



**FACULTAD DE INGENIERÍA  
INGENIERÍA EN AUTOMATIZACIÓN Y ROBÓTICA**

**“Prototipado de un Sistema de Monitoreo de Signos Vitales Inalámbrico”**

**ALEX EDUARDO VEGA PROVOSTE**

**PROFESOR GUÍA: MIGUEL ANDRES SOLIS CID**

**MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO EN AUTOMATIZACIÓN Y ROBÓTICA**

**SANTIAGO – CHILE  
Diciembre, 2020**



**FACULTAD DE INGENIERÍA  
INGENIERÍA EN AUTOMATIZACIÓN Y ROBÓTICA**

**DECLARACIÓN DE ORIGINALIDAD Y PROPIEDAD**

Yo, **ALEX EDUARDO VEGA PROVOSTE**, declaro que este documento no incorpora material de otros autores sin identificar debidamente la fuente.

Santiago, 07 de diciembre de 2020

---

Firma del alumno

*Dedicatoria*

*Este proyecto está dedicado a mi querido tío y a todas las personas que me acompañaron en esta aventura.*

**AGRADECIMIENTOS**

Dedico este trabajo en primer lugar a Dios, quien me ha dado salud y fuerza para superar todos los momentos difíciles que encontré durante mis años como estudiante. A mi madre que a pesar de todas las dificultades me ayudó a realizar mis sueños, a mi padre por todos los años de apoyo moral que me brindó, a mi hermana por darme todo el apoyo en este proceso, a mi pareja por darme su paciencia, comprensión y amor para poder seguir adelante frente a todas las dificultades que tuve durante mi proceso de formación profesional y por último a toda mi familia y conocidos por alentarme a ser una mejor persona y no renunciar a mis sueños.

## ÍNDICE GENERAL

Capítulo 1: Antecedentes Generales. ....	10
1.1 Introducción.....	10
1.2 Importancia de resolver el problema de monitorear los signos vitales. ....	11
1.3 Breve discusión bibliográfica.....	11
1.4 Contribución del trabajo .....	14
1.5 Objetivos .....	15
1.5.1 Objetivo General.....	15
1.5.2 Objetivos Específicos .....	15
1.6 Organización y presentación de este trabajo .....	16
Capítulo 2: MARCO TEÓRICO.....	17
2.1 Monitoreo de Signos Vitales en el área de la salud.....	17
2.2 Monitor De Signos Vitales .....	18
2.2.1 Clasificación de Monitores. ....	18
2.2.2 Tipos de Monitores en el Mercado.....	19
2.3 Variables Fisiológicas.....	19
2.3.1 Frecuencia Cardíaca.....	20
2.3.2 Frecuencia Respiratoria.....	21
2.3.3 Presión Arterial .....	22
2.3.4 Temperatura Corporal Exterior. ....	23
2.4 Planificación del trabajo de desarrollo.....	24
2.5 Descripción de problemas y oportunidades de mejora.....	25
2.6 Limitaciones y alcances del proyecto .....	26
2.7 Normativa y leyes asociadas al proyecto .....	26
Capítulo 3: Tecnología.....	27
3.1 Sistema Operativo.....	27
3.2 Placa de Desarrollo. ....	27
3.3 Sensores Encargados de Medir las Variables Fisiológicas. ....	27
3.3.1 Funcionamiento de los sensores para realizar un Electrocardiograma. ....	29
3.3.2 funcionamiento de los sensores para medir la saturación de oxígeno. ....	32
Capítulo 4: Hardware del Proyecto .....	35

4.1 Sensores Biomédicos.....	35
4.1.1 Sensor AD8232 ECG.....	35
4.1.2 Sensor MAX30102.....	37
4.2 Kit NodeMCU V2.....	39
4.3 Raspberry PI.....	41
4.3.1 Hardware.....	42
4.3.2 Software.....	43
4.3.3 Raspberry Pi Zero W.....	43
Capítulo 5: Software del proyecto.....	44
5.1 Arduino IDE.....	44
5.2 Raspbian.....	45
5.3 Mosquitto.....	45
5.3.1 MQTT.....	46
5.3.2 Funcionamiento de MQTT.....	46
5.3.3 Estructura de un mensaje MQTT.....	47
5.4 Python.....	48
5.5 InfluxDB.....	48
5.6 Grafana.....	49
Capítulo 6: Ingeniería del Proyecto.....	51
6.1 Configuración de NodeMCU y los Sensores.....	52
6.1.1 Conexión a la Red Inalámbrica Wifi.....	52
6.1.2 Esquema del Circuito del Prototipo.....	52
6.1.3 Configuración del sensor Biomédico AD8232.....	54
6.1.4 Configuración del Sensor Biomédico MAX30102.....	54
6.1.5 Conexión al Broker utilizando MQTT.....	57
6.2 Configuración de Raspberry PI Zero W.....	59
6.2.1 Comprobación conexión MQTT y InfluxDB.....	61
6.2.2 configuración de Grafana.....	62
Capítulo 7: Discusión de Resultados y Conclusiones Generales.....	64
7.1 Discusión de Resultado.....	65
7.1.1 Resultado sensor AD8232 ECG.....	65
7.1.2 Resultado sensor MAX30102.....	67

7.1.3 Resultado conexión entre NodeMCU, Mosquitto y la base de dato InfluxDB. ....	67
7.1.4 Resultado de los datos procesados graficados utilizando Grafana .....	70
7.2 Conclusiones Generales .....	70
7.2.1 Trabajo Futuro .....	71
GLOSARIO .....	72
Apéndice 1: Configuración de Arduino IDE para usar NodeMCU .....	74
Apéndice 2: Instalación de Raspbian y Configuración IP Estática .....	76
Apéndice 3: Instalación del Broker Mosquitto .....	78
Apéndice 4: Instalación de InfluxDB .....	81
Apéndice 5: Instalación Grafana .....	83
REFERENCIAS BIBLIOGRAFICAS .....	84
ANEXOS .....	88
10.1 Programación del NodeMCU.....	88
10.2 Script de Python para comunicar MQTT con InfluxDB .....	90

## ÍNDICE DE FIGURAS

Figura 1, Esquema del galvanómetro de Einthoven. ....	17
Figura 2, Pulsos periféricos. ....	21
Figura 3, Frecuencia Respiratoria. ....	22
Figura 4, Presión arterial. (Cardiavant, 2020) ....	23
Figura 5, Temperatura Corporal Medida con Termómetro. ....	23
Figura 6, Funcionamiento de un sensor. ....	28
Figura 7, Diferencia de señales análogas y digitales. ....	29
Figura 8, Ondas P, Q, R, S, T, y U. ....	30
Figura 9: Posición de los electrodos ECG. ....	31
Figura 10, Derivaciones precordiales. ....	31
Figura 11, Calculo SpO2. ....	32
Figura 12, Posición transmisiva y reflectiva. ....	33
Figura 13, Absorción de luz en la sangre. ....	33
Figura 14, Pulso oxímetro. ....	34
Figura 15, Modulo AD8232 de la marca Sparkfun. ....	36
Figura 16, Cable de sensor con 3 electrodos. ....	36
Figura 17, Sensor MAX30102. ....	37
Figura 18, Diagrama de bloque Sensor Max30102. ....	38
Figura 19, Método Reflectivo. ....	38
Figura 20, Longitudes de onda de cada diodo. ....	39
Figura 21, Respuesta del fotodiodo del Max30102. ....	39
Figura 22, Modulo NodeMCU. ....	40
Figura 23, Pines NodeMCU. ....	41
Figura 24, Raspberry PI modelo 4 y pines GPO's. ....	42
Figura 25, Hardware de Raspberry PI. ....	43
Figura 26, Sistema Operativo Raspbian. ....	43
Figura 27, Raspberry PI Zero W. ....	44
Figura 28, Arduino IDE. ....	45

Figura 29, Raspbian de Raspberry pi. ....	45
Figura 30, Message Queue.....	46
Figura 31, Conexión entre el cliente y el Broker. ....	47
Figura 32, a) Conexión entre cliente y Broker, b) comunicación entre suscriptor y Broker. ....	47
Figura 33, Estructura MQTT. ....	48
Figura 34, Python.....	48
Figura 35, Formato de base de datos InfluxDB.....	49
Figura 36, InfluxDB. ....	49
Figura 37, Grafana. ....	50
Figura 38, Dashboard de grafana. ....	50
Figura 39, Diagrama del sistema de monitoreo de signos vitales. ....	51
Figura 40, Configuración Wifi para NodeMCU. ....	52
Figura 41, Circuito del Prototipo.....	53
Figura 42, Void Setup sensor AD8232.....	54
Figura 43, Void Loop de AD8232.....	54
figura 44, Comunicación I2C.....	55
Figura 45, Programación de SpO2. ....	56
Figura 46, Librería HeartRate para el MAX30102.....	57
Figura 47, Calculo pulso por minuto ....	57
Figura 48, Comunicación MQTT.....	58
Figura 49, Void Connect_MQTT. ....	58
Figura 50, Asignación de los Tópicos. ....	59
Figura 51, Creación de usuario, clave y base de datos. ....	60
Figura 52, Datos de MQTT e InfluxDB.....	60
Figura 53, Datos del sensor. ....	60
Figura 54, Realizar consultas en InfluxDB. ....	62
Figura 55, GUI de grafana. ....	62
Figura 56, Fuente de datos.....	63



Figura 57, Panel de control.....	64
Figura 58, Interfaz gráfica del monitor de signos vitales.....	64
Figura 59, Triangulo de Einthoven para un Electrocardiograma.....	65
Figura 60, Primera Prueba de medición.....	66
Figura 61, Ondas P y T en la forma a).....	66
Figura 62, Segunda prueba de medición.....	66
Figura 63, Datos fisiológicos procesados por NodeMCU.....	68
Figura 64, Conexión exitosa entre NodeMCU y Raspbian utilizando Mosquitto.....	69
Figura 65, Comunicación exitosa entre MQTT e InfluxDB.....	69
Figura 66, Datos guardados en InfluxDB.....	70
Figura 67, Panel de control.....	70
Figura 68, Preferencias de Arduino IDE.....	75
Figura 69, Gestor de URLs Adicionales de Tarjetas.....	75
Figura 70, Gestor de Tarjeta.....	75
Figura 71, Sistema operativo Raspbian e instalador Noobs.....	76
Figura 72, Escritorio Raspbian.....	77
Figura 73, IP y MASK de Raspberry PI.....	77
Figura 74, IP del Router.....	78
Figura 75, Configuración IP estática.....	78
Figura 76, Eclipse Mosquitto.....	79
Figura 77, Descarga del Fichero Mosquitto.....	79
Figura 78, Configuración de Mosquitto.....	80
Figura 79, Configuración de mosquitto.....	81
Figura 80, Comunicación exitosa MQTT.....	81
Figura 81, Repositorio InfluxData.....	82
Figura 82, Instalación InfluxDB.....	83

## Índice de tablas

Tabla 1, Diferencia entre placas de desarrollo. ....	13
Tabla 2, Diferencia tarjeta de desarrollo. ....	14
Tabla 3, Diferencias entre monitores de signos vitales. ....	19
Tabla 4, Valores normales de frecuencia cardiaca. ....	21
Tabla 5, Planificación del trabajo. ....	24
Tabla 6, Conexión AD8232. ....	36
Tabla 7, Características Técnicas NodeMCU. ....	40
Tabla 8, Conexión de sensores. ....	53
Tabla 9, Posición de los Electrodo. ....	65
Tabla 10, Comparación MAX30102 y un Pulsómetro genérico ....	67

## Resumen

El monitoreo de signos vitales es considerado uno de los controles médicos más importantes que permite observar el estado de salud en tiempo real de cada paciente. Generalmente los hospitales públicos tienen estos instrumentos para un número limitado de pacientes por lo que a veces no logran satisfacer la demanda que tienen diariamente, ya sea por su costo o incluso no tiene el financiamiento necesario para ayudar a un gran número de personas. Por estas razones se propone realizar un dispositivo económico y de fácil utilización para la medición de estos parámetros.

En este proyecto se diseñará un prototipo de sistema de monitoreo de signos vitales de forma inalámbrica que permitirá obtener los datos de las variables fisiológicas y graficarlas para poder monitorearlas en cualquier parte utilizando una red local.

Dentro de las variables fisiológicas mediremos la frecuencia cardiaca, temperatura corporal, saturación de oxígeno y podrá realizar electrocardiograma. Para el procesamiento de datos se utilizará el microcontrolador NodeMCU el cual procesará los datos obtenidos y los transmitirá por comunicación MQTT a la placa de desarrollo Raspberry PI Zero W el cual trabajará como servidor. Los datos procesados serán respaldados en la base de datos InfluxDB que permitirá guardar todos los registros en tiempo real y, por último, se utilizará Grafana para poder visualizar y graficar los datos guardados en aquella base de datos.

## Capítulo 1: Antecedentes Generales.

### 1.1 Introducción

En el área de la salud siempre ha sido importante conocer a cada instante la evolución fisiológica de los pacientes, esto le permite al personal de salud valorar de forma integral las condiciones generales y específicas de cada paciente respecto a su patología y junto con esto tomar decisiones para su tratamiento y posterior bienestar.

Por otro lado, a lo largo del tiempo, la población y sus necesidades han pasado por modificaciones, siendo una de las necesidades básicas del ser humano adquirir un servicio de salud adecuado, oportuno y de calidad. Las personas recurren a entidades públicas o privadas dependiendo de sus recursos económicos, las cuales deben velar por su salud y otorgar un diagnóstico, tratamiento, y gracias a eso su posterior mejora, de no ser así, colaborar a que el usuario mantenga una óptima calidad de vida dentro de lo que su patología le permita.

Conforme al transcurso de los años el avance tecnológico en salud es cada vez más rápido, favoreciendo que la atención de los pacientes sea mucho más expedita y eficiente, aportando un diagnóstico certero en un periodo de tiempo prudente, logrando de esta manera ofrecer un tratamiento efectivo para la respectiva patología.

En la actualidad se conocen distintas tecnologías capaces de brindar un mejor servicio para cada usuario, como por ejemplo las prótesis. En Chile la institución Teletón se dedica a realizar prótesis para personas que nacen con algún grado de discapacidad o que son víctimas de accidentes (Dote et al, 2020), la realidad aumentada que se está utilizando actualmente en universidades en el área de la medicina (Almenara et al, 2017), la tecnología "Wearable" que permite realizar medidas de variables fisiológicas de uso no invasivo como son los Smartwach deportivos (Abril, 2019), herramientas de prototipado, la nanotecnología aplicada en la medicina, entre otras que han surgido estos últimos años, gracias a estos avances se ha producido un cambio en la forma de trabajar, relacionarse, aprender y cuidar de la salud de los pacientes.

El concepto que ha nacido gracias a la unión de estos dos campos es la electromedicina (Benito, 2017), área que se ha desarrollado en estos últimos años, teniendo importantes avances que han mejorado la calidad de vida de los pacientes. Los objetivos de la electromedicina es brindar un servicio que permita realizar un chequeo médico y diagnosticar diversos padecimientos por medio de tecnologías,

Comentado [s1]: M

Comentado [s2]: M

Comentado [s3]: M

Comentado [MSC4]: Mejor servicio para cada usuario, en distintos tipos de servicio?

Comentado [MSC5R4]:

Comentado [avp6]: Se repite

Comentado [avp7]: Se repite

que favorecen a mejorar la salud. Uno de los instrumentos de medición que ha desarrollado esta área es el monitor de signos vitales (Benito, 2017).

El control de signos vitales es fundamental desde los inicios de la enfermería y la medicina, esta simple pero importante tarea ha pasado por cambios tecnológicos, que han sido propicios tanto para el equipo sanitario como para los pacientes ya que les permite estar en constante conocimiento de las variables fisiológicas y por ende del estado actual del usuario.

Comentado [avp8]: Se repite

Comentado [s9]: M

Comentado [s10]: M

Comentado [s11]: M

### 1.2 Importancia de resolver el problema de monitorear los signos vitales.

Algunos centros hospitalarios del país tienen la capacidad de implementar salas de atención con equipos médicos especializados, lo cual permite que las personas puedan recibir atención de una manera adecuada y segura, respondiendo a la alta demanda en lugares donde la densidad de población sea mayor.

El problema surge cuando la persona no puede trasladarse a los centros médicos por variados motivos: falta de transporte público o privado, pacientes postrados, pacientes inmunodeprimidos, entre otros, requiriendo adquirir el equipo correspondiente capaz de monitorear sus signos vitales constantemente, es en ese momento cuando es necesaria la implementación de un sistema de monitoreo sin tener que contratar a personal capacitado del área de salud, para poder realizar dicha actividad, si no que este sistema sea tan fácil de manejar de tal modo que el cuidador pueda controlar al paciente, visualizar su estado actual y tomar decisiones para su cuidado, requiriendo de una simple capacitación para el manejo del equipo y conocer los parámetros normales de cada signo vital para que pueda evidenciar cualquier cambio en el estado de salud del paciente.

Otro factor importante que se puede rescatar gracias a un sistema inalámbrico de monitoreo de signos vitales es la recopilación de información constante, esto permite tener respaldo del momento exacto en que ocurre una anomalía en el usuario. Gracias a esto el cuerpo médico podrá validar y determinar un diagnóstico adecuado a dicha irregularidad.

Comentado [MSC12]: Aquí sugeriría dejarlo más "general" (como "Algunos centros hospitalarios.."), o incluir fuente/estudio que indique que al menos más de la mitad de centros hospitalarios tienen dicha capacidad

Comentado [MSC13R12]:

Comentado [s14]: M

Comentado [s15]: M

Comentado [s16]:

Comentado [s17]: M

### 1.3 Breve discusión bibliográfica

A lo largo de esta investigación existen muchas fuentes de información que tienen como objetivo implementar un prototipo de sistema de monitoreo de signos vitales con comunicación inalámbrica, utilizando distintas tecnologías, software y tipos de comunicación que permite conectar diferentes dispositivos. Otro punto importante que se discute al momento de desarrollar dicho prototipo es el área donde lo quieren implementar o al tipo de paciente que se desea monitorear, muchas veces los

Comentado [s18]: M

Comentado [avp19]: Software o softwares

dispositivos que se encuentran en el comercio están enfocados al área deportiva, monitoreando los signos vitales mientras ejercen alguna actividad física, en el área de la salud, todos los hospitales y clínicas cuentan con servicio de monitoreo de signos vitales para cada paciente.

Uno de los temas que se pueden cuestionar dentro de las distintas fuentes es qué tipo de signos vitales pueden monitorearse utilizando sensores capaces de conectarse y entregar dicha información a un controlador, esto se discute debido a que existen signos vitales como la frecuencia respiratoria, la cual es medible a través de la observación, contando los números de respiraciones por minuto, observando el tipo de ruido o esfuerzo respiratorio. (Parra, 2016)

En conocimiento de que cada fuente bibliográfica utiliza algunos signos vitales que puedan ser monitoreados a través de un sensor, lo que genera mayor discordancia es si los sensores serán invasivos o no invasivos, esta problemática ocurre porque al tener sensores conectados en pacientes, le puede dificultar en varios aspectos fisiológicos como la movilidad, comodidad del paciente y la salud como tal, esto pasa normalmente con los electrodos que utiliza el electrocardiograma, cuando un paciente tiene por mucho tiempo los electrodos en la piel, es posible que le produzca un tipo de comezón o irritación en la piel que afecte el bien estar del paciente. Debido a ese problema se llega a la conclusión de que el mejor método de implementar esos tipos de sensores es del modo no invasivo, en el capítulo dos se explicará estos tipos de sensores. (Gutiérrez, 2016)

Para obtener todos estos datos fisiológicos es necesario utilizar ciertos tipos de sensores con una buena sensibilidad y que sean capaces de conectarse a un controlador o microcontrolador, por estas razones las distintas fuentes bibliográficas proponen ciertos microcontroladores tales como la serie PIC, Arduino o Raspberry, que sean capaces de cumplir con la función de adquirir y comunicar los datos entregados por las variables fisiológicas. Estos controladores cumplen con particulares características necesarias para cada fuente bibliográfica, un ejemplo de esta puede ser la cantidad de memoria, los números de pines, tamaño, tipo comunicación, entre otras características necesarias para cada proyecto. En la siguiente tabla se mostrarán las diferencias entre las placas de programación:

Comentado [s20]: M

Comentado [s21]: M

Comentado [s22]: M

Comentado [s23]: M

Comentado [s24]: M

Comentado [s25]: M

Comentado [s26]: M

Comentado [s27]: M

Comentado [s28]: M

Comentado [s29]: M

Comentado [s30]: M

Comentado [s31]: M

Nombre	Arduino	Arduino	Arduino	Arduino
Modelo	Pro mini	Uno R3	Mega 2560 R3	Micro
Precio	\$12.990	\$19.990	\$39.990	\$19.990
Dimensiones	18x33mm	68.6x53.4mm	102x53mm	18x33mm
Procesador	Atmega 328	Atmega 328	Atmega 2560	Atmega 32U4
Clock Speed	16MHz	16MHz	16MHz	16MHz
RAM	2KB	2KB	8KB	2.5KB
Flash	32KB	32KB	256KB	32KB
EEPROM	1KB	1KB	4KB	1KB
Voltaje in	3.3V - 5V	7V - 12V	7V - 12V	3.3V - 5V
Min Power	150mA	42mA	42mA	40mA
Pines Digitales	14	14	54	20
Pines Análogos	6	6	16	12
PWM	6	6	15	7
TWI/I2C	1	1	1	1
SPI	1	1	1	1
UART	1	1	4	1
Dev IDE	Arduino Tool	Arduino Tool	Arduino Tool	Arduino Tool
Ethernet	No	No	No	No
USB Master	No	No	No	No

Tabla 1, Diferencia entre placas de desarrollo. (MCIElectronic, 2020)

También existen las tarjetas de desarrollo capaces de cumplir más funciones que la serie de Arduino, como tener la capacidad de instalar sistemas operativos, conectarse a internet sin necesidad de periféricos, entrada y salida de video y audio, entre otros. En la siguiente tabla se explicarán las diferencias existentes entre estas tarjetas de desarrollo:

CARACTERISITICAS	RASPBERRY PI	BEAGLEBOARD	ORANGE PI
MODELO	3 B+	BEAGLEBON BLACK	ONE H3
PRECIO	\$43.990	\$86.990	\$29.800
GPU	1,4GHZ	1GHZ	600MHZ
MEMORIA	1 GB DDR2	512 MB DDR3	1 GB DDR3
ALMACENAMIENTO	TARJETA SD	TARJETA SD	TARJETA SD
RED A BORDO	ETHERNET/WIFI	ETHERNET	ETHERNET
ENTRADA DE VIDEO	SI	NO	SI
ENTRADA DE AUDIO	SI	NO	SI
SALIDA DE VIDEO	SI	SI	SI
SALIDA DE AUDIO	SI	SI	SI
FUENTE DE ALIMENTACIÓN	5V	5V	5V
PUERTOS USB 2.0	4	1	3
BOTONES	NO	SI	SI

Comentado [s32]: M

Comentado [avp33R32]:

Comentado [s34]: M

Comentado [s35]: M

PERIFERICOS	SI	SI	SI
PIN GPIO	40 PINES	46 PINES	40 PINES
OS	LINUX	LINUX	LINUX

Tabla 2, Diferencia tarjeta de desarrollo.

Según la investigación, la Raspberry PI cumple con todos los requisitos necesarios para conectar cada tipo de sensor, y gracias a su conectividad WIFI es capaz de trabajar como servidor. Su única desventaja en comparación con la Placa Arduino es la cantidad de pines de entrada y salida análoga/digital. (García et al, 2018)

Para que todos los datos entregados por los sensores sean procesados y visualizados por el personal de salud o cuidador a cargo del paciente que esté utilizando este prototipo, las distintas fuentes bibliográficas entregan diversas soluciones útiles para desarrollar este prototipo, la más común es utilizar bases de datos capaces de conectarse a una página web para observar los datos procesados por el microcontrolador.

#### 1.4 Contribución del trabajo

Si bien los sistemas de monitoreo de signos vitales existen en cada hospital y clínica del país, siendo capaces de cumplir funciones de controlar uno a más parámetros, otros con la capacidad de utilizar batería de litio de alta duración, a veces llegando a durar 8 horas continuas así permitiendo transportar de un área a otro sin problema de necesitar una fuente de poder cercana. En el sistema público a menudo los equipos de monitoreo no son suficientes para satisfacer la demanda de los pacientes y para dar solución a dicho contratiempo se piensa implementar un prototipo de sistema inalámbrico de monitoreo de signos vitales.

Lo novedoso de este proyecto es brindar a las personas una mayor comodidad en la implementación de esta tecnología, permitiendo monitorear desde cualquier parte del hospital, también facilita que los familiares puedan tener un mejor monitoreo de los signos vitales y con este sistema podrán determinar si el paciente ha sufrido alguna irregularidad en sus variables biológicas.

Comentado [s36]: M

Comentado [s37]: M

Comentado [s38]: M

Comentado [s39]: M

Comentado [s40]: M

Comentado [s41]: M

Comentado [s42]: M

Comentado [s43]: M

Comentado [s44]: M

Comentado [s45]: M



## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Desarrollar un prototipo para un sistema de monitoreo de signos vitales, el cual tiene como función medir distintas variables fisiológicas presentes en el cuerpo humano, utilizando distintas tecnologías capaces de realizar lectura de frecuencia cardiaca, oxigenación en sangre y temperatura.

### **1.5.2 Objetivos Específicos**

- Estudiar uso y avances de las distintas tecnologías asociadas al tema de monitorización de signos vitales y conocer los distintos signos vitales presentes en el cuerpo humano.
- Evaluar las características técnicas de sensores no invasivos, microcontroladores, protocolos de comunicación y dispositivo de comunicación inalámbrica para escoger la precisión adecuada en aplicaciones de monitoreo de signos vitales en “campo clínico”.
- Desarrollar un sistema de monitoreo para medir la frecuencia cardiaca, oxigenación en sangre, temperatura corporal y realizar electrocardiograma.
- Realizar una interfaz gráfica, que permita visualizar los datos procesados en base a los datos obtenidos por el microcontrolador.

## 1.6 Organización y presentación de este trabajo

El proyecto de monitoreo de signos vitales cuenta con siete capítulos donde se explica la investigación del tema detallando aspectos de las variables fisiológicas, los tipos de monitores existentes, el funcionamiento de los sensores y el desarrollo de proyecto.

- En el capítulo 1 se explica la importancia que tiene el personal de salud para responder a todas las patologías que un paciente puede llegar a padecer. También se explica como la tecnología permite al equipo médico a tomar la mejor decisión frente a una enfermedad. Además, contiene las discusiones bibliográficas donde se exponen los diferentes proyectos relacionados con el monitoreo de signos vitales y por ultimo los objetivos de este proyecto.
- En el capítulo 2, correspondiente al marco teórico del proyecto, se explica como la electrónica se introdujo en la medicina, también se exponen todas las variedades de monitores de signos vitales presentes en el mercado y las variables fisiológicas más importantes que tiene el cuerpo humano. Dentro de este capítulo se encuentra además el plan de trabajo para el primer prototipado del proyecto, la descripción del problema propuesto, las limitaciones y alcance del proyecto.
- En el capítulo 3 se explica el tipo de tecnologías presentes en los monitores y el funcionamiento general de los sensores encargados de hacer electrocardiogramas y de medir la saturación de oxígeno.
- Los capítulos 4 y 5 del proyecto presentan el hardware y software que se utilizarán para este proyecto.
- El capítulo 6 describe el desarrollo del monitoreo de signos vitales inalámbrico, en él se explicará las configuraciones de los microcontroladores y de los sensores.
- El capítulo 7 está centrado en los resultados de los sensores, microcontroladores, la información de la base de datos y las visualizaciones de los datos fisiológicos. También contará con conclusiones de los resultados obtenidos y las posibles mejoras a futuro.

Comentado [s46]: M

Comentado [s47]: M

Comentado [s48]: M

Comentado [s49]: M

Comentado [s50]: M

Comentado [s51]: M

Comentado [s52]: M

Comentado [s53]: M

Comentado [s54]: M

## Capítulo 2: MARCO TEÓRICO

### 2.1 Monitoreo de Signos Vitales en el área de la salud.

Para entender el objetivo de controlar los signos vitales, primero es necesario conocer la historia donde por primera vez se introdujo la electrónica y electricidad en la medicina. La implementación de aparatos electrónicos para diagnosticar fue introducida en 1903, gracias a Willem Einthoven (1860 – 1927), fue médico y fisiólogo ganador del premio nobel de medicina y fisiología en 1924. (Acevedo, 2017). Desarrolló los primeros registros de la actividad eléctrica del corazón, conocido actualmente como electrocardiograma, utilizando un galvanómetro modificando por él mismo cambiando la media espiral de la bobina en un hilo cubierto de plata, extendido entre dos soportes y todo esto sometido a un campo electromagnético de un electroimán, como se ilustra en la Figura 1.

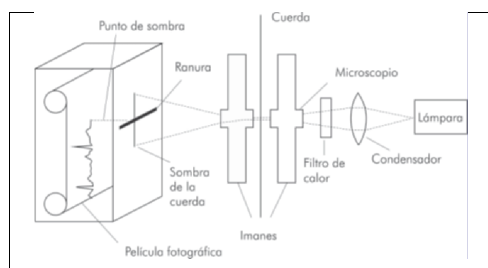


Figura 1, Esquema del galvanómetro de Einthoven. (Reynolds et al, 2011)

Durante la segunda guerra mundial, los avances tecnológicos fueron aumentando gracias al desarrollo de la industria electrónica permitiendo a los científicos y biólogos trabajar en laboratorios en contacto a la electrónica. Tras el término de la guerra y las consecuencias de las bombas atómicas, nació una nueva disciplina llamada ingeniería biomédica. (ARENI, 1996).

Con la creación de esta nueva rama de la ingeniería, los avances tecnológicos para el área de la salud fueron cada vez más sofisticados como el monitor de signos vitales, los cuales benefician en gran medida tanto al paciente como al equipo de salud, ya que proporciona datos de vital importancia sobre el estado de salud del usuario, lo que permite identificar anomalías e intervenir a tiempo.

Comentado [s55]: M

Comentado [s56]: M

Comentado [s57]: M

## 2.2 Monitor De Signos Vitales

El monitor de signos vitales surgió gracias a la evolución tecnológica y a la necesidad que tenía el equipo sanitario de valorar las condiciones generales y específicas de los pacientes, estos monitores dependiendo de su configuración son capaces de adquirir, amplificar, procesar, registrar y entregan valores numéricos de la mayoría de las variables fisiológicas, la mayoría también cuenta con un sistema de alarma capaz de alertar en caso de que los signos estén por fuera de los límites deseados. (Estrada et al, 2018)

Comentado [s58]: M

Dependiendo de su configuración, existen varios tipos de monitores de signos vitales, capaces de medir variables fisiológicas generales como el electrocardiograma (ECG), frecuencia cardiaca, frecuencia respiratoria, saturación de oxígeno (SpO2), saturación venosa de oxígeno (SvO2), temperatura corporal, presión invasiva y no invasiva, entre otros.

Existen monitores preconfigurados, estos son configurados por el fabricante, no pudiendo agregar ningún parámetro adicional, también están los monitores modulares, donde el usuario es capaz de configurar e implementar parámetros adicionales a través de módulos.

Comentado [s59]: M

Comentado [s60]: M

### 2.2.1 Clasificación de Monitores.

Existen dos tipos de monitores, fijos y transportables, el primero se encuentra normalmente en la cabecera del paciente, instalados en las paredes o fijados en la máquina de anestesia y los transportables que tienen la capacidad de moverse en cualquier área. Estos se pueden encontrar en distintas zonas de un hospital o clínica: (CENETEC, 2005)

- **Anestesia:** Estos monitores son específicos para esta zona ya que están diseñados para monitorear órganos que pueden sufrir daño por la falta de oxígeno o de circulación sanguínea, estos órganos pueden ser el corazón, cerebro y riñones.
- **Adulto y pediátrico:** Se utiliza en áreas críticas, es uno de los monitores más usados y con más consumo de insumos como electrodos, brazaletes y sensores. Estos equipos deben ser flexibles para cada paciente.
- **Neonatal:** Esta área de la medicina clínica es la más diferente de las demás, su función principal es monitorear a recién nacidos y prematuros, si bien las variables fisiológicas a evaluar son las mismas que en adultos, la diferencia es en el modo que son tomadas estas variables y sus parámetros normales los cuales también varían. Otro aspecto importante es que cuentan con un algoritmo especializado para recién nacidos.

- **Intrahospitalario:** Estos monitores son utilizado en pacientes que necesitan traslado de un área a otra dentro del mismo establecimiento hospitalario, también deben contar con una batería que dure a lo más 8 horas.
- **Interhospitalario:** Estos monitores son especializados para ser utilizados en pacientes que necesitan ser trasladados de un hospital a otro, estos monitores también deben tener una entrada de 12V para conectarse a las ambulancias.

### 2.2.2 Tipos de Monitores en el Mercado.

En conocimiento de los modelos relacionados en las distintas áreas de la medicina, ahora se clasificará por sus características técnicas y diferencias en donde se encuentran:

CARACTERÍSTICAS	BASICO	INTERMEDIO	AVANZADO
CONFIGURACIÓN	Configurado/Modular	Configurado/modular	Modular
PANTALLA	8 pulgadas	10,4 pulgadas	15 pulgadas
CURVA FISIOLÓGICA	3	4 a 6	8
ELECTROCARDIOGRAMA	Si	SI	Si
SATURACIÓN DE OXÍGENO	Si	SI	Si
PLETISMOGRAFÍA	Si	SI	Si
RESPIRACIÓN	Si	SI	Si
CAPNOGRAFÍA	No	Opcional	Si
PRESIÓN INVASIVA	No	Opcional	Si
PRESIÓN NO INVASIVA	No	No	Si
TEMPERATURA	No	No	Si
GASTO CARDIACO	No	OPCIONAL	Opcional
EEG	No	No	Opcional
ÍNDICE BIESPECTRAL	No	No	Opcional
ESPIROMETRÍA	No	No	Opcional

Tabla 3, Diferencias entre monitores de signos vitales. (CENETEC, 2005)

Estando al tanto de la existencia de una gran variedad de monitores de signos vitales, capaces de cumplir distintas funciones dependiendo del área donde se esté implementando, a continuación, se explicará las variables fisiológicas más importantes de monitorear.

### 2.3 Variables Fisiológicas.

Actualmente todos los hospitales tienen como labor controlar las variables fisiológicas, teniendo equipos de monitoreo comunes y otros especializados para

Comentado [s61]: M

Comentado [s62]: M

Comentado [s63]: M

Comentado [s64]: M

obtener estos parámetros, la mayoría de estos equipos suelen ser robustos. También se debe contar con especialistas encargados de manipular este tipo de instrumento para que los datos fisiológicos entregados por el monitor sean exactos. Estos datos fisiológicos, también conocidos como signos vitales, son fundamentales para determinar el estado de salud que presenta un paciente.

Los signos vitales o variables fisiológicas son indicadores que evidencian el estado de los órganos vitales como el cerebro, corazón y pulmón, expresando de forma inmediata los cambios en el organismo, estos cambios no se pueden cualificar ni cuantificar. (Penagos, 2005). Dentro de todos los signos vitales existen cuatro que cumplen funciones principales:

- Frecuencia cardiaca.
- Frecuencia respiratoria.
- Tensión arterial.
- Temperatura.

### 2.3.1 Frecuencia Cardiaca.

La frecuencia cardiaca o pulso arterial es la onda pulsátil de la sangre, que se origina en la contracción del ventrículo izquierdo del corazón y que da como resultado la expansión y contracción regular del calibre de la arteria.

Los latidos por minuto corresponden a la frecuencia cardiaca la cual varía con la edad, sexo, actividad física, estado emocional, fiebre, medicamento y hasta hemorragias. Para poder medir esta variable existe un examen llamado electrocardiograma que permite hacer un registro gráfico de la actividad eléctrica del corazón producidas por la excitación de las células miocárdicas del corazón causando el latido cardiaco. (Penagos, 2005).

Existen ocho puntos anatómicos para la palpación del pulso: temporal, carotideo, braquial, radial, femoral, poplíteo, tibial y pedio.

Comentado [s65]: M

Comentado [s66]: M

Comentado [s67]: M

Comentado [s68]: M

Comentado [s69]: M

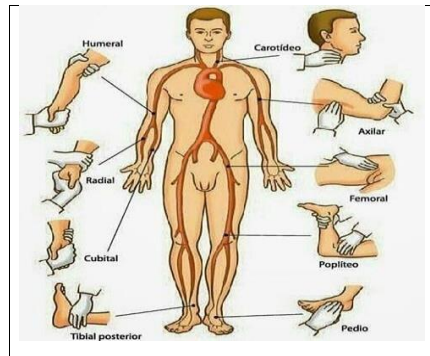


Figura 2, Pulsos periféricos. (Ferreira, 2015)

La frecuencia cardíaca dependerá en gran medida de la edad del paciente, esto debido a que sufre variaciones desde el momento del nacimiento hasta la madurez y la senectud, la siguiente tabla muestra los parámetros normales según la edad de cada paciente:

Comentado [s70]: M

Comentado [s71]:

Comentado [s72]: M

EDAD	PULSACIONES POR MINUTO
Recién nacido	120 – 170
Lactante menor	120 – 160
Lactante mayor	110 – 130
Niños de 2 a 4 años	100 – 120
Niños de 6 a 8 años	100 – 115
Adulto	60 – 80

Tabla 4, Valores normales de frecuencia cardíaca. (Penagos, 2005)

### 2.3.2 Frecuencia Respiratoria.

La frecuencia respiratoria corresponde a la cantidad de ciclos respiratorios por minuto y con esto se verifica si el paciente oscila dentro de los rangos normales o presenta alguna anomalía, cuando el paciente se encuentra dentro de los parámetros normales se considera eupneico, cuando la frecuencia respiratoria supera estos rangos, se considera taquipnea, cuando la frecuencia respiratoria esta disminuida se considera bradipnea, por último, cuando existe una ausencia de respiraciones por más de 20 segundos se considera una apnea. El ciclo respiratorio inicia con la inspiración, donde el diafragma se contrae y permite el paso del aire

Comentado [s73]: M

Comentado [s74]: M

Comentado [s75]: M

exterior hacia la cavidad torácica y termina con la expiración que es una fase pasiva y permite la eliminación de CO<sub>2</sub>. (Penagos, 2005)

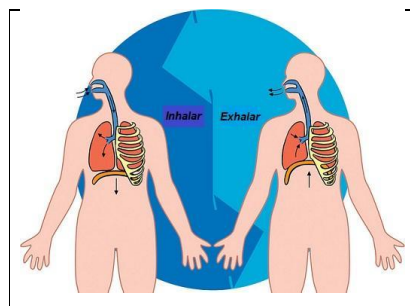


Figura 3, Frecuencia Respiratoria. (Robles, 2019)

### 2.3.3 Presión Arterial

Se conoce como presión arterial a la fuerza que ejerce la sangre sobre las paredes de los vasos sanguíneos, esta presión es importante ya que permite la circulación de la sangre por venas, arterias y capilares, aportando oxígeno y nutrientes a todos los órganos del cuerpo. Esta variable fisiológica es medida por un instrumento llamado esfigmomanómetro el cual es colocado en el brazo del paciente, después se debe poner la manguera en la arteria braquial, se debe cerrar sin apretar, teniendo todo esto instalado el paso siguiente es encontrar el pulso arterial para colocar la campana del estetoscopio, por otra parte se debe ubicar el pulso radial del mismo brazo, con la perilla del instrumento se debe insuflar hasta que se deje de sentir el pulso radial, y observar la aguja, al valor indicado por la aguja se deben sumar 30 mmHg, al llegar a la presión correspondiente, se debe abrir la perilla para escuchar los ruidos cardiacos (ruidos de Korotkoff), el primer ruido indica la presión sistólica y ultimo ruido indica la presión diastólica. (Penagos, 2005)

Comentado [s76]: M





Figura 4, Presión arterial. (Cardiavant, 2020)

#### **2.3.4 Temperatura Corporal Exterior.**

Se define temperatura corporal exterior al equilibrio entre la producción de calor por el cuerpo y su pérdida, esto se regula gracias al hipotálamo. Normalmente se toma con un termómetro, colocándolo debajo del brazo o en la boca, las temperaturas normales corresponden entre los 36 a 37 °C. (Penagos, 2005)



Figura 5, Temperatura Corporal Medida con Termómetro. (Zárate, 2020)

## 2.4 Planificación del trabajo de desarrollo

ID	% Real	% Planificado	Nombre de tarea	Duración	Comienzo	Fin
ID 1	56%	100%	<b>Monitoreo de Signos Vitales</b>	283 días	02-03-2020	31-12-2020
ID 2	100%	100%	Contacto con el profe	14 días	02-03-2020	15-03-2020
ID 3	100%	100%	Busqueda de proyectos en el area de interés	7 días	16-03-2020	22-03-2020
ID 4	100%	100%	Generación de ideas y enfoque a abordar	7 días	23-03-2020	29-03-2020
ID 5	100%	100%	<b>Introducción</b>	28 días	30-03-2020	26-04-2020
ID 6	25%	25%	Definición de objetivos generales	7 días	30-03-2020	05-04-2020
ID 7	25%	25%	Definición de objetivos específicos	7 días	06-04-2020	12-04-2020
ID 8	25%	25%	<b>Presentación 1 evaluado</b>	7 días	13-04-2020	19-04-2020
ID 9	25%	25%	Finalización del capítulo 1 "Introducción"	7 días	20-04-2020	26-04-2020
ID 10	100,00%	100%	<b>Marco Teórico</b>	73 días	27-04-2020	08-07-2020
ID 11	11,11%	11,11%	Investigación Exploratoria	7 días	27-04-2020	03-05-2020
ID 12	11,11%	11,11%	Revisión del estado de arte	7 días	04-05-2020	10-05-2020
ID 13	11,11%	11,11%	<b>Presentación 2 evaluado</b>	7 días	11-05-2020	17-05-2020
ID 14	11,11%	11,11%	Descripción de problema y mejoras	7 días	18-05-2020	24-05-2020
ID 15	11,11%	11,11%	Limitaciones y alcance del proyecto	7 días	25-05-2020	31-05-2020
ID 16	11,11%	11,11%	Normativas y leyes asociada al proyecto	7 días	01-06-2020	07-06-2020
ID 17	11,11%	11,11%	Planificación de trabajo de desarrollo	30 días	08-06-2020	07-07-2020
ID 18	11,11%	11,11%	<b>Presentación 3 evaluado</b>	1 días	08-07-2020	08-07-2020
ID 19	11,11%	11,11%	Finalización del capítulo 2 "Marco Teórico"	1 días	08-07-2020	08-07-2020
ID 20	0%	100%	<b>Tecnología</b>	77 días	08-06-2020	23-08-2020
ID 21	0%	20%	Definición de Tecnologías	54 días	08-06-2020	31-07-2020
ID 22	0%	20%	Investigar tipos de sensores	54 días	08-06-2020	31-07-2020
ID 23	0%	20%	cotización de sensores y microcontroladores	77 días	12-05-2020	31-07-2020
ID 24	0%	20%	Finalización del capítulo 3 "Tecnologías"	17 días	31-07-2020	16-08-2020
ID 25	0%	20%	Adquisición de sensores y tecnologías	7 días	17-08-2020	23-08-2020
ID 26	0%	100%	<b>Adquisición y Procesamiento de Datos</b>	35 días	24-08-2020	27-09-2020
ID 27	0%	25%	Adaptación del software a utilizar	7 días	24-08-2020	30-08-2020
ID 28	0%	25%	Acondicionamiento del hardware a utilizar	7 días	31-08-2020	06-09-2020
ID 29	0%	25%	Programación del nodo adquisidor de datos	7 días	07-09-2020	13-09-2020
ID 30	0%	25%	Pruebas Finales y ajustes del módulo adquisidor	14 días	14-09-2020	27-09-2020
ID 31	0%	100%	<b>Almacenamiento de Datos</b>	28 días	28-09-2020	25-10-2020
ID 32	0%	25%	Evaluación y Selección de Motor bd a Utilizar	7 días	28-09-2020	04-10-2020
ID 33	0%	25%	Conectividad y almacenamiento de datos en bd	7 días	05-10-2020	11-10-2020
ID 34	0%	25%	Visualización Gráfica de Datos Recolectados	7 días	12-10-2020	18-10-2020
ID 35	0%	25%	Pruebas Finales y ajustes	7 días	19-10-2020	25-10-2020
ID 36	0%	100%	<b>Desarrollo Web</b>	42 días	26-10-2020	06-12-2020
ID 37	0%	20%	Evaluación y selección de plataforma para visualización en línea	7 días	26-10-2020	01-11-2020
ID 38	0%	20%	Instalación y Configuración	7 días	02-11-2020	08-11-2020
ID 39	0%	20%	cargar contenido	7 días	09-11-2020	15-11-2020
ID 40	0%	20%	establecer conexión con base de dato	7 días	16-11-2020	22-11-2020
ID 41	0%	20%	realizar pruebas y ajustes	14 días	23-11-2020	06-12-2020

Tabla 5, Planificación del trabajo.

## 2.5 Descripción de problemas y oportunidades de mejora

Durante la investigación del proyecto existen varios factores que pueden dificultar el desarrollo de este prototipo, por ejemplo combinar el área de salud con la electrónica. En el área de la salud es necesario tener un conocimiento básico de los datos necesarios para medir las variables fisiológicas, también existe una dificultad con el lenguaje técnico que maneja el personal de salud, debido a este problema una de las posibles soluciones es hablar con un especialista en el área, como también leer distintos libros relacionados con la medicina básica.

Comentado [s77]: M

Comentado [s78]: M

Comentado [s79]: M

Comprendiendo qué tipos de signos vitales están presentes en el cuerpo humano, el siguiente paso es identificar qué tipos de variables fisiológicas son capaces de ser medidas a través de un sensor. Existen muchos sensores que sirven para hacer un prototipo de cualquier proyecto, pero lo más importante al elegir un sensor es la resolución que pueda tener, entre más resolución, es más probable que el valor que entrega dicho sensor se acerque a los valores físicos reales. Para poder solucionar el problema respecto a los sensores de signos vitales, será necesario buscar varios proveedores relacionados al área de la salud y sobre todo que vendan insumos médicos, realizar cotizaciones que permitan determinar monetariamente qué sensor conviene en relación a costo y calidad.

Comentado [s80]: M

Comentado [s81]: M

Comentado [s82]: M

Comentado [s83]: M

Ya teniendo toda la información sobre los sensores necesarios para las mediciones de signos vitales se debe cotizar un microcontrolador o tarjeta de desarrollo capaz de conectarse a los sensores y adquirir la información entregada por los mismos. Los factores que se deben considerar para la adquisición de un microcontrolador son los números de pines, análogos o digitales. Normalmente en los signos vitales las señales son análogas, por este motivo se debe buscar un microcontrolador que posea una gran cantidad de pines de entrada análoga. La conexión a internet hoy en día es muy importante ya que permite la comunicación con el entorno, lo mismo ocurre con los microcontroladores, es necesario que pueda ser capaz de conectarse a internet ya sea directamente a la red o inalámbricamente, para este proyecto se requiere una conexión inalámbrica debido a que el prototipo debe ser portátil y que pueda mandar la información a la base de datos en cualquier lugar.

Comentado [s84]: M

Comentado [s85]: M

Comentado [s86]: M

Comentado [s87]: M

Comentado [s88]: M

Lo que propone este proyecto es que no sea invasivo, permitiendo al usuario poder utilizarlo en cualquier lugar del recinto hospitalario, los sensores junto con el microcontrolador deben estar conectados en un área donde no importune, implementándolo ya sea en una polera, manga o muñequera.

Comentado [s89]: M

Comentado [s90]: M

El sistema operativo que se utilizará para el desarrollo de este proyecto debe ser compatible con la placa de desarrollo, en el caso de utilizar Raspberry PI Zero W se podrá utilizar Raspbian, este OS permite utilizar la Raspberry como un ordenador,

permitiendo programar, crear base de datos, crear consolas retro y varios proyectos de bajo costo.

Teniendo en cuenta que todos los datos recibidos por los sensores deben ser registrados, se debe realizar una comunicación inalámbrica entre el controlador que procese los datos y el microcontrolador que trabaje como base de datos. Toda esa información que recibe la Raspberry se debe enviar y guardar en una base de datos. Para eso se utilizará un script en Python que permita redirigir los datos obtenidos a la base de datos InfluxDB.

Lo más importante de este proyecto es que el personal de salud pueda visualizar estos datos en tiempo real, para esto se implementará la herramienta grafica Grafana, con esto se podrá realizar gráficos y visualizarlos en una página web.

## 2.6 Limitaciones y alcances del proyecto

Este proyecto consta de un prototipo de monitoreo de signos vitales, el cual cuenta con sensores que no cumplen con normativas que exige el campo clínico, por ende no puede ser usado como un monitor real, lo que puede hacer es entregar un apoyo a los monitores de signos vitales. Esto se debe al bajo costo de los sensores que se utilizaran.

Este proyecto constara de solo dos sensores que pueden medir cuatro variables fisiológicas, entre ellas el electrocardiograma, saturación de oxígeno, temperatura corporal y la frecuencia cardiaca. Estos tipos de sensores serán no invasivos para no incomodar al paciente. Se escogerán estos sensores debido a su importancia, valor y su facilidad de uso.

El microcontrolador que se utilizara es un NodeMCU V2 el cual tendrá la función de procesar los datos de los sensores. Este controlador solo se podrá conectar inalámbricamente a una Raspberry Pi Zero W mediante la comunicación MQTT.

Esta información será observada por el personal de salud, el mismo paciente o su cuidador.

Comentado [s91]: M

Comentado [s92]: M

Comentado [s93]: M

## 2.7 Normativa y leyes asociadas al proyecto

Existen varias normativas y leyes asociadas a este proyecto que permiten regular las acciones de una persona común, las siguientes leyes asociadas a este proyecto son las siguientes:

- **Ley Núm. 21.096:** Consagra el derecho a protección de los datos personales, esta ley permitirá resguardar los datos entregados por el prototipo de monitoreo de signos vitales.

- **Ley Núm. 20.584:** Regula los derechos y deberes que tienen las personas en relación con acciones vinculadas a su atención en salud.
- **Ley Núm.19.628:** Derecho a la privacidad, según el artículo 1 es el tratamiento de los datos de carácter personal en registros o bancos de datos por organismos público o particulares se sujetará a esta ley.

### Capítulo 3: Tecnología

#### 3.1 Sistema Operativo.

Se conoce como sistema operativo como una interfaz entre el ordenador y el usuario, que permite ejecutar programas y poder controlar dispositivos o periféricos conectado al ordenador. En la actualidad existen diversos sistemas operativos que controlan ordenadores como por ejemplo celulares o Smartwach, los más conocidos son los Windows de Microsoft, Mac OS de Apple, también existen OS para las placas de desarrollo como Debian de Linux. (Catalinas, 2002)

Comentado [s94]: Quieres decir ordenador?

Comentado [s95]: M

#### 3.2 Placa de Desarrollo.

Las placas o tarjetas de desarrollo son dispositivos que cuentan con la capacidad de instalar un sistema operativo en ellas, esto se puede porque cuentan con CPU, capacidad de almacenamientos ya sea interna o a través de una tarjeta SD, también cuentan con entradas y salidas análogas o digitales, estos permiten la interacción con dispositivos periféricos, como sensores, actuadores, en algunos casos estas placas cuentan con conexión a internet por red o inalámbrica, bluetooth. Estos equipos suelen ser muy útiles para el desarrollo de proyectos de bajo costo, utilizándolos como miniordenadores o bases de datos.

Comentado [s96]: M

Comentado [s97]: M

#### 3.3 Sensores Encargados de Medir las Variables Fisiológicas.

Para entender el funcionamiento de la medición de los signos vitales por un monitor, se debe manejar el lenguaje técnico del instrumento, estos monitores siempre están conectados con ciertos periféricos capaces de entregar y/o recibir información, la mayoría de estos periféricos son llamados sensores, que tienen como función registrar de forma directa, continua y reversible un parámetro físico, entregando lecturas en tiempo real (Conde, 1996).

Comentado [s98]: M

Los sensores son usados normalmente en circuitos eléctricos que deben comunicarse con el mundo exterior, estos son capaces de medir distintos rangos de

energía, como señales eléctricas, movimiento, radiación térmica o magnética, entre otras magnitudes físicas o eléctricas. Existe una gran variedad de sensores que se pueden clasificar como análogos o digitales y pasivos o activos, todos estos sensores transforman una señal física en una eléctrica.

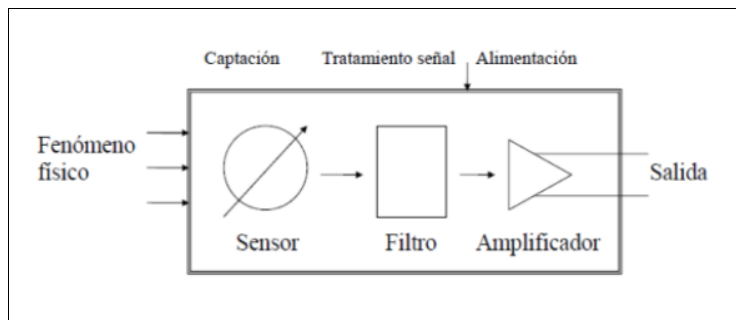


Figura 6, Funcionamiento de un sensor. (Brunete, 2020)

Los que se clasifican como activo requieren de una fuente de alimentación externa en su circuito, esto le permite producir una señal de salida, un ejemplo son los termistores y los inductores. Los sensores pasivos son capaces de generar una señal eléctrica en respuesta a factores externos, por ejemplo, las Termocuplas, y sensores Piezoeléctrico.

Los sensores análogos entregan una señal continua a diferencia de los sensores digitales que entregan señales en estado discreto, estos son capaces de producir señales binarias "1" o "0", un ejemplo de las magnitudes físicas en sensores análogos son la temperatura, desplazamiento, luminosidad, velocidad, presión. Y los digitales miden contactos, encoder, entre otros. (Vignoni, 2003).

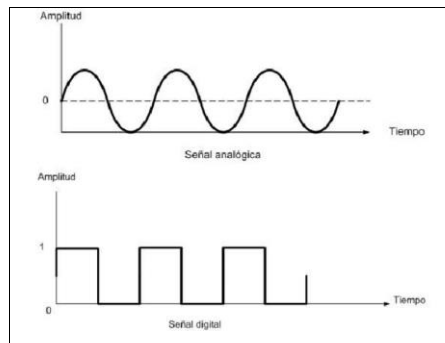


Figura 7, Diferencia de señales análogas y digitales. (Hernández, 2018)

Para poder adquirir los valores fisiológicos entregados por el cuerpo humano, es necesario contar con distintos instrumentos de medición adecuados para cada variable. Estos sensores son muy importantes para un sistema de monitoreo de signos vitales ya que cumplen con la función de adquirir los datos entregados por el cuerpo y transmitir esa información hacia los controladores y con ellos poder analizar dicha información.

Ya sabiendo qué es un sensor y los tipos de sensores que existen para medir cualquier variable, es necesario enfocarse en aquellos sensores capaces de detectar variables fisiológicas

### 3.3.1 Funcionamiento de los sensores para realizar un Electrocardiograma.

Una electrocardiograma o ECG, es un registro de la actividad eléctrica que presenta el corazón, también se considera como un conjunto de ondas donde el científico Einthoven denominó onda P, segmento QRS, onda T, y onda U.

Comentado [s99]: M

Comentado [s100]: M

Comentado [s101]: M

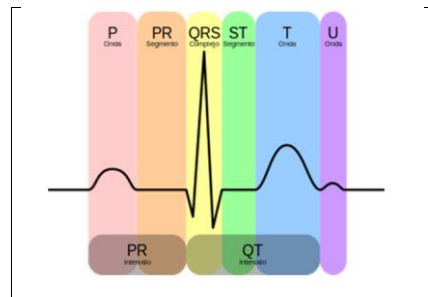


Figura 8, Ondas P, Q, R, S, T, y U. (Rozas 2007)

- **Onda P:** Es una señal eléctrica que corresponde a la despolarización auricular. Como características principales no debe superar los 0.25mV, en adulto no debe superar los 0.11 segundos y en niños no debe superar entre 0.07 a 0.09 segundos.
- **Intervalo PR:** representa el intervalo fisiológico que experimenta la conducción eléctrica del miocardio, tiene una duración entre los 0.12 a 0.20 segundos.
- **Complejo QRS:** corresponde a la corriente eléctrica que causa la contracción de los ventrículos izquierdo y derecho, la duración normal de esta onda es de 0.8 a 1.2 segundos. El complejo QRS está compuesto por tres vectores u ondas:
  1. Onda Q: Es una onda negativa, no entrega información para analizar.
  2. Onda R: Es una onda positiva, es la onda más grande.
  3. Onda S: Es una onda negativa, y representa el término del intervalo QRS.
- **Segmento ST:** representa una inactividad eléctrica de los ventrículos, su duración no tiene relevancia, pero si presenta una elevación o disminución debe considerarse anormal.
- **Onda T:** representa la repolarización ventricular, tiene una mayor amplitud y duración en comparación a la onda P, pero es más lenta, se pueden considerar asimétricas entre las dos.
- **Onda U:** Es una onda positiva en toda la derivación.

Este tipo de examen médico se realiza cuando un paciente presenta diversos tipos de dolores torácicos, disnea o arritmias cardiacas.



Para poder registrar esta actividad eléctrica se deben colocar electrodos en el tórax y en las extremidades, con la función de detectar 12 derivaciones diferentes de los aspectos de la actividad eléctrica del corazón, estos valores son registrados en papeles cuadriculados de ECG. (Zavala, 2017)

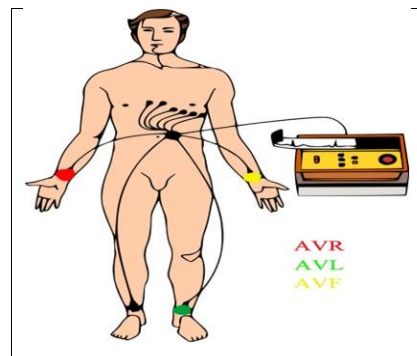


Figura 9: Posición de los electrodos ECG. (Campos, 2011)

Cada punto de se denomina derivación electrocardiográfica. La manera más efectiva de obtener un panorama completo de la actividad eléctrica del corazón es necesario tener doce derivaciones trabajando conjuntamente. De este grupo de derivaciones, tres son estándar y otras tres aumentadas, con esto es posible ver el corazón en el plano frontal.

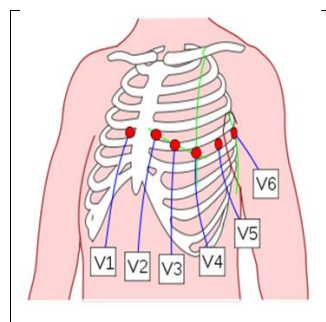


Figura 10, Derivaciones precordiales. (Campos, 2011)

De las restantes, seis son precordiales, que tiene la capacidad de medir en un plano horizontal. (Hampton, 2019) (Davis, 2007).

Las señales eléctricas recogidas por los electrodos se envían a un amplificador, el cual tiene como función, aumentar las pequeñas diferencias de potencial que produce el musculo cardiaco.

### 3.3.2 Funcionamiento de los sensores para medir la saturación de oxígeno.

Los sensores de SpO<sub>2</sub>, también conocido como pulsómetro es una tecnología capaz de medir la saturación de oxígeno en la sangre de un paciente. Estos tipos de sensores operan comparando la transmisión de la luz generada por dos tipos de led, rojo e infrarrojo, en el cuerpo hasta llegar a un fotorreceptor que provee la información sobre la proporción que existe entre la hemoglobina oxigenada y la desoxigenada en la sangre de un paciente. Existen dos tipos de saturaciones de oxígenos, la arterial (SaO<sub>2</sub>) y la periférica (SpO<sub>2</sub>). (Álvarez, 2019) (Soto et al, 2016)

- La saturación arterial de oxígeno (SaO<sub>2</sub>) corresponde al porcentaje de hemoglobina arterial funcional que es oxigenada. Esta se puede obtener a partir de una muestra de sangre arterial.
- La saturación periférica de oxígeno (SpO<sub>2</sub>) es el porcentaje de hemoglobina medido por el método de plétismógrafo óptica no invasiva.

La hemoglobina es una proteína que se encuentra en los glóbulos rojos, normalmente de color rojo y que es capaz de transportar oxígeno desde los órganos respiratorio hacia los tejidos. Existen dos tipos de hemoglobinas, funcionales y disfuncionales, la primera se encarga de enlazarse con el oxígeno, si se enlaza con cuatro moléculas de oxígeno se le considera como hemoglobina oxigenada (HbO<sub>2</sub>), si se enlaza con menos moléculas de oxígeno se le considera hemoglobina reducida (Hb) y las hemoglobinas disfuncionales son capaces de enlazarse con el dióxido de carbono, llamado Carboxihemoglobina (COHb). El porcentaje de SpO<sub>2</sub> se puede calcular con la siguiente formula. (Álvarez, 2019) (Soto et al, 2016)

$$\left[ \%SpO_2 = \frac{HbO_2}{Hb + HbO_2} \times 100 \right]$$

Figura 11, Calculo SpO<sub>2</sub>. (Álvarez, 2019) (Soto et al, 2016)

Los pulsómetros utilizan un sistema óptico para medir la SpO2 el cual está compuesta por un led de color rojo de 660 nm, un led infrarrojo de 920 nm y un fotorreceptor. Existen dos posiciones para la medición, la transmisiva y la reflectiva. (Álvarez, 2019) (Soto et al, 2016)

- Transmisiva, los leds y el fotorreceptor se posicionan en los extremos de una parte del cuerpo humano, normalmente se coloca en el dedo.
- Reflectivo, los leds y el fotorreceptor se colocan en serie, así el sensor detecta la luz reflejada por la sangre.

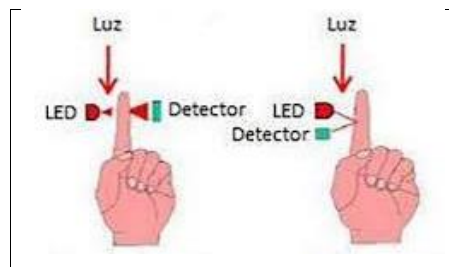


Figura 12, Posición transmisiva y reflectiva. (Álvarez, 2019)

Según este método la deoxihemoglobina tiene una absorción de luz roja próxima a los 660nm mientras que en la oxihemoglobina tiene una mayor absorción cercana a los 910 nm de luz infrarroja. En el siguiente gráfico se puede evidenciar la absorción de estas dos luces en la sangre.

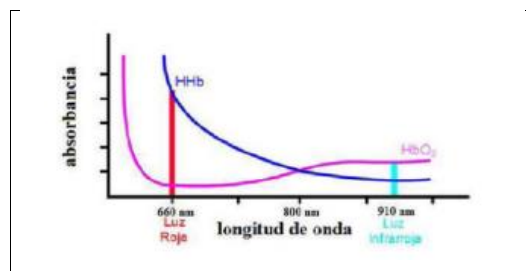


Figura 13, Absorción de luz en la sangre. (Soto et al, 2016)

Otro método es utilizar los pulsioxímetros portables, éstos tienen como ventaja ser compactos y transportables, este equipo puede medir tanto el SpO2 como las pulsaciones por minuto. (Soto et al, 2016)

Comentado [s102]: M



Figura 14, Pulso oxímetro. (Soto et al, 2016)

## Capítulo 4: Hardware del Proyecto

Para este proyecto se ha diseñado el monitoreo de signos vitales cumpliendo los siguientes requisitos: no ser invasivo, ser económico, y de uso intuitivo. Por eso se han escogido los siguientes materiales que permitirán medir cada variable fisiológica.

### 4.1 Sensores Biomédicos

#### 4.1.1 Sensor AD8232 ECG

El sensor biomédico AD8232 es un módulo integrado para realizar ECG y otras aplicaciones para la medición de biopotencial el cual está basado en el chip AD8232 de la empresa Analog Devices (Analog Devices, 2012). Este módulo está diseñado para extraer, amplificar y filtrar las pequeñas señales en condiciones de ruido, un ejemplo es el movimiento y colocación de los electrodos en cualquier parte del cuerpo. (Sparkfun, 2017)

Este sensor está compuesto por un ADC de potencia ultra baja para que cualquier microcontrolador pueda adquirir la señal de salida fácilmente, también cuenta con un filtro de paso de dos polos que sirve para eliminar el ruido producido por el movimiento, este filtro permanece acoplado por una arquitectura de instrumentación que permite tener una mejor ganancia permitiendo ahorrar espacio y costo, su rendimiento de operación es de 0 °C a 70 °C, este sensor se destaca por su bajo consumo de corriente, que es de 170 uA. (Sparkfun, 2017)

Este sensor biomédico es un módulo de bajo costo diseñado, que tiene la capacidad de medir la actividad eléctrica del corazón y registrarla como un electrocardiograma. Normalmente los ECG suelen ser extremadamente ruidosos y para poder tener una señal clara es necesario un amplificador operacional AD8232 que trabaja con intervalos PR y QT. (AD8232) (Sparkfun, 2017)

Como se puede visualizar en la Figura 15, la placa cuenta con un conector de 3.5 mm para conectar el cable con los electrodos y también cuenta con nueve pines como lo muestra en la siguiente tabla. (Sparkfun, 2017)

Comentado [s103]: M

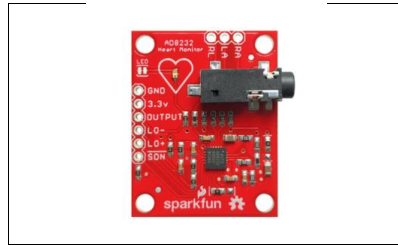


Figura 15, Modulo AD8232 de la marca Sparkfun. (Sparkfun, 2017)

Conexión del módulo AD8232:

GND	TIERRA
3.3v	FUENTE DE ALIMENTACIÓN
OUTPUT	SEÑAL DE SALIDA
LO -	DETECCIÓN DE DERIVACIONES -
LO +	DETECCIÓN DE DERIVACIONES +
SDN	DESCONECTAR LA PLACA
RA	CONEXIÓN BRAZO DERECHO
LA	CONEXIÓN BRAZO IZQUIERDO
RL	CONEXIÓN PIERNA DERECHA

Tabla 6, Conexión AD8232. (Sparkfun, 2017)

Este sensor incluye un cable sensor de tres conductores con electrodos, este cable tiene una largo de un metro aproximadamente, en un extremo tiene una conexión de salida formato Jack de 3.5mm y las otras tres son para los electrodos desechables.



Figura 16, Cable de sensor con 3 electrodos. (Sparkfun, 2017)

#### 4.1.2 Sensor MAX30102

El pulsímetro MAX30102 es un sensor fabricado por Maxim Integrated, considerado como un sensor fotoeléctrico no invasivo que permite medir el porcentaje de saturación de oxígeno y ritmo cardíaco. Este dispositivo está integrado con dos leds, un emisor de luz rojo de 660 nm y otro emisor infrarrojo de 920nm, también cuenta con un fotodetector, electrónica de bajo ruido que permite la eliminación de la luz ambiental y elementos ópticos. Este sensor es capaz de operar con fuentes de alimentación a 1.8V y 3.3V para los leds internos y tiene como módulo de comunicación que funciona con I2C. El sensor Max30102 convierte la señal análoga generada por el fotodetector a una señal digital utilizando un ADC interno de tipo sigma-delta con una resolución de hasta 18 bits, este ADC trabaja con una frecuencia de muestreo de 10.25MHz, lo que permite obtener de 50 muestras/segundos a 3200 muestras/segundos. (Maxim Integrated, 2020)

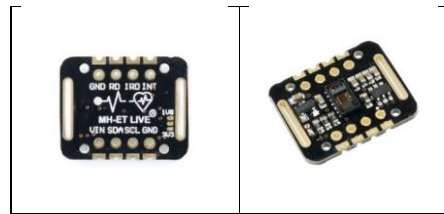


Figura 17, Sensor MAX30102. (Mechatronics, 2020)

El funcionamiento del sensor se puede visualizar en la arquitectura interna en bloque que contiene su datasheet. El sensor emite un pulso de luz roja o infrarroja, la cual será reflejada en la piel y se mezclará con la luz del ambiente, todo esto llega al fotodetector. El sensor utiliza un bloque de cancelación de luz ambiente para poder eliminarla y digitalizar la señal. Todos estos datos son almacenados en un registro FIFO y mediante el bloque I2C permitirá la transmisión de los datos obtenidos a cualquier microcontrolador. Todas estas funciones se pueden visualizar en el diagrama de bloque de la figura 18.

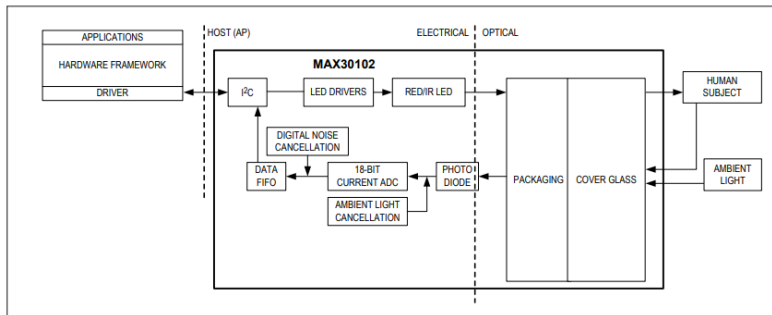


Figura 18, Diagrama de bloque Sensor Max30102. (Maxim Integrated, 2020)

Dentro del diagrama de bloque se puede observar que el sensor esta dividido en dos áreas: la óptica y la electrónica, la primera se conforma de dos tipos de leds, rojo e infrarrojo y un fotodetector, estos se encargan de generar los pulsos de luz y detectarlos utilizando el método reflexivo.

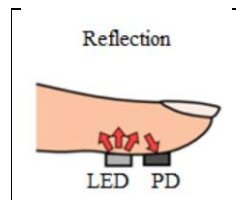


Figura 19, Método Reflectivo. (Bestbier, 2016)

Los diodos o leds que utilizan estos tipos de sensores uno rojo y otro infrarrojo que usan distintas longitudes de ondas las cuales varían entre los valores de 660 nm y 940 nm. Estos valores son recomendados en la ficha técnica del sensor. (Maxim Integrated, 2020)



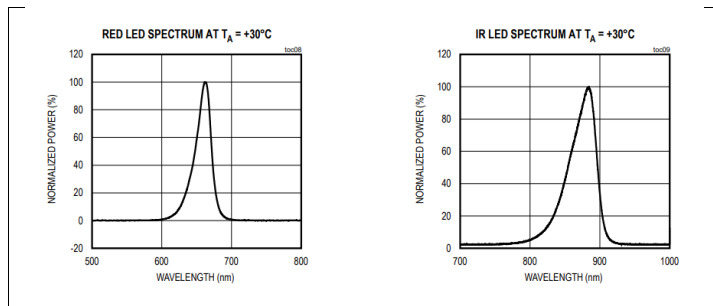


Figura 20, Longitudes de onda de cada diodo. (Maxim Integrated, 2020)

Estas longitudes de onda son reflejadas en los tejidos blandos del cuerpo, siendo detectados por un fotodetector, este se encarga de obtener una buena respuesta de los leds, en la siguiente figura se muestra la eficiencia que tiene el fotodiodo para medir los dos tipos de leds. (Maxim Integrated, 2020)

Comentado [s104]: M

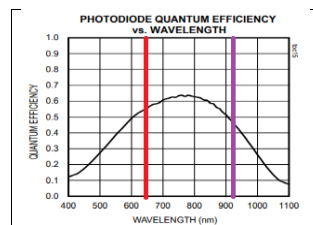


Figura 21, Respuesta del fotodiodo del Max30102. (Maxim Integrated, 2020)

## 4.2 Kit NodeMCU V2

El NodeMCU es considerado como un kit de desarrollo de código abierto integrado por el Chip ESP8232. Compatible con el lenguaje de programación Lua y compatible con el entorno Arduino IDE. Estos kits son ideales para proyectos portátiles debido a su gran potencial y comunicación Wifi. (Llamas, 2019)

El módulo ESP8232 integrado en el kit permite la comunicación con un Router inalámbricamente ya que cuenta con una conexión Wifi con el estándar de 802.11 b/g/n lo cual es compatible con el protocolo TCP/IP. (Llamas, 2019)

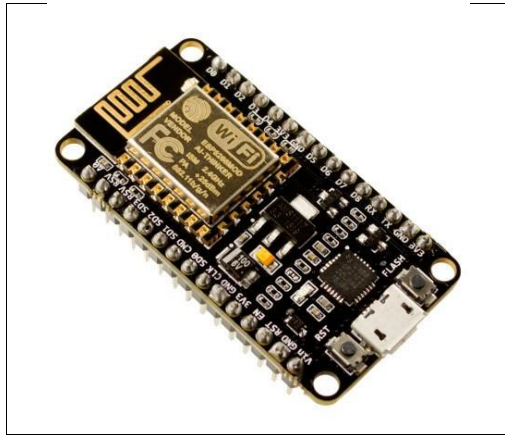


Figura 22, Modulo NodeMCU. (Llamas, 2019)

Se detalla a continuación las características técnicas de la placa:

Características	NodeMCU
Puertos	Adaptador SERIE/USB, adaptador microusb
Pines	17 digitales y 1 analógico
Tamaño	47 mm x 21 mm
Lenguaje de Programación	LUA, MicroPython, Processing en IDE de Arduino
Alimentación	5V o 3.3V
Facilidad de adaptación	Alta
Facilidad de Programación	Alta
Confiabilidad de hardware y software	Alta
Precio Promedio	\$3500 - \$5000
Frecuencia Reloj	80Mhz / 160Mhz

Tabla 7, Características Técnicas NodeMCU. (Llamas, 2019)

Otras características técnicas que contiene este módulo es una CPU Tensilica Xtensa LX3 de 32 bits, una memoria Flash Externa de 4MB y Data RAM de 96 KB, la antena está incorporada en la placa, también posee dos botones, RESET que permite reiniciar el software y Flash que permite que la placa active el modo de carga de Firmware. (Llamas, 2019)

Los pines 1 y 3 sirven para la comunicación Serial Tx/Rx que están protegidas con una resistencia de 470 ohm. En el puerto analógico utiliza un divisor de tensión que extiende el rango de 0 a 1V a un nuevo rango de 0 a 3.3V. (Llamas, 2019)

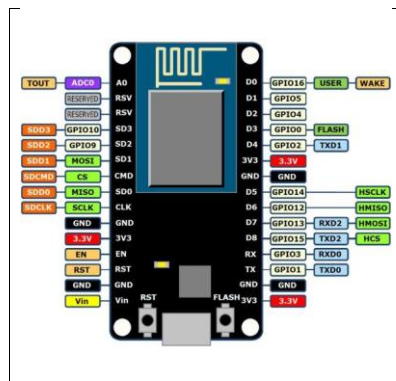


Figura 23, Pines NodeMCU. (Llamas, 2019)

### 4.3 Raspberry PI

Raspberry PI es considerado como un miniordenador de tamaño compacto, económico y de un gran rendimiento, sus características técnicas son variables dependiendo del modelo y cuentan con diversos sistemas operativos para usar. Uno de sus modelos cuenta con conexión a internet mediante WIFI y conexión Bluetooth, también cuenta con GPO's de entrada y salida que pueden ser configurables permitiendo establecer una comunicación serial con los puertos TX/RX y comunicación I2C mediante los puertos SDA/SCL.

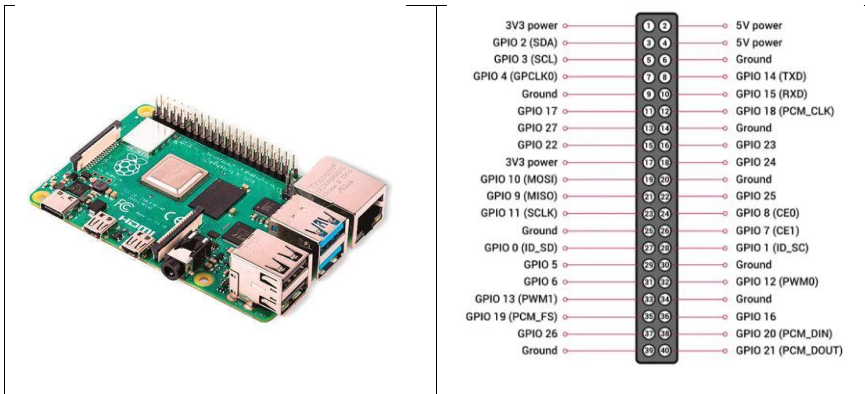


Figura 24, Raspberry PI modelo 4 y pines GPO's.

#### 4.3.1 Hardware

El hardware de una Raspberry PI es similar a un computador, pero varía según el modelo, lo que tienen en común todos los modelos es la memoria de almacenamiento, la cual consta de una ranura para insertar una memoria microSD, también cuentan con un procesador, memoria RAM DDR2, tarjeta de video y distintos módulos de comunicación como la conexión WIFI, Bluetooth y cable de red.

Para la comunicación con periféricos, ya sea sensores o actuadores, todas las Raspberry PI cuentan con entradas y salidas digitales llamadas GPO's, esto permite que la Raspberry sea compatible con diversos módulos de comunicación como el puerto serial, I2C y SPI. Por último, también cuenta con cuatros entradas USB, un HDMI que puede ser normal o mini dependiendo del modelo y entrada de alimentación.

Comentado [s105]: M

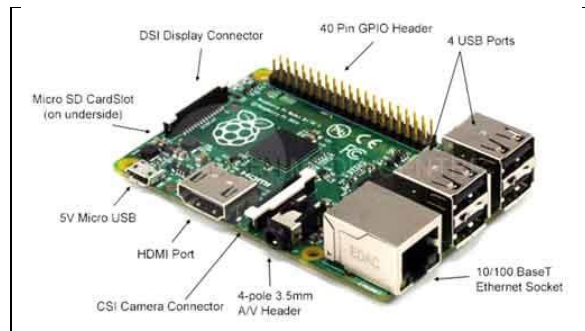


Figura 25, Hardware de Raspberry PI.

#### 4.3.2 Software

Actualmente existen diversos sistemas operativos para usar con Raspberry PI, pero normalmente funciona en base a GNU/LINUX, el más utilizado para esta placa es Raspbian que es un derivado del Debian. Este sistema operativo cuenta con varios programas preinstalados para su uso, uno de los lenguajes más utilizados y soportados para la realización de proyectos en esta tarjeta, es Python.

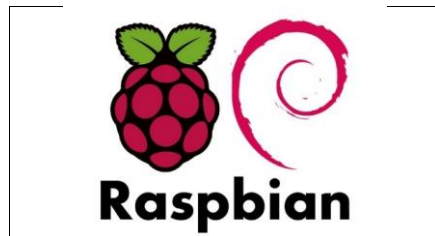


Figura 26, Sistema Operativo Raspbian.

#### 4.3.3 Raspberry Pi Zero W

La Raspberry Pi Zero W es uno de los modelos más pequeños dentro de la familia de las Raspberry PI ya que cuenta con dimensiones de 65 mm de largo y 30 de

ancho, este modelo es uno de los más útiles para usarlo en proyectos portables ya que cuenta con conexión WIFI y Bluetooth incorporados en la placa.



Figura 27, Raspberry PI Zero W.

Dentro de sus características técnicas se encuentra un procesador Broadcom BCM2835 de un GHz de un núcleo, una memoria RAM de 512 Mb, posee un puerto mini-HDMI con salida de 1080/60p, una ranura para la memoria microSD, Wifi 802.11n, Bluetooth 4.0, conector de cámara CSI, cabezal de 40 pines y posee dos entradas de micro-USB, una para la alimentación y otra para datos, su consumo es de 160mAh.

Las ventajas de utilizar esta tarjeta de desarrollo es la compatibilidad con diferentes módulos de comunicación como los puertos UART y I2C, otra ventaja es su bajo costo, posee un lenguaje de programación flexible y gran velocidad.

## Capítulo 5: Software del proyecto

En este capítulo se explicarán los softwares y herramientas que se utilizarán para este proyecto. Estos programas permitirán extraer, procesar, guardar y visualizar los datos obtenidos por los sensores.

### 5.1 Arduino IDE

Para poder usar cualquier placa de desarrollo ya sea de la familia de Arduino u otras placas compatibles con estas, es necesario descargar un IDE. Un Integrated Development Environment es un conjunto de herramientas computacionales recopiladas en un mismo software que permite en la misma aplicación desarrollar y

compilar todos los códigos necesarios para cualquier proyecto. Esta herramienta permite escribir, depurar, editar y grabar cada programa de manera sencilla. (Arduino, 2019)



Figura 28, Arduino IDE. (Arduino, 2019)

El área de trabajo esta dividido en cinco partes, barra de menú, barra de botones, el editor de código barra de consola de mensaje y la barra de estado. Otra herramienta muy importante al momento de poder visualizar los datos ya sea de un sensor o un cálculo matemático es el monitor serial. La comunicación serial entre el computador y Arduino se realiza utilizando un cable USB. (Arduino, 2019)

## 5.2 Raspbian

Raspbian es el sistema operativo más recomendado para utilizar en una Raspberry pi y está basada en GNU/Linux llamado Debian, el cual está orientado a la enseñanza de informática. La ventaja de este sistema operativo en comparación con las demás es que al momento de instalarlo en la Raspberry pi, viene con 35000 paquetes, software recompilado incluido en el instalador. (Raspbian, 2019)

Comentado [s106]: M

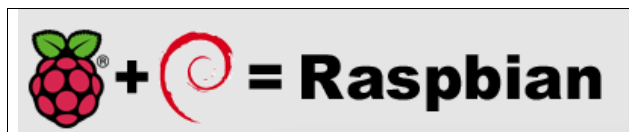


Figura 29, Raspbian de Raspberry pi. (Raspbian, 2019)

## 5.3 Mosquitto

Mosquitto es uno de los brókers MQTT más utilizados para este tipo de proyectos ya que permite comunicarse a través entre NodeMCU y una Raspberry PI.

### 5.3.1 MQTT

MQTT o en su sigla MQ telemetry Transport es un protocolo de comunicación M2M de tipo message queue el cual genera una cola de mensajes única para cada uno de los clientes que inician la subscripción. Cada mensaje enviado por el cliente viene con su tópic para que pueda ser identificado. (Llamas, 2020)

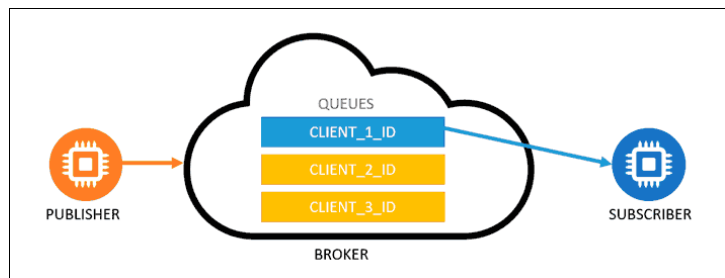


Figura 30, Message Queue. (Llamas, 2020)

Una de las ventajas que tiene este tipo de comunicación es que, si el cliente no está conectado, los datos enviados se mantendrán en el Router y serán enviadas cuando se conecten. Uno ejemplo claro de message Queue es la aplicación de mensajería WhatsApp. (Llamas, 2020)

### 5.3.2 Funcionamiento de MQTT

El principal funcionamiento de MQTT es trabajar de un servicio de mensajería push con patrón publicador/suscriptor el cual los clientes se conectan a un servidor local llamado Broker. (Llamas, 2019)

Para comenzar con la comunicación, los clientes deben iniciar con una conexión TCP/IP con el bróker, el cual tiene la capacidad de mantener un registro de los clientes conectados, por defecto se utiliza el puerto 1883. El funcionamiento de esta conexión es la siguiente: el cliente envía un mensaje CONNECT que contiene la información del usuario, contraseñas entre otras informaciones, por otro lado, el Broker responde con un mensaje CONNACK, la cual contiene el resultado de la conexión que puede ser aceptada o rechazada. (Llamas, 2019)



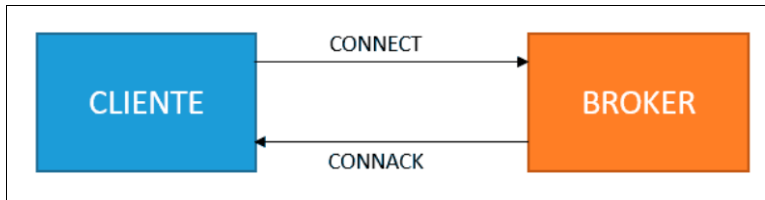


Figura 31, Conexión entre el cliente y el Broker. (Llamas, 2019)

Si el cliente desea enviar información, debe hacerlo implementando mensajes PUBLISH, donde la información enviada estará guardada en un tópic creado por el cliente. Por otro lado, para poder suscribirse se emplea un mensaje SUBSCRIBE y el servidor responderá con un SUBACK. (Llamas, 2019)

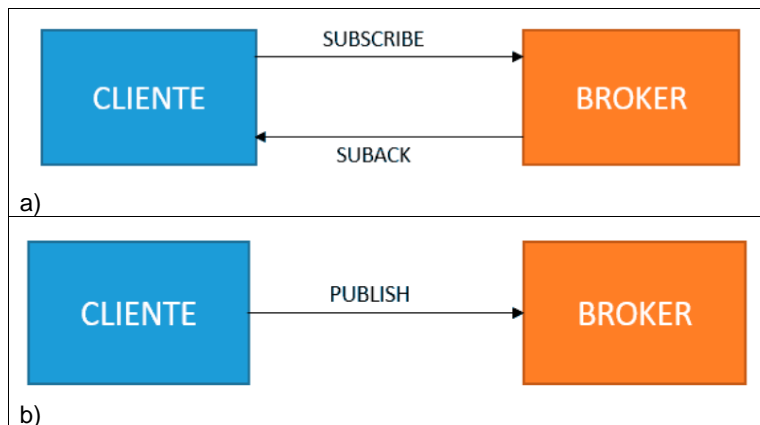


Figura 32, a) Conexión entre cliente y Broker, b) comunicación entre suscriptor y Broker. (Llamas, 2019)

### 5.3.3 Estructura de un mensaje MQTT

La estructura de la comunicación MQTT es una de las más importantes para este protocolo, en él está la tipología de los mensajes el cual consta de tres partes: la primera es la cabecera fija lo cual ocupa entre 2 a 5 bytes obligatorio, también contiene un código de control el cual identifica el mensaje enviado y la longitud del

Comentado [s107]: M

mensaje codificado entre 1 a 4 bytes de los cuales se utiliza los 7 primeros bits y el ultimo de continuidad, la segunda parte es la cabecera variable, la cual contiene información adicional que necesita algunos mensajes, esta es opcional, la última parte consta del contenido o payload, es la que contiene el mensaje real y puede tener un máximo de 256Mb. (Llamas, 2019)

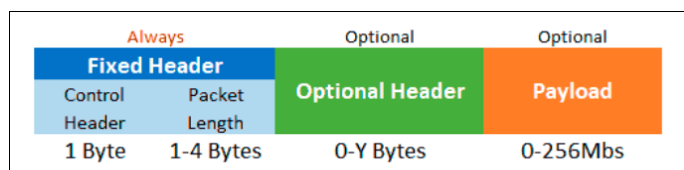


Figura 33, Estructura MQTT. (Llamas, 2019)

#### 5.4 Python

Python es un lenguaje de scripting de plataforma orientado a objetos el cual tiene la capacidad de realizar cualquier tipo de programa, desde aplicaciones o servidores hasta páginas web. La ventaja del lenguaje Python es la gran variedad de librerías que contienen pluralidad de tipos de datos, funciones y permiten realizar la gran mayoría de las tareas sin la necesidad de programar desde cero. La otra ventaja es existen muchas plataformas que permiten desarrollar los códigos Python. (Montoro, 2012)

Comentado [s108]: M

Comentado [s109]: M

Comentado [s110]: M



Figura 34, Python. (Montoro, 2012)

#### 5.5 InfluxDB

Influxdb es una base de datos especializada en almacenamiento de datos de series temporales, desarrollada por la empresa InfluxData. A diferencia de su competencia, influxdb fue creado para optimizar y minimizar el uso de recursos del sistema. (InfluxData, 2016) Las bases de datos de Influxdb puede ser muy compacta y solo puede contar con dos o tres columnas que pueden ser la fuente de los datos, el valor en sí y la marca temporal correspondiente, se puede ver en el ejemplo de la Figura 35. (InfluxData, 2016)

Sensor	Valor	Hora
Sensor 1	140,50	23/04/2020 @ 10:00
Sensor 2	110,02	23/04/2020 @ 10:00
Sensor 1	142,32	23/04/2020 @ 10:05
Sensor 2	110,50	23/04/2020 @ 10:05

Figura 35, Formato de base de datos InfluxDB. (InfluxData, 2016)

Esta base de datos tiene la capacidad de distinguir entre tags y fields, la función de los tags es contener los metadatos incluido en el índice, los fields integran los valores que serán evaluados. Por ejemplo, en la figura 35 los tags son los sensores y los fields son los valores de cada sensor. (InfluxData, 2016)

La ventaja de InfluxDB, en comparación a otras bases de datos, es que son mucho más rápidos a la hora de almacenar y procesar los datos de medición con marcas temporales. (InfluxData, 2016)



Figura 36, InfluxDB. (InfluxData, 2016)

## 5.6 Grafana

Grafana es una herramienta de código abierto que sirve para analizar y monitorear grandes volúmenes de datos en tiempo real, es capaz de admitir un amplio número de base de datos, tales como MySQL, Graphite, OpenTSDB, Prometheus, MariaDB e InfluxDB. Una de las ventajas que tiene esta herramienta grafica es crear alertas, en donde el usuario debe definir las condiciones en los paneles de *dashboard*, para que cuando el dato llegue al punto crítico, Grafana emita una notificación, la cual puede ser vía email o Slack. (Ubnunlog, 2018)

Comentado [s111]: M

Comentado [s112]: M



Figura 37, Grafana. (Ubunlog, 2018)

La otra ventaja de esta herramienta es que puede usarse como una página web la cual utiliza *dashboard* que representa distintos tipos de métricas obtenidas de los *data storages*. (adonis, 2018)



Figura 38, Dashboard de grafana. (adonis, 2018)

## Capítulo 6: Ingeniería del proyecto

Este proyecto consta de tres etapas para su funcionamiento, la primera consta de la adquisición de datos la cual tiene como función adquirir los datos entregados por las variables fisiológicas que son ECG, SpO2, la frecuencia cardiaca y la temperatura corporal. Los sensores que se utilizaran para esta etapa son AD8232 capaces de adquirir las señales eléctricas del corazón, el otro sensor es el MAX30102, la cual estará configurado para adquirir los valores de la saturación de oxígeno, temperatura corporal y frecuencia cardiaca.

La segunda etapa del proyecto consiste en el procesamiento de cada señal, para eso se utilizará el Kit NodeMCU el cual será programado como un Arduino y tendrá la capacidad de realizar cálculos para obtener los datos exactos de cada variable fisiológica, la ventaja de esta placa es su conexión inalámbrica vía Wifi que posee internamente lo cual permitirá comunicarse con una Raspberry PI Zero W inalámbricamente utilizando el protocolo de comunicación MQTT.

Comentado [s113]: M

La última etapa es el almacenamiento y visualización grafica de los datos procesados, al establecer la comunicación MQTT entre NodeMCU y Raspberry, el siguiente paso es guardar estos datos para eso se utilizará la base de datos InfluxDB, para eso se creará un script en Python que establecerá la comunicación entre MQTT e InfluxDB. Esto permitirá tener respaldos al momento sufrir alguna patología, por último, se utilizará la herramienta grafica Grafana que se conectará con InfluxDB y graficará los últimos datos guardados a tiempo real.

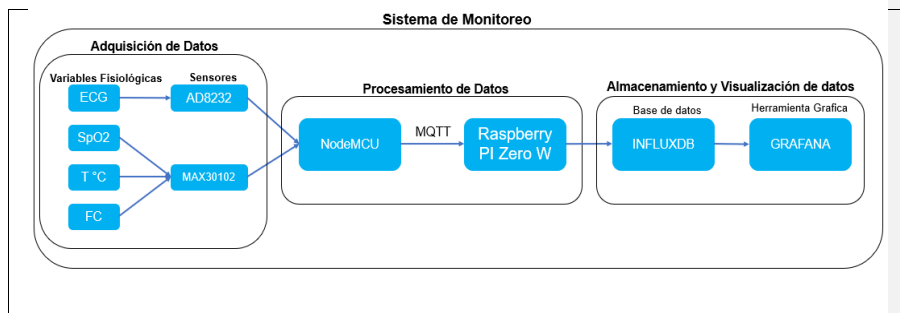


Figura 39, Diagrama del sistema de monitoreo de signos vitales.

## 6.1 Configuración de NodeMCU y los Sensores

Ya teniendo listo las configuraciones del apéndice 1, lo que sigue es programar los sensores que permitirán la adquisición de los datos de cada variable fisiológica.

### 6.1.1 Conexión a la Red Inalámbrica Wifi

Esta configuración es una de las más importantes para este proyecto ya que permitirá conectar el NodeMCU a internet de forma inalámbrica y transmitir los datos procesados por el microcontrolador a la base de datos InfluxDB instalada en Raspbian. Para ello será necesario utilizar la librería “*ESP8266WiFi.h*” y una serie de línea de código que permitirá habilitar estas funciones. (Zeroday, 2015)

```
#include <ESP8266WiFi.h>

const char* wifi_ssid = "VTR-2534266";
const char* wifi_password = "xxxxxx";

void setup_wifi()
{
  delay(10);
  WiFi.begin(wifi_ssid, wifi_password);
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
}
```

Figura 40, Configuración Wifi para NodeMCU.

### 6.1.2 Esquema del Circuito del Prototipo

En la siguiente figura se presentarán las conexiones necesarias para comunicar los sensores con la placa NodeMCU.

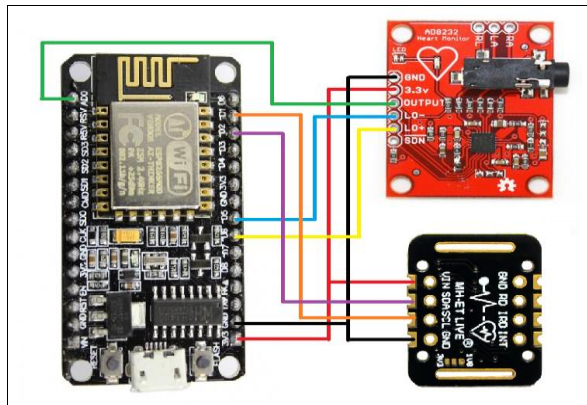


Figura 41, Circuito del Prototipo.

Los dos sensores tienen la capacidad de alimentarse en 3.3V, para la transmisión de datos del sensor AD8232, su salida va conectada a la entrada A0 del NodeMCU, en el caso del sensor Max30102 utiliza la comunicación I2C para transferir los datos obtenidos por el fotorreceptor.

		NodeMCU
AD8232	GND	GND
	3.3V	3.3V
	OUTPUT	A0
	LO-	D5
	LO+	D6
MAX30102	GND	GND
	VIN	3.3V
	SDA	D2
	SCL	D1

Tabla 8, Conexión de sensores.

Para realizar las primeras pruebas, se utilizará un Protoboard que facilitará la conexión de los sensores con la placa NodeMCU.

### 6.1.3 Configuración del sensor Biomédico AD8232

Este tipo de sensor no presenta dificultad al momento de adquirir o extraer la información a través de los electrodos conectado en el cuerpo. La programación de este sensor es simple.

```
void setup()
{
  Serial.begin(9600);
  pinMode(14, INPUT);
  pinMode(12, INPUT);
}
```

Figura 42, Void Setup sensor AD8232.

Como se puede observar en Void Setup, lo primero es establecer la velocidad del puerto serial, para este sensor se utilizará 9600 baudios y se establecerán los pines 12 y 14 como entradas en la placa NodeMCU.

```
void loop()
{
  if((digitalRead(12) == 1) || (digitalRead(14) == 1))
  {
    Serial.println('!');
  }
  else
  {
    Serial.print(analogRead(A0));
    delay(1);
  }
}
```

Figura 43, Void Loop de AD8232.

En el Void Loop se utilizará la condición IF la cual indica que si las dos entradas digitales son iguales a 1 no entregara resultado, pero si son distintas mostrara los datos obtenidos por el sensor conectado en la entrada análoga A0.

### 6.1.4 Configuración del Sensor Biomédico MAX30102

A diferencia del AD8232 en donde su programación es una de las sencilla a realizar, el sensor MAX30102 necesita implementar varias funciones para procesar los datos extraídos y convertirlos en datos reales de cada variable fisiológica.

Comentado [s114]: M



Este sensor consta de dos leds, una roja de 660nm y otro infrarrojo de 920nm, estos dos leds tienen la función de iluminar la piel y será absorbida por los tejidos y la sangre, después el fotodetector recogerá las luces reflejadas de cada longitud de onda y la retornará en flujo de datos de dos intensidades relativas que servirá para obtener la información de las variables fisiológicas.

Para este sensor se utilizará el método reflectivo, ya que tiene los leds y el sensor en paralelo. Teniendo las dos intensidades, el siguiente paso es calcular el valor Z, este valor es uno de los más importantes ya que será usado en distintas fórmulas para calcular otras variables fisiológicas.

$$Z = \text{Irojo} / \text{Infrarrojo}$$

(1)

Una de las ventajas al momento de utilizar este sensor es que tiene la capacidad de transferir los datos por comunicación I2C para eso debe utilizar la librería "Wire.h" y también se debe utilizar las librerías "MAX30105.h" que permite utilizar el sensor Max30102.

```
#include <Wire.h>
#include "MAX30105.h"

MAX30105 particleSensor;

void setup()
{
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST))
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
  Serial.println("Place your index finger on the sensor with steady pressure.");

  particleSensor.setup();
  particleSensor.setPulseAmplitudeRed(0x0A);
  particleSensor.setPulseAmplitudeGreen(0);
}
```

figura 44, Comunicación I2C.

En las últimas líneas de código se utilizará el ParticleSensor.Setup el cual permite al sensor estar configurado de fábrica. La siguiente línea indica que el led rojo del sensor indicará que el sensor está funcionando. La última línea indica que el led

Comentado [s115]: M

Comentado [s116]: M

verde se encuentra apagado, esto se debe a que el MAX30102 no cuenta con led verde a diferencia del MAX30105.

### **Cálculo de saturación de oxígeno**

Utilizando las especificaciones técnicas del sensor y los datos entregados por el datasheet, se recomienda utilizar la ecuación empírica para obtener el valor de SpO2 deseado, para eso se debe reemplazar el valor obtenido en la ecuación (1) en la ecuación (2).

$$\text{SpO}_2 (\%) = (30.354 - 45.06 \times Z) \times Z + 94.845$$

(2)

para la programación en NodeMCU, utilizaremos los valores que aparecen en la fórmula (2) como un Cont float. En la librería del sensor incluyen una función que permite extraer los datos de los leds. Se utilizará la ecuación (1) para tener el valor de Z y se reemplaza en la ecuación (2) para obtener SpO2 en porcentaje.

```
const float K1 = -45.06;
const float K2 = 30.354;
const float K3 = 94.845;

ir = particleSensor.getIR();
red = particleSensor.getRed();

float Z = (float)red / (float)ir;

if (checkForBeat(irValue) == true)
{
  oxygen = 0;
  if(beatAvg != 0)
  {
    oxygen = (int)(K1 * Z + K2) * Z + K3;
  }
}
```

Figura 45, Programación de SpO2.

### **Cálculo para la temperatura corporal**

Los datos de esta variable es fácil obtener debido a que la librería del sensor cuenta con la función para calcular la temperatura corporal en centígrados Celsius.

```
body_temp = particleSensor.readTemperature();
```

### **Cálculo de la Frecuencia Cardíaca**

Para el cálculo de la frecuencia cardiaca que viene definida por la inversa del periodo de tiempo entre dos latidos y se debe multiplicar por 60 ya que la variable que se quiere calcular son las pulsaciones por minuto para eso se utilizara la librería "HeartRate.h" que permite realizar los cálculos necesarios para dicha variable.

```
#include "heartRate.h"
```

Figura 46, Librería HeartRate para el MAX30102.

```
unsigned long currentTime = millis();
unsigned long delta = millis() - lastBeat;
lastBeat = millis();
bfM = 60 / (delta / 1000.0);

if (bfM < 255 && bfM > 20)
{
    rates[rateSpot++] = (byte)bfM;
    rateSpot %= RATE_SIZE;
    beatAvg = 0;
    for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
    beatAvg /= RATE_SIZE;
}
```

Figura 47, Calculo pulso por minuto

Dentro de esta programación también se calculará el promedio de las pulsaciones por minuto.

#### **6.1.5 Conexión al Broker utilizando MQTT**

Este tipo de comunicación es una de las más importantes para el proyecto porque permite transferir los datos procesados por NodeMCU inalámbricamente a la Raspberry PI Zero W el cual trabajara como bróker. La función del NodeMCU es de cliente y tiene la capacidad de enviar y pedir datos, por otro lado, la Raspberry solo trabajará como suscriptor el cual solo recibirá los datos obtenidos por los sensores.

Teniendo definida la función del NodeMCU, para comenzar con su programación es necesario instalar la librería “*PubSubClient.h*” y obtener el IP del servidor y crear los tópicos de cada variable fisiológica a utilizar. También es necesario crear un usuario, clave y un cliente ID dentro de la plataforma del bróker Mosquitto. Una vez listo todo, el siguiente paso es asignar el puerto 1883 que se utilizara para poder comunicarse entre los dispositivos.

Comentado [s117]: M

```
#include <PubSubClient.h>

const char* mqtt_server = "192.168.0.10";
const char* spo2_topic = "nodemcu/sensor/SPO2";
const char* bfm_topic = "nodemcu/sensor/bfm";
const char* temp_topic = "nodemcu/sensor/temp";
const char* irValue_topic = "nodemcu/sensor/irValue";
const char* ECG_topic = "nodemcu/sensor/ECG";
const char* mqtt_username = "tesis";
const char* mqtt_password = "6sbbu7";
const char* clientID = "nodemcu";

WiFiClient wifiClient;
PubSubClient client(mqtt_server, 1883, wifiClient);
```

Figura 48, Comunicación MQTT.

Ya creado los tópicos de cada variable fisiológica, se iniciará la conexión de MQTT con el siguiente Void.

```
void connect_MQTT()
{
  if (client.connect(clientID, mqtt_username, mqtt_password))
  {
    Serial.println("Connected to MQTT Broker!");
  }
  else {
    Serial.println("Connection to MQTT Broker failed...");
    Serial.println("NodeMCU is restarting now!");
    ESP.restart();
  }
}
```

Figura 49, Void Connect\_MQTT.

Ya iniciada la comunicación MQTT entre los dos dispositivos, el siguiente paso es asignar los valores procesados de las variables fisiológicas a los tópicos creados inicialmente como se muestra en la siguiente figura.

```
if (client.publish(spo2_topic, String(oxygen).c_str()))
{
  Serial.println("SPO2 sent!");
}
```

Figura 50, Asignación de los Tópicos.

## 6.2 Configuración de Raspberry PI Zero W

Como se menciona al principio de este capítulo la función principal que tiene esta placa de desarrollo es recibir los datos procesados por el NodeMCU y guardar dichos datos en la base de datos InfluxDB para luego ser visualizados utilizando la herramienta Grafana a tiempo real.

Teniendo claro lo anterior, lo primero que se debe realizar es la instalación del sistema operativo Raspbian a la placa, todos estos pasos se pueden visualizar en el apéndice 2 del proyecto. Una vez obtenido el sistema operativo listo y configurado, el siguiente paso es modificar la IP de la Raspberry a una IP estática, esto permite mantener la IP que uno desee al momento de reiniciar la placa de desarrollo, ver apéndice 2 del proyecto.

Comentado [s118]: M

Lo que queda por realizar son las instalaciones de los tres programas necesarios para este proyecto, los cuales son Mosquitto para la comunicación MQTT, la base de datos InfluxDB para guardar los datos procesados y Grafana el cual se encargara de graficar los datos guardados en la base de datos InfluxDB y poder realizar la monitorización de los signos vitales. Todas las configuraciones están en los apéndices del 2 al 4.

Comentado [s119]: M

Una vez configurado Mosquito para la transferencia de los datos procesados por NodeMCU, se realizará un script en Python para poder conectar y guardar los datos que se encuentran en los tópicos, pero para que este código funcione es necesaria la instalación de dos paquetes que son "paho-mqtt" que es una librería que permite comunicarse con un servidor vía MQTT y el segundo paquete es "InfluxDB" para la conexión con la base de datos. Para el desarrollo del script en Python crear un usuario y clave para InfluxDB.

```
pi@raspberrypi:~ $ influx
Connected to http://localhost:8086 version 1.8.3
InfluxDB shell version: 1.8.3
> CREATE USER tesis WITH PASSWORD '6sbbu7' WITH ALL PRIVILEGES
> CREATE DATABASE monitor
>
```

Figura 51, Creación de usuario, clave y base de datos.

Con la base de datos creada dentro del script se escribirán las siguientes variables, que permite conectarse a MQTT y a la base de datos InfluxDB.

```
INFLUXDB_ADDRESS = '192.168.0.10'
INFLUXDB_USER = 'tesis'
INFLUXDB_PASSWORD = '6sbbu7'
INFLUXDB_DATABASE = 'monitor'

MQTT_ADDRESS = '192.168.0.10'
MQTT_USER = 'tesis'
MQTT_PASSWORD = '6sbbu7'
MQTT_TOPIC = 'nodemcu/+/+'
MQTT_REGEX = 'nodemcu/([^/]+)/([^/]+)'
MQTT_CLIENT_ID = 'mqtt-influxdb'
```

Figura 52, Datos de MQTT e InfluxDB.

Lo que prosigue es crear una clase llamada “*SensorData*” que permite obtener de forma ordenada los datos de la ubicación, la medida o en este caso la variable fisiológica y el valor de dicha variable.

- La ubicación es dónde estará puesto el sensor, en este proyecto es NodeMCU.
- La medición corresponde a la variable a medir por ejemplo la temperatura, SpO2, frecuencia cardiaca y la ECG.
- El valor será el valor numérico real de la medición como por ejemplo el 99% de la saturación de oxígeno.

```
class SensorData(NamedTuple):
    location: str
    measurement: str
    value: float
```

Figura 53, Datos del sensor.

Comentado [s120]: M

Dentro del mismo script existen cinco funciones que permitirán recibir los datos y guárdalos en la base de datos.

1. On connect: Esta función permite conectar todos los clientes con el Broker, en ella existe la variable "rc" que contiene un código de error cuando la conexión no es exitosa.
2. Parse mqtt message: Se encarga de leer la ubicación, medición y los datos numéricos de cada sensor utilizando los tópicos asignados y los devuelve como NamedTuple.
3. Send sensor data to influxdb: Obtiene la variable NamedTuple como una entrada y crea una estructura json para que luego la herramienta Grafana pueda realizar las consultas. La ubicación la guarda en una estructura tag y el valor del sensor en una variable de campo.
4. On message: Esta función será llamada cada vez que se publique un dato, también se utilizaran la función 2 para obtener los datos y la función 3 para almacenar dichos datos.
5. Init influxdb database: permite iniciar la base de datos creada anteriormente en InfluxDB, si no existe una base de datos, esta función tiene la capacidad de crear una.

Tendiendo todas las funciones programadas y listas, se debe guardar el script con el nombre mqtt-influxdb.py.

### **6.2.1 Comprobación conexión MQTT y InfluxDB**

Esta parte del proyecto es una de las más importantes debido a que cada variable fisiológica será guardada en la base de datos, para ello se debe ejecutar el script creado anteriormente y observar si recibe los datos. Al ingresar a influxdb, se debe seleccionar la base de datos llamada monitor y ver cuantas variables existen en ella.

```
pi@raspberrypi:~$ influx
Connected to http://localhost:8086 version 1.8.3
InfluxDB shell version: 1.8.3
> use monitor
Using database monitor
> show measurements
name: measurements
name
----
ECG
SP02
bpm
irValue
temp
└─┘
```

Figura 54, Realizar consultas en InfluxDB.

### 6.2.2 configuración de Grafana

El último paso para realizar dentro de las configuraciones de la Raspberry PI Zero W, es la instalación de la interfaz gráfica llamada Grafana, todos los pasos para la instalación se encuentran en el apéndice 5. Para poder ingresar a la GUI de Grafana a través de un navegador, debes utilizar la IP del servidor más el puerto 3000.

- 192.168.0.10:3000



Figura 55, GUI de grafana.

Para iniciar sesión, grafana viene con un usuario y clave llamada "admin", al iniciar la sesión pedirá que se cambie la clave para tener mayor seguridad. Ya ingresado, lo primero que pedirá será una base de datos para conectarse, para este proyecto se está utilizando InfluxDB y dentro de la sección HTTP y en InfluxDB Details hay que completar los siguientes datos:

- URL: 192.168.0.10:8086
- Base de datos: "monitor"

Comentado [s121]: M

Comentado [s122]: M



- Usuario: “tesis”
- Contraseña: “6sbbu7”

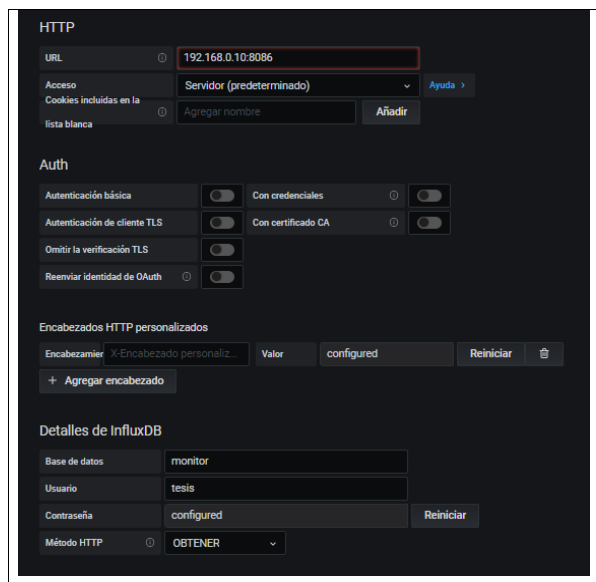


Figura 56, Fuente de datos.

Al guardar los datos, en el costado izquierdo de la página aparece el icono “CREAR”, se creará un nuevo tablero y dentro del mismo, se creará un nuevo panel de control.

Comentado [s123]: M

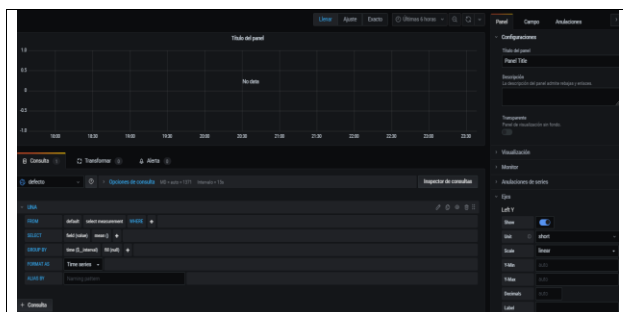


Figura 57, Panel de control.

Dentro del panel de control, encontramos en la parte inferior la tabla de consulta, en ella se puede elegir la base de datos y realizar las consultas de las variables fisiológicas. Las ventajas de este panel es poder realizar una consulta query sin la necesidad de programar, también permite realizar promedios de los datos entre otras funciones. En la parte superior se puede visualizar el comportamiento de los datos en tiempo real, una vez realizadas las consultas se debe apretar “aplicar” para que todas las consultas se ejecuten instantáneamente.

En el tablero de la derecha de la figura anterior, se encuentra una tabla llamada Panel, dentro del existe una sección llamada Visualización, en la cual se puede escoger qué tipo de grafico se desea visualizar. Como en este proyecto existen cuatro variables fisiológicas a medir, la visualización es la siguiente:



Figura 58, Interfaz gráfica del monitor de signos vitales.

## Capítulo 7: Discusión de Resultados y Conclusiones Generales

El estudio de los resultados obtenidos en este proyecto se puede dividir en tres partes: la primera es el estudio realizado para ver en qué partes del cuerpo se puede realizar las mediciones de las variables fisiológicas y las tecnologías que permitirán realizar este desarrollo, la segunda parte es la adquisición y procesamiento de los datos de cada variable fisiológica y por último es el monitoreo de los datos procesados durante un tiempo.

A continuación, se demostrarán los resultados obtenidos durante el desarrollo del proyecto.

Comentado [s124]: ¿?

Comentado [s125]: M

Comentado [s126]: M

Comentado [s127]: M

## 7.1 Discusión de Resultados

### 7.1.1 Resultado sensor AD8232 ECG

Como se puede observar en la Figura 59, es posible distinguir que existen dos formas de realizar un electrocardiograma con este sensor, ya que cuenta con solo tres electrodos. En la forma a) las derivaciones van en las muñecas y en la pierna derecha. La forma b) los electrodos van conectados en el pecho.

Comentado [s128]: M

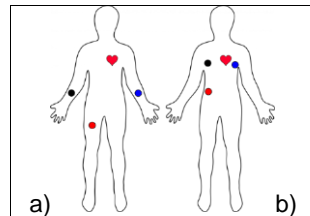


Figura 59, Triangulo de Einthoven para un Electrocardiograma.

Cada electrodo cumple la función de representar una señal y dicha señal va normalizada con su respectivo color para poder identificar la posición del electrodo, esto se puede observar en la Tabla 9.

Señal	Color del cable
R	Negro o Rojo
L	Azul o Amarillo
F	Rojo o Verde

Tabla 9, Posición de los Electrodo.

En la forma a), luego de compilar el programa en NodeMCU, al abrir el monitor serial plotter se puede observar en la Figura 60, la gráfica que el sensor entrega.

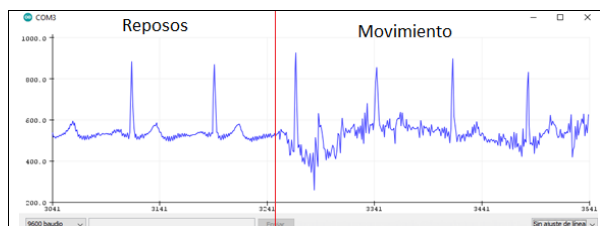


Figura 60, Primera Prueba de medición.

En este gráfico se realizaron dos pruebas, una en reposo y otra en movimiento, como caminar o correr. En la primera se puede observar un electrocardiograma limpio, en el cual se evidencian las ondas P y T dentro de la señal. En cambio, al realizar un movimiento de las extremidades, la señal presenta distorsiones, lo que causa que el electrocardiograma no sea tan limpio.

Comentado [s129]: M

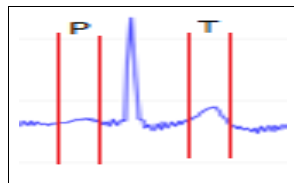


Figura 61, Ondas P y T en la forma a).

En la forma b), se puede observar en la Figura 62, donde los electrodos están conectados al pecho del paciente, la onda QRS aparece en el gráfico, lo cual es una gran diferencia con la forma a), también se realizó la prueba de movimiento la cual no presenta mucha diferencia con la señal en reposo.

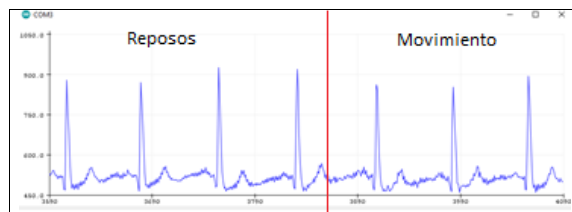


Figura 62, Segunda prueba de medición .

Obtenidos los gráficos de las señales en dos posiciones distintas, se decide implementar la forma b) para este proyecto, ya que no sufre mucha alteración en la señal al momento de realizar movimientos con las extremidades.

Comentado [s130]: M

### 7.1.2 Resultado sensor MAX30102

Para este sensor existen muchas formas de poder realizar las mediciones de SpO2, temperatura corporal y la frecuencia cardiaca, comúnmente se utilizan los dedos de la mano para realizar estas medidas, pero también puede medirse en la muñeca, en la oreja y hasta en la planta del pie, este último normalmente se realiza a recién nacidos. Para este proyecto solo se realizarán pruebas en el dedo, donde se compararán las mediciones con un pulsómetro que es un instrumento normalmente utilizado por los funcionarios de la salud. En la siguiente tabla se mostrarán los resultados obtenidos después de realizar las pruebas.

Comentado [s131]: M

	MAX30102	pulsómetro
SpO2	100%	97%
BPM	55 - 80	75 - 80

Tabla 10, Comparación MAX30102 y un Pulsómetro genérico

Tendiendo los resultados de cada sensor, se llega a la conclusión de que no hay mucha diferencia entre los dos sensores, ya que, aunque en la saturación de oxígeno el MAX30102 ganan por 3%, los dos se encuentran dentro de los rangos estables para esta variable fisiológica. Algo similar ocurre en la pulsación por minuto, donde los dos sensores oscilan entre los 50 a 80 el promedio.

Comentado [s132]: M

### 7.1.3 Resultado conexión entre NodeMCU, Mosquitto y la base de dato

#### InfluxDB.

Como se explicó en el capítulo anterior donde se puede ver la programación de los sensores y las configuraciones para que la placa NodeMCU se conecte a internet y envíe los datos procesados por MQTT a Raspberry PI. El microcontrolador tomara los datos cada un segundo.

Comentado [s133]: M



Figura 64, Conexión exitosa entre NodeMCU y Raspbian utilizando Mosquitto.

Después de realizar varias pruebas, se logró obtener la visualización de todos los datos fisiológicos medidos por el sensor en la Raspberry PI.

El siguiente paso es poder guardar todos los datos en InfluxDB, para eso se creó un script en Python que permite la comunicación entre mosquitto y la base de datos. Todas las configuraciones se encuentran en el capítulo anterior, ahora se debe crear un usuario llamado “**tesis**” con todos los privilegios, clave “**6sbbu7**” y una base de datos llamada “**monitor**”.

Ya teniendo el script configurado se debe correr iniciándolo en la consola con el siguiente comando: “**python3 mqtt-influxdb.py**”

Al correr el archivo mostrará todos los tópicos guardados en la base de datos y sus valores correspondientes.

```
pi@raspberrypi:~$ python3 mqtt-influxdb.py
MQTT to InfluxDB bridge
Connected with result code 0
nodemcu/sensor/bfm b'44'
nodemcu/sensor/temp b'33'
nodemcu/sensor/irValue b'108298'
nodemcu/sensor/ECG b'505'
nodemcu/sensor/SP02 b'99.78'
nodemcu/sensor/bfm b'44'
nodemcu/sensor/temp b'33'
nodemcu/sensor/irValue b'108294'
nodemcu/sensor/ECG b'501'
nodemcu/sensor/SP02 b'99.78'
nodemcu/sensor/bfm b'44'
nodemcu/sensor/temp b'33'
nodemcu/sensor/irValue b'108303'
nodemcu/sensor/ECG b'505'
nodemcu/sensor/SP02 b'99.78'
nodemcu/sensor/bfm b'44'
nodemcu/sensor/temp b'33'
```

Figura 65, Comunicación exitosa entre MQTT e InfluxDB

Con la comunicación lista y funcionando, se debe verificar si los datos están siendo registrados en la base de datos y si se están guardando en sus variables correspondiente, para eso utilizaremos el siguiente comando “**select \* from SPO2**”.

```

1605220007995780319 sensor 0
1605220008049139998 sensor 0
1605220008099424627 sensor 99.63
1605220008151500109 sensor 99.63
1605220008204289382 sensor 99.63
1605220008254032770 sensor 99.63
1605220008306688243 sensor 99.63
1605220008355594980 sensor 99.63
1605220008406372934 sensor 99.63
1605220008464734607 sensor 99.63

```

Figura 66, Datos guardados en InfluxDB.

Como se puede observar en la Figura 66, InfluxDB separa los datos en tres partes como se mencionó en el capítulo anterior, donde el primer dato es el “*time*”, el segundo dato es la “*localización*” que es el segundo datos de los tópicos creado y el ultimo valor es el dato de la variable.

Comentado [s134]: M

#### 7.1.4 Resultado de los datos procesados graficados utilizando Grafana

Con Grafana se creó un panel de control con gráficos y datos de cada una de las variables fisiológicas recogida por los sensores y se controlará en tiempo real la información recibida. La ventaja de esta herramienta es que tiene la capacidad de adaptar los gráficos en una gran variedad de periodo de tiempo como por ejemplo ver las lecturas de hace 15 minutos hasta un año.



Figura 67, Panel de control.

## 7.2 Conclusiones Generales

El objetivo general de este proyecto es realizar un sistema de monitoreo de signos vitales capaz de efectuar mediciones de la saturación periférica de oxígeno, temperatura corporal, frecuencia cardiaca y ejecutar electrocardiogramas, el cual ha cumplido con los objetivos propuestos al comienzo de este proyecto. Si bien una de

Comentado [s135]: M



las primeras dificultades fue decidir en qué parte del cuerpo se pueden medir estas variables fisiológicas, los sensores MAX30102 y AD8232 que son sensores de bajo costo, sirvieron para poder realizar la adquisición de datos.

Al utilizar el MAX30102, permitió obtener no solo la frecuencia cardiaca y la saturación de oxígeno, sino que también permitió poder obtener la temperatura corporal externa en grados Celsius. Pero uno de los inconvenientes que se obtuvo al momento de procesar los datos fue aplicar las fórmulas correspondientes para cada variable, tanto para la SpO2 como para la frecuencia cardiaca, en donde se tuvo que investigar en la ficha técnica y por la información entregada por los desarrolladores se pudo obtener la fórmula empírica para hacer el cálculo. Otra limitante que entrega el MAX30102 es muy sensible a pequeños movimientos, por lo que se creó un base donde será instalado y permitirá que el dedo se mantenga en una posición fija.

Comentado [s136]: M

Este prototipo demuestra fiabilidad en la transmisión inalámbrica por medio del protocolo MQTT y permite una monitorización continua sin pérdida de información.

### 7.2.1 Trabajo Futuro

Como consecuencia directa del cumplimiento de los objetivos propuestos en este proyecto, este trabajo se puede extender para la inclusión de mejoras en distintas líneas, tales como:

- **Ser portable:** El sistema de monitoreo sería más eficiente si pudiera trabajar de forma automática y autónoma, para eso se podría añadir una batería pequeña y evitar usar cables para su funcionamiento y por último agregarle unas correas para usarlo como reloj de muñeca, así permite que el paciente pueda tener el sistema puesto y no dificultar el movimiento de las extremidades.
- **Alertas generales que informen las anomalías en las variables fisiológicas:** Como este proyecto está desarrollado para ser usado por el paciente y poder saber el estado de salud de este, se puede implementar a futuro un sistema de alarmas por medio de correo electrónico o notificación por teléfono al personal de salud que estará monitoreando al paciente.
- **Iniciar script de Python cuando se enciende el proyecto:** Esta mejora es una de las más necesarias para este proyecto ya que permitirá iniciar el

sistema cuando la Raspberry inicie el sistema operativo y facilitará que el personal no capacitado, no ingrese al sistema interno del proyecto.

- **Configuración no-ip para el monitoreo de datos:** Es una mejora que permitirá que una persona no capacitada pueda ingresar a grafana para poder monitorear los signos vitales, sin la necesidad de ingresar la ip del servidor, sino que podrá ingresar con un DNS dinámico.
- **Uso de sensores calificados para el monitoreo de signos vitales:** Si bien los sensores lograron realizar la adquisición de los datos de cada variable fisiológica, a futuro se podría implementar sensores calificados para estas mediciones y poder usarlos sin problemas en hospitales.

## GLOSARIO

- °C: GRADOS CELCIUS.
- ADC: CONVERTIDOR ANALOGO DIGITAL.
- CO2: DIOXIDO DE CARBONO.
- COHb: CARBOXIHEMOGLOBINA.
- CPU: UNIDAD CENTRAL DE PROCESAMIENTO.
- DDR2: DOUBLE DATA RATE.
- ECG: ELECTROCARDIOGRAMA.
- EEG: ELECTROENCEFALOGRAMA.
- GUI: GRAPHICAL USER INTERFACE.
- HB: HEMOGLOBINA.
- HbO2: OXIHEMOGLOBINA.
- HTTP: HYPERTEXT TRANSFER PROTOCOL.
- I2C: CIRCUITO INTERINTEGRADO.
- IDE: INTEGRATED DEVELOPMENT ENVIRONMENT.
- KB: KILOBYTE.
- LUA: LENGUAJE DE PROGRAMACIÓN DE ALTO NIVEL.

- M2M: MACHINE TO MACHINE.
- mA: MILLIAMPERIOS HORAS.
- MASK: MASCARA.
- MB: MEGABITS.
- MHz: MEGAHERTZ.
- Mm: MILIMETRO.
- MMHG: MILÍMETRO DE MERCURIO.
- MQTT: MESSAGE QUEUE TELEMETRY TRANSPORT.
- mV: MILIVOLTIOS.
- nm: NANOMETRO.
- OS: SISTEMA OPERATIVO.
- PIC: CONTROLADO DE INTERFAZ PERIFERICO.
- ROUTE: ENRUTADOR.
- RX: ESCUCHAR.
- SaO<sub>2</sub>: SATURACIÓN ARTERIAL DE OXIGENO.
- SD: SECURE DIGITAL.
- SPI: SERIAL PERIPHERAL INTERFACE.
- SpO<sub>2</sub>: SATURACIÓN PERIFERICA DE OXIGENO.
- SvO<sub>2</sub>: SATURACIÓN VENOSA DE OXIGENO.
- TCP: PROTOCOLO DE CONTROL DE TRANSMISIÓN.
- TX: TRANSMITIR.
- $\mu$ A: MICROAMPERE.
- UART: UNIVERSAL ASYNCHRONOUS RECEIVER-TRANSMITTER.
- USB: UNIVERSAL SERIAL BUS.
- V: VOLTAJE.

## Apéndice 1: Configuración de Arduino IDE para usar NodeMCU

Para poder utilizar el Kit NodeMCU en Arduino IDE, lo primero que se debe hacer es instalar el plugin del ESP8232 para Arduino, para eso se debe abrir Arduino IDE, en la pestaña de archivos encontraras el ítem Preferencias.

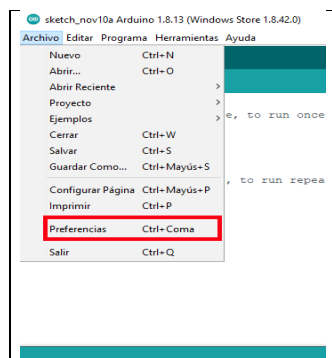


Figura 68, Preferencias de Arduino IDE.

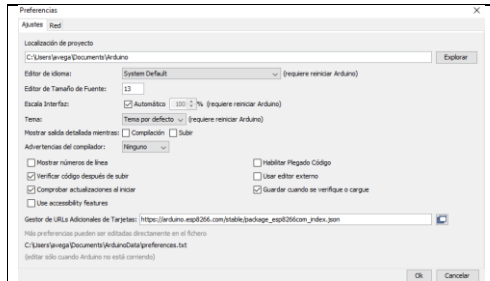


Figura 69, Gestor de URLs Adicionales de Tarjetas.

Al ingresar a Preferencias, se abrirá otra ventana que dice ajustes, en ella se encontrará el gestor de URLs adicionales de la tarjeta y se debe **digitar** el siguiente enlace.

Comentado [s137]: M

[https://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](https://arduino.esp8266.com/stable/package_esp8266com_index.json)

El siguiente paso es ir a herramientas, bajar donde dice placa y abrir el Gestor de tarjetas.

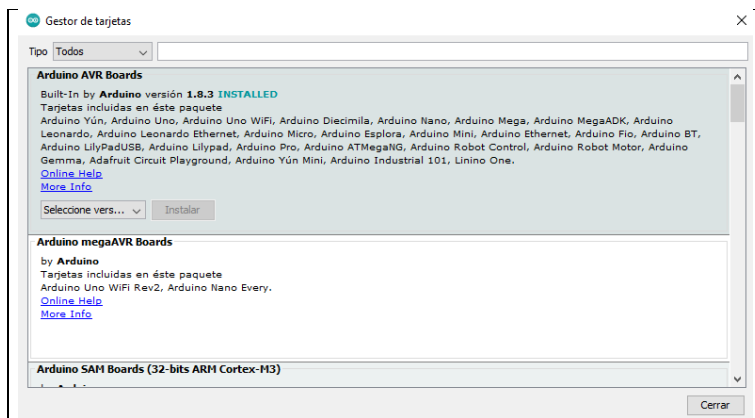


Figura 70, Gestor de Tarjeta.

Dentro de esta pestaña se debe buscar en la lista el “*esp8266 by ESP8266 Community*”, se selecciona y se instala. Al término de la instalación, se debe verificar si la instalación fue un éxito, para eso se debe ingresar nuevamente a herramientas, placas y aparecerá la nueva lista de ESP8266 y seleccionar “*Generic ESP8266 Module*”.

### **Apéndice 2: Instalación de Raspbian y Configuración IP Estática**

Para este proyecto se debe configurar la Raspberry para su uso. Para eso se debe instalar el sistema operativo Raspbian en la Raspberry PI.

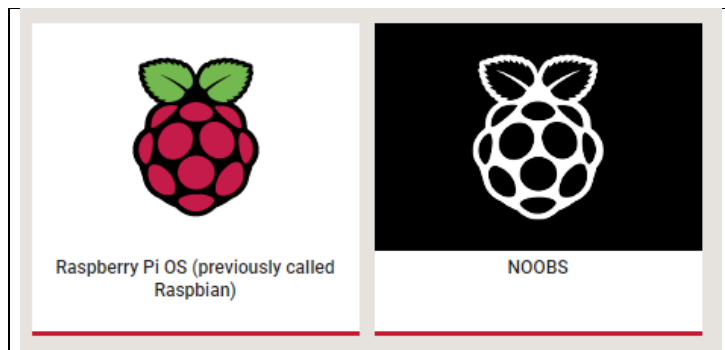


Figura 71, Sistema operativo Raspbian e instalador Noobs.

Existen dos formas para poder instalar el sistema operativo, el primero es cargar la imagen directo a la memoria microSD y el otro es descargar el instalador Noobs, este programa permite instalar diversos sistemas operativos que se encuentran para Raspberry PI.

Noobs se descargará en formato .RAR, el cual se debe descomprimir en una carpeta para luego pegar todos los archivos dentro de la memoria microSD. Teniendo todo listo se debe insertar la memoria en la placa e instalar el sistema operativo. Al finalizar la instalación aparecerá el escritorio de Raspbian.



Figura 72, Escritorio Raspbian.

El siguiente paso a realizar es la configuración de la IP estática de la Raspberry PI, pero primero se debe conectar a internet a través de Wifi después abrir la consola y obtener los siguientes datos:

- Una dirección IP
- Una máscara de red
- Un Gateway

Para las dos primeras se debe escribir el siguiente comando “*IFCONFIG*” en la consola.

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.66 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::a74b:723a:6e51:5114 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:9a:56:56 txqueuelen 1000 (Ethernet)
    RX packets 22172 bytes 24724050 (23.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2820 bytes 674377 (658.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 73, IP y MASK de Raspberry PI.

Como la placa está conectada a Wifi, la IP y la Mascará se obtendrá dentro de *WLAM0*.

Teniendo esos dos datos, el siguiente se puede obtener escribiendo el comando “*ROUTE -NE*” en la consola. Esta IP le pertenece al Router que está instalado en el edificio.

```
pi@raspberrypi:~ $ route -ne
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 192.168.10.1 0.0.0.0 UG 0 0 0 wlan0
192.168.10.0 0.0.0.0 255.255.255.0 U 0 0 0 wlan0
pi@raspberrypi:~ $
```

Figura 74, IP del Router.

Tras la recopilación de estos datos, lo siguiente es configurar la IP estática de la Raspberry Pi. Para eso se escribirá en la consola el siguiente comando “*sudo nano /etc/network/interfaces*” y se escriben las siguientes configuraciones.

```
# interfaces(5) file used by ifup(8) and ifdown(8)
# Please note that this file is written to be used with dhcpd
# For static IP, consult /etc/dhcpd.conf and 'man dhcpd.conf'
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0
auto wlan0

iface wlan0 inet static
address 192.168.10.66
netmask 255.255.255.0
gateway 192.168.10.1
wpa-ssid "RBeltranC_5G"
wpa-psk "ms6nrbt8wcg"
```

Figura 75, Configuración IP estática.

Teniendo todo listo se debe reiniciar la Raspberry Pi y al iniciar el sistema operativo se debe revisar la IP estática nuevamente para verificar si esta estática.

### Apéndice 3: Instalación del Broker Mosquitto

Para establecer una comunicación entre los módulos NodeMCU a una Raspberry es necesario instalar el servidor Eclipse Mosquitto.





Figura 76, Eclipse Mosquitto.

El primer paso es descargar la clave de firma utilizando el comando `wget`, lo cual permitirá descargar el fichero en el directorio en que se encuentra la Raspberry PI.

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

```
pi@raspberrypi:~$ sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg
.key
--2020-11-11 23:07:09-- http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
Resolviendo repo.mosquitto.org (repo.mosquitto.org)... 85.119.83.194, 2001:ba8:1
f1:f271::2
Conectando con repo.mosquitto.org (repo.mosquitto.org)[85.119.83.194]:80... cone
ctado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 3167 (3,1K) [application/octet-stream]
Grabando a: "mosquitto-repo.gpg.key"

mosquitto-repo.gpg. 100%[=====] 3,09K --.-KB/s en 0,001s
2020-11-11 23:07:10 (4,70 MB/s) - "mosquitto-repo.gpg.key" guardado [3167/3167]
```

Figura 77, Descarga del Fichero Mosquitto.

Se añadirá la clave para autenticar el paquete descargado utilizando el siguiente comando:

```
sudo apt-key add mosquitto-repo.gpg.key
```

Si todo resulta según lo esperado, dará aviso con un OK como respuesta. Luego se debe dirigir a la siguiente carpeta.

Comentado [s138]: M

```
cd /etc/apt/sources.list.d/
```

Y se debe descargar la lista de repositorios de Mosquitto con Wget, como el sistema operativo es la versión Buster, es necesario utilizar la siguiente línea de código:

`sudo wget http://repo.mosquitto.org/debian/mosquitto-buster.list`

```
pi@raspberrypi:~$  
pi@raspberrypi:~$  
pi@raspberrypi:~$ sudo apt-key add mosquito-repo.gpg.key  
OK  
pi@raspberrypi:~$ cd /etc/apt/sources.list.d/  
pi@raspberrypi:/etc/apt/sources.list.d$ sudo wget http://repo.mosquitto.org/debian/mosquitto-buster.list  
--2020-11-11 23:16:31-- http://repo.mosquitto.org/debian/mosquitto-buster.list  
Resolviendo repo.mosquitto.org (repo.mosquitto.org)... 85.119.83.194, 2001:ba8:1f1:f271::2  
Conectando con repo.mosquitto.org (repo.mosquitto.org)[85.119.83.194]:80... conectado.  
Petición HTTP enviada, esperando respuesta... 200 OK  
Longitud: 50 [application/octet-stream]  
Grabando a: "mosquitto-buster.list"  
  
mosquitto-buster.li 100%[=====] 50 --.-KB/s en 0s  
2020-11-11 23:16:31 (2,09 MB/s) - "mosquitto-buster.list" guardado [50/50]  
pi@raspberrypi:/etc/apt/sources.list.d$
```

Figura 78, Configuración de Mosquitto

Realizados los pasos anteriores, ya es posible instalar Mosquitto en la Raspberry PI, pero primero se debe realizar el *UPDATE* para actualizar la lista de paquete, para la instalación se utilizará dos líneas de código:

`Sudo apt-get install mosquitto`  
`Sudo apt-get install mosquitto-clients`

La primera es para instalar Mosquitto y la segunda es para instalar el cliente y poder realizar pruebas. Después de realizar estas instalaciones se debe modificar el archivo para indicarle en qué puerto debe conectar de Mosquitto utilizando la siguiente línea de código, dentro de la carpeta `mosquitto.conf` se tiene que agregar la siguiente línea que aparece en la figura.

`Sudo nano /etc/mosquitto/mosquitto.conf`

```
Place your local configuration in /etc/mosquitto/conf.d/
#
# A full description of the configuration file is at
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid_file /var/run/mosquitto.pid

persistence true
persistence_location /var/lib/mosquitto/

log_dest file /var/log/mosquitto/mosquitto.log

include_dir /etc/mosquitto/conf.d

listener 1883
```

Figura 79, Configuración de mosquitto.

Para mayor seguridad de que la instalación fue un éxito, se realizará una prueba como suscriptor y enviará un mensaje como cliente utilizando el Topic casa/dormitorio/luz.

```
pi@raspberrypi ~$ mosquitto_sub -h 192.168.10.66 -t casa/dormitorio/luz -m "recendido"
pi@raspberrypi ~$ mosquitto_pub -h 192.168.10.66 -t casa/dormitorio/luz -m "recendido"
pi@raspberrypi ~$ mosquitto_sub -h 192.168.10.66 -t casa/dormitorio/luz -m "recendido"
pi@raspberrypi ~$ mosquitto_pub -h 192.168.10.66 -t casa/dormitorio/luz -m "recendido"
pi@raspberrypi ~$
```

Figura 80, Comunicación exitosa MQTT.

#### Apéndice 4: Instalación de InfluxDB

Para almacenar los datos obtenidos por los sensores es necesario instalar una base de datos, se utilizará en este proyecto es InfluxDB ya que tiene una elevada velocidad de escritura. La primera línea de código que se debe utilizar es “*sudo apt-get update*”, la cual tiene función descargar paquetes repositorios y las actualiza para tener información sobre las versiones más reciente de los paquetes.

```
pi@raspberrypi:~$ sudo apt-get update
Des:1 http://archive.raspberrypi.org/debian buster InRelease [32,6 kB]
Des:2 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Des:3 http://archive.raspberrypi.org/debian buster/main armhf Packages [335 kB]
Obj:4 https://repo.mosquitto.org/debian buster InRelease
Des:5 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,4 MB en 13s (1.058 kB/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~$ wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -
OK
pi@raspberrypi:~$ source /etc/os-release
pi@raspberrypi:~$ echo "deb https://repos.influxdata.com/debian $(lsb_release -cs) stable"
| sudo tee /etc/apt/sources.list.d/influxdb.list
deb https://repos.influxdata.com/debian buster stable
pi@raspberrypi:~$ sudo apt-get update
Obj:1 http://archive.raspberrypi.org/debian buster InRelease
Obj:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Des:3 https://repos.influxdata.com/debian buster InRelease [4.731 B]
Obj:4 https://repo.mosquitto.org/debian buster InRelease
Des:5 https://repos.influxdata.com/debian buster/stable armhf Packages [926 B]
Descargados 5.657 B en 2s (2.672 B/s)
Leyendo lista de paquetes... Hecho
pi@raspberrypi:~$
```

Figura 81, Repositorio InfluxData.

Como aparece en la figura anterior, se debe agregar el repositorio de InfluxData, para eso se utiliza los siguientes códigos:

- `wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -`
- `source /etc/os-release`
- `echo 'deb https://repos.influxdata.com/debian $(lsb_release -cs) stable' | sudo tee /etc/apt/sources.list.d/influxdb.list`.

A continuación, se debe instalar Influxdb e iniciarlo.

- `Sudo apt-get update && sudo apt-get install influxdb`
- `Sudo service influxdb start`.

```
pi@raspberrypi:~$ sudo apt-get install influxdb
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
 libxiv2-14 libgfortran3 libgmime-2.6-0 libncurses5 libssl1.0.2 uuid-dev
 utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
 influxdb
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 4 no actualizados.
Se necesita descargar 60,0 MB de archivos.
Se utilizarán 153 MB de espacio de disco adicional después de esta operación.
Des:1 https://repos.influxdata.com/debian buster/stable armhf influxdb armhf 1.8.3-1 [60,0 MB]
Descargados 60,0 MB en 16s (3.674 KB/s)
Seleccionando el paquete influxdb previamente no seleccionado.
(Leyendo la base de datos ... 161435 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../influxdb_1.8.3-1_armhf.deb ...
Desempaquetando influxdb (1.8.3-1) ...
Configurando influxdb (1.8.3-1) ...
Created symlink /etc/systemd/system/influxd.service → /lib/systemd/system/influxdb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/influxdb.service → /lib/systemd/system/influxdb.service.
Procesando disparadores para man-db (2.8.5-2) ...
pi@raspberrypi:~$
```

Figura 82, Instalación InfluxDB.

### Apéndice 5: Instalación Grafana

Para instalar Grafana primero se debe observar la última actualización de este programa. Los siguientes comandos para su instalación son:

- *wget https://dl.grafana.com/oss/release/grafana\_6.5.1\_armhf.deb*
- *sudo dpkg -i grafana\_6.5.1\_armhf.deb*

Al tener estos comandos listos, se debe hacer un “update” para actualizar los paquetes que están instalados. El siguiente paso es instalar Grafana e iniciar el programa.

- *sudo apt-get install grafana*
- *sudo service grafana-server start*

Posterior a ello, se puede ingresar a la interfaz gráfica de usuario en el puerto 3000 con la siguiente estructura:

- *localhost:3000*

Comentado [s139]: M

## REFERENCIAS BIBLIOGRAFICAS

- Abril, X. B. (2019). *Tecnología wearable en indumentaria deportiva. Caso: Ciclismo BMX* (Bachelor's thesis, Universidad del Azuay).
- Acevedo, P. C. (2017). Einthoven y el electrocardiograma. *Revista de Hospital Italiano-Buenos Aires MCMLXXXI*, 29.
- AD8232, D. S. Single-Lead, Heart Rate Monitor Front End.
- Adonis, L. (2018, 4 diciembre). *Decidiendo cómo alertar con Grafana - devsChile*. Medium. <https://medium.com/devschile/decidiendo-como-alertar-con-grafana-6f21d508898f>
- Almenara, J. C., Osuna, J. B., & Obrador, M. (2017). Realidad aumentada aplicada a la enseñanza de la medicina. *Educación médica*, 18(3), 203-208.
- Álvarez Quispe, D. (2019). Diseño de un prototipo de monitoreo oximétrico Adulto-Pediátrico hospitalario.
- Analog Devices. (2012). *AD8232 Datasheet and Product Info | Analog Devices*. <https://www.analog.com/en/products/ad8232.html#product-overview>
- Arduino. (2019, 24 enero). *Software de Arduino*. Arduino.cl - Compra tu Arduino en Línea. <https://arduino.cl/programacion/>
- ARENI, R. P. (1996). LA INGENIERIA ELECTRONICA Y LA MEDICINA. *DYNA*, 71(6), 64-67.
- Benito Sánchez, R. (2017). Análisis de los campos electromagnéticos aplicados a la electromedicina para diagnosticar y tratar enfermedades.
- Bestbier, A. (2016). *Monitoring core temperature, heart rate, respiratory rate and EEG of an infant through a wireless ear probe* (Doctoral dissertation, Stellenbosch University).
- Brunete, H. R. S. S. P. A. Y. (2020, 28 junio). *2.1 Sensores industriales | Introducción a la Automatización Industrial*. <https://bookdown.org/>. [https://bookdown.org/alberto\\_brunete/intro\\_automatica/sensores-industriales.html](https://bookdown.org/alberto_brunete/intro_automatica/sensores-industriales.html)
- Campos, F. (2011, 13 diciembre). Electrocardiografía Básica: Colocación de electrodos. Recuperado de <http://formacionenemergencias.blogspot.com/2011/12/electrocardiografia-basica-colocacion.html>

- Cardiavant. 2020. *Holter De Tensión Arterial - Cardiavant*. [online] Available at: <<https://cardiavant.com/holter-tension-arterial/>> [Accessed 3 August 2020].
- Catalinas, E. Q. (2002). *Sistemas operativos y lenguajes de programación*. Editorial Paraninfo.
- CENETEC, S. (2005). Guía tecnológica No. 13 monitor de signos vitales. *Secretaría de Salud, Subsecretaría de Innovación y calidad, Centro Nacional de Excelencia Tecnológica en Salud. México*.
- Conde, C. P. (1996). *Sensores ópticos* (Vol. 30). Universitat de València.
- Davis, D. (2007). *Interpretación del ECG. Su dominio rápido y exacto*. Ed. Médica Panamericana.
- Diferencia entre lo analógico y digital. (2017, 19 octubre). Recuperado de <https://diferencias-entre.com/diferencia-entre-lo-analogico-y-digital/>
- Dote, J., Nahuelhual, P., Cubillos, R., Fuentes, G., & Zuñiga, J. (2020). Funcionalidad de prótesis de mano impresa en 3D en adolescentes con amputación congénita parcial de mano: una serie de casos. *Revista chilena de pediatría, (AHEAD)*, 0-0.
- Durán, T., & Ismael, E. (2015). *Diseño y elaboración de un prototipo de monitor de signos vitales aplicando métodos no invasivos con comunicación de datos a dispositivos móviles* (Bachelor's thesis).
- Estrada, H. G., Manzanero, A. P., Escobar, O. A. L., Vázquez, J. G., Sotelo, M. G. R., & Llanos, A. I. C. (2018). Monitor de signos vitales con comunicación inalámbrica Wi-Fi para unidad de cuidados intensivos desarrollado en LabVIEW y la tarjeta myRIO-1900. *Res. Comput. Sci*, 147(12), 203-212.
- Ferreira, M. (2015). Valoración de signos vitales. Recuperado de <http://24tes.blogspot.com/2015/10/valoracion-de-signos-vitales.html>.
- García, D. C., Martínez, G. Á., & García, M. E. (2018). Raspberry Pi y Arduino: semilleros en innovación tecnológica para la agricultura de precisión. *Informática y Sistemas: Revista de Tecnologías de la Informática y las Comunicaciones*, 2(1), 74-82.
- Gutiérrez Cisternas, C. A. (2016). *Sistema de monitoreo continuo de signos vitales con sensores no invasivos y transmisión inalámbrica de datos* (Doctoral dissertation, Universidad de Concepción. Facultad de Ingeniería. Departamento de Ingeniería Eléctrica).
- Hampton, J. (2019). *ECG fácil*. Elsevier.

- Hernández, M. (2018, 9 enero). *Robótica 2º ESO: Digital - Analógico*. <https://iesmiguelhernandez.es/>.  
<https://iesmiguelhernandez.es/moodle2/mod/page/view.php?id=9904>
- InfluxData (2016). InfluxDB: Storage Engine [Consulta: 16 de octubre de 2016] Recuperado de: [https://docs.influxdata.com/influxdb/v1.0/concepts/storage\\_engine/](https://docs.influxdata.com/influxdb/v1.0/concepts/storage_engine/)
- Llamas, L. (2019, 17 abril). *¿Qué es MQTT? Su importancia como protocolo IoT*. Luis Llamas. <https://www.luisllamas.es/que-es-mqtt-su-importancia-como-protocolo-iot/>
- Llamas, L. (2020, 12 septiembre). *Protocolos de comunicación para IoT*. Luis Llamas. <https://www.luisllamas.es/protocolos-de-comunicacion-para-iot/>
- Londoño, J. G., Duque, J. D. L., & Valdivieso, A. M. H. (2010). Diseño de un dispositivo portátil e inalámbrico para el monitoreo ambulatorio de la presión arterial no invasiva. *Revista Politécnica*, 6(11), 121-131.
- Maxim Integrated. (2020, 13 febrero). *MAX30102 High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health - Maxim Integrated*. [https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX30102.html/tb\\_tab0](https://www.maximintegrated.com/en/products/interface/sensor-interface/MAX30102.html/tb_tab0)
- MCI Electronics - Electrónica para makers y profesionales DIY. (s. f.). Recuperado 8 de mayo de 2020, de <https://www.mcielectronics.cl/page/homepage>
- Mechatronics, N. (s. f.). Pulsioxímetro MAX30102. Recuperado 28 de agosto de 2020, de <https://naylampmechatronics.com/biomedico/444-pulsioximetro-max30102.html>
- Melexis. (2019, septiembre). *Download datasheet for MLX90614*. <https://www.melexis.com/en/documents/documentation/datasheets/datasheet-mlx90614>
- Montoro, A. F. (2012). *Python 3 al descubierto*. Rc Libros.
- Núñez, C. V. (2009). Tecnologías de la información para la educación, investigación y aplicación en el área de la salud. Bondades y retos. *Salud Uninorte*, 25(2), 331-349.
- Parra, S. (2016). Cómo medir tus constantes vitales: más fácil que nunca. Recuperado de <https://muysaludable.sanitas.es/salud/como-medir-tus-constantes-vitales/>
- Penagos, S. P., Salazar, L. D., de Enfermería, E. C., Vera, F. E., & Cardioinfantil, F. (2005). Control de signos vitales. *Guías para manejo de Urgencias. Bogotá (Colombia): Fundación Cardioinfantil*, 1465-1473.



- Reyes, B. (2018). Diseño de sistema de monitoreo de signos vitales de pacientes en emergencias de la CÍA. de bomberos Salvadora Trujillo N 26.
- Reynolds, J., Barragán, L., & Cuéllar, F. (2011). Bioamplificadores. *Revista Colombiana de Cardiología*, 18(3), 131-143.
- Robles, F. (2019, 10 octubre). *¿Qué es la Frecuencia Respiratoria?* Lifeder. <https://www.lifeder.com/frecuencia-respiratoria/>
- Rozas, G. A. C. (2007). Electrocardiograma en Urgencias. Recuperado 10 de agosto de 2020, de <http://sintesis.med.uchile.cl/index.php/en/component/content/article/101-revision/r-de-urgencias/1885-electrocardiograma-en-urgencias>
- Ruiz Sánchez, W. R. (2016). *Redes de sensores inalámbricos enfocadas a la medicina con énfasis en control de los signos vitales en pacientes adultos mayores* (Master's thesis, PUCE).
- Soto, J. A. C., Gutiérrez, G. A. A., Romero, M. C., Rico, F. R., Duarte, S. G., & Lozoya, G. R. (2016). Diseño e implementación de un oxímetro de pulso con Labview y la NI MyDAQ. *Cultura Científica y Tecnológica*, (55).
- Sparkfun. (2017, 31 mayo). SparkFun Single Lead Heart Rate Monitor - AD8232. Recuperado de <https://www.sparkfun.com/products/12650>
- Ubunlog. (2018, 23 mayo). *Grafana, un software de código abierto para análisis y supervisión*. Ubunlog. <https://ubunlog.com/grafana-software-analisis-supervision/>
- Vallejo Mera, G. P. (2015). *Sistema de monitoreo de signos vitales y alerta de accidentes para personas con problemas de movilidad* (Bachelor's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, Electrónica e Industrial. Carrera de Ingeniería en Electrónica y Comunicaciones).
- Vignoni, J. (2003). Sensores.
- Zárate, J.R. (2020, 4 agosto). Recuperado de <https://www.diariomedico.com/opinion/jose-ramon-zarate/la-temperatura-corporal-media-ha-bajado-cinco-decimas-desde-1800.html>
- Zavala-Villeda, J. A. (2017). Descripción del electrocardiograma normal y lectura del electrocardiograma. *Revista Mexicana de Anestesiología*, 40(S1), 210-213.
- Zeroday. "A lua based firmware for wifi-soc esp8266". Github. Retrieved (2015).

## ANEXOS

### 10.1 Programación del NodeMCU

```
#include <ESP8266WiFi.h> //librería que permite conectar nodemcu a wifi
#include <PubSubClient.h> //permite comunicar mqtt
#include <Wire.h> //permite la comunicacion I2C
#include "MAX30105.h" // librería para poder usar el sensor MAX30102
#include "heartRate.h" // librería para poder realizar pulsaciones por minutos
MAX30105 particleSensor;
//***** conexión wifi *****//
const char* wifi_ssid = "VTR-2534266"; // nombre del WIFI
const char* wifi_password = "Hj6gqxwdbwy"; // clave del WIFI
//***** mqtt *****//
const char* mqtt_server = "192.168.0.10"; //IP de raspberry
const char* spo2_topic = "nodemcu/sensor/SPO2"; // topic de la spo2
const char* bfm_topic = "nodemcu/sensor/bfm";
const char* temp_topic = "nodemcu/sensor/temp";
const char* irValue_topic = "nodemcu/sensor/irValue";
const char* mqtt_username = "tesis";
const char* mqtt_password = "6sbbu7";
const char* clientID = "nodemcu";
WiFiClient wifiClient;
PubSubClient client(mqtt_server, 1883, wifiClient);
//*****//
const byte RATE_SIZE = 4; //Aumente esto para obtener más promedios. 4 es bueno.
byte rates[RATE_SIZE]; //Matriz de frecuencias cardiacas
byte rateSpot = 0;
long lastBeat = 0; //Hora a la que ocurrió el último latido
float bfm = 0.0; // Pulo por minutos
int beatAvg = 0; // promedio pulsaciones
float oxygen = 0; // oxígeno
int body_temp = 0.0; //temperatura corporal
unsigned long irValue = 0; //valor infrarrojo
unsigned long PTM = 0; // monitor de tiempo anterior
const int sending_interval = 1;
const float K1 = -45.06;
const float K2 = 30.354;
const float K3 = 94.845;
void setup()
{
  Serial.begin(115200);
  setup_wifi();
  connect_MQTT();
  // Initialize sensor
  if (!particleSensor.begin(Wire, I2C_SPEED_FAST)) //Use default I2C port, 400kHz speed
  {
    Serial.println("MAX30105 was not found. Please check wiring/power. ");
    while (1);
  }
  Serial.println("Place your index finger on the sensor with steady pressure.");
  particleSensor.setup(); //Configure sensor with default settings
  particleSensor.setPulseAmplitudeRed(0x0A); //Turn Red LED to low to indicate sensor is running
  particleSensor.setPulseAmplitudeGreen(0); //Turn off Green LED
}
void setup_wifi() //permite conectarse al wifi
{
  WiFi.begin(wifi_ssid, wifi_password);
  while(WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.print("");
  Serial.println("WiFi connected!");
}
```

```

Serial.print("IP address: ");
Serial.println(WiFi.localIP());
}
void connect_MQTT()
{
  if (client.connect(clientID, mqtt_username, mqtt_password))
  {
    Serial.println("Connected to MQTT Broker!");
  }
  else {
    Serial.println("Connection to MQTT Broker failed...");
    Serial.println("NodeMCU is restarting now!");
    ESP.restart();
  }
}
void loop()
{
  unsigned long currentTime = millis();

  irValue = particleSensor.getIR();
  unsigned long redValue = particleSensor.getRed();
  float Z = (float)redValue/(float)irValue;
  if (checkForBeat(irValue) == true)
  {
    body_temp = particleSensor.readTemperature();
    oxygen = 0;
    if(beatAvg != 0)
      oxygen = (int)((K1 * Z + K2) * Z + K3);
    //***** formula FC *****/
    unsigned long delta = millis() - lastBeat;
    lastBeat = millis();
    bfm = 60 / (delta / 1000.0);
    if (bfm < 255 && bfm > 20)
    {
      rates[rateSpot++] = (byte)bfm; //Store this reading in the array
      rateSpot %= RATE_SIZE; //Wrap variable
      //Take average of readings
      beatAvg = 0;
      for (byte x = 0 ; x < RATE_SIZE ; x++)
        beatAvg += rates[x];
      beatAvg /= RATE_SIZE;
    }
  }
  if(irValue > 50000)
  {
    Serial.print("IR=");
    Serial.print(irValue);
    Serial.print(", BPM=");
    Serial.print(bfm);
    Serial.print(", Avg BPM=");
    Serial.print(beatAvg);
    Serial.print(", Oxygen=");
    Serial.print(oxygen);
    Serial.print(", Body Temperature=");
    Serial.print(body_temp);
    Serial.println(" *C");
  }
  if(irValue < 50000)
  {
    bfm = 0;
    beatAvg = 0;
    oxygen = 0;
    body_temp = 0;
  }
  else{ Serial.println();
}
}

```

```

    if(currentTime - PTM > sending_interval)
    {
    connect_MQTT();
    if (client.publish(spo2_topic, String(oxygen).c_str()))
    {
    Serial.println("SPO2 sent!");
    }
    if (client.publish(bfM_topic, String(beatAvg).c_str()))
    {
    Serial.println("Pulso sent!");
    }
    if (client.publish(temp_topic, String(body_temp).c_str()))
    {
    Serial.println("temperatura sent!");
    }
    if (client.publish(temp_topic, String(irValue).c_str()))
    {
    Serial.println("irValue sent!");
    }
    PTM = currentTime;
    }
}

```

## 10.2 Script de Python para comunicar MQTT con InfluxDB

```

import re
from typing import NamedTuple

import paho.mqtt.client as mqtt
from influxdb import InfluxDBClient

INFLUXDB_ADDRESS = '192.168.0.10'
INFLUXDB_USER = 'tesis'
INFLUXDB_PASSWORD = '6sbbu7'
INFLUXDB_DATABASE = 'monitor'

MQTT_ADDRESS = '192.168.0.10'
MQTT_USER = 'tesis'
MQTT_PASSWORD = '6sbbu7'
MQTT_TOPIC = 'nodemcu/+/+'
MQTT_REGEX = 'nodemcu/([^\s]+)/([^\s]+)'
MQTT_CLIENT_ID = 'mqtt-influxdb'

influxdb_client = InfluxDBClient(INFLUXDB_ADDRESS, 8086, INFLUXDB_USER, INFLUXDB_PASSWORD, None)

class SensorData(NamedTuple):
    location: str
    measurement: str
    value: float

def on_connect(client, userdata, flags, rc):
    """ The callback for when the client receives a CONNACK response from the server. """
    print('Connected with result code ' + str(rc))
    client.subscribe(MQTT_TOPIC)

def _parse_mqtt_message(topic, payload):
    match = re.match(MQTT_REGEX, topic)
    if match:
        location = match.group(1)
        measurement = match.group(2)
        if measurement == 'status':
            return None
        return SensorData(location, measurement, float(payload))
    else:

```

```

        return None

def _send_sensor_data_to_influxdb(sensor_data):
    json_body = [
        {
            'measurement': sensor_data.measurement,
            'tags': {
                'location': sensor_data.location
            },
            'fields': {
                'value': sensor_data.value
            }
        }
    ]
    influxdb_client.write_points(json_body)

def on_message(client, userdata, msg):
    """The callback for when a PUBLISH message is received from the server."""
    print(msg.topic + ' ' + str(msg.payload))
    sensor_data = _parse_mqtt_message(msg.topic, msg.payload.decode('utf-8'))
    if sensor_data is not None:
        _send_sensor_data_to_influxdb(sensor_data)

def _init_influxdb_database():
    databases = influxdb_client.get_list_database()
    if len(list(filter(lambda x: x['name'] == INFLUXDB_DATABASE, databases))) == 0:
        influxdb_client.create_database(INFLUXDB_DATABASE)
    influxdb_client.switch_database(INFLUXDB_DATABASE)

def main():
    _init_influxdb_database()

    mqtt_client = mqtt.Client(MQTT_CLIENT_ID)
    mqtt_client.username_pw_set(MQTT_USER, MQTT_PASSWORD)
    mqtt_client.on_connect = on_connect
    mqtt_client.on_message = on_message

    mqtt_client.connect(MQTT_ADDRESS, 1883)
    mqtt_client.loop_forever()

if __name__ == '__main__':
    print('MQTT to InfluxDB bridge')
    main()

```