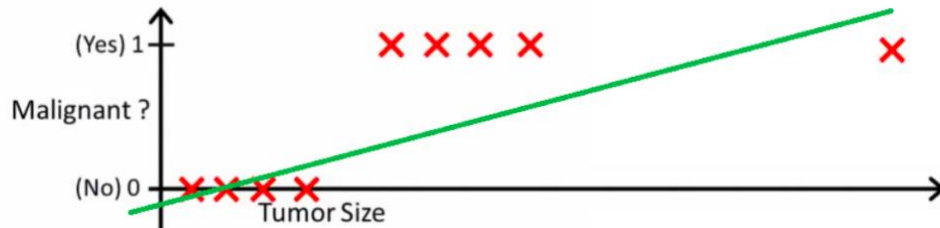
$$h_{\theta}(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x \quad (3.7)$$

Otra técnica bastante común en este tipo de regresión, se orienta a la utilización de Gradiente Descendiente, donde se toman los parámetros θ , antes mencionados, como un vector unidimensional de modo que se pueda obtener las derivadas parciales simultáneamente para cada uno de los atributos. La siguiente ecuación ilustra la derivada parcial de la función de coste $J(\theta)$ y la actualización simultanea para cada uno de los valores del vector θ [32].

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \quad \text{for } j := 0 \dots n \quad (3.8)$$

Este modelo desarrollado en Matlab, cuyas ecuaciones (3.6), (3.7) y (3.8) se configuran en [32], permite resaltar los requerimientos necesarios en la predicción de un episodio de hipertensión. En relación con el algoritmo de regresión lineal multivariada, este permite un valor de predicción futuro de acuerdo a la dispersión de los datos en torno a una variable objetivo. No obstante, para los requerimientos de predicción que se necesitan, esta técnica es susceptible a configurar una pendiente de gráfica acorde a la dispersión de datos, lo que puede significar interpretaciones incorrectas en la clasificación. Adicionalmente, con este algoritmo se obtienen valores superiores en magnitud a la variable objetivo, independientemente que la variable objetivo este con los valores 0 o 1. La **Figura 3-13** esquematiza el problema relacionado con el uso de un algoritmo de regresión lineal.

Figura 3-13: Problema de la hipótesis para el algoritmo regresión lineal [32].



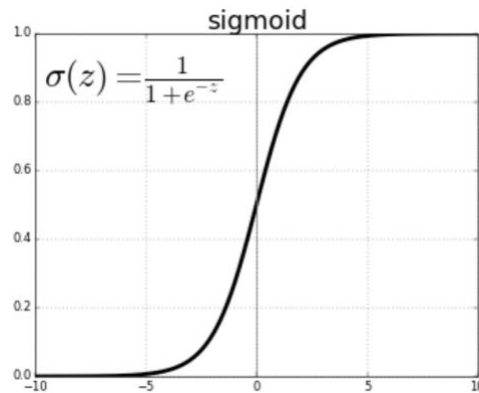
Debido al comportamiento de la función generada por el algoritmo de regresión lineal, una opción que provee una solución mas acorde y una clasificación idónea de la variable objetivo se relaciona al algoritmo de clasificación logística, el cual se explica seguidamente.

Regresión logística

Similar a la regresión lineal, este es un modelo estadístico utilizado para medir la relación entre una variable categórica dependiente y una o más variables independientes. De acuerdo a los valores que pueda tomar esta variable dependiente, el algoritmo puede denominarse como análisis multinomial o binario [13]. En el caso de análisis de regresión logística binario, la variable objetivo se suele codificar con un 1, si el sujeto padece de esta condición cardiaca, o 0 si no la padece.

En el caso que concierne a esta investigación, el modelo proporciona la probabilidad de sufrir de hipertensión arterial mediante una función de los parámetros de riesgo. En ese caso, se utiliza una función sigmoide para la tarea de clasificación, con esto se logra corregir una dispersión inesperada de los datos dada por uno o varios atributos de la DB. La **Figura 3-14** muestra el comportamiento de esa función, cuyos valores de convergencia son acordes a la función trazada en la **Figura 3-15**.

Figura 3-14: Función sigmoide utilizada en el algoritmo de regresión logística [32].



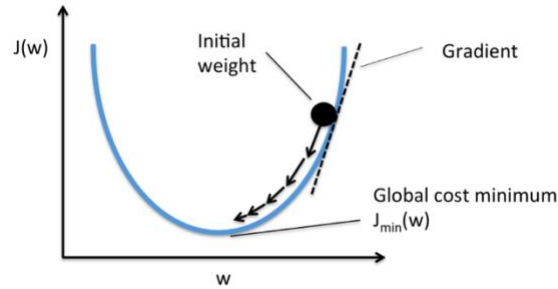
En la utilización de esta técnica se exige una variación en la función de coste, la cual permitirá converger apropiadamente para la variable objetivo. La función considerada para este caso de regresión logística se describe a continuación. Con esta variación se asegura

la probabilidad de predicción máxima o, de otro modo, el valor de convergencia mínimo, tal cual se muestra en la **Figura 3-15**.

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)})$$

$$\begin{aligned} \text{Cost}(h_{\theta}(x), y) &= -\log(h_{\theta}(x)) & \text{if } y = 1 \\ \text{Cost}(h_{\theta}(x), y) &= -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{aligned} \quad (3.9)$$

Figura 3-15: Valor de convergencia de la función de coste [33].



Acorde con la ecuación (3.9), si la respuesta correcta 'y' es 1, entonces la función de costo será 0, siempre que la función de hipótesis genere una salida de 1. Por el contrario, si la hipótesis se aproxima a 0, entonces la función de costo se acercará al infinito. También es importante resaltar que escribir la función de coste de esta manera garantiza que $J(\theta)$ sea convexo para la regresión logística. Seguidamente, se referencia la ecuación utilizada en la regresión logística multivariada.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] \quad (3.10)$$

Redes Neuronales Artificiales

El típico procedimiento de procesamiento de una red neuronal artificial (RNA) es: un conjunto de neuronas de entrada las cuales son activadas por las entradas, luego las activaciones de estas neuronas son pasadas, ponderadas y transformadas por funciones dadas por la red a otras neuronas, hasta que finalmente las neuronas de salida son activadas y generar resultados [13]. Este proceso permite el ajuste de los datos sin presentar ninguna especificación explícita en la distribución del modelo.

Además, las RNA son efectivas para modelar y reflejar las complejas relaciones de entradas y salidas de conjunto de datos destinado al entrenamiento para la clasificación o predicción de patrones, características o clusters de datos. La función de coste en este caso, comparada con la ecuación (3.9), es un poco más complicada y se presenta a

continuación. Seguidamente, se enumeran las variables que se usaran para el diseño del modelo de predicción:

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[y_k^{(i)} \log((h_{\Theta}(x^{(i)}))_k) + (1 - y_k^{(i)}) \log(1 - (h_{\Theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{j,i}^{(l)})^2 \quad (3.11)$$

L = Total de capas en la red

s_l = Numero de unidades en la capa 1

K = Numero de unidades de salida

Analizando la ecuación (3.11), se han agregado algunas sumas anidadas para dar cuenta de los múltiples nodos de salida. En la primera parte de la ecuación, antes de los corchetes, se tiene una suma adicional anidada que recorre el número de nodos de salida. Por otro lado, en la parte de regularización, después de los corchetes, se deben tener en cuenta las múltiples matrices Theta [32]. El número de columnas en la matriz theta actual es igual al número de nodos en la capa actual (incluida la unidad de sesgo).

Otro aspecto a considerar se refiere a la evaluación de las medidas proporcionadas por la matriz de confusión la cual provee los valores precisión, sensibilidad y especificidad, entre otros. Estas métricas son relevantes en el proceso de predicción de un modelo de clasificación, puesto que permiten la evaluación del desempeño de un algoritmo de aprendizaje supervisado [13]. En la **Tabla 3-4** se muestra la matriz de confusión para el caso de la predicción de la hipertensión.

Tabla 3-4: Matriz de confusión para la predicción de la hipertensión arterial [13].

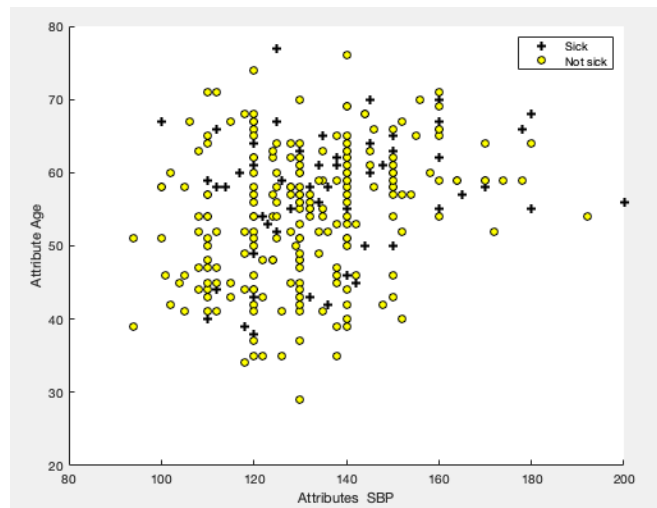
Class predicted	Real situation	
	Hypertension	Non-hypertension
Hypertension	TP	FP
Non-hypertension	FN	TN

3.2.5 Implementación de los algoritmos

Dados los principales detalles de los algoritmos diseñados en Matlab, el siguiente paso consiste en la búsqueda de una base de datos que contenga los atributos capturados por el tensiómetro arterial y la variable objetivo de interés, que en este caso se refiere a un valor de clasificación de hipertensión arterial. Un repositorio de importante trascendencia y referenciado en diversas investigaciones es Kaggle, una comunidad de científicos y personas interesadas en el aprendizaje máquina, que provee bases de datos para diversos tipos de investigación y desarrollo.

La base de datos seleccionada, provee 70.000 registros de pacientes con 11 atributos, de los cuales 5 de estos son de interés en esta investigación y se especifican la **Tabla 2-3**. Esta base de datos se carga a Matlab, de modo que se puedan visualizar a modo de prueba los registros etiquetados con edad y la variable de SBP. La siguiente figura muestra los resultados obtenidos contrastando los atributos de Edad y PAS. Analizando múltiples comparaciones de la información consignada en la DB, es imprescindible realizar una limpieza de los errores y datos incoherentes que se encontraron en los distintos atributos.

Figura 3-16: Distribución de datos en la DB seleccionado, (Fuente autor).



Pruebas con regresión lineal

Como se puede apreciar en la **Figura 3-16**, existe un alto índice de dispersión para los registros categorizados con la variable objetivo (enfermo o no enfermo). Esto significa que puede complicarse encontrar la función lineal que permita una clasificación idónea. Por lo

tanto, previamente se realizarán pruebas con la DB que permitan la manipulación de los datos e identificar los atributos claves para la predicción de las PAS y PAD.

Como primera exploración se toman los atributos de *Edad* y *PAS* en la identificación del valor correspondiente al *Colesterol* (Atributo de la BD). El diseño de código implementado realiza una normalización de los atributos seleccionados mediante el calculo de la *Desviación Estándar* en cada columna. De este modo se obtiene una matriz en la que se puede ejecutar el algoritmo de *Gradiente descendiente*, el cual se explica en la ecuación (3.8). Los resultados obtenidos del vector θ , mediante múltiples iteraciones y con la taza de aprendizaje α , son los siguientes:

Running gradient descent ...

Theta computed from gradient descent:

246.537211 *mg/dl*

7.405294

-9.684217

5.785819

3.660185

Predicted chol of 60 AGE , 132 mmHG (using gradient descent):

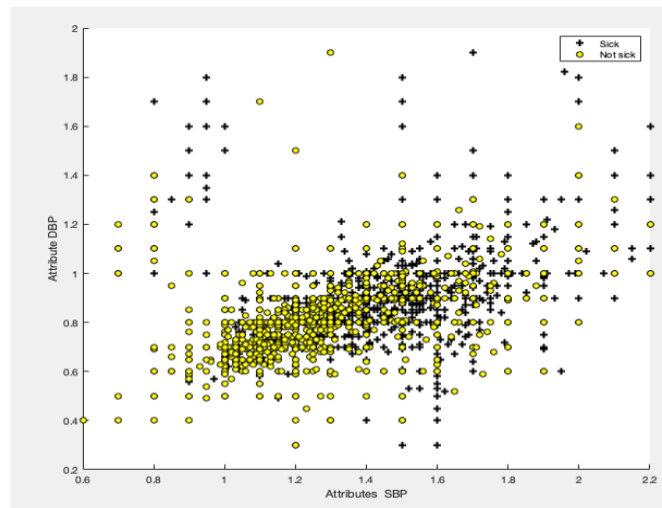
231.67 *mg/dl*

Lo anterior se evaluó en un sujeto de prueba de 60 años, con PAS de 132 y nivel de colesterol (*CHOL*) de 245.19 *mg/dl*. Comparando los valores de la variable *CHOL*, en diferentes iteraciones y cambios realizados, se obtuvo una predicción aproximada y superior al 90%. Si bien, la predicción lograda proporciona un valor cercano a dato real en la variable *CHOL*, para implementar el algoritmo en la predicción de valores de pre-hipertensión y superiores (**Tabla 1-1**), es necesario involucrar múltiples atributos del usuario en un algoritmo que permita la clasificación binaria, para este caso se utiliza la *Regresión Logística*.

Pruebas con Regresión Logística

Para la implementación de la función sigmoide (ver **Figura 3-14**) en el algoritmo de regresión logística, es necesario normalizar los atributos que se van a comparar. En este caso se visualiza en la **Figura 3-17** los registros de PAD y PAS, identificados con la variable objetivo. Este proceso sigue con el ajuste de la taza de regularización o lambda λ y el numero de iteraciones que se ejecute la función de optimización.

Figura 3-17: Distribución de muestras para los atributos PAS y PAD, (Fuente autor).

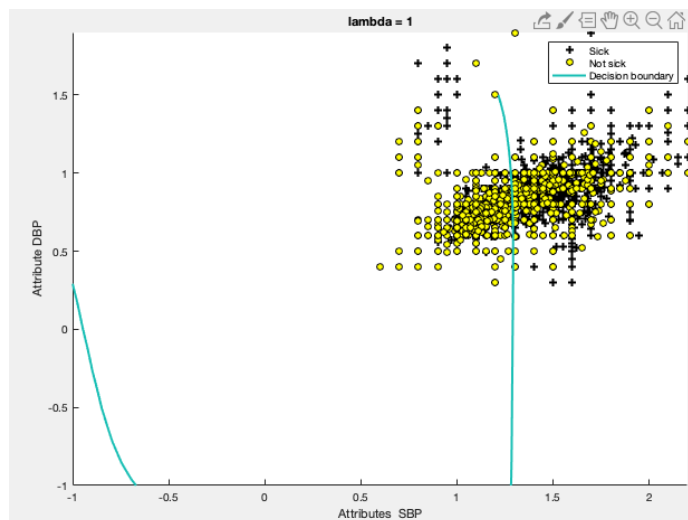


Posteriormente, se utiliza *fminunc* función que permite encontrar el mínimo de una función multivariable sin restricciones. Debido a este proceso, se logra un resultado de la función de coste de 0.615866 con un valor λ de 1, y una exactitud en el entrenamiento de 71.387 aproximadamente. En la **Figura 3-18** se aprecia la frontera de decisión configurada por el algoritmo y los resultados mencionados especificados enseguida.

Cost at test theta (with lambda = 1): 0.615866

Train Accuracy: 71.387143

Figura 3-18: Frontera de decisión obtenida del aprendizaje multivariable, (Fuente autor).



Pruebas con Redes Neuronales Artificiales

Para el desarrollo e implementación de esta técnica se utilizaron herramientas externas que permitieran su construcción y posteriores pruebas. Google Colab, es un entorno de desarrollo web, el cual permite la ejecución de código en Python utilizando el almacenamiento en Google Drive. Estas características, proveen flexibilidad al momento de interactuar con la DB de los registros de PA, además de acceder a los contenidos de ML y RNA de las librerías TensorFlow y Keras. A continuación, se describe el proceso de iteración con RNA y los resultados obtenidos.

eHeartBP

Haciendo uso del dispositivo para medir presión arterial, se obtienen 3 medidas:

- *Presión sanguínea sistólica: Mide la presión en los vasos sanguíneos cuando el corazón late.*
- *Presión sanguínea diastólica: Mide la presión en los vasos sanguíneos cuando el corazón descansa entre latidos.*
- *Pulsaciones por minuto.*

Información obtenida de CDC²

En la anterior descripción se exponen los principales atributos y métricas a tener en cuenta en los análisis posteriores (ver **Tabla 1-1**). Después, se contrastan los atributos capturados por el tensiómetro, con los registros existentes en la BD. Enseguida se muestran los 5 principales atributos que formaran parte del modelo de análisis realizado con la RNA.

Construcción del modelo

Haciendo uso del dataset Heart Attack Prediction disponible en Kaggle, se encuentran 14 atributos (de los cuales se posee 4 de estos valores a partir de las mediciones).

1. **Edad**
2. **Sexo**
3. *cp: Dolor de pecho*
4. **trestbps**: *Presión arterial. Sistólica*
5. *chol: Colesterol sérico*
6. *fbs: Glucemia en ayunas*
7. *restecg: Resultados electrocardiográficos en reposo.*
8. **thalach**: *Ritmo cardíaco máximo alcanzado*
9. *exang: Angina inducida por el ejercicio*
10. *oldpeak: Depresión ST inducida por el ejercicio en relación con el descanso.*
11. *slope: La pendiente del ejercicio pico.*
12. *ca: Número de vasos principales (0-3) coloreados por fluoroscopia*
13. *thal: Ritmo cardíaco.*

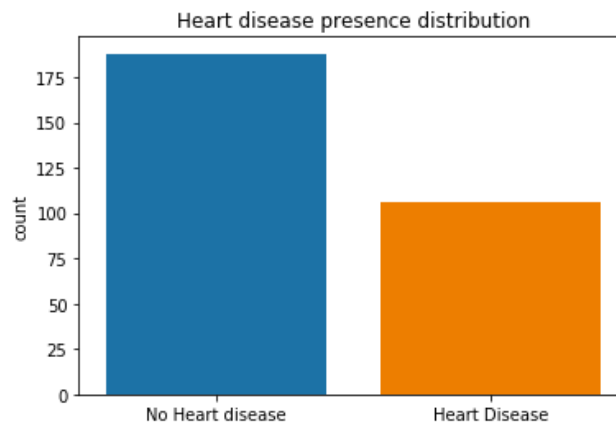
14. *num*: Diagnóstico de enfermedad de hipertensión

0* representa un estrechamiento del diámetro menor al 50%

1* representa un estrechamiento del diámetro mayor al 50%

Prevía visualización del panorama de trabajo y los atributos destacados, se procede a realizar una exploración de los datos, fundamentada en la agrupación de las personas con *Diagnostico de enfermedad*. La **Figura 3-19** muestra la distribución de muestras provenientes del dataset *Heart Attack Prediction* disponible en *Kaggle*³. El proceso detallado estará disponible en la sección de anexos de este documento.

Figura 3-19: Distribución de muestras del dataset, (Fuente autor).



Posteriormente, se crea el correspondiente tensor de modo que se discrimine la variable objetivo y los registros de la BD. Luego se separan los datos destinados al entrenamiento del algoritmo y los que servirán para realizar pruebas de precisión. Finalmente, después de la definición del modelo utilizando la librería de Keras, se implementan 5 capas de neuronas. Es importante resaltar que el problema abordado es de clasificación, por lo tanto, el número de unidades o neuronas en la última capa es de 1. A continuación, se generaliza el procedimiento antes descrito.

```
dataset = tf.data.Dataset.from_tensor_slices((data.values, target.values))
train, test = train_test_split(data, test_size = 100, random_state = RANDOM_SEED)

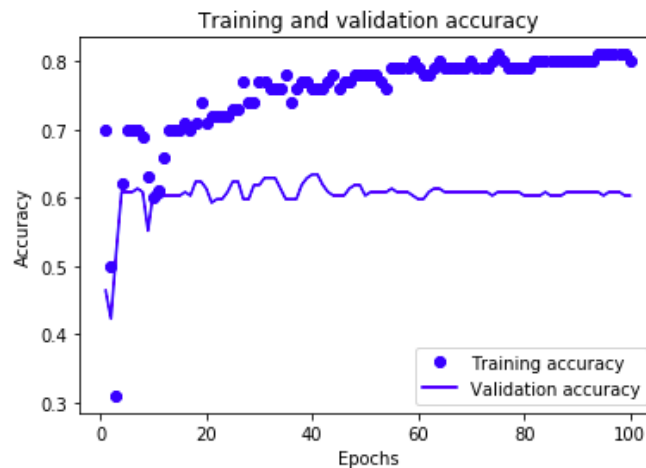
model = tf.keras.Sequential([
```

³ Plataforma destinada al aprendizaje de Machine Learning.

```
tf.keras.layers.Dense(10, activation = 'relu'),  
tf.keras.layers.Dense(32, activation = 'relu'),  
tf.keras.layers.Dense(64, activation = 'relu'),  
tf.keras.layers.Dense(16, activation = 'relu'),  
tf.keras.layers.Dense(1, activation = 'sigmoid')])
```

Finalmente, se obtiene un porcentaje cercano al 67% (*Model accuracy: 67.53%*), de la precisión en la predicción de los datos de prueba. En la **Figura 3-20** se observa la comparativa entre los resultados de precisión del dataset de entrenamiento y el valor obtenido en las pruebas, lo anterior respecto a las épocas o iteraciones de aprendizaje. Como paso final, se exporta el modelo para ser desplegado en el servidor, donde se podrán ejecutar las peticiones de predicción necesarias desde el dispositivo móvil.

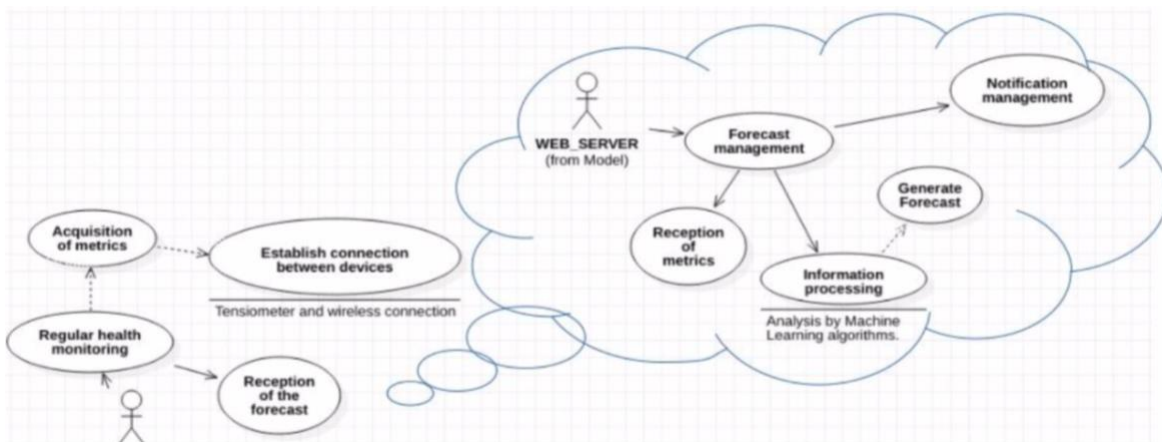
Figura 3-20: Métricas de exactitud obtenidas por el modelo realizado en TensorFlow, (Fuente autor).



4.Resultados

Previa explicación de las desventajas en el proceso rutinario de medición de la PA, descrito en el Capítulo 1, esta propuesta se enfoca en proporcionar una solución en el campo de la telecardiología, que proporcione información de fácil acceso para el usuario. Adicionalmente, El prototipo propuesto es consistente con el objetivo del Ministerio de Salud de Colombia de reducir en un 25% la mortalidad por ataques cardíacos en el territorio para 2025 [2], a través, de la convergencia de las TIC en el campo de la prevención y la atención médica. Lo anterior soporta y configura el horizonte de esta investigación, orientado en el autocuidado y la prevención en personas con factores de riesgo de sufrir hipertensión arterial. La **Figura 3-21** ilustra el esquema de comunicación propuesto y las diferentes tareas ejecutadas por el usuario de la plataforma de recepción y análisis de las medidas.

Figura 3-21: Esquema de comunicaciones del prototipo de monitoreo de la PA, (Fuente autor).



El diseño del prototipo (**Figura 3-21**) propone el uso de un monitor de presión arterial de muñeca, basado en el método oscilométrico para la medición de la PA, con lo cual se logra una precisión media-alta (ver **Tabla 2-2**). Bajo estas características, se proporciona un

índice de medición de confiabilidad aceptable para el usuario. Por otra parte, atendiendo la imposibilidad de interconectar el dispositivo, se agrega el módulo ESP8266 que permite exportar los datos de PA al servidor de Firebase para su posterior análisis.

En esta investigación, el desarrollo de los algoritmos de ML se fundamentó en el uso de RNA y un dataset disponible en Kaggle. La utilización de las librerías de TensorFlow y Keras permitieron implementar un modelo de RNA entrenado por el dataset antes mencionado, que posteriormente analiza los registros de PA almacenados en Firebase. El promedio de precisión del primer modelo de RNA es del 67% (ver **Figura 3-20**), sin embargo, es importante mencionar que el entrenamiento del mismo se realizó con 290 muestras, lo que posiblemente sea un factor relevante en este tipo de modelos. Además, cabe mencionar que, en este proceso de predicción, se usan 3 atributos cambiantes del conjunto de datos de entrenamiento; PAS, PAD y HR (ver **Tabla 2-3**). La utilización de los otros atributos podría significar un índice de precisión y confiabilidad más elevado en el modelo.

En cuanto a la usabilidad de los elementos físicos, estos se diseñaron según los requerimientos de funcionalidad y diseño planteados en la metodología. En cuanto al tensiómetro, este dispositivo es capaz de enviar inalámbricamente las mediciones de PA a la DB en tiempo real de Firebase. Si bien, la adhesión de la placa ESP8266 tiene un consumo extra en el consumo del tensiómetro, este no llega a afectar las funciones básicas de operación del mismo, debido a un diseño específico del código de la tarjeta que permite ponerla en estado de suspensión cuando no se utiliza. En cuanto a la captura de las variables de interés, este proceso se realiza sin inconvenientes gracias a las librerías de comunicación serial disponibles en el IDE de Arduino. Estas se utilizan para el reconocimiento de las señales eléctricas generadas por la memoria EEPROM interna del tensiómetro, cuando se realiza una medición de la PA.

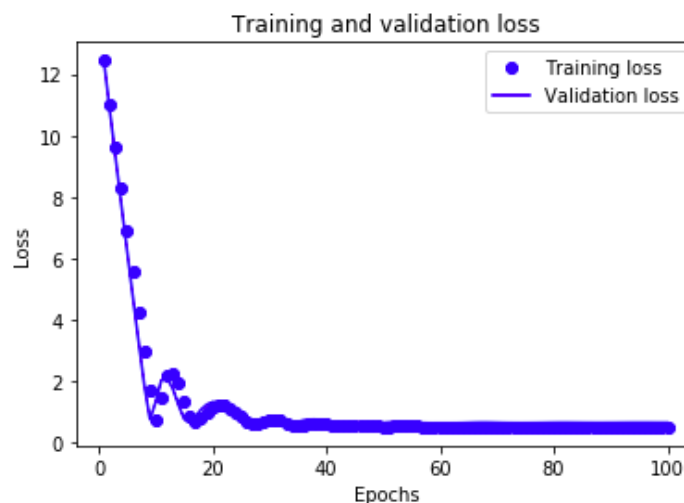
Con las anteriores modificaciones, los registros del usuario están disponibles en la app móvil, como para la plataforma de TensorFlow. Entorno al aplicativo móvil, este cumple con las principales funciones informativas y visuales sobre el estado de la PA del usuario, además de generar las peticiones necesarias para visualizar los registros de forma temporal en la pantalla del celular.

4.1 Discusión de resultados

Al contrastar los resultados obtenidos con la investigación realizada por Aborokbah et al en [34], los autores desarrollaron e implementaron algoritmos basados en SVM logrando una precisión alrededor del 72% en las predicciones de fallas cardíacas. En relación con esta métrica, la precisión alcanzada con el prototipo podría mejorarse cambiando el número de muestras en el entrenamiento, la tasa de aprendizaje y las épocas. En cuanto a los posibles cambios en el modelo, se pueden involucrar capas adicionales para el análisis y un número reducido de iteraciones, todo dependiente del número de atributos con que se vaya a entrenar la RNA.

Sin embargo, en relación a las muestras utilizadas para el entrenamiento del algoritmo, con que se logro una precisión cercana al 67%, es prudente mencionar que no se presentaron inconvenientes de *Overfitting* o *Underfitting*. Como se muestra en la **Figura 3-22** donde se evidencia como las curvas de entrenamiento y validación del algoritmo tienen valores similares. Caso contrario si los valores discretos de perdida en el entrenamiento fueran mucho menores respecto a los valores de validación, se presentaría un caso de *Overfitting*.

Figura 3-22: Comparativa de los resultados en el entrenamiento y la validación del modelo, (Fuente autor).



Por otra parte, es importante resaltar que el modelo expuesto en [34], permite la clasificación de pacientes con riesgo de padecer ataques cardíacos, con lo que se pueden relacionar con *eHearBP* los factores de riesgo asociados a condiciones cardíacas,

algoritmos de clasificación en incluso atributos de las bases de datos disponibles. No obstante, el prototipo presentado, aborda una perspectiva de promoción y prevención en el autocuidado de personas con factores de riesgo de hipertensión. De acuerdo a la anterior premisa, se abordó el problema desde el paradigma IoT abarcando los aspectos de captura de la señal, presentación de la información y procesamiento de los datos.

En este orden de ideas, la investigación presentada en [34], permite realizar un análisis comparativo respecto a las técnicas de aprendizaje automático utilizadas en la clasificación de pacientes con posible riesgo de padecer alguna condición cardíaca. En este punto se abordan caminos diferentes, en razón de que el prototipo expuesto en este documento, adicionalmente proporciona nuevas soluciones técnicas para proveer eficazmente un servicio de monitoreo entorno a la actividad diaria de un paciente utilizando las variables primarias de la presión arterial.

4.2 Productos realizados

Es meritorio mencionar que el desarrollo de este proyecto es consistente en el contexto tecnológico actual de IoT y consecuentemente la investigación se ha divulgado en múltiples escenarios académicos con el nombre de e-heartBP. En 2018, parte del proyecto se presentó en el concurso internacional TuApp realizado en Lima capital de Perú. En este escenario se logró el séptimo puesto a nivel Latinoamérica y el primer puesto de los proyectos presentados desde la Universidad Nacional de Colombia. Las múltiples contribuciones, sugerencias y contactos realizados en Perú, han contribuido al desarrollo de la investigación.

Otro producto generado, se compartió en el año 2019 a modo de publicación, donde se presentó un artículo corto para el congreso *Fourth International Workshop on ICT Solutions for Health (ICTS4eHealth'19)*, auspiciado por la IEEE Xplore (The Institute of Electrical and Electronics Engineers). Documento relacionado con el identificador DOI: [10.1109/CHASE48038.2019.00025](https://doi.org/10.1109/CHASE48038.2019.00025). Este evento se realizó en Washington D.C., EEUU. Lugar que congregó diversas investigaciones relacionadas el uso de Internet, detección, comunicaciones y técnicas inteligentes en apoyo de aplicaciones relacionadas con la salud.

Los anteriores productos y muestras de divulgación, pretenden contribuir al campo de las Tecnologías de Información y Comunicación, dirigidas a los sistemas de apoyo y monitoreo sustentados en dispositivos interconectados orientados a los hábitos saludables, en particular a la salud cardiovascular.

5. Conclusiones y recomendaciones

5.1 Conclusiones

Acorde al marco general de este estudio se destaca la culminación de las múltiples tareas que han permitido la realización del prototipo desde las perspectivas mencionadas en numerosas ocasiones (percepción, red y aplicación). Esto contribuye a la esquematización y arquitectura de comunicación que deben manejar estos dispositivos médicos.

Profundizando en la etapa de procesamiento y análisis de datos, referente al rendimiento del algoritmo de procesamiento utilizado, el porcentaje de precisión del modelo basado en RNA es cercano al 67%, no obstante, en el apartado de recomendaciones se mencionan puntos clave para la generación de un rendimiento superior. Por otra parte, este algoritmo basado en RNA, se ha diseñado considerando las posibles situaciones de overfitting o underfitting. Logrando así un adecuado balance, tal cual se muestra en la **Figura 3-22**.

Respecto a la interactividad del prototipo en las múltiples plataformas, el almacenamiento de los datos desde el tensiómetro hacia Firebase, y posteriormente la visualización de los registros a través de la aplicación, demora de 10 a 40 segundos. Con lo anterior se logra la disponibilidad de las mediciones de PA en tiempo cercano al real, posibilitando notificaciones o alertas a eventualidades críticas.

En cuanto a la interfaz de usuario, se posibilita una herramienta de fácil accesibilidad y gestión. Como resultado, se produce una mejora en el tiempo de visualización y análisis de los registros de PA. El personal médico puede analizar los registros históricos del usuario, de una manera dinámica y cómoda. Acorde a las pruebas realizadas, se logra una disminución del tiempo en cerca de 3 minutos por revisión médica en cada cita, en comparación con el proceso tradicional.

5.2 Recomendaciones

En relación con la precisión en la predicción del modelo, un aspecto a mejorar, se asocia a la modificación de los hiperparámetros del modelo de RNA en los cuales se pueden configurar las opciones de neuronas por capa, numero de capas, tasa de aprendizaje, entre otros. Lo anterior podría ajustarse a las necesidades y atributos de una DB en particular, obteniendo resultados propicios para el caso. Por otro lado, la implementación de un algoritmo de regresión lineal, podría proporcionar una precisión promedio del modelo en 71%, tal como se menciona en las pruebas realizadas con diferentes algoritmos, expuestas en la pagina 38.

En cuanto a la funcionalidad física del tensiómetro, la adhesión de la placa ESP8266 genera un consumo adicional a las baterías, aproximadamente del 10%. Este punto fue posible optimizarlo mediante el diseño del código y el requerimiento de transmisión de información exclusivamente en los eventos de captura y envío de la medida de PA.

En relación con las características y atributos del dataset utilizado para almacenar las variables propias del usuario, en un futuro desarrollo este podría almacenar información adicional sobre enfermedades o condiciones particulares del paciente. Lo anterior, significaría un modelo de análisis de datos mas completo, el cual podría proporcionar información mas precisa y promisoria entorno a la salud y cuidados preventorios de la persona.

Adicionalmente, este prototipo pretende contribuir a la digitalización y tecnificación de algunos procesos rutinarios en los cuales es posible involucrar herramientas de las TIC. En este caso, orientados hacia el autocuidado y promoción de los buenos hábitos de salud para evitar complicaciones relacionadas a la presión arterial. Esto fundamenta un esquema de posibilidades hacia la escalabilidad y análisis de información en grandes masas, posibilitando el descubrimiento de nueva información y patrones comportamentales, y una línea de investigación en el campo de las *Ciudades Inteligentes* o *Smart City*.

A. Anexo: Código de captura de datos de la tarjeta ESP8266

```
//FirebaseESP8266.h must be included before ESP8266WiFi.h
//Parametrización de librerías

#include "FirebaseESP8266.h"
#include < ESP8266WiFi.h >
#include < Wire.h >
#include < EEPROM.h >
#include < ArduinoJson.h >
////////////////////////////////////
//Declaración de los pines de a utilizar en la ESP8266
const int I2C_ERROR = 10;
const int sda = 5; //D1
const int scl = 4; //D2

//Ruta de los registros en Firebase
String path = "/iot/records";
String jsonStr = "";

//Parametros de conexión WIFI y BD Firebase
#define WIFI_SSID "Redmi"
#define WIFI_PASSWORD "811b3f739c850"
#define FIREBASE_HOST "eheartbp.firebaseio.com"
#define FIREBASE_AUTH "B3eFksOVtZq7m6URGmJgPISfJPshCBzBTH9MTW8E"

//Declaración de variables a capturar

boolean busActivity = false;
int error = 0;
int systolic = 0;
int diastolic = 0;
int hr = 0;

DynamicJsonBuffer jBuffer;
JsonObject& root = jBuffer.createObject();
JsonObject& tempTime = root.createNestedObject("timestamp");

FirebaseData firebaseData;
void setup()
{
  //Sincronizaciones iniciales
  pinMode(14, OUTPUT);
  digitalWrite(14, HIGH);

  delay(100);
  Serial.println("Wire initialized");
  Serial.begin(115200);
  Wire.begin(sda, scl);
  delay(100);
  Serial.println("Wire initialized");
  //////////////////////////////////
```

```

WiFi.begin(WIFI_SSID,WIFI_PASSWORD);
Serial.print("Connecting to Wi - Fi");
while (WiFi.status() != WL_CONNECTED)
{
  Serial.print(".");
  delay(300);
}
Serial.println();
Serial.print("Connected with IP: ");
Serial.println(WiFi.localIP());
Serial.println();

Firebase.begin(FIREBASE_HOST,FIREBASE_AUTH);
Firebase.reconnectWiFi(true);

//Lectura de la BD y configuración del tiempo de espera en escritura
Firebase.setReadTimeout(firebaseData,1000 * 60);
Firebase.setwriteSizeLimit(firebaseData,"tiny");
digitalWrite(14,LOW);
EEPROM.begin(4);
}
void loop()
{
  //Ciclo de lectura continua sobre nuevas mediciones
  int readCount = 0;
  int ourCount = 0;
  int addr = 0x00;

  // Trigger on SDA activity,user might have stored a new measurement
  busActivity = false;
  attachInterrupt(sda,busActivityDetected,CHANGE);

  Serial.println("Waiting for bus activity.");

  // Lets take a nap while we wait.
  while (busActivity == false){
    yield();
  }
  detachInterrupt(sda);
  // Delay a bit so the monitor's MCU can finish talking
  delay(500);

  // Get current measurement count
  readCount = getMeasurementCount();

  // The user might have turned off the unit causing an interrupt
  if (error == I2C_ERROR){
    Serial.println("Can't communicate with the chip.");
    return;
  }
  // Check count in our own eeprom, so we know if we have sent this data before
  ourCount = EEPROM.read(0x00);

  if (readCount != 0 && readCount != ourCount){
    Serial.println("New measurement detected!");
    // Get last measurement

    addr = (readCount - 1) * 4; // address for last measurement
    Serial.print("Reading from 0x");
    Serial.println(addr,HEX);
    refreshLastMeasurements(addr);

    Serial.print("Your last measurement is ");
    Serial.print(systolic,DEC);
    Serial.print(" mmHG X ");
    Serial.print(diastolic,DEC);
  }
}

```

```

Serial.print(" mmHG and ");
Serial.print(hr, DEC);
Serial.println(" bpm. Bye! ");

// Upload to firebase
uploadMeasurements();
// Store updated count in eeprom
EEPROM.write(0x00, readCount);
EEPROM.commit();

// Blink led so user knows everything is ok
blinkLED();

} else if (readCount == 0 && ourCount != 0) {
// BPM's memory was probably erased by the user
// So, update the count in EEPROM to keep it in sync

EEPROM.write(0x00, 0);
EEPROM.commit();
Serial.println("Memory was reset -- syncing EEPROM counter.");
// Blink it
blinkLED();

    } else {
        Serial.println("Hmm, I2C bus activity detected, but nothing to do.");
    }
}

void blinkLED() {
//Esta función cumple la función de indicador visual en el tensiometro

for (int i = 0; i < 4; i++) {
digitalWrite(14, HIGH); // Turn the LED off
delay(100);
digitalWrite(14, LOW); // Turn the LED on (active low)
delay(100);
}
}

void refreshLastMeasurements(int addr) {
//Esta función pretende actualizar la última medición obtenida
byte rdata = 0x00;

// read measurement
Wire.beginTransmission(0x50);
Wire.write(addr);
Wire.endTransmission();
Wire.requestFrom(0x50, 4);

if (Wire.available() != 4) {
Serial.println("Less than 4 bytes found?");
return;
}
hr = Wire.read();
diastolic = Wire.read();
rdata = Wire.read();
systolic = Wire.read();
// systolic is stored halved
systolic *= 2;
// if this byte is 0x80, add 1 to make it odd
if (rdata == 0x80)
systolic += 1;
}

//////////
int getMeasurementCount() {
//Esta función permite la adquisición de los valores de la EEPROM
int addr = 0x00;
byte rdata = 0x00;
// Get current number of measurements at 0xFF

```

```

Wire.beginTransmission(0x50);
Wire.write(0xFF);
Wire.endTransmission();
Wire.requestFrom(0x50,1);

if (Wire.available()){
  rdata = Wire.read();
  error = 0;
}else{
  error = I2C_ERROR;
  // Wait a little for things to settle
  delay(300);
  return 0;
}
if (rdata == 0){
  Serial.println("Nothing stored.");
  return 0;
}
return rdata;
}
////////////////////////////////////
void uploadMeasurements(){
  //En esta función se suben los valores de medida a la BD
  root["diastolic"] = diastolic;
  root["pulse"] = hr;
  root["systolic"] = systolic;

  root.printTo(jsonStr);
  String cam;

  if (Firebase.pushJSON(firebaseData,path,jsonStr))
  {
    cam = firebaseData.pushName();

    Serial.println("PASSED");
    Serial.println("PATH: " + firebaseData.dataPath());
    Serial.print("PUSH NAME: ");
    Serial.println(cam);
    Serial.println("ETag: " + firebaseData.ETag());
    Serial.println("-----");
    Serial.println();
  }
  else
  {
    Serial.println("FAILED");
    Serial.println("REASON: " + firebaseData.errorReason());
    Serial.println("-----");
  }
  if (Firebase.setTimestamp(firebaseData,path + "/" + cam + "/timestamp"))
  {
    Serial.println("PASSED");
    Serial.println("PATH: " + firebaseData.dataPath());
    Serial.println("TYPE: " + firebaseData.dataType());
    Serial.print("TIMESTAMP: ");
    Serial.println(firebaseData.intData());
  }
}
////////////////////////////////////
void busActivityDetected(){
  ///Función flag la cual indica si existe una nueva medida disponible
  busActivity = true;
}

```


B. Anexo: Modelo para RNA desarrollado en Google Colab

eHeartBP

Haciendo uso del dispositivo para medir presión arterial, obtenemos 3 medidas:

- Presión sanguínea **sistólica**: Mide la presión en los vasos sanguíneos cuando el corazón late.
- Presión sanguínea **diastólica**: Mide la presión en los vasos sanguíneos cuando el corazón descansa entre latidos.
- **Pulsaciones** por minuto.

Niveles de presión arterial

Normal

Sistólica: menos de 120 mmHg

Diastólica: menos de 80mmHg

En riesgo (prehipertensión)

Sistólica: 120–139 mmHg

Diastólica: 80–89 mmHg

Alto

Sistólica: 140 mmHg o superior

Diastólica: 90 mmHg o superior

Construcción del modelo

Haciendo uso del dataset Heart Attack Prediction disponible en Kaggle, encontramos 14 atributos (de los cuales se posee 4 de estos valores a partir de las mediciones).

Información de los atributos:

Edad

Sexo

cp: Dolor de pecho

trestbps: Presión arterial. Sistólica

chol: Colesterol sérico

fbs: Glucemia en ayunas

restecg: Resultados electrocardiográficos en reposo.

thalach: Ritmo cardíaco máximo alcanzado

exang: Angina inducida por el ejercicio

oldpeak: Depresión ST inducida por el ejercicio en relación con el descanso.

slope: La pendiente del ejercicio pico.

ca: Número de vasos principales (0-3) coloreados por fluoroscopia

thal: Ritmo cardíaco.

num: Diagnóstico de enfermedad cardíaca (estado de enfermedad angiográfica).

0* representa un estrechamiento del diámetro menor al 50%.

1* representa un estrechamiento del diámetro mayor al 50%

```
[ ]
!pip install h5py pyyaml
[ ]
# import tensorflow, keras and pandas
import numpy as np
import tensorflow as tf
from tensorflow import keras
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import rc
from pylab import rcParams
from google.colab import drive
from sklearn.model_selection import train_test_split

drive.mount("/content/drive")
# 4/lgFSUAbn9J9jaYFcR3lCAO7_bhY9FDV-NkvXeuEUW5qQfjJlKd9XcQ
RANDOM_SEED = 42
np.random.seed(RANDOM_SEED)
[ ]
#from google.colab import files
#uploades = files.upload()
%cd DataSets
[ ]
heart_csv_path = "/content/drive/My Drive/Colab Notebooks/DataSets/data5.csv"
data = pd.read_csv(heart_csv_path)
data = data.sample(frac=1)
#data = data.astype('float32')
data.shape
[ ]
f = sns.countplot(x='num', data=data)
f.set_title("Heart disease presence distribution")
f.set_xticklabels(['No Heart disease', 'Heart Disease'])
plt.xlabel("");
[ ]
target = data.pop('num')
[ ]
dataset = tf.data.Dataset.from_tensor_slices((data.values, target.values))
[ ]
for feat, targ in dataset.take(10):
    print('Features: {}, Target: {}'.format(feat, targ))
[ ]
print(len(data))
train_dataset = dataset.shuffle(buffer_size=len(data)).batch(32)
[ ]
model = get_compiled_model()
model.fit(train_dataset, epochs=10)
```

Ahora se separan los datos para entrenamiento y pruebas.

```
[ ]
def create_dataset(dataframe, batch_size=32):
    dataframe = dataframe.copy()
    labels = dataframe.pop('num')
    return tf.data.Dataset.from_tensor_slices((dict(dataframe), labels)) \
```

```

        .shuffle(buffer_size=len(dataframe)) \
        .batch(batch_size)
[]
train, test = train_test_split(data, test_size=100, random_state=RANDOM_SEED)

[]
train_ds = create_dataset(train)
test_ds = create_dataset(test)

[]
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(units=128, activation='relu'),
    tf.keras.layers.Dropout(rate=0.2),
    tf.keras.layers.Dense(units=128, activation='relu'),
    tf.keras.layers.Dense(units=1, activation='sigmoid')
])
[]
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
history = model.fit(train_ds, validation_data=test_ds, epochs=100)
[]
def get_compiled_model():
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(10, activation='relu'),
        tf.keras.layers.Dense(32, activation='relu'),
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(16, activation='relu'),
        tf.keras.layers.Dense(1, activation='sigmoid')
    ])
    model.compile(optimizer='adam',
                  loss='binary_crossentropy',
                  metrics=['accuracy'])
    return model
[]
# create model
model = keras.Sequential()

# add layers
model.add(keras.layers.Dense(10, activation=tf.nn.relu, input_shape=(train_x_data.shape[1],)))
model.add(keras.layers.Dense(32, activation=tf.nn.relu))
model.add(keras.layers.Dense(64, activation=tf.nn.relu))
model.add(keras.layers.Dense(16, activation=tf.nn.relu))

# last layer has only two possible outcomes
# either 0 or 1 indicating not diagnosed and diagnosed respectively
model.add(keras.layers.Dense(1, activation=tf.nn.sigmoid))

# get summary of the model
#model.summary()

[]
# compile the model
model.compile(optimizer=tf.train.AdamOptimizer(),
              loss='binary_crossentropy',
              metrics=['accuracy'])
[]
# model fitting, we train and test the data
history = model.fit(train_x_data,
                    train_y_data,
                    epochs=100,
                    batch_size=100,
                    validation_data=(test_x_data, test_y_data),
                    verbose=1)
[]
# evaluate the model
# results = model.evaluate(test_x_data, test_y_data)
loss, acc = model.evaluate(test_x_data, test_y_data)
print("Model accuracy: {:.2f}%".format(100*acc))

```

```
[ ]
# Create graph for accuracy and loss
history_dict = history.history
history_dict.keys()
import matplotlib.pyplot as plt
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)

# "bo" is for "blue dot"
plt.plot(epochs, loss, 'bo', label='Training loss')
# b is for "solid blue line"
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()

[ ]
plt.clf() # clear figure
acc_values = history_dict['acc']
val_acc_values = history_dict['val_acc']

plt.plot(epochs, acc, 'bo', label='Training accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation accuracy')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.show()
```

Se usa el modelo para predecir la posibilidad de hipertensión.

```
[ ]
import io
from google.colab import files
uploaded = files.upload()
new_data = pd.read_csv(io.BytesIO(uploaded['data_for_prediction_1.csv']))
print("Predicción 1.\n")
print(new_data)
print("\n")
x_test = new_data[0:].values
classes = model.predict(x_test)
score = classes[0][0]

print("Score:"+str(score))
if score > 0.5:
    print("La medición presenta un ALTO riesgo de padecer una enfermedad cardiaca.")
    sendPush(score)
else:
    print("La medición presenta un bajo riesgo de padecer una enfermedad cardiaca.")

[ ]
import io
from google.colab import files
uploaded = files.upload()
new_data = pd.read_csv(io.BytesIO(uploaded['data_for_prediction_2.csv']))
print("Predicción 2.\n")
print(new_data)
print("\n")
x_test = new_data[0:].values
classes = model.predict(x_test)
score = classes[0][0]

print("Score:"+str(score))
if score > 0.5:
    print("El analisis de presenta un ALTO riesgo de padecer un episodio de hipertensión.")
```

```

    sendPush(score)
else:
    print("El analisis de presenta un bajo riesgo de padecer un episodio de hipertensión.")

```

A continuación, se exporta el modelo para ser desplegado en el servidor.

```

[]
saved_model_path = tf.contrib.saved_model.save_keras_model(model, "./saved_models")
!ls saved_models
[]
!python --version
!python -c 'import tensorflow as tf; print(tf.__version__)'
[]
import requests
def sendPush(score):
    url = "https://fcm.googleapis.com/fcm/send"

    payload = "{\n  \"notification\": {\n    \"title\": \"Alto riesgo \",\n    \"body\": \"Visita a tu medico!\",\n  },\n  \"data\": {\n    \"title\": \"Alto riesgo!\",\n    \"body\": \"Visita a tu medico!\",\n  },\n  \"to\": \"/topics/notifications\"\n}"
    headers = {
        'Content-Type': "application/json",
        'Authorization': "key=AAAAo6pBpyg:APA91bFM24yJPpmaJpSwCABZTDdb7Y_sKo53n4GmN8FvzQOI tq_bTSKLhN_Gh0wqZKh11mYdnccKjUerxmG-s7kG0OV4WMjw84gbYGdKDEjpvGBU0GgWSoV_3xAhdVX_qch4EE mNPsgS",
        'cache-control': "no-cache",
        'Postman-Token': "8d15176c-3962-44b6-a3b7-575c80320d76"
    }
    response = requests.request("POST", url, data=payload, headers=headers)
    print(response.text)

```

C. Anexo: Manual de uso del prototipo de medición de la PA

El monitor de presión arterial de muñeca automático oscilométrico modelo Panasonic EW3106 es un dispositivo orientado a medir las presiones arteriales sistólica y diastólica y la frecuencia de pulso de un individuo adulto. El aparato no está diseñado para ser usado en niños. Las medidas de PA obtenidas con este dispositivo son equivalentes a las obtenidas por un observador entrenado usando el método de auscultación con estetoscopio, dentro de los límites prescritos por el American National Estándar de esfigmomanómetros Manuales, electrónicos o automatizados [26].



Instrucciones Importantes Previas al Uso

1. Es importante no confundir el concepto de autocontrol con el de autodiagnóstico. Solamente un profesional familiarizado con su historia clínica debe interpretar las mediciones de la presión arterial.
2. La presión arterial puede variar según diversos factores incluidos la edad, el sexo, el peso y el estado físico. Puesto que las mediciones de la PA tomadas en un entorno clínico varían considerablemente respecto de las tomadas en casa, se debe tomar la presión no solo en el consultorio sino en el hogar de manera frecuente.
3. Tómese la presión todos los días a la misma hora y en las mismas condiciones. El momento ideal para tomar la presión (para obtener la denominada “presión de la

sangre base") es por la mañana, justo después de levantarse, antes de desayunar y antes de hacer actividad o ejercicio alguno.

4. Las siguientes situaciones pueden provocar variaciones considerables en las lecturas de la presión de la sangre y, por lo tanto, se las debe evitar al menos 30 minutos antes de tomar la presión.

La presión de la sangre es mayor que lo normal:

- En estado de excitación o tensión
- Al tomar un baño
- Durante el ejercicio o inmediatamente después
- Cuando hace frío
- En un lapso de una hora después de haber comido
- Después de tomar café, té u otras bebidas que contengan cafeína
- Después de fumar tabaco
- Con la vejiga llena
- En un vehículo en movimiento

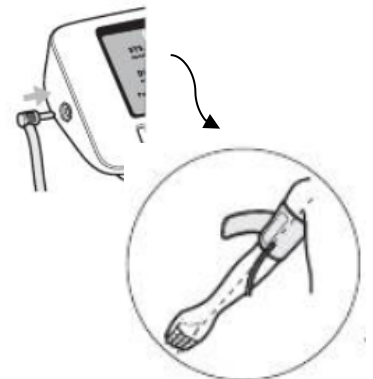
La presión de la sangre es menor que lo normal:

- Después de tomar un baño
- Después de beber alcohol

5. El manejo incorrecto de las pilas puede provocar la ruptura de las mismas. Tenga en cuenta lo siguiente para garantizar el uso correcto de las pilas.
 - a. Apague el tensiómetro después de utilizarlo.
 - b. No mezcle pilas de diferentes tipos ni tamaños.
 - c. Cambie todas las pilas al mismo tiempo. No mezcle pilas viejas con nuevas.
 - d. Inserte las pilas con la polaridad correcta, según las instrucciones.
 - e. Extraiga las pilas cuando estén agotadas y deséchelas según las disposiciones ambientales vigentes

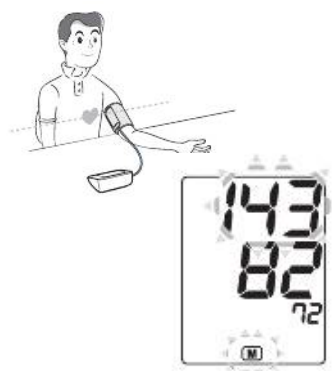
Ajuste del cuff de presión

1. Insertar la punta infladora en el conector de la bomba
2. Posteriormente ajuste el cuff al antebrazo, como se muestra en la imagen.



3. Posteriormente, descansa el codo sobre una mesa y atienda las recomendaciones mencionadas en este manual.

4. Prenda el dispositivo en el botón *On/Off*, espere a que se efectué la medición de PA. Usted visualizara en la pantalla los valores de PA.



5. Para almacenar la medición obtenida en la memoria del dispositivo y en la base de datos de Firebase, debe botón de memoria.



oprimir el

6. Finalmente, apague el dispositivo y desconecte los elementos para almacenarlos de forma segura y prolongar la vida útil de los mismos.



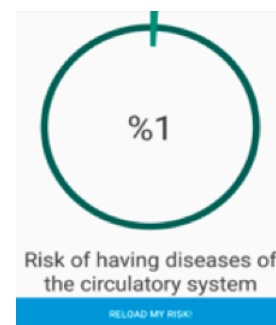
7. El registro de PA ya se encuentra disponible en Firebase. También puede ser visible en la app de eHeartBP.

```
-L_3fH_uMp6XLLPiloDP
  |
  | diastolic: 69
  | pulse: 77
  | systolic: 137
  | timestamp: 1542315847172
```

8. En el aplicativo, en la sección del menú principal se puede encontrar la opción para evaluar los registros almacenados en la DB y obtener una predicción sobre un posible episodio de hipertensión arterial.

SYS.	DIA.	PULSO
130	83	86
mmHg/KPa	mmHg/KPa	Pulse / min

9. Es importante reiterar que la utilización del dispositivo se debe enfocar con fines preventivos sobre la atención que se debe prestar a esta condición cardiaca. Por otra parte, del proyecto se orienta a personas con factores de riesgo de padecer hipertensión arterial, por tal motivo es imprescindible atender las recomendaciones médicas.



Adicionalmente, el uso de esta herramienta no pretende suplir o proporcionar un dictamen médico profesional, sino por el contrario estimular el autocuidado y los buenos hábitos de salud en el usuario.

Bibliografía

- [1] W. H. O. WHO Press, "Prevention of Cardiovascular Disease Guidelines for assessment and management of cardiovascular risk WHO Library Cataloguing-in-Publication Data," *WHO Library Cataloguing-in-Publication Data Prevention*, Geneva, Switzerland, pp. 2–18, 2007.
- [2] F. M. I. Javier, "Promocion de la salud cardiovascular, prevención y control de la diabetes, la obesidad y el sobrepeso en Colombia," Ministerio de Salud y Protección Social de Colombia, Colombia, 2017.
- [3] B. Y. Escobar Restrepo, "Sistema para la medición de la presión arterial continua no invasiva sin brazalete," vol. 1, pp. 25–50, Apr. 2014.
- [4] WHO, "Technical package for cardiovascular disease management in primary health care," Geneva, Switzerland, 2016.
- [5] N. G. Bolivar Pulgarin, L. D. Cangrejo Aljure, and O. J. Salcedo Parra, "eHeart-BP, Prototype of the Internet of Things to Monitor Blood Pressure," in *2019 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, 2019, pp. 58–63.
- [6] J. E. Sociedad Colombiana de Cardiología., *Revista colombiana de cardiología : órgano oficial de la Sociedad Colombiana de Cardiología.*, vol. 19, no. 6. Sociedad Colombiana de Cardiología, 2012.
- [7] F. Paganelli and D. Giuli, "An Ontology-Based Context Model for Home Health Monitoring and Alerting in Chronic Patient Care Networks," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*, 2007, pp. 838–845.
- [8] P. Unnikrishnan, D. K. Kumar, S. Poosapadi Arjunan, H. Kumar, P. Mitchell, and R. Kawasaki, "Development of Health Parameter Model for Risk Prediction of CVD Using SVM.," *Comput. Math. Methods Med.*, vol. 2016, p. 3016245, 2016.
- [9] J.-C. Hsieh, A.-H. Li, and C.-C. Yang, "Mobile, Cloud, and Big Data Computing: Contributions, Challenges, and New Directions in Telecardiology," *Int. J. Environ. Res. Public Health*, vol. 10, no. 11, pp. 6131–6153, Nov. 2013.
- [10] S. J. Hsu, S. S. Lin, T. W. Pai, and H. Fujita, "Proactive healthcare and an early warning mechanism for coronary artery disease patients using Internet-of-Thing devices," in *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, 2017, pp. 1400–1405.

- [11] S. Hijazi, A. Page, B. Kantarci, and T. Soyata, "Machine Learning in Cardiac Health Monitoring and Decision Support," *Computer (Long. Beach. Calif.)*, vol. 49, no. 11, pp. 38–48, 2016.
- [12] A. Acharya, "Comparative Study of Machine Learning Algorithms for Heart Disease Prediction," *Thesis Inf. Technol.*, no. April, p. 28, 2017.
- [13] A. Wang, N. An, Y. Xia, L. Li, and G. Chen, "A Logistic Regression and Artificial Neural Network-Based Approach for Chronic Disease Prediction: A Case Study of Hypertension," in *2014 IEEE International Conference on Internet of Things(iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 2014, pp. 45–52.
- [14] P. Unnikrishnan, D. K. Kumar, S. Poosapadi Arjunan, H. Kumar, P. Mitchell, and R. Kawasaki, "Development of Health Parameter Model for Risk Prediction of CVD Using SVM.," *Comput. Math. Methods Med.*, vol. 2016, p. 3016245, 2016.
- [15] A. R. Kumar, "Difference between Machine Learning, Deep Learning and Artificial Intelligence," *Udacity*, 2018. [Online]. Available: <https://medium.com/@UdacityINDIA/difference-between-machine-learning-deep-learning-and-artificial-intelligence-e9073d43a4c3>. [Accessed: 05-Nov-2019].
- [16] S. K. Sen, "Predicting and Diagnosing of Heart Disease Using Machine Learning Algorithms," *Int. J. Eng. Comput. Sci.*, 2017.
- [17] N. Gupta, N. Ahuja, S. Malhotra, A. Bala, and G. Kaur, "Intelligent heart disease prediction in cloud environment through ensembling," *Expert Syst.*, vol. 34, no. 3, 2017.
- [18] D. Singh, G. Tripathi, and A. J. Jara, "A survey of Internet-of-Things: Future vision, architecture, challenges and services," in *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, 2014, pp. 287–292.
- [19] A. Čolaković and M. Hadžialić, "Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues," *Computer Networks*, vol. 144. Elsevier B.V., pp. 17–39, 24-Oct-2018.
- [20] L. Rose, Karen; Eldridge, Scott; Chapin, "La internet de las cosas — Una breve reseña," *Internet Society*, vol. 1, Geneva, Switzerland, pp. 63–112, 2015.
- [21] A. M. Khairuddin, K. N. F. Ku Azir, and P. E. Kan, "Limitations and future of electrocardiography devices: A review and the perspective from the Internet of Things," in *International Conference on Research and Innovation in Information Systems, ICRIS*, 2017, pp. 2–4.
- [22] A. Ghosh, "Personal Healthcare Agents for Monitoring and Predicting Stress and Hypertension from Biosignals." Università degli Studi di Trento, Trento TN, Italia, pp. 22–40, 2017.
- [23] J. C. Hsieh, A. H. Li, and C. C. Yang, "Mobile, cloud, and big data computing: Contributions, challenges, and new directions in telecardiology," *International Journal of Environmental Research and Public Health*, vol. 10, no. 11.

- Multidisciplinary Digital Publishing Institute, pp. 6131–6153, 13-Nov-2013.
- [24] C. A. Alexander and L. Wang, “Big Data Analytics in Heart Attack Prediction,” *J. Nurs. Care*, vol. 06, no. 02, 2017.
- [25] P. Fernando and V. V. Asorey, “Claves para abordar el diseño metodológico » Maestría Diseño Comunicacional | diCom,” 2013. [Online]. Available: <https://maestriadicom.org/articulos/claves-para-abordar-el-diseno-metodologico/>. [Accessed: 28-Oct-2019].
- [26] “PANASONIC EW3106 OPERATING INSTRUCTIONS MANUAL Pdf Download.” [Online]. Available: <https://www.manualslib.com/manual/744498/Panasonic-Ew3106.html>. [Accessed: 27-Nov-2019].
- [27] S. W. Liou, D. Kurniadi, B. R. Zheng, W. Q. Xie, C. J. Tien, and G. J. Jong, “Classification of biomedical signal on IoT platform using support vector machine,” in *Proceedings of 4th IEEE International Conference on Applied System Innovation 2018, ICASI 2018*, 2018, pp. 50–53.
- [28] S. Eduardo, “Hacking a Blood Pressure Monitor – Edu Steinhorst,” 2015. [Online]. Available: <https://www.edusteinhorst.com/hacking-a-blood-pressure-monitor/>. [Accessed: 30-Oct-2019].
- [29] C. É. Rodríguez, “Telesalud en Colombia,” *Acreditación en Salud ICONTEC*, p. 5, 2011.
- [30] P. Borges, P. Monteiro, and R. J. Machado, “Tailoring RUP to small software development teams,” in *Proceedings - 37th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2011*, 2011, pp. 306–309.
- [31] S. Gayathri, A. S. Priyadarshini, and P. T. V. Bhuvaneswari, “Multivariate linear regression based activity recognition and classification,” in *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 2014, pp. 1–6.
- [32] V. Boyarshinov, “Machine Learning,” *Computer*, vol. 2005, no. April. Universidad de Stanford, p. 414, 1997.
- [33] G. Ranjan, “Global Cost Minimum,” *Stack Exchange*, 2019. [Online]. Available: <https://datascience.stackexchange.com>.
- [34] M. M. Aborokbah, S. Al-Mutairi, A. K. Sangaiah, and O. W. Samuel, “Adaptive context aware decision computing paradigm for intensive health care delivery in smart cities—A case analysis,” *Sustain. Cities Soc.*, vol. 41, pp. 919–924, 2018.