



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

**SISTEMA DE TELEMEDICINA BASADO EN IOT PARA MONITOREO DE
PACIENTES CON ENFERMEDADES RESPIRATORIAS.**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Electrónica y Comunicaciones

ÁREA: Física y Electrónica

LÍNEA DE INVESTIGACIÓN: Sistemas Electrónicos

AUTOR: Andrés Josué Carrillo Molina

TUTOR: Ing. Carlos Diego Gordón Gallegos, PhD

Ambato - Ecuador

febrero – 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA DE TELEMEDICINA BASADO EN IOT PARA MONITOREO DE PACIENTES CON ENFERMEDADES RESPIRATORIAS, desarrollado bajo la modalidad Proyecto de Investigación por el señor Andrés Josué Carrillo Molina, estudiante de la Carrera de Ingeniería Electrónica y Comunicaciones, 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 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, febrero 2022.

Ing. Carlos Gordon, Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA DE TELEMEDICINA BASADO EN IOT PARA MONITOREO DE PACIENTES CON ENFERMEDADES RESPIRATORIAS es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2022.



Andrés Josué Carrillo Molina

C.C. 1805225610

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Andrés Josué Carrillo Molina, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Titulación, titulado SISTEMA DE TELEMEDICINA BASADO EN IOT PARA MONITOREO DE PACIENTES CON ENFERMEDADES RESPIRATORIAS, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, febrero 2022.

Ing. Pilar Urrutia, Mg.
PRESIDENTA DEL TRIBUNAL

Ing. Patricio Córdova
PROFESOR CALIFICADOR

Dr. Freddy Benalcázar
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

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

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

Ambato, febrero 2022.

A handwritten signature in blue ink that reads "Andrés CM" with a small flourish underneath.

Andrés Josué Carrillo Molina

C.C. 1805225610

AUTOR

DEDICATORIA

Dedico este trabajo de investigación a mi Madre por ser un ejemplo de fe y amor por Dios, lucha, valentía y esfuerzo constante, lo cual sirve como un pilar fundamental en mi crecimiento y búsqueda de superación.

Y principalmente a la fuente de todo cuanto necesité para esta travesía:

Romanos 11:36

“Porque de Él, y por Él, y para Él, son todas las cosas. A Él sea la gloria por los siglos.
Amén”

Andrés Josué Carrillo Molina

AGRADECIMIENTO

Doy gracias a Dios por su propósito en mi vida y todo cuanto a dispuesto en mi entorno conforme a Su voluntad.

A mi madre por toda su entrega, guía, por inculcarme todos los valores necesarios para crecer como persona y lograr todo lo que hasta este momento he alcanzado, se lo debo todo.

A mi familia por todo el apoyo y motivación constante que me ha servido para avanzar y superarme.

A mi tutor, Carlos Gordon quien con su guía y consejo ha permitido el desarrollo de este proyecto.

A los docentes y administrativos de la Facultad, quienes durante toda la carrera supieron compartir sus conocimientos a fin de formar profesionales y personas apropiadas.

Andrés Josué Carrillo Molina

ÍNDICE GENERAL

DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL	viii
ÍNDICE DE TABLAS	xi
ÍNDICE DE FIGURAS	xii
RESUMEN EJECUTIVO	xv
ABSTRACT	xvi
CAPITULO I	1
MARCO TEÓRICO	1
1.1 Tema de Investigación	1
1.2 Antecedentes Investigativos	1
1.2.1 Contextualización del problema.....	1
1.2.2 Estado del Arte	2
1.2.3 Fundamentación teórica	4
1.2.3.1 Tecnologías de la Información y Comunicación	4
1.2.3.2 Internet de las Cosas (IOT Internet Of Things)	5
1.2.3.3 Telemedicina	23
1.2.3.4 Métodos de Adquisición de Parámetros Vitales	28
1.2.3.5 Neumología	31
1.3 Objetivos	34
1.3.1 Objetivo General	34
1.3.2 Objetivos Específicos.....	35
CAPITULO II	37
METODOLOGÍA	37
2.1 Materiales.....	37
2.1.1 Selección de software y hardware	37
2.1.1.1 Capa de Sensorización	37
2.1.1.2 Capa de Intercambio de Datos	42
2.1.1.3 Aplicación IoT y Servicios de Nube	45
2.2 Métodos.....	46

2.2.1 Modalidad de la Investigación	46
2.2.2 Recolección de Información	47
2.2.3 Procesamiento y Análisis de Datos	47
2.2.4 Propuesta de Solución	47
2.2.5 Desarrollo del Proyecto	48
CAPITULO III.....	49
RESULTADOS Y DISCUSIÓN	49
3.1 Análisis y Discusión de Resultados	49
3.1.1 Desarrollo de la propuesta.....	49
3.1.1.1 Capa de Sensorización	49
3.1.1.2 Capa de Intercambio de Datos	50
3.1.1.3 Capa de Aplicación IoT	50
3.1.2 Implementación de circuito de monitoreo de signos vitales.	51
3.1.2.1 Adaptación de sensor MAX30100	51
3.1.2.2 Adaptación de sensor LM35	56
3.1.2.3 Adaptación de sensor MQ135	57
3.1.2.4 Programación de Microcontrolador ATMEGA 328p	59
3.1.2.5 Programación de Tarjeta ESP32	66
3.1.2.6 Diseño de circuito electrónico.....	69
3.1.2.7 Ubicación de sensores	71
3.1.2.8 Diseño de prototipo 3D	72
3.1.3 Implementación de aplicación IoT en plataforma ThingSpeak.	76
3.1.3.1 Plataforma ThingSpeak.....	76
3.1.3.2 Creación de Canales en ThingSpeak.....	77
3.1.3.3 Diseño de Interfaz en ThingSpeak	80
3.1.3.4 Adaptación de IFTTT – App MatLab Analysis y TimeControl	82
3.1.4 Implementación final del prototipo.....	89
3.1.5 Pruebas de funcionamiento	90
3.1.5.1 Prueba de Frecuencia Cardíaca	90
3.1.5.2 Prueba de Saturación de Oxígeno	94
3.1.5.3 Prueba de Temperatura Corporal	98
3.1.5.4 Prueba de tiempo de vida de batería	103
3.1.6 Resultados	104

3.1.6.1 Resultados de Lecturas de prototipo y Error Porcentual.....	104
3.1.6.2 Análisis de tiempo de vida de batería	105
3.1.7 Costos del prototipo	105
3.1.7.1 Costo de Hardware	105
3.1.7.2 Costo de Software	106
CAPITULO IV	107
CONCLUSIONES Y RECOMENDACIONES.....	107
4.1 Conclusiones	107
4.2 Recomendaciones.....	108
ANEXOS	113
ANEXO A: Manual de Operación del Sistema.....	113
ANEXO B: Vista 3D de componentes del prototipo	121
ANEXO C: Datasheet sensor MAX 30100.....	125
ANEXO D: Código fuente de archivo “MAX30100.h”	126
ANEXO E: Código fuente de archivo “MAX30100_BeatDetector.cpp”	128
ANEXO F: Código fuente de archivo “MAX30100_SpO2Detector.cpp”	131
ANEXO G: Datasheet sensor LM35	133
ANEXO H: Datasheet sensor MQ135	134
ANEXO I: Datasheet Microcontrolador ATmega328p.....	135
ANEXO J: Código fuente de Microcontrolador ATmega328p.....	137
ANEXO K: Datasheet Tarjeta ESP32.....	140
ANEXO L: Código fuente de Tarjeta ESP32	142
ANEXO M: Código fuente de MatLab Analysis en Plataforma ThingSpeak	144

ÍNDICE DE TABLAS

Tabla I. Estándares de IoT en desarrollo.....	8
Tabla II. Protocolo de Internet de las Cosas.	11
Tabla III. Características de datos de Plataforma IoT ThingSpeak.	16
Tabla IV. Comparativa de Plataformas Software IoT.....	19
Tabla V. Tabla comparativa de Hardware IoT.....	21
Tabla VI. Tabla de resumen de métodos no invasivos de detección de parámetros vitales.	30
Tabla VII. Valores de constantes vitales según grupo etario.	34
Tabla VIII. Tabla comparativa de selección de Microcontrolador.	37
Tabla IX. Tabla comparativa de selección de Sensor de Oximetría.	38
Tabla X. Tabla comparativa de selección de sensor de temperatura.	39
Tabla XI. Recursos Capa de Sensado	41
Tabla XII. Tabla comparativa de selección de dispositivo de intercambio de datos.	42
Tabla XIII. Recursos de Capa de Intercambio de Datos.....	45
Tabla XIV. Curva de gases del sensor MQ135.....	59
Tabla XV. Conexiones de circuito de sensorización.	60
Tabla XVI. Conexiones de ESP32.....	66
Tabla XVII. Características de diseño de placa de circuito impreso.\.....	70
Tabla XVIII. Muestras de prueba de Frecuencia Cardíaca.....	91
Tabla XIX. Muestras de prueba de Saturación de Oxígeno.....	94
Tabla XX. Muestras de prueba de Temperatura Corporal.	98
Tabla XXI. Consumo de Corriente de dispositivos.	103
Tabla XXII. Costos de Implementación – Hardware.....	106

ÍNDICE DE FIGURAS

Figura 1. Frameworks de IoT.....	6
Figura 2. Clasificación de APIs en función de su distribución.	10
Figura 3. Esquema de desarrollo ThingSpeak.....	18
Figura 4. Arquitectura típica de sensores wearables en IoT	20
Figura 5. Ventajas y Desventajas presentes en la Telemedicina.....	26
Figura 6. Aspectos de infraestructura tecnológica en Telemedicina.....	27
Figura 7. Resumen de aplicaciones de Telemedicina.	28
Figura 8. Diagrama de funcionamiento de Capa de sensorización.	41
Figura 9. Diagrama de funcionamiento de Capa de Intercambio de datos.	44
Figura 10. Diagrama de funcionamiento de Capa de aplicación IoT.....	46
Figura 11. Arquitectura de prototipo de sistema de telemedicina basado en IoT.	49
Figura 12. Ubicación de sensor en la yema del dedo.....	22
Figura 13. Funcionamiento de MAX30100 sobre el dedo.....	23
Figura 14. Grafica de los datos del sensor con oscilaciones y componente CC y CA.	52
Figura 15. Grafica de los datos del sensor luego de eliminar la componente en DC.	52
Figura 16. Grafica de señal aplicando filtro de media	53
Figura 17. Grafica de señal a la salida del filtro pasa bajo.	54
Figura 18. Tabla de relación para cálculo de SpO2.	55
Figura 19. Ubicación del sensor para temperatura axilar.....	56
Figura 20. Grafica temperatura vs voltaje.....	57
Figura 21. Diagrama de conexiones para programación.....	59
Figura 22. Diagrama de flujo de programación ATMEGA328p	65
Figura 23. Diagrama de conexiones ESP32.....	66
Figura 24. Diagrama de conexión de componentes del circuito eléctrico	70
Figura 25. Diseño de PCB del sistema.....	71
Figura 26. PCB final del prototipo.....	71
Figura 27. Ubicación de sensores en el cuerpo del paciente.....	72
Figura 28. Diseño 3D - Estructura de carcasa del prototipo.	73
Figura 29. Diseño 3D - Dimensiones de Carcasa de prototipo.....	73

Figura 30. Diseño 3D - Vista lateral 1 de la carcasa con sus aberturas.	74
Figura 31. Diseño 3D - Vista lateral 2 de la carcasa con sus aberturas.	74
Figura 32. Diseño 3D - Vista superior de bases para correas.	74
Figura 33. Diseño 3D - Vista de Tapa de carcasa.	75
Figura 34. Estructura final del prototipo.	75
Figura 35. ThingSpeak ingreso a plataforma	76
Figura 36. Índice de Plataforma ThingSpeak.....	76
Figura 37. Panel de creación de Canales en Plataforma ThingSpeak.	77
Figura 38. Configuración de Canal en ThingSpeak para el sistema de telemedicina.	78
Figura 39. Plataforma ThingSpeak, API Keys para la lectura o escritura de datos. .	79
Figura 40. Plataforma ThingSpeak modelos de sintaxis de pedidos de API.	79
Figura 41. Opciones de modificación de interfaz de presentación de Datos.	80
Figura 42. Opciones de Widgets para la interfaz de presentación de los Datos.	80
Figura 43. Interfaz de visualización de BPM y SpO2.....	81
Figura 44. Interfaz de visualización de Temperatura y Calidad del Aire.	82
Figura 45. IF THIS THEN THAT ventana principal.....	82
Figura 46. IFTTT - Creación de Applet.....	83
Figura 47. IFTTT - Selección de servicio Webhooks	83
Figura 48. IFTTT - Descripción de Trigger Webhooks.....	84
Figura 49. IFTTT - Configuración de Trigger.	84
Figura 50. IFTTT - Configuración de acción "Then That".	85
Figura 51. IFTTT - Selección de Servicio Google Sheets.	85
Figura 52. IFTTT - Configuración de celdas de Google Sheets.	85
Figura 53. IFTTT - URL de evento configurado en el servicio Webhooks.....	86
Figura 54. IFTTT - Evento configurado y funcionando.	86
Figura 55. ThingSpeak - Creación de Apps.....	86
Figura 56. ThingSpeak - Configuración de App Analysis MATLAB.....	87
Figura 57. ThingSpeak - Escritura de código de para Analysis MATLAB.....	87
Figura 58. ThingSpeak - Configuración de App Time Control.	87
Figura 59. ThingSpeak - Verificación de App Time Control.	88
Figura 60. Google Drive - Comprobación de carpeta generada por servicio IFTTT.	88

Figura 61. Google Drive - Google Drive - Comprobación de archivo generado por servicio IFTTT.	88
Figura 62. Google Drive Sheets - Google Drive - Comprobación de Datos generados por servicio IFTTT.	89
Figura 63. Implementación del Prototipo Final.	89
Figura 64. Implementación de Prototipo - Lectura de Frecuencia Cardíaca.....	90
Figura 65. ThingSpeak - Lecturas de Frecuencia Cardíaca.	93
Figura 66. Implementación de prototipo - Lectura de Saturación de Oxígeno.....	94
Figura 67. ThingSpeak - Lecturas de Saturación de Oxígeno.	97
Figura 68. Implementación de prototipo - Lectura de Temperatura.	98
Figura 69. ThingSpeak - Lecturas de Temperatura Corporal.	101
Figura 70. Pruebas de funcionamiento del prototipo.	102
Figura 71. Pruebas de funcionamiento del prototipo pacientes con enfermedades asociadas.	102
Figura 72. Pruebas de funcionamiento del prototipo en pacientes.....	103

RESUMEN EJECUTIVO

El presente proyecto de investigación describe el desarrollo de un sistema de telemedicina basado en “Internet de las Cosas”, diseñado para contar con un monitoreo de frecuencia cardíaca, saturación de oxígeno y temperatura; constantes vitales cuyo monitoreo es fundamental en pacientes con enfermedades respiratorias. El sistema de telemedicina trabaja con Computación en la Nube, lo que permite tanto una visualización local como adquisición y almacenamiento de datos de los sensores biomédicos wearables en una plataforma “Internet de las Cosas”, lo que es ampliamente útil para el monitoreo y para la interpretación de los resultados por parte de un médico especializado, evitando de esta manera complicaciones y riesgos en el paciente.

La actual situación de pandemia ha ocasionado que la atención médica en los servicios de salud se vea afectada, lo que dificulta el acceso al control de pacientes ya sea por riesgo de contagio o dificultad de movilización. Esto permite el uso de dispositivos locales de lectura de constantes vitales, pero con una interpretación de datos casi nula, de esta manera se introduce el concepto de telemedicina, que relacionado con el “Internet de las Cosas” y sensores biomédicos wearables permite cubrir esta necesidad actual.

El proyecto posee un impacto en el ámbito de la Biomedicina y la Telemedicina, pues al brindar una tecnología con un sistema adaptable y wearable, con visualización local e interfaz de monitoreo en plataforma ThingSpeak, permite ser implementada de acuerdo a las necesidades del paciente como las del médico, con mediciones dentro de un rango de error aceptable en prototipos de sistemas de telemedicina con sensores biomédicos.

Palabras clave: Telemedicina, IoT, sensores biomédicos, enfermedad respiratoria.

ABSTRACT

This research project describes the development of a Telemedicine system based on the “Internet of Things”, designed to monitor heart rate, oxygen saturation and temperatura; vital signs that mainly in patients with with respiratory diseases it necessary to monitored. The telemedicine system works with Cloud Computing which allows local visualization, acquisition and storage of data from wearable biomedical sensors in an “Internet of Things” platform, which is widely useful for monitoring and interpretation of the results by a specialized doctor, thus avoiding complications and risks for the patient.

The current pandemic situation has caused that medical attention in health services is affected, which makes it difficult to access patient’s checkups, either due to the risk of catching the virus or difficulty commuting. This allows the use of local sign reading devices, but as an almost null interpretation of data, in this way the concept of Telemedicine is introduced, which in relation with the “Internet of Things” and wearable biomedical sensors allows to cover this current need.

The project has an impact in the field of Biomedicine and Telemedicine, because by providing technology with adaptable system and wearable, with local visualization and monitoring interface on the ThingSpeak platform, it can be implemented according to the needs of the patient and also the doctor, with an acceptable error range in the measurements in prototypes of Telemedicina systems with their biomedical sensors.

Keywords: Telemedicine, IoT, biomedical sensors, respiratory disease.

CAPITULO I

MARCO TEÓRICO

1.1 Tema de Investigación

Sistema de Telemedicina basado en IoT para monitoreo de pacientes con enfermedades respiratorias.

1.2 Antecedentes Investigativos

1.2.1 Contextualización del problema

La contaminación ambiental es una problemática de salud en Ecuador. El desarrollo de las ciudades principales ha llevado a un desordenado crecimiento industrial y sobrepoblación vehicular en el Ecuador razón por la cual el ser humano está constantemente expuesto a contaminación ambiental. El aire es un elemento indispensable para la vida y el hombre está expuesto a una masa de 15Kg de aire al día, lo que está llevando a que la contaminación sea llevada al cuerpo de los seres humanos. Según el Control de Calidad Ambiental de la capital del Ecuador el contaminante atmosférico mayor es el monóxido de carbono que constituye un 71% de carga contaminante. Según el Ministerio de Salud Pública del Ecuador existen dos periodos a lo largo del año en donde se registra mayor cantidad de problemas respiratorios en los ciudadanos y un dato preocupante es el que un 10% de las personas que adquieren el virus de la Influenza podrían presentar dificultades más graves como una Infección Respiratoria Aguda Grave. El presentar dificultad de salud en la actualidad conlleva un problema mayor debido a la saturación de los servicios de Salud del MSP en Ecuador, lo que dificulta la atención médica a pacientes con afecciones médicas [1] [2].

A lo largo de muchos años se ha intentado utilizar sensores para el monitoreo de pacientes, sin embargo, se presentan inconvenientes y limitación en cuanto a portabilidad de los sistemas, configuración de los dispositivos y movilidad del usuario para el monitoreo. En medio de situaciones conflictivas sociales, se busca la

implementación de soluciones accesibles tanto para el paciente como el médico, mejorar el servicio médico por medio de tele monitoreo, el uso de hardware sencillo y económico para la adquisición de parámetros vitales, todo esto permitirá que los datos de parámetros vitales sean accesibles para el médico de forma que se tenga un control adecuado de la salud de manera remota [3].

Actualmente dispositivos para la medición de constantes vitales, específicamente en pacientes con enfermedades respiratorias, como es el caso de la saturación de oxígeno en la sangre, están disponibles en el mercado, pero estos modelos no cuentan con la capacidad de tele monitoreo médico directo, los oxímetros actuales requieren de personal de asistencia y actúan de manera local sin interpretación de los resultados [4].

La propuesta se basa en el uso de tecnologías de comunicación a distancia y protocolos para el Internet de las Cosas (IOT Internet of Things), brindar una alternativa de monitoreo para las personas con enfermedades respiratorias, con el uso de sensores en el campo IoT de forma que se automatice la adquisición de datos y el envío de manera remota empleando comunicación inalámbrica a la estación del médico tratante para el monitoreo y evaluación de los datos.

1.2.2 Estado del Arte

El desarrollo del proyecto está basado en electrónica y las telecomunicaciones, busca la integración de un prototipo electrónico con la tecnología IoT para alcanzar un sistema de Telemedicina. De esta manera el proyecto de investigación “Sistema de Telemedicina basado en IoT para el monitoreo de pacientes con enfermedades respiratorias” al estar basado en una tecnología altamente estudiada a nivel mundial se estudiado investigaciones que aporten en el desarrollo del tema propuesto.

Jaime Calderón de la Universidad de Ciencias y Humanidades de Lima-Perú en su Tesis “Implementación de un oxímetro de pulsos para monitorizar desaturación del paciente a distancia” muestra que se puede obtener un oxímetro inalámbrico que permite acceder a los datos por medio de internet; en su desarrollo se identificó inconvenientes tales como el acceso a una Ip publica por parte del paciente, además el uso de diodos emisores encapsulados para mayor sensibilidad de la señal; finalmente es necesario una etapa de amplificación de la señal para que las lecturas sean óptimas

[5]. Este proyecto aporta en cuanto a la programación de un microprocesador para capturar los datos adquiridos por los sensores en este caso el oxímetro compuesto por diodos, lecturas que serán enviadas por parte de un módulo wifi por conexión TCP/IP para su análisis en una presentación HTML.

David Álvarez de la Universidad Tecnológica del Perú en su Tesis “Diseño de un prototipo de monitoreo oximétrico Adulto-Pediátrico hospitalario” muestra en su trabajo que el manejo de sensores hospitalarios de monitoreo es una opción óptima para el censado de datos, puesto que la etapa de acondicionamiento de la señal es externa al dispositivo, lo que permitiría un mejor procesamiento de las señales tomadas. Por otro lado, menciona que es de importancia la presentación de datos, pero este dispositivo maneja una presentación por medio de una pantalla LCD, lo que limita la visualización de los datos solo al paciente [6]. Este proyecto aporta un análisis del método de medición de saturación de oxígeno en la sangre, basado en la técnica de pulsioximetría que permite la adquisición de datos de frecuencia cardíaca y oxigenación de la sangre, lecturas que se adquieren por medio del pulso empleado diodos emisores para captar señales emitidas que no son absorbidas por el cuerpo; es decir, se basa en pulsos de luz a través de la piel.

Luis Ruiz de la Facultad de Ingeniería en Sistemas Electrónica e Industrial (FISEI) de la Universidad Técnica de Ambato en su tesis “Sistema de Telemedicina para monitoreo continuo de constantes vitales en lactantes menores para evitar el síndrome de muerte súbita” presenta en su trabajo un prototipo de dispositivo de medición de constantes vitales con visualización en una aplicación de Smartphone, para esto se basa en el envío de tramas con los datos recibidos por el sensor y estos datos son presentados en una interfaz amigable para el usuario; por otro lado la sensibilidad en este tipo de dispositivos juega un papel importante y Luis Ruiz en su investigación logró obtener un rango de error absoluto de 0,55 % teniendo una confiabilidad de 99,43% del prototipo [7]. Este proyecto de investigación aporta el manejo de redes de área corporal para el monitoreo de constantes vitales, redes que están constituidas por nodos y que su método de transmisión es inalámbrico; las tecnologías que se enmarcan en las redes WBAN y la búsqueda de la más óptima en el ámbito de la telemedicina como fuente de adquisición de datos y manejo hacia el usuario final.

José Vega, Mario Lagos, Francisco Sánchez, José Cosme, del Instituto Nacional de México en Celaya en su artículo “Monitoreo de Signos Vitales usando IoT”, presentan una opción para la presentación de información muy optima basado en el uso de IoT, para esto el dispositivo de medición de signos emplean sensores hospitalarios que integra una tarjeta PyBoard, en cuanto a la presentación de los datos se basa en una plataforma ThingSpeak que provee el servicio de análisis de información en la nube, recolectando los datos de sensores instalados en dispositivos inteligentes y generando un reporte en MatLab para analizar y presentar remotamente sobre aplicaciones web, móviles y redes sociales [8]. Esta investigación aporta al trabajo una plataforma IoT para el uso de servicio de almacenamiento de datos en la nube y el servicio de mensajería privada. La plataforma ThingSpeak se puede combinar con el código de MatLab que permitirá analizar un registro de los datos adquiridos por hardware IoT. Por otro lado, brinda un análisis de las distancias de alcance de la transmisión por WiFi del dispositivo hacia el punto de acceso, lo que permitirá la ubicación adecuada del prototipo, dispositivos intermediarios y la estación del médico tratante.

1.2.3 Fundamentación teórica

1.2.3.1 Tecnologías de la Información y Comunicación

Las Tecnologías de la Información y Comunicaciones son un tipo de las tecnologías que son necesarias para el procesamiento y gestión de la información, aún más de una manera particular en el uso de ordenadores y software que brinda la oportunidad de crear, modificar, almacenar, proteger y recuperar la información. Uno de los elementos más representativos de las nuevas tecnologías es el Internet, pues supone un gran salto cualitativo en el manejo de la información y su relación con el usuario [9].

Algunas de las características fundamentales de las TIC son:

- Inmaterialidad, su materia prima es la información en múltiples códigos y formas.
- Interconexión, aunque se presentan de forma independiente, pueden combinarse y ampliar sus conexiones.
- Interactividad, permite la interacción del sujeto con la máquina logrando la adaptación de diversas áreas educativas y cognoscitivas de las personas.

- Instantaneidad, facilita la rapidez de acceso e intercambio de la información.
- Calidad de imagen y/o sonido, lo que da fiabilidad y fidelidad a la información transferida.
- Penetración en diversos sectores de la sociedad: salud, educación, economía, entre otros.

Sistema de Telecomunicación

Las telecomunicaciones son una rama de ingeniería que maneja la generación, transmisión, recepción y procesado de señales sean estos por medios ópticos, electromagnéticos, eléctricos, entre otros; de ahí que los sistemas de telecomunicaciones están basados en técnicas y dispositivos adecuados para el transporte de información entre una fuente y uno o varios usuarios finales [10].

1.2.3.2 Internet de las Cosas (IOT Internet Of Things)

Introducción

En 1999, Kevin Ashton describe que un objeto del mundo real se puede conectar mediante el internet mediante sensores. Basa sus conocimientos en que no solo el internet puede actuar en el computador sino más bien puede aplicarse en otros objetos; basado en esto varios sectores son beneficiados, por ejemplo, automotrices, salud, industria y hogar. En un futuro se pronostica que el internet tomara otro sentido, pues gracias a los avances se evidenciara cambios importantes y relevantes [11].

El Internet de las Cosas (Internet Of Things) o conocido por sus siglas como IOT, ha tenido cambios impresionantes a través del tiempo. Considerado por muchos como una maravilla tecnológica pues gracias a ellos se ha visto grandes avances en la inteligencia artificial, nuevos avances en nano y biotecnología, así como ciencias de la salud y otras ramas. Como se mencionó anteriormente la evolución de la IOT no solo abarca el aspecto de ordenadores o teléfonos inteligentes, sino más bien también abarcan otros objetos como ropas, pulseras, gafas, electrodomésticos, automóviles entre otros. Una aportación esencial de la IOT es que es capaz de conectar personas con objetos e incluso animales con objetos la IOT es una base de datos que recaba información de manera constante [12].

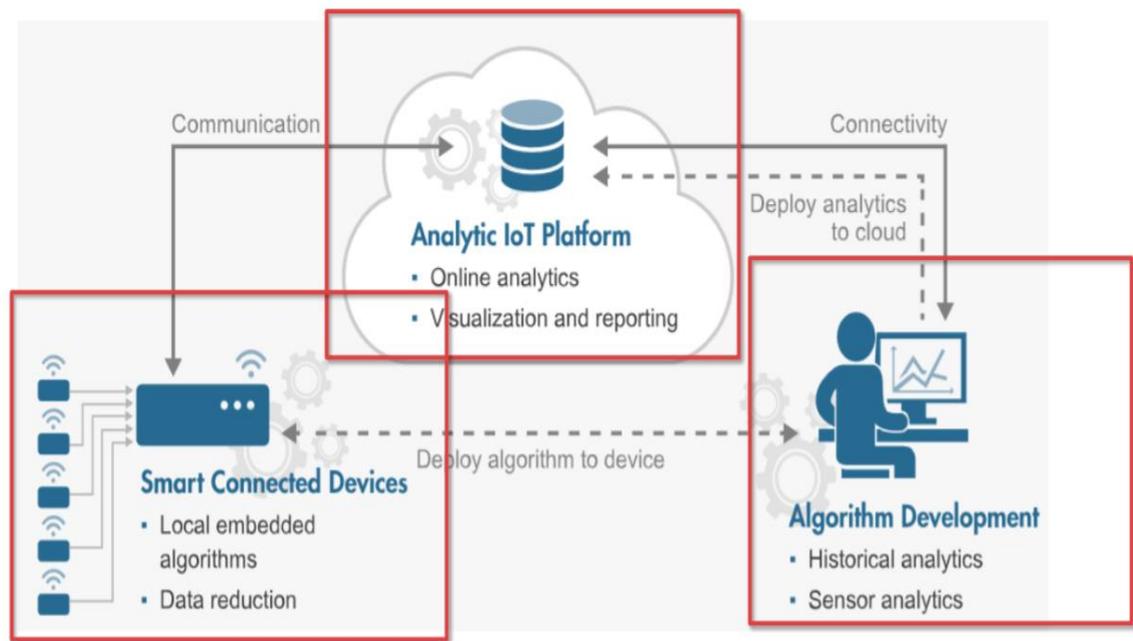


Figura 1. Frameworks de IoT.
Fuente: [13].

Una participación activa tanto de máquinas como de hombres es de gran utilidad para el desarrollo de la sociedad. Pues ambas trabajan de manera activa mediante internet específicamente de la utilización de redes y tecnologías inalámbricas que se encuentren a su disposición. La IOT permite que los objetos sean capaces de controlar, monitorear las situaciones que se puedan presentar en una aplicación. Es decir, es capaz de un funcionamiento ideal sin la participación del ser humano. Para llegar a cumplir todo esto es importante recabar, almacenar la mayor información posible que se pueda extraer día con día [14].

Tipos

Los tipos de IOT se clasifican en cuatro apartados, el primero está destinado a los objetos simples considerados como primitivos, pero aun así son capaces de conectarse a las interacciones. Los segundos son los sensitivos o también conocidos como los comunicativos, mediante sus sensores pueden ser capaces de comunicar y brindar la información necesaria. Por otra parte, los objetos controlables, estos son más avanzados pues son capaces de detectar, controlar y recibir información. Finalmente, los objetos autónomos, estos son los más sofisticados y actualizados [15] [16].

Arquitectura del Internet de las Cosas (IOT Internet Of Things)

Existen maneras diferentes por las cuales se pueden reconocer una arquitectura IoT, sin embargo, se plantea una estructura básica como base de cada implementación, se puede describir la arquitectura IoT en cuatro capas:

- Hardware o Dispositivos IoT

Un dispositivo para que se considere como hardware IoT debe cumplir con especificaciones adecuadas para que integre las funciones de IoT. Un dispositivo IoT es capaz de comunicarse por diferentes protocolos de comunicación a otros dispositivos, a internet o bluetooth. Un aspecto fundamental destaca que posea sensores o actuadores, o a su vez puertos de entrada y salida, ya que por estos sensores se recopila información o datos estáticos o dinámicos del mundo físico. Estos dispositivos deben contar con un procesador que permita la captura y procesamiento de los datos. Dentro de estos dispositivos contamos con Arduino, RaspberryPi, Pinnocio entre otros [17].

- Redes o Intercambio de Datos

La capa de Red de comunicación o intercambio de datos es comparable con internet, misma que está compuesta con diferentes capas, sean estas: física, enlace, red, transporte y aplicación, por otro lado, posee protocolos de comunicación que operan en diferentes capas. Esta capa llega a ser la encargada de intercomunicar los datos adquiridos en la capa de hardware IoT y la Capa de Nube, de esta manera se cuenta con un proceso de adquisición de datos, procesamiento y almacenamiento de los datos [17] [18].

- Nube o Servidor

Esta capa se consolida como la capa de borde del sistema IoT puesto que es el punto de almacenamiento de datos provenientes de la capa de hardware y transmitidos por la capa de intercambio de datos, esta capa nube permite la comunicación con otros servicios de nube sean privados públicos de forma que se pueda analizar y visualizar por medio de la Aplicación IoT [17] [18].

- Aplicación IoT

En la capa de aplicación se tiene el procesamiento, extracción y análisis de los datos que se han almacenado en la capa de Nube o Servidor. La aplicación IoT es fundamental en el sistema puesto que extrae los datos, para su presentación en una interfaz que integre el usuario con los datos adquiridos para su análisis por medio de gráficas, texto, gadgets, entre otros [17].

Estándares IoT

El estado actual de la estandarización de IoT es lejano respecto a establecer un único estándar universal, los análisis respecto a los estándares se llevan en cuanto a si son o no necesarios, considerando que los desarrolladores llegan a considerar que son centrales, pero eventualmente los dispositivos hardware de IoT pueden conectarse y transmitir entre sí. Es necesario considerar que el Internet de las Cosas requiere tecnología diversa para su funcionamiento que va desde comunicaciones móviles, a seguridad de información, intercomunicaciones e interoperabilidad con más dispositivos, y es aquí en donde se considera que un solo estándar no lograría cubrir todo el funcionamiento de la IoT.

Como consecuencia de esto se puede considerar que existen cuatro etapas o capas en las que se está desarrollando los estándares de IoT:

Tabla I. Estándares de IoT en desarrollo.
Fuente: [18].

CAPAS	DESCRIPCIÓN	DESARROLLADORES
Aplicación	Mira los protocolos para desarrollo de aplicaciones IoT	IETF, OASIS, OMA, W3C
Servicios	Se desarrollan los marcos que permiten los servicios de IoT	M2M, OIX, AllSeen

Redes	Se presta atención a las optimización que apoyan IoT	-
Tecnologías de Acceso	Busca optimizar las capas de aplicación y servicios para el uso de servicios de IoT y optimización específicas de acceso a redes.	3GPP, IEEE, Bluetooth SIG, Weightless SIG

Interfaz de Programación de Aplicaciones API

Las API Application Programming Interface son un conjunto de funcionalidades y protocolos que se emplea para la integración y desarrollo de software de las aplicaciones, aporta por otro lado un servicio de software de manera que pueda ser empleado por un distinto software para la optimización de resultados. Generalmente no tienen un resultado sobre sí mismo, sino que cumplen como enlace entre un software existente con otro que requiere integrar algún servicio, esto se conoce como “Software to Software”. Las API pueden estar distribuidas hacia ciertos sectores, estos pueden ser Internos, externos o partner [19].

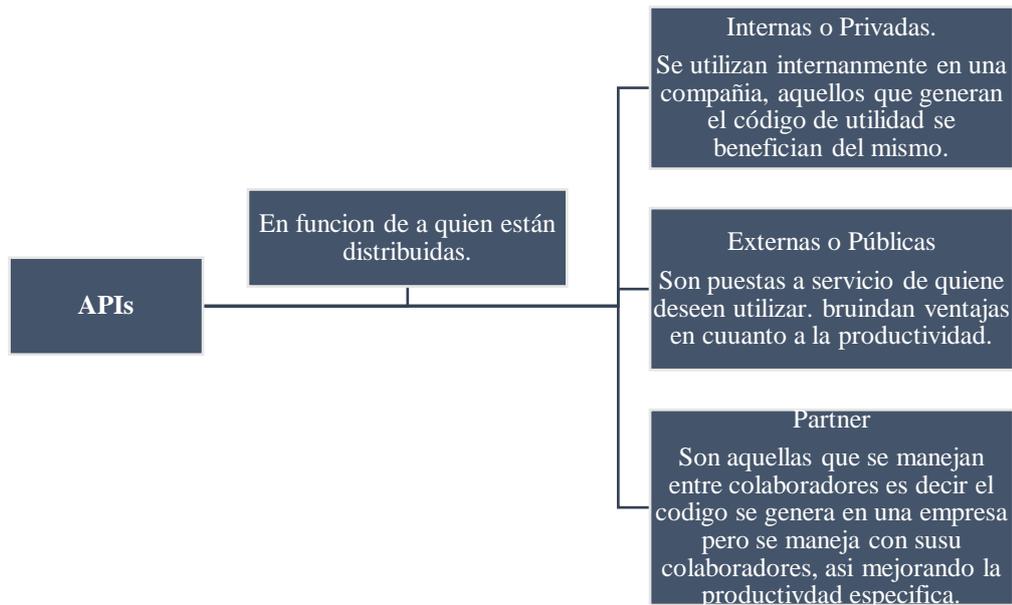


Figura 2. Clasificación de APIs en función de su distribución.
Fuente [19].

Además, podemos mencionar que las APIs se pueden también describir en función de orientación.

- APIs de servicios Web, proporcionan acceso a el servicio en base a una dirección web e implicación de comunicación de una red. REST, SOAP, XML-RPC, JSON-RPC.
- APIs basadas en librerías, son de importancia al manejar distintos tipos de lenguajes de programación, puesto que aquellas APIs brindan el manejo de librerías que proporcionan las plataformas software.
- APIs de sistemas operativos, son un tipo de API que permite el análisis de la estructura que poseen las funciones del Sistema Operativo que las requiere o implementa.

Las API son un medio que optimiza el conectar una infraestructura propia por medio de aplicaciones nativas de la nube, además permite compartir datos de usuarios externos con clientes, podemos decir que permiten el acceso de usuarios a recursos o datos manteniendo seguridad y control de los mismos [19].

Protocolos y Seguridad IoT

El modelo asociado a la conectividad y navegabilidad del Hardware IoT se ajustan al modelo de protocolos TCP/IP, sobre este modelo conviene emplear protocolos para la transferencia de datos o información según las características que posean los procesadores de los dispositivos de Hardware IoT.

Existen protocolos dedicados para Internet de las Cosas como el HTTP, REST, WebSocket y XMPP, los cuales están sustentados en los protocolos TCP o UDP, en la Tabla #, se describe los protocolos en base al modelo de interconexión de sistemas abiertos (OSI).

Tabla II. Protocolo de Internet de las Cosas.

Fuente [20].

NIVEL		PROTOCOLO	
Nivel de Aplicación	Este nivel actúa como una interfaz entre el usuario y los dispositivos IoT con un protocolo IoT.	Advanced Message Queuing Protocol AMQP	Este protocolo crea interoperabilidad entre el middleware de mensajería. Ayuda a que varios sistemas de mensajería funcionen juntos.
		Protocolo de Aplicación Restringida CoAP	Este protocolo de red posee un ancho de banda limitado orientado a dispositivos con capacidad limitada.
		Message Queue Telemetry Transport MQTT	Este protocolo de mensajería está diseñado para comunicación ligera entre dispositivos o equipos, es usado principalmente para conexiones con poco ancho

			<p>de banda con ubicaciones remotas.</p> <p>Usa un patrón de publicación-suscripción y es ideal</p>
		<p>HYPertext TRANSFER PROTOCOL HTTP</p>	<p>Protocolo de tipo cliente servidor de código abierto, efectivo para el envío de grandes cantidades de información, como lecturas de sensores, se recomienda asegurar por protocolos SSL/TLS sobre HTTP.</p>
<p>Nivel de Transporte</p>	<p>Este nivel habilita y da protección a la comunicación de los datos entre los puntos de comunicación.</p>	<p>Protocolo de Control de Transmisión TCP</p>	<p>Este protocolo ofrece comunicación entre hosts, de forma que divide los datos en paquetes individuales que envía, reenvía y los ensambla según sea necesario.</p>
		<p>Protocolo de Datagramas de Usuario UDP</p>	<p>Protocolo que permite comunicación entre procesos y estos se ejecutan sobre IP, en cuanto a la transferencia de datos sobre TCP mejora la velocidad.</p>
<p>Nivel de Red</p>	<p>Nivel que permite la comunicación entre dispositivos y el enrutador.</p>	<p>IP</p>	<p>La mayoría de protocolos IoT trabajan sobre IPv4 y más actualizados sobre IPv6.</p>

		6LoWPAN	Protocolo para mejorar procesamiento en dispositivos con baja potencia.
Nivel Físico	Nivel que funciona como canal de comunicación entre dispositivos del entorno de la red.	Bluetooth Low Energy BLE	Reducción en cuanto a consumo de energía y costo. Distancia de conectividad similar a Bluetooth clásico.
		Evolución a Largo Plazo LTE	Estándar para comunicación inalámbrica de banda ancha en dispositivos móviles celulares, existe aumento en cuanto a velocidad y capacidad de las redes celulares.
		WiFi 802.11	Requiere un consumo energético ininterrumpido, alcance con buena distancia y habitual.
		Zigbee	IEEE 802.15.4 conjunto de protocolos para comunicaciones de alto nivel, usados en redes de área personal.

Seguridad en Protocolo HTTP

El protocolo Hypertext Transfer Protocol HTTP por sí mismo no implementa seguridad, por esta razón usa Transport Layer Security TLS que corresponde a un protocolo de seguridad usado en la capa de transporte, de esta manera el protocolo HTTPS Hypertext Transfer Protocol Secure define la manera en que se transporta HTTP sobre un canal seguro TLS.

Con el uso de HTTPS o HTTP sobre TLS se brinda características de seguridad como:

- Integridad: Las modificaciones de los datos enviados por el canal seguro serán detectables luego de la implementación.
- Autenticación: Se realiza un proceso en que se autentica al servidor y al cliente.
- Confidencialidad: Los datos que se transmiten por el canal seguro establecido son solo visibles en los extremos de la comunicación.

Este protocolo se basa en la aplicación y manejo conjunto de criptografía simétrica, o asimétrica, certificados digitales y firmas digitales para de esta manera conseguir un medio seguro o canal certificado para la comunicación a través de internet [21].

Seguridad en Prototipo

Para la prevención de ataques o amenazas dentro de los sistemas IoT se implementan medidas de prevención como:

- Autenticación, control de acceso y administración

Se tiene consideraciones en cuanto a contraseñas predeterminadas. Para el acceso a la configuración del dispositivo es necesario la implementación de credenciales de acceso. Esta característica de seguridad se desarrolla en la programación del prototipo con el archivo *"secrets.h"*, es decir se cuenta con un archivo de certificados y credenciales de acceso. Archivo que posee credenciales de acceso a la Red WiFi y credenciales de la Plataforma *"API Key"* y *"Chanell ID"* esto para la comunicación Hardware y Plataforma IoT. Estos datos se describen en la programación del prototipo.

- Protocolos seguros

En este aspecto se debe tener consideraciones respecto a los protocolos a utilizar en las comunicaciones entre dispositivos.

- Protección de datos almacenados

La plataforma que controle los datos debe considerar acceso bajo credenciales. Se debe controlar los datos enviados y almacenados en la plataforma y en los dispositivos, la

plataforma posee datos de Login para el acceso a la misma y además maneja un tipo de visualización *Private Visualization* que permite que los datos no sean públicos.

- Plan de continuidad del servicio

Un dispositivo que trabaja de forma autónoma sin supervisión alguna, durante un fallo en la red o sistema, debe contar con un registro de logs o últimos datos enviados, de forma que se pueda analizar el instante del fallo [20].

Plataformas IoT

En la actualidad las plataformas más utilizadas son las presentadas por Google, Microsoft, Amazon y de Software Libre, las mismas que serán detalladas a continuación:

- ***Plataforma IOT AWS de Amazon***

Estas plataformas están diseñadas mediante sensores que ayudan a la conexión de varios objetos. Para que esta aplicación sea un éxito la empresa se ha asociado con Intel, Qualcomm, Texas Instrument, que son las más reconocidas en fabricación de Hardware, para lo cual ellos se especializan en crear productos que sean compatibles con su plataforma [22].

- ***Plataforma IOT Azure de Microsoft***

Esta plataforma trabaja conjuntamente con SaaS, PaaS y IaaS. Azure es una plataforma inteligente, que permite la expansión de los dispositivos, basándose en información obtenidas del mundo real. En si la plataforma permite a los administradores a realizar su trabajo de una manera más sencilla y eficiente, ya que ayuda a la visualización de la infraestructura operativa que maneja [22].

- ***Plataforma IOT de Google***

La plataforma IoT de Google es aquella encargada en dar alerta de si los equipos requieren mantenimiento o no, además de que ayuda a optimizar el rendimiento, e incluso es capaz de ayudar a la detección a tiempo de anomalías o errores que puedan existir. Dicha plataforma puede ser usada por cualquier tipo de empresa independiente

de su tamaño [23]. Una plataforma IoT consta principalmente de ocho componentes, el primero está relacionado con la conectividad y la normalización, el segundo es la gestión de cada uno de los dispositivos, como tercero se encuentra las bases de datos, seguidamente de los procesamientos y gestión, por otra parte, se encuentra lo analítico y la visualización, finalmente las herramientas y las interfaces externas [24].

Plataforma ThingSpeak

ThingSpeak es una plataforma abierta de aplicaciones, diseñada para permitir conectar personas con objetos. Se caracteriza por ser una plataforma Open Source con una API para almacenar y recuperar datos de los objetos usando el protocolo HTTP sobre Internet o vía LAN. Se trata de una plataforma basada en Ruby on Rails 3.0, este es un framework de aplicaciones web de código abierto basado en Ruby, cuya arquitectura está basada en el Modelo Vista Controlador. ‘

Se caracteriza por su simplicidad a la hora de programar aplicaciones del mundo real, escribiendo menos código y con una configuración mucho más sencilla que otros Frameworks. Otra de las características que hacen de RoR un Framework perfecto para el desarrollo de aplicaciones es que permite el uso de meta programación, haciendo que su sintaxis sea más legible y llegue a un gran número de usuarios [25].

Tabla III. Características de datos de Plataforma IoT ThingSpeak.

Fuente: [26].

Característica	Valor
Costo	Gratis y Pagada
Cantidad de Mensajes	Limitados
Vinculación con otras aplicaciones	Si
Procesamiento de la Plataforma	Ilimitado
Dispositivos	Limitados
Programación	Interna en la aplicación y en dispositivos
Lenguajes de Programación en Uso	MatLab / Java

Características de Diseño

API

ThingSpeak dispone de una API contenida en GitHub, que para desarrolladores es un mecanismo de suma importancia para el desarrollo de aplicaciones, en este caso es código abierto de manera que se pueda montar un servidor propio [25].

Canales

Esta plataforma da la facilidad de almacenar y publicar datos por medio de Canales “Channels”, su creación no es compleja, sino que más bien facilita su desarrollo. Los canales pueden ser Públicos o Privados según los requerimientos de los desarrolladores [25].

Plugins

La plataforma ThingSpeak para aumentar la funcionalidad de las aplicaciones permite el desarrollo de plugins, Soporta HTML, CSS y JavaScript. Posee graficas como Google Gauge Visualization, que posee un amplio nivel de personalización [25].

Integración

Un factor importante en cualquier plataforma IoT es que posea una amplia integración con hardware y software, en este caso ThingSpeak permite la integración con: Arduino, Raspberry Pi, IoBridge, Electric Imp, App Web, Redes Sociales, Análisis de datos con MatLab [25].

Apps en ThingSpeak

ThingTweet, es una app que actúa como proxy que permite el envío de actualizaciones de estado a Twitter, para esto se necesita una API de ThingTweet.

- TweetControl, esta app permite escuchar los hashtags de Twitter para controlar demás cosas, se consigue conectando a través de ThingHTTP.
- ThingHTTP, por medio de esta app se puede permitir un microcontrolador o dispositivo de bajo nivel enlazarse a un servicio web por internet o una red

local. Soporta métodos como GET, POST, PUT, DELETE y autenticación básica.

- React, una app que permite ejecutar una acción al cumplirse cierta condición para el manejo de datos en el canal de ThingSpeak.
- TallBackEsta app permite que cualquier dispositivo actúe sobre los comandos en cola.
- TimeControl, app que controla a una determinada hora podremos ejecutar un ThingHTTP o un ThingTweet [25].

Fases de desarrollo

Para la creación de una plataforma, lo ideal es seguir un esquema que permita el desarrollo correcto del mismo.

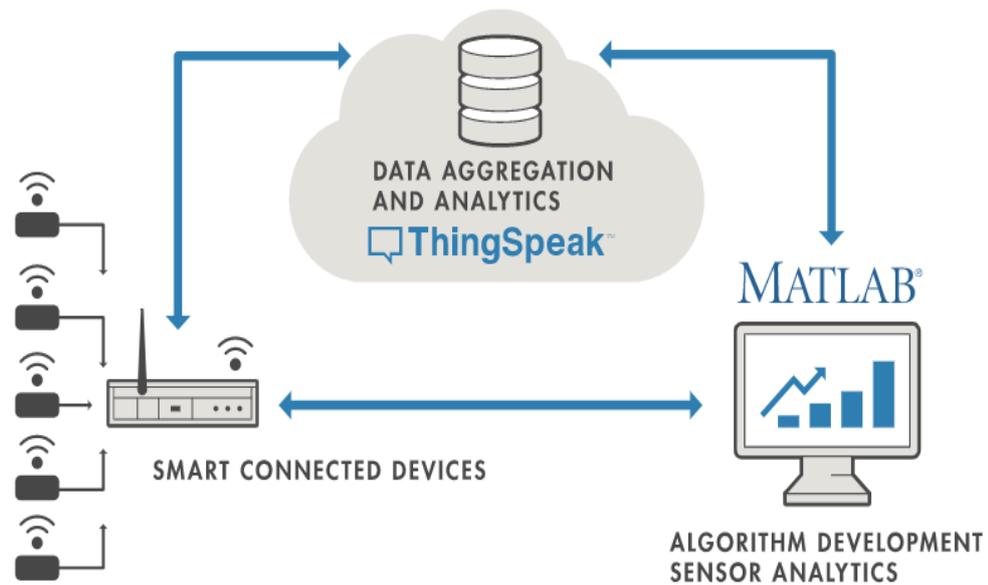


Figura 3. Esquema de desarrollo ThingSpeak.
Fuente [27].

El primer paso es la idea, donde la imaginación cuenta una parte importante, pues así se permitirá dar variedad de soluciones a problemas comerciales que puedan existir. Por otra parte, se encuentra el diseño, dentro de la misma es necesario respetar cada uno de los protocolos ya establecido, es decir que sean capaces de soportar dentro de la plataforma.

La tercera fase es el desarrollo, dentro de esta se procede a la ejecución de todo lo planificado en el paso anterior. Seguidamente se encuentra la validación, donde los criterios de Bottom- up y Up-down serán importantes pues serán destinados a la validación de protocolos que representen los datos, así como mantener un enfoque de tipo aplicativo. La penúltima fase es la puesta en producción, dentro de la misma tanto programación, así como validación deberán estar listas. Finalmente, la evolución, se enfocará en la actualización constante de la misma. es decir que sea adaptable a nuevos protocolos [28].

Comparativa de Plataformas Software IoT

Tabla IV. Comparativa de Plataformas Software IoT
Realizado por el Autor.

PLATAFORMA	ÁMBITO	HARDWARE
Carriots	Smart City Energía Edificios, Smart Home Logística Consumidor	Arduino Raspberry Electric Imp Beagle TST Industrial
Spark	Smart Home Prototipos	SPark
ThigSpeak	Smart Home Prototipos	Arduino Spark Raspberry Pi Electric Imp
BlueLabs	Smart City Energía Industria y Manufactura	SATEL Grid
Thinking Things	Smart Home Prototipos	Propietario
ZATAR	Logística	RFID

Hardware IoT

Los dispositivos “Smart” como hardware IoT pueden ser englobados como parte de “Things” en el Internet de las cosas y llegan a ser los encargados de proporcionar información de interacción en la red. Los medios de desarrollo software y hardware para IoT es variada, pero los dispositivos se pueden categorizar en dos grupos o categorías: “Wearables Devices and Gadgets” y “Embedded Systems and Boards”, de aquí que la segunda categoría tanto hardware como software están abiertas para desarrolladores y se aplican para el control de dispositivos, adquisición de datos o desarrollo de aplicación [29].

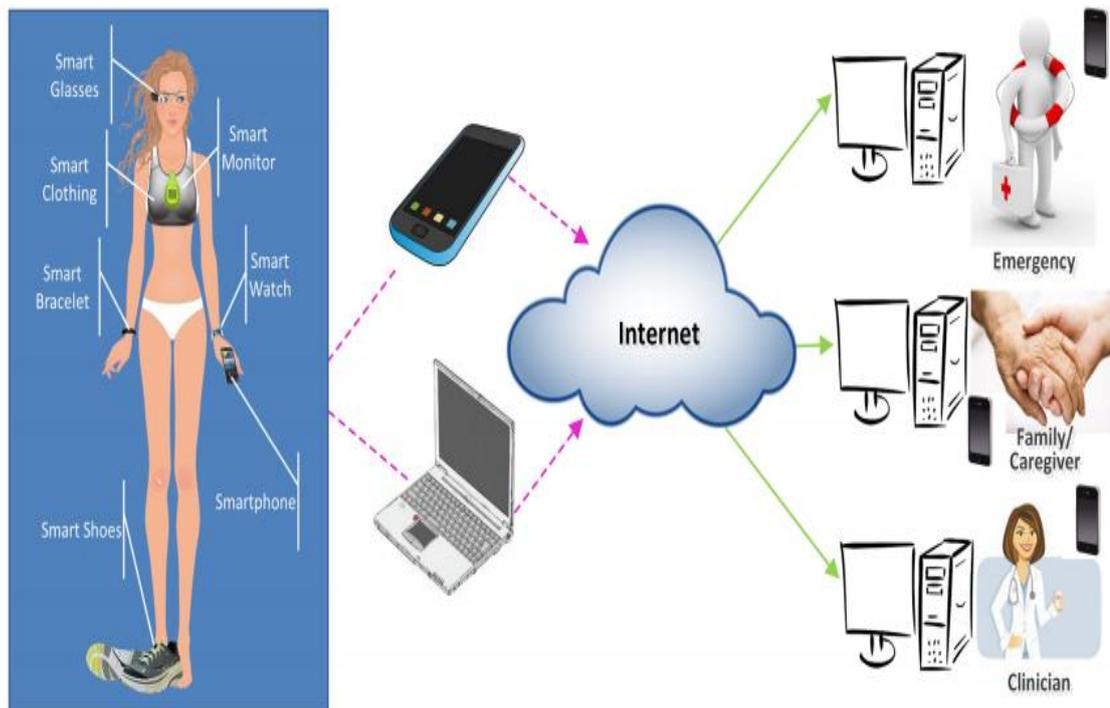


Figura 4. Arquitectura típica de sensores wearables en IoT
Fuente [30]

IoT permite la integración de varias tecnológicas de sensores que permiten la adquisición de varios parámetros en el usuario, estos sensores permiten integrar los dispositivos Smart que en la actualidad brindan la capacidad de conectarse hacia internet, y a la vez intercambiar datos de información en este caso de salud con los profesionales de salud, servicios de emergencia, o familiares, esto brinda una cobertura de todos los servicios en eHealth.

Tabla V. Tabla comparativa de Hardware IoT.
Fuente: [29]

Parámetros	RaspBerry Pi	Arduino	ESP32	Intel Galileo
Ámbito de Aplicación	Prototipos	Prototipos, Educación	Prototipos	Prototipos, Educación
Microprocesador	Broadcom BCM2837 and ARM COrtex-A53 64-b Quad Core	ATmel, Atmega328p – Atmega 128p	CPU Tensilica Xtensa LX6	Intel Quark SoC x1000
Memoria de Sistema	512 Mb	16 Kb	520 Kb	256 Mb
Velocidad de Reloj	320 MHz	16 MHz – 8MHz	240 MHz	400 MHz
Voltaje de Operación	3.3 V	2.5 ~ 3.6 V	3.3 V	5 V
Ancho de Banda del Bus	32 bits	8 bits	32 bits	32 bits
Ambiente de Desarrollo	Noobs	Arduino IDE	Arduino IDE	Arduino IDE
Puertos I/O	UART, GPIO, SPI, CSI, I2C, SPI	GPIO, UART, I2S, SPI, I2C	ADC, DAC, UART, SPI, I2S, I2C, PWM	GPIO, SPI, I2C, UART
Estándares de Comunicación	IEEE 802.11, IEEE 802.15.4, BLE 4.0, Ethernet, Serial	IEEE 802.11, Serial	IEEE 802.11 2.4GHz, BLE, BR/EDR v4.2,	IEEE 802.11, IEEE 802.15.4, BLE 4.0, Ethernet, Serial

Sensor MAX30100

MAX30100 es un sensor que permite el monitoreo de frecuencia cardiaca y saturación de oxígeno. Es un circuito que cuenta con un diodo que emite luz roja de 650nm y un fotodiodo que emite luz infrarroja de 950nm, cuenta con un filtro de luz ambiental, un ADC de 16bits, algo importante es que se puede configurar por medio de la programación el ancho de pulso, la corriente y la frecuencia de muestreo de los diodos, los datos que vienen del ADC se almacenan en una memoria FIFO (First In, First Out), para la configuración de los registros y la lectura de la memoria FIFO de 64 bytes se cuenta con comunicación I2C, esta comunicación se da por medio de dos puertos: SCL y SDA que deben necesariamente conectarse a resistencias pull up para asegurar estados lógicos correctos.

La adquisición de frecuencia cardiaca se tiene aprovechando los cambios en la absorción de la luz relacionado con el flujo sanguíneo y la desoxigenación de la hemoglobina. La oximetría de pulso brinda un enfoque en cuanto a la adquisición de parámetros como frecuencia cardiaca y saturación de oxígeno, puesto que aprovecha la absorción diferencial de la luz hallada en la hemoglobina en los estados de oxigenación y desoxigenación. La ubicación del sensor es fundamental para lecturas estables y adecuadas, este tipo de sensor se puede ubicar en el lóbulo de la oreja, la muñeca o en el dedo que son lugares donde la piel es suficientemente delgada como para que las longitudes de onda radiadas por los diodos puedan penetrar en la piel. El dedo y el lóbulo de la oreja por otro lado son lugares donde existe mayor microcirculación, así al pasar la luz polarizada por el dedo se permite la medición de glóbulos rojos que circulan por el organismo. La figura 5 muestra la ubicación adecuada en el dedo del sensor.

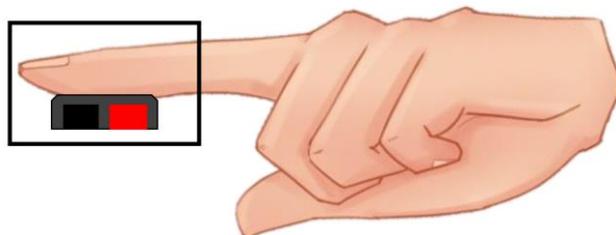


Figura 5. Ubicación de sensor en la yema del dedo.
Realizado por el autor

El sensor logra penetrar la epidermis y al existir latidos del corazón la absorción de la luz roja emitida por el diodo rojo hacia el fotodiodo varía debido al flujo sanguíneo. La variación de esta absorción se puede utilizar para el cálculo del nivel de oxígeno en la sangre específicamente en la Hemoglobina, debido a que poseen moléculas de oxígeno y son las encargadas de transportar la sangre; el sensor logra captar estas variaciones de esta manera su funcionamiento se observa en la figura 6.

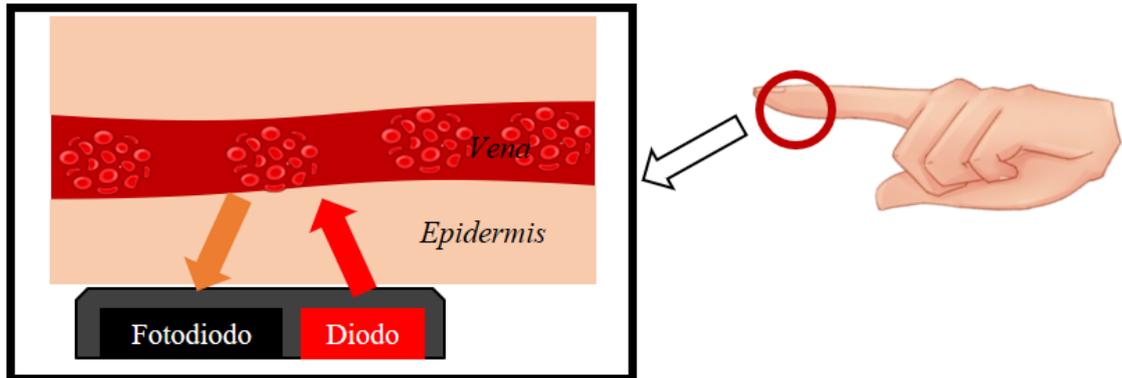


Figura 6. Funcionamiento de MAX30100 sobre el dedo
Realizado por el autor.

1.2.3.3 Telemedicina

Introducción

La Telemedicina es considerada como una gran ayuda tanto para el paciente como para el servicio de salud, pues ayuda a la atención oportuna y rápida a largas distancias. Utilizado principalmente en países desarrollados como Estados Unidos y Europa. El uso correcto de las TIC juega un papel importante dentro de la Telemedicina, pues mediante aquella es posible emitir un diagnóstico oportuno, sin importar la distancia en que se encuentre el paciente.

La Telemedicina es de gran ayuda en épocas de pandemia, pues mediante un diagnóstico virtual, se puede detener de alguna manera el aumento del agente contagioso o virus. Un ejemplo eficiente de su uso, son en las petroleras, quienes mediante Telemedicina se atiende por cualquier enfermedad o accidente que puedan tener. Fue creada principalmente con el objetivo de mejorar todos los servicios que brindan el sector de la salud, así como de contar con un gran número de especialistas, además de aportar grandes beneficios a la economía del país [31].

Con el paso del tiempo, el sector salud también ha tenido que evolucionar e incluso enfatizarse en enfermedades crónicas, donde el paciente no se puede movilizar por sí solo, es por este motivo que la Telemedicina ha sido de gran ayuda, pues muchas de las personas han sido beneficiadas. En el caso de Cuba se impulsó más en el ámbito de tele diagnóstico, donde existía una amplia gama de especialidades, donde la población podía acceder en cualquier momento.

Se podría decir que tuvo sus inicios en el año de 1900 con la creación del telégrafo, pues mediante ella se podía enviar una radiografía. Con el paso del tiempo se empezó a escuchar el Doctor por radio. Sin embargo, para el año de 1950 la NASA, creo un sistema que ayude a el control de la salud de sus tripulantes. Mientras transcurría el tiempo era más evidente escuchar temas como la asistencia médica mediante líneas telefónicas, así como la Tele radiología e incluso él envió de imágenes radiológicas, tele psiquiatría mediante las líneas telefónicas. Un punto importante en la evolución fue la primera videoconferencia de médicos realizada en Noruega en el año de 1986. Años siguientes fue evolucionando la Telemedicina hasta el punto de que un doctor fue capaz de operar de la vesícula mediante los avances tecnológicos, es decir la utilización de un brazo robot, en este punto se ve la importancia de la robótica con la medicina. Finalmente, para el 2003, se impulsa un nuevo proyecto denominado como Argonauta, impulsado por la Universidad de Chile [32].

Tipos de Telemedicina

Con el transcurso del tiempo la telemedicina ha ido evolucionando, motivo por el cual en la actualidad es evidente observar varios tipos de la misma. Uno de ellos es la Tele consulta, conformada principalmente por la indagación de información o el asesoramiento de un profesional. Seguidamente se encuentra la Teleeducación, destinada para los alumnos de medicina quienes quieren adquirir un mayor conocimiento en la materia, es por este motivo que es la mejor elección ya que así podrá educarse a larga distancia. Entorno a Tele monitoreo, los médicos la utilizan para mantener bajo control a sus pacientes, la utilización de la misma ayuda a que el medico sea capaz de obtener la información inmediata de resultados de exámenes, imágenes entre otros y así poder brindar el tratamiento ideal de su paciente [33].

Finalmente, la Tele cirugía, dentro del mismo el cirujano no se encuentra presente en la cirugía, motivo por el cual el uso de dispositivos de alta tecnología es de vital importancia, pues mediante ellos, el será capaz de realizar un trabajo eficiente, fue creado pensando en los pacientes que son inaccesibles debido a que se puede evidenciar ambientes peligrosos, así como ser un factor de riesgo para todo el equipo [33].

Ventajas y Desventajas

La ventaja principal de la Telemedicina, es la reducción de la desigualdad entre la población, pues mediante el esta es posible que cualquier persona pueda acceder directamente a este servicio. Otra ventaja radica en la accesibilidad, pues en muchos casos los pacientes se encuentran en lugares inaccesibles para el personal médico e incluso los mismos pobladores tiene problemas para poder acceder a un centro de salud. Otro punto a favor de la Telemedicina, es que ayudara al descongestionamiento de los centros de salud, e incluso el seguimiento constante de las personas que se encuentren en tratamiento. La atención de 24 horas es de gran utilidad, ya que así se podrá cubrir cualquier tipo de emergencia sin importar la hora [34].

Entorno a las desventajas existe varias, siendo la principal donde el cuerpo médico deje todo su trabajo a sus asistentes, por lo cual pueda existir malos diagnósticos. Otro aspecto, es que la información no se encuentra totalmente protegida, e incluso pueda existir errores o interferencias en el transcurso de la consulta. Es importante tener en cuenta que existirán varios médicos que no aceptaran adaptarse a este nuevo estilo de atención, por lo cual podrá existir inconvenientes [34].

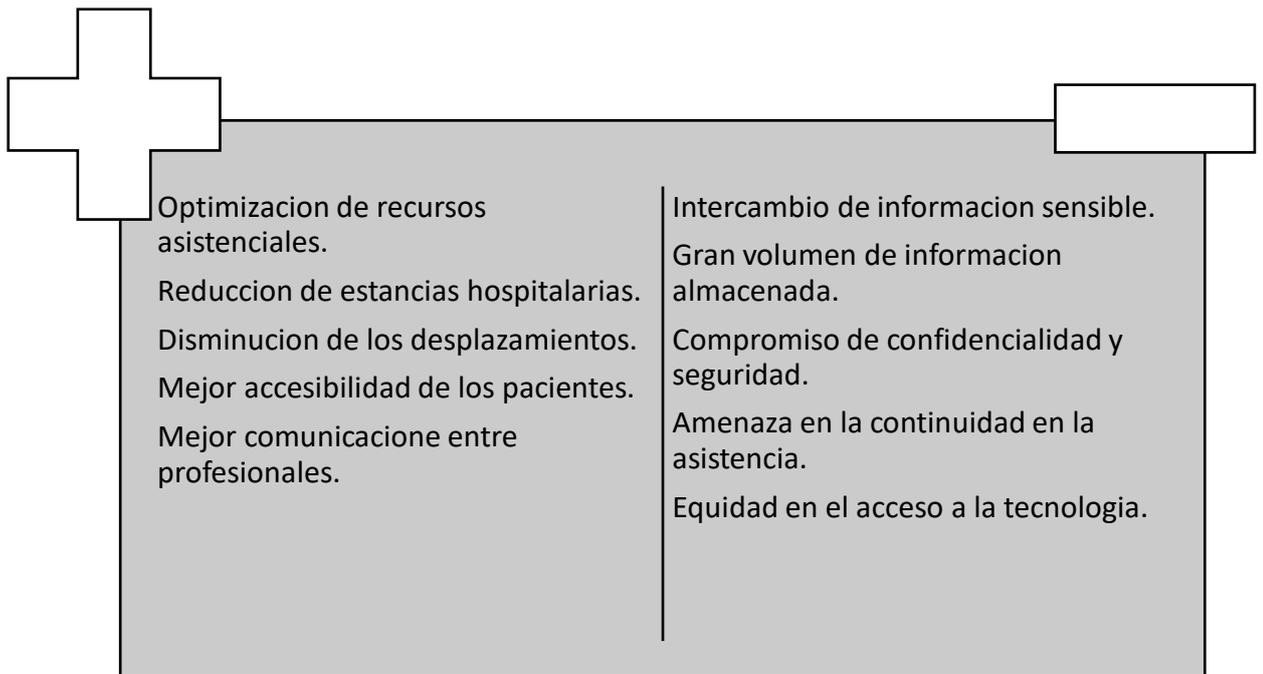


Figura 7. Ventajas y Desventajas presentes en la Telemedicina.
Realizado por el Autor.

Aspectos de Infraestructura Tecnológica

La telemedicina posee uno de los ámbitos más complejos desde la perspectiva de implementación, debido a la gran cantidad de información almacenada por sistemas anteriores heredados. Los aspectos tecnológicos e infraestructuras deben ser tenidos en cuenta como fundamentos esenciales a la hora de la implementación de servicios de telemedicina.

- Aspectos de interoperabilidad

Los datos generados en el proceso asistencial de telemedicina tanto para los pacientes como el personal médico deben ser comprensibles para los dispositivos de forma que se logre interactuar entre usuarios. La interoperabilidad es la capacidad que poseen las aplicaciones y los procesos que dan soporte al sistema, para el intercambio de información y el compartir datos entre aquellos implicados en el servicio de telemedicina [35].

- Aspectos de Infraestructura tecnológica

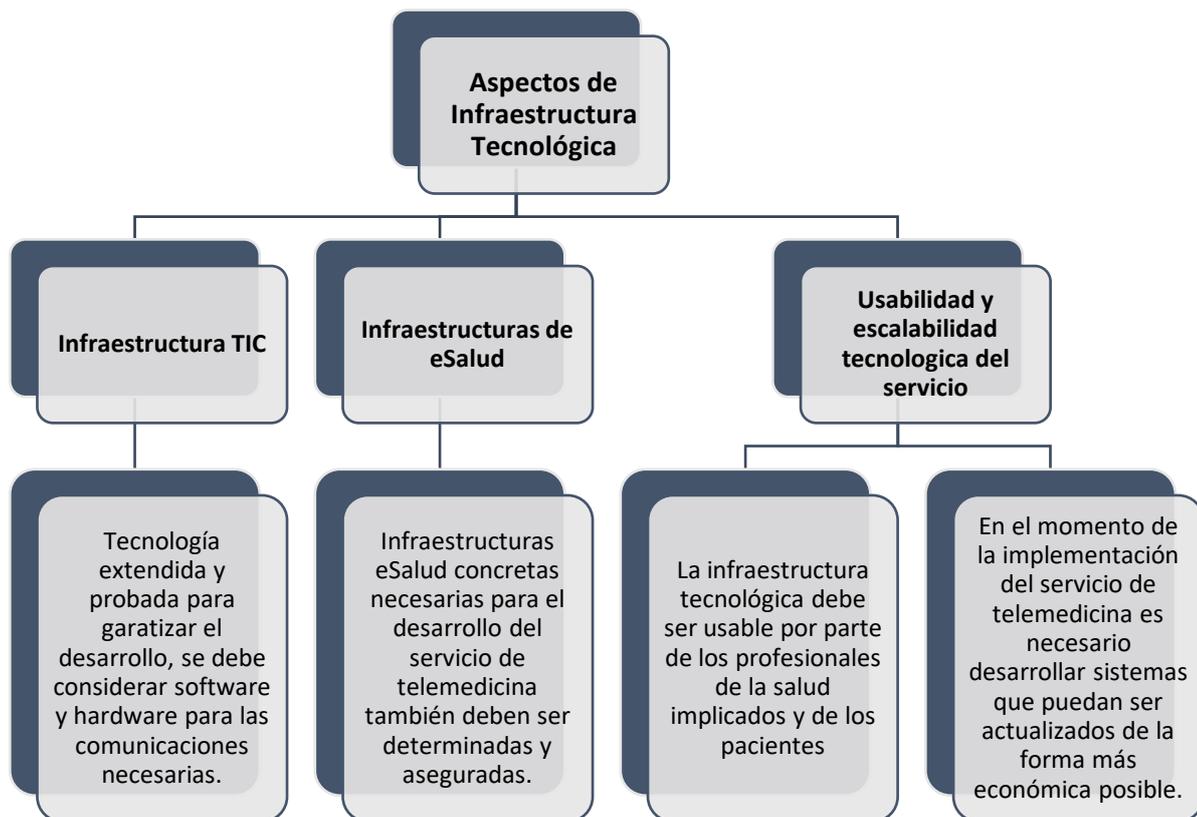


Figura 8. Aspectos de infraestructura tecnológica en Telemedicina.
Fuente [35]

Aplicaciones de la Telemedicina

Las principales aplicaciones de la telemedicina son: la Tele radiología, creado en el año de 1929, se podría considerar como el más longevo de la telemedicina, encargado en la transmisión de imágenes de los pacientes. Por otra parte, Tele oftalmología, como su nombre lo indica está destinado para el diagnóstico de enfermedades oculares, estas serán diagnosticados mediante videocámaras [33].

Entorno a la Tele patología, ha ido cambiando con el paso del tiempo, pues actualmente es posible que el patólogo pueda realizar su trabajo mediante el uso del microscopio a largas distancias. La Tele dermatología, encargado en el diagnóstico de lesiones en el campo de la dermatología, mediante él envió de imágenes en tiempo real. La Tele cardiología ha sido utilizado por más de cien años, trabaja conjuntamente con el dispositivo ECG que envía información mediante la web. La Tele pediatría, creada con la finalidad de brindar la información pertinente en tiempo real a niños.

Según la Tele psiquiatría la relación que es posible establecer tanto paciente con doctor ayuda a una mejor atención y resultados óptimos.

La Tele-neurofisiología, enfocado al cerebro pues se enfocan en el sistema nervioso tanto central como periférico. Tele obstetricia permite que la doctora mantenga un control prenatal e incluso obtener información acerca de las contracciones que puedan presentar sus pacientes y así poder brindarle la atención inmediata que se merecen [33].

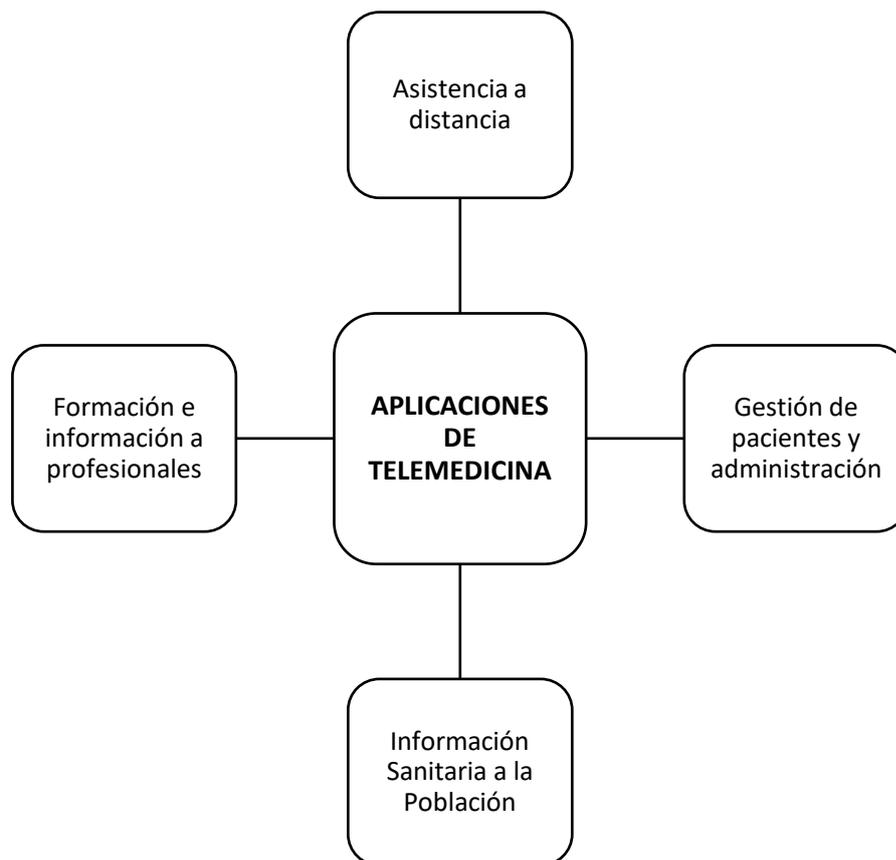


Figura 9. Resumen de aplicaciones de Telemedicina.

Realizado por el Autor.

1.2.3.4 Métodos de Adquisición de Parámetros Vitales

Los métodos que se emplean para la adquisición de parámetros vitales son definidos para cada uno, en el caso de este proyecto se monitorizara frecuencia cardiaca, saturación de oxígeno en la sangre y temperatura corporal del paciente, de esta manera describimos cada uno de los métodos que se emplean en la adquisición de estos parámetros.

Métodos no invasivos o mínimamente invasivos

Todos los métodos de medición que implican la medición de la superficie de la piel son no invasivos.

- **Frecuencia cardiaca**

La frecuencia cardiaca también se conoce como onda pulsátil de la sangre y es producida por la contracción del ventrículo izquierdo del corazón y de esta manera las extensiones de las arterias se pueden regular mediante la contracción y expansión. Además, la frecuencia cardiaca es la cantidad de sangre que ingresa a las arterias con cada sístole ventricular y la expansión para adaptación de las arterias, mediante la contracción y dilatación de las mismas. Para este parámetro se puede determinar con base en métodos como:

Método Empírico que consiste la detección de la dilatación arterial, de forma que se contabilice la cantidad de latidos que producen las arterias en el intervalo de un minuto. Otro método es el Electrocardiógrafo, que consiste en el registro durante un intervalo de tiempo de la actividad de corriente eléctrica generada por el corazón, para esto se utiliza electrodos de plata con un líquido conductor en diferentes partes del tórax.

Para esta constante vital su medida se basa también en la pulsioximetría instrumento basado en un método no invasivo [7].

- **Saturación de Oxígeno (Oximetría)**

La Oximetría es un método de adquisición de porcentaje de saturación de oxígeno SpO_2 transportado por la hemoglobina en la sangre, es un método no invasivo, el instrumento que se utiliza para esta medición basado en este método es el Oxímetro de pulso o Pulsioxímetro.

Este instrumento es aceptado como un dispositivo estándar de detección y monitoreo de hipoxemia, nivel de oxígeno en la sangre por debajo de lo normal. El procedimiento de esta medición está basado en la sangre que cambia la coloración dependiendo de cuanta saturación de oxígeno se encuentre. El instante en que las partículas de

hemoglobina permiten el paso oxígeno pierde su color rojizo o rosado, adquiriendo un tono más azulado, esto no permite el paso de la luz roja en su totalidad.

El Oxímetro de pulso determina la saturación de oxígeno en la sangre midiendo espectro-foto-métricamente el nivel de absorción de la luz roja determinando que grado de oxigenación tiene la sangre y se expresa en términos de saturación con unidad de porcentaje. Puesto que la atracción de la luz en los tejidos y en la sangre venosa es constante en todo momento, al presentarse un cambio en la atracción de la luz durante un intervalo de tiempo se debe exclusivamente a la sangre arterial. Los oxímetros miden las diferencias de absorción de las luces rojas e infrarrojas. La diferencia de absorción se relaciona directamente con la saturación de oxihemoglobina [5].

- **Temperatura**

La práctica habitual de cálculo de temperatura corporal es mediante termómetros de mercurio, mas este tipo de instrumentos conlleva una lectura lenta y de difícil apreciación y conlleva un riesgo en la liberación de mercurio. Las técnicas actuales se llevan por medio de medición electrónica, que se basan en termistores y elementos térmicos ofreciendo una precisión casi exacta de $\pm 0,1^\circ \text{C}$, otro factor favorable en este tipo de instrumentos es que su composición permite propiedades de expansión térmica hacia al sensor, pueden miniaturizarse y adaptarse a lugares adecuados para la medición es decir sobre la piel.

Los métodos de medición para temperatura que están basado en la detección de calor radiante desde la piel son de gran ventaja para el desarrollo de dispositivos de medición remota. Para la práctica de un sistema de telemedicina con instrumentos de medición no invasivos se puede considerar la medición de temperatura cutánea axilar, que es uno de los métodos habituales en la actualidad y que, gracias a sensores como Termopares, LM35 y más en electrónica se pueden realizar mediciones apropiadas y adaptables al lugar, la medición de temperatura axilar se realiza mediante la colocación de un instrumento sensor en la axila cerca de la arteria axilar y sujetándolo mediante la presión del brazo hacia el tórax [36].

Tabla VI. Tabla de resumen de métodos no invasivos de detección de parámetros vitales.

Fuente: [7] [36].

	FRECUENCIA CARDIACA	SATURACIÓN DE OXIGENO EN LA SANGRE	TEMPERATURA
Método	No Invasivo	No Invasivo	No Invasivo
Adquisición	Pulso	Pulsioximetría	Temperatura Axilar
Descripción	Detección de la dilatación arterial, de forma que se contabilice la cantidad de latidos que producen las arterias en el intervalo de	Método de adquisición de porcentaje de saturación de oxígeno SpO ₂ transportado por la hemoglobina en la sangre.	Método basado en elementos térmicos adaptables a mediciones sobre la piel.
Instrumento	Pulsómetro	Oxímetro de Pulso	Termómetro Axilar

1.2.3.5 Neumología

Es la especialidad médica que se ocupa del análisis y estudio de la patología y fisiología del aparato respiratorio, así como las técnicas de diagnóstico, métodos de terapia y prevención de enfermedad [37].

Principales Problemas en Enfermedades respiratorias

En la actualidad debido a varios factores del ambiente externo al ser humano, el pulmón se ha convertido en el órgano más vulnerable a una infección o lesiones, esto debido al incremento de la exposición que se tiene a partículas, organismos infecciosos y productos químicos que circulan en el aire del entorno global. A nivel mundial varios

millones de personas inhalan contaminantes atmosféricos del aire libre, mientras que otro porcentaje se ve expuesto a humo de fábricas y tabaco. Las enfermedades producidas por estas causas llegan a convertirse en una carga sanitaria para varios países a nivel mundial, y cinco enfermedades con trastornos respiratorios han llegado a ser incluidas entre causas comunes de mortalidad en todo el mundo [38].

- La Organización Mundial de Salud OMS cataloga a la Enfermedad Pulmonar Obstructiva Crónica (EPOC) como una enfermedad pulmonar con desarrollo progresivo, pero con alto grado de mortalidad. En base al último estudio de Carga de Morbilidad realizado en 2016 por la OMS la prevalencia de la EPOC fue de 251 millones de casos. Alrededor de un 90% de las muertes producidas por la EPOC se agudizan en países de ingresos bajos y medianos. Una de las causas más trascendentes se evidencia en los fumadores pasivos y activos, puesto que la exposición al humo del tabaco llega a afectar en gran manera la calidad de aire que fluye a través de los pulmones. Es necesario aclarar que la EPOC es una enfermedad sin cura, pero que al ser tratada llega a ser una enfermedad con alta probabilidad de mejora de vida y reducción en el riesgo de mortalidad [39].
- Dentro del grupo de enfermedades respiratorias con mayor mortalidad se puede enunciar el asma como una de las enfermedades que afecta a las vías respiratorias. Esta enfermedad provoca que las vías respiratorias se inflamen y lleguen a estrecharse razón por la cual los pulmones llegan a recibir menos aire del normal. Por tanto, el asma se convierte en un síndrome tanto complejo que aproximadamente afecta a unos 300 millones de personas entre niños y adultos jóvenes a nivel mundial. Existe una amplia variedad de criterios que llevan a un diagnóstico del asma, pero de manera general disnea, obstrucción variable de flujo de aire, inflamación de vías respiratorias, o presencia de tos llegan a ser factores cualitativos de diagnóstico. A pesar de su dificultad de diagnóstico llega a tener una prevalencia de 6.1 a 24% de la población estudiada, en Latinoamérica la media de prevalencia alcanza el 17% con fluctuaciones en países con climas variables [40].
- Durante varias décadas, las Infecciones Respiratorias Agudas IRA se encontraban dentro de las cinco causas de muerte y discapacidad entre niños y

adultos jóvenes, en la actualidad la aparición de COVID-19 coronavirus ha evidenciado que las enfermedades respiratorias han llegado al punto de convertirse en las más contagiosas y con mayor mortalidad en el mundo. Este tipo de enfermedad pueden llegar a producir fiebre alta, dolor de garganta, dolores musculares y dificultad respiratoria que es uno de los síntomas más graves de la IRA. En general muchos casos llegan a producir una enfermedad leve y en un 98% de personas con IRA logran sobrevivir, pero un 5% de las personas contagiadas llegan a producir una enfermedad con síntomas graves, provocando enfermedades críticas como neumonía grave, llegando incluso a producir la muerte. Respecto a este tipo de enfermedad respiratoria resulta complejo el definir cuantitativamente los porcentajes de mortalidad que ha alcanzado [41] [39].

- La Tuberculosis se presenta como una infección de carácter bacteriano provocado por el germen *Mycobacterium Tuberculosis*, enfermedad que ataca los pulmones y se puede extender a varias partes del cuerpo. Llega a ser una enfermedad que prevalecientemente ataca a niños, adolescentes y personas adultas con sistema inmunológico deprimido. Por otro lado, una de las enfermedades respiratorias con mayor incidencia en países Latinoamérica y de bajos ingresos es el cáncer de pulmón, enfermedad que causa un déficit de aire en el pulmón y un desgastamiento del órgano afectando a varios sistemas del cuerpo humano [38] [42].
- Para estas enfermedades se puede lograr una prevención muy alta, puesto que para Tuberculosis TB se puede evitar cumpliendo un esquema de vacunación adoptado a nivel mundial hace décadas, por otro lado, el cáncer de pulmón se puede evitar en gran medida manteniendo un alto control del tabaco, puesto que el tabaquismo es una de las causas más graves del cáncer de pulmón [38] [42].

Análisis de constantes vitales en pacientes con enfermedades respiratorias

Para el análisis de los pacientes con enfermedades respiratorias es necesario considerar el análisis individual de cada enfermedad englobada en enfermedades respiratorias, síntomas generales que pueden ser tos, fatiga, anosmia, entre otros detallados en anterioridad; mas existen síntomas que pueden ser comparados con parámetros

normales dentro de las constantes vitales, estos parámetros que pueden ser evaluados y monitoreados en pacientes con enfermedades respiratorias son: temperatura, pulso, y muy importante saturación de oxígeno en la sangre.

Tabla VII. Valores de constantes vitales según grupo etario.
Fuente [43] [36].

Cifras normales de constantes vitales según la edad				
Edad	Temperatura	Respiración	Pulso	Tensión Arterial
Recién Nacido	36.6° C - 37.8° C	>95%	120-160/min	70/50
De 1 a 4 Años	36.6° C – 37.8° C	>95%	120-130/min 90-100/min	90/50
De 4 a 8 Años	36.5° C – 37° C	>95%	86-90/min	Sistólica: #años x 2 + 80 Diastólica: mitad de sistólica +10
De 8 a 15 Años	36.5° C – 37° C	>95%	80-86/min	Sistólica: #años +100 Diastólica: mitad de sistólica +10
Edad Adulta	36.5° C	>95%	60-80/min	120/80 +o- 10
Vejez	36.0° C o menos	>95%	60 o menos	

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema de telemedicina basado en IoT para monitoreo de pacientes con enfermedades respiratorias.

1.3.2 Objetivos Específicos

- Analizar el estado del arte de pacientes con enfermedades respiratorias y equipos para su monitoreo.

En esta sección del proyecto se realizará una investigación bibliográfica para conocer la situación actual de pacientes con enfermedades respiratorias y que equipos actualmente se emplean para su monitoreo.

Para el cumplimiento de este objetivo se debe desarrollar con las siguientes actividades:

1. Análisis de la situación global actual de las enfermedades respiratorias.
2. Establecer relación de la situación actual con la situación local de las enfermedades respiratorias.
3. Identificación de los problemas que presenta un paciente con enfermedades respiratorias.
4. Análisis de las constantes vitales de un paciente con enfermedades respiratorias.
5. Estudio de los métodos de medición de saturación de oxígeno en la sangre.

- Desarrollar el modelo prototipo de una plataforma de monitoreo y control de pacientes con enfermedades respiratorias.

Para esta etapa del proyecto se comienza con el desarrollo del prototipo de sistema de telemedicina basado en el análisis del objetivo anterior.

Para el cumplimiento de este objetivo se debe desarrollar con las siguientes actividades:

1. Selección de hardware y software a utilizarse en el desarrollo del proyecto.
2. Programación de los componentes programables por medio de un software cumpliendo con las necesidades del prototipo.

- Construir un dispositivo electrónico que permita el monitoreo y control de constantes vitales en pacientes con enfermedades respiratorias.

En esta etapa del proyecto se ejecuta la construcción del prototipo basado en el diseño de hardware y software IoT del objetivo anterior a fin de que cumpla los requerimientos planteados.

Para el cumplimiento de este objetivo se de desarrollar las siguientes actividades:

1. Construcción del prototipo de dispositivo electrónico de adquisición y procesamiento de valores de saturación de oxígeno.
2. Diseño de interfaz de para presentación de datos para el paciente y el doctor con los parámetros e indicadores del sistema de monitoreo.

- Realizar pruebas para la validación de la plataforma de monitoreo y control de pacientes con enfermedades respiratorias.

Para finalizar con el sistema de telemedicina se realizará pruebas de funcionamiento del dispositivo a fin de comprobar de que cumple con los requerimientos planteados, y se tiene un monitoreo en tiempo real del paciente para su implementación.

Para el cumplimiento de este objetivo se debe desarrollar las siguientes actividades:

1. Pruebas de funcionamiento del prototipo de sistema de telemedicina para el monitoreo de saturación de oxígeno.
2. Implementación del dispositivo en pacientes con enfermedades respiratorias.

CAPITULO II

METODOLOGÍA

2.1 Materiales

2.1.1 Selección de software y hardware

2.1.1.1 Capa de Sensorización

Para la selección de hardware de la Capa de Sensorización se realizó un análisis técnico de dispositivos con características acordes a los requerimientos para el prototipo.

Microcontrolador

Tabla VIII. Tabla comparativa de selección de Microcontrolador.
Realizado por Autor.

Microcontrolador			
	ATMEGA 328P	DSPIC30F4013	PIC18F4550
Parámetros Técnicos			
Fabricante	Atmel Microchip	dsP Microchip	PIC18
Voltaje de Operación	1.8 a 5.5 VDC	2.5 a 5.5 VDC	4.2 5.5 VDC
Pines DIO	14	30	35
Pines AIO	6		
Pines PWM	6	4	2
Corriente I/O	20 mA	25 mA	25 mA
EEPROM	2 kb	1 kb	256 Bytes
RAM	2 kb	2 kb	2 kb

Memoria Flash	32 kb	48 kb	32 kb
Velocidad de Reloj	16 MHz	48 MHz	48MHz
Comunicaciones	UART, TWI (2-Wire) (I2C), SPI	CAN, I2C, SPI, UARTx2	UART, A/E/USART, SPI, I2C, MSSP
Dimensiones	4.6 mm x 34.8 mm x 7.5mm	3.81 mm x 52.26 mm x 13.84 mm	2.095 x 0.58 x 0.195plg
Precio	4,00 USD	8,90 USD	7,50 USD

Considerando todas las características técnicas que presta cada Microcontrolador para la operatividad del prototipo del sistema, se ha seleccionado el Microcontrolador ATmega 328p que como controlador cumple con las condiciones necesarias: resolución, voltaje de funcionamiento, pines de entradas y salidas, comunicaciones y costo.

Sensor de Frecuencia Cardiaca

Tabla IX. Tabla comparativa de selección de Sensor de Oximetría.
Realizado por Autor.

Sensor de Frecuencia Cardiaca			
	MAX 30100	MAX30102	Gravity Heart Rate
Parámetros Técnicos			
Método de Adquisición	Pulsioximetría	Pulsioximetría	Fotopletismografía
Datos	Frecuencia Cardiaca, Saturación de Oxígeno	Frecuencia Cardiaca, Saturación de Oxígeno	Frecuencia Cardiaca
Comunicaciones	I2C	I2C	Señal Analógica o Digital

Rango	-40 a +85°C temperatura del sensor -70 a +380°C temperatura del objeto	-50 ° C + 125 ° C	-55°C y 125°C
Precisión	0.02° C	+/- 0.25° C	+/- 0.5° C
Comunicaciones	PWM y SMBus (TWI, I2C)	Analógica	Digital (1-Wire)
Voltaje de Operación	5 VDC	4VDC – 20VDC	3 – 5.5 VDC
Wearable	Si	Si	Si
Dimensiones	16.9mm x 11.3mm	17.02 mm x 3.5 mm	6 x 50 mm
Precio	14,99 USD	2,00 USD	4,99 USD

Para la adquisición de temperatura se ha optado por el sensor LM35, puesto que brinda características adaptables al prototipo de sistema de telemedicina: voltaje de operación, precisión, costos, rango de temperatura, que hacen que sea un sensor óptimo para la adquisición de temperatura corporal.

La capa de sensorización de parámetros vitales del paciente posee dos componentes: los dispositivos de sensado y el dispositivo de procesamiento de datos. Para el componente de sensado se plantea utilizar el chip microcontrolador ATMEGA 328P que permite la adquisición de señales analógicas y digitales, posee 8 bits que permite un buen desempeño de lecturas, además posee comunicaciones por el protocolo I2C que permite la adaptación de los dispositivos sensores, el dispositivo MAX30100 permite la adquisición de datos de saturación de oxígeno en la sangre y de pulsaciones por minuto del corazón, es de tamaño reducido que ayuda al manejo e implementación en un case adaptable para el dedo, por otro lado se cuenta con el sensor MQ135 sensor que permite la lectura de la calidad del aire en el ambiente estopor medio de un puerto digital del microcontrolador, elemento que al estar ubicado en el case del dispositivo podrá tener lecturas constantes del ambiente cercano del paciente, además el sensor

LM35 que posee comunicación analógica permite la adquisición de datos de temperatura corporal de forma no invasiva pues trabaja de por expansión de temperatura en un objeto en este caso la temperatura corporal, los datos adquiridos de los sensores se procesaran en el microcontrolador para ser visualizados además en una pantalla OLED de 128x64 pixeles, estos elementos componen la capa de sensado que maneja el dispositivo.

En la figura 8 contamos con el diagrama de funcionamiento de la adquisición de datos de los parámetros vitales obtenidos por los sensores.

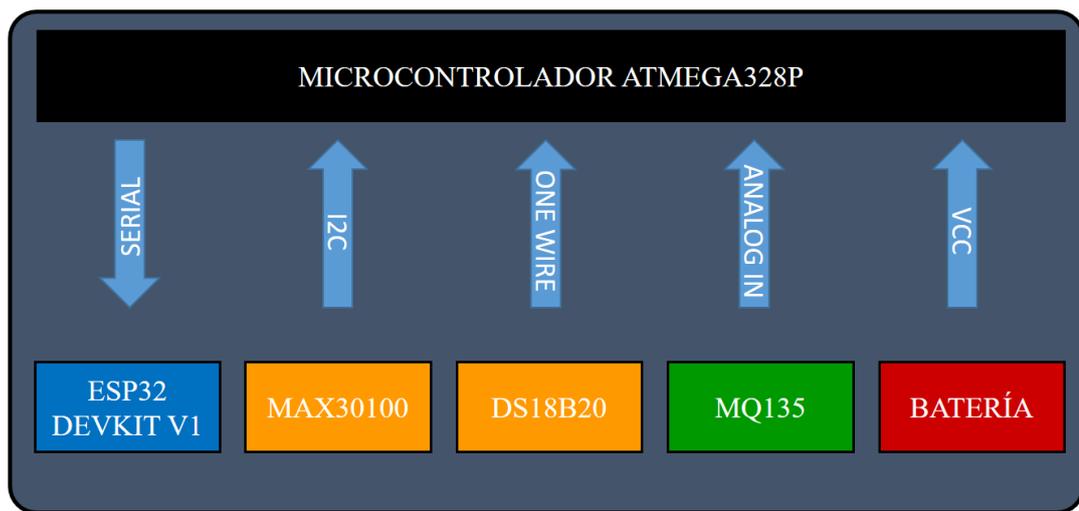


Figura 10. Diagrama de funcionamiento de Capa de sensorización.
Realizado por Autor.

Tabla XI. Recursos Capa de Sensado
Realizado por el Autor.

DISPOSITIVO	DESCRIPCIÓN	RESULTADO
MICROCONTROLADOR ATMEGA 328P	Procesamiento de señales analógicas y digitales de sensores para el muestreo de datos de constantes vitales.	Adquisición de datos de constantes vitales de pacientes obtenidos por los sensores.

MAX 30100	Sensor de adquisición de datos de frecuencia cardíaca y oxigenación de sangre basado en circuito fotoeléctrico.	Adquisición de datos de Bpm y SPO2 para procesamiento en microcontrolador.
LM35	Sensor de temperatura corporal por comunicación analógico 1Mv/°C, basado en expansión de temperatura en un objeto.	Obtención de temperatura corporal del paciente, por medio de puerto analógico hacia el microcontrolador.
MQ 135	Sensor de calidad del aire basado en la detección de gases peligrosos en el ambiente, basado en la conductividad del mismo.	Adquisición de valores ppm (partículas por millón) de forma analógica con valores TTL para su procesamiento en el microcontrolador.

2.1.1.2 Capa de Intercambio de Datos

Para la selección de hardware de la Capa de Intercambio de Datos se realizó un análisis técnico de dispositivos con características acordes a los requerimientos para el prototipo.

Tabla XII. Tabla comparativa de selección de dispositivo de intercambio de datos.
Realizada por Autor.

Módulo de Comunicación			
	MODULO ESP 32 DEV KIT V1	MODULO NODEMCU ESP8266	ETHERNET SHIELD Arduino
Parámetros Técnicos			

Dispositivo Interno	Tensilica Xtensa Dual-Core de 32 bits LX6	ESP8266 @ 80MHz (3.3V) (ESP-12E)	Ethernet W5100 de Wiznet
Comunicación	Serial, WiFi + Bluetooth, SPI, I2C	WiFi, SPI, I2C,	~
WiFi	Si 802.11 b / g / n / e / i (802.11n 2.4 GHz hasta 150 Mbit /s).	Si 802.11 b/g/n	No Posee jack RJ-45 estándar y transformador de línea integrado.
Autenticación	WEP o WAP / WAP2	WEP o WAP / WAP2	~
Bluetooth	Si Bluetooth: v4.2 BR / EDR y Bluetooth Low Energy (BLE).	No	No
Voltaje de Operación	3.3 VDC	3.3 VDC	5 VDC
Puertos	Digitales IO 25 Analógicos 6 Input 2 Output	GPIO 9 Analógicos 1	SPI MicroSD
Tipo de Antena	Interna	Interna PCB	No posee
Dimensiones	54.6 mm x 24mm	49mm x 26mm	68.6mm x 53.3mm
Precio	13 USD	7,99 USD	19 USD

Para el dispositivo WiFi de la capa de intercambio de datos, se ha optado por el Modulo ESP32 Dev Kit V1, puesto que brinda las características de hardware IoT para la aplicación requerida en el prototipo del sistema de telemedicina: Comunicación WiFi, I2C, puertos, Velocidad y Protocolos de comunicación y Autenticación.

La capa de intercambio de datos entrelaza la Capa de Sensorización con la Aplicación IoT, para esto la transmisión de datos provenientes de la capa de sensado por medio del microcontrolador se efectúa en un dispositivo wifi en este caso una tarjeta ESP-32 DEVKIT V1 – DOIT que al poseer características aplicables al IoT permite la combinación de estos aspectos. La ESP32 DEVKIT posee una gran cantidad de periféricos que permiten la adquisición de datos de sensores o comunicaciones de tipo I2C, SPI, UART, protocolos que permiten la transmisión de datos. Los datos de los sensores provenientes del microcontrolador en la capa de sensado se transmiten por comunicación serial entre los dos dispositivos, en este punto se posee adquisición de datos de Sensores de constantes vitales del paciente, visualización de datos por una pantalla OLED. El dispositivo cuenta con conectividad WiFi 802.11 b/g/n/e/i con velocidad de hasta 150 Mbit/s, por otro lado, Bluetooth 4.2. BLE, la conectividad WiFi permite la conexión con la red en un Access Point para la integración de la aplicación IoT.

En la Figura 9 se muestra el diagrama de funcionamiento de la etapa de intercambio de datos entre el Microcontrolador y la Aplicación IoT.

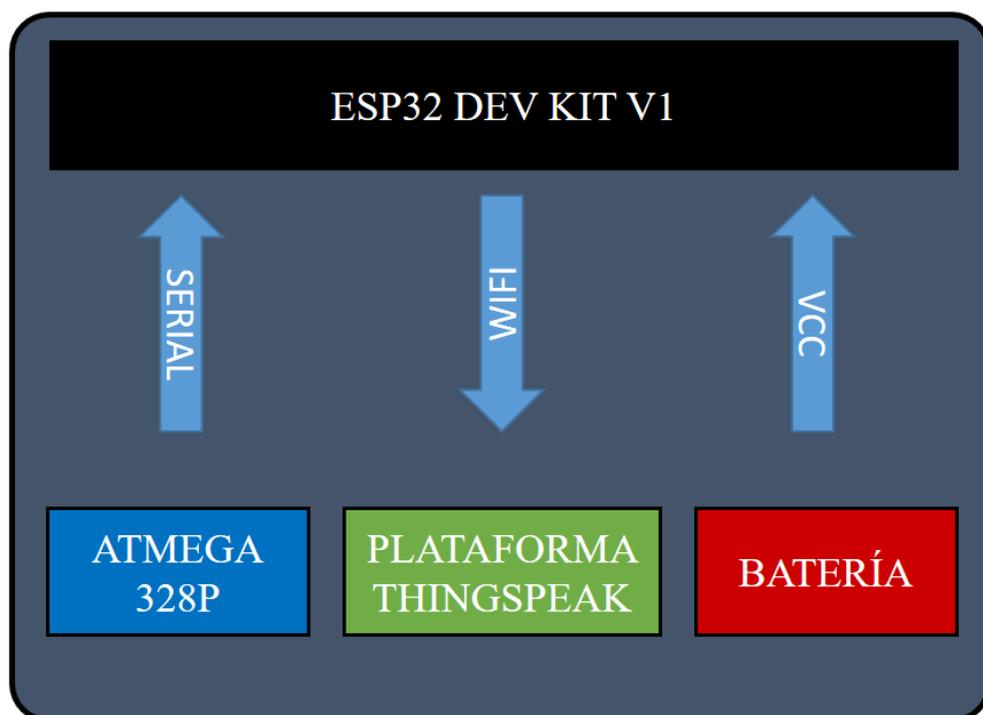


Figura 11. Diagrama de funcionamiento de Capa de Intercambio de datos.
Realizado por Autor.

Tabla XIII. Recursos de Capa de Intercambio de Datos.
Realizado por el Autor.

DISPOSITIVO	DESCRIPCIÓN	RESULTADO
ESP32 DEVKIT V1 – DOIT	Micronrolador con tecnología WiFi y Bluetooth dual para el control y manejo de datos de sensores, actuadores o módulos con aplicación de hardware en IoT, cuenta con protocolos de comunicación SPI, UART, I2S, I2C, fundamentales para la aplicación del Internet de las Cosas.	Dispositivo conectado a una red WiFi que recibe los datos sensados del microcontrolador ATMEGA 328P para la transmisión a la aplicación IoT, datos que se transmitirán en tiempo real para su visualización y monitoreo.
ROUTER DE CONEXIÓN A INTERNET	Router brinda conexión a internet por medio de IEEE 802.11 b/g/n, con asignación DHCP Protocolo de Configuración Dinámica de Host, hacia los dispositivos del lugar, que posean características de comunicación WiFi.	El Router permite el acceso a la red WiFi con conexión a Internet a los dispositivos, el ESP32 se enlazara a la red por ese dispositivo para la salida de datos hacia la aplicación IoT y su plataforma.

2.1.1.3 Aplicación IoT y Servicios de Nube

Para el manejo y alojamiento de datos se propone el uso de IoT ThingSpeak, plataforma abierta de aplicaciones, que permite conectar al usuario con datos de objetos, una de las características principales es que esta plataforma es Open Source Ruby on Rails 3.0 RoR con una API que permite el almacenamiento y recuperación de datos de objetos en este caso sensores usando el protocolo HTTP sobre red LAN o Internet. ThingSpeak permite la integración con hardware IoT en este caso con el dispositivo ESP32. ThingSpeak proporciona visualizaciones instantáneas de los datos

publicados por hardware IoT a ThingSpeak, con la capacidad de ejecutar el código MatLab en ThingSpeak, que permite el realizar análisis y procesamiento en línea de los datos conforme estos ingresan.

En la figura 10 se muestra el diagrama de funcionamiento de la capa de aplicación de plataforma IoT.

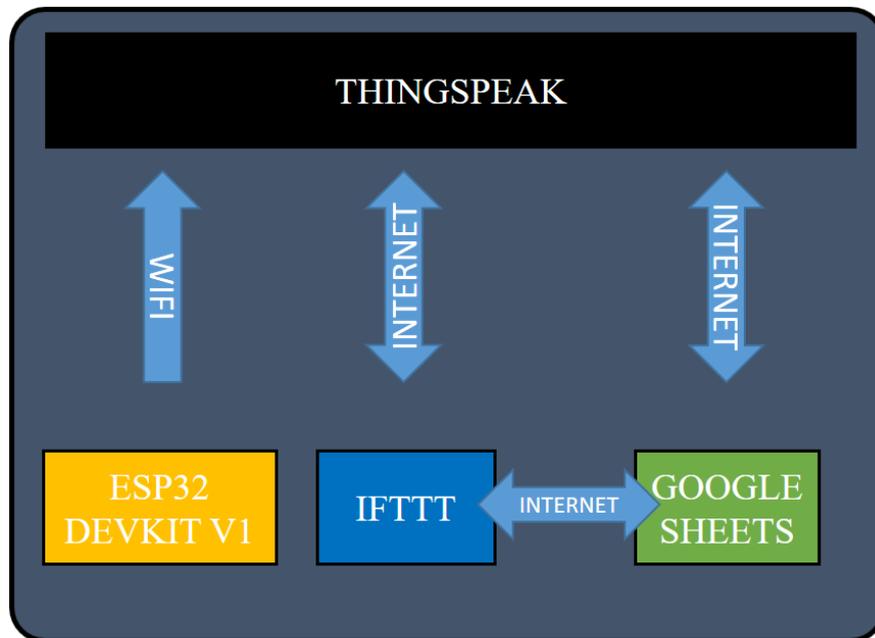


Figura 12. Diagrama de funcionamiento de Capa de aplicación IoT.
Realizado por Autor.

2.2 Métodos

2.2.1 Modalidad de la Investigación

a. Investigación Bibliográfica

El presente proyecto se basa en investigación bibliográfica puesto que la recopilación de información se obtiene por medio de artículos científicos, revistas con publicaciones científicas, trabajos previos de investigación relacionados con el área y tema que se está investigando, además de la búsqueda en repositorios de universidades del país, esto con el fin de proporcionar la fundamentación teórica, métodos, instrumentos e información usadas para el desarrollo del proyecto.

b. Investigación Aplicada

El proyecto está basado en Investigación Aplicada puesto que el fin del proyecto es el prototipo de un dispositivo funcional como producto de la Investigación desarrollada en el proyecto.

c. Investigación de Campo

Este tipo de Investigación se usa para el estudio y análisis del sistema de telemedicina para la medición de saturación de oxígeno, de esta manera obtener información de las características que poseen los pacientes con enfermedades respiratorias.

2.2.2 Recolección de Información

Para el desarrollo investigativo del proyecto, se utilizó documentación que aporte de manera teórica el desarrollo de la misma. Para esto se analizará documentos como artículos científicos, proyectos de investigación previos de repositorios de Universidades del Ecuador y libros.

2.2.3 Procesamiento y Análisis de Datos

El procesamiento y análisis de datos contemplan diferentes etapas que se mencionan a continuación:

- Recolección y análisis de información recopilada.
- Desarrollo y estudio de propuestas de solución para el prototipo de medición de saturación óptimo con sensibilidad adecuada y presentación visual de datos.
- Determinación de la mejor propuesta de solución.
- Planteamiento de la propuesta de solución.

2.2.4 Propuesta de Solución

La realización de un dispositivo electrónico de medición de saturación de oxígeno, frecuencia cardíaca y temperatura aplicado a personas con enfermedades respiratorias que permita tanto a los pacientes como al médico tratante conocer los parámetros vitales en la persona durante un periodo de dos horas máximo por medio de una aplicación IoT con interfaz gráfica.

2.2.5 Desarrollo del Proyecto

- Análisis de la situación global actual de las enfermedades respiratorias.
- Relación de la situación actual con la situación local de las enfermedades respiratorias.
- Identificación de los problemas que presenta un paciente con enfermedades respiratorias.
- Análisis de las constantes vitales de un paciente con enfermedades respiratorias.
- Estudio de los métodos de medición de saturación de oxígeno en la sangre.
- Identificación de dispositivos actuales de medición de saturación de oxígeno en la sangre por pulso.
- Selección de hardware y software a utilizarse en el desarrollo del proyecto.
- Programación de componentes por medio de un software de manera que se cumpla con las características del prototipo.
- Construcción del prototipo de dispositivo electrónico de adquisición y procesamiento de valores de saturación de oxígeno.
- Diseño de interfaz de para presentación de datos para el paciente y el médico con los parámetros e indicadores del sistema de monitoreo.
- Pruebas de funcionamiento del prototipo de sistema de telemedicina para el monitoreo de saturación de oxígeno.
- Implementación del dispositivo en pacientes con enfermedades respiratorias.
- Análisis de los resultados obtenidos con la implementación del prototipo.

CAPITULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y Discusión de Resultados

3.1.1 Desarrollo de la propuesta

En este proyecto se desarrolla un sistema de telemedicina basado en IoT para el monitoreo de pacientes con enfermedades respiratorias. Para esto se establece una arquitectura del prototipo del sistema de telemedicina, basado en la arquitectura básica de un sistema de telemedicina IoT.

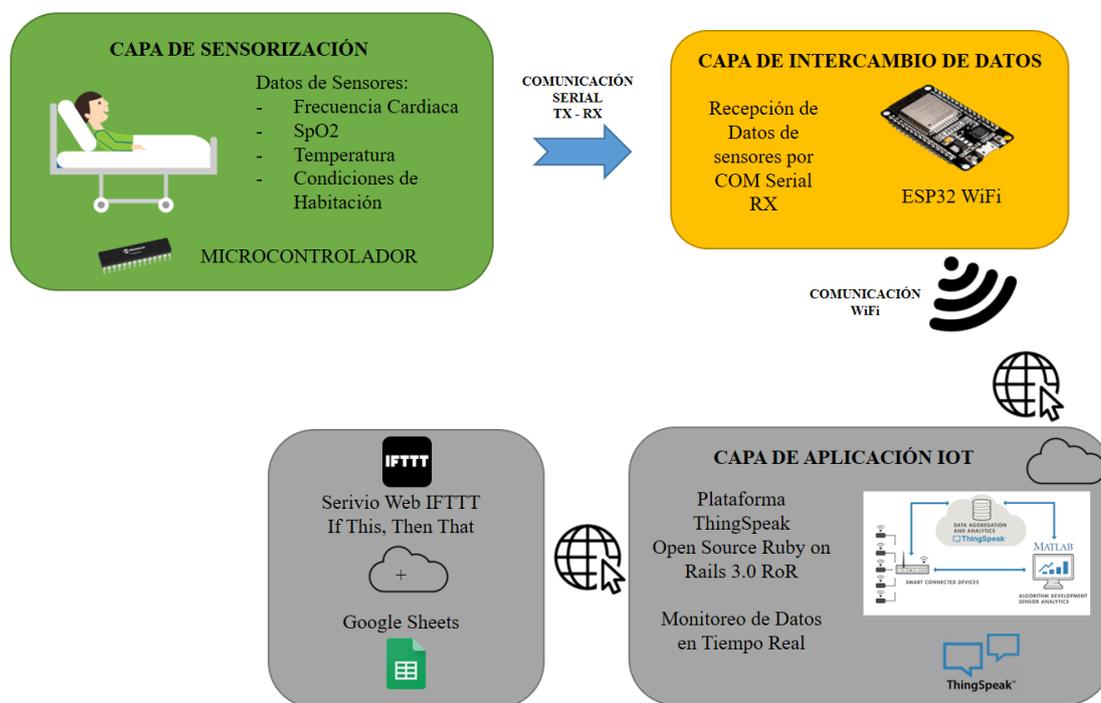


Figura 13. Arquitectura de prototipo de sistema de telemedicina basado en IoT.
Elaborado por Autor.

3.1.1.1 Capa de Sensorización

Se diseñó un dispositivo de adquisición de datos de sensores para la Capa de Sensorización, el cual se compone por dos dispositivos de adquisición de parámetros vitales (sensores biomédicos), microcontrolador para el procesamiento de datos y que

además transmitirá los datos hacia el componente de intercambio de información por medio de comunicación Serial. El sensor MAX30100 permite la adquisición de datos de Frecuencia Cardíaca y Saturación de Oxígeno, trabaja con un voltaje de 3.3 volts en corriente directa y como puerto de comunicación hacia el controlador utiliza comunicación I2C. El sensor LM35 encapsulado permite la adquisición de Temperatura Corporal, tiene un voltaje de operación de 5 volts y posee una salida analógica hacia el microcontrolador, posee una resolución de 0,010 volts por cada grado centígrado detectado. El sensor MQ135 permite comprobar la calidad del aire, posee un voltaje de operación de 5 volts de corriente directa, posee una salida dual analógica y digital, en este caso se comunica por medio de la salida digital hacia el microcontrolador. El microcontrolador ATMEGA 328p es el encargado del procesamiento de los datos adquiridos por los sensores, trabaja con 5 volts de corriente directa, posee puerto I2C para la comunicación con el sensor MAX30100 y puerto analógico para el sensor LM35 y puerto analógico para MQ135. La visualización de estas lecturas se realiza mediante una pantalla OLED de 128 x 64 pixeles trabaja con un voltaje de corriente directa de 3.3 y posee comunicación I2C.

3.1.1.2 Capa de Intercambio de Datos

Esta capa se encarga del entrelazado de datos de la Capa de Sensorización con la Capa de Aplicación IoT, utiliza hardware IoT que para este caso será una tarjeta ESP32, la cual cuenta con comunicación serial para la recepción de datos provenientes del microcontrolador, además en este punto se desempaquetan los datos y se actualizan en la plataforma ThingSpeak por medio de la interacción con API's de la Capa de Aplicación de IoT mediante el protocolo HTTP. La tarjeta al contar con conectividad WiFi 802.11 b/g/n/e/i a velocidad de hasta 150 Mbit/s permite la comunicación hacia un punto de acceso a internet para la actualización de los datos.

3.1.1.3 Capa de Aplicación IoT

La plataforma ThingSpeak basada en Open Source cuenta con una API que permite almacenar los datos provenientes de la Capa de Intercambio de Datos para recuperarlos y visualizarlos por medio de canales creados para la interpretación de los resultados, al ser una plataforma que permite la integración de hardware IoT es adecuada para el enlace con la tarjeta ESP32, se utiliza una de las aplicaciones que brinda Things Speak:

MatLab Analysis que permite explorar los datos que se han almacenado en el canal de ThingSpeak; la plataforma IFTTT permite enlazar Google Sheets con ThingSpeak, se utiliza la aplicación TimeControl, aplicaciones que permiten la generación de reportes cada 5 minutos con los datos de Frecuencia Cardiaca, Saturación de Oxígeno y Temperatura Corporal.

3.1.2 Implementación de circuito de monitoreo de signos vitales.

3.1.2.1 Adaptación de sensor MAX30100

Para el uso del sensor MAX30100 se requiere de la programación en el microcontrolador en este caso se utilizará la librería “MAX30100.h”, esta librería se puede instalar desde Arduino IDE o por GitHub. La librería permite leer datos del fotodiodo del sensor, estos datos se presentan con una oscilación ligera y con una componente en corriente directa como se observa en la figura 12, esta componente deberá ser eliminada para poder leer los datos de pulso cardiaco y nivel de saturación de oxígeno. Para eliminar esta componente en corriente directa se cuenta con un filtro IIR, filtro de primer orden, que es un filtro digital con entrada de señal impulso y con salida un número infinito de términos que no vuelven a reposo, la ventaja de este filtro es que cuenta con un orden menor de filtro lo que representa una menor carga del sistema computacional del microcontrolador. El filtro IIR usa las ecuaciones:

$$w(t) = x(t) + \alpha * w(t - 1).$$

$$y(t) = w(t) - w(t - 1).$$

Donde:

- $y(t)$: Salida del filtro
- $x(t)$: Valor actual (Entrada del Filtro)
- $w(t)$: Valor intermedio (Historial de valor DC)
- α : Constante de respuesta del filtro

Para la eliminación de la componente en DC, α deberá tener un valor muy cercano a 1, por esta razón es necesario un filtro pasa alto, de esta manera la señal de salida pasando por la eliminación de la componente en DC quedaría como la figura 14.

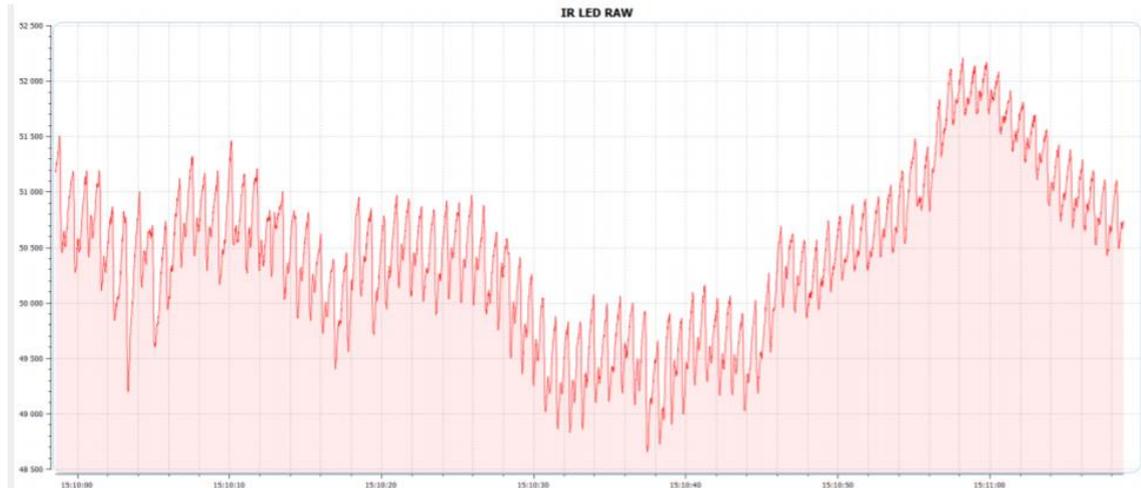


Figura 14. Gráfica de los datos del sensor con oscilaciones y componente CC y CA.
Fuente: [44].

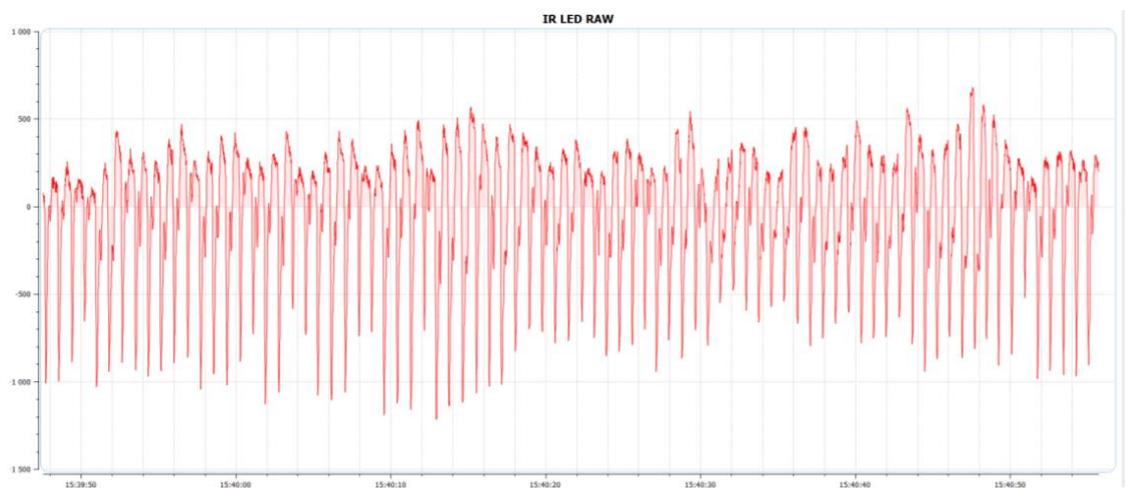


Figura 15. Gráfica de los datos del sensor luego de eliminar la componente en DC.
Fuente: [44].

Una vez que de la señal ha sido eliminada la componente DC se debe mejorar la capacidad de detección de los pulsos, tomando el diferencial de la señal se obtiene los cambios más bruscos, así aplicando un filtro se obtendrá la media del valor de cambio en la gráfica como se muestra en la figura 15, al pasar por este filtro se tiene que aún existen oscilaciones en la parte inferior de la señal, para esto es necesario aplicar un filtro pasa bajo para la suavización de esta señal, el filtro Buttersworth como FPB se empleó ante lo cual se debe establecer Frecuencia de muestreo FS y Frecuencia de corte FC. En el sensor MAX30100 la FS es de 1 KHz, de acuerdo al fabricante del sensor el ancho de pulso y la frecuencia de muestreo están relacionadas obteniendo

como resultado la resolución. La librería del sensor MAX30100 selecciona una resolución de 16 bits lo que significaría que trabaja con una frecuencia de muestreo de 100Hz. Por otro lado, suponiendo que la mayor cantidad de batidos por minuto BPM en un paciente es de hasta 220 BPM, la cantidad más baja sería de 50 BPM, la frecuencia quedaría:

$$f = \frac{220 \text{ BPM}}{60} = 3.66 \text{ Hz}$$

$$f = \frac{50 \text{ BPM}}{60} = 0.83 \text{ Hz}$$

Entonces asumiendo que la cantidad de batidos por minuto es 220 BPM o 3.66 Hz la frecuencia objetivo, el filtro Butterworth permitiría el paso de frecuencias deseadas con una tasa de muestreo con valores tan bajos como:

$$\frac{3.66 \text{ Hz}}{0.1} = 36.6 \text{ Hz}$$

Debido a que la mayor frecuencia que se puede contar en la señal es 3.66 Hz, la frecuencia de corte por tanto del filtro será 4 Hz, pero con esta frecuencia el filtro podría eliminar señales útiles para la medición, en este proceso se establece una frecuencia de 10 Hz para la frecuencia de corte FC, a esta FC será suficiente como para limpiar y mejorar la señal a fin de medir los picos adecuados para la Sensorización, así se tiene la salida de la señal del filtro como se muestra en la figura 16.

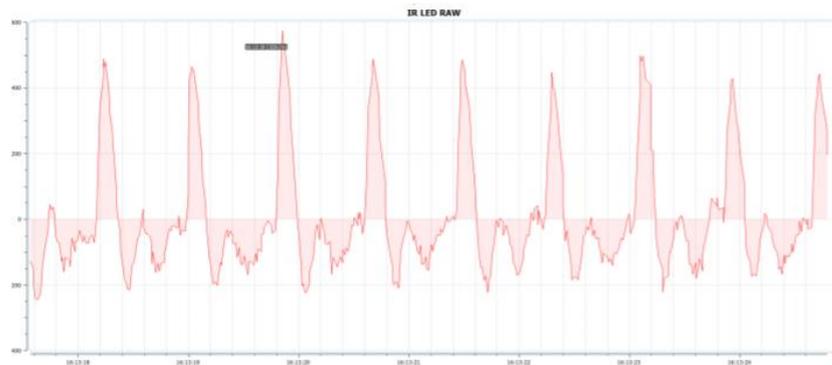


Figura 16. Gráfica de señal aplicando filtro de media
Fuente: [44].

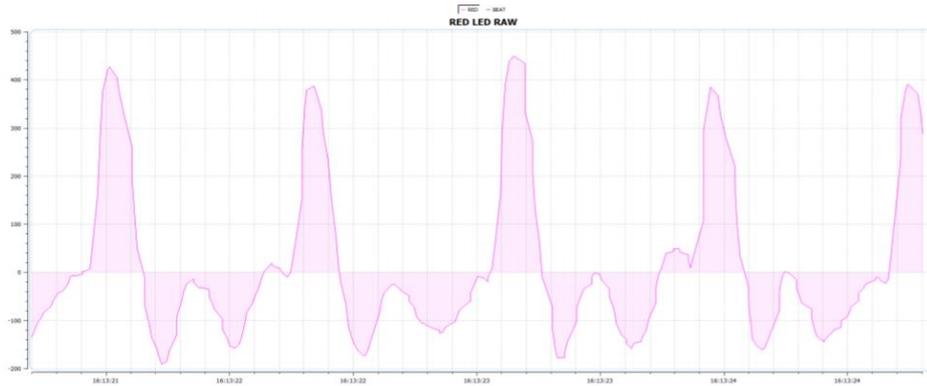


Figura 17. Gráfica de señal a la salida del filtro pasa bajo.
Fuente: [44].

Detección de batidos por minuto BPM

Con la señal del sensor MAX30100 filtrada, se puede calcular la frecuencia cardiaca, para esto se utiliza una máquina de estado que, cuando la señal alcanza el umbral de alto este sigue la onda hasta un punto donde la señal caiga, en este punto se guarda una marca de tiempo. Durante este intervalo de tiempo, la diferencia entre el inicio y el final del umbral de la señal será nuestro retraso medido entre los dos tiempos, a partir de aquí se puede calcular los batidos por minuto BPM. En la librería existe la función `milis()`, que brinda una marca de tiempo en mili segundos y al contar ya con dos marcas de tiempo anteriores guardadas, se logra el cálculo de frecuencia cardiaca mediante:

$$BPM = \frac{6000}{\text{Marca de Tiempo Actual} - \text{Marca de Tiempo Anterior}}$$

Detección de Saturación de Oxígeno

La oximetría de pulso o arterial está basada en valores medidos en cada instante de pulsación de sangre arterial, esto se debe a que el lugar de ubicación del sensor debe ser un lugar que la sangre realice pulsaciones. La luz absorbida varía respecto a la cantidad de sangre, esto permite la estimación de SpO2 arterial. El cálculo se obtiene al encontrar los máximos y mínimos de las pulsaciones detectadas, y estas corresponden a las absorciones de la luz máxima y mínima, al dividir las se obtiene un cociente R, que determinará la Saturación SpO2.

$$R = \frac{R_{ab}}{I_{ab}}$$

Donde:

- R_{ab} : cociente entre absorción max y min del diodo rojo
- I_{ab} : cociente entre absorción max y min del diodo infrarrojo

Por otro lado, se puede medir la amplitud RMS de cada señal: diodo rojo y diodo infrarrojo, al contar con la componente en AC que posee el dato de absorción de luz en las arterias, para el cálculo de los valores RMS se puede elevar al cuadrado los valores sensados por el MAX30100, este cociente R permite una aproximación con la tabla de proporción de cálculo de oxigenación de la sangre, así el cociente R quedaría:

$$R = 100 * \frac{\log(\text{Valor RMS de intensidad diodo rojo})}{\log(\text{Valor RMS de intensidad diodo infrarrojo})}$$

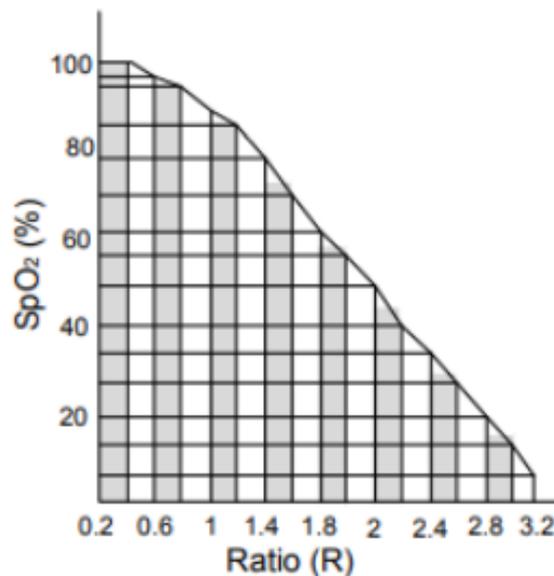


Figura 18. Tabla de relación para cálculo de SpO2.
Fuente [44]

Así la librería “MAX30100.h” cuenta con archivos para el cálculo de cada uno de los datos, tanto Frecuencia Cardíaca como Saturación de Oxígeno en la Sangre en los archivos: “MAX30100_SpO2Calculator” y “MAX30100_BeatDetector” archivos que se anexan en este documento.

3.1.2.2 Adaptación de sensor LM35

El sensor LM35 encapsulado permite la adquisición de temperatura, con datos en grados centígrados, posee un rango de lectura de temperaturas desde -55°C hasta 150°C , una de las ventajas que brinda este dispositivo es que no necesita de configuración externa, cuenta con una precisión de $0,5^{\circ}\text{C}$, posee una salida de tipo lineal alcanzando una ganancia de $10\text{mV}/^{\circ}\text{C}$. Este sensor cuenta con propiedades de expansión térmica lo que brinda un manejo y ubicación óptimas para la aplicación como sensor Wearable. Considerando que uno de los lugares óptimos para la toma de temperatura corporal es la axila, el sensor de tipo encapsulado ayuda de esta manera en su ubicación para la toma de temperatura axilar como se muestra en la figura 19.

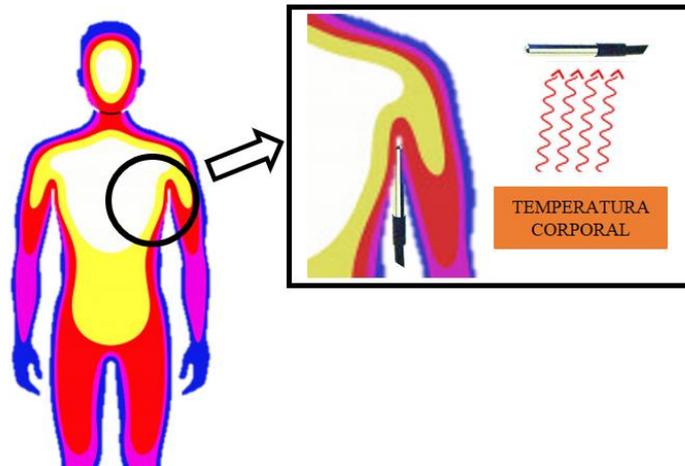


Figura 19. Ubicación del sensor para temperatura axilar.
Realizado por el autor.

Este sensor posee una salida analógica por lo que para su utilización en el sistema se asigna un puerto analógico para su lectura. El sensor posee una alimentación de 5 volts, para realizar mediciones se considera que en una lectura de 25°C el voltaje de salida en el puerto del sensor será de 0,25 volts por tanto se tendría una ganancia de $10\text{mV}/^{\circ}\text{C}$; es decir, que para obtener el valor de salida en volts el valor de temperatura adquirido habrá que multiplicar por 100, obteniendo la fórmula de Temperatura de la siguiente manera:

$$T(^{\circ}\text{C}) = V_{OUT} * 100$$

Considerando que el microcontrolador que se emplea para el procesamiento de los datos trabaja a 5 volts en corriente directa y asigna un rango de 0 a 5 volts para los puertos de entrada analógica; es decir, un valor de 0 a 1023. De esta manera el cálculo de la temperatura en el microprocesador sería:

$$V_{OUT} = \frac{5 * analogRead}{1023}$$

$$T(^{\circ}C) = \frac{500 * analogRead}{1023}$$

Así obtendremos una gráfica de Temperatura respecto a voltaje como se muestra en la figura 20.

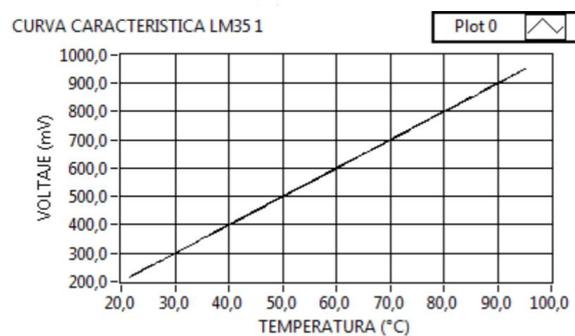


Figura 20. Gráfica temperatura vs voltaje.
Fuente [45]

3.1.2.3 Adaptación de sensor MQ135

MQ135 es un sensor que permite controlar la calidad del aire, está dirigida para lugares cerrados como edificios u oficinas, este sensor es capaz de detectar NH₃, NO_x, alcohol, benceno, smoke, CO₂, entre otros. El sensor se compone de: un tubo compuesto de cerámica con óxido de aluminio cuyo compuesto es Al₂O₃, una capa que es sensible de dióxido de estaño SnO₂, posee también un electrodo de medida y por último un calentador que es fundamental para proporcionar las condiciones del sensor adecuadas. Este sensor no proporciona directamente las partículas por millón ppm del gas, sino que devuelve una cantidad proporcional a la resistencia que produce cierta cantidad de gas en el sensor. El sensor necesita buscar la presencia de agentes externos fuera de los normales para esto se ubicará en el case del dispositivo de sensorización para poder prevenir agentes externos y dañinos cercanos al paciente.

El sensor cuenta con una salida dual, lo que permite contar con puertos de salida digital y analógica, en este caso se trabaja con la salida analógica para realizar lecturas por el puerto analógico del microcontrolador.

Cálculo de resistencia de carga

Para calcular los datos de partículas por millón ppm del sensor se utiliza la fórmula que se basa en datos de la tabla de curva de gases del sensor MQ135 extraído del Datasheet del sensor.

$$RL = Rs * e^{\left(\frac{\log\left(\frac{\text{intercepto}}{\text{ppm en el aire limpio}}\right)}{\text{pendiente}}\right)}$$

Donde:

- RS: Resistencia superficial
- RL: Resistencia de carga

Por otro lado, para el cálculo de la resistencia superficial se basa en un análisis de voltajes:

$$Rs = \frac{Vc - VRL}{VRL} * RL$$

Donde:

- RS: Resistencia superficial
- Vc: Voltaje in
- VRL: Voltaje out
- RL: Resistencia de carga

Cálculo de partículas por millón PPM

Con los datos de las resistencias de carga y superficial se puede realizar un cálculo adecuado de las partículas por millón presentes en el sensor, lo que dictaminaría la calidad del aire presente en la habitación o lugar que se encuentre el paciente.

$$ppm = \text{Intercepto} * \frac{R_{S^{\text{pendiente}}}}{R_o}$$

Donde:

- RS: Resistencia superficial
- Ro: Resistencia interna

Tabla XIV. Curva de gases del sensor MQ135.
Fuente [46]

Gas	Intercepto	Pendiente
CO2	113.71052	-3.01971
NH4	84.07117	-4.41107
CH3	47.01770	-3.28190

3.1.2.4 Programación de Microcontrolador ATMEGA 328p

La programación del microcontrolador conlleva conocer las conexiones que se realizan entre los sensores: MAX30100, LM35, MQ135, además la pantalla OLED, para esto las conexiones entre estos dispositivos se muestra en la figura 21.

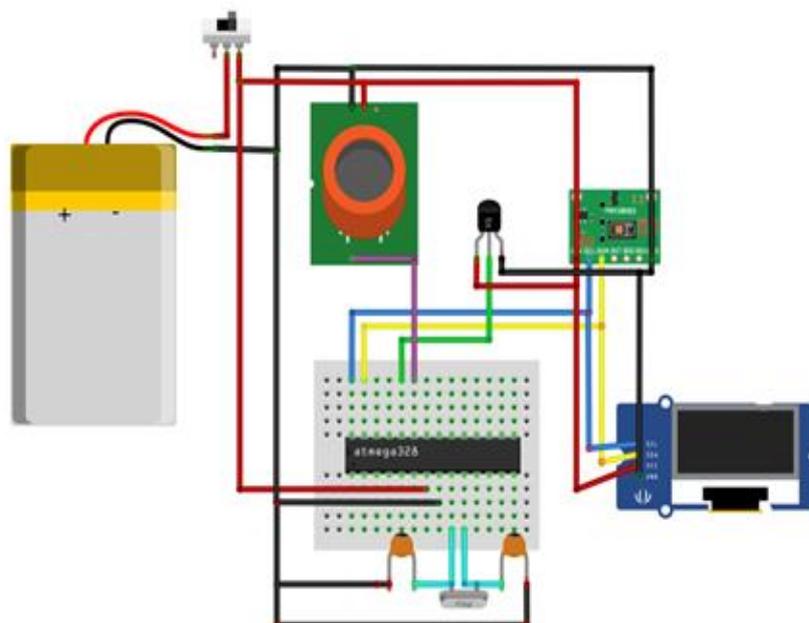


Figura 21. Diagrama de conexiones para programación.
Elaborado por Autor.

Tabla XV. Conexiones de circuito de sensorización.
Realizado por Autor.

Dispositivo	Puerto	Tipo de Comunicación	Descripción
MAX30100	27	I2C	PC5 - ADC5/SCL
	28	I2C	PC4 - AD4/SDA
LM35	24	Analógico	PC1 – ADC1
MQ135	25	Analógico	PC0 – ADC0
PANTALLA OLED	27	I2C	PC5 - ADC5/SCL
	28	I2C	PC4 - AD4/SDA
COMUNICACIÓN SERIAL	3	Serial	PD1 - TXD

Lectura de Frecuencia Cardíaca BPM y Saturación de Oxígeno en la Sangre SpO2

Los dos parámetros se obtienen del sensor MAX30100 la comunicación se realiza por medio de I2C agregando la librería “*MAX30100_PulseOximeter.h*” se puede realizar la adquisición de BPMs y SpO2 mencionado en el apartado 3.1.2.1, para esto se define una variable “*PulseOximeter pox*” y se inicializa la librería por medio de “*pox.begin*”.

```

if (!pox.begin())
{
    for(;;);
} else
{
    pox.setOnBeatDetectedCallback(onBeatDetected);
}
pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

```

Luego es necesario actualizar los datos constantemente esto se realiza por medio del método “*pox.update()*” que está ingresado en el loop del código de forma que cada ciclo se actualicen los datos. Además se obtienen los datos de BPMs y SpO2 con base en la librería con las funciones “*pox.getHeartRate()*” y “*pox.getSpO2*”.

Además es fundamental establecer la corriente del diodo rojo, esto se realiza por medio de “*pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA)*”.

```

void loop()
{
    pox.update();
    BPM = pox.getHeartRate();
    SpO2 = pox.getSpO2();

    pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
}

```

Lectura de Temperatura Corporal

La lectura de temperatura corporal se obtiene por medio del sensor LM35 que posee una salida analógica, de esta manera se asigna un puerto analógico del microcontrolador para realizar la lectura, en este caso se define el puerto Analógico 1 “*Lm35=A1*” además se define el puerto como una entrada “*pinMode(Lm35, INPUT)*”.

```

int AirQ=A0, Lm35=A1;
float tempC;

void setup()
{
    pinMode(AirQ, INPUT);
    pinMode(Lm35, INPUT);
}

```

El acondicionamiento de la lectura se realiza con la asignación de esta lectura analógica a una variable de tipo flotante “*tempC = analogRead(Lm35)*” y la conversión de mili volts a grados centígrados por medio de “*tempC = (5.0*tempC*100.0)/1024.0*”

```
float Aire = analogRead(AirQ);
tempC = analogRead(Lm35);
tempC = (5.0 * tempC * 100.0)/1024.0;
```

Lectura de Calidad del Aire

La lectura de la calidad del aire se obtiene en base al número de partículas por millón presentes en el aire, estos valores obtenemos por medio del sensor MQ135 que posee una salida analógica, para esto se asigna un puerto analógico del microcontrolador de forma que se realice esta lectura, en este caso el puerto analógico A0 “*AirQ=A0*”, además se establece el puerto A0 como puerto de entrada “*pinMode(AirQ, INPUT)*”.

```
int AirQ=A0, Lm35=A1;

void setup()
{
    pinMode(AirQ, INPUT);
    pinMode(Lm35, INPUT);
}
```

La lectura del puerto analógico se tendrán valores de enteros, así se puede hacer un análisis para determinar a qué cantidad de ppm se considera que el aire está expuesto a factores de contaminación o que involucren riesgo para el paciente.

```
float Aire = analogRead(AirQ);
```

Envío de datos de forma serial

Una vez que se han adquirido los datos de los sensores con los parámetros vitales frecuencia cardiaca, saturación de oxígeno en la sangre y temperatura además de la calidad del aire, para el envío de los mismos de forma serial se realiza una impresión serial “*Serial.print()*” de cada dato con un separador entre cada dato, en este caso se utiliza la coma “*Serial.print(‘, ’)*”

```

void loop()
{
  pox.update();
  BPM = pox.getHeartRate();
  SpO2 = pox.getSpO2();
  float Aire = analogRead(AirQ);
  tempC = analogRead(Lm35);
  tempC = (5.0 * tempC * 100.0)/1024.0;

  //Serial.print("BPM: ");
  Serial.print(BPM);
  Serial.print(',');

  //Serial.print("SpO2: ");
  Serial.print(SpO2);
  Serial.print(',');

  Serial.print(tempC);
  Serial.print(',');

  Serial.print(Aire);
  Serial.print(',');
}

```

De esta manera se tiene el envío de los datos por comunicación serial para esto se emplea el puerto de transmisión serial del microcontrolador puerto 3 PD1 – TXD.

Impresión de Datos por Pantalla OLED

La impresión de los datos por medio de la pantalla OLED con resolución de 128x64 pixeles se realiza por medio de comunicación I2C, para esto se agrega la librería “*OakOLED.h*” y “*Adafruit_GFX.h*” para la impresión se ha definido una variable “*OakOLED oled*” además se define las dimensiones de la pantalla OLED “*define SCREEN_WIDTH 128*” y “*define SCREEN_HEIGHT 64*”.

```

#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <OakOLED.h>

#define REPORTING_PERIOD_MS    1000
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

```

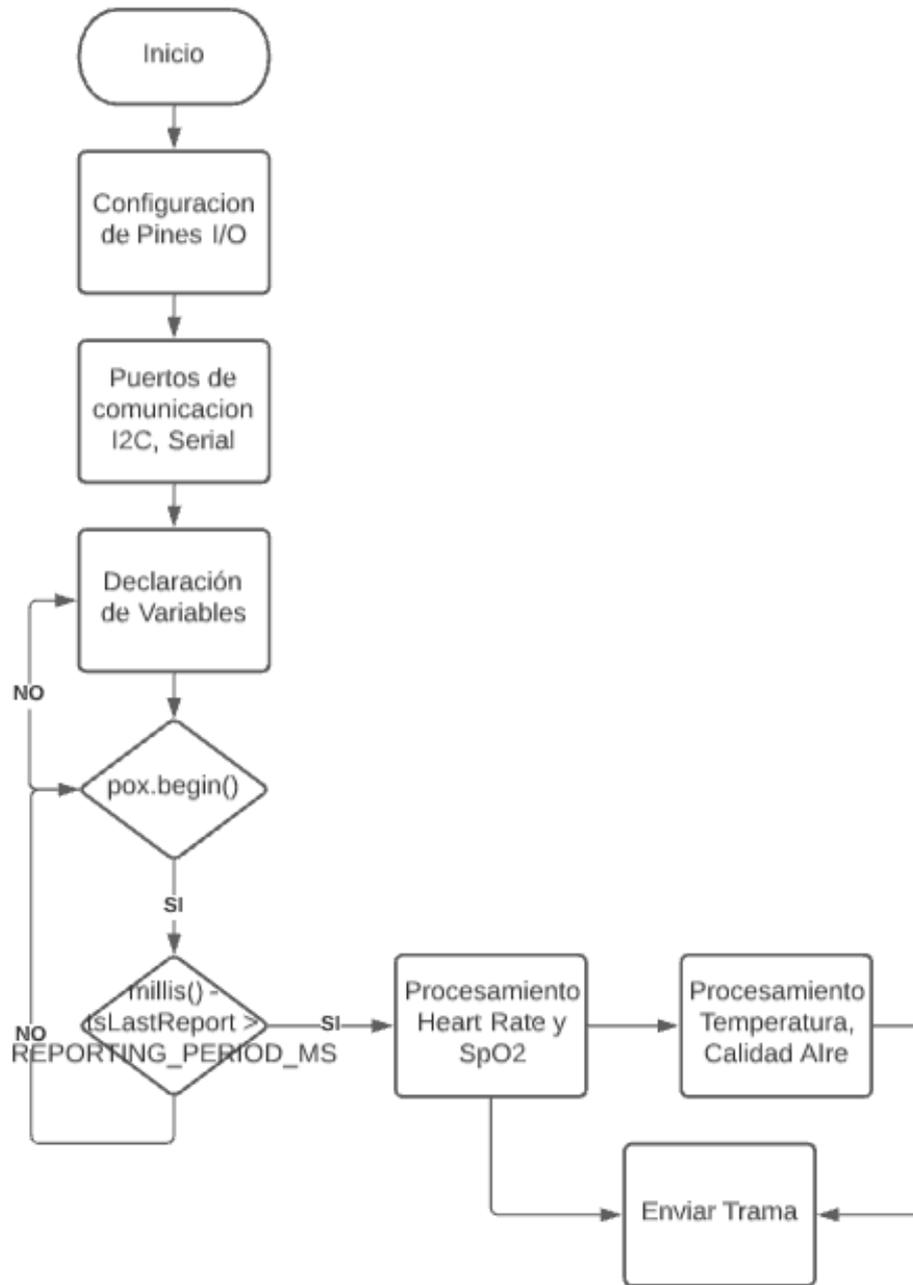



Figura 22. Diagrama de flujo de programación ATMEGA328p
Realizado por Autor

3.1.2.5 Programación de Tarjeta ESP32

Para la programación de la tarjeta ESP32 es necesario conocer el diagrama de conexiones que se poseen para de esta forma identificar los puertos a utilizar, las conexiones quedan de la manera que se muestran en la figura 23.

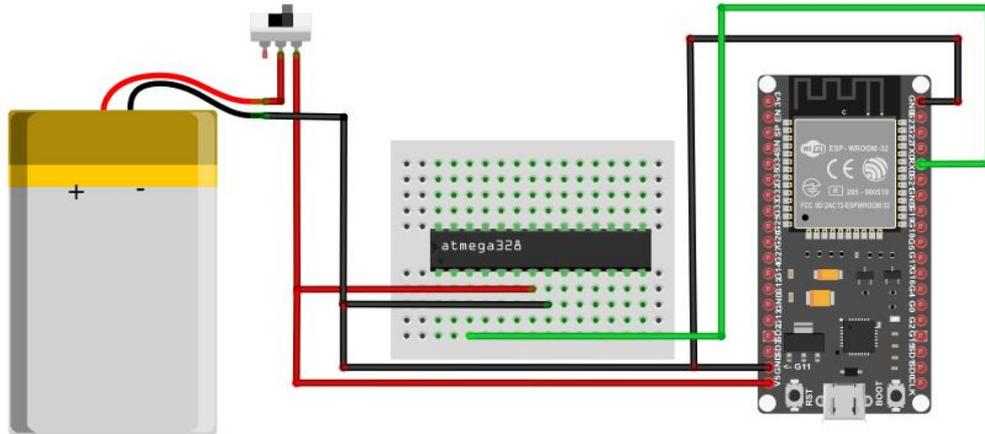


Figura 23. Diagrama de conexiones ESP32.
Realizado por Autor.

Tabla XVI. Conexiones de ESP32.
Realizado por Autor.

Dispositivo	Puerto	Tipo de Comunicación	Descripción
ATMEGA 328P	34	Serial	GIO3 – RX0
Alimentación	19	VCC	Vin 5V
	14, 38	GND	GND

La comunicación Serial se realiza por el puerto 34 GIO3 – RX debido a que la tarjeta ESP 32 forma parte de la capa de intercambio de datos, así recibe los datos provenientes de la capa de sensorización. Estos datos se han enviado por medio de una trama, por esta razón se requiere descomponer esta trama.

```

#define DEBUG_ARRAY(a)
{
  for (int index = 0; index < sizeof(a) / sizeof(a[0]); index++)
  {
    Serial.print(a[index]);
    Serial.print('\t');
  }
  Serial.println();
};

```

Para la recepción de los datos se emplea la clase String, esto para recibir, dividir el array y realizar las conversiones. Se define en primera instancia el separador que se tendrá entre cada dato “*const char separator*”, además se establece la longitud de los datos “*const int dataLength*”.

```

String str = "";
const char separator = ',';
const int dataLength = 4;
float data[dataLength];

```

Se constata que exista comunicación serial, “*Serial.available*”, así se procede a la descomposición del array identificando la longitud de los datos y el separador para poder extraer cada dato, de esta manera obtener los datos de los sensores enviados desde el microcontrolador.

```

void loop()
{

  if (Serial.available())
  {
    str = Serial.readStringUntil('\n');
    for (int i = 0; i < dataLength ; i++)
    {
      int index = str.indexOf(separator);
      data[i] = str.substring(0, index).toFloat();
      str = str.substring(index + 1);
    }
  }

  BPM = data[0];
  SpO2 = data[1];
  tempC = data[2];
  Aire = data[3];

```

Para la aplicación IoT el envío de datos se realiza por medio de conexión WiFi, la tarjeta ESP32 cuenta con el estándar de comunicación IEEE 802.11. Para esto se emplea la librería “WiFi.h”, se establece las credenciales de la red a la cual se realizará la conexión por medio de “char ssid[]” y “char pass[]”, estas credenciales están establecidas en el archivo “secrets.h”. Con las credenciales se inicializa la librería y se establece un bucle while para determinar si la tarjeta se ha conectado con la red “WiFi.begin” y “while (WiFi.status() != WL_CONNECTED)”.

```

|
#define SECRET_SSID "RED PIAZA"
#define SECRET_PASS "BLACKCM94@"

#define SECRET_CH_ID 1387476
#define SECRET_WRITE_APIKEY "P4FV7CSCA3I4SQRX"

#define SECRET_SHA1_FINGERPRINT "27 18 92 DD A4 26 C3 07 09 B9 7A E6 C5 21 B9 5B 48 F7 16 E1"

void setup()
{
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.begin(115200);

    while (WiFi.status() != WL_CONNECTED)
    {
      delay(1000);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi OK");
  }

  if(fingerprint!=NULL){
    client.setFingerprint(fingerprint);
  }
  else{
    client.setInsecure(); // To perform a simple SSL Encryption
  }

  ThingSpeak.begin(client); // Initialize ThingSpeak
}

```

Para la aplicación IoT se emplea la plataforma ThingSpeak para esto se realiza la escritura de los datos adquiridos para su monitoreo en la plataforma, por otro lado es necesario establecer el código del canal de la plataforma y la API Key del canal, datos que se encuentran en la plataforma ThingSpeak.

The screenshot shows the 'Write API Key' page in the ThingSpeak interface. At the top, there are navigation links: 'Private View', 'Public View', 'Channel Settings', 'Sharing', and 'API Keys'. The channel name 'TELEMEDICINA IOT ACM' is displayed prominently. Below it, the channel details are listed: 'Channel ID: 1387476', 'Author: mwa0000022518953', and 'Access: Private'. On the right side, the 'Write API Key' section is active, showing a 'Key' field with the value 'P4FV7CSCA3I4SQRX' entered in a text box.

Información que se ingresará en el código de la tarjeta ESP32 en el archivo de seguridad SSL “*secrets.h*”. Se emplea la librería “*ThingSpeak.h*”, para esto en primera instancia se definen los datos del canal de la plataforma.

```
#define SECRET_SSID "RED PIAZA"
#define SECRET_PASS "BLACKCM94@"

#define SECRET_CH_ID 1387476
#define SECRET_WRITE_APIKEY "P4FV7CSCA3I4SQRX"

#define SECRET_SHA1_FINGERPRINT "27 18 92 DD A4 26 C3 07 09 B9 7A E6 C5 21 B9 5B 48 F7 16 E1"
```

Una vez que se tienen los datos de los sensores en la tarjeta ESP32 se deben escribir estos datos en los Field’s creados en el canal con sus respectivos nombres, esto se realiza con la función “*ThingSpeak.setField()*” así cada uno de los datos son escritos con su nombre y posición “*1, BPM*”. Si los datos son escritos correctamente en el canal se cerrarán los datos con un mensaje de datos actualizados, caso contrario se intentará volver a escribir los datos cada 5 segundos.

```
ThingSpeak.setField(1, BPM);
ThingSpeak.setField(2, SpO2);
ThingSpeak.setField(3, tempC);
ThingSpeak.setField(4, Aire);

int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
if(x == 200){
  Serial.println("Channel update successful.");
}
else{
  Serial.println("Problem updating channel. HTTP error code " + String(x));
}
delay (5000);
```

3.1.2.6 Diseño de circuito electrónico

EL diseño del circuito electrónico permite que todos los componentes del sistema se interconecten, de manera que se tenga la medición y visualización de los datos de BPM, SpO2, Temperatura Corporal y Calidad del aire. Este diseño del circuito impreso se realizará en el software Proteus Profesional en la versión 8.8.

El proyecto en Proteus permite en primera instancia realizar la conexión de los elementos esto se realiza en la sección de “Schematic Capture”. La figura 24 muestra las conexiones que se tiene entre cada componente del circuito antes de pasar al diseño del circuito impreso.

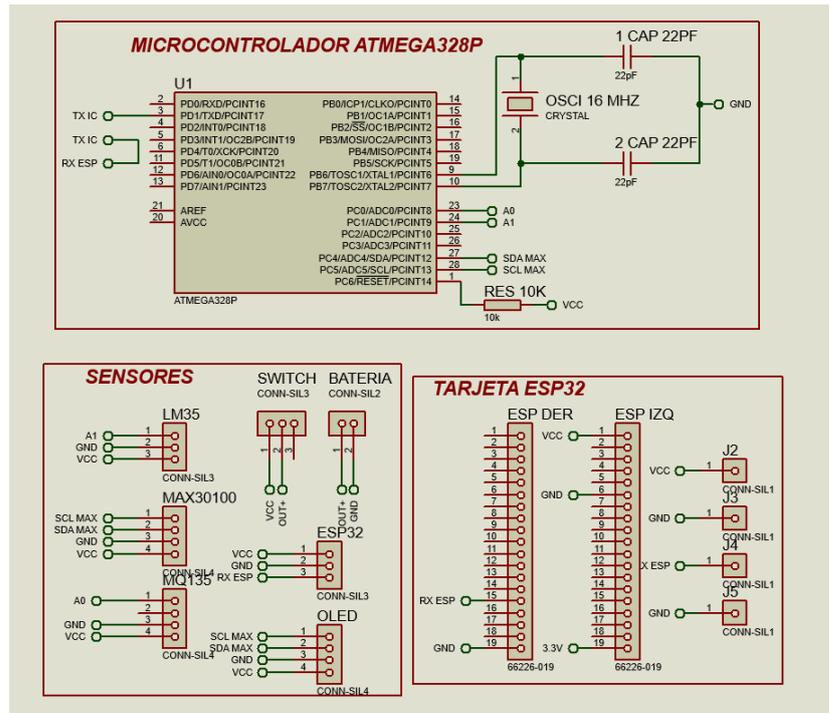


Figura 24. Diagrama de conexión de componentes del circuito eléctrico
Realizado por Autor.

Una vez que se cuente con el diagrama de conexiones se puede pasar a la sección de PCB Layout donde realizaremos el diseño de la PCB, para esto se define las características de diseño.

Tabla XVII. Características de diseño de placa de circuito impreso.\n
Realizado por Autor.

Parámetro	Valor
Ancho de pistas	70th - 1.778 mm
Agujeros de perforación de componentes	80th – 40th diámetro interno y externo 2.032mm - 1.016mm
Placa	50 mm x 50mm

Con las características de diseño definidas, la figura 25 muestra el diseño final de la PCB.

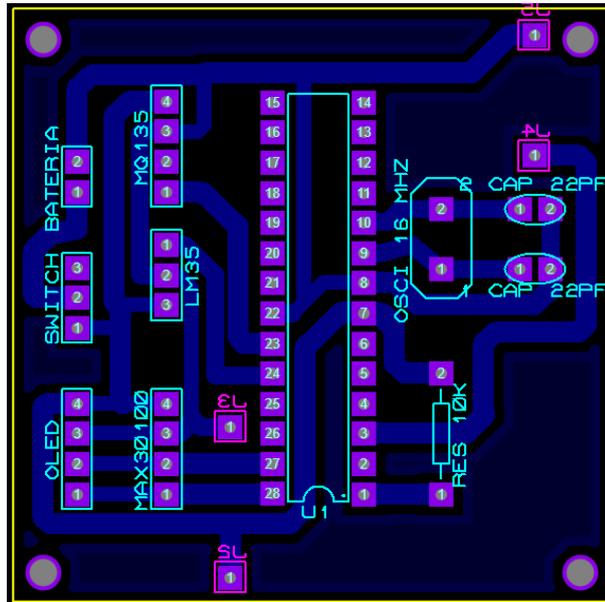


Figura 25. Diseño de PCB del sistema.
Realizado por Autor.

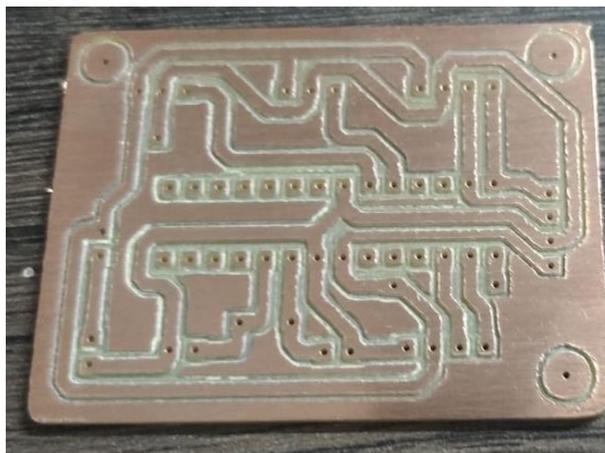


Figura 26. PCB final del prototipo.
Realizado por Autor.

3.1.2.7 Ubicación de sensores

Un factor fundamental para la medición de las constantes vitales por medio de los sensores mencionados, es la ubicación de dichos sensores dentro del sistema de telemedicina, puesto que la ubicación determina la calidad de la lectura de los datos.

El análisis realizado en los apartados 3.1.2.1 y 3.1.2.2 aclaran la ubicación de los sensores de forma que se tengan lecturas adecuadas de las constantes de los pacientes, siendo estas ubicaciones mostradas en la figura 27.

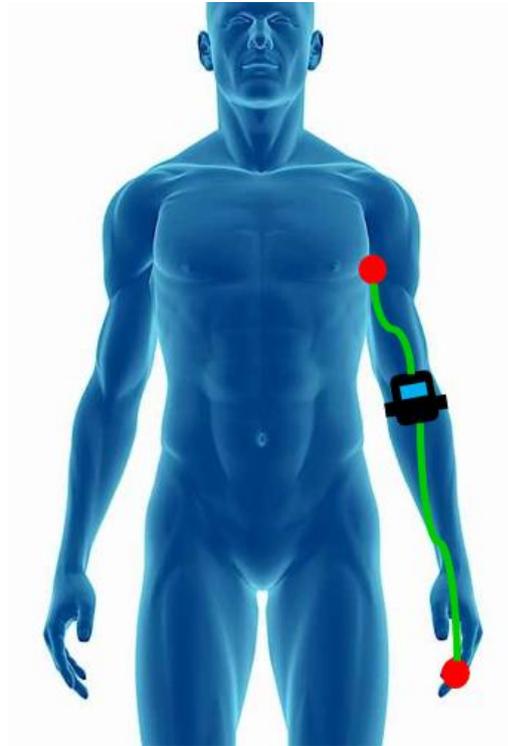


Figura 27. Ubicación de sensores en el cuerpo del paciente.
Realizado por Autor

Específicamente cada uno de los sensores cumple su función de adquisición de los valores de parámetros vitales de acuerdo a su ubicación.

3.1.2.8 Diseño de prototipo 3D

El diseño del prototipo 3D se realizará en el software “Solid Works” con el uso de las herramientas del programa que permite generar un archivo STL Estereolitografía para su impresión en 3D.

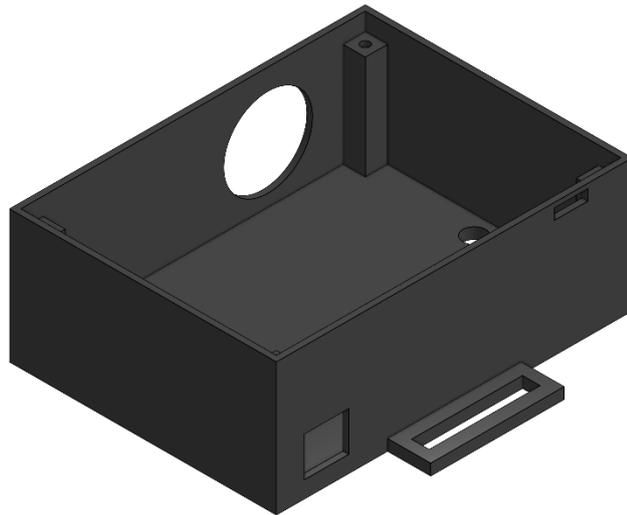


Figura 28. Diseño 3D - Estructura de carcasa del prototipo.
Realizado por Autor.

El prototipo asemeja una pulsera considerando esto y el tamaño del circuito impreso, la carcasa tiene las dimensiones como se muestra en la figura 29.

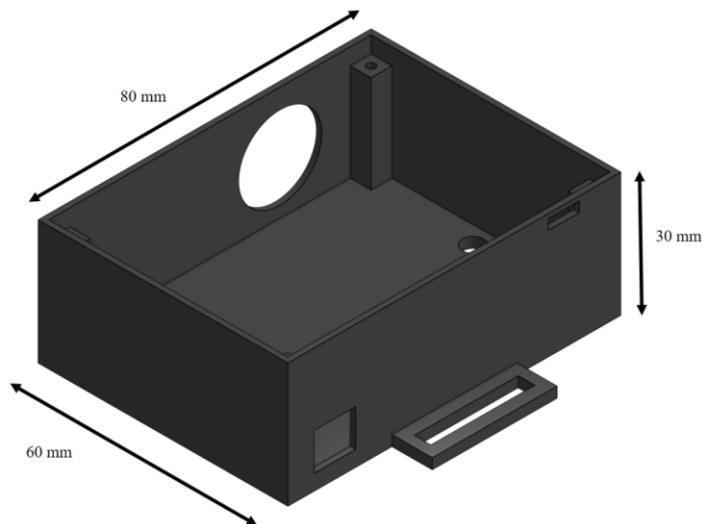


Figura 29. Diseño 3D - Dimensiones de carcasa de prototipo.
Realizado por Autor.

La carcasa posee aberturas para conexiones de los sensores, switch de encendido y apagado y pestaña de sujeción de la tapa, en la figura 30 y figura 31 se muestra la carcasa con las aberturas.

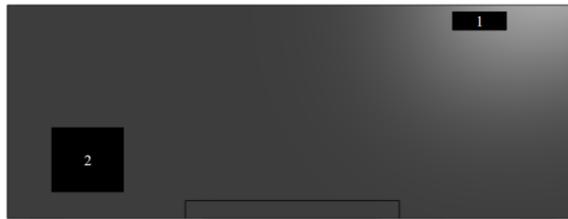


Figura 30. Diseño 3D - Vista lateral 1 de la carcasa con sus aberturas.
Realizado por Autor.

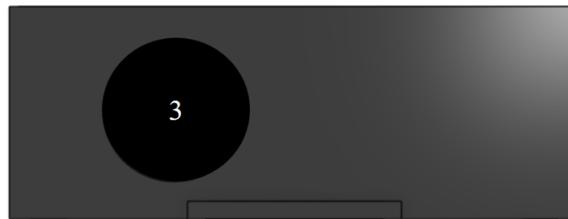


Figura 31. Diseño 3D - Vista lateral 2 de la carcasa con sus aberturas.
Realizado por Autor.

Las aberturas de la carcasa:

1. Abertura para pestaña de sujeción de tapa
2. Abertura de switch de encendido
3. Abertura de sensor MQ135

Para contar con correas para la colocación de la pulsera se tiene una base a cada lado de la carcasa.

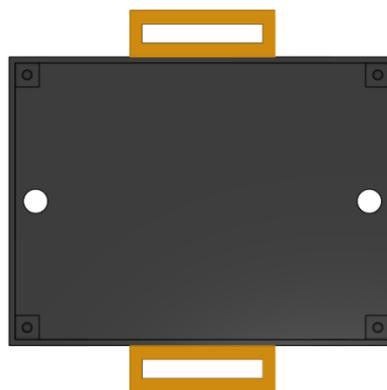


Figura 32. Diseño 3D - Vista superior de bases para correas.
Realizado por Autor.

Esta carcasa es un módulo rectangular hueco, la tapa de la misma posee las dimensiones mencionadas en la figura 33.

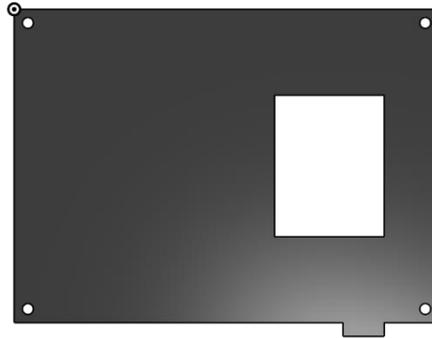


Figura 33. Diseño 3D - Vista de tapa de carcasa.
Realizado por Autor.

La tapa cuenta con abertura para la pantalla OLED, además cuenta con una pestaña para la sujeción a la carcasa.

Una vez que las piezas se han diseñado e impreso se puede armar la pulsera con correas en las bases de sujeción, los sensores se colocan en cada extremo de acuerdo a la conexión del circuito tanto para el MAX30100 como para el LM35, así se verá como se muestra en la figura 34.



Figura 34. Estructura final del prototipo.

Realizado por Autor.

3.1.3 Implementación de aplicación IoT en plataforma ThingSpeak.

3.1.3.1 Plataforma ThingSpeak

Para la implementación de la aplicación IoT se seleccionó la plataforma ThingSpeak esto tras el análisis de las plataformas IoT que manejan el Cloud Server como se menciona en el apartado 1.2.3.2. La plataforma ThingSpeak se empleará para el almacenamiento y visualización de los datos de parámetros vitales del paciente, para esto se requiere una cuenta de usuario en la plataforma, aquí se empieza con seguridad de SingUp, la pantalla de ingreso se muestra en la figura 35.

Al ingresar a la plataforma podremos empezar con el desarrollo de los canales y el enlace de aplicaciones que posee la plataforma, esto lo podemos encontrar en el índice de la plataforma como se muestra en la figura 36.

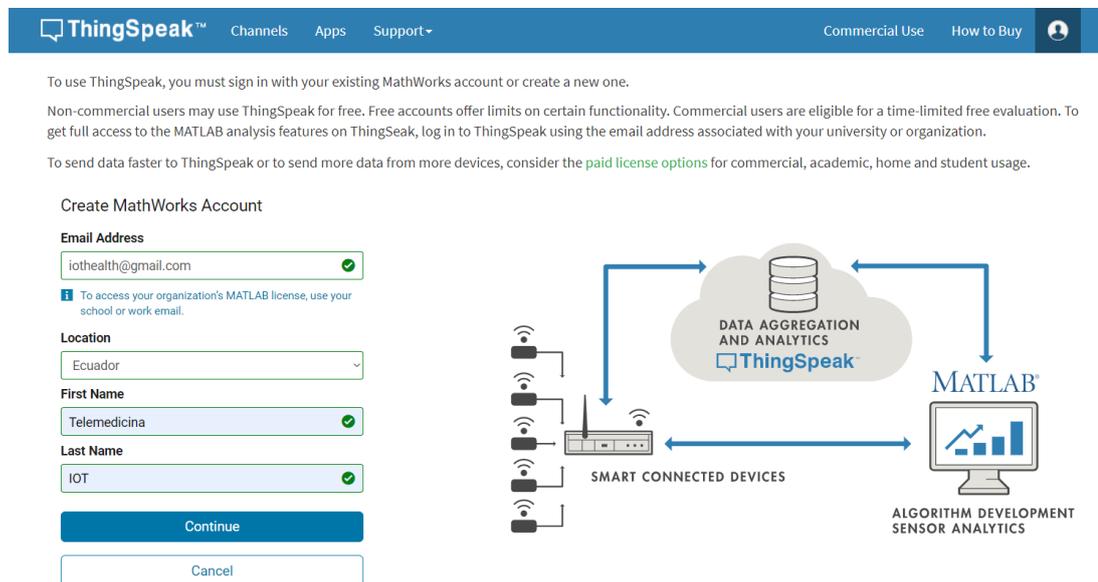


Figura 35. ThingSpeak ingreso a plataforma
Fuente [25]



Figura 36. Índice de Plataforma ThingSpeak
Fuente: [25].

3.1.3.2 Creación de Canales en ThingSpeak

Los Canales “Channels” en ThingSpeak son los componentes en los cuales se almacenan los datos que se envían a la plataforma. Para poder leer o escribir en un canal, la capa de intercambio de datos envía las solicitudes al Cloud Server de ThingSpeak esto mediante la emisión de solicitudes HTTP. Cada canal de ThingSpeak puede tener hasta ocho campos de datos de hasta 255 caracteres en formato numérico o alfanumérico. Además, el canal puede tener información como ubicación y un campo de actualización de estado como se muestra en la figura 37. Cada entrada de actualización de datos del canal se almacena con una fecha y una marca de tiempo.

The screenshot shows the 'New Channel' page in the ThingSpeak web interface. The top navigation bar includes 'Channels', 'Apps', 'Support', 'Commercial Use', and 'How to Buy'. The main content area is split into two columns. The left column, titled 'New Channel', contains a form with the following elements: a 'Name' text input, a 'Description' text area, eight 'Field' entries (Field 1 to Field 8), each consisting of a text input and a checkbox, a 'Metadata' text area, and a 'Tags' text area. The right column, titled 'Help', contains a 'Channel Settings' section with a list of instructions: 'Percentage complete', 'Channel Name', 'Description', 'Field', 'Metadata', 'Tags', 'Link to External Site', 'Show Channel Location' (with sub-points for Latitude, Longitude, and Elevation), and 'Video URL'.

Figura 37. Panel de creación de Canales en Plataforma ThingSpeak.
Fuente [25]

Para la aplicación IoT del sistema de telemedicina para el monitoreo de pacientes con enfermedades respiratorias se considerado la creación de un canal con cuatro campos de datos, considerando el análisis de los parámetros a ser monitorizados en pacientes con enfermedades respiratorias en el apartado 1.2.3.5. teniendo así el Canal “TELEMEDICINA IOT ACM” con los campos de:

- Field 1: BPM Frecuencia Cardiaca [BPM]
- Field 2: SPO2 Saturación de Oxígeno en la Sangre [%]
- Field 3: TEMP CORP Temperatura Corporal [° C]
- Field 4: AIR Calidad del Aire [ppm]

Los nombres de cada “Field” son considerados a la hora de la programación del dispositivo de la capa de intercambio de datos en el apartado 3.1.2.5. puesto que para la escritura de los datos de los sensores se debe asignar un Field determinado, así se evidencia en la figura 38, que muestra la configuración final del Canal.

ThingSpeak™ Channels Apps Support

Channel Settings

Percentage complete 30%

Channel ID

Name

Description

Field 1

Field 2

Field 3

Field 4

Field 5

Figura 38. Configuración de Canal en ThingSpeak para el sistema de telemedicina. Realizado por Autor. Fuente [25]

Algo fundamental en el desarrollo de la aplicación IoT para el monitoreo es la privacidad de los datos del paciente, por lo que el canal creado se desarrolla como un Canal Privado “Private View” sin compartir el canal de forma pública a manera de seguridad.

Otro factor fundamental en cuanto a seguridad del canal por lo tanto en la privacidad de datos, es la asignación de una contraseña especial que se asigna a cada canal “API KEY” además esto es con el fin de que los datos se almacenen en el canal correcto. La figura 39 muestra que existe una contraseña especial para escritura de datos “WRITE API KEY” y para lectura “READ API KEY”, además en esta ventana se tiene “API REQUEST” que presenta las sintaxis de peticiones de API para la integración de plataformas o servicios externos, esto se muestra en la figura 40.

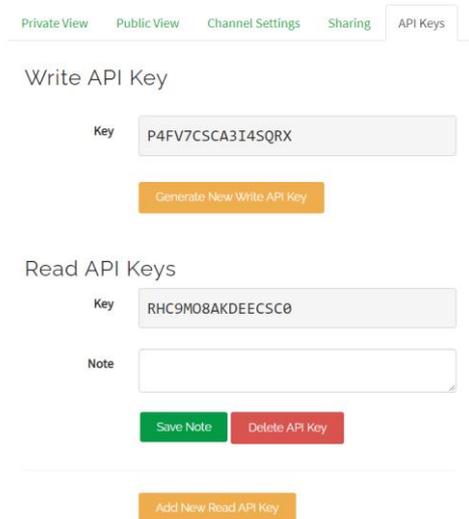


Figura 39. Plataforma ThingSpeak, API Keys para la lectura o escritura de datos. Realizado por Autor. Fuente [25]

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=P4FV7CSCA3I4SQRX&field=
```

Read a Channel Feed

```
GET https://api.thingspeak.com/channels/1387476/feeds.json?api_key=
```

Read a Channel Field

```
GET https://api.thingspeak.com/channels/1387476/fields/1.json?api_key=
```

Read Channel Status Updates

```
GET https://api.thingspeak.com/channels/1387476/status.json?api_key=
```

Figura 40. Plataforma ThingSpeak modelos de sintaxis de pedidos de API. Realizado por Autor. Fuente [25]

Una vez que se ha creado el canal con las configuraciones de campos de acuerdo a los requerimientos del sistema prototipo, y se ha programado la escritura de los datos de los sensores de parámetros vitales del paciente se puede empezar la visualización de los datos, la primera pantalla muestra las graficas de Datos vs Tiempo.

3.1.3.3 Diseño de Interfaz en ThingSpeak

Para el diseño de la presentación de los datos, se cuenta con la opción de mostrar los datos por medio de gráficas de tiempo, por otro lado, se puede agregar “*Widgets*” que pueden ser “*Gauges*”, “*Numeric Display*” y “*Lamp Indicator*”, cada uno de los cuales se puede asociar a los “*Field*”.

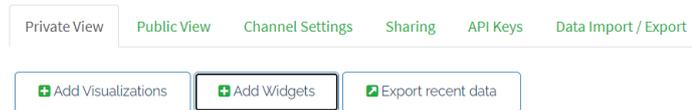


Figura 41. Opciones de modificación de interfaz de presentación de Datos.
Realizado por Autor.

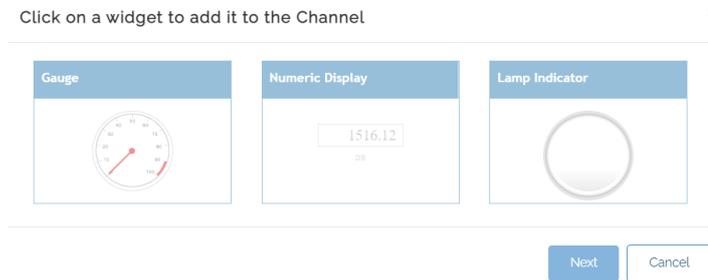


Figura 42. Opciones de Widgets para la interfaz de presentación de los Datos.
Realizado por Autor.

Una vez seleccionado el Widget acorde al dato que se desee representar, se configura los parámetros del Widget a fin de contar con una correcta visualización y representación de los datos.

Para el Widget “*Gauge*” tiene parámetros:

- Nombre
- Field
- Valores Máximo y Mínimo
- Visualizar el Valor
- Unidades
- Intervalos de Actualización
- Rangos

Para el Widget “*Numeric Display*” se tiene parámetros como:

- Nombre
- Field
- Intervalo de Actualización
- Unidades
- Tipo de Dato (Decimal, Integral)

Finalmente, el Widget “*Indicator Lamp*”

- Nombre
- Condicion If
- Intervalo de Actualización

Así una vez asociados los Field con cada Widget para su representación, tendremos la interfaz como se muestra:



Figura 43. Interfaz de visualización de BPM y SpO2.
Realizado por Autor.

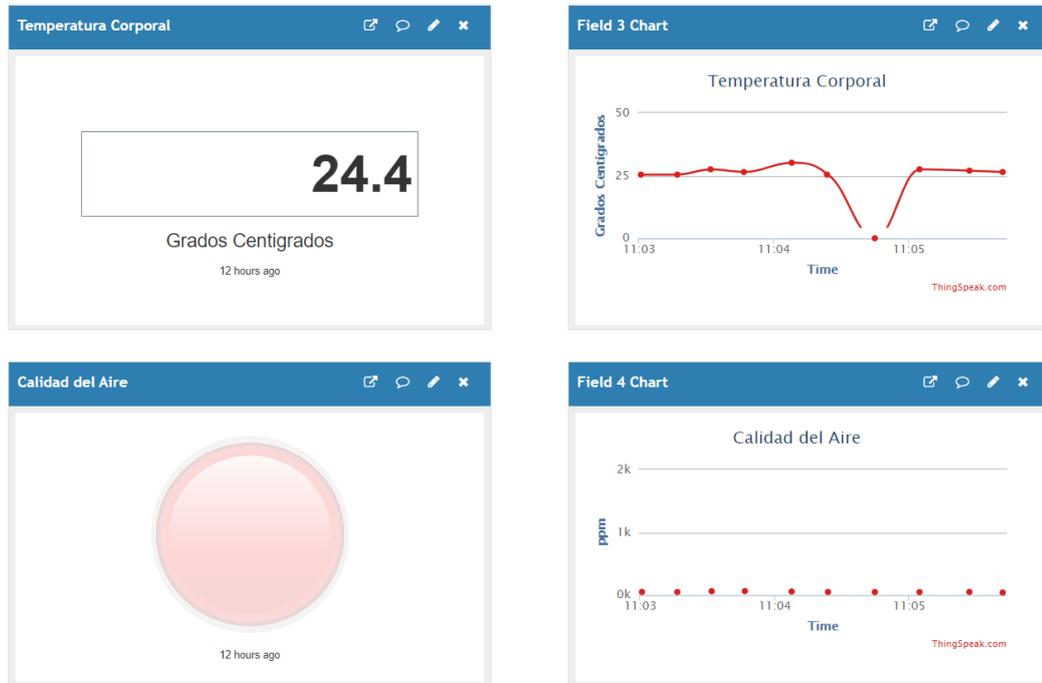


Figura 44. Interfaz de visualización de Temperatura y Calidad del Aire.
Realizado por Autor.

3.1.3.4 Adaptación de IFTTT – App MatLab Analysis y TimeControl

IFTTT es un servicio web que permite crear applets que son programas que se pueden incrustar en formato HTML, para que estos applets actúen en respuesta de una acción. Este servicio web permite el uso de WebHooks donde se puede crear solicitudes web para desencadenar acciones. Puede ser que la acción entrante sea una solicitud HTTP al servidor web y tenga una acción resultante como un mensaje o una asignación de hojas de cálculo de Google.

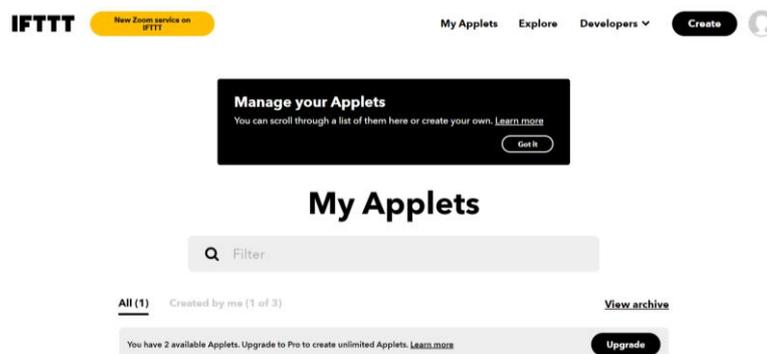


Figura 45. IF THIS THEN THAT ventana principal.
Realizado por Autor.

En la página de IFTTT crearemos los Applets, iremos a “*My Applets*” y a “*New Applet*”, ahora tenemos la opción de configuración de acciones y condiciones del servicio If This Then That (IFTTT), En principio se selecciona el servicio, en este caso “*Webhooks*”.



Figura 46. IFTTT - Creación de Applet
Realizado por Autor.

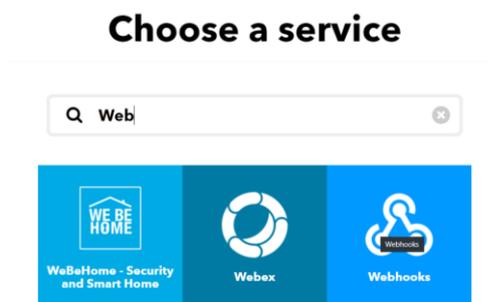


Figura 47. IFTTT - Selección de servicio Webhooks
Realizado por Autor.

Webhooks posee características de ser una solicitud de HTTP POST que tendrá una reacción en dependencia de acción inicial, es empleada en notificaciones de tiempo real, por esta razón es el servicio a seleccionar.

Al seleccionar el servicio Webhooks, corresponde escoger un “*Trigger*” en este servicio contamos con “*Receive a web request*” que permite su activación cada vez que el servicio recibe una solicitud web para notificar un evento. En el Trigger se completará el nombre del evento “*Event Name*”.

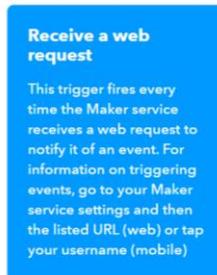


Figura 48. IFTTT - Descripción de Trigger Webhooks.
Realizado por Autor.

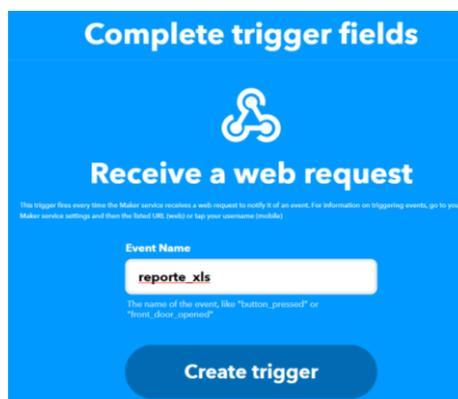


Figura 49. IFTTT - Configuración de Trigger.
Realizado por Autor.

Ahora corresponde la configuración de la acción que ocurrirá respecto al evento, esto en la opción *“that”*. Aquí se selecciona el servicio de *“Google Sheets”*. En este servicio corresponde configurar el nombre de la *“Spreadsheet name”* y el formato de la tabla con sus celdas en *“Formatted row”*, y finalmente la carpeta de destino del archivo.

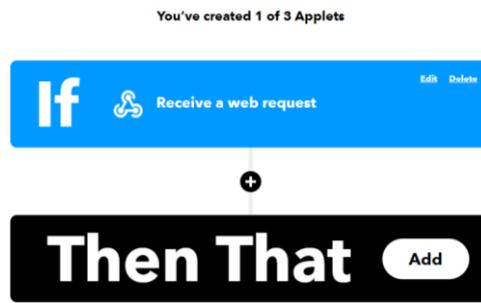


Figura 50. IFTTT - Configuración de acción "Then That".
Realizado por Autor.

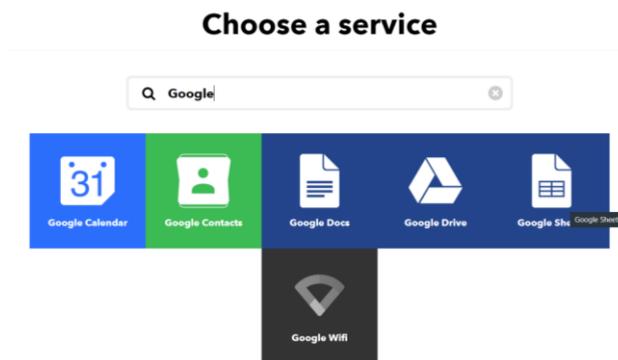


Figura 51. IFTTT - Selección de Servicio Google Sheets.
Realizado por Autor.

The screenshot shows the 'Add row to spreadsheet' configuration form. It includes fields for 'Spreadsheet name' (IFTTT_Maker_Webhooks_Events), 'Formatted row' (OccuredAt || EventName || Value1 || Value2 || Value3), and 'Drive folder path' (IFTTT/MakerWebhooks/EventName). Each field has an 'Add ingredient' button. A 'Create action' button is at the bottom.

Figura 52. IFTTT - Configuración de celdas de Google Sheets.
Realizado por Autor.

Ahora ya contamos con el servicio de "Webhooks" y necesitamos la URL del evento configurado.

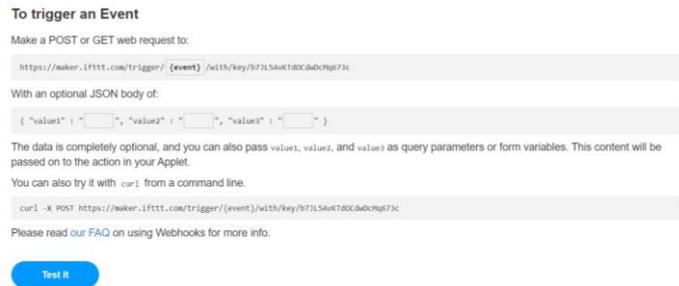


Figura 53. IFTTT - URL de evento configurado en el servicio Webhooks. Realizado por Autor.



Figura 54. IFTTT - Evento configurado y funcionando. Realizado por Autor.

En la plataforma “*ThingSpeak*” se necesita crear la aplicación “*Create a MATLAB Analysis*”, esto se realiza en la pestaña de “*Apps*”.



Figura 55. ThingSpeak - Creación de Apps. Realizado por Autor.

Al empezar a crear la App se cuenta con la opción de plantillas de código para análisis de datos en MatLab, podremos seleccionar “*Trigger Email from IFTTT*” código que contamos en el Anexo M, cabe recalcar que las plantillas de código son propias de la plataforma. Cambiamos el nombre del análisis y guardamos.

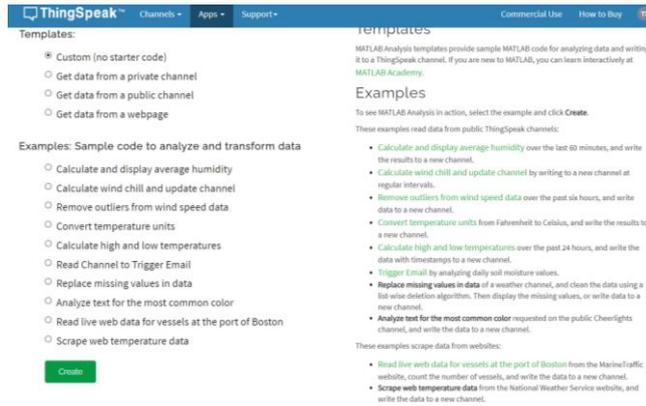


Figura 56. ThingSpeak - Configuración de App Analysis MATLAB. Realizado por Autor.

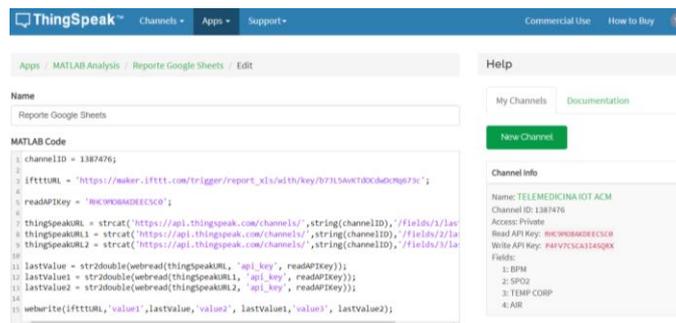


Figura 57. ThingSpeak - Escritura de código de para Analysis MATLAB. Realizado por Autor.

Ahora para establecer un intervalo de tiempo para generar los reportes se crea una “App” esto lo realizamos en la Aplicación “TimeControl”, aquí deberemos configurar la App.

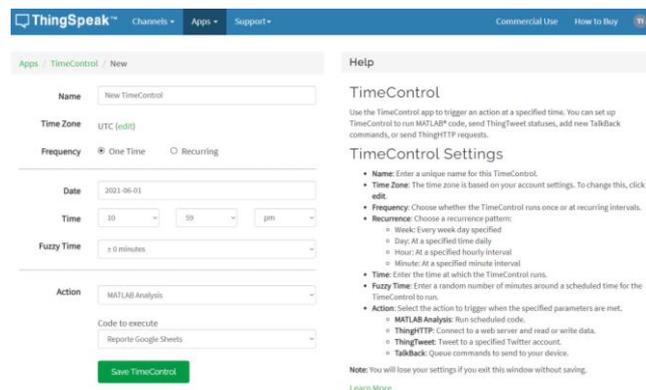


Figura 58. ThingSpeak - Configuración de App Time Control. Realizado por Autor.

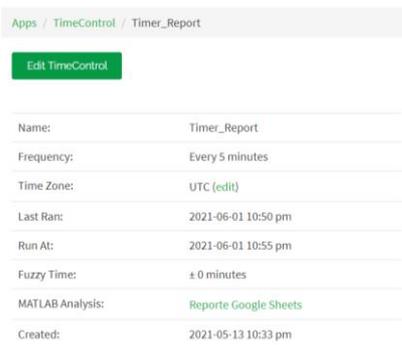


Figura 59. ThingSpeak - Verificación de App Time Control.
Realizado por Autor.

Una vez que se haya configurado la Applet en IFTTT y el Análisis de Datos en Matlab de ThingSpeak, el reporte configurado para Google Sheets se crea y actualiza en una carpeta de Drive de la cuenta de Google enlazada con IFTTT.

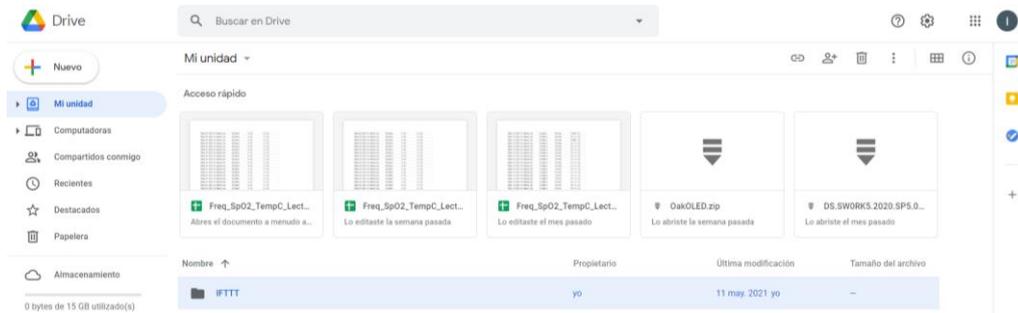


Figura 60. Google Drive - Comprobación de carpeta generada por servicio IFTTT.
Realizado por Autor.

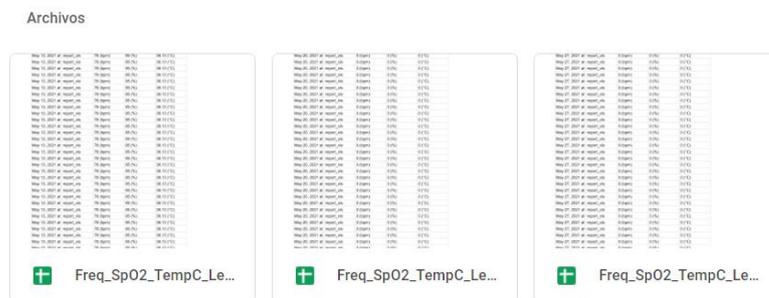


Figura 61. Google Drive - Google Drive - Comprobación de archivo generado por servicio IFTTT.
Realizado por Autor.

Freq_SpO2_TempC_Lecturas			
Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda			
43	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
44	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
45	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
46	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
47	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
48	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
49	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
50	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
51	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
52	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
53	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
54	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
55	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
56	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
57	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)
58	May 13, 2021 at report_xls	78 (bpm)	95 (%) 36.13 (°C)

Figura 62. Google Drive Sheets - Google Drive - Comprobación de Datos generados por servicio IFTTT.
Realizado por Autor.

3.1.4 Implementación final del prototipo

El prototipo final implementado en un paciente se estructura como se muestra en la figura 63.



Figura 63. Implementación del Prototipo Final.

Realizado por Autor.

Donde cada uno de sus componentes:

1. Brazaletes Controlador del prototipo
2. Sensor de Frecuencia Cardíaca y Saturación de Oxígeno
3. Sensor de Temperatura.

3.1.5 Pruebas de funcionamiento

La comprobación de funcionamiento del prototipo se realiza por medio de pruebas prácticas del prototipo funcional, a fin de que el mismo cumpla con los requisitos y características requeridas para un control adecuado y preciso de las constantes vitales del paciente a monitorear. Para la comprobar el funcionamiento se ha utilizado como referencia para la lectura de Frecuencia Cardíaca y Saturación de Oxígeno a un “Oxímetro de Pulso OxiMED” y “Aplicación Android Pulsómetro” y para la medición de temperatura corporal un “Termómetro axilar convencional” y un “Termómetro infrarrojo”.

Con las pruebas y comparaciones de las lecturas se evalúa el funcionamiento del prototipo en comparación con dispositivos actuales. Para obtener un valor aproximado entre las lecturas y el error porcentual entre las mediciones se utiliza las ecuaciones:

$$|\text{Valor Exacto} - \text{Valor Real}| = \text{Error absoluto}$$

$$\frac{\text{Valor Exacto} - \text{Valor Promedio}}{\text{Valor Exacto}} * 100\% = \text{Error relativo}$$

3.1.5.1 Prueba de Frecuencia Cardíaca

El prototipo se coloca en el paciente como se describe en el Apartado 3.1.4 quedando como muestra la figura 64, de lo cual se obtuvieron los datos detallados en la Tabla XVIII.

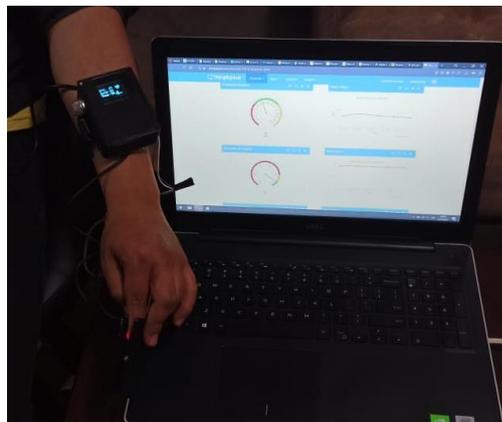


Figura 64. Implementación de Prototipo - Lectura de Frecuencia Cardíaca.

Realizado por Autor.

Tabla XVIII. Muestras de prueba de Frecuencia Cardiaca.

Realizado por el Autor.

#	BPM's Oxímetro	BPM's Pulsómetro App	BPM'S Prototipo	Error Absoluto	Error Relativo
1	69	70	71	2	2,899
2	69	69	68	1	1,449
3	69	70	70	1	1,449
4	69	70	70	1	1,449
5	69	69	69	0	0,000
6	68	68	68	0	0,000
7	68	67	66	2	2,941
8	68	69	70	2	2,941
9	68	68	67	1	1,471
10	68	69	70	2	2,941
11	70	71	71	1	1,429
12	70	70	70	0	0,000
13	70	70	69	1	1,429
14	70	70	69	1	1,429
15	70	71	71	1	1,429
16	71	71	71	0	0,000
17	71	72	72	1	1,408
18	71	70	69	2	2,817
19	71	71	71	0	0,000
20	71	71	70	1	1,408
21	69	69	69	0	0,000
22	69	70	70	1	1,449
23	70	70	69	1	1,429
24	71	71	70	1	1,408
25	70	70	69	1	1,429
26	70	70	70	0	0,000
27	69	69	69	0	0,000
28	70	70	70	0	0,000
29	70	70	70	0	0,000
30	81	81	81	0	0,000
31	81	81	81	0	0,000
32	81	81	80	1	1,235
33	81	81	80	1	1,235
34	81	81	81	0	0,000

35	81	81	81	0	0,000
36	69	69	68	1	1,449
37	69	69	69	0	0,000
38	70	71	71	1	1,429
39	69	70	71	2	2,899
40	67	67	67	0	0,000
41	67	67	67	0	0,000
42	67	67	66	1	1,493
43	67	67	66	1	1,493
44	67	67	66	1	1,493
45	67	67	67	0	0,000
46	71	71	71	0	0,000
47	71	71	71	0	0,000
48	70	70	70	0	0,000
49	69	69	69	0	0,000
50	70	70	70	0	0,000
51	69	69	69	0	0,000
52	69	70	71	2	2,899
53	70	70	70	0	0,000
54	71	71	71	0	0,000
55	70	70	70	0	0,000
56	71	70	69	2	2,817
57	70	71	71	1	1,429
58	69	70	70	1	1,449
59	69	70	71	2	2,899
60	78	78	77	1	1,282
61	78	78	77	1	1,282
62	78	79	79	1	1,282
63	78	79	79	1	1,282
64	78	79	79	1	1,282
65	78	78	78	0	0,000
66	71	71	70	1	1,408
67	70	71	71	1	1,429
68	71	71	70	1	1,408
69	71	70	69	2	2,817
70	82	82	82	0	0,000
71	82	82	82	0	0,000
72	82	82	82	0	0,000
73	82	83	83	1	1,220
74	82	82	82	0	0,000
75	82	82	82	0	0,000
76	69	70	71	2	2,899

77	71	71	71	0	0,000
78	70	70	69	1	1,429
79	71	71	70	1	1,408
80	70	71	71	1	1,429
81	70	70	70	0	0,000
82	71	71	71	0	0,000
83	70	71	71	1	1,429
84	71	71	70	1	1,408
85	70	70	70	0	0,000
86	70	70	70	0	0,000
87	70	71	71	1	1,429
88	70	71	71	1	1,429
89	70	70	70	0	0,000
90	70	70	70	0	0,000
91	71	71	71	0	0,000
92	69	69	69	0	0,000
93	69	70	70	1	1,449
94	70	71	71	1	1,429
95	70	70	70	0	0,000
96	69	70	70	1	1,449
97	69	70	71	2	2,899
98	69	69	69	0	0,000
99	70	70	69	1	1,429
100	70	71	71	1	1,429
				0,68	0,965

En la figura 65 se muestra que él envió de las lecturas hacia la nube de ThingSpeak se realizó con éxito y que la gráfica muestra las lecturas de un intervalo de 30 minutos, con una actualización cada 15 segundos aproximadamente.

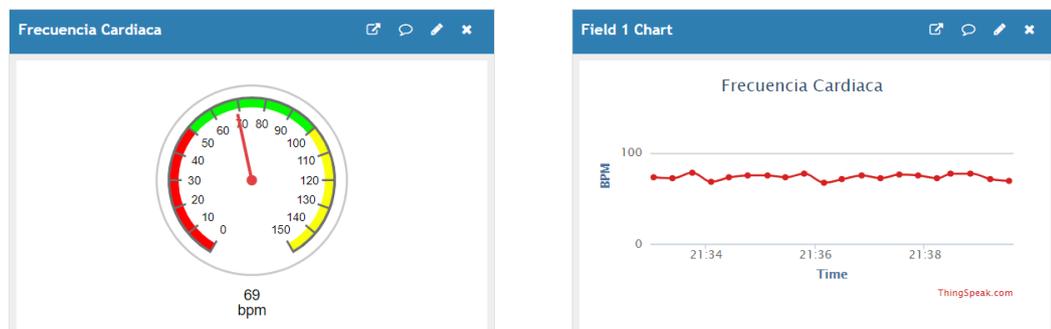


Figura 65. ThingSpeak - Lecturas de Frecuencia Cardiaca.

Realizado por Autor.

Basados en la Tabla de Lecturas de BPM's y empleado las formulas descritas en 3.1.5 se calcula el Valor Promedio y el Error Porcentual, cálculo del cual se ha obtenido los siguientes valores.

- Valor Promedio: 69.3
- Error absoluto: 0.68
- Error relativo: 0.965

3.1.5.2 Prueba de Saturación de Oxígeno

El prototipo se coloca en el paciente como se describe en el apartado 3.1.4 quedando como muestra la figura 66, de lo cual se obtuvieron los datos detallados en la tabla XIX.

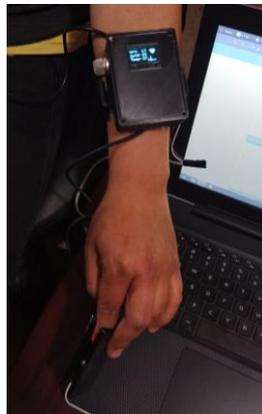


Figura 66. Implementación de prototipo - Lectura de Saturación de Oxígeno.

Realizado por Autor.

Tabla XIX. Muestras de prueba de Saturación de Oxígeno.

Realizado por el Autor.

#	SpO2 Oxímetro	SpO2 Prototipo	Error Absoluto	Error Relativo
1	96	97	1	1,042
2	96	97	1	1,042
3	99	98	1	1,010
4	97	97	0	0,000
5	99	98	1	1,010
6	98	97	1	1,020

7	99	98	1	1,010
8	99	98	1	1,010
9	98	97	1	1,020
10	97	98	1	1,031
11	98	99	1	1,020
12	96	97	1	1,042
13	97	97	0	0,000
14	99	97	2	2,020
15	96	97	1	1,042
16	96	96	0	0,000
17	99	98	1	1,010
18	96	97	1	1,042
19	99	99	0	0,000
20	98	98	0	0,000
21	97	96	1	1,031
22	98	98	0	0,000
23	96	97	1	1,042
24	98	98	0	0,000
25	96	97	1	1,042
26	96	98	2	2,083
27	99	99	0	0,000
28	97	96	1	1,031
29	97	98	1	1,031
30	97	97	0	0,000
31	98	97	1	1,020
32	97	97	0	0,000
33	99	98	1	1,010
34	98	98	0	0,000
35	98	99	1	1,020
36	99	99	0	0,000
37	98	99	1	1,020
38	98	98	0	0,000
39	98	98	0	0,000
40	99	98	1	1,010
41	96	95	1	1,042
42	98	98	0	0,000
43	99	98	1	1,010
44	98	98	0	0,000
45	98	97	1	1,020
46	99	98	1	1,010
47	97	98	1	1,031
48	99	99	0	0,000

49	97	98	1	1,031
50	98	98	0	0,000
51	96	95	1	1,042
52	99	99	0	0,000
53	99	98	1	1,010
54	98	98	0	0,000
55	97	97	0	0,000
56	96	95	1	1,042
57	98	98	0	0,000
58	96	97	1	1,042
59	98	98	0	0,000
60	97	97	0	0,000
61	99	99	0	0,000
62	96	95	1	1,042
63	98	99	1	1,020
64	96	95	1	1,042
65	96	96	0	0,000
66	99	99	0	0,000
67	96	96	0	0,000
68	98	97	1	1,020
69	96	97	1	1,042
70	99	98	1	1,010
71	99	99	0	0,000
72	97	96	1	1,031
73	97	97	0	0,000
74	97	96	1	1,031
75	99	98	1	1,010
76	98	97	1	1,020
77	97	98	1	1,031
78	96	96	0	0,000
79	97	96	1	1,031
80	97	96	1	1,031
81	99	98	1	1,010
82	99	99	0	0,000
83	97	96	1	1,031
84	98	98	0	0,000
85	97	96	1	1,031
86	99	98	1	1,010
87	96	97	1	1,042
88	96	98	2	2,083
89	99	99	0	0,000
90	96	96	0	0,000

91	96	97	1	1,042
92	96	97	1	1,042
93	99	97	2	2,020
94	96	97	1	1,042
95	97	99	2	2,062
96	99	99	0	0,000
97	98	97	1	1,020
98	96	98	2	2,083
99	99	99	0	0,000
100	98	99	1	1,020
			1	0,709

En la figura 67 se muestra que él envió de las lecturas hacia la nube de ThingSpeak se realizó con éxito y que la gráfica muestra las lecturas de un intervalo de 30 minutos, con una actualización cada 15 segundos aproximadamente.

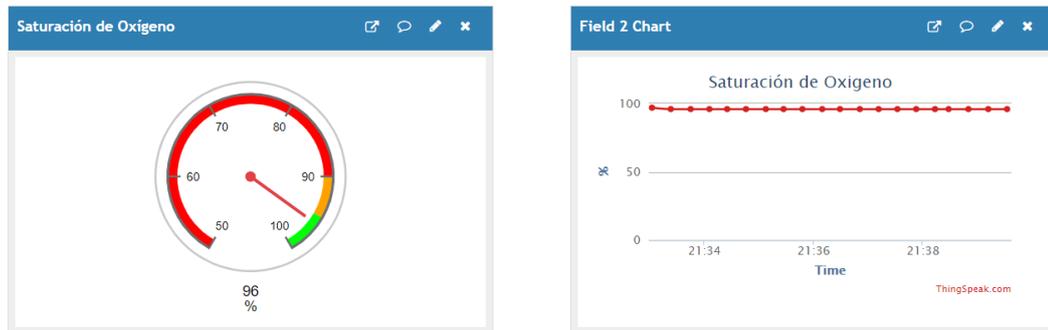


Figura 67. ThingSpeak - Lecturas de Saturación de Oxígeno.

Realizado por Autor.

Basados en la Tabla de Lecturas de SpO2 y empleado las formulas descritas en 3.1.5 se calcula el Valor Promedio y el Error Porcentual, cálculo del cual se ha obtenido los siguientes valores.

- Valor Promedio: 96.5
- Error absoluto: 1
- Error relativo: 0.709

3.1.5.3 Prueba de Temperatura Corporal

El prototipo se coloca en el paciente como se describe en el Apartado 3.1.4 quedando como muestra la figura 68, de lo cual se obtuvieron los datos detallados en la Tabla XX.



Figura 68. Implementación de prototipo - Lectura de Temperatura.

Realizado por Autor.

Tabla XX. Muestras de prueba de Temperatura Corporal.

Realizado por el Autor.

#	Temperatura Termómetro Infrarrojo	Temperatura Termómetro Convencional Axilar	Temperatura Prototipo	Error Absoluto	Error Relativo
1	36,1	36,2	36,5	0,4	1,108
2	36,0	36,2	36,5	0,5	1,389
3	36,2	36,2	36,5	0,3	0,829
4	36,1	36,2	36,5	0,4	1,108
5	36,1	36,2	36,5	0,4	1,108
6	36,3	36,2	36,5	0,2	0,551
7	36,0	36,2	36,5	0,5	1,389
8	36,1	36,2	36,5	0,4	1,108
9	36,1	36,2	36,5	0,4	1,108
10	36,2	36,2	36,5	0,3	0,829
11	36,0	36,2	35,9	0,1	0,278
12	36,0	36,2	35,9	0,1	0,278
13	36,0	36,2	36,0	0,0	0,000
14	36,1	36,2	36,0	0,1	0,277
15	36,1	36,2	36,7	0,6	1,662

16	36,3	36,2	36,7	0,4	1,102
17	36,2	36,2	36,7	0,5	1,381
18	36,2	36,2	36,7	0,5	1,381
19	36,1	36,2	36,7	0,6	1,662
20	36,2	36,2	36,7	0,5	1,381
21	36,3	36,2	36,7	0,4	1,102
22	36,3	36,2	36,7	0,4	1,102
23	36,2	36,2	36,7	0,5	1,381
24	36,2	36,2	36,7	0,5	1,381
25	36,2	36,2	36,7	0,5	1,381
26	36,2	36,2	36,7	0,5	1,381
27	36,2	36,2	36,7	0,5	1,381
28	36,1	36,2	36,7	0,6	1,662
29	36,1	36,2	36,7	0,6	1,662
30	36,1	36,2	35,9	0,2	0,554
31	36,0	36,2	36,8	0,8	2,222
32	36,0	36,2	36,8	0,8	2,222
33	36,0	36,2	36,8	0,8	2,222
34	36,0	36,2	36,8	0,8	2,222
35	36,0	36,2	36,8	0,8	2,222
36	36,1	36,2	36,8	0,7	1,939
37	36,1	36,2	36,8	0,7	1,939
38	36,1	36,2	36,8	0,7	1,939
39	36,1	36,2	35,9	0,2	0,554
40	36,1	36,2	35,9	0,2	0,554
41	36,1	36,2	35,9	0,2	0,554
42	36,1	36,2	35,9	0,2	0,554
43	36,1	36,2	35,9	0,2	0,554
44	36,2	36,2	35,9	0,3	0,829
45	36,2	36,2	35,9	0,3	0,829
46	36,3	36,2	35,9	0,4	1,102
47	36,3	36,2	35,9	0,4	1,102
48	36,2	36,2	35,9	0,3	0,829
49	36,2	36,2	35,9	0,3	0,829
50	36,2	36,2	36,1	0,1	0,276
51	36,2	36,2	36,1	0,1	0,276
52	36,2	36,2	36,1	0,1	0,276
53	36,1	36,2	36,1	0,0	0,000
54	36,1	36,2	36,1	0,0	0,000
55	36,1	36,2	36,1	0,0	0,000
56	36,1	36,2	36,1	0,0	0,000
57	36,1	36,2	36,1	0,0	0,000

58	36,1	36,2	36,1	0,0	0,000
59	36,1	36,2	36,1	0,0	0,000
60	36,1	36,2	35,9	0,2	0,554
61	36,0	36,2	35,9	0,1	0,278
62	36,0	36,2	35,9	0,1	0,278
63	36,0	36,2	35,9	0,1	0,278
64	36,0	36,2	35,9	0,1	0,278
65	36,0	36,2	35,9	0,1	0,278
66	36,3	36,2	35,9	0,4	1,102
67	36,3	36,2	35,9	0,4	1,102
68	36,2	36,2	35,9	0,3	0,829
69	36,2	36,2	35,9	0,3	0,829
70	36,2	36,2	35,9	0,3	0,829
71	36,2	36,2	35,9	0,3	0,829
72	36,2	36,2	35,9	0,3	0,829
73	36,3	36,2	35,9	0,4	1,102
74	36,3	36,2	35,9	0,4	1,102
75	36,2	36,2	35,9	0,3	0,829
76	36,2	36,2	35,9	0,3	0,829
77	36,2	36,2	35,9	0,3	0,829
78	36,3	36,2	35,9	0,4	1,102
79	36,3	36,2	35,9	0,4	1,102
80	36,2	36,2	36,0	0,2	0,552
81	36,2	36,2	36,0	0,2	0,552
82	36,3	36,2	36,0	0,3	0,826
83	36,2	36,2	36,0	0,2	0,552
84	36,2	36,2	36,0	0,2	0,552
85	36,2	36,2	36,0	0,2	0,552
86	36,2	36,2	36,0	0,2	0,552
87	36,2	36,2	36,0	0,2	0,552
88	36,1	36,2	36,0	0,1	0,277
89	36,1	36,2	36,0	0,1	0,277
90	36,1	36,2	35,9	0,2	0,554
91	36,1	36,2	35,9	0,2	0,554
92	36,1	36,2	35,9	0,2	0,554
93	36,1	36,2	36,9	0,8	2,216
94	36,1	36,2	36,9	0,8	2,216
95	36,1	36,0	36,9	0,8	2,216
96	36,0	36,2	36,9	0,9	2,500
97	36,0	36,2	36,9	0,9	2,500
98	36,0	36,2	36,9	0,9	2,500
99	36,0	36,2	36,9	0,9	2,500

100	36,0	36,2	36,9	0,9	2,500
				0,362	1,002

En la figura 69 se muestra que el envío de las lecturas desde el prototipo hacia la nube de ThingSpeak se realizó con éxito y que la gráfica muestra las lecturas en la plataforma en un intervalo de 30 minutos, con una actualización cada 15 segundos aproximadamente.

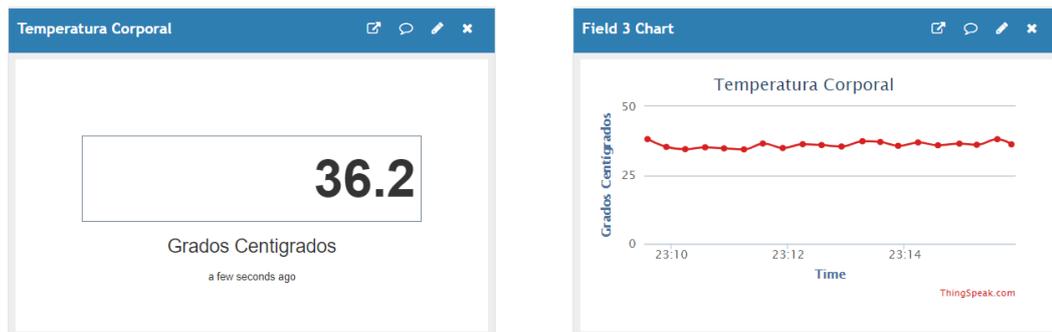


Figura 69. ThingSpeak - Lecturas de Temperatura Corporal.

Realizado por Autor.

Es necesario mencionar que la temperatura axilar requiere adicionar $0,5\text{ }^{\circ}\text{C}$ para ajustar la lectura del prototipo a la temperatura corporal, esta adición se hace tanto para el prototipo como para el termómetro axilar convencional. Por otro lado, la temperatura considerada para este análisis es la lectura final luego de 5 minutos de ubicado el sensor puesto que este tiempo es apropiado para tener una lectura estable y óptima de la temperatura axilar.

Basados en la tabla de lecturas de Temperatura Corporal y las fórmulas descritas en 3.1.5 se calcula el Valor Promedio y el Error Porcentual, cálculo del cual se ha obtenido los siguientes valores.

- Valor Promedio: $35.9\text{ }^{\circ}\text{C}$
- Error absoluto: 0.362
- Error relativo: 1.002

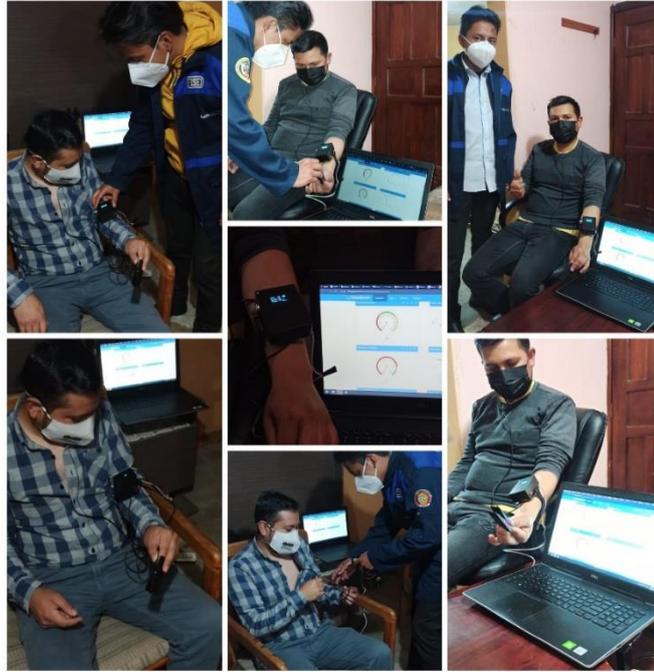


Figura 70. Pruebas de funcionamiento del prototipo.

Realizado por Autor.



Figura 71. Pruebas de funcionamiento del prototipo pacientes con enfermedades asociadas.

Realizado por Autor.



Figura 72. Pruebas de funcionamiento del prototipo en pacientes.

Realizado por Autor.

3.1.5.4 Prueba de tiempo de vida de batería

El prototipo requiere de una alimentación de 5 volts a 0,25 amperios aproximados esto se obtiene tras analizar el consumo que cada dispositivo posee como se muestra en la tabla XXI.

Tabla XXI. Consumo de corriente de dispositivos.

Realizado por Autor.

Dispositivo	Consumo Promedio de Corriente
Microcontrolador	0,0031 amperios
Tarjeta Esp 32	0,2 amperios

Sensores Biomédicos	MAX30100	0,0007 amperios
	LM35	0,00006 amperios
Total		0,20386 amperios

Con estos datos de consumo se puede y considerando una batería de 7.2 volts a 0,35 amperios se puede determinar el tiempo de vida de la batería bajo la siguiente ecuación:

$$\frac{\text{Capacidad de la batería (mili amperios)}}{\text{Consumo del prototipo (horas)}} = \text{Tiempo de vida de la batería}$$

De esta manera se obtiene que el funcionamiento independiente se tiene en un rango de 1.4 horas aproximado bajo un funcionamiento ideal.

3.1.6 Resultados

3.1.6.1 Resultados de Lecturas de prototipo y Error Porcentual

Frecuencia Cardiaca

- Error relativo: 0.965

El error obtenido de las lecturas de Frecuencia Cardiaca es menor a 1% de lo cual se puede concluir que la medición en el prototipo es acertada y que el funcionamiento del prototipo en cuanto a esta constante vital es adecuado.

Considerando que existieron variaciones entre lecturas de más de 1 punto de diferencia pero que durante el tiempo de transición del sistema las lecturas se estabilizaron.

Saturación de Oxígeno

- Error porcentual: 0.709

El error obtenido de las lecturas de Saturación de Oxígeno es de 0.709% menor a 1% que permite concluir que las mediciones del prototipo en cuanto a esta constante vital son óptimas y que el sistema permite el funcionamiento apropiado para las mediciones.

Por otro lado, es necesario mencionar que la diferencia entre cada lectura no varía con mayor diferencia lo que permite contar con mediciones acertadas aun en la etapa de transición del prototipo hasta la estabilización.

Temperatura Corporal

- Error porcentual: 0.683

EL error en cuanto al parámetro de Temperatura Corporal es mayor a 0.1 pero se mantiene en un valor menor a 1% lo que permite contar con lecturas apropiadas para mediciones médicas de esta constante vital, además que el prototipo cuenta con un funcionamiento adecuado y funcional tras la etapa de transición hasta contar con lecturas estables, considerando que para la temperatura al ser un sensor que trabaja con el concepto de expansión de calor se posee un tiempo de estabilización entre 4 a 5 minutos.

3.1.6.2 Análisis de tiempo de vida de batería

El prototipo posee una conexión de alimentación que puede emplear una fuente de voltaje externa gracias a la entrada de 5 volts para toda la PCB lo cual permite el uso de una fuente externa de 5 volts a 0,7 amperios, esto brinda una utilidad prolongada del prototipo sin necesidad de recargar batería.

Además, el prototipo posee una versión independiente que requiere una adaptación de batería, en este caso se emplea una batería de litio de 7.2 volts a 350m Amperios que de acuerdo al cálculo de consumo de corriente de cada dispositivo mostrado en la Tabla XXI es suficiente para proveer alimentación. Con los datos mostrados en el cálculo de vida útil de la batería se tiene un aproximado de 1.4 a 2 horas de uso continuo en un ámbito ideal.

3.1.7 Costos del prototipo

3.1.7.1 Costo de Hardware

Para el desarrollo del proyecto se utilizaron diferentes componentes y procesos electrónicos en cada una de las capas del prototipo, en la Tabla XXII se definen los costos de cada uno.

Tabla XXII. Costos de Implementación – Hardware.

Realizado por Autor.

Recurso	Precio Unitario	Cantidad	Total
Microcontrolador ATMega 328p	\$ 4,00	1	\$ 4,00
Resistencias	\$ 0,10	3	\$ 0,30
Capacitores	\$ 0,20	2	\$ 0,40
Oscilador de Cristal	\$ 0,25	1	\$ 0,25
Switch de Encendido	\$ 0,20	1	\$ 0,20
Sensor MAX 30100	\$ 9,00	1	\$ 9,00
Sensor LM35 Encapsulado	\$ 4,50	1	\$ 4,50
Sensor MQ135	\$ 4,00	1	\$ 4,00
Pantalla OLED 128x64	\$ 8,50	1	\$ 8,50
Tarjeta ESP32	\$ 13,00	1	\$ 13,00
Baquelita	\$ 3,50	1	\$ 3,50
Estaño	\$ 1,00	2 mts	\$ 2,00
Impresión 3D	\$ 2,00	2 hrs	\$ 4,00
Cables de Conexión	\$ 0,45	2 mts	\$ 0,90
Batería 3,7 Volts -	\$ 15,00	1	\$ 15,00
Total			\$ 69,55

3.1.7.2 Costo de Software

La ThingSpeak es una plataforma que permite el desarrollo de aplicaciones IoT en el ámbito Educativo y de prototipado, maneja dos opciones de uso, gratuito y de pago cuyas características no varían con notoriedad e importancia para el prototipado del sistema. Por esta razón no se toma en cuenta en el precio final esta sección. Por otro lado, Arduino IDE, es un software libre lo que permite la programación de los componentes programables sin costo incluido. El diseño 3D del prototipo está basado en el software Solid Works que trabaja bajo licencia pagada y una versión de prueba gratuita por lo que para el costo total no se considera.

CAPITULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Finalizado el presente trabajo de investigación para titulación se han obtenido las siguientes conclusiones:

- Se desarrolló un sistema de telemedicina basado en IoT para monitoreo de pacientes con enfermedades respiratorias, obteniendo resultados óptimos para la implementación del prototipo en aplicaciones de telemedicina comparado con dispositivos de uso médico, alcanzando un error relativo en Frecuencia Cardíaca de 0.965, Saturación de Oxígeno con un error relativo de 0.709 y de 1.002 de error absoluto en lecturas de Temperatura Corporal, lo que traduce una efectividad que permitirá contar con un monitoreo adecuado en pacientes con enfermedades respiratorias e implementaciones en pacientes en casa.
- El análisis de estudios e investigaciones permite concluir que el desarrollo de un sistema de telemedicina en la actualidad, con la situación de pandemia brinda un aporte significativo en cuanto al monitoreo de pacientes con diversas enfermedades puesto que permite analizar datos de constantes vitales evitando exposición de pacientes hacia lugares de servicios de salud. Los dispositivos actuales permiten contar con una lectura de constantes vitales sin embargo no cuentan con un análisis e interpretación de los mismos, por ello la importancia de este sistema que combina una solución para estas necesidades.
- El manejo de una Interfaz de Programación de Aplicaciones API como base para el almacenamiento, análisis y visualización de datos contribuye en el intercambio de información entre el dispositivo y la nube, ThingSpeak es una plataforma que permitió el manejo de los datos adquiridos desde hardware IoT e implementando un servidor de computación en la nube Cloud Computing de código abierto de manera que los datos no se limiten a un análisis local sino por medio de internet contar con la información en cualquier lugar.

- Se consideró sensores biomédicos wearables para la adquisición de constantes vitales para el caso de frecuencia cardiaca y saturación de oxígeno el sensor MAX30100 y para temperatura el sensor LM35; sensores que cuentan con características adecuadas para este tipo de aplicaciones, por otro lado, su tamaño permitió una fácil adecuación en los pacientes.
- El prototipo del sistema de telemedicina combina sensores biomédicos que permiten la adquisición de tres constantes vitales en un paciente, como son la Frecuencia Cardiaca, Saturación de Oxígeno y Temperatura Corporal, por otro lado permite contar con un lector de Calidad de Aire, parámetros que pueden ser monitorizados en una Plataforma IoT y que al combinarse alcanzan un costo del prototipo de 70 dólares, existiendo una diferencia de 10 dólares aproximados comparados con dispositivos comerciales, que no se venden por separado y no poseen monitoreo remoto por internet.

4.2 Recomendaciones

Finalizado el trabajo de investigación y tras la implementación se recomienda:

- Se debe considerar que constantes vitales son fundamentales para monitorización en pacientes con enfermedades respiratorias, puesto que el monitoreo de las mismas juegan un papel fundamental en la prevención de mortalidad esto con la oportuna intervención en el paciente.
- Para futuras investigaciones e implementaciones se recomienda el análisis de sensores wearables que permitan una adquisición no invasiva de mayores constantes vitales con una mayor resolución en las lecturas de los sensores además de un manejo de datos digitales para evitar interferencias y lecturas erróneas.
- Las seguridades en los sistemas de telemedicina juegan un papel fundamental, para esto se recomienda contar con un archivo de credenciales que se incluya en la programación para evitar que con el manejo de código libre se vulnere la información ya sea de redes, claves de plataforma y datos en la transmisión de información.

REFERENCIAS BIBLIOGRÁFICAS

- [1] V. D. Suárez, «Reporte Anual de la Calidad del Aire en el DMQ - 2019,» Secretaría de Ambiente Quito DM, Quito, 2019.
- [2] M. d. S. Pública, «Influenza Actualización Epidemiologica,» Ministerio de Salud Pública, Quito, 2019.
- [3] J. Sandoval, S. Monraz, R. Benitez y E. Mireles, «Utilidad de la telemedicina en las enfermedades respiratorias,» *Nuemología y Cirugía de Tórax*, vol. 79, nº 1, pp. 12-16, 2020.
- [4] M. Mesa y I. Perez, «El acto medico en la era de la Telemedicina,» *Revista Medica*, nº 148, pp. 852-857, 2020.
- [5] J. C. Quispe, «Implementación de un oxímetro de pulsos para monitorizar la desaturación del paciente a distancia,» UCH, Lima, 2019.
- [6] D. Alvarez, «Diseño de un prototipo de monitoreo oximétrico Adulto-Pediatrico hospitalario,» Universidad Tecnológica del Perú, Lima, 2019.
- [7] L. Ruiz, «Sistema de Telemedicina para monitoreo continuo de constantes vitales en lactantes menores para evitar el síndrome de muerte súbita,» Universidad Técnica de Ambato, Ambato, 2018.
- [8] J. Vega, M. Lagos, F. Sánchez y J. Cosme, «Monitoreo de Signos Vitales Usando IoT,» *Pistas Educativas*, vol. 41, nº 134, pp. 751-769, 2019.
- [9] G. Martín, V. Olmedo y J. Andoney, «Uso de las tecnologías de la información y comunicación (TIC) en las residencias médicas en México,» *Acta Medica Grupo Ángeles*, vol. 15, nº 2, pp. 150-153, 2017.
- [10] C. Pérez, «Unican,» 2015. [En línea]. Available: https://personales.unican.es/perezvr/pdf/CH1ST_web.pdf. [Último acceso: 15 Agosto 2020].
- [11] K. Rose, S. Eldridge y L. Chapin, *La Internet de las cosas*, España: Internet Society, 2015.
- [12] A. Moises, *Internet de las cosas*, España: Reus S.A, 2018.

- [13] D. Ning, «Developing and Deploying Analytics for IoT Systems,» MatLab, Australia, 2017.
- [14] I. Ramirez y B. Mazon, Analisis de datos Agropecuarios, Machala: UTMACH, 2018.
- [15] M. Cruz, J. Pinargote, G. Demera y D. Zambrano, «Tecnologias de internet de las cosas en la obtencion de informacion,» *Las Ciencias*, vol. 4, n° 2, pp. 147-160, 2018.
- [16] M. De Alcazar, «La Internet de las Cosas, el Big Data y los nuevos problemas de la comunicacion en el Siglo XXI,» *Mediciones Sociales*, vol. 17, pp. 11-24, 2018.
- [17] J. Melody, «Diseño de un sistema de monitoreo de signos vitales para personas adultas con afecciones cardiacas,» Universidad de Guayaquil, Guayaquil, 2019.
- [18] M. d. Modernización, «Internet de las Cosas,» Presidencia de la Nación, Buenos Aires, 2016.
- [19] R. Hervás y G. Mendez, «API de servicios web orientados a accesibilidad,» Universidad COMplutense de Madrid, Madrid, 2016.
- [20] J. Marquez, «Riesgos y vulnerabilidades de la denegación de servicio distribuidos en internet de las cosas,» *Bioética y Derecho*, n° 46, pp. 85-100, 2019.
- [21] J. Arias, «El protocolo HTTP,» 2021. [En línea]. Available: <http://www.it.uc3m.es/jaf/aw/transparencias/http.pdf>. [Último acceso: 29 Julio 2021].
- [22] O. Quiñones, Internet de las cosas, España: Ibukku, 2019.
- [23] GoogleCloud, «Soluciones de Google Cloud IoT: GoogleCloud,» GoogleCloud, 2020. [En línea]. Available: <https://cloud.google.com/solutions/iot?hl=es-419#:~:text=La%20plataforma%20de%20IoT%20de,la%20ubicaci%C3%B3n%20de%20los%20dispositivos..> [Último acceso: 10 Marzo 2021].
- [24] A. Cardenas , «¿Qué es una plataforma IoT?,» FIWOO, 28 Noviembre 2016. [En línea]. Available: <https://secmotic.com/plataforma-iot/#gref>. [Último acceso: 10 MARzo 2021].

- [25] R. L. Garrido, «Estudio de Plataformas IoT,» PFC, 2015.
- [26] F. Pérez, «Sistemas de prototipado rápido del Internet de las Cosas,» Escuela de Ingeniería ELéctrica, Valparaíso, 2019.
- [27] MathWorks, «ThingSpeak,» MathWorks, 2021. [En línea]. Available: <https://thingspeak.com/login?skipSSOCheck=true>. [Último acceso: 1 Mayo 2021].
- [28] M. Cruz, P. Oliete, C. Morales, C. Gonzalez, B. Cendon y A. Hernandez, Las tecnologías IoT dentro de la industria conectada 4.0, Madrid: EOI, 2015.
- [29] J. Campos, «Arquitectura IoT de bajo costo para redes de sensores,» Universidad Técnica Federico Santa María, Valparaiso Chile, 2020.
- [30] S. Sakr y A. Elgammal, «Towards a comprehensive data analytics framework for smart healthcare services,» ResearchGate, 2017.
- [31] E. Parrasi, L. Celis, J. Bocanegra y Y. Pascuas, «Estado actual de la telemedicina: una revisión de literatura*,» *Dialnet*, nº 20, pp. 105-120, 2016.
- [32] O. Soriano y L. Martin, «Telemedicina: ¿futuro o presente?,» *TECNOLOGÍA DE LA SALUD*, vol. 9, nº 1, pp. 127-139, 2010.
- [33] C. Ruiz, A. Zuluaga y A. Trujillo , «TELEMEDICINA: Introducción, aplicación y principios de desarrollo,» *Redalyc*, vol. 21, nº 1, pp. 77-93, 2007.
- [34] C. Martinez, «Telemedicina. Aspectos Generales.,» *REDUCA*, vol. 1, nº 1, pp. 61-79, 2009.
- [35] D. Novillo, «Marco de Implementación de un Servicio de Telemedicina,» OMS - OPS, Washington DC, 2016.
- [36] S. Quast y O. Kimberger, La importancia de la temperatura corporal central, Lübeck: Dräger Medical GmbH.
- [37] O. I. d. P. c. I. Primarias, «IDPS y las enfermedades respiratorias,» IPOPI, 2016.
- [38] F. d. I. S. R. Internacionales, «EL impacto mundial de la Enfermedad Respiratoria,» *Asociacion Latinoamerica de Torax*, vol. II, nº 2, p. 48, 2017.
- [39] O. M. d. I. Salud, «Organizacion Mundial de la Salud,» 1 Diciembre 2017. [En línea]. Available: <https://www.who.int/es/news-room/fact->

sheets/detail/chronic-obstructive-pulmonary-disease-(copd). [Último acceso: 19 Febrero 2021].

- [40] J. Campo, R. Gaviria y J. Sánchez, «Prevalence of asthma in Latin America,» *Revista Alergia México*, vol. 64, n° 2, pp. 188-197, 2017.
- [41] J. Hernández, «Aspectos clínicos relacionados con el Síndrome Respiratorio Agudo Severo (SARS-CoV-2),» Universidad de Ciencias Médicas de La Habana, Cuba, 2020.
- [42] G. Silva, F. Pérez y M. Diana, «Tuberculosis en niños y adolescentes en Ecuador: análisis de la notificación, las características de la enfermedad y el resultado del tratamiento,» *Rev Panam Salud Pública*, vol. 43, n° 104, 2019.
- [43] O. M. d. l. Salud, «Aspectos técnicos y regulatorios sobre el uso de oxímetros de pulso en el monitoreo de pacientes COVID-19,» OMS - OPS, 2020.
- [44] F. Medina, «Diseño y construcción de Oxímetro de Pulso Inalambrico,» Universidad Tecnológica de Bolívar, Cartagena de Indias, 2019.
- [45] J. Cajas y L. Yanchapaxi, «Diseño e implementación de un prototipo de módulo para visualizar por medio de touch screen, pc y dispositivos móviles curvas características de sensores de temperatura en el rango de 0 a 200 grados».
- [46] C. Castiblanco y N. Cañón, «Prototipo de bajo costo para monitoreo de calidad del aire en ambientes interiores,» Universidad Piloto de Colombia, Bogotá, 2019.
- [47] M. Garcia, C. C. Rey y P. Perez, «Infecciones Respiratorias Virales,» AEP, Madrid, 2015.

ANEXOS

ANEXO A: Manual de Operación del Sistema

MANUAL DE OPERACIÓN

Índice General

Introducción de Sistema.....	1
Objetivo del Sistema.....	1
Alcance.....	1
Descripción del Sistema.....	2
Partes.....	2
Sensores.....	3
Puesta en Marcha.....	3
Anexos.....	6

INTRODUCCIÓN DEL SISTEMA

Objetivo del Sistema

El sistema de telemedicina orientado a pacientes con enfermedades respiratorias que se encuentra en un monitoreo ya sea en un lugar de servicios de salud o en su hogar bajo un monitoreo constante, se emplea bajo supervisión de un médico que permita identificar si el sistema es adecuado para el paciente. Es prudente que consulte con su médico tratante si el dispositivo es apropiado para el uso en su situación de enfermedad.



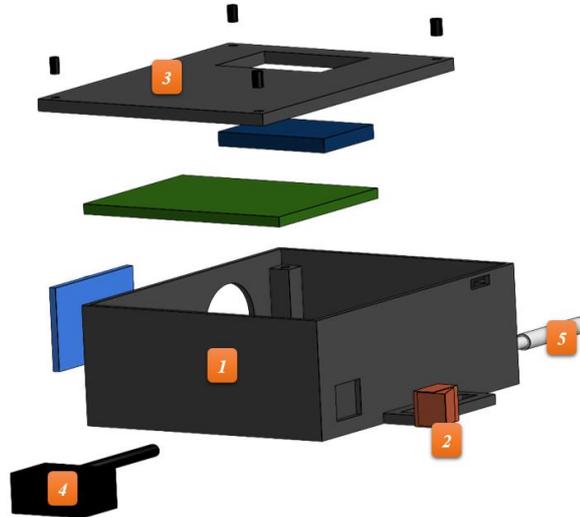
Alcance del Sistema

La actual situación del mundo ha permitido que sistemas de telemedicina se introduzcan como dispositivos para monitoreo y control en pacientes con enfermedades que requieran una supervisión y monitoreo de constantes vitales acordes a la necesidad. Pero varios dispositivos otorgan lecturas de manera local, permitiendo al usuario conocer sus constantes vitales, sin embargo, estos sistemas no cuentan con un análisis e interpretación de resultados. El sistema de medicina IOT HEALTH CM permite que el usuario cuente con lecturas de las constantes vitales y que estos datos se puedan acceder desde cualquier lugar con acceso a internet, logrando un acceso a la información desde el usuario hacia el médico tratante.

DESCRIPCIÓN DEL SISTEMA

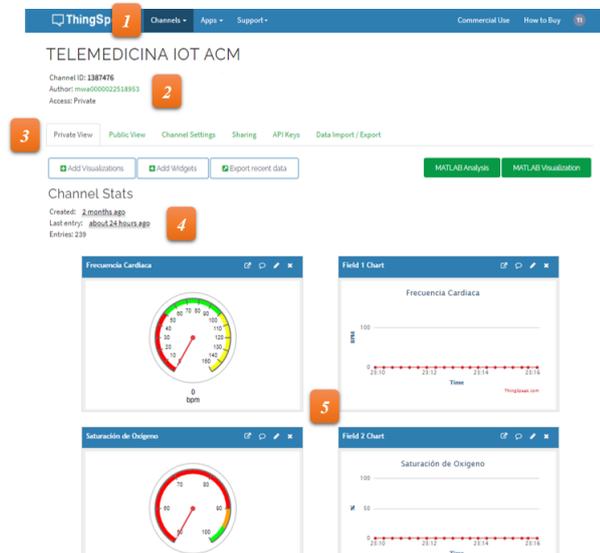
Partes del Sistema

Dispositivo de Medición de Constantes Vitales



1. Case del Sistema
2. Switch de Encendido y Apagado
3. Tapa de Cubierta del Case
4. Sensor de Frecuencia Cardiaca y Saturación de Oxigeno
5. Sensor de Temperatura Corporal

Plataforma de Monitoreo de Mediciones de Constantes Vitales



1. Selección de Canales
2. Información del Canal
3. Tipo de Visualización de Información
4. Estadísticas del Canal
5. Visualización de Lecturas

Sensores

Los sensores del sistema están basados en sensores wearables biomédicos que permiten contar con lecturas apropiadas para un sistema de telemedicina, considerando que fallos en las lecturas podrían traer consecuencias graves.

En este caso se emplearon para la medición de Frecuencia Cardíaca y Saturación de Oxígeno el sensor MAX30100.



El dispositivo MAX30100 permite la adquisición de datos de saturación de oxígeno en la sangre y de pulsaciones por minuto del corazón, es de tamaño reducido que ayuda al manejo e implementación en un case adaptable para el dedo.

El sensor LM35 permite la adquisición de temperatura corporal.



El sensor LM35 que posee comunicación analógica permite la adquisición de datos de temperatura corporal de forma no invasiva pues trabaja de por expansión de temperatura en un objeto en este caso la temperatura corporal.

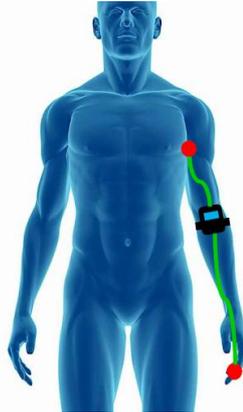
Puesta en Marcha

Para la puesta en marcha se debe tener consideración para cada uno de los procesos que se va a realizar, en este caso se describe por partes las acciones que permitirán al dispositivo funcionar de manera apropiada.

El dispositivo cuenta con un botón de encendido que al presionar podemos contar con el encendido total de todo el sistema. Una vez presionado el botón el sistema mostrara

en la pantalla la información preliminar del dispositivo. Y se empezara a ejecutar la adquisición de datos.

Para esto los sensores se deben colocar de acuerdo a lo descrito.



Una vez que los sensores se han colocado de manera apropiada, se deberá esperar un lapso de tiempo de estabilización de las lecturas, las mediciones se irán actualizando en la pantalla de visualización cada 12 segundos, de esta manera para frecuencia cardiaca y saturación de oxígeno este tiempo es adecuado para ya contar con una medición apropiada, por otro lado, la temperatura corporal requerirá de 4 a 5 minutos para poder contar con mediciones óptimas.

Para iniciar el monitoreo en la plataforma ThingSpeak es necesario hacer Login en la plataforma con las credenciales asignadas por el proveedor.

El formulario de inicio de sesión de MathWorks. Incluye el logo de MathWorks, un campo de entrada para el correo electrónico, un enlace para crear una cuenta si no se tiene, una declaración de consentimiento de la política de privacidad y un botón 'Next'.

Para esto se deberá acceder a la dirección web:

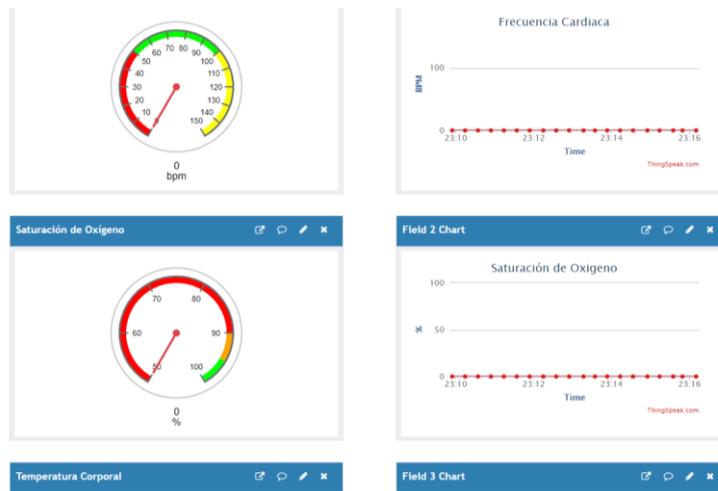
<https://thingspeak.com/login>

Aquí solicitara las credenciales y se realizara el acceso, ahora se deberá acceder al canal del paciente a monitorear.

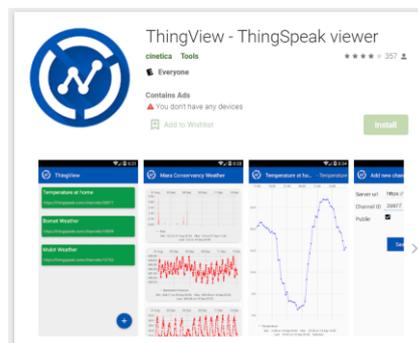
My Channels

Name	Created	Updated
New Channel <input type="text" value="Search by tag"/>		
TELEMEDICINA IOT ACM	2021-05-12	2021-05-13 21:36
Private Public Settings Sharing API Keys Data Import / Export		

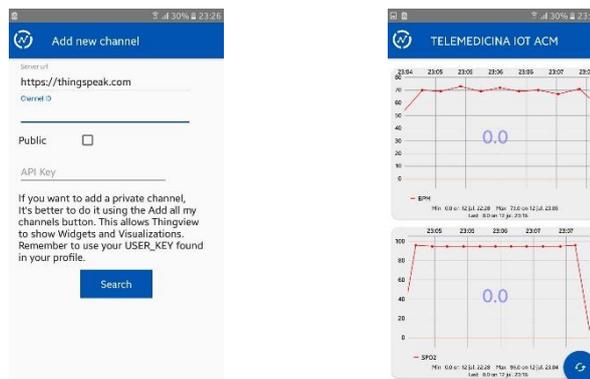
Ahora que las lecturas se han estabilizado en el dispositivo de medición las mediciones se podrán visualizar y monitorear desde cualquier ordenador con acceso a internet.



Además, se cuenta con una aplicación Android “ThingView Free” para el acceso remoto desde un dispositivo celular. Para esto se debe descargar la aplicación desde App Store.

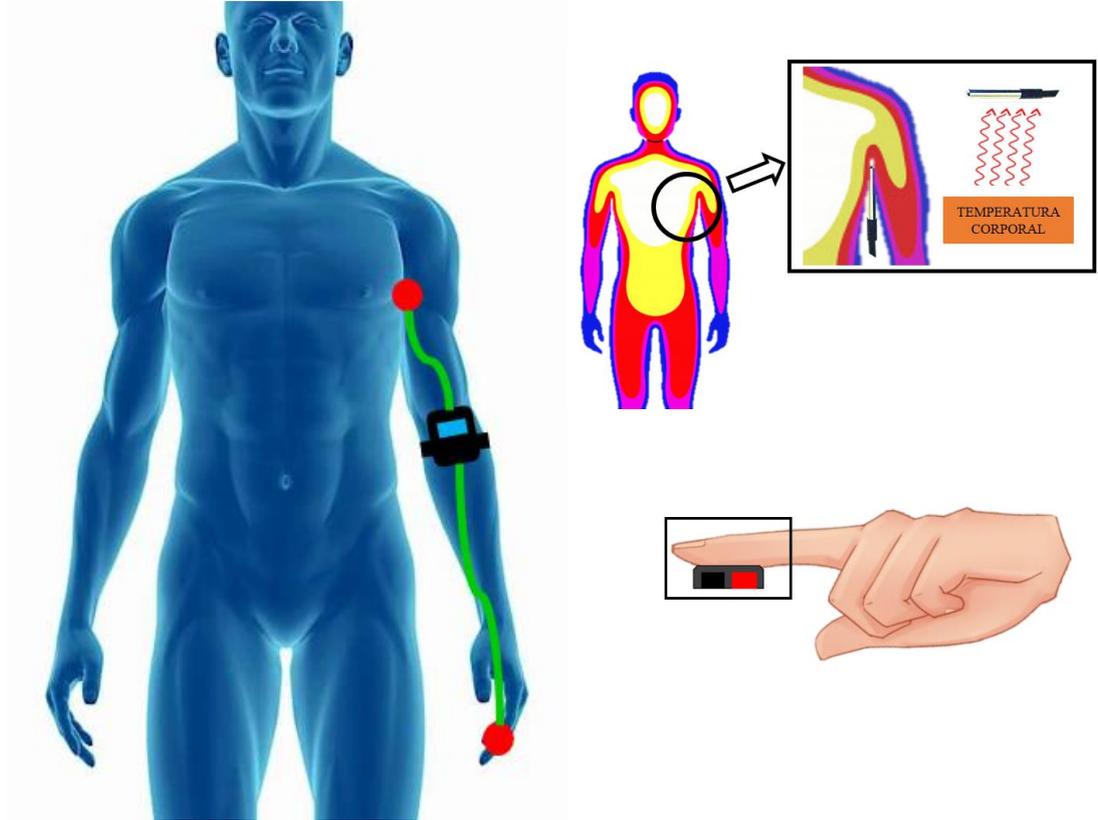


Una vez descargada la aplicación se puede realizar el monitoreo desde la misma, para esto es necesario acceder al Canal y se debe contar con la información de “API KEY” y “CHANNEL ID” esto debido a que el canal cuenta con seguridad de privacidad de manera que los datos se protejan.

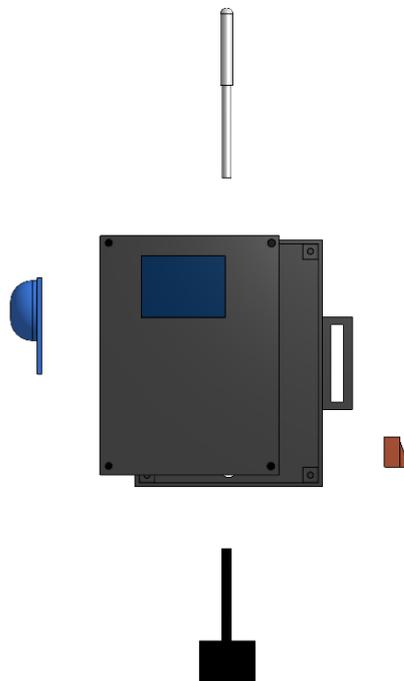


ANEXOS

Ubicación de Sensores



Vista General del Dispositivo



Acceso a Plataforma ThingSpeak

 Channels Apps Support
 Commercial Use How to Buy 

get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

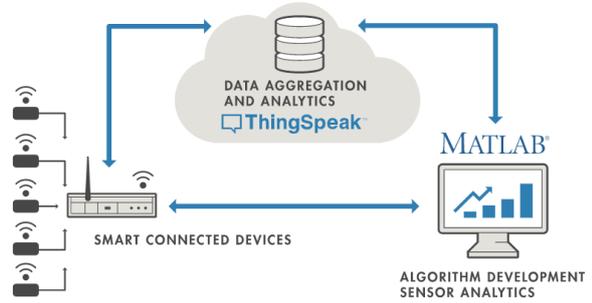


Email

No account? [Create one!](#)

By signing in you agree to our [privacy policy](#).

Next



Vista de Datos en Reportes de Google Sheets

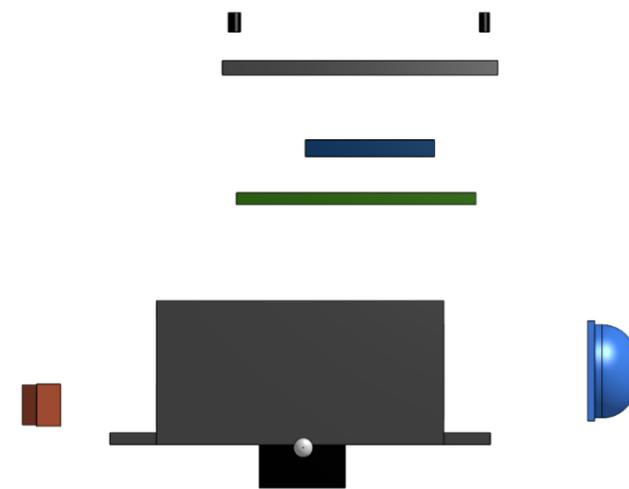
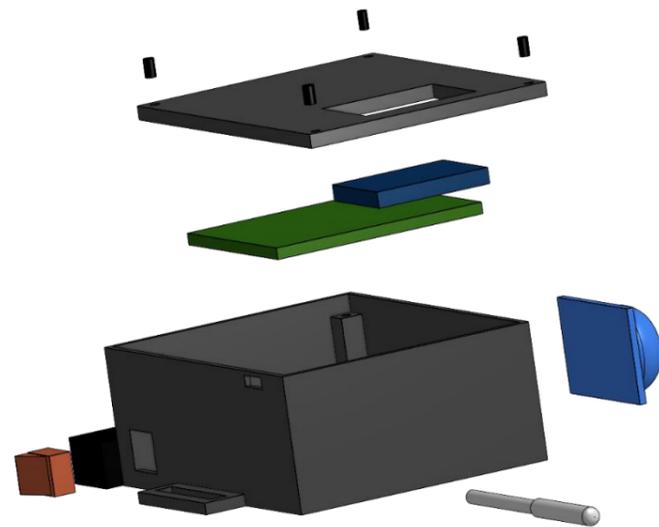
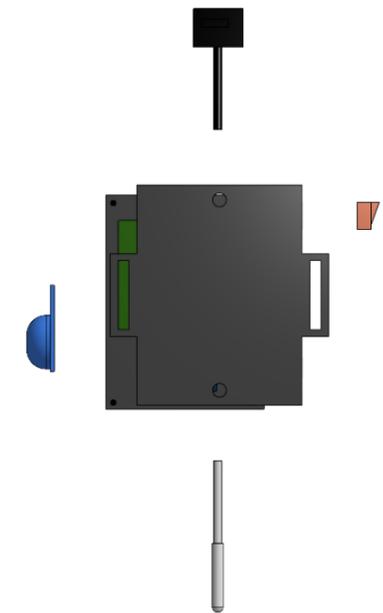
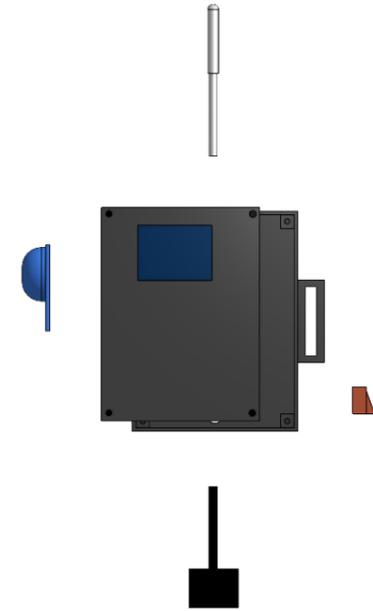
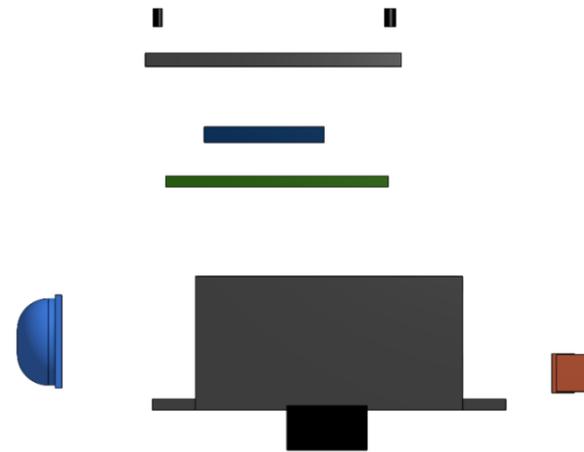
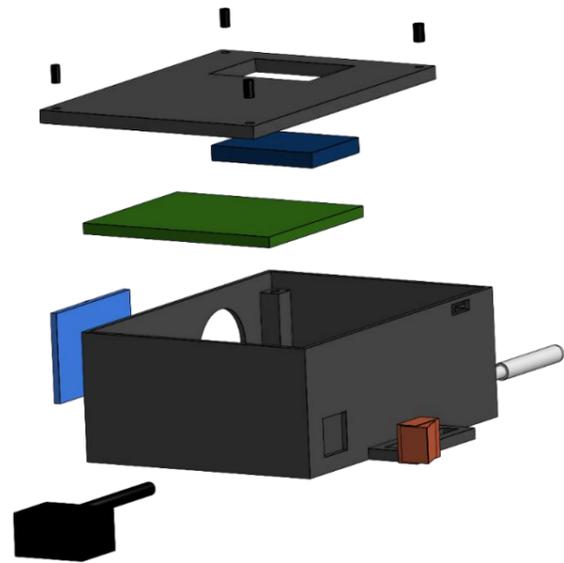
 Freq_SpO2_TempC_Lecturas (3) ☆ 📄 ☁

Archivo Editar Ver Insertar Formato Datos Herramientas Complementos Ayuda La última modificación se real

100% \$ % .0_ .00 123 Predetermi... 10 B I A

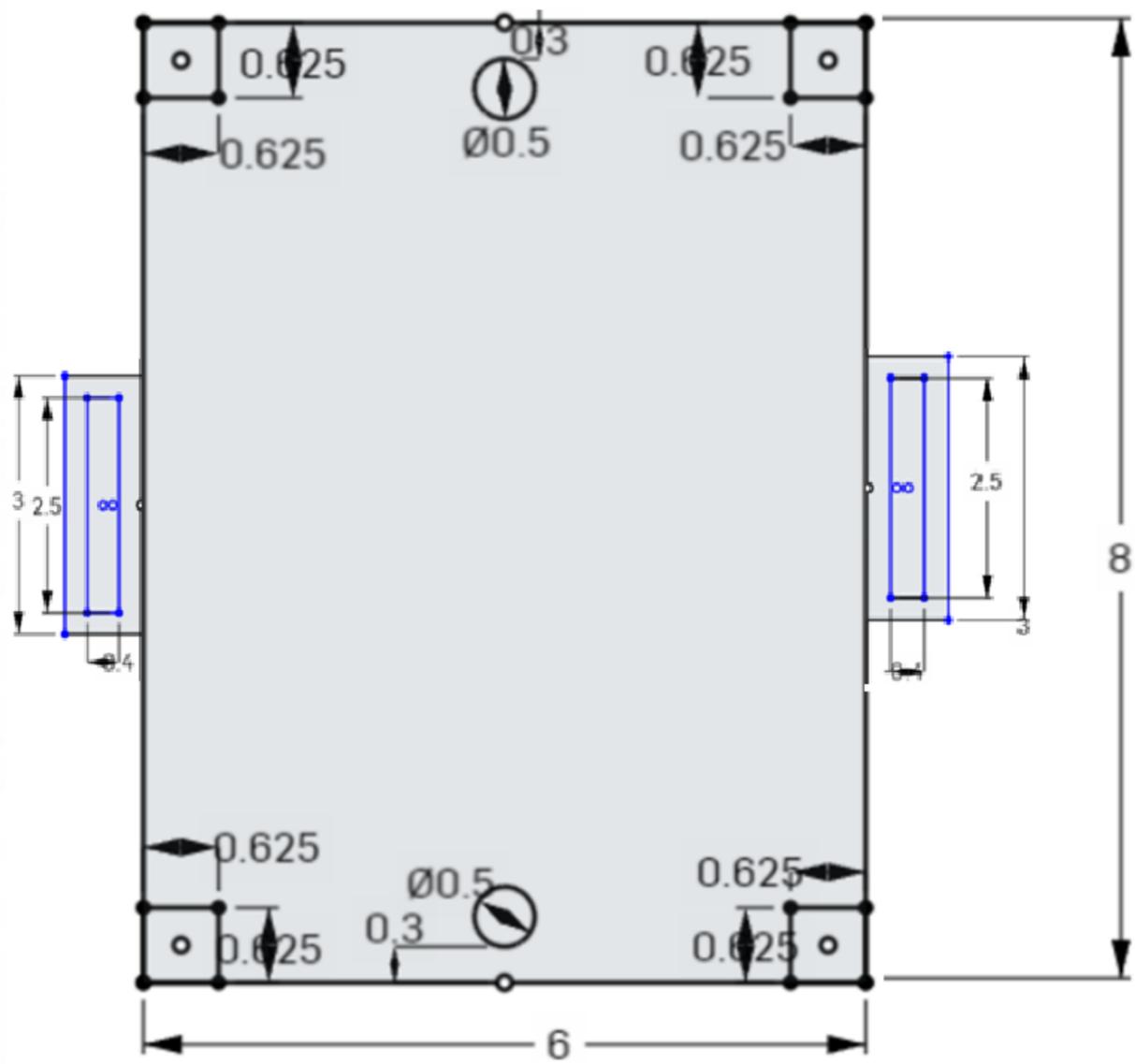
	A	B	C	D	E	F	G	H
596	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	24.98 (°C)			
597	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	30.36 (°C)			
598	July 12, 2021 at	report_xls	35 (bpm)	0 (%)	29.87 (°C)			
599	July 12, 2021 at	report_xls	35 (bpm)	0 (%)	29.87 (°C)			
600	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	25.33 (°C)			
601	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	24.22 (°C)			
602	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	2581.4399 (°C)			
603	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	2578.7 (°C)			
604	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	0 (°C)			
605	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	31.84 (°C)			
606	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	29.41 (°C)			
607	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	0 (°C)			
608	July 12, 2021 at	report_xls	70 (bpm)	95 (%)	34.07 (°C)			
609	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	35.24 (°C)			
610	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	36.44 (°C)			
611	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	36.2 (°C)			
612	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	36.2 (°C)			
613	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	36.2 (°C)			
614	July 12, 2021 at	report_xls	0 (bpm)	0 (%)	36.2 (°C)			

ANEXO B: Vista 3D de componentes del prototipo

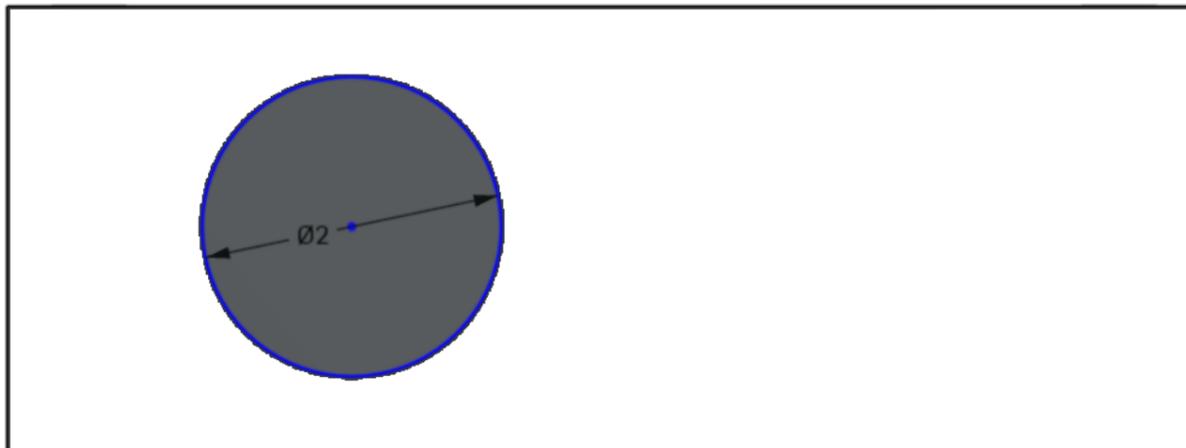


NOTA	FECHA	CURSO	NOMBRE
	2021		ANDRÉS CARRILLO
VISTA 3D DE CARCASA DEL PROTOTIPO			

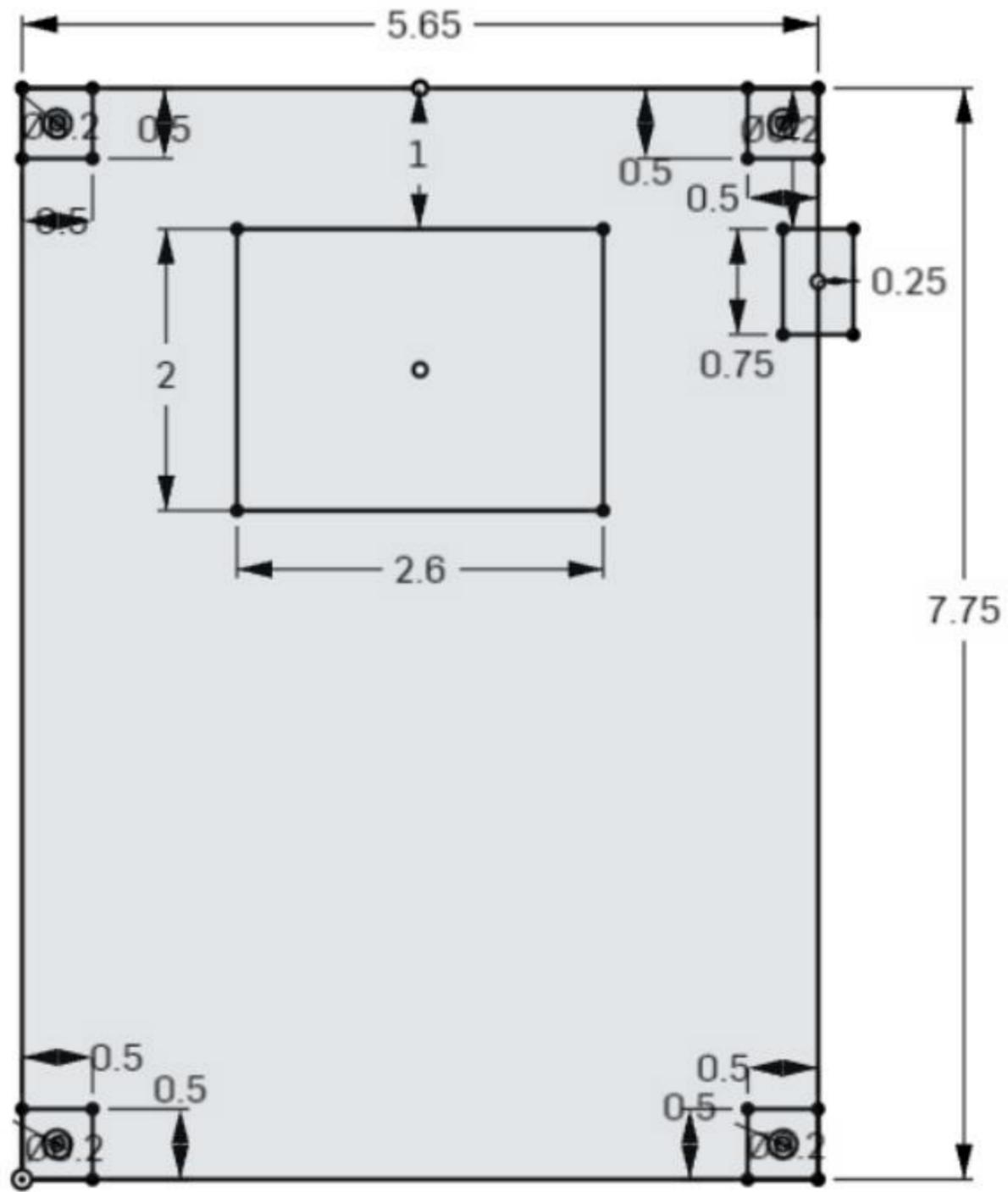
UNIVERSIDAD
TECNICA DE
AMBATO



	FECHA	NOMBRE	ANDRÉS CARRILLO	UNIVERSIDAD TÉCNICA DE AMBATO
DIBUJO				
				2021



	FECHA	NOMBRE	ANDRÉS CARRILLO	UNIVERSIDAD TÉCNICA DE AMBATO
DIBUJO				
				2021



	FECHA	NOMBRE	ANDRÉS CARRILLO	UNIVERSIDAD TÉCNICA DE AMBATO
DIBUJO				
				2021

MAX30100**Pulse Oximeter and Heart-Rate Sensor IC
for Wearable Health****General Description**

The MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor solution. It combines two LEDs, a photodetector, optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals.

The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to remain connected at all times.

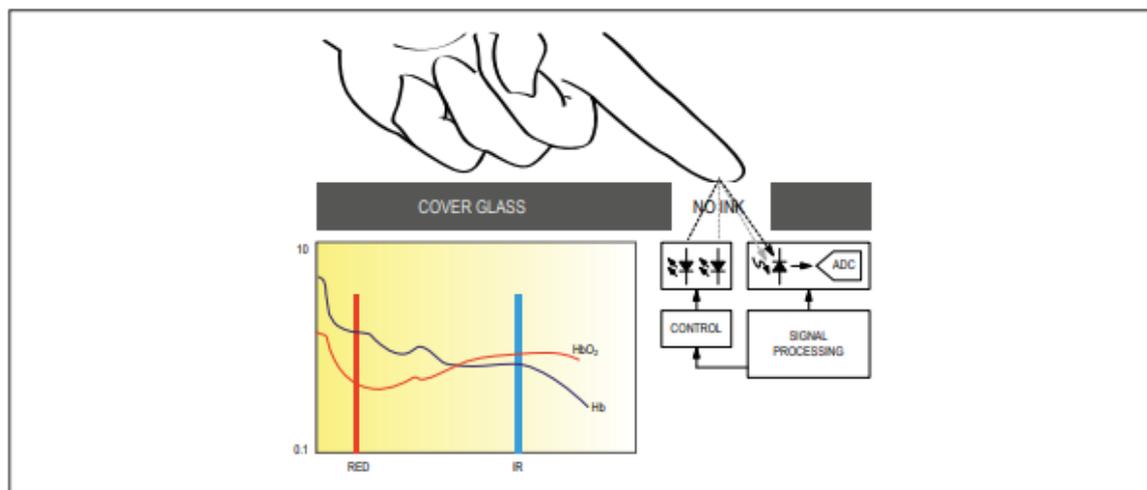
Applications

- Wearable Devices
- Fitness Assistant Devices
- Medical Monitoring Devices

Benefits and Features

- Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
 - Integrated LEDs, Photo Sensor, and High-Performance Analog Front -End
 - Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package
- Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
 - Programmable Sample Rate and LED Current for Power Savings
 - Ultra-Low Shutdown Current (0.7 μ A, typ)
- Advanced Functionality Improves Measurement Performance
 - High SNR Provides Robust Motion Artifact Resilience
 - Integrated Ambient Light Cancellation
 - High Sample Rate Capability
 - Fast Data Output Capability

Ordering Information appears at end of data sheet.

System Block Diagram

ANEXO D: Código fuente de archivo “MAX30100.h”

```
/*
Arduino-MAX30100 oximetry / heart rate integrated sensor
library
Copyright (C) 2016 OXullo Intersecans <x@brainrapers.org>

This program is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as
published by
the Free Software Foundation, either version 3 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be
useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public
License
along with this program. If not, see
<http://www.gnu.org/licenses/>.
*/

#ifndef MAX30100_H
#define MAX30100_H

#include <stdint.h>

#define CIRCULAR_BUFFER_XS
#include "CircularBuffer.h"
#include "MAX30100_Registers.h"

#define DEFAULT_MODE MAX30100_MODE_HRONLY
#define DEFAULT_SAMPLING_RATE MAX30100_SAMPRATE_100HZ
#define DEFAULT_PULSE_WIDTH
MAX30100_SPC_PW_1600US_16BITS
#define DEFAULT_RED_LED_CURRENT MAX30100_LED_CURR_50MA
#define DEFAULT_IR_LED_CURRENT MAX30100_LED_CURR_50MA
#define EXPECTED_PART_ID 0x11
#define RINGBUFFER_SIZE 16

#define I2C_BUS_SPEED 400000UL

typedef struct {
    uint16_t ir;
    uint16_t red;
} SensorReadout;

class MAX30100 {
public:
```

```

MAX30100();
bool begin();
void setMode(Mode mode);
void setLedsPulseWidth(LED PulseWidth ledPulseWidth);
void setSamplingRate(SamplingRate samplingRate);
void setLedsCurrent(LEDCurrent irLedCurrent, LEDCurrent
redLedCurrent);
void setHighresModeEnabled(bool enabled);
void update();
bool getRawValues(uint16_t *ir, uint16_t *red);
void resetFifo();
void startTemperatureSampling();
bool isTemperatureReady();
float retrieveTemperature();
void shutdown();
void resume();
uint8_t getPartId();

private:
    CircularBuffer<SensorReadout, RINGBUFFER_SIZE>
readoutsBuffer;

    uint8_t readRegister(uint8_t address);
    void writeRegister(uint8_t address, uint8_t data);
    void burstRead(uint8_t baseAddress, uint8_t *buffer,
uint8_t length);
    void readFifoData();
};

#endif

```

ANEXO E: Código fuente de archivo “MAX30100_BeatDetector.cpp”

```
/*
Arduino-MAX30100 oximetry / heart rate integrated sensor
library
Copyright (C) 2016 OXullo Intersecans <x@brainrapers.org>

This program is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as
published by
the Free Software Foundation, either version 3 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be
useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public
License
along with this program. If not, see
<http://www.gnu.org/licenses/>.
*/

#include <Arduino.h>

#include "MAX30100_BeatDetector.h"

#ifndef min
#define min(a,b) \
    ({ __typeof__ (a) _a = (a); \
      __typeof__ (b) _b = (b); \
      _a < _b ? _a : _b; })
#endif

BeatDetector::BeatDetector() :
    state(BEATDETECTOR_STATE_INIT),
    threshold(BEATDETECTOR_MIN_THRESHOLD),
    beatPeriod(0),
    lastMaxValue(0),
    tsLastBeat(0)
{
}

bool BeatDetector::addSample(float sample)
{
    return checkForBeat(sample);
}

float BeatDetector::getRate()
{

```

```

    if (beatPeriod != 0) {
        return 1 / beatPeriod * 1000 * 60;
    } else {
        return 0;
    }
}

float BeatDetector::getCurrentThreshold()
{
    return threshold;
}

bool BeatDetector::checkForBeat(float sample)
{
    bool beatDetected = false;

    switch (state) {
        case BEATDETECTOR_STATE_INIT:
            if (millis() > BEATDETECTOR_INIT_HOLDOFF) {
                state = BEATDETECTOR_STATE_WAITING;
            }
            break;

        case BEATDETECTOR_STATE_WAITING:
            if (sample > threshold) {
                threshold = min(sample,
BEATDETECTOR_MAX_THRESHOLD);
                state = BEATDETECTOR_STATE_FOLLOWING_SLOPE;
            }

            // Tracking lost, resetting
            if (millis() - tsLastBeat >
BEATDETECTOR_INVALID_READOUT_DELAY) {
                beatPeriod = 0;
                lastMaxValue = 0;
            }

            decreaseThreshold();
            break;

        case BEATDETECTOR_STATE_FOLLOWING_SLOPE:
            if (sample < threshold) {
                state = BEATDETECTOR_STATE_MAYBE_DETECTED;
            } else {
                threshold = min(sample,
BEATDETECTOR_MAX_THRESHOLD);
            }
            break;

        case BEATDETECTOR_STATE_MAYBE_DETECTED:
            if (sample + BEATDETECTOR_STEP_RESILIENCY <
threshold) {
                // Found a beat
                beatDetected = true;
                lastMaxValue = sample;
            }
    }
}

```

```

        state = BEATDETECTOR_STATE_MASKING;
        float delta = millis() - tsLastBeat;
        if (delta) {
            beatPeriod = BEATDETECTOR_BPFILTER_ALPHA *
delta +
            (1 - BEATDETECTOR_BPFILTER_ALPHA)
* beatPeriod;
        }

        tsLastBeat = millis();
    } else {
        state = BEATDETECTOR_STATE_FOLLOWING_SLOPE;
    }
    break;

    case BEATDETECTOR_STATE_MASKING:
        if (millis() - tsLastBeat >
BEATDETECTOR_MASKING_HOLDOFF) {
            state = BEATDETECTOR_STATE_WAITING;
        }
        decreaseThreshold();
        break;
    }

    return beatDetected;
}

void BeatDetector::decreaseThreshold()
{
    // When a valid beat rate readout is present, target the
    if (lastMaxValue > 0 && beatPeriod > 0) {
        threshold -= lastMaxValue * (1 -
BEATDETECTOR_THRESHOLD_FALLOFF_TARGET) /
        (beatPeriod / BEATDETECTOR_SAMPLES_PERIOD);
    } else {
        // Asymptotic decay
        threshold *= BEATDETECTOR_THRESHOLD_DECAY_FACTOR;
    }

    if (threshold < BEATDETECTOR_MIN_THRESHOLD) {
        threshold = BEATDETECTOR_MIN_THRESHOLD;
    }
}

```

ANEXO F: Código fuente de archivo “MAX30100_SpO2Detector.cpp”

```
/*
Arduino-MAX30100 oximetry / heart rate integrated sensor
library
Copyright (C) 2016 OXullo Intersecans <x@brainrapers.org>

This program is free software: you can redistribute it and/or
modify
it under the terms of the GNU General Public License as
published by
the Free Software Foundation, either version 3 of the License,
or
(at your option) any later version.

This program is distributed in the hope that it will be
useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public
License
along with this program. If not, see
<http://www.gnu.org/licenses/>.
*/

#include <math.h>

#include "MAX30100_SpO2Calculator.h"

// SaO2 Look-up Table
// http://www.ti.com/lit/an/slaa274b/slaa274b.pdf
const uint8_t SpO2Calculator::spO2LUT[43] =
{100,100,100,100,99,99,99,99,99,99,99,98,98,98,98,
98,97,97,97,97,97,97,96,96,96,96,96,96,95,95,
95,95,95,95,94,94,94,94,94,93,93,93,93,93};

SpO2Calculator::SpO2Calculator() :
    irACValueSqSum(0),
    redACValueSqSum(0),
    beatsDetectedNum(0),
    samplesRecorded(0),
    spO2(0)
{
}

void SpO2Calculator::update(float irACValue, float redACValue,
bool beatDetected)
{
    irACValueSqSum += irACValue * irACValue;
    redACValueSqSum += redACValue * redACValue;
```

```

++samplesRecorded;

if (beatDetected) {
    ++beatsDetectedNum;
    if (beatsDetectedNum == CALCULATE_EVERY_N_BEATS) {
        float acSqRatio = 100.0 *
log(redACValueSqSum/samplesRecorded) /
log(irACValueSqSum/samplesRecorded);
        uint8_t index = 0;

        if (acSqRatio > 66) {
            index = (uint8_t)acSqRatio - 66;
        } else if (acSqRatio > 50) {
            index = (uint8_t)acSqRatio - 50;
        }
        reset();

        spO2 = spO2LUT[index];
    }
}

void SpO2Calculator::reset()
{
    samplesRecorded = 0;
    redACValueSqSum = 0;
    irACValueSqSum = 0;
    beatsDetectedNum = 0;
    spO2 = 0;
}

uint8_t SpO2Calculator::getSpO2()
{
    return spO2;
}

```

ANEXO G: Datasheet sensor LM35

LM35 Precision Centigrade Temperature Sensors

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-μA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

3 Description

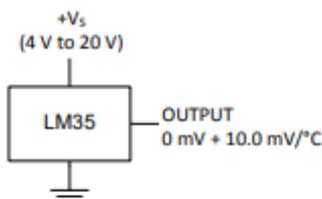
The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full -55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 μA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a -55°C to 150°C temperature range, while the LM35C device is rated for a -40°C to 110°C range (-10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Device Information⁽¹⁾

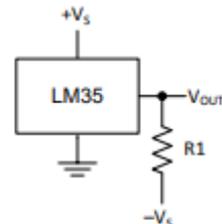
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM35	TO-CAN (3)	4.699 mm × 4.699 mm
	TO-92 (3)	4.30 mm × 4.30 mm
	SOIC (8)	4.90 mm × 3.91 mm
	TO-220 (3)	14.986 mm × 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Basic Centigrade Temperature Sensor (2°C to 150°C)



Full-Range Centigrade Temperature Sensor



Choose $R_1 = -V_S / 50 \mu\text{A}$
 $V_{\text{OUT}} = 1500 \text{ mV at } 150^\circ\text{C}$
 $V_{\text{OUT}} = 250 \text{ mV at } 25^\circ\text{C}$
 $V_{\text{OUT}} = -550 \text{ mV at } -55^\circ\text{C}$



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

ANEXO H: Datasheet sensor MQ135

TECHNICAL DATA

MQ-135 GAS SENSOR

FEATURES

Wide detecting scope
Stable and long life

Fast response and High sensitivity
Simple drive circuit

APPLICATION

They are used in air quality control equipments for buildings/offices, are suitable for detecting of NH₃, NO_x, alcohol, Benzene, smoke, CO₂, etc.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remarks
V _c	Circuit voltage	5V±0.1	AC OR DC
V _H	Heating voltage	5V±0.1	AC OR DC
R _L	Load resistance	can adjust	
R _H	Heater resistance	33 Ω ± 5%	Room Tem
P _H	Heating consumption	less than 800mw	

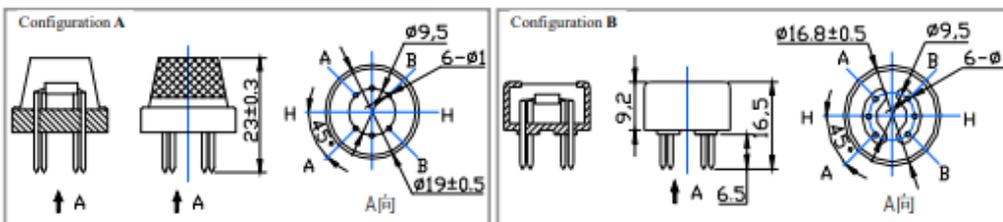
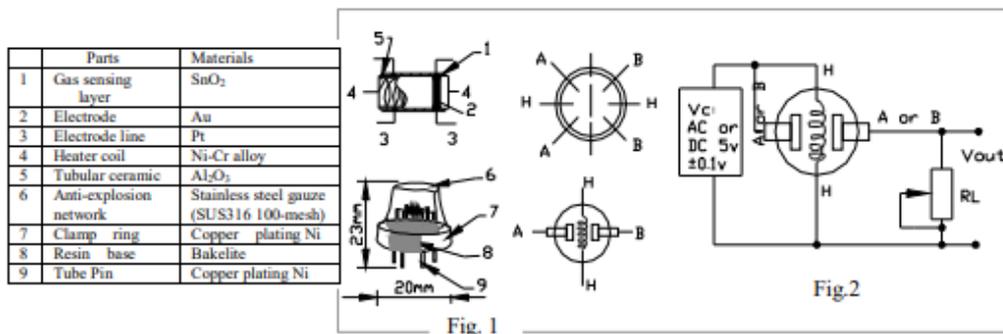
B. Environment condition

Symbol	Parameter name	Technical condition	Remarks
T _{ao}	Using Tem	-10°C-45°C	
T _{as}	Storage Tem	-20°C-70°C	
R _H	Related humidity	less than 95%Rh	
O ₂	Oxygen concentration	21%(standard condition)Oxygen concentration can affect sensitivity	minimum value is over 2%

C. Sensitivity characteristic

Symbol	Parameter name	Technical parameter	Remark 2
R _s	Sensing Resistance	30K Ω -200K Ω (100ppm NH ₃)	Detecting concentration scope: 10ppm-300ppm NH ₃ 10ppm-1000ppm Benzene 10ppm-300ppm Alcohol
α (200/50) NH ₃	Concentration Slope rate	≤0.65	
Standard Detecting Condition	Temp: 20°C ± 2°C Humidity: 65%±5%	V _c : 5V±0.1 V _H : 5V±0.1	
Preheat time	Over 24 hour		

D. Structure and configuration, basic measuring circuit



Structure and configuration of MQ-135 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro Al₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of

TEL: 86-371-67169070 67169080

FAX: 86-371-67169090

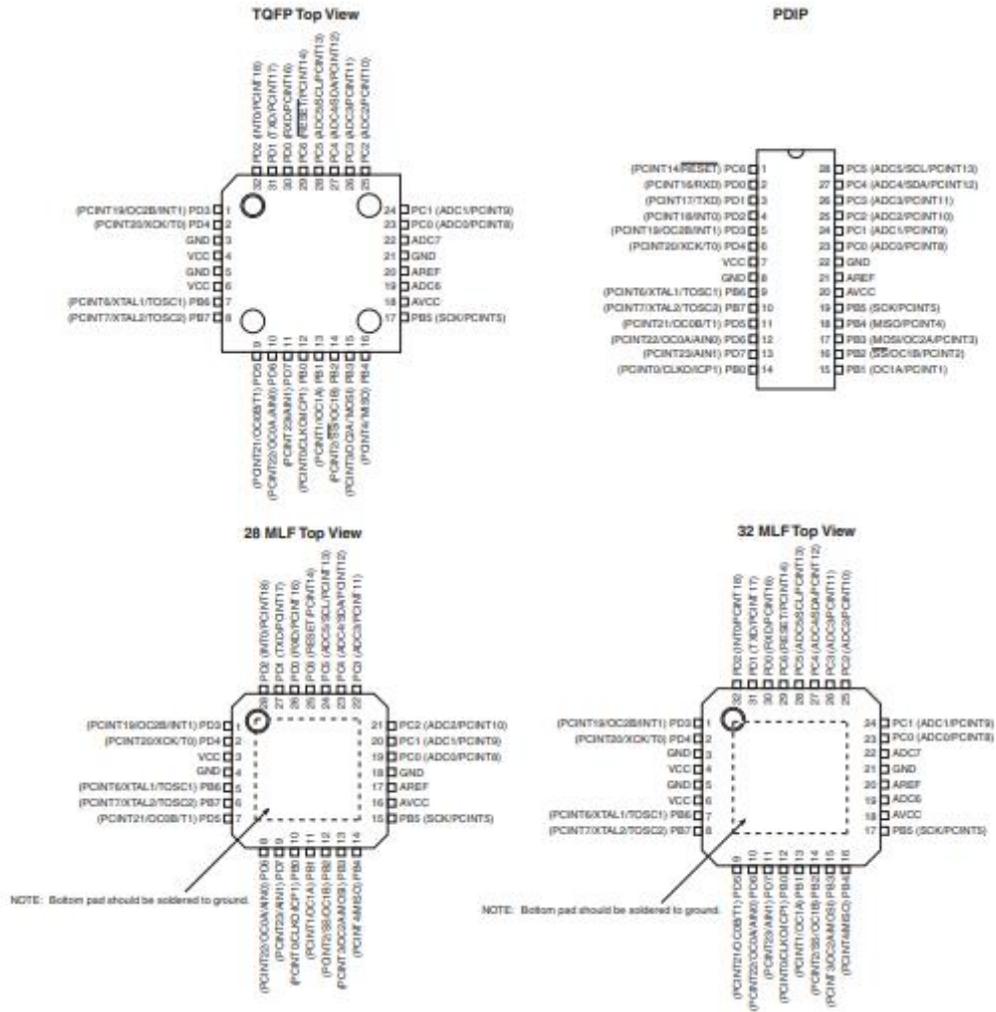
E-mail: sales@hwsensor.com

ANEXO I: Datasheet Microcontrolador ATmega328p

ATmega48PA/88PA/168PA/328P

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



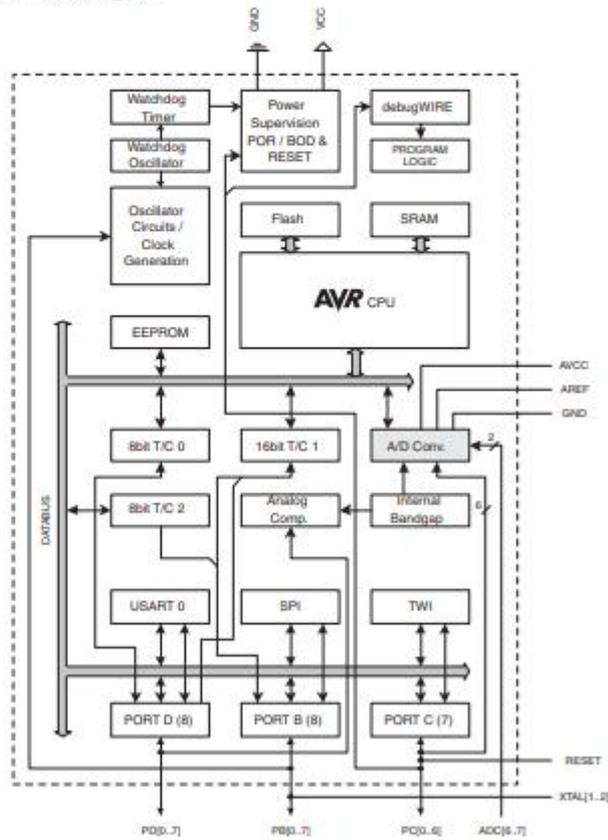
ATmega48PA/88PA/168PA/328P

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting

ANEXO J: Código fuente de Microcontrolador ATmega328p

```
#include "Wire.h"
#include "MAX30100_PulseOximeter.h"

#include "Adafruit_GFX.h"
#include "Adafruit_SSD1306.h"
#include "OakOLED.h"

#define REPORTING_PERIOD_MS      12000
#define SAMPLES_FAST_AVERAGE    64
#define SAMPLES_TO_AVERAGE      256

OakOLED oled;
PulseOximeter pox;

int AirQ=A0, LM35=A1;
int BPM, SpO2, Aire, Lectura;
float tempC,tempCO;
float RawTemp;

uint32_t tsLastReport = 0;

const unsigned char bitmap [] PROGMEM=
{
0x00, 0x00, 0x00, 0x00, 0x01, 0x80, 0x18, 0x00, 0x0f, 0xe0,
0x7f, 0x00, 0x3f, 0xf9, 0xff, 0xc0,
0x7f, 0xf9, 0xff, 0xc0, 0x7f, 0xff, 0xff, 0xe0, 0x7f, 0xff,
0xff, 0xe0, 0xff, 0xff, 0xff, 0xf0,
0xff, 0xf7, 0xff, 0xf0, 0xff, 0xe7, 0xff, 0xf0, 0xff, 0xe7,
0xff, 0xf0, 0x7f, 0xdb, 0xff, 0xe0,
0x7f, 0x9b, 0xff, 0xe0, 0x00, 0x3b, 0xc0, 0x00, 0x3f, 0xf9,
0x9f, 0xc0, 0x3f, 0xfd, 0xbf, 0xc0,
0x1f, 0xfd, 0xbf, 0x80, 0x0f, 0xfd, 0x7f, 0x00, 0x07, 0xfe,
0x7e, 0x00, 0x03, 0xfe, 0xfc, 0x00,
0x01, 0xff, 0xf8, 0x00, 0x00, 0xff, 0xf0, 0x00, 0x00, 0x7f,
0xe0, 0x00, 0x00, 0x3f, 0xc0, 0x00,
0x00, 0x0f, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00
};

void onBeatDetected()
{
oled.drawBitmap( 100, 00, bitmap, 28, 28, 1);
oled.display();
}

void setup()
{
pinMode(AirQ, INPUT);
pinMode(LM35, INPUT);
Serial.begin(115200);
}
```

```

oled.begin();
oled.clearDisplay();
oled.setTextSize(2);
oled.setTextColor(1);

if (!pox.begin())
{
    for(;;);
} else
{
    pox.setOnBeatDetectedCallback(onBeatDetected);
}
pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);

for (int i = 0; i < SAMPLES_TO_AVERAGE; i++)
{
    Lectura = analogRead(LM35);
    RawTemp = Lectura / 2.048;
    tempC = (tempC + ( RawTemp - tempC) / SAMPLES_TO_AVERAGE);
    tempCO=tempC+10;
}

}

void loop()
{
    pox.update();
    BPM = pox.getHeartRate();
    SpO2 = pox.getSpO2();
    Aire = analogRead(AirQ);
    Lectura = analogRead(LM35);
    RawTemp = Lectura / 2.048;

    tempC = (tempC + ( RawTemp - tempC) / SAMPLES_TO_AVERAGE);
    tempCO=tempC+10;

    if (millis() - tsLastReport > REPORTING_PERIOD_MS)
    {

        Serial.print(BPM);
        Serial.print(',');

        Serial.print(SpO2);
        Serial.print(',');

        Serial.print(tempCO);
        Serial.print(',');

        Serial.print(Aire);
        Serial.print(',');

        //////////////////////////////////////
oled.clearDisplay();

```

```
oled.setCursor(0,0);
oled.println("BPM:");

oled.setCursor(60,0);
oled.println(BPM);

oled.setCursor(0,20);
oled.println("SpO2:");

oled.setCursor(60,20);
oled.println(SpO2);

oled.setCursor(0,40);
oled.println("Temp:");

oled.setCursor(60,40);
oled.println(tempCO,1);
oled.display();
////////////////////////////////////
    tsLastReport = millis();

}

}
```

ANEXO K: Datasheet Tarjeta ESP32

2. Pin Definitions

2. Pin Definitions

2.1 Pin Layout

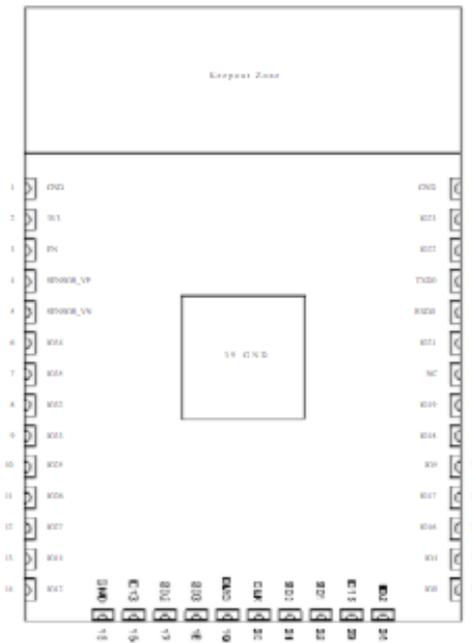


Figure 1: ESP32-WROOM-32D Pin Layout (Top View)

Note:

The pin layout of ESP32-WROOM-32U is the same as that of ESP32-WROOM-32D, except that ESP32-WROOM-32U has no keepout zone.

2.2 Pin Description

The ESP32-WROOM-32D and ESP32-WROOM-32U have 38 pins. See pin definitions in Table 3.

Table 3: Pin Definitions

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5

Table 2: ESP32-WROOM-32D and ESP32-WROOM-32U Specifications

Categories	Items	Specifications
Certification	RF Certification	FCC/CE-RED/IC/TELEC/KCC/SRRC/NCC
	Wi-Fi Certification	Wi-Fi Alliance
	Bluetooth certification	BQB
	Green Certification	REACH/RoHS
Test	Reliability	HTOL/HTSL@HAST/TC/ESD
Wi-Fi	Protocols	802.11 b/g/n (802.11n up to 150 Mbps)
		A-MPDU and A-MSDU aggregation and 0.4 μ s guard interval support
	Frequency range	2.4 GHz ~ 2.5 GHz
Bluetooth	Protocols	Bluetooth v4.2 BR/EDR and BLE specification
	Radio	NZIF receiver with -97 dBm sensitivity
		Class-1, class-2 and class-3 transmitter
		AFH
Audio	CVSD and SBC	
Hardware	Module interfaces	SD card, UART, SPI, SDIO, I ² C, LED PWM, Motor PWM, I ² S, IR, pulse counter, GPIO, capacitive touch sensor, ADC, DAC
	On-chip sensor	Hall sensor
	Integrated crystal	40 MHz crystal
	Integrated SPI flash ¹	4 MB
	Operating voltage/Power supply	2.7 V ~ 3.6 V
	Operating current	Average: 80 mA
	Minimum current delivered by power supply	500 mA
Recommended operating temperature range ²	-40 °C ~ +85 °C	

Notice:

1. ESP32-WROOM-32D and ESP32-WROOM-32U with 8 MB flash or 16 MB flash are available for custom order.
2. ESP32-WROOM-32D and ESP32-WROOM-32U with high temperature range (-40 °C ~ +105 °C) option are available for custom order. 4 MB SPI flash is supported on the high temperature range version.
3. For detailed ordering information, please see [Espressif Product Ordering Information](#).

ANEXO L: Código fuente de Tarjeta ESP32

```
#define DEBUG_ARRAY(a) { for (int index = 0; index <
sizeof(a) / sizeof(a[0]); index++) {
Serial.print(a[index]); Serial.print('\t'); }
Serial.println(); };
////////////////////////////////////
#include <WiFi.h>
#include <ThingSpeak.h>
////////////////////////////////////
#include <Wire.h>
////////////////////////////////////
WiFiClient client;
unsigned long myChannelNumber = 1387476;
const char * myWriteAPIKey = "P4FV7CSCA3I4SQRX";
////////////////////////////////////
String str = "";
const char separator = ',';
const int dataLength = 4;
float data[dataLength];
////////////////////////////////////
float BPM, SpO2, tempC, Aire;
////////////////////////////////////
int contconexion = 0;
int cont=0,dato_rx=0,valor;
////////////////////////////////////
char ssid[] = "RED PIAZA";
char pass[] = "BLACKCM94@";
////////////////////////////////////
unsigned long previousMillis = 0;
////////////////////////////////////
uint32_t tsLastReport = 0;
#define REPORTING_PERIOD_MS 5000
////////////////////////////////////

void setup()
{
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.begin(115200);

    while (WiFi.status() != WL_CONNECTED)
    {
      delay(1000);
      Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi OK");
  }
  ThingSpeak.begin(client); // Initialize ThingSpeak
}
```

```

void loop()
{
    if (Serial.available())
    {
        str = Serial.readStringUntil('\n');
        for (int i = 0; i < dataLength ; i++)
        {
            int index = str.indexOf(separator);
            data[i] = str.substring(0, index).toFloat();
            str = str.substring(index + 1);
        }
    }

    BPM = data[0];
    SpO2 = data[1];
    tempC = data[2];
    Aire = data[3];

    //////////////////////////////////////
    ThingSpeak.setField(1, BPM);
    ThingSpeak.setField(2, SpO2);
    ThingSpeak.setField(3, tempC);
    ThingSpeak.setField(4, Aire);

    int x = ThingSpeak.writeFields(myChannelNumber,
myWriteAPIKey);
    if(x == 200){
        Serial.println("Channel update successful.");
    }
    else{
        Serial.println("Problem updating channel. HTTP error code " +
String(x));
    }
    delay (5000);
    //////////////////////////////////////
}

```

ANEXO M: Código fuente de MatLab Analysis en Plataforma ThingSpeak

```
channelID = 1387476;

iftttURL =
'https://maker.ifttt.com/trigger/report_xls/with/key/b7JL5AvKTdOCdwDcMq
673c';

readAPIKey = 'RHC9M08AKDEECSC0';

thingSpeakURL =
strcat('https://api.thingspeak.com/channels/',string(channelID),'//field
s/1/last.txt');
thingSpeakURL1 =
strcat('https://api.thingspeak.com/channels/',string(channelID),'//field
s/2/last.txt');
thingSpeakURL2 =
strcat('https://api.thingspeak.com/channels/',string(channelID),'//field
s/3/last.txt');

lastValue = str2double(webread(thingSpeakURL, 'api_key', readAPIKey));
lastValue1 = str2double(webread(thingSpeakURL1, 'api_key',
readAPIKey));
lastValue2 = str2double(webread(thingSpeakURL2, 'api_key',
readAPIKey));

webwrite(iftttURL, 'value1',lastValue, 'value2', lastValue1, 'value3',
lastValue2);
```