



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL**

CARRERA DE TELECOMUNICACIONES

Tema:

**SISTEMA DE MONITOREO PARA LA DETECCIÓN DE PERSONAS
HIPERTENSAS EMPLEANDO INTELIGENCIA ARTIFICIAL.**

Trabajo de Integración Curricular Modalidad: Proyecto de Investigación, presentado
previo a la obtención del título de Ingeniero en Telecomunicaciones.

ÁREA: Electrónica

LÍNEA DE INVESTIGACIÓN: Sistemas Electrónicos

AUTOR: Juan Carlos Bayas Bustos

TUTOR: Ing. Santiago Manzano, Mg.

Ambato – Ecuador

marzo – 2023

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Integración Curricular con el tema: SISTEMA DE MONITOREO PARA LA DETECCIÓN DE PERSONAS HIPERTENSAS EMPLEANDO INTELIGENCIA ARTIFICIAL, desarrollado bajo la modalidad Proyecto de Investigación por el señor Juan Carlos Bayas Bustos, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del Título de Tercer Nivel, de Grado en la Universidad Técnica de Ambato y el numeral 7.4 del respectivo instructivo del reglamento.

Ambato, marzo 2023.

.....

Ing. Víctor Santiago Manzano Villafuerte, Mg.

TUTOR

AUTORÍA

El presente trabajo de Integración Curricular titulado: SISTEMA DE MONITOREO PARA LA DETECCIÓN DE PERSONAS HIPERTENSAS EMPLEANDO INTELIGENCIA ARTIFICIAL es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, marzo 2023.



Juan Carlos Bayas Bustos

C.C. 1805313846

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Integración Curricular como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Integración Curricular en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2023.



Juan Carlos Bayas Bustos

C.C. 1805313846

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Integración Curricular presentado por el señor Juan Carlos Bayas Bustos, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE MONITOREO PARA LA DETECCIÓN DE PERSONAS HIPERTENSAS EMPLEANDO INTELIGENCIA ARTIFICIAL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 de las segundas reformas al Reglamento para la ejecución de la Unidad de Integración Curricular y la obtención del título de tercer nivel, de grado en la Universidad Técnica de Ambato y al numeral 7.6 del respectivo instructivo del reglamento. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2023.

Ing. Elsa Pilar Urrutia Urrutia, Mg.
PRESIDENTE DEL TRIBUNAL

Ing. Marco Antonio Jurado Lozada, Mg.
PROFESOR CALIFICADOR

Ing. Juan Pablo Pallo Noroña, Mg.
PROFESOR CALIFICADOR

Dedicatoria

Dedicado a Dios por todo el apoyo y cuidado que siempre me ha entregado, a mis padres Carlos y Graciela, que jamás me dejan solo sin importar mis errores y dificultades, a mis hermanos David y Darío que siempre me apoyan y sienten alegría por mis triunfos, a mis abuelos Martín y Luz, a mis tíos, Roberto, María y Teresa, que gracias a sus consejos y motivaciones estoy cumpliendo cada una de mis metas.

A la mujer la cual durante toda mi carrera ha servido de consejera, compañera, amiga y pareja, Karen.

Juan Carlos Bayas Bustos

Agradecimiento

Agradezco principalmente a Dios por haberme dotado de fuerza para seguir adelante, sabiduría para poder tomar decisiones correctas, y la capacidad de valorar todo lo que la vida me ofrece.

A mi familia, en especial a mis padres, que nunca se rindieron trabajando y educándome, dándome siempre lo mejor, madre mía y padre amado, todo esto es gracias a ustedes.

Al Ingeniero Santiago Manzano, cuyo apoyo y ayuda, además de su ejemplo han servido de guía y motivación para seguir adelante.

A mi novia Karen, que siempre ha estado conmigo a pesar de todo, motivándome, apoyándome y buscando siempre lo mejor de mí. A mis amigos William, Dennis y Christian por su apoyo.

A la persona a la cual admiro y me ha sabido cuidar, apoyar y motivar, Paulo Acosta.

Juan Carlos Bayas Bustos

ÍNDICE GENERAL

PORTADA	<i>i</i>
APROBACIÓN DEL TUTOR	<i>ii</i>
AUTORÍA	<i>iii</i>
DERECHOS DE AUTOR	<i>iv</i>
APROBACIÓN DEL TRIBUNAL DE GRADO	<i>v</i>
Dedicatoria	<i>vi</i>
Agradecimiento	<i>vii</i>
ÍNDICE GENERAL	<i>viii</i>
ÍNDICE DE FIGURAS	<i>xii</i>
ÍNDICE DE TABLAS	<i>xiv</i>
RESUMEN EJECUTIVO	<i>xv</i>
ABSTRACT	<i>xvi</i>
CAPÍTULO I	<i>1</i>
MARCO TEÓRICO	<i>1</i>
1.1 Antecedentes Investigativos	<i>1</i>
1.2 Contextualización del problema	<i>3</i>
1.3 Fundamentación teórica	<i>4</i>
1.3.1 Presión Arterial	<i>4</i>
1.3.2 Medición de la presión arterial.....	<i>5</i>
1.3.3 Hipertensión Arterial.....	<i>8</i>
1.3.4 Diagnóstico de Hipertensión Arterial.....	<i>9</i>
1.3.5 Sensores	<i>11</i>
1.3.6 Sensores Biométricos	<i>14</i>
1.3.7 Fotoplestimografía (PPG)	<i>15</i>

1.3.8	Electrocardiogramas (ECG).....	16
1.3.9	IEEE Std 1708-2014	17
1.3.10	Placa de Desarrollo	18
1.3.11	Inteligencia Artificial	18
1.3.12	Aprendizaje Automático	19
1.3.13	Aprendizaje Profundo	20
1.3.14	Árboles De Decisión	21
1.3.15	Lenguajes de Programación	22
1.3.16	Python	24
1.3.17	Base de Datos	25
1.4	Objetivos	26
1.4.1	Objetivo general	26
1.4.2	Objetivos específicos	26
<i>CAPÍTULO II.....</i>		27
<i>METODOLOGÍA.....</i>		27
1.5	Materiales.....	27
1.6	Métodos	27
1.6.1	Modalidad de la Investigación	27
1.6.1.1	Investigación Aplicada.....	27
1.6.1.2	Investigación Bibliográfica	27
1.6.1.3	Investigación Experimental.....	27
1.6.1.4	Investigación de Campo	28
1.6.2	Recolección de Información	28
1.6.3	Procesamiento y Análisis de Datos	28
1.6.4	Propuesta de Solución	28
1.6.5	Desarrollo del Proyecto.....	29
<i>CAPÍTULO III.....</i>		30
<i>RESULTADOS Y DISCUSIONES.....</i>		30
1.7	Análisis y discusión de los resultados	30
1.8	Análisis de Factibilidad	30

1.8.1	Factibilidad Económica.....	30
1.8.2	Factibilidad Bibliográfica.....	30
1.9	Desarrollo de la propuesta.....	30
1.9.1	Requerimiento del Prototipo	30
1.9.2	Diagrama de bloques del sistema.....	31
1.9.3	Selección de los elementos para la implementación del sistema	32
1.9.4	Sensor de Signos Vitales	32
1.9.5	Placa de Desarrollo	37
1.9.6	Microcontrolador.....	41
1.9.7	Estructura del dispositivo electrónico para la toma de datos	45
1.9.8	Calibración del Sensor AS7038RB	46
1.9.9	Adquisición de Datos	48
1.9.10	Asistente Virtual.....	49
1.9.11	Clasificadores con Inteligencia Artificial.....	54
1.9.12	Clasificador para Personas Hipertensas	54
1.9.13	Clasificador para Riesgos Cardíacos.....	57
1.9.14	Clasificador para Riesgos de Derrame Cerebral	59
1.9.15	Visualización del Diagnóstico.....	60
1.9.18	Almacenamiento de Datos	63
1.10	Resultados	66
1.11	Pruebas de funcionamiento	67
1.12	Presupuesto.....	69
<i>CAPÍTULO IV</i>		70
<i>CONCLUSIONES Y RECOMENDACIONES</i>		70
1.13	Conclusiones	70
1.14	Recomendaciones.....	71
<i>BIBLIOGRAFÍA</i>.....		73
<i>ANEXOS</i>.....		79
ANEXO 1: Código fuente de entrenamiento del clasificador para hipertensión.		80

ANEXO 2: Código fuente de entrenamiento del clasificador para accidentes cardiovasculares.	83
ANEXO 3: Código fuente de entrenamiento del clasificador para accidentes cerebrales.	86
ANEXO 4: Código fuente correspondiente al Asistente Virtual	89
ANEXO 5: Ensamblaje de la estructura para los módulos.....	111

ÍNDICE DE FIGURAS

Figura 1 Medición de la Presión Arterial Invasiva.....	6
Figura 2 Medición de la Presión Arterial empleando un Esfigmomanómetro.	7
Figura 3 Tensiómetro Digital Para Medir Presión Arterial empleando el método oscilométrico.....	7
Figura 4 Hipertensión Arterial y sus Efectos en la salud del Paciente.	9
Figura 5 Sensor Inteligente	13
Figura 6 Diseño de un sensor MEMS	14
Figura 7 Señal PPG.....	15
Figura 8 Captura para Fotoplestimografía	16
Figura 9 Componentes de la Señal PPG	16
Figura 10 Electrocardiograma ECG y sus características	17
Figura 11 Clases de Aprendizaje Automático	20
Figura 12 Redes Neuronales	20
Figura 13 Composición de un árbol de decisión	21
Figura 14 Componentes de un Sistema de administración de base de datos.....	25
Figura 15 Diagrama de Bloques del Sistema.....	31
Figura 16 Sensor AS7038RB.	35
Figura 17 Pines de Conexión Del Sensor AS7038RB.	36
Figura 18 Placa de Desarrollo AS703XX	39
Figura 19 SMOD de los sensores AS703XX.	40
Figura 20 Puerto COM asignado al Microcontrolador Argon.	45
Figura 21 Diseño de la Estructura del Dispositivo Electrónico.	46
Figura 22 Software para la calibración y evaluación del Sensor AS7038RB.	46
Figura 23 PPG Obtenida por el Sensor AS7038RB.	47
Figura 24 Código para la comunicación serial.....	48
Figura 25 Interfaz Gráfica del Asistente Virtual.	50
Figura 26 Botones dentro de la interfaz gráfica.....	50
Figura 27 Función para la interpretación auditiva del asistente virtual	51
Figura 28 Función para la comunicación con el asistente virtual	51
Figura 29 Segunda Ventana del Asistente Virtual.....	52

Figura 30 Asignación de etiquetas para la ventana de diagnóstico.....	53
Figura 31 Reporte Final del Asistente Virtual.	53
Figura 32 Base de Datos para el Entrenamiento Del Clasificador de Personas Hipertensas.	55
Figura 33 Precisión del Algoritmo Clasificador de Personas Hipertensas.	56
Figura 34 Obtención del Clasificador de Personas Hipertensas.	56
Figura 35 Base de Datos para el Clasificador de Riesgo Cardiovascular.....	57
Figura 36 Datos Para el Entrenamiento del Clasificador para Riesgos Cardiovasculares.	58
Figura 37 Precisión del Clasificador Para Riesgos Cardiovasculares.....	58
Figura 38 Modelo Entrenado Riesgos Cardiovasculares.....	59
Figura 39 Base de Datos Para el Clasificador De Accidentes Cerebrales.	59
Figura 40 Precisión del Clasificador Para Riesgos Cerebrales.....	60
Figura 41 Visualización Del Diagnóstico.....	63
Figura 42 Creación de la Base de Datos	64
Figura 43 Almacenamiento en la Base de Datos.....	64
Figura 44 Datos tomados por el sistema de monitoreo	66

ÍNDICE DE TABLAS

Tabla 1 Parámetros para la Medición de la Presión Arterial.	8
Tabla 2 Clasificación de la Presión Arterial en Consultas Médicas.	9
Tabla 3 Características de los Sensores.	11
Tabla 4 Aplicaciones de la IA.	19
Tabla 5 Árboles de Decisión.	22
Tabla 6 Generaciones de Lenguaje de Programación.	22
Tabla 7 Sistemas de Gestión Para Bases de Datos.	25
Tabla 8 Sensores de Signos Vitales.	33
Tabla 9 Características Técnicas del Sensor De Signos Vitales AS7038RB.	35
Tabla 10 Descripción de Pines del AS7038RB.	36
Tabla 11 Placas de Desarrollo para el Sensor AS7038RB.	38
Tabla 12 Características de la placa AS703XX.	40
Tabla 13 Ejemplos de Microcontroladores.	41
Tabla 14 Modelos Particle y sus Características.	43
Tabla 15 Características del Microcontrolador Particle Argon.	44
Tabla 16 Valores de Presión Obtenidos con el Sensor AS7038RB y el Esfingomanómetro.	48
Tabla 17 Librerías Empleadas para el Asistente Virtual.	49
Tabla 18 Función de los botones de la Segunda Ventana.	52
Tabla 19 Ventajas y Desventajas de los Árboles de Decisión.	54
Tabla 20 Clasificación de los Casos de Presión Arterial.	55
Tabla 21 Variables Para los Modelos De Predicción.	61
Tabla 22 Parámetros Para La Tabla de Base de Datos.	65
Tabla 23 Medidas De Presión Durante La Mañana.	67
Tabla 24 Medidas De Presión Durante La Tarde.	67
Tabla 25 Medidas De Presión Durante La Noche.	68
Tabla 26 Tabla de Presupuesto.	69

RESUMEN EJECUTIVO

La hipertensión arterial se considera el enemigo silencioso del mundo entero, se caracteriza por ser una patología que no presenta síntomas y causa millones de muertes alrededor del mundo, de todas las personas que padecen hipertensión, aproximadamente solo un 46% lo sabe. Por esta razón, el presente proyecto se centra en implementar un sistema de monitoreo para la detección de personas hipertensas empleando inteligencia artificial.

El presente proyecto describe el uso de un dispositivo electrónico para la medición de presión arterial, el elemento electrónico cumple con ser un equipo no invasivo para el cuerpo humano. Los datos que se adquieran serán analizados por un asistente virtual que está programado para emitir diagnósticos empleando modelos entrenados de inteligencia artificial.

El funcionamiento del sensor principal encargado de tomar las medidas de presión arterial se basa en la fotoplestimografía, que básicamente emite haces de luz hacia la piel y analiza que tanto de esta luz fue absorbida. Los datos son recolectados por una tarjeta electrónica y enviados mediante comunicación serial USB hacia el asistente virtual programado en Python, se emplean librerías de uso libre y con excelentes respuestas de funcionamiento. El asistente realizará una serie de preguntas al paciente para conocer más sobre su estilo de vida y analizar mediante el uso de inteligencia artificial estos datos.

Como resultado final se obtiene un asistente virtual capaz de detectar cuando una persona padece hipertensión, y de ser el caso a que categoría dentro de esa patología pertenece. Este asistente emitirá un diagnóstico completo, adicionando las probabilidades existentes de que el paciente pueda sufrir algún accidente cardiovascular o cerebral, basándose en las respuestas que se ha ingresado, finalmente esta información se almacena en una base de datos.

Palabras clave: HRM, fotoplestimografía, sistólica, diastólica, reconocimiento, automatización.

ABSTRACT

Arterial hypertension is considered the silent enemy of the entire world, it is characterized by being a pathology that does not present symptoms and causes millions of deaths around the world, of all the people who suffer from hypertension, approximately only 46% know it. For this reason, this project focuses on implementing a monitoring system for the detection of hypertensive people using artificial intelligence.

This project describes the use of an electronic device for the measurement of blood pressure, the electronic element complies with being a non-invasive equipment for the human body. The data that is acquired will be analyzed by a virtual assistant that is programmed to issue diagnoses using trained artificial intelligence models.

The operation of the main sensor in charge of taking blood pressure measurements is based on photoplethysmography, which basically emits light beams towards the skin and analyzes how much of this light was absorbed. The data is collected by an electronic card and sent via USB serial communication to the virtual assistant programmed in Python, free-use libraries are used with excellent performance responses. The assistant will ask the patient a series of questions to learn more about her lifestyle and analyze this data using artificial intelligence.

As a final result, a virtual assistant capable of detecting when a person suffers from hypertension is obtained, and if so, to which category within that pathology it belongs. This assistant will issue a complete diagnosis, adding the existing probabilities that the patient may suffer a cardiovascular or cerebral accident, based on the answers that have been entered, finally this information is stored in a database.

Keywords: HRM, photoplethysmography, systolic, diastolic, recognition, automation.

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes Investigativos

La investigación presentada fue realizada partiendo de la información procesada de los distintos repositorios de las entidades educativas de nivel superior, tanto nacionales como internacionales, de la misma manera se ha procesado información proveniente de artículos científicos, informes de trabajos de investigación y reportes de salud referentes a la hipertensión.

En el año 2016, Franklin Placencia, de la Universidad Técnica de Ambato, publica una investigación con el tema: “Indumentaria Electrónica Para el Monitoreo Continuo de Señales Vitales”, misma que desarrolla una camiseta capaz de tomar signos vitales básicos y enviarlos a una aplicación. El sistema se desarrolla en base a dispositivos denominados “Wearables”, que son equipos electrónicos que emplean como vestimenta, conocidos también como vestimenta tecnológica, y que mediante sensores y comunicación inalámbrica envían los datos hacia una aplicación con interfaz gráfica amigable. La investigación también implementa un sistema de medición para la presión arterial, no obstante, el tema de la sensibilidad y confiabilidad no es algo que se tome muy en cuenta [1].

Uno de los indicadores fundamentales para que una mujer en estado de gestación pueda sufrir de preclamsia es padecer de hipertensión, por esta razón en el año 2018, Glenda Puco de la Universidad Técnica de Ambato desarrolla un sistema electrónico que permite detectar y monitorear esta patología en las mujeres en estado de gestación. El sistema toma medidas constantes de variables de presión arterial y las almacena en una base de datos, para posteriormente generar informes basándose en los valores de las últimas fechas [2].

En el mismo año, 2018, Diana Gonzalez Vargas y Dayana Vargas Vindas investigadoras de la Universidad Latina de Costa Rica emplean un brazalete creado por Blumio, para poder obtener el nivel de presión arterial en una persona, haciendo mucho énfasis en el hecho de la factibilidad de que cada vez más se sigan incorporando esta clase de dispositivos. En dicho documento, las investigadoras analizan el método empleado por el brazalete y determinan cuáles deberían ser los rangos de presión que

debería manejar el mismo, brindando una idea novedosa y aplicable dentro lo que es la biotecnología [3].

En el año 2020, un grupo de investigadores de la Universidad Don Bosco y la Universidad Centroamericana José Simeón Cañas proponen como solución un dispositivo de bajo costo que permite la adquisición de datos fisiológicos como lo son temperatura corporal, presión y pulso arterial, estos datos son enviados al dispositivo celular móvil con el objetivo de poder visualizarlos [4].

En el mismo año, 2020, Miguel Herrera junto con Eithel Meza de la Universidad Peruana de Ciencias Aplicadas, publican una tesis otorgando una solución tecnológica basada en factores de riesgo para el diagnóstico de Hipertensión Arterial, en donde emplean dispositivos wearables para enviar la información y que esta sea almacenada en una base de datos para futuras interpretaciones médicas [5].

María José De Mitri de la Universidad Católica de Córdoba, en el año 2019, expone una investigación médica que hace uso de la inteligencia artificial, como un factor clave para la predicción de marcadores cardíacos. La misma se basa en un estudio a 255 pacientes, entre los cuales se hallan sujetos que padecen diabetes o hipertensión, los cuales a su vez están bajo la prescripción médica donde deben ingerir ciertos medicamentos, cada uno de estos medicamentos poseen distintas características químicas que altera la variabilidad de los niveles de biomarcadores, dependiendo del tipo de medicamento suministrado y la enfermedad que padecen. En esta investigación el factor clave recae en la inteligencia artificial, ya que la misma ayudará al personal médico a definir el comportamiento del biomarcador, para de esta forma poder inclinar al paciente a que se le realice estudios médicos complejos [6].

Una vez que se ha recopilada la información obtenida de los antecedentes investigativos y como aporte al proyecto se concluye que para un sistema de monitoreo de presión arterial empleando inteligencia artificial es necesario el uso de un sensor no invasivo que comunique los datos, con tasas de errores aceptables, hacia un sistema donde puedan ser visualizados e interpretados, por otra parte, para la inteligencia artificial es necesario el uso de datos previamente organizados, a fin de que los algoritmos a emplearse puedan ser entrenados con esta información.

1.2 Contextualización del problema

Según la Fundación Española del Corazón (FEC), en su informe indica que durante cada año que transcurre la hipertensión mata a 7.5 millones de personas a nivel mundial, lo que se traduce a un 13% de las muertes totales que se producen a nivel global, esto según la Organización Mundial de la Salud (OMS). De estos datos según el Dr. Enrique Galve, encargado de la presidencia de la Sección de Riesgo Vascular y Rehabilitación Cardíaca de la Sociedad Española de Cardiología, manifiesta que el número total de personal a nivel mundial que padece esta patología es de aproximadamente un billón y medio [7].

Al momento de tomar una medida de presión arterial, es necesaria la valoración de un médico para aportar con un diagnóstico precoz, y de esta forma detectar a un paciente asintomático y prevenir futuros problemas. En todo este proceso se ve involucrado necesariamente un profesional de la salud para interpretar los datos obtenidos, sin esta intervención, las mediciones de la presión arterial de un paciente no se pueden descifrar, o no dan una idea clara del estado actual del individuo [8].

El proyecto de investigación se divide en dos partes, la primera se encarga de la toma de datos, empleando un dispositivo no invasivo, el cual debe tener la característica principal de entregar una medida con una tasa de error aceptable para aplicaciones médicas reales, además de poseer una tolerancia propia de un dispositivo de uso comercial y profesional, para garantizar la fiabilidad del dispositivo. La segunda parte es la recepción y procesamiento de datos empleando un asistente virtual, este tendrá la capacidad de emitir un diagnóstico efectivo que no requiera la presencia de un médico o personal capacitado.

El beneficio principal del proyecto es tener una perspectiva clara de la situación actual de las personas con respecto a la hipertensión, y que mediante los datos interpretados por una inteligencia artificial se pueda determinar si se requieren acciones correctivas o preventivas, o por otro lado requiera tratamiento médico urgente. En consecuencia, el objetivo de este proyecto es reducir la tasa de muertes producida por esta patología, a la vez que de mantener en los individuos censados un mejor estilo de vida con controles regulares que no requieran la presencia de un médico profesional, o a su vez

acudir a hospitales o centros especializados en el tema para conocer la situación actual de salud.

Para la elaboración del dispositivo de medición de presión arterial se considera el uso de un sensor que cumpla con especificaciones IEEE para uso comercial y aplicaciones en medicina, así como circuitos electrónicos que permitan el procesamiento de las mediciones obtenidas, las cuales serán almacenadas e interpretadas por una inteligencia artificial para poder determinar la hipertensión en los pacientes.

1.3 Fundamentación teórica

1.3.1 Presión Arterial

La presión arterial es cuando la sangre fluye por el cuerpo y se ve impulsada por el corazón, circulando por todo el sistema de arterias, la cual está sometida por una presión, o a su vez también conocida como tensión arterial. La presión arterial tiene una medida, la cual es expresada en milímetros de mercurio, o mejor conocida por siglas como “mmHg”, sin embargo, dicha tensión arterial se ve compuesta por dos factores más, estos son: la presión arterial sistólica, PAS, y la presión arterial diastólica, PAD.

La presión arterial sistólica hace referencia al impulso generado por el corazón cuando se contrae el ventrículo izquierdo, comúnmente se conoce esta presión como “alta”. La presión arterial diastólica por otro lado se considera como aquella resistencia que presentan las arterias del cuerpo humano hacia el paso de la sangre, también es conocida como “baja” [9].

En el cuerpo humano, la sangre fluye por medio del sistema arterial, el mismo está constituido por grandes arterias que poseen una gran capacidad, las cuales están coordinadas con el ritmo cardíaco, estas tienen la característica de disminuir, aflojar o relajar la tensión en sus paredes con cada oleada de sangre, las mismas se ven impulsadas por el corazón, específicamente por una parte del mismo que se conoce como ventrículo izquierdo, que a su vez cumple la función de amortiguar su presión, como el corazón se contrae cada vez que termina su ciclo, esto permite que la sangre fluya hacia arterias que son más pequeñas y que por su naturaleza oponen resistencia.

Finalmente, la sangre llega a los capilares, que son como pequeñas terminales de un árbol vascular, pero en este punto la presión de la misma llega muy amortiguada, en esta parte del sistema sanguíneo se realiza un censo de verificación del paso de oxígeno, así como los nutrientes que llegan a los tejidos.

La presión arterial se ve relacionada fundamentalmente con el trayecto y flujo de la sangre por el cuerpo, y depende de dos partes: la primera que es la cantidad o volumen de sangre que suministra el corazón, la cual se considera en unidad de tiempo, y la segunda que es la oposición que presentan las arterias al paso de la sangre, y donde fundamentalmente se tiene en consideración las arteriolas o dicho de otra manera las arterias de menor tamaño.

Sin embargo, existen otros factores que de una u otra manera influyen sobre la presión arterial, así pues, se tiene la actividad del sistema nervioso autónomo, que se encarga de controlar el ritmo cardíaco y también la oposición que presentan las arterias pequeñas. El volumen de la sangre es otro factor clave, y este tiene relación con el riñón, el cual actúa conforme a los niveles suministrados de agua y sal [9].

1.3.2 Medición de la presión arterial

La medición de la presión arterial en una persona da lugar a conocer como su sistema circulatorio se está comportando. En la actualidad existen varios métodos para lograr obtener la presión arterial, uno más efectivo que otro, se clasifican en dos grandes grupos, los métodos invasivos y los no invasivos. Los métodos invasivos se caracterizan especialmente porque se ingresa una sonda dentro del cuerpo humano, esto gracias a una intervención quirúrgica, que se encarga de registrar variables como la temperatura, el flujo sanguíneo, la presión arterial, entre otras. Una de las principales desventajas de este método radica en las variaciones que se generan por el paso de la sangre en el lugar donde se coloca la sonda o conocida como catéter, produciendo errores en la medición. Estos sistemas de medición invasiva generalmente se componen de monitores externos, interfaces de control y transductores, como se puede apreciar en la figura 1 [10].

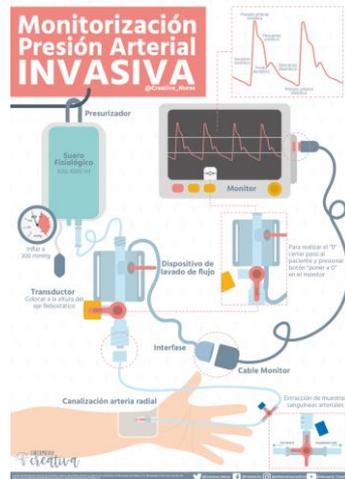


Figura 1 Medición de la Presión Arterial Invasiva [11]

Por otro lado, los métodos no invasivos, se caracterizan por utilizar generalmente equipos externos como brazaletes o sensores que se ubican en alguna extremidad del cuerpo. El método no invasivo más común usado para la toma de la presión arterial es el que emplea un esfigmomanómetro. Este elemento se coloca sobre la arteria del brazo, posteriormente, mediante una bomba de aire conocida como “manguito” se infla el brazalete, de esta manera se comprime de forma externa la arteria junto con cada uno de sus tejidos. La lógica que emplea para determinar la presión arterial se basa en el hecho que la presión ejercida sobre la arteria para cerrarla, es la misma a la existente dentro de ella.

Para poder medir la presión arterial se emplea un estetoscopio que se coloca dentro del brazalete, justo en la arteria. Cuando el brazalete tiende a desinflarse poco a poco se produce un ruido dentro de la arteria, y esto es lo que interpreta el médico junto con el manómetro para dar a conocer el valor de la presión arterial, como se puede apreciar en la figura 2. [12].

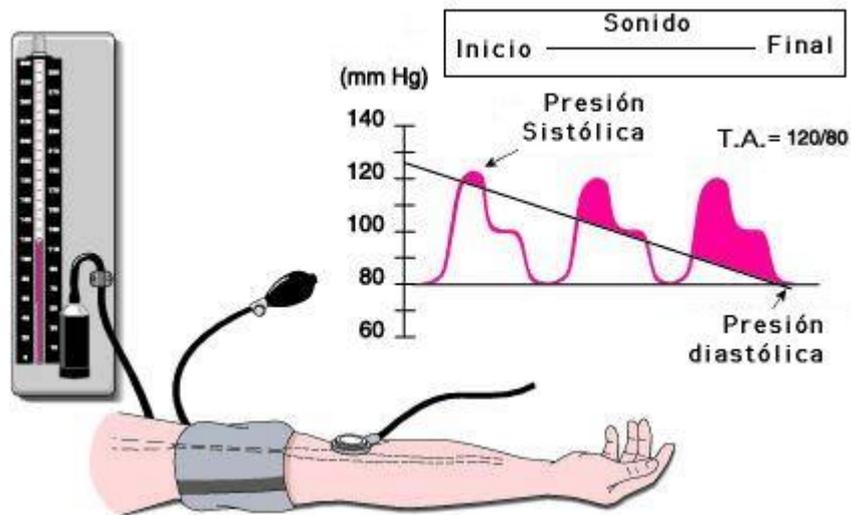


Figura 2 Medición de la Presión Arterial empleando un Esfigmomanómetro [13].

Este método es conocido como auscultatorio, sin embargo, también existe el método oscilométrico cuya diferencia radica en que este último no emplea el oído de un experto ni un estetoscopio, es netamente de los dispositivos electrónicos y usa sensores de presión que se encargan de analizar las oscilaciones que se produce cuando el brazalete se desinfla. Estos equipos son digitales y actualmente se los vende en muchas farmacias, tiendas médicas o electrónicas de forma libre, sin embargo, muchos de ellos no vienen homologados por lo que podrían entregar valores de presión arterial alejados de la realidad. En la figura 3 se puede observar un tensiómetro que emplea el método oscilométrico [14].



Figura 3 Tensiómetro Digital Para Medir Presión Arterial empleando el método oscilométrico [14].

Independientemente del método que se emplee para medir la presión arterial, se debe seguir los pasos presentados en la tabla 1 para realizar una buena toma de datos y evitar valores irreales [15].

Tabla 1 Parámetros para la Medición de la Presión Arterial [16].

Medición de la Presión Arterial
Mantener reposo durante 5 minutos antes de la toma de medidas.
Se deben registrar 3 mediciones con una diferencia de 1 a 2 minutos entre ellas, el valor final se calcula con un promedio entre las 2 últimas medidas.
En pacientes con arritmias cardíacas se deben considerar más mediciones.
El paciente debe permanecer con los pies apoyados.
Evitar apoyar las piernas.
La vejiga del paciente debe estar vacía.
Evitar comenzar charlas cuando se comience a tomar la presión arterial.
No se debe haber fumado o consumido licor antes de la consulta.

1.3.3 Hipertensión Arterial

La hipertensión arterial es una patología que se define como un incremento crónico de la tensión arterial o presión arterial, de tal forma que los valores de la presión sistólica es mayor o igual a 140 mmHg (milímetros de mercurio) y de presión diastólica es mayor o igual a 90 mmHg [16]. Los valores de presión arterial sistólica y diastólica se deben considerar siempre y cuando el paciente haya tomado un reposo de 5 minutos antes de la toma de datos, además de que el mismo se encuentre sin haber consumido café, alcohol, tabaco o cualquier sustancia que altere el ritmo cardíaco con una anticipación mínima de 30 minutos antes de la extracción de dichos valores de presión [17].

La hipertensión arterial, pertenece a las enfermedades cardiovasculares, además de que la misma está presente en la gran mayoría de las personas, y de forma más relevante en aquellos individuos que superan los 50 o 60 años de edad, indistintamente si son hombres o mujeres [18]. Si bien es cierto que la hipertensión arterial se define con el valor de dos números, esto no es todo, ya que la misma también se considera como un síndrome cardiovascular que avanza de manera progresiva, y que por consiguiente trae

muchas complicaciones al cuerpo humano. Padecer hipertensión arterial conlleva a generar causas muy graves que desencadenan en la muerte del paciente, por ejemplo, llega a producir retinopatía hipertensiva, que afecta al flujo de sangre en el ojo humano, produce accidentes cerebrovasculares como por ejemplo los conocidos “derrames cerebrales”, afecta al riñón generando insuficiencias, afecciones al corazón como insuficiencia cardíaca que puede producir paros, tal y como se puede observar en la figura 4. [19].

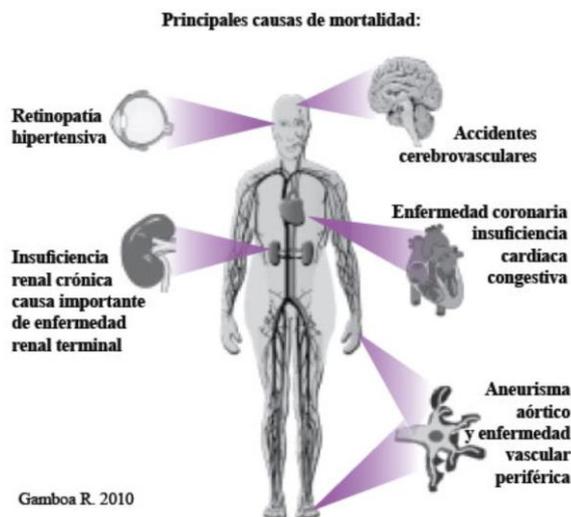


Figura 4 Hipertensión Arterial y sus Efectos en la salud del Paciente [19].

1.3.4 Diagnóstico de Hipertensión Arterial

Las medidas de la presión arterial se clasifican de acuerdo a la tabla 2. En ella se representa 7 posibles casos, en los cuales los 4 últimos corresponden a la clasificación de la hipertensión arterial.

Tabla 2 Clasificación de la Presión Arterial en Consultas Médicas [15].

Categoría	Presión Sistólica en mmHg	Presión Diastólica en mmHg
Óptima	< 120	<80
Normal	120-129	80-84
Normal-Alta	130-139	85-85
Hipertensión de Grado 1	140-159	90-99

Hipertensión de Grado 2	160-179	100-109
Hipertensión de Grado 3	≥ 180	≥ 110
Hipertensión Sistólica Aislada	≥ 140	< 90

La presión arterial elevada está estrictamente relacionada con complicaciones cardiovasculares y cerebrales en todos los grupos de edad. La presión arterial sistólica elevada se considera como una variable importante para predecir complicaciones futuras a partir de los 50 años. El aumento del riesgo de padecer patologías cardiovasculares cuando se tiene una presión diastólica elevada se ve presente en sujetos menores a 50 años. La hipertensión arterial comúnmente no se presenta sola, siempre se asocia a distintas variables que generan problemas cardiovasculares, por ende, se considera a la hipertensión como una puerta hacia esta enfermedad. Los factores que incrementan el riesgo cardiovascular en sujetos con hipertensión arterial son:

- Género (Influye más en los hombres que en mujeres)
- Edad
- Fumador
- Niveles de Colesterol
- Sobrepeso
- Familiares directos con registros de hipertensión
- En mujeres, cuando padecen de menopausia prematura
- Tipo de trabajo, economía, lugar de vivienda y todos los factores psicológicos y socioeconómicos
- Consumidor de Licor
- Sedentarismo
- Inactividad Física
- Ritmo Cardíaco Elevado

Estos factores sirven para predecir riesgos cardiovasculares a 10 años, el sistema que se emplea se conoce como SCORE, que usa bases de datos para realizar estas

predicciones. Las mujeres son más propensas a tener complicaciones cardiovasculares que los hombres, sin embargo, en los adultos mayores esta probabilidad incrementa mucho más [15].

1.3.5 Sensores

Un sensor es un dispositivo que se encarga de entregar un dato manipulable de una variable física que se ha medido. Estos dispositivos contienen transductores internamente, y solo admiten entradas. Existen dos clasificaciones para los sensores, activos y pasivos. Los sensores activos son aquellos que requieren de alimentación eléctrica para poder funcionar, por otro lado, los pasivos no necesitan dicha alimentación. Los parámetros que permiten diferenciar un sensor de otro se clasifican en estáticos, que no llegan a cambiar con el tiempo, y los dinámicos que describen a los sensores mientras avanza el tiempo. En la tabla 3 se describe cada una de estos parámetros [20].

Tabla 3 Características de los Sensores.

Características de los Sensores			
Estáticas		Dinámicas	
Sensitividad	Capacidad del sensor para poder procesar una señal mínima en su entrada.	Tiempo de Respuesta	Valor de tiempo en el cual el sensor puede registrar los cambios que se generan en la variable medida.
Rango	Valores máximos y mínimos que el sensor puede procesar.	Histéresis	Capacidad del sensor de poder seguir la curva ideal a los registros de salida.
Precisión	Estimación de la repetitividad con	Linealidad Dinámica	Capacidad del sensor para mantener la curva

	respecto a una medida.		de salida cuando se producen cambios bruscos en la variable censada.
Exactitud	Diferencia entre el valor real y el valor medido.	Error Dinámico	Producido por factores externos como por ejemplo el cambio en los valores de voltaje que alimenta al sensor.
Linealidad Estática	Condiciones ambientales en la que el sensor puede trabajar de mejor o de peor manera.		
Offset	Desplazamiento del eje de las ordenadas y de la curva de salida igualitario cuando se cumplen algunas condiciones de operación.		
Resolución	Capacidad del sensor de percibir los cambios más pequeños en las medidas.		

Error Estático	Fallas generadas al momento del registros de medidas.		
----------------	---	--	--

Fuente: Investigador

1.3.5.1 Sensores Inteligentes

Los sensores inteligentes nacen como respuesta a la necesidad de reducir el tamaño de las placas electrónicas y de integrar en un solo elemento varios dispositivos discretos. La IEEE define a un sensor inteligente en su estándar 1452.1 como un dispositivo que tiene funciones adicionales a las de únicamente censar una variable física, teniendo así una parte del sistema en su interior, la capacidad de comunicarse con protocolos especiales tiene convertidores analógicos digitales, digitales analógicos, memoria, entre otras funciones especiales. En el mundo de los sensores inteligentes, algunos de estos se encuentran en un controlador como por ejemplo una tarjeta electrónica embebida o en un chip, con el objetivo de mejorar las funciones del sensor [20]. En la figura 5 se puede observar el diagrama de bloques de un sensor inteligente.

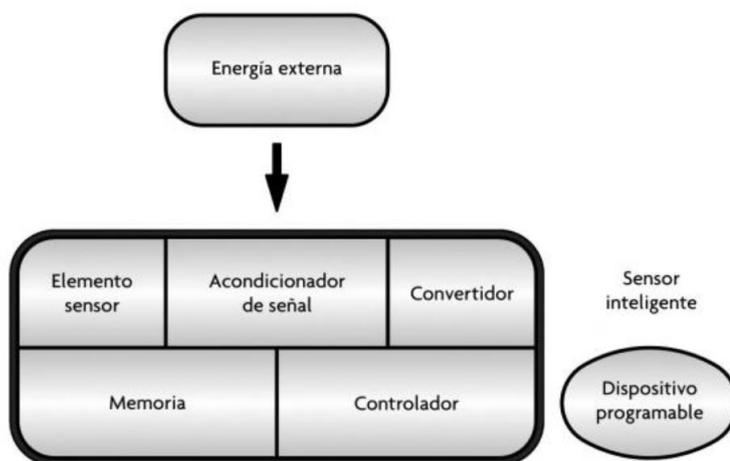


Figura 5 Sensor Inteligente [20].

1.3.5.2 Sensores MEMS

Los sensores MEMS se definen como Sistemas Micro-Electro-Mecánicos, y se caracterizan por tener elementos de tamaño micrométrico. Estos sensores están configurados para cumplir con una función específica dentro de un sistema, para ello

contiene elementos mecánicos y electrónicos. Principalmente en el núcleo de un sensor MEMS se encuentra un sensor y un transductor, esto se representa en la figura 6.

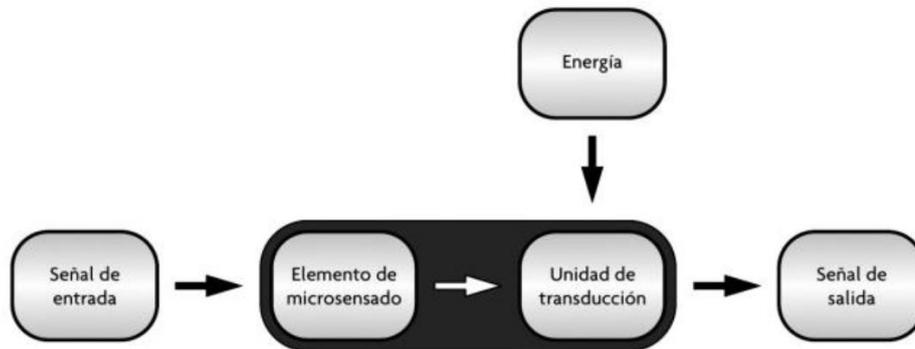


Figura 6 Diseño de un sensor MEMS [20].

Los sensores MEMS están orientados específicamente al para el censado de cantidades físicas y químicas, tales como la presión, la fuerza, la temperatura, la aceleración, flujo magnético, radiación o composiciones químicas. Cada sensor MEMS se clasifica de otros tipos como PolyMUMPs, SOI MEMS, LIGA, CMOS-MEMS debido a que utilizan una tecnología determinada. Por ejemplo, existen sensores que se encuentran acoplados un chip, y el sistema electrónico para su funcionamiento en otro chip, de tal manera que acoplan y se encapsulan juntos mediante un sistema de comunicación. Por otro lado, otras tecnologías juntan los sensores y la electrónica asociada dentro de un mismo elemento [20].

1.3.6 Sensores Biométricos

Los sensores biométricos se definen como dispositivos que se encargan de convertir una señal propia de un ser humano en impulsos eléctricos, permiten la toma de signos vitales como ritmo cardiaco, glucosa, electrocardiogramas, presión arterial, oxigenación de la sangre, entre otros factores que inciden en la calidad de la salud humana [9]. Estos sensores están presentes en hospitales como sistemas de monitoreo continuo. La presencia de estos dispositivos también está en equipos wearables como por ejemplo las bandas inteligentes, que permiten censar múltiples signos vitales, además, este tipo de sensores emplean haces de luz para obtener los valores de las variables antes mencionadas.

1.3.7 Fotoplestimografía (PPG)

La fotoplestimografía se encarga de recoger las distintas variaciones que se producen en los vasos sanguíneos por cada contracción cardíaca [21]. Un Fotoplestismógrafo se encarga de tomar las fotoplestimografías, está compuesto de un opto-acoplador que mediante diodos led ilumina la piel y mide la absorción de la luz que se genera por cada vez que el corazón bombea sangre. Una característica importante es que a pesar que los pulsos que genera el corazón tiende a amortiguarse por el tiempo que tarde en llegar a la piel, son suficientes para lograr en las arterias una dilatación. Una luz infrarroja mide los cambios de volumen que se generan en la piel, por otra parte, también se mide la cantidad de luz reflejada empleando un fototransistor. En la fotoplestimografía se representa el ciclo cardíaco como una onda, misma que varía conforme la localización y diseño del acoplador, esta señal se puede observar en la figura 7.

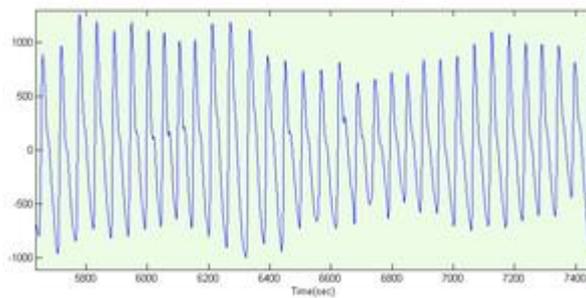


Figura 7 Señal PPG [22].

Estas señales se obtienen aplicando dichas emisiones de luces comúnmente en los dedos, de esta forma existen tres posibilidades de captura, que son la trans-iluminada, reflexión de luz y fibra óptica como se aprecia en la figura 8.

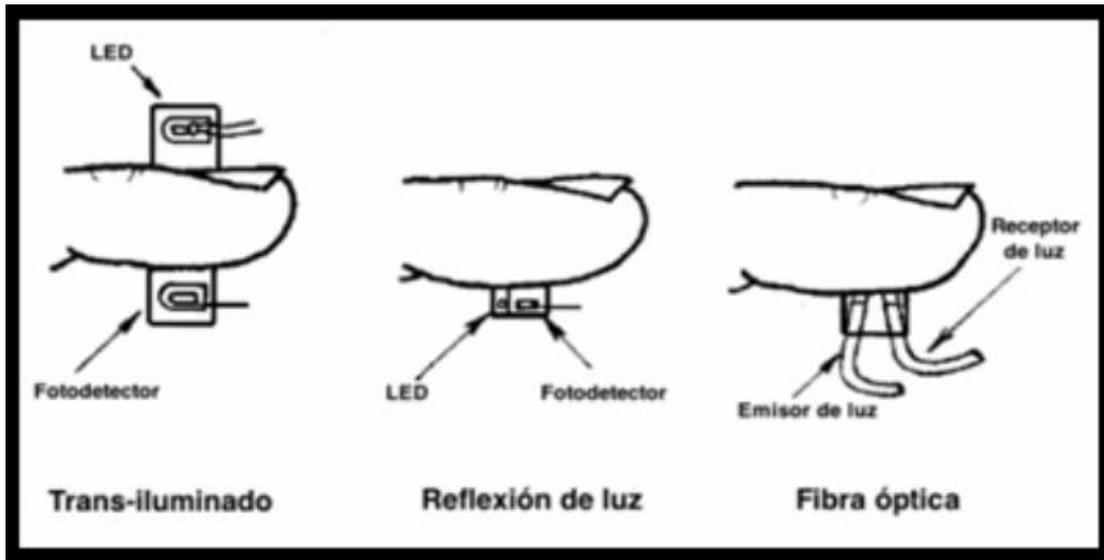


Figura 8 Captura para Fotoplestimografía [22].

Se puede emplear la señal de PPG obtenida para poder obtener la presión arterial, ya que la misma está relacionada con la frecuencia cardíaca, de tal forma que, como se puede apreciar en la figura 9, la presión sistólica estaría en el pico más alto, y la diastólica en el bajo. Esta gráfica generada permite verificar la estabilidad de la presión arterial además de generar un dato sobre el ritmo cardíaco [22].

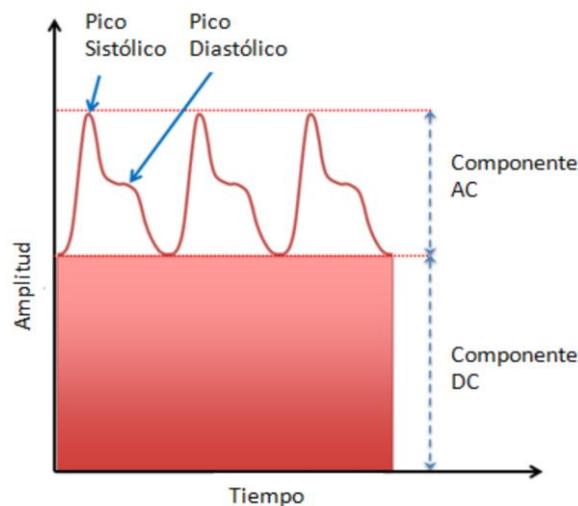


Figura 9 Componentes de la Señal PPG [23].

1.3.8 Electrocardiogramas (ECG)

Los electrocardiogramas son métodos que se emplean para poder obtener los estímulos eléctricos que se presentan en el corazón, específicamente cuando el mismo se

comprime o se expande. Este tipo de tecnología emplea electrodos que van conectados a la piel para registrar dichos impulsos eléctricos. Esta tecnología se emplea en el ámbito médico para poder obtener el ritmo cardíaco, verificar el correcto funcionamiento del corazón y sus latidos, también permite obtener el tamaño de las aurículas [24] . En la figura 10 se aprecia la salida de un electrocardiograma junto con sus principales características.

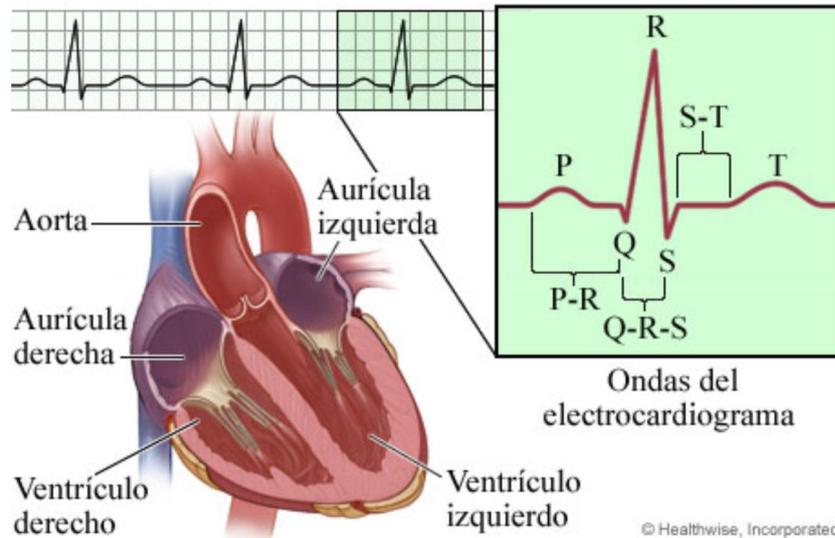


Figura 10 Electrocardiograma ECG y sus características [25].

La letra P en la figura 10 representa el impulso eléctrico que pasa a través de las aurículas, que son las cavidades superiores del corazón, QRS por otro lado representa el impulso eléctrico por las cavidades inferiores del corazón o ventrículos. ST aparece cuando se contrae el ventrículo sin presencia de algún impulso eléctrico. Finalmente T representa el tiempo en el cual los ventrículos se alistan para dar inicio a la siguiente contracción [25].

1.3.9 IEEE Std 1708-2014

El instituto de Ingenieros Eléctricos y Electrónicos es una agrupación técnica profesional alrededor del mundo, cuyo trabajo se ve centrado en el avance tecnológico y la estandarización de nuevas tecnologías que con el tiempo aparecen, buscando beneficios para la humanidad. Actualmente la organización cuenta con aproximadamente 1300 estándares activos, entre ellos el Std. 1708-2014 [26].

El estándar IEEE Std. 1708-2014 describe una normativa para los dispositivos de medición de presión arterial que no emplean una pera o manguito, así como la evaluación de este dispositivo. Esta normativa no regula la forma física del equipo, ya que puede ser de cualquier tamaño o forma, sin embargo, no garantiza una correcta medición si se emplean los manguitos. Esta normativa incluye a los equipos que usan la piel para tomar medidas de presión arterial, como es el caso de los dispositivos electrónicos que usan fotoplestimografía. El estándar 1708-2014 sirve de guía para que los profesionales de la salud o investigadores en busca de desarrollo de nuevos proyectos tengan una guía sobre las pautas de fabricación y de fidelidad de estos dispositivos. Las tasas de error que admite este estándar se mantiene en una tolerancia del +/- 4 mmHg, para la presión sistólica y diastólica [27].

1.3.10 Placa de Desarrollo

También conocidas como placas de programación, son elementos electrónicos que poseen un microcontrolador que es programable y sirve para ejecutar distintas acciones, esta característica les permite ser equipos aptos para aplicaciones inteligentes y de IoT (Internet de las Cosas). Usualmente estas placas de desarrollo incorporan protocolos de comunicación, entradas y salidas tanto digitales como analógicas, convertidores, memoria, inclusive en algunas placas se incorporan funciones adicionales como comunicaciones inalámbricas Wi-Fi y Bluetooth. Una de las placas más populares es la de Arduino, que se emplea tanto para proyectos sencillo como para complejos, sin embargo, no es la única que existe [28].

1.3.11 Inteligencia Artificial

La inteligencia artificial se define como el conjunto de algoritmos que busca dar solución a un problema o situación, es capaz de aprender y desarrollarse de mejor forma absorbiendo información y aprendiendo de ella. Representada por las siglas IA, tiene como objetivo que los computadores puedan realizar acciones propias de una mente humana. Algunas de estas son capaces de razonar, y comúnmente se las llaman inteligentes, por otro lado, otras como la visión no suelen llamarse así. Sin embargo, cada una está relacionada con cualidades como la percepción, la predicción, la planificación, control, etc. La inteligencia artificial emplea múltiples técnicas para poder resolver un problema o concretar una tarea [29].

La aplicación de la inteligencia artificial puede darse en casi todas las situaciones, por ejemplo, en la tabla 4 se detallan algunas de estas.

Tabla 4 Aplicaciones de la IA [30].

Aplicaciones de la Inteligencia Artificial	
Aplicación	Descripción
Procesamiento de imágenes en industrias	La IA puede reconocer imágenes para posteriormente clasificarlas y etiquetarlas.
Comercio	Algoritmos para mejorar las ventas
Medicina	Clasificación y diagnóstico de datos de personas.
Predicciones Industriales	Algoritmo de manteamientos preventivos.
Vehículos Automáticos	Reconoce, detecta y clasifica objetos.
Marketing Digital	Algoritmos encargados de distribuir contenido en redes sociales.
Ciberseguridad	Reconocimiento de malware para protección de datos y sistemas.

1.3.12 Aprendizaje Automático

El aprendizaje automático, Machine Learning, en inglés, es un apartado importante de la inteligencia artificial, se define como la capacidad de las máquinas de poder absorber conocimiento o aprender sin una programación dedicada para este fin. Un claro ejemplo de ello son las redes sociales de la actualidad que se acoplan a los gustos o necesidades de los usuarios, generando así sugerencias en base a búsquedas anteriores, es decir un algoritmo aprende sobre nosotros y con esos datos empieza a arrojar sugerencias basadas en esos gustos. Existen tres tipos de aprendizaje automático: supervisado, no supervisado y de refuerzo como se muestra en la figura 11.

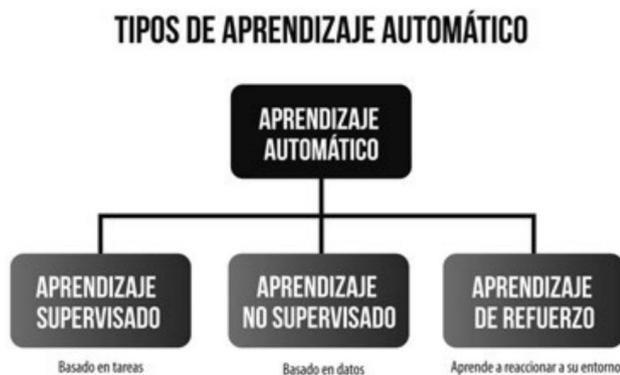


Figura 11 Clases de Aprendizaje Automático [30].

El aprendizaje supervisado requiere de bases de datos previamente clasificados para poder cumplir con una tarea, por otro lado, el aprendizaje no supervisado no necesita de dichos datos, sino que aprende en ese momento. Finalmente, el aprendizaje por refuerzo se caracteriza porque la inteligencia artificial aprende de la experiencia, algo así como un sistema de prueba y error, si se equivoca se corrige y si acierta se le incentiva, comúnmente se relaciona con el domesticar un perro, si cumple con lo propuesto se le asigna un premio [30].

1.3.13 Aprendizaje Profundo

El aprendizaje profundo o Deep learning, es un grupo de algoritmos de aprendizaje automático que se emplea para dar solución a tareas demasiado complejas. Este tipo de aprendizaje maneja grandes cantidades de información y emplea las redes neuronales, que son capas que se asocian de manera organizada para formar patrones complejos y de esta manera cumplir con un objetivo. Una de las aplicaciones de estas redes neuronales se encuentra en algoritmos de reconocimiento de voz, la visión artificial, conducción automática de vehículos, etc. En la figura 12 se puede apreciar cómo se componen las redes neuronales [29].

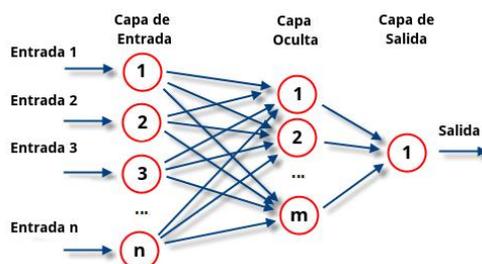


Figura 12 Redes Neuronales [31].

Las entradas se consideran las variables que permiten a la inteligencia artificial tomar una decisión, las capas o neuronas tiene asignados pesos que poco a poco se ajustan de acuerdo a los datos que se entregan y del cual aprenden, finalmente la salida viene a ser el dato que la inteligencia artificial ha procesado [32].

1.3.14 Árboles De Decisión

Los árboles de decisión son una clase de inteligencia artificial que pertenece al aprendizaje supervisado, básicamente está conformado por nodos, entre los que destacan la raíz, ramas, internos y hojas, como se puede apreciar en la figura 13.

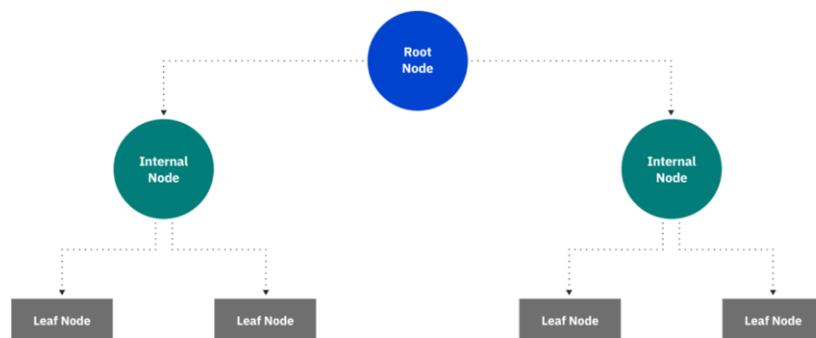


Figura 13 Composición de un árbol de decisión [33].

El funcionamiento de esta clase de inteligencia artificial relaciona el nodo raíz con los internos, donde se producen evaluaciones entre sí, dichas evaluaciones se indican por medio de los nodos hoja, mismos que representan las salidas que son posibles en una agrupación de datos. Este modelo es muy empleado cuando se requiere realizar toma de decisiones. Su aprendizaje se basa en un algoritmo que comúnmente se conoce como “divide y vencerás”, que busca dividir de forma correcta los puntos dentro del árbol. Este algoritmo se repite una y otra vez dentro del árbol, siguiendo un sentido descendente. Mientras más niveles existan dentro de un árbol, más complejo se vuelve el mismo, es por eso que usualmente se prefiere a los árboles con ramificaciones o niveles pequeños, esto bajo un principio que sugiere que no se debe incrementar las ramificaciones del árbol más allá de lo necesario. En la tabla 5 se presentan los tipos de árboles de decisión existentes, basados en el algoritmo de Hunt [33].

Tabla 5 Árboles de Decisión [33].

Tipos de Árboles de Decisión	
ID3	“Iterative Dichotomiser 3”, se basa en la ganancia de la información para elegir y estudiar las divisiones existentes.
C4.5	Desarrollado por Quinlan, es una evolución de ID3 que emplea la ganancia de la información, o a su vez proporciones de esta para validar puntos de división.
CART	Árboles de clasificación y regresión, se caracteriza por evaluar las veces que se clasifica de forma errónea una variable tomada al azar.

1.3.15 Lenguajes de Programación

Un lenguaje de programación se define como el software destinado a la construcción de otros softwares o programación informáticos, que emplea reglas dentro de un lenguaje para organizar acciones y procesos lógicos que deben ser ejecutados empleando un sistema computacional. En otras palabras, es aquella forma en la cual una persona comunica a un ordenador lo que se desea realizar. Permite describir tareas en notaciones que son entendibles tanto para el usuario como para la máquina [34]. Estos lenguajes han ido evolucionando con el tiempo, se destacan 5 generaciones en la actualidad, mismas que se detallan en la tabla 6.

Tabla 6 Generaciones de Lenguaje de Programación [34].

Generaciones de Lenguajes de Programación	
Primera Generación	Lenguaje empleado por los ordenadores, emplea 0 y 1, es de difícil comprensión para el ser humano. Funcionan solo en ordenadores específicos.

Segunda Generación	<p>Pertencen los lenguajes ensambladores, se emplean normativas para facilitar el tema de la lectura y escritura. Cada función recibe un nombre específico. Actualmente este lenguaje ensamblador es utilizado para la instalación de drivers en computadoras. Una dificultad con este lenguaje es que no es directamente ejecutable en un ordenador, ya que solamente entiende órdenes codificadas con 1 y 0.</p>
Tercera Generación	<p>Aparecen los lenguajes como C, Java, FORTRAN, entre otros. Son lenguajes de alto nivel, ya que poseen reglas definidas y es más entendible para el ser humano, no obstante, para que la máquina pueda entender es necesario un compilador, que se encarga de traducir a lenguaje de máquina. Estos lenguajes sirven para programar sistemas operativos, páginas Web y aplicaciones.</p>
Cuarta Generación	<p>Pertencen lenguajes de programación para fines específicos como SQL o SPSS. No se usan para diseñar aplicaciones o programar sistemas operativos, por ejemplo, SQL se emplea para bases de datos, permite realizar consultas, ingresar datos o modificaciones generales, pero se emplea únicamente para ese fin. Lo mismo sucede con SPSS que sirve para manejar grandes cantidades de datos para propósitos estadísticos.</p>

Quinta Generación	Lenguajes empleados en inteligencia artificial, posibilita el uso de restricciones para la resolución de problemas o tareas. Como ejemplo de lenguajes de quinta generación son Prolog y Mercury.
-------------------	---

1.3.16 Python

Python se define como un lenguaje de programación que puede ser empleado para casi todo, por esto se lo clasifica como un lenguaje de propósito general y cuya filosofía radica en ser legible y de fácil comprensión. Es empleado para distintos proyectos como aplicaciones de escritorio y Web, creación de software, ciencia de datos y también para crear algoritmos de inteligencia artificial [35]. Python presenta múltiples ventajas:

- Fácil comprensión ya que la mayoría de sus comandos son instrucciones en inglés.
- Emplea menos líneas de código para realizar acciones que en otros lenguajes.
- Contiene una gran cantidad de librerías disponibles que permiten realizar casi cualquier tarea.
- Es compatible con otros lenguajes de programación.
- Al ser de uso libre, y con una gran cantidad de usuarios alrededor del mundo, posee soporte inmediato ante cualquier problema.
- Es un lenguaje de programación demasiado conocido, por lo que existen miles de guías y tutoriales para su uso.
- Es multiplataforma, de tal forma que se puede ejecutar en sistemas operativos como Windows, Linux, macOS y Unix.
- Empleado para el desarrollo de algoritmos de inteligencia artificial.
- Es un lenguaje completo, ya que no se limita únicamente al desarrollo de scripts [36].

1.3.17 Base de Datos

Una base de datos es un sistema digital para almacenar registros, se considera como una especie de contenedor para distintos archivos que sobre los mismos se pueden realizar acciones como: insertar o eliminar datos, ingresar, modificar o eliminar nuevos archivos. Un conjunto de datos que pertenecen a una misma clase de registros se les denomina tabla. Para poder manipular este tipo de archivos o datos se emplea un lenguaje de programación, denominado SQL. Las bases de datos pueden ser utilizadas mediante los sistemas de administración de base de datos o DBMS, que está compuesto por datos, hardware, software y los usuarios. En la figura 14 se observa cómo está compuesto un DBMS [37].

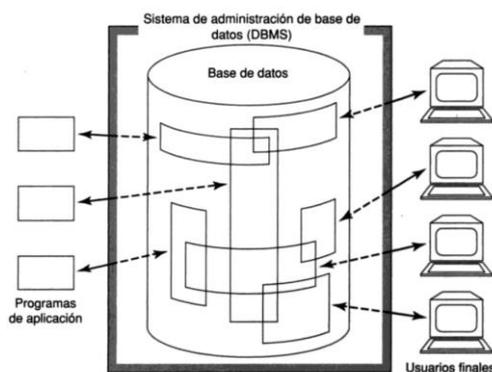


Figura 14 Componentes de un Sistema de administración de base de datos [37].

En la tabla 7 se presentan algunos ejemplos de software para DBMS, junto con sus características más representativas.

Tabla 7 Sistemas de Gestión Para Bases de Datos [38].

Ejemplos de DBMS	
Ejemplo	Descripción
MySQL	Diseñado bajo lenguaje de programación C y C++. Posee un analizador de sintaxis de SQL con scanner. Es multiplataforma, además posee un amplio soporte.
Microsoft SQL Server	Uno de los principales sistemas de gestión de base de datos a nivel de

	mercado gracias a que incorpora software adicional para empresas y análisis propios del mercado. Emplea un lenguaje de consulta Transact-SQL con un estándar Structured Query Language.
Oracle	Permite la administración de grandes cantidades de datos desde un solo archivo. Es multiplataforma con soporte integrado. Emplea el lenguaje PL/SQL, con características adaptables de acuerdo a las necesidades requeridas.

1.4 Objetivos

1.4.1 Objetivo general

- Implementar un sistema de monitoreo para la detección de personas hipertensas empleando inteligencia artificial.

1.4.2 Objetivos específicos

- Analizar los métodos para la obtención de la presión arterial.
- Diseñar un dispositivo electrónico para la detección de personas hipertensas.
- Desarrollar un asistente virtual para la recolección de la información e interpretación de los datos empleando inteligencia artificial.

CAPÍTULO II

METODOLOGÍA

1.5 Materiales

El proyecto que se presenta ha sido elaborado empleando materiales como un sensor biométrico, el cual tiene la capacidad de realizar fotoplestimografías para poder obtener los valores de presión arterial, también se ha empleado una placa de desarrollo programable para procesar los valores que emite el sensor, artículos científicos médicos y tecnológicos, libros, documentos en general sobre la hipertensión y sus enfermedades derivadas, bases de datos de pacientes y sus signos vitales generales para entrenar el algoritmo de inteligencia artificial.

1.6 Métodos

1.6.1 Modalidad de la Investigación

1.6.1.1 Investigación Aplicada

El presente proyecto empleo una Investigación Aplicada, haciendo uso de los conocimientos que se han adquirido durante la carrera profesional para diseñar la interfaz gráfica del asistente virtual, y automatizar la toma de datos y la entrega de diagnósticos del mismo, teniendo una fuerte base en programación y comunicaciones avanzadas.

1.6.1.2 Investigación Bibliográfica

El presente proyecto empleo una Investigación Bibliográfica, ya que se recurrió a información académica y científica de gran credibilidad para indagar sobre la hipertensión arterial y sus diagnósticos, así como también a bases de datos fiables para realizar el entrenamiento de la inteligencia artificial, todo esto extraído de documentos como tesis, artículos científicos, publicaciones en revistas académicas, guías médicas y libros.

1.6.1.3 Investigación Experimental

El proyecto también empleó una modalidad de Investigación Experimental ya que se realizó pruebas del dispositivo y diagnósticos generados por el asistente virtual, y de esta forma validar la información procesada por el sistema.

1.6.1.4 Investigación de Campo

Finalmente, el proyecto emplea una investigación de campo, debido a que es necesario consultar con profesionales de la salud acerca de esta patología, así como también realizar pruebas en individuos para calibrar el sistema y comparar las medidas que se van adquiriendo para garantizar la confiabilidad del sistema de monitoreo.

1.6.2 Recolección de Información

Para la recolección de la información en el proyecto se empleó distintos repositorios de universidades e institutos educativos tanto nacionales como internacionales, revistas de difusión científica, artículos científicos, libros, tesis, proyectos fiables y bases de datos para desarrollo de Inteligencia Artificial, accediendo a cada uno de ellos mediante el uso académico del Internet.

1.6.3 Procesamiento y Análisis de Datos

Una vez la información fue recolectada, se ha filtrado los datos de mayor importancia, dejando descartada la información con poca relevancia o redundante para el proyecto, procediendo así con las siguientes actividades desarrolladas:

- Recolección de la información obtenida.
- Revisión de la información disponible.
- Estudio de las diferentes soluciones existentes para la elaboración de dispositivos fiables que permitan recolectar datos y procesarlos.
- Deducción de la información más importante que ayude al correcto desarrollo del proyecto de investigación y que permita dar una solución adecuada a la propuesta.

1.6.4 Propuesta de Solución

Con el uso adecuado de elementos electrónicos se da paso a la construcción de un medidor de presión arterial cuyos resultados sean confiables, permitiendo así su uso comercial y aplicable a métodos médicos seguros, estos datos que son recolectados son interpretados por una inteligencia artificial que es capaz de emitir un diagnóstico oportuno para de esta forma detectar la hipertensión en las personas, y además mitigar las múltiples muertes o problemas de salud que son derivados por padecer esta patología, y con ello mejorar el estilo de vida y prolongar la misma de muchas

personas, además de que su uso no requiera de un personal médico capacitado para emitir un diagnóstico para que de esta forma pueda llegar a muchas más personas.

1.6.5 Desarrollo del Proyecto

En el presente proyecto se han ejecutado las siguientes actividades:

1. Determinación de los métodos actuales para obtención de la presión arterial.
2. Análisis de los tipos de presión arterial existentes.
3. Investigación del funcionamiento de los métodos que emplean sistemas electrónicos para la obtención de la presión arterial.
4. Selección de los métodos más eficientes para la obtención de la presión arterial.
5. Análisis de las tasas de error admisibles para los medidores de presión arterial comerciales.
6. Diseño esquemático del dispositivo electrónico para la medición de la presión arterial.
7. Simulación del circuito electrónico a emplearse para el dispositivo de medición arterial.
8. Elaboración del circuito electrónico para el dispositivo de medición de presión arterial.
9. Análisis de los pasos para emitir un diagnóstico médico usando el nivel de presión arterial.
10. Implementación de una base de datos para almacenar la información recolectada por el dispositivo de medición arterial.
11. Programación de la inteligencia artificial para la interpretación de los datos
12. Diseño del asistente virtual para la visualización de los resultados.

CAPÍTULO III

RESULTADOS Y DISCUSIONES

1.7 Análisis y discusión de los resultados

La implementación de un sistema de monitoreo para la detección de personas hipertensas empleando inteligencia artificial permite a los pacientes usuarios conocer a tiempo el estado de su presión sanguínea, y de esta manera lograr un control si se llegara a padecer anomalías en la misma, evitando así problemas futuros como lo son patologías cardíacas e inclusive prevenir riesgos como derrames cerebrales.

1.8 Análisis de Factibilidad

1.8.1 Factibilidad Económica

El proyecto es económicamente factible ya que todo lo requerido para el mismo corre por cuenta del investigador, además que existe un beneficio enorme con los resultados del mismo, que ha comparación de los problemas que se solucionan con la propuesta caen en una inversión de grandes beneficios.

1.8.2 Factibilidad Bibliográfica

La cantidad de información disponible sobre la hipertensión es abundante y de libre acceso, la misma se ha obtenido de libros, artículos científicos, tesis, proyectos, revistas académicas, foros, bases de datos, entre otras fuentes confiables.

1.9 Desarrollo de la propuesta

1.9.1 Requerimiento del Prototipo

El desarrollo del sistema de monitoreo para la detección de personas hipertensas empleando inteligencia artificial se han establecido diversos requerimientos para que el dispositivo sea confiable y su uso sea igual al de los dispositivos comerciales, adicional el software que procese todos estos datos sea intuitivo y de uso fácil para cualquier persona, de tal forma que se debe cumplir lo siguiente:

- Mediciones de presión arterial con errores máximos del +/- 4 mmHg.
- Interacción amigable con el asistente virtual.
- Interfaz principal controlada por medio de comandos de voz.

- Generación de diagnósticos finales visibles y de fácil comprensión.
- Almacenamiento de diagnósticos finales.

1.9.2 Diagrama de bloques del sistema

El presente sistema de monitoreo ha empleado el diagrama de bloques que se representa en la figura 15, el cual consta de tres partes principales. La primera se encargará de tomar los datos necesarios para poder obtener la presión arterial, estos son generados por el sensor de señales vitales y procesado por una placa de desarrollo junto con un microcontrolador. La segunda parte consta de la adquisición de los datos empleando una comunicación serial USB y un script en Python para almacenar estas variables. Finalmente, el tercer y último bloque se encarga del monitoreo de los datos, el asistente virtual mediante inteligencia artificial clasificará la información obtenida por el script en Python y generará un diagnóstico, que luego lo guardará en una base de datos.

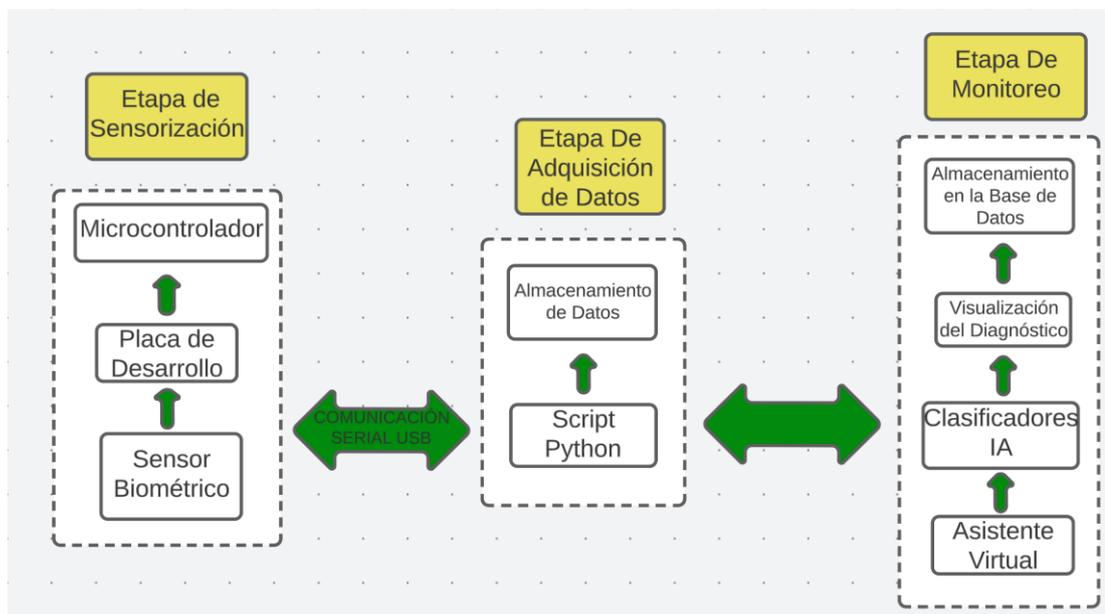


Figura 15 Diagrama de Bloques del Sistema.

Fuente: Investigador

1.9.3 Selección de los elementos para la implementación del sistema

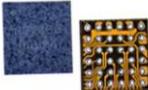
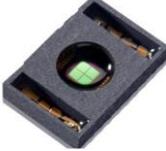
Para la selección de elementos se empleó el diagrama de bloques representado en la figura 15, para cumplir con ello se realiza un análisis analítico de cada uno de los elementos disponibles, con la finalidad de seleccionar el mejor componente para su posterior implementación, de igual forma se lo realizó con los distintos algoritmos de inteligencia artificial, en la rama de los árboles de decisión.

- Sensor de signos vitales
- Placa de desarrollo
- Tarjeta Electrónica
- Algoritmos de aprendizaje automático

1.9.4 Sensor de Signos Vitales

El sensor de signos vitales va a ser el encargado de tomar los datos del paciente, este dispositivo debe ser capaz de censar mediante la tecnología denominada fotoplestimografía (PPG). En la actualidad existen muchos sensores de este tipo, sin embargo, un grupo muy pequeño es capaz de satisfacer todas las necesidades que requiere el proyecto. El sensor debe ser capaz de entregar datos de tal forma que el color de la piel no sea un factor que altere las medidas tomadas, así como también la luz ambiental, ya que estos datos pueden ser tomados en cualquier momento del día, o la noche. Para poder seleccionar el sensor a emplearse se realiza una selección analítica de todos los elementos presentados en la tabla 8, basándose principalmente en los rangos de errores al momento de obtener la presión arterial, así como también los algoritmos que cada uno de estos sensores soporta. Todos los sensores expuestos en esta tabla cumplen con la capacidad de poder obtener los datos mediante PPG (fotoplestimografía), sin embargo, solamente uno de ellos, cumple con la norma IEEE, la Std 1708 – 2014, la cual ha sido descrita anteriormente.

Tabla 8 Sensores de Signos Vitales [39].

Sensor de Signos Vitales	Descripción Gráfica	Alimentación [V]	Algoritmos Soportados	Características
AS7038RB		2.7 – 5.5	SpO2, Presión Sanguínea, Ritmo Cardíaco, Variación del Ritmo Cardíaco.	Basado en tecnología PPG y ECG, cumple con la IEEE Std 1708-2014 para la medición de presión arterial sin manguito.
AS7050		1.7 – 1.98	HRM, HRV, SpO2, Presión Sanguínea	Basado en tecnología PPG, ECG y resistencia de la piel galvánica. Posee 8 Leds de salidas y 6 Leds de entrada.
AS7024		2.7 – 5.5	SpO2, Presión Sanguínea, Ritmo Cardíaco, Variación del Ritmo Cardíaco.	Basado en tecnología PPG y ECG, mediciones con rango de errores de +/- 8 mmHG, requiere configuraciones para cada tipo de piel para mejorar la precisión.

SFH 7072		2.7 – 5.5	SpO2, Presión Sanguínea, Ritmo Cardíaco.	Basado en tecnología PPG y ECG, sus valores de medida de precisión se centran en Oxímetro y Ritmo Cardíaco, la presión arterial se obtiene con ciertos rangos de errores.
AS7038GB		2.7 – 5.5	SpO2, Presión Sanguínea, Ritmo Cardíaco,	Basado en tecnología PPG y ECG, no cumple IEEE para Presión Sanguínea

Todos los sensores representados en la tabla 8 son compactos y emplean tecnologías como PPG, conocida como fotoplestimografía, y ECG, electrocardiogramas. Tienen voltajes de operación muy pequeños y consumos bajo de corriente, algunos poseen certificaciones por parte de la IEEE, que validan su uso en campos profesionales de la salud, haciéndolos aún más confiables [39].

Finalmente, el sensor seleccionado es el AS7038RB, debido a que es el único que cumple con la normativa IEEE Std. 1708 – 2014, el cual entrega valores confiables para mediciones de presión arterial que no emplean el típico manguito que se coloca en el brazo para obtener estas magnitudes. Este sensor es el último de su generación en su categoría, posee un tamaño relativamente pequeño, pero que ofrece múltiples

características interesantes y que son fiables. En la figura 16 se puede apreciar cómo es dicho sensor de forma física.

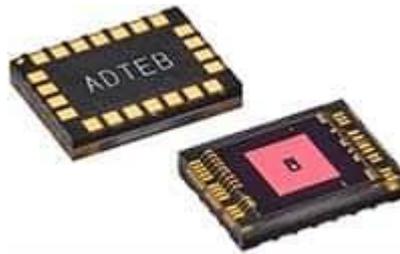


Figura 16 Sensor AS7038RB.

La tabla 9 detalla todas las características técnicas de este sensor.

Tabla 9 Características Técnicas del Sensor De Signos Vitales AS7038RB [40].

CARACTERÍSTICAS TÉCNICAS
<ul style="list-style-type: none">• Voltaje de Operación: 2.7 – 5.5 Voltios.• Consumo de Corriente: 200mA.• Bus de comunicación I2C: I2C Esclavo Estándar (100 kHz) y modo rápido (400 kHz).• Rango de Temperatura: -30 a 70 °C.• Dimensiones: 3.82 mm X 2.92 mm.• 22 Pines de conexión disponibles.• Aborda todos los tipos de piel.• SNR Óptico más alto.• Electrocardiograma ECG con electrodos secos.• Filtro de Interferencias Integrado.• Habilitación de mediciones de SpO2.• Tamaño de aplicación más pequeño.

El sensor AS7038RB posee 22 pines de conexión disponibles. Al ser un elemento relativamente pequeño es propenso a riesgos de descarga eléctrica estática, por lo que su manipulación debe ser muy cuidadosa. En la figura 17 se puede apreciar la descripción por bloques de cada uno de estos pines, así como su diseño interno.

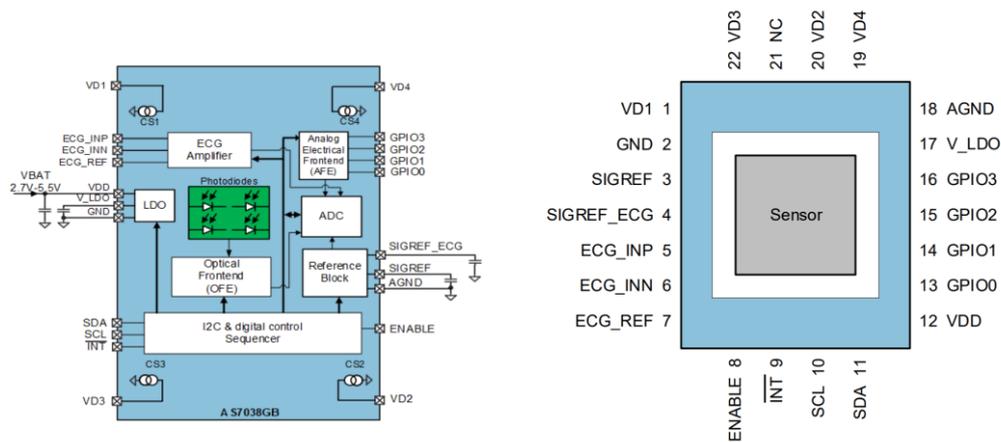


Figura 17 Pines de Conexión Del Sensor AS7038RB. [40]

Cada uno de los pines del AS7038RB cumple una función específica y ha sido dotado de características diferentes, en la tabla 10 se realiza un sumario sobre cada una de las entradas de este sensor y sus diferentes funciones o propósitos.

Tabla 10 Descripción de Pines del AS7038RB [40].

Número de PIN	Nombre de PIN	Descripción
1	VD1	Conexión al disipador de corriente 1.
2	GND	Tierra de la fuente de alimentación.
3	SIGREF	Salida de Referencia Analógica. Se debe conectar un condensador de 2.2 μ F a tierra.
4	SIGREF_ECG	Salida de referencia analógica.
5	ECG_INP	Entrada positiva del amplificador ECG.
6	ECG_INN	Entrada negativa del amplificador ECG.
7	ECG_REF	Salida de referencia del amplificador ECG.
8	ENABLE	Habilita el AS7038RB

9	INT	Salida de interrupción
10	SCL	Entrada de reloj para comunicación I2C
11	SDA	Terminal de datos para comunicación I2C
12	VDD	Tensión de alimentación
13	GPIO0	Entrada/Salida de propósito general
14	GPIO1	Entrada/Salida de propósito general
15	GPIO2	Entrada/Salida de propósito general
16	GPIO3	Entrada/Salida de propósito general
17	V_LDO	Tensión de salida de 1.9 Voltios. Se conecta un condensador de 2.2 μ F a tierra.
18	AGND	Tierra analógica. Conexiones de tierra de bajo ruido.
19	VD4	Conexión al disipador de corriente 4
20	VD2	Conexión al disipador de corriente 2
21	NC	No se conecta
22	VD3	Conexión al disipador de corriente 2

1.9.5 Placa de Desarrollo

Una placa de desarrollo se empleará para poder acondicionar el sensor AS7038RB, ya que el mismo requiere de ciertos circuitos preliminares para poder trabajar. Esta placa

permite evaluar o usar estos sensores de una manera sencilla. Tienen incluido en su circuito todas las etapas que el sensor de signos vitales necesita para poder entregar medidas con total fidelidad. Las placas de desarrollo disponibles se presentan junto con sus características en la tabla 11.

Tabla 11 Placas de Desarrollo para el Sensor AS7038RB [41].

Placa de Desarrollo	Descripción Gráfica	Alimentación [Voltios]	Descripción
Eval Kit AS7038RB		4.5 – 5.5	<p>Permite usar el sensor AS7038RB, evaluar cada una de sus funciones. Incorpora un microcontrolador programable. Admite únicamente comunicación USB.</p>
Eval Kit AS703XX		3.7 – 5.5	<p>Permite usar los sensores AS7030G, AS7038GB, AS7038RB para evaluar cada una de sus funciones. De acuerdo a las necesidades del usuario se puede acoplar el microcontrolador que se requiera.</p>

			Incorpora una batería.
--	--	--	------------------------

Al momento de seleccionar la placa de desarrollo adecuada se tomó en cuenta detalles como la posibilidad de poder personalizar a gusto la placa con cualquier microcontrolador que se desee, así como la universalidad de este, es decir que si se puede emplear en la placa más sensores del segmento. Estas características están presentes en el modelo AS703XX, el cual ha sido seleccionado para poder llevar a cabo el proyecto. El mismo consta de tres etapas, en la primera se encuentra el procesamiento de los datos, la segunda etapa consta de convertidores ADC aislados, amplificadores, y finalmente la tercera etapa es donde se localiza el sensor de signos vitales, así como los electrodos para realizar los electrocardiogramas, todo esto se puede apreciar en la figura 18.

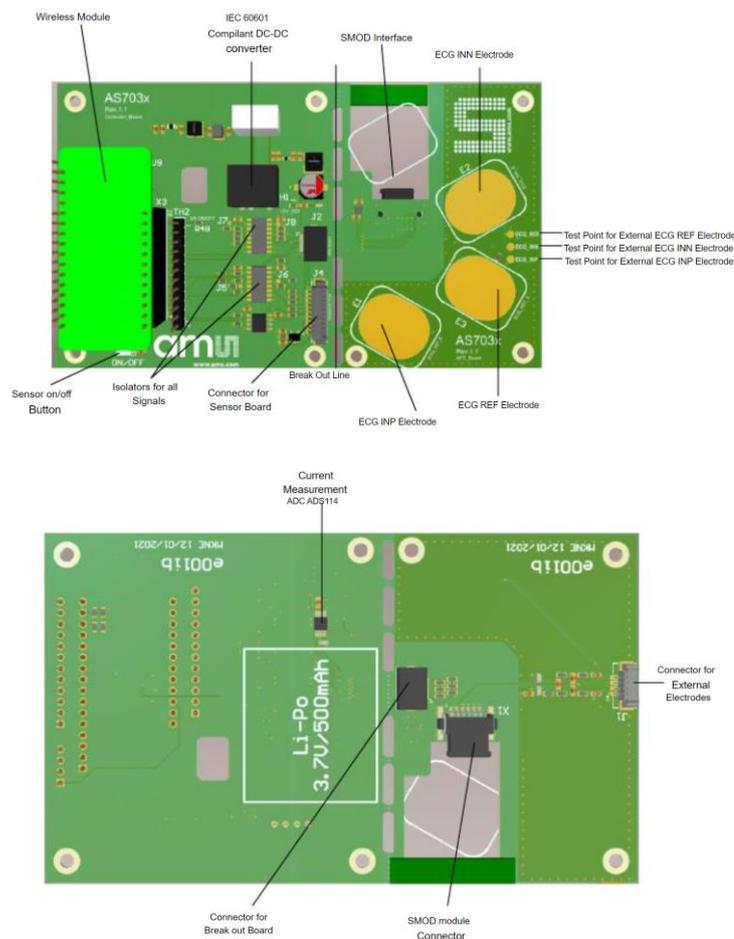


Figura 18 Placa de Desarrollo AS703XX [40].

El poseer una batería le da autonomía a esta placa de desarrollo, así mismo el poder acoplar la tarjeta electrónica de control que se desea le da libertad para poder mejorar de gran manera el procesamiento de los datos. En la tabla 12 se representan las características técnicas de esta placa a mayor detalle.

Tabla 12 Características de la placa AS703XX [41].

CARACTERÍSTICAS TÉCNICAS
<ul style="list-style-type: none">• Voltaje de Operación: 3.7 – 5.5 Voltios.• Corriente Suministrada: Hasta 500 mA.• Dimensiones: 130 mm X 80 mm.• IEC 60601 Compilador/Convertidor DC – DC.• Aisladores para señales.• Conectores para bandas inteligentes de desarrollo.• Batería recargable Li-Po de 3.7v/500mA.• Sensores Compatibles: AS7030, AS7038GB, AS7038RB.• Electrodo Incorporados: 3.• Posibilidad de romper la parte de los sensores y ser nuevamente incorporados por medio de un cable BUS.• Compatibilidad con microcontroladores IoT.

Con la placa seleccionada, se realizó el montaje del sensor, ya que cuenta con un socket de conexión, como se puede apreciar en la figura 19.

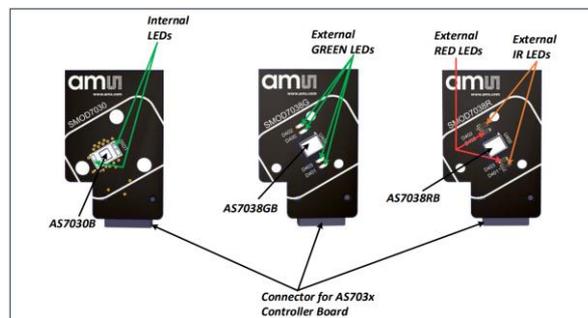


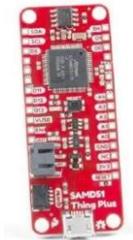
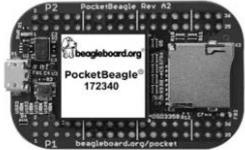
Figura 19 SMOD de los sensores AS703XX [39].

1.9.6 Microcontrolador

El microcontrolador se encargará de procesar los datos recibidos por parte del sensor, luego de haber sido acondicionada toda la señal por parte de la placa de desarrollo. En la tabla 13 se presentan algunos microcontroladores disponibles.

Tabla 13 Ejemplos de Microcontroladores [28].

Placas de Programación y Desarrollo		
Nombre	Descripción Gráfica	Características
Tennsy 3.6		Procesador de 32 bits, 52 entradas/salidas, puerto de comunicación USB, puerto para lectura de tarjetas SD, software compatible con Arduino.
Arduino		Posee muchas versiones como Arduino UNO, MEGA, NANO, BT, entre otros. El más popular el UNO, posee comunicación USB, software propio para programación, muchas librerías disponibles, uso libre.
Particle		Las características dependen del modelo de Particle, en este caso el más popular es Photon. 128Kb de memoria RAM, 1MB de memoria Cache, comunicación Wi-Fi, puerto USB, servicio

		Cloud para aplicaciones IoT.
SaparkFun Thing Plus		Múltiples periféricos como sensores táctiles. Entrada para memoria USB, Ethernet de alta velocidad, Wi-Fi y Bluetooth, conector para baterías tipo Li-Po.
Makey Makey		Para aplicaciones sencillas, no requiere un software para programar, 18 entradas analógicas y digitales, compatible con arduino.
Auricolari Bluetooth- FFTT BIANCO-T-136		Sus aplicaciones están basadas para pizarras inteligentes, puede ser programada a través de un navegador Web empleando Linux.

Para incluir estos datos se ha optado por las tarjetas Particle, que son compatibles con la placa de desarrollo especificada en la tabla 11. Una vez analizada la tabla 14 de los distintos equipos Particle se ha decidido emplear el modelo “Argon”, debido a que los datos que se van a extraer desde el sensor van a ser transmitidos vía USB, y este modelo posee esa conexión en modo rápido hasta 12Mbps, lo que facilitará la toma de datos. Así mismo, este modelo posee entrada para la batería externa y la posibilidad de emplear comunicaciones alternas como lo son Bluetooth o Wi-Fi.

Tabla 14 Modelos Particle y sus Características [42].

Particle	Descripción Gráfica	Dimensión	Alimentación	E/S Analógicas	E/S Digitales	Descripción
Argon		52 mm X 22 mm	3.7 – 5.5 V	6	8	Conexiones USB rápida de 12Mbps, Wi-Fi, Bluetooth. Entradas I2C, UART, SPI. Conexión a batería.
Boron		52 mm X 22 mm	3.7 – 5.5 V	6	8	Conexiones USB, Bluetooth, 2G/3G, LTE. Entradas I2C, UART, SPI. Conexión a batería.
Photon		52 mm X 22 mm	3.7 – 5.5 V	4	6	Conexiones USB y Wi-Fi. Entradas Tx/Rx.
Electron		52 mm X 22 mm	3.7 – 5.5 V	13	13	Conexiones USB, 2G/3G y

						LTE. Entradas Tx/Rx.
--	--	--	--	--	--	----------------------------

En la tabla 15 se exponen todas las especificaciones técnicas de este microcontrolador IoT.

Tabla 15 Características del Microcontrolador Particle Argon [43].

CARACTERÍSTICAS TÉCNICAS
<ul style="list-style-type: none"> • Procesador: ARM Cortex-M4F 32 Bits @64MHz. • 1MB de Flash, RAM de 256KB. • Bluetooth • 20 Entradas de propósito general GPIO. • 6 Entradas analógicas. • 8 Entradas digitales. • UART, I2C, SPI. • Wi-Fi con ESP32 de 2.4 GHz. • Micro USB 2.0 rápido de 12Mbps. • Integra un cargador para baterías Li-Po. • Leds RGB para estados. • Botones de Reset y Mode. • Antena Bluetooth de 2.4 Ghz. • Certificaciones FCC, CE, e IC. • Conector SWC.

El microcontrolador se programa mediante un archivo DFU que proporciona el fabricante del sensor. Se carga haciendo uso de la ventana de comandos de Windows y una librería de Python denominada “nrfutil”. Primero se debe identificar el puerto en el cual se ha reconocido el microcontrolador.

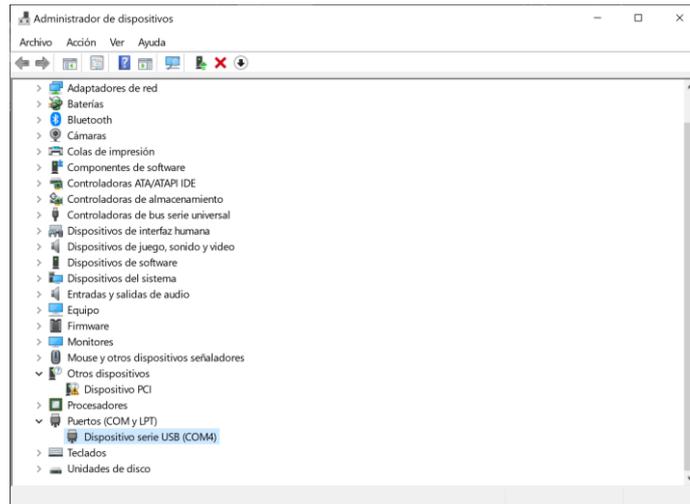


Figura 20 Puerto COM asignado al Microcontrolador Argon.

Fuente: Investigador

El puerto COM asignado es el 4, por ende, el comando que se emplea en la ventana CMD será: “nrfutil dfu usb-serial --port COM4 --package as703x-evk-firmware-v0.5.0.dfu”. La última parte hace referencia al nombre del archivo con extensión dfu proporcionado por el desarrollador del sensor.

1.9.7 Estructura del dispositivo electrónico para la toma de datos

El dispositivo electrónico consta del sensor, la placa y el microcontrolador, que adquirirán los datos cuando un usuario coloque su dedo sobre el dispositivo. Para el diseño de la estructura se emplea el software “SolidWorks”, tomando en cuenta las medidas del dispositivo para posteriormente imprimirlo en 3D.

Se ha diseñado el contenedor que se presenta en la figura 21, donde se puede apreciar las vistas superior, inferior, lateral y frontal, acompañado de las medidas con las que se ha modelado.

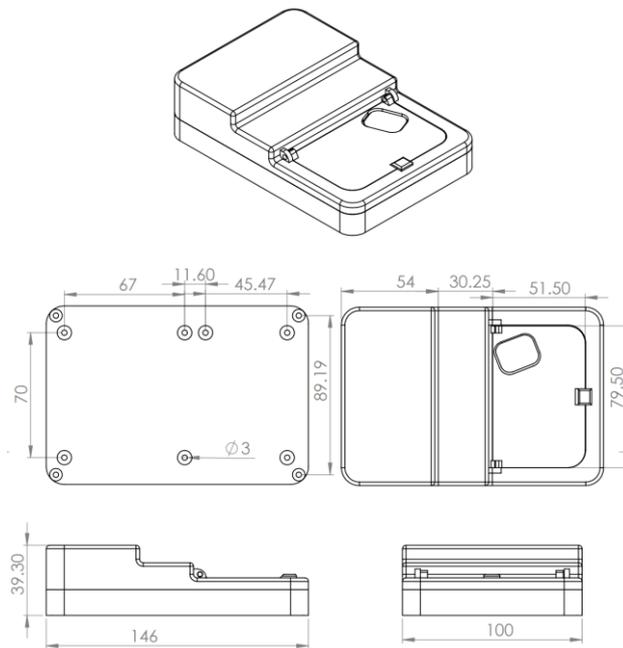


Figura 21 Diseño de la Estructura del Dispositivo Electrónico.

Fuente: Investigador

1.9.8 Calibración del Sensor AS7038RB

El sensor AS7038RB a pesar de poseer un tamaño relativamente pequeño, posee múltiples características que pueden ser modificadas, tal es el caso de la intensidad de los Leds, por ejemplo. De estas simulaciones se empleará la salida de fotoplestimografía, (PPG) para la obtención de la presión arterial, en la figura 22 se aprecia la interfaz gráfica del software del AS7038RB.

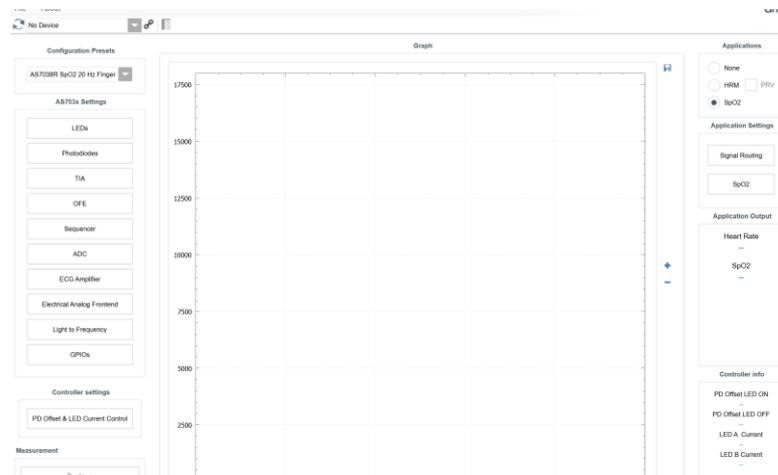


Figura 22 Software para la calibración y evaluación del Sensor AS7038RB.

Fuente: Investigador

En la figura 23 se puede apreciar la señal de PPG obtenida con el dispositivo electrónico, donde las líneas de color rojo y verde son representaciones de las señales amplificadas de transimpedancia (TIA) 1 y 2, las cuales tienen la función de convertir las señales de corriente que se han obtenido gracias a los fotodiodos del sensor en señales de tensión. En el eje de las abscisas se tiene el tiempo y en el eje de las ordenadas el conteo de las muestras de señales analógicas. La amplitud más alta se considera la presión sistólica, y la más baja la presión diastólica. La línea punteada de color gris ubica en el eje de las ordenadas representa un cambio en la corriente del led. El sensor toma estos datos y mediante un algoritmo propio determina la presión arterial sistólica y diastólica.

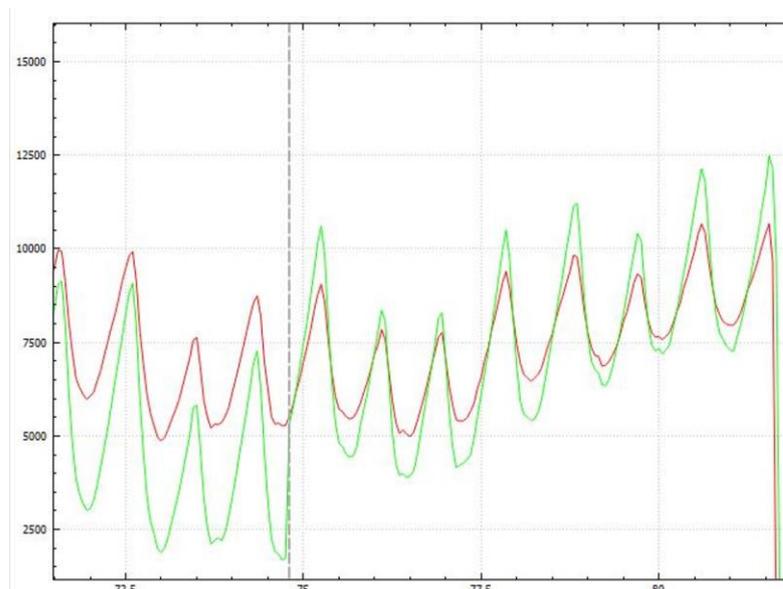


Figura 23 PPG Obtenida por el Sensor AS7038RB.

Fuente: Investigador

La presión comúnmente se toma con un esfigmomanómetro, así que para validar los datos obtenidos por parte del sensor se empleará este dispositivo médico. Se procede a tomar la presión arterial empleando el esfigmomanómetro y también el sensor AS7038RB, tomando en cuenta las recomendaciones detalladas en el apartado 1.3.2.

Tabla 16 Valores de Presión Obtenidos con el Sensor AS7038RB y el Esfingomanómetro.

Esfingomanómetro		Sensor	
Sistólica	Diastólica	Sistólica	Diastólica
113	64	117	65
118	66	120	69
115	68	112	70
111	64	108	69

Fuente: Investigador

Al comparar los valores de presión medidos por un esfingomanómetro y los obtenidos por el sensor, que se han representado en la tabla 16, se determinó que existe un error máximo de +/- 4mmHg, por lo que el dispositivo proporciona fidelidad en los datos capturados, validando así la configuración por defecto del sensor.

1.9.9 Adquisición de Datos

Para la adquisición de datos se desarrolla un script en Python empleando la librería “Pyserial” que permite la recolección de datos por comunicaciones USB. En el script se ingresan parámetros para establecer la comunicación como el puerto COM que emplea el microcontrolador junto con la velocidad en baudios de la comunicación, esto se aprecia en una parte del código de la figura 24.

```

Import serial
sensor = serial.Serial(port='COM4', baudrate=115200, timeout=1,
                      write_timeout=1)
    
```

Figura 24 Código para la comunicación serial

Fuente: Investigador

Una vez estos datos son recibidos se almacenan en variables globales para que puedan ser usadas por otros métodos. Los datos que se adquieren son los valores de presión sistólica y diastólica. Estos se obtienen empleando la librería de Serial con el método “get.value”, la trama de datos debe ser procesada y convertida a variables de tipo entero.

1.9.10 Asistente Virtual

El asistente virtual será desarrollado haciendo uso del lenguaje de programación Python, junto con librerías de uso libre, que deben ser instaladas e importadas dentro del programa. En la tabla 17 se detallan las librerías que se emplean junto con sus funciones.

Tabla 17 Librerías Empleadas para el Asistente Virtual.

Librería	Descripción
Speech_recognition	Permite al script escribir lo que se habla.
Subprocess	Abre procesos en segundo plano.
Pytsx3	Convierte el texto a voz.
Pywhatkit	Reproducción de contenido multimedia en internet.
Wikipedia	Consultas para la base de información en Wikipedia.
Datetime	Tiempo de la máquina o host.
Keyboard	Permite realizar acciones con el teclado.
Time	Ayuda a dar tiempos de espera a los scripts.
Joblib	Posibilita cargar los modelos de Inteligencia Artificial entrenados.
Pandas	Gestiona base de datos.
Numpy	Realiza operaciones matemáticas.
Tkinter	Diseña interfaces gráficas.
PIL	Carga imágenes en distintos formatos.
Pygame	Reproduce contenido multimedia.
Threading	Ejecuta procesos mediante hilos.
Mysql.connector	Administrador de base de datos para Mysql.
Pyserial	Habilita la comunicación serial.

Fuente: Investigador

Para la interfaz gráfica se empleó la librería “Tkinter” en la cual se especifica las dimensiones de la ventana que se va a generar, junto con parámetros de diseño como colores de contorno y de fondo, inclusión de imágenes, etc. Al tratarse de una ventana que servirá de guía para el usuario, se agregan mediante el uso de “canvas” consejos de uso de la aplicación y recomendaciones antes de la toma de la presión arterial, como se observa en la figura 25.



Figura 25 Interfaz Gráfica del Asistente Virtual.

Fuente: Investigador

Es necesario agregar botones para poder activar ciertas funciones dentro la interfaz, para ello se emplea el método “Button” incluido dentro de Tkinter, como se representa en la figura 26. Se especifican para cada uno de los botones sus posiciones, colores, dimensiones, contenidos de texto y el comando que los mismos invocan.

```
button_medir = Button(main_window, text="Medir Presión", fg="white",
bg="#F27121", font=("Arial", 14, "bold"), command=abrirventana2)
button_medir.pack(pady=35)
```

Figura 26 Botones dentro de la interfaz gráfica

Fuente: Investigador

Al momento de emplear la opción de “Hablar”, la máquina que aloja el programa habilitará el micrófono, escuchando y cumpliendo con las órdenes que el usuario comunique. Para lograrlo se implementó una función con la librería “Speech_recognition”, que habilita un algoritmo de inteligencia artificial para convertir lo que se diga por el micrófono a texto simple dentro del programa, tal como se aprecia en la figura 27.

```
def listen():
    try:
        with sr.Microphone() as source:
            talk("Dime. En que te ayudo")
            pc = listener.listen(source)
            rec = listener.recognize_google(pc, language="es")
            rec = rec.lower()
            if name in rec:
                rec = rec.replace(name, '')
    except:
        pass
    return rec
```

Figura 27 Función para la interpretación auditiva del asistente virtual

Fuente: Investigador

Con el texto dentro del programa solamente se usaron comparadores sencillos para invocar las funciones que el usuario requiera. Por ejemplo, decir “Medir Presión”, se ejecutará dicha función. El asistente virtual se comunica con el usuario hablando, para ello se empleó la librería “Pytt3x3”, que usa internet y algoritmos de inteligencia artificial para convertir el texto a voz, esta librería funciona en base a los fonemas.

```
def talk(text):
    engine.say(text)
    engine.runAndWait()
```

Figura 28 Función para la comunicación con el asistente virtual

Fuente: Investigador

Se integró una segunda ventana con el objetivo de dar inicio a toma de datos, en ella se incluyen botones que pueden ser accionados mediante comandos de voz. Esta ventana se aprecia en la figura 29 y las funciones de sus botones en la tabla 18.

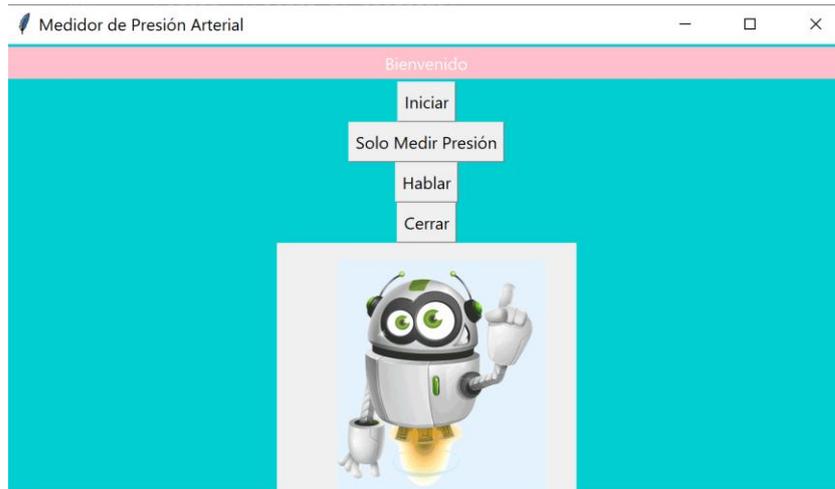


Figura 29 Segunda Ventana del Asistente Virtual.

Fuente: Investigador

Tabla 18 Función de los botones de la Segunda Ventana.

Botón	Descripción
Iniciar	Comienza el asistente virtual con las preguntas e interactúa con el usuario.
Solo Medir Presión	Toma la presión arterial de forma directa.
Hablar	Habilita el micrófono para emitir ordenes por comandos de voz.
Cerrar	Cierra el programa.

Fuente: Investigador

Una vez que se han registrado todos los datos necesarios, se diseñó una tercera ventana en la cual se podrá apreciar el diagnóstico emitido, para ello se emplean los “labels” de Tkinter. Para todas estas etiquetas se especifican dimensiones, contenido, diseño de colores y ubicación dentro la ventana, con las configuraciones de la figura 30.

```

main_window.withdraw()
win2=tk.Toplevel()
win2.geometry('1500x1300')
win2.title("Reporte Final")
win2.configure(background='white')
etiqueta3 = tk.Label(win2, text="Diagnóstico Final", bg="pink",
fg="white")
etiqueta3.pack(padx=10,pady=10, ipadx=10, ipady=10, fill=tk.X)

label_nombre = tk.Label(win2, text=("Nombre: "), bg="blue",
fg="white")
label_nombre.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

```

Figura 30 Asignación de etiquetas para la ventana de diagnóstico

Fuente: Investigador

En la figura 31 se observa el diseño final de la ventana de diagnóstico donde constan todos los campos que con la toma de datos se irán llenando con el fin de emitir una información completa.

Figura 31 Reporte Final del Asistente Virtual.

Fuente: Investigador

1.9.11 Clasificadores con Inteligencia Artificial

Después de analizar los tipos de aprendizaje en inteligencia artificial, se ha optado por emplear un clasificador que usa árboles de decisión, debido a las ventajas y desventajas analizadas en la tabla 19.

Tabla 19 Ventajas y Desventajas de los Árboles de Decisión [33].

Ventajas	Desventajas
Son algoritmos fáciles de interpretar, debido al uso de variables booleanas.	Cuando se trabaja con árboles de decisión complejos, los nuevos datos que se ingresan para el aprendizaje no se absorben de buena forma. Se requieren reajustes como la eliminación de datos innecesarios o añadir información cuando es escasa.
La preparación de datos es muy sencilla, los valores pueden ser convertidos a datos simples con valores booleanos.	Un ligero cambio en los datos cambia totalmente la estructura del árbol, por lo que se pierde fidelidad en las clasificaciones.
Aplicaciones muy efectivas en tareas que requieran clasificación.	Su entrenamiento es más costoso, dado que estos algoritmos buscan los caminos necesarios para lograr predicciones acertadas.

1.9.12 Clasificador para Personas Hipertensas

Para el diseño y entrenamiento del clasificador de personas hipertensas en el sistema de monitoreo, se empleó una base de datos con un total de 70000 entradas, que se observa en la figura 32, y cada una de ellas, ya asignada un caso específico de presión sanguínea. En total se pueden identificar 7 casos, los cuales se ven representados, junto con sus respectivos rangos de medida de presión en la tabla 20.

Tabla 20 Clasificación de los Casos de Presión Arterial [9].

CATEGORÍA	SISTÓLICA	DIASTÓLICA	CLASIFICACIÓN
1	<120	<80	ÓPTIMA
2	120-129	80-84	NORMAL
3	130-139	85-89	NORMAL-ALTA
4	140-159	90-99	HTA DE GRADO 1
5	160-179	100-109	HTA DE GRADO 2
6	≥180	≥110	HTA DE GRADO 3
7	≥140	<90	HTA SISTÓLICA ASILADA

Se empleó la librería de “Pandas” en Python para cargar la base de datos. Al emplear esta librería se puede apreciar la cantidad de entradas que contiene la base de información, así como la representación de las filas y columnas.

	ap_hi	ap_lo	DIAGNOSTICO	
id				
0	120	80	2	<class 'pandas.core.frame.DataFrame'>
1	140	90	4	Int64Index: 70000 entries, 0 to 99999
2	130	70	3	Data columns (total 3 columns):
3	150	100	4	# Column Non-Null Count Dtype
4	100	60	1	--- ---
				0 ap_hi 70000 non-null int64
				1 ap_lo 70000 non-null int64
				2 DIAGNOSTICO 70000 non-null object
				dtypes: int64(2), object(1)
				memory usage: 2.1+ MB

Figura 32 Base de Datos para el Entrenamiento Del Clasificador de Personas Hipertensas.

Fuente: Investigador

Se separó las variables predictoras de las variables a predecir, en este caso las variables de presión tanto alta como baja entran en la categoría de variables predictoras, y la información de diagnóstico se clasifica como una variable a predecir. Se indicó al modelo de entrenamiento las variables antes mencionadas, sin antes separar del total de los datos un 32 por ciento de los mismos, de forma aleatoria, para evaluar la precisión del algoritmo que se va a generar. Se dispuso de un total de 54600 entradas de tipo entero para el entrenamiento. Mediante el uso de la librería “DecisionTreeClassifier”, que se encuentra en el paquete de “Sklearn”, se especifican las variables de entrenamiento y los datos.

Con el algoritmo entrenado, se generó la matriz de confusión, empleando los datos que se dejaron para evaluación, es decir un 32% de la base total, básicamente compara los

datos que el árbol predijo con los datos que debía predecir, para de esta forma tener una idea de que tan confiable es el modelo. En este caso, se empleó una base de datos bastante generosa, y como las variables a predecir van de 1 a 7, la efectividad del algoritmo es de 1. Esta efectividad se obtiene gracias a la matriz de confusión generada en la figura 33, misma que a simple vista posee falsos positivos y falsos negativos en 0, lo cual indica que sus predicciones son correctas. También se obtiene mediante una sumatoria que relaciona la división de los valores positivos de la diagonal de la matriz de confusión entre cada uno de los valores conocidos como falsos positivos y falsos negativos.

```
In [29]: from sklearn.metrics import confusion_matrix
Matriz_de_Confesion = confusion_matrix(Y_test, Y_pred)
Matriz_de_Confesion

Out[29]: array([[ 1,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 2173,  0,  0,  0,  0,  0,  0],
 [ 0,  0, 6217,  0,  0,  0,  0,  0],
 [ 0,  0,  0, 2839,  0,  0,  0,  0],
 [ 0,  0,  0,  0, 2303,  0,  0,  0],
 [ 0,  0,  0,  0,  0,  707,  0,  0],
 [ 0,  0,  0,  0,  0,  0, 198,  0],
 [ 0,  0,  0,  0,  0,  0,  0, 962]], dtype=int64)

In [30]: import numpy as np
Precision_Global = np.sum(Matriz_de_Confesion.diagonal())/np.sum(Matriz_de_Confesion)
Precision_Global

Out[30]: 1.0
```

Figura 33 Precisión del Algoritmo Clasificador de Personas Hipertensas.

Fuente: Investigador

Una vez se ha finalizado con el entrenamiento, se procedió a guardar el algoritmo entrenado para poderlo usar en el asistente virtual. Para ello, se empleó la librería “joblib”, donde se le indica el modelo que se ha entrenado y el nombre con el cual el mismo será guardado. Todos los modelos entrenados deben ser almacenados con la extensión “pkl” como se muestra en la figura 34.

```
In [47]: import joblib
joblib.dump(arbol_enfermedad, 'modelo_hipertension.pkl')

Out[47]: ['modelo_hipertension.pkl']
```

Figura 34 Obtención del Clasificador de Personas Hipertensas.

Fuente: Investigador

1.9.13 Clasificador para Riesgos Cardíacos

Se empleó una base de datos de 70000 entradas de tipo entero, como se observa en la figura 35, posee 11 variables predictoras, y una variable a predecir cuyo tipo de dato corresponde a un valor booleano.

ID	EDAD	GENEERO	ALTURA	PESO	PA_ALTA	PA_BAJA	COLESTEROL	GLUCOSA	FUMADOR	ALCOHOL	EJERCICIO	RIESGO_CAR	EDAD_DIAS
0	50	2	168	62.0	120	80	1	1	0	0	1	0	18393
1	55	1	156	85.0	140	90	3	1	0	0	1	1	20228
2	52	1	165	64.0	130	70	3	1	0	0	0	1	18857
3	48	2	169	82.0	150	100	1	1	0	0	1	1	17623
4	48	1	156	56.0	100	60	1	1	0	0	0	0	17474

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 70000 entries, 0 to 99999
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   EDAD            70000 non-null  int64
1   GENEERO        70000 non-null  int64
2   ALTURA        70000 non-null  int64
3   PESO           70000 non-null  float64
4   PA_ALTA        70000 non-null  int64
5   PA_BAJA        70000 non-null  int64
6   COLESTEROL     70000 non-null  int64
7   GLUCOSA        70000 non-null  int64
8   FUMADOR        70000 non-null  int64
9   ALCOHOL        70000 non-null  int64
10  EJERCICIO       70000 non-null  int64
11  RIESGO_CAR     70000 non-null  int64
12  EDAD_DIAS      70000 non-null  int64
dtypes: float64(1), int64(12)
memory usage: 7.5 MB
```

Figura 35 Base de Datos para el Clasificador de Riesgo Cardiovascular.

Fuente: Investigador

Se empleó de toda la base datos, solamente un 80 por ciento de la misma para el entrenamiento del algoritmo, y el otro 20 por ciento para evaluar su fiabilidad. Estos datos son tomados de forma aleatoria, dejando un total de 56000 entradas para el entrenamiento. En base a la información tomada de la figura 36, se tomaron las 10 primeras columnas como variables predictoras y la columna 11 como la variable a predecir.

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 56000 entries, 62005 to 97497
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   EDAD         56000 non-null   int64
1   GENEERO      56000 non-null   int64
2   ALTURA      56000 non-null   int64
3   PESO         56000 non-null   float64
4   PA_ALTA     56000 non-null   int64
5   PA_BAJA     56000 non-null   int64
6   COLESTEROL  56000 non-null   int64
7   GLUCOSA     56000 non-null   int64
8   FUMADOR     56000 non-null   int64
9   ALCOHOL     56000 non-null   int64
dtypes: float64(1), int64(9)
memory usage: 4.7 MB

```

Figura 36 Datos Para el Entrenamiento del Clasificador para Riesgos Cardiovasculares.

Fuente: Investigador

Se entrenó el algoritmo, y se obtuvo la matriz de confusión para evaluar la precisión del modelo. Después de varios entrenamientos con la base de datos, se usó únicamente 4 niveles de decisión, para lograr mantener una precisión aceptable. En la figura 37, la matriz de confusión garantiza una precisión del 72 por ciento, lo cual es excelente, considerando que se trata de una salida netamente de advertencia para el usuario, más no de un diagnóstico médico final.

```

In [39]: from sklearn.metrics import confusion_matrix
Matriz_de_Confucion = confusion_matrix(Y_test, Y_pred)
Matriz_de_Confucion

Out[39]: array([[5188, 1881],
                [1933, 4998]], dtype=int64)

In [40]: #Calculo de La presiscion del modelo

import numpy as np

Precision_Global = np.sum(Matriz_de_Confucion.diagonal())/np.sum(Matriz_de_Confucion)
Precision_Global

Out[40]: 0.7275714285714285

```

Figura 37 Precisión del Clasificador Para Riesgos Cardiovasculares

Fuente: Investigador

Empleando la librería “joblib” se generó el modelo entrenado con el nombre “modelo_accidente_cardiaco”, como se observa en la figura 38.

```
In [50]: ► import joblib
joblib.dump(arbol_enfermedad, 'modelo_accidente_cardiaco.pkl')

Out[50]: ['modelo_accidente_cardiaco.pkl']
```

Figura 38 Modelo Entrenado Riesgos Cardiovasculares.

Fuente: Investigador

1.9.14 Clasificador para Riesgos de Derrame Cerebral

Si un paciente es diagnosticado con hipertensión arterial, y por el estilo de vida que lleva también es propenso a padecer algún riesgo cardiovascular, lo hace un candidato para desarrollar enfermedades cerebrovasculares, sin embargo, es importante analizar ciertos factores que pueden aumentar esta posibilidad, para ello se empleó una base de datos con un total de 4981 entradas con valores numéricos de tipo entero y decimal, la cual relaciona varios factores, algunos similares con la base de datos anterior, pero que relaciona características sociales como estado civil o el tipo de trabajo. La descripción de esta base de datos se observa en la figura 39.

	EDAD	HIPERTENSION	ENFERMEDAD_CORA	CASADO	TIPO_TRABAJO	TIPO_RESIDENCIA	GLUCOSA	IMC	FUMADOR	ACCIDENTE
GENERO										
Male	67.0	0	1	1	1	1	229	36.6	1	1
Male	80.0	0	1	1	1	0	106	32.5	0	1
Female	49.0	0	0	1	1	1	171	34.4	1	1
Female	79.0	1	0	1	2	0	174	24.0	0	1
Male	81.0	0	0	1	1	1	186	29.0	1	1

```
<class 'pandas.core.frame.DataFrame'>
Index: 4981 entries, Male to Female
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   EDAD                   4981 non-null   float64
1   HIPERTENSION           4981 non-null   int64
2   ENFERMEDAD_CORA       4981 non-null   int64
3   CASADO                 4981 non-null   int64
4   TIPO_TRABAJO          4981 non-null   int64
5   TIPO_RESIDENCIA       4981 non-null   int64
6   GLUCOSA                4981 non-null   int64
7   IMC                    4981 non-null   float64
8   FUMADOR               4981 non-null   int64
9   ACCIDENTE             4981 non-null   int64
10  ever_married           4981 non-null   object
11  TIPO_RESIDENCIA.1     4981 non-null   object
dtypes: float64(2), int64(8), object(2)
memory usage: 505.9+ KB
```

Figura 39 Base de Datos Para el Clasificador De Accidentes Cerebrales.

Fuente: Investigador

Se empleó un 78 por ciento del total de la base de datos para entrenar el algoritmo, mientras que el 32 por ciento servirá para evaluar la fiabilidad del mismo. El modelo entrenado, al ser evaluado con la matriz de confusión, mostró que tiene una precisión del 91%, como se observa en la figura 40, lo cual, para ser una información de carácter preventiva, entrega predicciones casi exactas. Este modelo se almacenó como un archivo con extensión “pkl” para su uso posterior dentro del asistente virtual.

```
In [60]: ▶ #Calculo de la presicion del modelo
import numpy as np
Precision_Global = np.sum(Matriz_de_Confucion.diagonal())/np.sum(Matriz_de_Confucion)
Precision_Global
Out[60]: 0.9124087591240876
```

Figura 40 Precisión del Clasificador Para Riesgos Cerebrales.

Fuente: Investigador

1.9.15 Visualización del Diagnóstico

Para visualizar el diagnóstico se emplea la tercera ventana. Mediante las funciones de “hablar” y “escuchar” programadas en Python, el asistente virtual entablilla una charla con el paciente para obtener datos como:

- Género
- Nombre
- Edad
- Altura
- Peso
- Nivel de colesterol en una escala general, de 1 a 3
- Nivel de glucosa en una escala general, de 1 a 3
- Si fuma
- Si consume licor
- Estado Civil
- Patologías del corazón
- Tipo de empleo

- Zona donde habita el paciente, urbana o rural.

Cada vez que usuario responda cada una de las preguntas, las mimas serán almacenadas dentro de variables en Python. La librería “joblib” permitirá cargar los clasificadores inteligentes, que en total son tres, cada uno de ellos tiene la extensión “pkl”:

- “modelo_hipertension.pkl”
- “modelo_accidente_cardiaco.pkl”
- “modelo_accidente_cerebral.pkl”

Con los modelos entrenados y cargados dentro de Python, se indican las variables predictoras, para esto, se creó un arreglo, y mediante la librería “Pandas” se generará un Frame, esto con el objetivo de que cada variable tenga de etiqueta identificadora el mismo nombre de las etiquetas de la base de datos con la que el modelo fue entrenado. En la tabla 21 se representa a cada modelo junto con los datos que se deben ingresar.

Tabla 21 Variables Para los Modelos De Predicción.

Modelo	Nombre de las Etiquetas	Variables Necesarias
Hipertensión	<ul style="list-style-type: none"> ➤ ap_hi ➤ ap_lo 	<ul style="list-style-type: none"> ➤ Presión Sistólica ➤ Presión Diastólica
Accidente Cardíaco	<ul style="list-style-type: none"> ➤ EDAD ➤ GEENERO ➤ ALTURA ➤ PESO ➤ PA_ALTA ➤ PA_BAJA ➤ FUMADOR ➤ ALCOHOL 	<ul style="list-style-type: none"> ➤ Edad ➤ Genero ➤ Altura en centímetros ➤ Peso en Kilogramos ➤ Presión Sistólica ➤ Presión Diastólica ➤ Fumador con valores de 1 o 0

		<ul style="list-style-type: none"> ➤ Alcohol con valores de 1 o 0
Accidente Cerebral	<ul style="list-style-type: none"> ➤ EDAD ➤ HIPERTENSION ➤ ENFERMEDAD_CORA ➤ CASADO ➤ TIPO_TRABAJO ➤ TIPO_RESIDENCIA ➤ IMC 	<ul style="list-style-type: none"> ➤ Edad ➤ Hipertensión con valores de 0 o 1 ➤ Enfermedades del corazón con valores de 0 o 1 ➤ Estado civil con valores de 0 o 1 ➤ Tipo de trabajo con valores de 1 a 3. ➤ Tipo de residencia con valores de 0 o 1 ➤ Índice de masa corporal

Fuente: Investigador

El asistente virtual hará uso de los modelos clasificadores para emitir el diagnóstico final de forma auditiva y textual, de tal manera que los datos se observarán como se presenta en la figura 41.



Figura 41 Visualización Del Diagnóstico.

Fuente: Investigador

1.9.18 Almacenamiento de Datos

Se almacenó la información obtenida en una base de datos, para ello se hará uso de Mysql Lite 3, debido a su sencillez y su compatibilidad con Python. En la misma, se guardará información como la fecha de toma de datos, género, edad, peso, altura, las medidas de presión, así como el diagnóstico emitido. La base de datos estará de forma local, en un archivo con extensión “db”, la cual podrá ser de libre acceso para consultas futuras. En Python esta base de datos se creó con la librería “sqlite3” y se genera siguiendo el código representado en la figura 42.

```
cur.execute("""CREATE TABLE PRESION(  
    Id AUTO_INCREMENT PRIMARY KEY,  
    Genero VARCHAR,  
    Nombre VARCHAR,  
    Edad INT,  
    Peso FLOAT,  
    Altura FLOAT,  
    Sistolica FLOAT,  
    Diastolica FLOAT,  
    Diagnostico INT,  
    Fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP)""")
```

Figura 42 Creación de la Base de Datos

Fuente: Investigador

Para el almacenamiento de los datos se asignan los valores de las variables a una cadena String y se ejecuta mediante un commit. Los signos de interrogación representan las variables a cargar, previamente deben ser almacenadas en un arreglo, el código empleado junto con la información almacena se observa en la figura 43.

```
cur.executemany("INSERT INTO PRESION VALUES(NULL, ?, ?, ?, ?, ?, ?,  
?, ?, CURRENT_TIMESTAMP)", data)  
con.commit() #Carga los datos dentro de la base
```

	Id AUTO_INCR	Genero VARCHAR	Nombre VARCHAR	Edad INT	Peso FLOAT	Altura FLOAT	Sistolica FLOAT	Diastolica FLOAT	Diagnostico INT	Fecha TIMESTAMP	
	1	(NULL)	Masculino	Juan Bayas	24	65.0	169.0	120.0	85.0	2	2023-01-25 16:44:40

Figura 43 Almacenamiento en la Base de Datos.

Fuente: Investigador

En la tabla 22 se detallan los parámetros empleados para la creación de la tabla en la base de datos.

Tabla 22 Parámetros Para La Tabla de Base de Datos.

Base de datos de Presión Arterial			
Número de Columna	Nombre	Tipo de Dato	Descripción
1	Id	PRIMARY KEY	Es la clave primera para cada dato.
2	Género	VARCHAR	Admite datos de Python del tipo String.
3	Nombre	VARCHAR	Admite datos de Python del tipo String.
4	Edad	INT	Admite datos de Python del tipo entero.
5	Peso	FLOAT	Admite datos de Python con decimales.
6	Altura	FLOAT	Admite datos de Python con decimales.
7	Sistolica	FLOAT	Admite datos de Python con decimales.
8	Diastolica	FLOAT	Admite datos de Python con decimales.
9	Diagnostico	INT	Admite datos de Python del tipo entero.
10	Fecha	TIMESTAMP	Asigna automáticamente

			la fecha y hora del ingreso de los datos.
--	--	--	---

Fuente: Investigador

1.10 Resultados

El sistema de monitorio para la detección de personas hipertensas que emplea inteligencia artificial logra que cada uno de los usuarios que hagan uso del mismo tengan una idea del estado actual de sus niveles de presión arterial, de ser el caso el asistente virtual dará a conocer el tipo de patología hipertensiva al que pertenecen. En la figura 44 se presenta el registro de algunas mediciones generadas por parte del asistente virtual, donde se aprecian distintos datos entre ellos el género, nombre, edad, peso altura y los valores de presión tanto sistólica como diastólica, así como el diagnóstico que se ha emitido.

	Id AUTO_INCR	Genero VARCHAR	Nombre VARCHAR	Edad INT	Peso FLOAT	Altura FLOAT	Sistolica FLOAT	Diastolica FLOAT	Diagnostico INT	
	4	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	5	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	6	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	7	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	8	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	9	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	10	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	11	(NULL)	Masculino	Gabriel	38	75.0	169.0	120.0	82.0	2
	12	(NULL)	Masculino	juan carlos	24	110.0	169.0	116.0	55.0	1
	13	(NULL)	Masculino	david	21	68.0	170.0	119.0	84.0	1
	14	(NULL)	Masculino	carlos	48	98.0	167.0	113.0	52.0	1
	15	(NULL)	Masculino	denis	23	82.0	171.0	115.0	86.0	3
	16	(NULL)	Femenino	teresa	42	72.0	163.0	136.0	84.0	3
	17	(NULL)	Femenino	veronica	21	69.0	163.0	115.0	72.0	1
	18	(NULL)	Femenino	karen	23	71.0	164.0	120.0	64.0	2
	19	(NULL)	Masculino	william	26	85.0	172.0	115.0	62.0	1
	20	(NULL)	Femenino	maria	33	72.0	168.0	114.0	56.0	1

Figura 44 Datos tomados por el sistema de monitoreo

Fuente: Investigador

1.11 Pruebas de funcionamiento

Para validar que el sistema de monitoreo para personas hipertensas funciona de manera correcta se realiza pruebas de medición con 10 personas de distinto sexo y edad, donde se registraron medidas en tres horarios distintos horarios. Nuevamente se hará uso de un esfigmomanómetro para comparar los valores medidos por el asistente y por este aparato de uso médico común. En las tablas 23, 24 y 25 se puede apreciar las tomas de estas medidas junto con sus comparaciones durante tres períodos del día.

Tabla 23 Medidas De Presión Durante La Mañana.

Datos del Paciente			Esfigmomanómetro		Sistema de Monitoreo de Presión Arterial	
Hora	Nombre	Edad (años)	Sistólica (mmHg)	Diastólica (mmHg)	Sistólica (mmHg)	Diastólica (mmHg)
07:00	David	21	118	80	119	84
07:20	Carlos	48	110	50	113	52
08:00	Dennis	23	110	80	115	86
08:25	Teresa	42	135	80	136	84
08:40	Verónica	21	113	75	115	72
08:55	Karen	23	118	60	120	64
09:00	William	26	115	60	115	62
09:15	María	33	110	55	114	56
09:30	Juan	22	105	60	108	62
09:45	Martha	23	110	60	111	62

Fuente: Investigador

Tabla 24 Medidas De Presión Durante La Tarde.

Datos del Paciente			Esfigmomanómetro		Sistema de Monitoreo de Presión Arterial	
Hora	Nombre	Edad (años)	Sistólica (mmHg)	Diastólica (mmHg)	Sistólica (mmHg)	Diastólica (mmHg)
12:20	David	21	125	86	123	85
12:55	Carlos	48	118	49	120	54

13:25	Dennis	23	117	89	116	85
13:55	Teresa	42	141	85	139	86
14:15	Verónica	21	119	65	122	69
14:35	Karen	23	122	75	120	78
14:55	William	26	122	64	118	68
15:20	María	33	113	65	116	69
15:50	Juan	22	109	76	111	74
16:20	Martha	23	108	65	112	69

Fuente: Investigador

Tabla 25 Medidas De Presión Durante La Noche.

Datos del Paciente			Esfigmomanómetro		Sistema de Monitoreo de Presión Arterial	
Hora	Nombre	Edad (años)	Sistólica (mmHg)	Diastólica (mmHg)	Sistólica (mmHg)	Diastólica (mmHg)
19:10	David	21	120	80	120	84
19:30	Carlos	48	115	60	113	56
20:05	Dennis	23	112	75	115	79
20:20	Teresa	42	130	75	134	79
20:45	Verónica	21	110	75	113	76
21:10	Karen	23	120	65	118	66
21:30	William	26	110	60	111	65
21:55	María	33	115	70	119	75
22:15	Juan	22	105	65	110	60
22:35	Martha	23	115	55	110	50

Fuente: Investigador

La medida obtenida por ambos dispositivos no posee grandes diferencias, esto debido a que también antes se ha calibrado de cierta forma el sensor. Sin embargo, en algunas medidas registradas, los rangos de errores fueron muy elevados, esto debido a una presión muy fuerte sobre el sensor, un aspecto muy importante a considerar. En la mayoría de las tomas de presión el error máximo presentado es de +/-4 mmHg,

independientemente del horario en el cual se tomen las medidas, por lo que el sistema de monitoreo es fiable en ese aspecto, sin embargo, muy pocas ocasiones este valor excede por un máximo de ± 6 mmHg, no obstante aun presentándose ese error poco común, el sistema de monitoreo sigue siendo fiable debido a que el rango permitido de fallo se extiende hasta los ± 7 mmHg.

1.12 Presupuesto

En la tabla 26 se detalla el presupuesto empleado en el proyecto, mediante la cual se especifican los costos de los materiales e implementación, dando un total de \$472,00, que es un costo aceptable considerando los riesgos a futuro que puede traer el padecer una hipertensión incontrolable y avanzada debido a los bajos controles médicos o desconocimiento total de esta enfermedad por su naturaleza asintomática.

Tabla 26 Tabla de Presupuesto.

Ítem	Descripción	Unidades	Cantidad	Valor Unitario	Valor Total
1	Particle Argon	c/u	1	\$35,00	\$35,00
2	Eval Kit AS703XX	c/u	1	\$289,00	\$289,00
3	AS7038RB SMOD	c/u	1	\$40,00	\$40,00
4	Cable USB a Micro USB	c/u	1	\$3,00	\$3,00
5	Caja Impresa 3D	c/u	1	\$25,00	\$25,00
6	Mano de Obra	c/u	1	\$50,00	\$50
Subtotal					\$442,00
Gastos Varios					\$30,00
Total					\$472,00

Fuente: Investigador

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

1.13 Conclusiones

Con el diseño e implementación de un sistema de monitoreo para la detección de personas hipertensas haciendo uso de inteligencia artificial se obtuvieron las siguientes conclusiones:

- Se analizó los métodos para la obtención de la presión arterial y se determinó que una de las técnicas más sencillas, compactas y fiables que existen son las que emplean fotoplestimografías o PPG, por ser un método no invasivo para el cuerpo humano, requiere únicamente posicionar el dedo o arteria sobre un pequeño dispositivo que trabaja con emisiones de luz, permitiendo obtener precisión de gran fidelidad en las medidas que se obtienen. Gracias a esta tecnología se concluyó que es posible su aplicación dentro del campo comercial debido a que muchos sensores cumplen con estándares de la IEEE.
- El dispositivo electrónico se ha diseñado haciendo uso de una placa de desarrollo compatible con comunicación USB, permitiendo velocidades de transmisión de 12 Mbps, junto con un sensor que realiza medidas con errores en promedio de +/- 4 mmHg, estos errores se reducen o aumentan dependiendo de la posición y fuerza con la que el dedo es colocado sobre el sensor. Gracias a las etapas de amplificación y filtrado de la placa de desarrollo donde se acondiciona la señal del sensor permite tomar datos incluso en lugares donde la luz es fuerte y también realiza medidas sobre todo tipo y color de piel.
- El asistente virtual se ha desarrollado empleando el lenguaje de programación Python, haciendo uso de múltiples librerías que ocupan pocos recursos del host donde se ejecuta, de tal forma que toma tiempos cortos de entre uno y dos segundos para solventar dudas o ejecutar tareas que el usuario requiera, dependiendo también de las características y velocidad del internet del equipo

huésped. El uso de modelos entrenados dentro del asistente virtual permitió aumentar la velocidad de respuesta, permitiendo realizar predicciones de forma rápida y ahorrando recursos de la máquina donde se ejecuta el programa.

1.14 Recomendaciones

- El sensor AS7038RB tiene requisitos para poder afinar aún más la precisión de sus valores emitidos, dos de ellas muy importantes recaen en el hecho del calor de los dedos de los usuarios, deben estar calientes antes de posicionarse sobre el sensor, otro aspecto es la distancia con el aparato electrónico, dedo no debe pegarse tanto al mismo, se debe dejar una presión normal entre el sensor y el dedo, evitando presionarlo con fuerza.
- El asistente virtual requiere de internet para poder realizar consultas al momento de que se le indique alguna orden o se inicie la comunicación, ya que la librería que hace todo esto posible realiza consultas en la nube, y mediante una matriz de posibilidades estima cuál es la palabra que se ha dicho y cuál es la que el asistente ha comprendido, dicho esto, la velocidad con la cual el asistente virtual responda o ejecute acciones, dependerá directamente de la velocidad del internet del usuario.
- A pesar de emitir valores de presión arterial con errores mínimos, no hay porque darle la totalidad de la credibilidad al asistente virtual, ya que el mismo para sus diagnósticos ha aprendido de bases de datos donde se han registrado miles de personas, y en medicina, se considera a cada ser humano un mundo diferente, por lo que, si bien es cierto, alguna persona puede tener sus valores de presión elevados, no quiere decir que la misma padezca hipertensión, muchas ocasiones los nervios, los problemas, la ansiedad o el estrés aumentan estos valores, en otros casos pacientes con pequeñas arritmias cardíacas pueden

presentar valores altos en su presión arterial, sin embargo son datos que cambian constantemente con el pasar del tiempo. Es recomendable tomar cada diagnóstico emitido como una advertencia, y tomar conciencia sobre la importancia de la presión arterial.

BIBLIOGRAFÍA

- [1] F. Placencia, «Indumentaria Electrónica Para El Monitoreo Continuo De Señales,» Repositorio UTA, Ambato, 2016.
- [2] G. Pujo y J. Pallo, «Repositorio de la Universidad Técnica de Ambato,» 2018. [En línea]. Available: <https://repositorio.uta.edu.ec/handle/123456789/27799>. [Último acceso: 26 12 2022].
- [3] D. Gonzalez y D. Vargas, «Medidor de Presión Arterial con Inyección Automática,» 04 05 2018. [En línea]. Available: <https://revistas.ulatina.ac.cr/index.php/gaudeamus/article/view/188/186>. [Último acceso: 27 06 2022].
- [4] C. A. Juárez, «Repositorio UCA,» 2020. [En línea]. Available: <http://repositorio.uca.edu.sv/jspui/bitstream/11674/2694/1/25-investigacion-uca-dispositivo-biosenales.pdf>. [Último acceso: 01 06 2022].
- [5] M. Herrera y E. Meza, «Repositorio Académico UPC,» 15 10 2020. [En línea]. Available: <https://renati.sunedu.gob.pe/handle/sunedu/3003995>. [Último acceso: 10 12 2022].
- [6] M. J. D. Mitri, «Universidad Católica de Córdoba,» 2019. [En línea]. Available: http://pa.bibdigital.ucc.edu.ar/1703/1/TF_DeMitri.pdf. [Último acceso: 28 12 2022].
- [7] «Fundación Española del Corazón,» [En línea]. Available: <https://fundaciondelcorazon.com/prensa/notas-de-prensa/2567-hipertension-mata-cada-ano-a-75-millones-de-personas-en-el-mundo.html#:~:text=en%20el%20mundo-,La%20hipertensi%C3%B3n%20mata%20cada%20a%C3%B1o%20a%207%2C5,de%20pe>. [Último acceso: 10 12 2022].
- [8] F. Contreras, M. Rivera, M. d. l. Parte, S. Rodríguez, O. Mendez, A. Papaprieto, J. Serrano y M. Velasco, «SciELO,» 01 2000. [En línea]. Available:

http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S0798-04692000000100003. [Último acceso: 20 12 2022].

- [9] A. L. Farré y C. M. Miguel, Libro De La Salud Cardiovascular Del Hospital Clínico San Carlos y La Fundación BBVA, Bilbao: Editorial Nerea, 2009.
- [10] R. Busoch y H. González, «Gale Academic Onefile,» 02 01 2001. [En línea]. Available:
<https://go.gale.com/ps/i.do?id=GALE%7CA146633359&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=02585944&p=AONE&sw=w&userGroupName=anon%7E33eb6d98>. [Último acceso: 02 01 2023].
- [11] «Enfermería Creativa,» 26 03 2021. [En línea]. Available:
<https://enfermeriacreativa.com/2021/03/26/monitorizacion-tension-arterial-invasiva/>. [Último acceso: 03 01 2023].
- [12] «Salud Comila de México,» [En línea]. Available:
https://saludcolima.gob.mx/images/documentos/5_e_METODO%20DE%20VERIFICACION%20PARA%20ESFIGMOMANOMETROS.pdf. [Último acceso: 04 01 2023].
- [13] J. A. Zumalacárregui, 12 11 2016. [En línea]. Available:
https://www.tuotromedico.com/temas/como_tomarse_la_tension.htm. [Último acceso: 04 01 2023].
- [14] «Fundación Española del Corazón,» 12 05 2018. [En línea]. Available:
<https://fundaciondelcorazon.com/blog-impulso-vital/2467-como-medir-la-tension-correctamente.html#:~:text=Habitualmente%20la%20toma%20de%20la,m%C3%A9todo%20auscultatorio%20y%20el%20oscilom%C3%A9trico..> [Último acceso: 05 01 2023].
- [15] B. Williams, G. Mancía, W. Spiering, E. A. Rosei, M. Azizi, M. Burnier, D. Clement, A. Coca, G. d. Simone, A. Dominiczak, T. Kahan, F. Mahfoud, J. Redon, L. Ruilope, A. Zanchetti y M. Kerins, «Peroxfarma,» 2018. [En línea].

Available: https://peroxfarma.com/wp-content/uploads/2022/07/Descargate_la_guia.pdf. [Último acceso: 02 01 2023].

- [16] D. Kasper, A. Fauci, S. Hauser, D. Longo, L. Jameson y J. Loscalzo, Harrison Manual de Medicina, México: Mc Graw Hill, 2017.
- [17] P. E. M. Soca y Y. S. Teruel, «Scielo,» 09 2009. [En línea]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=s1024-94352009000900007. [Último acceso: 26 12 2022].
- [18] J. D. L. R. Ferrera y M. A. Silva, «Scielo,» 02 06 2017. [En línea]. Available: http://scielo.sld.cu/scielo.php?pid=S1025-02552017000300007&script=sci_arttext&tlng=en. [Último acceso: 30 12 2022].
- [19] R. G. Aboado y A. R. Benavides, «Scielo,» 15 01 2010. [En línea]. Available: http://www.scielo.org.pe/scielo.php?pid=S1728-59172010000100009&script=sci_arttext&tlng=pt. [Último acceso: 30 12 2022].
- [20] L. Corona, G. Abarca y J. M. Carreño, Sensores y actuadores, Azcapotzalco: Patria, S.A. DE C.V., 2014.
- [21] D. M. Sánchez y M. d. M. E. Pérez, «Repositorio Biblus,» [En línea]. Available: <https://biblus.us.es/bibing/proyectos/abreproy/90024/fichero/Memoria.pdf>. [Último acceso: 05 01 2023].
- [22] G. G. Celi y M. E. Yapur, «ResearchGate,» 12 11 2011. [En línea]. Available: https://www.researchgate.net/publication/277801575_Mediciones_fotopletismograficas. [Último acceso: 06 01 2023].
- [23] J. C. B. Robles, «Repositorio del Instituto Politecnico Nacional de México,» 19 02 2015. [En línea]. Available: https://tesis.ipn.mx/bitstream/handle/123456789/19562/Presi%C3%B3n_Sist%C3%B3lica.pdf?sequence=1&isAllowed=y. [Último acceso: 06 01 2023].

- [24] M. R. Mañero, «Fundación Española del Corazón,» [En línea]. Available: <https://fundaciondelcorazon.com/informacion-para-pacientes/metodos-diagnosticos/electrocardiograma.html>. [Último acceso: 06 01 2023].
- [25] R. Pai, «Cigna,» 10 01 2022. [En línea]. Available: <https://www.cigna.com/es-us/knowledge-center/hw/componentes-e-intervalos-en-un-electrocardiograma-zm2308>. [Último acceso: 06 01 2023].
- [26] «IEEE,» 2022. [En línea]. Available: <https://ieeespain.org/quienes-somos/>. [Último acceso: 06 01 2023].
- [27] I. E. i. M. a. B. Society, «IEEE,» 21 08 2014. [En línea]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6882122>. [Último acceso: 06 01 2023].
- [28] L. G. López, «Educación 3.0,» [En línea]. Available: <https://www.educaciontrespuntocero.com/tecnologia/placas-de-programacion-2/#:~:text=Las%20placas%20de%20programaci%C3%B3n%20o,'inteligentes'%2C%20incluidos%20robots..> [Último acceso: 06 01 2023].
- [29] M. Boden, Inteligencia Artificial, Madrid: Turner Publicaciones, 2016.
- [30] L. Rouhiainen, «Planeta del Libro,» 2018. [En línea]. Available: https://static0planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificial.pdf. [Último acceso: 06 01 2023].
- [31] «ATRIA INNOVATION,» 22 10 2019. [En línea]. Available: <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>. [Último acceso: 06 01 2023].
- [32] T. Avila, P. Mayer y V. Quesada, «ELSEIVER,» 11 06 2020. [En línea]. Available: <https://reader.elsevier.com/reader/sd/pii/S0212656720301451?token=4DD0A94D2CB113B04110C5AB9692A3CA1D326EE5D442408AF0B29DC806E64604EBF25EE61BCD771E52CF6EDC35ED1439&originRegion=us-east-1&originCreation=20230130165358>. [Último acceso: 06 01 2023].

- [33] IBM, «IBM,» [En línea]. Available: <https://www.ibm.com/es-es/topics/decision-trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%20internos%20y%20nodos%20hoja..> [Último acceso: 06 01 2023].
- [34] F. C. Bellas, R. M. Unanue y V. F. Fernández, *Lenguajes de Programación y Procesadores*, Madrid: Centro de Estudios Ramón Areces S.A., 2016.
- [35] «AWS,» 2022. [En línea]. Available: <https://aws.amazon.com/es/what-is/python/>. [Último acceso: 07 01 2023].
- [36] V. Jimenez, «Repositorio de la Universidad de Jaume,» 2010. [En línea]. Available: <https://www3.uji.es/~vjimenez/AULASVIRTUALES/PL-0910/python-by-vilar.pdf>. [Último acceso: 06 01 2023].
- [37] C. J. Date, *Introducción a los Sistemas De Bases De Datos*, Mexico: Pearson Education, 2001.
- [38] S. D. Pérez, «Intelequia,» 09 09 2021. [En línea]. Available: [https://intelequia.com/blog/post/2949/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos#:~:text=Un%20sistema%20gestor%20de%20base%20de%20datos%20\(SGBD\)%20o%20Database,incluye%20una%20base%20de%20datos..](https://intelequia.com/blog/post/2949/gestor-de-base-de-datos-qu%C3%A9-es-funcionalidades-y-ejemplos#:~:text=Un%20sistema%20gestor%20de%20base%20de%20datos%20(SGBD)%20o%20Database,incluye%20una%20base%20de%20datos..) [Último acceso: 06 01 2023].
- [39] «OSRAM,» 15 12 2022. [En línea]. Available: <https://ams.com/vital-sign-sensing>. [Último acceso: 05 01 2023].
- [40] AMS, «AMS,» 2020. [En línea]. Available: https://ams.com/documents/20143/36005/AS7038RB_DS000726_2-00.pdf. [Último acceso: 08 01 2023].
- [41] OSRAM, «AMS,» 2020. [En línea]. Available: https://ams.com/documents/20143/4421878/AS703xxB_UG001014_1-00.pdf. [Último acceso: 08 01 2023].

- [42] Particle, «Particle Store,» 2022. [En línea]. Available: <https://store.particle.io/>.
[Último acceso: 06 01 2023].
- [43] Particle, «Particle Docs,» [En línea]. Available: <https://docs.particle.io/argon/>.
[Último acceso: 02 01 2023].

ANEXOS

ANEXO 1: Código fuente de entrenamiento del clasificador para hipertensión.

Descripción paso a paso para el entrenamiento del algoritmo encargado de la detección de hipertensión en los pacientes.

```
import pandas as pd
pacientes = pd.read_csv('C:/Users/Juan/Documents/TESIS/HIPERTENSION.csv',
engine='python', index_col=0)
pacientes.head()
```

```
Out[18]:
```

id	ap_hi	ap_lo	DIAGNOSTICO
0	120	80	2
1	140	90	4
2	130	70	3
3	150	100	4
4	100	60	1

```
pacientes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 70000 entries, 0 to 99999
Data columns (total 3 columns):
#   Column          Non-Null Count  Dtype
---  -
0   ap_hi           70000 non-null  int64
1   ap_lo           70000 non-null  int64
2   DIAGNOSTICO    70000 non-null  object
dtypes: int64(2), object(1)
memory usage: 2.1+ MB
```

```
#Variables Predictoras
```

```
X = pacientes.iloc[:,0:2]
```

```
#Variable a Predecir
```

```
Y = pacientes.iloc[:,2]
```

```
#Mostrar las primeras 5 filas
```

```
X.head()
```

```
Out[23]:
```

id	ap_hi	ap_lo
0	120	80
1	140	90
2	130	70
3	150	100
4	100	60

```
Y.head()
```

```
Out[24]: id
         0    2
         1    4
         2    3
         3    4
         4    1
         Name: DIAGNOSTICO, dtype: object
```

```
from sklearn.model_selection import train_test_split
```

```
#X_train and Y_train para entrenamiento
```

```
#Y_test and Y_test para prueba
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size = 0.78,
random_state=0)
```

```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 54600 entries, 55266 to 97497
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ---
 0   ap_hi   54600 non-null   int64
 1   ap_lo   54600 non-null   int64
dtypes: int64(2)
memory usage: 1.2 MB
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
#Se llama al constructor del arbol de decision
```

```
#arbol = DecisionTreeClassifier(max_depth=4)
```

```
arbol = DecisionTreeClassifier()
```

```
#Se entrena al modelo
```

```
arbol_enfermedad = arbol.fit(X_train, Y_train)
```

```
#Predicciones
```

```
Y_pred = arbol_enfermedad.predict(X_test)
```

```
Y_pred<
```

```
Out[28]: array(['2', '3', '2', ..., '4', '7', '3'], dtype=object)
```

```
from sklearn.metrics import confusion_matrix
```

```
Matriz_de_Confusion = confusion_matrix(Y_test, Y_pred)
```

```
Matriz_de_Confusion
```

```
Out[29]: array([[ 1,  0,  0,  0,  0,  0,  0,  0],
 [ 0, 2173,  0,  0,  0,  0,  0,  0],
 [ 0,  0, 6217,  0,  0,  0,  0,  0],
 [ 0,  0,  0, 2839,  0,  0,  0,  0],
 [ 0,  0,  0,  0, 2303,  0,  0,  0],
 [ 0,  0,  0,  0,  0, 707,  0,  0],
 [ 0,  0,  0,  0,  0,  0, 198,  0],
 [ 0,  0,  0,  0,  0,  0,  0, 962]], dtype=int64)
```

```
import numpy as np
```

```
Precision_Global
```

```
=
```

```
np.sum(Matriz_de_Confusion.diagonal())/np.sum(Matriz_de_Confusion)
```

```
Precision_Global
```

```
Out[30]: 1.0
```

```
#Prueba
```

```
nuevo_paciente = pd.DataFrame(np.array([[168,100]]),columns=['ap_hi',  
'ap_lo'])
```

```
nuevo_paciente
```

```
Out[14]:
```

	ap_hi	ap_lo
0	168	100

```
arbol_enfermedad.predict(nuevo_paciente)
```

```
Out[15]: array(['5'], dtype=object)
```

```
#Creación del Modelo
```

```
import joblib
```

```
joblib.dump(arbol_enfermedad, 'modelo_hipertension.pkl')
```

```
Out[16]: ['modelo_hipertension.pkl']
```

ANEXO 2: Código fuente de entrenamiento del clasificador para accidentes cardiovasculares.

Descripción paso a paso del entrenamiento del algoritmo para la predicción de accidentes cardiovasculares, junto con la matriz de confusión y porcentaje de fiabilidad.

```
import pandas as pd

pacientes = pd.read_csv('C:/Users/Juan/Documents/TESIS/base_presion_cardiovascular.csv',
engine='python', index_col=0)

pacientes.head()

Out[3]:
```

ID	EDAD	GENEERO	ALTURA	PESO	PA_ALTA	PA_BAJA	COLESTEROL	GLUCOSA	FUMADOR	ALCOHOL	EJERCICIO	RIESGO_CAR	EDAD_DIAS
0	50	2	168	62.0	120	80	1	1	0	0	1	0	18393
1	55	1	156	85.0	140	90	3	1	0	0	1	1	20228
2	52	1	165	64.0	130	70	3	1	0	0	0	1	18857
3	48	2	169	82.0	150	100	1	1	0	0	1	1	17623
4	48	1	156	56.0	100	60	1	1	0	0	0	0	17474

```
pacientes.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 70000 entries, 0 to 99999
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   EDAD                  70000 non-null  int64
1   GENEERO               70000 non-null  int64
2   ALTURA               70000 non-null  int64
3   PESO                 70000 non-null  float64
4   PA_ALTA              70000 non-null  int64
5   PA_BAJA              70000 non-null  int64
6   COLESTEROL           70000 non-null  int64
7   GLUCOSA              70000 non-null  int64
8   FUMADOR              70000 non-null  int64
9   ALCOHOL              70000 non-null  int64
10  EJERCICIO             70000 non-null  int64
11  RIESGO_CAR           70000 non-null  int64
12  EDAD_DIAS            70000 non-null  int64
dtypes: float64(1), int64(12)
memory usage: 7.5 MB

#Variables Predictoras
X = pacientes.iloc[:,0:10]

#Variable a Predecir
Y = pacientes.iloc[:,11]
```

```
#Mostrar las primeras 5 filas
```

```
X.head()
```

```
Out[5]:
```

ID	EDAD	GENEERO	ALTURA	PESO	PA_ALTA	PA_BAJA	COLESTEROL	GLUCOSA	FUMADOR	ALCOHOL
0	50	2	168	62.0	120	80	1	1	0	0
1	55	1	156	85.0	140	90	3	1	0	0
2	52	1	165	64.0	130	70	3	1	0	0
3	48	2	169	82.0	150	100	1	1	0	0
4	48	1	156	56.0	100	60	1	1	0	0

```
from sklearn.model_selection import train_test_split
```

```
#X_train and Y_train para entrenamiento
```

```
#Y_test and Y_test para prueba
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size = 0.80,  
random_state=0)
```

```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 56000 entries, 62005 to 97497  
Data columns (total 10 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   EDAD            56000 non-null  int64  
1   GENEERO        56000 non-null  int64  
2   ALTURA        56000 non-null  int64  
3   PESO           56000 non-null  float64  
4   PA_ALTA        56000 non-null  int64  
5   PA_BAJA        56000 non-null  int64  
6   COLESTEROL     56000 non-null  int64  
7   GLUCOSA        56000 non-null  int64  
8   FUMADOR        56000 non-null  int64  
9   ALCOHOL        56000 non-null  int64  
dtypes: float64(1), int64(9)  
memory usage: 4.7 MB
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
#Se llama al constructor del arbol de decision
```

```
arbol = DecisionTreeClassifier(max_depth=4)
```

```
#arbol = DecisionTreeClassifier()
```

```
#Se entrena al modelo
```

```
arbol_enfermedad = arbol.fit(X_train, Y_train)
```

```
#Predicciones
```

```

Y_pred = arbol_enfermedad.predict(X_test)
Y_pred

Out[38]: array([0, 0, 0, ..., 1, 0, 1], dtype=int64)

from sklearn.metrics import confusion_matrix
Matriz_de_Confusion = confusion_matrix(Y_test, Y_pred)
Matriz_de_Confusion

Out[39]: array([[5188, 1881],
                [1933, 4998]], dtype=int64)

#Calculo de la presicion del modelo

import numpy as np

Precision_Global = np.sum(Matriz_de_Confusion.diagonal())/np.sum(Matriz_de_Confusion)
Precision_Global

Out[40]: 0.7275714285714285

#Prueba de Funcionamiento

nuevo_paciente = pd.DataFrame(np.array([[80,1,169,80,140,80,2,1,1,1]]),columns=['EDAD', 'GENEERO', 'ALTURA', 'PESO', 'PA_ALTA', 'PA_BAJA', 'COLESTEROL', 'GLUCOSA', 'FUMADOR', 'ALCOHOL'])
nuevo_paciente

Out[45]:
   EDAD  GENEERO  ALTURA  PESO  PA_ALTA  PA_BAJA  COLESTEROL  GLUCOSA  FUMADOR  ALCOHOL
0     80         1     169     80     140         80           2         1         1         1

arbol_enfermedad.predict(nuevo_paciente)

Out[46]: array([1], dtype=int64)

#Generación del Modelo

import joblib
joblib.dump(arbol_enfermedad, 'modelo_accidente_cardiaco.pkl')

Out[50]: ['modelo_accidente_cardiaco.pkl']

```

ANEXO 3: Código fuente de entrenamiento del clasificador para accidentes cerebrales.

Descripción paso a paso del entrenamiento del algoritmo para la predicción de accidentes cerebrales, junto con la matriz de confusión y porcentaje de fiabilidad.

```
import pandas as pd

pacientes = pd.read_csv('C:/Users/Juan/Documents/TESIS/brain_stroke2.csv',
engine='python', index_col=0)

pacientes.head()

pacientes.info()

<class 'pandas.core.frame.DataFrame'>
Index: 4981 entries, Male to Female
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   EDAD                  4981 non-null   float64
1   HIPERTENSION          4981 non-null   int64
2   ENFERMEDAD_CORA      4981 non-null   int64
3   CASADO                4981 non-null   int64
4   TIPO_TRABAJO         4981 non-null   int64
5   TIPO_RESIDENCIA      4981 non-null   int64
6   GLUCOSA              4981 non-null   int64
7   IMC                  4981 non-null   float64
8   FUMADOR              4981 non-null   int64
9   ACCIDENTE            4981 non-null   int64
10  ever_married         4981 non-null   object
11  TIPO_RESIDENCIA.1   4981 non-null   object
dtypes: float64(2), int64(8), object(2)
memory usage: 505.9+ KB

#Variables Predictoras
X = pacientes.iloc[:,0:8]

#Variable a Predecir
Y = pacientes.iloc[:,9]

#Mostrar las primeras 5 filas

X.head()
```

Out[13]:

	EDAD	HIPERTENSION	ENFERMEDAD_CORA	CASADO	TIPO_TRABAJO	TIPO_RESIDENCIA	GLUCOSA	IMC
GENERO								
Male	67.0	0	1	1	1	1	229	36.6
Male	80.0	0	1	1	1	0	106	32.5
Female	49.0	0	0	1	1	1	171	34.4
Female	79.0	1	0	1	2	0	174	24.0
Male	81.0	0	0	1	1	1	186	29.0

```
from sklearn.model_selection import train_test_split
```

```
#X_train and Y_train para entrenamiento
```

```
#Y_test and Y_test para prueba
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, train_size = 0.78,
random_state=0)
```

```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3885 entries, Female to Female
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   EDAD                   3885 non-null   float64
1   HIPERTENSION           3885 non-null   int64
2   ENFERMEDAD_CORA       3885 non-null   int64
3   CASADO                 3885 non-null   int64
4   TIPO_TRABAJO          3885 non-null   int64
5   TIPO_RESIDENCIA       3885 non-null   int64
6   GLUCOSA                3885 non-null   int64
7   IMC                   3885 non-null   float64
dtypes: float64(2), int64(6)
memory usage: 273.2+ KB
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
#Se llama al constructor del arbol de decision
```

```
#arbol = DecisionTreeClassifier(max_depth=4)
```

```
arbol = DecisionTreeClassifier()
```

```
#Se entrena al modelo
```

```
arbol_enfermedad = arbol.fit(X_train, Y_train)
```

```
#Predicciones
```

```
Y_pred = arbol_enfermedad.predict(X_test)
```

```
Y_pred
```

```
Out[58]: array([0, 0, 0, ..., 0, 0, 1], dtype=int64)
```

```
from sklearn.metrics import confusion_matrix
```

```

Matriz_de_Confusion = confusion_matrix(Y_test, Y_pred)
Matriz_de_Confusion

Out[59]: array([[991, 51],
                [ 45,  9]], dtype=int64)

#Calculo de la presicion del modelo

import numpy as np

Precision_Global =
np.sum(Matriz_de_Confusion.diagonal())/np.sum(Matriz_de_Confusion)
Precision_Global

Out[60]: 0.9124087591240876

#Prueba

nuevo_paciente =
pd.DataFrame(np.array([[98,1,1,1,1,1,250,28.0]]),columns=['EDAD',
'HIPERTENSION', 'ENFERMEDAD_CORA', 'CASADO', 'TIPO_TRABAJO',
'TIPO_RESIDENCIA', 'GLUCOSA', 'IMC'])
nuevo_paciente

Out[66]:


|   | EDAD | HIPERTENSION | ENFERMEDAD_CORA | CASADO | TIPO_TRABAJO | TIPO_RESIDENCIA | GLUCOSA | IMC  |
|---|------|--------------|-----------------|--------|--------------|-----------------|---------|------|
| 0 | 98.0 | 1.0          | 1.0             | 1.0    | 1.0          | 1.0             | 250.0   | 28.0 |



arbol_enfermedad.predict(nuevo_paciente)

Out[67]: array([1], dtype=int64)

#Generación del Modelo

import joblib
joblib.dump(arbol_enfermedad, 'modelo_accidente_cerebral.pkl')

Out[69]: ['modelo_accidente_cerebral.pkl']

```

ANEXO 4: Código fuente correspondiente al Asistente Virtual

Se describe cada una de las librerías empleadas para el desarrollo del asistente virtual así como el diseño de la interfaz gráfica del mismo.

```
import speech_recognition as sr
import subprocess as sub
import pyttsx3
import pywhatkit
import wikipedia
import datetime
import keyboard
import colors
import time
import joblib
import pandas as pd
import numpy as np
import serial
import sqlite3
from tkinter import *
from tkinter import ttk
import tkinter as tk
from PIL import Image, ImageTk
from pygame import mixer
#import threading as tr

#carga de los modelos entrenados de IA

clf_hipertension = joblib.load('modelo_hipertension.pkl')
clf_cardiaco = joblib.load('modelo_accidente_cardiaco.pkl')
clf_cerebral = joblib.load('modelo_accidente_cerebral.pkl')

def abrirventana2():
    main_window.withdraw()
```

```

win=tk.Toplevel()
win.geometry('1500x800')
win.title("Medidor de Presión Arterial")
win.configure(background='dark turquoise')
e3 = tk.Label(win, text="Bienvenido", bg="pink", fg="white")
e3.pack(padx=5,pady=5, ipadx=5, ipady=5, fill=tk.X)
        boton3=tk.Button(win, text='Iniciar', command=lambda:[win.destroy
,iniciando()]
        boton3.pack(side=tk.TOP)
        boton4=tk.Button(win, text='Solo Medir Presión', command=solo_presion)
        boton4.pack(side=tk.TOP)
        boton5=tk.Button(win, text='Hablar', command=run_doctor)
        boton5.pack(side=tk.TOP)
        boton2=tk.Button(win, text='Cerrar', command=lambda:[win.destroy,
cerrarventana()])
        boton2.pack(side=tk.TOP)
        #Se carga la imagen GIF de la segunda ventana
        medico_gif_path = "C:/Users/Juan/Documents/TESIS/APP GUI/robot2.gif"
        info_gif = Image.open(medico_gif_path)
        gif_nframes = info_gif.n_frames
        doctor_gif_list = [PhotoImage(file=medico_gif_path, format=f'gif -index {i}')for
i in range(gif_nframes)]
        label_gif = Label(win)
        label_gif.pack()

        def animate_gif(index):
            frame = doctor_gif_list[index]
            index += 1
            if index == gif_nframes:
                index=0
            label_gif.configure(image=frame)
            win.after(50, animate_gif, index)

```

```

animate_gif(0)

def abrirventana3():
    main_window.withdraw()
    win2=tk.Toplevel()
    win2.geometry('1500x900')
    win2.title("Reporte Final")
    win2.configure(background='white')
    etiqueta3 = tk.Label(win2, text="Diagnóstico Final", bg="pink", fg="white")
    etiqueta3.pack(padx=10,pady=10, ipadx=10, ipady=10, fill=tk.X)
    boton3=tk.Button(win2, text='Iniciar', command=iniciando)
    boton3.pack(side=tk.TOP)
    boton4=tk.Button(win2, text='Solo Medir Presión', command=solo_presion)
    boton4.pack(side=tk.TOP)
    boton5=tk.Button(win2, text='Hablar', command=run_doctor)
    boton5.pack(side=tk.TOP)
    boton2=tk.Button(win2, text='Cerrar', command=lambda:[win2.destroy,
cerrarventana()]])
    boton2.pack(side=tk.TOP)

def solo_presion():
    talk("Se esta midiendo la presión")

def iniciando():
    talk("Estamos a punto de iniciar, por favor relájate y manten la calma. Recuerda
que antes de inciar con la medida, debes haber mantenido reposo por al menos 5
minutos. Ahora por favor, coloca el dedo sobre el sensor")
    h_a=116
    h_b=55
    talk("Por favor dime si perteneces al género masculino o femenino")
    genero=listen_genero()

```

```
talk("El paciente se ha identificado con el género "+genero)
talk("Muchas gracias..")
```

```
talk("Si no es mucha molestia, dime cual es tu nombre")
nombre=listen_nombre()
print=nombre
```

```
talk("Perfecto "+ nombre)
```

```
talk("Ahora dime tu edad")
edad=listen_edad()
talk("Excelente "+ nombre + "ahora sé que tienes " + edad + "años")
```

```
talk("Ayudame por favor diciéndome tu altura en centímetros")
altura=listen_altura()
talk("Mides " + altura + "centímetros")
```

```
talk("Ahora necesito saber tu peso en kilogramos. Dímelo por favor")
peso=listen_peso()
talk("Tu peso es " + peso + "kilos")
```

```
# talk("Este dato me ayudará a predecir factores de riesgo adicionales. Por favor
del uno al tres. Dime en que grado de colesterol te hallas. Siendo uno el más bajo y
tres el mas alto. Te escucho ")
```

```
# colesterol=listen_colesterol()
# talk("Has dicho el grado " + colesterol)
```

```
# talk("Nuevamente necesito que por favor cuantifiques tus niveles de glucosa.
Usa por favor las valoraciones anteriores. Donde uno es bajo y tres el mas alto")
```

```
# glucosa = listen_glucosa()
# talk("He guardado tu grado de glucosa con el número " + glucosa)
```

talk("Las siguientes dos preguntas necesito que por favor las responda con la mayor sinceridad posible")

talk("Responde con si, o con no")

talk("Usted Fuma?")

fumador=listen_fumador()

talk("Entiendo. Ahora se que usted " + fumador + "fuma")

talk("Me gustaría saber si usted consume licor")

licor=listen_licor()

talk("Acabo de guardar el dato de que usted " + licor + "consume licor")

talk("Me gustaría saber si usted practica algún deporte o hace ejercicio de forma regular. Responda con sí o no")

ejercicio=listen_ejercicio()

talk("Usted " + ejercicio + "hace ejercicio")

talk("Me gustaría saber si estado civil actual. Dígame si es casado o soltero.")

estado_civil=listen_estado_civil()

talk("Entiendo. Usted está " + estado_civil)

talk("Por favor dígame si usted padece alguna enfermedad del corazón. Responda con sí o no")

enfermedad_corazon=listen_enfermedad_corazon()

talk("Usted " + enfermedad_corazon + "padece alguna enfermedad del corazón")

talk("Ahora me gustaría saber un poco más de usted. Dígame que tipo de trabajo tiene usted. Si es un empleado público diga 1. Si tiene su propio negocio diga 2. Si es un trabajador independiente mencione 3")

trabajo=listen_trabajo()

talk("Usted se ha identificado con la clasificación de trabajo " + trabajo)

talk("Para finalizar. Dígame si su lugar de residencia se halla en una zona urbana o rural")

```

residencia=listen_residencia()
talk("He guardado el dato se que su vivienda se ubica en una zona " + residencia)
#Tratado de los datos para la interpretacion de la IA
#Se define el genero
if genero == "masculino":
    genero = 1
else:
    genero = 0

#Se convierte la edad a un numero entero
edad=int(edad)
altura=int(altura)
peso=int(peso)
colesterol=2
glucosa=2
trabajo=int(trabajo)
#calculo del indice de masa corporal
imc=peso/(altura/100)

#Paramatros para fumador
if fumador == "si":
    fumador = 1
else:
    fumador = 0

#Paramatros para licor
if licor== "si":
    licor = 1
else:
    licor = 0

#Paramatros para ejercicio
if ejercicio== "si":

```

```

ejercicio = 1
else:
    ejercicio = 0

#Paramatros para enfermedad del corazon
if enfermedad_corazon== "si":
    enfermedad_corazon = 1
else:
    enfermedad_corazon = 0

#Parametros para el estado civil
if estado_civil == "soltero":
    estado_civil= 1
else:
    estado_civil=0

#Parametros para el tipo de residencia

if residencia == "urbana":
    residencia=1
else:
    residencia=0

#Damos inicio a las predicciones

#Hipertension -AI
paciente_hipertension = pd.DataFrame(np.array([[h_a,h_b]]),columns=[
'ap_hi', 'ap_lo'])
diagnostico_hipertension = clf_hipertension.predict(paciente_hipertension)
diagnostico_hipertension = int(diagnostico_hipertension)

#Diagnostico proporcionado por la IA

```

```

if diagnostico_hipertension == 1:
    diagnostico_hipertension_final = "óptima"
    diagnostico_hipertension = 0
    hablar = "Usted tiene una presión óptima, nada de que preocuparse"
elif diagnostico_hipertension == 2:
    diagnostico_hipertension_final = "normal"
    diagnostico_hipertension = 0
    hablar = "Usted tiene una presión normal, nada de que preocuparse"
elif diagnostico_hipertension == 3:
    diagnostico_hipertension_final = "normal alta"
    hablar = "Usted tiene una presión normal alta, es normal, pero no se debe
descuidar el control y la comida"
    diagnostico_hipertension = 0
elif diagnostico_hipertension == 4:
    diagnostico_hipertension_final = "Hipertensión de Grado 1"
    diagnostico_hipertension = 1
    hablar = "Usted padece de Hipertensión en grado 1, necesita la atención de
un médico calificado"
elif diagnostico_hipertension == 5:
    diagnostico_hipertension_final = "Hipertensión de Grado 2"
    diagnostico_hipertension = 1
    hablar = "Usted padece de Hipertensión en grado 2. Debe asistir al médico de
forma urgente para proceder con un tratamiento"
elif diagnostico_hipertension == 6:
    diagnostico_hipertension_final = "Hipertensión de Grado 3"
    diagnostico_hipertension = 1
    hablar = "Usted padece de Hipertensión en grado 3. Debe asistir al médico de
forma urgente para proceder con un tratamiento"
elif diagnostico_hipertension == 7:
    diagnostico_hipertension_final = "Hipertensión Sistólica Aislada"
    diagnostico_hipertension = 1
    hablar = "Usted padece de Hipertensión Sistólica Aislada. Debe asistir al
médico de forma urgente para proceder con un tratamiento"

```

talk(hablar)

#Riesgo Cardiovascular -AI

talk("Ahora analizaré sus datos para predecir un posible riesgo cardiovascular")
time.sleep(2)

paciente_cardiovascular =
pd.DataFrame(np.array([[edad,genero,altura,peso,h_a,h_b,colesterol,glucosa,fu
mador,licor]]),columns=['EDAD', 'GENEERO', 'ALTURA', 'PESO', 'PA_ALTA',
'PA_BAJA', 'COLESTEROL', 'GLUCOSA', 'FUMADOR','ALCOHOL'])
diagnostico_cardiaco = clf_cardiaco.predict(paciente_cardiovascular)
if diagnostico_cardiaco == 1:
hablar22=("Riesgo Cardiovascular Presente")
hablar2="Según los datos analizados. Usted tiene una alta probabilidad de
padecer algún accidente grave cardiovascular. Necesita un control médico
profesional urgente"
else:
hablar22=("Riesgo Cardiovascular Ausente")
hablar2= "Usted no tiene probabilidades de padecer algún accidente
cardiovascular. Sin embargo. No olvide visitar al médico de forma regular y
mejorar su estilo de vida"
talk(hablar2)

#Riesgo de muerte cerebral -AI

talk("Ahora analizaré los datos para obtener las probabilidades de padecer algún
accidente cerebral")
time.sleep(2)

paciente_cerebral =
pd.DataFrame(np.array([[edad,diagnostico_hipertension,enfermedad_corazon,est
ado_civil,trabajo,residencia,glucosa,imc]]),columns=['EDAD', 'HIPERTENSION',

```

'ENFERMEDAD_CORA', 'CASADO', 'TIPO_TRABAJO', 'TIPO_RESIDENCIA',
'GLUCOSA', 'IMC'])

diagnostico_cerebral = clf_cerebral.predict(paciente_cerebral)
if diagnostico_cerebral == 1:
    hablar33=("Riesgo de Accidenete Cerebral Presente")
    hablar3=("Según la información obtenida debo informarle que usted es una
persona con alto riesgo de sufrir un accidente cerebral.Se recomienda acudir de
forma urgente donde un médico profesional")
else:
    hablar33=("Riesgo de Accidenete Cerebral Ausente")
    hablar3=("No posee ninguna probabilidad de padecer algún accidenete
cerebral. Sin embargo se recomienda los controles rutinarios y mejorar el estilo de
vida")

talk(hablar3)
time.sleep(1)
talk("Puede visualiazar el reporte final")

#abre la ventana número 3

main_window.withdraw()
win2=tk.Toplevel()
win2.geometry('1500x1300')
win2.title("Reporte Final")
win2.configure(background='white')
etiqueta3 = tk.Label(win2, text="Diagnóstico Final", bg="pink", fg="white")
etiqueta3.pack(padx=10,pady=10, ipadx=10, ipady=10, fill=tk.X)

label_nombre = tk.Label(win2, text=("Nombre: " + nombre), bg="blue",
fg="white")
label_nombre.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

label_edad = tk.Label(win2, text=("Edad: " + str(edad)), bg="blue", fg="white")

```

```

label_edad.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

label_peso = tk.Label(win2, text=("Peso: "+ str(peso)), bg="blue", fg="white")
label_peso.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

    label_altura = tk.Label(win2, text=("Altura: "+str(altura)), bg="blue",
fg="white")
label_altura.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

    label_sistolica = tk.Label(win2, text=("Presion Sistólica: "+str(h_a)),
bg="blue", fg="white")
label_sistolica.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

    label_diastolica = tk.Label(win2, text=("Presion Diastólica: "+str(h_b)),
bg="blue", fg="white")
label_diastolica.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

label_diagnostico = tk.Label(win2, text=(hablar), bg="blue", fg="white")
label_diagnostico.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

label_cardiovascula = tk.Label(win2, text=(hablar22), bg="blue", fg="white")
label_cardiovascula.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

label_cerebral = tk.Label(win2, text=(hablar33), bg="blue", fg="white")
label_cerebral.pack(padx=20,pady=20, ipadx=20, ipady=20, fill=tk.X)

boton5=tk.Button(win2, text='Hablar', command=run_doctor)
boton5.pack(side=tk.TOP)

#Almacenamiento en la Base de Datos

con = sqlite3.connect("PRESION_BASE.db")
cur = con.cursor()

```

#CREACION DE LA TABLA EN LA BASE DE DATOS

```
# cur.execute("""CREATE TABLE PRESION(
#     Id AUTO_INCREMENT PRIMARY KEY,
#     Genero VARCHAR,
#     Nombre VARCHAR,
#     Edad INT,
#     Peso FLOAT,
#     Altura FLOAT,
#     Sistolica FLOAT,
#     Diastolica FLOAT,
#     Diagnostico INT,
#     Fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP)""")

data = [
(genero,nombre, edad, peso, altura, h_a, h_b, diagnostico_hipertension),
#("Masculino","Gabriel", 38, 75, 169, 120, 82, 2),
#("Masculino","Gabriel", 38, 75, 169, 120, 82, 2),
]
cur.executemany("INSERT INTO PRESION VALUES(NULL, ?, ?, ?, ?, ?, ?, ?, ?, ?,
CURRENT_TIMESTAMP)", data)
con.commit() #Carga los datos dentro de la base

def cerrarventana():
    main_window.destroy()

main_window = Tk()
main_window.title("Asistente_Medico")
```

```

main_window.geometry("2200x1400")
main_window.resizable(0,0)
main_window.configure(bg='#4286f4')

label_title = Label(main_window, text="Asistente Médico", bg="#3b8d99",
fg="#ffd452", font=('Arial', 30, 'bold'))
label_title.pack(pady=10)

medico_photo = ImageTk.PhotoImage(Image.open("medico2.png"))
window_photo = Label(main_window,image=medico_photo)
window_photo.pack(pady=5)

opciones = """"
Comandos de voz que puede usar:

-Consulta Completa

-Medir la Presión

-Información sobre la Hipertensión

-"Busca" (Consulta información sobre temas médicos)

-"Reproduce" (Abre Youtube con cualquier criterio de búsqueda)

-"Cerrar" (Cierra el programa)

""""

opciones2 = """"
Consejos para la toma presión:

-Medir la presión arterial en un lugar

```

tranquilo y libre de ruidos.

*-Apoyar los pies. No apoyar los pies
agrega hasta 6 mmHg*

*-No cruzar las piernas. Tener las piernas
cruzadas agrega de 2 a 8 mm Hg*

*-Tener la vejiga vacía. Tener la vejiga
llena agrega hasta 10 mmHg*

*- No conversar. Hablar o escuchar de forma
activa agrega hasta 10 mmHg*

*-No haber fumado o ingerido licor antes de
la toma de presión.*

""

```
canvas_opciones=Canvas(bg="#E100FF", height=700, width=600)
canvas_opciones.place(x=50, y=350)
canvas_opciones.create_text(300, 300, text=opciones, fill="white", font='Arial 6')
```

```
canvas_opciones2=Canvas(bg="#E100FF", height=700, width=600)
canvas_opciones2.place(x=1550, y=350)
canvas_opciones2.create_text(300, 300, text=opciones2, fill="white", font='Arial
7')
```

```
name = "Doctor"
```

```
listener = sr.Recognizer()
```

```
engine = pyttsx3.init()
```

```
voices = engine.getProperty('voices')
```

```

engine.setProperty('voice', voices[0].id)

sites = {
    'google': 'google.com',
    'facebook': 'facebook.com'
}

def talk(text):
    engine.say(text)
    engine.runAndWait()

def listen():
    try:
        with sr.Microphone() as source:
            talk("Dime. En que te ayudo")
            pc = listener.listen(source)
            rec = listener.recognize_google(pc, language="es")
            rec = rec.lower()
            if name in rec:
                rec = rec.replace(name, "")
    except:
        pass
    return rec

def listen_nombre():
    try:
        with sr.Microphone() as source:
            talk("Te escucho")
            pc = listener.listen(source)
            rec1 = listener.recognize_google(pc, language="es")
            rec1 = rec1.lower()

```

```

        if "Mi nombre es" in rec1:
            rec1 = rec1.replace("Mi nombre es", "")
    except:
        pass
    return rec1

def listen_edad():
    try:
        with sr.Microphone() as source:
            talk("Dime únicamente el número")
            pc = listener.listen(source)
            rec2 = listener.recognize_google(pc, language="es")
            rec2 = rec2.lower()
            if "Tengo" in rec2:
                rec2 = rec2.replace("Tengo", "")
    except:
        pass
    return rec2

def listen_genero():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec3 = listener.recognize_google(pc, language="es")
            rec3 = rec3.lower()
            if "Soy" in rec3:
                rec3 = rec3.replace("Soy", "")
    except:
        pass
    return rec3

def listen_altura():
    try:

```

```

with sr.Microphone() as source:
    pc = listener.listen(source)
    rec4 = listener.recognize_google(pc, language="es")
    rec4 = rec4.lower()
    if "Centímetros" in rec4:
        rec4 = rec4.replace("Centímetros", "")
except:
    pass
return rec4

def listen_peso():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec5 = listener.recognize_google(pc, language="es")
            rec5 = rec5.lower()
            if "Kilos" in rec5:
                rec5 = rec5.replace("Kilos", "")
    except:
        pass
    return rec5

def listen_fumador():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec8 = listener.recognize_google(pc, language="es")
            rec8 = rec8.lower()
            if "fumo" in rec8:
                rec8 = rec8.replace("fumo", "")
    except:
        pass
    return rec8

```

```

def listen_licor():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec9 = listener.recognize_google(pc, language="es")
            rec9 = rec9.lower()
            if "tomo" in rec9:
                rec9 = rec9.replace("tomo", "")
    except:
        pass
    return rec9

def listen_ejercicio():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec10 = listener.recognize_google(pc, language="es")
            rec10 = rec10.lower()
            if "hago" in rec10:
                rec10 = rec10.replace("hago", "")
    except:
        pass
    return rec10

def listen_estado_civil():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec11 = listener.recognize_google(pc, language="es")
            rec11 = rec11.lower()
            if "soy" in rec11:
                rec11 = rec11.replace("soy", "")

```

```

except:
    pass
return rec11

def listen_enfermedad_corazon():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec12 = listener.recognize_google(pc, language="es")
            rec12 = rec12.lower()
            if "tengo" in rec12:
                rec12 = rec12.replace("tengo", "")
    except:
        pass
    return rec12

def listen_trabajo():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec13 = listener.recognize_google(pc, language="es")
            rec13 = rec13.lower()
            if "laboro" in rec13:
                rec13 = rec13.replace("laboro", "")
    except:
        pass
    return rec13

def listen_residencia():
    try:
        with sr.Microphone() as source:
            pc = listener.listen(source)
            rec14 = listener.recognize_google(pc, language="es")

```

```

    rec14 = rec14.lower()
    if "zona" in rec14:
        rec14 = rec14.replace("zona", "")
    except:
        pass
    return rec14

def run_doctor():
    rec = listen()
    if 'reproduce' in rec:
        music = rec.replace('reproduce', '')
        print("Reproduciendo " + music)
        talk("Reproduciendo " + music)
        pywhatkit.playonyt(music)

    elif 'busca' in rec:
        search = rec.replace('busca', '')
        wikipedia.set_lang("es")
        wiki = wikipedia.summary(search, 1)
        print(search + ": " + wiki)
        talk(wiki)

    elif 'abre' in rec:
        for site in sites:
            if site in rec:
                sub.call(f'start chrome.exe {sites[site]}', shell=True)
                talk(f'Abriendo {site}')

    elif 'medir presión' in rec:
        talk("Ahora mismo")
        abrirventana2()

    elif 'cerrar' in rec:
        talk("Hasta Pronto")
        main_window.destroy()

```

elif 'cerrar todo' in rec:

talk("Hasta Pronto")

main_window.destroy()

elif 'cierra todo' in rec:

talk("Hasta Pronto")

main_window.destroy()

elif 'cancelar' in rec:

talk("Hasta Pronto")

main_window.destroy()

elif 'información' in rec:

talk("Soy un asistente virtual que te ayudará tomando tu presión arterial. Y en base a esos valores y ciertas preguntas que te haré podré diagnosticar si padeces de hipertensión. Y si deseas también podré predecir que tan probable es que sufras un accidente cardíaco o cerebral. Estoy para servirte")

elif 'iniciar' in rec:

iniciando()

elif 'solo medir presión' in rec:

solo_presion()

else:

talk("No entendí la orden. Repítela por favor")

def informacion():

talk("Según la OMS, la tensión arterial es la fuerza que ejerce la sangre contra las paredes de las arterias, que son grandes vasos por los que circula la sangre en el organismo. La hipertensión es el término que se utiliza para describir la presión arterial alta. Es una enfermedad muy prevalente, sobre todo en países de ingresos bajos y medianos, que aumenta considerablemente el riesgo de padecer enfermedades cardíacas, cerebrales y renales. Por eso, recientemente, la OMS ha publicado unas nuevas directrices sobre el tratamiento farmacológico de la

hipertensión en el adulto. Las directrices formulan recomendaciones basadas en la evidencia sobre el comienzo del tratamiento de la hipertensión y los plazos de revisión recomendados. Sin embargo, según la OMS, el 46% de los adultos hipertensos desconocen que padecen esta afección y, de hecho, se diagnostica y se trata a menos de la mitad de los adultos que la presentan.")

#Botones de la Ventana Principal

```
button_medir = Button(main_window, text="Medir Presión", fg="white",  
bg="#F27121", font=("Arial", 14, "bold"), command=abrirventana2)
```

```
button_medir.pack(pady=35)
```

```
button_info = Button(main_window, text="Hipertensión", fg="white",  
bg="#F27121", font=("Arial", 14, "bold"), command=informacion)
```

```
button_info.pack(pady=35)
```

```
button_escuchar = Button(main_window, text="Hablar", fg="white",  
bg="#F27121", font=("Arial", 17, "bold"), command=run_doctor)
```

```
button_escuchar.pack(pady=35)
```

```
button_comandos = Button(main_window, text="Salir", fg="white",  
bg="#F27121", font=("Arial", 14, "bold"), command=main_window.destroy)
```

```
button_comandos.pack(pady=35)
```

```
main_window.mainloop()
```

ANEXO 5: Ensamblaje de la estructura para los módulos.

Diseño con medidas para la impresión 3D de la estructura del dispositivo para la toma de presión arterial.

