



# ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y  
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERA EN ELECTRÓNICA Y  
TELECOMUNICACIONES**

**TEMA: SISTEMA DE MONITOREO REMOTO Y VISUALIZACIÓN  
PARA DISPOSITIVOS DE ANÁLISIS DE SIGNOS VITALES  
ORIENTADOS A E-HEALTH.**

**AUTOR: TERÁN FLORES, ESTEFANÍA DE LOS ÁNGELES**

**DIRECTOR: ING. TRIVIÑO CEPEDA, ROBERTO DANIEL MSc**

**SANGOLQUÍ**

**2019**



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**CERTIFICACIÓN**

Certifico que el trabajo de titulación, “**SISTEMA DE MONITOREO REMOTO Y VISUALIZACIÓN PARA DISPOSITIVOS DE ANÁLISIS DE SIGNOS VITALES ORIENTADOS A E-HEALTH**” fue realizado por la señorita **Terán Flores, Estefanía de los Ángeles** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustenten públicamente.

**Sangolquí, 06 de junio del 2019**

Firma:

**Ing. Roberto Daniel Triviño Cepeda, MSc**

**DIRECTOR**



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES**  
**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**AUTORÍA DE RESPONSABILIDAD**

Yo **Terán Flores, Estefanía de los Ángeles**, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Sistema de monitoreo remoto y visualización para dispositivos de análisis de signos vitales orientados a e-health”** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

**Sangolquí, 06 de junio del 2019**

Firma:

**Estefanía de los Ángeles Terán Flores**

**CI: 1003536339**



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES  
CARRERA DE INGENIERÍA EN ELECTRÓNICA Y TELECOMUNICACIONES**

**AUTORIZACIÓN**

Yo **Terán Flores, Estefanía de los Ángeles**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Sistema de monitoreo remoto y visualización para dispositivos de análisis de signos vitales orientados a e-health”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

**Sangolquí, 06 de junio del 2019**

Firma:

**Estefanía de los Ángeles Terán Flores**

**CI: 1003536339**

## **DEDICATORIA**

*El siguiente trabajo se lo dedico a mis padres Ángel y Magdalena, que han sido la luz y ejemplo en mi vida, ellos han sabido guiarme y nunca me han dejado desvanecer.*

*A mis hermanos, Rubén, Diego y Lady, por ser mi ejemplo, mis amigos y sobre todo estar en mi vida en los buenos y malos momentos.*

*A toda mi familia por ayudarme a obtener este logro que es de todos.*

*A Luis por creer en mí y apoyarme incondicionalmente.*

**Estefanía de los Ángeles Terán Flores**

## **AGRADECIMIENTOS**

A Dios, por siempre guiarme en mi camino y darme las fuerzas suficientes para no rendirme.

A mis padres por apoyarme a lo largo de mi vida en todo momento y por ser el motor para lograr concluir etapa de mi vida.

A la prestigiosa Universidad de las Fuerzas Armadas ESPE, donde adquirí muchas experiencias, que me ha preparado para un futuro competitivo y donde forjé grandes lazos de amistad.

A mi tutor de tesis, Ing. Daniel Triviño por brindarme sus conocimientos, ayuda y apoyarme en la realización del proyecto de investigación.

Gracias a todos.

**Estefanía de los Ángeles Terán Flores**

## ÍNDICE DE CONTENIDOS

<b>CERTIFICADO DEL DIRECTOR.....</b>	<b>i</b>
<b>AUTORÍA DE RESPONSABILIDAD.....</b>	<b>ii</b>
<b>AUTORIZACIÓN.....</b>	<b>iii</b>
<b>DEDICATORIA.....</b>	<b>iv</b>
<b>AGRADECIMIENTOS.....</b>	<b>v</b>
<b>ÍNDICE DE CONTENIDOS.....</b>	<b>vi</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>x</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>xii</b>
<b>RESUMEN.....</b>	<b>xvii</b>
<b>ABSTRACT.....</b>	<b>xviii</b>
<b>CAPÍTULO 1.....</b>	<b>1</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
1.1 Antecedentes.....	1
1.2 Justificación e Importancia.....	3
1.3 Alcance.....	5
1.4 Objetivos.....	6
1.4.1 Objetivo General.....	6
1.4.2 Objetivos Específicos.....	6
<b>CAPÍTULO 2.....</b>	<b>7</b>
<b>ESTADO DEL ARTE.....</b>	<b>7</b>
2.1 E-Health.....	7
2.1.1 Repercusión de e-Health.....	7
2.1.2 Sistemas de monitoreo remoto de e-Health.....	8
2.2 Internet de las Cosas (IoT).....	9
2.2.1 Dispositivos utilizados en IoT.....	11
2.2.2 Arquitectura de IoT.....	13
Capa de percepción.....	13

Capa de red.....	14
Capa de aplicación .....	15
2.2.3 Protocolos de Mensajería para IoT.....	15
2.2.4 Estándar de e-Health .....	23
IEEE 1073 Point of Care Medical Device Communication.....	23
<b>CAPÍTULO 3.....</b>	<b>26</b>
<b>DISEÑO E IMPLEMENTACIÓN .....</b>	<b>26</b>
3.1 Requerimientos de hardware y software .....	27
3.1.1 Requerimientos de Hardware .....	27
3.1.2 Requerimientos de Software .....	28
3.1.2.1 Requisitos funcionales.....	29
3.1.2.2 Requisitos no funcionales.....	30
3.2 Diseño del Hardware .....	31
3.2.1 Módulos de adquisición de datos .....	31
3.2.1.1 Sensor de temperatura ambiente .....	31
3.2.1.2 Sensor de temperatura corporal.....	33
3.2.1.3 Sensor de ritmo cardiaco .....	35
3.2.1.4 Módulo detector de caídas.....	38
3.2.2 Repotenciación de la placa.....	40
3.2.2.1 Tarjeta ESP32.....	40
3.2.2.2 Modificaciones de la placa.....	43
3.2.2.3 Dimensiones de la placa .....	45
3.2.2.4 Peso del dispositivo e-Health .....	46
3.3 Diseño del software.....	48
3.3.1 Proceso de adquisición e instalación del servidor en la nube. ....	49
3.3.2 Protocolo de mensajería: MQTT.....	52
3.3.3 Proceso instalación de Node-RED .....	53
3.3.3.1 Inicialización de Node-RED .....	56
3.3.4 IDE de Arduino .....	56
3.3.4.1 Conexión a WiFi y al servidor en la nube.....	56



3.3.4.2	Desarrollo del software del procesamiento de temperatura ambiente y corporal. ....	60
3.3.4.3	Desarrollo del software del procesamiento para detección de caídas. ....	61
3.3.4.4	Desarrollo del software del procesamiento para el ritmo cardiaco. ....	63
3.3.5	Aplicación web.....	64
3.3.5.1	Inicio de sesión.....	66
3.3.5.2	Cierre de sesión.....	67
3.3.5.3	Registro de pacientes.....	68
3.3.5.4	Visualización de los datos monitoreados.....	70
3.3.5.5	Consulta de datos monitoreados.....	72
3.3.5.6	Reglas de validación.....	76
3.3.6	Aplicación móvil.....	77
3.3.6.1	Desarrollo de la aplicación.....	78
3.3.7	Base de Datos (BD).....	85
3.4	Protocolo de pruebas.....	88
3.4.1	Funcionamiento de los sensores y el módulo detector de caídas. ....	89
	Parte A: Funcionamiento del sensor de temperatura ambiente. ....	89
	Parte B: Funcionamiento del sensor de temperatura corporal.....	90
	Parte C: Funcionamiento del sensor de ritmo cardíaco.....	92
	Parte D: Funcionamiento del módulo detector de caídas.....	93
3.4.2	Envío de datos mediante comunicación WiFi al Servidor en la nube y almacenamiento de datos en la Base de Datos. ....	95
	Parte A: Verificación del protocolo de mensajería MQTT.....	96
3.4.3	Envío de datos a la tarjeta de memoria.....	97
3.4.4	Funcionalidad del Sistema de monitoreo. ....	98
3.4.5	Prueba de carga y descarga de la batería del dispositivo e-Health.....	100
	Parte A: Carga de la batería del dispositivo e-Health.....	100
	Parte B: Descarga de la batería del dispositivo e-Health.....	102
	<b>CAPÍTULO 4.....</b>	<b>104</b>
	<b>ANÁLISIS DE RESULTADOS.....</b>	<b>104</b>
4.1	Evaluación del hardware.....	104

4.1.1	Resultados: “Funcionamiento de los sensores y el módulo detector de caídas” .....	104
	Resultados parte A: Análisis del sensor de temperatura ambiente.....	104
	Resultados parte B: Análisis del sensor de temperatura corporal .....	105
	Resultados parte C: Análisis del sensor de ritmo cardiaco .....	108
	Resultados parte D: Análisis del módulo detector de caídas .....	109
4.2	Evaluación del software .....	114
4.2.1	Resultados: “Envió de datos mediante comunicación WiFi al Servidor en la Nube y almacenamiento de datos en la Base de Datos” .....	114
	Resultados Parte A: Verificación del protocolo de mensajería MQTT .....	119
4.2.2	Resultados: Envío de datos en la tarjeta de memoria.....	124
4.2.3	Resultados: “Funcionalidad del Sistema de Monitoreo”.....	126
	Datos del dispositivo e-Health .....	126
	Aplicación web.....	128
	Aplicación móvil.....	134
	Análisis de resultados.....	135
4.2.4	Resultados: Carga y descarga de la batería del dispositivo e-Health.....	136
	Resultados parte A: Carga de la batería del dispositivo.....	136
	Resultados parte B: Descarga de la batería del dispositivo e-Health.....	138
	Análisis de resultados.....	138
	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>141</b>
	Conclusiones .....	141
	Recomendaciones.....	144
	<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>145</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> <i>Brokers compatibles con MQTT</i> .....	18
<b>Tabla 2</b> <i>Paquetes de control MQTT</i> .....	19
<b>Tabla 3</b> <i>Análisis comparativo de Protocolos de Mensajería para Sistemas IoT.</i> .....	21
<b>Tabla 4</b> <i>Características Sensor de temperatura LM35</i> .....	31
<b>Tabla 5</b> <i>Valores promedios del sensor de temperatura ambiente</i> .....	33
<b>Tabla 6</b> <i>Características Termistor NXFT15H103FA2B100</i> .....	33
<b>Tabla 7</b> <i>Valores promedios del sensor de temperatura corporal</i> .....	35
<b>Tabla 8</b> <i>Características Sensor HRM-2511B</i> .....	36
<b>Tabla 9</b> <i>Valores promedios del sensor de ritmo cardiaco</i> .....	37
<b>Tabla 10</b> <i>Características del módulo ADXL335</i> .....	38
<b>Tabla 11</b> <i>Cambio de posición de los ejes</i> .....	39
<b>Tabla 12</b> <i>Características de la tarjeta ESP32</i> .....	41
<b>Tabla 13</b> <i>Pines de conexión de los módulos con la tarjeta ESP32</i> .....	42
<b>Tabla 14</b> <i>Comparación entre el dispositivo con tarjeta Teensy y ESP32</i> .....	44
<b>Tabla 15</b> <i>Peso de la placa anterior y actual</i> .....	46
<b>Tabla 16</b> <i>Estructura plástica del dispositivo e-Health</i> .....	47
<b>Tabla 17</b> <i>Comparación entre framework</i> .....	54
<b>Tabla 18</b> <i>Usuarios: Doctores</i> .....	65
<b>Tabla 19</b> <i>Usuarios: Enfermeras</i> .....	65
<b>Tabla 20</b> <i>Usuarios: Pacientes</i> .....	65
<b>Tabla 21</b> <i>Multiusuarios</i> .....	66
<b>Tabla 22</b> <i>Descripción de los nodos de inicio de sesión</i> .....	66
<b>Tabla 23</b> <i>Descripción de los nodos de cierre de sesión</i> .....	67
<b>Tabla 24</b> <i>Descripción de los nodos para el registro de pacientes</i> .....	68
<b>Tabla 25</b> <i>Descripción de los nodos para la visualización de datos</i> .....	70
<b>Tabla 26</b> <i>Descripción de los nodos para la consulta de datos</i> .....	72
<b>Tabla 27</b> <i>Reglas de validación de la aplicación web</i> .....	76
<b>Tabla 28</b> <i>Propiedades básicas de la interfaz de inicio de sesión.</i> .....	80

<b>Tabla 29</b> <i>Propiedades básicas de la interfaz para el monitoreo de los datos</i> .....	81
<b>Tabla 30</b> <i>Contenido de cada tabla</i> .....	85
<b>Tabla 31</b> <i>Tipo de datos en la tabla usuarios</i> .....	86
<b>Tabla 32</b> <i>Tipo de datos en la tabla pacientes</i> .....	86
<b>Tabla 33</b> <i>Tipo de datos en la tabla datos</i> .....	86
<b>Tabla 34</b> <i>Descripción de los nodos para el almacenamiento de los datos</i> .....	87
<b>Tabla 35</b> <i>Tabla para temperatura ambiente</i> .....	90
<b>Tabla 36</b> <i>Tabla para temperatura corporal</i> .....	92
<b>Tabla 37</b> <i>Tabla para ritmo cardiaco</i> .....	93
<b>Tabla 38</b> <i>Sensor de temperatura ambiente</i> .....	105
<b>Tabla 39</b> <i>Sensor de temperatura corporal</i> .....	106
<b>Tabla 40</b> <i>Sensor de temperatura corporal</i> .....	107
<b>Tabla 41</b> <i>Sensor de ritmo cardiaco</i> .....	109
<b>Tabla 42</b> <i>Ejemplo caída frontal</i> .....	110
<b>Tabla 43</b> <i>Ejemplo caída hacia la izquierda</i> .....	111
<b>Tabla 44</b> <i>Ejemplo caída hacia la izquierda</i> .....	112
<b>Tabla 45</b> <i>Cambio N°2 de posición de los ejes</i> .....	113
<b>Tabla 46</b> <i>Envío de paquetes al Servidor en la nube</i> .....	115
<b>Tabla 47</b> <i>Envío de paquetes al Servidor en la nube (diferente tamaño)</i> .....	116
<b>Tabla 48</b> <i>Cantidad de pruebas</i> .....	125
<b>Tabla 49</b> <i>Tiempo de carga de la batería</i> .....	138

## ÍNDICE DE FIGURAS

<i>Figura 1.</i> Cronología de políticas de e-Health en diferentes países. ....	7
<i>Figura 2.</i> Repercusiones de e-Health. ....	8
<i>Figura 3.</i> Aplicaciones de IoT. ....	10
<i>Figura 4.</i> Sistema de SmarHome IoT. ....	11
<i>Figura 5.</i> Sensores de e-Health. ....	12
<i>Figura 6.</i> Arquitectura básica de IoT. ....	13
<i>Figura 7.</i> Arquitectura de MQTT. ....	18
<i>Figura 8.</i> Entorno de CoAP. ....	20
<i>Figura 9.</i> Arquitectura AMQP. ....	21
<i>Figura 10.</i> Pila de protocolos de IEEE 1073. ....	23
<i>Figura 11.</i> Diagrama de paquetes según el modelo de información (DIM). ....	24
<i>Figura 12.</i> Diagrama de bloques general. ....	26
<i>Figura 13.</i> Diagrama de red del Sistema de monitoreo. ....	27
<i>Figura 14.</i> Requisitos del diseño del hardware. ....	28
<i>Figura 15.</i> Diagrama de conexión del sensor LM35. ....	32
<i>Figura 16.</i> Evaluación del sensor LM35. ....	32
<i>Figura 17.</i> Diagrama de conexión del sensor NTC. ....	34
<i>Figura 18.</i> Evaluación del sensor NTC. ....	34
<i>Figura 19.</i> Posición de ubicación del sensor de ritmo cardiaco. ....	35
<i>Figura 20.</i> Configuración interna del sensor HRM-2511B. ....	36
<i>Figura 21.</i> Evaluación del sensor HRM-2511B. ....	37
<i>Figura 22.</i> Diagrama de conexión del sensor ADLX335. ....	38
<i>Figura 23.</i> Posición de los ejes del acelerómetro. ....	39
<i>Figura 24.</i> Evaluación del sensor ADXL335. ....	40
<i>Figura 25.</i> Comparación de la tarjeta Teensy 3.6 con ESP32. ....	42
<i>Figura 26.</i> Diagrama de bloques del dispositivo de e-Health. ....	42
<i>Figura 27.</i> Comparación de dimensiones entre placas. ....	45
<i>Figura 28.</i> Dispositivo visto desde el lado 1. ....	46
<i>Figura 29.</i> Dispositivo visto desde el lado 2. ....	46

<b>Figura 30.</b> Esquema general de tecnologías usadas. ....	49
<b>Figura 31.</b> Creación del servidor en la nube .....	50
<b>Figura 32.</b> Herramienta MobaXterm.....	51
<b>Figura 33.</b> Pasos de inicio de sesión del droplet. ....	51
<b>Figura 34.</b> Proceso de creación de la llave privada.....	52
<b>Figura 35.</b> Instalación del mosquito.....	53
<b>Figura 36.</b> Instalación de Node-RED.....	54
<b>Figura 37.</b> Entorno de Node-RED.....	55
<b>Figura 38.</b> Inicialización de Node-RED.....	56
<b>Figura 39.</b> Diagrama de flujo de la conexión a WiFi y al servidor en la nube. ....	58
<b>Figura 40.</b> Conexión a WiFi.....	59
<b>Figura 41.</b> Diagrama de publicación y suscripción de los tópicos.....	59
<b>Figura 42.</b> Publicación de tópicos .....	60
<b>Figura 43.</b> Diagrama de flujo para obtener temperatura ambiente y corporal .....	60
<b>Figura 44.</b> Obtención valor de temperatura a) ambiente y b) corporal .....	61
<b>Figura 45.</b> Diagrama de flujo para Detectar caídas.....	62
<b>Figura 46.</b> Obtención de ángulos .....	63
<b>Figura 47.</b> Diagrama de flujo para obtener ritmo cardiaco .....	63
<b>Figura 48.</b> Obtención del valor de ritmo cardiaco.....	64
<b>Figura 49.</b> Flujo de inicio de sesión. ....	67
<b>Figura 50.</b> Flujo de cierre de sesión. ....	68
<b>Figura 51.</b> Flujo de registro de pacientes. ....	69
<b>Figura 52.</b> Flujo de datos monitoreados .....	71
<b>Figura 53.</b> Flujo de consulta de datos.....	72
<b>Figura 54.</b> Flujo total.....	73
<b>Figura 55.</b> Estructura de la página de inicio de sesión.....	74
<b>Figura 56.</b> Estructura de la página de monitoreo de datos, usuario:doctor .....	75
<b>Figura 57.</b> Estructura de la página de monitoreo de datos, usuario:enfermera .....	75
<b>Figura 58.</b> Estructura de la página de monitoreo de datos, usuario:paciente .....	76
<b>Figura 59.</b> Estructura de la página de descarga de datos, usuario:paciente .....	76

<b>Figura 60.</b> Características de Servicio de Paho.....	78
<b>Figura 61.</b> Líneas de código para el Servicio de Paho de Android.....	79
<b>Figura 62.</b> Líneas de código para conexión y permiso.....	79
<b>Figura 63.</b> Líneas de código para crear y establecer conexión Mqtt.....	79
<b>Figura 64.</b> Líneas de código de suscripción de los tópicos.....	80
<b>Figura 65.</b> Estructura de la selección de los usuarios de la aplicación móvil.....	81
<b>Figura 66.</b> Estructura de inicio de sesión.....	81
<b>Figura 67.</b> Estructura de datos para Usuario: Doctores.....	83
<b>Figura 68.</b> Estructura de datos para Usuario: Enfermeras.....	83
<b>Figura 69.</b> Estructura de datos para Usuario: Pacientes.....	84
<b>Figura 70.</b> Diagrama de flujo de uso de la aplicación móvil.....	84
<b>Figura 71.</b> Relación de las tablas de la base de datos.....	86
<b>Figura 72.</b> Flujo de almacenamiento de datos.....	87
<b>Figura 73.</b> Escenario de la Parte A.....	90
<b>Figura 74.</b> Escenario de la Parte B.....	92
<b>Figura 75.</b> Cambio de posición del sensor de temperatura corporal.....	92
<b>Figura 76.</b> Escenario de la Prueba C.....	93
<b>Figura 77.</b> Escenario de la Prueba D.....	94
<b>Figura 78.</b> Movimientos que debe realizar el sujeto de prueba.....	94
<b>Figura 79.</b> Escenario de la Prueba 3.4.2.....	96
<b>Figura 80.</b> Formato de la tabla “datos”.....	98
<b>Figura 81.</b> Escenario de la Prueba 3.4.3.....	98
<b>Figura 82.</b> Posición del sujeto de prueba 3.4.4.....	99
<b>Figura 83.</b> Escenario de la prueba del Sistema de monitoreo.....	100
<b>Figura 84.</b> Escenario de la Parte A.....	101
<b>Figura 85.</b> Modificación de la Parte A.....	102
<b>Figura 86.</b> Escenario de la Parte B.....	103
<b>Figura 87.</b> Salida del sensor de temperatura ambiente.....	105
<b>Figura 88.</b> Salida del sensor de temperatura corporal.....	106
<b>Figura 89.</b> Salida del sensor de temperatura corporal.....	107

<b>Figura 90.</b> Evolución de salida del sensor de ritmo cardiaco.....	108
<b>Figura 91.</b> Ejemplo prueba del detector de caída.....	109
<b>Figura 92.</b> Ejemplo prueba con caída hacia al frente.....	110
<b>Figura 93.</b> Ejemplo prueba de caída hacia el lado derecho.....	111
<b>Figura 94.</b> Ejemplo prueba de caída hacia el lado izquierdo.....	112
<b>Figura 95.</b> Simulación de convulsiones.....	113
<b>Figura 96.</b> ICMP enviado al servidor, 100 paquetes.....	115
<b>Figura 97.</b> ICMP enviado al servidor, 1000 paquetes.....	115
<b>Figura 98.</b> ICMP enviado al servidor, 10000 paquetes.....	115
<b>Figura 99.</b> ICMP enviado al servidor, tamaño del paquete 10 bytes.....	115
<b>Figura 100.</b> ICMP enviado al servidor, tamaño del paquete 100 bytes.....	116
<b>Figura 101.</b> ICMP enviado al servidor, tamaño del paquete 1000 bytes.....	116
<b>Figura 102.</b> Datos monitor serial.....	117
<b>Figura 103.</b> Datos en el “Debug messages”.....	117
<b>Figura 104.</b> Datos en la Base de Datos.....	118
<b>Figura 105.</b> Datos en la Base de Datos, cambio de red WiFi.....	118
<b>Figura 106.</b> Datos en la SD card, cambio de red WiFi.....	119
<b>Figura 107.</b> Captura de datos utilizando Wireshark.....	119
<b>Figura 108.</b> Tamaño del paquete capturado.....	120
<b>Figura 109.</b> Características del paquete enviado 1.....	120
<b>Figura 110.</b> Características del paquete enviado 2.....	120
<b>Figura 111.</b> Características del paquete analizado.....	121
<b>Figura 112.</b> Suscripción a un tópico.....	122
<b>Figura 113.</b> Diagrama de suscripción del Cliente MQTT.....	122
<b>Figura 114.</b> Suscripción a tópicos de tcorp, rcard y dcaid.....	123
<b>Figura 115.</b> Suscripción a tópicos de acorp, aritm y ccaid.....	123
<b>Figura 116.</b> Suscripción a tópicos de ccorp, critm, pamb y pcorp.....	124
<b>Figura 117.</b> Características del archivo creado .txt.....	125
<b>Figura 118.</b> Archivo almacenado en la tarjeta de memoria.....	126
<b>Figura 119.</b> Configuración de la red en el programa.....	127



<b>Figura 120.</b> Mensajes de la pantalla OLED .....	127
<b>Figura 121.</b> Ventana de inicio de sesión del usuario doctor y enfermera .....	128
<b>Figura 122.</b> Ventana de datos, usuario:doctor.....	129
<b>Figura 123.</b> Promedio de datos monitoreados, usuario:doctor.....	129
<b>Figura 124.</b> Registro del paciente.....	130
<b>Figura 125.</b> Consulta de los datos moniteoreados.....	130
<b>Figura 126.</b> Registro del paciente para el almacenamiento en la BD .....	131
<b>Figura 127.</b> Almacenamiento de datos en la BD.....	131
<b>Figura 128.</b> Terminar el almacenamiento de datos en la BD .....	132
<b>Figura 129.</b> Ventana de datos, usuario: enfermera.....	132
<b>Figura 130.</b> Ventana de datos, usuario: paciente.....	133
<b>Figura 131.</b> Ventana de datos, usuario paciente.....	133
<b>Figura 132.</b> Ventana de elección de usuarios.....	134
<b>Figura 133.</b> Ventana de inicio de los diferentes usuarios.....	134
<b>Figura 134.</b> Ventana de datos monitoreados (A) enfemera, (B) doctor y (C) paciente .....	135
<b>Figura 135.</b> Datos monitoreados del dispositivo, prueba de Funcionalidad del sistema .....	136
<b>Figura 136.</b> Datos monitoreados del dispositivo, Prueba A.....	137
<b>Figura 137.</b> Voltaje de la batería cargada, Prueba A.....	137
<b>Figura 138.</b> Voltaje de la batería carga .....	138
<b>Figura 139.</b> Aviso de alarmas, prueba N°2 .....	139
<b>Figura 140.</b> Voltaje de batería descargada .....	140

## **RESUMEN**

El presente proyecto de investigación describe la repotenciación, diseño e implementación de un sistema de monitoreo y visualización de signos vitales orientado a un dispositivo e-Health. Para la repotenciación del dispositivo del grupo Bio4Tech de la Universidad de las Fuerzas Armadas ESPE, se ha utilizado la tarjeta de adquisición ESP 32 con pantalla OLED que incluye un módulo WiFi lo que permitió la reducción de tamaño para el subsistema de comunicaciones y por tanto el peso. Este dispositivo realiza la adquisición de datos de temperatura ambiente, temperatura corporal, ritmo cardiaco y posee un detector de caídas. Los datos recolectados de los sensores son enviados a un Bróker (Servidor en la nube) utilizando el protocolo de mensajería MQTT a través de una red de comunicación inalámbrica (IEEE 802.11) con acceso a Internet, de esta forma se envían los datos para el almacenamiento en una Base de Datos. La información almacenada se visualiza en una aplicación web, desarrollada con la herramienta visual Node-RED, que permite el acceso y visualización casi en tiempo real a doctores, enfermeras y pacientes. Entre la información disponible se muestran alarmas tales como caídas del paciente o irregularidades en los datos monitoreados. Además, se dispone de una aplicación móvil desarrollada en Android Studio para teléfonos inteligentes, que muestra los datos monitoreados del paciente como complemento para el sistema.

### **PALABRAS CLAVE:**

- **DISPOSITIVO E-HEALTH**
- **MQTT**
- **APLICACIONES**

## **ABSTRACT**

This research project describes the repowering, design and implementation of a vital signs monitoring and visualization system oriented to an e-Health device. For the repowering of the device of the Bio4Tech group of the University of the Armed Forces ESPE, has been used the acquisition card ESP 32 with OLED display that includes a WiFi module which allowed the reduction of size for the communications subsystem as well as weight. This device performs the acquisition of data from ambient temperature, body temperature, heart rate and has a fall detector. The data collected from the sensors, is sent to a Broker (Cloud Server) using the MQTT messaging through wireless communication (IEEE 802.11) with Internet access, thus sending the data for the storage in a Database. The stored information is displayed on a web application, developed with the visual tool Node-RED, which allows access and visualization almost in real time for doctors, nurses and patients. Among the available information are alarms such as patient falls or irregularities in the monitored data. In addition, there is a mobile application developed in Android Studio for smart phones, which shows the monitored data of the patient as a complement to the system.

### **KEYWORDS:**

- **E-HEALTH DEVICE**
- **MQTT**
- **APPLICATIONS**

# CAPÍTULO 1

## INTRODUCCIÓN

### 1.1 Antecedentes

En los últimos años la importancia de la salud en la sociedad ha tomado mayor interés; alrededor del mundo el ser humano necesita poseer una condición sana y se ha convertido en una necesidad acudir a los hospitales con frecuencia para realizar una consulta médica, pero este procedimiento es lento y no garantiza el acceso a todas las personas (Anand, Nisar, Naresh, Nagendra, & Mithun, 2018). Existen diferentes factores económicos, geográficos, sociales, entre otros, que afectan la calidad de la asistencia hospitalaria, por esta razón se han planteado soluciones como: diagnósticos en tiempo real de problemas médicos, telemedicina, monitoreo remoto de pacientes, así como el transporte inteligente asistido por computadora en caso de emergencias (Kotronis et al., 2018), que proporcionan diferentes sistemas eficaces y accesibles a todos.

Además, al implementar un servicio de e-Health basado en Internet de las Cosas, por sus siglas en inglés (IoT), que posee gran potencial y que conjuntamente con la popularidad de los dispositivos médicos móviles inteligentes, los dispositivos de sensores biomédicos portátiles, la computación en la nube y el análisis de big data (Hussien et al., 2016), se lograría beneficios tanto para pacientes como para profesionales del campo de la salud. De esta forma según Borton (2014), en e-Health los sensores corporales son herramientas que detectan procesos biológicos, químicos, o físicos y luego transmiten o reportan esta información; algunos sensores trabajan fuera del cuerpo, mientras que otros están diseñados para ser implantados dentro del mismo, es decir son objetos inteligentes capaces de proporcionar el monitoreo remoto como signos vitales de pacientes a un bajo costo y en tiempo real (Maia et al., 2015).

Los sensores biomédicos y el Internet, en el campo de la salud, trabajan en conjunto, por eso se han desarrollado gran variedad de aplicaciones móviles, durante el año 2017 se calcula que se descargaron 262 692 millones de aplicaciones en salud con buenos y malos resultados (Alonso-Arévalo & Mirón-Canelo, 2017). Por ejemplo, Google en el 2008 lanzó Google Health, una base de datos personal de información online (Personal Health Record – PHR) que no fue aceptable entre los usuarios y fracasó, siendo cerrada en 2013. Con el lanzamiento de Google Fit en el ecosistema Android en el año 2014, intentó reparar del problema de la falta de usuarios en temas de salud enfocándose más al control de ejercicio y a la nutrición (Llordachs, 2016).

Por otro lado, muchos usuarios desinstalan las diferentes aplicaciones por que estas no cumplen sus expectativas (Alonso-Arévalo & Mirón-Canelo, 2017). Para evitar estos inconvenientes se debe desarrollar diversas aplicaciones que contengan una interfaz gráfica que sea agradable para el usuario y de fácil uso, además que se proteja la seguridad de los datos personales de los beneficiarios.

Las aplicaciones móviles de e-Health son diversas y se puede clasificar en dos grupos según su gestión de administración:

- **App para pacientes:** Cuando los usuarios disponen de la aplicación sin ninguna supervisión de un especialista.
- **App para doctores:** Aplicaciones móviles dirigidas a los proveedores de atención médica, es decir para los profesionales de la salud.

Según el informe “The mobile health global market report 2013-2017: the Commercialization of mHealth apps”, el 70 % de las aplicaciones son destinadas a pacientes y el 30 % son aplicaciones de uso profesional (Alonso-Arévalo & Mirón-Canelo, 2017)

Actualmente existe gran cantidad de plataformas de e-Health en el mercado que ofrecen diversidad de opciones para los usuarios, entre las principales se encuentran: MediSafe que permite recordar la toma de medicamentos a través del envío de notificaciones; Endomondo es un entrenador personal de bolsillo; Mobile MIM dirigido a profesionales que permite la visualización, intercambio y registro de imágenes, radiografías y ultrasonidos; My Suger admite realizar un control referente a la enfermedad de la diabetes, entre otras.

## **1.2 Justificación e Importancia**

En los últimos años, los sistemas y aplicaciones orientados a la salud por medio de dispositivos electrónicos han ganado gran popularidad, debido a los costos reducidos y la mejora que se ha logrado en la calidad de los servicios de salud (Ivascu, Frincu, & Negru, 2016). De hecho la sinergia entre e-Health e IoT generará alrededor de 8 billones de dólares de valor en juego durante los próximos diez años (Noronha, Moriarty, O'Connell, & Villa, 2014). Por esta razón se ha seleccionado este campo de aplicación como una perspectiva a futuro.

Con e-Health se modificarán completamente la comunicación de persona a persona, de persona a máquina y de máquina a máquina para la ayuda de todos (Maksimovic, M; Vujovic, 2017). El desarrollo de e-Health se relaciona directamente con la creación de una variedad de aplicaciones móviles para atención médica que proporcionan múltiples beneficios como la supervisión remota y continúa de datos de los pacientes; permitiendo prevención, diagnóstico, tratamiento y seguimiento.

Por otro lado la Telemedicina/Telesalud es un modelo de atención que aprovecha las modernas y actuales Tecnologías de Información y Comunicación (TICs) con el objeto de brindar asistencia médica a quien lo requiera en sitios distantes (MINTEL, 2013), que facilita la gestión, capacitación,

consulta bibliográfica, promoción y prevención, investigación e interculturalidad de la salud. Así, combinadas con los nuevos dispositivos y aplicaciones se pretende lograr el llamado e-Health de fácil acceso para las personas.

Por otra parte Ecuador tiene problemas en la prestación de servicios médicos, en particular en las unidades de salud que están distantes de los centros especializados (López, Chiriboga, & Carrera, 2016). Sin embargo el país forma parte de la Red Iberoamericana de Tecnologías Móviles en Salud (RITMOS), que busca impulsar el uso de dispositivos móviles en los servicios de salud y el uso de tecnología fundamentalmente en promoción y prevención (Mejía, 2017). Dentro de este campo múltiples son las aplicaciones de e-Health entre ellos están medición de ritmo cardiaco, temperatura corporal (Ibhaze, Eleanor, & Idachaba, 2016) , monitoreo remoto (Raj, Jain, & Arif, 2018), señal Electrocardiográfica ECG (Neyja et al., 2018), incluso entrega de turnos para consulta (López et al., 2016).

Como parte de estos desarrollos el grupo Bio4Tech de la Universidad de las Fuerzas Armadas-ESPE, en conjunto con otras Universidades y financiado por CEDIA, desarrollaron el proyecto “Evaluación del conocimiento sobre alimentación, ejercicio, redes sociales y desarrollo de prototipos para la toma de medicación y monitorización de signos vitales orientados al adulto mayor mediante la integración y el uso de las nuevas tecnologías de información y comunicaciones, aplicado a personas mayores de 65 años” (CEDIA, 2017) , donde se desarrolló un dispositivo para monitoreo de signos vitales usando módulos de hardware abierto, sin embargo el alcance de ese proyecto no incluía un sistema de monitoreo remoto y visualización de los datos obtenidos a través de Internet, por lo que realizar este proyecto es importante, con el fin de complementar la solución de monitoreo de signos vitales de personas, repotenciando el dispositivo desarrollado, reduciendo

tamaño y mejorando los medios para la transmisión de los datos recolectados por los sensores, con el fin de enviarlos inalámbricamente hacia los servidores de Base de Datos (BD) y Web, que posteriormente permitan realizar la consulta remota a través de aplicaciones multiplataforma (web y móvil), obteniendo un sistema de e-Health.

### **1.3 Alcance**

El presente proyecto de investigación se plantea el desarrollo de un sistema de monitoreo remoto y visualización para dispositivos de signos vitales orientados a e-Health.

Para ello se realizó la caracterización del funcionamiento del dispositivo de monitoreo de signos vitales del grupo Bio4Tech, con el fin de conocer cómo se adquieren los datos, la estructura del paquete de datos creado y la forma de transmisión inalámbrica del dispositivo; para luego repotenciarlo con la nueva tarjeta ESP32, que tiene los medios inalámbricos embebidos, que ofrece mayor estabilidad, procesamiento y reducción de tamaño.

Los datos monitoreados de la temperatura corporal, el ritmo cardíaco y el detector de caídas luego son enviados hacia el Broker (servidor) por medio de comunicación inalámbrica mediante el estándar IEEE 802.11, para almacenarlos en una Base de Datos. Así la información podrá ser desplegada y visualizada; teniendo acceso médicos, enfermeras y pacientes dependiendo de la política de acceso a través de una plataforma web; además se dispone de una aplicación móvil con Sistema Operativo Android, que muestra los datos monitoreados promedios; esta multiplataforma (web y móvil) implementa además una alarma en el caso que detecte una caída del paciente, por esta razón esta plataforma es de tipo multiacceso con usuario diferenciado, control de seguridad y reactiva a la alarma.



Finalmente se realizaron pruebas cuantitativas del funcionamiento del sistema integrado de Hardware y Software del dispositivo de monitoreo de signos vitales repotenciado, para determinar su funcionalidad, conectividad y la visualización de la información monitoreada en las aplicaciones web y móvil.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar un sistema de monitoreo remoto y visualización para dispositivos de análisis de signos vitales orientados a e-Health.

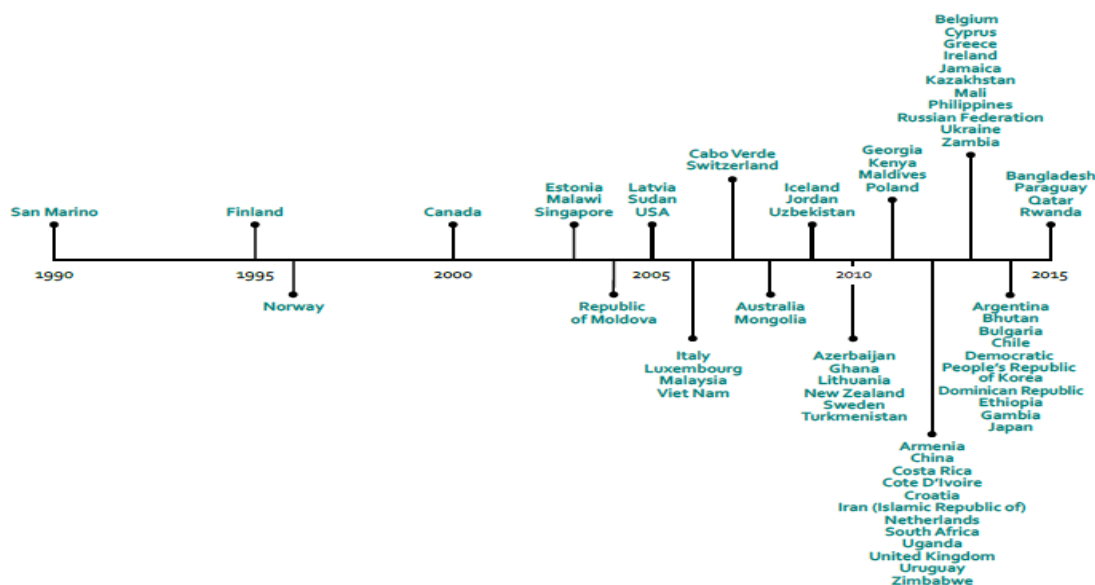
### **1.4.2 Objetivos Específicos**

- Analizar el estado del arte acerca de las aplicaciones de e-Health, estandarización e interoperabilidad de dispositivos.
- Caracterizar el funcionamiento del dispositivo de análisis de signos vitales repotenciado.
- Identificar el tipo de protocolo de IoT para la transmisión de información del dispositivo a Internet.
- Implementar una base de datos para el almacenamiento de los datos de los signos vitales del paciente.
- Desarrollar una aplicación multiplataforma (web y móvil) para la visualización de los datos monitoreados y comprobar su comunicación con el Bróker (Servidor).
- Desarrollar el protocolo de pruebas para el sistema integrado.
- Analizar los resultados obtenidos.

## CAPÍTULO 2

### ESTADO DEL ARTE

#### 2.1 E-Health



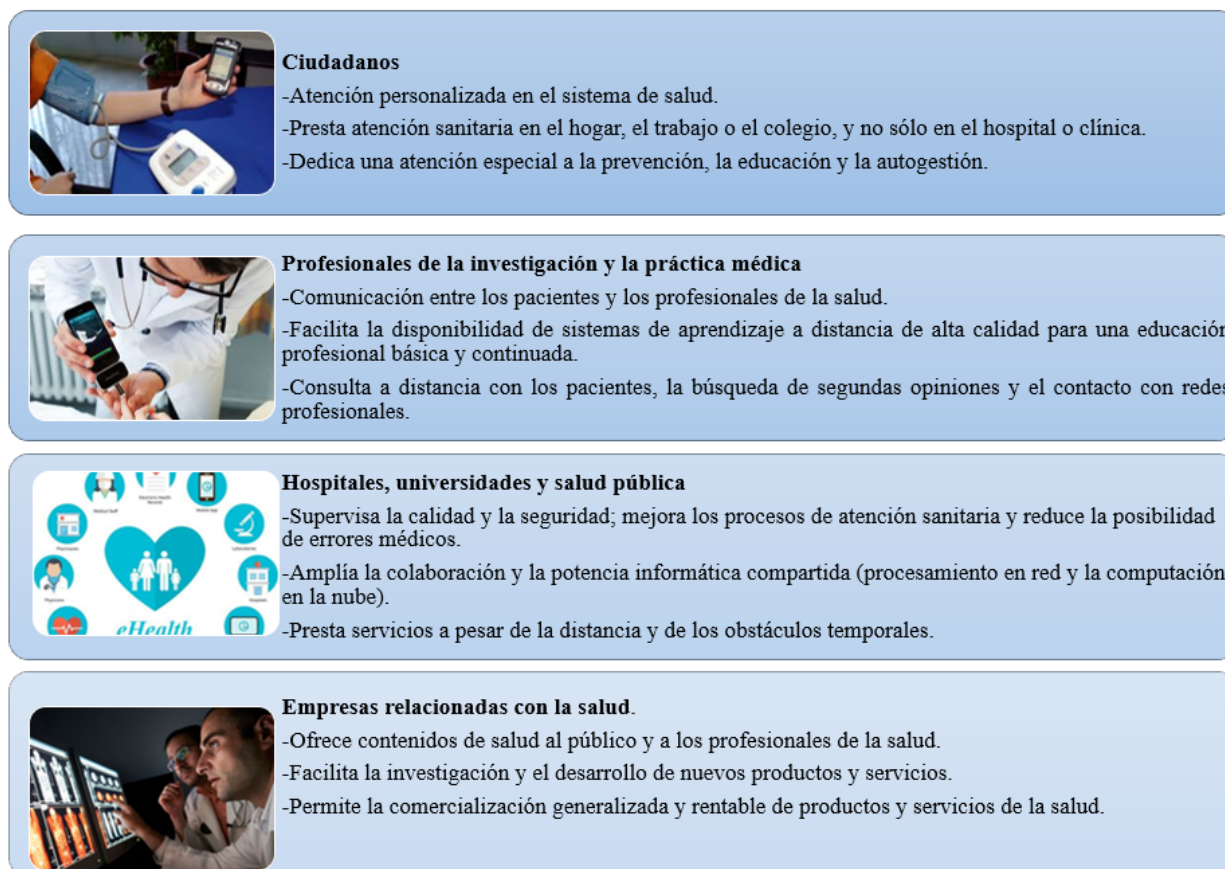
*Figura 1.* Cronología de políticas de e-Health en diferentes países.

Fuente: (World Health Organization, 2016)

Con el pasar de los años el término de e-Health ha tomado mayor relevancia; las personas de todas las edades requieren un monitoreo eficiente y continuo de su estado de salud (Imdoukh, Khalil, Rady, & Khanafer, 2017); pero todo este desarrollado ha ocurrido por cada uno de los avances que ha ocasionado la salud electrónica, en la Figura 1 se puede visualizar la cronología como e-Health ha tomado parte de los diferentes países.

##### 2.1.1 Repercusión de e-Health

La Figura 2 muestra algunos ejemplos de repercusión de e-Health, es decir los beneficios para las diferentes partes interesadas al aludir que el campo de la salud electrónica se expande a varios grupos de la sociedad.



**Figura 2.** Repercusiones de e-Health.

Fuente: (World Health Organization, 2016)(Organización Mundial de la Salud (OMS) y la Unión Internacional de Telecomunicaciones (UIT), 2012)

### 2.1.2 Sistemas de monitoreo remoto de e-Health.

La implementación de Sistemas de monitoreo cada día se convierte en indispensables por la diversidad de ventajas que ofrecen tanto a los pacientes como a los médicos, por lo que se presentan diferentes sistemas de monitoreo:

Chiuchisan & Geman (2016), propusieron un dispositivo que tiene por objetivo la monitorización remota de pacientes con obesidad mórbida entre 26 y 45 años, este dispositivo además presenta alerta temprana en el caso que los valores registrados tanto para el pulso y la presión arterial excedan los rangos máximos; al mismo tiempo se utilizó varios módulos dedicados a aplicaciones médicas.

De igual manera Hernández et al. (2017), presentaron un sistema e-Health para el monitoreo y control de temperatura y luminosidad de un prototipo de incubadora neonatal, a través de un sistema que permite la conexión con herramientas web, que a su vez, envían los datos a través de Internet, para observar constantemente las variables, generar alarmas y mantener informados a los profesionales de la salud.

También Noaman et al. (2018), plantearon un sistema de vigilancia de la salud en línea basado en el conocimiento para predecir infecciones adquiridas en el hospital, mediante un sistema web se puede recopilar los datos de pacientes, conjuntamente tiene un simulador incorporado para generar registros de datos; obteniendo toma de decisiones, aprobación de políticas y planes de vigilancia para reducir y prevenir las infecciones adquiridas en el hospital, sus resultados experimentales alcanzaron un valor de mejora igual al 87%.

Como se ha detallado se han desarrollado variedad de trabajos de e-Health y todos ellos ofrecen crear una nueva forma de adaptación de la medicina permitiendo una relación cercana entre el paciente y el médico; indistintamente que se encuentren en lugares diferentes, siendo una solución óptima y eficaz en el campo de la salud.

## **2.2 Internet de las Cosas (IoT)**

Desde que Kevin Ashton en 1999 acuñó el término de Internet de las Cosas (IoT) se afianzó un nuevo paradigma en el escenario de las comunicaciones inalámbricas (Rodríguez, Montenegro, & Cueva, 2015), siendo la interconexión de objetos a Internet. Originalmente, los datos en Internet fueron creados por personas, mientras que hoy en día la mayor cantidad de ellos son generados por objetos (Colucci, Poletti, Stefanelli, & Trincherro, 2017)

Además las tendencias de IoT incluye tres pasos: Inteligencia Embebida, que integra acciones que se desarrollan automáticamente; Conectividad, que todos dispositivos inteligentes estén conectados e Interacción, donde se crean nuevos objetos inteligentes que pueden procesar la información (Rodríguez et al., 2015).

IoT brinda una gran oportunidad para las aplicaciones en el sistema de atención médica, la combinación de sistemas integrados, sensores y actuadores asociados con el monitoreo constante de los pacientes logra un amplio alcance en el sistema de atención médica debido a la asequibilidad y la posibilidad de desarrollar sistemas compactos (Colucci et al., 2017).

Además el número de aplicaciones y servicios que pueden proporcionar es ilimitado y se puede adaptar a muchos campos de la actividad humana, facilitando y mejorando su calidad de vida en múltiples formas (Salazar & Silvestre, 2014). La Figura 3 indica la diversidad de aplicaciones de IoT.

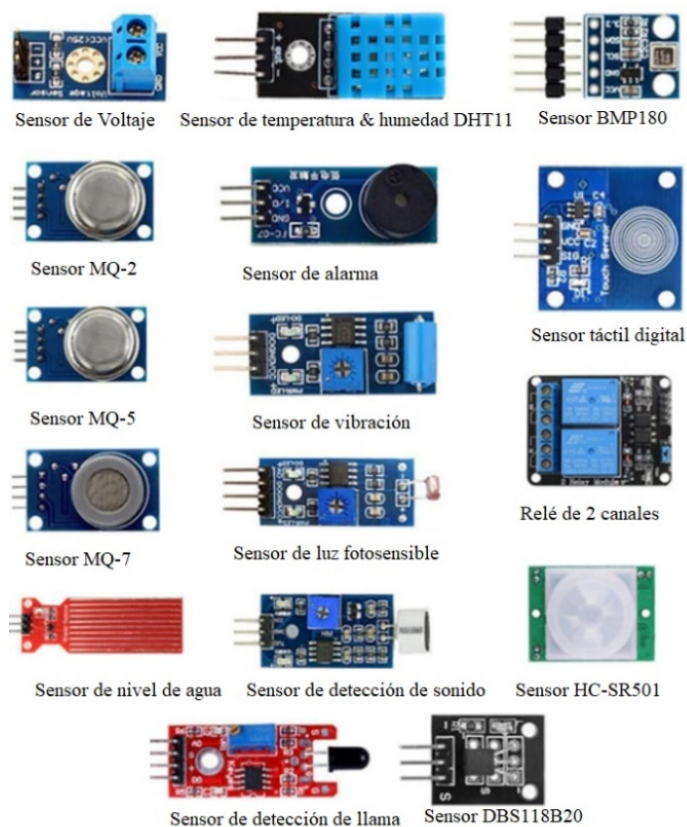


**Figura 3.** Aplicaciones de IoT.

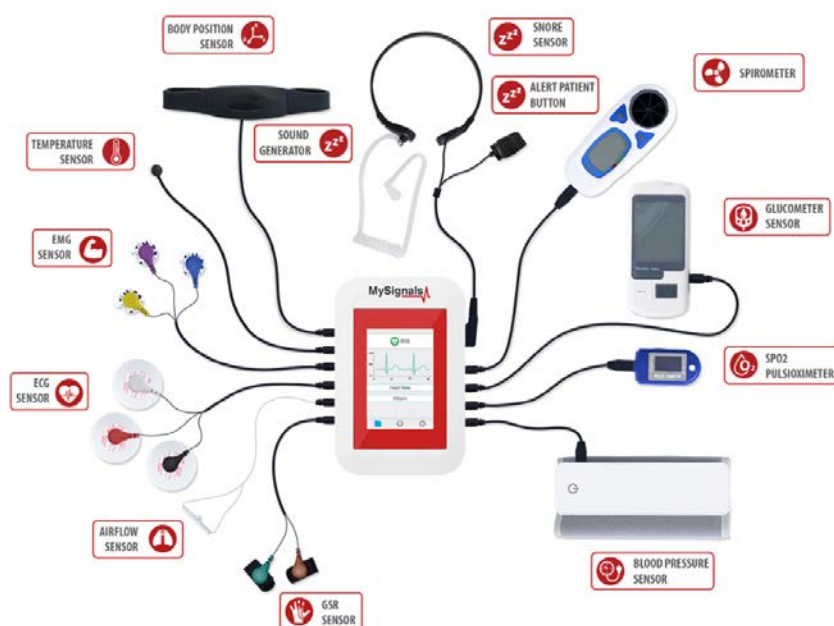
### 2.2.1 Dispositivos utilizados en IoT.

Los dispositivos utilizados en IoT poseen sensores y actuadores, que son indispensables porque facilitan que objetos de la vida cotidiana interactúen con los seres humanos a través de Internet o redes dedicadas, mediante la recopilación de información del entorno o interacción con él (González, 2017).

Es así que los sensores se han convertido en necesarios desde el campo de la salud hasta la construcción (Salazar & Silvestre, 2014), por esta razón existen sensores físicos que transforman una magnitud física en información, sensores químicos y bioquímicos que miden concentraciones de distintos elementos, y proveen a la salida de estos es una señal analógica proporcional a la medida, que debe ser adaptada mediante un circuito (González, 2017).



**Figura 4.** Sistema de SmarHome IoT.  
 (“Kit 16 sensores SmartHome Arduino (SKU 504G2) || HUBOT,” 2018)



**Figura 5.** Sensores de e-Health.

Fuente: (MySignals, 2016)

Los diferentes tipos de sensores para aplicaciones de IoT se pueden denominar a las piezas de hardware que hacen el trabajo de los procesos de monitoreo, mediciones y recolección de datos, para su elección se debe considerar: precisión, condiciones ambientales, alcance, calibración, costo, entre otras características. La Figura 4 muestra los diferentes tipos de sensores que se pueden utilizar para sistema de SmartHome, mientras que la Figura 5 muestra los diferentes sensores que se pueden utilizar para un dispositivo de e-Health.

Por otro lado, los actuadores también desempeñan un papel importante en el campo de IoT, estos trabajan con información digital para interactuar con el mundo exterior como motores, servomotores, motores paso a paso, electroválvulas, etc (González, 2017).

Actualmente por ejemplo un teléfono inteligente existe gran cantidad de sensores y actuadores incorporados (acelerómetro, magnetómetro, giroscopio, sensores de temperatura, GPS, entre

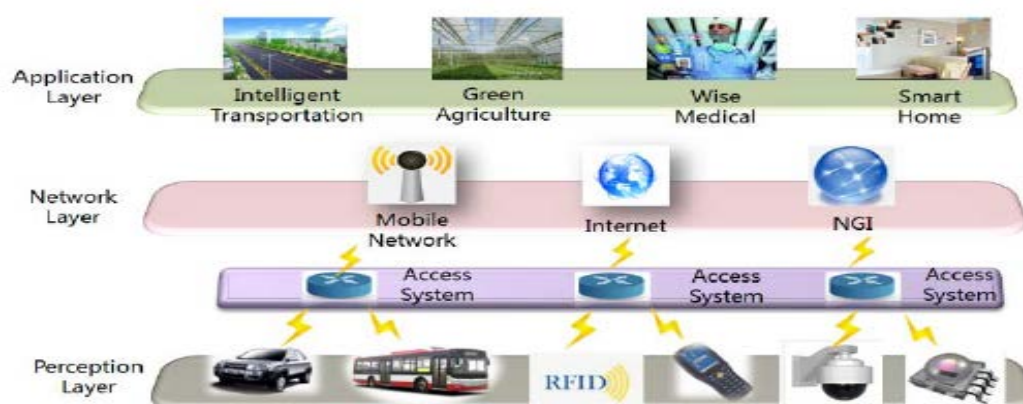
otros), que hacen posibles aplicaciones más inteligentes, siendo una herramienta útil para el reconocimiento de acciones humanas. (Bao, Ye, & Dou, 2016).

### 2.2.2 Arquitectura de IoT

La arquitectura popular de IoT se muestra en la Figura 6, y consta de tres capas: capa de percepción, capa de red y capa de aplicación (Wang, Hou, Gao, & Ji, 2017), además cumple un conjunto de principios:

- Escalabilidad, ampliación de capacidades y soporte de nuevo estándares.
- Ser horizontal, al lograr la integración de diferentes soluciones IoT.
- Gestión garantizada.
- Incluir la posibilidad de programar aplicaciones para los sistemas de IoT

(Benítez & Calderón, 2017).



**Figura 6.** Arquitectura básica de IoT.

Fuente: (Wang et al., 2017)

### Capa de percepción

Se encuentra en la base de la arquitectura y se encarga de varias funciones, relacionadas fundamentalmente con el medio físico en el cual se encuentra el dispositivo IoT, entre sus



principales funciones se encuentra: sensar, actuar, generar peticiones, responder peticiones, enviar reporte y modo de ahorro (Benítez & Calderón, 2017).

### Capa de red

Su función es conectar los sensores y servidores entre sí para transmitir y procesar los datos generados en la capa anterior de la arquitectura (González, 2017). Está compuesta por todas aquellas redes alámbricas e inalámbricas a través de las cuales se puede acceder a Internet, algunas de ellas como se muestran a continuación en:

- **Ethernet:** Basada en el estándar IEEE 802.3, está diseñado para ser implementado sobre distintos medios físicos con distintas velocidades de transmisión. Es posible encontrar en el mercado dispositivos diseñados para operar en este tipo de redes a velocidades de transmisión de 10, 100, 1000 Mbps y 10 Gbps (Barbancho, 2014) .
- **Wireless-Fidelity (WiFi):** Basada en el estándar IEEE 802.11, su objetivo fue crear una marca que permitiese fomentar la tecnología inalámbrica y asegurar la compatibilidad entre equipos. Existen varias versiones entre las más importantes:

IEEE 802.11b: Velocidades de 11 Mbps en la banda de 2.4 GHz.

IEEE 802.11a: Alcanza velocidades de 54bps en la banda de 2.4 GHz.

IEEE 802.11g: Velocidades de 54Mbps en la banda de 2.4GHz.

IEEE 802.11n: Velocidad real de transmisión de hasta 300 Mbps (límite teórico 600 Mbps), además puede trabajar en dos bandas (2.4GHz y 5GHz).

IEEE 802.11ad: Alcanza velocidades de hasta 6.7 Gbps, pero en la banda de 60 GHz.

IEEE 802.11ac: Permite velocidades de hasta 7 Gbps en la banda de 5 GHz.

(Barbancho, 2014)

- **Bluetooth:** Basada en el estándar IEEE 802.15.1, diseñada para corto alcance y tasas de transferencias hasta 1Mbps, en la frecuencia de 2.4GHz (González, 2017).
- **Zigbee:** Basada en el estándar IEEE 802.15.4 que define el nivel físico y el control de acceso a las redes de corto alcance Wireless Personal Area Networks (WPAN), además son de bajo costo y baja velocidad entre dispositivos, pero permiten mayor ancho de banda (González, 2017).
- **Worldwide Interoperability for Microwave Access (WiMAX):** Basada en el estándar IEEE 802.16, se trata de una especificación de redes de acceso metropolitanas inalámbricas, cuyo objetivo es promover la compatibilidad e interoperabilidad entre los dispositivos tanto en entornos fijos como móviles, además es posible disponer para coberturas de grandes distancias (Corral, 2016).
- **6LoWPAN:** Este estándar permite el uso de IPv6 sobre redes basadas en el estándar IEEE 802.15.4 y es una tecnología de bajo consumo de red (González, 2017).

### Capa de aplicación

Esta capa brinda servicios del contenido que se muestra a los usuarios (Malhotra, Bhattacharya, Kumar, Reddy, & Cao, 2011).

#### 2.2.3 Protocolos de Mensajería para IoT

- **Hyper Text Transport Protocol (HTTP)**

HTTP soporta solicitud / respuesta de arquitectura web RESTful. El servidor envía datos a

través del URI y el cliente recibe datos a través de un URI particular. Este un protocolo basado en texto y no define el tamaño de la carga útil del encabezado y el mensaje, sino que depende del servidor web o de la tecnología de programación, utiliza TCP como protocolo de transporte predeterminado y TLS / SSL para seguridad. Por lo tanto, la comunicación entre el cliente y el servidor es orientada a la conexión. HTTP es un estándar de mensajería web aceptado a nivel mundial que ofrece varias características, como conexiones persistentes, canalización de solicitudes y codificación de transferencia fragmentada (Naik, 2017). Cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

1. Un usuario accede a una URL.
2. El cliente Web descodifica la URL, identificando el protocolo de acceso, la dirección DNS o IP del servidor, el puerto opcional y el objeto requerido del servidor.
3. El Servidor inicia una conexión TCP/IP, llamando al puerto TCP correspondiente. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,...), la dirección del objeto requerido, la versión del protocolo HTTP empleada y un conjunto variable de información.
4. El servidor devuelve la respuesta al cliente.
5. Se cierra la conexión TCP (Networking and Emerging Optimization, 2018)

### **Desarrollo Web: Front-End y Back-End**

**Front-End:** Trabaja del lado Cliente, ocupa los componentes de la aplicación web, generalizándose en tres lenguajes, Html , CSS Y JavaScript.

**Back-End:** Trabaja del lado del servidor, es decir, esta persona se encarga de lenguajes como PHP, Python, .Net, Java, etc (campusMVP, 2018).

### **Java Messaging Service (JMS)**

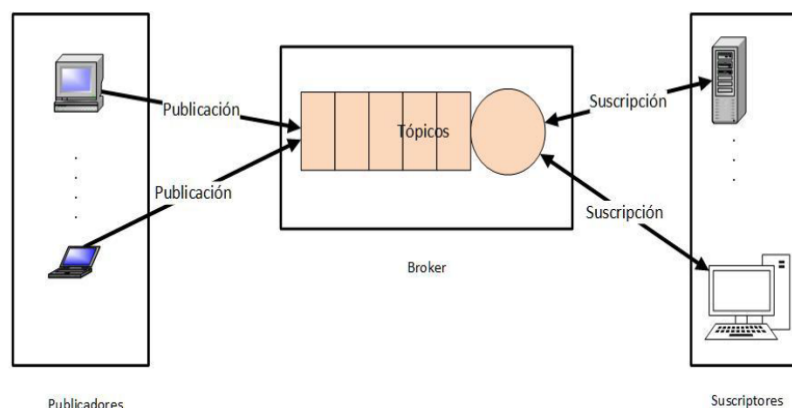
Es el sistema de mensajería de Java, proporciona un estándar entre plataformas para la comunicación asíncrona de sucesos y de datos. Los objetos de aplicaciones que normalmente se ejecutan en máquinas distintas se comunican accediendo a los servicios del middleware orientado a mensajes (MOM) mediante un conjunto estándar de interfaces que JMS define existente (Universidad de Alicante, 2014). Además, existen dos modelos de mensajería:

**Punto a Punto (PTP):** consta de dos clientes uno que envía el mensaje y otro que lo recibe (1:1). Se trata de un modelo de cola FIFO, en el cual el mensaje encolado será el primero en salir de la cola.

**Publicación/Subscripción:** consta de varios clientes (1:n). El destino de un mensaje se conoce como tópico. Los mensajes en los tópicos no se encolan. Un nuevo mensaje en el tópico sobrescribirá cualquier mensaje existente (Universidad de Alicante, 2014).

- **Message Queue Telemetry Transport (MQTT)**

Protocolo de mensajería de publicación/suscripción, se basa en la transmisión de mensajes entre diferentes dispositivos con un elemento central, el broker, que gestiona las comunicaciones de manera centralizada y garantiza que no existe pérdida de mensajes en la mayoría de veces, existe múltiples brokers compatibles con MQTT que se muestran en la , entre los principales Mosquitto, ActiveMQ o RabbitMQ (González, 2017). La Figura 7 muestra la arquitectura de MQTT.



**Figura 7.** Arquitectura de MQTT.

Fuente: (González, 2017)

**Tabla 1**

*Brokers compatibles con MQTT*

Broker	Descripción
Mosquitto	Broker de código abierto ligero. Probablemente el broker MQTT más popular.
Mosca	Se basa en Node.js que requiere la instalación de node.js.
ActiveMQ	Ofrece balanceo de carga de mensajes y alta disponibilidad para sus datos.
RabbitMQ	Permite reducir las cargas y los tiempos de entrega por parte de los servidores de aplicaciones web,
VerneMQ	Diseñado desde cero para funcionar como intermediario de mensajes distribuidos, que mantiene una baja latencia y tolerancia a fallas.
Kafka	Manejar fácilmente alta velocidad de los datos.

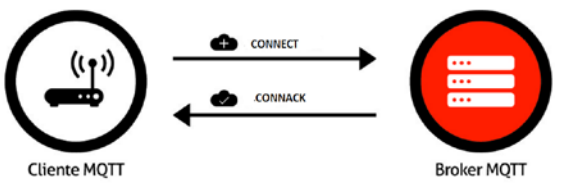
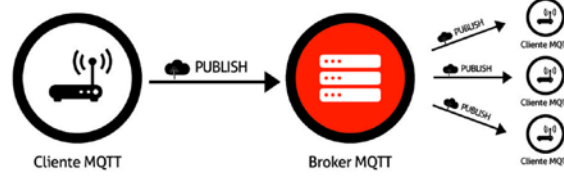
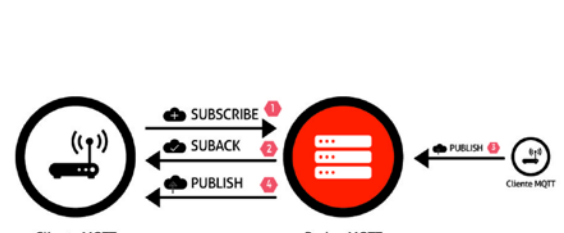
Fuente: (Cope, 2019)

El cliente MQTT publica mensajes en un broker, que están suscritos por otros clientes o pueden ser retenidos para la suscripción futura, cada mensaje se publica en una dirección, conocido como tópico, los clientes pueden suscribirse a múltiples tópicos y recibir cada mensaje publicado en cada tópico. El tópico técnicamente es la cola de mensajes, que permiten

el intercambio de mensajes bajo cierta estructura semántica. (Naik, 2017).

Además, MQTT es un protocolo de conexión que explica cómo los bytes de datos son organizados y transmitidos por la red TCP/IP. En la Tabla 2 se detalla los diferentes paquetes de control que posee MQTT para el envío de mensajes.

**Tabla 2**  
*Paquetes de control MQTT*

Paquetes de control	Proceso
 <p>Cliente MQTT</p> <p>Broker MQTT</p>	<p>El cliente se conecta el broker enviando un mensaje CONNECT que permite establecer la conexión y posteriormente el cliente recibe un mensaje CONNACK del broker.</p>
 <p>Cliente MQTT</p> <p>Broker MQTT</p> <p>Cliente MQTT</p> <p>Cliente MQTT</p> <p>Cliente MQTT</p>	<p>El cliente puede enviar mensajes PUBLISH al broker. Después, el broker remite el mensaje a todos los clientes que se suscriben a este tópico.</p>
 <p>Cliente MQTT</p> <p>Broker MQTT</p> <p>Cliente MQTT</p>	<p>El cliente podrá enviar uno o más mensajes SUSCRIBE al broker para indicar que recibirá mensajes de varios tópicos. Después que el cliente se haya suscrito a un tópico con éxito, el broker retornará un mensaje SUBACK. Según el mensaje SUSCRIBE, el cliente también podrá cancelar la suscripción mediante el mensaje UNSUBSCRIBE</p>

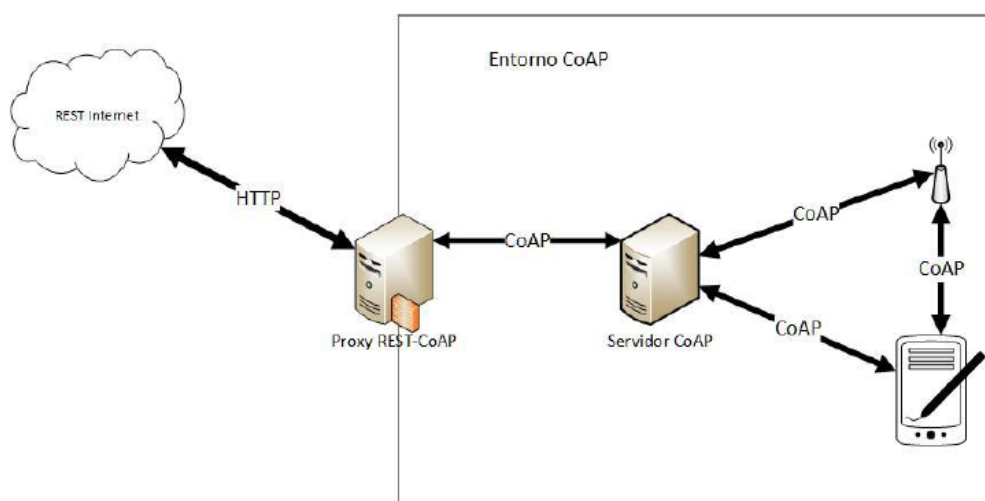
Fuente : (OASIS, 2014)

- **Constrained Application Protocol (CoAP)**

CoAP admite tanto la arquitectura de solicitud/respuesta como la de recurso/observación (una variante de publicación/suscripción), se desarrolla para interoperar con Hyper Text

Transport Protocol (HTTP) y la Web RESTful (Naik, 2017).

Se define como un protocolo de transferencia web que se basa en la representación del estado de la transferencia encima de las funcionalidades de HTTP. REST representa una forma simple de intercambiar datos entre clientes y servidores sobre HTTP, puede verse como un protocolo de conexión que depende de la arquitectura cliente-servidor (González, 2017); la Figura 8 muestra el entorno de desarrollo del protocolo CoAP.



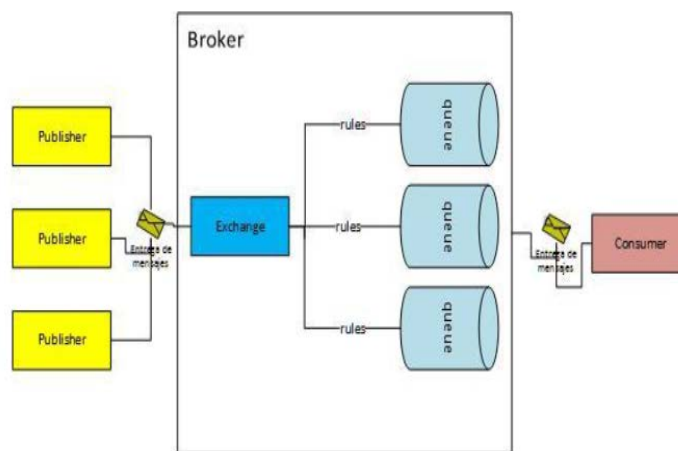
**Figura 8.**Entorno de CoAP.

Fuente: (González, 2017)

- **Advanced Message Queuing Protocol (AMQP)**

Protocolo de mensajería corporativa diseñado para la confiabilidad, la seguridad, el aprovisionamiento y la interoperabilidad, admite tanto la arquitectura de solicitud/respuesta como la de publicación/suscripción. Este sistema de comunicación AMQP requiere que el editor o el consumidor cree un "intercambio" con un nombre de pila y luego transmita ese nombre, los editores y consumidores usan el nombre de este intercambio para descubrirse. Posteriormente, un consumidor crea una "cola" y la conecta al intercambio al mismo tiempo.

Los mensajes recibidos por el intercambio deben coincidir con la cola a través de un proceso llamado "enlace". La Figura 9 muestra la arquitectura de este protocolo, además éste intercambia mensajes de varias maneras: directamente, en forma de fanout, por tópicos o en base a encabezados (Naik, 2017).



**Figura 9.** Arquitectura AMQP.

Fuente: (González, 2017)

- **Comparación entre los diferentes Protocolos de Comunicación**

La Tabla 3 presenta un análisis comparativo de los Protocolos de Mensajería para sistemas de IoT, se muestra diferentes características de cada uno.

**Tabla 3**

*Análisis comparativo de Protocolos de Mensajería para Sistemas IoT.*

Criterio	MQTT	CoAP	AMQP	HTTP
Año	1999	2010	2003	1997
Arquitectura	Cliente/Bróker	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor
		Cliente/Bróker	Cliente/Bróker	

CONTINÚA 



Abstracción	Publicar / Suscribir	Solicitar/respuesta	Publicar / Suscribir	Solicitar/respuesta
		Publicar / Suscribir	Solicitar/respuesta	
Tamaño de la cabecera	2 Byte	4 Byte	8 Byte	Indefinido
Tamaño del mensaje	Pequeño e indefinido	Pequeño e indefinido	Negociable e Indefinido	Grande e Indefinido
Semántica y métodos	Connect, Disconnect, Publish, Subscribe, Unsubscribe, Close	Get, Post, Put, Delete	Consume, Deliver, Publish, Get, Select, Ack, Delete, Nack, Recover, Reject, Open, Close	Get, Post, Head, Put, Patch, Options, Connect, Delete
Caché y Proxy	Parcial	Si	Si	Si
Calidad de Servicio (QoS) / Confiabilidad	QoS 0: a lo sumo una vez QoS 1: al menos una vez QoS 2: Exactamente una vez	Mensaje confirmado Mensaje no confirmado	Settle Format o Unsettle Format	Limitado (a través del transporte Protocolo - TCP)
Estándares	OASIS, Eclipse Foundations	IETF, Foundation	Eclipse OASIS, ISO/IEC	IETF and W3C
Protocolo de Transporte	TCP	UDP, SCTP	TCP, SCTP	TCP

Seguridad	TLS/SSL		DTLS, IPsec	TLS/SSL, IPsec, SASL	TLS/SSL
Default Port	1883/ (TLS/SSL)	8883	5683 (UDP Port)/ 5684 (DLTS)	5671 (TLS/SSL), 5672	80/ 443 (TLS/SSL)
Formato de codificación	Binario		Binario	Binario	Texto
Modelo de Licencia	Código abierto	Código Abierto	Código Abierto	Código Abierto	Libre

Fuente : (Naik, 2017)

## 2.2.4 Estándar de e-Health

Los estándares se los describe como herramientas que admiten mantener una conversación común, siendo la llave para la interoperabilidad, a continuación, se describe uno de los estándares utilizados para e-Health:

### IEEE 1073 Point of Care Medical Device Communication

Nivel OSI	ISO# 11073	IEEE# 1073	Contenidos (Fuentes relacionadas)
7	1xxxx	1.x.x	<b>MDDL – Medical Device Data Language</b> (related Vital-Intermed/ISO17109)
	10101	1.1.1	MDDL – Common Nomenclature (vital+intermed)
	10201	1.2.1	MDDL – Domain Information Model (DIM)
	103xx	1.3.x	Virtual Medical Device (VMD) specializations: 3.1 – Infusion device      3.9 – Airway flow 3.2 – Vital signs monitor    3.10 – Cardiac output 3.3 – Ventilator                3.11 – Capnometer <b>3.4 – Pulse oximeter</b> 3.12 – Hemodynamic 3.5 – Defibrillator <b>3.13 – Pulmonary</b> <b>3.6 – ECG</b> 3.14 – Respirator <b>3.7 – Blood Pressure</b> <b>3.15 – Weighing scale</b> 3.8 – Temperature
7-5	2xxxx	2.x.x	<b>MDAP - Medical Device Application Profiles</b> (related Intermed/1073.2/CEN1427)
	20101	2.1.1	MDAP - Base Standard
	20102	2.1.2	MDAP - MIB elements
	20201	2.2.1	MDAP - Polling Mode Profile
	20202	2.2.2	MDAP - Baseline Profile
	20301	2.3.1	MDAP - Optional Package, remote control
	20302	2.3.2	MDAP - Optional Package, symmetric commun.
4-1	3xxxx	3.x.x	<b>TPP - Transport &amp; Physical Profiles (common)</b>
	30100	3.1.x	TPP - Connection Mode (3.1a - Amendment 1)
	30200	3.2.x	TPP - IrDA Based.cable connected (3.2a - Amnd1)
	30300	3.3.x	TPP - Infrared wireless
1	4xxxx	4.x.x	<b>Physical Layer Interface Profiles</b>
3	5xxxx	5.x.x	<b>Internetworking Support</b>
4	6xxxx	6.x.x	<b>Application Gateways (related HL7 messages)</b>

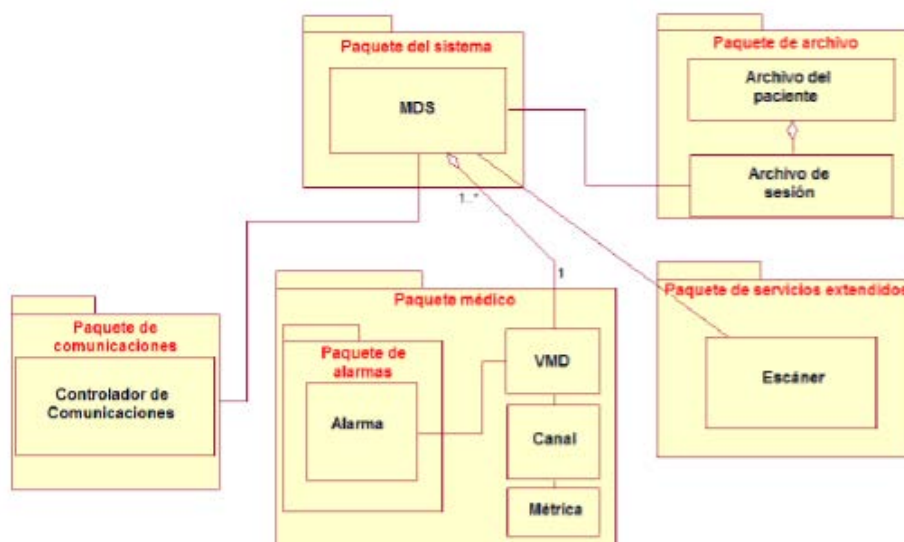
**Figura 10.** Pila de protocolos de IEEE 1073.

Fuente: (Galarraga et al., 2013)

Familia de estándares de comunicación que permite lograr interoperabilidad plug and play entre dispositivos médicos (reducido tamaño y capacidad de procesamiento) y sistemas de información en salud, dirigido a ambientes de atención aguda (Organización Panamericana de la Salud, 2016).

La Figura 10 muestra cómo posibilitar la comunicación siguiendo cada nivel de la pila de protocolos según el estándar.

**Niveles superiores:** Proporcionan definiciones para el formato de intercambio de información y representación de perfiles/componentes. Definen un Modelo de Información del Dominio (Domain Information Model, DIM) orientado a objetos, en que los objetos se categorizan en distintos paquetes (Galarraga et al., 2013). La Figura 11 presenta un diagrama general de paquetes para un sistema médico, según se define en el DIM.



**Figura 11.** Diagrama de paquetes según el modelo de información (DIM)

Fuente: (Martinez, 2015)

- **Paquete del sistema:** Representa a todo el sistema médico, es decir, los dispositivos, los eventos de alarma, las mediciones, los medios de almacenamiento de información y los actores del proceso de comunicación.

- **Paquete Médico:** Contiene a los objetos relacionados con el dispositivo, las medidas y los eventos de alarmas que se generan dentro de él, sus objetos a continuación se describen:

*Objeto VMD:* Representa a un subsistema médico que puede ser tipo Hardware y Software.

*Objeto Métricas:* Representa medidas tomadas por medio del dispositivo.

*Objeto Métricas:* Agrupa objetos de tipo métricas que se relacionan entre sí.

- **Paquete de Alarmas:** Contiene objetos que representan alarmas que ocurren dentro de un objeto VMD o dispositivo.
- **Paquete de Servicios Extendidos:** Estos son los encargados de identificar cuando se presentan sucesos anormales en otros objetos.
- **Paquete de Archivo:** Representa el medio de almacenamiento.
- **Paquete de Comunicaciones:** Involucra a todos los objetos que representan entidades participantes un proceso de comunicación en el cual se intercambia información sobre señales médicas (Martinez, 2015)

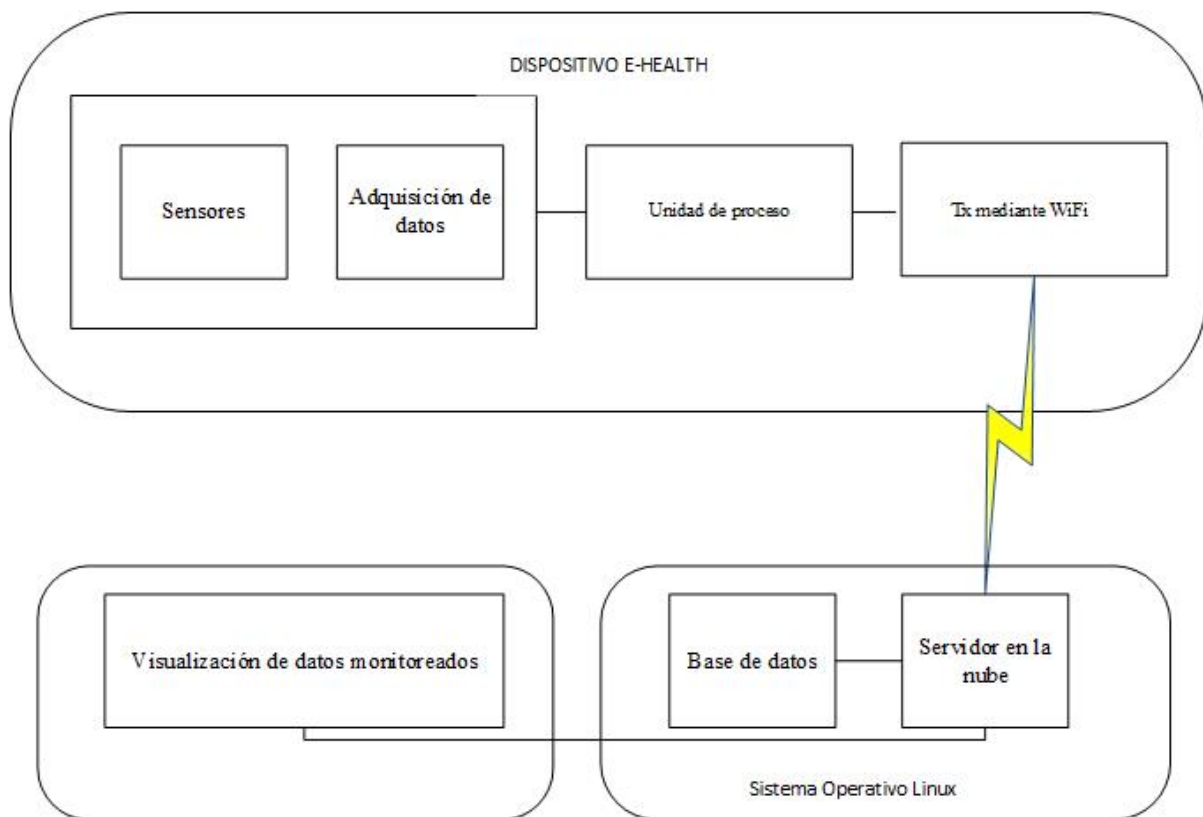
**Niveles intermedios:** Definen los perfiles de aplicación para un VMD (Medical Device Application Profiles, MDAP) distinguiendo entre modo de transmisión básico (baseline) y sondeo (polling) (Galarraga et al., 2013).

**Niveles inferiores:** Especifican los protocolos y servicios para las conexiones y mensajes de transporte usando los estándares internacionales existentes, donde sea posible (Galarraga et al., 2013).

## CAPÍTULO 3

### DISEÑO E IMPLEMENTACIÓN

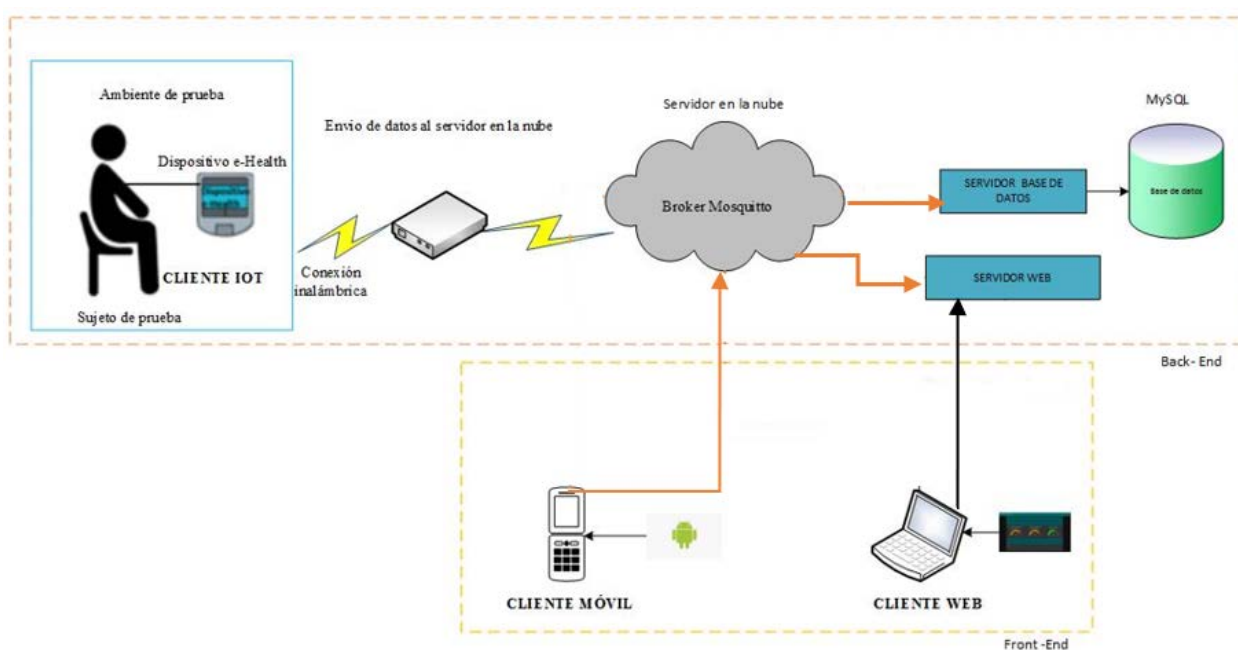
En este capítulo se presenta el diseño e implementación del Sistema de monitoreo remoto y visualización para dispositivos de signos vitales orientados a e-Health. Los datos recolectados de los signos vitales se envían a un Broker (servidor en la nube), se almacena en una Base de Datos y se presentan en un sistema de visualización multiplataforma (web y móvil), que permita observar la información a los múltiples usuarios (médicos, enfermeras y pacientes). En la Figura 12, se muestra un diagrama de bloque del sistema de monitoreo y visualización.



*Figura 12.* Diagrama de bloques general.

### 3.1 Requerimientos de hardware y software

El proyecto de investigación consta de dos partes principales que son el hardware y software, por lo tanto, se debe considerar los requerimientos necesarios para su implementación. En la Figura 13 se muestra un diagrama de red del Sistema de monitoreo, que consta de un Cliente IoT que es el dispositivo e-Health que se encuentra en el Back-End, el cliente web y móvil permite la visualización de los datos monitoreados que forman parte del Front-End.



*Figura 13.* Diagrama de red del Sistema de monitoreo.

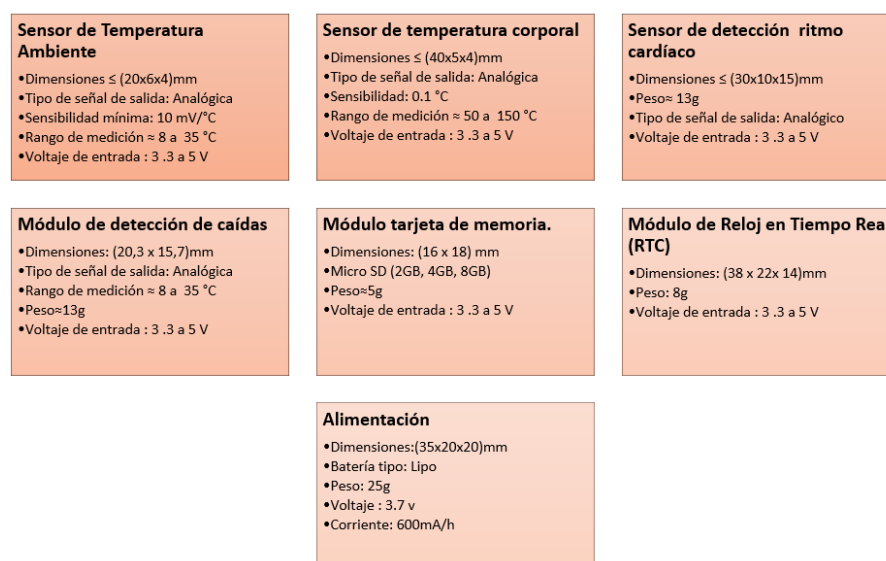
#### 3.1.1 Requerimientos de Hardware

En esta sección se describe el hardware con las características necesarias para el diseño e implementación del dispositivo de e-Health que busca repotenciar el hardware de la placa anterior que incluye las siguientes modificaciones:

- Cambio de tarjeta de desarrollo.
- Reubicación del sensor de temperatura ambiente.
- Cambio del sensor de temperatura corporal.

- Incluir módulo de memoria y reloj.
- Eliminar módulo de bluetooth.
- Módulo WiFi incluido en la tarjeta de adquisición.
- Reducir el tamaño y consumo eléctrico.
- Incluir LEDS indicadores.

La Figura 14 muestra los requerimientos de los elementos que debe contar el dispositivo de e-Health para obtener el funcionamiento propuesto y lograr las modificaciones.



**Figura 14.** Requisitos del diseño del hardware

### 3.1.2 Requerimientos de Software

En esta sección se detallan los componentes lógicos indispensables para la implementación del sistema de monitoreo, visualización y almacenamiento de los datos. A continuación, se detallan los requerimientos funcionales y no funcionales del sistema.

### **3.1.2.1 Requisitos funcionales**

Este tipo de requisitos especifican lo que debe hacer o los servicios que proporciona el sistema, en este proyecto de investigación se considera: aplicación web, aplicación móvil y la base de datos. Además, la instalación del servidor en la nube.

#### **a. Aplicación web**

En la aplicación web se puede visualizar los datos monitoreados como temperatura ambiente, temperatura corporal, ritmo cardíaco y detector de caídas.

##### **Inicio de sesión:**

- El usuario deberá iniciar la página web, se solicita elegir el tipo de usuario (doctor, enfermera o paciente) y posteriormente se requiere el usuario y contraseña.

##### **Ejecución de la aplicación web:**

- Ingreso de datos personales del paciente (nombre, apellido, cédula, edad, sexo).
- Consulta de datos monitoreados a través del atributo cédula, filtrar por fecha y se puede descargar un archivo con extensión .xls
- Mostrar los datos.

##### **Cierre de sesión:**

- El usuario puede cerrar la sesión a través de un botón.

#### **b. Aplicación móvil**

La aplicación móvil permite a los usuarios visualizar los datos monitoreados del dispositivo de e-Health en un smartphone.

##### **Inicio de sesión:**

- El usuario deberá iniciar la aplicación móvil, se solicita elegir el tipo de usuario (doctor,



enfermera o paciente) y posteriormente se requiere el usuario y contraseña.

### **Ejecución de la aplicación móvil**

- Conexión y desconexión al servidor en la nube.

### **Cierre de sesión:**

- El usuario puede cerrar la sesión a través de un botón.

### **c. Base de datos**

En la base de datos se almacenan los datos monitoreados y se desarrolla lo siguiente:

- Definir las tablas de almacenamiento.
- Creación de la tabla de almacenamiento de los datos.
- Recepción de los datos monitoreados desde el broker (servidor), con su respectiva fecha y hora.
- Almacenamiento de variables (temperatura ambiente, temperatura corporal, ritmo cardiaco, alarmas de caídas del paciente, temperatura corporal, contador de alarmas de detección de caídas, contador de alarmas de temperatura corporal, promedio de temperatura ambiente y temperatura corporal).

#### **3.1.2.2 Requisitos no funcionales**

Este tipo de requisitos se relaciona directamente a los aspectos visibles para los usuarios (médicos, enfermeras y pacientes). A continuación, se detalla lo que se consideró para el diseño del sistema:

- Aplicación web de fácil uso para el usuario.
- Aplicación web con fondo de color negro para no causar molestias en la vista del usuario.

- Aplicación móvil intuitiva y amigable.
- Visualización fácil de entender.

## 3.2 Diseño del Hardware

### 3.2.1 Módulos de adquisición de datos

Tomando en consideración los requerimientos del hardware de la sección 3.1.1 se ha elegido los siguientes sensores que permitirán la repotenciación del dispositivo de e-Health. Las características de cada de uno se detallan en tablas que se indican posteriormente.

#### 3.2.1.1 Sensor de temperatura ambiente

Se debe considerar la temperatura ambiente de la Provincia de Pichincha, que según los reportes de Instituto Nacional de Meteorología e Hidrología (INAMHI) varía entre 5.0 y 29.2 °C, por esta razón se ha elegido el sensor de temperatura LM35 que cumple con los requerimientos, además es uno de los sensores más adquiridos en el mercado, ya que tiene un rango de funcionamiento amplio y sensibilidad dentro de las temperaturas que pueden presentarse en el ambiente de prueba. La Tabla 4 presenta las características de dicho sensor.

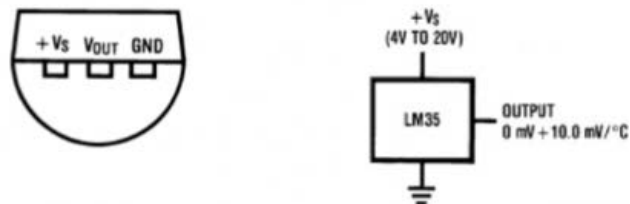
**Tabla 4**  
*Características Sensor de temperatura LM35*

Sensor	Características	
	Precisión	0.5 °C
	Rango de medición	-55 a 150 °C
	Sensibilidad	10 mV/°C
	Voltaje de entrada	3.3 a 30 V
	Consumo de corriente	Inferior 60 µA
	Impedancia de salida	0.1 Ω a 1 mA

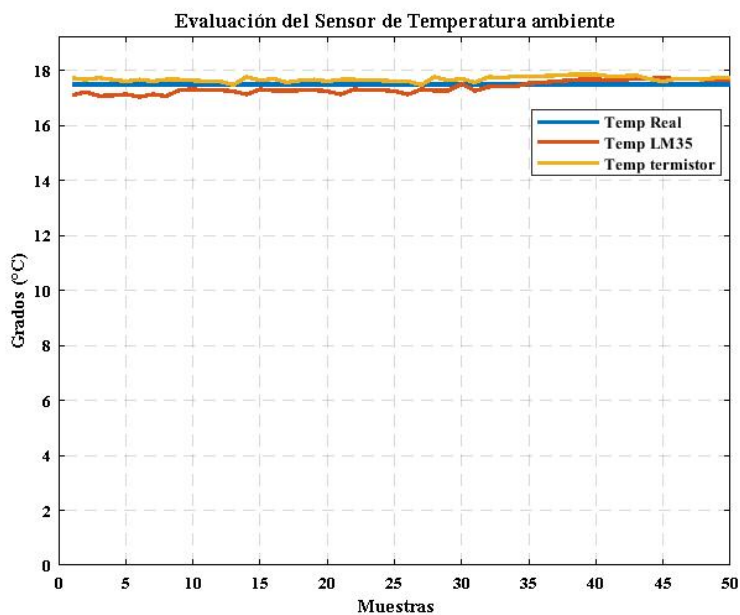
Fuente: (Instrument, 2016)

## Caracterización

La Figura 15 presenta el diagrama de conexión del sensor LM35, utilizando ese diagrama se realizó una prueba preliminar en un ambiente controlado, al interior, obteniendo 50 muestras durante una hora. Se ha utilizado como referencia un termómetro de mercurio y el termistor de serie NXFT15XH103FA2B100 para comparar resultados y obtener el posible error. La Figura 16 muestra los datos obtenidos de la temperatura ambiente, identificando que se mantiene un valor estable aproximadamente de 17.5 °C en el tiempo de duración de la prueba.



*Figura 15.* Diagrama de conexión del sensor LM35



*Figura 16.* Evaluación del sensor LM35

La Tabla 5 muestra los resultados obtenidos de la comparación de los sensores, identificando que existe un error entre el termómetro de mercurio y el sensor LM35 de 0.68%, mientras que el resultado obtenido con el Termistor es de 1.08% esto puede ser compensado en la implementación.


**Tabla 5**  
*Valores promedios del sensor de temperatura ambiente*

	Referencia	LM35	Error	Termistor	Error
<b>Promedio de muestras</b>	17.5°C	17.38°C	0.68%	17.69°C	1.08%

### 3.2.1.2 Sensor de temperatura corporal

La temperatura corporal es un factor que determina la regulación térmica del cuerpo humano que puede presentar cambios de acuerdo con el sexo, factores climatológicos, edad, entre otros. Por esta razón es dispensable que el dispositivo de e-Health disponga de un sensor que determine la temperatura en la que se encuentra el paciente, se considera valores normales de temperatura corporal entre 35°C y 37°C caso contrario la temperatura del paciente es anormal.

**Tabla 6**  
*Características Termistor NXFT15H103FA2B100*

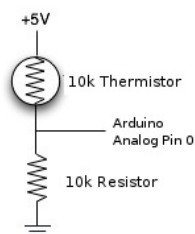
Sensor	Características	
	Precisión	±1%
	Rango de medición	-40 a 125 °C
	Potencia máxima	100 mW
	Voltaje de entrada máximo	5 V
	Corriente máxima	0.1 mA
	Sensibilidad	0.1°C
	Impedancia (25°C)	10KΩ

Fuente: (Manufacturing Co, 2018)

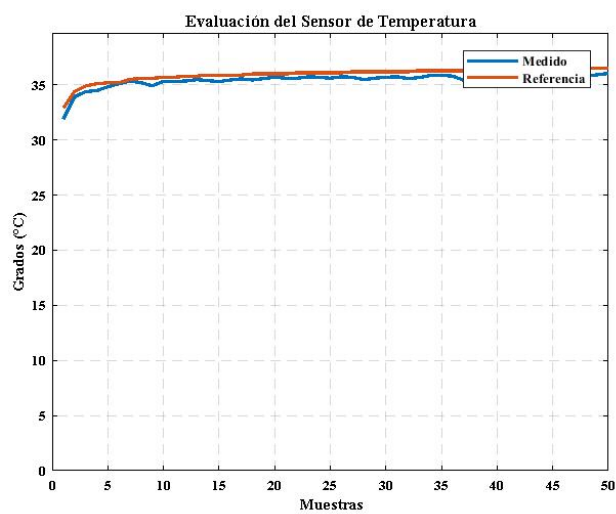
Es indispensable que el sensor a utilizarse cumpla con los requerimientos propuestos, posea una respuesta rápida a los cambios de temperatura y que esté disponible en el mercado. En base a ello se eligió el termistor de serie NXFT15XH103FA2B100, este sensor está formado por un encapsulado y con dos terminales, en la Tabla 6 se muestran sus principales características.

### Caracterización

La Figura 17 presenta el diagrama de conexión del sensor. Una vez implementado se realizó la evaluación preliminar para medir la temperatura axilar en un sujeto, la prueba se realizó durante media hora y se obtuvo 50 muestras. Como referencia se utilizó un termómetro clínico digital flexible para comparar resultados y obtener el posible error. En la Figura 18 se observa los datos obtenidos, el valor de la medición indicó una temperatura promedio de 35.5°C aproximadamente.



**Figura 17.**Diagrama de conexión del sensor NTC



**Figura 18.**Evaluación del sensor NTC

La Tabla 7 muestra los resultados obtenidos identificando que existe una variación entre el termómetro digital y el termistor de 1.39%, el cual puede ser compensado por software.

**Tabla 7**

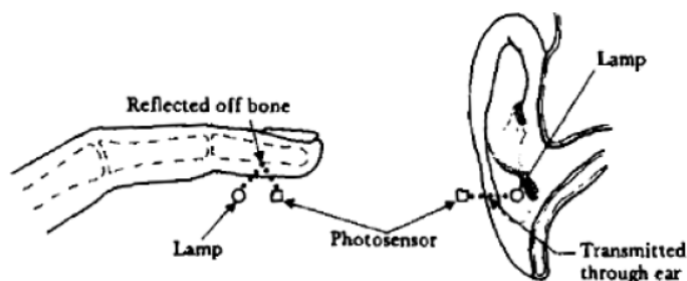
*Valores promedios del sensor de temperatura corporal*

	Referencia	Termistor	Error
Promedio de muestras	35.96 °C	35.46°C	1.39%

### 3.2.1.3 Sensor de ritmo cardiaco

La medición del ritmo cardiaco es indispensable en el dispositivo de e-Health, de esta manera se puede identificar el número de pulsaciones por minuto. Los valores considerados dentro del rango normal son de 60 a 100 latidos por minuto, estos valores pueden variar considerando la edad, sexo, actividad física, entre otros factores del paciente.

Basado en los requerimientos, para obtener los datos se utilizó el sensor de pulso HRM-2511B, se ha elegido este sensor porque posee dos modos de medida, se puede colocar en el dedo (modo de reflexión) o en la oreja (modo de transmisión) como se muestra en la Figura 19.



**Figura 19.** Posición de ubicación del sensor de ritmo cardiaco.


El principio de funcionamiento de este sensor se basa en la absorción de la luz infrarroja por parte de la sangre. Durante la fase de sístole se concentra mayor cantidad de sangre en los vasos sanguíneos permitiendo así una mayor absorción de luz, mientras que durante en la diástole la

cantidad de sangre disminuirá (Langereis, 2010). En la Tabla 8 se muestra las características básicas del sensor.

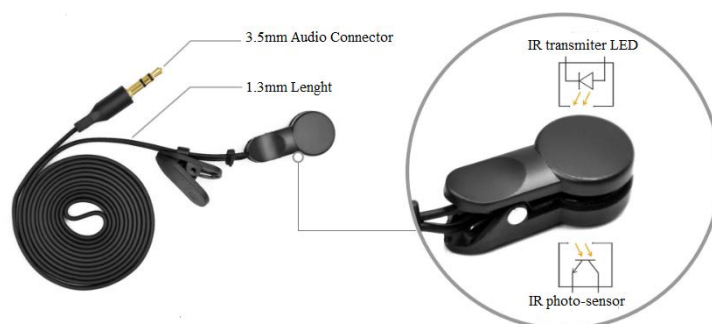
La Figura 20 muestra la configuración interna del sensor de ritmo cardiaco, este tipo de sensor posee un conector de tipo Jack de 3.5mm.

**Tabla 8**

*Características Sensor HRM-2511B*

Sensor	Características	
	Voltaje de entrada	3- 5 V
	Consumo de corriente	65 mA
	Rango de temperatura de funcionamiento	-50°C a 85°C

Fuente: (Langereis, 2010)

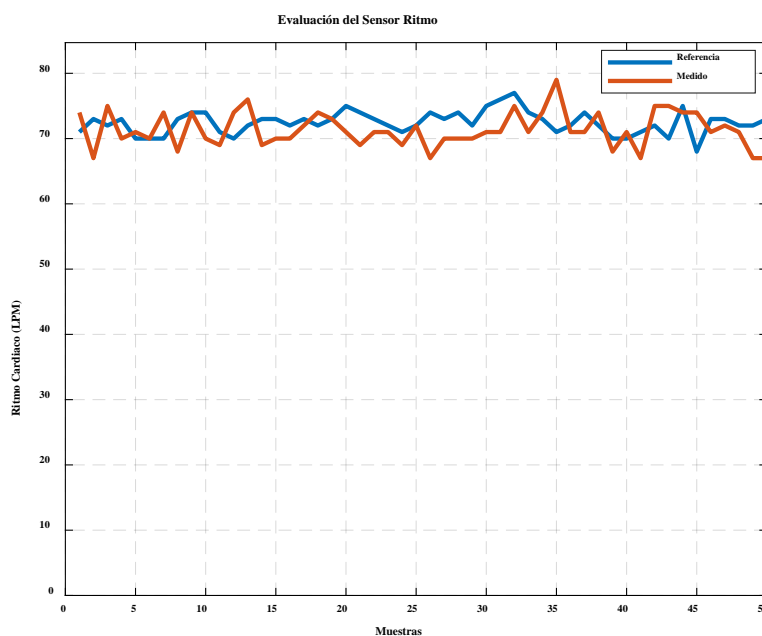


**Figura 20.** Configuración interna del sensor HRM-2511B.  
Fuente: (“Ear clip infrared heart rate pulse sensor - KYTO2511B,” 2018)

### Caracterización

Se realizó la prueba preliminar en una persona de 25 años de género masculino, durante la prueba se obtuvieron un total de 50 muestras de pulso cardiaco. Se ha utilizado como referencia un oxímetro de pulso para comparar los resultados.

La Figura 21 muestra que aproximadamente el ritmo cardiaco varía entre 72 Latidos por Minuto (LMP), existen algunos datos inferiores de los 70 LMP o superiores, esto se debe a que el sujeto de prueba adopta algún cambio repentino en su posición provocando falsos positivos. Por esta razón es indispensable que el sujeto no adopte cambios bruscos durante la toma de datos.



**Figura 21.** Evaluación del sensor HRM-2511B.

La Tabla 9 muestra los resultados obtenidos, donde se toma como referencia al oxímetro y se compara con lo medido encontrando un error de 2.77% que puede ser compensado para disminuirlo posteriormente.

**Tabla 9**

*Valores promedios del sensor de ritmo cardiaco*

	Referencia	Sensor HRM-2511B	Error
Promedio de muestras	70 LPM	72 LPM	2.77%




### 3.2.1.4 Módulo detector de caídas.

El detector de caídas en el dispositivo de e-Health, permite identificar si el paciente ha sufrido una caída; esto permite notificar a las personas a cargo y que se tome una rápida respuesta para ayuda del paciente. En la Tabla 10 se muestra las características principales del sensor ADXL335, el cual fue elegido por su alta sensibilidad a los cambios de posición de los tres ejes.

**Tabla 10**

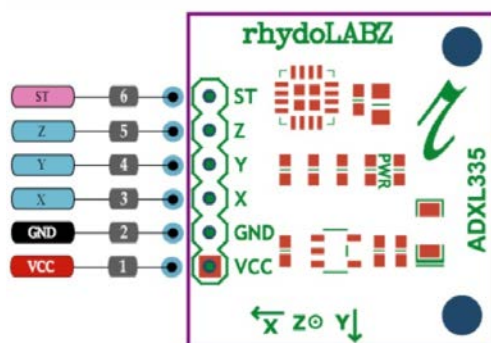
*Características del módulo ADXL335*

Sensor	Características	
	Voltaje de entrada	3.3 a 5 V
	Consumo de corriente	350 uA
	Sensibilidad	Alta
	Grados de libertad	3 ejes

Fuente: (Naylamp Mechatronics, 2018)

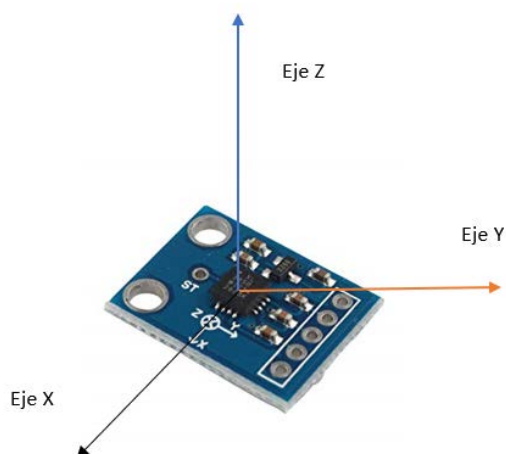
### Caracterización

La Figura 22 muestra el diagrama de conexión del sensor que permite detectar las caídas, en función de la variación de los ejes X e Z, la Figura 23 muestra dichos ejes, la evaluación se realizó durante 2 minutos con la obtención de 50 muestras.



**Figura 22.** Diagrama de conexión del sensor ADLX335.

La Figura 24 (a) muestra los datos obtenidos correspondientes a los tres ejes (X, Y, Z), sin variar la posición del sensor y se visualiza que no existe ningún cambio. La Figura 24 (b) muestra la variación de la posición del sensor, donde se ha simulado caídas, el primer cambio de posición se realizó un movimiento frontal, el siguiente cambio hacia la derecha y el último hacia la izquierda.



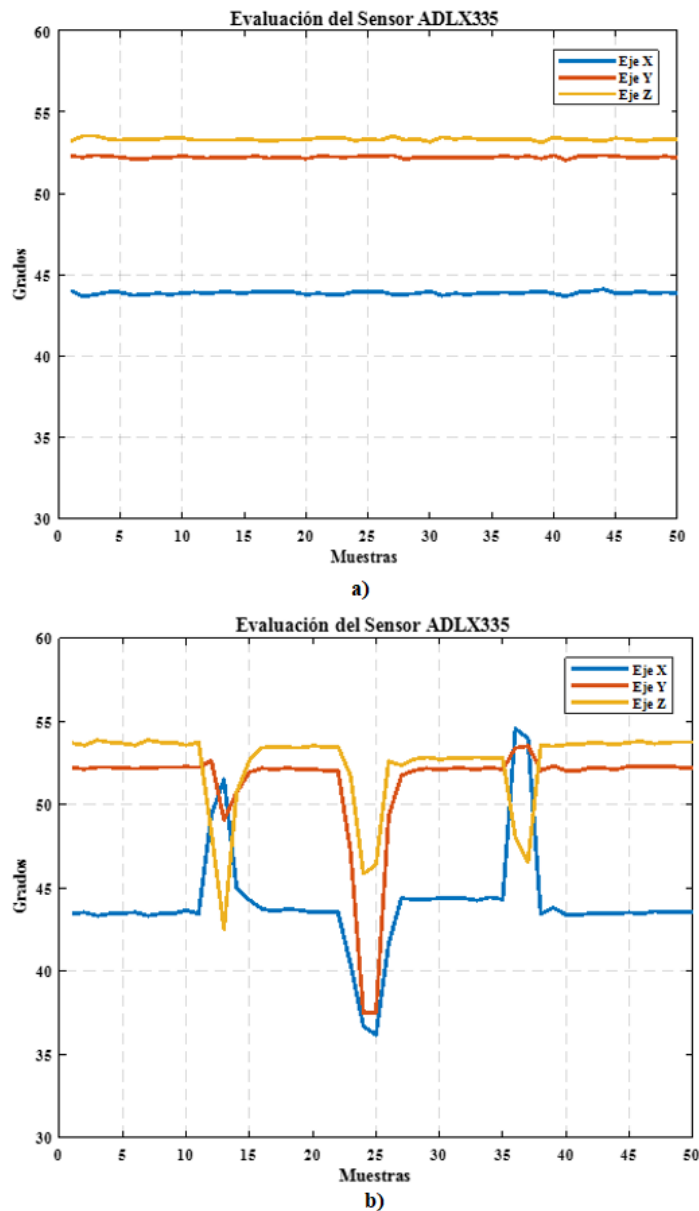
**Figura 23.** Posición de los ejes del acelerómetro.

Debido a la posición del sensor detector de caídas (ver Figura 23) los ejes que tienen una mayor variación son el eje X y Z porque el acelerómetro se encuentra perpendicular al eje horizontal, esto se aplica para caídas frontales y hacia los lados (derecha o izquierda). En la Tabla 11 se observa el cambio de los ejes con los diferentes movimientos que se utilizaron al simular las caídas del sujeto de prueba.

**Tabla 11**

*Cambio de posición de los ejes*

Movimiento	Eje X (Grados)	Eje Y (Grados)	Eje Z (Grados)
Reposo	44	52	53
Frontal	51	49	48
Derecha	36	37	46
Izquierda	54	53	47



*Figura 24.* Evaluación del sensor ADXL335.

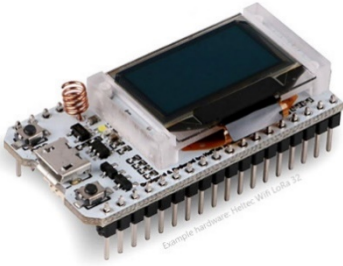
## 3.2.2 Repotenciación de la placa

### 3.2.2.1 Tarjeta ESP32

Para la repotenciación de la placa, se propuso la tarjeta ESP32 que tiene incorporado una pantalla de Diodo Orgánico de Emisión de Luz (OLED), bastante utilizada en proyectos de IoT.

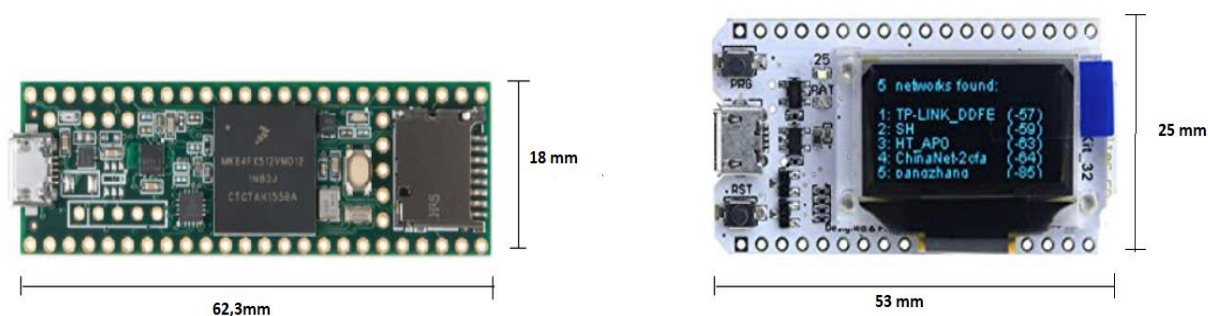
La ESP32 permite la conectividad inalámbrica mediante el estándar 802.11 b/g/n mediante un módulo incluido y además posee varios pines de conexión analógicos. En la Tabla 12 se muestra las características de la tarjeta ESP 32.

**Tabla 12**  
*Características de la tarjeta ESP32*

ESP 32	Características		
	Procesador	Xtensa 32-Bit LX6 Dual Core	
	Memoria ROM	448 KB	
	Memoria SRAM	540 Kb	
	Memoria flash externa	32-Bit de acceso	
	Voltaje de alimentación	2.7 a 3.6 V	
	Consumo de corriente	80 mA(típica) – 500 mA (máxima)	
	WiFi	802.11 b / g / n (2.4 a 2.5 GHz)	
	Interfaces	UART, SPI, SDIO I2C, I2S, IR, PWM	
		LED y PWM motor.	
	Tipos GPIO	Digital IO (36), ADC 12 Bits (18 canales), DAC 8-Bits (2 canales), Sensor capacitivo (10 canales)	
Pantalla OLED, dimensiones		190 mm x 273 mm	
Pantalla OLED, resolución	128 x 64 píxeles		

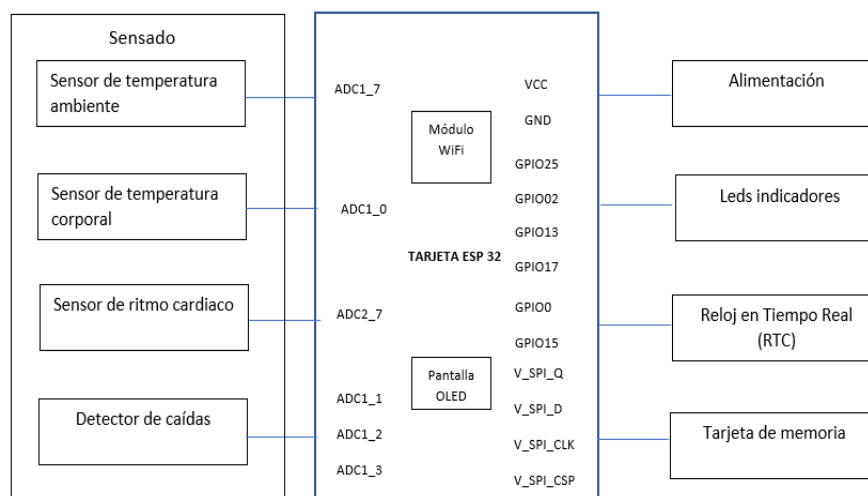
Fuente: (“Placa WiFi LoRa 32 - ESP32 / LoRa / Display OLED | Curto Circuito,” 2016)

La placa del dispositivo e-Health desarrollada por el grupo de investigación Bio4Tech utiliza originalmente la tarjeta Teensy, mientras que el dispositivo repotenciado utiliza la tarjeta ESP 32 con pantalla OLED incluida, con el fin de reducir el tamaño del PCB. La Figura 25 muestra las dos tarjetas de adquisición de datos comparando las dimensiones entre ellas.



**Figura 25.** Comparación de la tarjeta Teensy 3.6 con ESP32.

La Figura 26, muestra la nueva distribución en el diagrama de bloques para el dispositivo de e-Health repotenciado que usa la tarjeta ESP 32. Además, en la Tabla 13 se indica la conexión de los pines perteneciente a cada módulo.



**Figura 26.** Diagrama de bloques del dispositivo de e-Health

**Tabla 13**

*Pines de conexión de los módulos con la tarjeta ESP32*

Módulos	Pines de conexión
Sensor de temperatura ambiente (LM35)	ADC1_7

*CONTINÚA*



Sensor de temperatura Ambiente (Termistor)	ADC1_0
Sensor de ritmo cardiaco (HRM-2511B)	ADC2_7
Detector de caídas (Acelerómetro ADLX335)	ADC1_1 ADC1_2 ADC1_3
Leds indicadores:	GPIO25 GPIO2 GPIO13 GPIO17
Tarjeta de memoria	V_SPI_Q V_SPI_D V_SPI_CLK V_SPI_CSO
Reloj de Tiempo Real (RTC)	GPIO0 GPIO15

### 3.2.2.2 Modificaciones de la placa

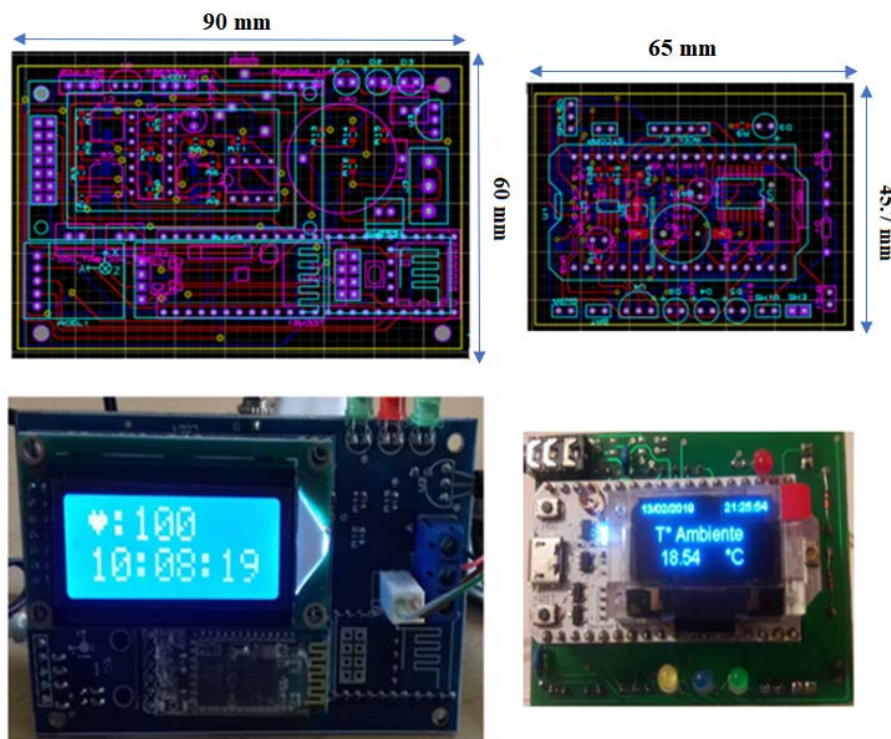
En la Tabla 14 se realiza una comparación entre el dispositivo anterior y el actual repotenciado. Se debe mencionar que se agregó una tarjeta de memoria para almacenamiento de datos y un Reloj en Tiempo Real (RTC) para la hora y fecha, debido a que este tipo de módulos no están incluidos dentro del hardware de la ESP32.

**Tabla 14***Comparación entre el dispositivo con tarjeta Teensy y ESP32*

Elementos del dispositivo	Dispositivo con la tarjeta Teensy 3.6	Dispositivo con la tarjeta ESP32 con OLED	Observación
Sensor de temperatura ambiente.	✓	✓	Cambio de posición del sensor
Sensor de temperatura corporal.	✓	✓	Se ha cambiado el sensor Ds18b20 por el sensor NTC.
Sensor de ritmo cardiaco.	✓	✓	Se ha modificado el circuito de detección del ritmo cardiaco por el recomendado en el datasheet (Langereis, 2010).
Detector de caídas.	✓	✓	No se ha realizado ningún cambio.
RTC	✓	✓	Módulo externo
Tarjeta de memoria	✓	✓	Módulo externo
Módulo de WiFi	✓	✓	Incluido en la tarjeta ESP 32
Módulo de bluetooth	✓	x	No se incluye el módulo.
Pantalla LCD	✓	✓	Incluida en la tarjeta ESP 32
Led indicador de alarmas	x	✓	Led de color rojo
Led indicador de conexión a WiFi	x	✓	Led de color verde
Led indicador de batería baja	x	✓	Led de color amarillo
Led indicador de lectura de ritmo cardiaco	x	✓	Led de color azul
Interruptor encendido/apagado	x	✓	Encendido/ apagado
Pulsador para actualización de fecha y hora	x	✓	En caso de que la fecha y la hora no sea la correcta.

### 3.2.2.3 Dimensiones de la placa

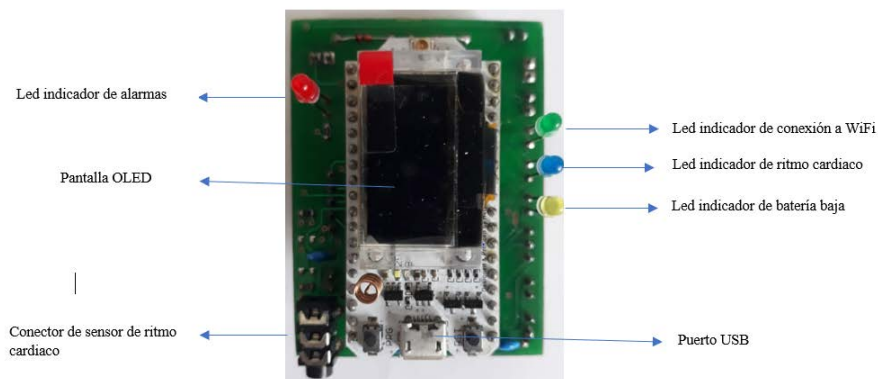
En la Figura 27 se presenta el diseño PCB realizado en el software de diseño electrónico Proteus, se muestra una comparación de dimensiones entre la placa desarrollado por el grupo Bio4Tech (a) que tiene dimensiones de 90 mm x 60 mm y la placa repotenciada (b) de dimensiones 45.7 mm x 65 mm, con esto se aprecia que se ha reducido el tamaño del dispositivo notablemente. Además, se muestra ensamblados todos sus componentes.



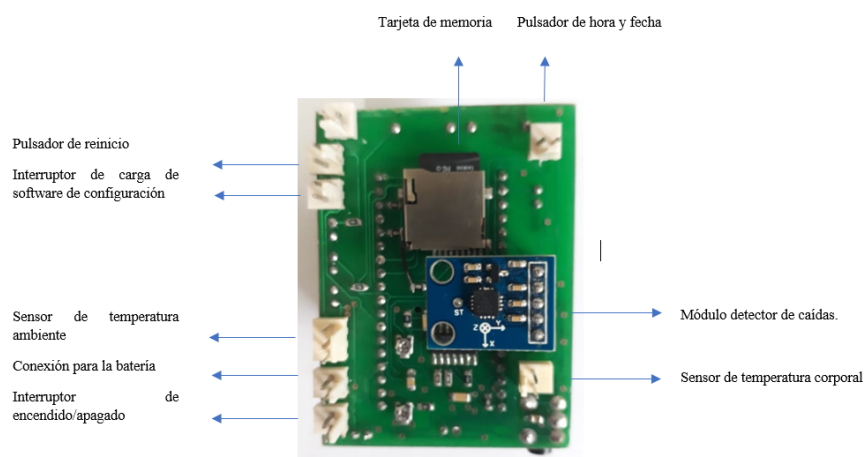
*Figura 27.* Comparación de dimensiones entre placas

La placa del dispositivo de e-Health fue realizado a doble cara para distribuir cada uno de sus componentes en ambos lados logrando reducir el tamaño, las Figuras 28 y 29 muestran los componentes.





**Figura 28.** Dispositivo visto desde el lado 1



**Figura 29.** Dispositivo visto desde el lado 2

### 3.2.2.4 Peso del dispositivo e-Health

Uno de los requerimientos de diseño era la reducción del peso, al cambiar de placa se logra este objetivo. La Tabla 15 muestra el peso de la placa anterior y la placa actual verificando que se ha logrado la reducción de 44.7 gramos aproximadamente de la placa anterior.




**Tabla 15**  
*Peso de la placa anterior y actual*

	Placa anterior	Placa actual
Peso de la placa	77.7 gramos	33 gramos

Para protección y manejo del dispositivo se ha incluido una carcasa externa de plástico con lo que el peso final del dispositivo es de 114.2 gramos siendo menor al prototipo anterior. En la Tabla 16 se indica cada una de las partes de la estructura del dispositivo.

**Tabla 16**

*Estructura plástica del dispositivo e-Health.*

	Vistas	Contenido
	Vista superior	Posee una cubierta, que contiene el acceso a los leds indicadores y la pantalla OLED
	Vista lateral 1	Salida del sensor de temperatura corporal.
	Vista lateral 2	Salida del sensor de temperatura ambiente, el interruptor de encendido/ apagado y el pulsador de reset.

*CONTINÚA*





Vista lateral 3

Salida del sensor de ritmo cardiaco, la conexión del puerto USB y el interruptor de carga de software de configuración.



Vista lateral 4

Pulsador de actualización de fecha y hora.



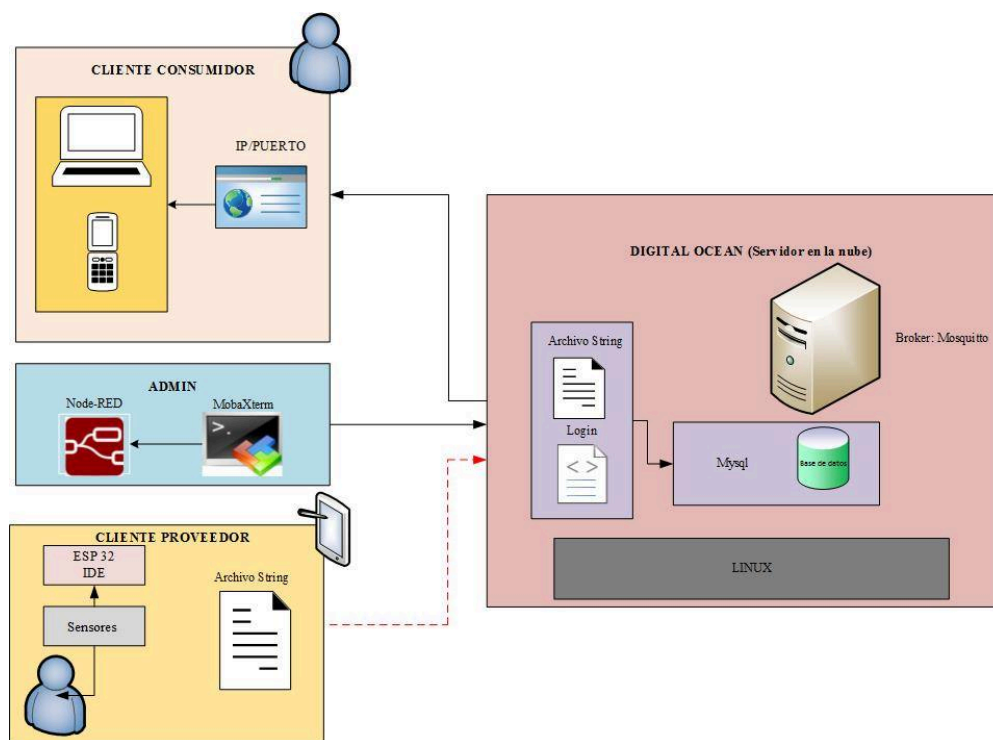
Vista posterior

Se creado está cubierta para acceder a la tarjeta de memoria que posee el dispositivo, esta debe ser retirada para extraer la tarjeta.

### 3.3 Diseño del software

El diseño del software incluye la instalación del servidor en la nube y el protocolo de mensajería, además un programa desarrollado en el IDE de Arduino y la implementación multiplataforma (web, móvil) y una Base de Datos.

La Figura 30 muestra un esquema general de las tecnologías utilizadas para el desarrollo del Sistema de monitoreo remoto y visualización de datos para el dispositivo de e-Health.



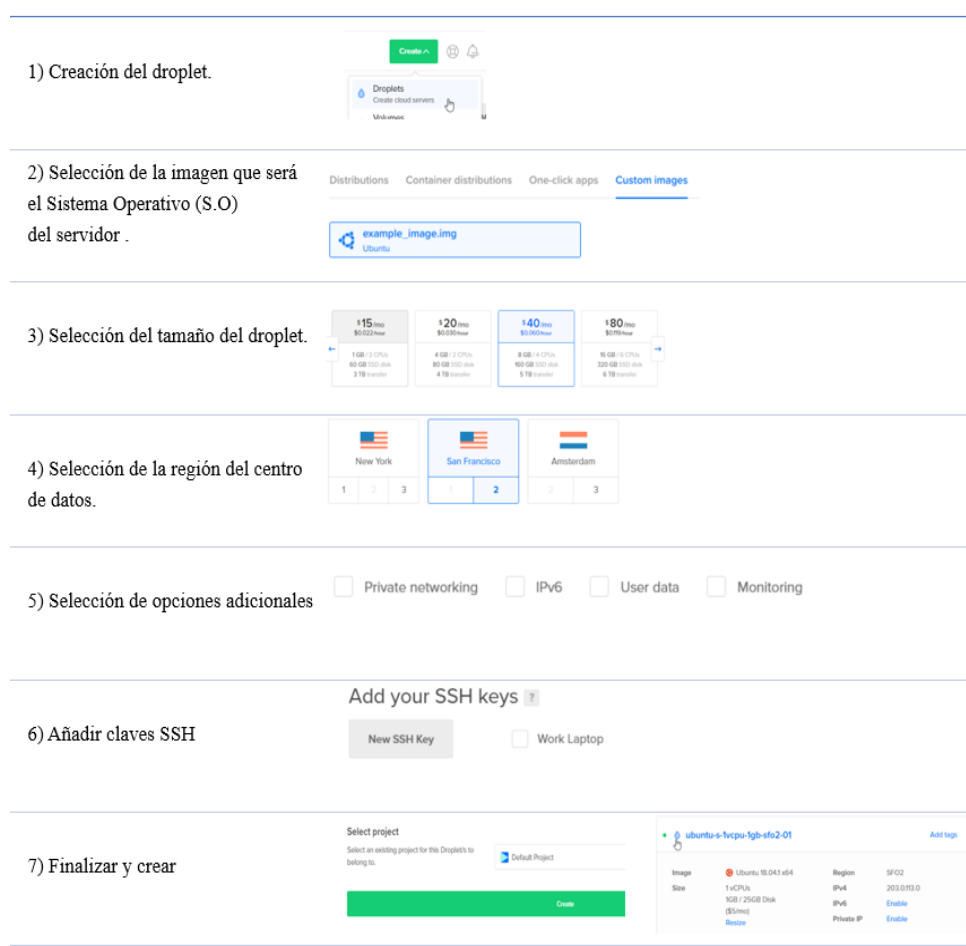
**Figura 30.** Esquema general de tecnologías usadas.

### 3.3.1 Proceso de adquisición e instalación del servidor en la nube.

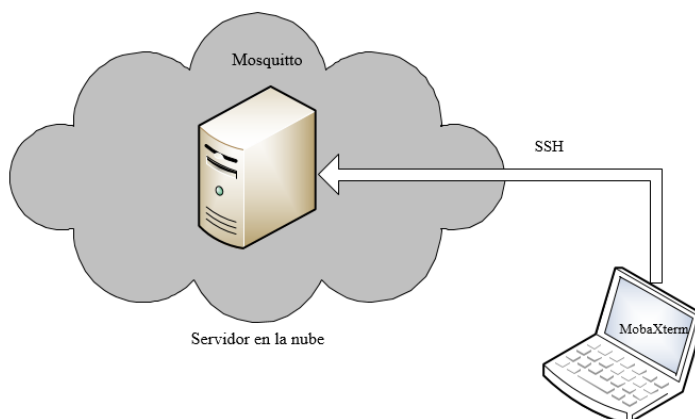
Un servicio en la nube proporciona procesos tecnológicos (publicidad, acceso a aplicaciones, seguridad, infraestructura, gestión, entre otros) a través de una red, por esta razón se ha elegido a Digital Ocean, el cual es un proveedor de servidores virtuales privados que permite la creación de máquinas virtuales conocidas como droplet, totalmente escalable de acuerdo con las necesidades requeridas como aumentar el espacio de almacenamiento sin problema y obtener un alto nivel de seguridad. En Digital Ocean existe la opción del Sistema Operativo Linux en las siguientes distribuciones: Ubuntu, CentOS, Debian, Fedora, CoreOS o FreeBSD, en el desarrollo del sistema se eligió el Sistema Operativo Ubuntu. En la Figura 31 se muestra el proceso de instalación del droplet, sin embargo previamente se debe crear una cuenta ingresando a la siguiente dirección web: <https://www.digitalocean.com/> para poder usarla.

Una vez que se creó el droplet y a través de MobaXterm, que es una herramienta de red de acceso remoto (SSH, X11, RDP, VNC, FTP, MOSH) que permite comandos de Unix (bash, ls, cat, sed, grep, awk, rsync) en el escritorio de Windows, se realiza el proceso mostrado en las Figuras 32 y 33 para el ingreso a la herramienta y el proceso para acceder al droplet respectivamente.

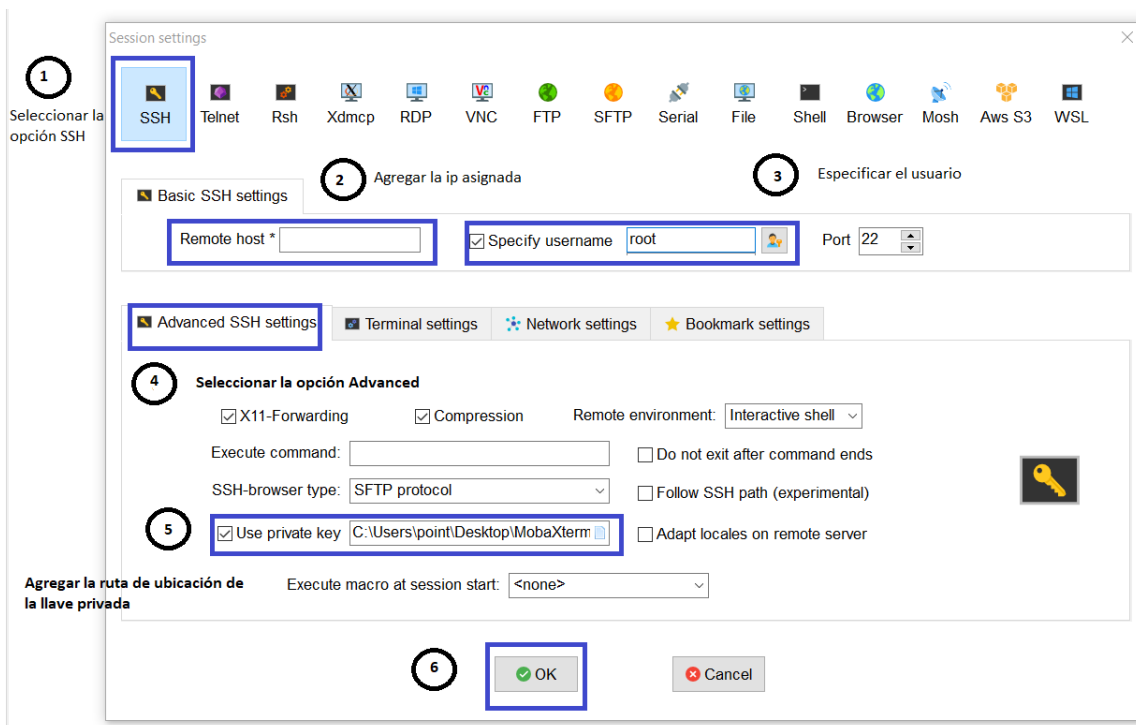
La Figura 34 indica el proceso de creación de la llave privada, esta llave se debe agregar en el paso 5 de la Figura 33 y es indispensable en el proceso de inicio de sesión del droplet caso contrario no se puede acceder al mismo.



**Figura 31.** Creación del servidor en la nube

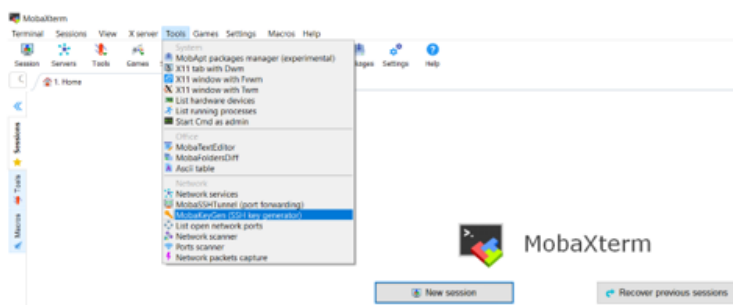


**Figura 32.** Herramienta MobaXterm

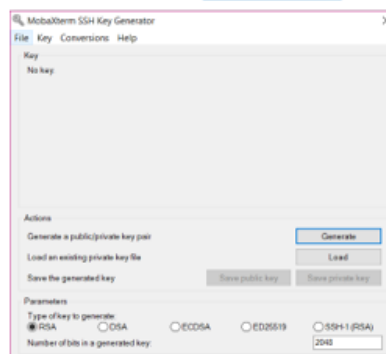


**Figura 33.** Pasos de inicio de sesión del droplet.

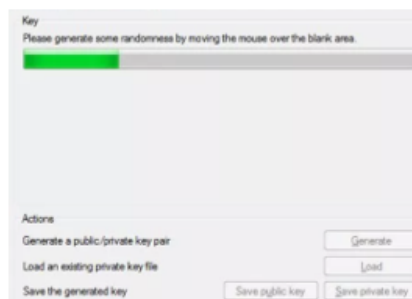
1) Crear una llave SSH en MobaXterm a través de la pestaña Herramientas MobaKeyGen ->



2) Elegir las características de la llave



3) Generar la llave



4) Guardar o copiar la llave en la ubicación que se desee.

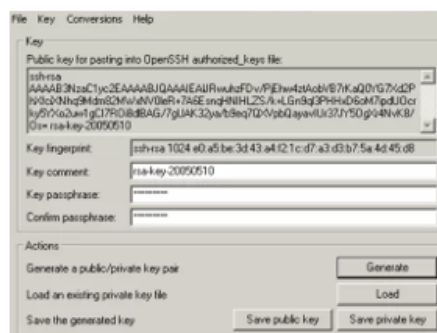
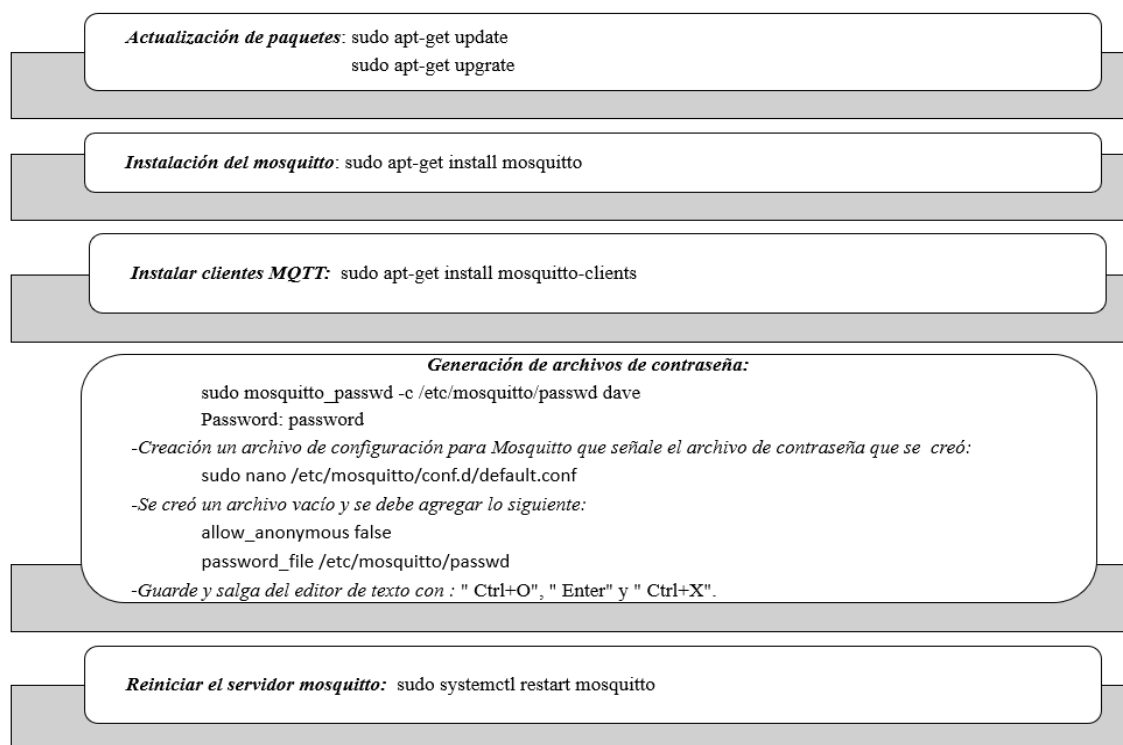


Figura 34. Proceso de creación de la llave privada

### 3.3.2 Protocolo de mensajería: MQTT

El protocolo que se eligió para la implementación es “Message Queue Telemetry Transport” (MQTT) porque permite el envío de mensajes cortos que utiliza un ancho de banda mínimo, es

ideal para en redes inalámbricas, consume poca energía y no requiere grandes recursos como memoria. Dentro de varias soluciones, uno de los brokers de mensajes MQTT es Mosquitto siendo un software de código abierto, de sencilla instalación y configuración. Mosquitto es el encargado de gestionar la red y de transmitir los mensajes, su instalación se muestra en la Figura 35.



**Figura 35.** Instalación del mosquitto.

### 3.3.3 Proceso instalación de Node-RED

Node-RED es un motor de flujos con enfoque IoT para protocolos estándares como MQTT, AMQP, entre otros. Es una herramienta visual, programada en Node.js, que proporciona un editor de flujo con una amplia gama de nodos. Node.js es una forma de ejecutar JavaScript en el servidor, y está diseñado para generar aplicaciones web (Node-RED, 2016). Se ha elegido esta herramienta por su gran variedad de nodos con diferentes funcionalidades listos para configurarse con lo que



desea el programador y formar los diferentes flujos. Además, la Tabla 17 muestra una comparación entre los diferentes framework.

**Tabla 17**  
*Comparación entre framework*

Framework	Descripción
Node-RED	Es una herramienta visual de código abierto, que está compuesta de nodos que permiten hacer una tarea concreta, como recibir una llamada HTTP, un mensaje MQTT o la activación de un pulsador.
Spring Boot	Permite el desarrollo de aplicaciones, de código abierto para la plataforma Java. Su trabajo es instanciar, inicializar y conectar objetos de la aplicación.
Ballerina	Es un enfoque para simplificar la integración y la programación de microservicios mediante el diseño conjunto de un lenguaje y una plataforma que sea ágil y simple a la integración.

Fuente: (InfoQ, 2018)

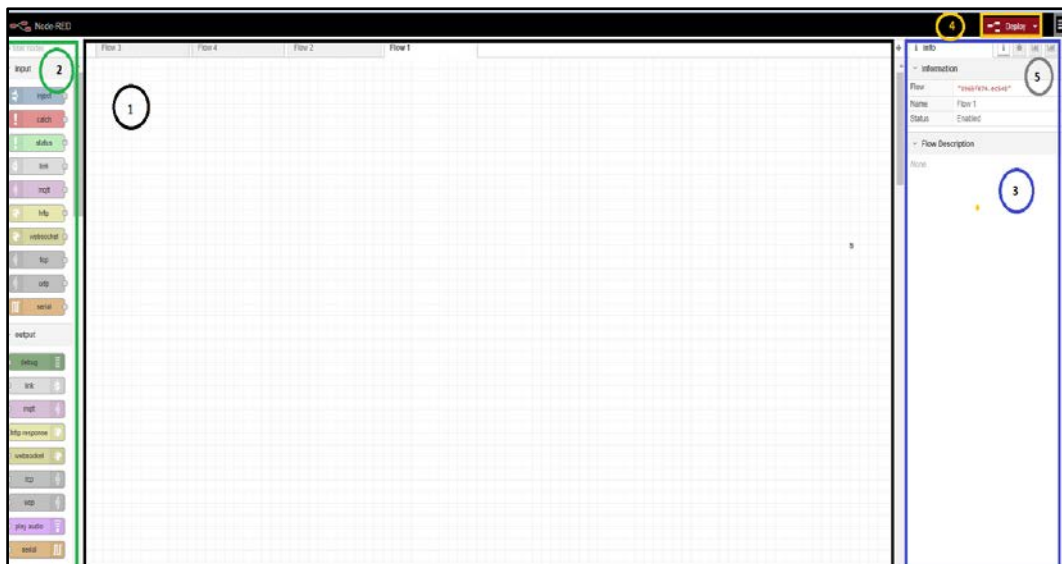
La Figura 36 muestra el proceso de instalación.



**Figura 36.** Instalación de Node-RED.

En la Figura 37 se muestra las principales componentes del entorno de Node-RED:

1. **Área de trabajo:** en este espacio se arrastra los diferentes nodos y se interconecta para formar flujos.
2. **Barra lateral izquierda:** se muestran los diferentes nodos como de entrada, salida, función, social, almacenamiento, análisis, etc.
3. **Barra lateral derecha:** muestra varias pestañas como información, dashboard, configuración, debug y datos de contexto.
4. **Botón Deploy:** permite compilar y ejecutar el programa para ver los resultados a través de la pestaña de depuración.
5. **Menú esquina superior:** permite importar o exportar flujos, eliminar flujos entre otras funcionalidades.



*Figura 37.*Entorno de Node-RED.

### 3.3.3.1 Inicialización de Node-RED

Para inicializar la herramienta visual Node-RED se debe ejecutar MobaXterm como se indicó en la Figura 32. En consola se debe colocar “Node-red” y se obtendrá el resultado de la Figura 38, indicando que se ha conectado con el broker mosquitto.

```

*** System restart required ***
Last login: Fri Feb 15 05:11:40 2019 from 181.113.97.168
root@ubuntu-s-lvcpu-lgb-nyc3-01:~# node-red
16 Feb 22:56:18 - [info]

Welcome to Node-RED
=====

16 Feb 22:56:18 - [info] Node-RED version: v0.19.5
16 Feb 22:56:18 - [info] Node.js version: v8.15.0
16 Feb 22:56:18 - [info] Linux 4.4.0-124-generic x64 LE
16 Feb 22:56:18 - [info] Loading palette nodes
16 Feb 22:56:19 - [warn] rpi-gpio : Raspberry Pi specific node set inactive
16 Feb 22:56:19 - [warn] rpi-gpio : Cannot find Pi RPi.GPIO python library
16 Feb 22:56:19 - [info] Dashboard version 2.12.2 started at /ui
16 Feb 22:56:19 - [info] Settings file : /root/.node-red/settings.js
16 Feb 22:56:19 - [info] Context store : 'default' [module:memory]
16 Feb 22:56:19 - [info] User directory : /root/.node-red
16 Feb 22:56:19 - [warn] Projects disabled : editorTheme.projects.enabled=false
16 Feb 22:56:19 - [info] Flows file : /root/.node-red/flows_ubuntu-s-lvcpu-lgb-nyc3-01.json
16 Feb 22:56:20 - [info] Server now running at http://127.0.0.1:1880/
16 Feb 22:56:20 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

16 Feb 22:56:20 - [info] Starting flows
16 Feb 22:56:20 - [info] Started flows
16 Feb 22:56:20 - [info] [mqtt-broker:a0e1d16f.6b85f] Connected to broker: mqtt://159.65.174.156:1883

```

*Figura 38.* Inicialización de Node-RED

### 3.3.4 IDE de Arduino

Es un entorno en el se realiza la programación de gran variedad de placas de desarrollo. En este proyecto se ha elegido la ESP32 de Heltec la cual puede ser configurada mediante el IDE de Arduino.

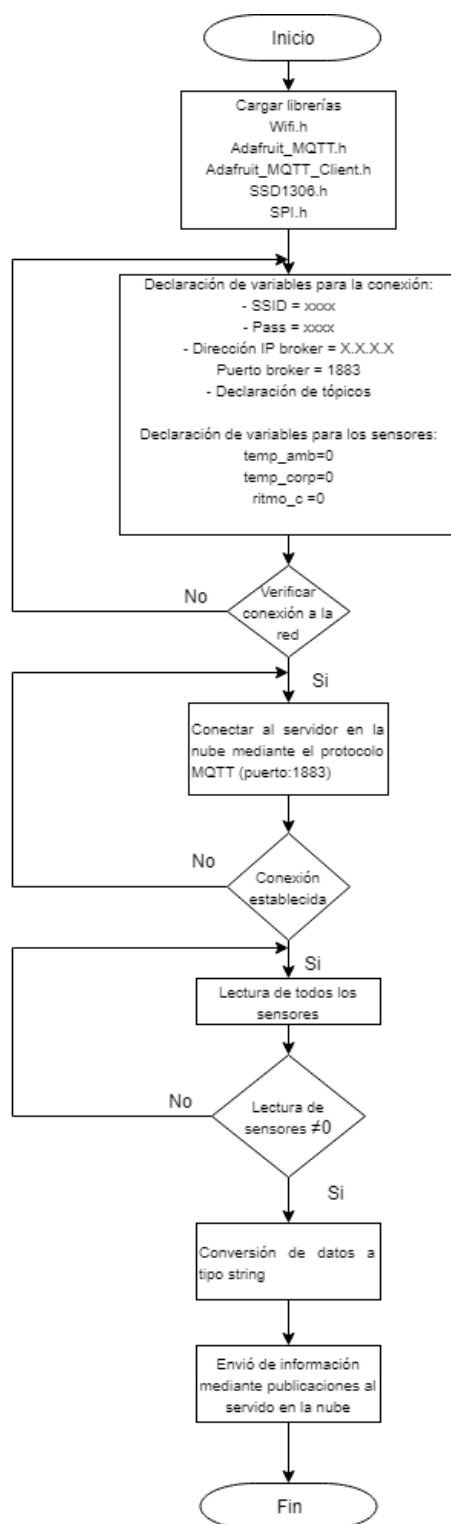
#### 3.3.4.1 Conexión a WiFi y al servidor en la nube

En la Figura 39 se muestra el diagrama de flujo para la conexión inalámbrica mediante WiFi y al servidor en la nube utilizando el protocolo MQTT, primero se ejecuta la conexión a la red y posteriormente al servidor. Se ha implementado las respectivas librerías para realizar este proceso como muestra la **¡Error! No se encuentra el origen de la referencia..**

**Tabla 18****Librerías**

<b>Librerías</b>	<b>Definición</b>
<code>#include &lt;WiFi.h&gt;</code>	Permite la conexión a WiFi.
<code>#include "Adafruit_MQTT.h"</code>	Librería para el protocolo de mensajería MQTT
<code>#include "Adafruit_MQTT_Client.h"</code>	Provee un cliente que permite tanto la suscripción como la publicación de datos usando el protocolo MQTT.
<code>#include "FS.h"</code>	Facilita la gestión del sistema de archivos para dispositivos integrados de almacenamiento
<code>#include "SD.h"</code>	Permite la lectura y escritura en la tarjeta de memoria.
<code>#include &lt;SPI.h&gt;</code>	Permite comunicarse con dispositivos SPI, con Arduino como dispositivo maestro.
<code>#include &lt;NTPClient.h&gt;</code>	Solicita la fecha y hora de un servidor NTP.
<code>#include &lt;WiFiUdp.h&gt;</code>	Crea una instancia de la clase WiFi UDP que puede enviar y recibir mensajes UDP.
<code>#include &lt;Wire.h&gt;</code>	Permite comunicarse con dispositivos I2C / TWI.
<code>#include "RTClib.h"</code>	Permite obtener un Reloj en tiempo real

En la Figura 40 muestra una sección de código con respecto a la conexión WiFi, se ha realizado un bucle que se repetirá 10 veces permitiendo la conexión con el punto de acceso.



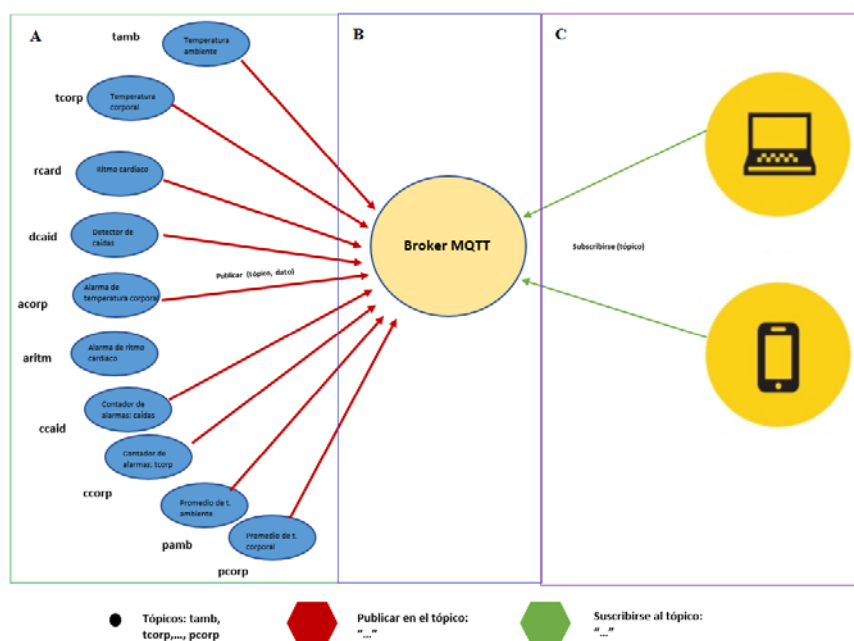
**Figura 39.** Diagrama de flujo de la conexión a WiFi y al servidor en la nube.

```

//CONEXION A INTERNET
WiFi.begin(ssid, password);
while ((WiFi.status() != WL_CONNECTED) && contwifi<=10) {
  contwifi++;
  delay(500);
}
if (WiFi.status() != WL_CONNECTED) {
  // Serial.println("Desconectado");
  WiFi.disconnect();
  display.drawString(5,5,"Iniciando...");
  display.drawString(20,20,"WiFi Desconectado");
  display.display();
}
else if (WiFi.status() == WL_CONNECTED) {
  Serial.println("Conectado a la red de WiFi");
  Serial.println("Dirección IP: ");
  Serial.print(WiFi.localIP());
  display.drawString(5,5,"Iniciando...");
  display.drawString(20,20,"Conectando a WiFi");
  display.display();
}
}

```

**Figura 40.** Conexión a WiFi



**Figura 41.** Diagrama de publicación y suscripción de los tópicos

Además, se han declarado los tópicos que son los temas donde se suscriben los receptores para recibir los mensajes, que se representa a través de una cadena de texto, en la Figura 41 se muestra la publicación y suscripción a los diferentes tópicos. La parte A representan los tópicos que se han creado para el envío de los datos monitoreados a través de la publicación, la parte B representa el

broker y la parte C los diferentes clientes suscritos a los tópicos creados en la parte A. En la Figura 42 se muestra una sección del código para realizar la publicación de los tópicos mencionados anteriormente.

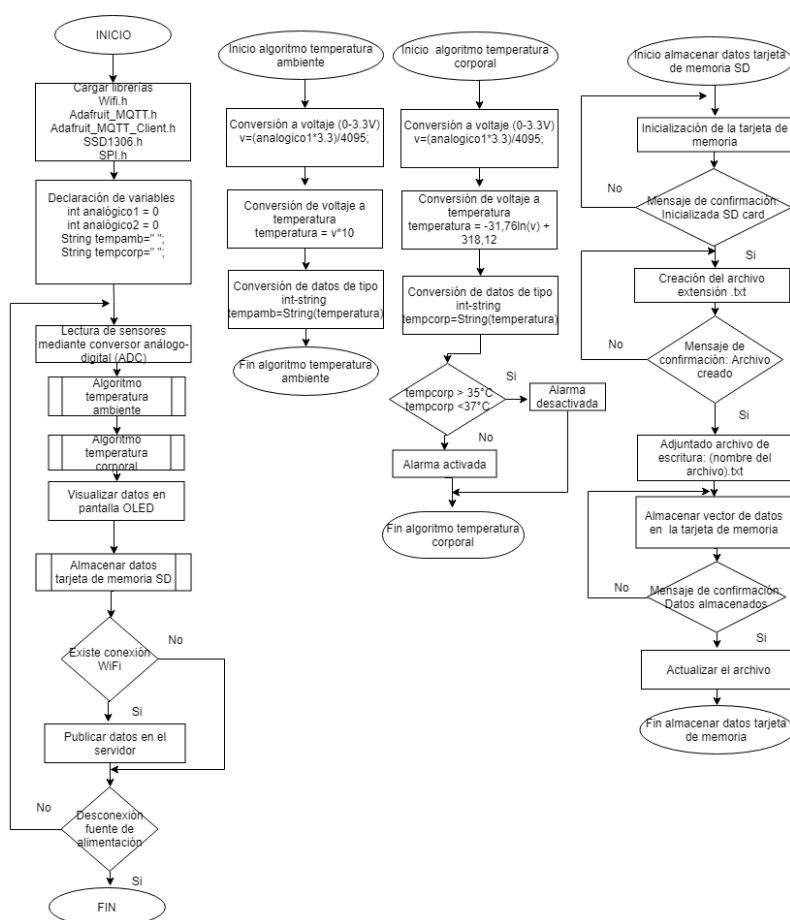
```

Adafruit_MQTT_Publish temp_amb = Adafruit_MQTT_Publish(&mqtt, "tamb");
Adafruit_MQTT_Publish temp_corp = Adafruit_MQTT_Publish(&mqtt, "tcorp");
Adafruit_MQTT_Publish ritmo_cardiaco = Adafruit_MQTT_Publish(&mqtt, "rcard");
Adafruit_MQTT_Publish detector_caída = Adafruit_MQTT_Publish(&mqtt, "dcaid");
Adafruit_MQTT_Publish alarma_temp = Adafruit_MQTT_Publish(&mqtt, "acorp");
Adafruit_MQTT_Publish alarma_pulso = Adafruit_MQTT_Publish(&mqtt, "aritm");
Adafruit_MQTT_Publish conteo_caída = Adafruit_MQTT_Publish(&mqtt, "ccaid");
Adafruit_MQTT_Publish conteo_temperatura = Adafruit_MQTT_Publish(&mqtt, "ccorp");
Adafruit_MQTT_Publish conteo_ritmo = Adafruit_MQTT_Publish(&mqtt, "critm");
Adafruit_MQTT_Publish prom_tambiente = Adafruit_MQTT_Publish(&mqtt, "pamb");
Adafruit_MQTT_Publish prom_tcorporal = Adafruit_MQTT_Publish(&mqtt, "pcorp");

```

**Figura 42.** Publicación de tópicos

### 3.3.4.2 Desarrollo del software del procesamiento de temperatura ambiente y corporal.



**Figura 43.** Diagrama de flujo para obtener temperatura ambiente y corporal

La Figura 43 muestra el diagrama de flujo que se ha desarrollado para el funcionamiento tanto del sensor de temperatura ambiente como el de temperatura corporal, los dos sensores hacen uso de entradas analógicas de la tarjeta ESP32. Además, se han agregado alarmas cuando los datos presentan anomalías, tomando en consideración lo mencionado por Health Encyclopedia (Healthwise, 2017):

- Temperatura corporal baja  $> 35\text{ }^{\circ}\text{C}$
- Temperatura corporal alta  $< 37\text{ }^{\circ}\text{C}$

La Figura 44 muestra una sección de código de los algoritmos utilizados para la obtención de los valores de temperatura ambiente (a) y temperatura corporal (b).

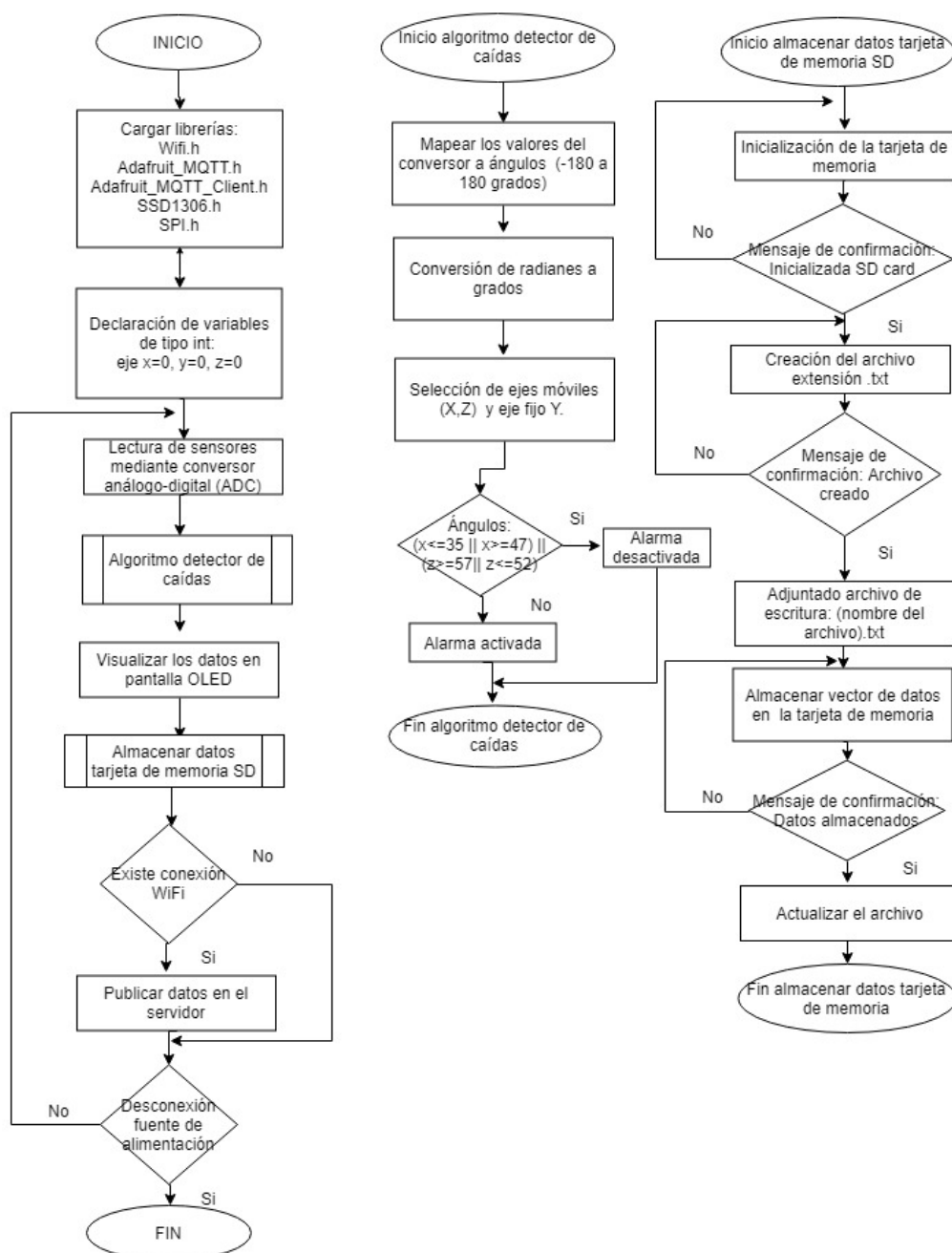
<pre> analogico2 = analogRead(35); v = analogico2*3.3; v = v_ambiente/4095.0; temperatura = v*10; </pre> <p style="text-align: center;">a)</p>	<pre> temperatura = -31,76ln(x) + 318,12 analogico=analogRead(36); v=analogico*3.3; v=v/4095; if (v!=0) {     r=33000/v;     r=r-10000;     temperatura=log (r);     temperaturaf=temperatura </pre> <p style="text-align: center;">b)</p>
--	--

**Figura 44.** Obtención valor de temperatura a) ambiente y b) corporal

### 3.3.4.3 Desarrollo del software del procesamiento para detección de caídas.

La Figura 45 muestra el diagrama de flujo del programa desarrollado para la detección de caídas del paciente. Cabe indicar que los valores de los ángulos para la implementación del software se obtienen de la Tabla 11.





**Figura 45.** Diagrama de flujo para Detectar caídas.

La Figura 46 indica una sección de código del algoritmo para la detección de caídas, se muestra el rango de los ángulos para determinar la caída del paciente.

```

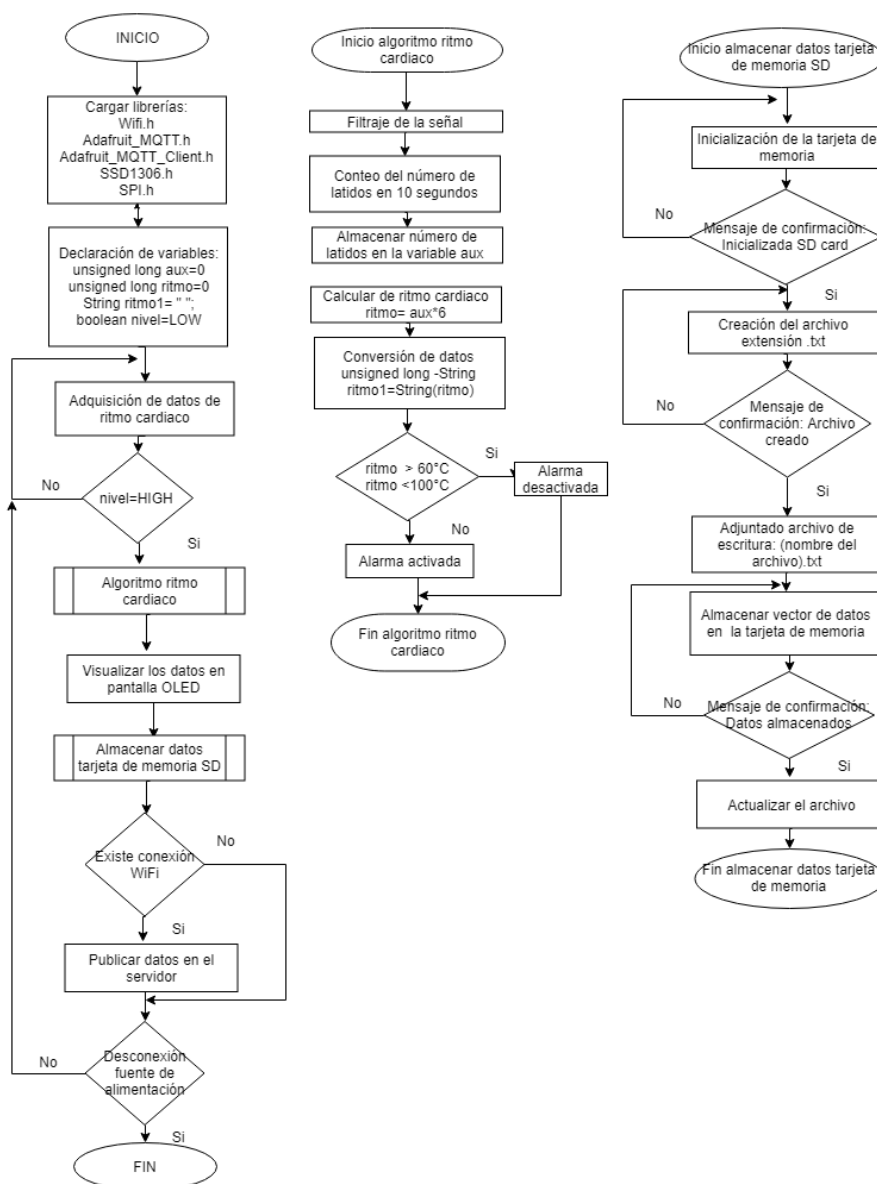
y = RAD_TO_DEG*(atan2(-yAng,-zAng)+PI);
x = RAD_TO_DEG*(atan2(-xAng,-zAng)+PI);
z = RAD_TO_DEG*(atan2(-yAng,-xAng)+PI);

if ((x<=35 || x>=47) || (z>=57 || z<=52)){
  alarmal=1;
  aviso1 = "Caída";
  aviso = "Caída";
}

```

**Figura 46.** Obtención de ángulos

### 3.3.4.4 Desarrollo del software del procesamiento para el ritmo cardiaco.



**Figura 47.** Diagrama de flujo para obtener ritmo cardiaco

La Figura 47 muestra el diagrama de flujo del programa desarrollado para la detección del ritmo cardiaco del paciente. También se ha agregado alarmas considerando lo descrito por la organización Heart Rhythm Society (Society, 2013) en el caso que se presente anomalías en los datos monitoreados como se muestra a continuación:

- Ritmo cardiaco bajo  $> 60$  °C
- Ritmo cardiaco alta  $< 100$  °C

La Figura 48 indica una sección de código del algoritmo para la obtención del ritmo cardiaco.

```
tiempo_ppg = pulseIn(pulso, HIGH);  
aux = tiempo_ppg/1000;  
aux =total/1000;  
aux = total*2;  
ritmo =total *60;  
ritmo = ritmo *6
```

**Figura 48.** Obtención del valor de ritmo cardiaco.

### 3.3.5 Aplicación web

Para el desarrollo de la página web se ha utilizado la herramienta Node-RED que permite a través de nodos que se encuentran previamente configurados para la creación de un Dashboard, que permite visualizar los datos monitoreados. Para acceder a la página web se ingresa en el navegador con la dirección que se compone de la IP del Broker, dos puntos (:), el puerto por el que se ejecuta, barra oblicua (/) y *ui*, <http://159.65.174.156:1880/ui/>.

Se ha considerado que el registro de los multiusuarios se debe realizar previamente en la herramienta Node-RED, esto permite seguridad y privacidad en la obtención de los datos de los pacientes, es decir solo el administrador tiene acceso el nombre de usuario y su respectiva

contraseña. Los multiusuarios (doctores, enfermeras y pacientes) registrados que se muestran en las Tablas 18, 19 y 20 son tanto para la aplicación móvil y web.

**Tabla 18**

*Usuarios: Doctores*

Usuario	Contraseña
doctor1	doctor1
doctor2	doctor2

**Tabla 19**

*Usuarios: Enfermeras*

Usuario	Contraseña
enfermera1	enfermera1
enfermera2	enfermera2

**Tabla 20**

*Usuarios: Pacientes*

Usuario	Contraseña	Cédula
paciente1	paciente1	1003536339
paciente2	paciente2	1000795800

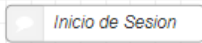
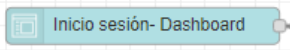

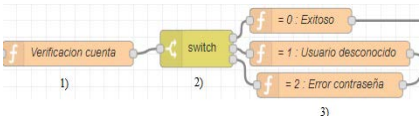
El Dashboard posee tres pestañas para cada multiusuario con su respectivo inicio de sesión, la Tabla 21 muestra los datos monitoreados para cada usuario, esto se ha personalizado dependiendo la función que cada usuario desempeña. En las Tablas 22, 23, 24, 25 y 26, se detallan cada uno de los flujos que se han creado para la construcción del Dashboard.

**Tabla 21**  
*Multiusuarios*

Usuario	Datos monitoreados visualizados	Observación
Doctor	Temperaturas ambiente y corporal, ritmo cardiaco, promedio de temperaturas, alarmas y contador de alarmas.	Este usuario puede registrar a los pacientes con sus datos personales, guardar los datos monitoreados, y descargar la información de cada paciente a través del atributo cédula.
Enfermera	Temperaturas ambiente y corporal, ritmo cardiaco, alarmas y contador de alarmas.	Este usuario puede visualizar las alarmas de los datos monitoreados con sus respectivos contadores.
Paciente	Temperaturas ambiente y corporal, ritmo cardiaco, alarma de detección de caídas.	Este usuario solo puede visualizar datos monitoreados y descargar su propia información.

### 3.3.5.1 Inicio de sesión

**Tabla 22**  
*Descripción de los nodos de inicio de sesión*

Nodo	Propiedades del nodo
<p>1) </p> <p>2) </p>	<p>2) Nodo “comment” permite agregar comentarios en el flujo de trabajo.</p> <p>3) Nodo “form” permite crear etiquetas para la pantalla de inicio de sesión, contiene campos para el ingreso del usuario y la contraseña.</p>
	<p>Nodo “function” permite la creación diferentes cuentas con su respectiva usuario y contraseña</p>
	<p>1) Nodo “function” permite la verificación de las cuentas creadas, con su respectiva usuario y contraseña, además se ha configurado varias condiciones como: ingreso correcto tanto usuario como contraseña, si el usuario es erróneo y en caso de error en la contraseña.</p> <p>2) Nodo “switch” permite elegir la condición de acuerdo a lo configurado en el nodo función.</p>

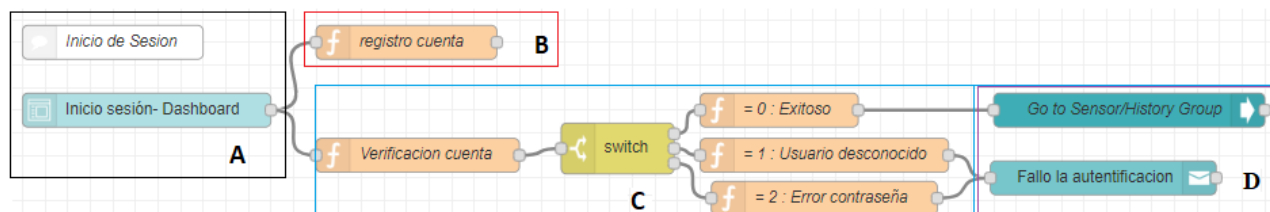
CONTINÚA 

- 3) Nodo “function”: con la etiqueta Exitoso permite acceder al Dashboard que contiene todos los datos monitoreados, mientras que en las etiquetas Usuario desconocido y Error contraseña permite la configuración de estos mensajes en caso del ingreso erróneo de los mismo.



- 1) Nodo “UI-control”, permite control con el Dashboard y el acceso a los datos monitoreados.
- 2) Nodo “notification”, crea alertas para el usuario, mediante una ventana emergente, estas se mencionaron anteriormente.

A continuación, en la Figura 49 se muestra el flujo completo de inicio de sesión en la herramienta Node-RED.

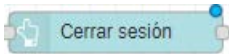


**Figura 49.**Flujo de inicio de sesión.

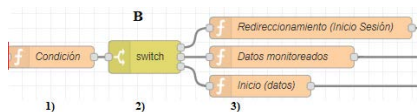
### 3.3.5.2 Cierre de sesión

**Tabla 23**

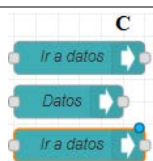
*Descripción de los nodos de cierre de sesión*

Nodo	Propiedades del nodo
	<p>1) Nodo “button” permite agregar un botón para el cierre de sesión a través de un clic, esta acción se ha representado a través del envío a través de un dato String.</p>

CONTINÚA 

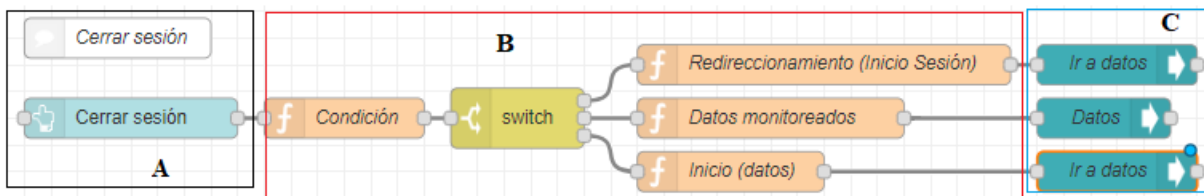


- 1) Se ha configurado algunas condiciones en caso de que el dato recibido es 0, se cierra la sesión caso contrario se mantiene en la página de datos monitoreados.
- 2) Se elige la condición de acuerdo con el dato recibido.
- 3) De acuerdo con la condición elegida se realiza la acción correspondiente.



Estos nodos trabajan juntamente con lo configurado en el nodo “function” anteriormente, permite la redirección hacia la página de inicio de sesión.

A continuación, en la Figura 50 se muestra el flujo completo de cierre de sesión en la herramienta Node-RED.



**Figura 50.**Flujo de cierre de sesión.

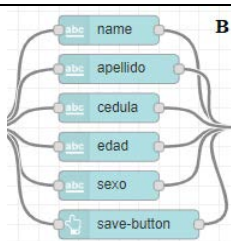
### 3.3.5.3 Registro de pacientes

**Tabla 24**

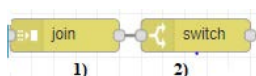
*Descripción de los nodos para el registro de pacientes*

Nodo	Propiedades del nodo
	<ol style="list-style-type: none"> <li>1) En el nodo “set” se ha configurado la condición para borrar los campos de nombre, apellido, cédula, edad y sexo.</li> <li>2) A través de este nodo mediante un clic se borran todos los campos.</li> </ol>

CONTINÚA



Nodo “text input” permite agregar campos de texto en este caso son: nombre, apellido y sexo datos de tipo String mientras que cédula y edad son datos de tipo entero.

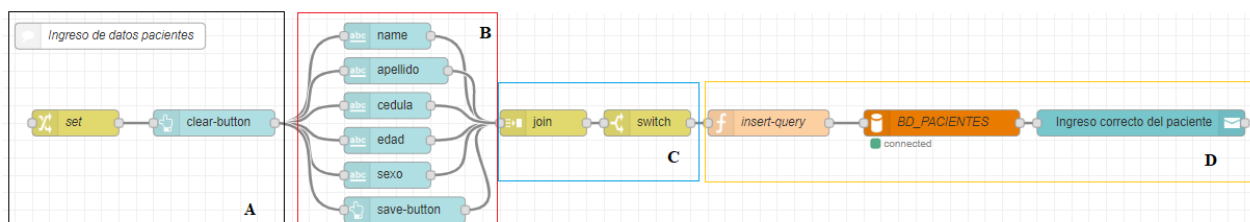


- 1) El nodo “join” une la secuencia de mensajes en este caso de nombre, apellido, cedula, edad, sexo en un solo mensaje esta acción se realiza para posteriormente lograr el almacenamiento de información en la tabla denominada pacientes.
- 2) Relacionado con la acción de almacenamiento de los datos.



- 1) Función de almacenamiento de datos a través de la sentencia “Insert” de MySQL (INSERT INTO (nombre de la tabla) VALUES (atributos)).
- 2) El node “MySQL” permite realizar la conexión con la base de datos denominada tesis
- 3) Se visualiza mediante una ventana emergente el correcto ingreso de datos.

A continuación, en la Figura 51 se muestra el flujo de registro de pacientes en la herramienta Node-RED.




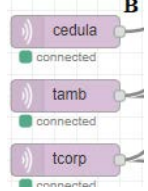
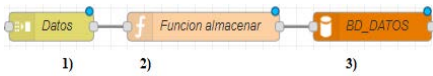

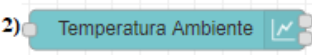
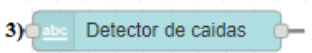
**Figura 51.**Flujo de registro de pacientes.



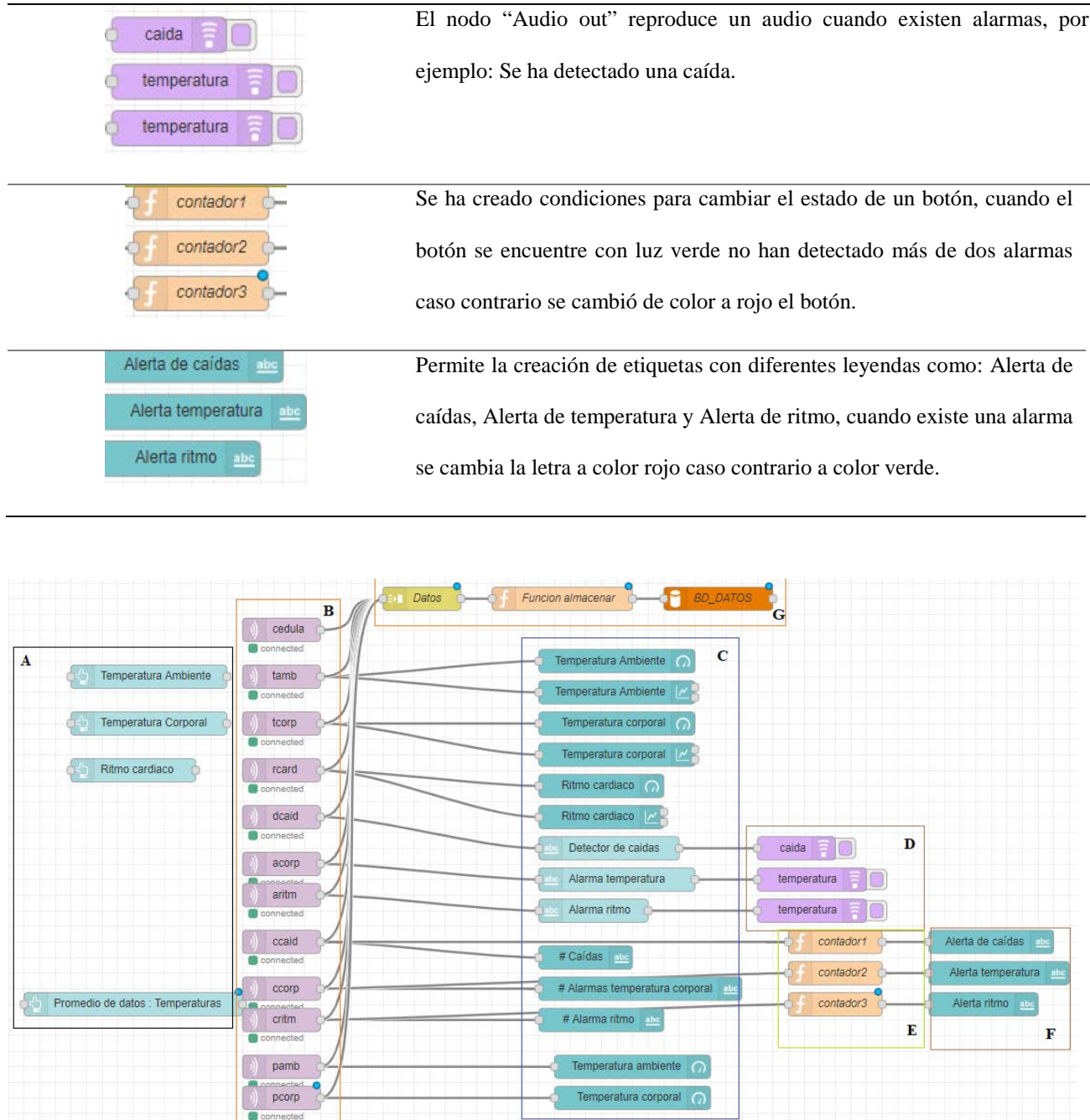
### 3.3.5.4 Visualización de los datos monitoreados

**Tabla 25**

*Descripción de los nodos para la visualización de datos*

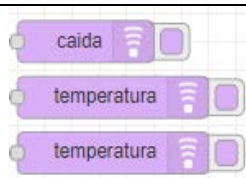
Nodo	Propiedades del nodo
<p><b>A</b></p> 	<p>Se agregaron botones en la página de visualización de datos en este caso los hemos utilizado como etiqueta de leyenda por ejemplo: “Temperatura Ambiente”</p>
<p><b>B</b></p> 	<p>El nodo “Mqtt” permite realizar la conexión con el bróker. En este nodo se ejecuta la suscripción a los diferentes tópicos creados anteriormente en la Figura 41.</p>
	<ol style="list-style-type: none"> <li>1) El nodo “join” une la secuencia de mensajes en este caso de todos los tópicos en un solo mensaje esta acción se realiza para lograr el almacenamiento de información en la tabla denominada datos.</li> <li>2) Se realiza la función de almacenamiento de datos a través de la sentencia “Insert” de MySQL, con la estructura que se muestra a continuación: I NSERT INTO (nombre de la tabla) VALUES (tamb, tcorp..., etc).</li> <li>3) El node “MySQL” permite realizar la conexión con la base de datos.</li> </ol>
<ol style="list-style-type: none"> <li>1) </li> <li>2) </li> <li>3) </li> </ol>	<ol style="list-style-type: none"> <li>1) El nodo “Gauge” es un indicar del dato recibido de forma gráfica.</li> <li>2) El nodo “Chart”, muestra una gráfica a lo largo del tiempo de los datos monitoreados.</li> <li>3) El nodo “Text input” muestra las diferentes alarmas que se presentan en el monitoreo de los datos</li> </ol>

CONTINÚA 

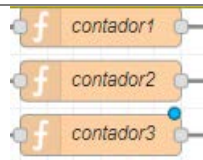


**Figura 52.**Flujo de datos monitoreados

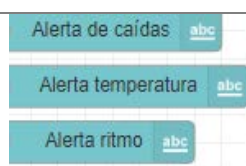
En la Figura 52 se muestra el flujo de datos monitoreados en la herramienta Node-RED.



El nodo “Audio out” reproduce un audio cuando existen alarmas, por ejemplo: Se ha detectado una caída.



Se ha creado condiciones para cambiar el estado de un botón, cuando el botón se encuentre con luz verde no han detectado más de dos alarmas caso contrario se cambió de color a rojo el botón.


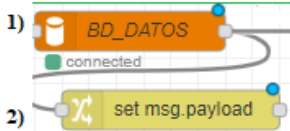
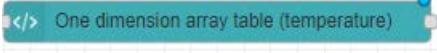


Permite la creación de etiquetas con diferentes leyendas como: Alerta de caídas, Alerta de temperatura y Alerta de ritmo, cuando existe una alarma se cambia la letra a color rojo caso contrario a color verde.

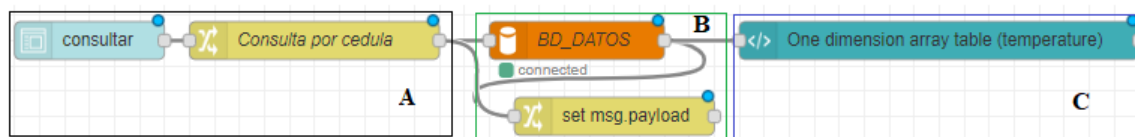
### 3.3.5.5 Consulta de datos monitoreados

**Tabla 26**

*Descripción de los nodos para la consulta de datos*

Nodo	Propiedades del nodo
	<p>1) El nodo “form” permite agregar un campo para el ingreso de la cedula, a través de este atributo se obtendrá información del paciente.</p> <p>2) El nodo “change” permite utilizar la sentencia “Select” de MySQL con la estructura que se muestra a continuación: SELECT (columnas) FROM (nombre de la tabla) WHERE (condición).</p>
	<p>1) El node “MySQL” permite realizar la conexión con la base de datos.</p> <p>2) Se realiza la creación de un vector para permitir el almacenamiento contiguo de los datos monitoreados.</p>
	<p>El nodo “template” permite la creación del diseño de la tabla para la visualización de los datos consultados, además crea un botón de descargar archivo de datos con extensión .xls</p>

A continuación, en la Figura 53 se muestra el flujo de la consulta de datos a través del atributo cédula en la herramienta Node-RED.



**Figura 53.** Flujo de consulta de datos.

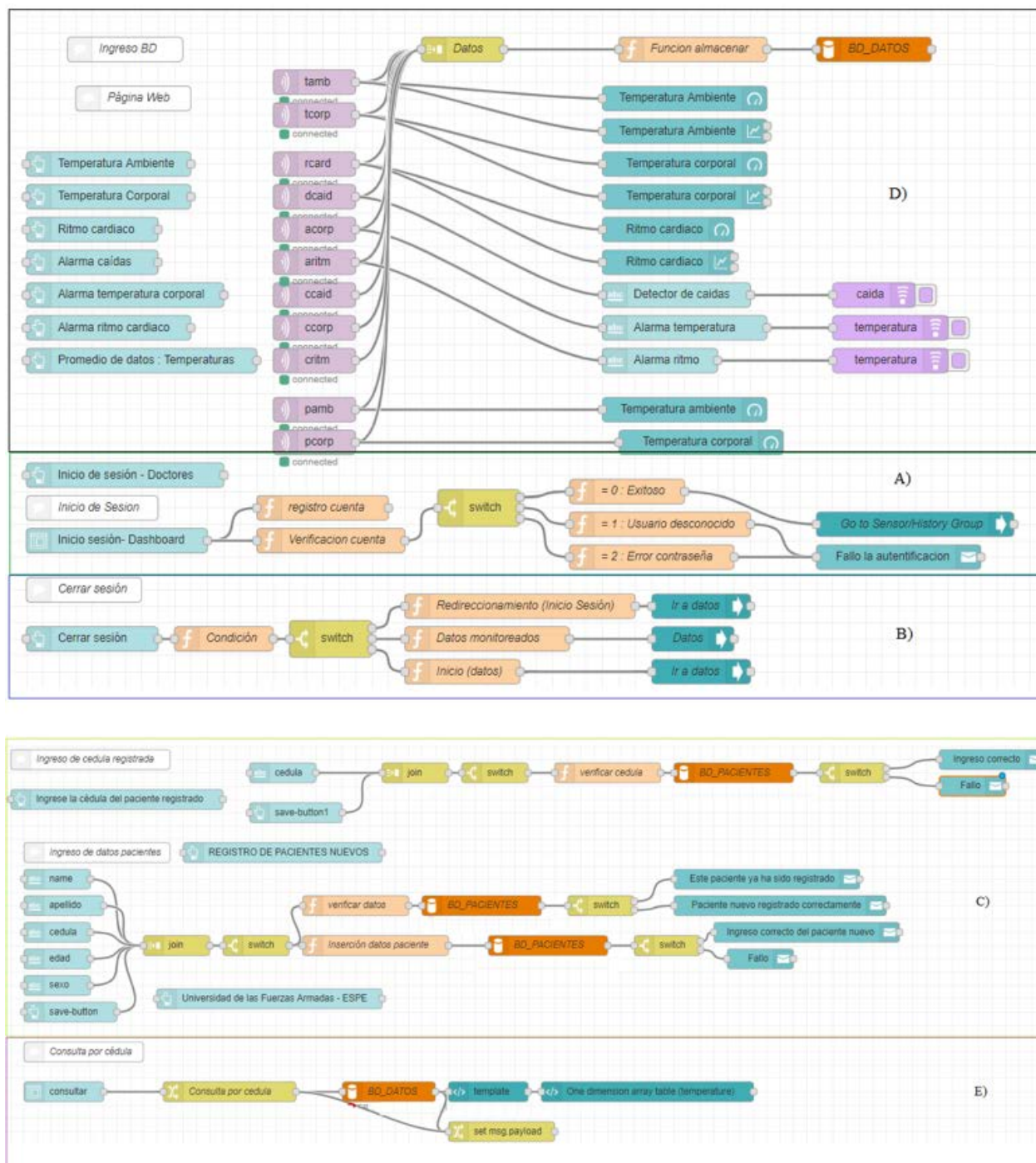
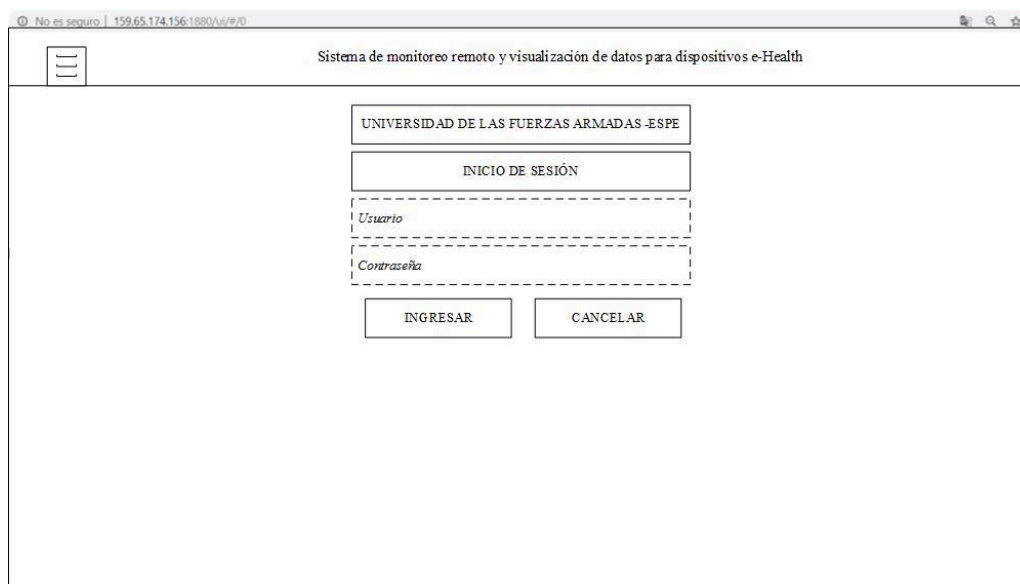


Figura 54. Flujo total.

La Figura 54 indica el flujo total para el desarrollo de la página web la parte A indica el inicio de sesión, B el cierre de sesión, C el ingreso de datos del paciente y D visualización de datos monitoreados.

La Figura 55 muestra un bosquejo de la página de inicio de sesión, se presenta con la misma estructura para los diferentes usuarios: doctores, enfermeras y pacientes. Se debe mencionar que la parte superior izquierda despliega un menú para elegir a qué tipo de usuario se desea acceder.

Las Figuras 56, 57, 58 y 59 muestran un bosquejo de los datos monitoreados para los diferentes multiusuarios.



The image shows a web browser window with the address bar displaying "No es seguro | 159.65.174.156:1880/a/#/0". The page title is "Sistema de monitoreo remoto y visualización de datos para dispositivos e-Health". The page content includes a hamburger menu icon in the top left corner. The main content area features a centered login form with the following elements: a header box containing "UNIVERSIDAD DE LAS FUERZAS ARMADAS -ESPE", a box labeled "INICIO DE SESIÓN", a dashed-line input field for "Usuario", a dashed-line input field for "Contraseña", and two buttons at the bottom: "INGRESAR" and "CANCELAR".

**Figura 55.** Estructura de la página de inicio de sesión

Sistema de monitoreo remoto y visualización de datos para dispositivos e-Health

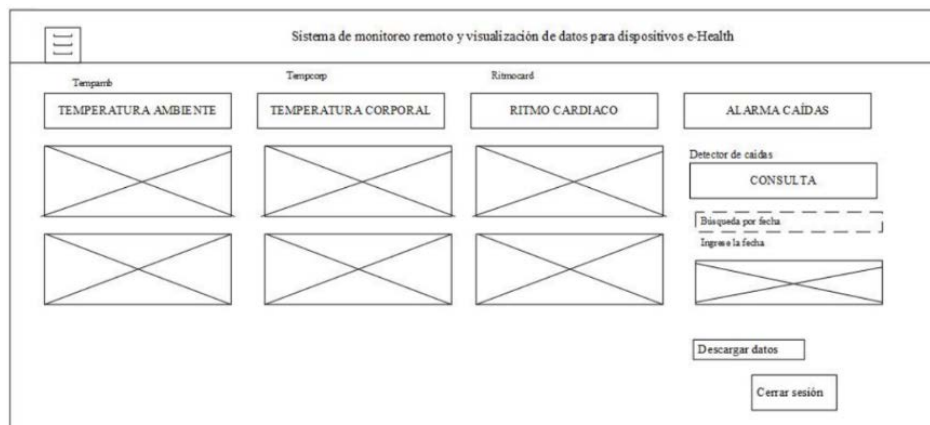
Pacientes	Tempamb	Tempcorp	Ritmocard
REGISTRO DE PACIENTES NUEVOS	TEMPERATURA AMBIENTE	TEMPERATURA CORPORAL	RITMO CARDIACO
Nombre	<input type="text"/>	<input type="text"/>	<input type="text"/>
Apellido	<input type="text"/>	<input type="text"/>	<input type="text"/>
Cédula	<input type="text"/>	<input type="text"/>	<input type="text"/>
Edad	<input type="text"/>	<input type="text"/>	<input type="text"/>
Sexo	<input type="text"/>	<input type="text"/>	<input type="text"/>
Guardar información del paciente			
INGRESE LA CÉDULA DEL PACIENTE REGISTRADO			
Cédula	Alarma caídas	Alarma temperatura	Alarma ritmo
Guardar datos monitoreados	ALARMA CAÍDAS	ALARMA TEMPERATURA CORPORAL	ALARMA RITMO CARDIACO
Terminar el almacenamiento de datos	Detector de caídas:	Alarma temperatura:	Alarma ritmo:
Reiniciar contadores	# Caídas Totales	# Alarmas T. Corporal	# Alarmas de Ritmo Cardíaco
Cerrar sesión	PROMEDIO DE DATOS	CONSULTA	
	<input type="text"/>	Consulta de datos	
	<input type="text"/>	Ingrese el número de cédula	
		<input type="text"/>	
		<input type="button" value="Consultar"/> <input type="button" value="Cancelar"/>	
		Búsqueda por fecha	
		Ingrese la fecha	
		<input type="text"/>	
		<input type="button" value="Descargar datos"/>	

**Figura 56.** Estructura de la página de monitoreo de datos, usuario:doctor

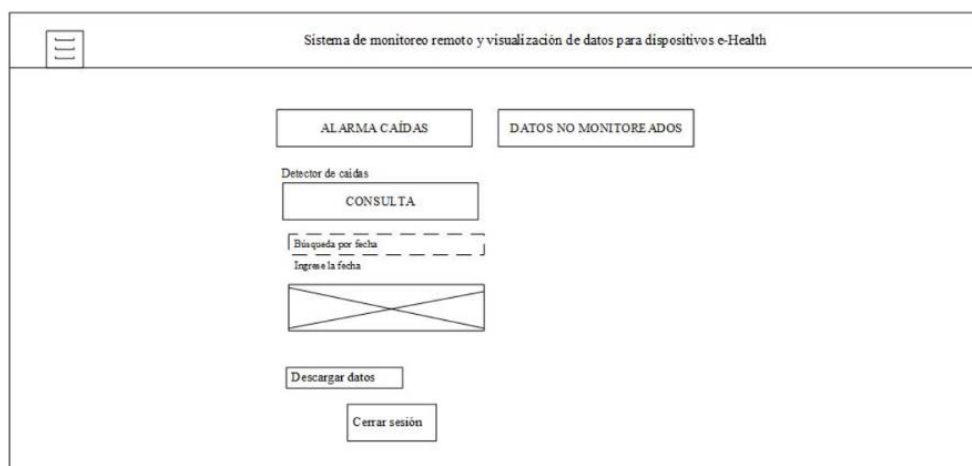
Sistema de monitoreo remoto y visualización de datos para dispositivos e-Health

Tempamb	Tempcorp	Ritmocard	
TEMPERATURA AMBIENTE	TEMPERATURA CORPORAL	RITMO CARDIACO	ALARMA CAÍDAS
<input type="text"/>	<input type="text"/>	<input type="text"/>	Detector de caídas:
<input type="text"/>	<input type="text"/>	<input type="text"/>	# Caídas
			Alerta de caídas
			ALARMA TEMPERATURA CORPORAL
			Alarma temperatura:
			# Alarmas temperatura corporal
			Alarma temperatura
			ALARMA RITMO CARDIACO
			Alarma ritmo:
			# Alarmas ritmo cardíaco
			Alarma ritmo

**Figura 57.** Estructura de la página de monitoreo de datos, usuario:enfermera



**Figura 58.** Estructura de la página de monitoreo de datos, usuario: paciente



**Figura 59.** Estructura de la página de descarga de datos, usuario: paciente

### 3.3.5.6 Reglas de validación

Se ha realizado diferentes validaciones mostradas en la Tabla 27 en la aplicación web para los diferentes usuarios:

**Tabla 27**

*Reglas de validación de la aplicación web*

Usuario	Validación	Observación
		El Doctor debe ingresar cada uno de los datos
	Registro de pacientes	personales del paciente para registrarlo en la BD y se

CONTINÚA 

		emite el mensaje “Ingreso correcto del paciente nuevo”.
Doctor		En el caso que el paciente haya sido registrado se emite un mensaje “Este paciente ya ha sido registrado”.
	Almacenamiento de datos en la BD	Se debe ingresar la cédula del paciente registrado y presionar el botón de “Guardar datos monitoreados”.
	Terminar el almacenamiento de datos	Se debe presionar el botón de “Terminar el almacenamiento” y posteriormente reiniciar los contadores, esta acción también se realiza por medio de un botón.
Paciente	Visualización de los datos	El paciente solo podrá visualizar sus datos monitoreados cuando el usuario doctor haya registrado su atributo cédula en el campo de “Ingreso de la cédula del paciente”, caso contrario solo puede acceder a descargar su información personal.

### 3.3.6 Aplicación móvil

La aplicación móvil del proyecto de investigación fue desarrollada en Andriod Studio que es un entorno de desarrollo integrado (IDE) para la plataforma de Andriod, se ha elegido este entorno porque permite la ejecución de la app directamente desde el móvil y además por la cantidad de herramientas que posee como: Sistema de compilación basado en Gradle flexible, emulador rápido, instant Run para aplicar cambios mientras la app se ejecuta sin la necesidad de compilar un nuevo APK.



### 3.3.6.1 Desarrollo de la aplicación.

Para el desarrollo de la aplicación se ha utilizado el Servicio Paho porque es una biblioteca del cliente de MQTT escrita en Java para desarrollar aplicaciones en Android.

El Servicio de Paho, tuvo sus inicios en el 2008, se había desarrollado el cliente síncrono, llamado MQTTClient, que tuvo los siguientes puntos de diseño:

- Subprocesos: no hay subprocesos en segundo plano, o solo uno, independientemente de cuántas conexiones se creen.
- Bloqueo de llamadas MQTT, para simplificar la programación de aplicaciones.
- Rastreo interno y rastreo de memoria, para la capacidad de servicio y la eliminación de fugas de memoria (Craggs, 2015).

En la Figura 61 se muestra las características del Servicio de Paho.

Característica		Característica	
MQTT 3.1	✓	QoS 0	✓
MQTT 3.1.1	✓	QoS 1	✓
LWT	✓	QoS 2	✓
SSL / TLS	✓	Autenticación	✓
Reconexión automática	✗	Estrangulamiento	✗
Persistencia del disco	✓	Mensaje sin conexión Buffering	✗

**Figura 60.** Características de Servicio de Paho  
Fuente: (Craggs, 2015)

En el desarrollo para la aplicación móvil se utiliza la herramienta Gradle que es el sistema de compilación y administración de dependencias (librerías) para añadir el servicio mencionado.

La Figura 61 muestra las líneas que se deben añadir, la parte A indica que se agrega el repositorio de Eclipse Maven al *build.gradle* archivo, permitiendo que este pueda encontrar el JAR de Servicio Paho para Android, mientras que la parte B agrega este servicio como una dependencia de la aplicación.

```

a)
repositories {
    maven {
        url "https://repo.eclipse.org/content/repositories/paho-releases/"
    }
}

b)
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:28.0.0'
    compile('org.eclipse.paho:org.eclipse.paho.android.service:1.0.2')
    {
        exclude module: 'support-v4'
    }
}

```

**Figura 61.** Líneas de código para el Servicio de Paho de Android

Para crear un enlace al Servicio Paho para Android, este debe ser declarado en el archivo de configuración *AndroidManifest.xml* como se muestra en la Figura 62, además se debe agregar los siguientes permisos para su funcionamiento.

```

<service
android:name="org.eclipse.paho.android.service.MqttService"></service>

<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.VIBRATE" />

```

**Figura 62.** Líneas de código para conexión y permiso.

En la Figura 63 se muestra las líneas de código para crear y establecer una conexión MQTT, para ello se debe añadir la IP del servidor en la nube seguido del puerto MQTT, esto debe ser declarado en *MainActivity.java* que es una clase donde se definen los métodos.

```

String clientId = MqttClient.generateClientId();
client =
    new MqttAndroidClient(this.getContext(),
        "tcp://159.65.174.156:1883", clientId);
options = new MqttConnectOptions();

```

**Figura 63.** Líneas de código para crear y establecer conexión Mqtt

Para realizar el proceso de suscripción a los diferentes tópicos, se describe en la parte A de la Figura 64, mientras que la parte B muestra la creación a del método setSubscription, en la línea client.suscribe(topic, 0) se indica el tópicos y la Calidad de Servicio (QoS) en este caso de cero.

<pre>String topicStr1= "tamb"; String topicStr2= "teorp"; String topicStr3= "rcard"; String topicStr4= "dcaid"; String topicStr5= "acorp"; String topicStr6= "aritm"; String topicStr7= "ccaid"; String topicStr8= "ecorp"; String topicStr9= "critm"; String topicStr10= "pamb"; String topicStr11= "pcorp";</pre>	<pre>private void setSubscription(String topic) {     try{         client.subscribe(topic,0);     }catch (MqttException e){         e.printStackTrace();     } }</pre>
a)	b)

**Figura 64.** Líneas de código de suscripción de los tópicos.

La aplicación móvil se constituye de dos interfaces, en la Tabla 28 se detalla las propiedades de la pantalla general de inicio de sesión de la aplicación móvil, se especifica el tipo de ID, evento y acción. La Figura 65 muestra el bosquejo de la ventana de elección del usuario (doctores, enfermeras o pacientes) y la Figura 66 el inicio de sesión para acceder a los datos monitoreados.

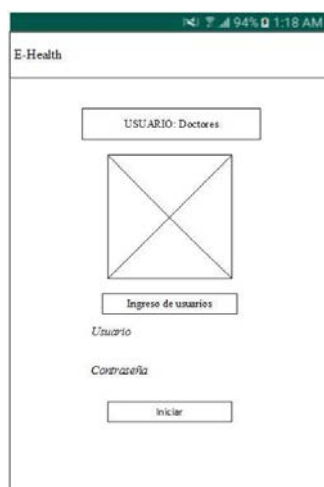
**Tabla 28**

*Propiedades básicas de la interfaz de inicio de sesión.*

Texto	Tipo	Id	Evento	Acción
Dispositivo de e-Health	TextView	txttítulo	Ninguno	-
	ImageView	imagen	Ninguno	-
Ingreso de usuarios	TextView	txtingreso	Ninguno	Introducir usuario
Usuario	EditText	txtusuario	Ninguno	Introducir password
Password	EditText	txtpassword	Ninguno	-
Iniciar	Button	ingresar	onClick	Ir a la interfaz de monitorización.



**Figura 65.** Estructura de la selección de los usuarios de la aplicación móvil.



**Figura 66.** Estructura de inicio de sesión.

En la Tabla 29 se detalla cada una de las propiedades para el desarrollo de la pantalla donde se puede visualizar los datos monitoreados.

**Tabla 29**

*Propiedades básicas de la interfaz para el monitoreo de los datos.*

Texto	Tipo	Id	Evento	Acción
Temp. Ambiente (°C)	TextView	sub1	Ninguno	-
Temp. Corporal (°C)	TextView	sub2	Ninguno	-
Ritmo Cardíaco (LPM)	TextView	sub3	Ninguno	-

CONTINÚA 

Detector de caídas	TextView	sub4	Ninguno	-
Alarma temp. Corp	TextView	sub5	Ninguno	-
Alarma de ritmo	TextView	sub6	Ninguno	-
Cont. de caídas	TextView	sub7	Ninguno	-
Cont. alarma tcorp	TextView	sub8	Ninguno	-
Cont. alarma ritmo	TextView	sub9	Ninguno	-
Prom. temp-amb	TextView	sub10	Ninguno	-
Prom. Temp-corp	TextView	sub11	Ninguno	-
	TextView	subText1	Ninguno	-
	TextView	subText2	Ninguno	-
	TextView	subText3	Ninguno	-
	TextView	subText4	Ninguno	-
	TextView	subText5	Ninguno	-
	TextView	subText6	Ninguno	-
	TextView	subText7	Ninguno	-
	TextView	subText8	Ninguno	-
	TextView	subText9	Ninguno	-
	TextView	subText10	Ninguno	-
	TextView	subText11	Ninguno	-
Conectado	Button	connBtn	onClick	Conexión con el bróker
Desconectado	Button	disconnBtn	onClick	Desconexión con el bróker
	ImageView	imagen1	Ninguno	-
	ImageView	imagen2	Ninguno	-
	ImageView	imagen3	Ninguno	-

Las Figuras 67, 68 y 69 muestran un bosquejo de la interfaz de los datos monitoreados para cada usuario. En la parte superior derecha se presenta un menú que permite cerrar la sesión del usuario. Además, una vez desarrollada la aplicación la Figura 70 presenta el diagrama de flujo del manejo de la aplicación web creada.

The screenshot shows the E-Health application interface for Doctors. The interface includes a status bar at the top with the text 'E-Health' and a menu icon. Below the status bar, there are several monitoring parameters, each with an input field and a checkbox:

- Temp. Ambiente (°C): [input field]
- Temp. Corporal (°C): [input field] [checked checkbox]
- Ritmo cardiaco (LPM): [input field]
- Detector de caídas: [input field]
- Alarma temp. Corp.: [input field] [checked checkbox]
- Alarma ritmo: [input field] [checked checkbox]
- Prom. Ambiente: [input field]
- Prom. Tcorporal: [input field] [checked checkbox]

At the bottom of the interface, there are two buttons: 'CONECTADO' and 'DESCONECTADO'.

**Figura 67.** Estructura de datos para Usuario: Doctores.

The screenshot shows the E-Health application interface for Nurses. The interface includes a status bar at the top with the text 'E-Health' and a menu icon. Below the status bar, there are several monitoring parameters, each with an input field and a checkbox:

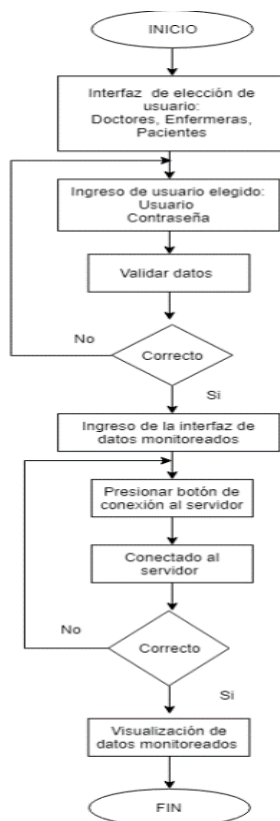
- Temp. Ambiente (°C): [input field]
- Temp. Corporal (°C): [input field] [checked checkbox]
- Ritmo cardiaco (LPM): [input field]
- Detector de caídas: [input field]
- Alarma temp. Corp.: [input field] [checked checkbox]
- Alarma ritmo: [input field] [checked checkbox]
- Cont. de caídas: [input field]
- Cont. Alarma tcorp: [input field]
- Cont. ritmo: [input field] [checked checkbox]
- Prom. Ambiente: [input field]
- Prom. Tcorporal: [input field]

At the bottom of the interface, there are two buttons: 'CONECTADO' and 'DESCONECTADO'.

**Figura 68.** Estructura de datos para Usuario: Enfermeras.

The screenshot shows a mobile application interface titled 'E-Health'. At the top, there is a status bar with signal strength, 94% battery, and the time 1:18 AM. Below the title, there are four input fields: 'Temp. Ambiente (°C):', 'Temp. Corporal (°C):', 'Ritmo cardiaco (LPM):', and 'Alarma de caídas:'. To the right of the 'Temp. Corporal' and 'Ritmo cardiaco' fields, there are square boxes containing an 'X' mark. At the bottom of the screen, there are two rectangular buttons: 'CONECTADO' on the left and 'DESCONECTADO' on the right.

**Figura 69.** Estructura de datos para Usuario: Pacientes.



**Figura 70.** Diagrama de flujo de uso de la aplicación móvil.

### 3.3.7 Base de Datos (BD)

Para el almacenamiento de los datos monitoreados se utilizó MySQL que es un sistema de administración de bases de datos de código abierto y se eligió esta BD porque es multiplataforma y puede trabajar el Sistema Operativo Linux que es utilizado para el Sistema de monitoreo.

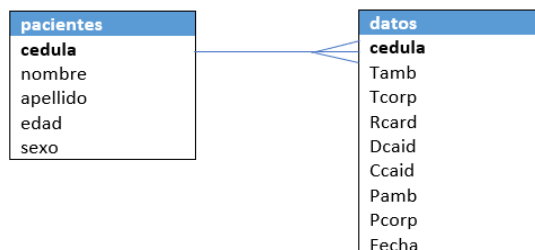
La Base de datos denominada “tesis” consta de las siguientes tablas: datos, pacientes, doctores, enfermeras, enfermo. El contenido de cada una se muestra en la Tabla 30.

**Tabla 30**  
*Contenido de cada tabla*

<b>Tabla</b>	<b>Contenido</b>
Datos	Se almacenan los datos monitoreados como temperatura ambiente, corporal, ritmo cardiaco, alarmas y los datos promedios de la temperatura ambiente y corporal.
Pacientes	Se almacena la información personal del paciente: nombre, apellido, edad, cédula, y sexo.
Doctores	Se almacena los nombres de los usuarios y las contraseñas para el inicio de sesión.
Enfermeras	Se almacena los nombres de los usuarios y las contraseñas para el inicio de sesión.
Enfermos	Se almacena los nombres de los usuarios y las contraseñas para el inicio de sesión.

En la Figura 71 se muestra la relación de las tablas, se utilizó la cédula como un identificador único que cada paciente posee, utilizando un modelo entidad-relación de uno a varios, siendo una base relacional, ya que por un paciente se tiene información como temperatura, ritmo cardiaco, detector de caídas, etc.





**Figura 71.** Relación de las tablas de la base de datos.

En las Tablas 31, 32 y 33 se detallan el tipo de dato que se ha utilizado en las variables de las tablas.

### Tabla 31

*Tipo de datos en la tabla usuarios*

Variable	Tipo de dato
usuario	varchar (256)
contraseña	varchar (256)

### Tabla 32

*Tipo de datos en la tabla pacientes*

Variable	Tipo de dato
<b>cédula</b>	int (10)
nombre	varchar (256)
apellido	varchar (256)
edad	int(10)
sexo	varchar (256)

### Tabla 33

*Tipo de datos en la tabla datos*




Variable	Tipo de dato
<b>cédula</b>	int (10)
Tamb	varchar (256)
Tcorp	varchar (256)
Rcard	varchar (256)

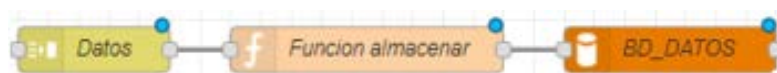
Dcaid	varchar (256)
Ccaid	varchar (256)
Pamb	varchar (256)
Pcorp	varchar (256)
Fecha	timestamp

Para el proceso de almacenamiento de los datos se ha utilizado Node-RED porque en esta herramienta previamente los datos han sido recolectados como se indicó en la Sección 3.3.5.4. La Tabla 34 muestra los diferentes nodos utilizados.

**Tabla 34**

*Descripción de los nodos para el almacenamiento de los datos*

Nodo	Propiedades del nodo
	El nodo “Split” permite convertir un solo mensaje en una secuencia de mensajes, en este caso todos los tópicos recibidos.
	El nodo “Function” permite realizar la inserción de los datos en la BD.
	El nodo “MySQL”, permite realizar la conexión con la BD.



**Figura 72.** Flujo de almacenamiento de datos.

### **3.4 Protocolo de pruebas**

Para la validación del dispositivo y los sensores que involucran el sistema de monitoreo son necesarias realizar diferentes pruebas. Estas pruebas son de carácter funcional para comprobar el funcionamiento y confiabilidad del dispositivo. Además, se comprueba el envío de datos, almacenamiento y visualización.

#### **Características de las pruebas de funcionamiento**

- Es indispensable que la red de conexión a WiFi no tenga bloqueados los puertos del Protocolo MQTT (1883 /8883).
- El servidor en la nube debe estar inicializado como se indicó en la Sección 3.3.3.1

#### **Característica de los sujetos de prueba**

En este caso se eligió sujetos de género masculino, pero este dispositivo puede ser utilizado para el género femenino también.

- El sujeto 1 de prueba es de género masculino, edad 25 años, no presenta ninguna enfermedad y realiza moderada actividad física, los datos muestreados serán utilizados en la evaluación del hardware.
- El sujeto 2 de prueba es de género masculino, edad 35 años, no presenta ninguna enfermedad y realiza baja actividad física, los datos muestreados serán utilizados en la evaluación del software.

#### **Ubicación del dispositivo e-Health en el sujeto de prueba**

- El dispositivo de e-Health se ubicará en el pecho del sujeto de prueba.

**Tiempo de monitoreo**

- Para el sistema de monitoreo el envío de datos se lo realiza cada treinta segundos con la finalidad que la ventana de muestreo sea la misma para cada sensor.

**Número de veces de repetición de la prueba**

- Se repetirá cada prueba diez veces, porque a mayor número de veces el error será más pequeño, caso contrario si solo se repetiría una vez la prueba el error sería del 100%.

**3.4.1 Funcionamiento de los sensores y el módulo detector de caídas.**

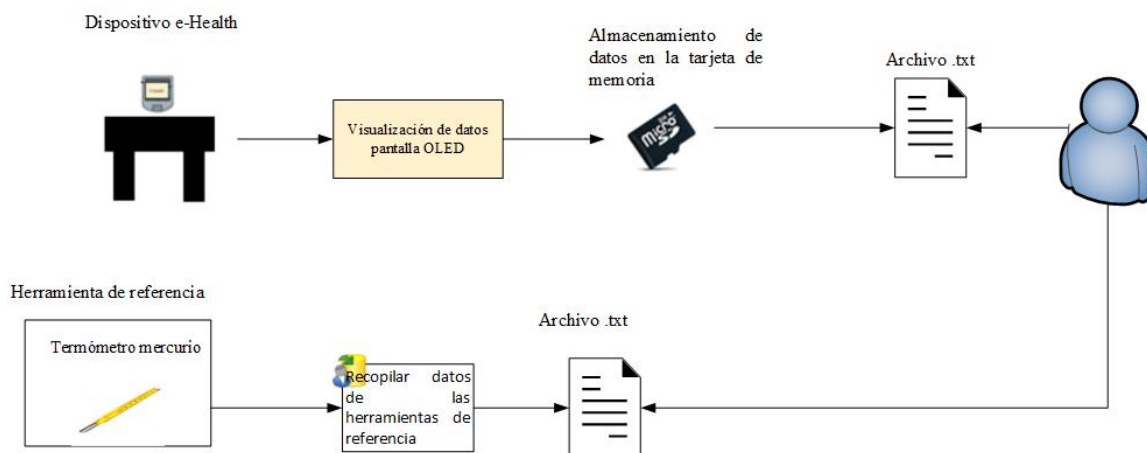
Esta prueba pretende la validación de los diferentes sensores y el módulo del detector de caídas. Se realizará cada prueba durante un periodo de 10 minutos ya que el tiempo de funcionamiento del dispositivo es de 1h 20 min, pero se debe considerar que el sensor de ritmo cardiaco no debe exceder este tiempo colocado en el sujeto de prueba porque disminuye el flujo sanguíneo (American Thoracic Society, 2013) y se obtendría falsos positivos en las pruebas por esta razón el tiempo no debe exceder 10 minutos.

**Parte A: Funcionamiento del sensor de temperatura ambiente.****Procedimiento:**

- Colocar el dispositivo de e-Health en un lugar estable y en ambiente interno.
- Encender el dispositivo de e-Health.
- Recolectar los datos monitoreados del sensor de temperatura ambiente durante un periodo de diez minutos.
- Recolectar la información de la herramienta de referencia, se ha elegido un termómetro de mercurio durante un periodo de diez minutos.
- Repetir la prueba diez veces para validar los resultados.

- Los datos monitoreados se almacenarán en la tarjeta de memoria que posee el dispositivo de e-Health.

En la Figura 73 se muestra el escenario de la Parte A.



**Figura 73.** Escenario de la Parte A

**Tabla 35**

*Tabla para temperatura ambiente*

	Referencia	LM35	Error
<b>Promedio de muestras</b>			

### Parte B: Funcionamiento del sensor de temperatura corporal.

Esta prueba pretende la validación del sensor de temperatura corporal con una herramienta de referencia.

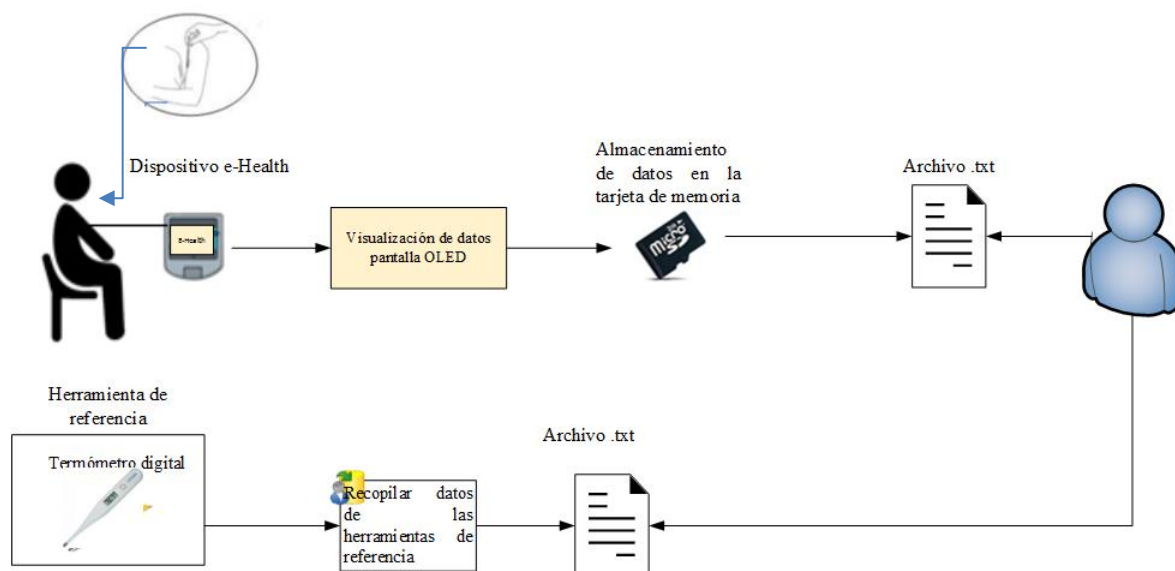
#### Procedimiento:

- Colocar el sensor de temperatura corporal en el sujeto de prueba 1. Coloque el extremo del encapsulado de color celeste debajo de la axila. Mantenga el brazo presionado contra su cuerpo.

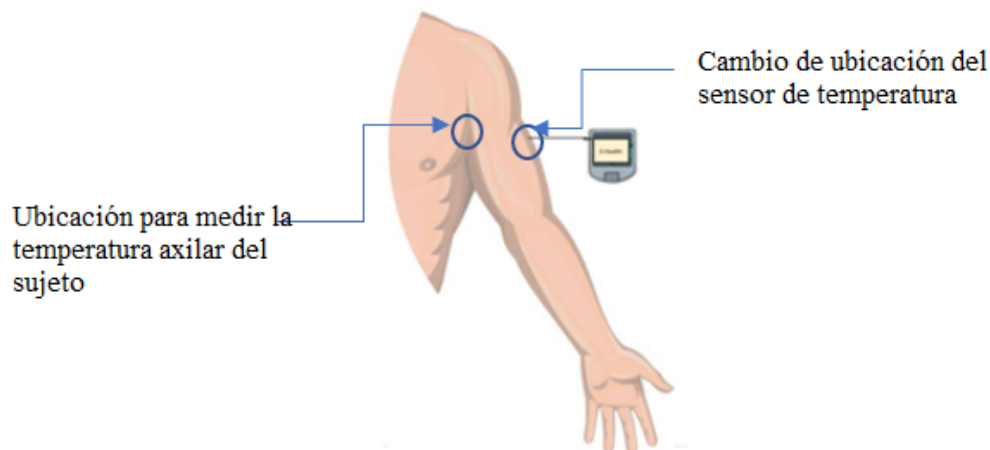
- Encender el dispositivo de e-Health.
- Recolectar los datos monitoreados del sensor de temperatura corporal durante un periodo de diez minutos.
- Recolectar la información de la herramienta de referencia, se ha elegido un termómetro digital durante un periodo de diez minutos.
- Repetir la prueba diez veces para validar los resultados.
- Los datos monitoreados se almacenarán en la tarjeta de memoria que posee el dispositivo de e-Health.
- Al terminar la prueba retirar el sensor de temperatura corporal sin tocar el encapsulado del sensor.

Se debe repetir todo el procedimiento anterior, pero en este caso se coloca el sensor de temperatura corporal en la parte externa del brazo, se adherirá el encapsulado del sensor por medio de una cinta adhesiva médica como se indica en la Figura 75 .

En la Figura 74 se muestra el escenario de la Parte B.



**Figura 74.** Escenario de la Parte B



**Figura 75.** Cambio de posición del sensor de temperatura corporal

**Tabla 36**

*Tabla para temperatura corporal*

	Referencia	Termistor	Error
Promedio de muestras			

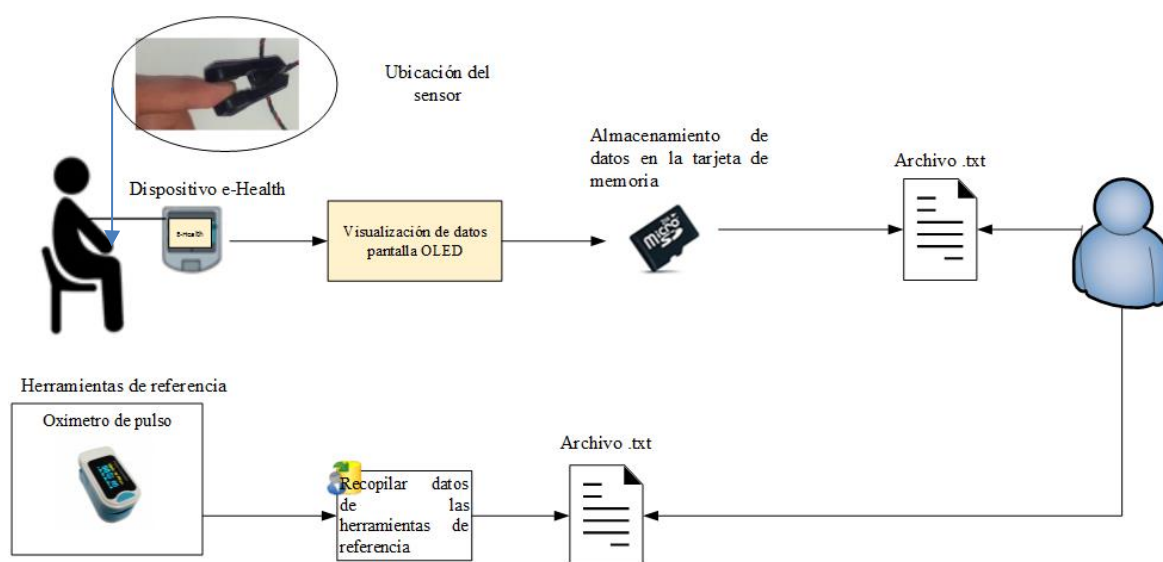
### Parte C: Funcionamiento del sensor de ritmo cardiaco.

#### Procedimiento:

- Colocar el sensor de ritmo cardiaco en el sujeto de prueba 1. Coloque el sensor en el dedo índice.
- Encender el dispositivo de e-Health.
- Recolectar los datos monitoreados del sensor de ritmo cardiaco durante un periodo de diez minutos.
- Recolectar la información de la herramienta de referencia, se ha elegido un oxímetro de pulso para el dedo durante un periodo de diez minutos.

- Repetir la prueba diez veces para validar los resultados.
- Los datos monitoreados se almacenarán en la tarjeta de memoria que posee el dispositivo de e-Health.
- Al terminar la prueba retirar el sensor de ritmo cardiaco.

En la Figura 76 se muestra el escenario de la Parte C.



**Figura 76.** Escenario de la Prueba C

**Tabla 37**

*Tabla para ritmo cardiaco*

Referencia	Sensor de pulso	Error
<b>Promedio de muestras</b>		

#### Parte D: Funcionamiento del módulo detector de caídas.

##### Procedimiento:

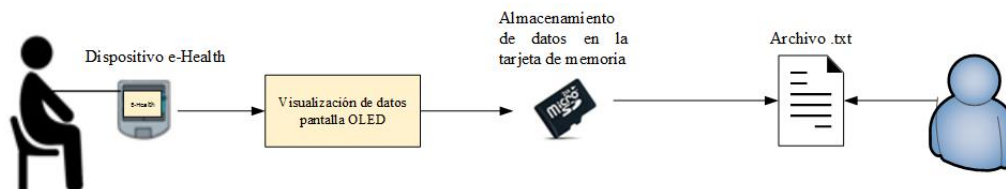
- Colocar el dispositivo de monitoreo en el sujeto de prueba 1. Coloque el dispositivo en el



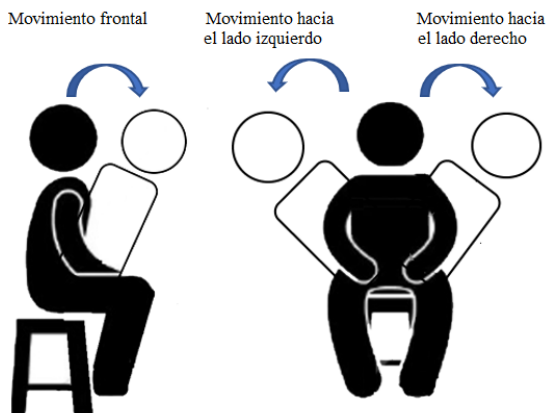
pecho del sujeto.

- Encender el dispositivo de e-Health.
- Simular caídas con movimientos frontales, hacia el lado derecho e izquierda del sujeto de prueba (ver Figura 78).
- Recolectar los datos monitoreados del módulo detector de caídas.
- Los datos monitoreados se almacenarán en la tarjeta de memoria que posee el dispositivo de e-Health.
- Al terminar la prueba retirar el dispositivo de e-Health y verificar mediciones.

En la Figura 77 se muestra el escenario de la Parte D.



**Figura 77.** Escenario de la Prueba D



**Figura 78.** Movimientos que debe realizar el sujeto de prueba.

NOTA: Con el fin de validar otros movimientos involuntarios se debe repetir todo procedimiento anterior, pero en este caso se puede simular movimientos similares a una convulsión, estiramientos, bostezos u otros.

### **3.4.2 Envío de datos mediante comunicación WiFi al Servidor en la nube y almacenamiento de datos en la Base de Datos.**

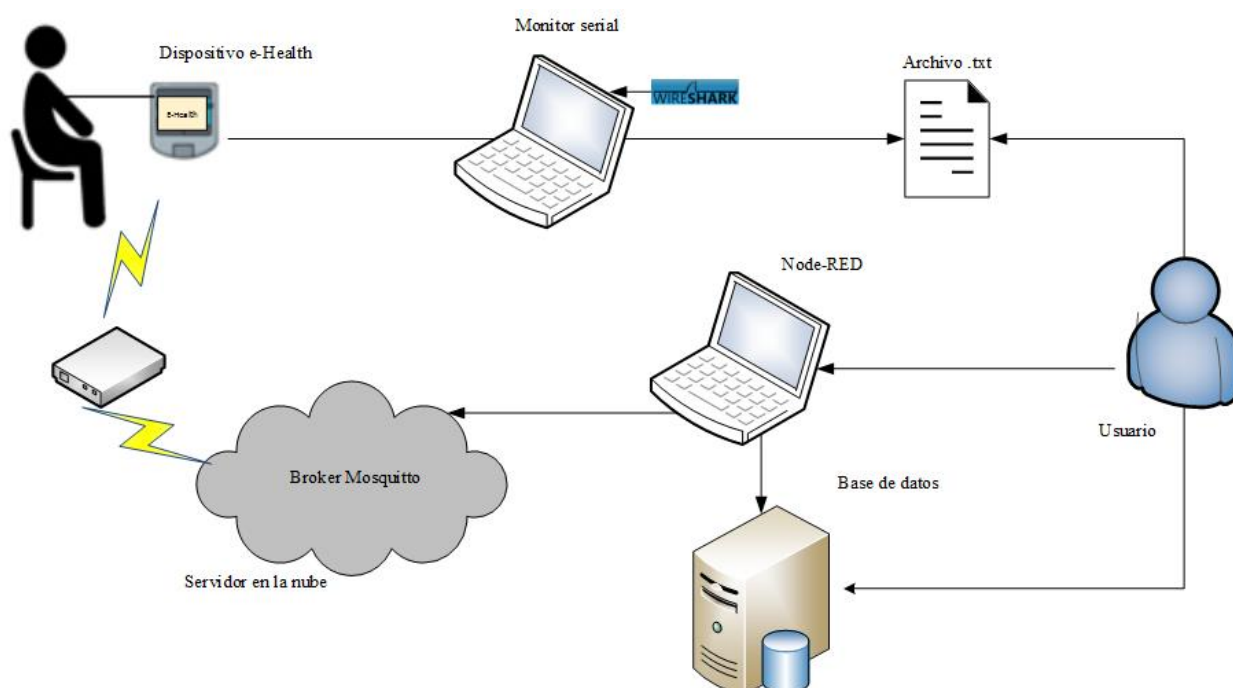
A través de esta prueba se valida los datos enviados desde el dispositivo e-Health al Servidor en la nube. Se utilizará una red de conexión WiFi estable y una red móvil de telefonía celular.

#### **Procedimiento:**

- Alimentar el dispositivo e-Health a través del cable USB al computador y habilitar la opción de monitor serial del IDE de Arduino.
- Verificar que el dispositivo tenga conexión a la red disponible mediante el led indicador que debe estar encendido y el mensaje que se muestra en la pantalla OLED “Conexión a WiFi”, en este caso la red será de conexión WiFi estable.
- Para comprobar la conexión con el Servidor en la nube se debe realizar un ping a la dirección IP 159.65.174.156 que es la dirección del Servidor, enviando paquetes de diferentes cantidades (100, 1000, 10000) y de diferente tamaño (10, 100 y 1000 bytes).
- Verificar envío de datos monitoreados en la opción “Debug messages” (ver Figura 37) en Node-RED.
- Recolectar los datos monitoreados durante un periodo de tiempo de 10 minutos.
- Repetir la prueba diez veces para validar los resultados.
- Analizar los datos recolectado a través de una gráfica elaborada en Matlab y concluir.

La Figura 79 muestra el escenario de la prueba 3.4.2.

NOTA: Con el fin de validar la conexión a WiFi en este caso se realizará una prueba utilizando una red móvil de telefonía celular como medio de acceso a los servidores, se debe repetir el procedimiento anterior de recolección de datos y obtener resultados.



**Figura 79.** Escenario de la Prueba 3.4.2

### Parte A: Verificación del protocolo de mensajería MQTT

A través de esta prueba se verifica que se utiliza el protocolo de mensajería MQTT mediante la herramienta de Wireshark.

- Durante la recolección de datos de la Prueba 3.4.2, en el computador se accede a la herramienta Wireshark que es un analizador de protocolos, que previamente ha sido instalado.
- Se filtra en la herramienta el protocolo MQTT.
- Se analiza los resultados obtenidos.

También se realizará la verificación de los paquetes de control para el protocolo de mensajería MQTT de la Tabla 2 a través de la herramienta MobaXterm.

- Durante la recolección de datos de la Prueba 3.4.2, en el computador se accede a la herramienta MobaXterm.
- En la consola se realizará la suscripción a un tópico específico a través del comando: `mosquitto_sub -d -t 'tamb'`.
- Se analiza los resultados obtenidos.

### **3.4.3 Envío de datos a la tarjeta de memoria**

A través de esta prueba se valida el almacenamiento de los datos monitoreados en la tarjeta de memoria en caso de que no esté disponible la conexión a la red.

#### **Procedimiento:**

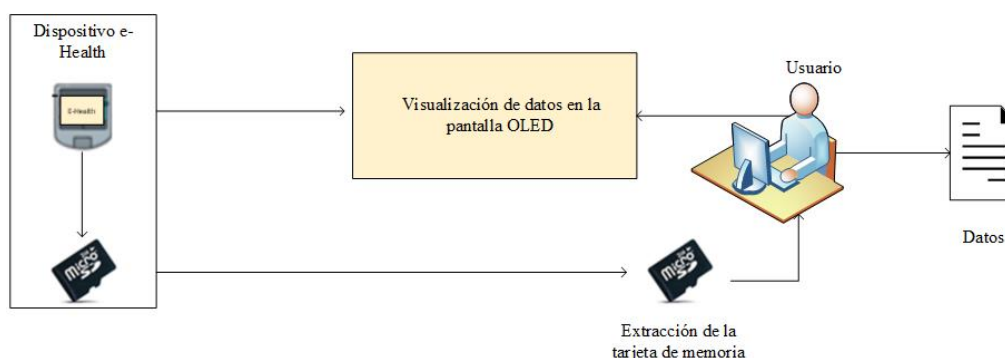
- Colocar la tarjeta de memoria en el computador por medio de su adaptador. Formatear la tarjeta para colocar en el dispositivo e-Health, esto se realizará solo la primera vez antes de crear un archivo.
- Alimentar el dispositivo e-Health a través del cable USB y cargar el programa desarrollo en Arduino, donde previamente se ha establecido el nombre del archivo que se desea crear.
- Colocar la tarjeta de memoria en el dispositivo e-Health.
- Encender el dispositivo y verificar que no se haya conectado a ninguna red mediante el led indicar de conexión debe estar apagado.
- Visualizar los datos monitoreados en la pantalla OLED.
- Recolectar los datos durante 10 minutos.
- Repetir la prueba diez veces para validar los resultados.

- Extraer la tarjeta de memoria del dispositivo y colocar en la computadora para verificar la existencia del archivo con extensión .txt
- Analizar los datos recolectados, en la Figura 80 se muestra el formato de la tabla y concluir.

	Fecha	Hora	Tamb	Tcorp	Ritmo	Deaidas	Ccaid	Atemp	Aritmo	Ccaidas	Ctemp	Critmo	Ptamb	Ptcorp
Datos														

**Figura 80.** Formato de la tabla “datos”

La Figura 81 muestra el escenario de la Prueba 3.4.3



**Figura 81.** Escenario de la Prueba 3.4.3

### 3.4.4 Funcionalidad del Sistema de monitoreo.

Prueba de validación del sistema de monitoreo de forma total, la prueba se realizará en un periodo de 1h 20 min porque es el tiempo del correcto funcionamiento del dispositivo.

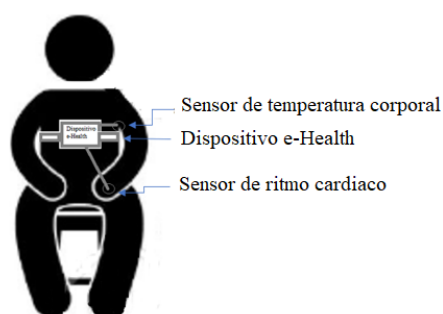
#### Procedimiento:

- Colocar el dispositivo e-Health en el sujeto de prueba 2, cada uno de sus sensores en las siguientes partes del cuerpo (ver Figura 82 ):

Sensor de temperatura corporal debajo de la axila del sujeto y su brazo debe apoyarlo contra su cuerpo para evitar que el dispositivo se mueva.

Sensor de ritmo cardiaco en el dedo índice y su mano apoyado sobre sus piernas.

El sujeto debe estar sentado en posición erguida y no puede realizar ningún movimiento brusco.



**Figura 82.** Posición del sujeto de prueba 3.4.4

- Encender el dispositivo, verificar que tenga conexión a la red disponible mediante el led indicador que debe estar encendido y el mensaje que se muestra en la pantalla OLED “Conexión a WiFi”.
- Verificar los datos monitoreados en la aplicación multiplataforma:

*Página web:*

Ingresa en el navegador con la dirección <http://159.65.174.156:1880/ui/> elegir el usuario al cual tiene acceso.

En la ventana de inicio de sesión colocar el usuario y contraseña.

En la página que se muestra después del inicio de sesión se podrá visualizar los datos monitoreados cada 30 segundos.

*Aplicación móvil:*

En el celular previamente se debe haber instalado la aplicación e-Health por medio de la herramienta Android Studio.

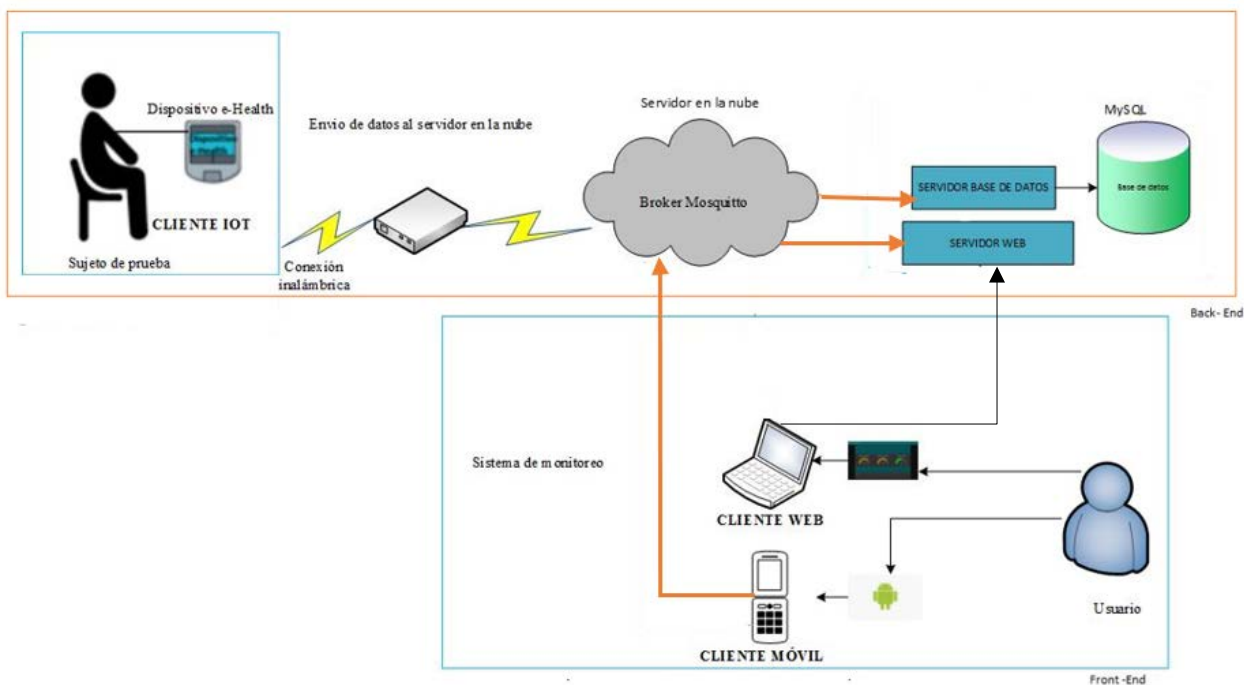
Presionar en el icono de e-Health y se mostrará una interfaz para elección del usuario.

En la interfaz de inicio de sesión colocar el usuario y contraseña.

En la ventana que se muestra después del inicio de sesión se podrá visualizar los datos monitoreados cada 30 segundos.

Visualizar, graficar y comprobar la funcionalidad del dispositivo de e-Health.

La Figura 83 muestra el escenario de la Prueba 3.4.4.



*Figura 83.* Escenario de la prueba del Sistema de monitoreo.

### 3.4.5 Prueba de carga y descarga de la batería del dispositivo e-Health

Mediante esta prueba se determinará el tiempo de carga y descarga de la batería del dispositivo.

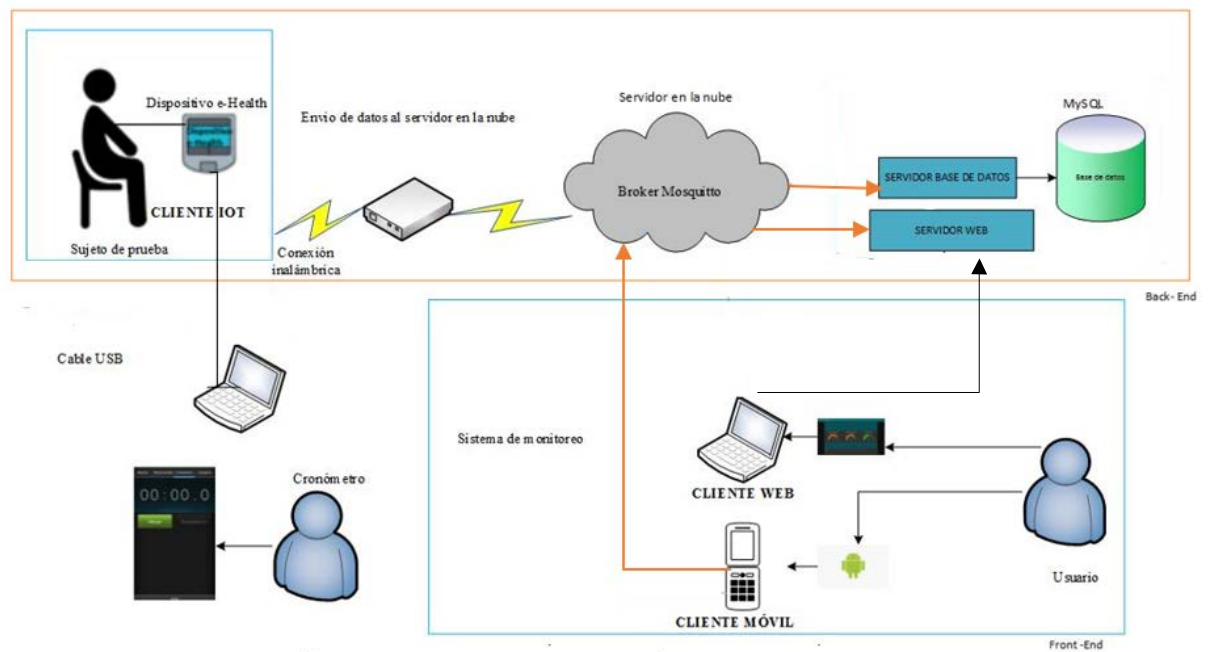
#### Parte A: Carga de la batería del dispositivo e-Health

##### Procedimiento:

- Colocar el dispositivo e-Health en el sujeto de prueba (ver Figura 82 ).

- Alimentar el dispositivo e-Health a través del cable USB al computador y el interruptor encendido / apagado, debe estar en la opción encendido.
- Utilizar un cronómetro, esto permitirá determinar el tiempo de carga del dispositivo, para verificar que la batería está totalmente cargada se mide el voltaje con un multímetro el cual debe ser de 3.7 V.
- Verificar los datos monitoreados tanto en la aplicación web y móvil.
- Detener el cronómetro cuando la batería indique aproximadamente 3.7 V.
- Desconectar el cable USB del dispositivo e-Health.
- Apagar el dispositivo.
- Analizar los datos obtenidos

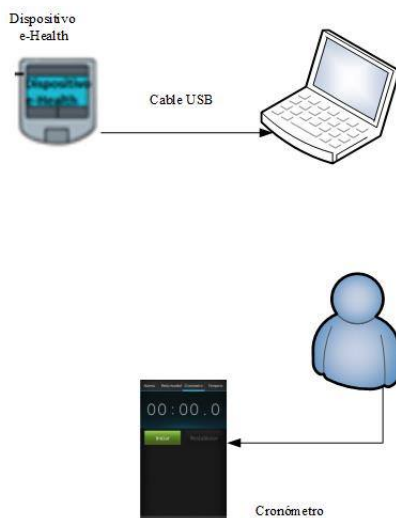
La Figura 84 muestra el escenario de Parte A.



**Figura 84.** Escenario de la Parte A.



NOTA: Con el fin de verificar el tiempo de carga del dispositivo se repite el procedimiento anterior, pero no se colocará el dispositivo en el sujeto de prueba (ver Figura 85).



*Figura 85.* Modificación de la Parte A.

### **Parte B: Descarga de la batería del dispositivo e-Health**

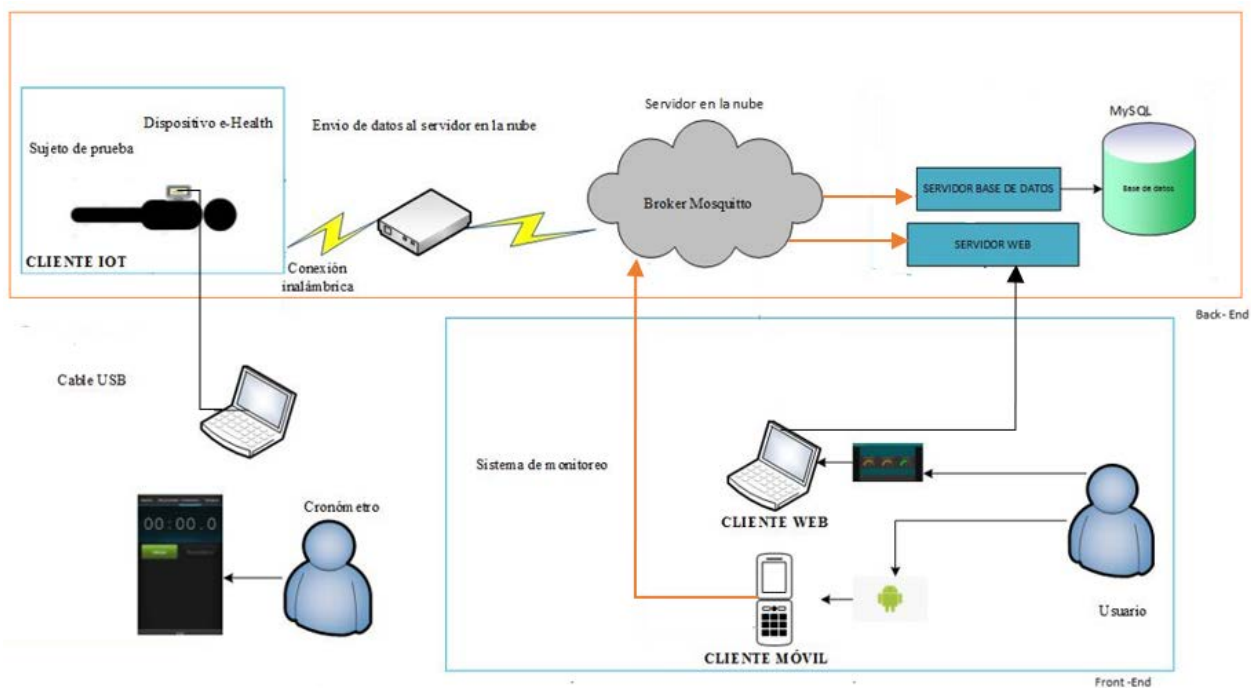
Mediante esta prueba se determina el tiempo de descarga de la batería, en este caso el sujeto de prueba se encuentra en posición horizontal y/o dormido, la posición del sujeto será como se muestra en la Figura 86.

#### **Procedimiento:**

- Colocar el dispositivo e-Health en el sujeto de prueba.
- Encender el dispositivo, verificar que tenga conexión a la red disponible mediante el led indicador que debe estar encendido y el mensaje que se muestra en la pantalla OLED “Conexión a WiFi”.
- Utilizar un cronómetro, esto permitirá saber con exactitud el tiempo de descarga del dispositivo.

- Verificar los datos monitoreados tanto en la aplicación web y móvil.
- Para verificar si el dispositivo e-Health se encuentra descargado se enciende un LED indicador de color amarillo, esto indica que el voltaje de la batería es inferior a 2.9 V.
- Detener el cronómetro.
- Apagar el dispositivo.
- Analizar los datos obtenidos.

La Figura 86 muestra el escenario de la Prueba B.



**Figura 86.** Escenario de la Parte B.

## **CAPÍTULO 4**

### **ANÁLISIS DE RESULTADOS**

#### **4.1 Evaluación del hardware**

Para evaluar el sistema, en aspectos de Hardware es indispensable realizar varias pruebas de los sensores del dispositivo de e-Health y con los resultados obtenidos realizar un análisis para determinar la precisión y la confiabilidad de los datos. Para evaluar el sistema se utilizaron las pruebas definidas en el Capítulo 3.

##### **4.1.1 Resultados: “Funcionamiento de los sensores y el módulo detector de caídas”**

Las pruebas de funcionamiento se realizaron durante un periodo de 10 minutos y esta se repite 10 veces porque a mayor número de repeticiones en las pruebas el error de las mediciones será más pequeño permitiendo validar los resultados de los sensores de temperatura ambiente, temperatura corporal, ritmo cardiaco y el módulo detector de caídas.

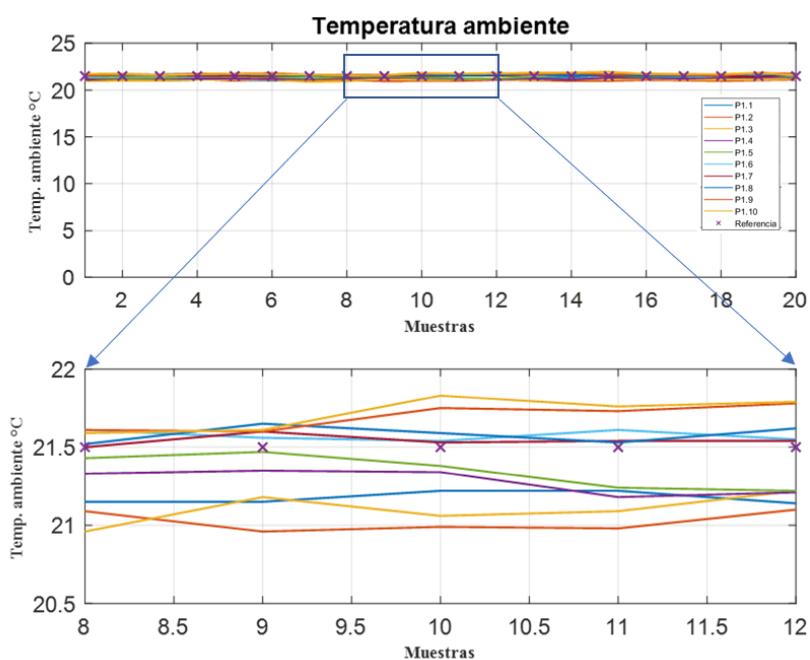
Para el sistema de monitoreo él envió de datos se lo realiza cada treinta segundos con la finalidad que la ventana de muestreo sea la misma para cada sensor. Considerando que la prueba será en un periodo de 10 minutos se obtendrá un total de 20 muestras por cada prueba.

##### **Resultados parte A: Análisis del sensor de temperatura ambiente**

Durante esta prueba se analizó los resultados del sensado de temperatura ambiente obtenidos y así realizar la validación de los datos del dispositivo de e-Health. En base al procedimiento descrito en el protocolo de pruebas, punto 3.4.1. La Figura 87 muestra la gráfica de temperatura ambiente.

En la Tabla 38 se muestran los valores promedio de la temperatura ambiente del dispositivo de referencia y el sensor LM35. Como dispositivo de referencia se ha considerado un termómetro de

mercurio y durante el periodo de prueba de funcionamiento del sensor se detecto una temperatura ambiente de 21.45°C aproximadamente tanto para la referencia, como para el LM35. Con estos datos se obtuvo un error de 0.41%, para obtener este error se realizó una compensación por software de 0.04 °C logrando ajustar el valor del sensor a la referencia.



**Figura 87.** Salida del sensor de temperatura ambiente

**Tabla 38**

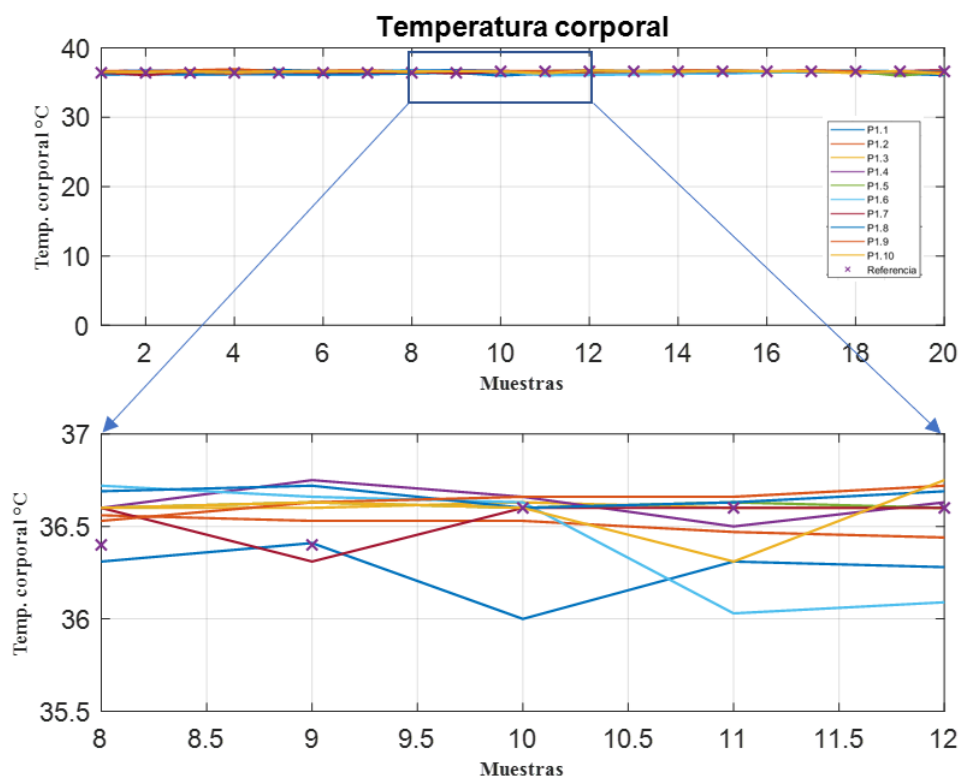
*Sensor de temperatura ambiente*

Promedio	Referencia	LM35	Error antes compensación	Error después compensación
Muestras	21.5 °C	21.41°C	0.68%	0.41%

### Resultados parte B: Análisis del sensor de temperatura corporal

Al probar el otro sensor de temperatura se analiza los resultados obtenidos de temperatura corporal y así realizar la validación de los datos del dispositivo de e-Health. Para ello se utiliza un

termómetro digital con precisión de  $\pm 0.1^{\circ}\text{C}$  como instrumento de comparación. La Figura 88 muestra la gráfica de temperatura corporal.



**Figura 88.** Salida del sensor de temperatura corporal

**Tabla 39**

*Sensor de temperatura corporal*

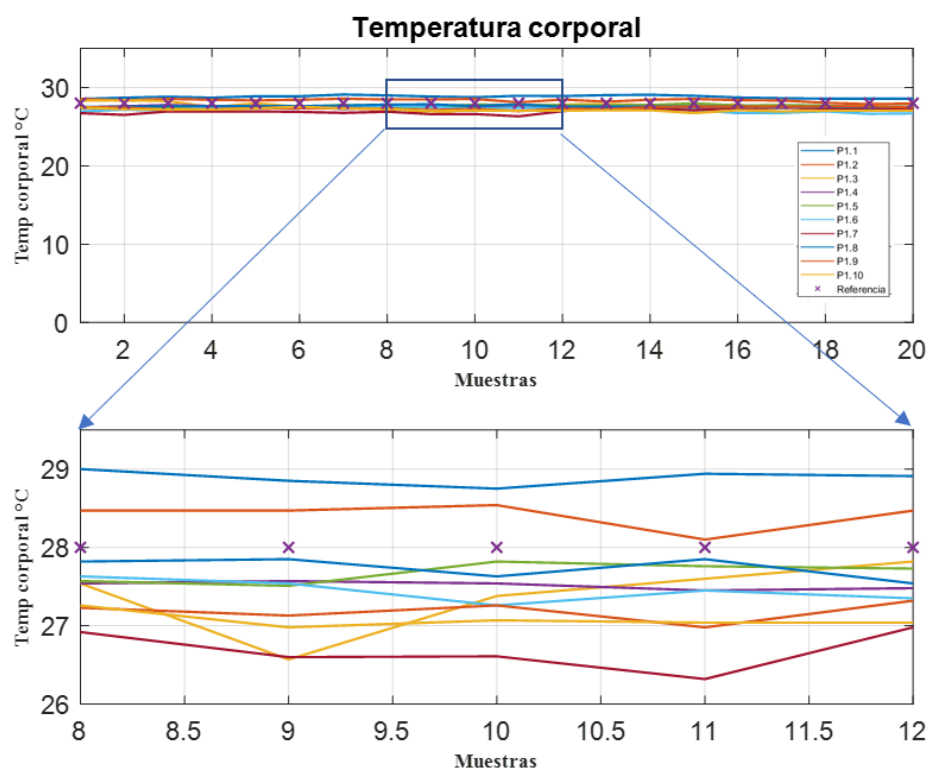
Promedio	Referencia	Termistor	Error antes compensación	Error después compensación
Muestras	36.51 °C	36.54°C	1.39%	0.08%

En la Tabla 39 se muestran los valores promedio de la temperatura corporal una vez compensado el sensor por software. Durante el periodo de prueba de funcionamiento del sensor se detectó una temperatura corporal de  $36.52^{\circ}\text{C}$  aproximadamente, tanto para la referencia como para el Termistor encontrándose un error de 0.08%.

Para asegurar el funcionamiento del dispositivo se realizó la prueba ya planteada, pero cambiando la posición del sensor de temperatura corporal de la axila al brazo como se indica en la

Figura 75, obteniendo datos distintos a los esperados y por lo tanto erróneos ya que la temperatura corporal de una persona varía entre 35 °C y 37°C y en este nuevo punto externo se ha registrado datos de 28°C aproximadamente, de donde se obtuvo un error de 1.39% con respecto a la referencia como se muestra en la Tabla 40.

La cual indica que la posición del sensor es importante para medir la temperatura corporal, en este caso la axila genera una medición de temperatura corporal más precisa que en otro punto sobre la piel.



**Figura 89.** Salida del sensor de temperatura corporal

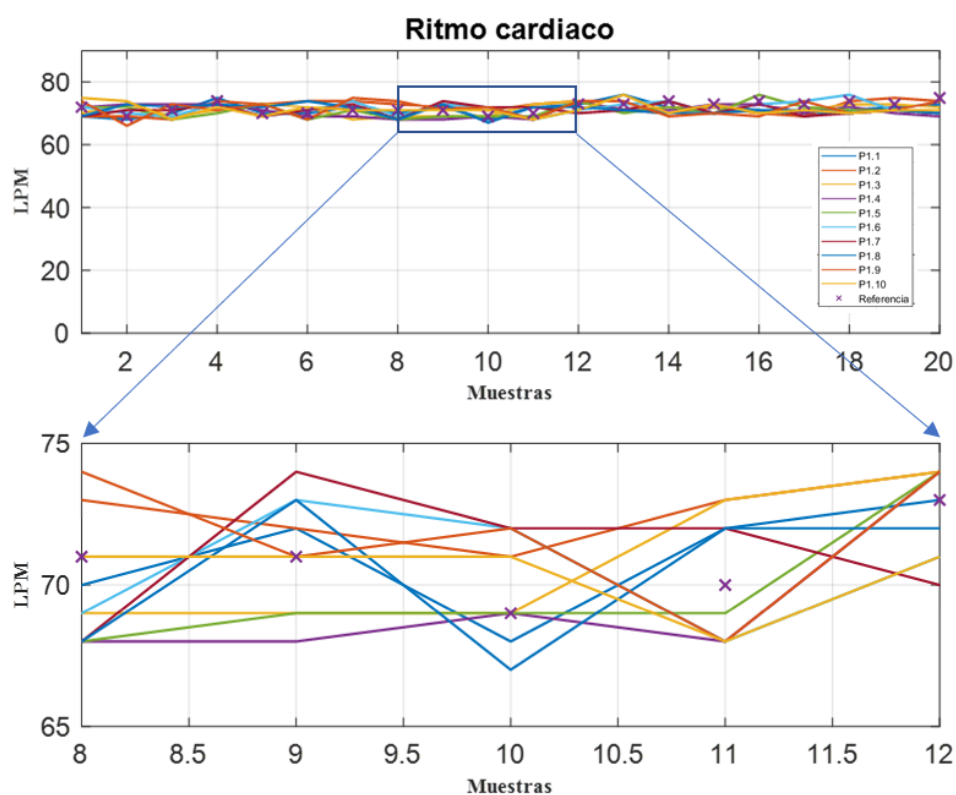
**Tabla 40**

*Sensor de temperatura corporal*

Promedio	Referencia	Termistor	Error
Muestras	27.61 °C	28°C	1.39%

### Resultados parte C: Análisis del sensor de ritmo cardiaco

Durante esta prueba se analiza los resultados del sensado de ritmo cardiaco obtenidos y así realizar la validación de los datos del dispositivo de e-Health. En base al procedimiento descrito en el protocolo de pruebas del punto 3.4.1. Esta prueba se realizó en el sujeto de prueba 1 y el sensor estaba ubicado en el dedo índice. La Figura 90 muestra la gráfica del valor instantáneo del ritmo cardiaco en el tiempo.



**Figura 90.** Evolución de salida del sensor de ritmo cardiaco

La Tabla 41 muestra los valores promedios del ritmo cardiaco. Se ha considerado un oxímetro de pulso como instrumento de referencia que posee una precisión de  $\pm 2$  Latidos Por Minuto (LPM). Durante el periodo de prueba de funcionamiento del sensor se detectó un ritmo cardiaco promedio de 72 LPM aproximadamente tanto para el dispositivo de referencia como para el sensor de

ritmo cardiaco, de donde se obtuvo un error de 1.38% que representa 1 Latido Por Minuto (LPM) respecto a la referencia.

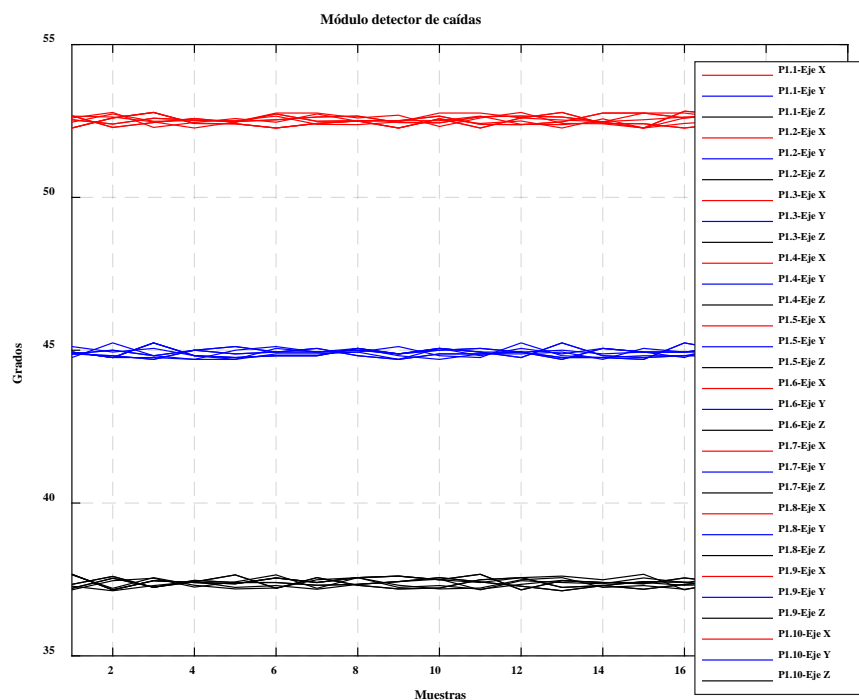
**Tabla 41**

*Sensor de ritmo cardiaco*

Promedio	Referencia	HRM-2511B	Error antes compensación	Error después compensación
Muestras	71 LPM	72 LPM	2.77%	1.38%

### Resultados parte D: Análisis del módulo detector de caídas

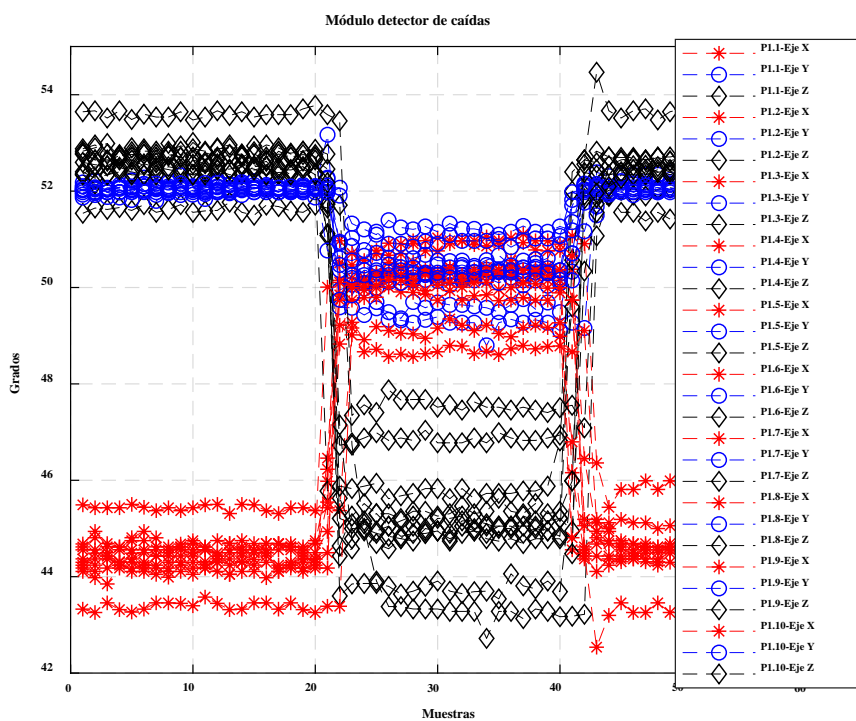
Durante esta prueba se analiza los resultados obtenidos del módulo detector de caídas, así con el objetivo de realizar la validación de los datos del dispositivo de e-Health. La Figura 91 muestra la gráfica del módulo detector de caídas en una sola posición estable en este caso se ha tomado 20 muestras, y se observa que los ejes permanecen sin cambios.



**Figura 91.** Ejemplo prueba del detector de caída



La simulación de caídas hacia al frente, hacia el lado izquierdo y el lado derecho pueden verse a continuación. En la Figura 92 se muestra una caída hacia al frente del paciente donde se observa el cambio de ejes, se puede verificar que existen cambios notables tanto en el eje X como en el eje Z, mientras que el eje Y su diferencia es tan solo  $2^\circ$ . La Tabla 42 indica el cambio de los ejes de una posición inicial que es representada cuando el sujeto está en posición fija y una posición final cuando sufre una caída.

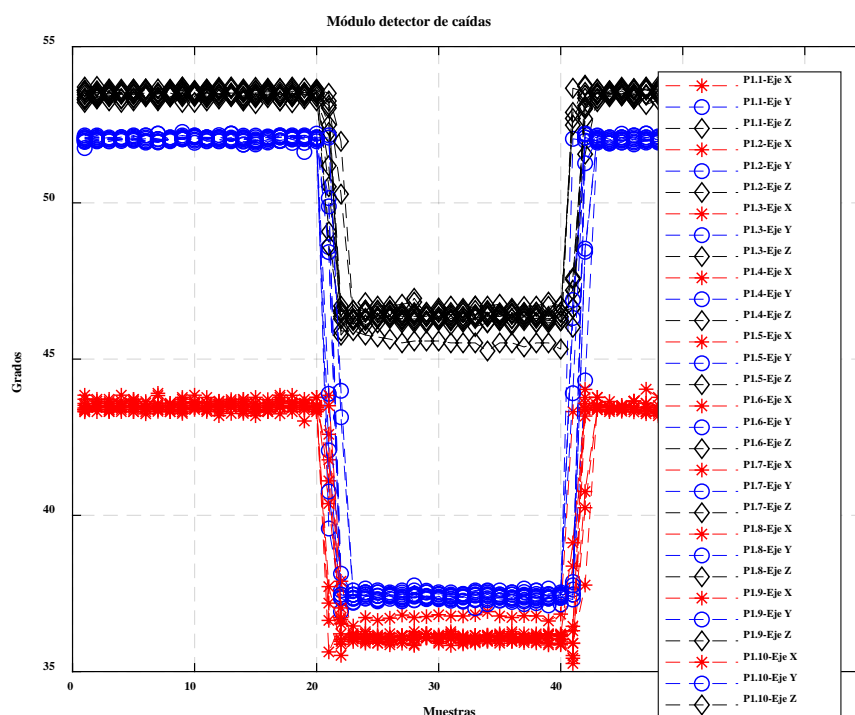


**Figura 92.** Ejemplo prueba con caída hacia al frente.

**Tabla 42**  
*Ejemplo caída frontal*

Movimiento	Eje X (Grados)	Eje Y (Grados)	Eje Z (Grados)
Posición inicial	44	52	53
Posición final	50	50	45

En la Figura 93 se muestra una caída hacia el lado derecho del paciente y los datos de la Tabla 43 indican el cambio de los ejes, se puede verificar que los tres ejes presentan cambios, pero el eje Y ha cambiado notablemente con una diferencia de 15 °.



**Figura 93.** Ejemplo prueba de caída hacia el lado derecho.

**Tabla 43**

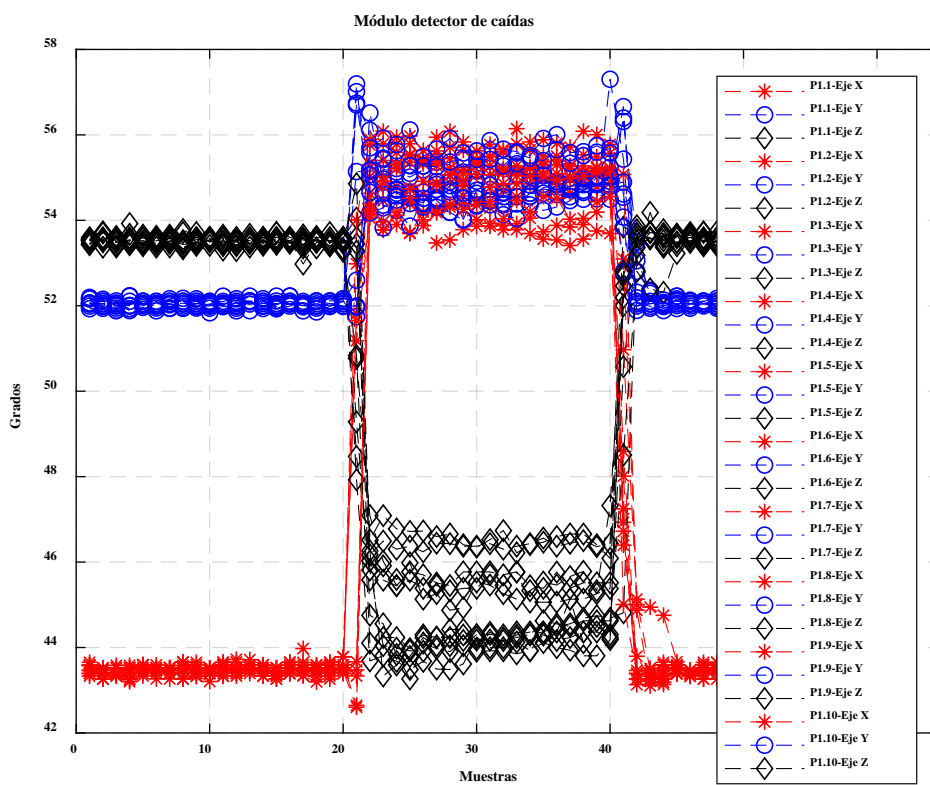
*Ejemplo caída hacia la izquierda*

Movimiento	Eje X (Grados)	Eje Y (Grados)	Eje Z (Grados)
Posición inicial	44	52	53
Posición final	36	37	46

En la Figura 94 se muestra una caída hacia el lado izquierdo del paciente y la Tabla 44 indica el cambio de los ejes. Se puede verificar que el eje X y Z presentan cambios repentinos mientras que el eje Y tiene una diferencia de 4°.

**Tabla 44***Ejemplo caída hacia la izquierda*

Movimiento	Eje X (Grados)	Eje Y (Grados)	Eje Z (Grados)
Posición inicial	44	52	53
Posición final	55	56	43

**Figura 94.** Ejemplo prueba de caída hacia el lado izquierdo.

Para otra validación del módulo detector de caída, se ha simulado que el paciente sufra convulsiones, las cuales no deben ser reconocidas como caídas. En la Figura 95 se muestra los cambios de los ejes consecutivamente sin embargo esa variación no es considerada como caída según las validaciones del software.



Reposo	44	52	53	44	52	53
Frontal	49	53	53	51	49	48
Derecha	36	38	38	36	37	46
Izquierda	48	56	56	54	53	47

## 4.2 Evaluación del software

Para evaluar el sistema, en aspectos de Software es indispensable realizar varias pruebas de conexión con el Servidor en la nube y la aplicación multiplataforma (web y móvil) para realizar un análisis de resultados obtenidos para determinar la precisión y la confiabilidad del dispositivo de e-Health para el Sistema de monitoreo remoto.

### 4.2.1 Resultados: “Envío de datos mediante comunicación WiFi al Servidor en la Nube y almacenamiento de datos en la Base de Datos”

Un aspecto importante es la conexión entre el dispositivo e-Health y el Servidor en la Nube para ello es indispensable validar la comunicación WiFi entre ellos, además se verificará que los datos monitoreados se han enviado al servidor al obtener la información de la Base de Datos.

Antes de realizar la prueba se debe verificar si existe conexión a WiFi del dispositivo a través del mensaje indicador en la pantalla OLED, así como el LED indicador de color verde.

Para comprobar la conexión del Servidor en la nube se ha utilizado el protocolo de control de mensajes de Internet (ICMP), se envió un ping a la dirección IP 159.65.174.156 que es la dirección del Servidor en la nube, en el que se realizó el envío de diferentes cantidades de paquetes (100, 1000 y 10000) para verificar si existe pérdida de paquetes.

En las Figuras 94 y 95 muestran si existe pérdida de paquetes y se obtuvo una confiabilidad del 100%, pero en la Figura 96 se obtuvo una confiabilidad del 99% porque existe 134 paquetes perdidos. Además, en la Tabla 46 se indica el tiempo mínimo y máximo del envío de paquetes.

```

Estadísticas de ping para 159.65.174.156:
 Paquetes: enviados = 100, recibidos = 100, perdidos = 0
 (0% perdidos),
 Tiempos aproximados de ida y vuelta en milisegundos:
 Mínimo = 108ms, Máximo = 133ms, Media = 111ms

```

**Figura 96.**ICMP enviado al servidor, 100 paquetes

```

Estadísticas de ping para 159.65.174.156:
 Paquetes: enviados = 1000, recibidos = 1000, perdidos = 0
 (0% perdidos),
 Tiempos aproximados de ida y vuelta en milisegundos:
 Mínimo = 108ms, Máximo = 189ms, Media = 110ms

```

**Figura 97.**ICMP enviado al servidor, 1000 paquetes

```

Estadísticas de ping para 159.65.174.156:
 Paquetes: enviados = 10000, recibidos = 9866, perdidos = 134
 (1% perdidos),
 Tiempos aproximados de ida y vuelta en milisegundos:
 Mínimo = 108ms, Máximo = 504ms, Media = 110ms

```

**Figura 98.**ICMP enviado al servidor, 10000 paquetes

**Tabla 46**

*Envió de paquetes al Servidor en la nube*

Cantidad de paquetes	Tiempo mínimo (ms)	Tiempo máximo (ms)	Pérdida de paquetes (%)
100	108	113	0
1000	108	189	0
10000	108	504	1

También se realizó una prueba con él envió de paquetes de diferente tamaño de 10,100 y 1000 bytes. El tamaño puede estar definido entre 32 a 65.527 siendo 32 el valor por defecto.

En la prueba realizada no existe pérdida de paquetes, pero el tiempo aproximado de ida y vuelta aumenta cuando el paquete es de mayor tamaño en bytes, esto se muestra en la Tabla 47

```

Estadísticas de ping para 159.65.174.156:
 Paquetes: enviados = 100, recibidos = 100, perdidos = 0
 (0% perdidos),
 Tiempos aproximados de ida y vuelta en milisegundos:
 Mínimo = 82ms, Máximo = 130ms, Media = 83ms

```

**Figura 99.**ICMP enviado al servidor, tamaño del paquete 10 bytes

```

Estadísticas de ping para 159.65.174.156:
Paquetes: enviados = 100, recibidos = 100, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 82ms, Máximo = 137ms, Media = 84ms

```

**Figura 100.**ICMP enviado al servidor, tamaño del paquete 100 bytes

```

Estadísticas de ping para 159.65.174.156:
Paquetes: enviados = 100, recibidos = 100, perdidos = 0
(0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
Mínimo = 84ms, Máximo = 140ms, Media = 85ms

```

**Figura 101.**ICMP enviado al servidor, tamaño del paquete 1000 bytes

**Tabla 47**

*Envío de paquetes al Servidor en la nube (diferente tamaño)*

Tamaño del paquete (bytes)	Tiempo mínimo (ms)	Tiempo máximo (ms)	Pérdida de paquetes (%)
10	82	130	0
100	82	137	0
1000	84	140	0

Continuando con la prueba se verificó el envío de datos en las Figuras 97, 98 y 99, a través del monitor serial del IDE de Arduino, el “Debug messages” de la herramienta Node-RED y de la Base de datos que representan los datos monitoreados.

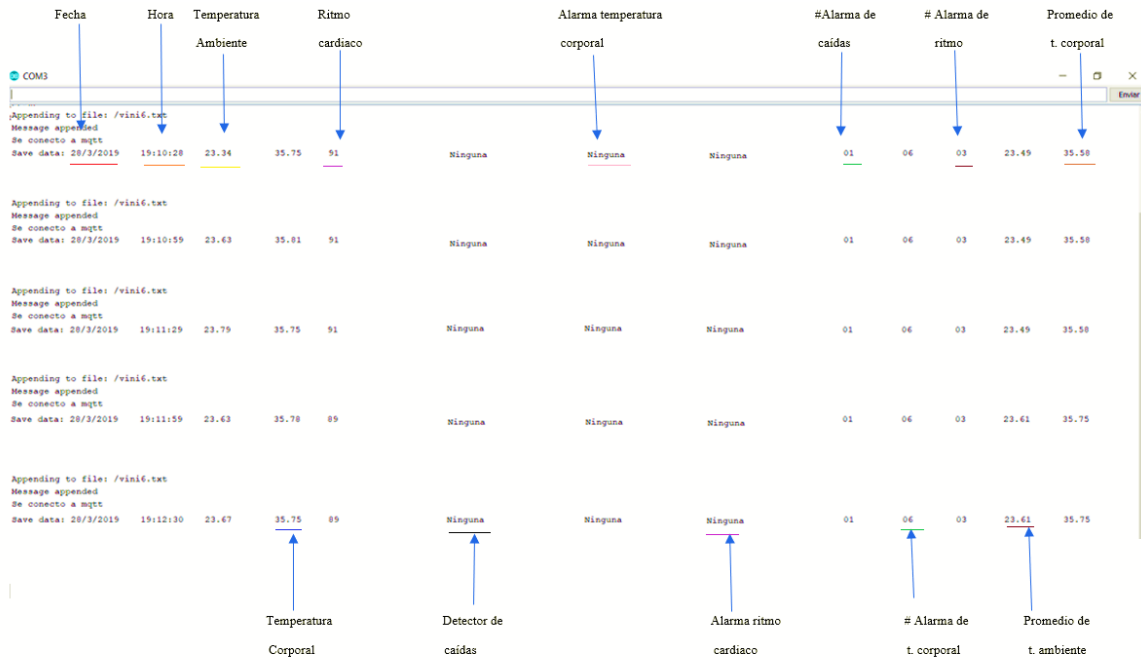


Figura 102. Datos monitor serial

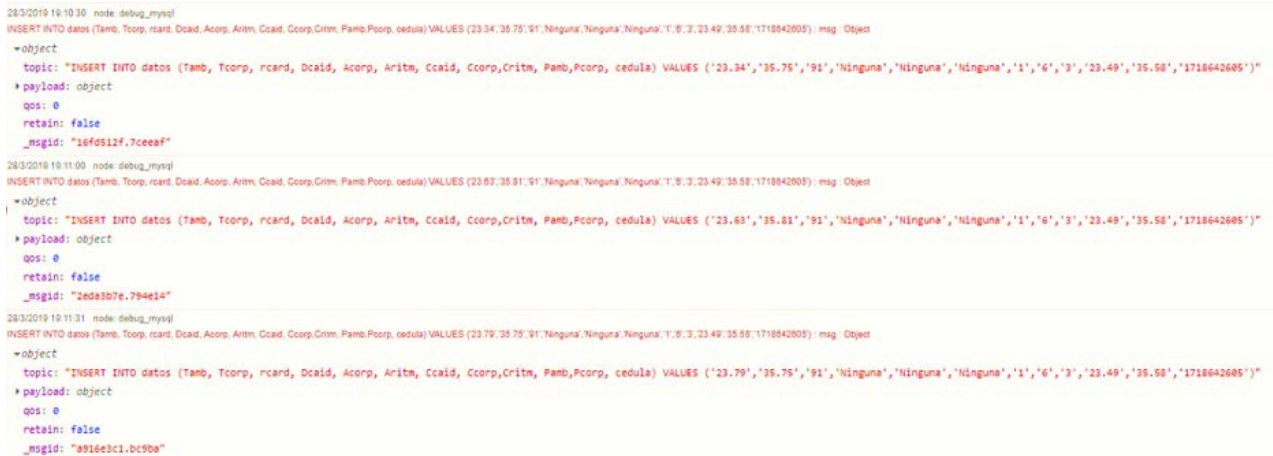


Figura 103. Datos en el “Debug messages”.



Fecha/Hora	Temperatura corporal	Detector caída	Alarma ritmo cardiaco	#Alarma de t. corporal	Promedio t.ambiente	Cédula
2019-03-28 19:10:30	23.34	35.75   91	Ninguna	Ninguna	Ninguna	1   6   3   23.49   35.58   17186
2019-03-28 19:11:00	23.63	35.81   91	Ninguna	Ninguna	Ninguna	1   6   3   23.49   35.58   17186
2019-03-28 19:11:30	23.79	35.75   91	Ninguna	Ninguna	Ninguna	1   6   3   23.49   35.58   17186
2019-03-28 19:12:00	23.63	35.78   89	Ninguna	Ninguna	Ninguna	1   6   3   23.61   35.75   17186
2019-03-28 19:12:31	23.67	35.75   89	Ninguna	Ninguna	Ninguna	1   6   3   23.61   35.75   17186

Labels below the table: Temperatura Corporal, Ritmo cardiaco, Alarma t. corporal, #Alarma de caídas, #Alarma de ritmo, Promedio t. corporal

**Figura 104.**Datos en la Base de Datos.

Además se ha realizado una prueba utilizando una red móvil de telefonía celular como medio de acceso a internet, utilizando un teléfono como punto de acceso inalámbrico, donde los resultados obtenidos demostraron, por ejemplo que una conexión estable de buen nivel de señal no pierde la conexión, en este caso la conexión red móvil después de 37 minutos provocó que los datos no se enviaran al Servidor en la nube y como consecuencia no se almacenaran en la BD, pero el dispositivo e-Health dispone de una tarjeta de memoria en la cual los datos se almacenan durante el tiempo de funcionamiento del dispositivo. La prueba inicio a las 19:07:42 y se perdio la conexión WiFi a las 19:34:23 (Figura 105), pero el almacenamiento en la tarjeta se realizó hasta las 20:08:06 (Figura 106).

Inicio de almacenamiento de datos

2019-06-05 19:07:42	19.62	36.28   97	Ninguna	Ninguna	Ninguna	0   0   0   19.53   36.27   10007
2019-06-05 19:08:04	19.45	36.25   98	Ninguna	Ninguna	Ninguna	0   0   0   19.53   36.27   10007
...						
2019-06-05 19:33:23	20.55	36.25   90	Ninguna	Ninguna	Ninguna	0   0   0   20.47   36.25   10007
2019-06-05 19:33:53	20.44	36.28   94	Ninguna	Ninguna	Ninguna	0   0   0   20.47   36.25   10007
2019-06-05 19:34:23	20.44	36.19   91	Ninguna	Ninguna	Ninguna	0   0   0   20.47   36.25   10007

Fin de almacenamiento de datos

**Figura 105.**Datos en la Base de Datos, cambio de red WiFi

Inicio de almacenamiento de datos

5/6/2019	19:07:40	19.62	36.28	97	Ninguna	Ninguna	Ninguna	0	0	0	19.53	36.27
5/6/2019	19:08:02	19.45	36.25	98	Ninguna	Ninguna	Ninguna	0	0	0	19.53	36.27
...												
...												
5/6/2019	20:07:46	20.30	36.19	89	Ninguna	Ninguna	Ninguna	0	0	0	20.29	36.19
5/6/2019	20:08:06	20.35	36.28	92	Ninguna	Ninguna	Ninguna	0	0	0	20.29	36.19

Fin de almacenamiento de datos

**Figura 106.**Datos en la SD card, cambio de red WiFi

Cabe señalar que en el caso de utilizar un teléfono celular como punto de acceso WiFi el funcionamiento y el almacenaje de los datos del sistema dependerá del nivel de cobertura de cada proveedor de telefonía móvil celular.

### Resultados Parte A: Verificación del protocolo de mensajería MQTT

Durante la recolección de datos de la Prueba 4.2.1, se ha utilizado la herramienta Wireshark para verificar el protocolo MQTT utilizado para el envío de datos. La Figura 107 muestra la captura de los datos enviados a través de la herramienta Wireshark, para ello se filtra el tráfico del protocolo MQTT. Se puede visualizar que se ha realizado la publicación de los diferentes tópicos que se han establecido en la sección anterior, la fuente de origen es la dirección IP del servidor en la nube y el destino es la dirección IP del usuario. La Figura 108 indica el tamaño del paquete capturado de 173 bytes (1384 bits).

No.	Time	Source	Destination	Protocol	Length	Info
99	7.000411	192.168.100.7	159.65.174.156	MQTT	56	Ping Request
100	7.000737	192.168.100.7	159.65.174.156	MQTT	56	Ping Request
103	7.168882	159.65.174.156	192.168.100.7	MQTT	56	Ping Response
104	7.168883	159.65.174.156	192.168.100.7	MQTT	56	Ping Response
226	19.359006	159.65.174.156	192.168.100.7	MQTT	67	Publish Message [tamb]
233	19.598088	159.65.174.156	192.168.100.7	MQTT	65	Publish Message [rcard]
257	19.747380	159.65.174.156	192.168.100.7	MQTT	173	Publish Message [tcorp], Publish Message [ccald], Publish Message [ccorp], Publish Message [critw], Publish Message [dcald], P...
272	22.012416	192.168.100.7	159.65.174.156	MQTT	56	Ping Request
273	22.012763	192.168.100.7	159.65.174.156	MQTT	56	Ping Request
275	22.208854	159.65.174.156	192.168.100.7	MQTT	56	Ping Response
276	22.210201	159.65.174.156	192.168.100.7	MQTT	56	Ping Response
1285	37.021839	192.168.100.7	159.65.174.156	MQTT	56	Ping Request

**Figura 107.**Captura de datos utilizando Wireshark

```

✓ Frame 257: 173 bytes on wire (1384 bits), 173 bytes captured (1384 bits) on interface 0
  > Interface id: 0 (\Device\NPF_{4A7340C6-F00A-4ED3-B720-2B34F5CF45C2})
    Encapsulation type: Ethernet (1)
    Arrival Time: Feb 25, 2019 10:49:14.289019000 Hora est. Pacifico, Sudamérica
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1551109754.289019000 seconds
    [Time delta from previous captured frame: 0.008058000 seconds]
    [Time delta from previous displayed frame: 0.149292000 seconds]
    [Time since reference or first frame: 19.747380000 seconds]
    Frame Number: 257
    Frame Length: 173 bytes (1384 bits)
    Capture Length: 173 bytes (1384 bits)

```

**Figura 108.** Tamaño del paquete capturado

```

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 11
    Topic Length: 4
    Topic: tamb
    Message: 17.95

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 9
    Topic Length: 5
    Topic: rcard
    Message: 69

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 12
    Topic Length: 5
    Topic: tcorp
    Message: 37.16

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 8
    Topic Length: 5
    Topic: ccaid
    Message: 0

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 8
    Topic Length: 5
    Topic: ccorp
    Message: 0

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 8
    Topic Length: 5
    Topic: critm
    Message: 0

```

**Figura 109.** Características del paquete enviado 1.

```

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 8
    Topic Length: 5
    Topic: ccorp
    Message: 0

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 8
    Topic Length: 5
    Topic: critm
    Message: 0

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 14
    Topic Length: 5
    Topic: dcaid
    Message: Ninguna

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 14
    Topic Length: 5
    Topic: acorp
    Message: Ninguna

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 14
    Topic Length: 5
    Topic: aritm
    Message: Ninguna

MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 11
    Topic Length: 4
    Topic: pamb
    Message: 18.03

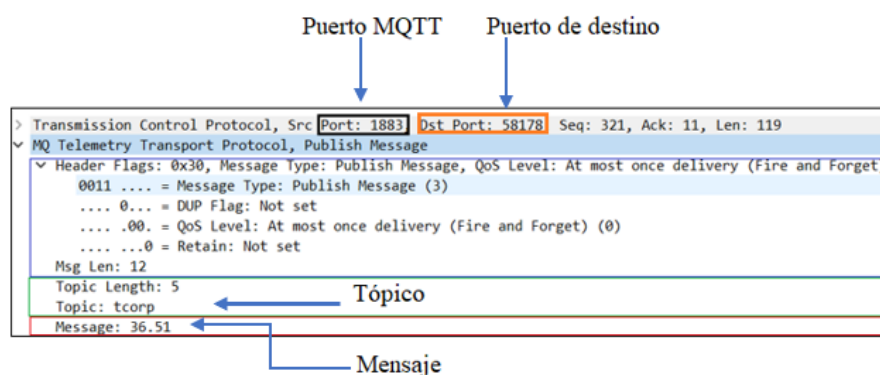
MQ Telemetry Transport Protocol, Publish Message
  > Header Flags: 0x30, Message Type: Publish Message, QoS Level: At most once delivery (Fire and Forget)
    Msg Len: 12
    Topic Length: 5
    Topic: pcorp
    Message: 36.51

```

Paquete a analizar

**Figura 110.** Características del paquete enviado 2.

Las Figuras 107 y 108 muestran todos los tópicos con su respectivo mensaje que son los datos que están siendo monitoreados del paciente. En la Figura 111 se indica las características de los paquete analizado, en este caso se confirma que le puerto de origen es 1883 utilizado por el protocolo MQTT, el puerto de destino 58178, el tópico “tcorp” que representa a la temperatura corporal y el mensaje de 36.51°C.



**Figura 111.** Características del paquete analizado

Se ha realizado la suscripción a un tópico, que es el mensaje de los datos monitoreados, esto permite verificar su envío al Servidor en la nube, en este procedimiento se utilizó la herramienta MobaXterm y en la línea de comandos se colocó lo siguiente:

- mosquitto\_sub: para suscribirse a un tópico.
- d: habilitar el mensaje de depuración
- t: indica a que tópico que desea filtrar para obtener la información.

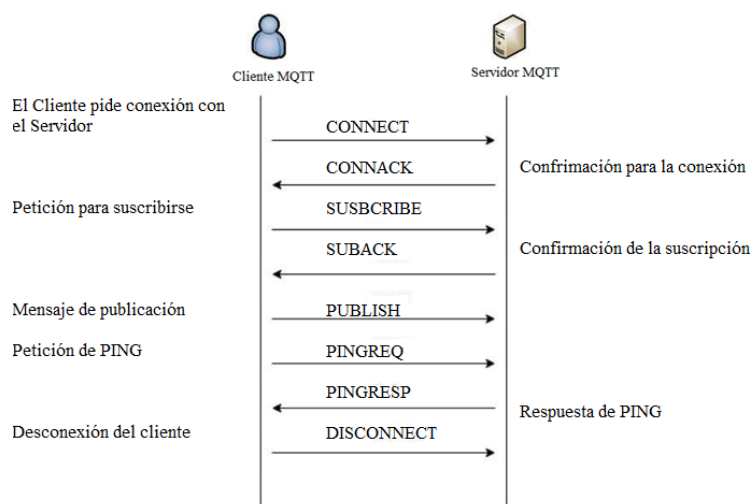
También se puede comprobar cada uno de los paquetes que se muestra en la Figura 112 son los que se ha mencionado en la Tabla 2, además se ha representado con un diagrama de secuencia de los paquetes enviados (ver Figura 113).

```

root@ubuntu-s-lvcpu-1gb-nyc3-01:/# mosquitto_sub -d -t 'tamb'
Client mosqsub/32491-ubuntu-s- sending CONNECT
Client mosqsub/32491-ubuntu-s- received CONNACK
Client mosqsub/32491-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: tamb, QoS: 0)
Client mosqsub/32491-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32491-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'tamb', ... (5 bytes))
23.63
Client mosqsub/32491-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'tamb', ... (5 bytes))
23.86
Client mosqsub/32491-ubuntu-s- sending PINGREQ
Client mosqsub/32491-ubuntu-s- received PINGRESP
Client mosqsub/32491-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'tamb', ... (5 bytes))
23.89

```

**Figura 112.** Suscripción a un tópico



**Figura 113.** Diagrama de suscripción del Cliente MQTT

En las Figuras 108, 109 y 110, se realizó la suscripción a todos los tópicos de manera independiente, para identificar los respectivos paquetes de control de MQTT. Se identifica que se enviaron paquetes de tamaño mínimo de 1 Byte y el paquete máximo enviado fue de 7 bytes.

```

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'tcorp'
Client mosqsub/32456-ubuntu-s- sending CONNECT
Client mosqsub/32456-ubuntu-s- received CONNACK
Client mosqsub/32456-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: tcorp, QoS: 0)
Client mosqsub/32456-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32456-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'tcorp', ... (5 bytes))
36.38
Client mosqsub/32456-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'tcorp', ... (5 bytes))
36.48
Client mosqsub/32456-ubuntu-s- sending PINGREQ
Client mosqsub/32456-ubuntu-s- received PINGRESP

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'rcard'
Client mosqsub/32455-ubuntu-s- sending CONNECT
Client mosqsub/32455-ubuntu-s- received CONNACK
Client mosqsub/32455-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: rcard, QoS: 0)
Client mosqsub/32455-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32455-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'rcard', ... (2 bytes))
96
Client mosqsub/32455-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'rcard', ... (2 bytes))
93

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'dcaid'
Client mosqsub/32457-ubuntu-s- sending CONNECT
Client mosqsub/32457-ubuntu-s- received CONNACK
Client mosqsub/32457-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: dcaid, QoS: 0)
Client mosqsub/32457-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32457-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'dcaid', ... (7 bytes))
Ninguna
Client mosqsub/32457-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'dcaid', ... (7 bytes))
Ninguna
Client mosqsub/32457-ubuntu-s- sending PINGREQ

```

**Figura 114.** Suscripción a tópicos de tcorp, rcard y dcaid

```

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'acorp'
Client mosqsub/32465-ubuntu-s- sending CONNECT
Client mosqsub/32465-ubuntu-s- received CONNACK
Client mosqsub/32465-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: acorp, QoS: 0)
Client mosqsub/32465-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32465-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'acorp', ... (7 bytes))
Ninguna
Client mosqsub/32465-ubuntu-s- sending PINGREQ
Client mosqsub/32465-ubuntu-s- received PINGRESP

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'aritm'
Client mosqsub/32468-ubuntu-s- sending CONNECT
Client mosqsub/32468-ubuntu-s- received CONNACK
Client mosqsub/32468-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: aritm, QoS: 0)
Client mosqsub/32468-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32468-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'aritm', ... (7 bytes))
Ninguna
Client mosqsub/32468-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'aritm', ... (7 bytes))
Ninguna
Client mosqsub/32468-ubuntu-s- sending PINGREQ
Client mosqsub/32468-ubuntu-s- received PINGRESP

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'ccaid'
Client mosqsub/32469-ubuntu-s- sending CONNECT
Client mosqsub/32469-ubuntu-s- received CONNACK
Client mosqsub/32469-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: ccaid, QoS: 0)
Client mosqsub/32469-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32469-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'ccaid', ... (1 bytes))
Client mosqsub/32469-ubuntu-s- sending PINGREQ
Client mosqsub/32469-ubuntu-s- received PINGRESP

```

**Figura 115.** Suscripción a tópicos de acorp, aritm y ccaid

```

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'ccorp'
Client mosqsub/32473-ubuntu-s- sending CONNECT
Client mosqsub/32473-ubuntu-s- received CONNACK
Client mosqsub/32473-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: ccorp, QoS: 0)
Client mosqsub/32473-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32473-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'ccorp', ... (1 bytes))
5
Client mosqsub/32473-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'ccorp', ... (1 bytes))
5
Client mosqsub/32473-ubuntu-s- sending PINGREQ
Client mosqsub/32473-ubuntu-s- received PINGRESP

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'critm'
Client mosqsub/32474-ubuntu-s- sending CONNECT
Client mosqsub/32474-ubuntu-s- received CONNACK
Client mosqsub/32474-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: critm, QoS: 0)
Client mosqsub/32474-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32474-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'critm', ... (2 bytes))
79
Client mosqsub/32474-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'critm', ... (2 bytes))
85
Client mosqsub/32474-ubuntu-s- sending PINGREQ
Client mosqsub/32474-ubuntu-s- received PINGRESP

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'pamb'
Client mosqsub/32470-ubuntu-s- sending CONNECT
Client mosqsub/32470-ubuntu-s- received CONNACK
Client mosqsub/32470-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: pamb, QoS: 0)
Client mosqsub/32470-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32470-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'pamb', ... (5 bytes))
20.46
Client mosqsub/32470-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'pamb', ... (5 bytes))
20.46
Client mosqsub/32470-ubuntu-s- sending PINGREQ
Client mosqsub/32470-ubuntu-s- received PINGRESP

root@ubuntu-s-1vcpu-1gb-nyc3-01:~# mosquitto_sub -d -t 'pcorp'
Client mosqsub/32463-ubuntu-s- sending CONNECT
Client mosqsub/32463-ubuntu-s- received CONNACK
Client mosqsub/32463-ubuntu-s- sending SUBSCRIBE (Mid: 1, Topic: pcorp, QoS: 0)
Client mosqsub/32463-ubuntu-s- received SUBACK
Subscribed (mid: 1): 0
Client mosqsub/32463-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'pcorp', ... (5 bytes))
36.41
Client mosqsub/32463-ubuntu-s- received PUBLISH (d0, q0, r0, m0, 'pcorp', ... (5 bytes))
36.41
Client mosqsub/32463-ubuntu-s- sending PINGREQ
Client mosqsub/32463-ubuntu-s- received PINGRESP

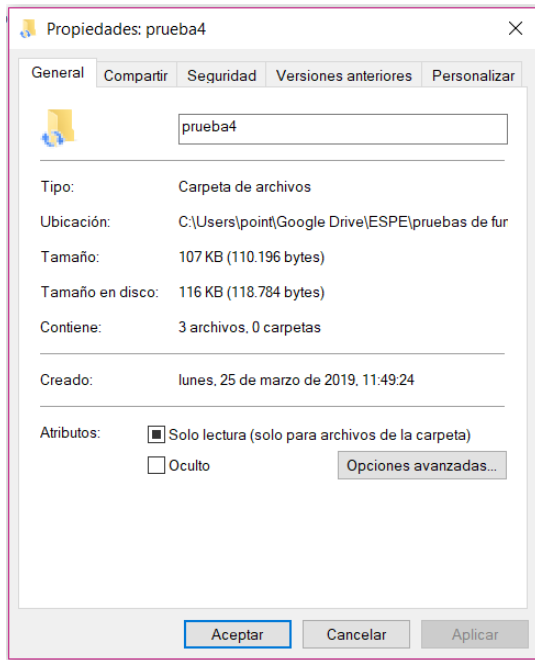
```

*Figura 116.* Suscripción a tópicos de ccorp, critm, pamb y pcorp

## 4.2.2 Resultados: Envío de datos en la tarjeta de memoria

Se ha realizado esta prueba según el protocolo definido para obtener un respaldo de los datos monitoreados en caso de que no exista conectividad a WiFi, de esta manera no existe pérdida de información.

La tarjeta de memoria microSD utilizada tiene un espacio de almacenamiento de 16 GB, para la validación de esta prueba se corroboró la existencia del archivo creado durante la recolección de datos, este archivo ocupa un espacio de 107 KB. La Figura 117 indica las características del archivo con nombre prueba4 y extensión .txt como su respectiva fecha de creación y el tamaño del archivo.



**Figura 117.** Características del archivo creado .txt

La Tabla 48 indica la cantidad de pruebas a realizarse en función del tiempo, se ha considerado el tamaño del archivo y el espacio de almacenamiento de la tarjeta de memoria para obtener el resultado mostrado.

**Tabla 48**  
*Cantidad de pruebas*

Tiempo	Tamaño archivo	#Pruebas
1 min	561 bytes	$28 * 10^6$
5 min	2.03 KB	$14 * 10^6$
10 min	3.62 KB	$10 * 10^6$
190 min	107 KB	$5 * 10^6$

En la Figura 118 se puede observar los datos monitoreados del dispositivo de e-Health, además se ha agregado etiquetas para la identificación de cada dato almacenado. Estos datos son almacenados en forma de vector. La variable cédula no es almacenada en el archivo prueba4.txt



porque el proceso de registro de cédula se realiza en la aplicación web y este atributo se almacena solo en la BD.

Fecha	Hora	Temperatura corporal	Detector de caídas	Alarma ritmo cardiaco	#Alarmas t. corporal	Promedio t. ambiente					
28/3/2019	17:33:31	23.31	35.13	90	Ninguna	Ninguna	Ninguna	0	0	23.18	35.54
28/3/2019	17:34:22	23.40	35.62	94	Caída	Ninguna	Ninguna	1	0	23.18	35.54
28/3/2019	17:34:53	23.21	35.56	94	Ninguna	Ninguna	Ninguna	1	0	23.18	35.54
28/3/2019	17:35:23	23.06	35.65	95	Ninguna	Ninguna	Ninguna	1	0	23.18	35.54
28/3/2019	17:35:54	22.96	35.75	93	Ninguna	Ninguna	Ninguna	1	0	23.18	35.54
28/3/2019	17:36:24	23.04	35.68	94	Ninguna	Ninguna	Ninguna	1	0	23.13	35.65
28/3/2019	17:36:55	23.15	36.18	94	Ninguna	Ninguna	Ninguna	1	0	23.13	35.65
28/3/2019	17:37:25	23.23	36.25	92	Ninguna	Ninguna	Ninguna	1	0	23.13	35.65
28/3/2019	17:37:55	23.29	36.18	89	Ninguna	Ninguna	Ninguna	1	0	23.13	35.65
28/3/2019	17:38:26	23.16	35.62	91	Ninguna	Ninguna	Ninguna	1	0	23.13	35.65
28/3/2019	17:38:56	23.22	35.81	90	Ninguna	Ninguna	Ninguna	1	0	23.21	36.01
28/3/2019	17:39:27	23.16	36.09	93	Ninguna	Ninguna	Ninguna	1	0	23.21	36.01
28/3/2019	17:39:57	23.18	35.65	90	Ninguna	Ninguna	Ninguna	1	0	23.21	36.01
28/3/2019	17:40:27	23.32	36.09	93	Ninguna	Ninguna	Ninguna	1	0	23.21	36.01
28/3/2019	17:40:58	23.26	36.15	97	Ninguna	Ninguna	Ninguna	1	0	23.21	36.01
28/3/2019	17:41:28	23.16	35.93	90	Ninguna	Ninguna	Ninguna	1	0	23.22	35.98
28/3/2019	17:41:59	23.18	36.12	96	Ninguna	Ninguna	Ninguna	1	0	23.22	35.98
28/3/2019	17:42:29	23.40	36.31	92	Ninguna	Ninguna	Ninguna	1	0	23.22	35.98
28/3/2019	17:42:59	23.54	35.93	93	Ninguna	Ninguna	Ninguna	1	0	23.22	35.98
28/3/2019	17:43:30	23.42	36.28	90	Ninguna	Ninguna	Ninguna	1	0	23.22	35.98

Annotations below the table:

- ↑ Temperatura ambiente
- ↑ Ritmo cardiaco
- ↑ Alarma temperatura corporal
- ↑ #Alarmas de caída
- ↑ #Alarmas de ritmo
- ↑ Promedio t. corporal

Figura 118. Archivo almacenado en la tarjeta de memoria

#### 4.2.3 Resultados: “Funcionalidad del Sistema de Monitoreo”

En este escenario el sujeto de prueba 2 adoptará una posición sentado según lo definido en el protocolo de pruebas, sin realizar ningún movimiento brusco y con el dispositivo colocado en su pecho, se realizará la prueba en un ambiente interior. Primero se mostrará cómo se debe acceder a la aplicación multiplataforma y posteriormente se analizará los resultados obtenidos de las 160 muestras que representan los datos monitoreados.

##### Datos del dispositivo e-Health

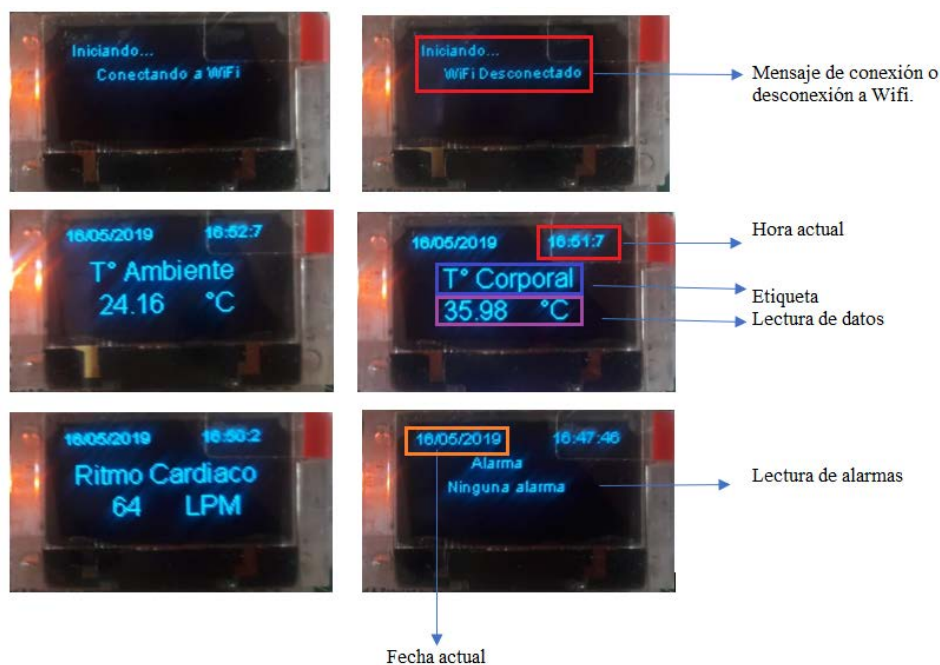
Cuando se inicie la prueba el dispositivo e-Health se debe encender y en la pantalla OLED se muestra un mensaje principal en el cual permite al usuario verificar si está conectado a una red WiFi mediante el mensaje: “Iniciando ... Conectado a WiFi”, caso contrario su mensaje es “WiFi

Desconectado”, además existe un LED indicador de conexión de color verde. El tiempo que tarda el dispositivo conectarse a la red es aproximadamente de 5 a 10 segundos.

La red a la cual el dispositivo se desea conectar esta previamente configurada en el programa desarrollado en Arduino, en el caso que se desee modificar la red se debe realizar el cambio en las variables SSID y password en el programa como se muestra en la Figura 119.

```
//Red y contraseña
const char* ssid    = "NETLIFE-MASABANDA";
const char* password = "777777777777";
```

**Figura 119.** Configuración de la red en el programa



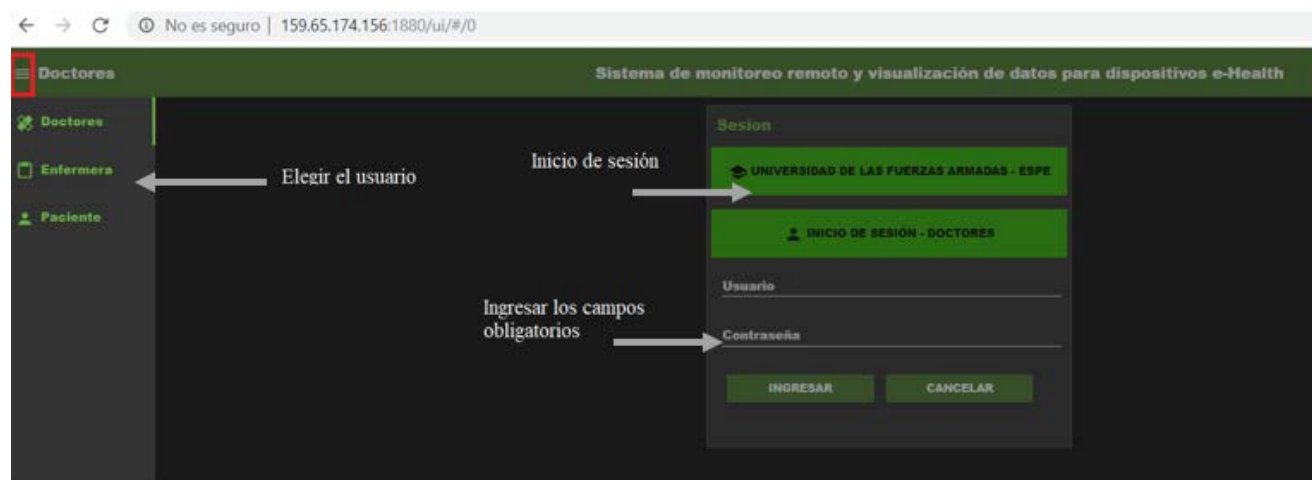
**Figura 120.** Mensajes de la pantalla OLED

En la Figura 120 se muestra cada una de los mensajes que se muestran en la pantalla OLED durante el funcionamiento del dispositivo, primero se visualiza la Temperatura ambiente, luego temperatura corporal, posteriormente ritmo cardiaco y como último mensaje las alarmas de las variables de temperatura corporal, ritmo cardiaco y detector de caídas.

## Aplicación web

Se ha desarrollado una aplicación web que cumple con los requerimientos de software que se detallaron en la sección 3.1.2.1.

Para acceder a los datos monitoreados que muestra la página web se debe ingresar en el navegador con la dirección <http://159.65.174.156:1880/ui/>. En la Figura 121 se observa la ventana principal, en donde se elige el usuario a cuál desea acceder a través de la opción que se muestra en la parte superior izquierda. Una vez que se ha elegido el usuario se presenta el inicio de sesión correspondiente, el usuario debe ingresar los campos obligatorios.



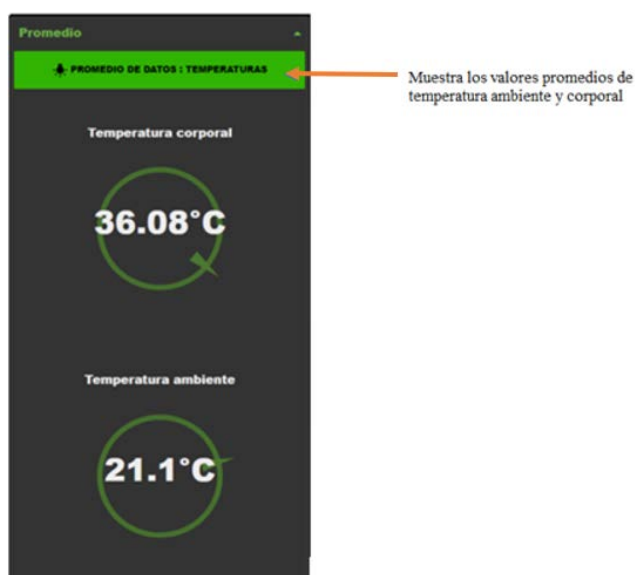
**Figura 121.** Ventana de inicio de sesión del usuario doctor y enfermera

Si se ha elegido el usuario Doctor, se accede a los datos que muestra la Figura 122, en esta ventana se debe realizar el registro del paciente a) en el caso que el paciente no ha sido registrado, b) ingrese la cédula del paciente registrado para que los datos se almacenen en la BD, y c) muestra el mensaje que se emite en el caso que el paciente no esté registrado (Figura 124). Los datos se presentan cada treinta segundos. Además, se puede obtener la información del paciente a través de la pestaña consulta. Para ello se debe ingresar el número de cédula del usuario y se puede filtrar

por fecha con el formato de aaaa-mm-dd, también se puede descargar los datos en un archivo de extensión .xls esto se realiza a través del botón descargar que posee la aplicación web (ver Figura 125).



**Figura 122.** Ventana de datos, usuario:doctor



**Figura 123.** Promedio de datos monitoreados, usuario:doctor

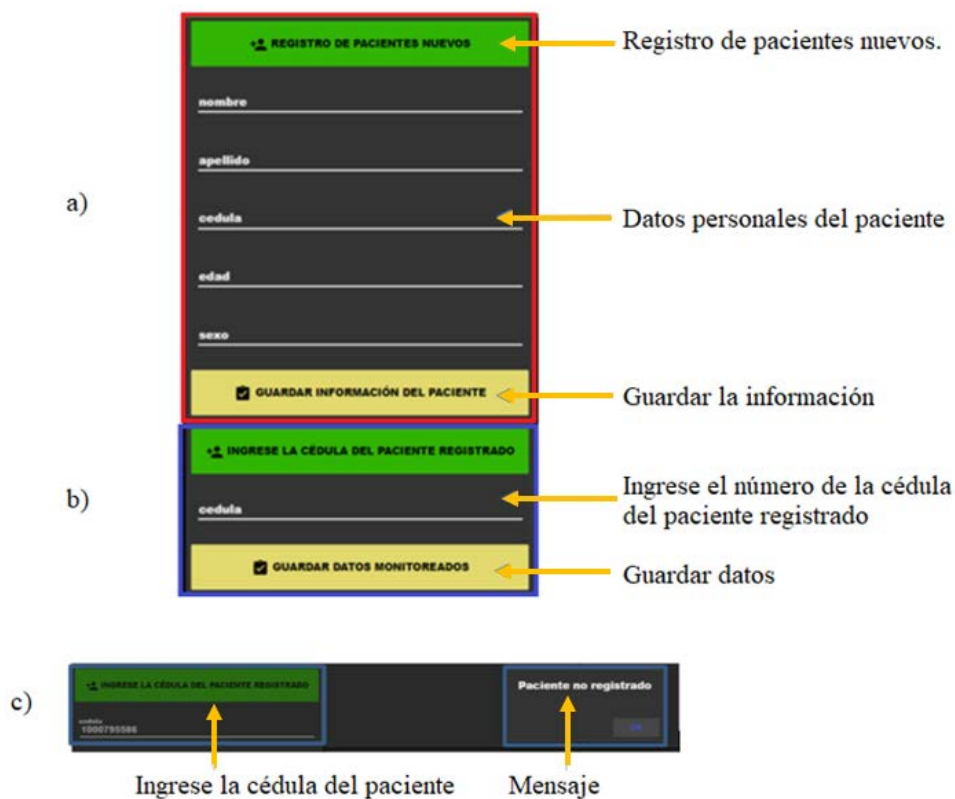


Figura 124.Registro del paciente

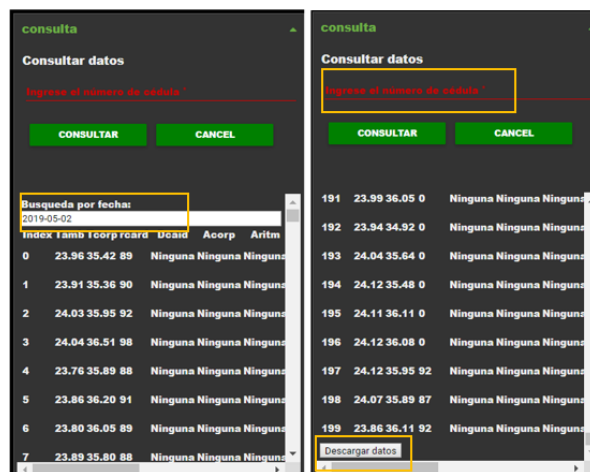
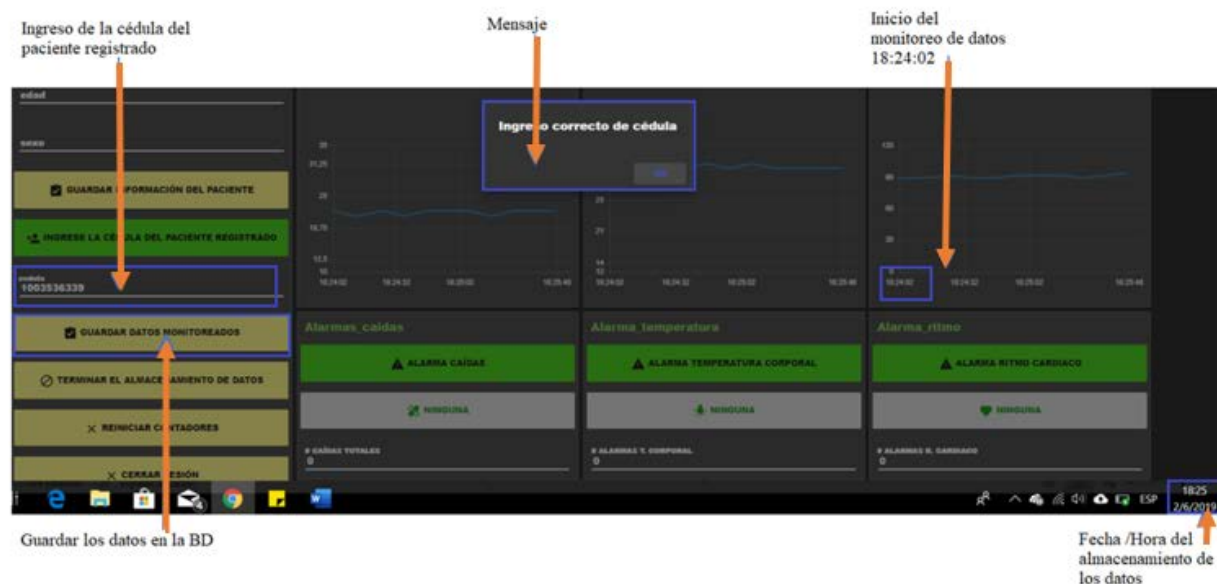


Figura 125.Consulta de los datos monitoreados

Se realizó una prueba para la verificación que los datos de monitoreo sean almacenados en la BD después de haber registrado la cédula del paciente. Por ejemplo en la Figura 126 se muestra que los datos ya estan siendo monitoreados una vez encendido el dispositivo a las 18:24:02 pero no se almacenan hasta el ingreso de la cédula del paciente, verificando así que luego del registro del paciente a las 18:25, esta información esta en la BD (Figura 127).



**Figura 126.** Registro del paciente para el almacenamiento en la BD

Último ingreso en la BD						
95800	2019-05-30 16:32:16	25.03	27.70	71		Caída
36339	2019-06-02 18:25:50	22.04	35	94		Ninguna
Ingreso en la BD						

**Figura 127.** Almacenamiento de datos en la BD

Para terminar el almacenamiento de datos se debe apagar el dispositivo, luego presionar el botón de “Terminar el almacenamiento de datos” y posteriormente el botón de “Reiniciar contadores” (ver Figura 128).



Figura 128. Terminar el almacenamiento de datos en la BD

Para el caso que el usuario sea enfermera los datos monitoreados son los que se muestran en la Figura 129 y si el usuario es paciente son los de la Figura 130 según los requerimientos de software.

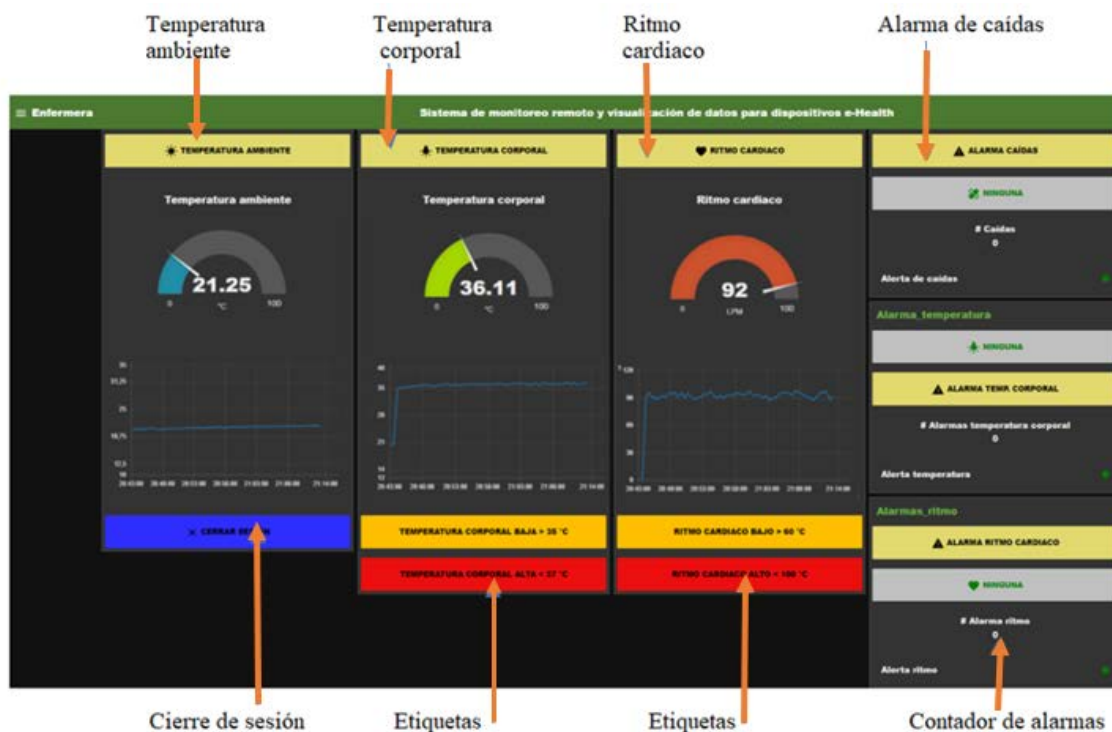


Figura 129. Ventana de datos, usuario: enfermera



Figura 130. Ventana de datos, usuario: paciente

Si un usuario es paciente y requiere ver su perfil cuando no se estén monitoreando sus datos, este puede acceder con su usuario, contraseña y cédula, pero solo se permite descargar los datos anteriores (ver Figura 131).

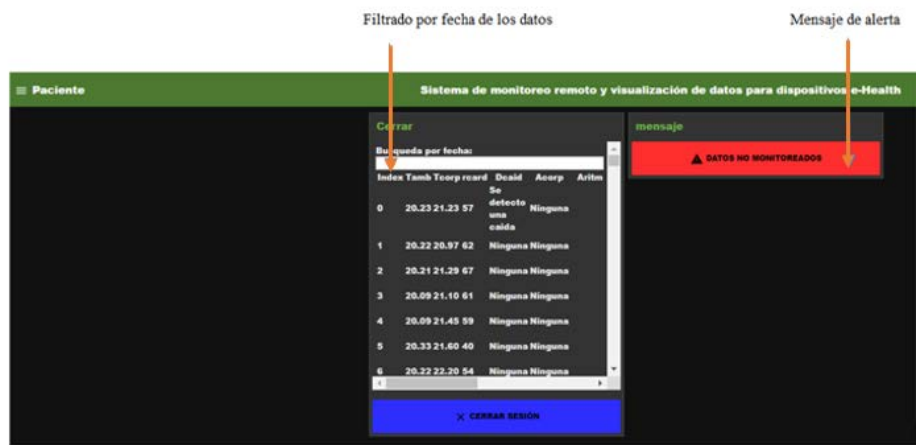


Figura 131. Ventana de datos, usuario paciente

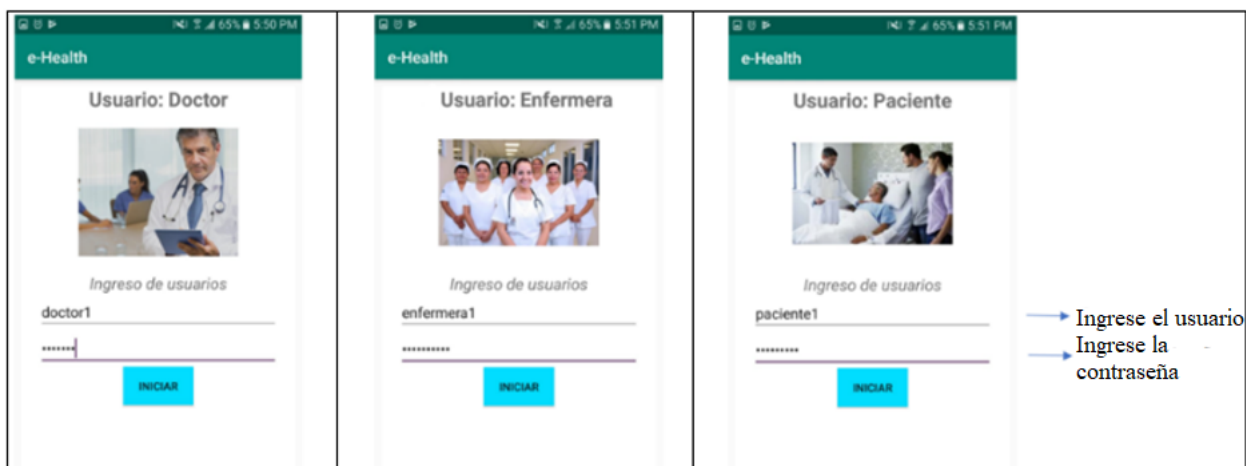


## Aplicación móvil

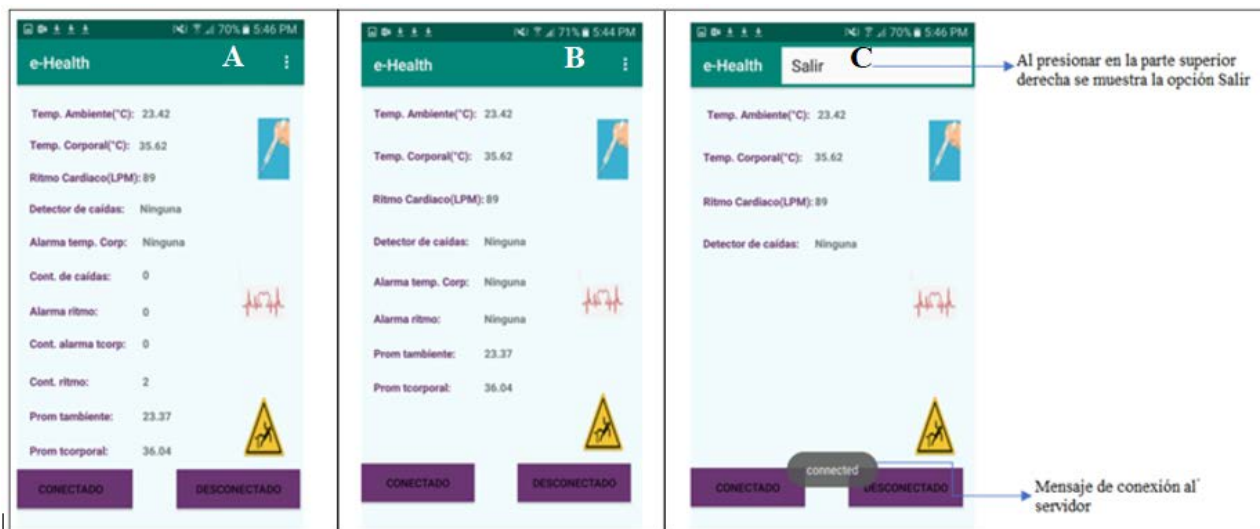
La Figura 132 muestra la ventana de elección de usuarios (doctor, paciente o enfermera), una vez que se elegido el usuario se accede al inicio de sesión (Figura 133). Los datos monitoreados se muestran en la Figura 134.



**Figura 132.** Ventana de elección de usuarios



**Figura 133.** Ventana de inicio de los diferentes usuarios

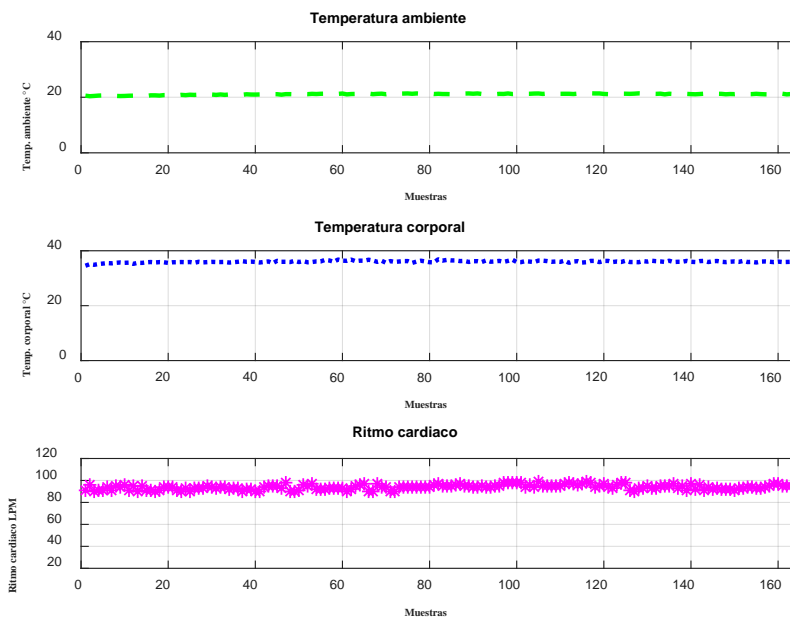


**Figura 134.** Ventana de datos monitoreados (A) enfermera, (B) doctor y (C) paciente

### Análisis de resultados

La Figura 135 se muestra los resultados obtenidos, durante 1h 20 min donde se han tomado 160 muestras, el sensor de temperatura ambiente (línea verde) registro un valor promedio de 21.09 °C, el sensor de temperatura corporal (línea azul) registró un valor promedio de 36.01°C y el sensor de ritmo cardíaco (Línea rosada) un valor promedio de 93 LPM.

Durante la prueba ninguno de los sensores registraron falsos positivos, porque el sujeto de prueba todo el tiempo se mantuvo en una posición erguida que es la posición recomendada en el protocolo de pruebas y no realizó ningun movimiento que pueda afectar a la toma de los datos monitoreados.



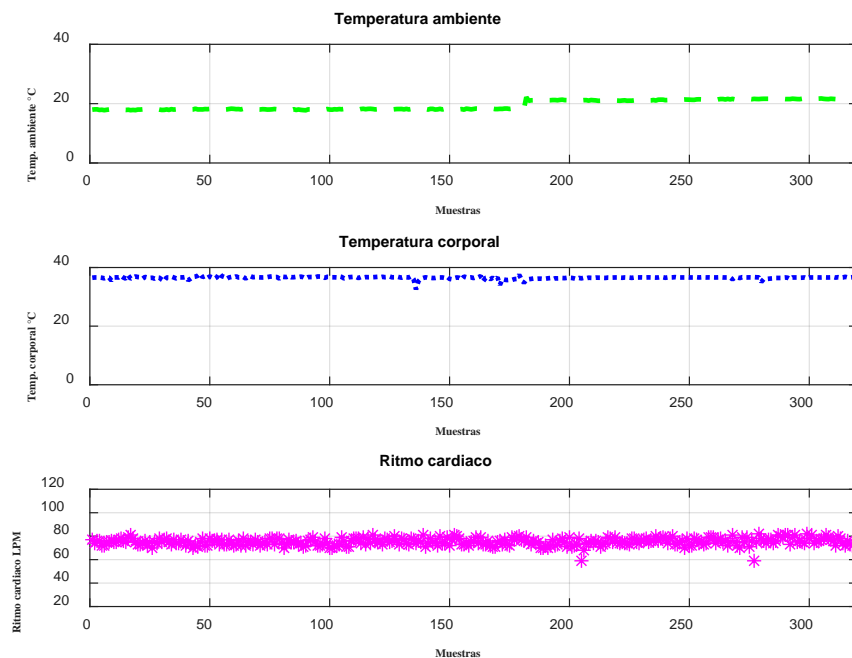
**Figura 135.** Datos monitoreados del dispositivo, prueba de Funcionalidad del sistema

#### 4.2.4 Resultados: Carga y descarga de la batería del dispositivo e-Health.

##### Resultados parte A: Carga de la batería del dispositivo.

Se debe conectar el cable USB a la computadora y el interruptor de Encendido/Apagado debe estar en la posición encendido. Además, se inicia el cronómetro para determinar el tiempo de carga.

En la Figura 136 se muestra los datos monitoreados del sujeto de prueba, durante la carga de la batería, los datos que se obtuvieron sufren un aumento de  $0.8^{\circ}\text{C}$  en la temperatura ambiente y corporal, mientras que el sensor de ritmo cardiaco presenta un aumento de 6 LPM. Los datos reales del paciente se han verificado utilizando los diferentes instrumentos de referencia. Este error se produce porque la referencia de voltaje ha cambiado, debido a que se tiene una conexión externa que alimenta con mayor corriente al dispositivo e-Health.



**Figura 136.** Datos monitoreados del dispositivo, Prueba A

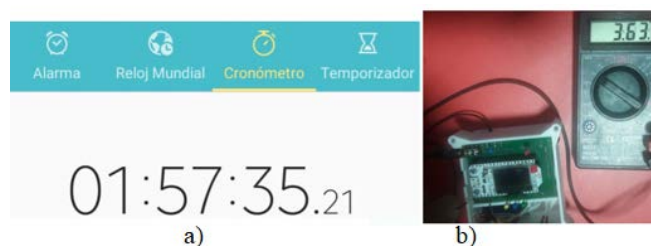
La Figura 137 a) indica el tiempo de carga de la batería a través de un cronómetro, y la Figura 137 b) muestra la medición realizada directamente a la batería del dispositivo de e-Health con el multímetro.



**Figura 137.** Voltaje de la batería cargada, Prueba A

Se ha realizado una prueba de carga del dispositivo pero sin colocarlo en el sujeto de prueba, la Figura 138 a) indica el tiempo de carga de la batería a través de un cronómetro, y la Figura 138 b)

muestra la medición realizada directamente a la batería del dispositivo de e-Health con el multímetro.



**Figura 138.** Voltaje de la batería carga

En la Tabla 49 se verifica que cuando se carga el dispositivo sin el sujeto de prueba su tiempo de carga es menor, con una diferencia de 15 minutos.

**Tabla 49**

*Tiempo de carga de la batería*

Sujeto de prueba	Tiempo de carga
SI	02:12:16
NO	01:57:35

### Resultados parte B: Descarga de la batería del dispositivo e-Health

En este escenario se determinará el tiempo de funcionamiento del dispositivo, el sujeto de prueba 2 adoptará una posición horizontal como se indica en la Figura 86 y/o estará dormido.

### Análisis de resultados

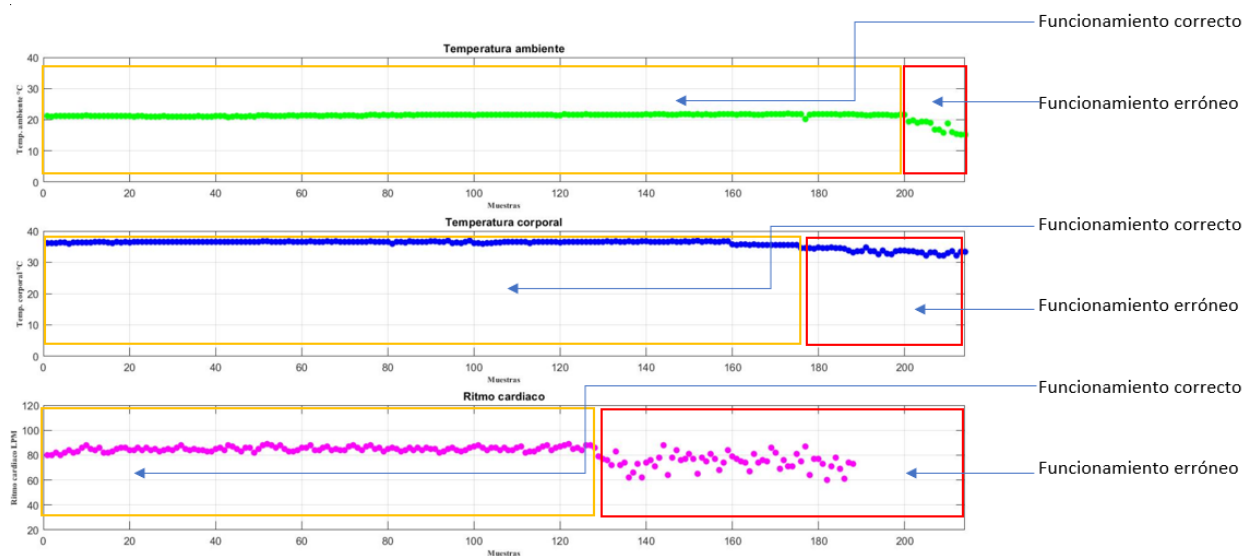
La Figura 139 indica que la prueba se ha dividido en dos secciones, se muestra el periodo del correcto funcionamiento del dispositivo y el periodo de la lectura de datos erróneos. A continuación, se realiza un análisis de cada uno de los sensores en la prueba realizada:

- Sensor del ritmo cardiaco (línea rosada): es el primer sensor que muestran una lectura de falsos positivos pudiendo ser provocados por la incorrecta posición del sensor en el sujeto

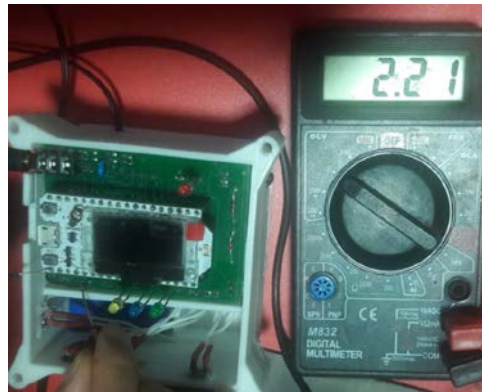
de prueba o si el sujeto de prueba ha realizado algún movimiento inesperado mientras estaba dormido.

- Sensor de temperatura corporal (línea azul): este sensor debe estar ubicado debajo de la axila del sujeto y debe apoyar su brazo contra su cuerpo como se indica en la Figura 86, pero en esta prueba cuando se adoptó un cambio de posición del sujeto la lectura de datos fue errada.
- Sensor de temperatura ambiente (línea verde): presenta falsos positivos cuando la carga de la batería disminuyó totalmente.

El dispositivo e-Health funcionó correctamente por un tiempo de una hora y veinte minutos, porque la batería estaba cargada con un voltaje de aproximadamente 3.7 V, posteriormente la carga disminuye provocando que la lectura de los sensores sea errónea, en la Figura 140 se indica el valor de la batería ha disminuido a 2.21 V cuando ha transcurrido aproximadamente 2 horas



**Figura 139.** Aviso de alarmas, prueba N°2



*Figura 140.* Voltaje de batería descargada

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

Con la repotenciación del dispositivo e-Health se logró la reducción del tamaño y el peso del dispositivo, esto se consiguió con la utilización de la tarjeta ESP 32 con pantalla OLED que incluye un módulo WiFi. El diseño de la placa PCB fue realizado a doble lado de esta manera cada uno de sus componentes SMD (Surface Mounting Device) están dispuestos de manera que se redujo el tamaño de la placa en un 55% y el peso en un 42%.

Con la implementación del dispositivo e-Health se logró obtener datos de temperatura ambiente, temperatura corporal y ritmo cardiaco. Primero se desarrolló la caracterización de cada sensor de forma independiente obteniendo el error, y posteriormente con esa información se corrigió mediante software realizando una compensación:

- Sensor de temperatura ambiente (LM35): en la caracterización presento un error de 0.68% y posteriormente a la compensación un error a 0.41%.
- Sensor de temperatura corporal (Termistor NXFT15H103FA2B100): en la caracterización presento un error de 1.39% y posteriormente a la compensación un error a 0.08%.
- Sensor de ritmo cardiaco (HRM-2511B): en la caracterización presento un error de 2.77% y posteriormente a la compensación un error a 1.38%.

En el desarrollo de la prueba del funcionamiento se colocó el sensor de temperatura corporal debajo de la axila del paciente, al colocar el sensor en otra área de piel se pudo observar que las lecturas son erróneas como se muestra en la Prueba 4.1.1, si se cubre con otro material el encapsulado del sensor, la temperatura puede variar.



Se identificó que el sensor de ritmo cardiaco al estar un tiempo excesivo en el dedo del sujeto de prueba puede ocasionar que la circulación de sangre disminuya obteniendo falsos positivos en la lectura de datos, es por ello que se implementó un LED indicador que se enciende cada vez que se detecta un latido y las pruebas se limitan a 10 minutos.

Se ha identificado que la posición del sujeto de prueba influye sobre las mediciones, en el caso que se registre una caída podría existir falsos positivos tanto para los datos monitoreados de temperatura corporal y de ritmo cardiaco, por tanto, los sensores pueden cambiar su posición y desconectarse del sujeto de prueba.

Se identificó que durante el funcionamiento el dispositivo e-Health puede desconectarse de la red WiFi, por esta razón se implementó un LED indicador de conexión. En caso de existir la desconexión los datos son almacenado en una tarjeta de memoria interna en un archivo con nombre prueba4 y de extensión .txt

El protocolo MQTT permitió la publicación de diferentes tópicos a través del broker Mosquitto, siendo este un protocolo asíncrono donde cada sensor puede enviar las lecturas a los tópicos simultáneamente y posibilita un tiempo de respuesta rápido para aplicaciones de tiempo real y que a diferencia del protocolo HTTP que es síncrono y el cliente debe esperar la respuesta del servidor.

Se ha establecido una Calidad de Servicio (QoS) de 0 para el envío de los datos monitoreados, porque se envían estos datos al menos una vez, y posteriormente se eliminan de la cola de mensajes salientes, evitando su duplicidad.

Digital Ocean es una proveedor que ofrece una Infraestructura como Servicio (IaaS), que permitió la instalación de varios componentes de software como el broker Mosquitto, el framework

Node-RED y la Base de Datos MySQL que son indispensables para lograr el desarrollo del proyecto.

Para el desarrollo de la aplicación web se ha utilizado la herramienta Node-RED programada en Node.js que es una plataforma de lenguaje de programación JavaScript, esta herramienta visual que contiene diferentes nodos, permitieron el desarrollo de flujos para obtener la interfaz multiusuario. Uno de los principales inconvenientes se refiere a que la mayoría de páginas web están desarrolladas con el lenguaje PHP, por esta razón la información de referencia para el diseño en la herramienta visual se debía aprender de acuerdo a los requerimientos del proyecto.

Para el almacenamiento de los datos se utilizó la herramienta Node-RED a través de diferentes nodos, que se programaron con las sintaxis de sentencia de MySQL para lograr la inserción o consulta de datos, además se utilizó una base de datos SQL porque esta requiere que toda la información que se adquiere de los sensores siga la misma estructura, es decir no se agregan o se eliminan sensores que proporcionan la información de datos monitoreados del dispositivo e-Health.

La aplicación web cumplió con los requerimientos de software, permitiendo un uso multiusuario los cuales tiene diferentes perfiles como doctor, enfermera y paciente. Las funciones a las que se tienen acceso varían dependiendo del usuario, por ejemplo, el doctor puede monitorear, almacenar información e ingresar nuevos pacientes, la enfermera recibe alarmas en caso de anomalías en los datos, mientras que el usuario paciente puede visualizar los datos monitoreados y descargar su información.

Para el desarrollo de la aplicación móvil se utilizó la herramienta Android Studio y es indispensable incluir la librería Cliente Paho MQTT porque permite establecer la conexión con el broker MQTT, además el Cliente móvil se suscribe a los diferentes tópicos declarados.

El Cliente IoT por medio de la librería de Adafruit\_MQTT\_Client.h en el IDE de Arduino, realiza la publicación de los diferentes tópicos que contiene los datos monitoreados del dispositivo e-Health.

### **Recomendaciones**

Es indispensable que la conexión a Internet sea estable, de esta manera se evitará la pérdida de datos, además la red de conexión que se utilice debe tener habilitados los puertos de MQTT (ver Tabla 3). No se recomienda utilizar redes móviles de telefonía celular por la capacidad de su cobertura que puede afectar en la Calidad de Servicio (QoS).

Se recomienda que durante el proceso de grabado del programa en la tarjeta ESP 32 es importante que el interruptor de carga de software de configuración este en modo ON, caso contrario se presentará un error. Una vez que se ha finalizado el proceso de grabación se debe cambiar el estado del interruptor a modo OFF para su funcionamiento.

En caso de que la Fecha /Hora del dispositivo de e-Health no sea la correcta se debe presionar el pulsador Fecha/Hora que modifica estos parámetros a los correctos, para ello es necesario que el dispositivo tenga acceso a una conectividad WiFi para realizar este procedimiento.

Para realizar la carga de la batería del dispositivo e-Health se debe conectar el cable de alimentación USB a un computador y el interruptor de encendido/apagado se debe cambiar de posición apagado a la posición encendido.

La herramienta Node-RED utilizada para el desarrollo de la aplicación web contiene una paleta de nodos que facilita las tareas del programador, pero este debe saber diferenciar el tipo de nodo que desea utilizar ya que algunos solo admiten entradas, salidas o entradas y salidas de datos. Además, no son personalizados.

## REFERENCIAS BIBLIOGRÁFICAS

- Alonso-Arévalo, J., & Mirón-Canelo, J. A. (2017). Aplicaciones móviles en salud: potencial, normativa de seguridad y regulación Mobile health applications: potential, regulation and security. *Revista Cubana de Información En Ciencias de La Salud*, 2828(33), 1–13. Retrieved from <http://www.acimed.sld.cu/index.php/acimed/article/view/1136/690>
- Anand, V. K. M., Nisar, P. S. B., Naresh, G. D., Nagendra, R. S., & Mithun, S. (2018). An improved performance of home E-health portable monitoring system. *RTEICT 2017 - 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology, Proceedings, 2018–Janua, 2222–2226*. <https://doi.org/10.1109/RTEICT.2017.8256995>
- Bao, J., Ye, M., & Dou, Y. (2016). Mobile Phone-based Internet of Things Human Action Recognition for E-health. *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, 957–962.
- Barbancho, J. (2014). *Redes locales : sistemas microinformáticos y redes*. Paraninfo.
- Benítez, D., & Calderón, C. A. (2017). *Architectural proposal for Internet of Things*. Retrieved from <https://www.researchgate.net/publication/320353907>
- Borton, D. (2014). *Biosensors Fact Sheet*. Retrieved from [www.nibib.nih.gov](http://www.nibib.nih.gov)
- campusMVP. (2018). Desarrollador web: Front-end, back-end y full stack. ¿Quién es quién? | [campusMVP.es](http://campusMVP.es). Retrieved June 14, 2019, from <https://www.campusmvp.es/recursos/post/Desarrollador-web-Front-end-back-end-y-full-stack-Quien-es-quien.aspx>
- CEDIA. (2018). Evaluación del conocimiento sobre alimentación, ejercicio, redes sociales y

desarrollo de prototipos para la toma de medicación y monitorización de signos vitales orientados al adulto mayor mediante la integración y el uso de las nuevas tecnologías de inf. Retrieved June 8, 2018, from <https://www.cedia.edu.ec/es/proyectos-ganadores/cepra-x/evaluacion-conocimiento-alimentacion-ejercicio-redes-sociales-desarrollo-prototipos-toma-medicacion-monitorizacion-signos-vitales-orientados-adulto-mayor-integracion-uso-nuevas-tecnologias-informacio>

Chiuchisan, I., & Geman, O. (2016). Trends in embedded systems for e-Health and biomedical applications. In *Proceedings of the 2016 International Conference and Exposition on Electrical and Power Engineering, EPE 2016* (pp. 304–308). IEEE. <https://doi.org/10.1109/ICEPE.2016.7781352>

Colucci, G. P., Poletti, M., Stefanelli, R., & Trinchero, D. (2017). Internet of Things as a means to improve agricultural sustainability. *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 1–4. <https://doi.org/10.1109/BIOCAS.2017.8325182>

Cope, S. (2019). MQTT Brokers / Servers y guía de alojamiento. Retrieved June 13, 2019, from <http://www.steves-internet-guide.com/mqtt-hosting-brokers-and-servers/>

Corral, P. (2016). *Simulación de técnicas de diversidad y filtrado Kalman en redes inalámbricas*. (Universidad Miguel Hernández de Elche, Ed.). Retrieved from [https://books.google.com.ec/books?id=i3AYDQAAQBAJ&pg=PA19&dq=ieee+802.15+versiones&hl=es&sa=X&ved=0ahUKEwj10eS1z-HdAhWvt1kKHY-sCuQQ6AEIJTAA#v=onepage&q=ieee 802.16&f=false](https://books.google.com.ec/books?id=i3AYDQAAQBAJ&pg=PA19&dq=ieee+802.15+versiones&hl=es&sa=X&ved=0ahUKEwj10eS1z-HdAhWvt1kKHY-sCuQQ6AEIJTAA#v=onepage&q=ieee%20802.16&f=false)

Craggs, I. (2015). MQTT Client Library Encyclopedia - Paho MQTT C Client. Retrieved June 14, 2019, from <https://www.hivemq.com/blog/mqtt-client-library-pahocclient/>

- Ear clip infrared heart rate pulse sensor - KYTO2511B. (2018). Retrieved December 3, 2018, from <https://kytofitness.com/products/ear-clip-infrared-heart-rate-pulse-sensor-kyto2511b>
- Galarraga, M., Martínez, I., De Toledo, P., Jiménez, S., García, J., & Serrano, L. (2013). *Norma ISO11073 / IEEE1073 sobre interoperabilidad de dispositivos médicos aplicada a telemonitorización*. Retrieved from <http://diec.unizar.es/intranet/articulos/uploads/05caseibSTD.pdf.pdf>
- González, A. (2017). IoT : Dispositivos , tecnologías de transporte y aplicaciones. Retrieved from <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/64286/3/agonzalezgarcia0TFM0617 memoria.pdf>
- Healthwise. (2017). Temperatura corporal | NorthShore. Retrieved January 2, 2019, from <https://www.northshore.org/healthresources/encyclopedia/encyclopedia.aspx?DocumentHwid=hw198785&Lang=es-us>
- Hernández, A., Alexander, G., Rueda, H., Andrés, F., Escobar, V., Juliana, L., & Alexander, S. (2017). Sistema e-Salud para el monitoreo de un prototipo de incubadora neonatal E-health system used for monitoringaprototype neonatal incubator, *38*(2), 107–112.
- Hussien, Z. A., Jin, H., Abduljabbar, Z. A., Hussain, M. A., Yassin, A. A., Abbdal, S. H., ... Zou, D. (2016). Secure and Efficient E-health Scheme Based on the Internet of Things.
- Ibhaze, A. E., Eleanor, E. C., & Idachaba, F. E. (2016). E-health monitoring system for the aged. *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies, EmergiTech 2016*, 298–299. <https://doi.org/10.1109/EmergiTech.2016.7737355>
- Imdoukh, M., Khalil, Y., Rady, A., & Khanafer, M. (2017). Zealth: ZigBee-Based WBAN System

- for e-Health Monitoring. In *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)* (pp. 1–9). IEEE. <https://doi.org/10.1109/IEEEGCC.2017.8448094>
- InfoQ. (2018). Tutorial de bailarina: un lenguaje de programación para la integración. Retrieved June 14, 2019, from <https://www.infoq.com/articles/ballerina-integration-tutorial-part-2/>
- Instrument, T. (2016). LM35 Datasheet. *LM35 Datasheet*, 1. Retrieved from <https://www.electroschematics.com/6393/lm35-datasheet/>
- Ivascu, T., Frincu, M., & Negru, V. (2016). Considerations towards security and privacy in Internet of Things based eHealth applications. *2016 IEEE 14th International Symposium on Intelligent Systems and Informatics (SISY)*, 275–280. <https://doi.org/10.1109/SISY.2016.7601512>
- Kit 16 sensores SmartHome Arduino (SKU 504G2) | HUBOT. (2018). Retrieved June 13, 2019, from <https://hubot.cl/product/kit-16-sensores-smarthome/>
- Kotronis, C., Minou, G., Dimitrakopoulos, G., Nikolaidou, M., Anagnostopoulos, D., Amira, A., ... Djelouat, H. (2018). Managing criticalities of e-health IoT systems. *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband, ICUWB 2017 - Proceedings, 2018-Janua*, 1–5. <https://doi.org/10.1109/ICUWB.2017.8251004>
- Langereis, G. (2010). Photoplethysmography ( PPG ) system, (February), 1–22.
- Llordachs, F. (2016). E-health: internet de las cosas médicas. Retrieved from <http://www.sumandohistorias.com/reportajes/e-health-internet-cosas-medicas/>
- López, R., Chiriboga, M., & Carrera, A. (2016). The Present Situation of e-Health and mHealth in Ecuador Resumo Resumen Abstract. *Latin American Journal of Telehealth*, 4.
- Maia, P., Baffa, A., Cavalcante, E., Delicato, F. C., Batista, T., & Pires, P. F. (2015). 2015 XXXIII Brazilian Symposium on Computer Networks and Distributed Systems A Middleware

- Platform for Integrating Devices and Developing Applications in e-Health, 10–18.  
<https://doi.org/10.1109/SBRC.2015.13>
- Maksimovic, M; Vujovic, V. (2017). Internet of Things Based E-health Systems: Ideas, Expectations and Concerns. In A. A. Samee U. Khan, Albert Y. Zomaya (Ed.), *Handbook of Large-Scale Distributed Computing in Smart Healthcare* (1st ed.). Springer International Publishing.
- Malhotra, R., Bhattacharya, A., Kumar, A., Reddy, N. V., & Cao, J. (2011). A new methodology for multi-pass single point incremental forming with mixed toolpaths. *CIRP Annals - Manufacturing Technology*, 60(1), 323–326. <https://doi.org/10.1016/j.cirp.2011.03.145>
- Manufacturing Co, M. (2018). *NTC Thermistors R44E.pdf*. Retrieved from <https://www.murata.com/>
- Martinez, D. A. (2015). IEEE 1073 Standard implementation to manage and storage corporal variables using mobile devices. *IEEE Latin America Transactions*, 13(3), 835–842. <https://doi.org/10.1109/TLA.2015.7069112>
- Mejía, J. (2017). El país requiere una estrategia de e-Health articulado al sistema nacional. Retrieved June 4, 2017, from <https://www.redaccionmedica.ec/secciones/gestion/el-pa-s-requiere-una-estrategia-de-e-health-articulado-al-sistema-nacional-91052>
- MINTEL/Ministerio de Telecomunicaciones y de la Sociedad de la Informacion. (2013). Expansión Del Programa De Telemedicina a Nivel Nacional, 49. Retrieved from <https://www.telecomunicaciones.gob.ec/wp-content/uploads/2015/06/Expansion-del-Programa-de-Telemedicina-Nivel-Nacional.pdf>
- MySignals. (2016). MySignals - eHealth and Medical IoT Development Platform. Retrieved



January 17, 2019, from <http://www.my-signals.com/>

Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP

and HTTP. Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In 2017 IEEE International Symposium on Systems Engineering, ISSE 2017 -. In *2017 IEEE International Symposium on Systems Engineering, ISSE 2017 - Proceedings* (pp. 1–7). IEEE. <https://doi.org/10.1109/SysEng.2017.8088251>

Naylamp Mechatronics. (2018). Módulo ADXL335 Acelerómetro Analógico. Retrieved December 25, 2018, from <https://naylampmechatronics.com/sensores-posicion-inerciales-gps/160-modulo-adxl335-acelerometro-analogico.html>

Networking and Emerging Optimization. (2018). protocolo http. Retrieved February 17, 2019, from <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>

Neyja, M., Mumtaz, S., Huq, K. M. S., Busari, S. A., Rodriguez, J., & Zhou, Z. (2018). An IoT-based e-health monitoring system using ECG signal. *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings, 2018-Janua*, 1–6. <https://doi.org/10.1109/GLOCOM.2017.8255023>

Noaman, A. Y., Al-Abdullah, N., Jamjoom, A., Ragab, A. H. M., Nadeem, F., & Ali, A. G. (2018). Knowledge Based e-Health Surveillance System for Predicting Hospital Acquired Infections. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)* (pp. 345–351). IEEE. <https://doi.org/10.1109/COMPSAC.2018.10255>

Node-RED. (2016). Node-RED. Retrieved January 30, 2019, from <https://nodered.org/>

Noronha, A., Moriarty, O'Connell, K., & Villa, N. (2014). El valor de IoT: cómo pasar de conectar cosas a obtener información. *Cisco, I(1)*, 20.

- OASIS. (2014). MQTT Versión 3.1.1. Retrieved February 15, 2019, from [http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#\\_Toc398718027](http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718027)
- Organización Mundial de la Salud (OMS) y la Unión Internacional de Telecomunicaciones (UIT). (2012). Conjunto de herramientas para una estrategia de eSalud nacional. Retrieved September 23, 2018, from <http://www.paho.org/ict4health/images/docs/conjuntoherramientasestrategiaesaludnacional.pdf>
- Organización Panamericana de la Salud. (2016). *Revisión de estándares de interoperabilidad para la eSalud en Latinoamérica y el Caribe*.
- Placa WiFi LoRa 32 - ESP32 / LoRa / Display OLED | Curto Circuito. (2016). Retrieved December 1, 2018, from <https://www.curtocircuito.com.br/placa-wifi-lora-32-esp32-lora-display-oled.html>
- Raj, C., Jain, C., & Arif, W. (2018). HEMAN: Health monitoring and nous: An IoT based e-health care system for remote telemedicine. *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017, 2018–Janua*, 2115–2119. <https://doi.org/10.1109/WiSPNET.2017.8300134>
- Rodríguez, J., Montenegro, C., & Cueva, J. (2015). Introducción al Internet de las Cosas introduction of internet of things &quot; IOT. *Redes de Ingeniería*, 6, 53–59. <https://doi.org/10.14483/udistrital.jour.redes.2016.1.a04>
- Salazar, J., & Silvestre, S. (2014). Internet de las cosas. *Internet de Las Cosass*, 24. <https://doi.org/10.1016/j.cirp.2011.03.145>
- Society, H. R. (2013). Ritmo cardíaco lento.

Universidad de Alicante. (2014). *Introducción a JMS (Java Message Service) Índice*. Retrieved from <http://www.jtech.ua.es/j2ee/publico/mens-2010-11/sesion01-apuntes.pdf>

Wang, S., Hou, Y., Gao, F., & Ji, X. (2017). A novel IoT access architecture for vehicle monitoring system. In *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016* (pp. 639–642). IEEE. <https://doi.org/10.1109/WF-IoT.2016.7845396>

World Health Organization. (2016). *Global diffusion of eHealth: making universal health coverage achievable. Report of the third global survey on eHealth. Global Observatory for eHealth*. Retrieved from <http://apps.who.int/bookorders.%0Ahttp://www.wipo.int/amc/en/mediation/rules%0Ahttp://www.wipo.int/amc/en/mediation/rules%0AReport of the third global survey on eHealth Global Observatory for eHealth>