



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE INGENIERÍA ELECTRÓNICA**

**“IMPLEMENTACIÓN DE UN PROTOTIPO PARA REALIZAR EL MONITOREO DE
SIGNOS VITALES Y VISUALIZARLO EN UN DASHBOARD DE NODE-RED
UTILIZANDO UN PIC18F4550 Y RASPBERRY PI ”**

Trabajo de titulación previo a la obtención del
Título de **Ingeniero Electrónico**

AUTOR:

JULIO ANDRÉS CAGUA CHIQUITO

TUTOR:

**MSC. VICENTE AVELINO PEÑARANDA IDROVO
GUAYAQUIL – ECUADOR
2023**

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo **Julio Andrés Cagua Chiquito** con cédula de identificación N° **0926575697** manifiesto que:

Soy el autor y responsable del presente trabajo; 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, 19 de febrero del año 2024.



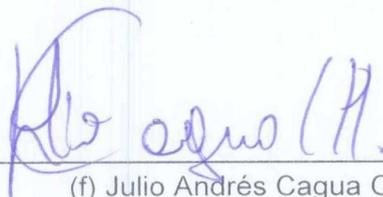
(f) Julio Andrés Cagua Chiquito
C.I: 0926575697

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

Yo **Julio Andrés Cagua Chiquito** con cédula de identificación N° **0926575697**, 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 el autor del **Proyecto técnico: “IMPLEMENTACIÓN DE UN PROTOTIPO PARA REALIZAR EL MONITOREO DE SIGNOS VITALES Y VISUALIZARLO EN UN DASHBOARD DE NODE-RED UTILIZANDO UN PIC18F4550 Y RASPBERRY PI”**, 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, 19 de febrero del año 2024.



(f) Julio Andrés Cagua Chiquito

C.I: **0926575697**

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo Vicente Avelino Peñaranda Idrovo con documento de identificación N° 0916113426, docente de la **Universidad Politécnica Salesiana**, declaro que bajo mi autoría fue desarrollado el trabajo de titulación: **"IMPLEMENTACIÓN DE UN PROTOTIPO PARA REALIZAR EL MONITOREO DE SIGNOS VITALES Y VISUALIZARLO EN UN DASHBOARD DE NODE-RED UTILIZANDO UN PIC18F4550 Y RASPBERRY PI "**, realizado por **Julio Andrés Cagua Chiquito** con cédula de identificación N° **0926575697** obteniendo como resultado final el trabajo de titulación bajo la opción **Artículo Académico**, que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 19 de febrero del año 2024.

Atentamente,



Ing. Vicente Avelino Peñaranda Idrovo, MSC

C.I: 0916113426

DEDICATORIA

Le dedico este proyecto primero a mis hijos, Mateo y Alejandro, quienes representan mi mayor fuente de inspiración y la razón por la cual me esfuerzo cada día para ser un mejor ser humano. Quiero ser un ejemplo positivo en sus vidas y su constante motivación me impulsa a superarme continuamente. Además, esta dedicación no solo se trata de mi deseo de ser un buen modelo para ellos, sino también de mi compromiso de prepararme de manera constante para brindarles la mejor educación posible.

Quisiera también dedicarle este proyecto a mis padres, Nelly y Andrés, quienes, a pesar de enfrentar diversas dificultades a lo largo de sus vidas, han sido una fuente constante de inspiración y guía. Esta dedicación es un reconocimiento a su amor incondicional y a la resiliencia que demostraron ante las adversidades. A través de su ejemplo, aprendí la importancia de la perseverancia y la fortaleza. Su legado es una parte fundamental de este logro, y quiero expresar mi profundo agradecimiento por haberme brindado las bases sólidas que me han permitido llegar hasta aquí.

Cagua Chiquito J.

AGRADECIMIENTO

Quiero expresar mi profundo agradecimiento a mis padres, quienes han sido un pilar inquebrantable no solo en mi trayectoria educativa sino también en mi vida diaria. Agradezco mucho a toda mi familia, en especial a mis hijos, quienes son mi alegría constante.

Asimismo, quiero extender mi gratitud a cada persona que formó parte de las instituciones donde me eduqué, desde el personal de limpieza y los guardias hasta los administrativos y los profesores. Cada uno de ustedes desempeñó un papel fundamental en mi formación académica, y les estoy muy agradecido.

Un reconocimiento especial va para los creadores de contenido educativo en internet. Gracias a muchos de ustedes, pude comprender conceptos complejos de manera didáctica y accesible.

Finalmente, quiero expresar mi agradecimiento más sincero a la comunidad "open source" en internet. Su dedicación a compartir conocimiento, basándose en principios de libertad, igualdad, solidaridad y apoyo mutuo, ha transformado la experiencia en la red. Sin ellos, el internet no sería lo que es hoy en día. Gracias a todos por su invaluable contribución a mi formación y desarrollo.

Cagua Chiquito J.

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO	TEMA DE PROYECTO DE TITULACIÓN
2023	CAGUA CHIQUITO JULIO ANDRÉS	MSC. VICENTE AVELINO PEÑARANDA HIDROVO	"IMPLEMENTACIÓN DE UN PROTOTIPO PARA REALIZAR EL MONITOREO DE SIGNOS VITALES Y VISUALIZARLO EN UN DASHBOARD DE NODE-RED UTILIZANDO UN PIC18F4550 Y RASPBERRY PI"

El proyecto implica crear un prototipo de monitorización de signos vitales para seguir la temperatura corporal, los niveles de oxígeno en la sangre (oximetría) y la presión arterial. Estos parámetros son presentados en un panel de control desarrollado en Node-RED, el cual puede ser accedido desde cualquier dispositivo con conexión a internet, sin importar la ubicación geográfica. Es importante destacar que para poder acceder al dashboard del prototipo, se requiere que este esté conectado a internet a través de una conexión alámbrica (ethernet).

En este prototipo, se emplearon diversos componentes para realizar las mediciones. Para la medición de la temperatura, se utilizó una termocupla. Para llevar a cabo la oximetría, se empleó una sonda comercial compatible con varios modelos de pulsioxímetros en el mercado, destacando su especial compatibilidad con el modelo Nellcor DS100A. La elección de esta sonda se fundamenta en su reconocimiento en el mercado y su reputación por ofrecer mediciones de alta calidad y precisión. En lo que respecta a la presión arterial, se empleó un brazalete de la marca CK, en el cual se conecta a su memoria EEPROM para obtener los datos de presión diastólica, sistólica y la frecuencia de pulsos por minuto.

Para llevar a cabo el procesamiento de las distintas señales, se emplearon los microcontroladores PIC 18F4550 y Arduino nano. La termocupla se encuentra conectada a una placa amplificadora de señal que trabaja con el protocolo de comunicación SPI. En relación a la sonda de oximetría, se diseñó un circuito encargado de procesar la señal emitida, el cual cumple con funciones de amplificación y filtrado para mejorar la calidad de la señal capturada.

En el caso del brazalete de presión arterial, el Arduino nano establece comunicación con la memoria EEPROM a través del protocolo de comunicación I2C. Para la transmisión de todos estos datos al panel de control de Node-RED, el PIC y el Arduino nano se conecta de manera serial con una Raspberry Pi, que cumple la función de servidor. La Raspberry Pi, a su vez, se conecta a internet para compartir el panel de control en la red global. Esta comunicación entre el Raspberry Pi y el PIC se facilita mediante el uso de una tarjeta interfaz serial TTL a USB.

Este proyecto encuentra su pertinencia en la capacidad de realizar un seguimiento exhaustivo de pacientes que requieran aislamiento debido a diversas circunstancias, tal como se evidenció durante la pandemia de COVID-19. Además, esta solución es adecuada para la implementación de telemedicina, permitiendo al personal médico monitorizar los signos vitales en tiempo real desde cualquier parte del planeta.

Palabras Claves: PIC 18F4550, Rapsberry PI, Microprocesador, Node-RED, IOT, Procesamiento de Señales, Monitor de Signos Vitales, Telemedicina.

ABSTRACT

YEAR	STUDENTS	PRJ. DIRECTOR	SUBJECT
2023	CAGUA CHIQUITO JULIO ANDRÉS	MSC. VICENTE AVELINO PEÑARANDA HIDROVO	"IMPLEMENTATION OF A PROTOTYPE FOR VITAL SIGNS MONITORING AND VISUALIZATION IN A NODE-RED DASHBOARD USING A PIC18F4550 AND RASPBERRY PI"

The project involves creating a prototype for monitoring vital signs to track body temperature, blood oxygen levels (oximetry), and blood pressure. These parameters are displayed on a control panel developed in Node-RED, which can be accessed from any internet-connected device, regardless of geographical location. It is important to note that to access the prototype dashboard, it is required to be connected to the internet via a wired connection (ethernet).

Various components were used in this prototype to perform the measurements. A thermocouple was used for temperature measurement. For oximetry, a commercial probe compatible with various models of pulse oximeters on the market was used, with special compatibility with the Nellcor DS100A model. The choice of this probe is based on its recognition in the market and reputation for providing high-quality and accurate measurements. As for blood pressure, a CK brand cuff was used, which connects to its EEPROM memory to obtain diastolic and systolic pressure data and pulse frequency per minute.

Microcontrollers PIC 18F4550 and Arduino nano were used for processing the different signals. The thermocouple is connected to a signal amplification board that works with the SPI communication protocol. For the oximetry probe, a circuit was designed to process the emitted signal, which performs amplification and filtering functions to improve the quality of the captured signal.

In the case of the blood pressure cuff, the Arduino nano communicates with the EEPROM memory through the I2C communication protocol. To transmit all this data to the Node-RED control panel, the PIC and Arduino nano are serially connected to a Raspberry Pi, which acts as a server. The Raspberry Pi, in turn, connects to the internet to share the control panel on the global network. This communication between the Raspberry Pi and the PIC is facilitated

by using a TTL serial interface card to USB.

This project is relevant for comprehensive monitoring of patients requiring isolation due to various circumstances, as evidenced during the COVID-19 pandemic. Additionally, this solution is suitable for telemedicine implementation, allowing medical personnel to monitor vital signs in real-time from anywhere in the world.

Keywords: PIC 18F4550, Raspberry Pi, Microprocessor, Node-RED, IoT, Signal Processing, Vital Signs Monitor, Telemedicine.

ABREVIATURAS

GUI: Graphical User Interface.

PIC: Periferal Interface Controller.

INDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN.....	2
CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA.....	3
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	4
DEDICATORIA.....	5
AGRADECIMIENTO.....	6
RESUMEN.....	7
ABSTRACT.....	9
INDICE GENERAL.....	12
INDICE DE FIGURAS.....	15
INDICE DE TABLAS.....	18
INTRODUCCIÓN.....	19
1. EL PROBLEMA.....	20
1.1. Descripción del problema.....	20
1.2. Antecedentes.....	20
1.3. Importancia y alcances.....	21
1.4. Delimitación del problema.....	22
1.4.1. Temporal.....	22
1.4.2. Espacial.....	22
1.4.3. Académica.....	23
1.5. Objetivos.....	23
1.5.1. Objetivo General.....	23
1.5.2. Objetivo Específicos.....	23
2. MARCO TEÓRICO REFERENCIAL.....	24
2.1. Microprocesadores y Microcontroladores: Caracterización.....	24
2.1.1. Microcontroladores PIC.....	24
2.1.2. Microcontroladores de gama alta.....	25
2.1.3. Interrupciones en Microcontroladores PIC.....	25
2.1.4. Introducción a la transmisión de datos en serie.....	26
2.1.5. El Bus I2C.....	26
2.1.6. El puerto serial USART en los microcontroladores PIC.....	27
2.1.7. El convertor de 10-Bits A/D en los microcontroladores PIC.....	27
2.2. PIC18F4550.....	28
2.2.1. Especificaciones.....	28
2.2.2. PINOUT.....	29
2.2.3. Diagrama de bloques.....	30

2.2.4. Interrupciones.....	31
2.2.5. Puertos de Entradas y Salidas.....	31
2.3. Raspberry Pi.....	32
2.3.1. Especificaciones.....	32
2.3.2. Sistema Operativo Raspbian.....	34
2.3.3. Pines GPIO.....	34
2.4. Arduino nano.....	34
2.4.1. Características.....	35
2.4.1. PINOUT.....	37
2.4.2. Diagrama de Bloques.....	37
2.5. Node-RED.....	38
2.5.1. Generalidades.....	38
2.5.2. El Editor de Flujos y el Entorno de Ejecución.....	38
2.5.3. Node-RED junto a IOT.....	39
2.6. Oximetría de Pulso.....	41
2.6.1. Transporte Normal de Oxígeno.....	41
2.6.2. Principios del Oxímetro de Pulso.....	42
2.6.3. SpO2 como Monitor de Oxigenación de la Hemoglobina.....	42
2.6.4. Saturación de oxígeno arterial.....	43
2.6.5. Hipoxia e Hipoxemia.....	43
2.6.6. Hiperoxia.....	44
2.6.7. Limitaciones.....	44
2.6.8. Funcionamiento de un Oxímetro de Pulso.....	46
2.7. Presión Arterial.....	50
2.7.1. Presión arterial vs. frecuencia cardíaca.....	50
2.7.2. Por qué la presión arterial fluctúa.....	51
2.7.3. Importancia de la medición de presión arterial.....	51
2.8. Temperatura Corporal.....	52
2.8.1. Termopares.....	52
2.8.2. Principios Generales de Operación.....	53
2.8.3. Alteraciones en la Temperatura Corporal: Hipertermia e Hipotermia.....	53
3. REVISIÓN DE LA LITERATURA.....	55
3.1. Estado del Arte.....	55
3.1.1. Placas de Desarrollo y Microcontroladores.....	55
3.1.2. Diseño de Tarjetas PCB.....	55
3.1.3. Programación y Lenguajes de Programación.....	55
3.1.4. Prototipo para el Monitoreo de Signos Vitales.....	56
3.2. Revisión de Alcance.....	56

3.2.1. Delimitación del Tema.....	56
3.2.2. Alcance de los Signos Vitales.....	56
3.2.3. Exclusión del Alcance.....	56
3.2.4. Limitaciones y Desafíos.....	57
3.2.5. Importancia del Alcance.....	57
4. METODOLOGÍA.....	58
4.1. Investigación Inicial.....	58
4.2. Diseño y Desarrollo.....	58
4.3. Métodos y técnicas de Recopilación de datos empleadas.....	74
5. INTERPRETACIÓN DE RESULTADOS.....	76
6. CONCLUSIONES.....	92
7. RECOMENDACIONES.....	93
8. REFERENCIAS BIBLIOGRÁFICAS.....	94
9. ANEXOS.....	96

INDICE DE FIGURAS

Figura 1. Diagrama de bloques genéricos de una microcomputadora.....	24
Figura 2. Esquema de una interrupción en un Microcontrolador.....	26
Figura 3. Formula de velocidad de transmisión de datos.....	26
Figura 4. Varios dispositivos conectados en el bus I2C.....	27
Figura 5. Bloque funcional de convertidor A/D.....	28
Figura 6. PIC18F4550-PINOUT.....	29
Figura 7. PIC18F4550-Diagrama de bloques.....	30
Figura 8. PIC18F4550-Puertos I/O.....	32
Figura 9. Raspberry Pi 3.....	33
Figura 10. Rasperry PI-GPIO.....	34
Figura 11: Arduino nano Pinout.....	37
Figura 12: Arduino nano - Diagrama de bloques.....	37
Figura 13. El Editor de Flujos de Node-RED como una herramienta FBP.....	38
Figura 14. Las 6 capas de IOT.....	40
Figura 15. Diagrama de Bloques de un sistema de Pulso Oxímetro.....	47
Figura 16. Ilustración del funcionamiento de la sonda del Pulso Oxímetro.....	49
Figura 17. Rango de mediciones de Presión Arterial.....	51
Figura 18: Diagrama Esquemático Tarjeta Interfaz con el PIC.....	59
Figura 19: Diseño de la PCB de la Tarjeta Interfaz con el PIC.....	59
Figura 20: Diagrama Esquemático de la Tarjeta Interfaz con el sensor de Oximetría.....	61
Figura 21: Diseño de la PCB de la Tarjeta Interfaz con el sensor de Oximetría.....	62
Figura 22: Imagen Frontal del Diseño en 3D del Prototipo.....	62

Figura 23: Imagen Lateral del Diseño en 3D del Prototipo.....	63
Figura 24: Imagen Frontal sin tapa del Diseño en 3D del Prototipo.....	64
Figura 25: Imagen Lateral sin tapa del Diseño en 3D del Prototipo.....	64
Figura 26: Imagen Trasera sin tapa del Diseño en 3D del Prototipo.....	65
Figura 27: Imagen Superior sin tapa del Diseño en 3D del Prototipo.....	66
Figura 28: Tarjeta Interfaz con el PIC (terminada).....	67
Figura 29: Tarjeta Interfaz con el Sensor de Oximetría (terminada).....	67
Figura 30: Proceso de Soldadura para la Tarjeta Interfaz con el PIC.....	68
Figura 31: Proceso de soldadura para la Tarjeta Interfaz con el Sensor de Oximetría.....	68
Figura 32: Montaje de las Tarjetas Electrónicas.....	69
Figura 33: Tarjetas Electrónicas colocadas en su sitio.....	69
Figura 34: Imagen final del prototipo.....	70
Figura 35: Maleta con el prototipo.....	71
Figura 36: Dashboard en Node-RED.....	71
Figura 37: Prueba de Temperatura.....	78
Figura 38: Medicion de Temperatura.....	78
Figura 39: Medicion de Temperatura.....	79
Figura 40: Código para el calculo del error en la medición de la Temperatura.....	80
Figura 41: Comparación de Temperaturas.....	80
Figura 42: Error entre las mediciones de Temperatura.....	81
Figura 43: Medición de Saturación.....	82
Figura 44: Medición de Saturación.....	83
Figura 45: Medición de Saturación.....	83

Figura 46: Medición de Saturación.....	84
Figura 47: Medición de Saturación.....	84
Figura 48: Código para el calculo del error en la medición de la Temperatura.....	85
Figura 49: Comparación de Saturación de Oxígeno.....	85
Figura 50: Error Absoluto de Saturación de Oxígeno.....	86
Figura 51: Medición de Presión Arterial.....	87
Figura 52: Medición de Presión Arterial.....	88
Figura 53: Medición de Presión Arterial.....	88
Figura 54: Medición de Presión Arterial.....	88
Figura 55: Medición de Presión Arterial.....	89
Figura 56: Medición de Presión Arterial.....	89
Figura 57: Código para el calculo del error en la medición de la Temperatura.....	90
Figura 58: Comparación de Presión Arterial.....	90
Figura 59: Error en la medición de Presión Arterial.....	91

INDICE DE TABLAS

Tabla 1: Tiempos de Anoxia.....	44
Tabla 2: Limitaciones de la Oximetría de Pulso.....	46
Tabla 3: Mediciones de Temperatura.....	77
Tabla 4: Mediciones de Saturación de Oxígeno.....	82
Tabla 5: Mediciones de Presión Arterial.....	87

INTRODUCCIÓN

En respuesta a los desafíos planteados durante la pandemia global de COVID-19, surgió la necesidad apremiante de soluciones avanzadas para el monitoreo de la salud y la telemedicina. Aunque la crisis sanitaria ha quedado atrás, la importancia de contar con sistemas eficientes de monitoreo de signos vitales persiste. Este proyecto, concebido en medio de la pandemia, se propuso abordar esa necesidad mediante el desarrollo de un sistema prototipo de monitoreo de signos vitales. Diseñado para ofrecer un seguimiento en tiempo real de parámetros fisiológicos críticos, como la temperatura corporal, los niveles de oxígeno en la sangre y la presión arterial, el sistema aprovecha tecnologías de vanguardia para facilitar el monitoreo remoto de pacientes y permitir intervenciones médicas eficaces.

Este proyecto se enfoca en desarrollar un prototipo completo que combina diversas tecnologías y componentes para el monitoreo de signos vitales. Utiliza un microcontrolador PIC 18F4550 y dos tarjetas Arduino Nano junto con una Raspberry Pi como servidor para procesar y transmitir datos. Con técnicas avanzadas de procesamiento de señales, se busca mejorar la precisión y confiabilidad de la información recopilada. La inclusión de Node-RED como plataforma de panel de control permite a los profesionales de la salud monitorear los signos vitales de los pacientes desde cualquier dispositivo conectado a Internet.

Este proyecto aborda la necesidad de protocolos de aislamiento en entornos médicos, especialmente en contextos similares a los vividos durante la pandemia de COVID-19. Se destaca la capacidad única del prototipo para facilitar el monitoreo remoto, alineándose con los principios de la telemedicina y permitiendo intervenciones oportunas sin la necesidad de presencia física.

Las secciones siguientes detallan el diseño, desarrollo e implementación del prototipo de monitoreo de signos vitales. Analizan la metodología del proyecto, los detalles técnicos y los resultados clave, considerando las posibles implicaciones para el panorama general de la atención médica. Al converger los campos de la tecnología médica, el procesamiento de señales y la telemedicina, este proyecto representa un avance significativo y contribuye a la evolución continua de las prácticas de atención médica en un mundo cada vez más interconectado.

1. EL PROBLEMA

1.1. Descripción del problema

La emergencia global causada por la pandemia de COVID-19 expuso la necesidad crítica de soluciones de monitoreo médico innovadoras que puedan adaptarse rápidamente a situaciones de aislamiento y a la creciente adopción de prácticas de telemedicina. Durante la pandemia, la limitación de los contactos físicos se convirtió en un imperativo para prevenir la propagación del virus, lo que a su vez aumentó la demanda de sistemas de monitoreo remoto capaces de mantener una vigilancia constante sobre los pacientes.

En este contexto, surge el desafío de diseñar y desarrollar un sistema de monitoreo de signos vitales altamente confiable y accesible. La necesidad de realizar un seguimiento continuo y preciso de la temperatura corporal, los niveles de oxígeno en la sangre y la presión arterial se convierte en una prioridad tanto en situaciones de aislamiento médico como en la evolución constante de las prácticas de atención médica.

Los métodos tradicionales de monitoreo a menudo requieren la presencia física de profesionales de la salud en la ubicación del paciente, lo que no solo es un riesgo potencial en escenarios de contagio, sino que también limita la capacidad de proporcionar atención inmediata en situaciones críticas. Asimismo, la adopción acelerada de la telemedicina ha demostrado la necesidad de sistemas de monitoreo versátiles accesibles y manejados a distancia.

Este proyecto se propone abordar estas cuestiones fundamentales al crear un prototipo de sistema de monitoreo de signos vitales que integra tecnologías de vanguardia para lograr una monitorización precisa y continua. Al brindar a los profesionales de la salud la capacidad de monitorear a los pacientes en tiempo real y a distancia, este sistema se posiciona como una respuesta viable y eficaz a los desafíos planteados por la pandemia y las demandas cambiantes de la atención médica moderna. (Cisneros Caicedo et al., 2022)

1.2. Antecedentes

Las pandemias a lo largo de la historia han ejercido un profundo impacto en la salud pública y han catalizado la búsqueda de soluciones innovadoras para el monitoreo y manejo de enfermedades. Ejemplos notables incluyen la pandemia de gripe española de 1918, la pandemia de gripe aviar H5N1 en la década de 2000, y la pandemia de gripe porcina H1N1 en 2009. Estos eventos han destacado la necesidad de contar con sistemas eficientes de

monitoreo y control que puedan adaptarse rápidamente a las dinámicas cambiantes de la propagación de enfermedades.

En particular, la pandemia de gripe porcina H1N1 enfatizó la importancia del monitoreo en tiempo real de los signos vitales de los pacientes para permitir una intervención temprana y un manejo adecuado. Durante esta pandemia, los sistemas tradicionales de monitoreo en los hospitales se encontraron limitados debido a la alta demanda y a la necesidad de mantener un entorno de bajo riesgo de contagio. Esta situación impulsó la exploración de soluciones de monitoreo remoto que permitieran a los médicos supervisar la evolución de la enfermedad sin estar físicamente presentes.

En respuesta a estas demandas, diversas soluciones de monitoreo de signos vitales han surgido a lo largo de los años. Los avances en la tecnología han permitido el desarrollo de dispositivos portátiles y no invasivos que pueden medir la temperatura corporal, la frecuencia cardíaca y los niveles de oxígeno en la sangre. Además, la incorporación del Internet de las Cosas (IoT) ha posibilitado la transmisión en tiempo real de estos datos a través de plataformas de control accesibles desde múltiples ubicaciones.

Sin embargo, a pesar de estos avances, la pandemia de COVID-19 reveló la necesidad continua de sistemas de monitoreo de signos vitales aún más flexibles y eficaces. La rápida propagación del virus y las restricciones de movilidad subrayaron la importancia de contar con soluciones que permitieran la monitorización remota y constante de pacientes en cuarentena o aislamiento.

En este contexto, el presente proyecto se propone abordar los desafíos identificados por pandemias anteriores y las limitaciones de las soluciones existentes. A través de la integración de tecnologías de vanguardia, como microcontroladores, IoT y plataformas de control en línea, se busca crear un sistema de monitoreo de signos vitales que satisfaga las demandas de la atención médica moderna y las situaciones de emergencia como la que se experimentó con el COVID-19. (Fojo, 2020)

1.3. Importancia y alcances

La importancia de este proyecto se deriva de la convergencia de varios factores cruciales en el panorama actual de la atención médica. La pandemia de COVID-19 reveló las deficiencias en los sistemas de monitoreo y atención médica tradicionales, así como la necesidad de adoptar enfoques innovadores para garantizar un monitoreo efectivo y una intervención temprana. La capacidad de monitorear de manera precisa y continua los signos vitales de

los pacientes adquiere una nueva urgencia en situaciones de aislamiento y telemedicina.

Este proyecto aborda necesidades críticas mediante la implementación de un sistema de monitoreo de signos vitales. Proporciona a los profesionales de la salud la capacidad de supervisar y analizar datos en tiempo real, independientemente de la ubicación física del paciente o del médico. La aplicación de tecnologías avanzadas, como microcontroladores y la Internet de las Cosas (IoT), facilita una transmisión de datos fluida y en tiempo real. Esto contribuye a fortalecer la atención médica basada en datos y a respaldar la toma de decisiones informada.

Los logros concretos de este proyecto incluyen la creación de un prototipo de monitoreo de signos vitales. Se han abordado aspectos clave, como el diseño y desarrollo integral de hardware y software, la exitosa integración de tecnologías, la optimización de la precisión en las mediciones y la creación de una interfaz de usuario intuitiva y accesible.

El impacto potencial de este proyecto es considerable. La implementación exitosa del sistema podría mejorar significativamente la atención médica en situaciones de pandemia, aislamiento y telemedicina. Además de su aplicación en emergencias de salud, el sistema también puede encontrar utilidad en entornos hospitalarios convencionales, hogares de cuidado, y otras situaciones donde el monitoreo constante de signos vitales es esencial para la atención y el bienestar del paciente.

Este proyecto no solo aborda las necesidades actuales de la atención médica en un mundo en constante cambio, sino que también abre la puerta a nuevas formas de monitoreo y atención basadas en la tecnología. La capacidad de proporcionar cuidados médicos efectivos y personalizados, independientemente de las limitaciones físicas, confirma el potencial innovador y la importancia duradera de este proyecto.

1.4. Delimitación del problema

1.4.1. Temporal

La implementación de este proyecto se realizó en el periodo lectivo número 62, en el año 2023.

1.4.2. Espacial

La delimitación espacial de este prototipo ya que es portable sería en cualquier lugar con la única limitante que posea conexión a internet.

1.4.3. Académica

La delimitación académica de este proyecto, en el mismo se refuerzan conocimientos de las materias: Electrónica Analógica 1 y 2, Electrónica Digital, Sistemas Microprocesados 1 y 2, Redes de Computadoras 1 y 2, Sensores y Transductores, Instrumentación, Informática Industrial, DSP, Teoría del Diseño.

1.5. Objetivos

1.5.1. Objetivo General

Implementar un prototipo que realice el monitoreo de signos vitales y también permita la visualización de estos parámetros desde un dispositivo con acceso a internet, utilizando una tarjeta interfaz programada con un microcontrolador PIC18F4550 y Arduino nano que se comunican con un raspberry Pi para su conexión a internet.

1.5.2. Objetivo Específicos

- Diseñar y enviar a fabricar la tarjeta PCB del prototipo.
- Implementar el montaje, configuración y conexión de todos los elementos eléctricos y electrónicos de la PCB y del equipo prototipo.
- Implementar la programación del microcontrolador PIC18F4550, Arduino nano y el dashboard de Node-Red en la tarjeta Raspberry Pi.
- Implementar la configuración del Raspberry Pi para su conexión a internet y usarlo como servidor.

2. MARCO TEÓRICO REFERENCIAL

2.1. Microprocesadores y Microcontroladores: Caracterización

La figura 1 presenta el esquema de una microcomputadora estándar, compuesta por tres elementos clave: la unidad central de procesamiento (CPU), la memoria y el sistema de entrada/salida (E/S). Estos componentes están conectados mediante grupos de líneas eléctricas denominadas buses. Los buses de direcciones transportan información de memoria o E/S, los buses de datos llevan datos e instrucciones, y los buses de control transmiten señales de control. (Pallas-Areny & Valdes-Perez, 2017)

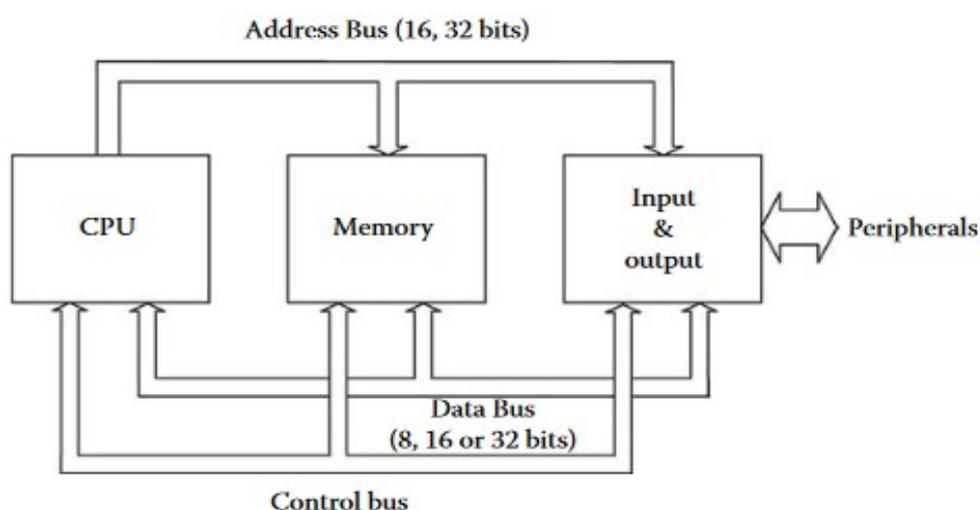


Figura 1. Diagrama de bloques genéricos de una microcomputadora
(Pallas-Areny & Valdes-Perez, 2017)

2.1.1. Microcontroladores PIC

Todos los microcontroladores PIC adoptan la arquitectura Harvard, caracterizada por separar las memorias de programa y datos. Como es habitual en muchos microcontroladores, la memoria de programa suele ser mayor que la memoria de datos. La memoria de programa se organiza en palabras de 12, 14 o 16 bits, mientras que la de datos consiste en registros de 8 bits. El acceso a los dispositivos de E/S se realiza a través de registros especiales en la memoria de datos llamados registros de funciones especiales (SFR). Además, varios microcontroladores PIC incluyen EEPROM adicional para almacenamiento de datos de manera no volátil.

2.1.2. Microcontroladores de gama alta

Los microcontroladores de alta gama se destacan por emplear instrucciones de 16 bits, una pila de mayor profundidad y un sistema de interrupción capaz de manejar tanto interrupciones internas como múltiples entradas para interrupciones externas. Algunos de estos PIC cuentan con una arquitectura abierta que permite la expansión tanto de la memoria de programa como de la memoria de datos, así como una mayor variedad de dispositivos disponibles. Además, los microcontroladores de alta gama pueden ser clasificados adicionalmente como:

- PIC17
- PIC18.

2.1.3. Interrupciones en Microcontroladores PIC

Una solicitud de interrupción, conocida comúnmente como interrupción, es un evento tanto interno como externo que, cuando se produce, interrumpe la ejecución del programa actual en el microcontrolador para ejecutar otro programa en su lugar. Una vez que el programa que maneja la interrupción ha concluido, el microcontrolador vuelve a ejecutar las instrucciones del programa que estaba en curso antes de la interrupción, como se representa en la Figura 2.

Las solicitudes de interrupción suelen ser eventos asíncronos respecto al programa en ejecución, lo que significa que pueden ocurrir en cualquier momento durante la ejecución del mismo. Por lo tanto, es imposible predecir qué instrucción se estaba ejecutando en el momento de la interrupción. Los microcontroladores tienen diversas fuentes de interrupciones, tanto internas como externas. Las interrupciones internas pueden originarse en los módulos de E/S, la memoria o la CPU del microcontrolador, siendo los temporizadores y otros módulos de E/S las fuentes de interrupción interna más comunes. Por otro lado, las interrupciones externas provienen de periféricos y se comunican al microcontrolador a través de sus pines y puertos asociados.

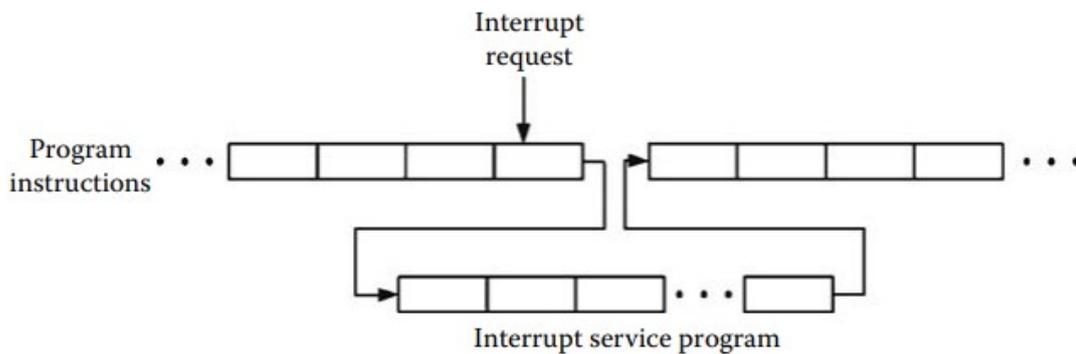


Figura 2. Esquema de una interrupción en un Microcontrolador
(Pallas-Areny & Valdes-Perez, 2017)

2.1.4. Introducción a la transmisión de datos en serie

La señal de datos puede ser generada de manera sincronizada con una señal de reloj, cuyo período determina la duración de un bit, tal como se ilustra en la figura 3. La velocidad de transmisión de la señal de datos (v_T) se define como el inverso de la duración de un bit. Si cada bit tiene una duración de τ segundos, entonces la velocidad de transmisión se calcula como:

$$v_T = \frac{1}{\tau} \text{ bit/s .}$$

Figura 3. Formula de velocidad de transmisión de datos

(Pallas-Areny & Valdes-Perez, 2017)

2.1.5. El Bus I2C

Philips desarrolló el bus de circuito integrado (I2C) para conectar circuitos integrados en una misma placa de circuito impreso utilizando únicamente tres líneas. Este bus se ha establecido como un estándar para la interconexión y la comunicación de datos seriales síncronos entre diferentes dispositivos ubicados a corta distancia, como microcontroladores, memorias, convertidores D/A y A/D, entre otros. Junto con la línea de tierra, el bus utiliza dos líneas adicionales: una para la transferencia de datos, llamada línea de datos seriales

(SDA), y otra para la señal de reloj, denominada línea de reloj serial (SCL). El bus I2C puede alcanzar velocidades de transmisión de hasta 100 kbits/s en modo de baja velocidad, 400 kbits/s en modo rápido y 3.4 Mbits/s en modo de alta velocidad. La Figura 4 ilustra varios dispositivos conectados utilizando el bus I2C.

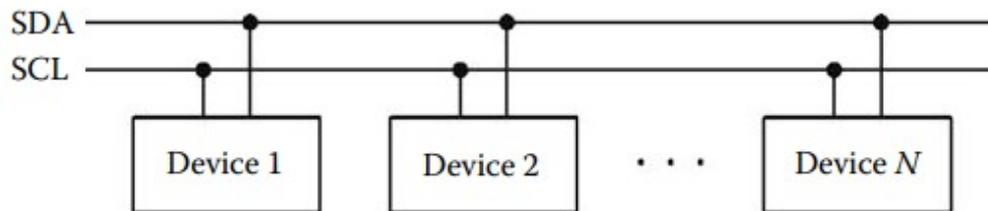


Figura 4. Varios dispositivos conectados en el bus I2C
(Pallas-Areny & Valdes-Perez, 2017)

2.1.6. El puerto serial USART en los microcontroladores PIC

Los microcontroladores PIC de nivel medio incluyen un puerto de comunicaciones serie conocido como receptor transmisor universal síncrono asíncrono (USART, por sus siglas en inglés) o interfaz de comunicación serie (SCI). Este puerto se puede configurar para facilitar una comunicación bidireccional asíncrona simultánea (full duplex) o una comunicación bidireccional síncrona no simultánea (transmitiendo la señal de reloj) (half duplex). (Welcome to Real Digital, s.f.)

2.1.7. El convertor de 10-Bits A/D en los microcontroladores PIC

Los microcontroladores PIC de nivel medio emplean convertidores A/D de aproximaciones sucesivas, típicamente de 10 bits. Una representación simplificada de la estructura interna de estos convertidores se muestra en la Figura 5. Los elementos principales de este módulo son:

- Multiplexor analógico con hasta ocho canales de entrada.
- Amplificador de muestreo y retención sin búferes de entrada o salida.
- Convertidor A/D de aproximaciones sucesivas de 10 bits.

- Registros para el control del módulo (ADCON0 y ADCON1), y registros para almacenar el resultado de la conversión (ADRESH y ADRESL).

Este módulo puede admitir hasta ocho entradas analógicas, las cuales se encuentran disponibles como funciones alternativas en las entradas del puerto paralelo. El número de entradas analógicas varía según el modelo específico del PIC. Por ejemplo, el PIC16F873 ofrece cinco entradas analógicas disponibles en cinco pines del puerto paralelo A (RA0/AN0, RA1/AN1, RA2/AN2/VREF-, RA3/AN3/VREF+ y RA5/AN4).

En el caso de los PIC con más de cinco entradas analógicas, como el PIC16F874 que dispone de ocho entradas analógicas, se utilizan tres pines del puerto E para las entradas analógicas AN5, AN6 y AN7. La selección de canales se realiza mediante los bits CHS2:CHS0 del registro ADCON0.

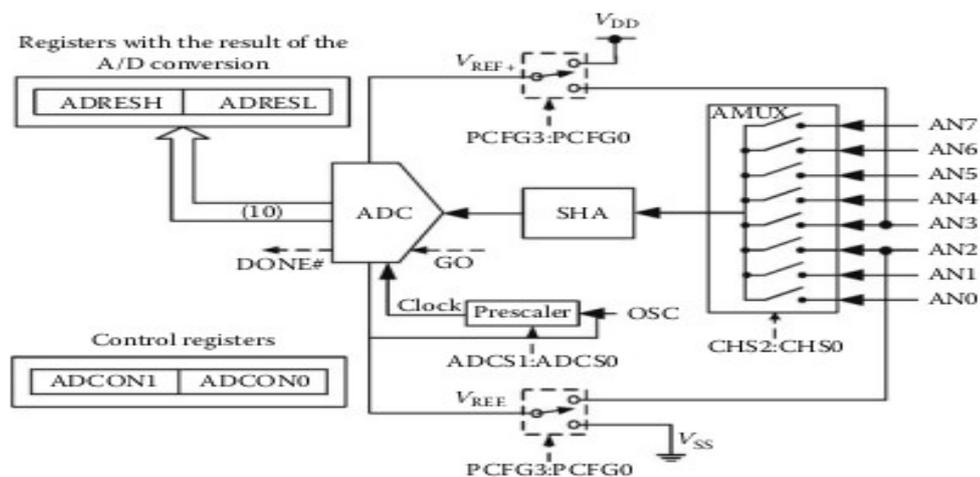


Figura 5. Bloque funcional de convertidor A/D
(Pallas-Areny & Valdes-Perez, 2017)

2.2. PIC18F4550

2.2.1. Especificaciones

Esta serie de dispositivos ofrece las ventajas típicas de todos los microcontroladores PIC18, que incluyen un alto rendimiento computacional a un precio asequible, junto con una memoria de programa Flash mejorada y resistente. Además de estas características, la serie PIC18F2455/2550/4455/4550 presenta mejoras de diseño que los convierten en una opción lógica para muchas aplicaciones que requieren alto rendimiento y sensibilidad al consumo de energía.

Los dispositivos de la serie PIC18F2455/2550/4455/4550 incorporan un módulo completo de comunicaciones por Bus Universal Serial (USB) que cumple con la Especificación USB Revisión 2.0. Este módulo admite tanto la comunicación de baja velocidad como la de velocidad completa para todos los tipos de transferencia de datos compatibles. También incluye su propio transceptor integrado en el chip y un regulador de voltaje de 3.3V, además de ser compatible con el uso de transceptores externos y reguladores de voltaje.

2.2.2. PINOUT

El pic 18F4550 y el pic 18F4555 poseen 40 pines:

40-Pin PDIP

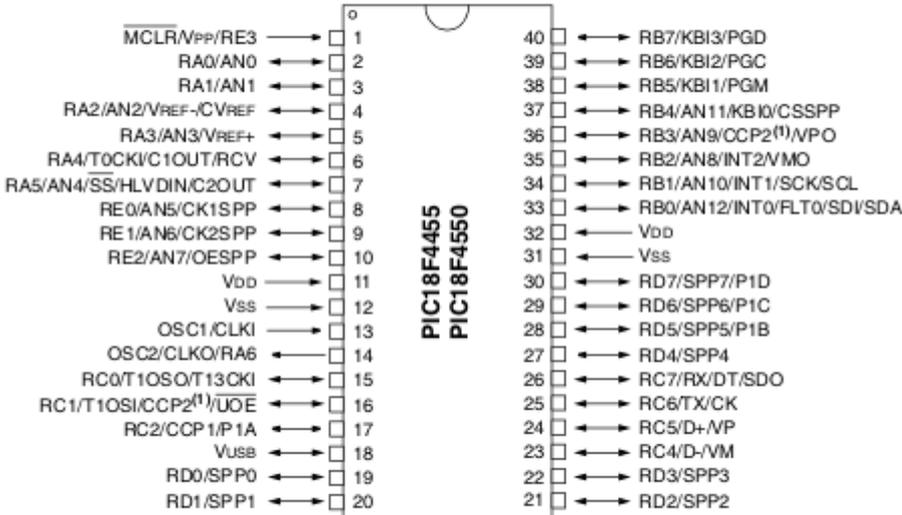


Figura 6. PIC18F4550-PINOUT
(Microchip, 2009)

2.2.3. Diagrama de bloques

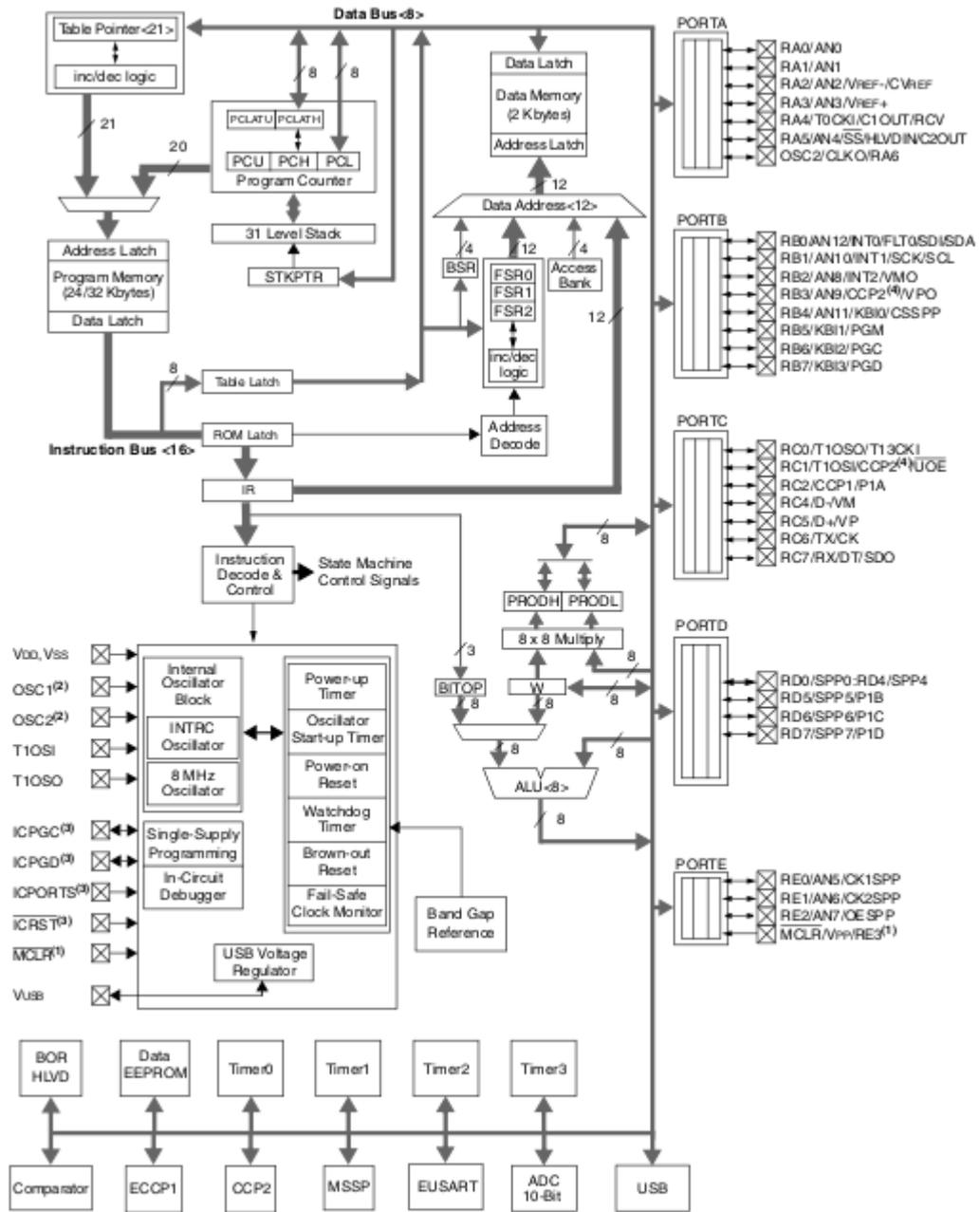


Figura 7. PIC18F4550-Diagrama de bloques

(Microchip, 2009)

2.2.4. Interrupciones

Los dispositivos PIC18F2455/2550/4455/4550 poseen diversas fuentes de interrupción, junto con una función de prioridad de interrupción que permite asignar un nivel de prioridad alto o bajo a cada fuente de interrupción. El vector de interrupción de alta prioridad se localiza en la dirección 000008h, mientras que el vector de interrupción de baja prioridad se encuentra en la dirección 000018h. Las interrupciones de alta prioridad tienen precedencia sobre las de baja prioridad que puedan estar en curso.

Para controlar el funcionamiento de las interrupciones, se emplean diez registros específicos, que son:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

2.2.5. Puertos de Entradas y Salidas

Según el dispositivo elegido y las funciones activadas, pueden haber hasta cinco puertos disponibles. Varios pines de estos puertos de entrada/salida pueden estar multiplexados con funciones alternativas provenientes de características periféricas en el dispositivo. En términos generales, cuando un periférico está habilitado, ese pin no puede utilizarse como un pin de entrada/salida de propósito general.

Cada puerto dispone de tres registros para su funcionamiento. Estos registros son:

- Registro TRIS (registro de dirección de datos)
- Registro PORT (lee los niveles en los pines del dispositivo)
- Registro LAT (latch de salida)

- Bluetooth de baja energía (BLE) incorporado.
- 40 pines GPIO extendidos.
- 4 puertos USB2.
- Salida estéreo de 4 polos y puerto de video compuesto.
- Puerto HDMI de tamaño completo.
- Puerto de cámara CSI para conectar la cámara Raspberry Pi.
- Puerto de pantalla DSI para conectar la pantalla táctil Raspberry Pi.
- Puerto MicroSD para cargar el sistema operativo y almacenar datos.
- Fuente de alimentación Micro USB mejorada con interruptor (ahora compatible con hasta 2.4 Amperios).
- Mismo factor de forma que la Raspberry Pi 2 Model B, aunque los LEDs cambiarán de posición.



Figura 9. Raspberry Pi 3
(Kurniawan,2016)

2.3.2. Sistema Operativo Raspbian

Raspbian es un sistema operativo basado en Debian Linux diseñado para el hardware de Raspberry Pi. Es recomendable descargar el archivo de imagen del sistema operativo en <http://www.raspberrypi.org/downloads>.

2.3.3. Pines GPIO

Los pines de entrada/salida de propósito general (GPIO) son pines genéricos en un circuito integrado cuyo comportamiento, como si actúan como entrada o salida, puede ser controlado por el usuario durante la ejecución del programa. Estos pines GPIO no tienen un propósito específico definido y, por defecto, no están asignados a ninguna función particular.

Para comprender mejor los pines GPIO en la placa Raspberry Pi 3, se puede consultar la figura que se muestra a continuación. (Kurniawan, 2016)

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	⬛	DC Power 5v	02
03	GPIO02 (SDA1 , I ² C)	⬛	DC Power 5v	04
05	GPIO03 (SCL1 , I ² C)	⬛	Ground	06
07	GPIO04 (GPIO_GCLK)	⬛	(TXD0) GPIO14	08
09	Ground	⬛	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	⬛	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	⬛	Ground	14
15	GPIO22 (GPIO_GEN3)	⬛	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	⬛	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	⬛	Ground	20
21	GPIO09 (SPI_MISO)	⬛	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	⬛	(SPI_CE0_N) GPIO08	24
25	Ground	⬛	(SPI_CE1_N) GPIO07	26
27	ID_SD (I ² C ID EEPROM)	⬛	(I ² C ID EEPROM) ID_SC	28
29	GPIO05	⬛	Ground	30
31	GPIO06	⬛	GPIO12	32
33	GPIO13	⬛	Ground	34
35	GPIO19	⬛	GPIO16	36
37	GPIO26	⬛	GPIO20	38
39	Ground	⬛	GPIO21	40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

Figura 10. Rasperry PI-GPIO
(Kurniawan, 2016)

2.4. Arduino nano

Arduino Nano es una placa de desarrollo compacta diseñada para facilitar la creación rápida de prototipos en un espacio reducido. Como el miembro más antiguo de la familia Nano, Arduino Nano ofrece numerosas interfaces adecuadas para proyectos que utilizan protoboards. En su núcleo, se encuentra el microcontrolador ATmega328, con una

frecuencia de reloj de 16 MHz, que proporciona funcionalidades similares a las del Arduino Duemilanove. Esta placa dispone de 20 pines de entrada/salida digital, 8 pines analógicos y un puerto mini-USB.

Arduino Nano es el primer microcontrolador integrado en la serie Nano con funciones básicas, diseñado especialmente para proyectos de pequeña escala dentro de la comunidad maker. Con su amplio conjunto de pines de entrada/salida, ofrece la ventaja de utilizar diversas comunicaciones serie como UART, SPI e I2C. Además, el hardware es compatible con Arduino IDE, Arduino CLI y el editor web.

Seguridad: Las capacidades de alto rendimiento y bajo consumo de energía brindan la oportunidad de desarrollar aplicaciones de seguridad, como sistemas de control de acceso mediante sensores de huellas dactilares. La flexibilidad para conectar sensores y dispositivos externos mediante comunicación serie ha ampliado su utilidad en este campo.

Medio ambiente: La característica de bajo consumo de energía del microcontrolador y las opciones de suministro de energía para la placa han mejorado la capacidad para implementar proyectos remotos de IoT relacionados con problemas ambientales.

Robótica: La robótica siempre ha sido un área de interés para la comunidad Maker y, con este pequeño hardware integrado, ahora es posible crear aplicaciones robóticas complejas y avanzadas. (Arduino, 2024)

2.4.1. Características

Microcontrolador Atmega328

- Procesador de 8 bits de alto rendimiento y bajo consumo de energía
- Alcanza hasta 16 MIPS para una frecuencia de reloj de 16 MHz
- 32 kB, de los cuales 2 kB son utilizados por el cargador de arranque
- 2 kB de SRAM interna
- 1 kB de EEPROM
- 32 x 8 registros de trabajo de propósito general
- Contador de tiempo real con oscilador separado

- Seis canales PWM
- USART Serie Serial Programable
- Interfaz serial SPI maestro/esclavo

Alimentación

- Conexión Mini-B USB
- Fuente de alimentación externa no regulada de 7-15 V (pin 30)
- Fuente de alimentación externa regulada de 5 V (pin 27)

Modos de Suspensión

- Inactivo
- Reducción de ruido ADC
- Ahorro de energía
- Apagado
- En espera
- En espera extendida

Entradas/Salidas (E/S)

- 20 digitales
- 8 analógicas
- 6 salidas PWM

2.4.1. PINOUT

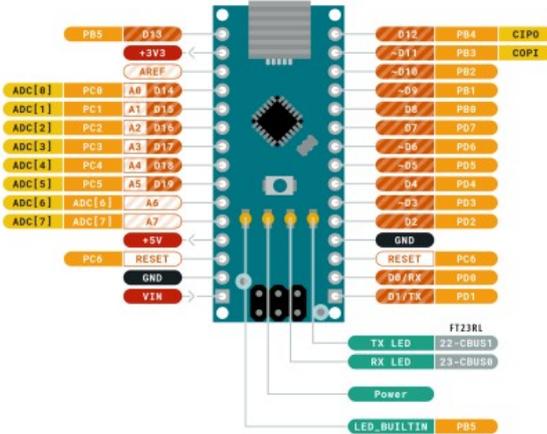


Figura 11: Arduino nano Pinout (Arduino, 2024)

2.4.2. Diagrama de Bloques

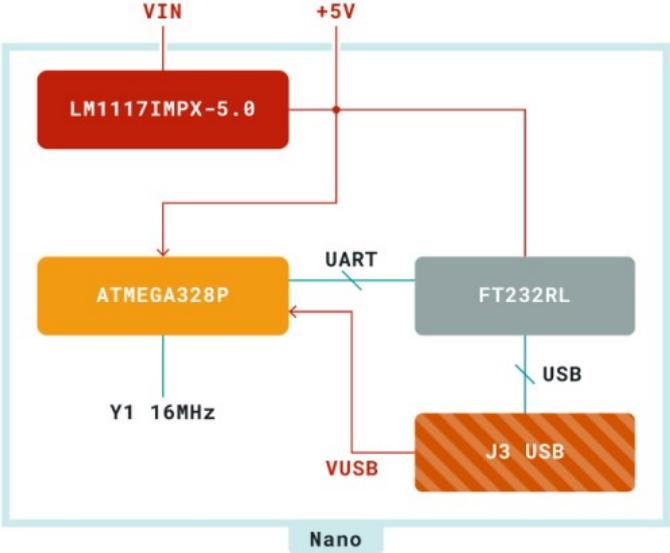


Figura 12: Arduino nano - Diagrama de bloques (Arduino, 2024)

2.5. Node-RED

2.5.1. Generalidades

Node-RED es una de las herramientas FBP. Desarrollada por el equipo de Servicios de Tecnología Emergente de IBM, Node-RED ahora está bajo la Fundación OpenJS.

FBP fue inventado por J. Paul Morrison en la década de 1970. FBP describe el comportamiento de la aplicación como una red de cajas negras, que en Node-RED se describe como un 'nodo'. El procesamiento está definido en cada nodo; se le proporcionan datos, se realiza el procesamiento con esos datos y esos datos se pasan al siguiente nodo. La red cumple la función de permitir que los datos fluyan entre los nodos.

Este tipo de método de programación es muy fácil de usar para crear un modelo visual y facilita el acceso para varios usuarios. Cualquiera puede entender lo que hace el flujo si se descompone un problema en cada paso. Es por eso que no es necesario el código dentro de los nodos:

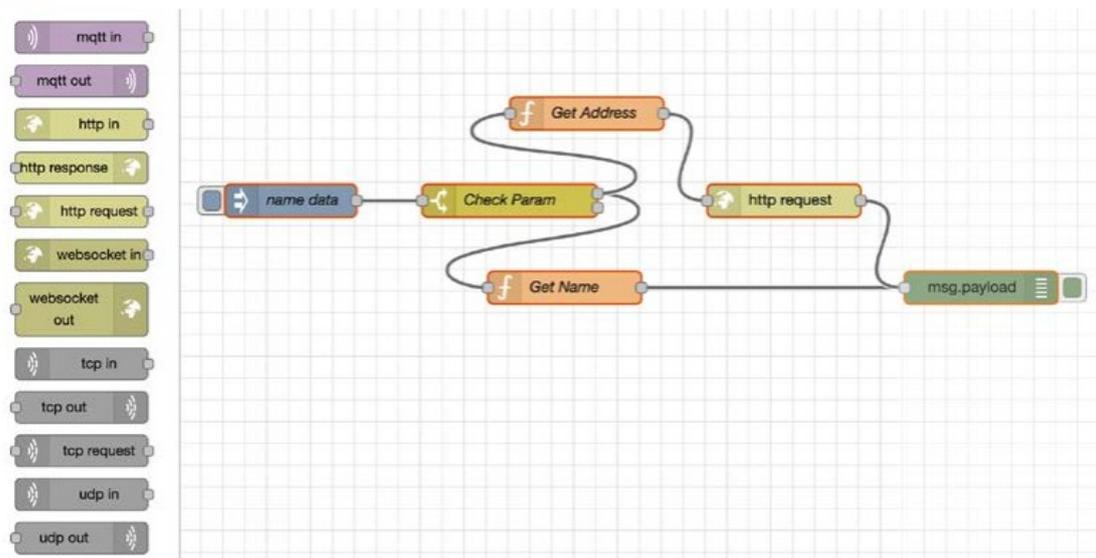


Figura 13. El Editor de Flujos de Node-RED como una herramienta FBP.

(Hagino, 2021)

2.5.2. El Editor de Flujos y el Entorno de Ejecución

Node-RED no solo se presenta como una herramienta de programación, sino también como una plataforma de ejecución que abarca el tiempo de ejecución de Node.js para aplicaciones construidas en Node-RED. Se emplea un editor de flujos para crear diversas aplicaciones, incluyendo aquellas orientadas hacia IoT y servicios web. Este editor de flujos, que constituye el núcleo de Node-RED, en realidad es una aplicación web desarrollada con Node.js y opera dentro del navegador. Al utilizarlo, los usuarios seleccionan nodos de una paleta y los arrastran al espacio de trabajo, donde pueden conectarlos para crear una aplicación mediante el proceso de cableado. Una vez completada, la aplicación puede ser implementada en el tiempo de ejecución deseado con un solo clic. Además, la paleta de nodos se puede ampliar fácilmente mediante la instalación de nuevos nodos creados por desarrolladores, lo que permite compartir los flujos creados como archivos JSON de manera sencilla.

2.5.3. Node-RED junto a IOT

Una vez más, Node-RED representa un entorno virtual que integra dispositivos hardware, APIs y servicios en línea de manera innovadora, todo ello dentro de un navegador web. Sus características destacadas son las siguientes:

- Interfaz de usuario basada en navegador.
- Funciona con Node.js y es liviano.
- Encapsula funciones y puede ser utilizado como un nodo (lo que significa que las funciones están bloqueadas en una cápsula abstracta).
- Puedes crear y añadir tus propios nodos.
- Acceso fácil a los servicios en la nube de IBM.

En pocas palabras, esta herramienta se adecua para la creación de servicios vinculados con el Internet de las cosas (IoT), tales como la gestión de datos en dispositivos y la interconexión entre dispositivos periféricos y servicios en la nube. Dado que el concepto inicial de desarrollo de Node-RED se centraba en IoT, resulta coherente. Ahora, se examinará la estructura fundamental del IoT, la cual se compone de seis capas, como se ilustra en el siguiente diagrama:

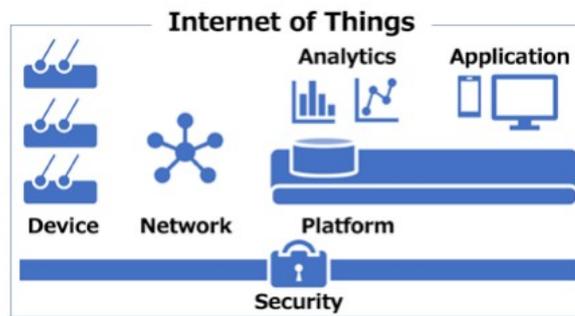


Figura 14. Las 6 capas de IOT
(Hagino, 2021)

Dispositivo

El dispositivo, que se identifica como un dispositivo periférico o "edge device" en el contexto del Internet de las Cosas (IoT), está equipado con una variedad de sensores para la recolección de datos. Sin embargo, almacenar la información exclusivamente en el dispositivo periférico resulta insuficiente, por lo que se requiere la transmisión de estos datos a través de una pasarela (gateway) hacia la red.

Red

Esta red se encarga de enviar los datos recolectados desde el dispositivo hacia un servidor en internet. Normalmente, se refiere a la conexión a internet. Además de internet, también se pueden establecer conexiones P2P mediante Bluetooth o serie.

Plataforma

La plataforma es la parte encargada de recibir y utilizar los datos. Aquí es donde se pueden encontrar bases de datos que permiten activar y autenticar dispositivos, gestionar comunicaciones y almacenar los datos recibidos.

Analítica

Esta capa se encarga de analizar los datos recibidos. Básicamente, funciona como una aplicación que prepara los datos para ser procesados de manera significativa.

Aplicación

Una aplicación ofrece un servicio específico basado en los resultados del análisis de datos. Puede tratarse de una aplicación web, móvil o integrada en hardware. Esta capa representa la interfaz utilizada por el usuario final de la solución IoT.(Hagino, 2021)

2.6. Oximetría de Pulso

La oximetría de pulso, introducida en 1983, representa un método no invasivo para monitorizar la saturación de oxígeno arterial en la sangre de los pacientes. Ampliamente reconocida como el estándar de atención en el campo de la anestesiología, su aplicación se extiende a diversas áreas como cuidados intensivos, quirófanos, emergencias, transporte de pacientes, áreas generales, partos, cuidados neonatales, laboratorios de sueño, atención domiciliaria y medicina veterinaria. Este procedimiento proporciona una detección temprana de problemas relacionados con la entrega de oxígeno a los tejidos, los cuales pueden surgir debido a factores como mezclas de gases incorrectas, obstrucciones en las vías respiratorias, ventilación inadecuada, difusión o circulación insuficiente, entre otros. Actualmente, más de 35 empresas fabrican y distribuyen más de 300,000 oxímetros de pulso en uso en los Estados Unidos.

2.6.1. Transporte Normal de Oxígeno

El oxígeno desempeña un papel crucial en el funcionamiento de cada célula del cuerpo humano. La ausencia prolongada de oxígeno conduce a la muerte celular, lo que subraya la importancia del suministro de oxígeno como un indicador clave de la salud del paciente.

Diversos métodos han sido desarrollados para evaluar la entrega de oxígeno, siendo la oximetría de pulso uno de los enfoques más comunes y no invasivos empleados en entornos clínicos.

El proceso de entrega de oxígeno a las células implica la colaboración entre el sistema respiratorio y el sistema circulatorio. La ventilación es el primer paso, moviendo el aire dentro y fuera de los pulmones. Dentro de los pulmones, se produce el intercambio gaseoso, donde el oxígeno se difunde hacia la sangre, mientras que el dióxido de carbono, producto residual del metabolismo celular, se difunde hacia los pulmones.

La sangre oxigenada circula por todo el cuerpo, llegando a áreas con necesidades de oxígeno, donde el oxígeno se difunde hacia las células y el dióxido de carbono se transfiere a la sangre que retorna a los pulmones. Este proceso respiratorio está controlado por

neuronas en el tronco encefálico. Además, el sistema circulatorio puede modular el gasto cardíaco para influir en la entrega de oxígeno.

2.6.2. Principios del Oxímetro de Pulso

Un oxímetro de pulso emite luz a dos longitudes de onda a través de un área tisular como el dedo o el lóbulo de la oreja, y luego mide la señal de luz transmitida. El funcionamiento del dispositivo se basa en los siguientes principios:

Absorbancia de la luz: La absorbancia de la hemoglobina oxigenada y desoxigenada a las dos longitudes de onda es distinta. Este modelo parte del supuesto de que solo la hemoglobina oxigenada y desoxigenada están presentes en la sangre, con suficiente variación en los coeficientes de extinción asociados para garantizar una sensibilidad adecuada sin que la sangre parezca opaca a las fuentes de luz.

Naturaleza pulsátil de la sangre arterial: La sangre arterial pulsátil genera una forma de onda en la señal transmitida, lo que permite distinguir los efectos de la absorbancia de la sangre arterial de los de la sangre venosa no pulsátil y otros tejidos corporales. Al calcular el cociente de los dos efectos a diferentes longitudes de onda, se puede obtener una medida que no requiere calibración absoluta en relación con la absorbancia total del tejido, lo cual constituye una ventaja significativa de los oxímetros de pulso sobre los tipos anteriores de oxímetros.

Efecto de dispersión de luz: Con suficiente intensidad lumínica, la dispersión en la sangre y los tejidos ilumina suficiente sangre arterial para permitir la detección confiable de la señal pulsátil. Si bien este efecto requiere una calibración empírica del oxímetro de pulso, también posibilita una ruta de transmisión alrededor del hueso en el dedo.

2.6.3. SpO2 como Monitor de Oxigenación de la Hemoglobina

La carencia de oxígeno puede ocasionar rápidamente daños irreversibles en los tejidos celulares con un metabolismo elevado, siendo el corazón y el sistema nervioso central ejemplos destacados. A pesar de la notable resistencia del cuerpo humano en muchos aspectos, el proceso fisiológico de mantener una función celular óptima a través del transporte de oxígeno constituye un sistema de control delicado y complejo. Si este sistema se ve alterado de manera significativa, podría volverse inestable e insuficiente para satisfacer las demandas de oxígeno de los tejidos.

Para resaltar la importancia de una adecuada oxigenación de los tejidos, se presentan en la Tabla 1 los tiempos de supervivencia para diferentes lechos de órganos después de que se inicia la anoxia o el paro cardíaco. Por ende, resulta crucial protegerse contra condiciones patológicas que puedan dar lugar a una oxigenación inadecuada de los tejidos.

Tabla 1

Tiempos de Anoxia

Organo	Tiempo de supervivencia después del inicio de la anoxia
Corteza Cerebral	Menos de 1 minuto
Corazón	5 minutos
Higado y Riñones	10 minutos
Musculos Esqueléticos	2 horas

(Webster, 1997)

2.6.4. Saturación de oxígeno arterial

La saturación de oxígeno arterial se refiere al nivel de oxígeno en la sangre presente en las arterias y arteriolas en todo el cuerpo. Esta saturación permanece constante en todo el sistema arterial, pero varía a nivel capilar. En adultos sanos, el rango típico de saturación de oxígeno en sangre (SaO₂) es superior al 90%, lo que se correlaciona con una presión parcial arterial de oxígeno (PaO₂) de 60 a 100 mmHg.

Dada la complejidad del proceso de oxigenación, resulta desafiante abordar la amplia gama de aplicaciones de la saturación arterial en entornos como unidades de cuidados intensivos, salas de operaciones y laboratorios de investigación. Los médicos están interesados en monitorear la SaO₂ por varias razones, ya sea para realizar evaluaciones cuantitativas o como una variable crítica para detectar posibles problemas, aunque no constituye una indicación directa de la peligrosa condición de hipoxia celular.

2.6.5. Hipoxia e Hipoxemia

La hipoxia se refiere a una situación en la que los tejidos no reciben suficiente oxígeno para funcionar correctamente, mientras que la hipoxemia indica una baja concentración de

oxígeno en la sangre. Aunque relacionadas, estas son dos condiciones distintas. La hipoxia representa una condición críticamente peligrosa en la que la función celular se ve comprometida. Dentro de la categoría de hipoxia, la hipoxia hipóxica surge como resultado de una saturación arterial baja, lo que la convierte en una situación de riesgo. Sin embargo, es importante destacar que la presencia de hipoxemia no siempre conduce a hipoxia, y viceversa. Es posible experimentar hipoxia sin evidencia de hipoxemia, y viceversa.

Por lo tanto, los resultados de la monitorización del contenido de oxígeno en la sangre deben ser interpretados con precaución por parte de los profesionales de la salud, ya que la saturación arterial de oxígeno (SaO₂) y, por ende, la saturación periférica de oxígeno (SpO₂) solo ofrecen una medida de hipoxemia, no de hipoxia.

2.6.6. Hiperoxia

La hiperoxia se caracteriza por una concentración elevada de oxígeno en la sangre, por encima de los niveles normales. La determinación de niveles excesivos de oxígeno es crucial en diversas circunstancias debido al potencial tóxico de los radicales de oxígeno. Sin embargo, los estudios indican que la oximetría de pulso no resulta eficaz para este propósito específico. Por ejemplo, en el caso de la retinopatía de la prematuridad en neonatos, el papel de la saturación arterial de oxígeno (SaO₂) para su determinación no está completamente esclarecido. Además, la imprecisión del 2 al 3% en la medición de SaO₂ para estimar la saturación de oxígeno arterial (SaO₂) introduce incertidumbre en este contexto.

2.6.7. Limitaciones

Limitaciones del Instrumento y Operativas

Muchas de las limitaciones de la oximetría de pulso se harán evidentes a lo largo del resto de este texto. Sin embargo, la tabla 2 resume algunas limitaciones proporcionadas por Severinghaus y Kelleher (1992). Estas se describen en detalle en la fuente original.

Tabla 2

Limitaciones de la Oximetría de Pulso

Limitaciones de la oximetría de pulso
Incidencia de fallas del instrumento
Baja relación señal-ruido
Derivación de luz y sonda mal aplicada
Vasoconstrictores
Límites de baja perfusión
Artefactos por movimiento
Pulso anormal
Interferencia de pulsos inducida por ventilador e interferencia de pulso venoso
Tiempos de respuesta
Luz ambiental
Electrocirugía
Interferencia de resonancia magnética (MRI)
Selección del sitio para la colocación de la sonda
Pigmentos cutáneos, tintes y esmalte de uñas
Hemoglobinas disfuncionales
Quemaduras y otros peligros
Falsas alarmas y falsas no alarmas

(Webster, 1997)

Limitaciones en SaO₂

Con frecuencia, en la literatura médica de anestesiología, se encuentran artículos que abordan las limitaciones de la oximetría de pulso, aunque al analizarlos detenidamente, lo que realmente se discute es la limitación en la monitorización de la saturación arterial de oxígeno, como señalan Hutton y Clutton-Brock (1993) y Mak (1993). Los autores de tales estudios resaltan esta distinción. Es interesante observar cómo la medición de SpO₂ se ha asociado estrechamente con la medición de SaO₂. No obstante, cualquier limitación relacionada con SaO₂ como variable monitoreada también se aplica a SpO₂.

Al emplear SaO₂ para evaluar la función pulmonar, es crucial tener precauciones. Es complicado considerar cualquier técnica de monitoreo como infalible, ya que pueden darse combinaciones engañosas de condiciones que ocasionen que la variable monitoreada parezca normal cuando, en realidad, existe una situación potencialmente riesgosa para el paciente. Por ejemplo, Hutton y Clutton-Brock (1993) y Mak (1993) señalan que la oximetría de pulso (es decir, SaO₂) resulta poco confiable para detectar la hipoventilación cuando la concentración de oxígeno inspirado es elevada. En situaciones como esta, es esencial adoptar un enfoque integral para la evaluación de la oxigenación mediante otras variables monitoreadas.

2.6.8. Funcionamiento de un Oxímetro de Pulso

El oxímetro de pulso resuelve numerosos problemas presentes en tecnologías previas al aprovechar el flujo pulsátil de la sangre, siguiendo el cambio en la absorbancia de la luz con cada pulso sanguíneo. Este dispositivo rastrea esta componente alterna de pico a pico, lo que permite que la medición no se vea afectada por la absorbancia causada por la sangre venosa o el tejido circundante.

A pesar de los avances, la dispersión de la luz sigue siendo una fuente de imprecisión en los oxímetros de pulso. Dado que la Ley de Beer no considera este fenómeno, no es posible calcular directamente la saturación de oxígeno en sangre (S_{O₂}). Por lo tanto, el oxímetro de pulso registra las absorbancias en dos longitudes de onda y utiliza los datos de co-oxímetros para buscar de manera empírica un valor estimado para SpO₂, que es una estimación de S_{O₂}.

LEDs (Diodos Emisores de Luz)

Una de las principales mejoras del oxímetro de pulso con respecto a sus predecesores radica en el uso de LEDs como fuente de luz. Los LEDs tienen la capacidad de emitir intensidades de luz considerables en proporción a la corriente que los impulsa.

En la figura 15, el bloque de control de los LEDs regula tanto la cantidad de corriente de accionamiento como la duración de los pulsos de luz. La sincronización de estas pulsaciones es crucial, ya que el fotodiodo no puede distinguir entre distintas longitudes de onda de luz. Por ende, el oxímetro de pulso depende del sistema de microprocesador para coordinar los pulsos de los LEDs con las muestras tomadas por el ADC, permitiendo que las absorbancias detectadas por el fotodiodo se atribuyan al LED correspondiente.

Las longitudes de onda seleccionadas para la oximetría de pulso son 660 nm y 940 nm. Estas elecciones se basan en la disponibilidad de LEDs que emiten luz en estas longitudes de onda, además de la variación máxima posible en los coeficientes de extinción de Hb y HbO₂.

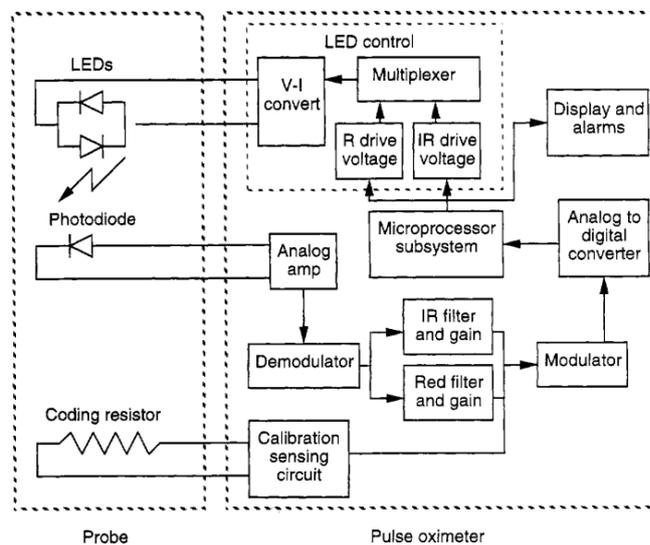


Figura 15. Diagrama de Bloques de un sistema de Pulso Oxímetro
(Webster, 1997)

Desventajas de usar LEDs como fuente de luz

Una desventaja asociada con el uso de LEDs como fuente de luz radica en la posible variación en la longitud de onda exacta de cada LED individual, que puede oscilar hasta 15 nm. Esta variabilidad podría provocar errores significativos si no se aborda adecuadamente. Para abordar este problema, diversos fabricantes realizan una caracterización detallada de cada LED y lo identifican con un valor de resistencia específico. Al suministrar una corriente constante a través de la resistencia codificada (como se muestra en la figura 15), el oxímetro

de pulso puede medir el voltaje resultante y ajustar los cálculos de SpO₂ teniendo en cuenta las características individuales de cada LED.

Fotodiodo

El fotodetector empleado es un fotodiodo de silicio que genera una corriente de manera proporcional a la intensidad de la luz que incide sobre él, en una relación lineal. Gracias a los avances en la tecnología del silicio, el tamaño reducido del fotodiodo permite su integración en sondas pequeñas colocadas en las puntas de los dedos. Estos avances han contribuido significativamente a mejorar la precisión y la comodidad del oxímetro de pulso en comparación con dispositivos previos. En el pasado, los oxímetros requerían una calibración frecuente debido a la inconsistencia de los dispositivos fotoeléctricos utilizados como sensores.

El fotodiodo no es capaz de distinguir entre la luz roja y la infrarroja. Para solucionar este problema, el sistema de microprocesador alterna la activación y desactivación de cada LED de manera secuencial. Durante la activación del LED rojo, el oxímetro de pulso registra repetidamente la salida del fotodiodo, lo mismo ocurre durante la activación del LED infrarrojo, así como cuando ambos LEDs están apagados. Al registrar la salida del fotodiodo con ambos LEDs apagados, el oxímetro de pulso puede eliminar cualquier interferencia de luz ambiental que pueda estar presente.

Sondas

Los avances tecnológicos en fotodiodos y LEDs también ofrecen otra ventaja significativa: permiten que las sondas sean más pequeñas y se conecten al oxímetro de pulso mediante cables convencionales. Por ejemplo, el oxímetro de ocho longitudes de onda de Hewlett-Packard se consideraba preciso, pero la necesidad de cables de fibra óptica voluminosos para llevar la fuente de luz al paciente y la luz transmitida de vuelta a un sensor de luz lo hacía poco práctico. Las sondas utilizadas en los oxímetros de pulso modernos no solo son más compactas, sino que también pueden ser desechables.

La figura 16 ilustra un oxímetro de pulso de transmisión y uno de reflectancia. Como sus nombres indican, un oxímetro de pulso de transmisión evalúa la cantidad de luz que atraviesa el tejido, como en el caso de una sonda de dedo, mientras que un oxímetro de pulso de reflectancia evalúa la cantidad de luz reflejada de vuelta a la sonda. Ambos tipos de oxímetros utilizan la misma tecnología, diferenciándose únicamente en la ubicación de las sondas y su calibración.

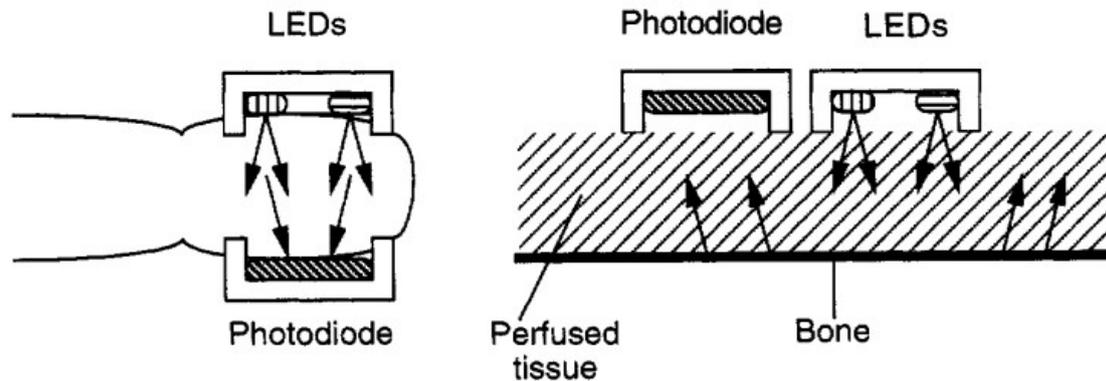


Figura 16. Ilustración del funcionamiento de la sonda del Pulso Oxímetro (Webster, 1997)

Amplificador analógico y procesamiento de señales

El fotodiodo convierte la intensidad de la luz en una corriente proporcional, la cual es luego convertida en voltaje por un amplificador analógico. Dado que el cambio en voltaje causado por las pulsaciones arteriales es pequeño en comparación con la componente de corriente continua de la señal, el demodulador resta esta componente del resto de la señal. Además, el demodulador emplea un circuito de temporización de muestreo y retención para distinguir las muestras del LED rojo de las del LED infrarrojo. Las partes de corriente alterna (CA) de estas señales se eliminan mediante un filtro paso bajo para reducir la interferencia electromagnética. Luego, cada señal se somete a un circuito de ganancia programable, tras lo cual un multiplexor, junto con otro circuito de muestreo y retención, combina las señales roja e infrarroja de nuevo en una sola para su conversión digital mediante un convertidor analógico a digital (ADC) para su posterior procesamiento por el microprocesador.

Usando los datos obtenidos del ADC, el microprocesador calcula una medida conocida como "relación de relaciones". A partir de esta relación y el valor de la resistencia de codificación, el microprocesador accede a una tabla de búsqueda empírica para determinar el valor de la saturación de oxígeno en la sangre (S_{O_2}). Esta tabla empírica se elabora mediante pruebas de laboratorio realizadas con un co-oxímetro por parte del fabricante. Además, los algoritmos de procesamiento de señales contribuyen a reducir el ruido en los datos recopilados. Ciertos oxímetros de pulso incorporan un electrocardiograma (ECG) en su algoritmo de procesamiento de señales para minimizar los errores debidos a artefactos de movimiento.

Oxímetro de pulso de tres longitudes de onda para determinación de COHb

Los oxímetros de pulso modernos estiman la saturación arterial de oxígeno midiendo las absorbancias de la luz en dos longitudes de onda específicas. Debido a esta metodología, estos dispositivos solo pueden considerar la hemoglobina (Hb) y la oxihemoglobina (HbO₂). Por lo tanto, niveles elevados de carboxihemoglobina (COHb), por ejemplo, pueden resultar en una sobreestimación de la saturación de oxígeno arterial (S_{O2}), ya que el oxímetro de pulso no puede distinguir entre HbO₂ y COHb. Esta limitación puede tener consecuencias graves en casos de intoxicación por monóxido de carbono, si el profesional médico no está al tanto de esta situación.

Buinevicius (1987) propuso una solución a este problema al diseñar un oxímetro de pulso que emplea tres longitudes de onda de luz. En este diseño, se incluyó un LED adicional que emite luz a 810 nm, con el objetivo de detectar la presencia de COHb en la sangre. Además, Buinevicius desarrolló un método para calibrar este oxímetro de pulso de tres longitudes de onda utilizando soluciones tridimensionales basadas en la Ley de Beer. (Webster, 1997)

2.7. Presión Arterial

La presión arterial se refiere a la fuerza ejercida por la sangre dentro de las arterias del cuerpo. Cada vez que el corazón se contrae, impulsa la sangre a través de las arterias, distribuyéndola por todo el cuerpo. Este proceso ocurre de forma continua, con el corazón latiendo aproximadamente entre 60 y 100 veces por minuto, durante todo el día. Las arterias desempeñan un papel crucial al transportar oxígeno y nutrientes a todas las partes del cuerpo para mantener su funcionamiento adecuado.

2.7.1. Presión arterial vs. frecuencia cardíaca

Ambas están vinculadas al funcionamiento de tu corazón, pero representan conceptos distintos. La presión arterial se refiere a la fuerza con la que la sangre circula por tus vasos sanguíneos, mientras que la frecuencia cardíaca indica el número de latidos que realiza tu corazón en un minuto.

Es importante tener en cuenta que un aumento en la frecuencia cardíaca no necesariamente indica un incremento en la presión arterial. La única manera de conocer tu presión arterial con certeza es utilizando un dispositivo de medición de presión arterial, como un brazalete y un monitor de presión arterial.

2.7.2. Por qué la presión arterial fluctúa

La presión arterial no se mantiene constante y varía según la actividad que estés realizando. Por ejemplo, durante el ejercicio o momentos de emoción, la presión arterial tiende a aumentar, mientras que en periodos de reposo suele disminuir.

Además, factores como la edad, los medicamentos que consumes y los cambios de posición también pueden influir en los cambios de la presión arterial.

2.7.3. Importancia de la medición de presión arterial

La hipertensión arterial, a menudo denominada el "asesino silencioso", generalmente no presenta síntomas evidentes. Puede ocasionar daños en el corazón, los riñones y el cerebro sin que el individuo sea consciente de ello.

Es un factor de riesgo significativo para diversas enfermedades cardiovasculares. Sin un tratamiento adecuado, la hipertensión arterial puede provocar condiciones como el ataque isquémico transitorio (AIT), accidente cerebrovascular, infarto de miocardio, agrandamiento del corazón, insuficiencia cardíaca, enfermedad arterial periférica, aneurismas, enfermedad renal y ruptura de vasos sanguíneos en los ojos.

CATEGORÍA DE LA PRESIÓN ARTERIAL	SISTÓLICA mmHg (número de arriba)	y/o	DIASTÓLICA mmHg (número de abajo)
NORMAL	MENOS DE 120	y	MENOS DE 80
ELEVADA	120 – 129	y	MENOS DE 80
PRESIÓN ARTERIAL ALTA (HIPERTENSIÓN) NIVEL 1	130 – 139	o	80 – 89
PRESIÓN ARTERIAL ALTA (HIPERTENSIÓN) NIVEL 2	140 O SUPERIOR	o	90 O SUPERIOR
CRISIS HIPERTENSIVA (consulte a su médico inmediatamente)	SUPERIOR A 180	y/o	SUPERIOR A 120

Figura 17. Rango de mediciones de Presión Arterial
(Grodzinsky & Sund Levander, 2020)

2.8. Temperatura Corporal

En 1871, se publicó la obra significativa "Medical Thermometry and Human Temperature" (Termometría Médica y Temperatura Humana), que marcó un hito en la comprensión de la temperatura corporal. Este trabajo, realizado por el médico alemán Carl August Wunderlich (1815-1910), se basó en meticulosas mediciones de la temperatura axilar. Fue el primero en establecer la temperatura corporal normal en 37.0 °C (98.6 °F) y considerar 38.0 °C (100 °F) como fiebre. Sin embargo, en ese período, la fisiología de la termorregulación no estaba completamente comprendida, había un conocimiento limitado sobre inmunología y microbiología, y la tecnología para termómetros clínicos precisos estaba en sus primeras etapas de desarrollo. Además, las mediciones de Wunderlich se realizaron en pacientes, lo que sugiere que muchos de ellos podrían haber estado experimentando fiebre en el momento de la medición. Hacia finales del siglo XIX, el médico inglés Sir Thomas Clifford Allbutt introdujo el termómetro clínico rectal de mercurio más corto, seguido más tarde por el termómetro oral de mercurio. No fue hasta la década de 1960 que surgieron dispositivos digitales, junto con nuevas técnicas para la medición no invasiva de la temperatura corporal.

2.8.1. Termopares

Thomas Seebeck, nacido en Estonia en 1770, es reconocido por su contribución fundamental al desarrollo del termopar como un dispositivo de medición de temperatura. Durante su tiempo en la Academia de Ciencias de Berlín en 1820, investigó la relación magnética de una corriente eléctrica. Un año después, anunció el descubrimiento de que dos metales distintos en un circuito cerrado exhibirían propiedades magnéticas cuando hubiera una disparidad de temperatura entre los dos puntos de contacto. Este fenómeno, conocido como el efecto Seebeck, sentó las bases de la termoelectricidad y condujo al desarrollo de termopares para la medición de temperatura por contacto.

En años recientes, esta tecnología ha evolucionado para ofrecer dispositivos de medición de calor altamente precisos, capaces de abarcar desde temperaturas cercanas al cero absoluto hasta elevadas temperaturas superiores a los 1600 °C (2912 °F). Si bien sus principales aplicaciones generalmente se sitúan fuera del rango de temperatura del cuerpo humano, un porcentaje de los dispositivos utilizados en la monitorización de pacientes en cuidados críticos emplean termopares adheridos a la piel para obtener mediciones continuas a lo largo del tiempo. Además, los termopares y termistores se emplean en catéteres sellados para medir la temperatura interna del cuerpo.

2.8.2. Principios Generales de Operación

Todos los termómetros clínicos poseen componentes esenciales que incluyen siquiera un sensor y un medio para mostrar la salida del sensor, como una pantalla digital. El sensor se integra en la sonda, la cual se posiciona de manera que esté térmicamente conectada al lugar de medición, es decir, el cuerpo. Este sensor convierte la energía térmica del lugar de medición en una señal eléctrica, a partir de la cual se puede determinar la temperatura del sitio en cuestión. Los termómetros pueden ser dispositivos de contacto, donde el termómetro se coloca directamente en contacto con el cuerpo, o dispositivos sin contacto, como los infrarrojos, que estiman la temperatura a través de la radiación infrarroja. Un ejemplo de termómetro de contacto es el termómetro eléctrico de tipo lápiz para mediciones sublinguales, mientras que un ejemplo de termómetro sin contacto es el termómetro timpánico (o de oído). Los termómetros de contacto logran un equilibrio térmico con el cuerpo, mientras que los sin contacto miden la radiación infrarroja emitida por la superficie del cuerpo.

Es crucial que los termómetros clínicos proporcionen una indicación precisa de la temperatura del paciente para asegurar un diagnóstico y tratamiento correctos. Deben ser adecuados para su propósito y apropiados para el uso al que están destinados, lo que implica que deben ser lo suficientemente compactos para insertarse correctamente en el lugar de medición y seguros de usar sin causar daño al paciente. Existen diferentes tipos y diseños de termómetros para diversas aplicaciones y para distintos grupos de pacientes. En su empleo en entornos clínicos, surgen varios problemas relacionados con la precisión, estabilidad y repetibilidad, entre otros, que los usuarios deben tener en cuenta.

2.8.3. Alteraciones en la Temperatura Corporal: Hipertermia e Hipotermia

La hipertermia, también conocida como golpe de calor, puede dividirse en dos categorías: la clásica y la relacionada con el esfuerzo físico. La primera suele afectar a personas mayores, enfermas crónicas, discapacitadas física o mentalmente, así como a niños pequeños, mientras que la segunda se observa principalmente en personal militar, de rescate y atletas. Además de la alta temperatura y humedad ambiental, los factores de riesgo incluyen enfermedades circulatorias y respiratorias, así como el uso de ciertos medicamentos como antipiréticos, betabloqueantes, diuréticos, psicofármacos, antidepresivos, anticolinérgicos, antihistamínicos y antiepilépticos. Otros factores de riesgo comprenden la obesidad, afecciones de la piel, enfermedades neurológicas, deshidratación, infecciones y actividad física. La hipertermia puede también surgir de la incapacidad para adaptarse a la

temperatura ambiente, como cuando un niño queda encerrado en un automóvil caliente o se duerme al aire libre en verano con demasiada ropa, lo que puede llevar a una retención excesiva de calor que sobrepasa la capacidad del cuerpo para disiparlo.

La hipertermia es una condición en la cual la temperatura corporal es tan elevada que existe riesgo de daño a las proteínas celulares, así como de una alteración o incluso una pérdida de la capacidad de regulación térmica. Se trata de una situación potencialmente mortal que requiere una acción inmediata mediante la aplicación de métodos de enfriamiento activo hasta que se restablezca la regulación térmica normal. El tratamiento de la hipertermia se basa en principios de disipación de calor mediante radiación, convección y evaporación.

Por otro lado, cuando la disipación de calor excede la producción de calor, puede ocurrir una disminución severa de la temperatura corporal, conocida como hipotermia. La hipotermia se clasifica en cinco etapas: leve (de 32 °C a 34 °C), moderada (de 28 °C a 31.9 °C), profunda (de 11 °C a 27.9 °C), severa (de 6 °C a 10.9 °C) y extremadamente severa (por debajo de 6 °C). Las temperaturas corporales por debajo de 35 °C conllevan el riesgo de un rendimiento cardíaco y una función muscular deteriorados, así como efectos cerebrales y liberación de hormonas del estrés. A los 34 °C, el metabolismo celular y las funciones termorreguladoras cesan, mientras que a los 30 °C, la termorregulación deja de funcionar por completo. La hipotermia puede ocurrir de manera accidental, como resultado de la exposición al frío después de un trauma, la ingesta de alcohol seguida de una pérdida de conciencia, o la exposición al agua fría o la nieve. Además, la hipotermia puede inducirse intencionalmente en ciertos procedimientos quirúrgicos con el fin de reducir el metabolismo. (Grodzinsky & Sund Levander, 2020)

3. REVISIÓN DE LA LITERATURA

3.1. Estado del Arte

El campo de sistemas embebidos, la investigación y desarrollo de tarjetas PCB con microcontroladores se ha convertido en un área de creciente importancia. Los sistemas embebidos desempeñan un papel fundamental en numerosas aplicaciones, desde dispositivos médicos hasta sistemas de automatización industrial y electrónica de consumo, un ejemplo de este tipo de desarrollos es (Morales, Hornak, Solari & Oreggioni, 2023).

3.1.1. Placas de Desarrollo y Microcontroladores

Los avances tecnológicos han permitido la proliferación de placas de desarrollo como Arduino, Raspberry Pi y microcontroladores PIC. Estas plataformas ofrecen versatilidad y facilidad de uso en el desarrollo de sistemas embebidos. El conocimiento y la experiencia en el manejo de estas plataformas son esenciales para implementar soluciones eficaces, ejemplos de esto son los artículos (Kazan, Terzioglu & Agacayak, 2015) y (Tank, Warriar & Jakhiya, 2017)

3.1.2. Diseño de Tarjetas PCB

El diseño de tarjetas PCB personalizadas es fundamental para adaptar los sistemas embebidos a aplicaciones específicas. El desarrollo de tarjetas PCB implica la disposición de componentes, trazado de pistas y consideraciones de diseño eléctrico. Además, la capacidad de diseñar y fabricar tarjetas PCB personalizadas brinda un alto grado de control sobre el desarrollo de proyectos. (Wang, Poorfakhraei, Narimani & Emadi, 2021)

3.1.3. Programación y Lenguajes de Programación

La programación desempeña un papel crucial en los sistemas embebidos. El conocimiento de lenguajes de programación como C, Java, JavaScript y PHP permite el desarrollo de software personalizado para controlar y monitorear sistemas embebidos. La programación efectiva es fundamental para la interconexión de dispositivos y la adquisición de datos de sensores. (Lu, Wu, Sheng, Liu & Yang, 2020)

3.1.4. Prototipo para el Monitoreo de Signos Vitales

Este proyecto de titulación se centra en la creación de un prototipo para el monitoreo de signos vitales, es una contribución valiosa al campo de la salud y los dispositivos médicos. La integración de tecnologías de sistemas embebidos para medir la oximetría, la presión arterial y la temperatura corporal es un ejemplo de cómo estas plataformas pueden aplicarse en entornos críticos de atención médica. (Amaya, Meléndez, Marín, & Cubías, 2023)

Esta investigación y desarrollo en este campo están en consonancia con las tendencias actuales en sistemas embebidos y dispositivos médicos, y se espera que ofrezca soluciones innovadoras y de gran utilidad. (Cuckov et al., 2021)

3.2. Revisión de Alcance

En esta sección, se llevará a cabo una revisión exhaustiva del alcance de la tesis. Esto implica definir los límites de la investigación y proporcionar una visión general de lo que se incluirá y lo que se excluye en el proyecto. La clarificación del alcance es esencial para garantizar que la tesis sea manejable y se centre en los objetivos establecidos.

3.2.1. Delimitación del Tema

La tesis se centra en el diseño, desarrollo y evaluación de un prototipo para el monitoreo de tres signos vitales: oximetría, presión arterial y temperatura corporal. La investigación abarcará aspectos técnicos, de hardware y software relacionados con sistemas embebidos, así como la interacción con el microcontrolador PIC18f4550, Arduino nano, Raspberry Pi y los dispositivos médicos usados.

3.2.2. Alcance de los Signos Vitales

Oximetría: La tesis abordará la medición de la saturación de oxígeno en sangre y el pulso del paciente mediante la tecnología de oximetría.

Presión Arterial: Se incluirá la comunicación serial con un equipo para medir la presión arterial no invasiva y la capacidad de registrar tanto la presión sistólica como la diastólica.

Temperatura Corporal: La tesis se centrará en la medición precisa y en tiempo real de la temperatura corporal utilizando una termocupla tipo K.

3.2.3. Exclusión del Alcance

Diagnóstico Médico: La tesis no tiene como objetivo proporcionar diagnósticos médicos, sino desarrollar un prototipo para la obtención de datos de signos vitales.

Implementación Clínica: La implementación clínica y la aprobación regulatoria no están dentro del alcance de esta tesis, ya que requieren evaluaciones adicionales y autorizaciones.

Ampliación de Signos Vitales: Aunque se monitorean tres signos vitales, no se expandirá el alcance para incluir otros parámetros médicos.

3.2.4. Limitaciones y Desafíos

Se reconoce que existen limitaciones en términos de recursos, tiempo y equipo para llevar a cabo la tesis. Además, se anticipan desafíos técnicos y de diseño durante el desarrollo del prototipo. Estos aspectos se abordarán en detalle en capítulos posteriores de la tesis.

3.2.5. Importancia del Alcance

La delimitación precisa del alcance garantiza que la tesis sea factible y que se alcancen los objetivos establecidos. Además, proporciona una base sólida para la toma de decisiones y la gestión de expectativas a lo largo del proyecto.

4. METODOLOGÍA

La metodología de esta tesis se basa en una serie de pasos y enfoques estratégicos para lograr los objetivos de diseño, desarrollo y evaluación del prototipo para el monitoreo de signos vitales. Estos pasos se dividen en las siguientes fases:

4.1. Investigación Inicial

Revisión de la literatura: Se realizó una revisión exhaustiva de la literatura existente relacionada con sistemas embebidos, sensores de signos vitales, microcontroladores, placas de desarrollo y tecnologías médicas.

Recopilación de requisitos: Se identificaron y documentaron los requisitos técnicos y funcionales del prototipo, incluyendo especificaciones para oximetría, presión arterial y medición de temperatura corporal.

4.2. Diseño y Desarrollo

4.2.1. Selección de Componentes

Se eligieron sensores, microcontroladores y otros componentes de acuerdo a su funcionalidad, costo, fiabilidad y durabilidad.

4.2.2. Diseño del Hardware

Se diseñaron dos tarjetas PCB para este proyecto, cada una con un propósito específico. La primera tarjeta está destinada a trabajar con el microcontrolador PIC18F4550 y la termocupla tipo K, utilizada para la lectura precisa de la temperatura corporal. En la Figura 18, se muestra el esquemático detallado de esta tarjeta, donde se puede apreciar la disposición de los componentes y las conexiones entre ellos. La Figura 19 presenta el enrutado completo de la PCB, también elaborado con el software Eagle. Este diseño garantiza una disposición óptima de los componentes y una ruta eficiente para las señales eléctricas, lo que es crucial para el funcionamiento correcto y confiable del sistema en su conjunto. Además, la utilización del software Eagle facilita la creación y modificación de los diseños, permitiendo ajustes precisos según sea necesario para cumplir con los requisitos del proyecto.

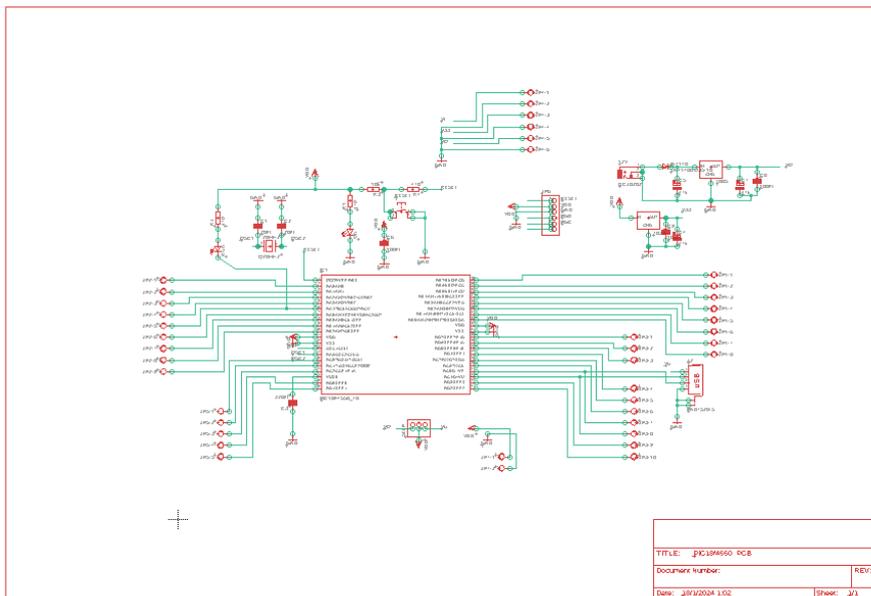


Figura 18: Diagrama Esquemático Tarjeta Interfaz con el PIC

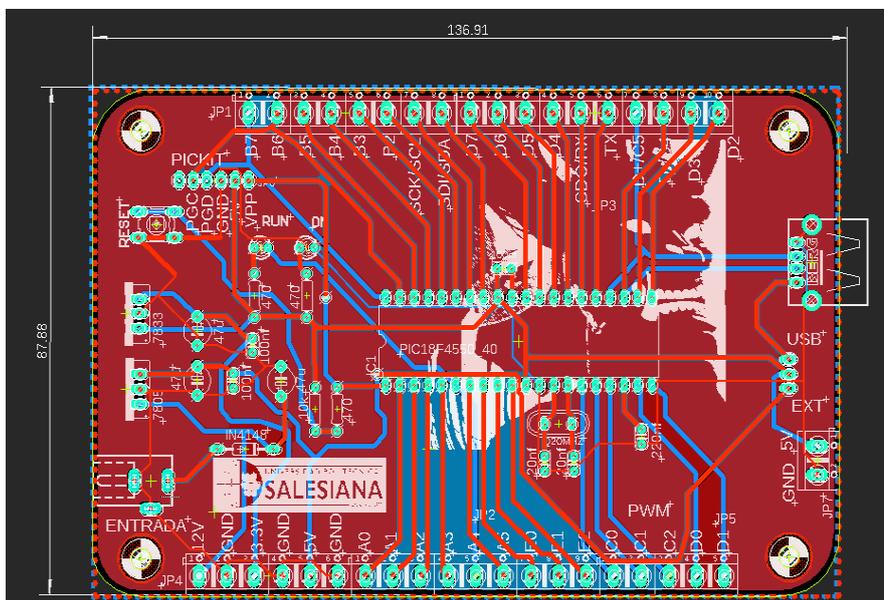


Figura 19: Diseño de la PCB de la Tarjeta Interfaz con el PIC

Se diseñó una segunda tarjeta PCB para funcionar como interfaz entre el sensor de oximetría de la marca Nellcor y el Arduino Uno, encargado del procesamiento de la señal adquirida. En la Figura 20, se presenta el diagrama esquemático detallado de esta tarjeta, donde se ilustra claramente la conexión entre los componentes necesarios para el correcto funcionamiento del sensor y el Arduino. La Figura 21 muestra el diseño completo de la PCB, también elaborado con el software Eagle. Este diseño proporciona una plataforma sólida y confiable para la comunicación efectiva entre el sensor de oximetría y el Arduino, lo que garantiza una correcta adquisición y procesamiento de los datos de oximetría en el sistema.

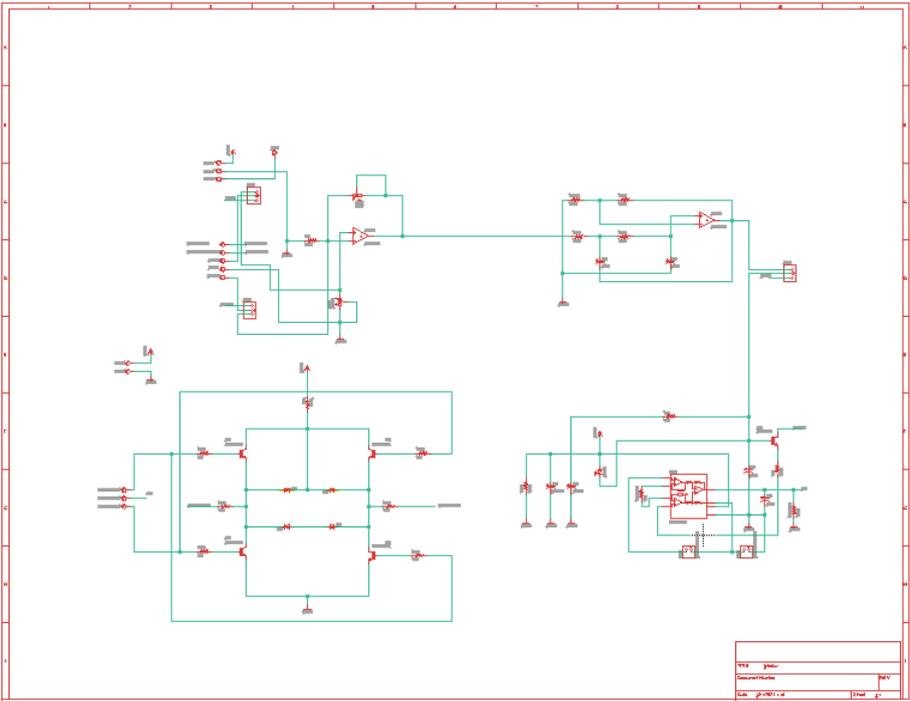


Figura 20: Diagrama Esquemático de la Tarjeta Interfaz con el sensor de Oximetría

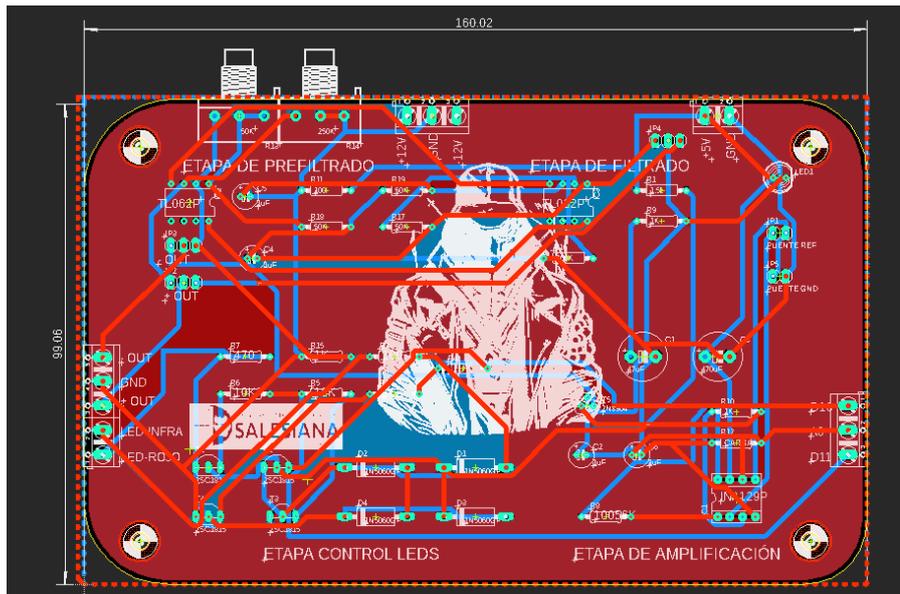


Figura 21: Diseño de la PCB de la Tarjeta Interfaz con el sensor de Oximetría

En las figuras 22 y 23 se muestra con la tapa, la caja diseñada en el software Fusion 360.



Figura 22: Imagen Frontal del Diseño en 3D del Prototipo

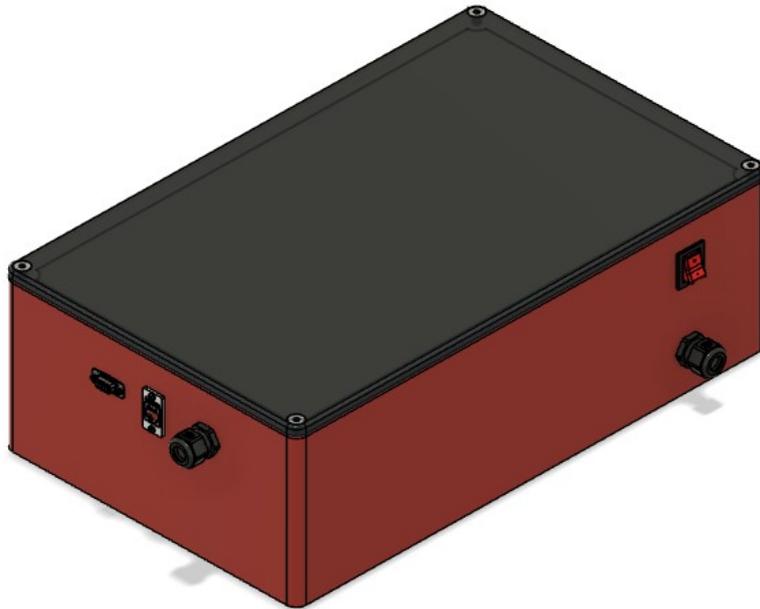


Figura 23: Imagen Lateral del Diseño en 3D del Prototipo

En las figuras 24 y 25 se presenta una vista lateral de la caja, esta vez sin la tapa superior, que muestra todos los componentes electrónicos del prototipo diseñados a tamaño real. Utilizando el software Fusion 360 se pudo exportar los diseños de PCB realizados en Eagle en formato 3D y a tamaño real. Además, los elementos comunes como Arduino, Raspberry Pi, fuente de alimentación, ventilador, conectores, entre otros, pueden descargarse de forma gratuita de la red, ya que la comunidad de código abierto comparte los diseños que crean para estos elementos.

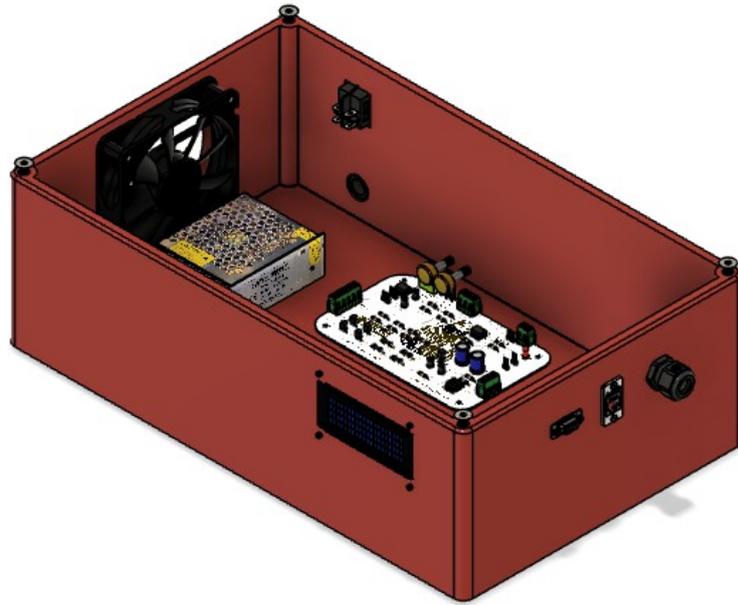


Figura 24: Imagen Frontal sin tapa del Diseño en 3D del Prototipo

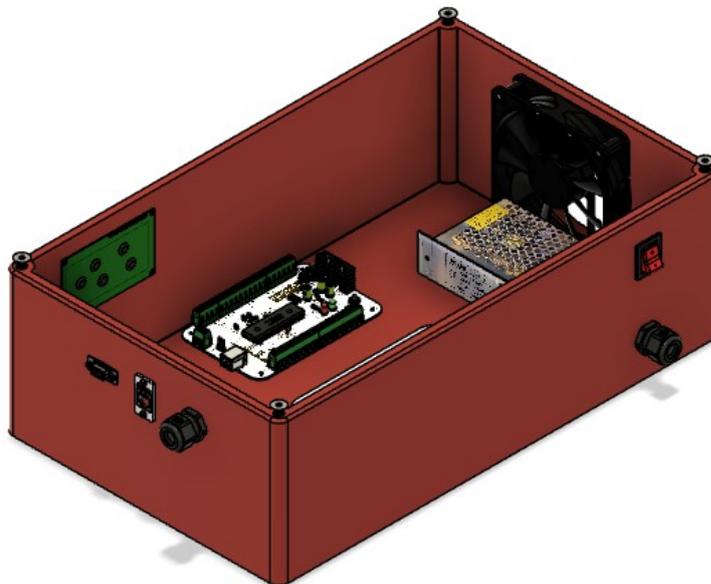


Figura 25: Imagen Lateral sin tapa del Diseño en 3D del Prototipo

En las figuras 26 y 27, se muestra la caja sin la tapa desde una vista superior, lo que brinda una perspectiva más detallada de cómo se distribuirían los elementos y el espacio disponible para el cableado. Esta vista nos permite visualizar claramente la organización interna del prototipo y planificar la disposición óptima de los componentes electrónicos para garantizar una estructura ordenada y eficiente. Además, facilita la identificación de posibles áreas de congestión o interferencia, lo que es crucial para asegurar un diseño funcional y de fácil mantenimiento.

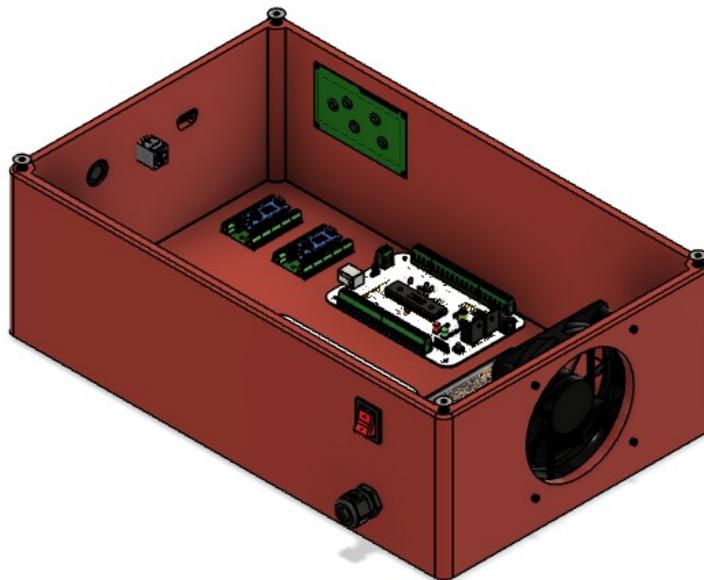


Figura 26: Imagen Trasera sin tapa del Diseño en 3D del Prototipo

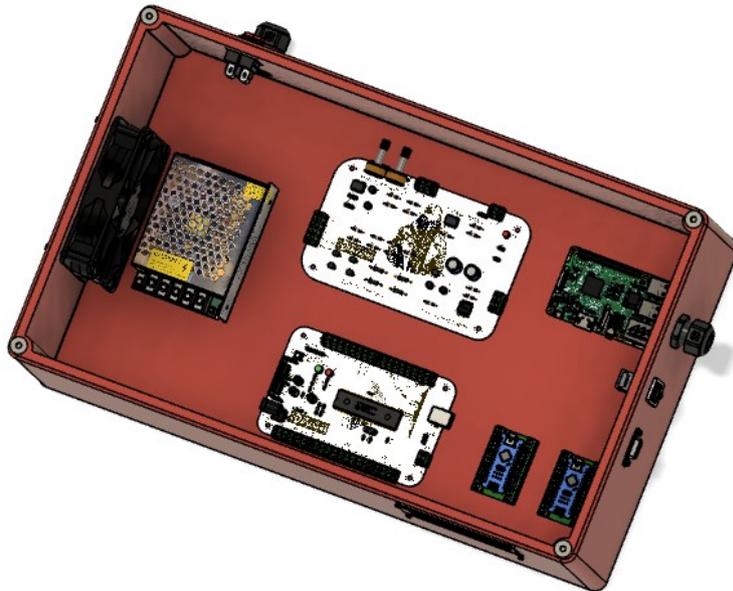


Figura 27: Imagen Superior sin tapa del Diseño en 3D del Prototipo

En estas imágenes, se logra apreciar con mayor claridad cómo se distribuyen los componentes dentro de la caja, lo que nos brinda una visión más completa del diseño y nos ayuda a evaluar la accesibilidad de los elementos para futuras actualizaciones o reparaciones. La disposición cuidadosa de los componentes y el espacio disponible para el cableado son aspectos fundamentales en el proceso de diseño, ya que contribuyen significativamente a la funcionalidad y la durabilidad del prototipo. Esta vista superior nos permite realizar ajustes precisos y optimizar el diseño para garantizar un rendimiento óptimo del dispositivo en su conjunto.

Las tarjetas terminadas se muestran en detalle en las Figuras 28 y 29, tras haber sido enviadas para su fabricación al proveedor extranjero JLCPCB. Este proveedor entregó un trabajo de calidad, como se puede apreciar claramente en las imágenes. La meticulosidad en la producción de las tarjetas garantiza un rendimiento óptimo y confiable del sistema en su conjunto. La colaboración con proveedores especializados como JLCPCB permite obtener resultados profesionales.

En la figura 30 y 31 se puede observar el proceso de soldadura que se llevo a cabo en las dos tarjetas PCBs enviadas a fabricar.



Figura 30: Proceso de Soldadura para la Tarjeta Interfaz con el PIC



Figura 31: Proceso de soldadura para la Tarjeta Interfaz con el Sensor de Oximetría

En la figura 32 se presenta el proceso de taladrado y fijación de los elementos al doble fondo, en la figura 33 se visualiza como se realizó el cableado del prototipo.



Figura 32: Montaje de las Tarjetas Electrónicas

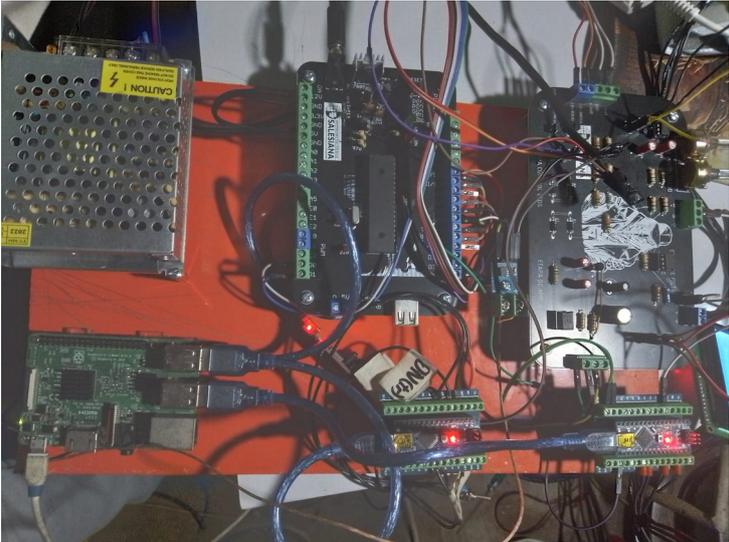


Figura 33: Tarjetas Electrónicas colocadas en su sitio

Finalmente, se optó por colocar el doble fondo con todos los elementos dentro de una maleta debido a que la caja diseñada para ser impresa en 3D resultaba demasiado grande y ninguna máquina impresora podía manejarla. Esta maleta, equipada con ruedas para facilitar su transporte, se muestra en las figuras 34 y 35. Esta solución brinda comodidad y practicidad al trasladar el equipo de un lugar a otro, asegurando su protección y facilitando su movilidad en diferentes entornos. La elección de una maleta con ruedas garantiza un transporte seguro y sin complicaciones, lo que resulta esencial para su aplicación en diversas situaciones y escenarios.



Figura 34: Imagen final del prototipo



Figura 35: Maleta con el prototipo

4.2.3. Desarrollo del Software

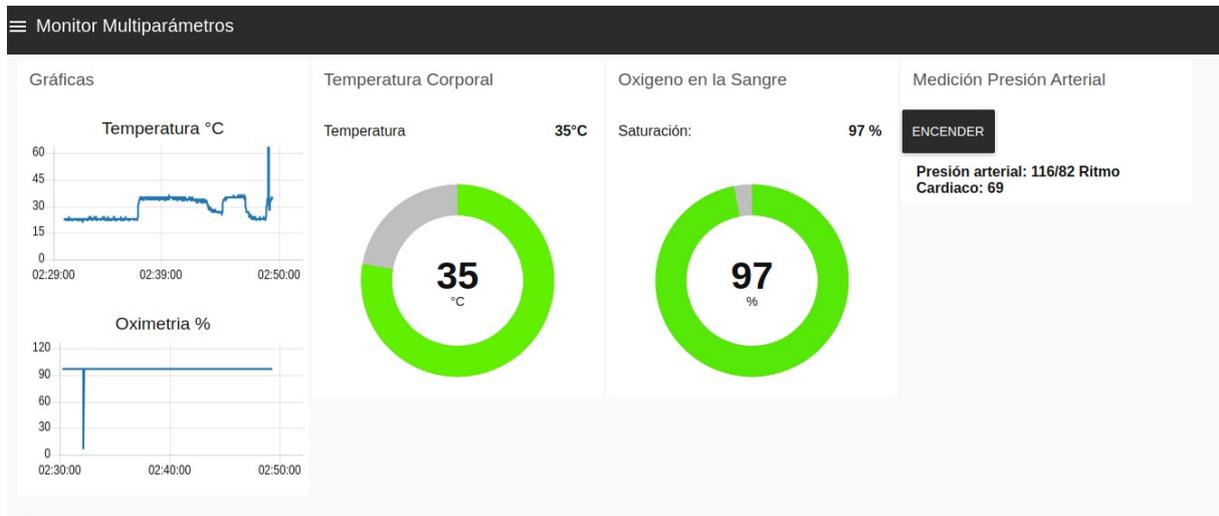


Figura 36: Dashboard en Node-RED

Se realiza una descripción del funcionamiento del software desarrollado, los códigos desarrollados se los adjunta en Anexos.

4.2.3.1. Oxímetro de Pulso:

Inclusión de bibliotecas:

Se incluyen las bibliotecas necesarias para el funcionamiento del código, como Wire, LiquidCrystal_I2C y SPI.

Definición de constantes:

Se definen varias constantes, como maxperiod_siz (número máximo de muestras en un período), measures (número de períodos almacenados), samp_siz (número de muestras para el promedio) y rise_threshold (umbral de medidas ascendentes para determinar un pico).

Creación de objeto LCD:

Se crea un objeto LCD para mostrar la información.

Definición de variables y pines:

Se definen las variables y pines necesarios para el sensor, LEDs y otras operaciones.

Configuración inicial en la función setup:

Se inicia la comunicación serial, se configuran los pines y se inicializa la pantalla LCD.

Bucle principal en la función loop:

El código principal se ejecuta en un bucle continuo.

Preparación de variables y arreglos:

Se inicializan variables y arreglos utilizados para el procesamiento de datos.

Lecturas de sensores:

Se realiza la lectura de los sensores infrarrojo y rojo para medir la intensidad de la luz transmitida a través del dedo.

Cálculo del coeficiente R:

Se calcula el coeficiente R, que se utiliza para determinar la relación entre la intensidad de la luz roja e infrarroja.

Verificación de presencia del dedo:

Se verifica si el dedo está presente en el sensor. Si no hay dedo, se muestra un mensaje en la pantalla.

Proceso de detección de latido cardíaco y cálculo de parámetros:

Se detectan los latidos cardíacos a partir de la curva de intensidad de luz. Se calcula la frecuencia cardíaca promedio y la saturación de oxígeno en la sangre.

Visualización de datos en la pantalla LCD:

Se muestran los resultados en la pantalla LCD, incluyendo la frecuencia cardíaca, saturación de oxígeno y otros indicadores.

Espera entre iteraciones del bucle:

Se introduce un retardo de 60 milisegundos entre iteraciones del bucle principal.

Este código es complejo y se basa en la detección de cambios en la intensidad de la luz a medida que pasa a través del dedo. La frecuencia cardíaca y la saturación de oxígeno se calculan a partir de estas mediciones.

4.2.3.2. Medición de Presión Arterial:

Inclusión de Librerías:

Se incluyen las librerías necesarias para el funcionamiento del programa, como Wire para la comunicación I2C.

Declaración de Variables:

Se declara una variable ``incomingByte`` para datos seriales entrantes.

Se definen constantes (``BP_START_PIN``, ``VALVE_PIN``, ``MEASURE_BEGIN_PIN``) para los pines del botón de inicio, la válvula y el inicio de la medición.

Se declaran variables volátiles (``i2c_data_rx``, ``count``, ``sys``, ``dia``, ``hr``) para manejar datos I2C y almacenar mediciones.

Configuración Inicial en ``setup()``:

Se inicializa la comunicación serial.

Se configuran los pines como entrada/salida según sea necesario.

Se inicia la comunicación I2C con una dirección específica (0x50) y se configura la función de interrupción para recibir datos I2C.

Bucle Principal en `loop()`:

Se verifica si hay datos disponibles en la comunicación serial.

Si se recibe la instrucción para comenzar la medición (`true` o señal en el pin `MEASURE_BEGIN_PIN`), se simula el botón de inicio y se espera a que la válvula se cierre.

Luego, se simula nuevamente el botón de inicio y se espera a que la válvula se cierre.

Se imprime por serial el resultado de la medición si hay datos, de lo contrario, se indica que no hay datos.

Rutina de Recepción de I2C (`receiveEvent`):

Esta función maneja la interrupción para datos I2C.

Lee los datos I2C y los almacena en las variables correspondientes (`sys`, `dia`, `hr`) según el número de datos recibidos.

4.2.3.3. Medición de Temperatura

Configuración y Librerías:

Se incluyen las librerías necesarias y se configuran los fusibles y la frecuencia del reloj.

Definiciones de Pines:

Se definen los pines para el control del LCD.

Definición de Constantes:

Se define la cantidad de mediciones (NUM_MEDICIONES) que se realizarán.

Arreglo de Mediciones:

Se declara un arreglo mediciones para almacenar las temperaturas medidas.

Función de Ordenamiento (Bubble Sort):

Se implementa la función ordenarMediciones() para ordenar las mediciones.

Función de Cálculo de Mediana:

La función calcularMediana() ordena las mediciones y calcula la mediana.

Función Principal (main()):

Se inicializan los pines y el LCD.

En un bucle infinito, se realizan las mediciones, se calcula la mediana y se muestra en el LCD.

Además, se imprime la mediana por el puerto serie.

4.3. Métodos y técnicas de Recopilación de datos empleadas

En esta sección, se detallan los métodos y técnicas utilizados para adquirir datos relevantes a lo largo de la investigación y el desarrollo del prototipo para el monitoreo de signos vitales. Estas técnicas se dividen en las siguientes categorías:

4.3.1. Adquisición de Datos de Sensores

Para obtener mediciones precisas de signos vitales como la oximetría, la presión arterial y la temperatura corporal, se emplearon las siguientes técnicas:

Sensores médicos: Se utilizaron sensores médicos especializados para oximetría y presión arterial. Estos sensores proporcionaron mediciones directas y precisas de la saturación de oxígeno en la sangre y la presión arterial.

Termopares: Se emplearon termopares para medir la temperatura corporal. Estos sensores de temperatura proporcionan mediciones sensibles y de alta precisión.

4.3.2. Procesamiento de Señales y Datos

Para garantizar mediciones precisas y confiables, se aplicaron las siguientes técnicas de procesamiento de señales y datos:

Filtrado de señales: Se implementaron técnicas de filtrado digital para eliminar el ruido de las señales de los sensores y garantizar mediciones suaves.

Calibración de sensores: Se realizaron procedimientos de calibración para ajustar los valores medidos por los sensores y garantizar mediciones precisas.

Algoritmos de procesamiento: Se desarrollaron algoritmos específicos para el cálculo de oximetría y la estimación de la frecuencia cardíaca a partir de las señales adquiridas.

4.3.3. Pruebas en Sujetos de Prueba

Para validar la precisión y la confiabilidad del prototipo, se llevaron a cabo pruebas en sujetos de prueba:

Entorno controlado: Las pruebas se llevaron a cabo en un entorno controlado para minimizar la influencia de factores externos.

4.3.4. Recopilación y Documentación

Todos los datos adquiridos a lo largo de la investigación se recopilaron, registraron y documentaron de manera exhaustiva. Esta documentación incluye registros de pruebas, resultados, observaciones y análisis de datos.

Los métodos y técnicas mencionados se seleccionaron cuidadosamente para garantizar mediciones precisas y resultados confiables en el desarrollo del prototipo para el monitoreo de signos vitales. Además, se aplicaron procedimientos de control de calidad y se mantuvo un enfoque riguroso en la recopilación y el análisis de datos.

5. INTERPRETACIÓN DE RESULTADOS

Considerando la naturaleza prototípica del sistema implementado, es imprescindible contextualizar los resultados obtenidos en un marco realista. A pesar de las limitaciones inherentes a un prototipo en desarrollo, los datos recopilados proporcionan una visión prometedora de su funcionalidad y precisión.

Para examinar los resultados de las pruebas de temperatura, en la tabla 3 se muestran cinco mediciones realizadas, cada una con un conjunto de datos específico. Cada prueba incluyó la medición de la temperatura en grados Celsius utilizando tanto el prototipo desarrollado como un equipo comercial de referencia. Además, se recopilaron datos adicionales, como la edad y el sexo de los sujetos de prueba.

Tabla 3

Mediciones de Temperatura

TEMPERATURA	PROTOTIPO	EQUIPO	EDAD	SEXO
°C		COMERCIAL		
PRUEBA 1	36	37	67	M
PRUEBA 2	36	35.2	54	F
PRUEBA 3	37	36	35	M
PRUEBA 4	35	36	16	M
PRUEBA 5	36	37	78	F

En la primera prueba, representada en la figura 37, se observa que la temperatura registrada por el prototipo fue de 36°C, mientras que el equipo comercial mostró una lectura ligeramente más alta de 37°C. Este resultado sugiere una ligera discrepancia entre las mediciones del prototipo y el equipo comercial.



Figura 37: Prueba de Temperatura

En la segunda prueba figura 38, se observó una temperatura de 36°C con el prototipo, mientras que el equipo comercial registró 35.2°C. Esta vez, el prototipo mostró una temperatura ligeramente más alta que el equipo comercial. Es interesante notar que esta diferencia podría ser resultado de variaciones normales en la temperatura corporal entre individuos, así como diferencias en la precisión y calibración de los dispositivos de medición.



Figura 38: Medicion de Temperatura

En la tercera prueba, figura 39, la temperatura registrada por el prototipo fue de 37°C, mientras que el equipo comercial mostró una lectura más baja de 36°C. Esta discrepancia puede sugerir una posible sobreestimación por parte del prototipo o una subestimación por parte del equipo comercial.

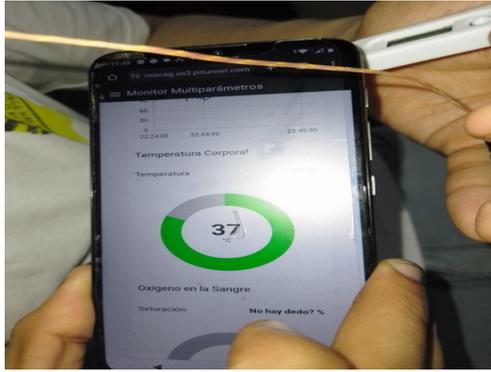


Figura 39: Medicion de Temperatura

En la cuarta prueba, se registraron temperaturas de 35°C con el prototipo y 36°C con el equipo comercial. Esta vez, el prototipo mostró una temperatura ligeramente más baja en comparación con el equipo comercial. La diferencia podría atribuirse a varias razones, como la precisión de los sensores utilizados en cada dispositivo o posibles errores de medición.

Finalmente, en la quinta prueba, se observó una temperatura de 36°C con el prototipo y 37°C con el equipo comercial. Nuevamente, se presenta una ligera discrepancia entre las mediciones de ambos dispositivos.

Los resultados obtenidos de las pruebas revelan que el prototipo desarrollado demostró ser capaz de medir la temperatura corporal, aunque se observó cierta variabilidad en las mediciones en comparación con un equipo comercial de referencia. Esta variabilidad puede atribuirse a diversos factores, como la precisión de los sensores, la calibración de los dispositivos y las diferencias individuales en la temperatura corporal de los sujetos evaluados.

En la figura 40 se muestra el código realizado en Python para calcular el error del prototipo con respecto al equipo comercial. Como resultado, el error promedio entre las mediciones del prototipo y del equipo comercial es de 0.96 °C.

CÓDIGO PARA CALCULAR EL ERROR EN LA MEDICIÓN DE LA TEMPERATURA

```
✓ [1] # Datos de temperatura obtenidos del prototipo y del equipo comercial
0s temperatura_prototipo = [36, 36, 37, 35, 36] # Temperatura en grados Celsius
temperatura_comercial = [37, 35.2, 36, 36, 37] # Temperatura en grados Celsius

# Calcular el error entre las mediciones del prototipo y del equipo comercial
errores = [abs(tp - tc) for tp, tc in zip(temperatura_prototipo, temperatura_comercial)]

# Calcular el error promedio
error_promedio = sum(errores) / len(errores)

# Mostrar el error promedio
print(f"Error promedio entre las mediciones del prototipo y del equipo comercial: {error_promedio:.2f} °C")
```

Error promedio entre las mediciones del prototipo y del equipo comercial: 0.96 °C

Figura 40: Código para el calculo del error en la medición de la Temperatura

También se realizó un código en python para realizar gráficas representativas para las mediciones de temperatura realizadas, en la figura 41 se realiza la comparación de Temperaturas entre el prototipo y el equipo comercial y en la figura 42 se grafica el error entre cada medición.

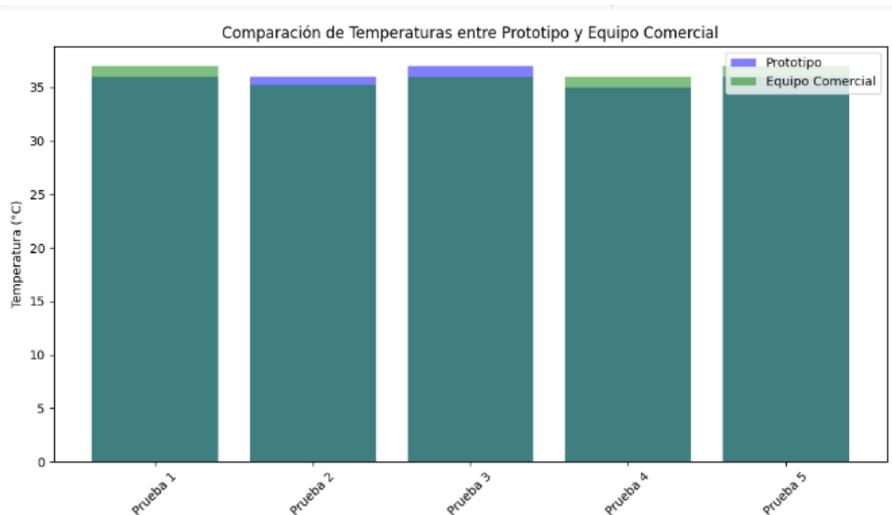


Figura 41: Comparación de Temperaturas

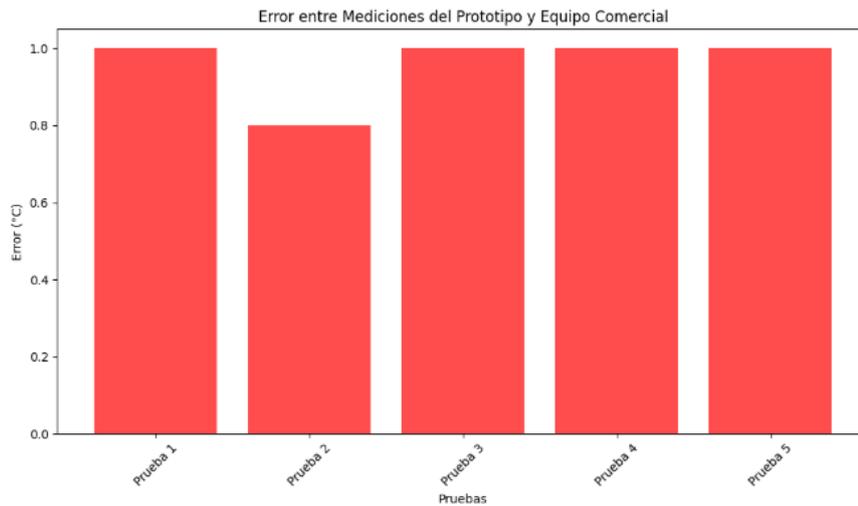


Figura 42: Error entre las mediciones de Temperatura

Es importante resaltar que, a pesar de esta variabilidad, el prototipo logró acercarse bastante a las mediciones obtenidas por el equipo comercial, lo que sugiere un desempeño aceptable para un proyecto universitario en fase de prototipo. Este logro indica un buen potencial para el desarrollo futuro del prototipo, destacando la viabilidad de realizar ajustes o mejoras para aumentar su precisión y fiabilidad en aplicaciones posteriores.

En consecuencia, los resultados obtenidos ofrecen una base sólida para continuar refinando el prototipo y explorar su potencial en diferentes escenarios y aplicaciones. Al considerar estas discrepancias como oportunidades de mejora, el proyecto puede avanzar hacia la optimización de su desempeño y su eventual implementación en entornos clínicos o de monitoreo de la salud.

Ahora se presentan las mediciones de saturación de oxígeno realizadas a los mismos sujetos de prueba. En la tabla 4, se visualizan los datos obtenidos en estas mediciones.

Tabla 4

Mediciones de Saturación de Oxígeno

SATURACIÓN %	PROTOTIPO	EQUIPO COMERCIAL	EDAD	SEXO
PRUEBA 1	94	95	67	M
PRUEBA 2	93	92	54	F
PRUEBA 3	95	97	35	M
PRUEBA 4	94	96	16	M
PRUEBA 5	96	95	78	F

En la figura 43 se observa la prueba uno, la saturación de oxígeno medida por el prototipo es ligeramente inferior en comparación con el equipo comercial. La diferencia de 1 punto porcentual puede considerarse dentro del rango aceptable de variabilidad.

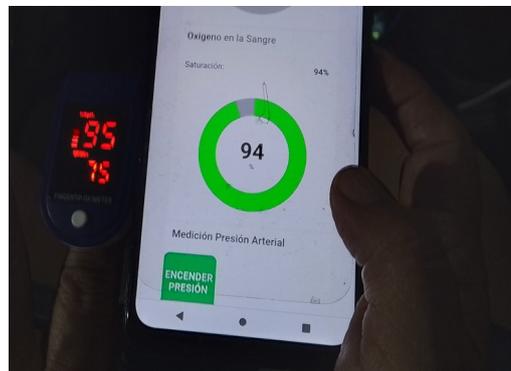


Figura 43: Medición de Saturación

En la prueba dos, figura 44, la saturación de oxígeno medida por el prototipo es ligeramente superior en comparación con el equipo comercial. Nuevamente, la diferencia de 1 punto porcentual es relativamente pequeña.



Figura 44: Medición de Saturación

En la Figura 45 se observa la medición de la prueba 3. La saturación de oxígeno que el prototipo mide es levemente inferior en comparación con el equipo comercial. La diferencia de 2 puntos porcentuales es algo más significativa, pero aún se considera aceptable.



Figura 45: Medición de Saturación

En la cuarta prueba, figura 46, la saturación de oxígeno medida por el prototipo es ligeramente inferior en comparación con el equipo comercial. La diferencia de 2 puntos porcentuales también podría considerarse dentro del rango aceptable de variabilidad.



Figura 46: Medición de Saturación

En la figura 47 se presenta la quinta prueba. La saturación de oxígeno que el prototipo mide es levemente superior en comparación con el equipo comercial. La diferencia de 1 punto porcentual es relativamente pequeña.

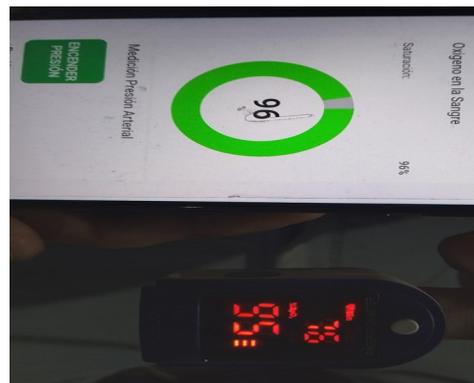


Figura 47: Medición de Saturación

En la figura 48 se muestra el código en Python para calcular el error del prototipo con respecto al equipo comercial. El resultado del cálculo indica que el error promedio entre las mediciones del prototipo y del equipo comercial es de 1.4 %.

CÓDIGO PARA CALCULAR EL ERROR EN LA MEDICIÓN DE LA SATURACIÓN

```
import numpy as np

# Datos de saturación de oxígeno
saturacion_prototipo = [94, 93, 95, 94, 96]
saturacion_comercial = [95, 92, 97, 96, 95]

# Cálculo del error absoluto
error_absoluto = np.abs(np.array(saturacion_prototipo) - np.array(saturacion_comercial))

# Cálculo del error relativo porcentual
error_relativo_porcentual = (error_absoluto / np.array(saturacion_comercial)) * 100

# Cálculo del error promedio
error_promedio = np.mean(error_absoluto)

# Impresión de los resultados
print("Error absoluto:", error_absoluto)
print("Error relativo porcentual:", error_relativo_porcentual)
print("Error promedio:", error_promedio)
```

Error absoluto: [1 1 2 2 1]
Error relativo porcentual: [1.05263158 1.08695652 2.06185567 2.08333333 1.05263158]
Error promedio: 1.4

Figura 48: Código para el cálculo del error en la medición de la Temperatura

También se realizó un código en python para realizar gráficas representativas para las mediciones de saturación realizadas, en la figura 49 se realiza la comparación de Saturaciones entre el prototipo y el equipo comercial y en la figura 50 se grafica el error entre cada medición.

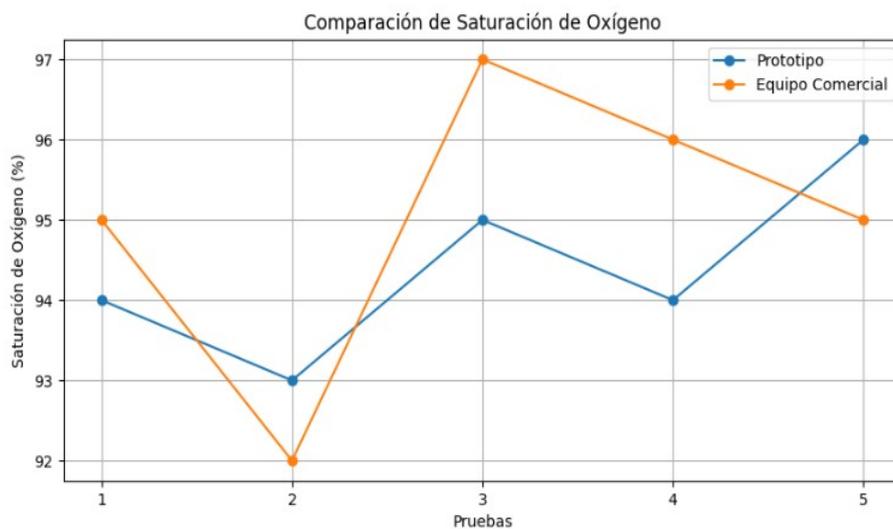


Figura 49: Comparación de Saturación de Oxígeno

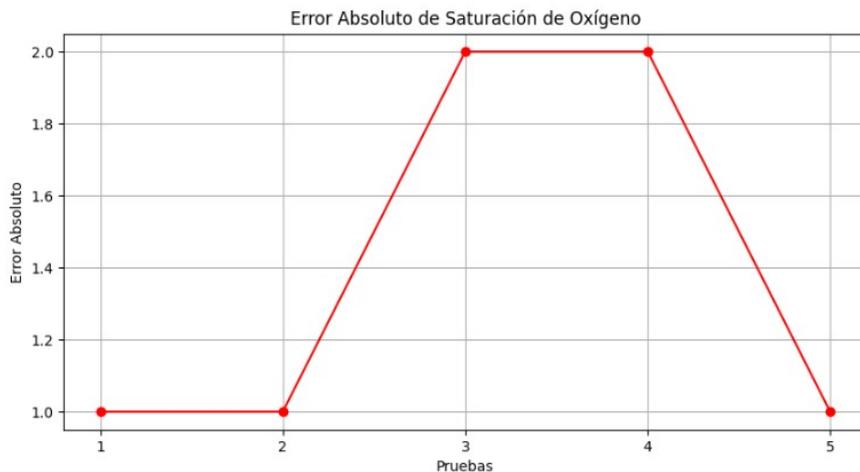


Figura 50: Error Absoluto de Saturación de Oxígeno

Los resultados muestran que, en general, la saturación de oxígeno medida por el prototipo es comparable a la medida por el equipo comercial. Sin embargo, existen algunas discrepancias entre las mediciones de ambos dispositivos.

En las pruebas 1, 2 y 4, la saturación de oxígeno medida por el prototipo es ligeramente menor que la medida por el equipo comercial, mientras que en las pruebas 3 y 5, la saturación medida por el prototipo es ligeramente mayor.

Aunque se observan algunas discrepancias en las mediciones de saturación de oxígeno entre el prototipo y el equipo comercial en cada una de las pruebas, las diferencias son generalmente pequeñas y podrían considerarse dentro del rango aceptable de variabilidad para un prototipo en desarrollo. Sin embargo, es importante tener en cuenta estas diferencias al interpretar los resultados y considerar posibles ajustes o mejoras en el prototipo para aumentar su precisión y fiabilidad en futuras aplicaciones.

En la tabla 5 se presentan las mediciones de presión arterial obtenidas tanto con el prototipo desarrollado como con un equipo comercial de referencia. Es importante resaltar que el tensiómetro utilizado en el prototipo es de origen comercial, y el método empleado para obtener los datos fue mediante la conexión a su memoria EEPROM. Esta conexión representó un desafío técnico que se superó con éxito, destacando la capacidad del prototipo para integrarse con dispositivos comerciales y aprovechar su funcionalidad en un entorno digital de salud. Esta demostración refuerza la viabilidad y utilidad del prototipo en aplicaciones de monitoreo de la salud, ofreciendo una solución efectiva y adaptable a las necesidades de la atención médica moderna.

Tabla 5:

Mediciones de Presión Arterial

PRESIÓN ARTERIAL	PROTOTIPO	EQUIPO COMERCIAL	EDAD	SEXO
PRUEBA 1	118/79	112/76	67	M
PRUEBA 2	138/98	134/95	54	F
PRUEBA 3	116/79	110/74	35	M
PRUEBA 4	106/77	101/73	16	M
PRUEBA 5	150/94	147/91	78	F

En la figura 51 se muestra la prueba uno, La presión arterial sistólica (118 mmHg) y diastólica (79 mmHg) medidas por el prototipo están ligeramente elevadas en comparación con las del equipo comercial. En este caso, la presión arterial podría considerarse en el rango normal-alto para un hombre de 67 años.



Figura 51: Medición de Presión Arterial

En la prueba dos, figura 52 y 53, tanto la presión arterial sistólica como la diastólica medidas por el prototipo son elevadas en comparación con las del equipo comercial. Estas mediciones sugieren una presión arterial elevada, lo cual podría ser preocupante, especialmente para una mujer de 54 años.



Figura 52: Medición de Presión Arterial

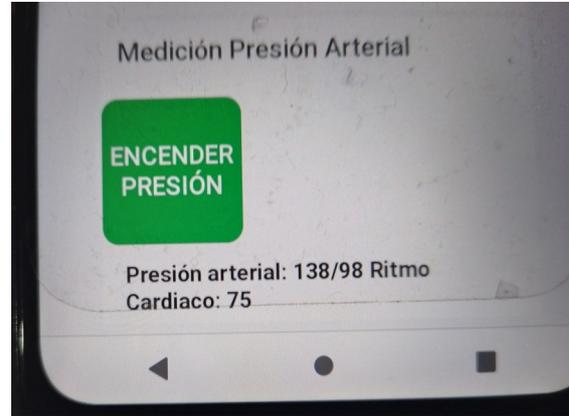


Figura 53: Medición de Presión Arterial

En la figura 54 se muestra la prueba tres, las mediciones de presión arterial del prototipo están ligeramente elevadas en comparación con las del equipo comercial. Aunque no se consideran altamente elevadas, podrían indicar una presión arterial en el límite superior de lo normal, especialmente para un hombre de 35 años.



Figura 54: Medición de Presión Arterial

La figura 55 representa la prueba cuatro, La presión arterial sistólica y diastólica medidas por el prototipo son ligeramente elevadas en comparación con las del equipo comercial. Estas mediciones podrían indicar una presión arterial elevada, lo cual es importante considerar incluso a una edad temprana.



Figura 55: Medición de Presión Arterial

Por ultimo en la prueba cinco, figura 56, Las mediciones de presión arterial del prototipo son ligeramente elevadas en comparación con las del equipo comercial. Estas mediciones sugieren una presión arterial elevada, lo cual es relevante considerando la edad avanzada y el sexo femenino.



Figura 56: Medición de Presión Arterial

En la figura 57 representa el código en Python para calcular el error del prototipo con respecto al equipo comercial. El resultado del cálculo indica que el error promedio entre las mediciones del prototipo y del equipo comercial es de 4.8 mmHg.

CÓDIGO PARA CALCULAR EL ERROR EN LA MEDICIÓN DE LA PRESIÓN ARTERIAL

```
✓ [13] import numpy as np
0 s      import matplotlib.pyplot as plt

# Datos de presión arterial del prototipo y del equipo comercial
presion_prototipo = [118, 138, 116, 106, 150]
presion_comercial = [112, 134, 110, 101, 147]

# Calcular el error
error = np.abs(np.array(presion_prototipo) - np.array(presion_comercial))

# Imprimir el error promedio
error_promedio = np.mean(error)
print("Error promedio:", error_promedio)
```

Error promedio: 4.8

Figura 57: Código para el calculo del error en la medición de la Temperatura

También se realizó un código en python para realizar gráficas representativas para la mediciones de saturación realizadas, en la figura 58 se realiza la comparación de Saturaciones entre el prototipo y el equipo comercial y en la figura 59 se grafica el error entre cada medición.

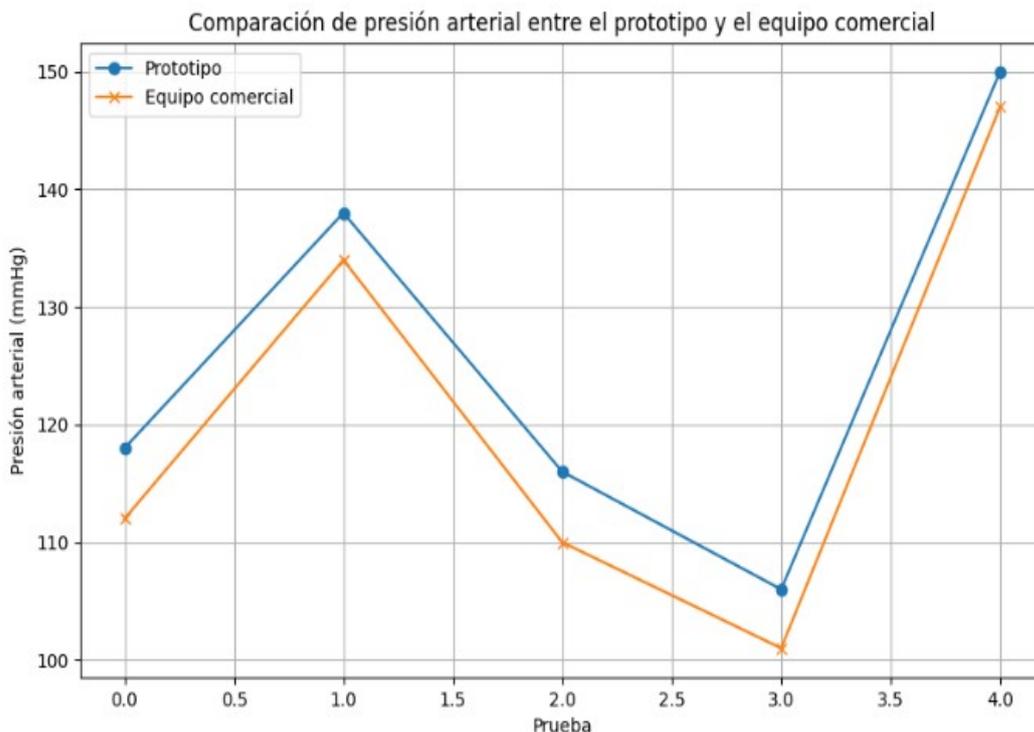


Figura 58: Comparación de Presión Arterial

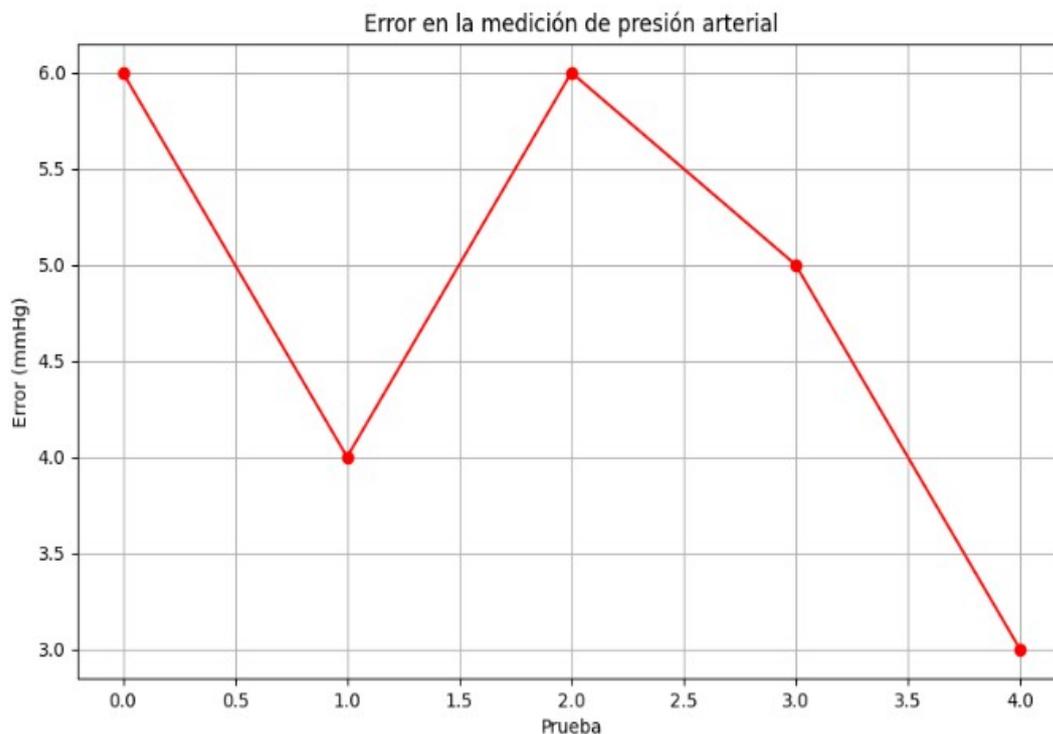


Figura 59: Error en la medición de Presión Arterial

Las mediciones de presión arterial muestran cierta variabilidad entre el prototipo y el equipo comercial, con el prototipo mostrando en general mediciones ligeramente más altas. Estas diferencias pueden ser importantes al considerar el riesgo de hipertensión y otras condiciones relacionadas con la presión arterial. Es esencial realizar un seguimiento adecuado de la presión arterial y considerar ajustes en el prototipo para mejorar su precisión y fiabilidad en futuras aplicaciones de monitoreo de la salud. La visualización de estos datos en un dashboard de Node-RED proporciona una herramienta valiosa para la monitorización continua y la toma de decisiones informadas en el cuidado de la salud.

Los resultados obtenidos tras la implementación y evaluación del prototipo revelan un desempeño altamente satisfactorio en la monitorización de signos vitales. La saturación de oxígeno, la temperatura corporal, la presión arterial y el ritmo cardíaco, medidas con precisión, confirman la fiabilidad y coherencia del sistema desarrollado.

La exactitud demostrada en la medición de estos parámetros es fundamental, destacando la capacidad del prototipo para proporcionar datos clínicamente relevantes. La respuesta consistente y confiable ante diferentes condiciones fortalece la confianza en la integridad de

las mediciones, respaldando la idoneidad del sistema para aplicaciones de monitoreo médico.

Estos resultados, obtenidos en un entorno controlado, sugieren que el prototipo ha cumplido con los objetivos planteados durante la fase de diseño y desarrollo. La interacción armoniosa entre los microcontroladores PIC 18F4550 y Arduino nano, junto con los diversos componentes utilizados para las mediciones, ha dado como resultado un sistema integral y funcional.

No obstante, se destaca la importancia de considerar estos logros en el contexto de un proyecto universitario, reconociendo las limitaciones inherentes a un entorno académico. Aunque los resultados son prometedores, se insta a futuras investigaciones y aplicaciones prácticas para validar la efectividad del prototipo en situaciones del mundo real.

En suma, el prototipo de monitorización de signos vitales demuestra ser una contribución valiosa al campo de la telemedicina y el monitoreo remoto de pacientes. Su capacidad para proporcionar mediciones precisas y su integración exitosa con tecnologías como Node-RED y Raspberry Pi resaltan su potencial en aplicaciones médicas diversas, promoviendo la innovación y la mejora continua en el ámbito de la salud.

6. CONCLUSIONES

El desarrollo exitoso de este prototipo de monitorización de signos vitales representa un avance significativo en el campo de la telemedicina y el seguimiento remoto de pacientes. La combinación de una termocupla, una sonda de oximetría y un brazalete de presión arterial permitió la medición aceptable de la temperatura corporal, los niveles de oxígeno en la sangre y la presión arterial, respectivamente. La implementación de un panel de control en Node-RED proporciona un acceso conveniente a los datos de los signos vitales desde cualquier dispositivo con conexión a internet. Esto facilita la monitorización remota y la telemedicina. El prototipo ofrece una solución viable para el seguimiento exhaustivo de pacientes en situaciones de aislamiento, como evidenciado durante la pandemia de COVID-19. Además, su capacidad para la implementación de telemedicina brinda oportunidades valiosas para el personal médico.

7. RECOMENDACIONES

Limpiar regularmente sensores y dispositivos de medición, recordar la importancia de mantener la privacidad de los datos del paciente, proporcionar pautas claras para verificar la precisión de los sensores según sea necesario. Mantener una fuente de alimentación estable para evitar interrupciones durante la monitorización, realizar el almacenamiento adecuado del equipo cuando no esté en uso. Manejar instrucciones claras sobre qué hacer en situaciones de emergencia o mal funcionamiento del equipo.

8. REFERENCIAS BIBLIOGRÁFICAS

A. Ruiz, P. Córdova and C. Gordón. (2018). Telemedicine System to Avoid Sudden Death Syndrome by Continuous Monitoring of Vital Signs. : 2018 International Conference on eDemocracy & eGovernment (ICEDEG).

Agus Kurniawan. (2016). Getting Started with Raspberry Pi 3. : Agus Kurniawan.

Arduino. (2024). Arduino nano Datasheet. : Arduino.

Cisneros Caicedo, A. J., Guevara García, A. F., Urdánigo Cedeño, J. J., & Garcés Bravo, J. E. (2022). Técnicas e instrumentos para la recolección de datos que apoyan a la investigación científica en tiempo de pandemia. *Dominio de las Ciencias*, 8(1), 1165–1185. <https://doi.org/10.23857/dc.v8i1.2546>

D. Amaya, D. Meléndez, M. Marín and J. Cubías. (2023). Construction and Design of a Didactic Prototype of a Vital Signs Monitor programmed with LabVIEW. : 2023 IEEE Central America and Panama Student Conference (CONESCAPAN).

D. Lu, J. Wu, Y. Sheng, P. Liu and M. Yang. (2020). Analysis of the popularity of programming languages in open source software communities. : 2020 International Conference on Big Data and Social Sciences (ICBDSS).

Ewa Grodzinsky; Marta Sund Levander. (2020). Understanding Fever and Body Temperature. Suiza: Springer Nature.

F. A. Kazan, H. Terzioglu and A. C. Agacayak. (2015). The Design of a Test & Development Board for the Training of PIC18F4550 Microcontroller. Shanghai, China: 2015 2nd International Conference on Information Science and Control Engineering.

F. Cuckov et al. (2021). Towards a Reconfigurable Cyber-Physical Systems Framework for Rapid Development of Scalable Next-Generation Smart Medical Devices. : 2021 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART).

F. L. Oliveira and J. C. B. Mattos. (2022). JSGuide: A Tool to Improve JavaScript Algorithms Focusing on IoT Devices. : 2022 Symposium on Internet of Things (SIoT).

Fernando E. Valdes-Perez; Ramon Pallas-Areny. (2009). Microcontrollers Fundamentals and applications with PIC. 6000 Broken Sound Parkway NW, Suite 300: Taylor & Francis Group.

Fojo, F. J. (2020). Viejas y Nuevas Pandemias: Un Recorrido Por la Historia. UnosOtrosEdiciones.

I. Morales, L. M. Hornak, A. Solari and J. Oreggioni. (2023). Respiratory rate estimation on embedded system. : IEEE Embedded Systems Letters.

J G Webster. (1997). Design of Pulse Oximeters. Londres: IOP Publishing Ltd.

MICROCHIP. (2009). PIC18F4550 Data Sheet. U.S.A: MICROCHIP.

Taiji Hagino. (2021). Practical Node-RED Programing. Livery Place 35 Livery Street Birmingham B3 2PB, UK.: Pack Publishing Ltd.

V. Tank, S. Warriar and N. Jakhiya. (2017). Medicine dispensing machine using raspberry pi and arduino controller. : 2017 Conference on Emerging Devices and Smart Systems (ICEDSS).

Westat, Inc. (1989). PULSE AND BLOOD PRESSURE PROCEDURES. 1650 Research Boulevard Rockville, MD 20850: Westat, Inc.

Y. Wang, A. Poorfakhraei, M. Narimani and A. Emadi. (2021). A 70kW 3-Level Active Neutral Point Clamped Traction Inverter PCB Design for Stray Inductance and Thermal Performance Optimization. : IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society.

Welcome to real digital. (s.f.). Welcome to Real Digital.
<https://www.realdigital.org/doc/7d93df01f5ae9bcb7a0e6859badee09c>


```
B11000,  
B00100,  
B01000,  
B10000,  
B11100  
},  
{  
B00000,  
B00100,  
B01010,  
B00010,  
B00100,  
B00100,  
B00000,  
B00100  
}  
};
```

```
void setup() {  
  Serial.begin(9600);  
  Serial.flush();  
  pinMode(sensorPin, INPUT);  
  pinMode(REDLed, OUTPUT);  
  pinMode(IRLed, OUTPUT);
```

```
// Inicializar la pantalla LCD  
lcd.init();  
lcd.backlight();
```

```
// Apagar LEDs  
digitalWrite(REDLed, LOW);  
digitalWrite(IRLed, LOW);
```

```
for (int i = 0; i < 8; i++) lcd.createChar(i, sym[i]);  
}
```

```
void loop()  
{  
  bool finger_status = true;
```

```
float readsIR[samp_siz], sumIR, lastIR, reader, start;  
float readsRED[samp_siz], sumRED, lastRED;
```

```
int period, samples;  
period = 0;  
samples = 0;  
int samplesCounter = 0;
```

```

float readsIRMM[maxperiod_siz], readsREDMM[maxperiod_siz];
int ptrMM = 0;
for (int i = 0; i < maxperiod_siz; i++) {
readsIRMM[i] = 0;
readsREDMM[i] = 0;
}
float IRmax = 0;
float IRmin = 0;
float REDmax = 0;
float REDmin = 0;
double R = 0;

float measuresR[measures];
int measuresPeriods[measures];
int m = 0;
for (int i = 0; i < measures; i++) {
measuresPeriods[i] = 0;
measuresR[i] = 0;
}

int ptr;

float beforeIR;

bool rising;
int rise_count;
int n;
long int last_beat;
for (int i = 0; i < samp_siz; i++) {
readsIR[i] = 0;
readsRED[i] = 0;
}
sumIR = 0;
sumRED = 0;
ptr = 0;

while (1)
{
// Encender el LED IR
digitalWrite(REDLed, LOW);
digitalWrite(IRLed, HIGH);

// Calcular un promedio del sensor durante un período de 20 ms (T)
n = 0;
start = millis();
reader = 0.0;
do {

```

```

reader += analogRead(sensorPin);
n++;
} while (millis() < start + T);
reader /= n; // Se obtiene el promedio
sumIR -= readsIR[ptr];
sumIR += reader;
readsIR[ptr] = reader;
lastIR = sumIR / samp_siz;

// Encender el LED ROJO y hacer lo mismo
digitalWrite(REDLed, HIGH);
digitalWrite(IRLed, LOW);

n = 0;
start = millis();
reader = 0.0;
do {
reader += analogRead(sensorPin);
n++;
} while (millis() < start + T);
reader /= n;
sumRED -= readsRED[ptr];
sumRED += reader;
readsRED[ptr] = reader;
lastRED = sumRED / samp_siz;

// Cálculo de R
readsIRMM[ptrMM] = lastIR;
readsREDMM[ptrMM] = lastRED;
ptrMM++;
ptrMM %= maxperiod_siz;
samplesCounter++;

if (samplesCounter >= samples) {
samplesCounter = 0;
IRmax = 0;
IRmin = 1023;
REDmax = 0;
REDmin = 1023;
for (int i = 0; i < maxperiod_siz; i++) {
if (readsIRMM[i] > IRmax) IRmax = readsIRMM[i];
if (readsIRMM[i] > 0 && readsIRMM[i] < IRmin ) IRmin = readsIRMM[i];
readsIRMM[i] = 0;
if (readsREDMM[i] > REDmax) REDmax = readsREDMM[i];
if (readsREDMM[i] > 0 && readsREDMM[i] < REDmin ) REDmin =
readsREDMM[i];
readsREDMM[i] = 0;

```

```

}
R = ((REDmax - REDmin) / REDmin) / ((IRmax - IRmin) / IRmin);
}

if (lastRED < lastIR) {
if (finger_status == true) {
finger_status = false;
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("No finger?");
Serial.println("No finger?");
}
} else {
if (finger_status == false) {
lcd.clear();
finger_status = true;

lcd.setCursor(10, 0);
lcd.print("c=");
lcd.setCursor(0, 0);
lcd.print("bpm");
lcd.setCursor(0, 1);
lcd.print("SpO"); lcd.write(1);
lcd.setCursor(10, 1);
lcd.print("R=");
}
}

float avR = 0;
avBPM = 0;

if (finger_status == true) {
if (lastIR > beforeIR) {
rise_count++;
if (!rising && rise_count > rise_threshold) {
lcd.setCursor(3, 0);
lcd.write(0);

rising = true;
measuresR[m] = R;
measuresPeriods[m] = millis() - last_beat;
last_beat = millis();
int period = 0;
for (int i = 0; i < measures; i++) period += measuresPeriods[i];
period = period / measures;
samples = period / (2 * T);
}
}
}

```

```

int avPeriod = 0;

int c = 0;
for (int i = 1; i < measures; i++) {
  if ((measuresPeriods[i] < measuresPeriods[i-1] * 1.1) &&
      (measuresPeriods[i] > measuresPeriods[i-1] / 1.1)) {
    c++;
    avPeriod += measuresPeriods[i];
    avR += measuresR[i];
  }
}

m++;
m %= measures;

lcd.setCursor(12, 0);
lcd.print(String(c) + " ");

avBPM = (60000 / (avPeriod / c) + 2);
avR = avR / c;

lcd.setCursor(12, 1);
if (c == 0) {
  lcd.print(" ");
} else {
  lcd.print(String(avR) + " ");
}

if (c > 4) {
  SpO2 = -19 * R + 121;
  lcd.setCursor(4, 0);
  if (avBPM > 40 && avBPM < 220) {
    lcd.print(String(avBPM) + " ");
  } else {
    lcd.print("---");
  }
}

lcd.setCursor(4, 1);
if (SpO2 > 20 && SpO2 < 100) {
  Serial.println(String(SpO2) + "%");
  lcd.print(" " + String(SpO2) + "% ");
} else {
  lcd.print("--% ");
}
} else {
  if (c < 3) {
    lcd.setCursor(3, 0); lcd.write(2); lcd.print("---");
  }
}

```

```
lcd.setCursor(4, 1); lcd.write(2); lcd.print("----");
}
}
}
} else {
rising = false;
rise_count = 0;
lcd.setCursor(3, 0);
lcd.print(" ");
}
beforeIR = lastIR;
} // dedo está dentro

delay(60);
ptr++;
ptr %= samp_siz;
} // bucle while 1
}
```

Presión Arterial

```
#include <Wire.h> // Librería para la comunicación I2C en Arduino.
```

```
int incomingByte = 0; // Variable para datos seriales entrantes.
```

```
// Definiciones de MACROS
```

```
#define BP_START_PIN (2) // Botón de inicio del dispositivo de medición de presión arterial.
```

```
#define VALVE_PIN (3) // Indica si la medición ha terminado.
```

```
#define MEASURE_BEGIN_PIN (4) // Indica que debe comenzar una medición. Puede conectarse a un interruptor o a otro microcontrolador (MCU) o Raspberry Pi.
```

```
volatile byte i2c_data_rx; // Indica que hay datos disponibles desde el bus I2C.
```

```
volatile uint16_t count; // Indica el número total de datos recopilados.
```

```
volatile uint8_t sys, dia, hr; // Almacena los valores de medida: sistólica, diastólica y frecuencia cardíaca.
```

```
void setup()
```

```
{
```

```
Serial.begin(115200); // Inicia la comunicación serial a 115200 baudios.
```

```
pinMode(BP_START_PIN, OUTPUT); // Configura el pin de inicio como salida.
```

```
pinMode(VALVE_PIN, INPUT); // Configura el pin de la válvula como entrada.
```

```
pinMode(MEASURE_BEGIN_PIN, INPUT_PULLUP); // Configura el pin de inicio de medición como entrada con resistencia de pull-up.
```

```
Wire.begin(0x50); // La dirección del EEPROM es 0x50. El Arduino debe tener la misma dirección.
```

```
Wire.onReceive(receiveEvent); // Inicialización de la interrupción para datos I2C.
```

```
}
```

```
void loop()
```

```
{
```

```
if (Serial.available() > 0) {
```

```
  // Lee el comando desde la comunicación serial
```

```
  String comando = Serial.readStringUntil('\n');
```

```
  if (digitalRead(MEASURE_BEGIN_PIN) == 0 || comando == "true") // El Arduino recibe la instrucción de comenzar la medición.
```

```
  {
```

```
    digitalWrite(BP_START_PIN, HIGH); // Simula presionar el botón.
```

```
    delay(200);
```

```
    digitalWrite(BP_START_PIN, LOW);
```

```
    delay(2000);
```

```
    // Espera a que se cierre la válvula.
```

```

while (digitalRead(VALVE_PIN) == 0)
{
delay(1000);
}
delay(2000);

// Simula presionar nuevamente el botón.
digitalWrite(BP_START_PIN, HIGH);
delay(200);
digitalWrite(BP_START_PIN, LOW);

delay(500);

if (count == 0)
{
Serial.println("Sin datos");
}
else if (count == 35)
{
Serial.print(sys);
Serial.print("/");
Serial.print(dia);
Serial.print("/");
Serial.println(hr);
}
else
{
Serial.println("error");
}

count = 0;
}
}
}

void receiveEvent(int iData) // Rutina de servicio de interrupción.
{
if (iData > 0)
{
while (iData--)
{
i2c_data_rx = Wire.read();
count++;

// Almacena valores en variables específicas según el número de datos
recibidos.
if (count == 28)

```

```
{
sys = i2c_data_rx;
}
if (count == 29)
{
dia = i2c_data_rx;
}
if (count == 30)
{
hr = i2c_data_rx;
}
}
}
}
```

Temperatura Corporal

```
#include <18f4550.h>
#fuses NOWDT , HS, NOPROTECT, CPUDIV1, PLL1
#use delay(clock=20000000)
#use rs232(baud=9600, parity=N, xmit=pin_c6, rcv=pin_c5, bits=8)
#include "MAX6675_Lib.c"
#include <lcd.c>

#define LCD_ENABLE_PIN PIN_D0
#define LCD_RS_PIN PIN_D1
#define LCD_RW_PIN PIN_D2
#define LCD_DATA4 PIN_D4
#define LCD_DATA5 PIN_D5
#define LCD_DATA6 PIN_D6
#define LCD_DATA7 PIN_D7

#define NUM_MEDICIONES 40

int16 mediciones[NUM_MEDICIONES];
int16 Tm; // Definimos las variable de temperatura del sensor como flotante
char msg[32];

// Función de ordenamiento simple (Bubble Sort)
void ordenarMediciones() {
    int16 temp;
    for (int i = 0; i < NUM_MEDICIONES - 1; i++) {
        for (int j = 0; j < NUM_MEDICIONES - i - 1; j++) {
            if (mediciones[j] > mediciones[j + 1]) {
                // Intercambiar elementos si están en el orden incorrecto
                temp = mediciones[j];
                mediciones[j] = mediciones[j + 1];
                mediciones[j + 1] = temp;
            }
        }
    }
}

int16 calcularMediana() {
    // Ordenar el arreglo de mediciones
    ordenarMediciones();

    // Calcular la mediana
    if (NUM_MEDICIONES % 2 == 0) {
        // Número par de elementos, promedio de los dos centrales
    }
}
```

```

return (mediciones[NUM_MEDICIONES / 2 - 1] + mediciones[NUM_MEDICIONES / 2]) /
2;
} else {
// Número impar de elementos, valor central
return mediciones[NUM_MEDICIONES / 2];
}
}

void main() {
set_tris_d(0x00);
lcd_init();

int16 mediana;

do {
for (int i = 0; i < NUM_MEDICIONES; i++) {
Tm = read_termocupla_k();
Tm = Tm - 24;

mediciones[i] = Tm;
}

mediana = calcularMediana();

lcd_gotoxy(1, 1);
printf(lcd_putc, "#####");
lcd_gotoxy(1, 2);
printf(lcd_putc, "# TERMOMETRO #");
lcd_gotoxy(17, 1);
printf(lcd_putc, "# TEMP: %ld C #", mediana);
lcd_gotoxy(17, 2);
printf(lcd_putc, "#####");
sprintf(msg,"%ld\r\n",mediana);
printf("%s",msg);
delay_ms(250);

} while (1);
}

```

Código Python

```
# Datos de temperatura obtenidos del prototipo y del equipo comercial
temperatura_prototipo = [36, 36, 37, 35, 36] # Temperatura en
grados Celsius
temperatura_comercial = [37, 35.2, 36, 36, 37] # Temperatura en
grados Celsius

# Calcular el error entre las mediciones del prototipo y del
equipo comercial
errores = [abs(tp - tc) for tp, tc in zip(temperatura_prototipo,
temperatura_comercial)]

# Calcular el error promedio
error_promedio = sum(errores) / len(errores)

# Mostrar el error promedio
print(f"Error promedio entre las mediciones del prototipo y del
equipo comercial: {error_promedio:.2f} °C")

import matplotlib.pyplot as plt

# Crear una lista de índices para las pruebas
pruebas = [f"Prueba {i}" for i in range(1, 6)]

# Crear una gráfica de barras para las mediciones del prototipo y
del equipo comercial
plt.figure(figsize=(10, 6))
plt.bar(pruebas, temperatura_prototipo, color='blue', alpha=0.5,
label='Prototipo')
plt.bar(pruebas, temperatura_comercial, color='green', alpha=0.5,
label='Equipo Comercial')
plt.xlabel('Pruebas')
plt.ylabel('Temperatura (°C)')
plt.title('Comparación de Temperaturas entre Prototipo y Equipo
Comercial')
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()

# Mostrar la gráfica
plt.show()

# Crear una gráfica de barras para visualizar el error entre las
mediciones
```

```

plt.figure(figsize=(10, 6))
plt.bar(pruebas, errores, color='red', alpha=0.7)
plt.xlabel('Pruebas')
plt.ylabel('Error (°C)')
plt.title('Error entre Mediciones del Prototipo y Equipo
Comercial')
plt.xticks(rotation=45)
plt.tight_layout()

# Mostrar la gráfica
plt.show()

```

#####

```

import numpy as np

# Datos de saturación de oxígeno
saturacion_prototipo = [94, 93, 95, 94, 96]
saturacion_comercial = [95, 92, 97, 96, 95]

# Cálculo del error absoluto
error_absoluto = np.abs(np.array(saturacion_prototipo) -
np.array(saturacion_comercial))

# Cálculo del error relativo porcentual
error_relativo_porcentual = (error_absoluto /
np.array(saturacion_comercial)) * 100

# Cálculo del error promedio
error_promedio = np.mean(error_absoluto)

# Impresión de los resultados
print("Error absoluto:", error_absoluto)
print("Error relativo porcentual:", error_relativo_porcentual)
print("Error promedio:", error_promedio)

```

#####

```

import numpy as np
import matplotlib.pyplot as plt

# Gráfico de comparación
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(saturacion_prototipo) + 1),
saturacion_prototipo, marker='o', label='Prototipo')
plt.plot(range(1, len(saturacion_comercial) + 1),
saturacion_comercial, marker='o', label='Equipo Comercial')

```

```

plt.title('Comparación de Saturación de Oxígeno')
plt.xlabel('Pruebas')
plt.ylabel('Saturación de Oxígeno (%)')
plt.xticks(range(1, len(saturacion_prototipo) + 1))
plt.legend()
plt.grid(True)
plt.show()

# Gráfico de error
plt.figure(figsize=(10, 5))
plt.plot(range(1, len(error_absoluto) + 1), error_absoluto,
marker='o', color='red')
plt.title('Error Absoluto de Saturación de Oxígeno')
plt.xlabel('Pruebas')
plt.ylabel('Error Absoluto')
plt.xticks(range(1, len(error_absoluto) + 1))
plt.grid(True)
plt.show()

#####

import numpy as np
import matplotlib.pyplot as plt

# Datos de presión arterial del prototipo y del equipo comercial
presion_prototipo = [118, 138, 116, 106, 150]
presion_comercial = [112, 134, 110, 101, 147]

# Calcular el error
error = np.abs(np.array(presion_prototipo) -
np.array(presion_comercial))

# Imprimir el error promedio
error_promedio = np.mean(error)
print("Error promedio:", error_promedio)

# Realizar la comparación de presión arterial entre el prototipo y el
equipo comercial
plt.figure(figsize=(10, 6))
plt.plot(presion_prototipo, label='Prototipo', marker='o')
plt.plot(presion_comercial, label='Equipo comercial', marker='x')
plt.title('Comparación de presión arterial entre el prototipo y el
equipo comercial')
plt.xlabel('Prueba')
plt.ylabel('Presión arterial (mmHg)')
plt.legend()

```

```
plt.grid(True)
plt.show()
```

```
# Graficar el error
```

```
plt.figure(figsize=(10, 6))
plt.plot(error, marker='o', color='red')
plt.title('Error en la medición de presión arterial')
plt.xlabel('Prueba')
plt.ylabel('Error (mmHg)')
plt.grid(True)
plt.show()
```