



UNIVERSIDAD TÉCNICA DE AMBATO

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

Tema:

**SISTEMA DE MONITOREO NO INVASIVO Y SEGURIDAD
PREVENTIVA DE FACTORES CAUSANTES DEL SÍNDROME DE
MUERTE SÚBITA DE LOS LACTANTES DEL CENTRO DE
ESTIMULACIÓN TEMPRANA Y PRENATAL “NEURO PLACE” A
TRAVÉS DE SMART TEXTIL**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Telecomunicaciones

ÁREA: Comunicaciones

LÍNEA DE INVESTIGACIÓN: Tecnología de la Información y
Sistemas de Control

AUTOR: John David Lema Yungán

TUTOR: Dr. Freddy Geovanny Benalcázar Palacios

**Ambato – Ecuador
agosto - 2023**

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: SISTEMA DE MONITOREO NO INVASIVO Y SEGURIDAD PREVENTIVA DE FACTORES CAUSANTES DEL SÍNDROME DE MUERTE SÚBITA DE LOS LACTANTES DEL CENTRO DE ESTIMULACIÓN TEMPRANA Y PRENATAL “NEURO PLACE” A TRAVÉS DE SMART TEXTIL, desarrollado bajo la modalidad Proyecto de Investigación por el señor John David Lema Yungán, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2023.

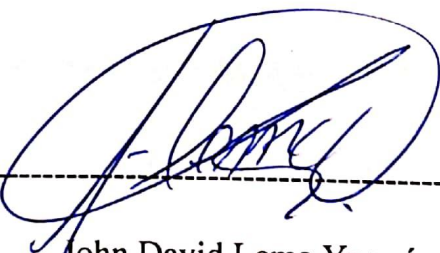
Dr. Freddy Geovanny Benalcázar Palacios

TUTOR

AUTORÍA

El presente trabajo de titulación titulado: SISTEMA DE MONITOREO NO INVASIVO Y SEGURIDAD PREVENTIVA DE FACTORES CAUSANTES DEL SÍNDROME DE MUERTE SÚBITA DE LOS LACTANTES DEL CENTRO DE ESTIMULACIÓN TEMPRANA Y PRENATAL “NEURO PLACE” A TRAVÉS DE SMART TEXTIL es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2023



John David Lema Yungán

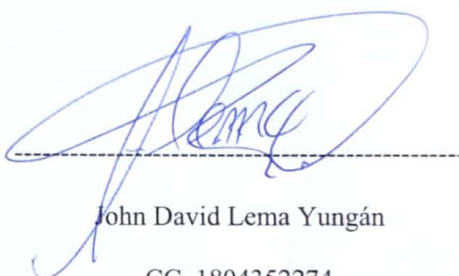
CC. 1804352274

AUTOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, agosto 2023



John David Lema Yungán

CC. 1804352274

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor John David Lema Yungán, estudiante de la Carrera de Telecomunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad de Proyecto de Investigación titulado SISTEMA DE MONITOREO NO INVASIVO Y SEGURIDAD PREVENTIVA DE FACTORES CAUSANTES DEL SÍNDROME DE MUERTE SÚBITA DE LOS LACTANTES DEL CENTRO DE ESTIMULACIÓN TEMPRANA Y PRENATAL “NEURO PLACE” A TRAVÉS DE SMART TEXTIL, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2023.

Ing. Elsa Pilar Urrutia Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Mg. Martha Esperanza Sevilla Abarca
PROFESOR CALIFICADOR

Ing. Mg. Santiago Mauricio Altamirano Meléndez
PROFESOR CALIFICADOR

DEDICATORIA

Este trabajo de investigación está dedicado a Dios, quien ha sido un constante faro de bondad en mi camino. Expreso mi profundo agradecimiento a mi familia, en particular a mi padre, José, y mi madre, María. Ellos han sido los pilares fundamentales que han guiado mis pasos y han infundido en mí los valores que me han moldeado no solo como profesional, sino también como persona íntegra. Quiero extender mi reconocimiento a mis hermanos, en especial a Cristian, y a mi sobrino Danny, quienes siempre han depositado una confianza inquebrantable en mí. Este logro es también un homenaje a esa confianza que han tenido en mi

John

AGRADECIMIENTO

"Agradezco a Dios por el milagro de la vida y por otorgarme el don de la ciencia y la sabiduría, elementos fundamentales en la consecución de mis objetivos.

A mi padre, José, y mi madre, María, les dedico mi más profundo agradecimiento por el amor incondicional y el apoyo constante que me han brindado a lo largo de mi trayectoria. Su respaldo ha sido un pilar fundamental en cada paso que he dado en mi carrera.

Agradezco a mis hermanos, quienes han sido fuente de consejos y aliento constante, nunca permitiendo que claudicara en mi recorrido universitario.

Mis amigos han sido pilares invaluableles en mi camino. Especialmente, quiero expresar mi gratitud a Alexis y Francis, compañeros de carrera con quienes he compartido todas las vivencias, tanto las exitosas como las desafiantes. Sin su apoyo, el culminar mi carrera no hubiera sido posible.

Jhonatan, Gerson, Michelle y Tamia, queridos amigos que han sido un apoyo inquebrantable más allá de los días universitarios, merecen un lugar muy especial en mi corazón. Sus palabras de aliento, siempre presentes, han sido como rayos de luz en los momentos más oscuros y complicados. Su amistad genuina ha sido un regalo invaluable que atesoro con profundo agradecimiento.

A mis compañeros de toda la vida, Nicolás, Josu, Raphael y Cristian, les agradezco por nuestra inquebrantable amistad. Siempre estuvieron a mi lado en los momentos complejos, demostrando que juntos enfrentamos cualquier desafío que se presentara."

John

TABLA DE CONTENIDO

APROBACIÓN DEL TUTOR	ii
AUTORÍA	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO	v
Dedicatoria.....	vi
Agradecimiento.....	vii
RESUMEN EJECUTIVO	xiii
ABSTRACT.....	xv
CAPÍTULO I	1
MARCO TEÓRICO	1
1.1. Antecedentes investigativos	3
1.2. Contextualización del problema.....	1
1.3. Fundamentación teórica	6
1.3.1 Muerte Súbita.....	6
1.3.2 Factores Asociados a la Muerte Súbita	7
1.3.3 Frecuencia Cardíaca.....	8
1.3.4 Saturación de oxígeno.....	9
1.3.5 Tecnologías de la Información.....	10
1.3.6 Sistemas electrónicos de adquisición de datos y el procesamiento de datos.....	10
1.3.7 Baterías LiPo.....	13
1.3.8 Frameworks para la creación de aplicaciones móviles	14
1.3.9 Backend.....	16
1.4. Objetivos	19
1.4.1. Objetivo General.....	19
1.4.2. Objetivos Específicos.....	19
CAPÍTULO II	20
METODOLOGÍA	20
2.1. Materiales.....	20
2.2. Métodos.....	21
2.2.1 Modalidad de investigación	21

2.2.2 Recolección de información.....	22
2.2.3 Procesamiento y análisis de datos	22
2.2.4 Desarrollo de Proyecto.....	22
CAPÍTULO III.....	24
RESULTADOS Y DISCUSIÓN	24
3.1. Análisis y discusión de los resultados	24
3.2. Desarrollo de la propuesta.....	24
3.2.1 Arquitectura del sistema.....	25
3.2.2 Diseño del sistema	30
3.2.3 Implementación de Textil	35
3.2.4 Procesamiento y envío de señales	37
3.2.5 Desarrollo de aplicación móvil	55
3.2.6 Prueba de Funcionamiento.....	80
3.2.7 Presupuesto	90
CAPÍTULO IV	92
CONCLUSIONES Y RECOMENDACIONES	92
4.1. Conclusiones	92
4.2. Recomendaciones.....	94
BIBLIOGRAFÍA	95
ANEXOS	101

ÍNDICE DE FIGURAS

Figura 1 Cinco pasos propuestos en vía respiratoria terminal asociada al síndrome de	7
Figura 2 Arquitectura del sistema	25
Figura 3 Conexión I2C Maestro-Eslavos	27
Figura 4 Venas disponibles en la pierna de un lactante	31
Figura 5 Diseño esquemático	32

Figura 6	Diseño de PCB.....	33
Figura 7	Circuito 3D esquemático	33
Figura 8	Medidas del armazón del circuito y de la batería	34
Figura 9	Implementación del circuito en el armazón.....	35
Figura 10	Microfibra.....	35
Figura 11	Implementación de Velcro.....	36
Figura 12	Vista superior e Inferior del Smart Textil.....	37
Figura 13	Diagrama de flujo de la función de la frecuencia cardiaca.....	39
Figura 14	Diagrama de flujo de la función del nivel de Oxígeno en la Sangre	41
Figura 15	Posición de descanso de un lactante	47
Figura 16	Posiciones de detección de un MPU6050	47
Figura 17	Diagrama de flujo de la función de MPU6050.....	48
Figura 18	Batería de polímero de litio con cable conector JST	51
Figura 19	Ejemplo de la estructura JSON.....	53
Figura 20	Arquitectura de las funciones del aplicativo móvil	56
Figura 21	Flujograma de la aplicación Móvil.....	57
Figura 22	Creacion de proyecto en Firebase.....	58
Figura 23	Carpeta creada del proyecto de Flutter	59
Figura 24	Contenido de un proyecto en Flutter	59
Figura 25	Pantalla de Registro/Edición de datos	62
Figura 26	Ejemplo de creación de un nodo de Usuario	64
Figura 27	Datos del usuario en Firebase	65
Figura 28	Pantalla de Inicio de la app.....	66
Figura 29	Pantalla de inicio sin base de datos.....	67
Figura 30	Visualización de dato en tiempo real.....	68
Figura 31	Base de datos, almacenada en Firebase	71
Figura 32	Botón de Terminar Medición.....	72
Figura 33	Valores almacenados en base de Datos	74
Figura 34	Estructura de la base de datos de los Usuarios	74
Figura 35	Vista de datos generales.....	76
Figura 36	Pantalla de los datos de lactantes.....	79
Figura 37	Registro de base de datos de un usuario	80
Figura 38	Vista de datos.....	81

Figura 39 Implementación de Smart Textil	82
Figura 40 Dispositivo conectado a la Red.....	83
Figura 41 Proceso de muestreo de datos del Smart Textil	83
Figura 42 Comparativa de datos con Oxímetro comercial.....	84
Figura 43 Datos en aplicación en tiempo real	84
Figura 44 Comparativa de la frecuencia cardiaca entre el Smart Textil y el oxímetro GL320	85
Figura 45 Comparativa de oxígeno en la sangre entre el Smart Textil y el oxímetro GL320	86
Figura 46 Notificaciones de la aplicación.....	88

ÍNDICE DE TABLAS

Tabla 1 Defunciones de Mortalidad Infantil en Región Sierra	2
Tabla 2 Frecuencia Cardíaca por Rango de edades	9
Tabla 3 Comparativa Microcontroladores.....	11
Tabla 4 Comparación de Sensores de Ritmo Cardíaco y Oxigenación en la sangre. 12	
Tabla 5 Comparación de Acelerómetros	13
Tabla 6 Comparación de Frameworks.....	15
Tabla 7 Comparativa SQL y NoSQL	17
Tabla 8 Comparativa de servicio Backend.....	18
Tabla 9 Pines para la comunicación I2C	31
Tabla 10 Error Porcentual para las mediciones del Ritmo Cardíaco	44
Tabla 11 Error Porcentual para las mediciones de nivel de Oxígeno en la Sangre... 45	
Tabla 12 Estado del porcentaje de oxigenación en la sangre	77
Tabla 13 Datos del lactante	81
Tabla 14 Gasto energético de los componentes del Smart Textil.....	89
Tabla 15 Presupuesto de construcción	91
Tabla 16 Presupuesto total de proyecto.....	91

RESUMEN EJECUTIVO

El presente proyecto se enfoca en el desarrollo de un Smart Textil innovador para el monitoreo de signos vitales en lactantes, con el objetivo de prevenir el síndrome de muerte súbita. El Smart Textil, diseñado específicamente para el Centro de Estimulación Temprana y Prenatal "Neuro Place", será utilizado por los profesionales de este centro para el monitoreo de los lactantes. El proyecto tiene como objetivo proporcionar una solución tecnológica confiable y efectiva para el monitoreo continuo de los signos vitales de los lactantes a través de un tejido inteligente.

Durante el proceso de desarrollo de este proyecto, se empleó el microcontrolador ESP32 Feather como la unidad central encargada de controlar el sensor GY-MAX30102 y el sensor MPU-6050 dentro del Smart Textil. El sensor GY-MAX30102, basado en el principio de fotopletismografía, permite medir de manera precisa el ritmo cardíaco y la saturación de oxígeno en la sangre del lactante. Por otro lado, el sensor MPU-6050, equipado con un acelerómetro, se utiliza para identificar y rastrear la posición del lactante. Además, se incorporó una pantalla OLED con tecnología I2C en el Smart Textil, lo cual permite la visualización en tiempo real de los datos recopilados. En cuanto a la transmisión de datos, se utilizó tecnología Wifi, cuyo módulo está integrado en el microcontrolador elegido, para enviar los datos recolectados desde el microcontrolador hacia la nube. Para facilitar la interacción con el Smart Textil, se desarrolló una aplicación móvil utilizando el Framework Flutter. Esta aplicación móvil actúa como interfaz entre los encargados del centro de estimulación y el Smart Textil, permitiéndoles registrar al lactante, realizar mediciones de los signos vitales, acceder a los registros de datos y verificar el estado de la batería del Smart Textil, usando la plataforma de Firebase como backend y el desarrollo de scripts específicos, se habilitó la capacidad de enviar alertas al dispositivo móvil a través de notificaciones push en caso de detectar anomalías en los signos vitales del lactante. Esto garantiza una respuesta rápida y eficiente ante situaciones de riesgo, brindando a los encargados del centro de estimulación la oportunidad de tomar medidas adecuadas y oportunas.

Finalmente se obtiene un porcentaje de confiabilidad en las mediciones de frecuencia cardíaca fue del 98.02%, mientras que el porcentaje de confiabilidad en las mediciones

de saturación de oxígeno en la sangre fue del 98.27%. Estos valores reflejan la precisión y la fiabilidad de las mediciones realizadas a través del Smart Textil. La autonomía de la batería del Smart Textil es de aproximadamente 12.41 horas, lo que proporciona un tiempo de monitoreo prolongado. Estos resultados respaldan el uso efectivo del Smart Textil como una herramienta confiable para el monitoreo de signos vitales en lactantes y la prevención del síndrome de muerte súbita.

Palabras claves: Smart textil, muerte súbita, ritmo cardíaco, saturación de oxígeno, aplicación móvil

ABSTRACT

This project focuses on the development of an innovative Smart Textile for monitoring vital signs in infants, with the goal of preventing sudden infant death syndrome. The Smart Textile, specifically designed for the "Neuro Place" Early and Prenatal Stimulation Center, will be used by professionals at the center to monitor the infants. The project aims to provide a reliable and effective technological solution for continuous monitoring of infants' vital signs through smart fabric.

During the development process of this project, the ESP32 Feather microcontroller was utilized as the central unit responsible for controlling the GY-MAX30102 sensor and the MPU-6050 sensor within the Smart Textile. The GY-MAX30102 sensor, based on photoplethysmography principles, accurately measures the infant's heart rate and blood oxygen saturation. On the other hand, the MPU-6050 sensor, equipped with an accelerometer, is used to identify and track the infant's position. Additionally, an OLED display with I2C technology was incorporated into the Smart Textile, enabling real-time visualization of the collected data. For data transmission, Wi-Fi technology was employed, with the module integrated into the chosen microcontroller, to send the collected data from the microcontroller to the cloud. To facilitate interaction with the Smart Textile, a mobile application was developed using the Flutter Framework. This mobile application serves as an interface between the staff at the stimulation center and the Smart Textile, allowing them to register the infant, measure vital signs, access data records, and check the battery status of the Smart Textile. By utilizing the Firebase platform as the backend and developing specific scripts, the capability to send alerts to the mobile device through push notifications was enabled in case any anomalies in the infant's vital signs are detected. This ensures a prompt and efficient response to risky situations, providing the staff at the stimulation center with the opportunity to take appropriate and timely actions.

Finally, a reliability percentage of 98.02% was achieved for heart rate measurements, while the reliability percentage for blood oxygen saturation measurements was 98.27%. These values reflect the accuracy and dependability of the measurements conducted through the Smart Textile. The Smart Textile's battery autonomy is approximately 12.41 hours, offering ample operational time.

Keywords: Smart textile, sudden death, heart rate, oxygen saturation, mobile application.

CAPITULO I

MARCO TEÓRICO

1.1 Tema de Investigación

SISTEMA DE MONITOREO NO INVASIVO Y SEGURIDAD PREVENTIVA DE FACTORES CAUSANTES DEL SÍNDROME DE MUERTE SÚBITA DE LOS LACTANTES DEL CENTRO DE ESTIMULACIÓN TEMPRANA Y PRENATAL “NEURO PLACE” A TRAVÉS DE SMART TEXTIL

1.1.1 Contextualización del problema

En los países desarrollados, el Síndrome de Muerte Súbita del Lactante (SIDS) es la causa principal de la muerte infantil en bebés de entre uno y doce meses de edad. [1] Los factores genéticos y ambientales son los principales implicados en esta triste situación. A nivel mundial, la incidencia del SIDS varía de un país a otro, oscilando entre 0.31 y 3.5 por cada 1000 nacidos vivos. Hasta la década de los 90, la incidencia del SIDS en países occidentales fluctuaba entre 1 y 3 por cada 1000 nacidos vivos, mientras que en Australia y Nueva Zelanda llegaba a ser de hasta 7 por cada 1000 nacidos vivos. Aunque se han producido avances en los países desarrollados para disminuir su incidencia, el SIDS sigue siendo una de las principales causas de muerte en bebés lactantes [2].

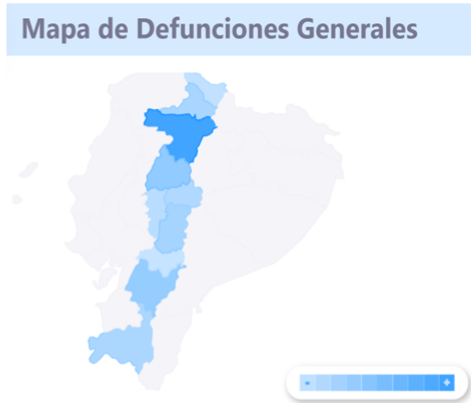
En Latinoamérica, el cuidado de los niños no siempre alcanza los mismos estándares de calidad que en los países desarrollados. Como resultado, se registra una tasa de Síndrome de Muerte Súbita del Lactante (SIDS) más alta en la región. En México, por ejemplo, la tasa de SIDS es de aproximadamente 0.6 por cada 1000 lactantes [3]. INEC reporta que el 5.8% de las muertes en niños menores de 4 años se deben a dificultades respiratorias, ahogamiento, altas temperaturas y alteraciones repentinas del ritmo cardíaco que no recibieron tratamiento a tiempo, factores que pueden contribuir al desarrollo de SIDS [4].

Actualmente, el cuidado de las personas ha evolucionado gracias al avance de la tecnología. Sin embargo, en los países de Latinoamérica, el acceso a las tecnologías actuales no está garantizado al 100%. Esto se debe a diversos factores, como la falta de investigación, la limitación de recursos y la escasez de laboratorios tecnológicos equipados para realizar pruebas. En el caso de Ecuador, el uso de IoT aún no se ha explorado en su totalidad y no está completamente desarrollado. Las instituciones que han intentado utilizar esta tecnología se limitan exclusivamente a los hospitales y requieren equipos costosos que solo están disponibles en grandes ciudades como Guayaquil y Quito. [5]

El INEC en el 2021, la región Sierra de Ecuador presento un total de 1290 muertes infantiles los principales datos se muestran en la tabla 1:

Tabla 1 Defunciones de Mortalidad Infantil en Región Sierra [6]

Provincias	Mortalidad Infantil
Pichincha	602
Cotopaxi	134
Azuay	124
Chimborazo	93
Imbabura	83
Loja	80
Tungurahua	68
Bolivar	49
Carchi	34
Cañar	23
TOTAL	1290



Considerando que en la región 3 se reportan 1290 casos de mortalidad infantil, de un total de 2655 en todo el país, es relevante resaltar la crucial importancia de prevenir la mortalidad en este grupo de edad.

La muerte súbita es un fenómeno complejo y multifactorial, ya que los lactantes son particularmente vulnerables a las condiciones ambientales que afectan la respiración, la temperatura y el entorno en el que viven. Por lo tanto, es esencial que estos factores sean monitoreados constantemente para garantizar la seguridad y bienestar del niño en los primeros años de vida. La actuación rápida en caso de una situación crítica es vital para el lactante y puede marcar una gran diferencia.

La muerte de un hijo por Síndrome de Muerte Súbita del Lactante es una experiencia devastadora para los padres, ya que suele ser repentina y genera sentimientos de culpa por no haber actuado a tiempo. Además, los efectos psicológicos de este tipo de pérdida pueden ser duraderos y afectar a los padres durante toda su vida. Actualmente, los factores que pueden desencadenar una muerte súbita solo pueden ser monitoreados constantemente por los padres o cuidadores, lo que puede resultar agotador, especialmente durante la noche, cuando los signos vitales del niño son más propensos a sufrir alteraciones.

Desafortunadamente, en la mayoría de los hogares no existen dispositivos que permitan medir los factores asociados al Síndrome de Muerte Súbita del Lactante, ni registrar los signos vitales que permitan a los padres reaccionar rápidamente en caso de cualquier anomalía. Los productos disponibles en el mercado para este propósito suelen ser costosos, por lo tanto, no están al alcance de todas las familias.

1.1 Antecedentes investigativos

Al finalizar la investigación bibliográfica en los principales repositorios de las universidades acerca de instrumentaria Electrónica y de dispositivos de monitoreo de signos vitales en lactantes sin las necesidades de grandes equipos médicos, así como en diversas bases datos se determinaron los siguientes antecedentes.

En marzo de 2017, Alajo Elvis y Pullopaxi Guaman, investigadores de la Universidad de las Fuerzas Armadas ESPE, presentaron un proyecto titulado "Construcción de un medidor cardiorrespiratorio para el monitoreo de señales pletismográficas en lactantes mediante el uso de dispositivos móviles". Este estudio propone el diseño y desarrollo de una solución de monitoreo remoto que captura la onda pletismográfica a través de un dispositivo electrónico de pulso acoplado al dedo del paciente. Para el procesamiento de las señales adquiridas, se emplea una tarjeta PSoC 5LP. Posteriormente, la información es transmitida vía WiFi a un dispositivo móvil tipo smartphone. El monitoreo de las señales se realiza mientras el paciente se encuentra en estado de reposo, específicamente durante el sueño. Este enfoque permite la detección temprana de anomalías cardiorrespiratorias en lactantes. El dispositivo propuesto demostró un margen de error del 2.5%, lo que se traduce en un 98% de

confiabilidad en las mediciones de las condiciones cardiorrespiratorias en lactantes [7].

En noviembre de 2017, Loachamin Mercy y Ribadeneira Verónica, investigadoras de la Escuela Superior Politécnica de Chimborazo, presentaron el proyecto titulado "Diseño e implementación de un prototipo multimodal en telemedicina para emergencia utilizando redes WSN y tecnología móvil para transmitir medidas fisiológicas humanas en tiempo real". Este proyecto propone el uso de redes de sensores inalámbricos (WSN) y tecnología móvil, específicamente el servicio general de paquetes vía radio (GPRS), con el objetivo de mejorar el tiempo de respuesta en la adquisición y transmisión de medidas fisiológicas humanas en tiempo real a profesionales de la salud o centros médicos cercanos para el monitoreo y diagnóstico de pacientes. Para llevar a cabo el proyecto, se empleó una tarjeta de desarrollo Arduino Uno debido a su capacidad analógica, compatible con el módulo E-Health Sensor Shield V2.0. Este módulo permite la realización de aplicaciones biométricas al contar con sensores que miden temperatura, pulso, oxigenación de sangre, entre otros parámetros fisiológicos. Los resultados obtenidos indican una mejora significativa en la eficiencia del prototipo en comparación con los métodos manuales de medición, con tiempos promedio de 1.23 y 5.57 minutos, respectivamente. Esto representa que el prototipo es un 77.87% más rápido que los métodos tradicionales. Además, el dispositivo mostró un nivel de confianza del 98%, resultado que fue evaluado mediante pruebas prácticas y análisis estadísticos durante el proceso de validación [8].

En abril de 2018, Ruiz Narvárez Luis presentó su proyecto titulado "Sistema de telemedicina para monitoreo continuo de constantes vitales en lactantes menores para evitar el síndrome de muerte súbita". El proyecto propone un sistema de telemedicina de uso doméstico diseñado para el monitoreo continuo de signos vitales en lactantes menores, con el objetivo de supervisar, alertar y registrar dichos signos, así como proporcionar información detallada sobre el estado del bebé. El sistema está orientado principalmente a lactantes prematuros, con bajo peso al nacer, recién operados o con problemas congénitos. Para el desarrollo del proyecto, se empleó un microcontrolador Arduino Pro-Micro junto al sensor de oximetría Max30102, seleccionados por su voltaje de operación y precisión. La transmisión de información se realiza mediante

un módulo de comunicación Bluetooth HC-05. El prototipo resultante es adecuado para lactantes de 0 a 6 meses y, en comparación con equipos médicos certificados, ofrece una confiabilidad del 99.1%. Esta alta confiabilidad permite reducir los casos de muerte súbita en lactantes y facilita la continuidad de investigaciones en esta área [9].

En junio de 2019, Md Nakib-Ul Hasan publicó un artículo titulado "Wearable technology for baby monitoring", en el cual destaca la importancia de la portabilidad en sistemas de monitoreo de lactantes, utilizando diferentes medios como movimiento, audio y video. Hasan menciona la existencia de un sensor desarrollado con LilyPad, que emplea tecnología Bluetooth para la conexión con la nube y dispositivos móviles inteligentes, permitiendo la detección de la respiración del bebé, su posición, patrones de sueño y temperatura corporal. La aplicación móvil asociada es compatible con sistemas operativos iOS y Android, y ofrece una interfaz amigable para el seguimiento de signos vitales. La información se transmite directamente a los smartphones conectados a la nube. La estación base muestra una luz verde para indicar que todo está en orden, es decir, que la frecuencia cardíaca y los niveles de oxígeno se encuentran dentro de rangos normales. Como resultado, el dispositivo permite la correcta recolección de datos, como temperatura, frecuencia cardíaca, nivel de oxígeno y otros factores relacionados con las causas del síndrome de muerte súbita del lactante (SIDS). Por ejemplo, se logra calcular la oxigenación de la sangre del pulso (SpO₂) utilizando la ley de Beer-Lambert. El sistema incorpora optoelectrónica orgánica avanzada, LED orgánicos flexibles y fotodetectores orgánicos OPD, proporcionando comodidad y facilidad de montaje en el antebrazo del lactante [10].

En el artículo "Automatic Posture and Movement Tracking of Infants with Wearable Movement Sensors", publicado en 2020 por Airaksinen M. y Räsänen O., se describe el desarrollo de un dispositivo portátil para bebés equipado con un multisensor inteligente. Este dispositivo permite la recopilación de datos de acelerómetros y giroscopios móviles durante los movimientos del lactante al utilizar el traje diseñado para tal fin. El objetivo principal es analizar los movimientos espontáneos y voluntarios de los bebés, dado que estos reflejan la integridad del desarrollo de las redes cerebrales. La detección de bebés con desarrollo motor atípico resulta

prometedora para identificar a aquellos en riesgo de padecer diversos trastornos del neurodesarrollo. [11]

Posteriormente, se entrenó un algoritmo de aprendizaje automático basado en redes neuronales convolucionales profundas (CNN) para la detección automática de clases de postura y movimiento utilizando los datos y anotaciones recabados. Este estudio demuestra la factibilidad de construir un dispositivo portátil inteligente para bebés que sea cómodo y ofrezca un procesamiento de señales que permita el seguimiento cuantitativo de las actividades de movimiento independientes de los lactantes con alta precisión. Gracias a estos avances, el registro multisensorial del tipo actual abre múltiples vías potenciales para análisis posteriores y aplicaciones en el monitoreo del desarrollo infantil [11].

1.2 Fundamentación teórica

1.3.1 Muerte Súbita

El síndrome de muerte súbita del lactante (SMSL) es un término que se utiliza para describir la muerte inesperada y repentina de un niño menor de un año, sin una explicación clara después de una investigación exhaustiva, que incluye una autopsia, revisión del historial clínico y examen del lugar donde ocurrió el deceso. [12] Aunque la causa exacta del SMSL sigue siendo desconocida, existen diversos factores de riesgo asociados con este fenómeno, incluyendo problemas de salud previos en el niño, como trastornos respiratorios o infecciones, y factores ambientales. El SMSL ocurre con mayor frecuencia durante el período de sueño del lactante, lo que ha llevado a los investigadores a estudiar la relación entre las condiciones del entorno donde duerme el bebé y la aparición de este síndrome. Un factor de riesgo importante es la superficie de descanso del lactante. Una práctica relacionada con un mayor riesgo de SMSL se debe los bebés duerman boca abajo o de lado. Cuando un bebé duerme en estas posiciones, existe una mayor probabilidad de que sus vías respiratorias queden obstruidas, lo que puede conducir a una asfixia involuntaria. [13]

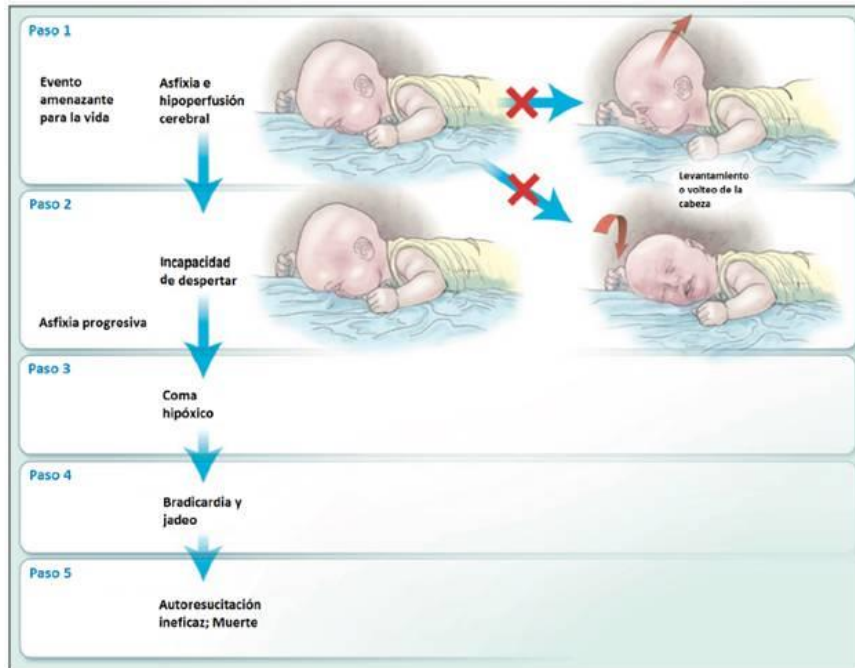


Figura 1 Cinco pasos propuestos en vía respiratoria terminal asociada al síndrome de muerte súbita en el lactante [14]

1.3.2 Factores Asociados a la Muerte Súbita

Existen una serie de factores que se relacionan al fenómeno de la muerte súbita en lactantes, estos pueden ser internos desde el embarazo o externos posterior al parto.

- **Factores asociados con la madre**

Durante el embarazo, es posible identificar diversas enfermedades o condiciones que el bebé podría padecer después del nacimiento. Por ello, existen varios factores que pueden estar asociados a la muerte súbita, como el consumo de tabaco durante el embarazo, la alimentación con leche de fórmula en lugar de leche materna (lactancia artificial), y el consumo de drogas o alcohol [15].

- **Factores asociados al niño**

Algunos aspectos relacionados con el lactante, como haber nacido prematuramente y con un peso por debajo de los límites ideales, o tener antecedentes familiares de un

hermano que haya fallecido debido a muerte súbita, pueden ser factores desencadenantes de este evento trágico. [15]

- **Factores Ambientales**

El entorno en donde el bebé crece y se desarrolla juega un papel crucial en la prevención del síndrome de muerte súbita del lactante. Hay varios factores relacionados que debemos tener en cuenta para asegurar un entorno seguro. Por ejemplo, la posición cuando él bebe duerme es importante. Algunos bebés tienden a dormir boca abajo, asociado con la principal causa de muerte súbita. Además, también debemos considerar las condiciones climáticas y evitar abrigo en exceso. [15]

1.3.3 Frecuencia Cardíaca

La frecuencia cardíaca (FC) es el número de veces que el corazón late por minuto, y es un indicador fundamental de la función cardiovascular y el estado de salud general [16]. La FC varía entre individuos y se ve afectada por factores como la edad, el género, el nivel de actividad física y el estado emocional [17].

En el caso de los lactantes, la frecuencia cardíaca es especialmente importante debido a que su sistema cardiovascular se encuentra en desarrollo y se adapta constantemente a los cambios en la demanda de oxígeno y nutrientes [18]. Los lactantes suelen tener una frecuencia cardíaca más alta que los adultos, lo que les permite mantener una adecuada perfusión de órganos y tejidos [19].

En la tabla 2 se muestran los diferentes valores de la frecuencia cardíaca por rango de edades:

Tabla 2 Frecuencia Cardiaca por Rango de edades [20]

EDAD	BPM - Latidos por minuto DORMIDO	BPM - Latidos por minuto PROMEDIO	BPM - Latidos por minuto DESPIERTO
0 – 3 Meses	80 - 160	140	85 - 205
3 meses – 2 años	75 – 160	130	100 -190
2 años – 10 años	60 – 90	80	60 – 140
Mayor a 10 años	50 – 90	75	60 - 100

Elaborado por: El investigador

Fuente: *Paediatric reference values for heart rate variability during sleep* [20]

1.3.4 Saturación de oxígeno

La saturación de oxígeno (SpO₂) hace referencia a la proporción de hemoglobina en la sangre que está unida al oxígeno, lo cual es esencial para mantener la vida y las funciones celulares. La inhalación de aire proporciona a los pulmones el oxígeno necesario para que sea captado por los glóbulos rojos y transportado a diferentes partes del cuerpo. La adecuada oxigenación es fundamental para el correcto funcionamiento del sistema respiratorio, y la falta de oxígeno en los tejidos (hipoxia) puede afectar negativamente el funcionamiento de los órganos y las células [21].

En términos generales, los niveles normales de SpO₂ se encuentran entre el 95% y el 97%, con una variabilidad del 2%. Las mediciones de SpO₂ por debajo del 95% en reposo pueden estar asociadas con condiciones patológicas, y niveles entre el 92% y el 90% se observan en pacientes con antecedentes de insuficiencia respiratoria crónica [22].

1.3.5 Tecnologías de la Información

- **WI-FI**

Es una tecnología de comunicación inalámbrica que permite la transmisión de datos y el acceso a internet a través de ondas de radio en lugar de cables. Se basa en el estándar IEEE 802.11 y ha experimentado numerosas actualizaciones a lo largo de los años para aumentar la velocidad, el alcance y la seguridad de las conexiones inalámbricas. [23]

En el campo de la telemedicina, Wi-Fi ha desempeñado un papel crucial al permitir la prestación de servicios de atención médica a distancia y mejorar el acceso a la atención médica para pacientes en áreas remotas o con movilidad limitada. La telemedicina utiliza conexiones Wi-Fi para transmitir datos de pacientes, como imágenes médicas, registros de salud electrónicos y señales biomédicas, entre profesionales médicos y pacientes. Además, Wi-Fi permite la implementación de dispositivos médicos conectados a Internet, como monitores de signos vitales y dispositivos de diagnóstico, lo que facilita la monitorización remota y el seguimiento de pacientes en tiempo real. [24]




1.3.6 Sistemas electrónicos de adquisición de datos y el procesamiento de datos

La adquisición de datos conlleva un complejo sistema como cualquier otro tipo de señal, partiendo principalmente de la obtención de todos los signos vitales necesarios mediante el uso de sensores como aquellos dispositivos usables para la Salud.

- **Microcontrolador**

La Tabla 3 presenta diversas opciones de microcontroladores que satisfacen los requisitos de la instrumentaria electrónica. Estos parámetros incluyen: memoria, puertos de entrada y salida, voltaje de operación, dimensiones, entre otros aspectos.

Tabla 3 Comparativa Microcontroladores

Características	<p style="text-align: center;">Adafruit ESP32 Feather</p> 	<p style="text-align: center;">Arduino Nano</p> 	<p style="text-align: center;">Mini NodeMCU (ESP8266)</p> 
Microcontrolador	ESP32	ATmega328P	ESP8266
Frecuencia del reloj	240 MHz	16 MHz	80 MHz (160 MHz opcional)
Memoria Flash	4 MB	32 KB	4 MB
RAM	520 KB	2 KB	80 KB
E/S digitales	21	14 (6 PWM)	11 (9 PWM)
Entradas analógicas	6	8	1
Salidas analógicas (PWM)	16	6	9
Comunicación	Wi-Fi, Bluetooth	-	Wi-Fi
Voltaje de operación	3.3V	5V	3.3V
Voltaje de entrada	USB: 5V; Batería: 3.5-6.5V	7-12V	Micro USB: 5V
Dimensiones	51 x 23 mm	45 x 18 mm	48 x 25 mm
Socket de batería	Sí (JST-PH 2-pin)	No	No
Circuito de carga de batería	Sí	No	No

Elaborado por: El investigador

Fuente: *Fabricantes* [25] [26] [27]

- **Sensor de Ritmo Cardíaco y Oxígeno en la sangre**


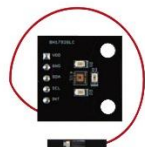

Los sensores de ritmo cardíaco y oxígeno en la sangre son dispositivos electrónicos diseñados para medir la frecuencia cardíaca (número de latidos del corazón por minuto) y la saturación de oxígeno en la sangre (porcentaje de hemoglobina en los glóbulos rojos que transporta oxígeno). [28] Estos sensores son comunes en dispositivos médicos y de bienestar, como oxímetros de pulso, monitores de ejercicio y relojes inteligentes.

Una técnica ampliamente utilizada para medir tanto la frecuencia cardíaca como la saturación de oxígeno en la sangre es la oximetría de pulso. La oximetría de pulso utiliza luz LED en dos longitudes de onda diferentes para calcular la saturación de

oxígeno en la sangre y, a menudo, también proporciona una lectura de la frecuencia cardíaca. [29]

La tabla 4 muestra una comparativa se presentan los detalles técnicos de diversas opciones de sensores de frecuencia cardíaca.

Tabla 4 Comparación de Sensores de Ritmo Cardíaco y Oxigenación en la sangre.

Características	GY-MAX30102 	BH1792GLC 	AFE4404 
Fabricante	Maxim Integrated	ROHM Semiconductor	Texas Instruments
Tipo de sensor	Óptico de PPG	Óptico de PPG	Óptico de PPG
LEDs	Rojo e Infrarrojo	Verde	Rojo, Infrarrojo y Verde
Comunicación	I2C	I2C	SPI
Voltaje de operación	1.7V - 2V (sensor)	1.71V - 3.6V	2V - 3.6V
Consumo de corriente	600µA (típico)	0.6 mA (modo de medición)	5.5 mA (LED encendido), 100 µA (reposo)
Rango de medición HR	30 - 210 BPM	30 - 210 BPM	30 - 210 BPM
Rango de medición SpO2	70% - 100%	70% - 100%	70% - 100%
Resolución ADC	16 bits	16 bits	18 bits
Filtro de ruido	Filtro digital integrado	Filtro digital integrado	Filtro digital integrado
Temperatura de operación	-40°C a +85°C	-40°C a +85°C	-40°C a +85°C
Tamaño	5.6 x 3.3 x 1.55 mm (MAX30102)	2.8 x 2.8 x 0.9 mm	5.2 x 4.7 x 1 mm

Elaborado por: El investigador

Fuente: Fabricantes [30] [31]




- **Acelerómetro**

Un giroscopio es un sensor que mide la velocidad angular en torno a uno o más ejes de rotación [3]. Los giroscopios son útiles para determinar la orientación de un objeto en movimiento y son ampliamente utilizados en sistemas de navegación y

estabilización, así como en dispositivos electrónicos, como teléfonos móviles y drones. [32]

La tabla 5 muestra una comparativa se presentan los detalles técnicos de diversas opciones de acelerómetros.

Tabla 5 Comparación de Acelerómetros

Características	MPU-6050 	ADXL345 	LIS3DH 
Ejes	3 (acelerómetro) + 3 (giroscopio)	3	3
Rango de medida	±2g, ±4g, ±8g, ±16g	±2g, ±4g, ±8g, ±16g	±2g, ±4g, ±8g, ±16g
Resolución	16 bits	13 bits	16 bits
Salida de datos	I2C	I2C, SPI	I2C, SPI
Consumo de energía	3.9 mA (typical)	23 µA - 130 µA (typical)	6 µA - 11 µA (typical)
Rango de voltaje	2.375 V - 3.46 V	2.0 V - 3.6 V	1.71 V - 3.6 V

Elaborado por: El investigador

Fuente: Fabricantes [33] [34]

Estos acelerómetros ofrecen características similares en términos de ejes, rangos de medición y resolución. Sin embargo, difieren en su consumo de energía, opciones de comunicación y funciones adicionales.

1.3.7 Baterías LiPo

Las baterías de polímero de litio (LiPo) son una clase de baterías recargables que emplean polímeros de litio como material de electrolito en lugar de líquidos utilizados en baterías de iones de litio convencionales [35]. Estas baterías tienen una serie de ventajas en comparación con otros tipos de baterías, como su capacidad de ser moldeadas en diferentes formas y tamaños, lo que permite una mayor flexibilidad en el diseño de dispositivos electrónicos. Además, las baterías LiPo son conocidas por su

alta relación energía-peso y capacidad de proporcionar corrientes de descarga altas [36].

Estas baterías han ganado popularidad en aplicaciones electrónicas portátiles y vehículos eléctricos debido a sus ventajas en comparación con otras tecnologías de baterías, como las de iones de litio (Li-ion) o níquel-metal hidruro (NiMH) [37].

Ventajas de las baterías LiPo:

- **Flexibilidad de forma:** A diferencia de las baterías cilíndricas de Li-ion, las baterías LiPo son flexibles y pueden fabricarse en diversas formas y tamaños, lo que permite a los diseñadores adaptarlas a dispositivos electrónicos de diferentes formas.
- **Peso ligero:** Son generalmente más ligeras que las baterías Li-ion de capacidad similar, lo que las hace ideales para aplicaciones portátiles donde el peso es una consideración importante.
- **Baja tasa de autodescarga:** Poseen una tasa de autodescarga relativamente baja, lo que les permite mantener su carga durante largos períodos de inactividad.
- **Ciclos de carga rápidos:** Logran cargarse rápidamente en comparación con otros tipos de baterías, lo que reduce el tiempo necesario para recargar dispositivos electrónicos. [37]

1.3.8 Frameworks para la creación de aplicaciones móviles




Un "framework" (marco de trabajo) es una estructura de soporte que proporciona un esquema predefinido y componentes reutilizables para simplificar y acelerar el desarrollo de aplicaciones de software. Los frameworks están diseñados para abordar desafíos comunes y proporcionar soluciones estandarizadas, lo que permite a los desarrolladores centrarse en las características específicas de su aplicación en lugar de preocuparse por detalles de bajo nivel. [38]

Los frameworks pueden incluir bibliotecas de funciones, módulos de código, herramientas de desarrollo y conjuntos de reglas y mejores prácticas [3]. Los desarrolladores pueden adaptar y extender los componentes de un framework para

satisfacer sus necesidades específicas, lo que resulta en un proceso de desarrollo más eficiente y una mayor calidad del software. [39]

Existen varios frameworks para el desarrollo de aplicaciones móviles, tanto para plataformas específicas como para enfoques multiplataforma. En la tabla 6 se muestra una comparación de los más populares:

Tabla 6 Comparación de Frameworks

Características	Flutter 	Xamarin 	NativeScript 
Empresa	Google	Microsoft	Progress Software
Lenguajes	Dart	C# y .NET	JavaScript, TypeScript, Angular, Vue.js
Plataformas	iOS, Android, Web, Desktop	iOS, Android, Windows	iOS, Android
Rendimiento	Alto (compilación AOT)	Bueno (compilación JIT y AOT)	Bueno (compilación JIT)
UI Nativa	Sí (widgets propios)	Sí (controles nativos)	Sí (controles nativos)
Ecosistema	En crecimiento rápido	Maduro y estable	Maduro pero más pequeño que los otros
Comunidad	Gran comunidad y en rápido crecimiento	Gran comunidad y amplio soporte empresarial	Comunidad activa pero más pequeña
Código compartido	Alto (UI y lógica de negocio)	Alto (lógica de negocio, UI parcialmente)	Alto (UI y lógica de negocio)
Curva de aprendizaje	Moderada (requiere aprender Dart)	Moderada (necesita conocimientos de C# y .NET)	Moderada (conocimientos de JS y TS)
Licencia	Código abierto (Licencia BSD)	Código abierto (Licencia MIT)	Código abierto (Licencia Apache 2.0)

Elaborado por: El investigador

Fuente: *Fabricantes* [40]

1.3.9 Backend

El término "backend" se refiere a la parte de un sistema o aplicación que se encarga de la lógica y funcionalidad detrás de la interfaz de usuario. En otras palabras, es el conjunto de herramientas, tecnologías y servicios que trabajan juntos para gestionar los datos y la lógica de una aplicación. El backend es responsable de recibir las solicitudes del cliente, procesarlas y devolver los resultados a través de la interfaz de usuario. El desarrollo de un backend implica la creación de una arquitectura adecuada para la gestión de datos, la lógica empresarial y la integración con servicios externos, como bases de datos, servicios en la nube, API y sistemas de autenticación y autorización. El objetivo principal de un backend es proporcionar a los usuarios una experiencia sin interrupciones y asegurarse de que la aplicación sea escalable y segura. [41]

- **Base de Datos**

Una base de datos es un conjunto de información organizada y almacenada en una computadora de manera sistemática, para que pueda ser accedida y utilizada con facilidad. La información se guarda en tablas compuestas por filas y columnas, donde cada columna representa un atributo de la información y cada fila representa una entidad o registro. [42]

Existen dos tipos principales de bases de datos: SQL y NoSQL. Las bases de datos SQL (Structured Query Language) son las más utilizadas y tradicionales. Están diseñadas para manejar datos estructurados y relacionales, lo que significa que la información se organiza en tablas con relaciones establecidas entre ellas. Estas bases de datos utilizan un lenguaje común (SQL) para manipular y consultar la información. [43]

Por otro lado, las bases de datos NoSQL (Not Only SQL) están diseñadas para manejar datos no estructurados y no relacionales. Estas bases de datos son altamente escalables y flexibles, lo que las hace ideales para manejar grandes volúmenes de datos no estructurados como documentos, gráficos y datos geoespaciales. Utilizan diferentes

modelos de datos, como documentos, columnas, clave-valor y grafos, y no requieren un esquema fijo como en las bases de datos SQL. [44]

A continuación, se presenta la tabla 7 con la comparativa entre los aspectos más importantes de las bases de datos SQL y NoSQL:

Tabla 7 Comparativa SQL y NoSQL

Característica	Bases de datos SQL	Bases de datos NoSQL
Estructura	Datos estructurados y relacionales	Datos no estructurados y no relacionales
Escalabilidad	Menos escalables	Altamente escalables
Flexibilidad	Menos flexibles	Altamente flexibles
Consultas	Utilizan SQL para consultar y manipular datos	Utilizan diferentes lenguajes de consulta
Esquema	Utilizan un esquema fijo y predefinido	No requieren un esquema fijo
Transacciones	Manejan transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad)	No manejan necesariamente transacciones ACID
Uso	Ideal para datos estructurados y relacionales, como registros de clientes y transacciones financieras	Ideal para datos no estructurados y no relacionales, como datos geoespaciales y documentos

Elaborado por: El investigador

Fuente: *Fabricantes* [44]

Servicios Backend:

Los servicios Backend son una combinación de servicios alojados en servidores remotos que proporcionan servicios web, bases de datos y almacenamiento en la nube. [45] Estos servicios son esenciales para la mayoría de las aplicaciones móviles y web, ya que proporcionan una forma escalable de procesamiento de datos y almacenamiento.

A continuación, se presenta en la tabla 8 una comparativa entre Firebase, Apache Cassandra y Redis, tres de los servicios Backend más populares, en función de sus características y funcionalidades.

Tabla 8 Comparativa de servicio Backend

Características	Firestore	Apache Cassandra	Redis
Tipo de base de datos	NoSQL	NoSQL	NoSQL
Soporte para múltiples lenguajes de programación	Sí	Sí	Sí
Soporte para almacenamiento de datos en tiempo real	Sí	No	No
Soporte para autenticación y autorización de usuarios	Sí	No	No
Soporte para notificaciones push	Sí	No	No
Soporte para análisis de aplicaciones móviles	Sí	No	No
Facilidad de uso	Muy fácil	Moderadamente fácil	Moderadamente fácil
Escalabilidad	Altamente escalable	Altamente escalable	Altamente escalable
Costo	Gratis/De pago	Gratis/De pago	Gratis/De pago

Elaborado por: El Investigador [46]

1.4 Objetivos

1.4.1. Objetivo General

- Desarrollar un sistema de monitoreo no invasivo y seguridad preventiva de factores causantes del síndrome de muerte súbita de los Lactantes del Centro de Estimulación Temprana y Prenatal “Neuro Place” a través de Smart Textil.

1.4.2. Objetivos Específicos

- Analizar los factores que afecten los signos vitales que conllevan a muerte súbita de un lactante en sus primeros meses de vida.
- Determinar las características de hardware y software para la creación de un textil electrónico que permita un monitoreo constante de factores causantes del síndrome de la muerte súbita de los lactantes a través de un dispositivo móvil.
- Elaborar un Smart Textil con un sistema de monitoreo de factores causantes del síndrome de muerte súbita que permita alertar sobre los valores anormales que presenta en sus signos vitales.
- Diseñar una Interfaz móvil para la visualización de signos vitales y posición del lactante para él envío de alertas al presentarse valores fuera de rango y anormales.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

Este proyecto tiene como objetivo desarrollar un sistema de monitoreo para lactantes que brinde tranquilidad a los padres y cuidadores al supervisar los signos vitales del bebé en tiempo real. Para lograr esto, se utilizaron diferentes componentes electrónicos de alta calidad.

El sensor Gy-MAX30102 es un sensor de ritmo cardíaco de alta precisión, que se utiliza para medir la frecuencia cardíaca del bebé. El sensor es muy sensible y puede detectar incluso pequeños cambios en el ritmo cardíaco, permite una supervisión precisa y en tiempo real de la salud del bebé. El sensor MPU6050 es un sensor de movimiento que se utiliza para medir la actividad y posición del lactante. El sensor es muy preciso puede detectar cambios en la posición del bebé, esto permite una supervisión detallada y en tiempo real de la actividad del bebé. Además, la combinación de ambos sensores permite una supervisión completa y precisa de la salud del bebé.

Para el procesamiento y la transmisión de los datos, se empleó el microcontrolador Esp32 Feather por sus características avanzadas de conectividad inalámbrica y su alta capacidad de procesamiento, ideal para aplicaciones de monitoreo en tiempo real. Además, su diseño compacto, bajo consumo de energía lo hacen perfecto para aplicaciones portátiles, cuenta con un módulo WiFi y Bluetooth integrado, lo que permite la transmisión de datos en tiempo real a una aplicación móvil. Además, su arquitectura de doble núcleo permite la separación de tareas críticas, como el monitoreo a través de los sensores y la transmisión de datos, asegurando una alta precisión y una operación sin interrupciones.

Otro factor clave en la elección del Esp32 Feather fue su capacidad para conectarse a una batería de polímero de litio de alta capacidad. La batería proporciona una fuente de energía estable y duradera, garantizando que el sistema de monitoreo pueda operar durante largos períodos de tiempo sin interrupciones. Además, su capacidad de carga rápida y su bajo peso la hacen ideal para aplicaciones portátiles.

La aplicación móvil desarrollada en Flutter es una parte importante del sistema de monitoreo de lactantes, permite a los padres y cuidadores supervisar los signos vitales del bebé en tiempo real y recibir alertas si se detecta algún problema. La aplicación cuenta con una interfaz de usuario clara y sencilla, que permite una visualización fácil de los datos. Además, la aplicación permite el almacenamiento de datos y la posibilidad de compartir información con los profesionales de la salud.

Por último, la plataforma Firebase de Google se utilizó para garantizar la seguridad y la privacidad de los datos del bebé. Firebase ofrece servicios de autenticación de usuarios y almacenamiento de datos en la nube, lo que garantiza que los datos del bebé estén protegidos y accesibles solo para los usuarios autorizados. Además, Firebase ofrece herramientas de monitoreo y análisis que permiten a los desarrolladores mejorar y optimizar el sistema de monitoreo.

2.2. Métodos

2.2.1 Modalidad de investigación

Este proyecto consistió en una Investigación Aplicada que buscaba solucionar los problemas del monitoreo de signos vitales dentro de un centro de estimulación temprana y prenatal, mediante la implementación de alertas que sobrepasen los valores establecidos de los signos vitales a través de un aplicativo móvil. Para lograrlo, se emplearon conocimientos ya existentes, aplicándolos en un contexto específico.

La Investigación fue Bibliográfica, el proyecto de investigación se sustentó mediante la recopilación de información de diversas fuentes, tales como revistas técnicas, libros, artículos científicos, publicaciones en internet y tesis, relacionados a los sistemas de monitoreo, textiles y normativas para los avales médicos. Esta información permitió fundamentar el proyecto en el conocimiento previo y las mejores prácticas en el área de monitoreo de signos vitales en lactantes.

De igual manera, el proyecto fue una Investigación de Campo, recopilando información e implementando el sistema dentro del establecimiento Centro de Estimulación Temprana y Prenatal “Neuro Place”, dedicado al cuidado de lactantes. La recolección de información y la implementación del sistema en un entorno real permitió validar la efectividad y eficiencia de este, así como identificar posibles mejoras en el proceso de monitoreo de signos vitales en lactantes.

2.2.2 Recolección de información

Con el fin de recopilar información para la investigación, se utilizaron diversas fuentes, como libros, recursos en línea, proyectos anteriores, revistas científicas, manuales de construcción y guías prácticas. Se seleccionaron cuidadosamente bases de datos respaldadas por la comunidad científica para garantizar la validez y confiabilidad de la información recopilada.

2.2.3 Procesamiento y análisis de datos

Para el procesamiento y análisis de los datos se siguieron los siguientes pasos:

- Revisión de la información recopilada.
- Estudio las propuestas de solución para el dispositivo de monitoreo del lactante mediante Smart Textil.
- Evaluación de la información y datos obtenidos durante la ejecución del proyecto que llevaron a una solución del problema.
-

2.2.4 Desarrollo de Proyecto

Se establecen una serie de actividades para poder ejecutar el presente proyecto:

1. Recolección de datos de la situación actual de la muerte súbita de lactantes dentro del país.

2. Análisis y selección de los factores externos que pueden afectar la vida de un lactante en sus primeros meses de vida y que puedan ser monitoreados.
3. Comparación de dispositivos existentes para la medición de posición y frecuencia cardíaca de un lactante.
4. Ajuste de rangos que permitan establecer parámetros de operación para una emisión de alerta.
5. Selección, comparación de sensores y microcontroladores que se ajusten a las necesidades del textil electrónico.
6. Análisis de las tecnologías inalámbricas existentes.
7. Elaboración de la tabla comparativa de protocolos de envío de datos.
8. Estudio del protocolo y del método de almacenamiento de los datos.
9. Elaboración de la prenda acorde a las dimensiones necesarias para los lactantes.
10. Programación del microcontrolador y ensamblaje de los sensores.
11. Elaboración del aplicativo móvil con sus respectivas funciones.
12. Elaboración de la interfaz para la interacción con el usuario final.
13. Programación de notificación de alertas cuando se detecten valores fuera de rango de signos vitales y posición del niño.
14. Realización de pruebas de funcionamiento y análisis de los resultados obtenidos para la corrección de posibles errores y la determinación del nivel de confianza de los valores finales obtenidos con el dispositivo generado.
15. Análisis de resultados y la obtención de conclusiones del sistema en funcionamiento.
16. Elaboración de Informe Final.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de los resultados

El sistema de monitoreo no invasivo y seguridad preventiva de factores causantes del Síndrome de Muerte Súbita de los Lactantes (SMSL) en el Centro de Estimulación Temprana y Prenatal "Neuro Place", utilizando smart textil, puede prevenir el SMSL al detectar cualquier anomalía en tiempo real y alertar a los cuidadores. Además, este sistema también previene factores de riesgo para el SMSL, como la posición de sueño insegura, para garantizar la seguridad del bebé mientras duerme.

En este sentido, se ha desarrollado una aplicación móvil que permite observar el registro de los datos capturados por la indumentaria utilizada por los bebés. Esta aplicación cuenta con una interfaz de usuario intuitiva que permite visualizar los datos de forma clara y sencilla. Además, la aplicación notifica automáticamente si se produce alguna alerta, como exceder el umbral de frecuencia cardiaca máxima, la oxigenación de la sangre en sangre o la posición del lactante.

Es importante destacar que el dispositivo utilizado en el monitoreo no invasivo de los lactantes cuenta con una batería de litio segura y de fácil colocación. Esta batería es capaz de almacenar la energía necesaria para el funcionamiento del dispositivo, convirtiéndolo en un equipo energéticamente autónomo y fácil de transportar. En caso de ser necesario, la batería es fácilmente reemplazable

3.2. Desarrollo de la propuesta

Para la elaboración del proyecto se desarrolló mediante una metodología teórico-práctica. En primer lugar, se realizó una Investigación Bibliográfica en bases de datos nacionales e internacionales respaldadas por la comunidad científica, sobre los métodos de monitoreo no invasivo en bebés, los factores de riesgo del SMSL y sus consecuencias en la salud del lactante.

Posteriormente, se diseñó una prenda que permitirá el monitoreo continuo de los factores de riesgo del SMSL, la detección temprana de cualquier anomalía en tiempo real y la alerta a los cuidadores, así como una aplicación móvil que permitirá visualizar los datos sensados por la indumentaria y notificar si se produce alguna alerta. La interfaz de usuario de la aplicación es clara y sencilla de utilizar. Finalmente, se empleó una batería de litio segura y fácil de colocar para garantizar la autonomía energética del dispositivo y su portabilidad.

3.2.1 Arquitectura del sistema

Para la elaboración del sistema, se presenta su arquitectura mediante un esquema compuesto por varias etapas, como se muestra en la figura.

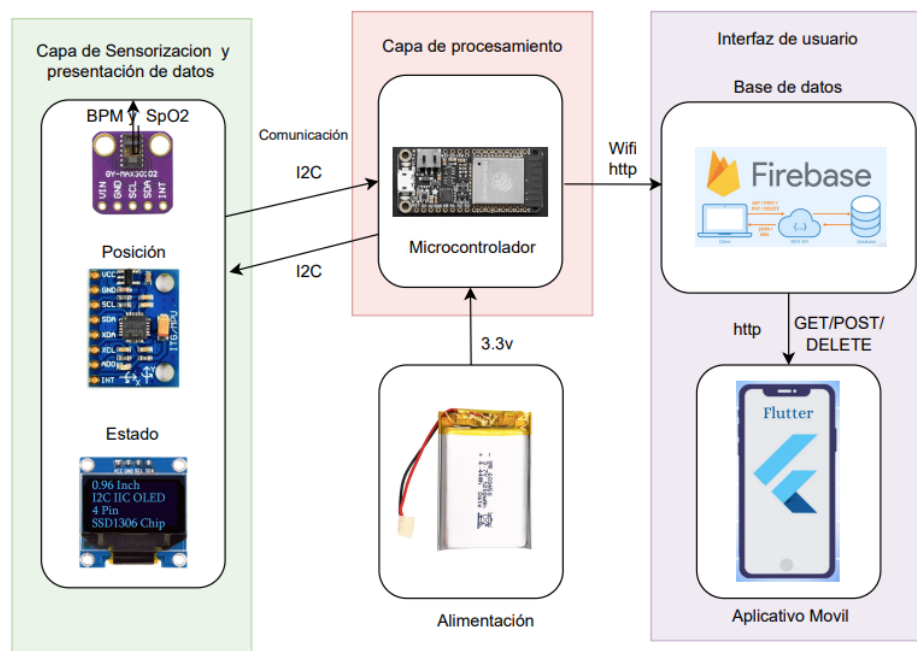


Figura 2 Arquitectura del sistema
Elaborado por: El investigador

Dentro de la figura 1 se pueden observar tres capas principales y los modos de comunicación entre cada etapa, que son descritas a continuación:

3.2.1.1 Capa de monitoreo y presentación de datos

En la fase de monitoreo del sistema, se lleva a cabo la captura y adquisición de datos o señales provenientes de los sensores, lo cual resulta crucial para la medición de

magnitudes físicas y fenómenos de interés. Durante esta etapa, se emplean diversos tipos de sensores diseñados para detectar y convertir magnitudes físicas específicas en señales eléctricas que puedan ser procesadas por el sistema. En esta capa se encuentra la presentación de datos que serán visualizados por la persona que utilice el dispositivo. A continuación, se detalla los sensores que son utilizados dentro de esta capa con los signos vitales serán medidos:

- **Sensor para Oxigenación en la sangre y Ritmo Cardíaco**

El sensor GY-MAX30102 es una elección ideal luego de realizar una comparativa que se presenta en la tabla 3. Aquí un resumen destacado de sus ventajas:

- Mediciones precisas de frecuencia cardíaca y oxigenación de la sangre, lo cual resulta vital en aplicaciones de monitoreo de la salud.
- Diseño compacto y ligero que se adapta fácilmente a proyectos portátiles con limitaciones de espacio, sin comprometer la comodidad del usuario.
- Utiliza la interfaz I2C para la comunicación, lo cual simplifica la conexión y permite una mayor flexibilidad de diseño al conectar varios dispositivos en un solo bus con solo dos líneas de comunicación.
- Eficiente en el consumo de energía, lo cual maximiza la duración de la batería en proyectos portátiles, permitiendo un uso prolongado sin agotar rápidamente la energía del dispositivo.

- **Giroscopio**

A través de la comparativa realizada en la tabla 4 el sensor MPU6050 es la mejor opción para este proyecto ya que presenta las siguientes características:

El MPU6050 es un sensor de seis ejes que combina un acelerómetro de tres ejes y un giroscopio de tres ejes, esta combinación permite detectar y medir tanto el movimiento lineal y angular como la orientación del dispositivo en el espacio tridimensional, además resulta especialmente útil en aplicaciones como la navegación, juegos y principalmente en este caso el control de movimiento presenta un diseño compacto, lo cual lo hace ideal para este proyecto donde el espacio es limitado.

- LCD Oled I2C (Presentación de datos)

El LCD OLED (Diodo Emisor de Luz Orgánica) con interfaz I2C es una pantalla que utiliza tecnología OLED para mostrar información de manera clara y nítida. Se caracteriza por tener píxeles individuales que emiten luz propia, resultando en un contraste y una saturación de color superiores en comparación con las pantallas LCD convencionales. Además, gracias a su bajo consumo de voltaje, el LCD OLED muestra información importante como el estado de la conexión WiFi, el progreso del proceso de adquisición y envío de datos, y el porcentaje de la batería restante, lo que permite una rápida recarga cuando los niveles son bajos.

3.2.1.2 Comunicación I2C

El protocolo que se utiliza es I2C debido a que facilita la comunicación entre un microcontrolador maestro y varios dispositivos esclavos, como se muestra en la figura 3, posibilitando el intercambio de información de manera eficiente y confiable. Permite la conexión de múltiples dispositivos en un solo bus utilizando únicamente dos líneas de comunicación: una línea de datos (SDA) y una línea de reloj (SCL).

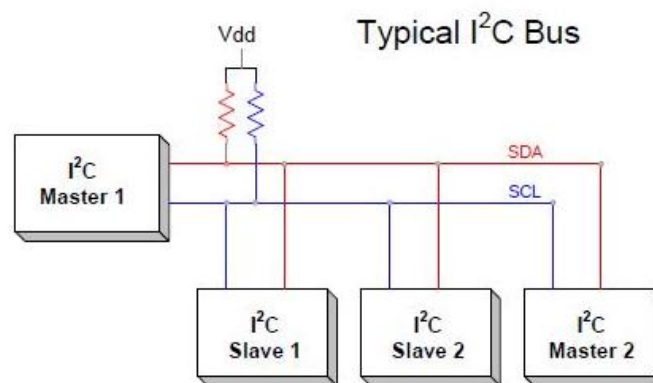


Figura 3 Conexión I2C Maestro-Eslavos [47]

A través de operaciones de lectura, el microcontrolador solicita y recibe datos del GY-MAX30102 y el MPU6050. Estos datos incluyen mediciones de frecuencia cardíaca, saturación de oxígeno y orientación. El microcontrolador procesa esta información de acuerdo con las necesidades del proyecto, como el monitoreo de la salud o la detección

de movimiento, permitiendo así tomar decisiones o ejecutar acciones en base a los datos obtenidos.

Por otro lado, mediante operaciones de escritura, el microcontrolador controla el LCD OLED. A través del envío de comandos y datos, el microcontrolador configura el LCD OLED para mostrar información relevante en la pantalla. Esta información puede incluir el estado de la conexión a la red WiFi, el progreso de la adquisición y envío de datos, así como el nivel de la batería

3.2.1.3 Capa de Procesamiento

Para el procesamiento de los datos se opta por el Esp32 Feather luego de la comparativa de la tabla 3, a diferencia de los otros microcontroladores nos permite conectar una batería de Litio con su conector JST-PH 2-pin integrado, permitiendo cargarlo directamente, sin la necesidad de módulos externos, además soporta la comunicación I2C y la conexión Wifi, con un tamaño compacto que permite crear un dispositivo portátil.

3.2.1.4 Alimentación

La alimentación del sistema se lleva a cabo mediante una batería de polímero de litio de 3.7 V y 1800 mAh. Esta batería está equipada con un conector JST que se conecta directamente al microcontrolador. Además de su rendimiento fiable y su larga vida útil, la batería cuenta con una función de protección PCM. Esta función proporciona protección contra sobrecargas, sobredescargas, sobrecorrientes, cortocircuitos y sobretemperatura, lo que garantiza un funcionamiento seguro y confiable del sistema. También se destaca su excelente relación costo-rendimiento, lo que la convierte en una opción óptima para alimentar el dispositivo.

3.2.1.5 Conectividad inalámbrica

- Wifi

La tecnología WiFi proporciona una serie de ventajas significativas para la conexión del microcontrolador destinado a medir los signos vitales. En primer lugar, ofrece una conexión inalámbrica que elimina la necesidad de cables y permite una mayor movilidad para el usuario. Además, el WiFi presenta un amplio alcance y velocidades rápidas de transferencia de datos, a diferencia del Bluetooth que tiene un alcance más

limitado y requiere una proximidad más cercana. Estas características son esenciales para lograr un monitoreo preciso y en tiempo real de los signos vitales.

- **Protocolo HTTP**

HTTP es el protocolo utilizado para la comunicación en la World Wide Web y permite la transferencia de información entre un cliente y un servidor a través de Internet.

Cuando se utiliza HTTP en esta configuración, el microcontrolador puede enviar y recibir datos relacionados con los signos vitales a través de solicitudes y respuestas HTTP. El microcontrolador envía una solicitud HTTP al servidor para enviar datos de los signos vitales medidos, como la frecuencia cardíaca, la saturación de oxígeno y la posición. A su vez, el servidor responde con una confirmación de recepción o con información adicional, como recomendaciones médicas o gráficos de tendencias de los signos vitales.

El uso de HTTP permite la integración con servicios web y aplicaciones en la nube, lo que facilita el monitoreo remoto de los signos vitales. Los datos son enviados y almacenados en servidores remotos, lo que permite el acceso a la información desde cualquier ubicación con conexión a Internet. Además, HTTP ofrece opciones de seguridad, como la encriptación de datos mediante HTTPS, para garantizar la privacidad y la integridad de los datos transmitidos.

3.2.1.6 Capa de Interacción con el Usuario

Para el desarrollo de la aplicación móvil, se utilizó el framework de desarrollo Flutter, como se pudo apreciar en la comparativa presentada en la tabla 6. Para alojar la base de datos, se empleó un backend que será detallado a continuación junto a las tecnologías utilizadas:

- a) Se eligió Firebase como plataforma de backend debido a sus servicios adicionales a base de datos, como el envío de notificaciones push en tiempo real. Esto resulta beneficioso para poder enviar alertas al usuario mediante la aplicación móvil. Además, Firebase ofrece una librería específica para trabajar con Flutter, lo que simplifica el desarrollo. Durante la evaluación de opciones, se consideraron tanto el plan gratuito (Spark) como el plan de pago (Blaze), basado en el costo por uso. Para este proyecto, se decidió utilizar el plan de

pago Blaze, el cual proporciona acceso a todos los servicios y su costo está determinado por el porcentaje de uso de dichos servicios.

- b) En función de la información presentada en la tabla 7, se optó por utilizar un modelo de base de datos no relacional (NoSQL) para el proyecto. Este tipo de base de datos ofrece ventajas notables en términos de capacidad para gestionar grandes volúmenes de datos y tráfico, lo cual resulta fundamental para el correcto funcionamiento de la aplicación. Dado que la instrumentaria estará enviando datos de forma continua, es crucial contar con una base de datos que pueda manejar esta carga constante. Además, la aplicación móvil requiere realizar consultas rápidas para sincronizar los datos adquiridos por el sensor y garantizar una experiencia fluida para el usuario.
- c) Finalmente, como se mencionó La elección del Framework para el desarrollo móvil (Front-End) recayó en Flutter. Esta decisión se basó en varias razones, entre las cuales se destaca su excelente rendimiento y velocidad, gracias a la capacidad de ejecutar aplicaciones como aplicaciones nativas. Además, Flutter ofrece la posibilidad de crear aplicaciones multiplataforma en el futuro utilizando el código base existente, lo cual fue un factor determinante.

3.2.2 Diseño del sistema

Para el diseño del dispositivo destinado a medir los signos vitales en bebés de 1 a 12 meses de edad, se consideró encontrar un tamaño lo más compacto posible. Como sensor principal, se utilizó el Gy-max30102, que emplea la técnica de fotoplethismografía (PPG) para detectar la absorción de luz por los vasos sanguíneos. Para ubicar el sensor en el cuerpo del lactante, se decidió colocarlo en la pantorrilla, esta zona cuenta con las venas Safena Mayor y Safena Menor, como se ilustra en la figura 4. Se optó por esta ubicación debido a que otras áreas, como la muñeca, podrían ser manipuladas y causar molestias al niño, afectando el correcto funcionamiento del dispositivo.

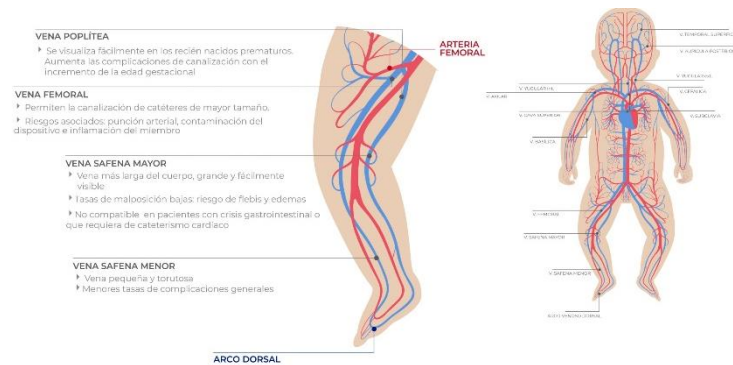


Figura 4 Venas disponibles en la pierna de un lactante [55]

3.2.2.1 Diseño del sistema de medición

Para el diseño del dispositivo, se utiliza el software EasyEDA debido a su facilidad de uso y la capacidad de visualizar los resultados en 3D con los sensores. En este caso, se implementa una configuración de maestro (microcontrolador) y esclavo (sensores). La tabla 8 muestra los pines utilizados y proporciona una descripción de cada uno de ellos.

Tabla 9 Pines para la comunicación I2C

Conexión	Descripción
VCC	Conexión de voltaje positivo. Proporciona la alimentación necesaria para el dispositivo o componente.
GND	Conexión de tierra o voltaje de referencia. Sirve como punto de retorno para la corriente eléctrica y establece el potencial de referencia para las demás conexiones.
SCL	Línea de reloj (Clock) del bus de comunicación. Se utiliza para sincronizar la transferencia de datos entre dispositivos.
SDA	Línea de datos (Data) del bus de comunicación. Se utiliza para enviar y recibir datos entre dispositivos.

Fuente: Elaborador por autor

El microcontrolador y los sensores requieren un total de 4 pines para su conexión. Tanto los sensores como el LCD I2C recibirán alimentación directamente del microcontrolador, y los pines de comunicación SDA y SCL serán los mismos. Con esta información en mente, se procede a realizar el siguiente diseño:

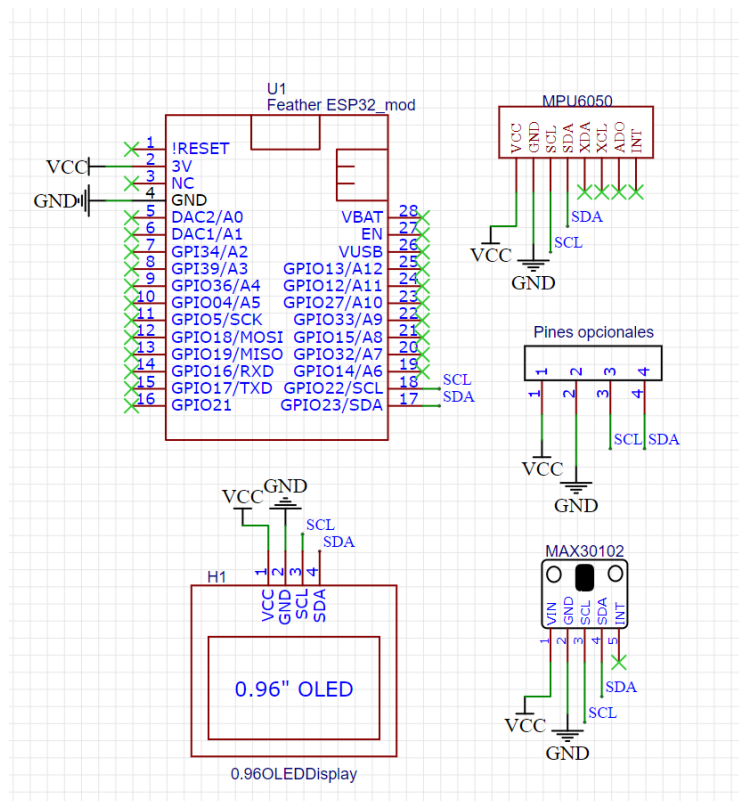


Figura 5 Diseño esquemático
Elaborador por: El investigador

En la figura 5 se muestra la conexión directa de todos los pines al controlador. En este caso, no se ha reservado un espacio específico para la batería, fue conectada de forma externa utilizando el conector JST hembra del ESP32 Feather. De esta manera, la batería de litio puede ser conectada directamente. Una vez descrito el esquema, se procede a realizar el circuito esquemático e implementarlo.

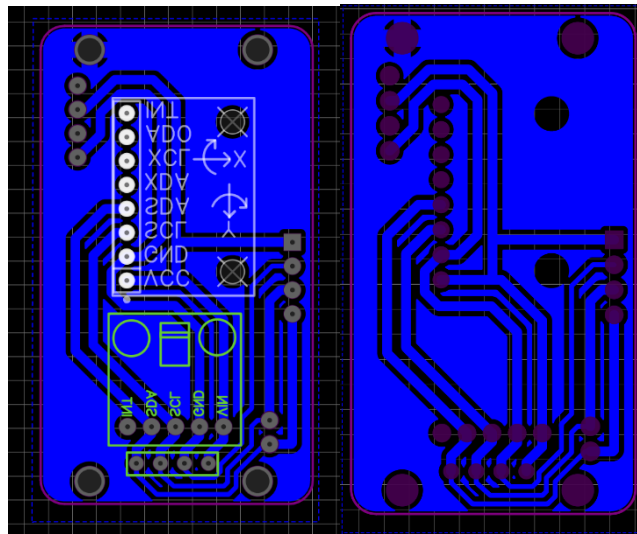


Figura 6 Diseño de PCB

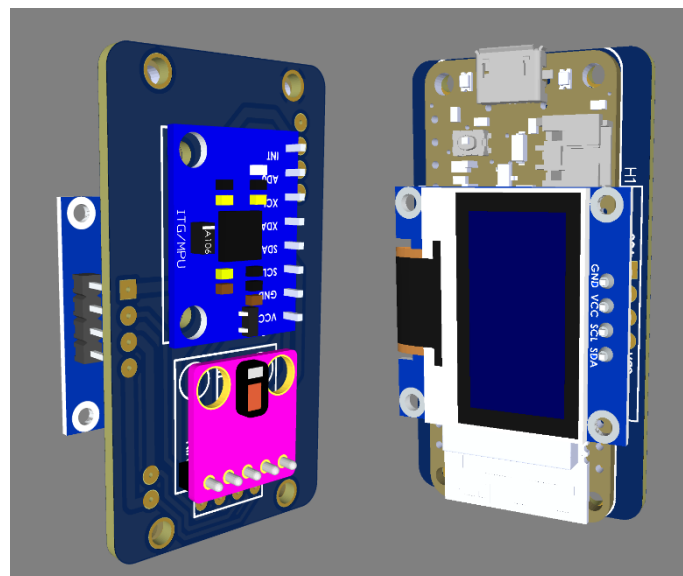


Figura 7 Circuito 3D esquemático

El objetivo principal en el diseño del circuito impreso es lograr un tamaño reducido y compacto para el módulo, de manera que los sensores estén colocados de forma eficiente y ocupando el menor espacio posible. Como se puede observar en las figuras 6 y 7, se ha logrado un diseño que tiene dimensiones de 6 centímetros en sentido vertical y 3 centímetros en sentido horizontal. Esta optimización de tamaño permite

una mayor portabilidad y facilidad de integración en diferentes dispositivos, sin comprometer la funcionalidad ni la calidad del sistema.

3.2.2.2 Diseño de armazón

Después de haber completado el circuito con el microcontrolador y los sensores, se procedió a diseñar una armazón 3D que cumple dos funciones importantes: proteger y asegurar la PCB, y permitir la integración del dispositivo en prendas textiles. Este armazón ha sido cuidadosamente elaborado para brindar un ajuste perfecto a la PCB, garantizando la seguridad y estabilidad de los componentes electrónicos. La utilización de esta carcasa 3D ofrece la posibilidad de envolver el dispositivo con tela, lo cual resulta especialmente beneficioso al utilizarlo en contextos textiles. Esta combinación de carcasa y tela proporciona una mayor comodidad y facilita la integración del dispositivo en diferentes prendas o accesorios, permitiendo así su uso de forma discreta y cómoda en diversas aplicaciones, el diseño fue realizado en 3D Fusión 360.

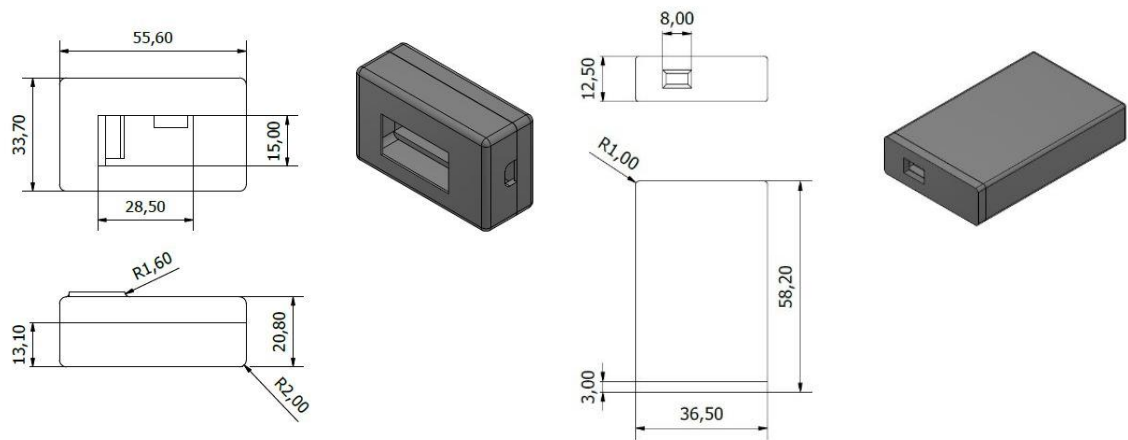


Figura 8 Medidas del armazón del circuito y de la batería
Elaborado por: El investigador

Como se puede apreciar en la figura 8, las medidas se encuentran expresadas en milímetros. El armazón que rodea al circuito presenta bordes redondeados, lo cual contribuye a brindar comodidad y evita posibles daños al lactante. Además, su diseño permite un recubrimiento sencillo con la tela que se aplicará posteriormente. La figura también muestra las dimensiones del armazón que protege la batería de litio, la cual cuenta con un interruptor de encendido. Esta disposición garantiza una protección

adecuada del circuito, minimizando cualquier problema potencial, la figura 9 muestra la implementación del circuito en el armazón principal.



Figura 9 Implementación del circuito en el armazón
Elaborado por: El investigador

3.2.3 Implementación de Textil

En la sección 3.2.2, se mencionó que el circuito se colocará en la pantorrilla. Por lo tanto, al diseñar el textil para este dispositivo, se tomaron en cuenta las siguientes consideraciones:

Comodidad: El diseño textil debe ser lo suficientemente cómodo para ser usado alrededor de la pantorrilla durante períodos prolongados. Debe evitar puntos de presión excesiva y asegurar una sensación agradable en la piel.



Figura 10 Microfibra
Elaborado por: El investigador

La microfibra es la elección ideal al diseñar ropa para bebés debido a su suavidad excepcional, transpirabilidad superior y ligereza que brinda comodidad sin restricciones. Además, su durabilidad y resistencia garantizan una larga vida útil,

mientras que su facilidad de cuidado simplifica el lavado frecuente que requiere la ropa de bebé. Con la microfibra, los bebés disfrutaban de una prenda suave al tacto, que permite la circulación del aire, regula la temperatura corporal y evita irritaciones cutáneas, proporcionando así una experiencia de vestir cómoda y segura.

Ajuste adecuado: El diseño textil para la colocación del circuito alrededor de la pantorrilla requiere un ajuste preciso que se adapte a la forma de la pierna del bebé. Debido a que el tamaño de la pantorrilla varía considerablemente desde los 2 meses hasta aproximadamente un año, es fundamental contar con un sistema de ajuste versátil. En este sentido, el uso de Velcro se presenta como la mejor opción, ya que permite un ajuste personalizado y seguro, adaptándose fácilmente a diferentes tamaños y formas de pantorrilla. Esto garantiza un contacto óptimo con la piel y un rendimiento efectivo del circuito, brindando comodidad y seguridad al bebé durante su uso, en la figura 11 se puede observar la implementación del Velcro con la microfibra.



Figura 11 Implementación de Velcro
Elaborado por: El investigador

Alergias: La elección de la microfibra como material principal para la fabricación de ropa de bebé presenta ventajas significativas, especialmente por que evita posibles alergias. La microfibra es un tejido sintético que se caracteriza por su suavidad y por ser hipoalergénica, además su estructura fina y suave evita la irritación y el roce excesivo en la piel sensible del bebé, minimizando así el riesgo de alergias o reacciones adversas.

3.2.3.1 Construcción del Textil

Una vez que se definen las necesidades del textil, se procede a colocar el circuito y la batería como se muestra de la siguiente manera:

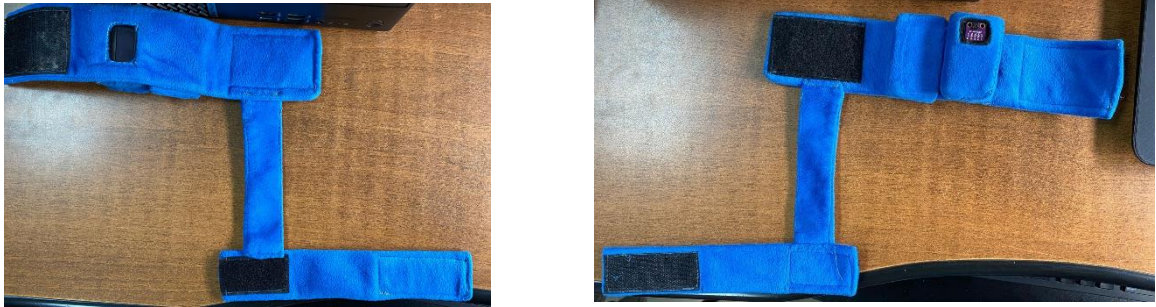


Figura 12 Vista superior e Inferior del Smart Textil
Elaborado por: El investigador

La figura 12 nos muestra una vista de la implementación del dispositivo con el textil, en la parte superior del dispositivo se ubica la batería y el armazón que alberga tanto los sensores como la pantalla. Es importante resaltar que el sensor GY-Max30102 se encuentra sin ningún tipo de recubrimiento, dado que es necesario que este establezca un contacto directo con la piel para realizar las mediciones de manera precisa. Para asegurar un ajuste adecuado alrededor del pie, se utiliza el sistema de cierre con velcro, permitiendo así adaptar el dispositivo a diferentes tamaños y formas de pie. Además, en la parte inferior, se utiliza un envoltorio especial para mantener una posición óptima y un ajuste seguro durante su uso. Esta combinación de elementos garantiza un uso cómodo y efectivo del dispositivo, asegurando un monitoreo preciso de los signos vitales en la zona de interés.

3.2.4 Procesamiento y envío de señales

Dentro del proyecto el ESP32 Feather es utilizado como maestro en la comunicación I2C, permitiendo controlar y enviar instrucciones a otros dispositivos en el bus. Para ello, se requiere configurar e inicializar adecuadamente el bus I2C en el ESP32 Feather. A continuación, se generan instrucciones que se envían a los dispositivos esclavos a través del bus I2C, pudiendo realizar solicitudes de lectura o escritura de datos, así como enviar comandos específicos. Posteriormente, el ESP32 Feather puede recibir datos de los dispositivos esclavos, procesarlos según sea necesario y tomar decisiones adicionales, si corresponde. Los sensores serán configurados de la siguiente manera:

3.2.4.1 Configuración del sensor GY-MAX30102

Para optimizar la configuración del sensor Gy-MAX30102 se tomó en cuenta que el dispositivo se encuentra en la pantorrilla; por lo tanto, la superficie de la piel es más gruesa, los siguientes puntos fueron tratados:

- a) **Aumentar la intensidad de la fuente de luz:** La superficie de la piel es más gruesa, por lo tanto, se aumenta la intensidad de la fuente de luz utilizada por el sensor. Esto puede mejorar la penetración de la luz a través de las capas de la piel y permitir una lectura más precisa.
- b) **Ajuste del tiempo de integración:** El tiempo de integración es el período durante el cual se acumula la luz recibida antes de convertirla en una señal digital. Como la piel es un poco más gruesa, se necesita aumentar el tiempo de integración para capturar una señal suficiente.

Diagrama de flujo y programación del GY-MAX30102

Para la incorporación de las funciones al momento de enviar los datos, se describen dos bloques que engloban la función de cada uno, en la figura 7 se presenta el diagrama de flujo correspondiente a la medición de la frecuencia cardíaca.

El proceso comienza configurando la comunicación serial y luego inicializando el sensor MAX30105. Si el sensor no se encuentra, se muestra un mensaje de error y el programa se detiene.

Después de recopilar las 10 muestras, se imprimen los valores de infrarrojo, frecuencia cardíaca y frecuencia cardíaca promedio, al no detectar un dedo en el sensor, se muestra un mensaje de advertencia. Finalmente, el proceso concluye.

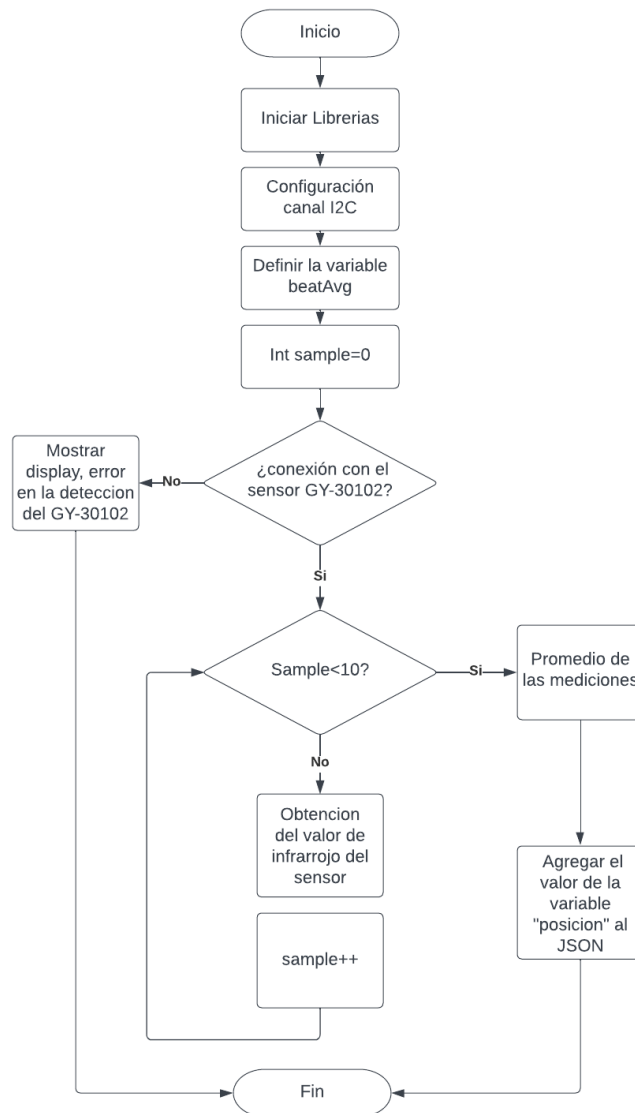


Figura 13 Diagrama de flujo de la función de la frecuencia cardiaca
Elaborado por: El investigador

Como se observa en la figura 7 es un diagrama de flujo dentro del cual tiene más procesos que van a ser descritos en el siguiente bloque de programación que corresponde a la función...

```

while (samples < 10) {

  irValue = particleSensor.getIR();

  if (checkForBeat(irValue) == true) {
    //We sensed a beat!
    long delta = millis() - lastBeat;
    lastBeat = millis();
  }
}

```

```

beatsPerMinute = 60 / (delta / 1000.0);

if (beatsPerMinute < 255 && beatsPerMinute > 20 && irValue > 50000) {
    rates[rateSpot++] = (byte)beatsPerMinute;
    rateSpot %= RATE_SIZE;

    beatAvg = 0;
    for (byte x = 0; x < RATE_SIZE; x++)
        beatAvg += rates[x];
    beatAvg /= RATE_SIZE;
    Serial.println(samples++);
}
}

```

En la línea `irValue = particleSensor.getIR();` se obtiene el valor de infrarrojo del sensor y se almacena en la variable `irValue`. A continuación, se verifica si se ha detectado un latido utilizando la función `checkForBeat(irValue)`. Si la función retorna true, significa que se ha detectado un latido. Dentro del bloque condicional `if(checkForBeat(irValue) == true)`, se realiza lo siguiente:

- Se calcula el tiempo transcurrido desde el último latido en la variable `delta` utilizando la función `millis()`. Se actualiza el tiempo del último latido en la variable `lastBeat` con el valor actual de `millis()`.
- Se verifica si la frecuencia cardíaca calculada (`beatsPerMinute`) está dentro del rango válido (menor a 255 y mayor a 20) y si el valor de infrarrojo (`irValue`) es mayor a 50000.
- Si se cumplen las condiciones anteriores, se almacena el valor de frecuencia cardíaca en el arreglo `rates` en la posición `rateSpot`, y luego se actualiza el índice `rateSpot` de manera circular utilizando el operador `%`.
- Se calcula el promedio de las mediciones de frecuencia cardíaca en la variable `beatAvg`, sumando todos los valores en el arreglo `rates` y dividiéndolos por el tamaño del arreglo (`RATE_SIZE`).
- Se imprime el número de muestra actual en el monitor serial utilizando `Serial.println(samples++)`. De esta manera se guarda dentro de la variable `beatAvg`, luego que se obtienen estos valores se procede a realizar el diagrama para calcular el oxígeno de la sangre, que es similar al realizado para calcular la frecuencia cardíaca.

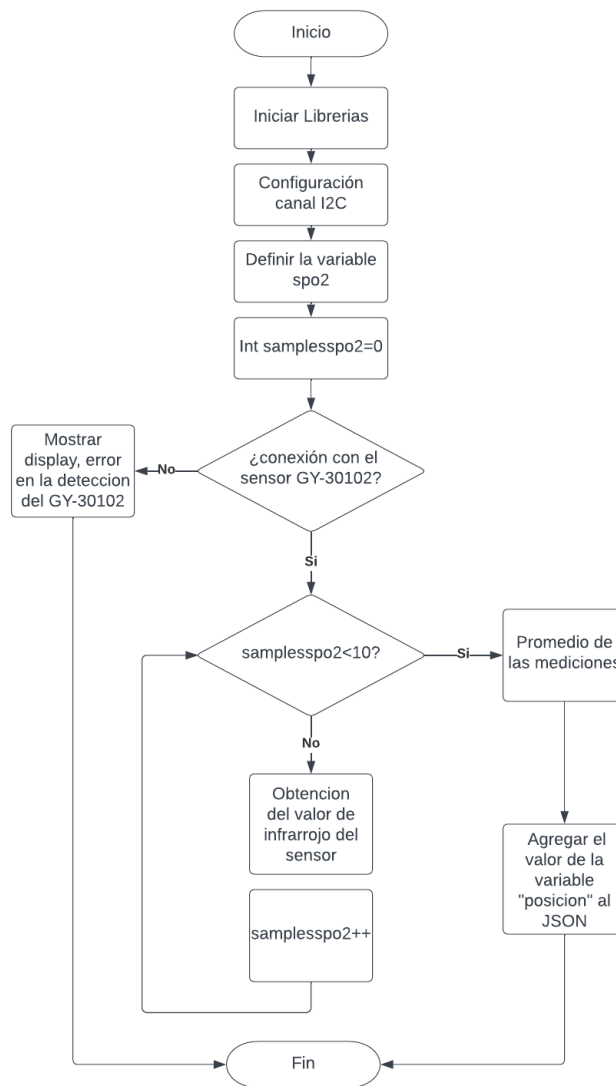


Figura 14 Diagrama de flujo de la función del nivel de Oxígeno en la Sangre
Elaborado por: El investigador

En la figura 14 se presenta un diagrama simplificado que representa el flujo general del proceso; sin embargo, es importante destacar que dentro de este diagrama existen varios procesos adicionales que serán detallados en el próximo bloque de programación correspondiente a la función...

```

void getSP02() {
    bufferSize = 100;

    for (byte i = 0; i < bufferSize; i++) {
        while (particleSensor.available() == false)
  
```

```

        particleSensor.check();

        redBuffer[i] = particleSensor.getRed();
        irBuffer[i] = particleSensor.getIR();
        particleSensor.nextSample();
    }

    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer,
&spo2, &validSPO2, &heartRate, &validHeartRate);

    while (samplespo2 <= 10) {

        for (byte i = 25; i < 100; i++) {
            redBuffer[i - 25] = redBuffer[i];
            irBuffer[i - 25] = irBuffer[i];
        }

        for (byte i = 75; i < 100; i++) {
            while (particleSensor.available() == false)
                particleSensor.check();
            digitalWrite(readLED, !digitalRead(readLED));

            redBuffer[i] = particleSensor.getRed();
            irBuffer[i] = particleSensor.getIR();
            particleSensor.nextSample();

            Serial.print(F("SPO2= "));
            Serial.println(spo2, DEC);
            if (spo2 > 80) {
                samplespo2++;
            }
        }

        if (samplespo2 <= 9) {
            maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer,
&spo2, &validSPO2, &heartRate, &validHeartRate);
        }
    }
    samplespo2 = 0;
    Serial.print(F("SPO2= "));
    Serial.println(spo2, DEC);
}

```

El código presentado implementa la función "getSPO2" para obtener y calcular la saturación de oxígeno en la sangre (SpO2) utilizando el sensor MAX30102. El proceso se lleva a cabo de la siguiente manera:

- Se establece la longitud del búfer de muestras en 100 para almacenar las lecturas del sensor. Se realizan las primeras 100 lecturas del sensor, esperando hasta que estén disponibles los datos. Los valores de la señal roja se almacenan en el arreglo *redBuffer* y los valores de la señal infrarroja se almacenan en *irBuffer*. A continuación, se avanza a la siguiente muestra.
- Se utiliza la función *maxim_heart_rate_and_oxygen_saturation* para calcular la frecuencia cardíaca y la *SpO2* utilizando los datos almacenados en *irBuffer* y *redBuffer*.
- Se inicia un ciclo mientras el número de mediciones de *SpO2* sea menor o igual a 10, luego se imprime el valor de SpO2 en la salida serial y se verifica si es mayor a 80 para contar como una medición válida. Si el número de mediciones válidas de *SpO2* es menor o igual a 9, se realiza nuevamente el cálculo de la frecuencia cardíaca y la *SpO2* utilizando los datos del búfer, se reinicia el contador de mediciones de *SpO2* y se imprime el valor final de *SpO2* obtenido en la salida serial. De esta manera se guarda dentro de la variable *SpO2*, para ser utilizado más adelante, al momento de enviar los datos.

Cálculo del error Porcentual

Una vez se obtiene los datos tanto para la frecuencia cardiaca y la oxigenación de la sangre en la sangre, se elabora una tabla para calcular el porcentaje de error entre las mediciones de este sensor y un oxímetro comercial. Se tomaron un total de 30 muestras para calcular el porcentaje de error del sensor, se utiliza la ecuación 1 para su cálculo:

$$Ep = \left| \frac{x_i - x_v}{x_i} \right| \cdot 100 \% \quad (1) \text{ Ecuación de error porcentual}$$

donde,

$x_i - x_v$: Es la diferencia entre el valor medio del sensor y el dispositivo comercial respectivamente.

$\left| \frac{x_i - x_v}{x_i} \right|$: es el valor absoluto de la relación entre la diferencia entre las mediciones y el valor del sensor. Al dividir la diferencia por x_i , obtienes la diferencia

relativa en relación con la magnitud del sensor. El uso del valor absoluto asegura que el resultado sea siempre positivo.

Tabla 10 Error Porcentual para las mediciones del Ritmo Cardiaco

Muestras	Gy- Max30102 x_i	Oxímetro de pulso (comercial) x_v	$\left \frac{x_i - x_v}{x_i} \right $	Porcentaje de error
1	77,35	81	0,047	4,72 %
2	78,67	83	0,055	5,50 %
3	82,23	79	0,039	3,93 %
4	76,2	78	0,024	2,36 %
5	79,58	83	0,043	4,30 %
6	78,91	81	0,026	2,65 %
7	80,49	83	0,031	3,12 %
8	74,13	78	0,052	5,22 %
9	76,84	79	0,028	2,81 %
10	81,82	86	0,051	5,11 %
11	80,39	84	0,045	4,49 %
12	78,16	79	0,011	1,07 %
13	78,32	82	0,047	4,70 %
14	80,01	83	0,037	3,74 %
15	80,25	85	0,059	5,92 %
16	82,79	79	0,046	4,58 %
17	79,41	80	0,007	0,74 %
18	79,92	79	0,012	1,15 %
19	80,85	81	0,002	0,19 %
20	79,07	80	0,012	1,18 %
21	78,02	77	0,013	1,31 %
22	79,86	81	0,014	1,43 %
23	80,76	79	0,022	2,18 %

24	81,25	83	0,022	2,15 %
25	78,01	76	0,026	2,58 %
26	83,73	80	0,045	4,45 %
27	83,96	81	0,035	3,53 %
28	82,68	83	0,004	0,39 %
29	80,65	77	0,045	4,53 %
30	80,69	81	0,004	0,38 %
PROMEDIO				3,01 %

Elaborado por: El investigador

Tabla 11 Error Porcentual para las mediciones de nivel de Oxígeno en la Sangre

Muestras	Gy- Max30102 x_i	Oxímetro de pulso (comercial) x_v	$\left \frac{x_i - x_v}{x_i} \right $	Porcentaje de error
1	95,44	95	0,005	0,46
2	95,86	98	0,022	2,23
3	95,93	95	0,010	0,97
4	96,73	97	0,003	0,28
5	95,9	96	0,001	0,10
6	96,78	95	0,018	1,84
7	96,58	98	0,015	1,47
8	95,99	95	0,010	1,03
9	94,6	99	0,047	4,65
10	94,4	96	0,017	1,69
11	95	96	0,011	1,05
12	96,96	95	0,020	2,02
13	96,69	97	0,003	0,32
14	97,38	95	0,024	2,44
15	95,59	99	0,036	3,57
16	94,79	98	0,034	3,39

17	98,59	98	0,006	0,60
18	96,18	99	0,029	2,93
19	95,84	99	0,033	3,30
20	96,83	95	0,019	1,89
21	97,23	95	0,023	2,29
22	97,21	97	0,002	0,22
23	94,4	98	0,038	3,81
24	94,32	99	0,050	4,96
25	96,41	95	0,015	1,46
26	94,3	97	0,029	2,86
27	94,86	97	0,023	2,26
28	97,04	99	0,020	2,02
29	96,99	95	0,021	2,05
30	97,71	95	0,028	2,77
PROMEDIO				2,03 %

Elaborado por: El investigador

Basado en un análisis de 30 muestras, se ha determinado que el sensor GY-MAX30102 presenta un error porcentual del 3,01% en la medición de la frecuencia cardíaca (BPM) y del 2,03% en la medición de la saturación de oxígeno en la sangre en comparación con un oxímetro comercial. Estos resultados se encuentran documentados en las tablas 10 y 11.

El rendimiento del sensor GY-MAX30102 ha demostrado ser satisfactorio en términos de precisión y confiabilidad en el monitoreo de la frecuencia cardíaca y la saturación de oxígeno, lo que lo convierte en una opción viable para aplicaciones de alerta y monitoreo en tiempo real. Cabe destacar que este sensor se encuentra diseñado para aplicaciones de alerta y monitoreo general, y no está destinado a ser utilizado para diagnósticos médicos, donde se requiere una mayor precisión.

3.2.4.2 Configuración del sensor MPU6050

En primer lugar, la necesidad de identificar y distinguir estas tres posiciones clave en el contexto del monitoreo del lactante. La posición supina corresponde a cuando el niño se encuentra acostado boca arriba, la posición prona se refiere a cuando el niño

está acostado boca abajo, y la posición neutra se da cuando el niño se encuentra en posición vertical o sentado como se muestra en la figura 7. Estas posiciones son relevantes para obtener información valiosa sobre la salud y el bienestar del lactante.

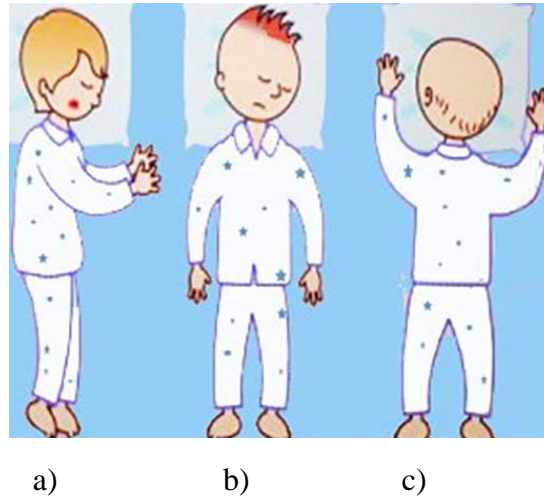


Figura 15 Posición de descanso de un lactante a) Neutral b) Supino c) Prono [48]

La posición prona, en la cual el lactante se encuentra acostado boca abajo, es conocida por aumentar el riesgo de ahogamiento en los bebés. Es por eso por lo que es crucial detectar esta posición y tomar medidas adecuadas para prevenir cualquier situación de riesgo.

Para la detección de la posición del bebé utilizando el sensor MPU6050, se puede utilizar el eje de aceleración z (eje vertical) como referencia principal como se muestra en la figura ya que esta hace una rotación completa con referencia al chip principal. Este eje permite distinguir entre las posiciones supino, prono y neutra del lactante, como se muestra en la figura 15

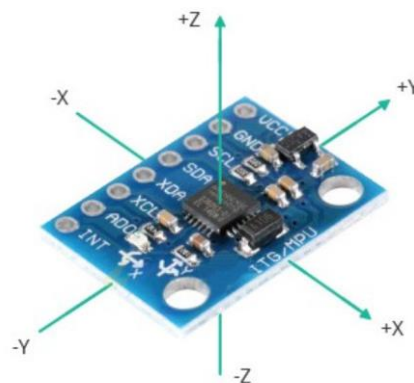


Figura 16 Posiciones de detección de un MPU6050 [49]

El acelerómetro del MPU6050 se utiliza para medir la velocidad angular o la tasa de cambio de la orientación del sensor en los tres ejes (x, y, z) como se muestra en la figura 16. Esto permite detectar movimientos de rotación o cambios de posición más sutiles, como el acercamiento a una posición prono a partir de una posición neutral.

Diagrama de flujo y programación del MPU6050

La librería que es utilizada es *Adafruit_MPU6050.h* que es propia del fabricante, la referencia del acelerómetro parte desde un valor máximo de 9.8 m/s^2 , y de un valor negativo cuando este se encuentra girado en 360 grados, la programación se realizó de la siguiente manera:

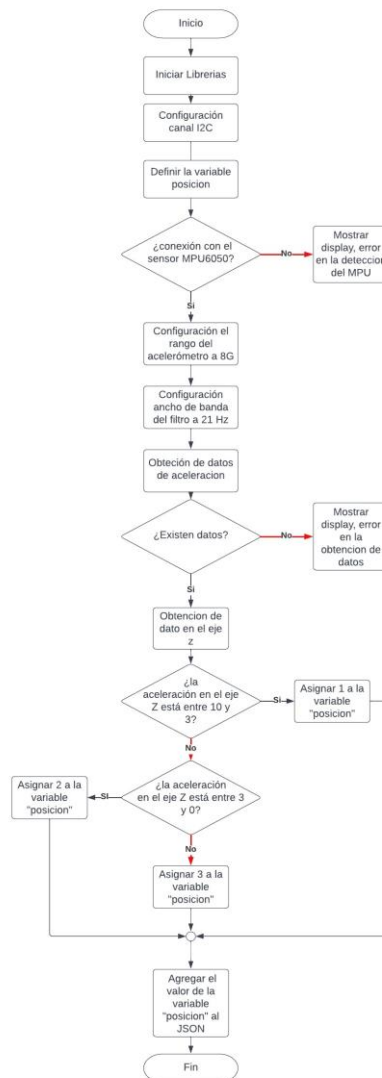


Figura 17 Diagrama de flujo de la función de MPU6050

Elaborado por el Investigador

La figura 17 muestra el diagrama de flujo de función de MPU6050, describe el proceso de configuración y envío de datos utilizando el sensor. Primero, se configura el puerto serie y se verifica la conexión con el sensor. Si no se encuentra el chip MPU6050, se muestra un mensaje de error y se detiene el programa. Luego, se configuran los rangos del acelerómetro y el giroscopio, así como el ancho de banda del filtro. Después de una espera de 2 segundos, se inicia un ciclo repetitivo. En cada iteración, se obtienen los datos de aceleración, giroscopio y temperatura del sensor. Estos datos se agregan a un objeto JSON. Luego, se verifica el valor de la aceleración en el eje Z y se asigna una posición correspondiente. La posición se agrega al JSON. A continuación, el JSON se convierte en una cadena de texto y se envía. El ciclo se repite con un retardo de 500 milisegundos entre cada iteración.

El fragmento de la programación se muestra a continuación:

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;
byte posicion = 0;
void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10);

  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");

  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  Serial.print("Accelerometer range set to: ");
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
  Serial.print("Filter bandwidth set to: ");

  Serial.println("");
  delay(2000);
}
```

```

void loop() {

  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  if (a.acceleration.z <= 10 && a.acceleration.z > 3) {
    posicion = 1;
  } else if (a.acceleration.z < 3 && a.acceleration.z > 0) {
    posicion = 2;
  } else if (a.acceleration.z < 0) {
    posicion = 3;
  }

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");

  Serial.println("");
  delay(500);
}

```

Dentro de este código se encuentra la posición del lactante, se adjunta a un objeto JSON.

3.2.4.3 Configuración para el cálculo del nivel de batería

Uno de los objetivos principales del sistema es lograr la portabilidad sin depender de un cable de alimentación. Para ello, se utiliza una batería de polímero de litio de 3.7V y 1800mAh de la marca EEMB. Esta batería se selecciona por varias razones. En primer lugar, viene equipada con un conector JST como se muestra en la figura 18, lo que facilita su reemplazo y permite una conexión sencilla al dispositivo. Además, la capacidad de 1800mAh proporciona una duración de batería adecuada para el funcionamiento del sistema sin interrupciones.



Figura 18 Batería de polímero de litio con cable conector JST [50]

El ESP32 Feather tiene un pin analógico, denominado ADC_BAT, que se puede utilizar para medir el voltaje de la batería; sin embargo, antes de utilizar este pin para medir el voltaje de la batería, se recomienda seguir los pasos a continuación:

- Conectar el terminal positivo de la batería al pin ADC_BAT del ESP32 Feather y el terminal negativo de la batería al pin GND (tierra) del ESP32 Feather.
- Hay que asegurar de que el ESP32 Feather esté alimentado únicamente por la batería de 3.7V 1800mAh, sin ninguna fuente de alimentación externa.

Para determinar el porcentaje de carga de la batería de la indumentaria, se considera que opera a un voltaje de 3.3V. Dado que la batería de litio funciona con este voltaje, podemos realizar el siguiente cálculo utilizando la ecuación 2 en el pin analógico:

$$\%V = \left(\frac{rawValue}{4095} \right) \cdot 2 \cdot 1.1 \cdot 3.3 \cdot 100$$

donde,

- **rawValue:** Es el valor crudo (sin procesar) obtenido de la lectura analógica del pin ADC (Convertidor Analógico-Digital) del ESP32.
- **4095.0:** Es el valor máximo que puede tener el ADC, que corresponde a una lectura de referencia de 3.3V.
- **2:** Se utiliza para ajustar el rango de valores de 0 a 3.3V a un rango de 0 a 6.6V, ya que se necesita tener en cuenta la referencia de voltaje de 1.1V del ESP32.
- **1.1:** Es la referencia de voltaje interna del ESP32.
- **3.3:** Es el voltaje de alimentación del ESP32.

A través de la programación del microcontrolador, podemos implementar una función que nos permita obtener el nivel de voltaje. A continuación, se presenta el bloque de código correspondiente a esta funcionalidad:

```
int getBattery() {
    int rawValue = analogRead(ADC_BAT);
    float voltageLevel = (rawValue / 4095.0) * 2 * 1.1 * 3.3;
    float batteryFraction = voltageLevel / MAX_BATTERY_VOLTAGE;
    return batteryFraction * 100;
}
```

El código proporciona una función llamada "getBattery" utilizada para obtener el nivel de carga de la batería en forma de un porcentaje. A continuación, se detalla cómo funciona el código:

- Se lee el valor crudo de voltaje de la batería utilizando la función *analogRead* en el pin ADC_BAT. El valor leído se almacena en la variable *rawValue*.
- El ESP32, tiene una referencia de voltaje de 1.1V. Esta referencia se utiliza para convertir el valor crudo leído en un nivel de voltaje real. Se multiplica el valor crudo por 2 para ajustar el rango a 0-1.
- Se multiplica por 1.1 y 3.3 para escalar el valor a la tensión real de la batería, el resultado obtenido es el nivel de voltaje de la batería en volts y se almacena en la variable *voltageLevel*. A continuación, se calcula la fracción de la batería dividiendo el nivel de voltaje actual entre el voltaje máximo de la batería (MAX_BATTERY_VOLTAGE). Esto proporciona una medida relativa de la carga de la batería. Finalmente, se multiplica la fracción de la batería por 100 para obtener el nivel de carga de la batería en forma de un porcentaje.
- El valor resultante se devuelve como un entero. Una vez que se encuentra este valor se adjunta a un objeto JSON que se verá más adelante con todos los datos hallados.

3.2.4.5 Envío de Señales

El envío eficiente de datos requiere utilizar un formato que sea sencillo y fácil de procesar. En este sentido, el formato JSON se destaca como una solución idónea. JSON (JavaScript Object Notation) ofrece la capacidad de representar información de

manera concisa y estructurada, lo que facilita su manipulación y procesamiento. Al utilizar JSON, los datos se vuelven más accesibles y comprensibles tanto para el sistema de monitoreo como para la base de datos, además de que este es el formato que utiliza la base de datos NoSQL, el formato que se va a enviar es el siguiente:

```
1  {
2    "bateria": "100",
3    "bpm": "80",
4    "oxigenacion": "93",
5    "posicion": "1",
6    "timestamp": "1650329217000"
7  }
```

Figura 19 Ejemplo de la estructura JSON
Elaborado por el Investigador

En la figura 19 se observa un objeto JSON, los datos se organizan en pares clave-valor, donde una clave es una cadena que identifica un valor asociado. El valor puede ser de diferentes tipos, como una cadena de texto, un número, un booleano, otro objeto JSON o una lista.

La explicación de cada una de las propiedades:

- **bateria:** representa el nivel de batería en forma de porcentaje, en este caso, el valor es "100"
- **bpm:** indica las pulsaciones por minuto del usuario, siendo "80" el valor actual.
- **oxigenacion:** muestra el nivel de oxigenación en la sangre, siendo "93" el valor actual.
- **posicion:** indica la posición actual, donde "1" podría representar una posición específica.
- **timestamp:** representa la marca de tiempo en formato "since epoch" en milisegundos. En este caso, el valor es "1650329217000", lo que indica que se generó en un momento específico.

Para que estos datos sean enviados dentro de la programación del microcontrolador se utiliza el siguiente bloque de programación:

```

void httpSend() {
    WiFiClientSecure* client = new WiFiClientSecure;
    if (client) {
        client->setCACert(rootCACertificate);
        {
            HTTPClient https;
            Serial.print("[HTTPS] begin...\n");
            if (https.begin(*client, serverName)) {
                Serial.print("[HTTPS] POST...\n");
                String valores = "{\"bateria\": \"" + String(BatteryP) + "\", \"bpm\": \"" +
                    String(beatAvg) + "\", \"oxigenacion\": \"" + String(spo2) +
                    "\", \"posicion\": \"" + String(posicion) +
                    "\", \"timestamp\": \"" + String(getTime()) + "000\"}";
                int httpCode = https.POST(valores);
                Serial.printf("[HTTPS] POST... code: %d\n", httpCode);
                if (httpCode = 200) {
                    u8g2.clearBuffer();
                    u8g2.drawStr(30, 39, "Enviado");
                    u8g2.sendBuffer();
                }
                https.end();
            } else {
                Serial.printf("[HTTPS] Unable to connect\n");
            }
        }

        delete client;
    } else {
        Serial.println("Unable to create client");
    }
}

```

Dentro de este código se crea un objeto *WiFiClientSecure** llamado *client* para establecer una conexión utilizando WiFi. Si se ha creado correctamente, se procede con el siguiente paso. Se configura el certificado de autoridad de certificación (CA) del cliente mediante la función *setCACert()* en *client*. El certificado de CA es necesario para establecer una conexión segura con el servidor, luego se crea un objeto *HTTPClient* llamado *https*, se muestra un mensaje en el puerto serial indicando el inicio de la comunicación HTTPS. Se utiliza la función *begin()* de *https* para establecer la comunicación con el servidor especificado en *serverName*.

Se muestra un mensaje en el puerto serial indicando el inicio del envío de la solicitud POST. Se construye una cadena de texto en formato JSON utilizando diferentes valores de variables, como "bateria", "bpm", "oxigenacion", "posicion" y "timestamp". Estos valores se concatenan a la cadena en formato JSON, luego de ser guardado en sus respectivas variables al cumplir las funciones vistas anteriormente.

Se realiza una solicitud POST al servidor mediante la función "*POST()*" de la biblioteca HTTPS, donde se pasa la cadena de texto JSON como parámetro. Si el código de respuesta es 200 (indicando éxito), se ejecutan las siguientes instrucciones: se borra el contenido de la pantalla OLED, se muestra el mensaje "Enviado" en la posición especificada de la pantalla, y se finaliza la comunicación HTTPS utilizando la función *end()* de la biblioteca HTTPS. En caso de que no se pueda establecer la conexión con el servidor, se muestra un mensaje en el puerto serial indicando la incapacidad de conexión.

3.2.5 Desarrollo de aplicación móvil

Al desarrollar la aplicación móvil, se consideran los siguientes requisitos que deben cumplirse:

- Almacenamiento de información de usuarios
- Recepción en tiempo real de datos recopilados por la instrumentaria.
- Generación de alertas según valores anormales.

En la sección 1.3.7 y 1.3.8 se estableció el marco de trabajo y el servicio de backend que se emplearán en el diseño de la aplicación móvil. De esta manera, se define la arquitectura que guiará el proceso de desarrollo y codificación.

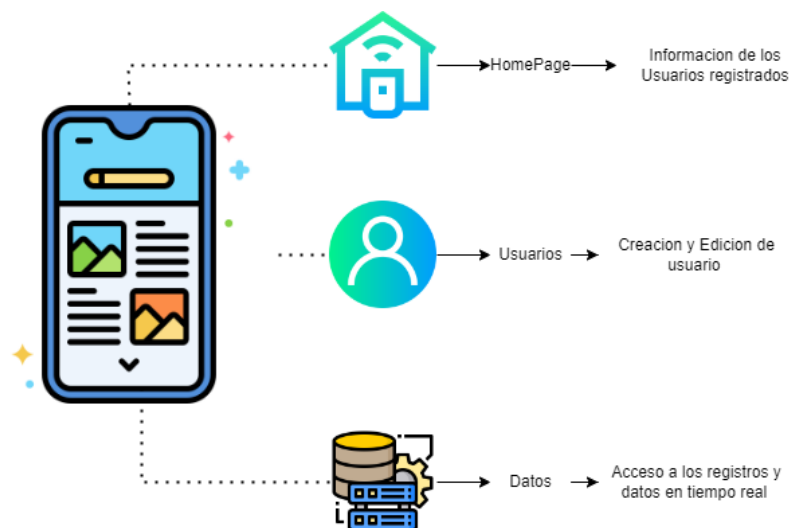


Figura 20 Arquitectura de las funciones del aplicativo móvil
Elaborado por: El investigador

La figura 20 muestra la arquitectura de nuestra aplicación móvil, la cual ha sido diseñada de manera intuitiva y sencilla para ser utilizada por los encargados del cuidado de los lactantes. Aunque cuenta con un conjunto limitado de funciones, estas se han seleccionado cuidadosamente para facilitar su uso. A continuación, se describirán en detalle dichas funciones:

Homepage: La pantalla de la aplicación variará según la información almacenada en la base de datos. Si no hay información disponible, se mostrará un mensaje de bienvenida y un botón que permitirá crear nuevos usuarios. Si hay información registrada, se mostrará una lista de todos los usuarios previamente registrados. Además, mientras el dispositivo envía datos, se puede visualizar en tiempo real en el perfil del niño seleccionado.

Usuarios: La función muestra un formulario donde se pueden ingresar los datos necesarios para crear un nuevo usuario. Los campos incluyen nombre, apellido, sexo, edad, tipo de prenda utilizada, lugar de descanso, opción para indicar si está alimentado y un campo para agregar observaciones adicionales.

Datos: Al igual que en la página de inicio, esta función mostrará los datos según la información almacenada en la base de datos. Si el dispositivo está enviando datos, se visualizará una tabla en tiempo real con los signos vitales. En caso de que el dispositivo

no esté enviando datos, se mostrará una lista con los registros previos de datos medidos de cada usuario.

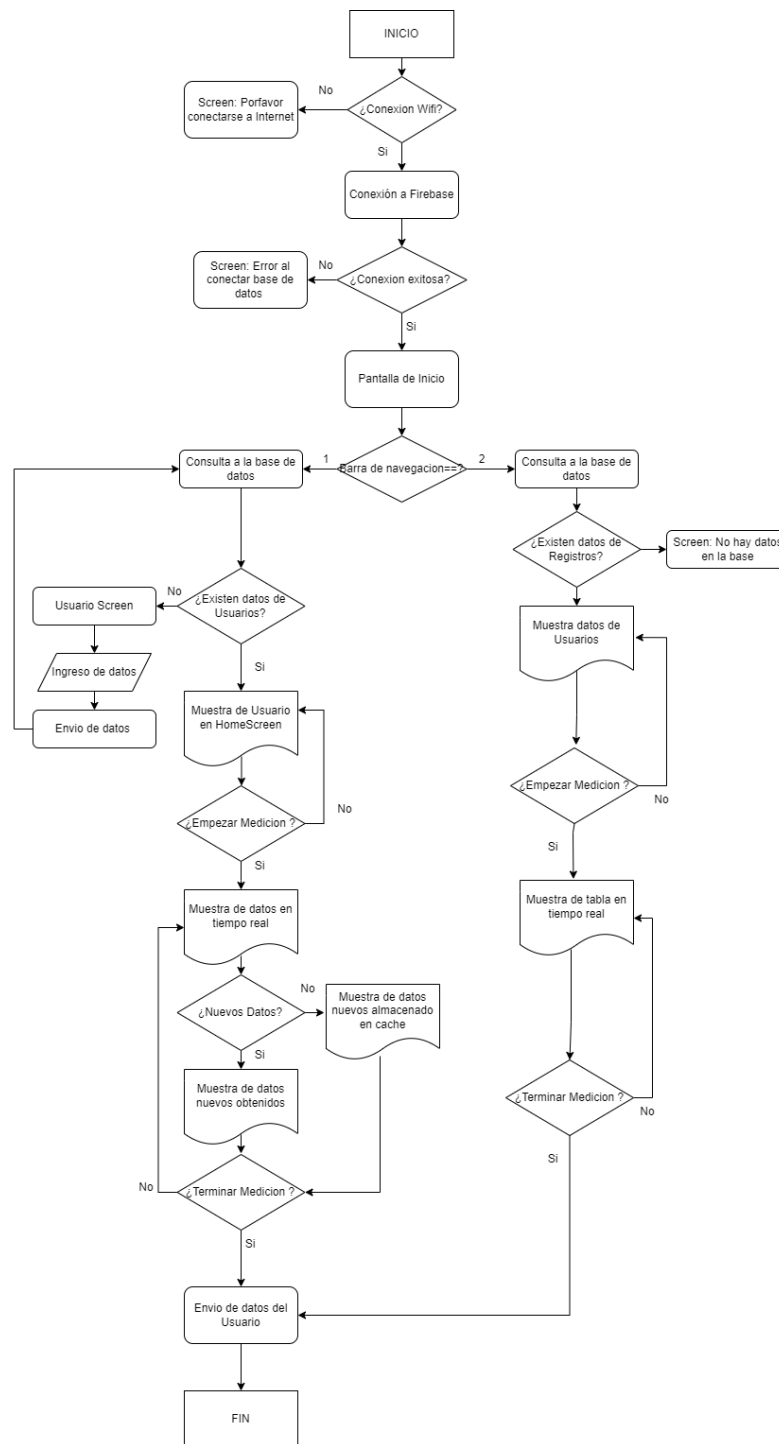


Figura 21 Flujograma de la aplicación Móvil

La figura 21 representa la lógica de nuestro proyecto, estableciendo así el punto de partida para la creación de la aplicación en el framework Flutter. Es importante tener en cuenta que Firebase será nuestra plataforma de backend.

3.2.5.1 Desarrollo de proyecto en Firebase

No es necesario proporcionar una explicación detallada sobre cómo crear un proyecto en Firebase, debido a que el proceso es relativamente sencillo y se puede encontrar una guía completa y precisa en la documentación oficial de Firebase.

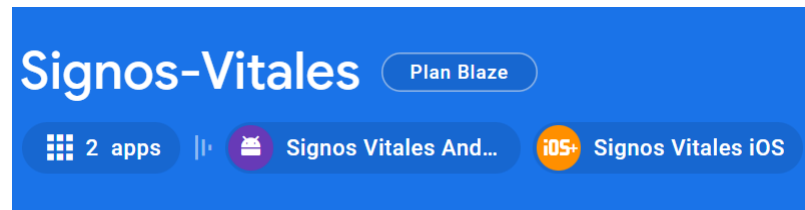


Figura 22 Creacion de proyecto en Firebase
Elaborado por: El investigador

En la figura 22 se creó el proyecto con el nombre Signos-Vitales, en esta también se puede observar las configuraciones que son utilizados en un proyecto de Flutter, en nuestro caso es Android, por lo tanto, nuestras configuraciones son las siguientes:

Primero, se crea un proyecto en Firebase Console, se descarga el archivo de configuración google-services.json. A continuación, se agregan las dependencias de Firebase en el archivo build.gradle y se importa el plugin de Firebase en el archivo MainActivity. Después de sincronizar las dependencias y realizar un *rebuild*, el proyecto estará listo para utilizar Firebase y sus servicios, como Firebase Authentication o Firebase Cloud Firestore.

3.2.5.2 Desarrollo del proyecto en Flutter

Para la creación del proyecto se ejecuta en modo administrador en la terminal windows, el siguiente comando:

```
flutter create signos_vitales
```

Una vez que el comando se ejecuta correctamente, se crea una carpeta que contiene todos los archivos necesarios para el proyecto. Para acceder a esta carpeta, utilizamos el IDE Visual Studio Code. Abrimos Visual Studio Code y navegamos hasta la ubicación donde se ejecutó el comando anterior.

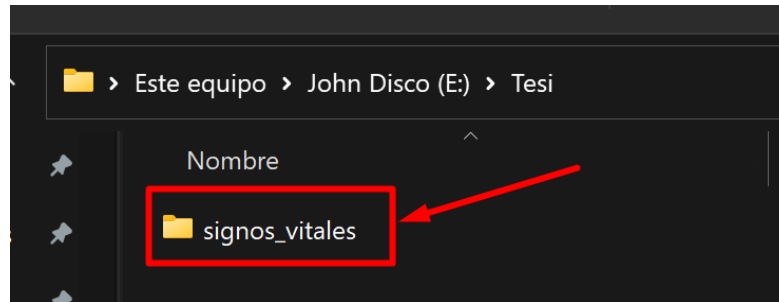


Figura 23 Carpeta creada del proyecto de Flutter
Elaborado por: El investigador

En la figura 23 se puede observar la carpeta, una vez abierto el editor Visual Studio Code, accedemos a esta, y podremos visualizar el contenido a continuación

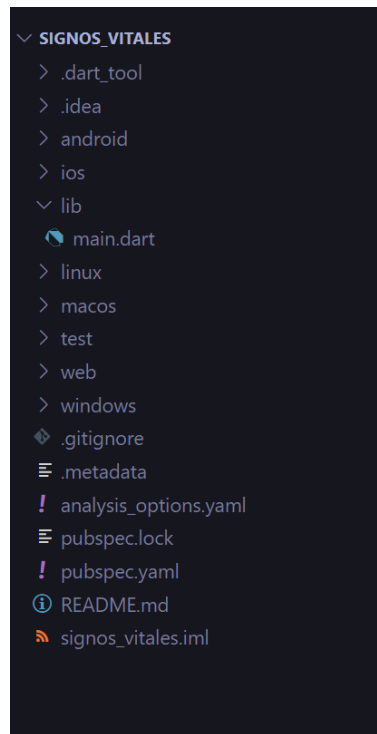


Figura 24 Contenido de un proyecto en Flutter
Elaborado por: El investigador

En la figura 24 se observa el contenido creado para el proyecto, empezaremos la codificación es el archivo “*main.dart*” que se encuentra en la carpeta “*lib*”, este es un archivo de Dart con el nombre "*main.dart*". Este archivo contiene la función "*main()*", que es el punto de entrada para la ejecución de la aplicación. Dentro de la función "*main()*" se define la estructura inicial de la aplicación y se inicia la ejecución del código.

- **Conexión del proyecto de Flutter con Firebase**

Para establecer la conexión entre el backend y el proyecto en Flutter utilizaremos una librería proporcionada por la documentación de Firebase llamada "*firebase_core*". Esta librería nos permitirá integrar de manera efectiva las funcionalidades de Firebase en nuestra aplicación Flutter. En la figura 18 se puede observar un archivo llamado "*pubspec.yaml*" un archivo de configuración que se utiliza para gestionar las dependencias del proyecto. En este archivo se especifican las dependencias de paquetes externos que se requieren en el proyecto.

Agregamos la dependencia de "*firebase_core*" en la sección "dependencias" como se muestra en el siguiente bloque:

```
dependencies:  
  firebase_core: ^2.8.0
```

Una vez guardado el archivo, en la terminal de Visual Studio Code ejecutamos el comando "*flutter pub get*" para descargar e instalar las dependencias de la librería. Después de completar esta acción, importamos "*firebase_core*" en el archivo "*main*" para asegurarnos de que la conexión se establezca antes de generar la pantalla de inicio de la aplicación, la línea para importar la librería es la siguiente:

```
import 'package:firebase_core/firebase_core.dart';
```

Luego, inicializamos Firebase en nuestro proyecto:

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firebase.initializeApp();  
  runApp(MyApp());  
}
```

Este código es el punto de entrada de una aplicación Flutter. Aquí se establece la conexión con Firebase y se inicia la ejecución de la aplicación.

- ***void main() async***: Esta es la función principal del programa. El modificador *async* indica que la función puede contener operaciones asíncronas y el tipo de retorno *void* indica que la función no devuelve ningún valor.

- ***await Firebase.initializeApp():*** Esta línea espera a que se complete la inicialización de Firebase antes de continuar. La función `initializeApp()` inicializa la conexión con Firebase y configura los servicios necesarios para interactuar con la plataforma.
- ***runApp(MyApp()):*** Esta línea inicia la ejecución de la aplicación. `MyApp()` es una clase que define la estructura y el comportamiento de la aplicación y se pasa como argumento a la función `runApp()`, que la renderiza y muestra en la pantalla del dispositivo.

Programación de Pantallas

El desarrollo del código en Flutter se basará en el diagrama de flujo presentado en la figura 21. Se enfocará en implementar el funcionamiento de cada una de las tres pantallas requeridas. El código completo del diseño se encuentra disponible en los Anexos debido a su extensión.

Pantalla de Registro de Usuario

En el centro de Estimulación Temprana, es fundamental recopilar y registrar datos de los niños con el fin de conocer su estado actual y proporcionar un seguimiento adecuado. Estos datos nos brindan información valiosa sobre el desarrollo y las necesidades individuales de cada niño. Mediante la recopilación de datos relevantes,

Figura 25 Pantalla de Registro/Edición de datos
Elaborado por: El investigador

En la figura 25 se muestra el diseño de un formulario que recolecta los siguientes datos:

Nombre y Apellido: Estos datos nos permiten identificar de manera única a la persona, lo cual es fundamental para asegurar que los registros y mediciones correspondan a la persona correcta.

Sexo: El sexo puede influir en ciertos aspectos fisiológicos, como los rangos de referencia de los signos vitales. Por ejemplo, los valores normales de la presión arterial pueden variar entre hombres y mujeres.

Edad: La edad es un factor importante en la interpretación de los signos vitales.

Prenda que usa: La ropa que el lactante está usando puede afectar la medición de ciertos signos vitales. Por lo tanto, es importante registrar este dato para considerarlo al realizar las mediciones.

Lugar de Descanso: El entorno donde encuentra el lactante también puede influir en los signos vitales. Registrar el lugar de descanso nos permite tener en cuenta este factor al interpretar los resultados.

Alimentación: La alimentación puede tener un impacto en los signos vitales, especialmente en aquellos relacionados con la digestión, como la frecuencia cardíaca y la presión arterial. Conocer el estado de alimentación de la persona nos ayuda a comprender mejor los cambios en los signos vitales.

Observaciones: Las observaciones proporcionan detalles adicionales, pueden incluir síntomas, cambios en el estado de salud o cualquier otro factor relevante. Estas observaciones complementan los datos cuantitativos de los signos vitales y nos brindan un contexto más completo para la interpretación.

Implementación de la funcionalidad de Registro de Usuarios

La base de datos que es creada en Firebase es conocida como Firebase Realtime Database, que es una base de datos en tiempo real que permite almacenar y sincronizar datos en la nube. En este caso, el código se centra en la funcionalidad de registro de usuarios. Al inicializar la pantalla, el mapa de datos tiene la siguiente estructura:

```
final _data = {
  'nombre': '',
  'apellido': '',
  'sexo': 'hombre',
  'edad': '',
  'prenda': '',
  'descanso': '',
  'alimentado': false,
  'observaciones': '',
};
```

Una vez completado el formulario, al presionar el botón de guardar, se activa la función `send()` que está definida dentro de la clase `SendData()`. El código correspondiente se muestra a continuación:

```
class SendData {
  final Map<String, dynamic> data;

  SendData({required this.data});

  Future<void> send() async {
    final databaseRef = FirebaseDatabase.instance.ref().child('Usuario');
    ;

    final String nodeName = '${data['nombre']} + ' ' + data['apellido']};
```

```
DatabaseEvent snapshot = await databaseRef.child(nodeName).once();

if (snapshot.snapshot.value != null) {
    print('Ya existe un nodo con ese nombre');
    return;
}

data['fecha_creacion'] = DateTime.now().toString();

await databaseRef.child(nodeName).set(data);
print('Datos enviados exitosamente a Firebase');
}
}
```

La clase *SendData* recibe un parámetro llamado *data*, que es un mapa de clave-valor que contiene los datos del usuario a registrar. La función *send()* se encarga de enviar los datos a Firebase Realtime Database. En primer lugar, se obtiene una referencia a la instancia de la base de datos mediante *FirebaseDatabase.instance.ref()*, y se especifica el nodo "Usuario" como destino de los datos. A continuación, se construye un nombre de nodo único utilizando los valores de nombre y apellido del mapa de datos. Se verifica si ese nodo ya existe en la base de datos mediante *databaseRef.child(nodeName).once()*. Si el valor de *snapshot.snapshot.value* no es nulo, significa que ya existe un nodo con ese nombre y se imprime un mensaje indicando dicha situación. En ese caso, se finaliza la función sin realizar ninguna acción adicional.

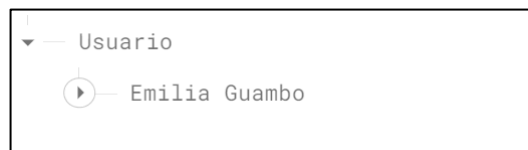


Figura 26 Ejemplo de creación de un nodo de Usuario
Elaborado por: El investigador

En el ejemplo de la figura 26, se muestra la creación de un nodo específico dentro del nodo principal. Esto indica el almacenamiento de un conjunto de datos relacionados bajo un subnodo específico, lo que ayuda a mantener la organización y la integridad de la base de datos. La estructura de datos jerárquica facilita la organización y recuperación de la información, debido a que pueden acceder rápidamente a los

subnodos dentro del nodo principal. Además, permite establecer relaciones entre los datos y mantener una coherencia en la estructura de la base de datos.



Figura 27 Datos del usuario en Firebase
Elaborado por: El investigador

En la figura 27 se puede apreciar que los datos se han guardado correctamente en Firebase Realtime Database. Además, se puede observar que se ha incluido la fecha de creación junto con los demás datos. Esta incorporación de la fecha de creación permite una organización ascendente de los registros en la base de datos, de esta manera se puede observar que el bloque funciona correctamente.

Pantalla Principal

La pantalla principal es una parte fundamental de la interfaz que presentará la información clave. Aquí se mostrarán los datos de cada usuario registrado, permitiendo tener un acceso rápido y sencillo a la medición de los signos vitales. Además, se dispondrán botones intuitivos que agilizarán el proceso de inicio de las mediciones.



Figura 28 Pantalla de Inicio de la app
Elaborado por: El investigador

En la figura 28, se puede apreciar la representación visual de los datos almacenados, resaltados en el recuadro negro número 2. Esta sección proporciona una vista rápida de la situación actual del perfil, brindando una visión general de los datos relevantes. Justo debajo de esta sección, se encuentran los botones de inicio, los cuales permiten seleccionar el perfil correspondiente y comenzar a capturar los datos específicos de dicho perfil. Además, se ofrecen opciones para editar o eliminar los datos en caso de corregir alguna información. Es importante destacar que el recuadro 1 abarca todos estos elementos y se presenta como un panel deslizante que se adapta dinámicamente a la cantidad de perfiles de usuarios existentes, ofreciendo una experiencia intuitiva y funcional. En el caso de que no se encuentren datos tendremos la pantalla de presentación.

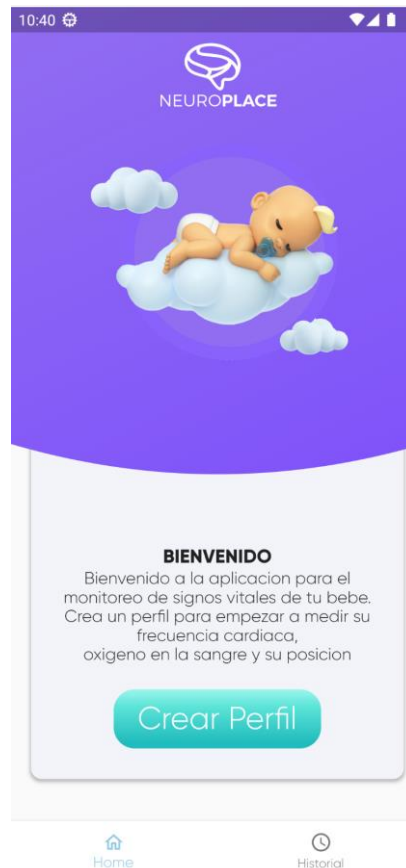


Figura 29 Pantalla de inicio sin base de datos
Elaborado por: El investigador

En la figura 29 se presenta un diseño que muestra información relevante. En este caso específico, se observa que no hay datos disponibles en la base de datos. Para iniciar la recolección de datos de los usuarios, se muestra un botón que permite iniciar dicho proceso. En la parte inferior se encuentran los botones de navegación que nos facilitan desplazarnos entre la pantalla de datos y la pantalla de inicio.

Adquisición de datos

Una vez que se comienza a enviar los datos a nuestra base de datos, estos se reflejarán en tiempo real en la misma pantalla. No se produce un cambio completo de pantalla, pero se realizan pequeñas actualizaciones para mostrar la información actualizada en tiempo real. De esta manera, los usuarios pueden ver de forma inmediata los datos que han sido enviados y almacenados en la base de datos sin necesidad de cambiar de pantalla.



Figura 30 Visualización de datos en tiempo real
Elaborado por: El investigador

En la parte inferior de la figura 30 se pueden observar los tres datos principales que serán medidos. Estos valores se actualizarán en tiempo real en la pantalla a medida que haya cambios en la base de datos de Firebase. Esto significa que los códigos de los datos se modificarán automáticamente en la pantalla cuando se produzcan cambios en la base de datos. El código para almacenar los datos que se reciben es el siguiente:

```
class DatosRealTime {
    final String bpm;
    final String posicion;
    final String oxigenacion;
    final String bateria;
    final String timestamp;

    DatosRealTime(
        {required this.bpm,
        required this.posicion,
        required this.oxigenacion,
        required this.bateria,
        required this.timestamp});
}
```



```

factory DatosRealTime.fromRTDB(Map<dynamic, dynamic> data) {
  return DatosRealTime(
    bpm: data['bpm'] ?? 0,
    posicion: data['posicion'] ?? 0,
    bateria: data['bateria'] ?? 0,
    oxigenacion: data['oxigenacion'] ?? 0,
    timestamp: data['timestamp'] ?? 0);
}
}

```

La clase *DatosRealTime* tiene cinco propiedades: bpm, posicion, oxigenacion, bateria y timestamp, todas ellas son de tipo String. Estas propiedades representan diferentes aspectos de los datos en tiempo real, como la frecuencia cardíaca (bpm), la posición, la oxigenación de la sangre, el nivel de batería y fecha.

Dentro del constructor *factory*, se accede a las diferentes claves del mapa dinámico para obtener los valores correspondientes a cada propiedad de *DatosRealTime*. Si alguna clave no está presente en el mapa, se proporciona un valor predeterminado de 0 para evitar errores. Al definir una clase específica para estos datos se facilita la manipulación y visualización de los cambios en la interfaz de usuario de Flutter, esto permite una mejor organización del código y una mayor facilidad para trabajar con los datos en tiempo real en diferentes partes de la aplicación.

Luego de crear esta clase para el almacenamiento de datos continuamos con el siguiente bloque de código:

```

import 'package:flutter/foundation.dart';
import 'package:firebase_database/firebase_database.dart';
import 'package:signos_vitales/models/datos_tiemporeal.dart';

class RealtimeProvider with ChangeNotifier {
  final _database = FirebaseDatabase.instance.ref();
  DatosRealTime _datos = DatosRealTime(
    bpm: "0",
    posicion: "0",
    oxigenacion: "0",
    bateria: "0",
    timestamp: "0",
  );

  DatosRealTime get datos => _datos;
  List<DatosRealTime> _datosList = [];

  RealtimeProvider() {

```

```

    _initRealtimeDatabaseListener();
    // _tableData();
}

void _initRealtimeDatabaseListener() {
    final DatabaseReference dbRef = _database.child('DatoTiempoReal');
    dbRef.onChildAdded.listen((event) {
        DataSnapshot dataSnapshot = event.snapshot;
        Map<dynamic, dynamic> data = dataSnapshot.value as Map;
        _datos = DatosRealTime.fromRTDB(data);
        notifyListeners();
    });
}
}

```

El código proporcionado muestra la implementación de un proveedor en Flutter que utiliza la base de datos en tiempo real de Firebase para obtener y mostrar datos en tiempo real en la aplicación. En primer lugar, se importan los paquetes necesarios para el funcionamiento del código. El paquete *'flutter/foundation'* utiliza la clase *ChangeNotifier*, que permite notificar a los widgets interesados sobre los cambios en los datos. El paquete *'firebase_database'* se utiliza para interactuar con la base de datos en tiempo real de Firebase. Además, se importa el archivo *'signos_vitales/models/datos_tiemporeal.dart'*, que contiene la definición de la clase *DatosRealTime*, la cual representa los datos en tiempo real. La clase *RealtimeProvider* es un proveedor que extiende la clase *ChangeNotifier*, este proveedor es responsable de gestionar los datos en tiempo real y notificar a los widgets interesados sobre los cambios. Los datos que tenemos en la base de datos que fueron enviados por el dispositivo electrónico es el siguiente:

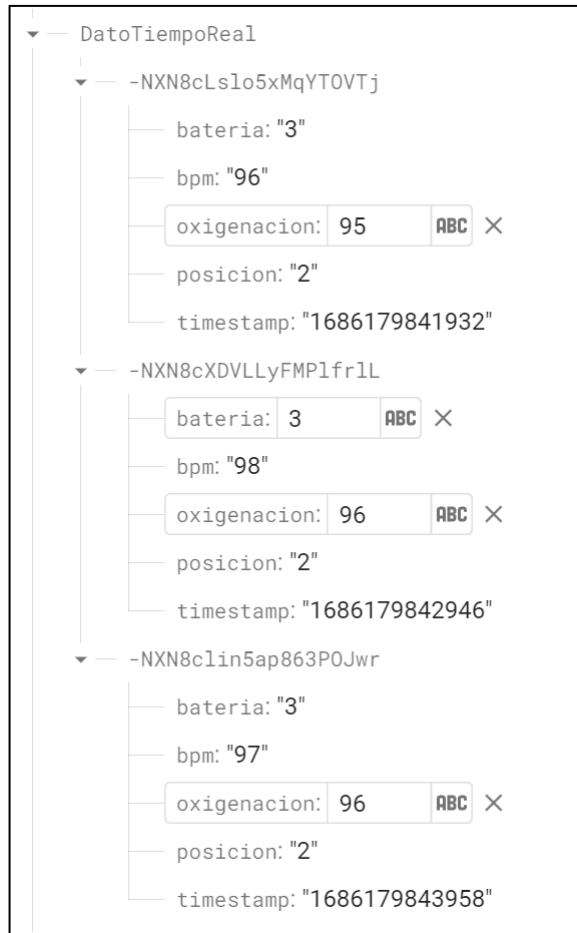


Figura 31 Base de datos, almacenada en Firebase
Elaborado por: El investigador

El nombre único del nodo se genera utilizando alguna forma de identificador único, como un ID generado automáticamente por Firebase o utilizando un identificador único personalizado. Este nombre garantiza que cada conjunto de datos enviado en modo POST tenga su propio nodo independiente. A continuación, se define una instancia inicial de la clase *DatosRealTime* en la variable *_datos*. Esta instancia tiene valores predeterminados para los distintos datos en tiempo real, como “bpm”, “posición”, “oxigenación”, “batería” y “timestamp”. Estos valores pueden ser actualizados a medida que se obtienen nuevos datos de la base de datos.

La clase *RealtimeProvider* también define un getter llamado *datos* que devuelve el objeto *_datos*. Esto permite a otros widgets obtener y mostrar los datos en tiempo real en la aplicación. La función *_initRealtimeDatabaseListener()* configura el *listener* para escuchar los nuevos datos agregados a la referencia '*DatoTiempoReal*' en la base

de datos. Se obtiene una referencia a esta ubicación mediante `_database.child('DatoTiempoReal')` y se suscribe al evento `onChildAdded.listen()`. De esta manera, los cambios en la base de datos se reflejan de manera automática y dinámica en la pantalla, proporcionando una visualización en tiempo real de los signos vitales y el estado de la batería.



Figura 32 Botón de terminar medición
Elaborado por: El investigador

Una vez completado la medición en el usuario o se desee finalizar, el botón mostrado en la figura 32 permite volver al estado donde se encuentran los diferentes perfiles. Sin embargo, su funcionalidad no se limita a eso, su identificador único también guarda todos los datos en su propia base de datos. Este proceso de guardado asegura que los datos recopilados durante la medición se almacenen de manera segura y se mantengan asociados al perfil correspondiente. De esta manera, se garantiza un registro completo y preciso de los datos de cada usuario, permitiendo un análisis posterior y un seguimiento adecuado de su estado de salud.

A continuación, se presenta el siguiente código:

```
import 'package:firebase_database/firebase_database.dart';

class MoveData {
  final String name;
  final now = DateTime.now();

  MoveData({required this.name});
}
```

```

Future<void> move() async {
  DateTime now = DateTime.now();
  String formattedDate =
    "${now.year.toString().padLeft(4, '0')}-${now.month.toString().padLeft(2,
'0')}-${now.day.toString().padLeft(2, '0')}-${now.hour.toString().padLeft(2, '0')}-
${now.minute.toString().padLeft(2, '0')}-${now.second.toString().padLeft(2, '0')}";
  final DatabaseReference sourceRef =
    FirebaseDatabase.instance.ref().child('DatoTiempoReal');
  final DatabaseReference destinationRef = FirebaseDatabase.instance
    .ref()
    .child('Basededatos')
    .child(name)
    .child(formattedDate);

  DataSnapshot dataSnapshot = await sourceRef.get();
  Map<dynamic, dynamic>? data = dataSnapshot.value as Map<dynamic, dynamic>?;

  data?.forEach((key, value) async {
    await destinationRef.update({key: value});
  });

  await sourceRef.remove();

  print('Datos movidos exitosamente de DatoTiempoReal a $name');
}
}

```

Como en toda la programación que incluye una interacción en Firebase, en primer lugar, se importa la biblioteca *firebase_database.dart*. La clase *MoveData* tiene dos atributos: *name*, que representa el nombre utilizado como destino para los datos movidos, y *now*, que almacena la fecha y hora actuales obtenidas mediante *DateTime.now()*. La función *move()* es una función asincrónica que se encarga de realizar el movimiento de los datos. En primer lugar, se genera una cadena de texto formateada con la fecha y hora actuales utilizando la variable *now*. A continuación, se itera sobre cada elemento del mapa *data* utilizando el método *forEach()*. Dentro del bucle se realiza una operación asincrónica en la referencia de destino para actualizar cada clave y valor mediante *destinationRef.update({key: value})*.



Figura 33 Valores almacenados en Base de Datos
Elaborado por: El investigador

En la figura 33 se presentan los valores obtenidos a partir de las mediciones del dispositivo. Al ejecutar el código mencionado anteriormente, estos valores serán transferidos al nodo denominado *"Basededatos"*. Los datos se recorren iterativamente utilizando el método *"forEach()"* en cada elemento del mapa *"data"*. Una vez que se han movido todos los datos, se elimina la referencia de origen utilizando *sourceRef.remove()*. Finalmente, se imprime un mensaje indicando que los datos se han movido exitosamente de la ubicación de origen a la ubicación de destino especificada por el atributo *name*. El resultado que tenemos es el siguiente:



Figura 34 Estructura de la Base de Datos de los usuarios
Elaborado por: El investigador

En la figura 34 se presenta la estructura jerárquica de almacenamiento en la base de datos. En primer lugar, encontramos el nodo principal llamado "Basededatos", que actúa como contenedor para todos los datos. Dentro de este modo, se crean subnodos con los nombres de los usuarios completada la medición. Posteriormente, se agrega un nodo adicional con la fecha que finaliza la medición. Dentro de este nodo de fecha, se almacenan las mediciones individuales. Cabe destacar que el uso del timestamp como parte de la estructura de los nodos permite una visualización ordenada de los datos en función de la fecha y hora en que se realizaron las mediciones. Esta estructura jerárquica garantiza una organización eficiente de los datos y facilita su acceso y consulta.

Pantalla de Registro de datos

En el apartado de Historial, se presenta una visualización de los datos en función del momento actual. Esta sección presenta la capacidad de visualizar los datos de acuerdo con si la medición está en curso en tiempo real o si ya ha finalizado. Dependiendo de la situación actual, se mostrarán los datos correspondientes, lo que permite a los usuarios tener acceso a la información relevante en el momento adecuado. Esta funcionalidad brinda una experiencia dinámica y personalizada.

Indice	Fecha	Frecuencia Cardíaca	Oxígeno en la sangre
1	2023-06-08 18:06:56	91 BPM Normal	97.00% Normal
2	2023-06-08 18:06:55	90 BPM Normal	98.00% Normal
3	2023-06-08 18:06:54	94 BPM Normal	97.00% Normal
4	2023-06-08 18:06:53	99 BPM Normal	94.00% Normal
5	2023-06-08 18:06:52	92 BPM Normal	96.00% Normal
6	2023-06-08 18:06:51	94 BPM Normal	96.00% Normal
7	2023-06-08 18:06:50	95 BPM Normal	97.00% Normal
8	2023-06-08 18:06:49	91 BPM Normal	95.00% Normal
9	2023-06-08 18:06:48	95 BPM Normal	96.00% Normal
10	2023-06-08 18:06:47	96 BPM Normal	97.00% Normal
11	2023-06-08 18:06:46	94 BPM Normal	98.00% Normal
12	2023-06-08 18:04:36	100 BPM Normal	98.00% Normal
13	2023-06-08 18:04:35	98 BPM Normal	95.00% Normal
14	2023-06-08 18:04:34	100 BPM Normal	95.00% Normal
15	2023-06-08 18:04:33	97 BPM Normal	95.00% Normal

Figura 35 Vista de datos generales
Elaborado por: El investigador

En la figura 35 se presenta la pantalla en la cual se reciben los datos en tiempo real desde la base de datos. Esta pantalla muestra una lista de los 15 registros más recientemente recopilados. Los datos que se exhiben incluyen la frecuencia cardíaca y la oxigenación de la sangre, junto con una etiqueta descriptiva que indica la situación actual. La tabla 11, ubicada en la sección 1.3.4, proporciona información detallada sobre las diferentes etiquetas y su significado.

Tabla 12 Estado del porcentaje de oxigenación en la sangre

Oxigenación	Estado
95% - 99%	Normal
90% - 94%	Hipoxia Leve
85% - 89%	Hipoxia Moderada
<85%	Hipoxia Severa

Elaborado por: El investigador

El código que nos permite visualizar los datos es el siguiente:

```
import 'dart:async';
import 'package:firebase_database/firebase_database.dart';
import 'package:flutter/material.dart';

class GetTablesdate with ChangeNotifier {
  List<Map<dynamic, dynamic>> _datosTiempoRealList = [];

  final _db = FirebaseDatabase.instance.ref();

  static const TIEMPO_REAL_PATH = 'DatoTiempoReal';

  late StreamSubscription _datosTiempoRealStream;

  List<Map<dynamic, dynamic>> get datosTiempoRealList => _datosTiempoRealList;

  GetTablesdate() {
    _listenToDatoTiempoReal();
  }

  void _listenToDatoTiempoReal() {
    _datosTiempoRealStream = _db
      .child(TIEMPO_REAL_PATH)
      .orderByChild('fecha')
      .limitToLast(20)
      .onValue
      .listen((event) {
        final data = (event.snapshot.value as Map<dynamic, dynamic>?) ?? {};
        _datosTiempoRealList = data.entries
          .map((entry) {
            final value = entry.value as Map<dynamic, dynamic>;
            return {
              'bateria': int.parse(value['bateria']),
              'bpm': int.parse(value['bpm']),
              'oxigenacion': int.parse(value['oxigenacion']),
              'posicion': int.parse(value['posicion']),
              'timestamp': int.parse(value['timestamp'])
            };
          });
      });
  }
}
```

```

        });
    })
    .toList()
    .reversed
    .toList();
    notifyListeners();
});
}

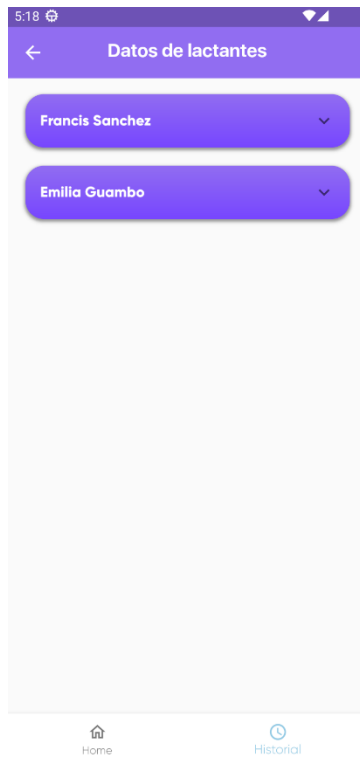
@override
void dispose() {
    _datosTiempoRealStream.cancel();

    super.dispose();
}
}

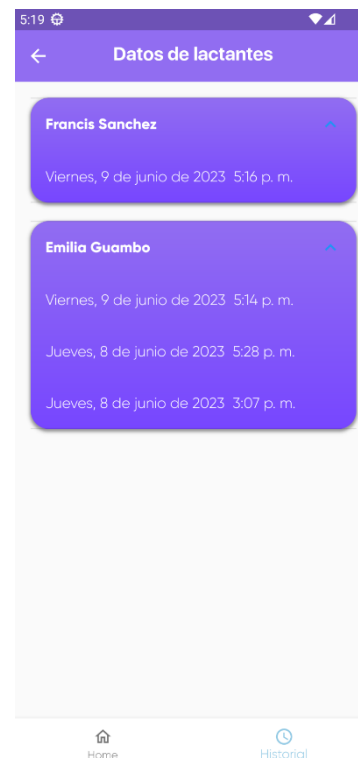
```

El código proporcionado corresponde a la clase *GetTablesdate*, la cual se encarga de obtener y gestionar datos en tiempo real desde una base de datos Firebase. Dentro de la clase, se define una lista llamada *_datosTiempoRealList*, la cual será utilizada para almacenar los datos obtenidos de la base de datos en tiempo real. Además, se crea una instancia de la base de datos Firebase mediante *FirebaseDatabase.instance.ref()* y se asigna a la variable *_db*. Se establece una constante llamada *TIEMPO_REAL_PATH*, que representa la ruta dentro de la base de datos donde se encuentran los datos en tiempo real.

Una vez finalizada la medición, la pantalla cambiará y nos mostrará la siguiente con la base de datos recopilada.



a) Usuario con datos



b) Fecha y hora de datos recopilados

Figura 36 Pantalla de los datos de lactantes
Elaborado por: El investigador

La figura 36 representa una interfaz que muestra los datos obtenidos. Al hacer clic en los elementos, se despliega una lista que incluye la fecha y hora en la que se tomaron las medidas correspondientes. Esta funcionalidad permite acceder a un registro ordenado y estructurado, facilitando la visualización de los datos individuales y su contexto temporal. De esta manera, los usuarios pueden examinar detalladamente cada conjunto de medidas registradas y obtener información precisa sobre los valores capturados en momentos específicos.

Se ha logrado ofrecer diversas opciones de acceso y visualización de la información. El resultado de estas opciones se muestra en la figura 37. A diferencia de la pantalla

que muestra los datos en tiempo real, esta funcionalidad permite acceder a todos los signos vitales registrados en una fecha específica.

Indice	Fecha	Frecuencia Cardíaca	Oxígeno en la sangre
1	2023-06-08 15:07:19	90 BPM Normal	96.00% Normal
2	2023-06-08 15:07:31	95 BPM Normal	98.00% Normal
3	2023-06-08 15:07:29	91 BPM Normal	96.00% Normal
4	2023-06-08 15:07:24	96 BPM Normal	97.00% Normal
5	2023-06-08 15:07:27	90 BPM Normal	94.00% Normal
6	2023-06-08 15:07:53	100 BPM Normal	98.00% Normal
7	2023-06-08 15:07:56	98 BPM Normal	97.00% Normal
8	2023-06-08 15:07:40	96 BPM Normal	98.00% Normal
9	2023-06-08 15:07:39	99 BPM Normal	96.00% Normal
10	2023-06-08 15:07:15	99 BPM Normal	98.00% Normal
11	2023-06-08 15:07:47	95 BPM Normal	99.00% Normal
12	2023-06-08 15:07:18	91 BPM Normal	97.00% Normal
13	2023-06-08 15:07:30	92 BPM Normal	99.00% Normal
14	2023-06-08 15:07:45	96 BPM Normal	98.00% Normal
15	2023-06-08 15:07:09	91 BPM Normal	98.00% Normal
	2023-06-08	91 BPM	95.00%

Figura 37 Registro de la Base de Datos de un lactante
Elaborado por: El investigador

3.2.6 Prueba de Funcionamiento

3.2.6.1 Dispositivo

A continuación, se tomará como ejemplo la Prueba N°06 de las pruebas realizadas, las cuales se encuentran en el Anexo D. Esta prueba en particular se selecciona debido a que contiene una mayor cantidad de datos en comparación con las demás. Esto nos permitirá realizar un análisis más exhaustivo. Los datos correspondientes al lactante son los siguientes:

Tabla 13 Datos del lactante

Nombre	Edad	Alimentado	Vestimenta	Lugar de descanso	Observaciones
LACTANTE NN6	8 meses	No	Body	Cuna	Ninguna

Elaborado por: El investigador

Una vez que estos datos fueron tomados se procede a llenarlos en la aplicación para poder empezar con las mediciones.

Una vez que tenemos los datos del lactante empezamos colocando el Smart textil elaborado anteriormente en el lactante cuando se encuentra dormido, como se muestra en la figura 38.

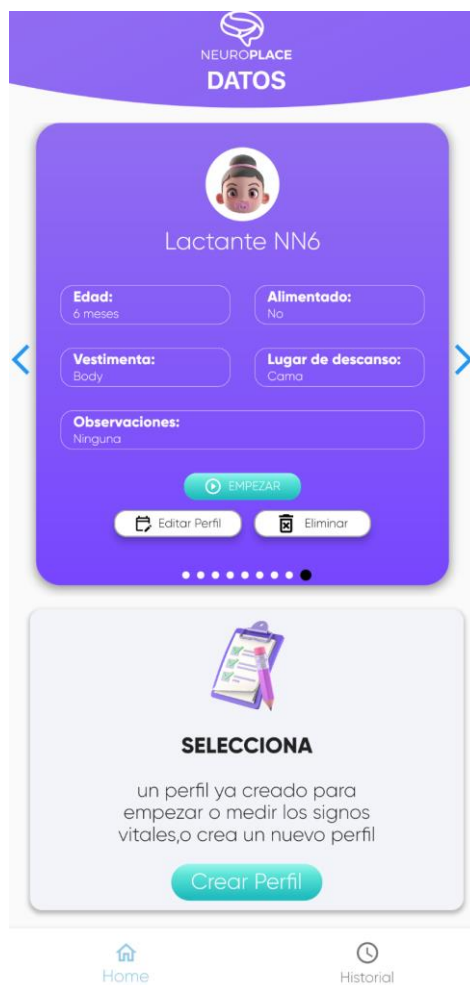


Figura 38 Vista de datos

Elaborado por: El investigador

Se sugiere utilizar el dispositivo cuando el lactante esté dormido, ya que de esta manera se logrará un funcionamiento más preciso sin que los movimientos del bebé lo afecten. En la figura 39 se puede observar cómo se encuentra implementado el dispositivo al momento de su descanso u hora de sueño.



Figura 39 Implementación de Smart Textil
Elaborado por: El investigador

Una vez encendido el dispositivo se conectará automáticamente a la red Wi-Fi configurada. Una vez establecida la conexión, se mostrará el tal como se ilustra en la figura siguiente.



Figura 40 Dispositivo conectado a la Red
Elaborado por: El investigador

Una vez conectado, el sistema mostrara el procedimiento para él envió de datos.



a) Muestreo de datos



b) Presentación de datos



c) Envió de datos

Figura 41 Proceso de muestreo de datos del Smart Textil
Elaborado por: El investigador

Como se puede apreciar en la figura 41, el proceso se repite con el fin de determinar los valores de frecuencia cardíaca y saturación de oxígeno en la sangre. La pantalla OLED proporciona información en tiempo real sobre el progreso del dispositivo, manteniéndonos informados sobre el estado actual del proceso. La figura 42 muestra los valores del oxímetro GL3200, así como los valores que se registran en la base de datos del Smart textil.

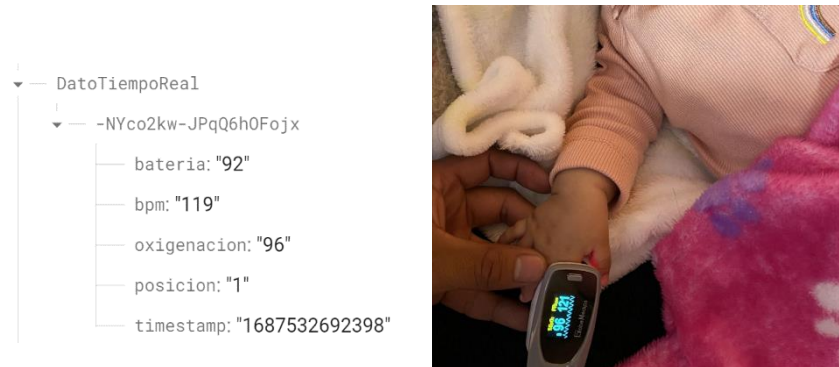


Figura 42 Comparativa de datos con un Oxímetro comercial
Elaborado por: El investigador

Se accede a la aplicación para verificar que los datos se vean reflejados en el instante que llegan a la aplicación como se muestra en la figura 43.

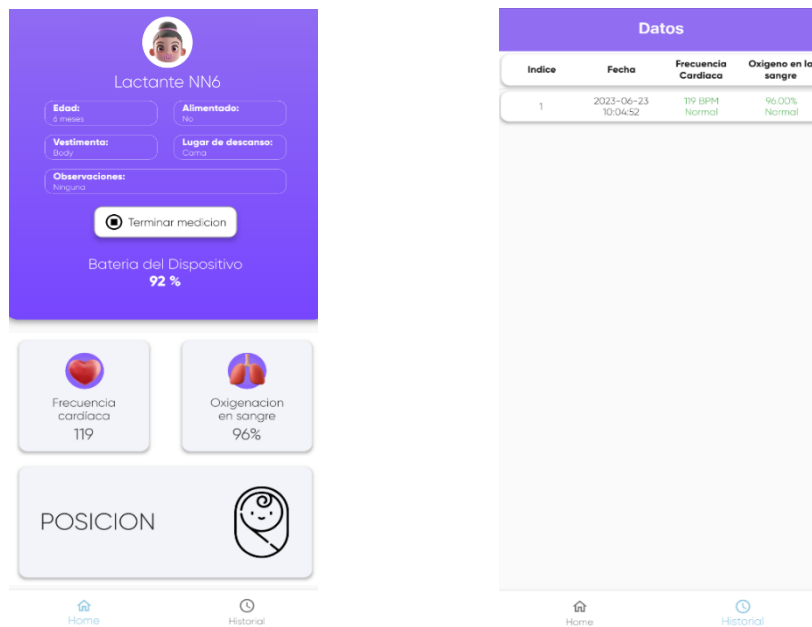


Figura 43 Datos en aplicación en tiempo real

Elaborado por: El investigador

3.2.6.2 Resultados

Mediciones de sensor

La medición se llevó con acabo con el Lactante NN6 el día 23 de junio de 2023, durante un lapso desde las 12:12 pm hasta las 12:55 pm lo que equivale a un total de 43 minutos. Las condiciones de la medición fueron una cuna con superficie firme y plana, colocándolo boca arriba para dormir, con una ventilación adecuada y luz tenue. Durante este período, se recopilaron 126 datos en total. Esto significa que se obtuvo un promedio de tiempo de 20,47 segundos por cada dato recopilado, lo cual permitió realizar una recolección precisa de los datos.

En las figuras 44 y 45 se muestra un gráfico comparativo entre el dispositivo creado y un oxímetro GL3200. Esta comparativa visual permite evaluar y analizar la concordancia entre ambos dispositivos en términos de las mediciones realizadas.

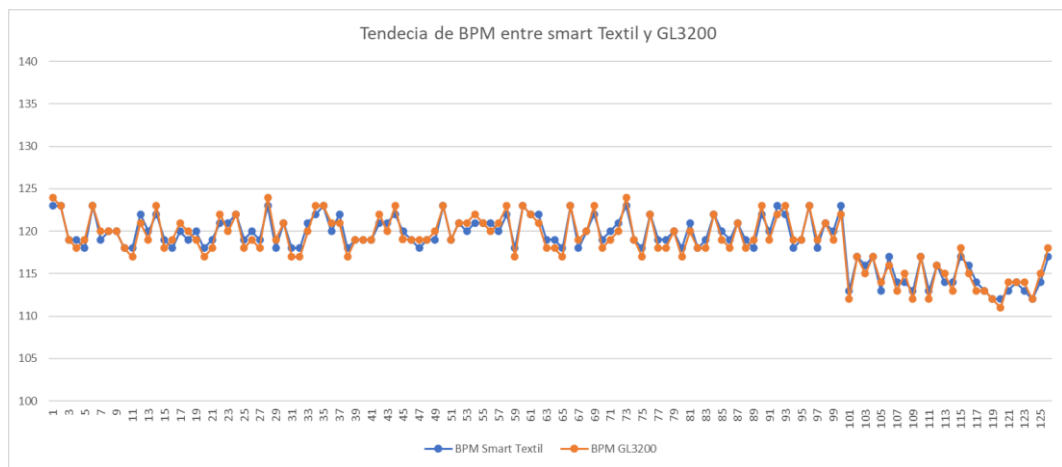


Figura 44 Comparativa de la frecuencia cardíaca entre el Smart Textil y el oxímetro GL3200

Elaborado por: El investigador

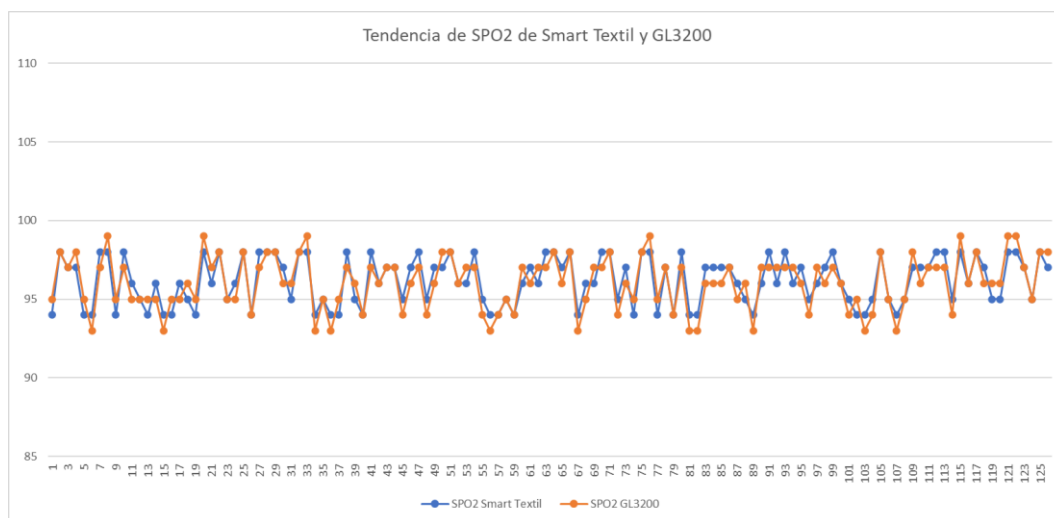


Figura 45 Comparativa de oxígeno en la sangre entre el Smart Textil y el oxímetro GL320

Elaborado por: El investigador

Se llevó a cabo un análisis de los resultados de las mediciones de bpm (pulsaciones por minuto) y oxigenación en la sangre utilizando el sensor GY-MAX30102. Con el fin de evaluar la confiabilidad de este sensor, se compararon nuevamente estas mediciones con las de un dispositivo comercial que contaba con una mayor cantidad de datos disponibles. Para determinar la fiabilidad del sensor GY-MAX30102, se utilizó la ecuación (2) que permite calcular el error absoluto, el cual se obtiene al tomar la diferencia entre el valor exacto (medida del dispositivo comercial GL3200) y el valor real (medida del sensor GY-MAX30102). Posteriormente, se calculó el error relativo utilizando el error absoluto, y este valor nos proporciona el porcentaje de fiabilidad mediante el uso de la ecuación (4).

$$Error\ absoluto = |Valor\ Exacto - Valor\ real| \quad (2)$$

$$Error\ relativo = \frac{Error\ absoluto}{Error\ exacto} * 100\% \quad (3)$$

$$Fiabilidad = 100 - Error\ Relativo \quad (4)$$

El porcentaje de error de la frecuencia cardíaca fue de 1,98%, lo que indica una fiabilidad del 98,02% en las mediciones. Por otro lado, el valor de error relativo para la medición del oxígeno en la sangre fue de 1,73%, lo que proporciona una fiabilidad

del 98,27%. Estos resultados muestran que tanto la frecuencia cardíaca como la medición de oxígeno en la sangre realizadas por el dispositivo demostraron nivel de precisión y confiabilidad aceptable. Los porcentajes de error relativamente bajos indican que las mediciones obtenidas son muy cercanas a los valores reales.

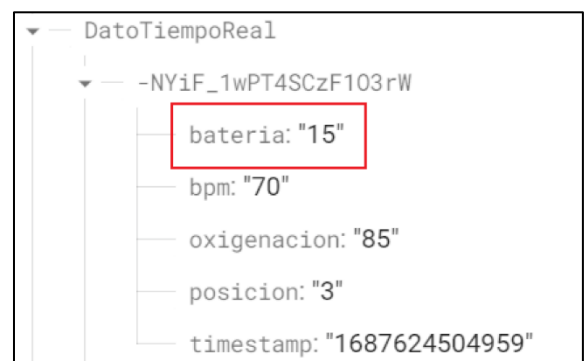
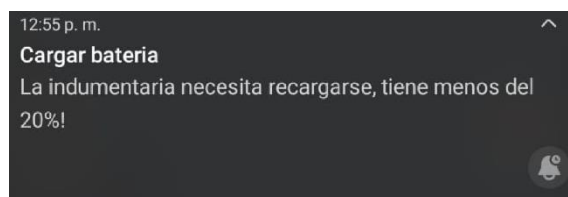
Validación Médica

Con el fin de asegurar que los valores medidos por el sensor de ritmo cardíaco se encuentren dentro de los rangos normales, se consultó a un médico general para que verificara los datos obtenidos. Tras su evaluación, se determinó que el sensor es apto para la medición del ritmo cardíaco. La certificación emitida por el médico se encuentra en la sección de anexos, específicamente en el ANEXO L.

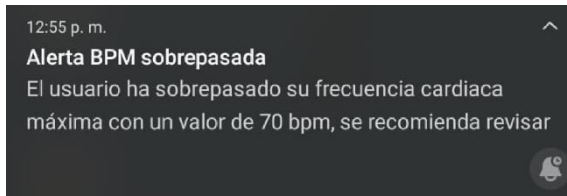
Sistema de alerta y Notificaciones

En este caso, se han establecido cuatro alertas en la aplicación. La primera alerta se activa cuando los valores de la frecuencia cardíaca están fuera de los rangos normales, ya sea por ser demasiado altos o bajos. De manera similar, la segunda alerta se dispara cuando los valores de oxígeno en la sangre están fuera de los rangos normales. La tercera alerta se activa cuando la energía del dispositivo alcanza un nivel inferior al 20%, y la última alarma se activa cuando la posición del lactante ha variado.

Estas alertas se envían a través de Push Notificación, lo que significa que se pueden recibir incluso si la aplicación está en segundo plano o no se está utilizando activamente. Para la elaboración de ejemplos, los datos se envían directamente desde el microcontrolador, evitando la necesidad de esperar a que ocurra alguna situación para activar las notificaciones.

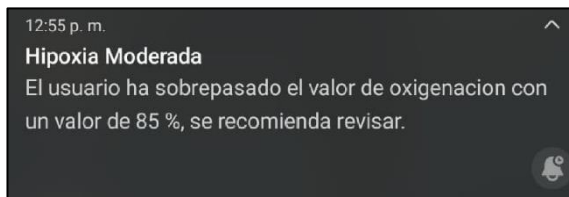


a) Notificación por nivel bajo de batería



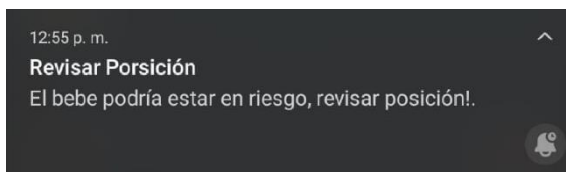
```
DatoTiempoReal  
└── -NYiF_1wPT4SCzF103rW  
    ├── bateria: "15"  
    ├── bpm: "70"  
    ├── oxigenacion: "85"  
    ├── posicion: "3"  
    └── timestamp: "1687624504959"
```

b) Notificación por niveles anormales de frecuencia cardiaca



```
DatoTiempoReal  
└── -NYiF_1wPT4SCzF103rW  
    ├── bateria: "15"  
    ├── bpm: "70"  
    ├── oxigenacion: "85"  
    ├── posicion: "3"  
    └── timestamp: "1687624504959"
```

c) Notificación por niveles anormales de oxígeno en la sangre



```
DatoTiempoReal  
└── -NYiF_1wPT4SCzF103rW  
    ├── bateria: "15"  
    ├── bpm: "70"  
    ├── oxigenacion: "85"  
    ├── posicion: "3"  
    └── timestamp: "1687624504959"
```

d) Notificación por posición de descanso del lactante

Figura 46 Notificaciones de la aplicación
Elaborado por: El investigador

La figura 46 ilustra cómo las notificaciones son entregadas de manera precisa al dispositivo. En el registro de la base de datos, la hora de recepción de los datos se encuentra en un formato especial llamado EPOCH, ampliamente utilizado en sistemas informáticos. En este caso, el número correspondiente es "1687624504959", el cual, al ser convertido, se traduce en la hora "12:35:05 pm". Esto demuestra que la respuesta es casi instantánea en el funcionamiento de nuestro sistema de alertas, el cual es eficiente y brinda una respuesta rápida.

Autonomía de la batería.

Para determinar la duración de la batería, se detalla el consumo de cada uno de los elementos que se utilizan en la indumentaria, la tabla 14, detalla el consumo de la corriente y el voltaje de operación:

Tabla 14 Gasto energético de los componentes del Smart Textil

DISPOSITIVO	Consumo de corriente (mA)	Voltaje de Operación (V)
ESP32 Feather	100 mA	3,3 V
GY-MAX30102	30 mA	3,3 V
MPU-6050	15 mA	3,3 V

Elaborado por: El investigador

En la tabla 14 se observa que el total de consumo de corriente fue de 145 mA, obtenido mediante la suma del consumo de todos los dispositivos. La batería de litio utilizada presentaba una capacidad de 3.7 V y 1800 mAh para alimentar dichos dispositivos. A continuación, se utilizó la fórmula para calcular la duración en horas de una batería en función de su capacidad en mAh (miliamperios-hora) y la corriente total en mA (miliamperios), el tiempo de funcionamiento es:

Duración (horas) = Capacidad de la batería (mAh) / Corriente total (mA)

Duración = 1800 mAh / 145 mA = 12.41 horas

Por lo tanto, con una batería de litio de 3.7 V y 1800 mAh, y con los dispositivos ESP32 Feather, GY-MAX30102 y MPU-6050 funcionando simultáneamente, se esperaba una duración de la batería de aproximadamente 12.41 horas. Esta capacidad permitió al Centro de Estimulación Temprana trabajar de manera ininterrumpida durante un día laboral sin la necesidad de recargarla durante este periodo.

3.2.7 Presupuesto

Se realizó el cálculo del presupuesto total del proyecto teniendo en cuenta dos aspectos fundamentales: el presupuesto de construcción y el presupuesto de mano de obra. El primero comprende los costos asociados a los componentes y software utilizados en el proyecto, mientras que el segundo se refiere al tiempo dedicado al diseño y construcción de mismo. En cuanto al presupuesto de mano de obra, se consideró el salario promedio establecido por el Ministerio de Trabajo para un Ingeniero en Telecomunicaciones, el cual asciende a \$485.7: [51]

$$Sueldo_d = \frac{Salario_m}{Dias Laborales}$$

$$Sueldo_d = \frac{485.70}{21} = 23.12$$

$$Sueldo_h = \frac{23.12}{8} = 2.89$$

Se ha calculado que el tiempo necesario para llevar a cabo las labores de diseño, desarrollo, pruebas y corrección de errores en este proyecto asciende a un total de 200 horas, el costo de la mano de obra es:

$$Costo_{Mano de obra} = Hora \times Sueldo_h$$

$$Costo_{Mano de obra} = 200 \times 2.89 = 578$$

La siguiente tabla muestra el costo de los materiales de construcción que fueron utilizados en la elaboración del dispositivo:

Tabla 15 Presupuesto de construcción

MATERIAL	CANTIDAD	VALOR/U	VALOR/T
ESP32 FEATHER	1	\$ 25.00	25.00
GY-MAX30102	1	\$ 15.00	\$15.00
Bateria Lipo	1	\$ 18.00	\$ 18.00
MPU-6050	1	\$ 8.00	\$ 8.00
Display Oled	1	\$ 10.00	\$ 10.00
PCB	1	\$ 5.00	\$ 5.00
Impresion 3D	1	\$ 20.00	\$ 20.00
Cables Arduino	1	\$ 3.50	\$ 3.50
Estaño	1	\$ 1.50	\$ 1.50
Microfibra	1	\$ 5.00	\$ 5.00
Velcro	2	\$ 1.00	\$ 2.00
Subtotal			113
IVA (12%)			13.56
TOTAL			126.56

Elaborado por: El investigador

Una vez definido el presupuesto de la mano de obra y el presupuesto de los materiales de construcción, podemos calcular el presupuesto total que se hace constar en la tabla 16:

Tabla 16 Presupuesto total de proyecto

DESCRIPCION	CANTIDAD	VALOR/U	VALOR/T
Presupuesto mano de obra	1	\$ 578.00	\$ 578.00
Presupuesto de construcción	1	\$ 126.56	\$ 126.56
TOTAL			\$ 704.56

Elaborado por: El investigador

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- La muerte súbita en lactantes es un fenómeno complejo influenciado por una combinación de factores internos y externos. Durante el embarazo y condiciones maternas como el consumo de tabaco, la lactancia artificial y el uso de drogas o alcohol se asocian con el mayor riesgo de muerte súbita en el lactante. Además, factores propios del bebé, como el nacimiento prematuro y el bajo peso al nacer, también aumentan esta posibilidad. Entre los factores externos, la mala posición de descanso puede obstruir las vías respiratorias y afectar el suministro de oxígeno, incrementando así el riesgo de muerte súbita. Estos factores, tanto internos como externos, tienen un impacto directo en el funcionamiento inadecuado de la frecuencia cardíaca y los niveles de oxígeno en la sangre.
- Se logró determinar las características fundamentales de hardware y software necesarias para la creación de un textil. En cuanto al hardware, se determinó que es crucial contar con microcontroladores con capacidad de comunicación WiFi, lo que permite una conexión estable y confiable a larga distancia. Asimismo, se identificó la importancia de utilizar un sensor no invasivo basados en la técnica como la pulsioximetría. En cuanto al software, se determinó la utilidad de las notificaciones push para una rápida detección y alerta ante cualquier irregularidad en los factores monitoreados.
- El diseño del Smart Textil logró ser una solución efectiva para el monitoreo constante del ritmo cardíaco y el nivel de oxígeno en lactantes. Mediante la colocación estratégica del sensor GY-MAX30102 en la pantorrilla y su sujeción con una correa de velcro se ha logrado obtener mediciones adecuadas cuando el lactante se encuentra descansando. Los resultados obtenidos revelan una precisión del 98.02% y 98.27% para el ritmo cardíaco y el nivel de oxígeno respectivamente, lo cual se encuentra dentro del rango de confiabilidad aceptable para una adecuada

detección de valores anormales. Además, la incorporación del MPU ha permitido detectar la posición del lactante, brindando una funcionalidad adicional al sistema de monitoreo.

- Se creó una interfaz móvil utilizando Flutter y la plataforma Firebase, que permite la visualización en tiempo real de los signos vitales y la posición del lactante. La interfaz tiene un diseño atractivo y su intuitiva usabilidad, mejorando la experiencia del usuario. Mediante la implementación de notificaciones push se estableció un sistema de alertas eficiente que notifica de forma inmediata sobre valores anormales sobre alguno de los parámetros monitoreados y de funcionamiento del dispositivo y que se encuentran fuera de los límites establecidos. Esta solución tecnológica, brinda una respuesta rápida y confiable para el monitoreo de los factores implicados en el síndrome de muerte súbita en lactantes, garantizando así un sistema de alerta en el Centro de Estimulación Temprana “Neuro Place”.

4.2. Recomendaciones

- Al utilizar el sensor GY-MAX30102, es importante seguir algunas recomendaciones para obtener mediciones precisas y confiables. Se recomienda colocar el sensor en áreas del cuerpo del lactante donde se obtengan lecturas consistentes del ritmo cardíaco y la saturación de oxígeno. Es importante evitar la exposición a fuentes de luz intensa y ruidos electromagnéticos, así como mantener un entorno tranquilo y libre de interferencias. Realizar un monitoreo constante de las lecturas obtenidas y consultar con un profesional de la salud en caso de presentarse dudas o lecturas anormales.
- Se recomienda utilizar el dispositivo como una herramienta complementaria y de monitoreo, pero no como un reemplazo del diagnóstico médico. Aunque este sensor proporciona datos sobre el ritmo cardíaco y la saturación de oxígeno en la sangre, es esencial recordar que su precisión puede verse afectada por diversos factores y limitaciones técnicas. Por lo tanto, es importante interpretar los resultados obtenidos con cautela y consultar a un profesional de la salud para un diagnóstico médico preciso y cualquier preocupación relacionada con la salud del lactante.
- Al utilizar un dispositivo que utiliza Wi-Fi, asegúrese de que el dispositivo esté dentro del rango de cobertura de su red Wi-Fi, evitando distancias excesivas o barreras físicas que puedan debilitar la señal. Si experimenta una conexión débil, considere la posibilidad de utilizar repetidores o extensores de señal para ampliar el alcance de su red.

BIBLIOGRAFÍA

- [1] K. O. T. a. R. Y. M. F. R. Hauck, Artist, *Racial and ethnic disparities in infant mortality*. [Art]. Seminars in Perinatology, 2018.
- [2] E. Garcia Garcia, M. Martinez Silva y M. Moreno Miravalles, Artists, *Síndrome de la muerte súbita del lactante*. [Art]. Sociedad Cubana de Cardiología.
- [3] F. C. Piquer, Artist, *Síndrome de la Muerte*. [Art]. Madrid: GEMPSI, 2003.
- [4] INEC, «Instituto nacional de estadística y censos,» 12 04 2023. [En línea]. Available: <https://www.ecuadorencifras.gob.ec/nacidos-vivos-y-defunciones-fetales/>.
- [5] G. P. M. G. y. C. G. D. Bautista Rico, Artist, *Caracterización de la situación actual de las tecnologías inteligentes en Latinoamérica..* [Art]. Revista Iberica de Sistemas y Tecnologías de Información , 2019.
- [6] INEC, «INEC,» [En línea]. Available: <https://app.powerbi.com/view?r=eyJrIjoiYmM4NWZjNTktNGRlZi00NDkxLWEzOWUtYmEwNDg3NTYwMjIiIiwidCI6ImYxNThhMmU4LWNhZWMTNDQwNi1iMGFiLWY1ZTI1OWJkYTEwMiJ9>.
- [7] E. M. A. GALLARDO, «CONSTRUCCIÓN DE UN MEDIDOR.,» *Universidad de las Fuerzas Armadas ESPE*.
- [8] M. A. L. CHANATAXI, Artist, *DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO..* [Art]. ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, 2016.
- [9] L. .. A. R. Narváez, Artist, *Sistema de telemedicina para monitoreo continuo de constantes vitales en lactantes menores para evitar el síndrome de muerte súbita..* [Art]. Universidad Técnica de Ambato, 2018.
- [10] N. U. H. y. N. I. I, Artist, *Wearable technology for baby monitoring*. [Art]. Med Crave , 2019.

- [11] M. A. y. O. Räsänen, Artist, *Automatic Posture and Movement Tracking of*. [Art]. Scientific Reports, 2020.
- [12] M. S. M. M. García GFE, Artist, *Síndrome de la muerte súbita del lactante..* [Art]. CorSalud, 2014.
- [13] S. J. L. o. Medicine, «National Library of Medicine,» Sudden infant death syndrome, 08 09 2021. [En línea]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8424793/>. [Último acceso: 17 04 2023].
- [14] R. D. G. y. P. B. M. L. Munkel Ramírez, «Medicina Legal de Costa Rica,» 01 01 2018. [En línea]. Available: https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S140900152018000100065. [Último acceso: 17 04 2023].
- [15] C. S. Ruiz, «redalyc,» 01 12 2007. [En línea]. Available: <https://www.redalyc.org/articulo.oa?id=366638698010>. [Último acceso: 17 04 20023].
- [16] P. P. a. W. B. Rosei, Artist, *Assessment of heart rate as a risk factor*. [Art]. European Heart Journal Supplements, May 2018.
- [17] J. C. D.-C. a. J. L. R. P. J. A. S. Moraes, Artist, *Heart rate variability: A review on methodological aspects, physiological and pathological influence*. [Art]. International Journal of Cardiology, Sep. 2021.
- [18] J. E. S. a. J. S. Jugovac, Artist, *Development of the Cardiovascular System in the Infant*. [Art]. Advances in Neonatal Care, Apr. 2020.
- [19] J. J. Volpe, de *Neurology of the Newborn*, Philadelphia, Saunders, 2008, pp. 455-457.
- [20] M. P. Cicogna, Artist, *Paediatric reference values for heart rate variability during sleep*. [Art]. European Journal of Applied Physiology, 2019.

- [21] J. H. L. a. S. H. C. J. S. Han, Artist, *Pulse Oximetry: Basics and Beyond*. [Art]. Journal of the Korean Medical Association, Sep. 2020.
- [22] E. B. R. a. R. C. S. C. M. Teixeira, Artist, *Oxygen saturation targets in the intensive care unit: A systematic review and meta-analysis*. [Art]. Heart & Lung, 2021.
- [23] M. Gast, 802.11 Wireless Networks: The Definitive Guide, Sebastopol: O'Reilly Media, 2005.
- [24] C. E. Hande Alemdar, Artist, *Wireless sensor networks for healthcare: A survey*. [Art]. Science Direct, 2010.
- [25] A. HUZZAH32, «adafruit,» [En línea]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-huzzah32-esp32-feather.pdf>. [Último acceso: 17 04 2023].
- [26] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>. [Último acceso: 17 04 2023].
- [27] Wemos, «Wemos,» [En línea]. Available: https://www.wemos.cc/en/latest/d1/d1_mini.html. [Último acceso: 17 04 2023].
- [28] Z. Y. a. G. C. B. Sun, «Photoplethysmography-Based Heart Rate Monitoring in Wearable Electronics,» de *IEEE International Conference on Consumer Electronics (ICCE)*, 2017 .
- [29] I. P. A. B. M. Kosacka, *Sclerostin in Obstructive Sleep Apnea*, Springer, 2015.
- [30] A. Devices, «Analog Devices,» [En línea]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30102.pdf>. [Último acceso: 18 04 2023].
- [31] R. Semiconductor, «Rohm Semiconductor,» [En línea]. Available: https://fscdn.rohm.com/en/products/databook/datasheet/ic/sensor/pulse_wave/bh1792glc-e.pdf. [Último acceso: 18 04 2023].

- [32] J. A. Yoder, Artist, *Three-axis inertial measurement using a planar vibratory MEMS gyroscope*. [Art]. IEEE Trans. Instrum. Meas., 2010.
- [33] InvenSense, «InvenSense,» [En línea]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf>. [Último acceso: 18 04 2023].
- [34] life.augmented, «life.augmented,» [En línea]. Available: <https://www.st.com/resource/en/datasheet/cd00274221.pdf>. [Último acceso: 18 04 2023].
- [35] M. W. a. R. J. Brodd, Artist, *What Are Batteries, Fuel Cells, and Supercapacitors?*. [Art]. ACS Publication, 2004.
- [36] A. V. Egorov, Artist, *Lithium polymer batteries*. [Art]. Russian Journal of Electrochemistry, 2012.
- [37] F. W. J. T. L. G. Y. Naoki Nitta, Artist, *Li-ion battery materials: present and future*. [Art]. Materials Today, 2015.
- [38] M. Fowler, Artist, *Inversion of Control Containers and the Dependency Injection Pattern*. [Art]. 2004.
- [39] S. S. a. N. Dahanayake, *Service-Oriented Software System Engineering: Challenges and Practices*, Hershey: Idea Group Publishing, 2005.
- [40] Medium. [En línea]. Available: <https://inapp-inc.medium.com/react-native-vs-flutter-vs-ionic-vs-xamarin-vs-nativescript-a-detailed-comparison-bfa99d72c4bb>. [Último acceso: 18 04 2023].
- [41] M. U. A. K. M. G. N. a. A. Y. Z. M. Nadeem, Artist, *A survey of backend as a service for mobile applications*. [Art]. IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), 2016.
- [42] Oracle, «OCI,» [En línea]. Available: <https://www.oracle.com/mx/database/what-is-database/>. [Último acceso: 23 04 2023].

- [43] M. & H. M. Ghanbari, Artist, *A comparative study of SQL and NoSQL databases*. [Art]. International Journal of Advanced Computer Science and Applications, 2021.
- [44] S. & D. S. Priyadarshini, Artist, *SQL vs NoSQL: A comparative study*. [Art]. International Journal of Advanced Research in Computer Science, 2021.
- [45] M. U. a. J. F. J. Orduña, Artist, *Diseño y desarrollo de servicios web RESTful para aplicaciones móviles*. [Art]. Revista Iberoamericana de Tecnologías del Aprendizaje, 2016.
- [46] DBMS, «DBMS,» [En línea]. Available: <https://db-engines.com/en/system/Cassandra%3BFirestore+Realtime+Database%3BRedis>. [Último acceso: 23 04 2023].
- [47] J. G. Carmenate, «Programar Facil,» [En línea]. Available: <https://programarfacil.com/blog/arduino-blog/comunicacion-i2c-con-arduino/>. [Último acceso: 21 06 2023].
- [48] B. Martínez, «Guia Infantil,» [En línea]. Available: <https://www.guiainfantil.com/blog/salud/sueno/que-dice-de-los-ninos-su-forma-de-dormir/>. [Último acceso: 01 06 2023].
- [49] H. Matheus, «USINAINFO,» [En línea]. Available: <https://www.usinainfo.com.br/acelerometro-arduino/acelerometro-e-giroscopio-mpu-6050-3-eixos-6-dof-gy521-3895.html>. [Último acceso: 01 06 2023].
- [50] E. Battery, «eemb Battery,» [En línea]. Available: <https://www.eemb.com/>. [Último acceso: 03 06 2023].
- [51] JEZL, «JEZL,» [En línea]. Available: <https://www.jezl-audidores.com/index.php/tributario/111-tabla-sueldos-minimos-2022>. [Último acceso: 24 06 2023].
- [52] J. Woodard, «What is a Peristaltic Pump and How Does It Work?,» 21 marzo 2019. [En línea]. Available:

<https://www.freshwatersystems.com/blogs/blog/what-is-a-peristaltic-pump-and-how-does-it-work>. [Último acceso: 10 noviembre 2022].

- [53] F. A. a. K. N. N. Yazdi, Artist, *Micromachined inertial sensors*. [Art]. Proc. IEEE, 1998.
- [54] R. S. S. a. M. W. D. S. T. V. D. Woods, Artist, *Inertial sensor-based methods in walking speed estimation: a systematic review*. [Art]. 2012.
- [55] P. Navarro, «Value Life,» 10 3 2022. [En línea]. Available: <https://campusvygon.com/venas-disponibles-picc-neonatos/>. [Último acceso: 29 05 2023].

ANEXOS

Anexo A: Código Fuente del microcontrolador ESP32 Feather

```
#include <Adafruit MPU6050.h>
#include <Adafruit_Sensor.h>
#include "MAX30105.h"
#include <U8g2lib.h>
#include "heartRate.h"
#include <Wire.h>
#include <HTTPClient.h>
#include "WiFi.h"
#include "spo2_algorithm.h"
#include <WiFiClientSecure.h>

const char* ssid = "NEURO-PLACE";
const char* password = "0502928740Carlos";

#define uS_TO_S_FACTOR 1000000ULL
#define TIME_TO_SLEEP 10

#define ADC_BAT 35

const char* rootCACertificate =
"-----BEGIN CERTIFICATE-----\n"
"MIIFVzCCAz+gAwIBAgINAgPlk28xsBNJiGuiFzANBgkqhkiG9w0BAQwFADBHMQsw\n"
"CQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFN1cnZpY2VzIEExMQZEU\n"
"MBIGA1UEAxMLR1RTIFJvb3Q0UjEwHhcNMjYwNjIyMDAwMDAwWhcNMzYwNjIyMDAw\n"
"MDAwWjBHMQswCQYDVQQGEwJVUzEiMCAGA1UEChMZR29vZ2x1IFRydXN0IFN1cnZp\n"
"Y2VzIEExMQZEUeUMBIGA1UEAxMLR1RTIFJvb3Q0UjEwGgIiMA0GCSqGSIb3DQEBAQUA\n"
"A4ICDwAwggIKAoICAQC2EQKLHu0hd5s73L+UPreVp0A8of2C+X0yBoJx9vaMf/vo\n"
"27xqLpeXo4xL+Sv2sfn0hB2x+cWX3u+58qPpvBKJXqeqUqv4IyfLpLgCY9vXmX7w\n"
"C17raKb0xlpHDu0QM+N0sR0jyBhsS+z8CZDfnWQpJSMHobTSPS5g4M/SCYe7zUjw\n"
"TcLCEoiKu7rPWRnWr4+wB7CeMfGCwCdfLqZtbBk0tdh+JhpFAz2weaSUKK0Pfyb1\n"
"qAj+lug8aJRT7oM6iCsVlgy4HqMLnXWnOunVmSP1k9orj2XwoSPwLxAwAtcvfah\n"
"szVsrbhQf4TgTM2S0yDpM7xSma8ytSmzJSq0SP1y4cPk9+aCEI3oncKKiPo4Zor8\n"
"Y/kb+Xj9e1x3+naH+uzfsQ551Ve0vSbv1gHR6xYKu44LtcXfilWr06zqkUspzBmk\n"
"MiV0kVf1RNACzqr0SbTqn3yDsEB7500rp2yjj32JgfpMpf/VjsPOS+C12L00Rc92\n"
"w01AK/1TD7Cn1TsNsYqiA94xrcx36m97PtbfkSIS5r762DL8EGMUUXLeXdYwk70p\n"
"aDPvOmsB4om3xPXV2V4J95eSRQAogB/mqghtqmx1bCluQ0WEdrHbEg8QOB+DvRn\n"
"VjzR1wW5y0vtOUucxD/SVRNuJLDwcfR0wbrM7Rv1/ofB2ACYPTTrIrnqYNxgFlQID\n"
"AQABO0IwQDAOBgNVHQ8BAF8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4E\n"
"FGQU5K8rJnEak0gnhS9SZizv8IkTcT4wDQYJKoZIhvcNAQEMBQADggIBAJ+qQibb\n"
"C5u+/x6Wki4+omVKapi6Ist9wTrYggoGxva13sB0h2Z5ofmmWJyq+bXmY0fg6LEe\n"
"QkEzCzc9zo1wFcq1JKjPa7XSQCgYzyI0zzvFIoTgxQ6kFF2I5DUkzps+G1Qebtuy\n"
"h6f88/qBVRriC1mpIguXpOLw7ttXNLwzldMXG+gnoot7TiYae1pkttGsN/H9oPM4\n"
"7HLwEXWdyzRSjeZ2axfg34arJ45JK3VmgRAhpuo+9K41/3wV3s6MJT/KYnAK9y8J\n"
"ZgfIPxz88ntFMM9iiMG1D53Dn0reW1HxYciNuaCp+0KueIHOI17eko8cdLiA6Ef\n"
"MgfdG+RCzgwARWGAatQsgWS14vf1Vy2PFPEz0tv/bal8xa5meLMFrUKTX5hgUvYU/\n"
```

```

"Z6tGn6D/Qqc6f1zLXbBwHsS09dR2CQzreExZBfMzQsNhFRAbd030IozUhfJfFbdT\n"
"6u9AWpQKXCbFTkBdYiJ23//OYb2MI3jSNwLgjt7RETeJ9r/tSQdirpLsQBqvFAnZ\n"
"0E6yove+7u7Y/9waLd64NnHi/Hm3lCXrSHNboTXns5lndcEZ0itHTtNCjv0xyBZm\n"
"2tIMPNUzjsmhDYAPexZ3FL//2wmUsp08IFgV6dtxQ/PeEMMA3KgqlbbC1j+Qa3bb\n"
"bP6MvPJwNQzcmRk13NfIRmPVNnGuV/u3gm3c\n"
"-----END CERTIFICATE-----\n";

WiFiClientSecure client;

MAX30105 particleSensor;
U8G2_SSD1306_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0);

const byte RATE_SIZE = 8;
byte rates[RATE_SIZE];
byte rateSpot = 0;
long lastBeat = 0;

float beatsPerMinute;
int beatAvg;
long irValue;
int samples = 0;
int samplespo2 = 0;
byte posicion = 0;

////
uint32_t irBuffer[100]; //infrared LED sensor data
uint32_t redBuffer[100]; //red LED sensor data

int32_t bufferLength; //data length
int32_t spo2; //SP02 value
int8_t validSPO2; //indicador para mostrar si el cálculo de SPO2 es válido
int32_t heartRate;
int8_t validHeartRate; //indicador para mostrar si el cálculo de la frecuencia cardíaca es
    válido
byte readLED = 13;
const int MAX_ANALOG_VAL = 4095;
const float MAX_BATTERY_VOLTAGE = 4.2;
int BatteryP;
byte update = 1;

RTC_DATA_ATTR int bootCount = 0;

Adafruit_MPU6050 mpu;

const char* serverName = "https://signos-vitales-81f9b-default-
rtdb.firebaseio.com/DatoTiempoReal.json";

```

```

// Servidor NTP para solicitar la hora de la época
const char* ntpServer = "pool.ntp.org";
// Variable para guardar el tiempo de época actual
unsigned long epochTime;
// Función que obtiene la hora actual
unsigned long getTime() {
    time_t now;
    struct tm timeinfo;
    if (!getLocalTime(&timeinfo)) {
        //Serial.println("Failed to obtain time");
        return (0);
    }
    time(&now);
    return now;
}

void print_wakeup_reason() {
    esp_sleep_wakeup_cause_t wakeup_reason;

    wakeup_reason = esp_sleep_get_wakeup_cause();

    switch (wakeup_reason) {
        case ESP_SLEEP_WAKEUP_EXT0: Serial.println("Wakeup caused by external signal using
RTC_IO"); break;
        case ESP_SLEEP_WAKEUP_EXT1: Serial.println("Wakeup caused by external signal using
RTC_CNTL"); break;
        case ESP_SLEEP_WAKEUP_TIMER: Serial.println("Wakeup caused by timer"); break;
        case ESP_SLEEP_WAKEUP_TOUCHPAD: Serial.println("Wakeup caused by touchpad"); break;
        case ESP_SLEEP_WAKEUP_ULP: Serial.println("Wakeup caused by ULP program"); break;
        default: Serial.printf("Wakeup was not caused by deep sleep: %d\n", wakeup_reason);
break;
    }
}

void setup(void) {
    Serial.begin(115200);
    pinMode(readLED, OUTPUT);
    delay(1000);
    u8g2.begin();
    u8g2.clearBuffer();
    u8g2.setFont(u8g2_font_logisoso16_tr);
    initWiFi();
    //Incrementar el número de arranque e imprimirlo cada vez que se reinicie
    ++bootCount;
    Serial.println("Boot number: " + String(bootCount));
    print_wakeup_reason();
    esp_sleep_enable_timer_wakeup(TIME_TO_SLEEP * uS_TO_S_FACTOR);
    if (!mpu.begin()) {
        Serial.println("Failed to find MPU6050 chip");
    }
}

```

```

    while (1) {
        delay(10);
    }
}

// Inicializar sensor
if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
{
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1)
        ;
}

particleSensor.setup();//Configurar el sensor con los ajustes por defecto
particleSensor.setPulseAmplitudeRed(0x0A); //Apagar el LED rojo para indicar que el
    sensor está funcionando
particleSensor.setPulseAmplitudeGreen(0); //Apagar el LED verde

mpu.setAccelerometerRange(MPU6050_RANGE_8_G);

mpu.setGyroRange(MPU6050_RANGE_500_DEG);

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
u8g2.clearBuffer(); // borrar la memoria interna

if (WiFi.status() != WL_CONNECTED) {
    u8g2.drawStr(15, 40, "No Conectado");
    //u8g2.sendBuffer(); // transferir la memoria interna a la pantalla
} else {
    u8g2.drawStr(20, 40, "Conectado");
    //u8g2.sendBuffer(); // transferir la memoria interna a la pantalla
}
u8g2.sendBuffer(); // transferir la memoria interna a la pantalla
delay(2000);

configTime(-18000, -18000, ntpServer);
}

void loop() {
    char oled_data[32] = { 0 };

    while (samples < 10) {

        irValue = particleSensor.getIR();

        if (checkForBeat(irValue) == true) {
            //We sensed a beat!

```

```

long delta = millis() - lastBeat;
lastBeat = millis();

beatsPerMinute = 60 / (delta / 1000.0);

if (beatsPerMinute < 255 && beatsPerMinute > 20 && irValue > 50000) {
    rates[rateSpot++] = (byte)beatsPerMinute; //Almacenar esta lectura en el array
    rateSpot %= RATE_SIZE;

    //Tomar la media de las lecturas
    beatAvg = 0;
    for (byte x = 0; x < RATE_SIZE; x++)
        beatAvg += rates[x];
    beatAvg /= RATE_SIZE;
    Serial.println(samples++);
    update = 1;
}
}

if (update == 1) {

    u8g2.clearBuffer();
    BatteryP = getBattery();
    // Se muestra el porcentaje de batería
    u8g2.setFont(u8g2_font_5x8_tf);

    memset(oled_data, 0, sizeof(oled_data));
    sprintf(oled_data, "%d%", BatteryP);
    u8g2.drawStr(100, 10, oled_data);
    u8g2.setFont(u8g2_font_logisoso16_tr);
    u8g2.drawStr(15, 30, "Muestreando");
    u8g2.setFont(u8g2_font_logisoso16_tr);
    memset(oled_data, 0, sizeof(oled_data));
    sprintf(oled_data, "%d%", (samples)*10);
    u8g2.drawStr(45, 50, oled_data);
    u8g2.sendBuffer();
    update = 0;
}
}

delay(1000);

samples = 0;
Serial.print("IR=");
Serial.print(irValue);
Serial.print(", BPM=");
Serial.print(beatsPerMinute);
Serial.print(", Avg BPM=");
Serial.print(beatAvg);

```

```

u8g2.clearBuffer();
memset(oled_data, 0, sizeof(oled_data));
sprintf(oled_data, "BPM: %d", beatAvg);
u8g2.drawStr(20, 39, oled_data);
u8g2.sendBuffer();
delay(1000);

epochTime = getTime();
Serial.print("Epoch Time: ");
Serial.println(epochTime);

u8g2.clearBuffer();
u8g2.drawStr(15, 30, "Muestreando");
u8g2.drawStr(45, 50, "SPO2");
u8g2.sendBuffer();

getSPO2();

u8g2.clearBuffer();

memset(oled_data, 0, sizeof(oled_data));
sprintf(oled_data, "SPO2: %d", spo2);

u8g2.drawStr(20, 39, oled_data);
u8g2.sendBuffer();
sensors_event_t a, g, temp;
mpu.getEvent(&a, &g, &temp);
Serial.print(", Z: ");
Serial.print(a.acceleration.z);
Serial.println(" m/s^2");

if (a.acceleration.z <= 4 && a.acceleration.z > -2) {
  posicion = 1;
} else if (a.acceleration.z > 4) {
  posicion = 2;
} else if (a.acceleration.z <= -2) {
  posicion = 3;
}

Serial.println("");
delay(2000);

httpSend();

delay(2000);

Serial.println("Going to sleep now");

```

```

Serial.flush();
u8g2.clearBuffer();
u8g2.drawStr(15, 39, "Hibernando");
u8g2.sendBuffer(); // transfer internal memory to the display
delay(2000);
u8g2.clearBuffer();
u8g2.sendBuffer(); // transfer internal memory to the display
esp_deep_sleep_start();
Serial.println("This will never be printed");
}

void httpSend() {
  WiFiClientSecure* client = new WiFiClientSecure;
  if (client) {
    client->setCACert(rootCACertificate);

    {
      HTTPClient https;
      Serial.print("[HTTPS] begin...\n");
      if (https.begin(*client, serverName)) {
        Serial.print("[HTTPS] POST...\n");
        String valores = "{\"bateria\": \"" + String(BatteryP) + "\", \"bpm\": \"" +
String(beatAvg) + "\", \"oxigenacion\": \"" + String(spo2) + "\", \"posicion\": \"" +
String(posicion) + "\", \"timestamp\": \"" + String(getTime()) + "000\"}";
        int httpCode = https.POST(valores);
        Serial.printf("[HTTPS] POST... code: %d\n", httpCode);
        if (httpCode = 200) {
          u8g2.clearBuffer();
          u8g2.drawStr(30, 39, "Enviado");
          u8g2.sendBuffer();
        }
        https.end();
      } else {
        Serial.printf("[HTTPS] Unable to connect\n");
      }
    }

    delete client;
  } else {
    Serial.println("Unable to create client");
  }
}

void getSP02() {
  bufferLength = 100;

```

```

for (byte i = 0; i < bufferLength; i++) {
  while (particleSensor.available() == false)
    particleSensor.check();

  redBuffer[i] = particleSensor.getRed();
  irBuffer[i] = particleSensor.getIR();
  particleSensor.nextSample();
}

maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2,
&validSPO2, &heartRate, &validHeartRate);

while (sampllesspo2 <= 10) {

  for (byte i = 25; i < 100; i++) {
    redBuffer[i - 25] = redBuffer[i];
    irBuffer[i - 25] = irBuffer[i];
  }

  for (byte i = 75; i < 100; i++) {
    while (particleSensor.available() == false)
      particleSensor.check();
    digitalWrite(readLED, !digitalRead(readLED));

    redBuffer[i] = particleSensor.getRed();
    irBuffer[i] = particleSensor.getIR();
    particleSensor.nextSample();

    Serial.print(F("SPO2= "));
    Serial.println(spo2, DEC);
    if (spo2 > 80) {
      sampllesspo2++;
    }
  }

  if (sampllesspo2 <= 9) {
    maxim_heart_rate_and_oxygen_saturation(irBuffer, bufferLength, redBuffer, &spo2,
&validSPO2, &heartRate, &validHeartRate);
  }
}

sampllesspo2 = 0;
Serial.print(F("SPO2= "));
Serial.println(spo2, DEC);
}

int getBattery() {
  int rawValue = analogRead(ADC_BAT);

```



```

    float voltageLevel = (rawValue / 4095.0) * 2 * 1.1 * 3.3; // calcular el nivel de
tensión
    float batteryFraction = voltageLevel / MAX_BATTERY_VOLTAGE;
    return batteryFraction * 100;
}

void initWiFi() {
    Serial.print("Attempting to connect to SSID: ");
    Serial.println(ssid);

    u8g2.clearBuffer();
    u8g2.drawStr(5, 30, "Conectando a:");
    u8g2.drawStr(20, 50, ssid);
    u8g2.sendBuffer();

    WiFi.begin(ssid, password);

    // intento de conexión a la red Wifi:
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        // esperar 1 segundo para reintentar
        delay(500);
    }

    Serial.print("Connected to ");
    Serial.println(ssid);
}

```

Anexo B: Código Fuente de notificación en la aplicación móvil “push_notification_service”.dart

```
import 'dart:async';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_messaging/firebase_messaging.dart';
class PushNotificationService {
  static FirebaseMessaging messaging = FirebaseMessaging.instance;
  static String? token;
  static StreamController<String> _messageStream =
    new StreamController.broadcast();
  static Stream<String> get messagesStream => _messageStream.stream;
  static Future _backgroundHandler(RemoteMessage message) async {
    print(message.data);
    _messageStream.add(message.data['product'] ?? 'No data');
  }

  static Future _onMessageHandler(RemoteMessage message) async {
    print(message.data);
    _messageStream.add(message.data['product'] ?? 'No data');
  }
  static Future _onMessageOpenApp(RemoteMessage message) async {
    print(message.data);
    _messageStream.add(message.data['product'] ?? 'No data');
  }
  static Future initializeApp() async {
    await Firebase.initializeApp();
    await requestPermission();
    token = await FirebaseMessaging.instance.getToken();
    print('Token: $token');
    FirebaseMessaging.onBackgroundMessage(_backgroundHandler);
    FirebaseMessaging.onMessage.listen(_onMessageHandler);
    FirebaseMessaging.onMessageOpenedApp.listen(_onMessageOpenApp);
  }
  static requestPermission() async {
    NotificationSettings settings = await messaging.requestPermission(
      alert: true,
      announcement: false,
      badge: true,
      carPlay: false,
      criticalAlert: false,
      provisional: false,
      sound: true);

    print('User push notification status ${settings.authorizationStatus}');
```

```
}  
static closeStreams() {  
    _messageStream.close();  
}  
}
```

Anexo C: Código Fuente de notificación en cloud function firebase para notificaciones

```
const functions = require('firebase-functions');
const admin = require('firebase-admin');
var i=0;
var fcm2;
var fcmReal;
const registrationTokens= [
  'fATaW8csTGCpfRh4Qe15dC:APA91bFv96vwaE0OyWz39j-8Hadxh1TtFI7U4RU11f07133ml-
GxpCYzfiptBDGRulsqc4XNvJbhcrvz04JyJaF6vluyQvuSW7SalmAzXGSJiy9JiowPnwuAOjWJX7ivmy5qj
1Y-c',
  'cB8eSGYzSLWVqMbgsjAIQc:APA91bHj5F910g-
4rD69ftseNkVlJacNlJgKrkhf2Ii_halmhXKb4KDevM9aQVG7w1G8cxBbt9z_JoCh01JUfWKSnb1UnkxQyX4
RluyCjMk6aTaLkv4hyq_TrS2saO_AsS_lT5iZ1Iq',
  'dlbQvx13Q402zZhx0QsDvv:APA91bFzGPuWwyYlLvIdo3V6p05gJXj0_pKtN_jpdMwx2zFXbgNEJZO1
JnIFdAjSS2NtXt3VbvoVb2kfYek7ln-KuscEImcs5PilnNRzNy1d7YIym4xmwwG-VgZgEtY5-1nK6CTqL42',
  'cC2SGbrqTSO4vR0a5wFX81:APA91bE7YwIa-
7dxkC5kK2wHyhkBW7RLpf45grYowJdDNqrPGz9DEEOSqbk3Ed5Eci_AkHiOqdmnxGnmQLF7ZdqmnfwBLcc2Ty
qpDVfvxF0dNczlkFGG-29iRwpJr3WKMGbbuHvnwgH9'
];
var options = {
  priority: "high",
};
admin.initializeApp(functions.config().firebase);
exports.oxigenacion= functions.database.ref('/DatoTiempoReal/').onWrite((change,
context) => {
  admin.database().ref('/DatoTiempoReal/').orderByChild('timestamp').limitToLast(1)
  .once('value', function(snapshot) {
    snapshot.forEach(function(childSnapshot) {
      var childData = childSnapshot.val();

      console.log("oxigenacion");
      console.log(childData.oxigenacion);
      var message='El usuario ha sobrepasado el valor de oxigenacion con un valor
de '+childData.oxigenacion+' %, se recomienda revisar.';
      if(childData.oxigenacion>=84 && childData.oxigenacion<=90){
        const payload = {
          notification:{
            title : 'Hipoxia Moderada',
            body : message,
            badge : '1',
            sound : 'default'
          }
        };
        console.log("Se fue la notificacion");
        admin.messaging().sendToDevice(registrationTokens, payload, options);
      }
      if(childData.oxigenacion>=91 && childData.oxigenacion<=94){
        const payload = {
          notification:{
            title : 'Hipoxia Leve',
            body : message,
            badge : '1',
            sound : 'default'
          }
        };
      }
    });
  });
});
```

```

    };
    console.log("Se fue la notificacion");
    admin.messaging().sendToDevice(registrationTokens, payload, options);
  }

  if(childData.oxigenacion<=83){
    const payload = {
      notification:{
        title : 'Hipoxia Severa',
        body : message,
        badge : '1',
        sound : 'ip.mp3'
      }
    };
    console.log("Se fue la notificacion");
    admin.messaging().sendToDevice(registrationTokens, payload, options);
  }
});
});
});

exports.posicion= functions.database.ref('/DatoTiempoReal/').onWrite((change,
context) => {

  admin.database().ref('/DatoTiempoReal/').orderByChild('timestamp').limitToLast(1)
.once('value', function(snapshot) {
  snapshot.forEach(function(childSnapshot) {
    var childData = childSnapshot.val();

    console.log(childData.posicion);
    var message='El bebe podría estar en riesgo, revisar posición!.';
    if(childData.posicion==2 || childData.posicion==3){
      const payload = {
        notification:{
          title : 'Revisar Porsición',
          body : message,
          badge : '1',
          sound : 'default'
        }
      };
      console.log("Se fue la notificacion");
      admin.messaging().sendToDevice(registrationTokens, payload, options);
    }
  });
});
});

exports.bpm= functions.database.ref('/DatoTiempoReal/').onWrite((change, context) =>
{

  admin.database().ref('/DatoTiempoReal/').orderByChild('timestamp').limitToLast(
1).once('value', function(snapshot) {

```

```

snapshot.forEach(function(childSnapshot) {
  var childData = childSnapshot.val();

  console.log("Este es el valor de bpm");
  console.log(childData.bpm);
  var message='El usuario ha sobrepasado su frecuencia cardiaca máxima con un
valor de '+childData.bpm+' bpm, se recomienda revision.';
  if(childData.bpm<80 || childData.bpm>130){
    const payload = {
      notification:{
        title : 'Alerta BPM sobrepasada',
        body : message,
        badge : '1',
        sound : 'default'
      }
    };
    console.log("Se fue la notificacion");
    admin.messaging().sendToDevice(registrationTokens, payload, options);
  }
});
});
});

exports.battery= functions.database.ref('/DatoTiempoReal/').onWrite((change,
context) => {

  admin.database().ref('/DatoTiempoReal/').orderByChild('timestamp').limitToLast(
1).once('value', function(snapshot) {
  snapshot.forEach(function(childSnapshot) {
    var childData = childSnapshot.val();
    console.log("Bateria!!!");
    console.log(childData.bateria);
    if(childData.bateria<=20){
      const payload = {
        notification:{
          title : 'Cargar bateria',
          body : 'La indumentaria necesita recargarse, tiene menos del 20%!',
          badge : '1',
          sound : 'default'
        }
      };
      console.log("Se fue la notificacion");
      admin.messaging().sendToDevice(registrationTokens, payload, options);
    }
  });
});
});
});

```

Anexo D: Código de la aplicación Signos Vitales en repositorio Github

Enlace de acceso:

[Codigo Fuente](#)

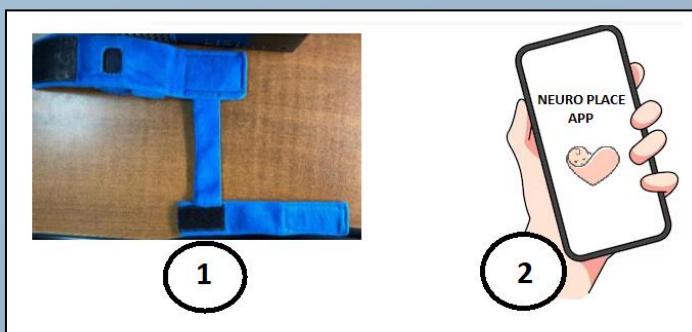
MANUAL DE USO

SMART TEXTIL PARA EL MONITOREO NO INVASIVO Y
SEGURIDAD PREVENTIVA DE FACTORES CAUSANTES DEL
SÍNDROME DE MUERTE SÚBITA DE LOS LACTANTES



Para monitorear al lactante se necesita de 2 elementos que son:

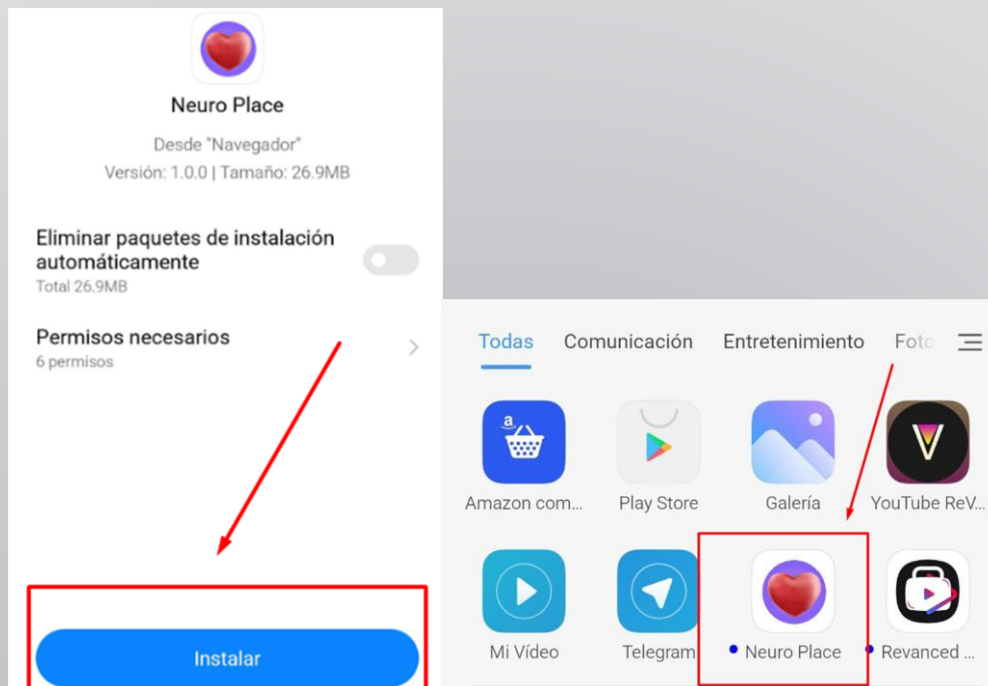
1. Smart textil
2. Smartphone con la aplicación móvil Neuro Place



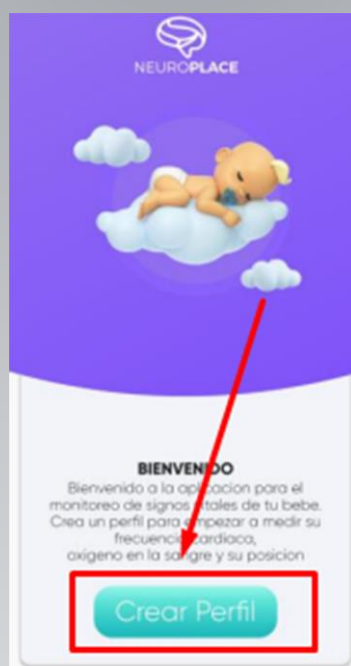
Instalación de aplicación

1. Hacemos clic en el hipervínculo a continuación o escanea el código QR.

[Descargar aplicación iHeart](#)



4. Una vez instalada la aplicación se procede a ingresar nuestros datos del lactante, presionamos el botón de *CREAR PERFIL*, que posteriormente nos llevará al formulario donde deberemos llenar la información necesaria.



The screenshot shows the 'NEUROPLACE DATOS' form. It includes fields for 'Nombre' (Name) and 'Apellido' (Last Name). Under 'Sexo:' (Sex), there are two options: 'Niño' (Boy) with a male emoji and 'Niña' (Girl) with a female emoji. Below these are fields for 'Edad:' (Age), 'Prenda que usa:' (Clothing), and 'Lugar de descanso:' (Resting place). At the bottom, there is a toggle switch for '¿Se encuentra recién alimentado?' (Is it recently fed?) with 'No' and 'Si' options.

Una vez que se llena el formulario de datos se procede a presionar el botón de guardar, esto nos llevará a la pantalla de Home, donde observaremos los perfiles creados, para empezar la medición debemos presionar el botón EMPEZAR.

The screenshot shows the user profile page for 'Jesus vasquez'. It features a profile picture of a baby and the name 'Jesus vasquez'. The profile includes the following details: 'Edad: 3 meses', 'Alimentado: Si', 'Vestimenta: saco', and 'Lugar de descanso: cama'. There is an 'Observaciones:' field. At the bottom, the 'EMPEZAR' button is highlighted with a red box and a red arrow. Below it are 'Editar Perfil' and 'Eliminar' buttons.

3

Uso de Smart Textil

Para poder visualizar los datos, se debe colocar el Smart Textil de forma correcta teniendo en cuenta las siguientes consideraciones:

- Para asegurar la correcta ubicación de los sensores se debe de usar el velcro que forma una correa por encima del tobillo del lactante



- Para encender la indumentaria únicamente se debe mover el switch ubicado en parte lateral izquierda, una vez encendido el dispositivo este se conectará a la red automáticamente



Smart textil en funcionamiento

Para comprobar que el Smart textil ya se encuentra en funcionamiento, observaremos la pantalla en el módulo extraíble con los siguientes mensajes

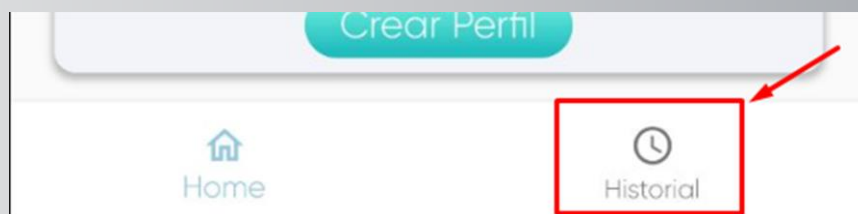


Una vez colocado el Smart Textil y este conectado a la red en la aplicación móvil, en la pantalla a continuación se muestran los datos recolectados en tiempo real, podremos observar:

1. Porcentaje de batería del Smart Textil
2. Frecuencia Cardíaca
3. Oxigenación en sangre
4. Posición

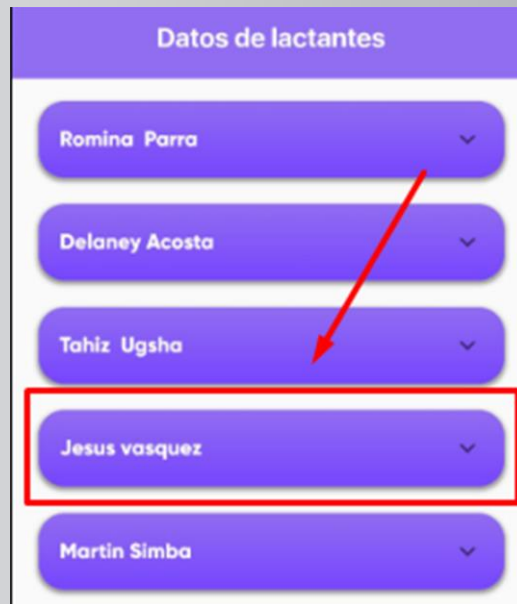


Una vez recolectado los datos necesarios se procede a terminar la medición, para poder revisar el historial de los datos recolectados se procede a presionar el apartado de historial:



6

A continuación se desplegará una pantalla con un lista de todos los lactantes con sus respectivas mediciones, procedemos a seleccionar el nombre del lactante que queremos ver el registro



A continuación se mostrará los datos que se han recolectado durante la medición, así como los niveles de los mismos, en este caso observamos que todas las mediciones estuvieron en un nivel normal.

← Jesus vasquez Martes, 27 de junio de 2023 9:46 a. m.			
Indice	Fecha	Frecuencia Cardíaca	Oxígeno en la sangre
1	2023-06-27 09:46:39	121 BPM Normal	94.00% Normal
2	2023-06-27 09:46:17	123 BPM Normal	94.00% Normal
3	2023-06-27 09:45:54	117 BPM Normal	95.00% Normal
4	2023-06-27 09:45:36	117 BPM Normal	96.00% Normal
5	2023-06-27 09:45:14	117 BPM Normal	94.00% Normal
6	2023-06-27 09:44:51	122 BPM Normal	95.00% Normal
7	2023-06-27 09:44:28	125 BPM Normal	94.00% Normal

Anexo F: Pruebas Realizadas

- Prueba N° 1

NEUROPLACE
DATOS

Lactante NN1

Edad: 9 meses

Alimentado: No

Vestimenta: Muselina

Lugar de descanso: Cama

Observaciones: Ninguna

EMPEZAR

Editar Perfil Eliminar

Lactante NN1			
#	BPM	SPO2	Fecha-hora
1	115	96	27/06/2023 15:32
2	115	96	27/06/2023 15:32
3	117	98	27/06/2023 15:33
4	116	97	27/06/2023 15:33
5	114	98	27/06/2023 15:33
6	112	93	27/06/2023 15:34
7	113	97	27/06/2023 15:34
8	111	94	27/06/2023 15:34
9	117	97	27/06/2023 15:35
10	115	95	27/06/2023 15:35
11	111	94	27/06/2023 15:35
12	112	98	27/06/2023 15:36
13	117	98	27/06/2023 15:36
14	111	96	27/06/2023 15:36
15	113	93	27/06/2023 15:37
16	113	97	27/06/2023 15:37
17	114	94	27/06/2023 15:37
18	112	96	27/06/2023 15:38
19	116	98	27/06/2023 15:38

20	114	96	27/06/2023 15:38
21	117	96	27/06/2023 15:39
22	117	94	27/06/2023 15:39
23	116	98	27/06/2023 15:39
24	115	94	27/06/2023 15:40
25	113	97	27/06/2023 15:40
26	116	97	27/06/2023 15:40
27	116	97	27/06/2023 15:41
28	111	98	27/06/2023 15:41
29	111	97	27/06/2023 15:41
30	116	94	27/06/2023 15:42
31	112	94	27/06/2023 15:42
32	117	97	27/06/2023 15:42
33	111	93	27/06/2023 15:43
34	117	97	27/06/2023 15:43
35	116	95	27/06/2023 15:43
36	116	94	27/06/2023 15:44
37	112	94	27/06/2023 15:44
38	115	94	27/06/2023 15:44
39	114	96	27/06/2023 15:45
40	113	97	27/06/2023 15:45
41	114	94	27/06/2023 15:45
42	116	97	27/06/2023 15:46
43	117	96	27/06/2023 15:46
44	117	94	27/06/2023 15:46
45	113	97	27/06/2023 15:46
46	113	97	27/06/2023 15:47
47	117	98	27/06/2023 15:47
48	114	93	27/06/2023 15:47
49	114	95	27/06/2023 15:48
50	113	93	27/06/2023 15:48
51	117	94	27/06/2023 15:48
52	114	97	27/06/2023 15:49
53	116	98	27/06/2023 15:49
54	117	95	27/06/2023 15:49
55	115	96	27/06/2023 15:50
56	113	97	27/06/2023 15:50
57	115	97	27/06/2023 15:50
58	111	93	27/06/2023 15:51
59	113	94	27/06/2023 15:51
60	113	95	27/06/2023 15:51
61	113	94	27/06/2023 15:52
62	111	96	27/06/2023 15:52
63	114	93	27/06/2023 15:52
64	114	96	27/06/2023 15:53
65	113	97	27/06/2023 15:53

66	117	98	27/06/2023 15:53
67	113	94	27/06/2023 15:54
68	115	93	27/06/2023 15:54
69	114	97	27/06/2023 15:54
70	115	98	27/06/2023 15:55
71	112	96	27/06/2023 15:55
72	117	93	27/06/2023 15:55
73	117	94	27/06/2023 15:56
74	114	93	27/06/2023 15:56
75	112	93	27/06/2023 15:56
76	113	95	27/06/2023 15:57
77	116	95	27/06/2023 15:57
78	116	98	27/06/2023 15:57
79	117	96	27/06/2023 15:58
80	113	95	27/06/2023 15:58
81	117	96	27/06/2023 15:58
82	114	93	27/06/2023 15:59
83	116	97	27/06/2023 15:59
84	112	98	27/06/2023 15:59
85	113	93	27/06/2023 16:00

- Prueba N° 2

Lactante NN2			
#	BPM	SPO2	Fecha-hora
1	120	96	27/06/2023 9:11
2	117	96	27/06/2023 9:11
3	123	97	27/06/2023 9:12
4	119	93	27/06/2023 9:12
5	118	94	27/06/2023 9:12
6	122	94	27/06/2023 9:13
7	121	97	27/06/2023 9:13
8	117	96	27/06/2023 9:13
9	124	94	27/06/2023 9:14
10	120	94	27/06/2023 9:14
11	122	93	27/06/2023 9:14
12	121	96	27/06/2023 9:15
13	120	97	27/06/2023 9:15
14	122	93	27/06/2023 9:15
15	123	96	27/06/2023 9:16
16	121	97	27/06/2023 9:16
17	119	94	27/06/2023 9:16
18	119	95	27/06/2023 9:17
19	121	94	27/06/2023 9:17
20	117	95	27/06/2023 9:17
21	121	93	27/06/2023 9:18

22	124	97	27/06/2023 9:18
23	125	96	27/06/2023 9:18
24	117	95	27/06/2023 9:19
25	121	95	27/06/2023 9:19
26	119	93	27/06/2023 9:19
27	122	97	27/06/2023 9:19
28	121	94	27/06/2023 9:20
29	118	94	27/06/2023 9:20
30	117	94	27/06/2023 9:20
31	120	94	27/06/2023 9:21
32	118	95	27/06/2023 9:21
33	118	95	27/06/2023 9:21
34	121	97	27/06/2023 9:22
35	122	94	27/06/2023 9:22
36	118	95	27/06/2023 9:22
37	123	94	27/06/2023 9:23
38	121	96	27/06/2023 9:23
39	120	94	27/06/2023 9:23
40	125	96	27/06/2023 9:24
41	119	93	27/06/2023 9:24
42	122	94	27/06/2023 9:24
43	125	93	27/06/2023 9:25
44	121	97	27/06/2023 9:25
45	122	97	27/06/2023 9:25
46	120	93	27/06/2023 9:26
47	124	97	27/06/2023 9:26
48	125	96	27/06/2023 9:26
49	125	94	27/06/2023 9:27
50	118	97	27/06/2023 9:27
51	119	95	27/06/2023 9:27
52	124	93	27/06/2023 9:28
53	117	94	27/06/2023 9:28
54	121	96	27/06/2023 9:28
55	119	95	27/06/2023 9:28
56	120	95	27/06/2023 9:29
57	121	93	27/06/2023 9:29
58	124	94	27/06/2023 9:29
59	125	94	27/06/2023 9:30
60	121	93	27/06/2023 9:30
61	123	95	27/06/2023 9:30
62	125	93	27/06/2023 9:31
63	124	94	27/06/2023 9:31
64	121	97	27/06/2023 9:31
65	118	93	27/06/2023 9:32
66	123	93	27/06/2023 9:32
67	125	94	27/06/2023 9:32

68	125	93	27/06/2023 9:33
69	119	94	27/06/2023 9:33
70	117	97	27/06/2023 9:33
71	124	93	27/06/2023 9:34
72	125	96	27/06/2023 9:34
73	119	96	27/06/2023 9:34
74	120	95	27/06/2023 9:35
75	122	97	27/06/2023 9:35
76	120	96	27/06/2023 9:35
77	118	96	27/06/2023 9:36
78	118	96	27/06/2023 9:36
79	119	95	27/06/2023 9:36
80	125	97	27/06/2023 9:36
81	125	93	27/06/2023 9:37
82	124	96	27/06/2023 9:37
83	119	93	27/06/2023 9:37
84	123	93	27/06/2023 9:38
85	124	95	27/06/2023 9:38
86	124	94	27/06/2023 9:38
87	123	95	27/06/2023 9:39
88	125	96	27/06/2023 9:39
89	121	97	27/06/2023 9:39
90	118	97	27/06/2023 9:40
91	123	95	27/06/2023 9:40
92	123	96	27/06/2023 9:40
93	117	97	27/06/2023 9:40
94	123	96	27/06/2023 9:41
95	120	95	27/06/2023 9:41
96	122	93	27/06/2023 9:41
97	117	93	27/06/2023 9:42
98	125	93	27/06/2023 9:42
99	117	95	27/06/2023 9:42
100	125	97	27/06/2023 9:43
101	125	97	27/06/2023 9:43
102	123	95	27/06/2023 9:43
103	124	96	27/06/2023 9:44
104	125	94	27/06/2023 9:44
105	122	95	27/06/2023 9:44
106	117	94	27/06/2023 9:45
107	117	96	27/06/2023 9:45
108	117	95	27/06/2023 9:45
109	123	94	27/06/2023 9:46
110	121	94	27/06/2023 9:46

- Prueba N° 3



Lactante NN3			
#	BPM	SPO2	Fecha-hora
1	120	96	27/06/2023 9:11
2	117	96	27/06/2023 9:11
3	123	97	27/06/2023 9:12
4	119	93	27/06/2023 9:12
5	118	94	27/06/2023 9:12
6	122	94	27/06/2023 9:13
7	121	97	27/06/2023 9:13
8	117	96	27/06/2023 9:13
9	124	94	27/06/2023 9:14
10	120	94	27/06/2023 9:14
11	122	93	27/06/2023 9:14
12	121	96	27/06/2023 9:15
13	120	97	27/06/2023 9:15
14	122	93	27/06/2023 9:15
15	123	96	27/06/2023 9:16
16	121	97	27/06/2023 9:16
17	119	94	27/06/2023 9:16
18	119	95	27/06/2023 9:17
19	121	94	27/06/2023 9:17
20	117	95	27/06/2023 9:17

21	121	93	27/06/2023 9:18
22	124	97	27/06/2023 9:18
23	125	96	27/06/2023 9:18
24	117	95	27/06/2023 9:19
25	121	95	27/06/2023 9:19
26	119	93	27/06/2023 9:19
27	122	97	27/06/2023 9:19
28	121	94	27/06/2023 9:20
29	118	94	27/06/2023 9:20
30	117	94	27/06/2023 9:20
31	120	94	27/06/2023 9:21
32	118	95	27/06/2023 9:21
33	118	95	27/06/2023 9:21
34	121	97	27/06/2023 9:22
35	122	94	27/06/2023 9:22
36	118	95	27/06/2023 9:22
37	123	94	27/06/2023 9:23
38	121	96	27/06/2023 9:23
39	120	94	27/06/2023 9:23
40	125	96	27/06/2023 9:24
41	119	93	27/06/2023 9:24
42	122	94	27/06/2023 9:24
43	125	93	27/06/2023 9:25
44	121	97	27/06/2023 9:25
45	122	97	27/06/2023 9:25
46	120	93	27/06/2023 9:26
47	124	97	27/06/2023 9:26
48	125	96	27/06/2023 9:26
49	125	94	27/06/2023 9:27
50	118	97	27/06/2023 9:27
51	119	95	27/06/2023 9:27
52	124	93	27/06/2023 9:28
53	117	94	27/06/2023 9:28
54	121	96	27/06/2023 9:28
55	119	95	27/06/2023 9:28
56	120	95	27/06/2023 9:29
57	121	93	27/06/2023 9:29
58	124	94	27/06/2023 9:29
59	125	94	27/06/2023 9:30
60	121	93	27/06/2023 9:30
61	123	95	27/06/2023 9:30
62	125	93	27/06/2023 9:31
63	124	94	27/06/2023 9:31
64	121	97	27/06/2023 9:31
65	118	93	27/06/2023 9:32
66	123	93	27/06/2023 9:32

67	125	94	27/06/2023 9:32
68	125	93	27/06/2023 9:33
69	119	94	27/06/2023 9:33
70	117	97	27/06/2023 9:33
71	124	93	27/06/2023 9:34
72	125	96	27/06/2023 9:34
73	119	96	27/06/2023 9:34
74	120	95	27/06/2023 9:35
75	122	97	27/06/2023 9:35
76	120	96	27/06/2023 9:35
77	118	96	27/06/2023 9:36
78	118	96	27/06/2023 9:36
79	119	95	27/06/2023 9:36
80	125	97	27/06/2023 9:36
81	125	93	27/06/2023 9:37
82	124	96	27/06/2023 9:37
83	119	93	27/06/2023 9:37
84	123	93	27/06/2023 9:38
85	124	95	27/06/2023 9:38
86	124	94	27/06/2023 9:38
87	123	95	27/06/2023 9:39
88	125	96	27/06/2023 9:39
89	121	97	27/06/2023 9:39
90	118	97	27/06/2023 9:40
91	123	95	27/06/2023 9:40
92	123	96	27/06/2023 9:40
93	117	97	27/06/2023 9:40
94	123	96	27/06/2023 9:41
95	120	95	27/06/2023 9:41
96	122	93	27/06/2023 9:41
97	117	93	27/06/2023 9:42
98	125	93	27/06/2023 9:42
99	117	95	27/06/2023 9:42
100	125	97	27/06/2023 9:43
101	125	97	27/06/2023 9:43
102	123	95	27/06/2023 9:43
103	124	96	27/06/2023 9:44
104	125	94	27/06/2023 9:44
105	122	95	27/06/2023 9:44
106	117	94	27/06/2023 9:45
107	117	96	27/06/2023 9:45
108	117	95	27/06/2023 9:45
109	123	94	27/06/2023 9:46
110	121	94	27/06/2023 9:46

- Prueba N° 4



Lactante NN4			
#	BPM	SPO2	Fecha-hora
1	120	98	30/06/2023 9:05
2	120	93	30/06/2023 9:05
3	125	98	30/06/2023 9:06
4	119	93	30/06/2023 9:06
5	121	98	30/06/2023 9:06
6	119	98	30/06/2023 9:07
7	125	97	30/06/2023 9:07
8	119	93	30/06/2023 9:07
9	123	97	30/06/2023 9:07
10	125	98	30/06/2023 9:08
11	122	93	30/06/2023 9:08
12	123	93	30/06/2023 9:09
13	120	94	30/06/2023 9:09
14	119	94	30/06/2023 9:09
15	119	94	30/06/2023 9:09
16	122	96	30/06/2023 9:10
17	121	97	30/06/2023 9:10
18	123	98	30/06/2023 9:10
19	125	93	30/06/2023 9:11
20	119	93	30/06/2023 9:11
21	119	95	30/06/2023 9:11

22	119	93	30/06/2023 9:12
23	121	98	30/06/2023 9:12
24	122	98	30/06/2023 9:12
25	124	95	30/06/2023 9:13
26	125	96	30/06/2023 9:13
27	123	98	30/06/2023 9:13
28	125	95	30/06/2023 9:14
29	124	93	30/06/2023 9:14
30	123	93	30/06/2023 9:15
31	124	95	30/06/2023 9:15
32	121	94	30/06/2023 9:15
33	123	93	30/06/2023 9:16
34	124	95	30/06/2023 9:16
35	125	98	30/06/2023 9:16
36	125	96	30/06/2023 9:17
37	123	94	30/06/2023 9:17
38	122	93	30/06/2023 9:17
39	123	95	30/06/2023 9:18
40	121	96	30/06/2023 9:18
41	125	96	30/06/2023 9:18
42	123	98	30/06/2023 9:19
43	119	93	30/06/2023 9:19
44	121	96	30/06/2023 9:19
45	123	96	30/06/2023 9:20

- Prueba N° 5



Lactante NN5			
#	BPM	SPO2	Fecha-hora
1	115	96	28/06/2023 15:12
2	113	99	28/06/2023 15:12
3	113	98	28/06/2023 15:13
4	111	96	28/06/2023 15:13
5	110	98	28/06/2023 15:13
6	114	98	28/06/2023 15:14
7	113	97	28/06/2023 15:14
8	111	96	28/06/2023 15:14
9	115	98	28/06/2023 15:15
10	112	99	28/06/2023 15:15
11	112	97	28/06/2023 15:15
12	115	99	28/06/2023 15:16
13	111	96	28/06/2023 15:16
14	112	97	28/06/2023 15:16
15	111	96	28/06/2023 15:17
16	113	97	28/06/2023 15:17
17	111	97	28/06/2023 15:17
18	115	98	28/06/2023 15:18
19	113	97	28/06/2023 15:18
20	110	99	28/06/2023 15:18
21	111	96	28/06/2023 15:18

22	114	99	28/06/2023 15:19
23	111	99	28/06/2023 15:19
24	111	96	28/06/2023 15:19
25	110	99	28/06/2023 15:20
26	115	99	28/06/2023 15:20
27	115	99	28/06/2023 15:20
28	112	96	28/06/2023 15:21
29	112	98	28/06/2023 15:21
30	114	97	28/06/2023 15:21
31	113	97	28/06/2023 15:22
32	111	99	28/06/2023 15:22
33	113	98	28/06/2023 15:22
34	113	99	28/06/2023 15:23
35	115	97	28/06/2023 15:23
36	114	97	28/06/2023 15:23
37	111	96	28/06/2023 15:24
38	111	96	28/06/2023 15:24
39	114	97	28/06/2023 15:24
40	112	99	28/06/2023 15:25
41	114	96	28/06/2023 15:25
42	111	97	28/06/2023 15:25
43	110	97	28/06/2023 15:26
44	112	98	28/06/2023 15:26
45	115	98	28/06/2023 15:26
46	114	96	28/06/2023 15:27
47	110	98	28/06/2023 15:27
48	114	96	28/06/2023 15:27
49	111	99	28/06/2023 15:28
50	110	96	28/06/2023 15:28
51	110	96	28/06/2023 15:28
52	111	96	28/06/2023 15:29
53	114	99	28/06/2023 15:29
54	113	97	28/06/2023 15:29
55	114	99	28/06/2023 15:29
56	111	98	28/06/2023 15:30
57	113	99	28/06/2023 15:30
58	112	97	28/06/2023 15:30
59	110	97	28/06/2023 15:31
60	112	97	28/06/2023 15:31
61	115	99	28/06/2023 15:31
62	112	97	28/06/2023 15:32
63	113	98	28/06/2023 15:32

- Prueba N° 6



Lactante NN6			
#	BPM	SPO2	Fecha-hora
1	123	94	23/06/2023 12:12
2	123	98	23/06/2023 12:12
3	119	97	23/06/2023 12:12
4	119	97	23/06/2023 12:13
5	118	94	23/06/2023 12:13
6	123	94	23/06/2023 12:14
7	119	98	23/06/2023 12:14
8	120	98	23/06/2023 12:14
9	120	94	23/06/2023 12:15
10	118	98	23/06/2023 12:15
11	118	96	23/06/2023 12:15
12	122	95	23/06/2023 12:16
13	120	94	23/06/2023 12:16
14	122	96	23/06/2023 12:16
15	119	94	23/06/2023 12:17
16	118	94	23/06/2023 12:17
17	120	96	23/06/2023 12:18
18	119	95	23/06/2023 12:18
19	120	94	23/06/2023 12:18
20	118	98	23/06/2023 12:19
21	119	96	23/06/2023 12:19
22	121	98	23/06/2023 12:19

23	121	95	23/06/2023 12:20
24	122	96	23/06/2023 12:20
25	119	98	23/06/2023 12:20
26	120	94	23/06/2023 12:21
27	119	98	23/06/2023 12:21
28	123	98	23/06/2023 12:21
29	118	98	23/06/2023 12:22
30	121	97	23/06/2023 12:22
31	118	95	23/06/2023 12:22
32	118	98	23/06/2023 12:23
33	121	98	23/06/2023 12:23
34	122	94	23/06/2023 12:23
35	123	95	23/06/2023 12:24
36	120	94	23/06/2023 12:24
37	122	94	23/06/2023 12:24
38	118	98	23/06/2023 12:25
39	119	95	23/06/2023 12:25
40	119	94	23/06/2023 12:25
41	119	98	23/06/2023 12:26
42	121	96	23/06/2023 12:26
43	121	97	23/06/2023 12:27
44	122	97	23/06/2023 12:27
45	120	95	23/06/2023 12:27
46	119	97	23/06/2023 12:28
47	118	98	23/06/2023 12:28
48	119	95	23/06/2023 12:28
49	119	97	23/06/2023 12:29
50	123	97	23/06/2023 12:29
51	119	98	23/06/2023 12:29
52	121	96	23/06/2023 12:30
53	120	96	23/06/2023 12:30
54	121	98	23/06/2023 12:30
55	121	95	23/06/2023 12:31
56	121	94	23/06/2023 12:31
57	120	94	23/06/2023 12:31
58	122	95	23/06/2023 12:32
59	118	94	23/06/2023 12:32
60	123	96	23/06/2023 12:32
61	122	97	23/06/2023 12:33
62	122	96	23/06/2023 12:33
63	119	98	23/06/2023 12:34
64	119	98	23/06/2023 12:34
65	118	97	23/06/2023 12:34
66	123	98	23/06/2023 12:35
67	118	94	23/06/2023 12:35
68	120	96	23/06/2023 12:35

69	122	96	23/06/2023 12:36
70	119	98	23/06/2023 12:36
71	120	98	23/06/2023 12:36
72	121	95	23/06/2023 12:37
73	123	97	23/06/2023 12:37
74	119	94	23/06/2023 12:37
75	118	98	23/06/2023 12:38
76	122	98	23/06/2023 12:38
77	119	94	23/06/2023 12:38
78	119	97	23/06/2023 12:39
79	120	94	23/06/2023 12:39
80	118	98	23/06/2023 12:39
81	121	94	23/06/2023 12:40
82	118	94	23/06/2023 12:40
83	119	97	23/06/2023 12:40
84	122	97	23/06/2023 12:41
85	120	97	23/06/2023 12:41
86	119	97	23/06/2023 12:42
87	121	96	23/06/2023 12:42
88	119	95	23/06/2023 12:42
89	118	94	23/06/2023 12:43
90	122	96	23/06/2023 12:43
91	120	98	23/06/2023 12:43
92	123	96	23/06/2023 12:44
93	122	98	23/06/2023 12:44
94	118	96	23/06/2023 12:44
95	119	97	23/06/2023 12:45
96	123	95	23/06/2023 12:45
97	118	96	23/06/2023 12:45
98	121	97	23/06/2023 12:46
99	120	98	23/06/2023 12:46
100	123	96	23/06/2023 12:46
101	113	95	23/06/2023 12:47
102	117	94	23/06/2023 12:47
103	116	94	23/06/2023 12:47
104	117	95	23/06/2023 12:47
105	113	98	23/06/2023 12:48
106	117	95	23/06/2023 12:48
107	114	94	23/06/2023 12:48
108	115	95	23/06/2023 12:49
109	113	97	23/06/2023 12:49
110	117	97	23/06/2023 12:49
111	113	97	23/06/2023 12:50
112	116	98	23/06/2023 12:50
113	114	98	23/06/2023 12:50
114	114	95	23/06/2023 12:51

115	117	98	23/06/2023 12:51
116	116	96	23/06/2023 12:52
117	114	98	23/06/2023 12:52
118	113	97	23/06/2023 12:52
119	112	95	23/06/2023 12:53
120	112	95	23/06/2023 12:53
121	113	95	23/06/2023 12:53
122	114	98	23/06/2023 12:54
123	113	97	23/06/2023 12:54
124	112	95	23/06/2023 12:54
125	114	98	23/06/2023 12:55
126	117	97	23/06/2023 12:55

- Prueba N° 7

NEUROPLACE
DATOS

Lactante NN7

Edad:
13 meses

Alimentado:
No

Vestimenta:
Camiseta

Lugar de descanso:
Cama

Observaciones:
Ninguna

EMPEZAR

Editar Perfil Eliminar

Lactante NN7			
#	BPM	SPO2	Fecha-hora
1	113	97	28/06/2023 11:37
2	108	97	28/06/2023 11:37
3	111	96	28/06/2023 11:37
4	109	97	28/06/2023 11:38
5	108	99	28/06/2023 11:38
6	108	99	28/06/2023 11:39
7	113	97	28/06/2023 11:39
8	111	99	28/06/2023 11:39
9	109	98	28/06/2023 11:39
10	112	96	28/06/2023 11:40
11	108	97	28/06/2023 11:40
12	112	96	28/06/2023 11:40
13	108	97	28/06/2023 11:41
14	112	99	28/06/2023 11:41
15	110	98	28/06/2023 11:42
16	111	98	28/06/2023 11:42
17	112	99	28/06/2023 11:42
18	108	96	28/06/2023 11:42
19	112	97	28/06/2023 11:43
20	109	97	28/06/2023 11:43
21	112	97	28/06/2023 11:43

22	109	97	28/06/2023 11:44
23	107	97	28/06/2023 11:44
24	107	97	28/06/2023 11:45
25	113	98	28/06/2023 11:45
26	110	98	28/06/2023 11:45
27	112	97	28/06/2023 11:46
28	109	98	28/06/2023 11:46
29	107	99	28/06/2023 11:46
30	110	98	28/06/2023 11:47
31	111	97	28/06/2023 11:47
32	110	98	28/06/2023 11:47
33	107	96	28/06/2023 11:48
34	107	96	28/06/2023 11:48
35	107	99	28/06/2023 11:48
36	110	97	28/06/2023 11:49
37	108	99	28/06/2023 11:49
38	107	98	28/06/2023 11:49
39	113	98	28/06/2023 11:49
40	113	96	28/06/2023 11:50
41	107	96	28/06/2023 11:50
42	112	98	28/06/2023 11:51
43	112	98	28/06/2023 11:51
44	113	98	28/06/2023 11:51
45	109	98	28/06/2023 11:52
46	110	96	28/06/2023 11:52
47	109	97	28/06/2023 11:52
48	111	97	28/06/2023 11:53
49	113	96	28/06/2023 11:53
50	108	96	28/06/2023 11:53
51	110	97	28/06/2023 11:54
52	108	99	28/06/2023 11:54
53	110	96	28/06/2023 11:54
54	107	99	28/06/2023 11:55
55	109	98	28/06/2023 11:55
56	109	98	28/06/2023 11:55
57	111	97	28/06/2023 11:55
58	109	97	28/06/2023 11:56
59	109	99	28/06/2023 11:56
60	113	98	28/06/2023 11:56
61	113	99	28/06/2023 11:57
62	109	97	28/06/2023 11:57
63	113	99	28/06/2023 11:57
64	113	96	28/06/2023 11:58
65	110	98	28/06/2023 11:58
66	111	99	28/06/2023 11:58
67	110	96	28/06/2023 11:59

68	109	96	28/06/2023 11:59
69	112	99	28/06/2023 11:59
70	111	97	28/06/2023 12:00
71	113	96	28/06/2023 12:00
72	109	98	28/06/2023 12:00
73	108	99	28/06/2023 12:01
74	109	97	28/06/2023 12:01
75	109	97	28/06/2023 12:01
76	112	96	28/06/2023 12:02
77	109	97	28/06/2023 12:02
78	112	98	28/06/2023 12:02
79	111	99	28/06/2023 12:02
80	112	97	28/06/2023 12:03
81	113	99	28/06/2023 12:03
82	113	97	28/06/2023 12:03
83	109	97	28/06/2023 12:04
84	113	98	28/06/2023 12:04
85	110	96	28/06/2023 12:04
86	112	98	28/06/2023 12:05
87	113	99	28/06/2023 12:05
88	110	96	28/06/2023 12:05
89	112	99	28/06/2023 12:06
90	111	97	28/06/2023 12:06
91	109	97	28/06/2023 12:06
92	109	96	28/06/2023 12:07
93	110	96	28/06/2023 12:07
94	112	96	28/06/2023 12:07
95	109	96	28/06/2023 12:08
96	112	96	28/06/2023 12:08
97	107	99	28/06/2023 12:08

- Prueba N° 8

Lactante NN8			
#	BPM	SPO2	Fecha-hora
1	104	94	28/06/2023 8:01
2	100	94	28/06/2023 8:02
3	102	94	28/06/2023 8:02
4	100	96	28/06/2023 8:02
5	104	94	28/06/2023 8:02
6	104	94	28/06/2023 8:03
7	104	94	28/06/2023 8:03
8	100	95	28/06/2023 8:03
9	104	95	28/06/2023 8:04
10	103	94	28/06/2023 8:04
11	100	96	28/06/2023 8:04
12	101	95	28/06/2023 8:05
13	101	96	28/06/2023 8:05
14	104	94	28/06/2023 8:05
15	103	94	28/06/2023 8:06
16	104	96	28/06/2023 8:06
17	100	95	28/06/2023 8:06
18	104	94	28/06/2023 8:07
19	104	96	28/06/2023 8:07
20	103	96	28/06/2023 8:07
21	100	95	28/06/2023 8:08
22	102	95	28/06/2023 8:08
23	101	94	28/06/2023 8:08

24	104	96	28/06/2023 8:09
25	104	94	28/06/2023 8:09
26	104	94	28/06/2023 8:09
27	103	96	28/06/2023 8:10
28	100	96	28/06/2023 8:10
29	101	94	28/06/2023 8:11
30	103	96	28/06/2023 8:11
31	104	96	28/06/2023 8:11
32	101	96	28/06/2023 8:12
33	103	95	28/06/2023 8:12
34	103	94	28/06/2023 8:12
35	103	95	28/06/2023 8:13
36	102	95	28/06/2023 8:13
37	102	95	28/06/2023 8:13
38	103	94	28/06/2023 8:14
39	104	94	28/06/2023 8:14
40	103	96	28/06/2023 8:14
41	103	94	28/06/2023 8:15
42	102	94	28/06/2023 8:15
43	102	94	28/06/2023 8:15
44	100	96	28/06/2023 8:16
45	101	96	28/06/2023 8:16
46	101	96	28/06/2023 8:16
47	101	95	28/06/2023 8:17
48	101	94	28/06/2023 8:17
49	103	94	28/06/2023 8:17
50	100	94	28/06/2023 8:18
51	104	95	28/06/2023 8:18
52	100	95	28/06/2023 8:18
53	103	96	28/06/2023 8:19
54	101	96	28/06/2023 8:19
55	100	94	28/06/2023 8:19
56	103	95	28/06/2023 8:20
57	100	95	28/06/2023 8:20
58	101	95	28/06/2023 8:20
59	103	94	28/06/2023 8:21
60	100	96	28/06/2023 8:21
61	103	94	28/06/2023 8:21
62	100	95	28/06/2023 8:22
63	101	96	28/06/2023 8:22

Anexo G: eficiencia del equipo utilizando programa estadístico: Minitab

BPM

Análisis de regresión: BPM GL3200 vs. BPM Smart Textil

La ecuación de regresión es

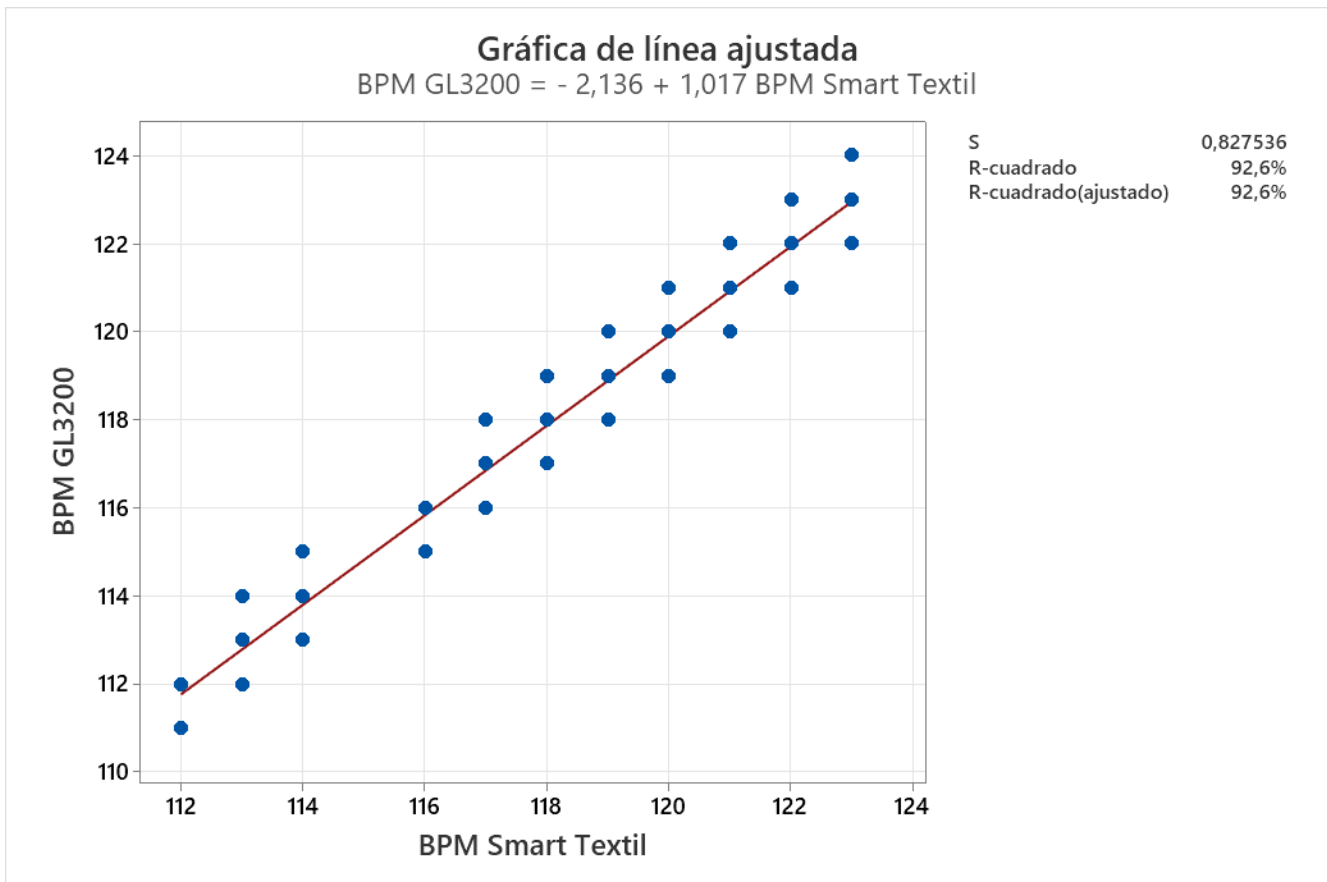
$$\text{BPM GL3200} = - 2,136 + 1,017 \text{ BPM Smart Textil}$$

Resumen del modelo

S	R-cuadrado	R-cuadrado(ajustado)
0,827536	92,64%	92,58%

Análisis de Varianza

Fuente	GL	SC	MC	F	P
Regresión	1	1069,22	1069,22	1561,32	0,000
Error	124	84,92	0,68		
Total	125	1154,13			



Análisis de regresión: SPO2 GL3200 vs. SPO2 Smart Textil

La ecuación de regresión es

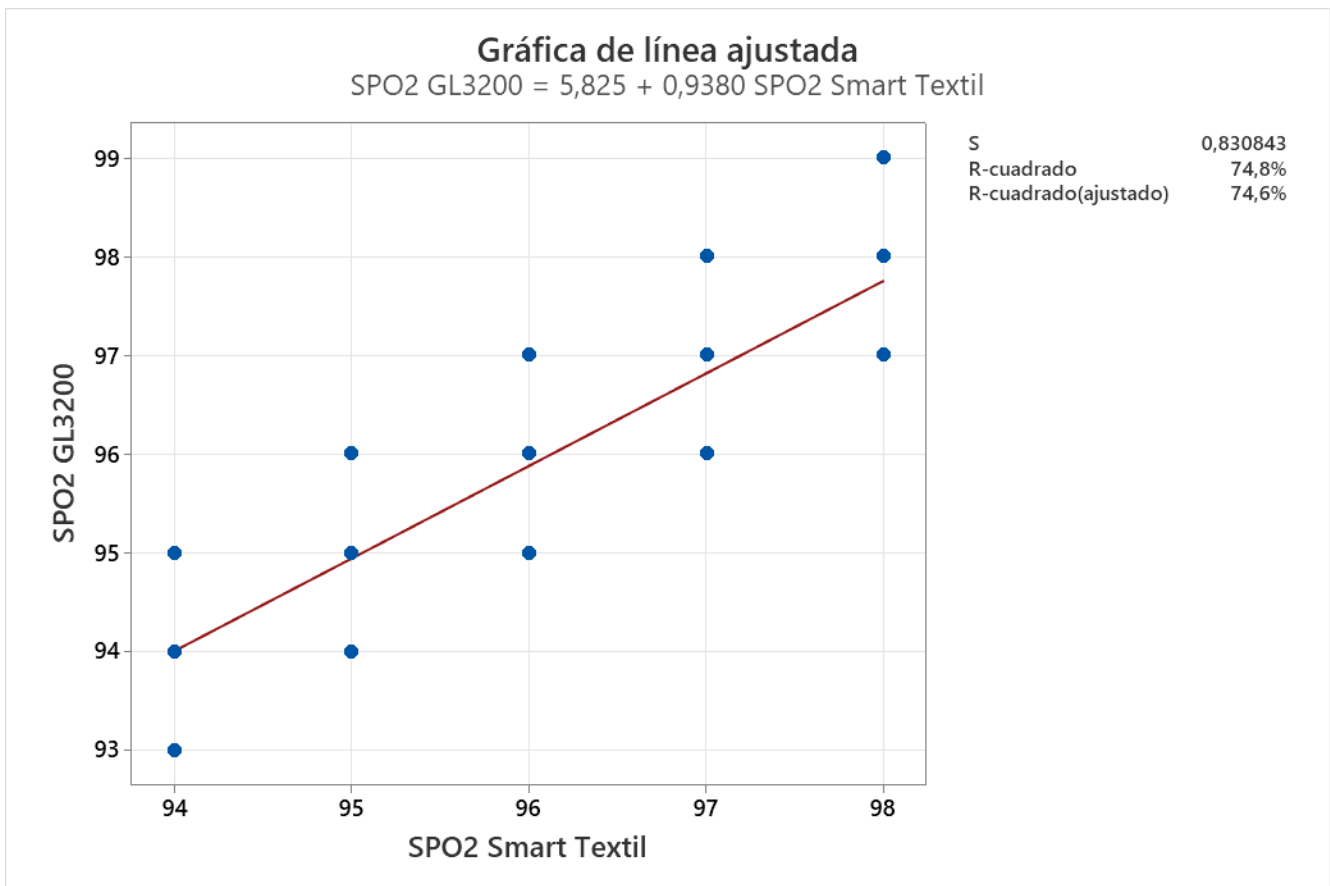
$$\text{SPO2 GL3200} = 5,825 + 0,9380 \text{ SPO2 Smart Textil}$$

Resumen del modelo

S	R-cuadrado	R-cuadrado(ajustado)
0,830843	74,80%	74,60%

Análisis de Varianza

Fuente	GL	SC	MC	F	P
Regresión	1	254,117	254,117	368,13	0,000
Error	124	85,597	0,690		
Total	125	339,714			



Anexo H: FOTOGRAFIAS DE PRUEBAS EN EL CENTRO DE ESTIMULACION TEMPRANA “NEUROPLACE”





**Anexo I: FOTOGRAFÍAS DE ENTREGA Y EXPLICACIÓN DEL USO
DISPOSITIVO**



CERTIFICACIÓN

Ingeniera Mg.

Pilar Urrutia

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

DECANA

Presente

Señora Decana:

Por medio de la presente yo, Dra. Jessica María Monge Segovia con cédula de identidad N° 0501847297 con registro del Senescyt N° 1027-2017-1806908 especialista en Pediatría, médico tratante del Hospital General de Latacunga certifico que he revisado el proyecto " SISTEMA DE MONITOREO NO INVASIVO Y SEGURIDAD PREVENTIVA DE FACTORES CAUSANTES DEL SÍNDROME DE MUERTE SÚBITA DE LOS LACTANTES DEL CENTRO DE ESTIMULACIÓN TEMPRANA Y PRENATAL "NEURO PLACE" A TRAVÉS DE SMART TEXTIL", el mismo que fue diseñado en su total autoría por el Sr. John David Lema Yungan con cédula de identidad N°1804352274 , estudiante de la carrera de Telecomunicaciones de la Universidad Técnica de Ambato.

Después de haber realizado las pruebas médicas necesarias me permito verificar la funcionalidad del equipo y puedo validar que los resultados obtenidos hasta la fecha son aceptables y de gran ayuda en el área de Pediatría; además, indico que datos de los sensores se visualizan dentro de los rangos médicos normales.

Es todo cuanto puedo certificar, pudiendo el portador de este documento hacer uso del mismo.



Dra. Jessica Monge Segovia

Médico Pediatra

C.I: 0501847297