



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA
MENCIÓN EN SISTEMAS COMPUTACIONALES**

**TEMA DEL TRABAJO DE TITULACIÓN
DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO
MULTIMODAL PARA MEDICIÓN DE VARIABLES
FISIOLÓGICAS EN HUMANOS, CON COMUNICACIÓN
INALÁMBRICA A INTERNET Y APLICACIONES A LA
TELEMEDICINA.**

Trabajo de titulación previo a la obtención del
Título de Ingeniería Electrónica, mención: Sistemas Computacionales

AUTOR: JUAN LUIS ÁLAVA MIELES

TUTOR: PHD. LUIS NEIRA

Guayaquil-Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Juan Luis Alava Mieles con documento de identificación N° 0913695029 manifiesto que:

Soy el autor y responsable del presente trabajo de Titulación: “Diseño e Implementación de un Prototipo Multimodal para Medición de Variables Fisiológicas en Humanos, con comunicación inalámbrica a internet y aplicaciones a la telemedicina”; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 15 de febrero del año

2022 Atentamente,



Firma

Juan Luis Álava Mieles

Cédula: 0913695029

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Juan Luis Alava Mieles, con documento de identificación No. 0913695029 expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Trabajo de Titulación: “Diseño e Implementación de un Prototipo Multimodal para Medición de Variables Fisiológicas en Humanos, con comunicación inalámbrica a internet y aplicaciones a la telemedicina “, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónico, mención Sistemas Computacionales, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 15 de febrero del año 2022

Atentamente,



Firma

Juan Luis Álava Mieles

0913695029

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Ing. Luis Antonio Neira Clemente, PhD. con documento de identificación N° 0909136582, docente de la Universidad Politécnica Salesiana Sede Guayaquil, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: “DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO MULTIMODAL PARA MEDICIÓN DE VARIABLES FISIOLÓGICAS EN HUMANOS, CON COMUNICACIÓN INALÁMBRICA A INTERNET Y APLICACIONES A LA TELEMEDICINA”, realizado por Juan Luis Alava Mieles con documento de identificación N° 0913695029, obteniendo como resultado final el trabajo de titulación bajo la opción Trabajo de titulación que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 15 de febrero del año 2022

Atentamente,



Ing. Luis Antonio Neira Clemente, PhD

0909136582

DEDICATORIA

Dedico este trabajo de titulación a mi madre, Águeda Aurora Mieles Macías, gracias a ella por estar a mi lado, por su apoyo incondicional y sus incesantes oraciones. Sus oraciones y bendiciones de cada día, me guardan de todo mal y me llevan por el camino de bien. Por eso le dedico este trabajo por su paciencia y su fe hacia mí. Y también agradezco a mi hermano Domingo Joaquín Alava Mieles por haber sido un padre para mí, y haberme guiado desde mi niñez por el bien e instruirme en lo correcto de mi vida.

AGRADECIMIENTO

Mi agradecimiento va primero hacia Dios, por haberme dado la vida y también de tener una familia que me ha apoyado, en especial a mí hermana Ida Aurora Alava Mieles, por motivar a terminar esta carrera que con su tesón y empuje que me animó a seguir adelante en la culminación de la misma.

RESUMEN

AÑO	ALUMNO	DIRECTOR DE PROYECTO TÉCNICO	TEMA DE PROYECTO TÉCNICO
2021	Juan Luis Álava Mieles	PhD. Luis Neira :	“Diseño e implementación de un prototipo multimodal para medición de variables fisiológicas en humanos, con comunicación inalámbrica a internet y aplicaciones a la telemedicina”

El presente proyecto técnico tiene como objetivo diseñar, desarrollar e implementar un prototipo multimodal de mediciones de variables biomédicas conectado mediante internet a un servidor con base de datos que junto al desarrollo de una página web y aplicación móvil permita realizar la administración y registro de la información enviada.

El proyecto apunta a ser una solución a los procesos médicos en la medición de variables fisiológicas específicamente para el cuerpo médico que labora en el centro médico Semedic Sur, localizado en la ciudad de Guayaquil. De esta manera, se contribuiría a la optimización de herramientas digitales en la gestión médica en relación con la telemedicina, misma que se ha implementado en la actualidad primigeniamente por el abrupto aparecer de la pandemia del COVID-19 y su relación en diversos campos de la humanidad.

Sin embargo, la telemedicina, marca una evolución en el manejo del trato médico con respecto a los pacientes, abriendo un abanico de factores positivos de la aplicación de la misma. Pacientes que por dolencias físicas no pudiesen movilizarse hasta los centros médicos; pacientes que sufran alguna molestia

causada por el traslado hasta el punto de revisión médica; o simplemente algún inconveniente por no tener el medio de transporte adecuado para la movilización del paciente ,teniendo en cuenta la alta posibilidad de largas distancias desde su hogar hasta el centro de salud u hospital agendado, para la revisión o chequeo del progreso en la aplicación de algún tratamiento de situaciones de salud no crítica en los pacientes; son algunas de las muchas ventajas que ofrece la aplicación apropiada en la telemedicina. Además, permite a los pacientes tener una referencia acerca de su salud mediante la red de telemedicina implementada.

Debido a todo lo antes mencionado surgió la idea de la elaboración del presente proyecto, mismo que sugiere ser una posible solución ante esta problemática detectada y ampliamente explicada a lo largo de esta investigación.

Palabras claves: variables fisiológicas, internet, servidor, prototipo, telemedicina.

ABSTRACT

YEAR	STUDENT	DIRECTOR OF THECINAL PROJECT	TECHNICAL PROJECT THEME
2021	Juan Luis Álava Mieleles	PhD. Luis Neira	Design and implementation of a multimodal prototype for measuring physiological variables in humans, with wireless internet communication and telemedicine applications

The objective of this technical project is to design, develop and implement a multimodal prototype for measurements of biomedical variables connected via the internet to a server with a database that, together with the development of a web page and mobile application, allows the administration and registration of information sent.

The project aims to be a solution to medical processes in the measurement of physiological variables specifically for the medical staff that works in the Semic Sur medical center, located in the city of Guayaquil. In this way, it would contribute to the optimization of digital tools in medical management in relation to telemedicine, which has been implemented today primarily due to the abrupt appearance of the COVID-19 pandemic and its relationship in various fields of the humanity.

However, telemedicine marks an evolution in the management of medical treatment with respect to patients, opening a range of positive factors in its application.

Patients who due to physical ailments could not travel to medical centers; patients who suffer any discomfort caused by the transfer to the medical examination point; or simply some inconvenience due to not having the adequate means of transport for the mobilization of the patient, taking into account the high possibility of long distances from home to the scheduled health center or hospital, for the review or check of progress in the application of some treatment of non-critical health situations in patients; are some of the many advantages that the appropriate application in telemedicine offers. In addition, it allows patients to have a reference about their health through the telemedicine network implemented.

Due to all the aforementioned, the idea of the development of this project arose, which suggests that it is a possible solution to this problem detected and widely explained throughout this research.

Keywords: physiological variables, internet, server, prototype, telemedicine.

INDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUDITORÍA.....	¡Error! Marcador no definido.
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR...	¡Error! Marcador no definido.
CERTIFICADO DE DIRECCIÓN DE TRABAJO DE TITULACIÓN.....	¡Error! Marcador no definido.
DEDICATORIA.....	iv
AGRADECIMIENTO.....	v
RESUMEN.....	vi
ABSTRACT.....	viii
INDICE GENERAL.....	x
ÍNDICE DE FIGURAS.....	xiii
ÍNDICE DE TABLAS.....	xvi
INTRODUCCIÓN.....	1
1. EL PROBLEMA.....	2
1.1. Antecedentes.....	2
1.2. Importancia y Alcances.....	3
1.3. Delimitación.....	4
1.3.1. Temporal.....	4
1.3.2. Espacial.....	4
1.3.3. Académica.....	4
1.4. Innovación.....	4
1.5. Objetivos.....	5
2. FUNDAMENTOS TEÓRICOS.....	6
2.1. Telemedicina.....	6
2.2. Variables Fisiológicas.....	6
2.3. Sensores Fisiológicos.....	7
2.3.1. Termómetro infrarrojo corporal.....	7
2.3.2. Tensiómetro.....	8
2.3.3. Oxímetro.....	10
2.3.4. Sensor de ritmo cardíaco.....	13
2.3.5. E-Health Sensor Shield.....	14

2.4.	Arduino	15
2.5.	Módulo Wifi ESP8266	15
2.6.	Internet de las cosas	16
2.7.	Base de datos documentales	16
2.8.	Node.js	17
2.9.	MongoDB	17
2.10.	Amazon Web Services	18
2.10.1.	Amazon Elastic Compute Cloud	18
2.10.2.	Amazon Simple Storage Service	19
2.11.	Flutter	19
2.11.1.	Dart	19
2.12.	API REST	20
2.13.	Arquitectura cliente servidor	20
2.14.	Diseño Web Responsive	21
3.	MARCO METODOLÓGICO	22
3.1.	Diseño del sistema	22
3.2.	Implementación del módulo de telemedicina	23
3.2.1.	Placa electrónica	23
3.2.2.	Programación del módulo	33
3.2.3.	Diseño de carcasa	39
3.3.	Arquitectura del servidor	41
3.3.1.	Servidor virtual	41
3.3.2.	Creación de los API REST	42
3.4.	Página web y aplicativo móvil	47
3.4.1.	Diseño de la página web	48
3.4.2.	Diseño del aplicativo móvil	48
3.4.3.	Desarrollo de la lógica de programación	49
3.4.4.	Hosting	53
4.	RESULTADOS	54
4.1.	Pruebas Iniciales	54
4.1.1.	Sensor MLX30100	54
4.1.2.	Módulo AD8232	55
4.2.	Base de datos	58

4.3. Página web y aplicación móvil	60
CONCLUSIONES	64
RECOMENDACIONES.....	65
REFERENCIAS BIBLIOGRÁFICAS	66
ANEXOS	69
Anexo 1 Creación y configuración del servidor	69
Anexo 1.0 Creación de una instancia en EC2	69
Anexo 2 REST API desarrollados.....	74
Anexo 2.0 Estado del API	74
Anexo 2.1 Crear registro	74
Anexo 2.2 Obtener todos los usuarios	76
Anexo 2.3 Obtener el registro de un usuario	76
Anexo 2.4 Crear Variable	77
Anexo 2.5 Obtener variables	78

ÍNDICE DE FIGURAS

Figura 1: Sensor de temperatura corporal MLX90614	8
Figura 2: Tensiómetro digital.....	9
Figura 3: Max30100 Sensor De Frecuencia Cardiaca Oximetro	10
Figura 4: Diagrama de funcionamiento del sensor MAX30100	11
Figura 5: Módulo de ritmo cardíaco Ecg Ad8232 Arduino	13
Figura 6: Electrodo del módulo de ritmo cardíaco	13
Figura 7: E-health sensor shield	14
Figura 8: Arduino Nano	15
Figura 9: Wifi ESP8266.....	16
Figura 10: Node js.....	17
Figura 11: Base de datos Mongo DB	17
Figura 12: Amazon Web Services.....	18
Figura 13: Flutter SDK 2.2.3	19
Figura 14: Estructura de la arquitectura cliente servidor	21
Figura 15: Diseño adaptativo o responsive	21
Figura 16: Diseño general del prototipo desarrollado	22
Figura 17: Modificación del monitor de tensión arterial (Cable amarillo: SDA, cable rojo: SCL y cable marrón: Negativo).....	25
Figura 18: Monitor de tensión arterial sellado luego de la modificación.....	25
Figura 19: Colocación de los electrodos (RA=R, LA=L y LL=COM) (Kanani & Padole, 2018).....	26
Figura 20: Esquema general de funcionamiento del módulo	27
Figura 21: Implementación del módulo sobre un protoboard.....	29
Figura 22: Diseño esquemático de los dos circuitos Arduino Nano	30
Figura 23: Diseño esquemático del circuito de la ESP8266.....	31
Figura 24: Circuito de alimentación del módulo	32
Figura 25: Diseño PCB del módulo.....	32
Figura 26: Placa electrónica del módulo de telemedicina	33
Figura 27: Diagrama de flujo (Configuración inicial) de la ESP8266.....	35
Figura 28: Diagrama de flujo (Void Loop) de la ESP8266.....	36
Figura 29: Diagrama de flujo del programa en Arduino Nano A1.....	37

Figura 30: Diagrama de flujo del programa en Arduino Nano A1.....	38
Figura 31: Partes individuales de la carcasa del módulo	39
Figura 32: Ensamblaje digital de toda la carcasa del módulo.	39
Figura 33: Impresión 3D de la carcasa (a. Carcasa ensamblada b. Soporte para el sensor de temperatura).....	40
Figura 34: Módulo de telemedicina completamente ensamblado	40
Figura 35: Funcionamiento del servidor a implementar	41
Figura 36: Diagrama de flujo del REST “estado”	42
Figura 37: Diagrama de flujo para registrar un paciente en la base de datos	43
Figura 38: Diagrama de flujo para la obtención de todos los pacientes registrados.	44
Figura 39: Diagrama de flujo que permite obtener los datos de un paciente específico	45
Figura 40: Diagrama de flujo para escribir la lectura de los sensores.....	46
Figura 41: Diagrama de flujo para obtener la lectura de los sensores.	47
Figura 42: Boceto de la página web.....	48
Figura 43: Boceto del aplicativo móvil.....	49
Figura 44: Diagrama de flujo para determinar el dispositivo y la distribución de pantalla a mostrarse	50
Figura 45: Adquisición de variables del servidor para representarlos en la interfaz visual	51
Figura 46: Diagrama de flujo de la pantalla “Pacientes”	52
Figura 47: Valores de medición de la saturación de oxígenos en la sangre y bpm con diferentes valores de corriente en los LEDs a) 50mA y b) 7mA	55
Figura 48: Señal resultante de la medición de ritmo cardiaco sin un intervalo entre cada muestra	56
Figura 49: Señal resultante de la medición de ritmo cardiaco con un tiempo de 50ms entre cada muestra.	56
Figura 50: Señales resultante de la medición de ritmo cardíaco con un tiempo de muestreo de: a) 5ms b) 1ms y c) 10ms.....	57
Figura 51: Señal resultante de la medición de ritmo cardiaco con un tiempo de 8ms entre cada muestra.	57
Figura 52: Señal resultante de la medición de ritmo cardiaco en un paciente de 6 años.	58
Figura 53: Colecciones generados en la base de datos Variables Fisiológicas en MongoDB	58
Figura 54: Colección de pacientes históricos guardados por la aplicación web	59

Figura 55: Colección que registra los datos obtenidos por el sistema de adquisición de variables.....	59
Figura 56: Página web implementada sección Revisión Online.....	60
Figura 57: Aplicación móvil equivalente en web	60
Figura 58: Diferentes elementos utilizados en la aplicación para mostrar	61
Figura 59: Panel de históricos ordenados por antigüedad en la versión web	62
Figura 60: Panel de históricos ordenados por antigüedad equivalente en su versión móvil	62
Figura 61: Paciente registrado en la base de datos.....	63
Figura 62: Cuenta de AWS consola de administración	69
Figura 64: Selección del sistema operativo Ubuntu	70
Figura 65: Selección de características del servidor virtual.	70
Figura 66: Configuraciones generales del servidor seleccionado	71
Figura 67: Selección de almacenamiento del computador virtual.	71
Figura 68: Configuración de seguridades del servidor.....	72
Figura 69: Resumen del servidor seleccionado	72
Figura 70: Creación de claves de acceso remoto	73

ÍNDICE DE TABLAS

Tabla 1	Valores de presión arterial	9
Tabla 2	Control de Corriente del LED	12
Tabla 3	Características del servidor virtual	41
Tabla 4	Dirección web para el ingreso a la aplicación web.....	53
Tabla 5	Datos generales del REST “estado”	74
Tabla 6	Códigos de respuesta del REST “estado”	74
Tabla 7	Datos generales del REST “crearRegistro”	74
Tabla 8	Parámetros enviados del REST “crearRegistro”	75
Tabla 9	Ejemplo de envío del REST “crearRegistro”	75
Tabla 10	Códigos de respuesta del REST “crearRegistro”	75
Tabla 11	Datos generales del REST “obtenerUsuarios”	76
Tabla 12	Códigos de respuesta del REST “obtenerUsuarios”	76
Tabla 13	Datos generales del REST “obtenerRegistro”	76
Tabla 14	Parámetros enviados del REST “obtenerRegistro”	76
Tabla 15	Ejemplo de envío del REST “obtenerRegistro”	77
Tabla 16	Códigos de respuesta del REST “obtenerRegistro”	77
Tabla 17	Datos generales del REST “crearVariables”	77
Tabla 18	Parámetros enviados del REST “crearVariables”	77
Tabla 19	Ejemplo de envío del REST “crearVariable”	78
Tabla 20	Códigos de respuesta del REST “crearVariables”	78
Tabla 21	Datos generales del REST “obtenerVariables”	78
Tabla 22	Códigos de respuesta del REST “obtenerVariables”	78

INTRODUCCIÓN

El desarrollo de la electrónica y sistemas computacionales han permitido automatizar y conectar una infinidad de procesos e información, y los tratamientos médicos no son la excepción.

En la actualidad, es cotidiano hablar de telesalud o telemedicina, términos usados para hacer referencia a consultas, tratamientos y procedimientos médicos realizados vía online. Es así como, la telemedicina se encuentra en su mejor momento debido a la emergencia mundial, ya que un médico puede cuidar a distancia de la salud de su paciente, a través de medios manejados remotamente.

El desarrollo de un sistema electrónico de gestión y adquisición de información basado en sensores, microprocesadores, servidores, bases de datos, páginas web y aplicaciones móviles pretenden dar una solución a la automatización de los procedimientos de salud preventiva para pacientes de un centro de salud privado, de tal forma que se promueva el desarrollo de la sociedad con estas nuevas tecnologías.

El proyecto abre la posibilidad al paciente de contar con información confiable a cerca de su salud y tomar recomendaciones de prevención de manera automática o bajo criterio médico, mediante contacto con el mismo de manera telemática; por lo que posibilita una mejor relación entre el profesional médico y su paciente.

1. EL PROBLEMA

1.1. Antecedentes

En la actualidad el uso de servicios de salud en línea, son una realidad que ha ayudado en mucho a descongestionar los distintos hospitales y centros de salud en el mundo entero, gracias al uso de la telemedicina y dispositivos inteligentes que permiten la transmisión, registro y análisis de datos del paciente, logrando así la automatización de procedimientos y tratamientos médicos en personas con enfermedades consideradas crónicas de carácter no transmisible (ECNT) que en su mayoría requieren un seguimiento regular de las mediciones de algunas variables biomédicas como presión arterial, nivel de glucosa, electrocardiogramas (ECG), electromiogramas (EMG) nivel de oxígeno en la sangre, temperatura corporal y otras, que con las respectivas indicaciones pueden ser tomadas por el propio paciente, facilitando y optimizando su propio tratamiento. En el Ecuador estas formas de tratamientos aún se ven algo limitadas debido al difícil acceso y poca difusión de estas tecnologías en la población, lo que ha repercutido en el desperdicio de tiempo y recursos en ciertos tratamientos médicos que se pueden llevar a cabo sin la necesidad de asistencia médica personalizada al paciente.

Ante esta situación y con el propósito de facilitar la interacción entre paciente y médico en el control de variables biomédicas de importancia para el diagnóstico médico, se ha planteado desarrollar un prototipo de dispositivo electrónico que permita realizar las mediciones de estas variables fisiológicas, y mediante internet enviarlas a una base de datos donde una página web y/o aplicación móvil pueda administrar esta información de forma eficiente.

1.2. Importancia y Alcances

El presente trabajo ayuda al avance de prototipos biomédicos conectados al internet de las cosas.

Este proyecto permite al personal médico que labora en el Centro Médico 'Semedic', localizado en el Sur de la ciudad de Guayaquil, tener una plataforma web donde se podrá visualizar las variables fisiológicas de sus pacientes obtenidas con el prototipo biomédico implementado, proporcionando un programa de automatización de los procedimientos de salud preventiva para pacientes del centro de salud particular.

Adicionalmente, la pandemia del COVID-19 ha sido gestora del cambio en el paradigma de las relaciones médico-pacientes, modificando el tradicional trato de los médicos exclusivamente en sus consultorios revisando a sus pacientes, para en pro evitar la propagación de la pandemia actual; la práctica médica ha evolucionado a la telemetría, misma que en situaciones no críticas de los enfermos, permite hacer un seguimiento en el tratamiento de alguna enfermedad, vigilando la evolución de la enfermedad para una pronta recuperación de manera remota a través de medios electrónicos y plataformas digitales. La telemedicina se ve condicionada por diversos factores, tales como la protección de datos, la privacidad y la seguridad de la información de los pacientes. Por los factores antes mencionadas, las entidades del sector médico, en este caso específico el centro médico Semedic, apuesta por establecer sistemas de gestión de calidad y de seguridad del paciente para garantizar la calidad de los diagnósticos y tratamientos. Es por ello la relevancia de la implementación de la telemedicina para ofrecer una mejor tanto en la gestión medica ofrecida a los pacientes de manera telemática como una optimización de los recursos informáticos para una rápida, oportuna y veraz en la entrega de los resultados y diagnósticos de los pacientes.

1.3. Delimitación

1.3.1. Temporal

El desarrollo del proyecto se llevó a cabo durante 6 meses a partir de su aprobación. Al final de este período, se obtuvo como producto final el dispositivo en correcto funcionamiento tanto en la parte de hardware como software.

1.3.2. Espacial

El proyecto se realizó en Semic Sur de la ciudad de Guayaquil, ubicado en Av. Circunvalación Sur 905, Guayaquil; en donde se implementó y probó el prototipo desarrollado.

1.3.3. Académica

El proyecto cuenta con todos los requerimientos exigidos por la Universidad Politécnica Salesiana en torno al desarrollo de proyectos técnicos y de investigación, en el cual se desarrolló un prototipo de medición de variables fisiológicas humanas con conexión a internet de las cosas, utilizando conocimientos de instrumentación biomédica, microcontroladores, sensores, electrónica analógica y digital, y programación web.

1.4. Innovación

El avance del internet de las cosas obliga a la sociedad a desarrollar nuevas tecnologías, el campo de la telemedicina no está exento, el implementar un prototipo con sensores biomédicos no invasivos con conexión a internet, en donde el servidor y una base de datos permite el intercambio de información entre médicos y pacientes con el fin de realizar una sanidad más sostenible y contribuir con la eficiencia de los métodos actuales mejorando la salud de las personas.

1.5. Objetivos

1.5.1. Objetivo general

- Diseñar, desarrollar e implementar un prototipo multimodo, mediante el uso de la plataforma e-health sensors y Arduino, para medición de variables fisiológicas en humanos, con comunicación inalámbrica wifi a internet, junto con una página web y aplicación móvil que permitan la gestión de datos y la interacción entre paciente y médico.

1.5.2. Objetivos específicos

- Determinar y diseñar el sistema de adquisición de datos de las variables fisiológicas de humanos, que será utilizado en este prototipo, con las respectivas tomas de pruebas y validación de forma comparativa con otros dispositivos estandarizados.
- Diseñar e implementar tanto el hardware como el software para el sistema microprocesador para el tratamiento de datos, mediante el uso de softwares de simulación para sistemas electrónicos.
- Diseñar y realizar la programación e implementación en hardware del sistema microprocesador para su conexión con el internet de las cosas, mediante el módulo de comunicación inalámbrica ESP8266.
- Diseñar e implementar una página web con un servidor web, hosting, y dominio gratuito, donde se permita la gestión de datos recibidos desde este prototipo y la interacción médico-paciente.
- Diseñar una aplicación móvil en la plataforma Android, que permita realizar las mismas funciones que nuestra página web, con comunicación wifi por medio del módulo ESP8266.

2. FUNDAMENTOS TEÓRICOS

2.1. Telemedicina

La telemedicina hace referencia a la atención médica a distancia, es decir, no es necesario que el doctor y el paciente se encuentren en contacto físico, para realizar un diagnóstico oportuno. Cabe mencionar, que este sistema puede ser aplicado para cualquier especialidad médica. Para esto, se ha hecho uso de la tecnología que ha facilitado y ha resuelto varios inconvenientes que se presentan a menudo como puede ser: la distancia, falta de personal, transporte, entre otros (Alvarado, 2005).

2.2. Variables Fisiológicas

De acuerdo con lo mencionado por Gutiérrez (2016), el control de los signos vitales se lo realiza a través de la medición de las variables fisiológicas mediante el uso de sensores que perciben la señal analógica y la transforma en digital con el uso de un microcontrolador para su visualización.

Además, el autor señala que la monitorización puede ser de dos tipos: invasiva y no invasiva. La monitorización invasiva consiste en un sensor que se introduce en el cuerpo o atraviesa la piel, y de esta forma percibe las señales fisiológicas buscadas. Por otro lado, la monitorización no invasiva no atraviesa la piel ni penetran de forma física el cuerpo humano. El presente trabajo involucra sensores no invasivos que buscan la medición de la frecuencia cardíaca, frecuencia respiratoria, presión arterial, saturación de oxígeno y temperatura corporal (Gutiérrez, 2016).

2.3. Sensores Fisiológicos

El sensor es un dispositivo que mide el cambio de una propiedad física, la almacena y responde a la misma. Un transductor es un aparato que transforma en una señal medible el cambio físico o químico generado (Eggins, 2002)

Las variables físicas como longitud, peso, temperatura, presión y electricidad pueden ser medidas con ayuda de sensores fisiológicos, dichas variables son de gran importancia para la evaluación del estado funcional de los seres humanos (Eggins, 2002).

2.3.1. Termómetro infrarrojo corporal

El termómetro es un instrumento que se utiliza para la medición de la temperatura. Los cambios de temperatura corporal es una manifestación de algunas enfermedades por lo que su medición es clave para el diagnóstico y monitorización del paciente, esta medida se lo puede realizar con un termómetro corporal, y así determinar la temperatura del individuo. En la actualidad, existen aparatos como termómetros infrarrojos que permiten obtener la medición de temperatura de forma práctica y a su vez confiable (Melexis, 2006).

El termómetro MLX90614, es un instrumento de tipo infrarrojo que permite obtener mediciones de temperatura sin ponerse en contacto directo con la superficie, ya que es sensible a la radiación infrarroja que generan los cuerpos y es mediante este concepto en conjunto con etapas de amplificación y digitalización que puede proporcionar un valor de temperatura, el cual puede ser desde -70°C hasta 380°C , con una precisión de 0.5°C (Melexis, 2006).

Su funcionamiento se apoya en la Ley de Stefan-Boltzmann, la cual menciona que un cuerpo emite radiación infrarroja, la misma que es proporcional a la temperatura de dicho cuerpo. Sin embargo, tanto la distancia como el campo

de visión del objeto a medir afecta de manera directa a la precisión del instrumento (Del Valle, 2021).

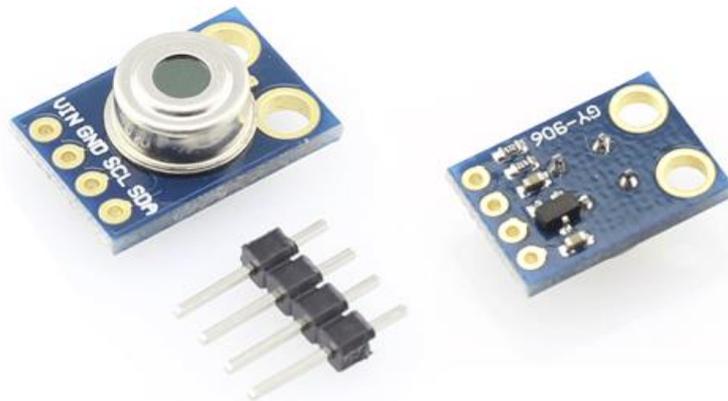


Figura 1: Sensor de temperatura corporal MLX90614 (Del Valle, 2021)

2.3.2. Tensiómetro

Es un dispositivo para medir la presión sanguínea del paciente, esta presión se produce por los latidos del corazón, cada latido empuja la sangre hacia las arterias, lo cual provoca que la sangre fluya a través de todo el cuerpo, durante este recorrido se ejerce fricción en las arterias lo que genera presión dentro de las mismas. A este dispositivo también se lo conoce como monitor de presión arterial, en el que sus mediciones están relacionadas con la fuerza que ejerce el flujo sanguíneo en las paredes de las arterias (Hiraoka, 2020).

Al realizar la medición de presión se obtienen dos valores: la presión sistólica y la presión diastólica. El primer valor de presión correspondiente a la presión sistólica se mide en el momento de máxima presión (durante el latido del corazón). El segundo tipo de presión llamada diastólica se mide en el momento de mínima presión, la cual ocurre durante el descanso entre los latidos. La forma correcta de presentar la medición obtenida es registrando la presión sistólica y luego la presión diastólica, por ejemplo 120/90. Esta variable fisiológica se mide en milímetros de mercurio (mmHg) y un valor elevado puede desencadenar enfermedades más graves como ataques al corazón, derrames cerebrales entre otras (Banerjee, 2006).

Mediante esta medición es posible establecer un diagnóstico contra la presión arterial alta, a continuación, se presenta un cuadro sobre los valores de presión en conjunto con su diagnóstico más común (Banerjee, 2006).

Categoría de presión arterial	Presión arterial sistólica		Presión arterial diastólica
Normal	Menos de 120	y	Menos de 80
Presión elevada	Menos de 130	y	Menos de 80
Presión arterial alta (Hipertensión nivel 1)	Menos de 140	o	Entre 80 y 89
Presión arterial alta (Hipertensión nivel 2)	Mayor o igual a 140	o	Mayor o igual a 90
Presión arterial peligrosamente alta (Crisis hipertensiva) – Consulte inmediatamente a su médico	Mayor o igual a 180	y	Mayor o igual 120

Tabla 1 Valores de presión arterial (Banerjee, 2006)



Figura 2: Tensiómetro digital (Hiraoka, 2020)

2.3.3. Oxímetro

El oxímetro es un sensor que realiza dos mediciones de manera simultánea, la primera medición son las pulsaciones por minuto y la segunda es la saturación de oxígeno en la sangre. Cabe mencionar que el pulso está ligado con la expansión y contracción de las arterias como consecuencia del bombeo de la sangre por el corazón, es decir, el pulso cardíaco es el número de latidos del corazón (Mejía & Mejía, 2012).

Este dispositivo busca medir la cantidad de oxígeno que es transportado por la hemoglobina en el torrente sanguíneo. Cada medida proporciona un valor porcentual; los valores normales de SpO₂ están en el rango de 95 a 100% (Maxim Integrated, 2014)

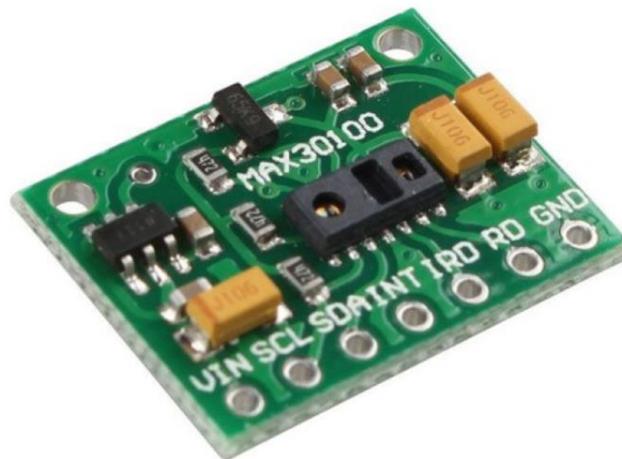


Figura 3: Max30100 Sensor De Frecuencia Cardiaca Oximetro (Mejía & Mejía, 2012)

Para realizar la medición este dispositivo electrónico se compone de 2 etapas importantes en el circuito. La primera etapa se encarga de emitir luz hacia el dedo del paciente a través de un diodo emisor de luz (LED) de color rojo que tiene una longitud de onda de 660nm y un diodo infrarrojo de 920nm.

La luz emitida debe ser dirigida al dedo del paciente, en donde la sangre oxigenada que circula absorberá la mayor cantidad de luz infrarroja, mientras que la sangre con una menor cantidad de oxígeno absorberá mayor luz roja emitida por el LED (Del Ángel & Escalera, 2017)

En la segunda etapa se mide la cantidad de luz reflejada mediante un fotodetector, además de complementarse con componentes adicionales para compensar factores ambientales y de la propia medición que pueden afectar a la interpretación de datos. Finalmente, mediante un conversor análogo-digital la información es tratada para que pueda ser utilizada a conveniencia (Del Ángel & Escalera, 2017). El funcionamiento de este dispositivo se representa de mejor manera en la Figura 4.

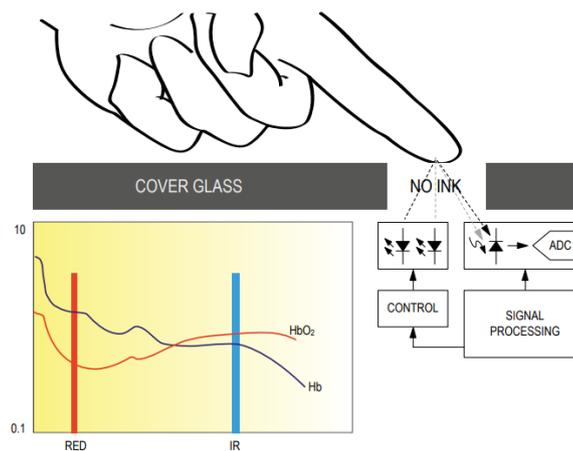


Figura 4: Diagrama de funcionamiento del sensor MAX30100 (Maxim Integrated, 2014)

Se debe mencionar que este dispositivo tiene la opción de modificar la cantidad de corriente con que se alimenta al LED; factor que llega a afectar a la medición dependiendo de la distancia a la que se coloque el dedo del paciente. Las posibles variaciones de este parámetro se presentan en la Tabla 1. Para comprobación del funcionamiento del sensor se dejó en el valor por defecto de corriente que es de 50mA, sin embargo, luego de realizar pruebas con el módulo terminado, este valor puede ser calibrado para obtener mejores resultados (Maxim Integrated, 2014)

Red_PA [3:0] OP IR_PA[3:0]	Typical LED Current [mA]
0000	0.0
0001	4.4
0010	7.6
0011	11.0
0100	14.2
0101	17.4
0110	20.8
0111	24.0
1000	27.1
1001	30.6
1010	33.8
1011	37.0
1100	40.2
1101	43.6
1110	46.8
1111	50.0

Tabla 2 Control de Corriente del LED (Maxim Integrated, 2014)

Este dispositivo es capaz de proporcionar un valor de pulsaciones por minuto para conocer el ritmo cardíaco del paciente, sin embargo, este valor puede obtenerse de otros sensores que brindan mayor precisión (Maxim Integrated, 2014)

2.3.4. Sensor de ritmo cardíaco

El corazón de una persona late cierta cantidad de veces por minuto, que dependiendo de la actividad física que esté realizando y del estado de salud de la persona este valor, llamado frecuencia cardiaca, puede rondar entre 60 a 90 latidos por minutos o bpm por su significado en inglés. Es necesario mencionar que el valor de frecuencia cardiaca no es el único relevante para determinar la salud del corazón, pues es el ritmo de sus latidos, que normalmente son presentados en un electrocardiograma, el que puede ayudar a detectar con mayor certeza algún signo de enfermedad cardiaca (Mayo Clinic, 2021)

El electrocardiógrafo está constituido por tres electrodos, los cuales son colocados en la zona torácica sobre la piel, estos electrodos reciben la señal electrocardiográfica originada en el miocardio del paciente, la señal receptada se representa en electrocardiogramas (EKG o ECG). El electrocardiograma es una prueba de diagnóstico no intrusiva para la evaluación del ritmo cardíaco por medio del registro de la actividad eléctrica cardiaca (Zavala, 2017).

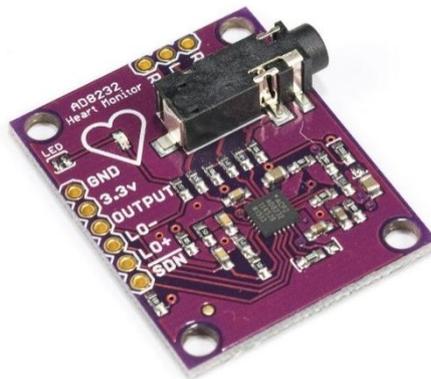


Figura 5: Módulo de ritmo cardíaco Ecg Ad8232 Arduino (Mayo Clinic, 2021)



Figura 6: Electrodos del módulo de ritmo cardíaco (Mayo Clinic, 2021)

2.3.5. E-Health Sensor Shield

Es una placa compatible con microcontroladores como Arduino, Raspberry Pi e Intel Galileo; usada para aplicaciones médicas y biométricas, especialmente en el monitoreo de pacientes en tiempo real con el fin de recopilar datos relevantes para su posterior análisis y llegar a un diagnóstico médico (SoloElectronicos, 2017). El equipo emplea varios sensores como:

- Oximetría de pulso (SpO₂)
- Flujo de aire (respiración)
- Temperatura corporal
- Electrocardiograma (ECG)
- Glucómetro
- Respuesta galvánica de la piel (GSR - sudoración).
- Presión arterial (esfigmomanómetro).
- Posición o movimiento del paciente (acelerómetro)

(SoloElectronicos, 2017)



Figura 7: E-health sensor shield (SoloElectronicos, 2017)

2.4. Arduino

Arduino es un sistema de desarrollo que consta de una monocapa electrónica de hardware libre que usa varias entradas y salidas de señales y datos conectadas a un microcontrolador reprogramable. El microcontrolador establece comunicación con diferentes sensores y actuadores de forma simple, lo que facilita usar la electrónica para la realización de diseños interactivos (ARDUINO.cl, 2021).

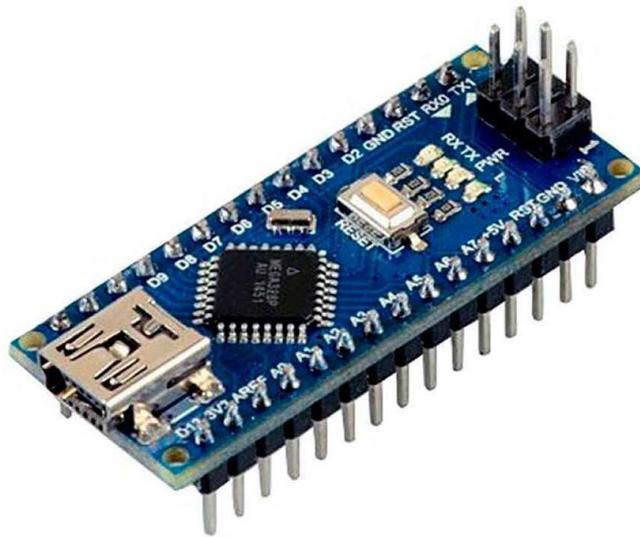


Figura 8: Arduino Nano (ARDUINO.cl, 2021)

2.5. Módulo Wifi ESP8266

El módulo ESP8266 permite la conectividad al microcontrolador asociado a la red WiFi utilizando el protocolo TCP/IP integrado. Este módulo puede procesar la aplicación o descarga de las funciones de Wifi completas. Los módulos individuales de ESP8266 son preprogramados con comandos AT de un grupo de firmware (SANDOROBOTICS, 2019)

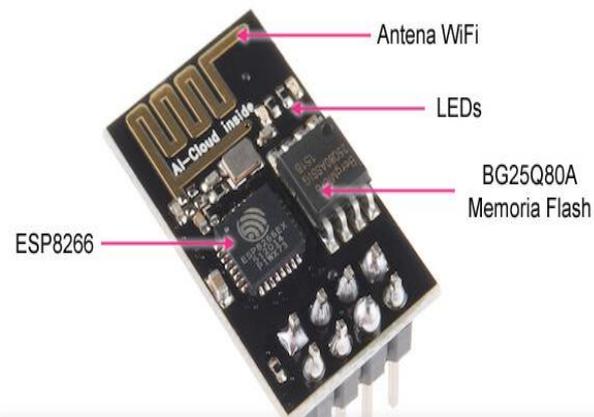


Figura 9: Wifi ESP8266 (SANDOROBOTICS, 2019)

2.6. Internet de las cosas

El internet de las cosas o internet de los objetos (IdC) es de gran importancia ya que recopila, estudia y reparte datos convirtiéndolos en información para el IdC. El internet es uno de los inventos más revolucionarios en la historia, ya que ha tenido un impacto importante en diferentes áreas como la educación, finanzas, y la forma de vida en general (Evans, 2011).

2.7. Base de datos documentales

Una base de datos es una cadena estructurada de información, ordenada en forma de registros y guardada de manera electrónica, la cual es descifrable por un ordenador. Cada unidad autónoma es un registro de información que puede estar formado por diferentes tipos de datos recopilados en esta base de datos. Una forma para entender esto de forma práctica se toma como ejemplo un directorio de integrantes de un grupo, cada integrante llena un registro, la cual es básicamente, una ficha con determinados datos como: nombre, profesión, teléfono o dirección, cada uno de estos datos constituye un campo (Rodríguez, 2001)

2.8. Node.js

Node.js utiliza una arquitectura de entradas y salidas para la programación basada en eventos asíncronos, el cual emplea JavaScript V8 de Google como motor de conversión de scripts (Tilkov & Vinoski, 2010).

Para la configuración de Node y V8 se utiliza el lenguaje C y C++, que ofrece dos ventajas, la primera es enfocarse en el rendimiento y la segunda es disminuir el consumo de memoria. Tanto el V8 como Node se complementan entre sí, mientras V8 accede con el navegador JavaScript, Node es el responsable de administrar los procesos del servidor en tiempo real mediante el tratamiento intensivo de datos (Tilkov & Vinoski, 2010)



Figura 10: Node js (Tilkov & Vinoski, 2010)

2.9. MongoDB

MongoDB parece en la era de la nube como una base de datos, en la cual su distribución está relacionada con documentos de uso general. Fue elaborada para desarrolladores que trabajan en el diseño de aplicaciones modernas (MongoDB, 2021)



Figura 11: Base de datos Mongo DB (MongoDB, 2021)

2.10. Amazon Web Services

Amazon Web Services es una serie de servicios computacionales dentro de la nube, los cuales conforman una plataforma adaptable y completa en la nube pública ya que ofrece alrededor de 200 servicios integrales de datos a nivel mundial. Estos servicios se ofertan en internet por medio de Amazon.com (Amazon Web Services, 2021a)



Figura 12: Amazon Web Services (Amazon Web Services, 2021a)

2.10.1. Amazon Elastic Compute Cloud

Amazon Elastic Compute Cloud o denominado Amazon EC2, se describe como un servicio web diseñado para brindar seguridad a la nube y modificar su tamaño. Su diseño simple permite a los desarrolladores reducir los pasos de la informática en la nube dentro de la escala web. La simplicidad de la interfaz que brinda Amazon EC2 ayuda a la configuración de la capacidad con una fricción baja, lo cual permite controlar de manera completa los recursos informáticos. Cabe mencionar, que la ejecución de estos recursos se realiza bajo la plataforma informática de Amazon (Amazon Web Services, 2021a).

2.10.2. Amazon Simple Storage Service

Amazon Simple Storage Service también conocido como Amazon S3 es un servicio que ofrece almacenamiento de objetos de amplia escalabilidad y base de datos disponible. Además, son líderes en el mercado ya que brindan alto nivel de seguridad y rendimiento (Amazon Web Services, 2021b).

2.11. Flutter

Flutter es un juego de herramientas para el interfaz de usuario de Google que tienen el objetivo de permitir la creación de aplicaciones compiladas de forma original para dispositivos móviles, plataformas web y computadoras de escritorio desde un código base único (Flutter, 2021).



Figura 13: Flutter SDK 2.2.3 (Flutter, 2021)

2.11.1. Dart

Dart es un tipo de lenguaje de programación orientado a objetos (OOP), el cual permite al cliente crear aplicaciones de manera rápida y ejecutarlas en cualquier plataforma, es por lo que se lo conoce como un lenguaje optimizado. Al hablar de lenguajes de programación orientado a objetos, se hace referencia a lenguajes como Java o C#, no obstante, Dart presenta varias similitudes con dichos lenguajes antes mencionados (Miola, 2020).

2.12. API REST

La Interfaz de programación de aplicaciones o su abreviatura API reúne varias requisiciones para la transmisión de información de datos de una aplicación a otra, como lo puede ser en aplicaciones de celulares y páginas web (Rosa, 2018).

De igual forma, REST es una interfaz que permite la conectividad de sistemas fundamentado en uno de los protocolos más tradicionales HTTP. Además, la generación y obtención de datos en base a las operaciones son presentados en formato XML y JSON (Rosa, 2018).

De tal modo, API REST es el conjunto de buenas prácticas implementadas conjunto el protocolo HTTP en una aplicación móvil o web (De Souza, 2020)

Las solicitudes destacadas son:

- POST: originar datos en el servidor
- GET: leer datos en el host
- DELETE: suprimir la información
- PUT: registrar actualización

(De Souza, 2020)

2.13. Arquitectura cliente servidor

Cliente/Servidor es un ejemplo de aplicación distribuida, cada asignación es compartida entre proveedores de servicio (servidores) y demandantes (clientes). La aplicación Cliente manda la orden a la aplicación Servidor, y este último al estar en ejecución, es el encargado de atender dichas peticiones. Cada máquina del sistema distribuido puede cumplir con la posición de servidor para desempeñar requisitos específicos o cambiar su posición a cliente para realizar otras (Marini, 2012).

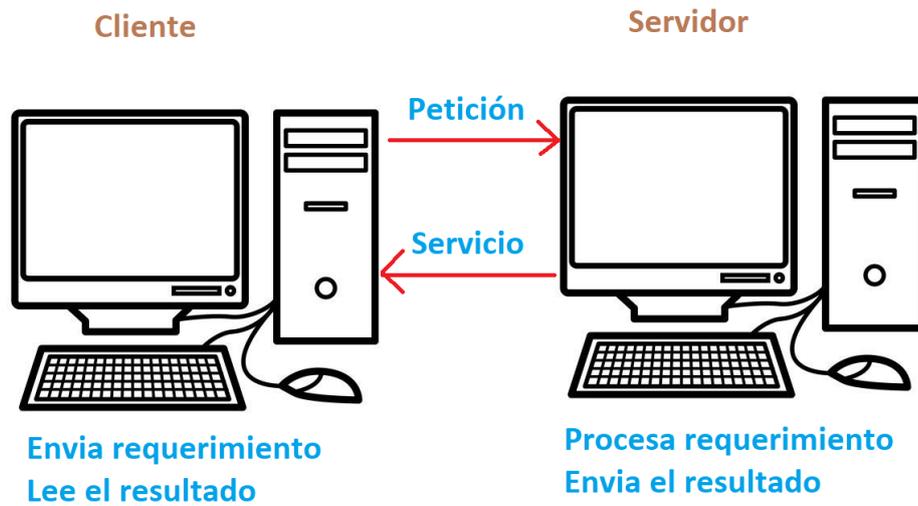


Figura 14: Estructura de la arquitectura cliente servidor (Marini, 2012)

2.14. Diseño Web Responsive

Diseño Web Responsive es una forma de diseño y desarrollo de contenido de sitios web mediante la implementación de estructuras e imágenes fluidas con un despliegue al ancho del área dentro de cada dispositivo (Labrada & Salgado, 2013).



Figura 15: Diseño adaptativo o responsive (Labrada & Salgado, 2013)

3. MARCO METODOLÓGICO

3.1. Diseño del sistema

El sistema planteado está comprendido por sensores de variables fisiológicas, los cuales permiten tener una lectura del paciente por medio del sistema de adquisición de datos, el mismo que está compuesto por tarjetas de desarrollo para complementar la etapa de adquisición y tratamiento de datos de las variables medidas. El dispositivo de conectividad a internet permite el intercambio de información con los diferentes usuarios de la red de telemedicina, el servidor o nube es el encargado de realizar la conexión al módulo de adquisición con la base de datos y las aplicaciones web o móvil, la base de datos es responsable de guardar variables y generar reportes. Para finalizar, la página web y la aplicación móvil se presentan como las interfaces para el usuario final, las cuales son las encargadas de mostrar las variables en tiempo real y las variables históricas.

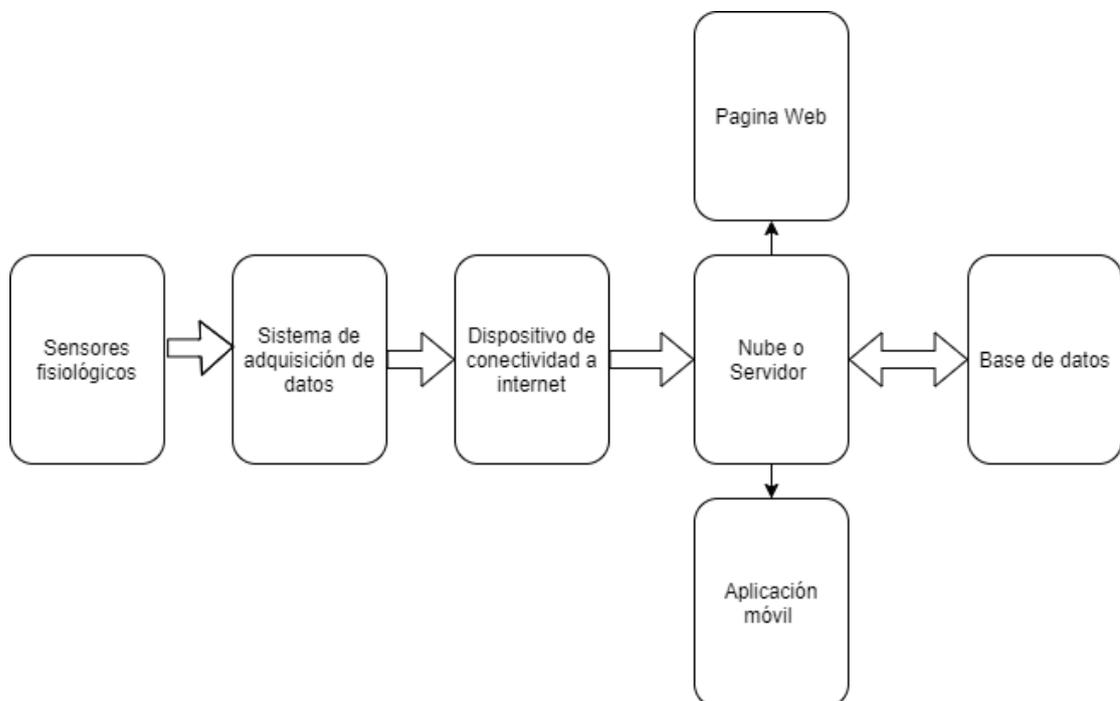


Figura 16: Diseño general del prototipo desarrollado

3.2. Implementación del módulo de telemedicina

En los inicios del proyecto, se pretendía utilizar como sistema de adquisición de datos la plataforma e-health sensor de la Figura 7 debido a su fácil y rápida interacción con los sensores disponibles en la marca. No obstante, de acuerdo con la web, el producto se encuentra descontinuado y para mantener el sistema de adquisición propuesto, se procedió a realizar un prototipo en la placa de desarrollo Arduino, el cual cumple con esta función, y se empleó sensores comerciales con comunicación compatible con esta placa.

Asimismo, para disponer de manera simultánea de la medición de diferentes variables fisiológicas, que complementan la información general sobre el estado de salud de un paciente, se dispone de la creación de un módulo compacto que incluye una tarjeta de control para el procesamiento de datos que recolectan los sensores. Así, como la propia electrónica que permite la emisión de toda esta información hacia un servidor, y esta se encuentre disponible para su presentación en una interfaz gráfica. La construcción e implementación del módulo será descrita en tres secciones correspondientes a: placa electrónica, programación y carcasa.

3.2.1. Placa electrónica

3.2.1.1. Sensores

Como parte esencial de la placa electrónica del módulo de telemedicina, se tiene la recolección de datos mediante el uso de dispositivos de medición apropiados, para cada una de las variables fisiológicas propuesta. Además, la mayoría fueron seleccionados debido a la forma en cómo se transmiten los datos, en este caso será el protocolo I2C, que mejora la integración de los sensores.

3.2.1.1.1. Sensor de temperatura corporal

Para la implementación de este módulo, se ha optado por utilizar el sensor MLX90614 que se muestra en la Figura 1, el cual ha sido fabricado por la empresa Melexis. Este sensor es capaz de medir la temperatura de un objeto sin necesidad de entrar en contacto con él.

3.2.1.1.2. Sensor de saturación de oxígeno en sangre

Con el propósito de facilitar la toma de mediciones en pacientes, se busca que el método a utilizar sea no invasivo, por lo que se optó por emplear el Pulsioxímetro MAX30100, que se observa en la Figura 3, mismo que emplea un circuito fotoeléctrico para proporcionar el valor de SpO₂ en la sangre.

Este dispositivo proporciona el ritmo cardíaco del paciente a través del número de pulsaciones por minuto. Sin embargo, este valor puede obtenerse de otros sensores que brindan mayor precisión.

3.2.1.1.3. Tensiómetro de presión arterial

Para conseguir una medida confiable de esta variable seleccionó un dispositivo comercial, como el que se presenta en la Figura 2. Este instrumento es capaz de realizar la medición de manera automática, mediante el envío de aire a una pulsera que expande su tamaño, el dispositivo escanea el pulso del paciente y así se obtiene el valor de la presión arterial. De la misma forma, proporciona un valor de los bpm, los cuales son visualizados en el display.

Al ser un dispositivo comercial, fue necesario investigarlo y examinarlo para conocer las modificaciones requeridas y así, obtener los datos deseados. De esta manera, se conoció que este dispositivo maneja comunicación I2C, por lo que se dispuso a localizar en esta placa electrónica los puntos accesibles correspondientes a los pines SCL y SDA.

También, se ubicó el punto negativo de la fuente de alimentación del dispositivo que debe unirse al RESTo conexiones y de esta forma, todos los elementos comparten la misma referencia. En la Figura 17 se muestra la placa del dispositivo y los cables añadidos.



Figura 17: Modificación del monitor de tensión arterial (Cable amarillo: SDA, cable rojo: SCL y cable marrón: Negativo)

Luego de agregado los cables, se selló el dispositivo asegurando una correcta disposición de los cables en la estructura del monitor de tensión arterial como se observa en la Figura 18.



Figura 18: Monitor de tensión arterial sellado luego de la modificación.

3.2.1.1.4. Sensor de ritmo cardíaco

Para monitorear la actividad eléctrica del corazón (electrocardiograma) se ha empleado el módulo AD8232 (ver Figura 5), el cual funciona en conjunto con unos electrodos colocados apropiadamente en el cuerpo del paciente, específicamente en pecho, brazos y piernas, que pueden captar las señales eléctricas que se envían al corazón durante su latido. De ese modo, el módulo amplifica y filtra las señales para obtener los valores a ser utilizados.

Si bien el manejo del módulo no es muy complicado, su medición depende ampliamente de una correcta posición de los electrodos, pues si, no son colocados en los puntos apropiados del cuerpo, los valores medidos pueden ser incorrectos y la señal del ritmo cardiaco puede alterarse (García, 2004). En la Figura 19 se muestra la forma como fueron colocados los electrodos en el cuerpo de los pacientes de prueba (Kanani & Padole, 2018).

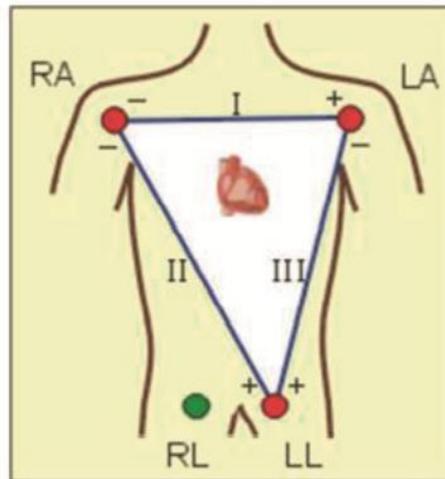


Figura 19: Colocación de los electrodos (RA=R, LA=L y LL=COM) (Kanani & Padole, 2018)

3.2.1.2. Conexión

La implementación del módulo usó dos tipos de tarjetas electrónicas: Arduino Nano y ESP8266, la primera trabaja con los datos recopilados, mientras, la segunda tarjeta tiene como función en el módulo, enviar todos los datos a un servidor. El esquema general de la distribución de estas tarjetas en conjunto con los sensores conectados es presentado en la Figura 20.

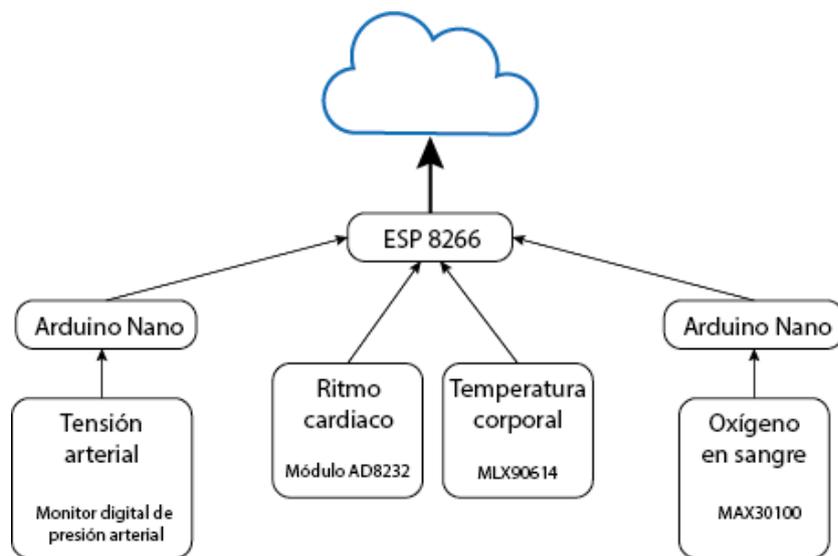


Figura 20: Esquema general de funcionamiento del módulo

Debido a la adaptación que se realizó al monitor de presión arterial comercial, se requirió de una tarjeta de desarrollo exclusivamente para este aparato, y así enviar los datos desde el monitor hacia el Arduino nano utilizando comunicación I2C.

El resto de los sensores involucrados en el módulo fueron distribuidos en dos tarjetas: ESP8266 y Arduino Nano, a causa de conflictos durante la inicialización en conjunto del sensor de temperatura y el oxímetro, cuando fueron conectados en la ESP8266; motivo por lo que se decidió emplear la tarjeta Arduino para comunicarse mediante I2C con el sensor MAX30100. Los sensores correspondientes al sensor de ritmo cardíaco y de temperatura corporal se mantuvieron en la tarjeta ESP8266.

Como se mencionó anteriormente, adicionalmente a la conexión de los dos sensores, la tarjeta ESP 8266 es la encargada de conectarse a una red Wifi para enviar los datos recopilados a un servidor y posterior a ello, estos puedan ser visualizados en la interfaz gráfica desarrollada.

3.2.1.2.1. Comunicación

El módulo maneja dos tipos diferentes de protocolos de comunicación: I2C y UART. El primer tipo de comunicación es empleado en la interacción entre los sensores y las tarjetas de desarrollo, específicamente el sensor de temperatura, el oxímetro y el monitor de presión arterial, con sus respectivas tarjetas. El otro tipo de comunicación se utiliza para enviar la información de las tarjetas Arduino hacia la tarjeta ESP.

3.2.1.3. Ensamblaje del módulo

3.2.1.3.1. Diseño de la placa electrónica

Para el armado, primero se comprobó el funcionamiento los diferentes sensores de manera individual. Posteriormente, se realizó la conexión de todos los componentes juntos sobre un protoboard para comprobar su correcto funcionamiento, tal como se presenta en la Figura 21.

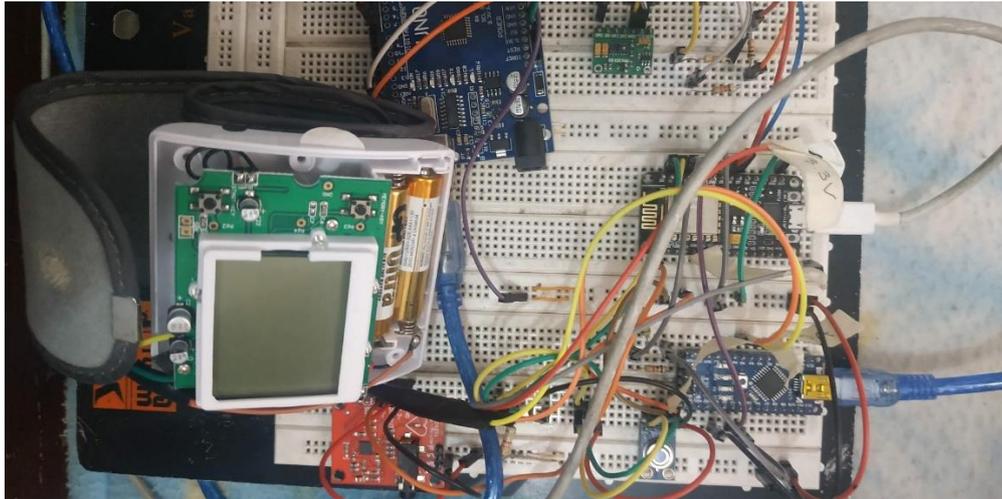


Figura 21: Implementación del módulo sobre un protoboard

Una vez que se comprobó el funcionamiento conjunto de sensores, comunicación y envío de datos hacia el servidor, se inició con el diseño de la placa electrónica utilizando el software Proteus. Para comenzar, se utilizó la aplicación ISIS de Proteus para realizar el diseño esquemático de todo el circuito.

A pesar de que el programa Proteus tiene por defecto una gran cantidad de elementos electrónicos dentro de su catálogo para la formación de circuitos, existen componentes, específicamente tarjetas, que no poseen un componente virtual para insertarlo en el circuito. Por tal motivo, es necesario crearlo en base al diagrama de pines que proporciona el proveedor. Este proceso se realizó para la tarjeta ESP8266 y para el Arduino Nano.

Para este módulo, la placa se diseñó de tal forma que las tres tarjetas y sus sensores se encuentren en la misma área; lo cual permite que sean alimentadas con una sola fuente de energía. En la Figura 22 se muestra el diseño esquemático de los dos circuitos Arduino Nano, dentro de esto, se incluye la conexión para: el sensor de saturación de oxígeno en la sangre, el monitor de tensión arterial y las resistencias de pull-up para la comunicación I2C.

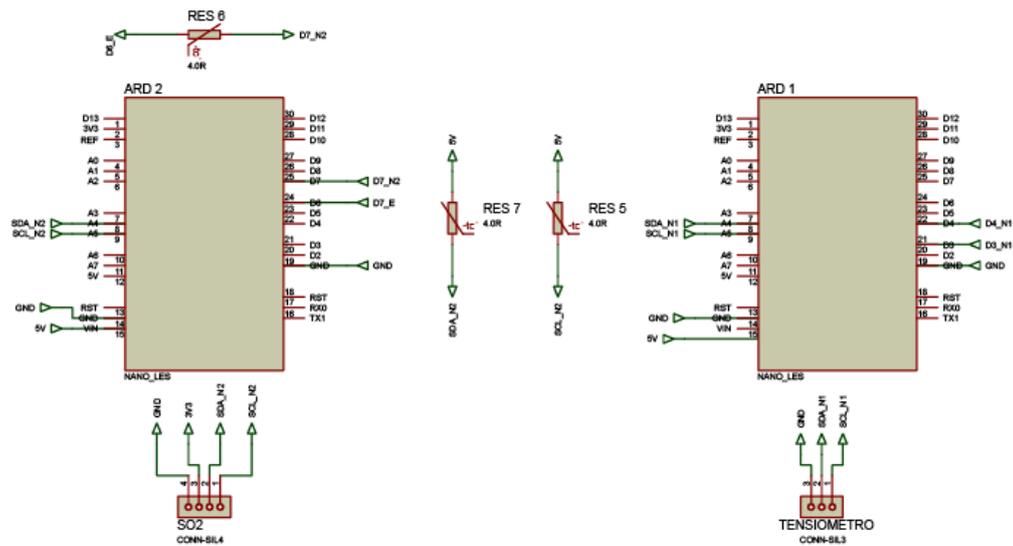


Figura 22: Diseño esquemático de los dos circuitos Arduino Nano

Por otro lado, en la Figura 23 se presenta el diseño del circuito esquemático de la tarjeta ESP8266. La gráfica consta principalmente de la tarjeta ESP8266, el sensor de temperatura, el sensor de ritmo cardiaco, los leds indicadores y las resistencias de pull-up para la comunicación I2C.

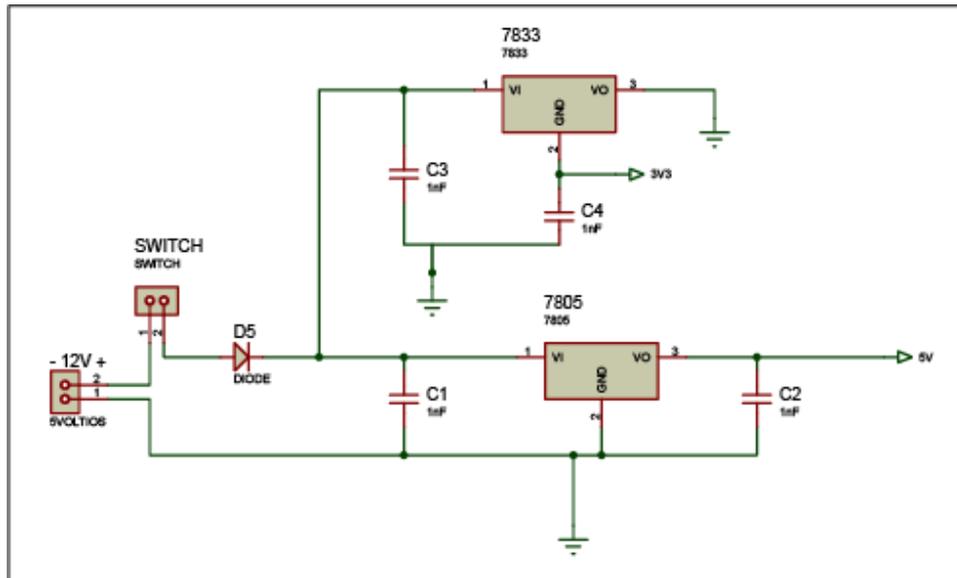


Figura 24: Circuito de alimentación del módulo

Con el diseño terminado, se procede a utilizar la aplicación ARES de Proteus para el diseño del PCB. Cada elemento fue colocado de tal manera que todas las rutas generadas estén presentes en un solo lado de la placa y que los elementos no presenten una posición que dificulte su conexión. El diseño final se presenta en la Figura 25.

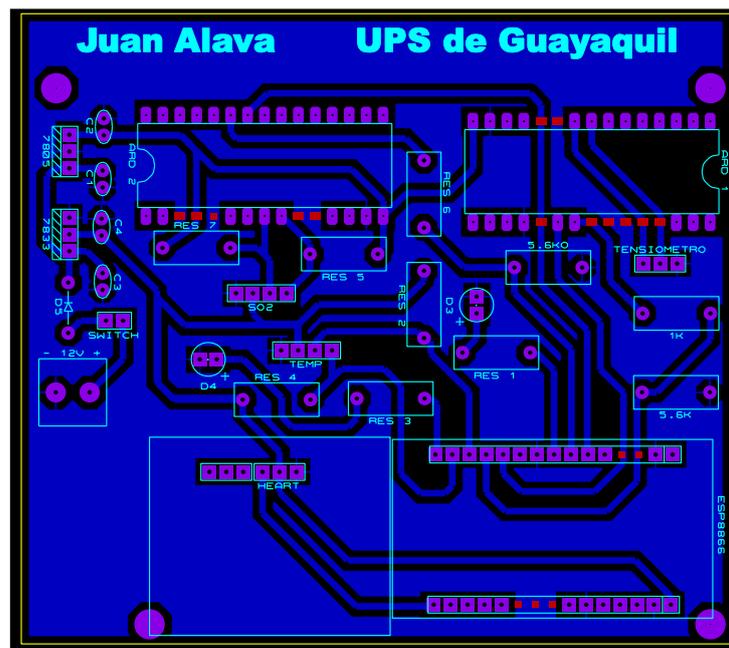


Figura 25: Diseño PCB del módulo

3.2.1.3.2. Montaje de la placa

El proceso de elaboración de la placa se realizó en baquelita y mediante el proceso de planchado se transfirió el diseño del PCB al lado de cobre de la placa. Luego, se utilizó ácido férrico para eliminar el cobre no deseado de la placa, y de esta forma solo quede las líneas de conexiones deseadas. Finalmente, se usó thinner para retirar la tinta que cubre las conexiones y se realizó los orificios para colocar los elementos.

Cabe mencionar que antes de colocar los elementos es necesario revisar la continuidad de las pistas de la placa, esto para verificar que no se presente algún corto circuito debido a fragmentos de cobre presentes. Una vez corroborado esto, se colocó y se soldó los elementos. La placa resultante se presenta en la Figura 26

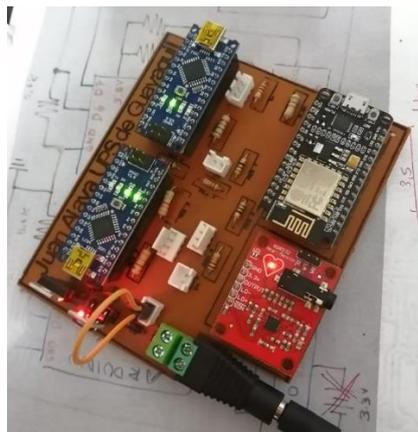


Figura 26: Placa electrónica del módulo de telemedicina

3.2.2. Programación del módulo

La programación total del módulo está dividida en las tres tarjetas (dos tarjetas Arduino Nano y una tarjeta ESP8266). El flujo principal de trabajo del módulo está enfocado en la tarjeta ESP8266, la cual recopila la información que será enviada a un servidor. Se emplearon varias librerías para el funcionamiento de los sensores, configuraciones de la comunicación y envío de datos hacia el servidor. Esto se esquematiza en los diagramas de flujo mostrados en la Figura 27 y Figura 28.

3.2.2.1. Programa principal (ESP8266)

La ESP8266 aloja el programa principal, ya que recopila los valores medidos de todos los sensores y arma la estructura de datos a enviar hacia el servidor. En la Figura 27 se muestra el diagrama de flujo del programa a detalle de la tarjeta ESP8266. En esta parte se incluyen a las respectivas librerías, declaración de variables, inicialización del sensor de temperatura, y establecimiento de la conexión a internet en conjunto con el encendido de un LED; que funciona como indicador en caso de una conexión exitosa.

El diagrama de flujo del lazo principal se observa en la Figura 28, donde se muestra la lectura de datos de los sensores en las otras tarjetas y la lectura de los sensores en la propia ESP8266. Cuando todos los datos son obtenidos se forma el cuerpo del mensaje con una estructura establecida por los API-REST. Por último, si la conexión a internet es correcta se realiza el envío de datos al servidor mediante el consumo del API-REST. Si el servidor responde de manera correcta un LED se enciende por un instante, esto indica que el envío de datos no tuvo ningún problema. Las instrucciones se vuelven a ejecutar en un lazo infinito, siempre y cuando el módulo esté energizado y exista conexión a internet.

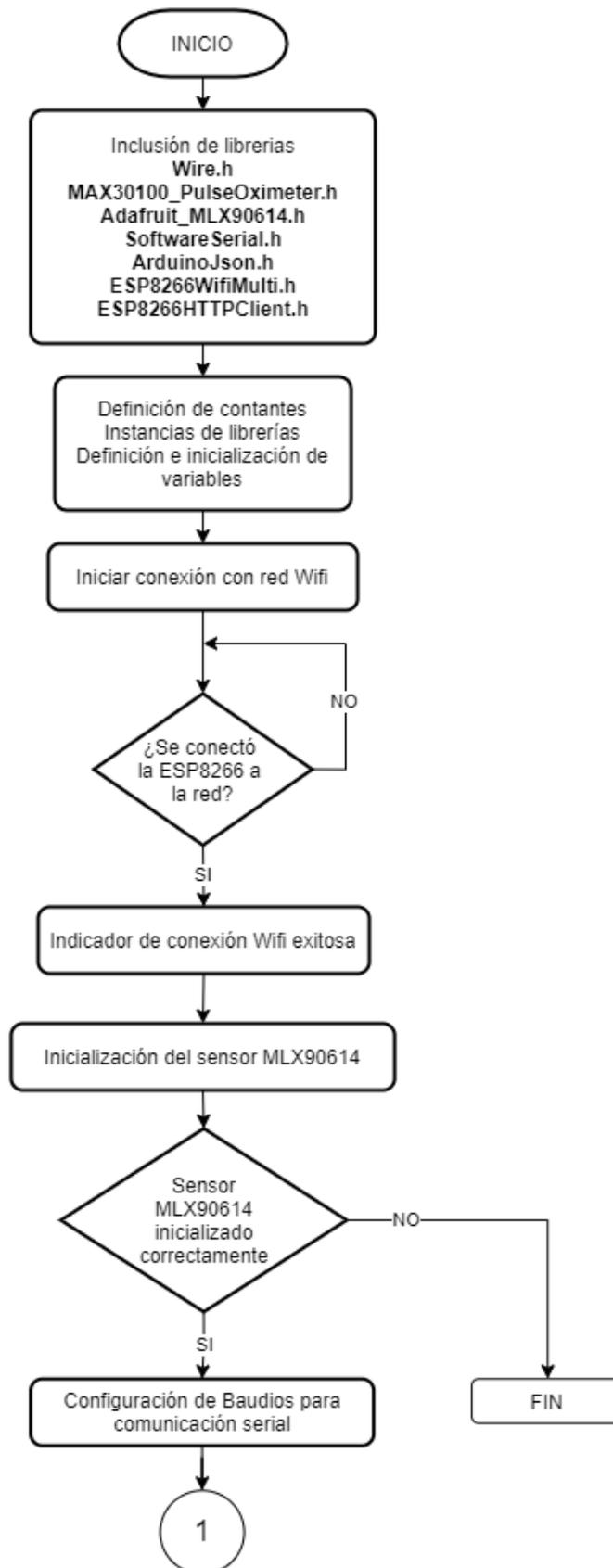


Figura 27: Diagrama de flujo (Configuración inicial) de la ESP8266

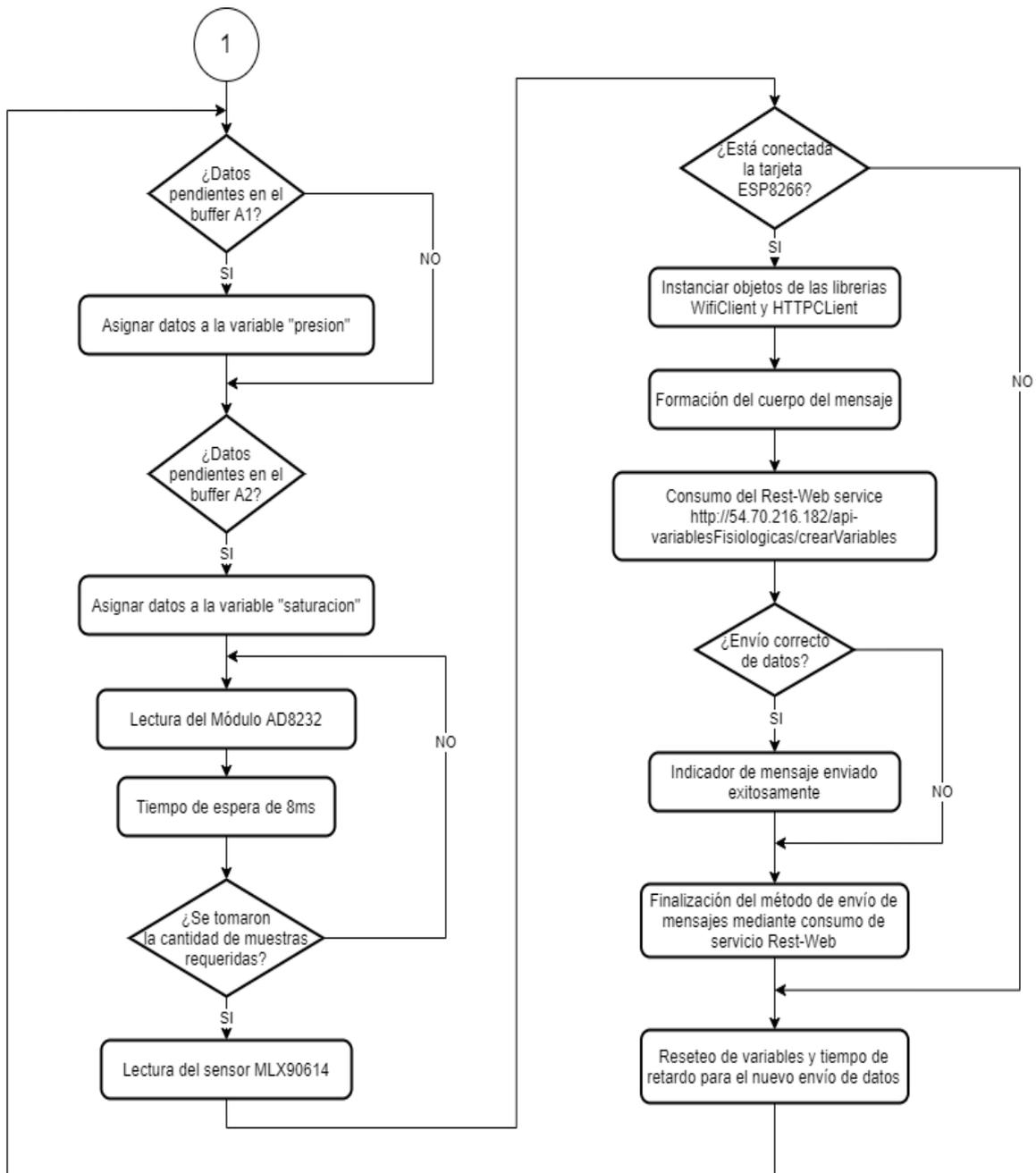


Figura 28: Diagrama de flujo (Void Loop) de la ESP8266

3.2.2.2. Programa Arduino Nano 1 (A1)

El programa principal de la tarjeta Arduino o A1 se conecta al monitor digital de tensión arterial. Su función es recopilar las mediciones realizadas por el dispositivo y establecer comunicación con la tarjeta ESP8266 para el envío.

Cuando el dispositivo termina de realizar la medición correctamente, los valores se muestran en el display, y son guardados en la memoria del dispositivo, después que el dispositivo se apague manualmente. Esta acción permite que los datos viajen por I2C a la memoria del monitor de tensión arterial, durante este proceso se asigna la misma dirección de la memoria ROM del dispositivo, al Arduino Nano y así se genera el evento de almacenamiento. El envío de datos es detectado por el Arduino y los datos pueden leerse.

Los datos llegan en una misma trama de bytes y para localizar los datos con la información requerida, se realizaron varias pruebas y se encontró que dichos datos llegan mediante la interrupción generada. El flujo de trabajo es mostrado en la Figura 29.

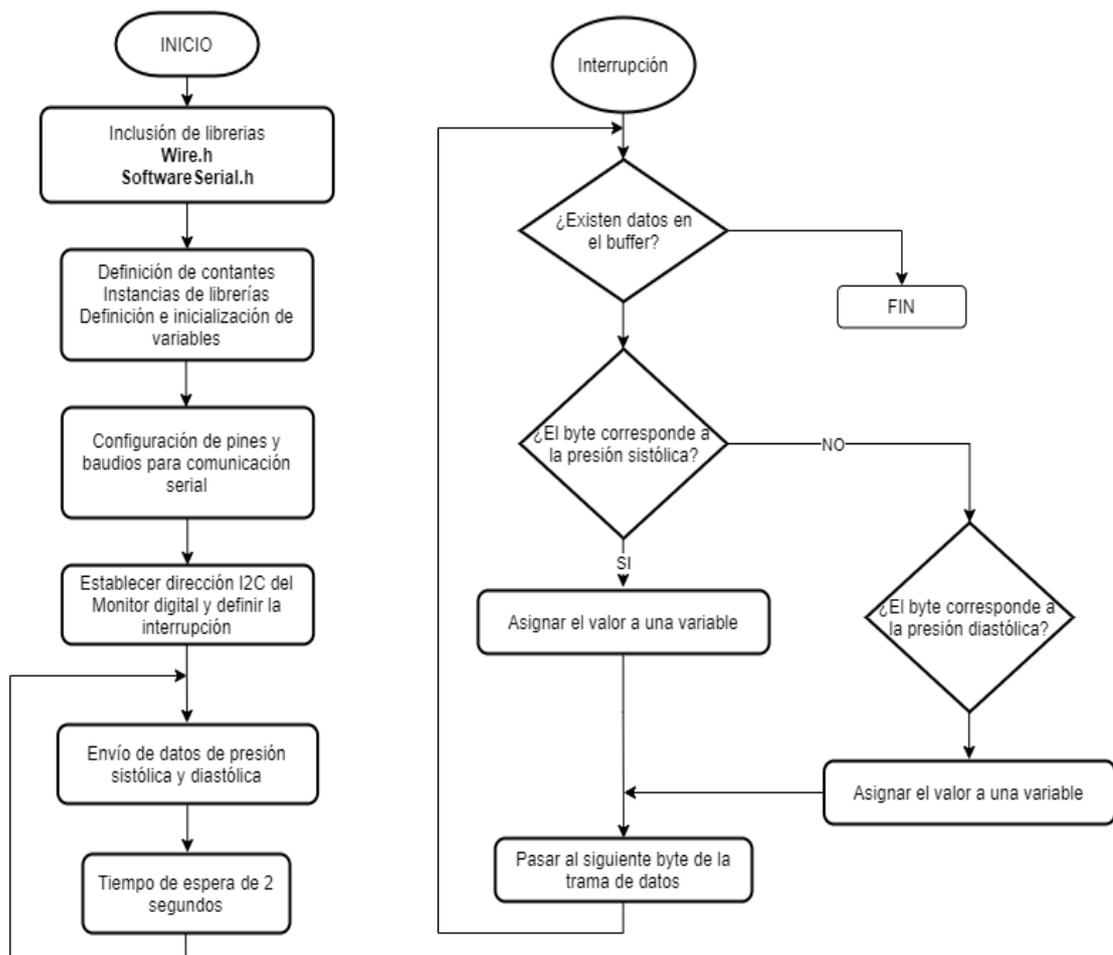


Figura 29: Diagrama de flujo del programa en Arduino Nano A1

3.2.2.3. Programa Arduino Nano 2 (A2)

La tarjeta de desarrollo se programó para manejar los datos obtenidos del sensor MAX30100, encargado de la saturación de oxígeno en la sangre. De manera similar al Arduino A1, las tareas principales de la tarjeta son: la medición del parámetro, asignación a la variable correspondiente y envío de datos hacia la tarjeta ESP8266. En la Figura 30 se presenta el flujo de trabajo en esta tarjeta.

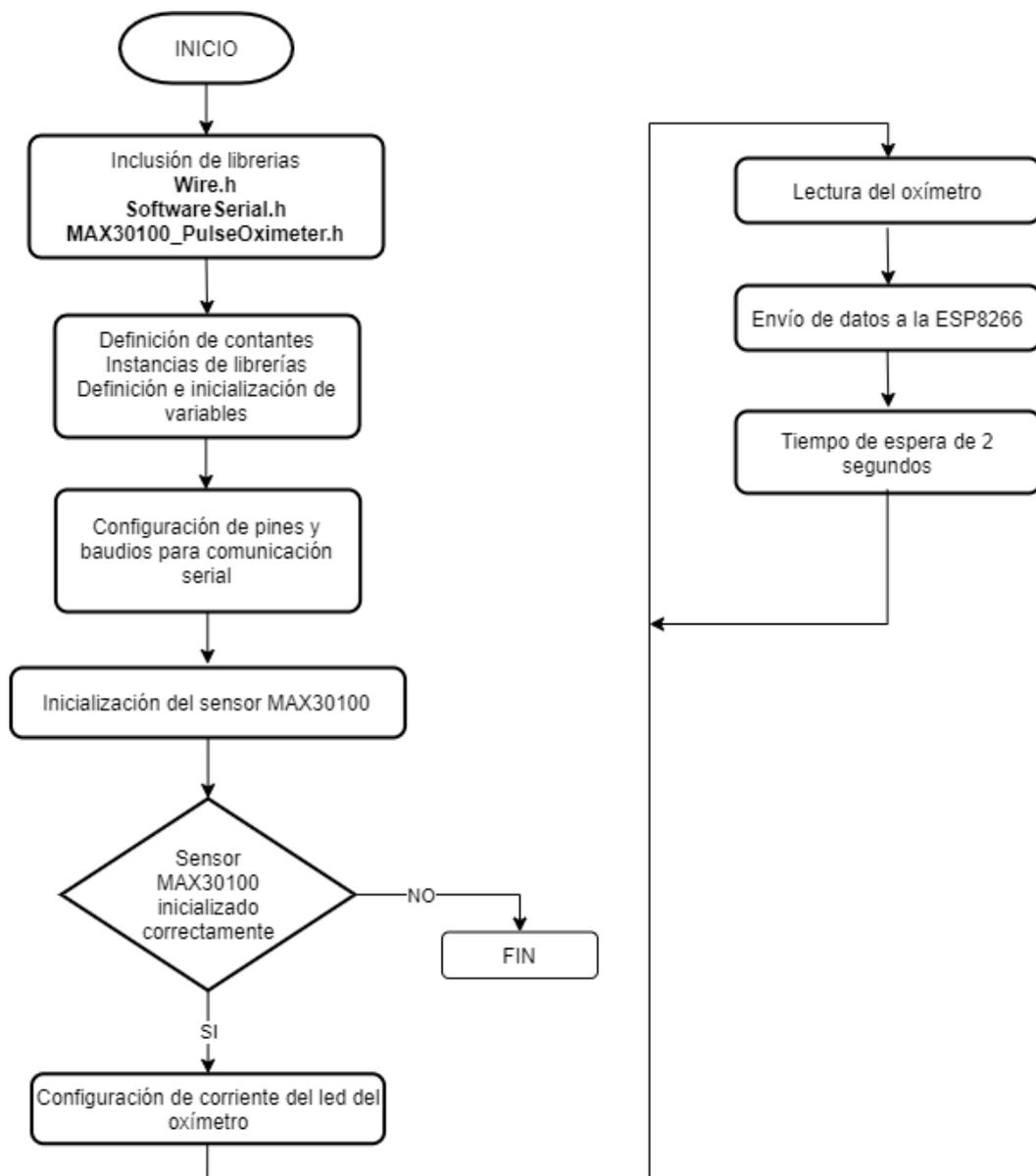


Figura 30: Diagrama de flujo del programa en Arduino Nano A1

3.2.3. Diseño de carcasa

Una vez armada la placa electrónica a la que se conectan todos los componentes, se procedió a diseñar la carcasa con apoyo en las medidas de la placa y de los sensores involucrados. Para su esquematización, se utilizó el software Inventor de Autodesk que permitió diseñar las cuatro partes de la carcasa correspondientes a la caja, tapa y dos partes adicionales para el sensor de temperatura. En la Figura 31 se puede apreciar cada una de las partes diseñadas de manera individual, mientras que en la Figura 32 se muestra el ensamblaje de estas partes, realizado en el mismo software.

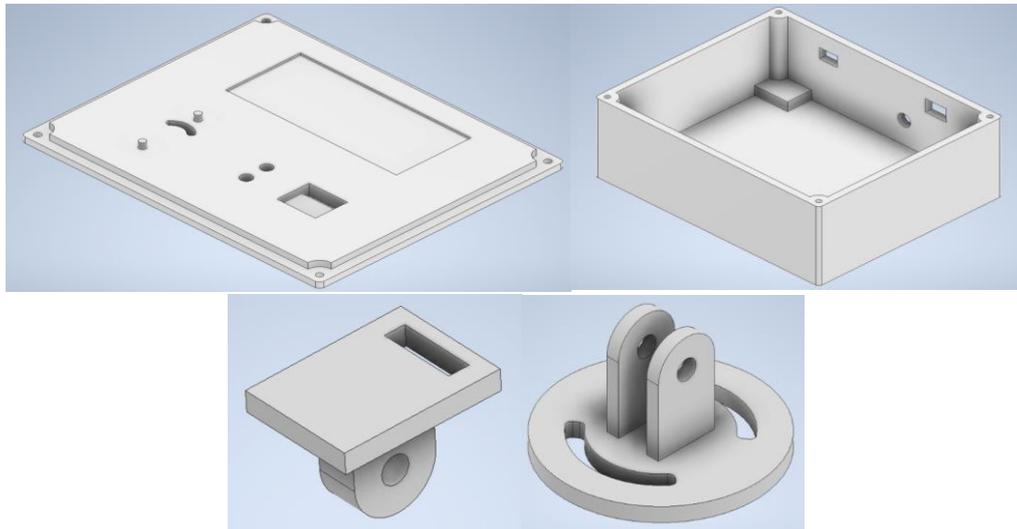


Figura 31: Partes individuales de la carcasa del módulo

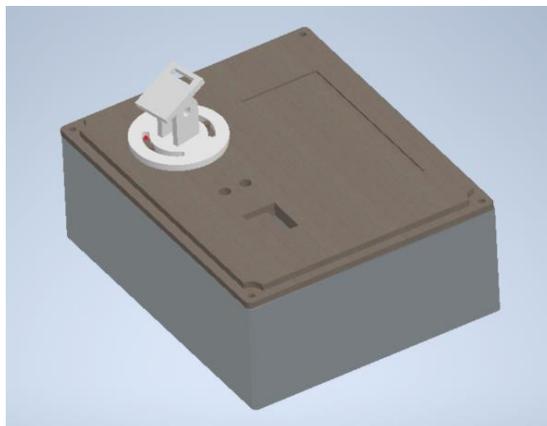


Figura 32: Ensamblaje digital de toda la carcasa del módulo.

El diseño de las partes en Inventor y el ensamble virtual permitió verificar que no se presenten inconvenientes en las medidas. Por las ventajas que presenta la impresión en 3D, fue el método seleccionado para la construcción de las partes diseñadas, las cuales fueron impresas en PLA de color negro y posterior a esto, fueron ensambladas. El producto resultante de la impresión 3D se muestra en la Figura 33.



Figura 33: Impresión 3D de la carcasa (a. Carcasa ensamblada b. Soporte para el sensor de temperatura)

Con la carcasa terminada, se colocó la placa electrónica en su interior, de manera que los cables y cada parte del módulo pueda ser manipulado sin ningún problema. El módulo final y los elementos adicionales necesarios para su funcionamiento son presentados en la Figura 34.



Figura 34: Módulo de telemedicina completamente ensamblado

3.3. Arquitectura del servidor

Se empleó la arquitectura cliente/servidor con el fin de que los clientes conectados realicen requerimientos a este servidor, por medio de solicitudes de tipo http del lado del servidor y formato JSON para las interfaces gráficas. El conjunto de buenas prácticas denominado REST API fue utilizado, tal como se muestra en la Figura 35.

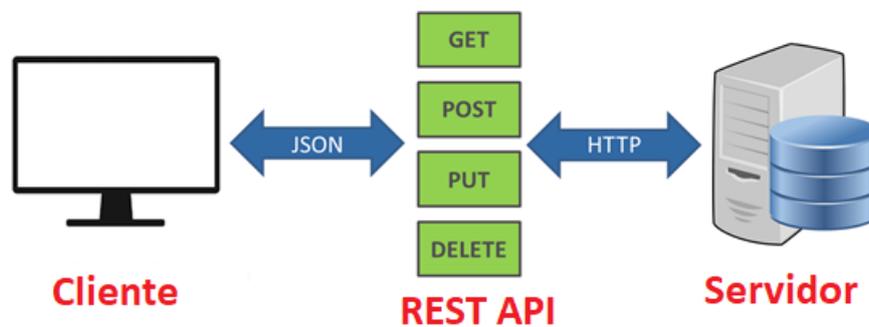


Figura 35: Funcionamiento del servidor a implementar

3.3.1. Servidor virtual

El servidor es una computadora virtual alojada en los servidores de Amazon Web Services, que cumple con las características presentadas en la Tabla 3.

Características	Valor
Sistema Operativo	Ubuntu Server 20.04 LTS
Número de núcleos	1
Memoria Ram	1GB

Tabla 3 Características del servidor virtual

En virtud que la aplicación está destinada para uso privado entre médico paciente, y solamente se procesan variables de tipo documento JSON, se ha determinado que estas son las características apropiadas para el funcionamiento ininterrumpido de la aplicación. Esto asegura la transmisión de datos entre la aplicación web/móvil y el dispositivo de adquisición, siempre que exista una conexión estable de internet.

3.3.2. Creación de los API REST

A continuación, se observa los diagramas de flujo de los servicios creados, los cuales muestran el establecimiento de la conexión entre el módulo de adquisición de datos, la base de datos y las interfaces de usuario.

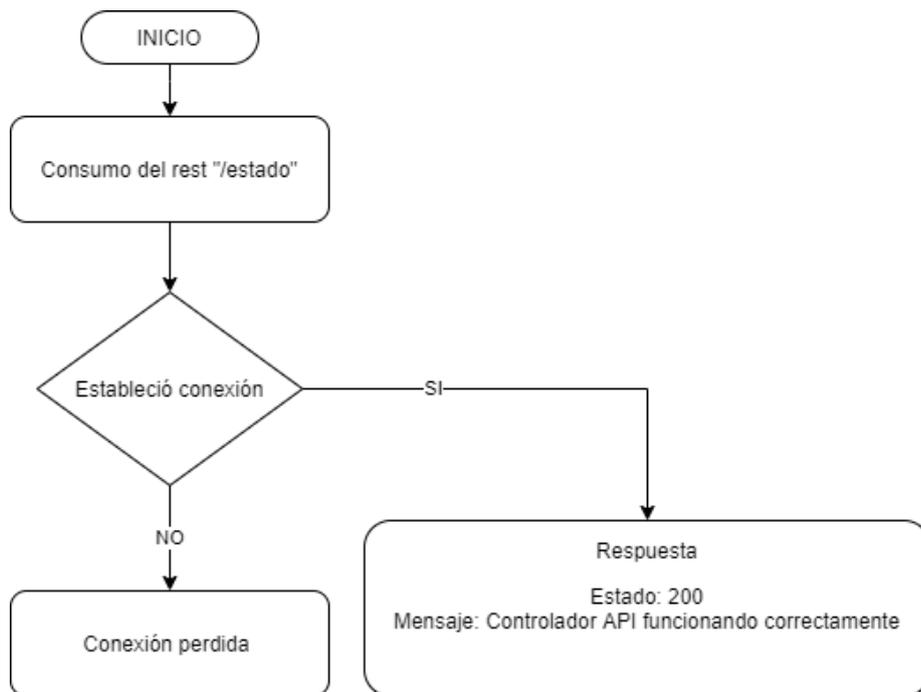


Figura 36: Diagrama de flujo del REST “estado”

La Figura 36 muestra el servicio que permite la verificación rápida del estado del servidor, es el más importante ya que al no estar disponible las subsecuentes peticiones del servidor no se podrán ejecutar, por lo que los

clientes pueden deducir que el servidor está en error debido a que se ha superado la cuota de almacenamiento.

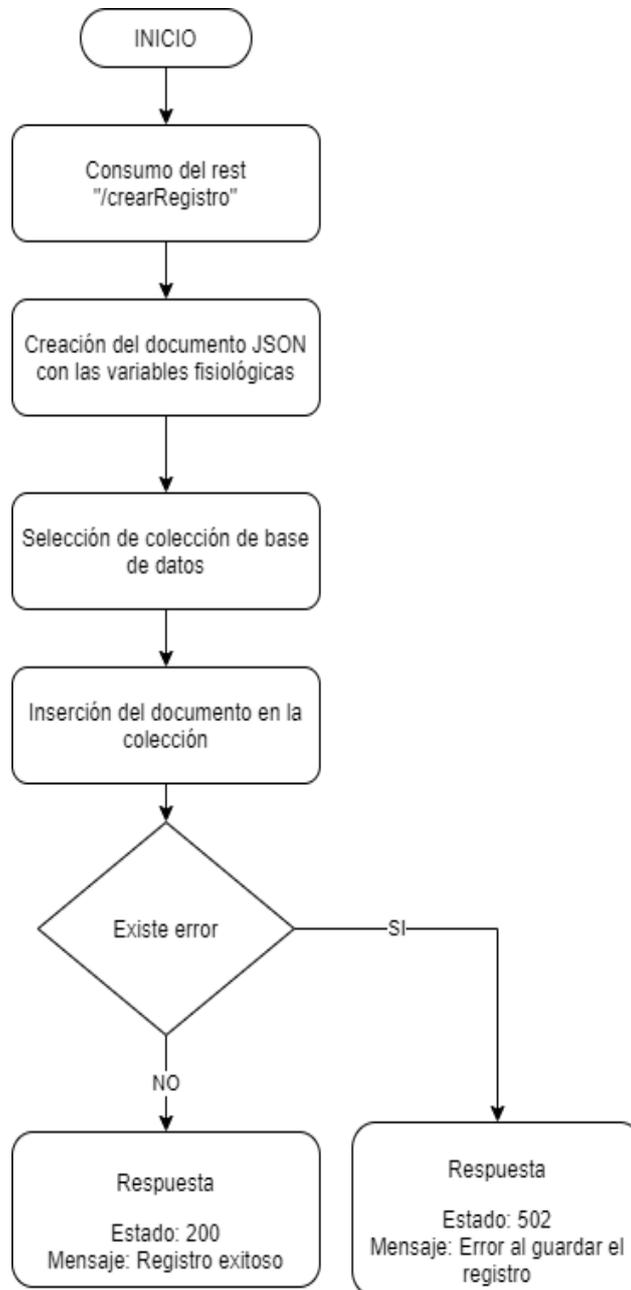


Figura 37: Diagrama de flujo para registrar un paciente en la base de datos.

La Figura 37 presenta el proceso de registro de un paciente en la base de datos con la creación de históricos. Si se presenta un error el cliente puede tomar acciones en la aplicación y desplegarlos de acuerdo con el código de estado.

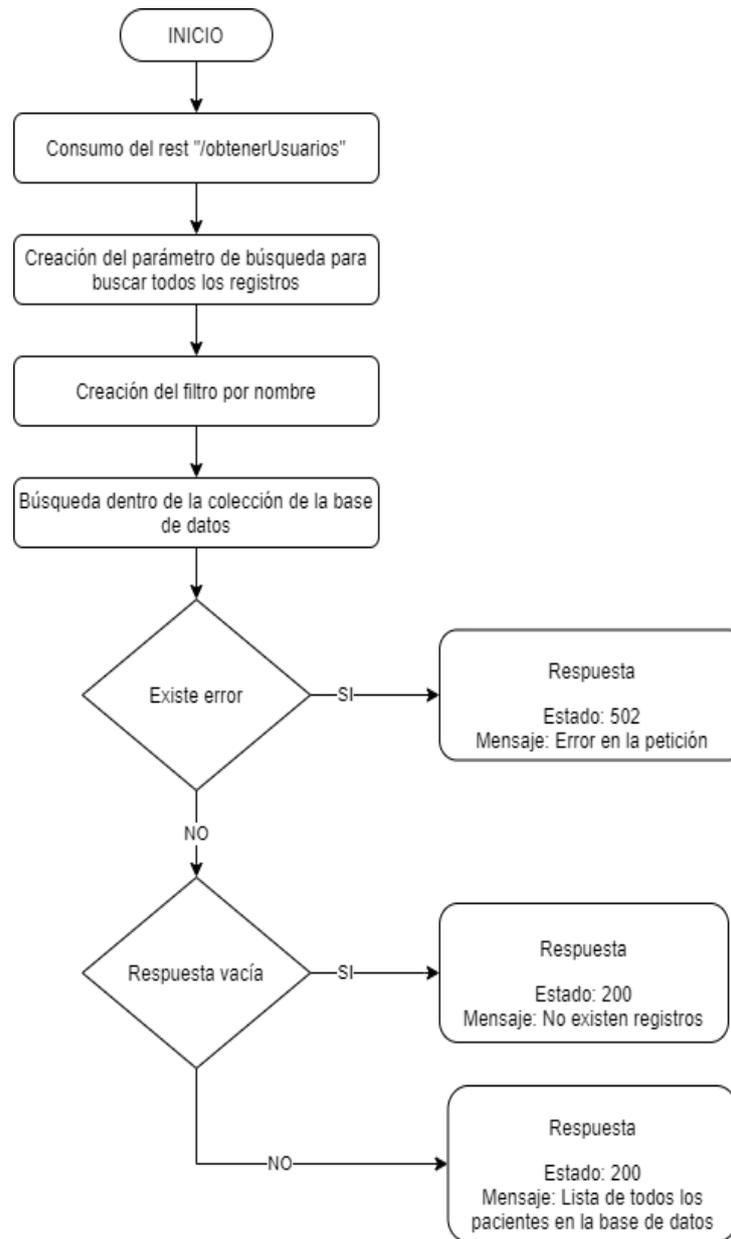


Figura 38: Diagrama de flujo para la obtención de todos los pacientes registrados.

En la Figura 38 se observa la lógica que permite a los clientes obtener la lista completa de todos los pacientes registrados en la plataforma. Adicionalmente, se encuentra el diagrama de flujo que permite acceder a las variables fisiológicas guardadas de un paciente específico, mediante el envío de su nombre como parámetro de búsqueda, tal como se muestra en la Figura 39.

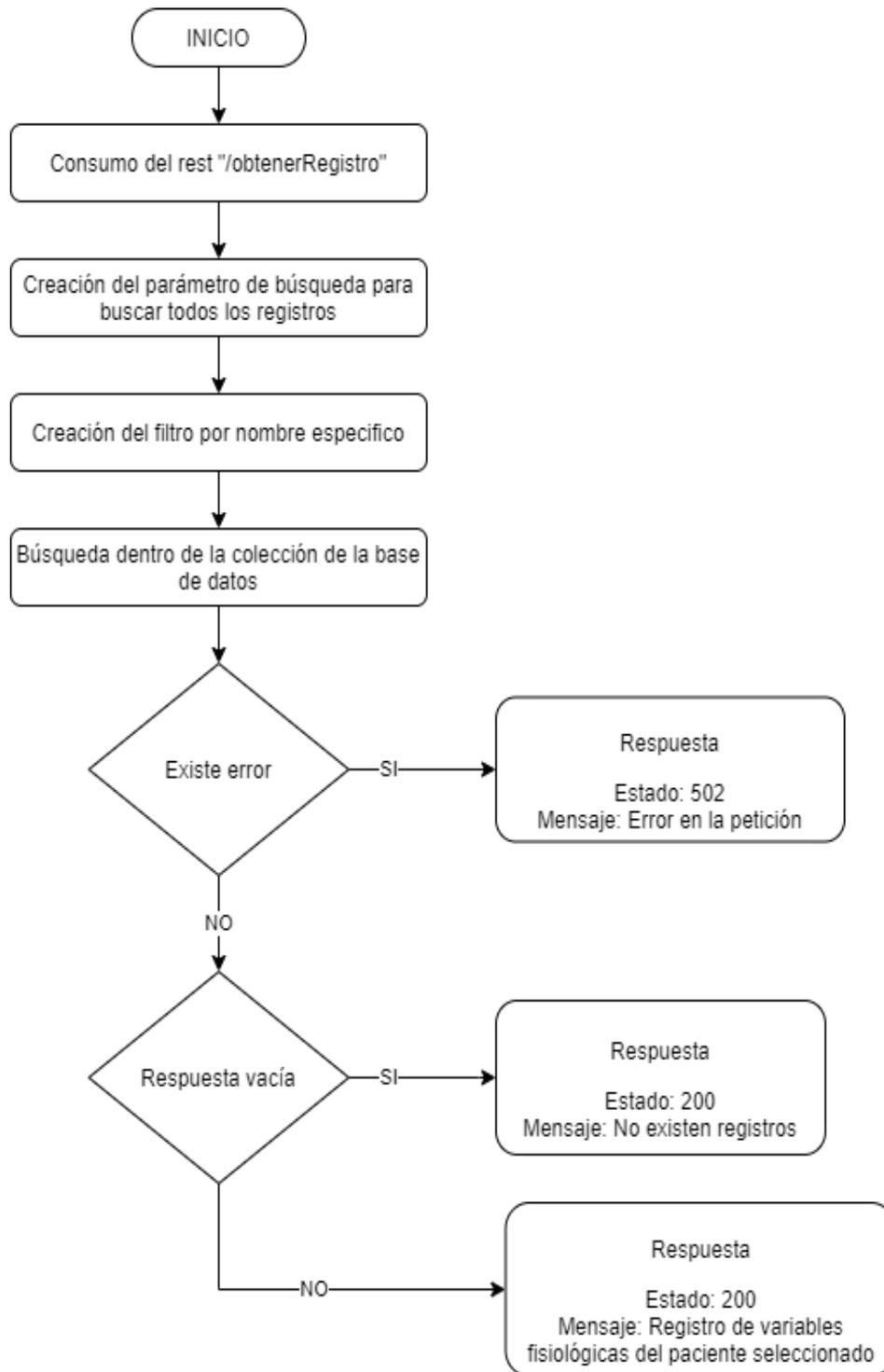


Figura 39: Diagrama de flujo que permite obtener los datos de un paciente específico

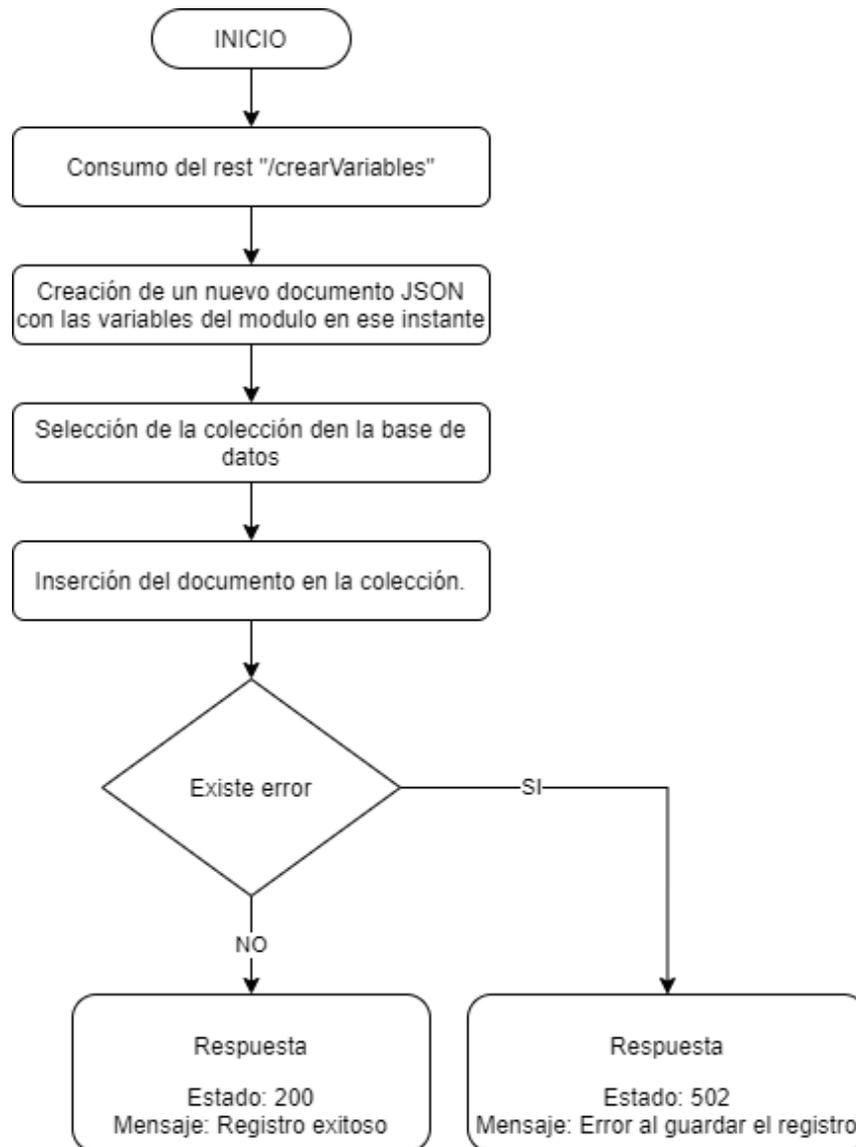


Figura 40: Diagrama de flujo para escribir la lectura de los sensores

La conectividad del sistema de adquisición de variables con la página web y/o aplicación móvil a través del módulo de conexión ESP, requiere de dos pasos: el primero paso es comunicar el dispositivo de internet wifi con la base de datos para un guardado momentáneo como se muestra en la Figura 40, y el segundo paso es realizar una petición de información guardada en la base de datos, esto lo pueden realizar los clientes conectados a la aplicación web o móvil como se muestra en la siguiente Figura.

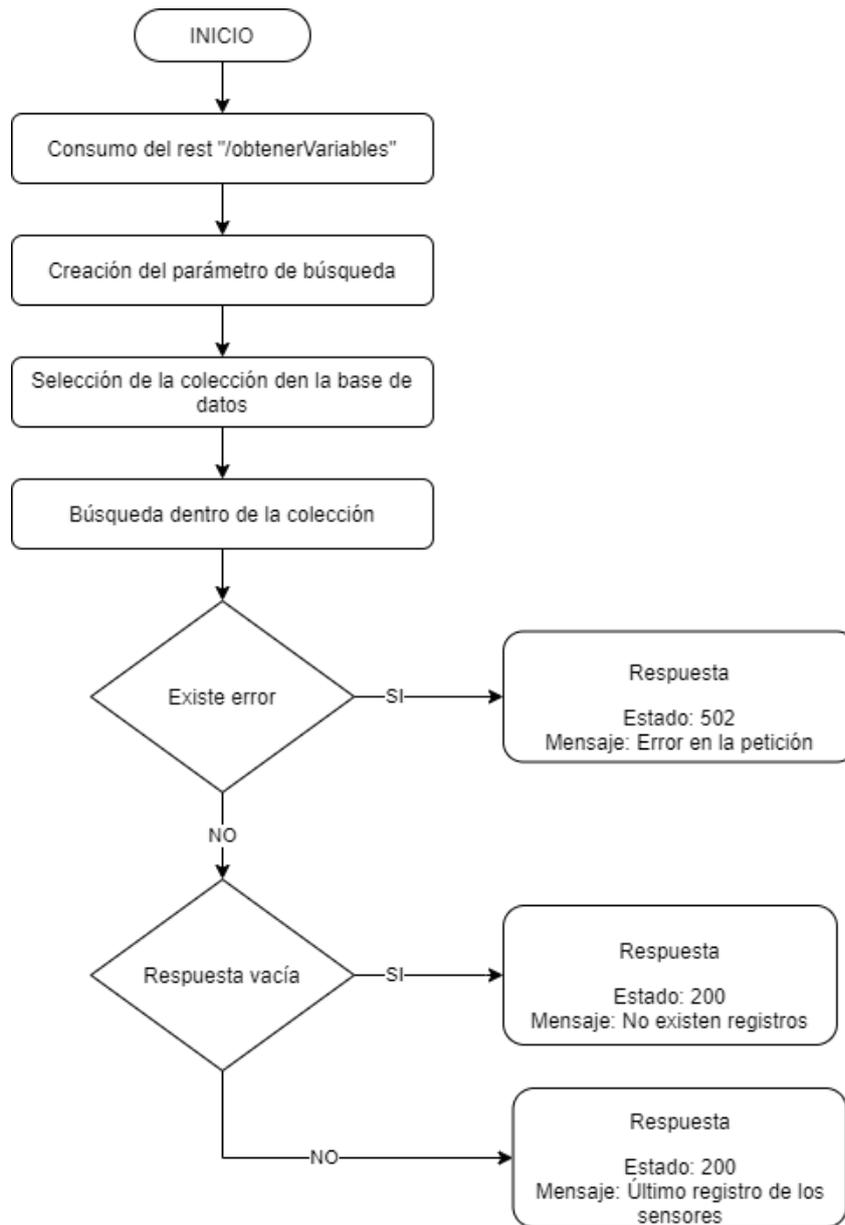


Figura 41: Diagrama de flujo para obtener la lectura de los sensores.

3.4. Página web y aplicativo móvil

Para la interfaz de usuario final se utilizó el framework Flutter, que permite realizar tanto aplicaciones móviles como aplicaciones web. Esta interfaz ofrece la facilidad de usar el mismo código para obtener aplicaciones en ambas plataformas, siempre que se tome en cuenta el criterio de diseño responsive para un resultado satisfactorio en ambas plataformas.

El diseño de los aplicativos se realizó desde un boceto inicial, en donde se visualiza los elementos a mostrar, en este caso, cada uno de los datos recibidos por el servidor, son desplegados en cada uno de sus elementos para su correcta visualización. El mismo boceto sirve para dividir la pantalla y acoplarla a la vista en móvil.

3.4.1. Diseño de la página web

La pantalla de una página web se caracteriza por disponer de un ancho mayor, esto permite generar un boceto en varias secciones; en esta ocasión se realizó cuatro secciones como se muestra en la Figura 42.

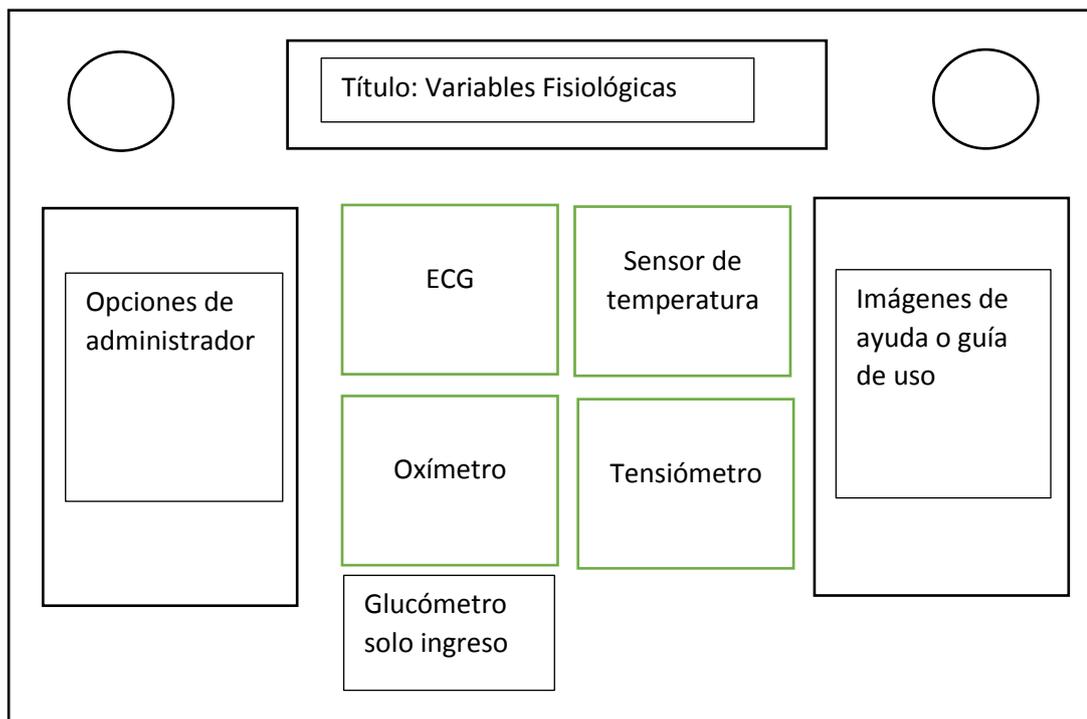


Figura 42: Boceto de la página web

3.4.2. Diseño del aplicativo móvil

Del diseño anterior, se usaron las mismas secciones para acoplar la pantalla a su versión móvil, sin cambiar su distribución.

En las opciones de administrador se emplea un botón tipo sanduche, ubicado al lado superior izquierdo y las demás secciones son distribuidas en una barra de navegación ubicada en la parte inferior de la pantalla, como se aprecia en la siguiente imagen.

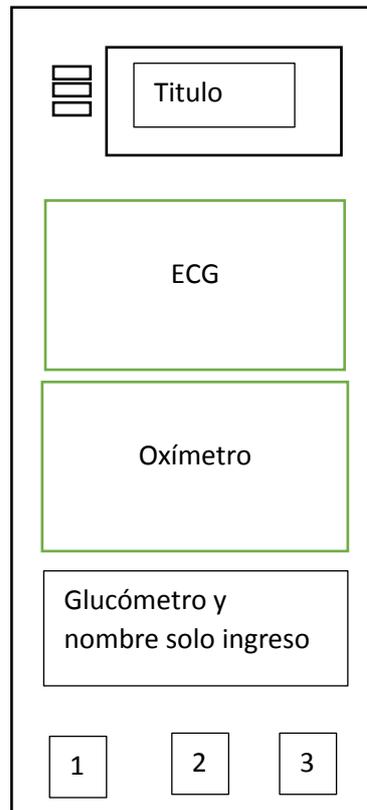


Figura 43: Boceto del aplicativo móvil

3.4.3. Desarrollo de la lógica de programación

A continuación, se presentan los diagramas de flujo utilizados en la creación de la aplicación, dentro de los cuales se encuentra la diferenciación entre web y móvil, y la conexión con el servidor para el manejo de reportes de pacientes.

En primer lugar, la programación determina el tipo de diseño a presentarse, esto se consigue con el despliegue del sistema operativo, que verifica las dimensiones de la pantalla en la cual se ejecuta el programa; este tipo de variables son denominadas runtime o en tiempo de ejecución.

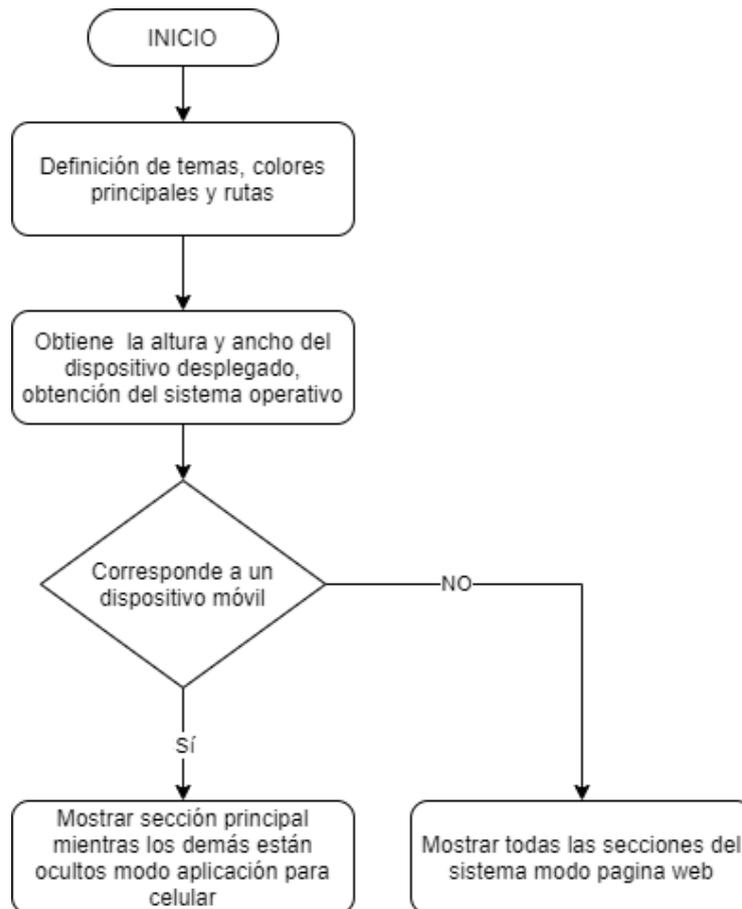


Figura 44: Diagrama de flujo para determinar el dispositivo y la distribución de pantalla a mostrarse

Las aplicaciones web y móvil permiten desarrollar códigos asincrónicos, esto con el fin de facilitar la fluidez entre la interfaz visual y los datos alojados en una base de datos, ya que por la velocidad de respuesta pueden tomar más tiempo en relación con el despliegue de la información.

Los diagramas de flujo que se muestran a continuación presentan el funcionamiento tanto para la aplicación web como para la móvil. Cabe mencionar que tienen una forma similar, sin embargo, el cambio radica en que a diferencia de la versión web, la versión móvil solo muestra la pantalla que está activa mientras las demás permanecen ocultas.

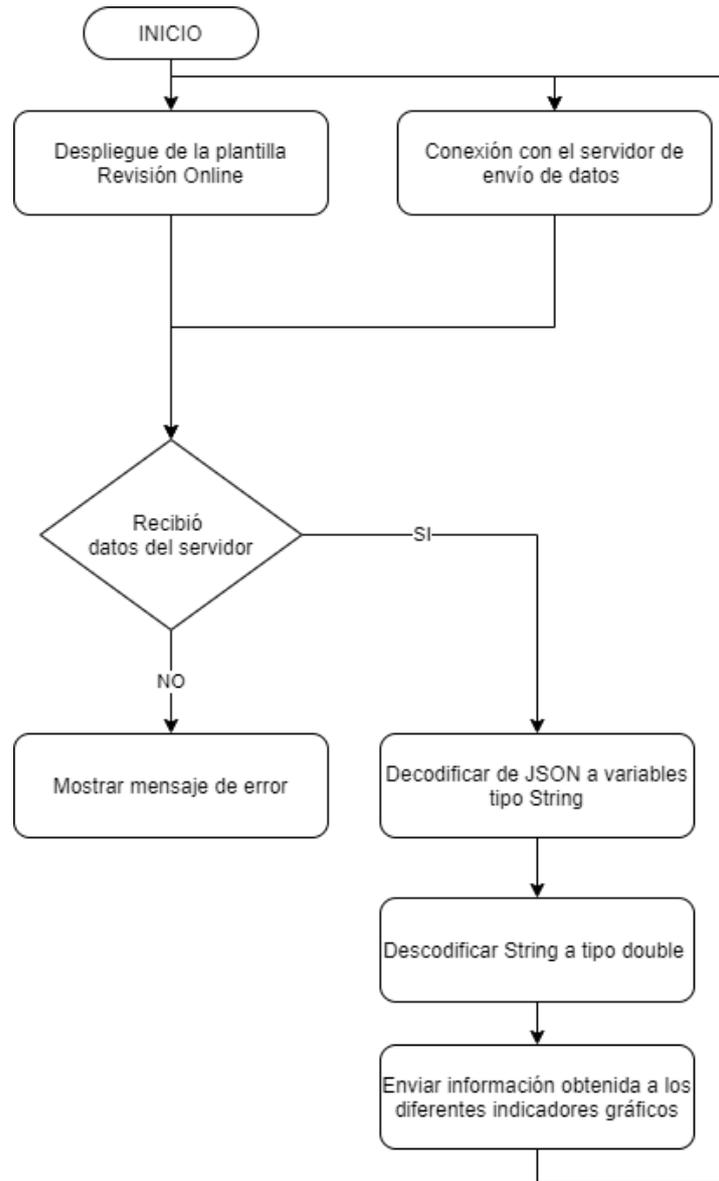


Figura 45: Adquisición de variables del servidor para representarlos en la interfaz visual

Como se observa en la Figura anterior, los requerimientos al servidor se los hace de manera asincrónica y cíclica, con un periodo de muestreo apropiado para establecer una conexión fluida entre la toma de datos del dispositivo y la interfaz gráfica, esto gracias a la arquitectura cliente/servidor implementado.

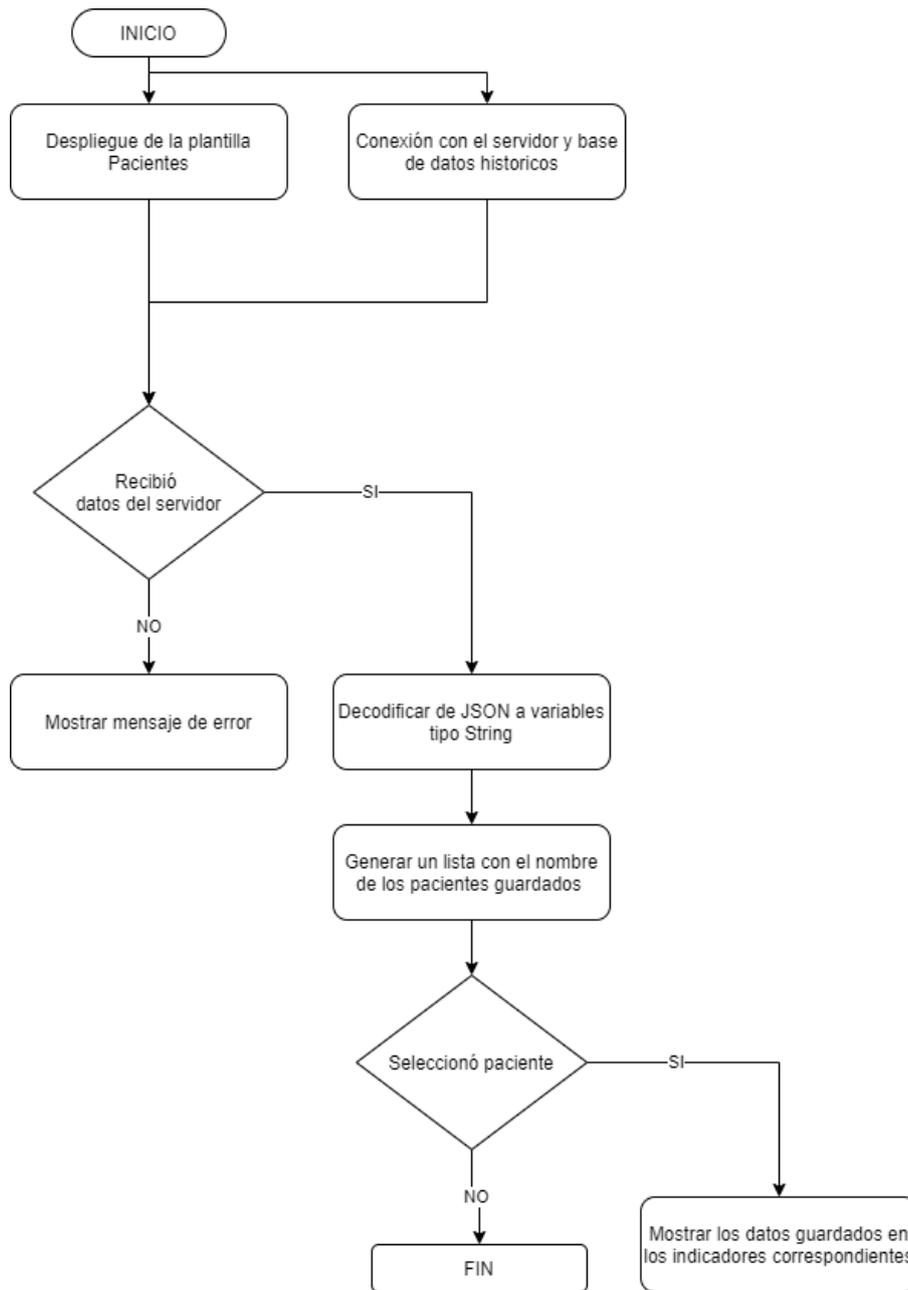


Figura 46: Diagrama de flujo de la pantalla “Pacientes”

De igual forma, para la generación de datos históricos de los pacientes registrados se necesita establecer conexión con la base de datos a través del servidor, el cual provee del listado de pacientes que han sido registrados con anterioridad, y sus datos principales como nombre y variables fisiológicas.

3.4.4. Hosting

El hosting utilizado es provisto por Amazon, el mismo que trae por defecto el dominio personalizado y su cambio puede incurrir en gastos extra que no satisfacen el planteamiento del problema. La dirección web resultante es la que se usa en los navegadores para ingresar a la aplicación web.

Dirección de la página web	http://pruebas.variables-fisiologicas.s3-website-us-west-2.amazonaws.com/#/
----------------------------	---

Tabla 4 Dirección web para el ingreso a la aplicación web

4. RESULTADOS

4.1. Pruebas Iniciales

Se realizó una verificación de todos los sensores involucrados al inicio del proceso de construcción. Sin embargo, dos de estos sensores se tuvieron que probar nuevamente después de terminado el módulo, esto para la calibración del software y obtener mejores resultados.

4.1.1. Sensor MLX30100

El sensor de saturación de oxígeno en la sangre MLX30100, tiene un valor de corriente de los LEDs que puede ser modificado. Después de realizar algunas pruebas, se decidió cambiar el valor de corriente predeterminado de 50mA a 7mA, debido a que con el primer valor era necesario alejar el dedo del paciente y mantenerlo en esa posición, lo que dificulta la medición. Por otro lado, con el valor de 7mA, el dedo del paciente puede descansar sobre el sensor sin ejercer demasiada presión sobre el mismo y de esta forma obtener una medición acertada.

En la Figura 47 se observa las capturas de los resultados conseguidos con las modificaciones realizadas, en la sección Figura 47a se muestran los valores con la corriente predeterminada, y en la Figura 47b se presentan los resultados con el nuevo valor de corriente. En esta última se puede observar progresión hasta alcanzar un valor final, mientras que en el otro caso (Figura 47a) se tiene más fluctuaciones en el resultado.

COM4	COM4
Inicio del Oximetro: Exitoso	Inicio del Oximetro: Exitoso
Ritmo cardiaco: 0.001pm / SpO2: 0%	Ritmo cardiaco: 0.001pm / SpO2: 0%
Ritmo cardiaco: 0.001pm / SpO2: 0%	Ritmo cardiaco: 0.001pm / SpO2: 0%
Ritmo cardiaco: 39.821pm / SpO2: 0%	Ritmo cardiaco: 34.221pm / SpO2: 0%
Ritmo cardiaco: 50.091pm / SpO2: 0%	Ritmo cardiaco: 34.221pm / SpO2: 0%
Ritmo cardiaco: 56.231pm / SpO2: 100%	Ritmo cardiaco: 0.001pm / SpO2: 0%
Ritmo cardiaco: 59.791pm / SpO2: 100%	Ritmo cardiaco: 37.851pm / SpO2: 0%
Ritmo cardiaco: 60.611pm / SpO2: 100%	Ritmo cardiaco: 50.821pm / SpO2: 0%
Ritmo cardiaco: 60.981pm / SpO2: 96%	Ritmo cardiaco: 57.531pm / SpO2: 93%
Ritmo cardiaco: 61.091pm / SpO2: 96%	Ritmo cardiaco: 61.121pm / SpO2: 93%
Ritmo cardiaco: 63.061pm / SpO2: 96%	Ritmo cardiaco: 61.561pm / SpO2: 93%
Ritmo cardiaco: 65.131pm / SpO2: 95%	Ritmo cardiaco: 63.261pm / SpO2: 95%
Ritmo cardiaco: 64.721pm / SpO2: 95%	Ritmo cardiaco: 63.561pm / SpO2: 95%
Ritmo cardiaco: 66.291pm / SpO2: 95%	Ritmo cardiaco: 68.001pm / SpO2: 96%
Ritmo cardiaco: 66.911pm / SpO2: 95%	Ritmo cardiaco: 67.151pm / SpO2: 96%
Ritmo cardiaco: 65.451pm / SpO2: 95%	Ritmo cardiaco: 65.461pm / SpO2: 96%
Ritmo cardiaco: 62.811pm / SpO2: 95%	Ritmo cardiaco: 64.471pm / SpO2: 96%
Ritmo cardiaco: 63.781pm / SpO2: 95%	Ritmo cardiaco: 66.601pm / SpO2: 96%
Ritmo cardiaco: 63.411pm / SpO2: 95%	Ritmo cardiaco: 64.871pm / SpO2: 96%
Ritmo cardiaco: 62.041pm / SpO2: 95%	Ritmo cardiaco: 61.441pm / SpO2: 96%
Ritmo cardiaco: 61.891pm / SpO2: 95%	Ritmo cardiaco: 60.531pm / SpO2: 96%
Ritmo cardiaco: 67.741pm / SpO2: 95%	Ritmo cardiaco: 62.041pm / SpO2: 96%
Ritmo cardiaco: 73.391pm / SpO2: 95%	Ritmo cardiaco: 61.891pm / SpO2: 96%
Ritmo cardiaco: 66.961pm / SpO2: 95%	Ritmo cardiaco: 59.581pm / SpO2: 96%
Ritmo cardiaco: 64.631pm / SpO2: 95%	Ritmo cardiaco: 61.301pm / SpO2: 96%
Ritmo cardiaco: 62.861pm / SpO2: 95%	Ritmo cardiaco: 62.751pm / SpO2: 96%
Ritmo cardiaco: 61.111pm / SpO2: 95%	Ritmo cardiaco: 61.001pm / SpO2: 96%
Ritmo cardiaco: 62.281pm / SpO2: 95%	Ritmo cardiaco: 63.021pm / SpO2: 96%
Ritmo cardiaco: 62.371pm / SpO2: 95%	Ritmo cardiaco: 64.321pm / SpO2: 96%
Ritmo cardiaco: 62.411pm / SpO2: 95%	Ritmo cardiaco: 63.141pm / SpO2: 96%
Ritmo cardiaco: 61.281pm / SpO2: 96%	Ritmo cardiaco: 62.751pm / SpO2: 96%
Ritmo cardiaco: 62.351pm / SpO2: 96%	Ritmo cardiaco: 65.431pm / SpO2: 96%
Ritmo cardiaco: 62.441pm / SpO2: 96%	Ritmo cardiaco: 63.371pm / SpO2: 96%
Ritmo cardiaco: 62.791pm / SpO2: 96%	Ritmo cardiaco: 65.701pm / SpO2: 96%
Ritmo cardiaco: 61.801pm / SpO2: 96%	Ritmo cardiaco: 66.231pm / SpO2: 96%
Ritmo cardiaco: 61.421pm / SpO2: 96%	Ritmo cardiaco: 63.091pm / SpO2: 96%
Ritmo cardiaco: 62.801pm / SpO2: 96%	Ritmo cardiaco: 64.311pm / SpO2: 96%
Ritmo cardiaco: 62.801pm / SpO2: 96%	Ritmo cardiaco: 62.001pm / SpO2: 96%

Figura 47: Valores de medición de la saturación de oxígenos en la sangre y bpm con diferentes valores de corriente en los LEDs a) 50mA y b) 7mA

4.1.2. Módulo AD8232

Este sensor es el responsable de la lectura constante de las señales eléctricas producidas por el corazón. El tipo de señal que se obtiene en el proceso de lectura depende del tiempo de muestreo, en la Figura 48 se exhibe la señal resultante del monitor serial de Arduino, luego de realizar las mediciones sin considerar un intervalo de tiempo entre cada una de ellas.

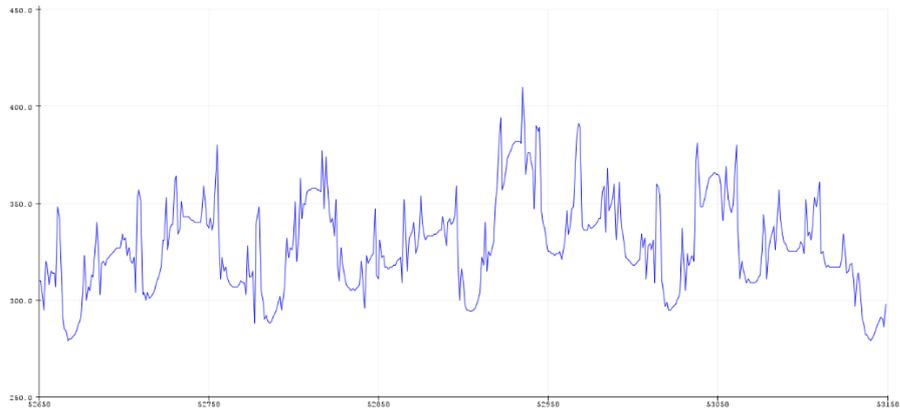


Figura 48: Señal resultante de la medición de ritmo cardiaco sin un intervalo entre cada muestra

A partir del resultado anterior, se amplió el tiempo de espera entre cada medición, y así de manera heurística se cambió este tiempo hasta obtener una señal semejante a otros equipos médicos. En la Figura 49 se indica la señal para un tiempo de 50ms entre cada medición, en esta señal se observa un cierto patrón en los picos.

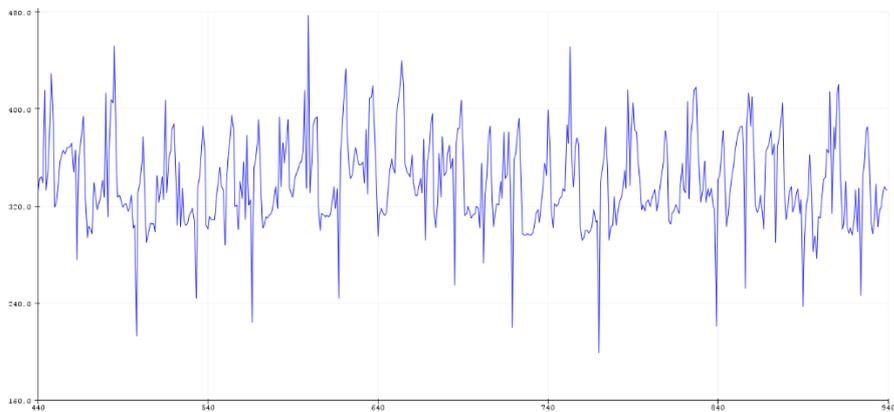


Figura 49: Señal resultante de la medición de ritmo cardiaco con un tiempo de 50ms entre cada muestra.

Dentro de este rango se realizó una evaluación de los valores, para lo cual se inició en 5ms, 1ms y 10ms, estas pruebas se muestran en la Figura 50a, Figura 50b y Figura 50c correspondientemente. Finalmente, se optó por un valor de

8ms como tiempo de muestreo, en el que se tiene una mejor forma en la señal resultante (Ver Figura 51).

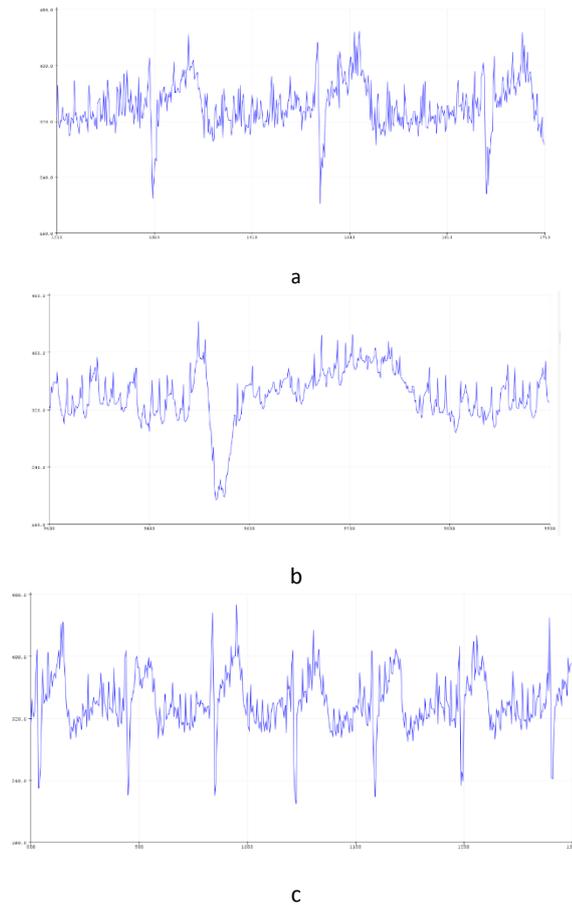


Figura 50: Señales resultante de la medición de ritmo cardíaco con un tiempo de muestreo de: a) 5ms b) 1ms y c) 10ms.

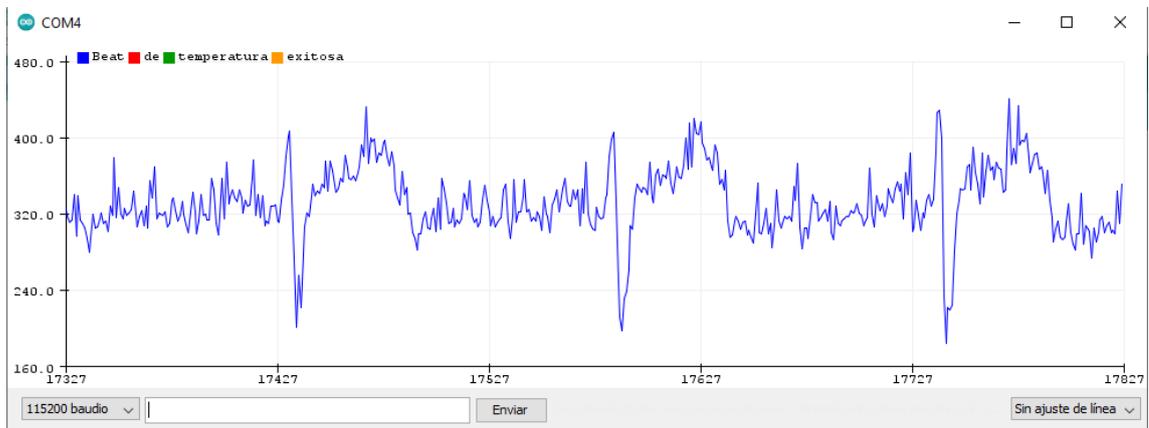


Figura 51: Señal resultante de la medición de ritmo cardíaco con un tiempo de 8ms entre cada muestra.

Con este valor se realizaron varias mediciones en diferentes pacientes, y los resultados fueron similares. Además, se observó una mejora en la visualización de la señal, al emplear menor cantidad de masa muscular en el área pectoral. La señal de un paciente con esta característica se observa en la Figura 52.

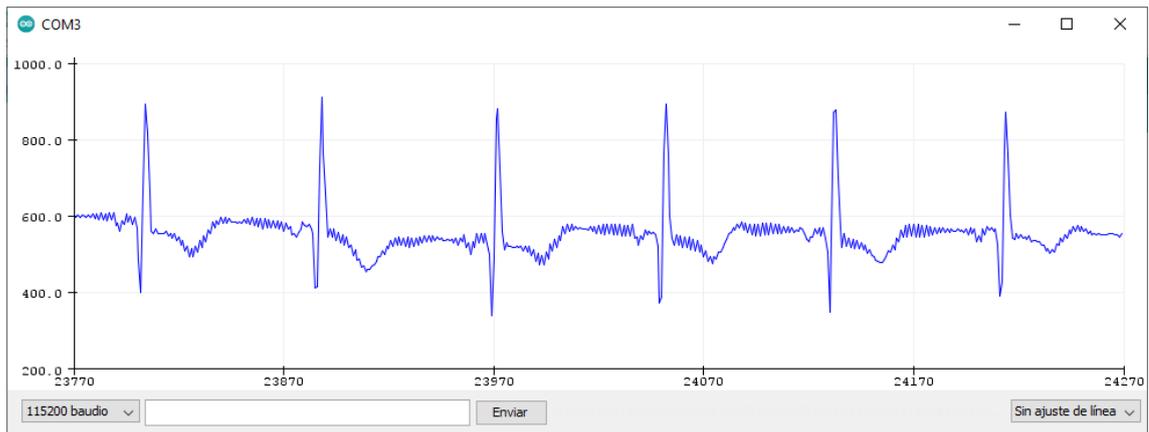


Figura 52: Señal resultante de la medición de ritmo cardíaco en un paciente de 6 años.

4.2. Base de datos

La base de datos utilizada fue MongoDB que, para esta ocasión, se creó dos tipos de colecciones, la primera es para los históricos o pacientes registrados y la segunda, para ser el nexo entre el dispositivo de adquisición de datos y la interfaz gráfica web o móvil.

La imagen muestra la interfaz de MongoDB Compass. En la parte superior izquierda, se indica 'MongoDB Compass - 52.43.113.195:38510/VariablesFisiologicas'. El menú de navegación incluye 'Connect', 'View' y 'Help'. A la izquierda, se muestra el estado 'Local' con '50 DBS' y '170 COLLECTIONS'. El panel principal muestra una tabla de colecciones:

Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size	Properties
registros	5	1.2 KB	6.2 KB	1	36.0 KB	[Icono]
variables	3,063	1.1 KB	3.2 MB	1	92.0 KB	[Icono]

Figura 53: Colecciones generados en la base de datos Variables Fisiológicas en MongoDB

Posteriormente, se expone un ejemplo de cada colección, el primero es un registro histórico de los pacientes guardados, y el siguiente son los últimos registros capturados por el prototipo electrónico. Cada colección muestra los documentos en formato JSON almacenados y desplegados por el software MongoDB.

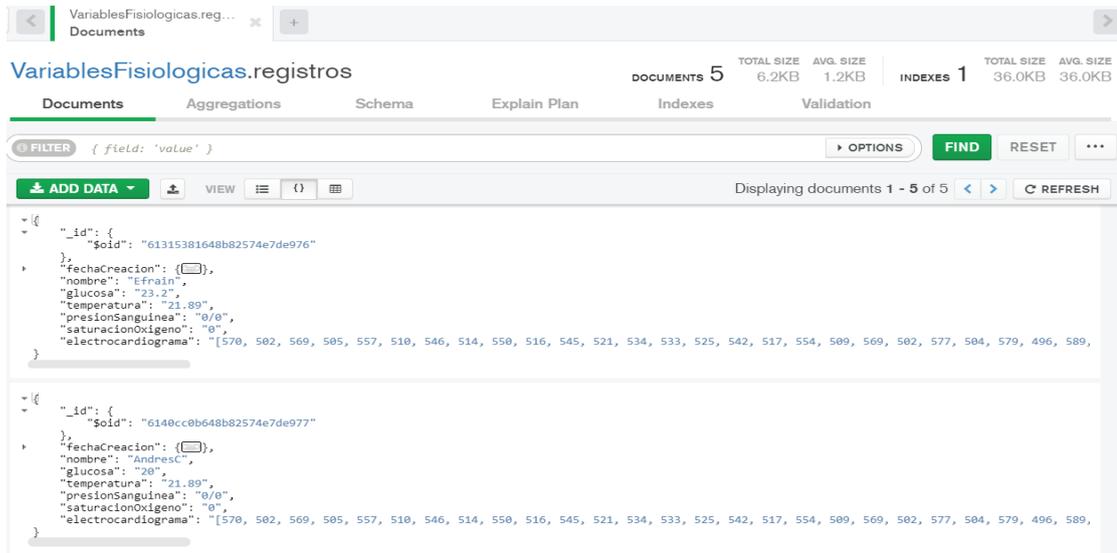


Figura 54: Colección de pacientes históricos guardados por la aplicación web

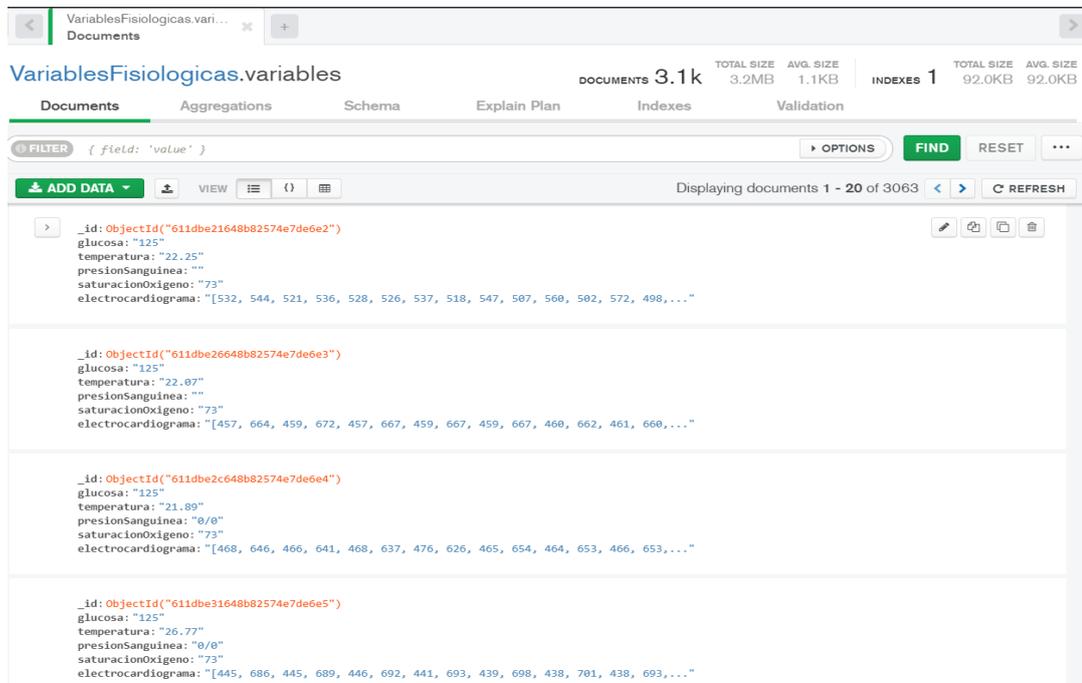


Figura 55: Colección que registra los datos obtenidos por el sistema de adquisición de variables.

4.3. Página web y aplicación móvil

A continuación, se presenta la página web y su equivalencia en su versión móvil con todas las secciones descritas anteriormente.

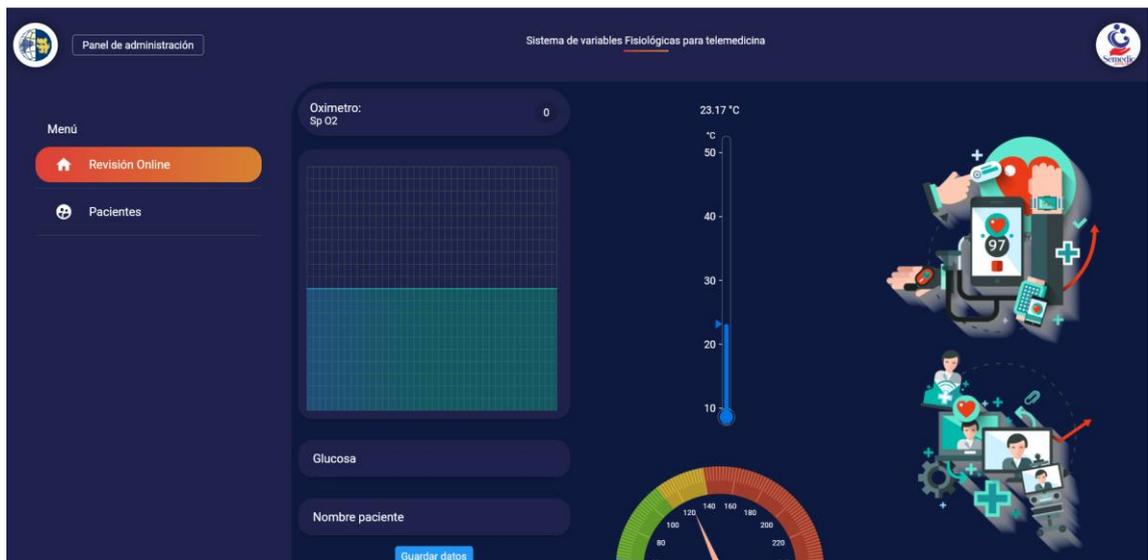


Figura 56: Página web implementada sección Revisión Online

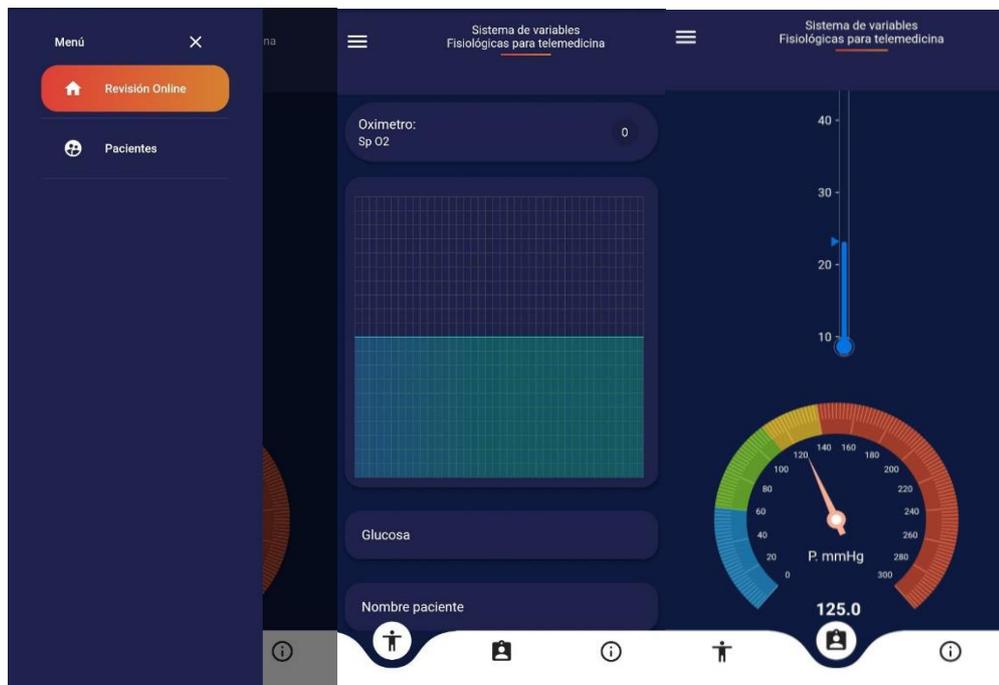


Figura 57: Aplicación móvil equivalente en web

En las Figuras anteriores, se observa la aplicación de los datos del sistema de adquisición de variables actuales, en donde se puede observar algunos indicadores, por ejemplo, para la temperatura un termómetro, para la presión un indicador con los niveles recomendados para un paciente, un monitor para el sensor ECG, indicadores tipo texto para el oxímetro y campos de entrada de texto para guardar información, como nombre y glucosa en conjunto con el botón de guardar. Además, permite la generación de un nuevo paciente en la base de datos histórico.

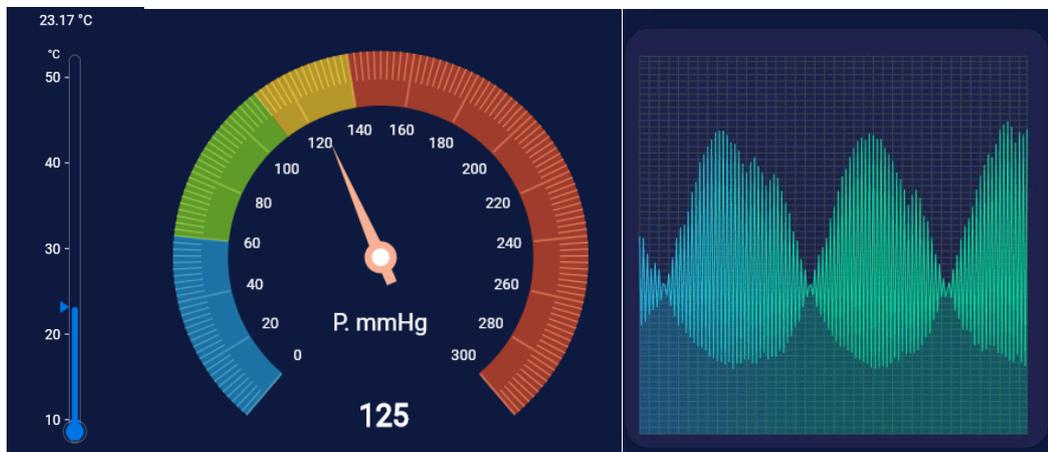


Figura 58: Diferentes elementos utilizados en la aplicación para mostrar

Igualmente, se presenta el panel correspondiente a la sección de pacientes guardados en la base de datos, desde esta pantalla se puede acceder a los datos almacenados del paciente.

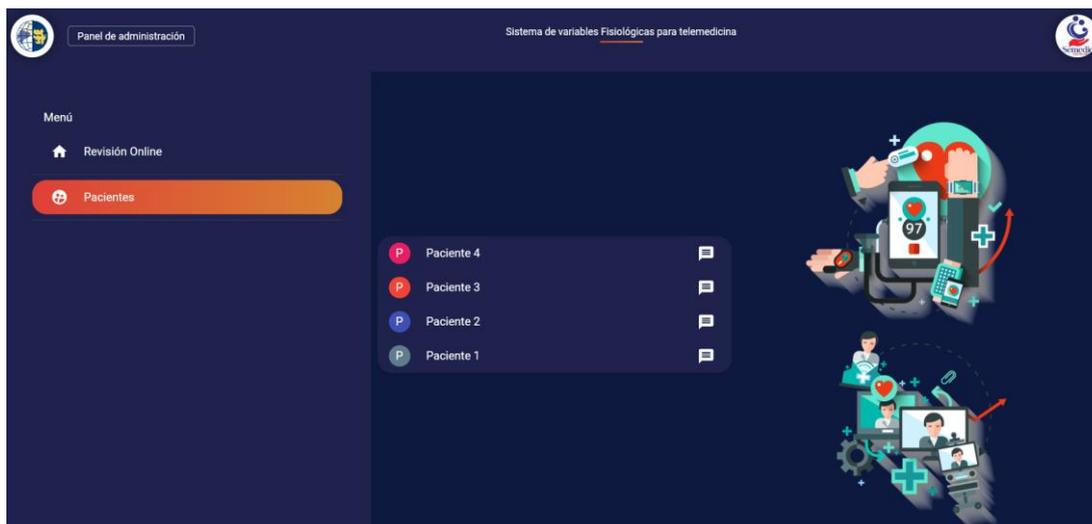


Figura 59: Panel de históricos ordenados por antigüedad en la versión web

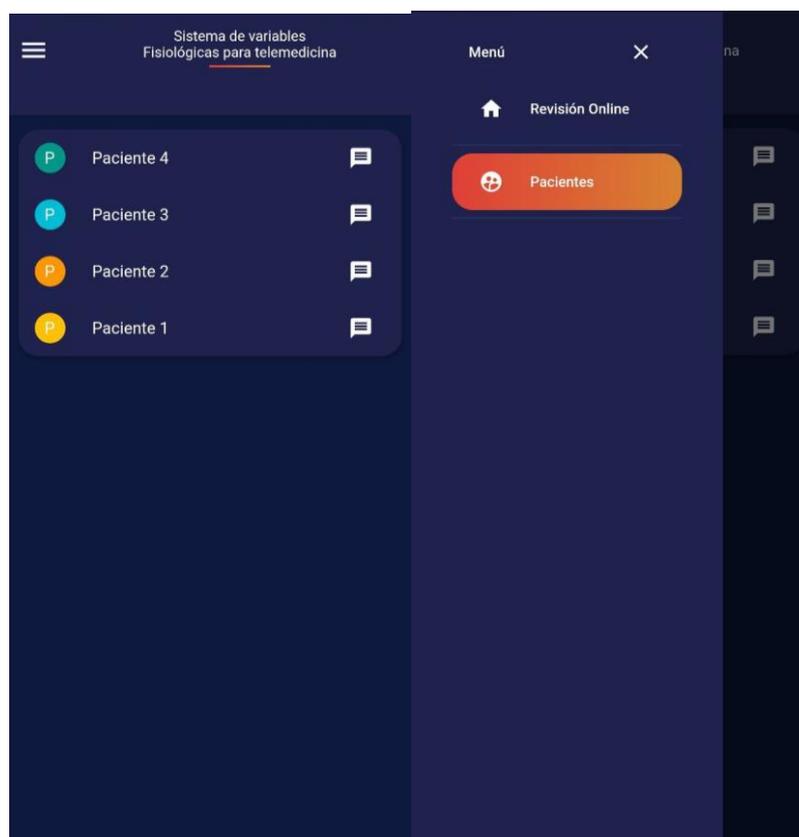


Figura 60: Panel de históricos ordenados por antigüedad equivalente en su versión móvil

Al dar clic sobre cualquiera de los pacientes aparece una pantalla similar a la revisión online, pero esta vez los datos están estáticos, es decir, no presenta campos configurables de texto, como se observa a continuación.



Figura 61: Paciente registrado en la base de datos

CONCLUSIONES

- Se desarrolló un sistema de adquisición de datos de las variables fisiológicas de humanos, a partir de un modelo comercial, en el que se implementó tanto hardware como software, mediante el IDE de Arduino y sensores compatibles con este.
- Se desarrolló la programación del módulo de comunicación inalámbrica, que establece la conexión entre servidor web y la base de datos, de esta forma se realiza el envío de datos que proveen los sensores del prototipo.
- Se diseñó e implementó una página web con dominio gratuito y una aplicación móvil para el sistema operativo Android, los cuales permitieron realizar la gestión de datos enviados por el dispositivo. Además, de permitir la interacción con el usuario final en este caso pacientes o médicos.
- Se implementó una estructura modular mediante impresión 3D, que facilita el transporte y conexión con los sensores, así como la configuración del sistema de adquisición de variables para la interacción medico paciente.

RECOMENDACIONES

- Tomar en cuenta los tiempos de retardo, para que no intervenga de manera negativa en la recepción de datos durante la comunicación UART.
- Debido a que el tensiómetro funciona con una fuente independiente, es importante asegurarse que esta tenga la carga suficiente para su correcto funcionamiento, de lo contrario es propenso a que no tenga la energía para el envío de datos durante la interrupción.
- Colocar apropiadamente los electrodos en el cuerpo del paciente ya que, según las pruebas realizadas, la masa corporal que tenga el paciente es un factor que afecta directamente a la toma de datos, es decir, en pacientes con menor masa corporal específicamente en el pecho, las señales son más fuertes y claras.
- Se aconseja a los usuarios que manejen el panel administrativo, tomen las siguientes medidas para su correcto funcionamiento, en primer lugar, el uso del tensiómetro, y luego continuar de manera paralela con los otros tres sensores (temperatura, oxímetro y ECG) de tal forma que estas medidas sean enviadas al servidor en conjunto.
- El servidor utilizado no es escalable, por lo que tiene un límite de clientes y al ser aplicaciones demandadas se recomienda mejorar el servidor utilizado, por uno de mayor capacidad que el implementado en este prototipo

REFERENCIAS BIBLIOGRÁFICAS

- Alvarado, D. (2005). *TELECIRUGÍA: UN ANÁLISIS TECNOLÓGICO DE LA TELECIRUGÍA LAPAROSCÓPICA* [UNIVERSIDAD DE SAN CARLOS DE GUATEMALA]. http://biblioteca.usac.edu.gt/tesis/08/08_0289_CS.pdf
- Amazon Web Services, I. (2021a). *¿Qué es AWS?*
<https://aws.amazon.com/es/what-is-aws/>
- Amazon Web Services, I. (2021b). *AWS | Almacenamiento de datos seguro en la nube (S3)*. <https://aws.amazon.com/es/s3/>
- ARDUINO.cl. (2021). *¿Qué es Arduino?*. <https://arduino.cl/que-es-arduino/>
- Banerjee, N. (2006). *Tension Arterial: Etiología Y Tratamiento* (Segunda Ed). B. Jain Publishers.
<https://books.google.com.sv/books?id=DpKmGEdHsvIC&printsec=frontcover&hl=es#v=onepage&q&f=false>
- De Souza. (2020, March 17). *API REST: ¿qué es y cómo funciona ese recurso? [con ejemplos]*. <https://rockcontent.com/es/blog/api-REST/>
- Del Ángel, F., & Escalera, S. (2017). Vista de Oxímetro de pulso con pletismografía por reflexión implementando el módulo MAX30100 mediante procesamiento digital con interfaz gráfica. *Sociedad Mexicana de Ingeniería Biomédica*, 4(1), 419–422.
<https://memorias.somib.org.mx/index.php/memorias/article/view/313/290>
- Del Valle, L. (2021). *Termómetro infrarrojo con Arduino MLX90614*.
<https://programarfacil.com/blog/arduino-blog/termometro-infrarrojo-con-arduino-mlx90614/>
- Eggs, B. (2002). *CHEMICAL SENSORS AND BIOSENSORS* (J. Wiley (ed.); Primera Ed). John Wiley & Sons Ltd. <https://download.e-bookshelf.de/download/0000/5741/24/L-G-0000574124-0002359078.pdf>
- Evans, D. (2011). The Internet of Things How the Next Evolution of the Internet Is Changing Everything. *Cisco IBSG*, 2–10.
https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/loT_IBSG_0411FINAL.pdf
- Flutter. (2021). *Beautiful native apps in record time | Flutter*. <https://flutter.dev/>
- García, J. (2004). Imágenes electrocardiográficas derivadas de una incorrecta

- colocación de los electrodos v1-v2. *Enfermería En Cardiología*, 3, 38–44.
<https://www.enfermeriaencardiologia.com/wp-content/uploads/3305.pdf>
- Gutiérrez, C. (2016). *Sistema de Monitoreo Continuo de Signos Vitales con Sensores No Invasivos y Transmisión Inalámbrica de Datos* [UNIVERSIDAD DE CONCEPCIÓN].
http://repositorio.udec.cl/bitstream/11594/1945/1/Tesis_Sistema_de_Monitoreo_Continuo_de_Signos_Vitales_con_Sensores.Image.Marked.pdf
- Hiraoka. (2020). ▷ *Tensiómetro: ¿qué es, para qué sirve y cómo funciona?*
<https://hiraoka.com.pe/blog/post/tensiometro-que-es-para-que-sirve-y-como-funciona>
- Kanani, P., & Padole, M. (2018). Recognizing real time ecg anomalies using arduino, ad8232 and java. *Communications in Computer and Information Science*, 905, 54–64. https://doi.org/10.1007/978-981-13-1810-8_6
- Labrada, E., & Salgado, C. (2013). Diseño Web Adaptativo o Responsivo. *Revista Digital Universitaria*, 14, 2–9.
<http://www.revista.unam.mx/vol.14/num1/art07/art07.pdf>
- Marini, E. (2012, October). *El Modelo Cliente/Servidor*. 5–6. <http://index-of.co.uk/REDES/linuxito - El Modelo Cliente-Servidor.pdf>
- Maxim Integrated. (2014, September). *MAX30100*. 10–30.
<https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf>
- Mayo Clinic. (2021, July 29). *Electrocardiograma (ECG) - Mayo Clinic*.
<https://www.mayoclinic.org/es-es/tests-procedures/ekg/about/pac-20384983>
- Mejía, H., & Mejía, M. (2012). Oximetría de pulso. *Sociedad Boliviana de Pediatría*, 51, 149–155. http://www.scielo.org.bo/pdf/rbp/v51n2/v51n2_a11.pdf
- Melexis. (2006). *MLX90614 family Single and Dual Zone Infra Red Thermometer in TO-39 Features and Benefits Ordering Information Part No. MLX90614 X X X (1) (2) (3) (1) Supply Voltage: A-5V power (adaptable for 12V) B-3V power (2) Number of thermopiles: A-single zone B-dua*.
- Miola, A. (2020). *Amazon.com: Flutter Complete Reference: Create beautiful, fast and native apps for any device* (F. Angelov & R. Rousselet (eds.); Primera Ed). Edición Kindle. <https://www.amazon.com/-/es/Alberto-Miola-ebook/dp/B08KHKK8TR>
- MongoDB, I. (2021). *La base de datos líder del mercado para aplicaciones*

- modernas | MongoDB*. <https://www.mongodb.com/es>
- Rodríguez, L. (2001). BASES DE DATOS DOCUMENTALES: ESTRUCTURA Y PRINCIPIOS DE USO. In *CINDOC-CSIC*.
http://biblioteca.udgvirtual.udg.mx/jspui/bitstream/123456789/3683/1/Base_datos_documental.pdf
- Rosa, J. (2018, May 17). *¿Qué es REST? Conoce su potencia | OpenWebinars*.
<https://openwebinars.net/blog/que-es-REST-conoce-su-potencia/>
- SANDORBOTICS. (2019). *Módulo WIFI ESP8266 ESP-01 | SANDORBOTICS*.
<https://sandorobotics.com/producto/hr0307/>
- SoloElectronicos. (2017, May 2). *Plataforma de sensores e-Health V1.0 para Arduino y Raspberry Pi [aplicaciones biométricas / medicina] – Soloelectronicos.com*. <https://soloelectronicos.com/2017/05/02/plataforma-de-sensores-e-health-v1-0-para-arduino-y-raspberry-pi-aplicaciones-biometricas-medicina/>
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80–83.
<https://doi.org/10.1109/MIC.2010.145>
- Zavala, A. (2017). *Revista Mexicana de Anestesiología Descripción del electrocardiograma normal y lectura del electrocardiograma*. 40, 210–213.
<http://www.medigraphic.com/rmawww.medigraphic.org.mxwww.medigraphic.org.mx>

ANEXOS

Anexo 1 Creación y configuración del servidor

Anexo 1.0 Creación de una instancia en EC2

En la consola de administración de AWS se selecciona el servicio EC2:

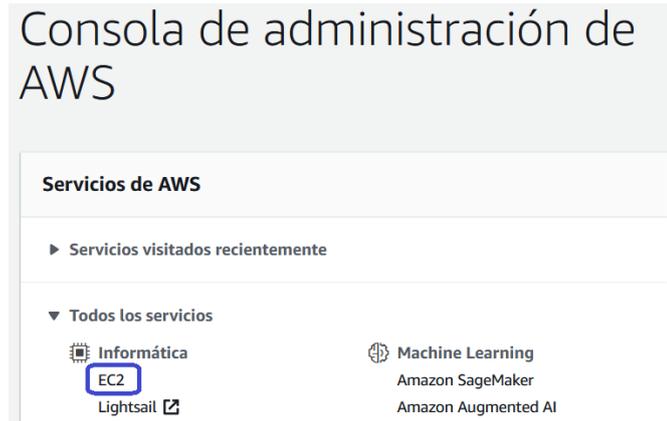


Figura 62: Cuenta de AWS consola de administración

Dentro de EC2 se elige la opción Instancias, luego de lo cual aparece la opción Lanzar instancias:

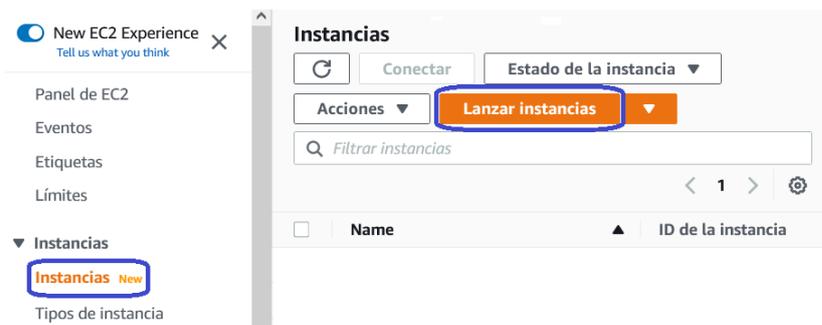


Figura 63: Sección EC2 configuración de instancias

Se elige la imagen (Sistema Operativo) a instalar en el servidor:

Paso 1: Elegir una imagen de Amazon Machine (AMI)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Puede seleccionar una AMI proporcionada por AWS, nuestra comunidad de usuarios o AWS Marketplace, o puede seleccionar una de sus propias AMI.



Figura 64: Selección del sistema operativo Ubuntu

Se escoge el procesador (número de núcleos y memoria RAM) con el que va a trabajar el servidor:

Paso 2: Página Choose an Instance Type

Amazon EC2 proporciona una amplia selección de tipos de instancias optimizadas para adaptarse a diferentes casos de uso. Las instancias son servidores virtuales que pueden ejecutar aplicaciones. Tienen distintas combinaciones de CPU, memoria, almacenamiento y capacidad de red, lo que proporciona una gran flexibilidad para elegir la combinación de recursos adecuada para las aplicaciones. [Más información](#) acerca de los tipos de instancias y cómo pueden satisfacer sus necesidades de computación.

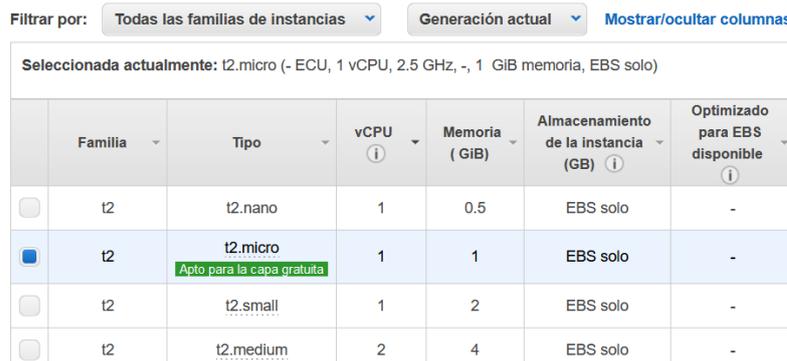


Figura 65: Selección de características del servidor virtual.

En el siguiente paso, es la configuración de los parámetros de red del servidor, entre los cuales se encuentran:

- VPC (Virtual Private Cloud): crea una LAN interna dentro de AWS.

- Subered: es el conjunto de direcciones pertenecientes a la subred dentro de la VPC.
- IP pública: activa una IP pública para acceder al servidor desde cualquier red.

1. Elija AMI 2. Elegir tipo de instancia 3. Configurar la instancia 4. Adición de almacenamiento

Paso 3: Página Configuración de los detalles de la instancia

Configure la instancia adecuada a sus requisitos. Puede lanzar varias instancias desde la misma AMI, solicitar instancias de spot para aprovecharse de los precios reducidos y asignar un rol de administración de acceso a la instancia, entre otras operaciones.

Número de instancias ⓘ

1 [Lanzar en grupo de Auto Scaling](#) ⓘ

Opción de compra ⓘ

Solicitar instancias de spot

Red ⓘ

vpc-1a66c67f (predeterminada) [Crear nueva VPC](#)

Subred ⓘ

Sin preferencia (subred predeterminada de cualquier) [Crear nueva subred](#)

Asignar automáticamente IP pública ⓘ

Usar configuración de subred (habilitar)

Figura 66: Configuraciones generales del servidor seleccionado

A continuación, se elige la opción de almacenamiento inicial del servidor, ya que cuando supere el límite indicado éste se incremente de manera automática:

1. Elija AMI 2. Elegir tipo de instancia 3. Configurar la instancia 4. Adición de almacenamiento

Paso 4: Adición de almacenamiento

Su instancia se lanzará con la siguiente configuración de dispositivo de almacenamiento. Puede asociar volúmenes de EBS y volúmenes del almacén de instancias adicionales a la instancia o editar la configuración del volumen raíz. También puede asociar volúmenes de EBS adicionales después de lanzar una instancia, pero no volúmenes del almacén de instancias. [Obtenga más información](#) acerca de las opciones de almacenamiento de Amazon EC2.

Tipo de volumen ⓘ	Dispositivo ⓘ	Snapshot ⓘ	Tamaño (GiB) ⓘ	Tipo de volumen ⓘ	IOPS ⓘ	Velocidad (MB/s) ⓘ
Raíz	/dev/sda1	snap-0f81f825ae3251a39	8	SSD de uso gener	100/3000	N/D

[Añadir nuevo volumen](#)

Figura 67: Selección de almacenamiento del computador virtual.

Se opta por el grupo de seguridad, el cual permite habilitar puertos dentro del servidor. En este paso se habilita SSH para acceder al servidor de manera remota, y el puerto de http o https para instalar un servidor web. Una vez creado el grupo de seguridad se lo guarda para futuras instancias EC2.

1. Elija AMI 2. Elegir tipo de instancia 3. Configurar la instancia 4. Adición de almacenamiento 5. Agregar etiquetas

Paso 6: Página Configure Security Group

Un grupo de seguridad es un conjunto de reglas del firewall que controlan el tráfico de la instancia. En esta página, puede agregar reglas para permitir que determinado tráfico llegue a la instancia. Por ejemplo, si desea configurar un servidor web y permitir que el tráfico de Internet llegue a la instancia, agregue reglas que permitan el acceso sin restricción a los puertos HTTP y HTTPS. Puede crear un nuevo grupo de seguridad o seleccionar uno existente a continuación. [Más información](#) sobre los grupos de seguridad de Amazon EC2.

Asignar un grupo de seguridad: Crear un nuevo grupo de seguridad
 Seleccionar un grupo de seguridad **existente**

Nombre del grupo de seguridad:

Descripción:

Tipo <small>i</small>	Protocolo <small>i</small>	Rango de puertos <small>i</small>	Origen <small>i</small>
SSH <small>v</small>	TCP	22	Personaliz. <small>v</small> 0.0.0.0/0
HTTP <small>v</small>	TCP	80	Cualquier I <small>v</small> 0.0.0.0/0, ::/0

Figura 68: Configuración de seguridades del servidor.

Se escoge la opción: Revisar y lanzar donde se muestra un resumen de todas las configuraciones que realizadas en el servidor.

1. Elija AMI 2. Elegir tipo de instancia 3. Configurar la instancia 4. Adición de almacenamiento 5. Agregar etiquetas

Paso 7: Página Review Instance Launch

▼ Detalles de la AMI [Editar AMI](#)

 **Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-03d5c68bab01f3496**

Apto para la capa Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).
 Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm

▼ Tipo de instancia [Editar tipo de instancia](#)

Tipo de instancia	ECU	vCPU	Memoria (GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	Desempeño de la red
t2.micro	-	1	1	EBS solo	-	Low to Moderate

Figura 69: Resumen del servidor seleccionado

Finalmente, se crea o se elige la llave de acceso al servidor. Esta llave se utiliza para la conexión SSH y es la única forma de acceder al servidor de manera remota.

Seleccione un par de claves existente o cree un nuevo par de claves

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Nota: El par de claves seleccionado se añadirá al conjunto de claves autorizadas para esta instancia. Obtenga más información sobre [cómo eliminar pares de claves existentes de una AMI pública](#).

Crear un nuevo par de claves

Nombre del par de claves

Clave_De_Acceso

Descargar par de claves

Tiene que descargar el **archivo de claves privadas** (archivo *.pem) para poder continuar. **Guárdelo en un lugar seguro y accesible**. No podrá descargar el archivo de nuevo después de crearlo.

Cancelar **Lanzar instancias**

Figura 70: Creación de claves de acceso remoto

Anexo 2 REST API desarrollados

Anexo 2.0 Estado del API

Permite obtener el estado del API.

Tipo	Descripción
Método	GET
URL	http://54.70.216.182/api-variablesFisiologicas/estado

Tabla 5 Datos generales del REST “estado”

Código	Respuesta
400	No entrega respuesta
200	{mensaje: "Controlador API funcionando correctamente"}

Tabla 6 Códigos de respuesta del REST “estado”

Anexo 2.1 Crear registro

API encargado de la creación de registros de un paciente.

Tipo	Descripción
Método	POST
URL	http://54.70.216.182/api-variablesFisiologicas/crearRegistro

Tabla 7 Datos generales del REST “crearRegistro”

Parámetros	Tipo	Observación	Descripción
nombre	string	Siempre se envía	Nombre del paciente
glucosa	string	Siempre se envía	Nivel de glucosa del paciente
temperatura	string	Siempre se envía	Temperatura corporal del paciente
presionSanguinea	string	Siempre se envía	Nivel de presión sanguínea del paciente
saturaciónOxigeno	string	Siempre se envía	Porcentaje de saturación del oxígeno
electrocardiograma	array	Siempre se envía	Valores del electrocardiograma

Tabla 8 Parámetros enviados del REST “crearRegistro”

body
<pre>{ "nombre": "Paciente 0", "glucosa": "120", "temperatura": "37", "presionSanguinea": "120/80", "saturacionOxigeno": "97", "electrocardiograma": [] }</pre>

Tabla 9 Ejemplo de envío del REST “crearRegistro”

Código	Respuesta
502	{mensaje: "Error al guardar el registro"}
200	{mensaje: "Registro exitoso"}

Tabla 10 Códigos de respuesta del REST “crearRegistro”

Anexo 2.2 Obtener todos los usuarios

API encargado de la obtener todos los registros de pacientes.

Tipo	Descripción
Método	GET
URL	http://54.70.216.182/api-variablesFisiologicas/obtenerUsuarios

Tabla 11 Datos generales del REST “obtenerUsuarios”

Código	Respuesta
502	{mensaje: "Error en la peticion"}
502	{mensaje: "No existen registros"}
200	Entrega todos los usuarios

Tabla 12 Códigos de respuesta del REST “obtenerUsuarios”

Anexo 2.3 Obtener el registro de un usuario

API encargado de la obtener el registro de un usuario en específico.

Tipo	Descripción
Método	GET
URL	http://54.70.216.182/api-variablesFisiologicas/obtenerRegistro

Tabla 13 Datos generales del REST “obtenerRegistro”

Parámetros	Tipo	Observación	Descripción
nombre	string	Siempre se envía	Nombre del paciente

Tabla 14 Parámetros enviados del REST “obtenerRegistro”

body
{ "nombre": "Paciente 0" }

Tabla 15 Ejemplo de envío del REST “obtenerRegistro”

Código	Respuesta
502	{mensaje: "Usuario incorrecto"}
502	{mensaje: "Error en la petición"}
502	{mensaje: "No existen registros"}
200	Entrega el registro de un usuario

Tabla 16 Códigos de respuesta del REST “obtenerRegistro”

Anexo 2.4 Crear Variable

API encargado de actualizar los valores de las mediciones

Tipo	Descripción
Método	POST
URL	http://54.70.216.182/api-variablesFisiologicas/crearVariables

Tabla 17 Datos generales del REST “crearVariables”

Parámetros	Tipo	Observación	Descripción
glucosa	string	Siempre se envía	Nivel de glucosa del paciente
temperatura	string	Siempre se envía	Temperatura corporal del paciente
presionSanguinea	string	Siempre se envía	Nivel de presión sanguínea del paciente
saturacionOxigeno	string	Siempre se envía	Porcentaje de saturación del oxígeno
electrocardiograma	array	Siempre se envía	Valores del electrocardiograma

Tabla 18 Parámetros enviados del REST “crearVariables”

body
<pre>{ "glucosa": "120", "temperatura": "37", "presionSanguinea": "120/80", "saturacionOxigeno": "97", "electrocardiograma": [] }</pre>

Tabla 19 Ejemplo de envío del REST “crearVariable”

Código	Respuesta
502	{mensaje: "Error al guardar el registro"}
200	{mensaje: "Registro exitoso"}

Tabla 20 Códigos de respuesta del REST “crearVariables”

Anexo 2.5 Obtener variables

API encargado de la obtener las ultimas variables registradas en la base de datos.

Tipo	Descripción
Método	GET
URL	http://54.70.216.182/api-variablesFisiologicas/obtenerVariables

Tabla 21 Datos generales del REST “obtenerVariables”

Código	Respuesta
502	{mensaje: "Error en la peticion"}
502	{mensaje: "No existen registros"}
200	Entrega el ultimo registro de las variables en la base de datos

Tabla 22 Códigos de respuesta del REST “obtenerVariables”