

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar el título de Ingeniero de Sistemas y computación.



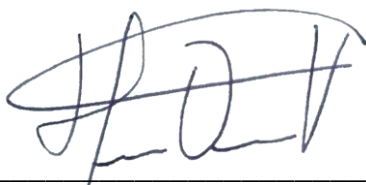
---

**Dr. CAMILO ROCHA NIÑO**  
Decano de la Facultad de Ingeniería



---

**Dr. LUIS EDUARDO TOBÓN LLANO**  
Director Carrera Ingeniería Electrónica.



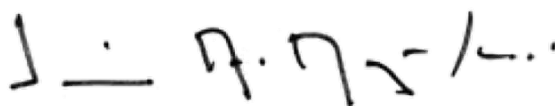
---

**Dr. HERNÁN DARÍO VARGAS CARDONA**  
Director(a) Trabajo



---

**Ing. Stephanie Ordoñez Medina**  
Jurado 1



---

**Dr. Jaime Alberto Aguilar Zambrano**  
Jurado 2



## **Acta de Correcciones al Proyecto de Grado Ingeniería Electrónica**

**Fecha:** 2 de Julio de 2021

**Autores:** Juan Felipe Cifuentes Agudelo, Juan Pablo Navarrete Morales.

**Nombre del Proyecto de Grado:** PROTOTIPO DE UN SISTEMA DE RETROALIMENTACIÓN VISUAL CON MONITOREO Y TRAZABILIDAD APLICADO A UN MANIQUÍ DE RCP LAERDAL RESUSCI ANNE.

**Director:** Dr. Hernán Darío Vargas Cardona

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de Director(a) del Proyecto de Grado

Pontificia Universidad Javeriana Cali  
Facultad de Ingeniería.  
Ingeniería Electrónica.  
Proyecto de Grado.

Prototipo de un sistema de retroalimentación visual con  
monitoreo y trazabilidad aplicado a un maniquí de RCP Laerdal  
Resusci Anne.

Juan Felipe Cifuentes Agudelo  
Juan Pablo Navarrete Morales

Director: Dr. Hernán Dario Vargas Cardona

30 de junio de 2021



Santiago de Cali, 30 de junio de 2021.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Luis Eduardo Tobón Llano

Director Carrera de Ingeniería Electrónica.

Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que los estudiantes de Ingeniería Electrónica Juan Felipe Cifuentes Agudelo (cod: 2030958) y Juan Pablo Navarrete Morales (cod: 2273242) trabajaron bajo mi dirección en el proyecto de grado titulado “Prototipo de un sistema de retroalimentación visual con monitoreo y trazabilidad aplicado a un maniquí de RCP Laerdal Resusci Anne.”.

Atentamente,

A handwritten signature in blue ink, appearing to read 'H D V', with a horizontal line drawn across the middle of the letters.

---

Dr. Hernán Dario Vargas Cardona

Profesor

Departamento de Electrónica y Ciencias de la Computación

Santiago de Cali, 30 de junio de 2021.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Luis Eduardo Tobón Llano

Director Carrera de Ingeniería Electrónica.

Cali.

Cordial Saludo.

Nos permitimos presentar a su consideración el proyecto de grado titulado “Prototipo de un sistema de retroalimentación visual con monitoreo y trazabilidad aplicado a un maniquí de RCP Laerdal Resusci Anne.” con el fin de cumplir con los requisitos exigidos por la Universidad y optar al título de Ingeniero Electrónico.

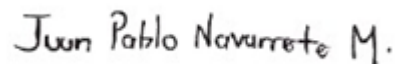
Al firmar aquí, damos fe que entendemos y conocemos las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del proyecto y del trabajo de grado.

Atentamente,



---

Juan Felipe Cifuentes Agudelo  
Código: 2030958



---

Juan Pablo Navarrete Morales  
Código: 2273242

# Resumen

El presente proyecto de grado tiene como objetivo el diseño y la implementación de un prototipo de sistema de retroalimentación visual con monitoreo y trazabilidad aplicado a un maniquí de reanimación cardio pulmonar (RCP) para el entrenamiento de los estudiantes de ciencias de la salud de la Pontificia Universidad Javeriana Cali.

Para el desarrollo de este trabajo fue necesario tener conocimientos de los elementos más importantes que se deben tener en cuenta en la práctica de RCP expuesto por las guías de la Asociación Americana del Corazón (AHA). A través del estudio de esta guía se logró identificar las variables más importantes a medir cuando se realiza esta actividad las cuales son la profundidad de las compresiones de pecho, el número de compresiones por minuto y el retroceso que se define como la liberación del pecho después de efectuar una compresión. Una vez teniendo claras las variables más importantes se procede a diseñar el sistema que logre cumplir los requerimientos establecidos anteriormente mencionados. Para cumplir con este objetivo se propuso programar un microcontrolador el cual realice mediante modelos la cuantificación de las variables obtenidas por un sensor ultrasónico y puedan ser presentadas en una interfaz gráfica al usuario. Una vez se obtiene la cuantificación de las principales variables de RCP en el prototipo del sistema, se realiza un registro de los datos generados durante la actividad en un tiempo de 1 minuto definido así por las guías de la AHA y finalmente el prototipo se conecta a una aplicación del internet de las cosas (IoT) donde se almacenan los datos y se lleva una trazabilidad de las variables que puedan servirle al estudiante para una mejor retroalimentación en su práctica.

A continuación, se implementa el sistema diseñado y para esto fue necesario un proceso de selección de materiales y componentes que se ajustaran a la necesidad del proyecto con base en antecedentes de trabajos relacionados en medición de variables básicas de la reanimación cardiopulmonar y que tuvieran una buena relación costo beneficio.

Para esta fase del proyecto se capturaron los datos del sensor y se caracterizaron. Fue necesario hacer un acondicionamiento de la señal para calcular de una manera más adecuada las variables de profundidad, frecuencia de las compresiones y el retroceso de la compresión. Una vez cuantificadas las variables se presentaron en la interfaz gráfica a través de objetos diseñados bajo la dirección y conocimientos del personal calificado del Hospital Simulado de la Universidad. Finalmente, para esta etapa el prototipo se conecta a la plataforma IoT Node Red donde se almacenan en una base de datos la información y se presenta a través de indicadores.

Por último, se hicieron pruebas de funcionamiento del prototipo para garantizar la funcionalidad del sistema verificando en primera instancia la funcionalidad del sensor, la interfaz gráfica y por último de la trazabilidad de los datos del estudiante. Estas validaciones se realizaron con personal calificado del hospital simulado de la universidad.

**Palabras Clave:** Maniquí de reanimación Cardiopulmonar, Monitoreo y trazabilidad, Internet de las cosas (IoT), Retroalimentación visual, Soporte Vital Básico.

# Abstract

The objective of this degree project is the design and implementation of a prototype of a visual feedback system with monitoring and traceability applied to a cardio pulmonary resuscitation (CPR) manikin for the training of health sciences students of the Pontificia Universidad Javeriana Cali.

For the development of this work, it was necessary to have knowledge of the most important elements that should be taken into account in the practice of CPR exposed by the guidelines of the American Heart Association (AHA). Through the study of these guidelines, it was possible to identify the most important variables to be measured when performing this activity, which are the depth of chest compressions, the number of compressions per minute and the recoil, which is defined as the release of the chest after performing a compression. Once the most important variables were clear, we proceeded to design the system to meet the requirements established above. To meet this objective, it was proposed to program a microcontroller which performs through models the quantification of the variables obtained by an ultrasonic sensor and can be presented in a graphical interface to the user. Once the quantification of the main CPR variables is obtained in the system prototype, a record of the data generated during the activity is made in a time of 1 minute defined by the AHA guidelines and finally the prototype is connected to an Internet of Things (IoT) application where the data is stored and a traceability of the variables that can serve the student for better feedback in their practice is carried.

Next, the designed system is implemented and for this it was necessary a process of selection of materials and components that fit the need of the project based on the background of works related to the measurement of basic variables of cardiopulmonary resuscitation and that had a good cost-benefit ratio.

For this phase of the project, sensor data were captured and characterized. It was necessary to condition the signal to more adequately calculate the variables of depth, compression frequency and compression recoil. Once the variables were quantified, they were presented in the graphic interface through objects designed under the direction and knowledge of the qualified personnel of the Simulated Hospital of the University. Finally, for this stage the prototype is connected to the IoT Node Red platform where the information is stored in a database and presented through indicators.

Finally, the prototype is tested to ensure the functionality of the system, verifying first the functionality of the sensor, the graphical interface and finally the traceability of the student's data. These validations were performed with qualified personal from the university's simulated hospital.

**Key Words:** RCP Manikin, Monitoring and traceability, Internet of Things (IoT), Visual feedback, Basic Life Support (BLS).

# Índice general

<b>1. Descripción del problema</b>	<b>2</b>
1.1. Planteamiento del problema	2
1.1.1. Formulación	4
1.1.2. Sistematización	4
1.2. Objetivos	4
1.2.1. Objetivo General	4
1.2.2. Objetivos Específicos	4
1.3. Justificación	5
1.4. Delimitaciones y Alcances	6
<b>2. Desarrollo del proyecto</b>	<b>7</b>
2.1. Antecedentes	7
2.2. Marco Teórico	12
2.2.1. Definiciones	14
2.3. Marco metodológico	19
2.3.1. Teoría de control aplicado en el sistema: Humano - Maniquí instrumentado	19
2.3.2. Definición de requerimientos	20
2.3.3. Modelo en bloques teórico propuesto para la solución a la pregunta problema	21
2.3.4. Proceso de selección de componentes de Hardware y Software	22
2.3.4.1. Sensores	22
2.3.4.2. Microcontrolador	28
2.3.4.3. Visualización / Interfaz de usuario	29
2.3.4.4. Servidor IoT	30
2.3.4.5. Protocolo de comunicación	32
2.3.5. Modelo en bloques detallado del sistema propuesto	34
2.3.6. Diagrama de flujo del algoritmo del sistema	35
2.3.6.1. Diseño en pseudocódigo para la programación de las rutinas en el microcontrolador	36
2.3.6.2. Diseño de los indicadores para la interfaz gráfica	40
2.3.7. Integración Hardware y Software	41
2.3.7.1. Conexión de los dispositivos y fuente de alimentación	41
<b>3. Resultados del proyecto</b>	<b>44</b>
3.1. Resultados de calibración sensor de distancia ultrasónico HY-SRF05	44
3.2. Resultados implementación de rutinas e indicadores	52
3.3. Prototipo #1 CPR-TRAINING	53
3.3.1. Implementación modelo experimental casero de torso RCP para pruebas	54

---

3.3.2. Socialización virtual prototipo #1 CPR-TRAINING con equipo profesional en Salud Universidad Javeriana Cali . . . . .	56
3.4. Prototipo #2 Sistema de retroalimentación visual CPR-TRAINING . . . . .	56
3.5. Implementación de indicadores en la plataforma IoT Node-Red . . . . .	62
3.5.1. Base de datos MySQL para almacenar los datos del prototipo . . . . .	65
3.6. Evaluación consumo de corriente del prototipo . . . . .	71
3.7. Evaluación del prototipo y resultados finales . . . . .	73
3.7.1. Evaluación del prototipo en el hospital simulado del campus universitario . . .	73
3.7.2. Resultados de la evaluación . . . . .	74
3.7.3. Comparación con otras tecnologías existentes en el hospital simulado . . . . .	79
3.8. Futuros trabajos . . . . .	80
3.9. Conclusiones . . . . .	81
<b>4. Anexos . . . . .</b>	<b>83</b>
4.1. Códigos . . . . .	83
4.2. Acta Reunión 001 . . . . .	111
4.3. Acta Reunión 002 . . . . .	115
4.4. Manual de usuario . . . . .	118
<b>Bibliografía . . . . .</b>	<b>129</b>

# Índice de figuras

1.1. Ubicación de las manos sobre el pecho del maniquí para realizar RCP. Imagen tomada de [26] . . . . .	3
2.1. Izquierda: Mal retroceso, derecha: Retroceso OK. Tomado de [16] . . . . .	14
2.2. Sistema Humano - Maniquí instrumentado. Fuente: Autores . . . . .	19
2.3. Modelo teórico en lazo cerrado Sistema Humano - Maniquí instrumentado. Fuente: Autores . . . . .	20
2.4. Modelo en bloques general del sistema. Fuente: Autores . . . . .	21
2.5. Bosquejo vista lateral izquierda maniquí de RCP . . . . .	23
2.6. Sensor de distancia de tiempo de vuelo VL6180x vista superior con dimensiones . . . . .	24
2.7. Sensor Ultrasónico HC-SR04 vista frontal . . . . .	24
2.8. Sensor Ultrasónico HY-SRF05 vista frontal . . . . .	24
2.9. Sensor distancia SHARP GP2Y0D815 . . . . .	25
2.10. Diagrama en bloques específico del sistema . . . . .	34
2.11. Diagrama de flujo del algoritmo del sistema . . . . .	35
2.12. Pseudocódigo para estimar la profundidad con el sensor HY-SRF05 . . . . .	37
2.13. Pseudocódigo para estimar las compresiones acumuladas con el sensor HY-SRF05 . . . . .	38
2.14. Pseudocódigo para estimar el retroceso del tórax con el sensor HY-SRF05 . . . . .	39
2.15. Indicadores Prototipo CPR-TRAINING #1. Fuente: Autores . . . . .	40
2.16. Indicador retroceso de 2 estados 1.Rojo: Incompleto 2. Negro: Completo Prototipo CPR-TRAINING #1. Fuente: Autores . . . . .	41
2.17. Módulo regulador - convertidor DC-DC ideal para sensores y microcontroladores [47].	42
2.18. Prototipo CPR-TRAINING #1 en protoboard. . . . .	43
3.1. Vista superior sensor HY-SRF05 con longitud de su Tx y Rx . . . . .	46
3.2. Calibración sensor imagen 1 . . . . .	46
3.3. Calibración sensor imagen 2 . . . . .	46
3.4. Calibración sensor imagen 3 . . . . .	47
3.5. Matriz de muestras del sensor ultrasónico . . . . .	48
3.6. Vector fila con el promedio de cada columna . . . . .	48
3.7. Curve Fitting Toolbox . . . . .	49
3.8. Desviación estándar datos del sensor ultrasónico . . . . .	49
3.9. Varianza datos del sensor ultrasónico . . . . .	50
3.10. Vector fila con el promedio de cada columna después de la recta de ajuste . . . . .	50
3.11. Curvas de distancia teórica vs experimental antes de aplicar linealización . . . . .	51
3.12. Curvas de distancia teórica vs experimental después de aplicar linealización . . . . .	52
3.13. Indicadores implementados en el microcontrolador Prototipo CPR-TRAINING #1. . . . .	53

3.14. Resorte maniquí RCP Laerdal vista frontal . . . . .	55
3.15. Resorte maniquí RCP Laerdal vista superior . . . . .	55
3.16. Prototipo CPR-TRAINING #1 vista completa . . . . .	55
3.17. Prototipo CPR-TRAINING #1 vista interior . . . . .	55
3.18. Indicadores Prototipo CPR-TRAINING #2. . . . .	57
3.19. Indicadores implementados en el microcontrolador Prototipo #2 CPR-TRAINING. . . . .	58
3.20. Circuito esquemático Prototipo #1 CPR-TRAINING. . . . .	59
3.21. Circuito PCB Prototipo #1 CPR-TRAINING. . . . .	59
3.22. Montaje componentes en PCB vista 1 . . . . .	60
3.23. Montaje componentes en PCB vista 2 . . . . .	60
3.24. Montaje componentes en PCB vista 3 . . . . .	60
3.25. Cubierta superior caja del proyecto . . . . .	60
3.26. Alojamiento de componentes . . . . .	61
3.27. Circuito insertado en la caja del proyecto . . . . .	61
3.28. Interruptor ON/OFF parte superior caja del proyecto . . . . .	61
3.29. CPR-TRAINING totalmente armado . . . . .	61
3.30. Ventana de comandos node-js. . . . .	62
3.31. Ejecución del servidor local Node-Red desde la ventana de comandos node-js . . . . .	63
3.32. Entorno de desarrollo integrado de Node-Red . . . . .	64
3.33. Panel de control XAMPP . . . . .	66
3.34. Conexión SQLyog con el <i>localhost</i> . . . . .	66
3.35. Tabla <i>entrenamiento_2</i> creada . . . . .	67
3.36. Desarrollo del algoritmo en Node-Red para construir indicadores con los datos cuantificados y enviados vía WiFi por el prototipo . . . . .	68
3.37. Consulta a la tabla <i>entrenamiento_2</i> con algunos registros de prueba enviados desde el prototipo . . . . .	69
3.38. DASHBOARD CPR-TRAINING . . . . .	70
3.39. Medición consumo de corriente DC prototipo . . . . .	72
3.40. Ubicación del sensor en el maniquí . . . . .	74
3.41. <i>Metrónomo beats</i> a 90 beat/s . . . . .	75
3.42. Indicador velocidad pantalla: <i>Lento</i> . . . . .	75
3.43. <i>Metrónomo beats</i> a 105 beat/s . . . . .	76
3.44. Indicador velocidad pantalla: <i>OK</i> . . . . .	76
3.45. <i>Metrónomo beats</i> a 125 beat/s . . . . .	76
3.46. Indicador velocidad pantalla: <i>Rápido</i> . . . . .	76
3.47. Prototipo CPR-TRAINING en maniquí NASCO sin armar . . . . .	77
3.48. Prototipo CPR-TRAINING en maniquí NASCO armado completamente . . . . .	77
3.49. Evaluación del prototipo en el maniquí NASCO . . . . .	78
3.50. Métricas del evaluador Oscar Echeverry en el Dashboard . . . . .	79
4.1. Firmware para el microcontrolador ESP32 del prototipo final . . . . .	110

# Índice de tablas

2.1. Selección de sensores de distancia . . . . .	26
2.2. Selección de microcontroladores . . . . .	29
2.3. Selección de pantalla para visualización . . . . .	30
2.4. Consumo en mA teórico de componentes . . . . .	42
3.1. Porcentajes de error antes de realizar linealización . . . . .	50
3.2. Porcentajes de error después de realizar linealización . . . . .	51
3.3. Comparación entre QCPR Laerdal y CPR-TRAINING . . . . .	79
3.4. Comparación entre PRESTAN CPR y CPR-TRAINING . . . . .	80

# Introducción

Según datos y cifras de la Organización Mundial de la Salud (OMS) las enfermedades cardiovasculares son la principal causa de muerte en el mundo y el paro cardiorrespiratorio inesperado lidera el porcentaje en las estadísticas de esta patología. Esto se considera como un problema sanitario de primera magnitud a nivel global [20].

Entre las estrategias que se emplean en el mundo para combatir este fenómeno de salud pública se encuentra la prevención y tratamiento de este tipo de enfermedades. Entre las formas de prevención básicas se encuentran seguir una alimentación saludable y balanceada consumiendo los nutrientes que el cuerpo necesita para su buen funcionamiento acompañado de la práctica del deporte para regular la energía en el organismo. De esta forma se busca evitar situaciones de parada cardiaca y primeros auxilios.

El 60 % de los paros cardíacos ocurren en presencia de testigos y por cada minuto que pasa una persona con un paro cardiaco sin recibir reanimación cardiopulmonar las posibilidades de supervivencia disminuyen en un 10 %, además el porcentaje de sufrir accidente cerebrovascular aumenta, es decir que posiblemente las secuelas neurológicas sean graves [7].

La técnica de la reanimación cardiopulmonar puede ser adoptada y utilizada en casos de emergencia para salvar una vida por cualquier persona en el mundo y no necesariamente debe pertenecer o laborar en el área de la salud, pues mientras llega un servicio médico de emergencia, un reanimador lego (Persona del común con conocimientos en reanimación cardiopulmonar) puede dar los primeros auxilios y garantizar la supervivencia de la víctima mientras el servicio médico hace presencia.

Este trabajo propone el diseño y la implementación de un sistema de retroalimentación visual con monitoreo y trazabilidad para un maniquí de RCP Laerdal Resusci Anne que se convierta en apoyo para los estudiantes de la facultad de ciencias de la salud de la Pontificia Universidad Javeriana Cali respecto al seguimiento de las recomendaciones de las guías de la Asociación Americana del Corazón (AHA).

# Descripción del problema

---

## 1.1. Planteamiento del problema

La salud en los seres humanos y todo organismo vivo en el mundo se define como un estado de equilibrio y bienestar. La Organización Mundial de la Salud (OMS) define el término salud como un estado de completo bienestar físico, mental y social [42].

Las personas para conservar un buen estado de salud deben seguir hábitos que contribuyan al equilibrio y buen funcionamiento del cuerpo humano. Entre estos hábitos se relacionan una alimentación saludable y balanceada que proporcione los nutrientes que el cuerpo necesita para reducir el riesgo de enfermedades y conservar el equilibrio [6]; la práctica del deporte es un hábito fundamental pues garantiza bienestar físico y mental. Cuando una persona en su vida no pone en práctica estos hábitos, las probabilidades de sufrir cualquier tipo de enfermedad son altas incluyendo enfermedades cardiovasculares [42].

Según datos y cifras de la OMS, las enfermedades cardiovasculares son la principal causa de muerte en el mundo y por este motivo es muy común que en cualquier actividad que el ser humano realice en su diario vivir como tomar el transporte público, caminar en un parque, realizar actividades en el área de trabajo o en una instalación deportiva, se presencie un paro cardíaco en el cual se requiera de atención inmediata de primeros auxilios y RCP [20].

En el área de la salud, la RCP es una técnica utilizada para salvar vidas en casos de emergencia cuando se presenta paro cardíaco. Según investigaciones académicas en el área de la salud y el deporte, las probabilidades de supervivencia de una persona que sufre un paro cardiorespiratorio son aproximadamente del 71 % si se actúa entre los primeros 5 minutos del paro cardíaco antes de que llegue un servicio médico de emergencia, además por cada minuto que pasa sin recibir RCP la probabilidad de supervivencia disminuye entre un 7 % y un 10 % si no se le practica esta técnica a la persona afectada [9].

La Universidad Javeriana Cali realiza jornadas de capacitación de RCP constantemente en el Hospital Simulado que está ubicado en el edificio guayacanes. Durante la práctica de RCP se utiliza un maniquí al cual se le dan ventilaciones en la boca (respiración boca a boca) y compresiones

de pecho ubicando las manos sobre la parte central del pecho del maniquí específicamente sobre el hueso esternón y en medio de los pezones (Ver figura:1.1). Las compresiones de pecho según las recomendaciones de las guías de la AHA deben oscilar entre 100 y 120 por minuto [40]. Los maniqués que se utilizan actualmente en la universidad tienen retroalimentación mecánica (sonido por contacto entre piezas mecánicas en el interior del maniquí) y el seguimiento al proceso del estudiante es realizado por cada instructor.



Figura 1.1: Ubicación de las manos sobre el pecho del maniquí para realizar RCP. Imagen tomada de [26]

La AHA diseña unas guías especiales para RCP y atención cardiovascular de emergencia (ACE), donde incluye toda clase de recomendaciones para ser adoptadas por organismos como por ejemplo Salamandra<sup>®</sup>, que certifican a personas profesionales de la salud en esta técnica con el fin de mejorar su formación integral. En la actualización de las guías de la AHA para el año 2015 incluyen una recomendación donde hacen énfasis en los dispositivos de retroalimentación incorporados en los maniqués de RCP para ayudar al aprendizaje de las habilidades psicomotoras en esta técnica [40].

Actualmente en el mercado existen dispositivos de retroalimentación modulares que se le pueden adaptar al maniquí dependiendo del fabricante como por ejemplo el QCPR de Laerdal Medical [46] ó el PRESTAN Adult Series 2000 [1]. Este tipo de dispositivos cumplen con la recomendación de la actualización de las guías de la AHA del 2015 pero les hace falta un sistema que presente los datos al estudiante en un reporte para un buen análisis y retroalimentación al final de la práctica.

Sin embargo, los maniqués que existen actualmente en el Hospital Simulado están en proceso de actualización y esta actualización tecnológica ha permitido identificar oportunidades de mejora para lograr un alto impacto en las habilidades de la técnica y la realización de mediciones para establecer objetivamente métricas relacionadas con la adquisición de habilidades en RCP. De acuerdo con estas oportunidades surgen las siguientes necesidades:

1. Un sistema de retroalimentación visual cuantitativo que indique el estado de las variables al estudiante.

2. Un sistema de monitoreo y trazabilidad con servicios en la nube para el acceso a los datos estadísticos que realice el estudiante.

Se plantea en este proyecto realizar mediciones de las habilidades de la técnica de RCP mediante datos estadísticos.

### **1.1.1. Formulación**

¿Cómo desarrollar un prototipo de sistema de retroalimentación visual con monitoreo y trazabilidad aplicado a un maniquí de RCP Laerdal Resusci Anne que apoye a los estudiantes de ciencias de la salud de la Pontificia Universidad Javeriana Cali en el seguimiento a los entrenamientos de RCP según las recomendaciones de las guías de la AHA?

### **1.1.2. Sistematización**

- ¿Qué características funcionales debe cumplir el diseño del prototipo ?
- ¿Cómo implementar un prototipo de sistema de retroalimentación visual con monitoreo y trazabilidad que incluya tecnologías acerca del Internet de las cosas - IoT?
- ¿Cómo se estimarán sus parámetros y evaluará su desempeño?

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Desarrollar un prototipo de sistema de retroalimentación visual con monitoreo y trazabilidad aplicado a un maniquí de RCP Laerdal Resusci Anne que apoye a los estudiantes de ciencias de la salud de la Pontificia Universidad Javeriana Cali en el seguimiento a los entrenamientos de RCP según las recomendaciones de las guías de la AHA.

### **1.2.2. Objetivos Específicos**

- Diseñar un prototipo de sistema de retroalimentación visual con monitoreo y trazabilidad de los datos generados por el estudiante en el momento de su práctica de RCP, que incorpore los elementos más importantes que debe cumplir de acuerdo a las recomendaciones de las guías de la AHA y que incluya servicios de las tecnologías del internet de las cosas para facilitar al estudiante la consulta de sus datos.
- Implementar un prototipo de sistema de retroalimentación visual con monitoreo y trazabilidad que incluya los servicios de las tecnologías del internet de las cosas.

- Evaluar el desempeño del prototipo con al menos un docente de la facultad de ciencias de la salud de acuerdo a las recomendaciones de las guías de la AHA 2015.

### 1.3. Justificación

En la actualidad existen diferentes laboratorios de investigación y diseño de dispositivos médicos que se enfocan en desarrollar soluciones para el mercado que estén a la vanguardia de la tecnología. Con respecto a esto, en el mercado se pueden encontrar dispositivos de retroalimentación para maniqués de RCP que cumplen con las recomendaciones de las guías de la AHA y se encargan de ofrecer al estudiante o profesional de salud una retroalimentación oportuna cuando se realiza la técnica de reanimación cardiopulmonar [43] sin embargo, estas soluciones no disponen para el usuario de un sistema que almacene sus datos. El almacenamiento de datos históricos es una actividad clave para los sistemas de información que por medio de reportes e indicadores presentan al usuario estadísticas de sus resultados.

Continuando con el mismo orden de ideas, se ha comprobado en distintos estudios científicos que el uso de dispositivos de retroalimentación visual de RCP en simuladores de soporte vital básico (maniqués de RCP) mejora el aprendizaje del usuario con la continua realización de las prácticas, por lo tanto, las personas que tienen la posibilidad de aprender y mejorar sus destrezas en esta técnica por medio de dispositivos de retroalimentación visual se encuentran en mayor capacidad de realizar esta labor de emergencia en comparación a las personas que aprenden por la forma convencional (Maniqués de retroalimentación mecánica). [44] [34] [19].

Gracias a que actualmente existen soluciones que satisfacen los lineamientos de las guías de la AHA, se realiza una investigación del estado del arte a estos desarrollos y encontramos que nuestra propuesta es viable técnicamente puesto que el acceso a las herramientas de la computación embebida y el IoT lo permiten [27] Con base en lo anterior se satisface la necesidad de los usuarios del Hospital Simulado de la universidad relacionada con el almacenamiento y tratamiento de los datos remotamente cumpliendo así con los requisitos dentro del dominio del problema.

Para complementar el desarrollo de la propuesta se hace un análisis de impactos que compromete la realización del proyecto. En el ámbito social se espera que los estudiantes logren adoptar la técnica de la reanimación cardiopulmonar potencializando sus habilidades en soporte vital básico y concientizando la importancia de salvar vidas en casos de emergencia [40]. En el ámbito económico se espera que el desarrollo del prototipo cumpla no solo con todas las especificaciones de medición de las variables de sus homólogos en el mercado, sino también con las especificaciones de su componente del internet de las cosas que será su factor distintivo [36] siendo así un valor agregado para la facultad de ciencias de la salud en la búsqueda del continuo mejoramiento de su infraestructura y herramientas de aprendizaje. Por otra parte, el impacto económico también afecta a los usuarios del sistema. Un estudio científico realizado a enfermeras en un hospital de Canadá demuestra que

los intervalos cortos de entrenamiento en RCP son la clave para mantener en alto el desempeño de los reanimadores en la técnica [3] pero implementar jornadas de entrenamiento de cortos periodos en reanimación sin equipos tecnológicos de retroalimentación es costoso debido a la logística que se debe implementar mas los recursos humanos necesarios para las capacitaciones [8], es por esto que la inversión en equipos de retroalimentación para maniqués de RCP es una buena alternativa para que los usuarios en el hospital simulado logren mantener sus estándares de calidad en un alto nivel ejercitando la técnica en cortos periodos de tiempo.

Para concluir, este proyecto propone la construcción de un prototipo de retroalimentación visual cuantitativo que ayude a mejorar el aprendizaje de las habilidades psicomotoras de la técnica de RCP y a su vez proveer de una retroalimentación oportuna tanto al estudiante como al instructor por medio de un sistema de monitoreo y trazabilidad que almacene los datos en la nube y permita realizar el seguimiento al estudiante de su proceso de formación.

## 1.4. Delimitaciones y Alcances

- El prototipo del sistema debe seguir las recomendaciones de las guías de la AHA del 2015.
- El prototipo debe medir al menos las 3 variables principales de la reanimación cardiopulmonar (1. Profundidad de las compresiones de pecho, 2. Velocidad de las compresiones de pecho, 3. Retroceso del pecho).
- El sistema debe generar métricas con los datos generados por el estudiante y gestión de aprendizaje del usuario por medio de reportes.
- La utilización de servicios en la nube para el internet de las cosas puede estar sujeto a costos.
- La verificación del aprendizaje que el prototipo aporte a uno o más estudiantes se encuentra fuera del alcance del proyecto.
- Este proyecto solo se enfoca en resolver las necesidades relacionadas con el almacenamiento y tratamiento de los datos del Hospital Simulado de la Pontificia Universidad Javeriana Cali y no de un mercado objetivo.

# Desarrollo del proyecto

---

## 2.1. Antecedentes

Para desarrollar el planteamiento del problema del presente trabajo se realizó una investigación acerca del estado del arte de las tecnologías relacionadas con esta dependencia de la medicina. Durante el estudio también se investigaron materiales bibliográficos y comerciales relacionados con el tema de manera que contribuyan al enriquecimiento de la información necesaria para el desarrollo del trabajo. A continuación, se enseñan algunos de estos estudios de investigación documentados en artículos académicos:

- **Using a smartphone application (PocketCPR) to determine CPR quality in a bystander CPR scenario a manikin trial [44]**

En este artículo se realizó un estudio comparativo con una muestra poblacional de 100 personas, sin experiencia en el ámbito de la salud, en donde fueron divididos en 4 grupos específicos:

1. Sin instrucciones sobre RCP
2. Guiados por teléfono por centro simulado de RCP
3. Guiados por una aplicación de RCP de teléfono inteligente
4. Asistidos telefónicamente y por medio de la aplicación de teléfono inteligente

Para la evaluación de los diferentes grupos se estudiaron parámetros como profundidad de las compresiones, frecuencia de compresión, correcta reexpansión torácica y adecuada posición de las manos en el tórax. Se obtuvieron resultados donde hubo mayor precisión en un adecuado RCP en el grupo de personas asistido telefónicamente y por medio de la app móvil. Finalmente, el estudio concluye que el uso de los sistemas de información como las aplicaciones móviles en personas sin experiencia en esta técnica favorece a salvar vidas en casos de emergencias.

- **Optimal training frequency for acquisition and retention of high quality CPR skills: A randomized trial [3]**

Este artículo realiza un estudio de investigación para determinar cuál es el intervalo mínimo de tiempo entre prácticas para una persona con poca experiencia en RCP (en este caso enfermeras de un hospital de Canadá) para que se garantice la retención adecuada de las habilidades obtenidas durante su etapa de aprendizaje en la técnica de RCP. Para este estudio se realizaron sesiones de práctica de RCP de dos minutos con maniqués los cuales tenían dispositivos para una posterior retroalimentación. La práctica se dividió en 4 grupos:

1. Grupo con intervalo mensual
2. Grupo con intervalo trimestral
3. Grupo con intervalo semestral
4. Grupo con intervalo anual

Este artículo concluye que el mejor intervalo de tiempo para permanecer con una RCP de “High-Quality” es con prácticas mensuales evidenciados en el programa AHA RQI (American Heart Association - Resuscitation Quality Improvement). Esta metodología la han denominado como RCP de baja dosis y alta frecuencia. Las prácticas con intervalos de 3, 6 y hasta de 12 meses son proporcionales con la pérdida de las habilidades en la técnica.

- **CPR feedback prompt device improves the quality of hands only CPR performed in manikin by laypersons following the 2015 AHA guidelines [34]**

En este artículo se presenta un estudio realizado a un grupo de 124 practicantes lego en RCP divididos en 3 subgrupos:

1. Subgrupo 1. Conformado por 42 personas, regido por las recomendaciones de las guías de la AHA para el año 2010 y con maniquí sin retroalimentación
2. Subgrupo 2. Conformado por 42 personas, regido por las recomendaciones de las guías de la AHA para el año 2015 y sin maniquí con retroalimentación
3. Subgrupo 3. Conformado por 40 personas, regido por las recomendaciones de las guías de la AHA para el año 2015 y con maniquí que incluía dispositivos de retroalimentación

Este estudio tenía como finalidad saber que tanto los dispositivos de retroalimentación en los maniqués le favorecen a los practicantes lego en RCP a mejorar sus habilidades.

En conclusión, el estudio demostró que las recomendaciones de la AHA para el año 2015 en combinación con dispositivos retroalimentados enriquecen satisfactoriamente las habilidades

de los practicantes lego, no obstante, cuando se entrena en la técnica de RCP, la variable más difícil de mantener controlada es la profundidad de la compresión.

- **Development of a smart backboard system for real-time feedback during CPR chest compression on a soft back support surface [19]**

En esta investigación que tiene relación directa con el trabajo que se está planteando se realiza todo un estudio en función a las variaciones que tiene realizar la técnica de RCP en una superficie rígida comparada con una superficie blanda en donde los dispositivos de retroalimentación para RCP disponibles en el mercado presentan variaciones significativas para las 2 distintas formas de apoyar el torso del cuerpo humano sobre una superficie.

La investigación plantea la utilización de 2 sensores tipo acelerómetro y un sensor de fuerza para mejorar la sensibilidad del usuario que se entrena en esta técnica para brindar los primeros auxilios. Según el estudio, la variable más difícil de controlar es la profundidad de la compresión y los cálculos arrojan que la utilización de estos sensores favorece el mejoramiento de la sensibilidad del usuario en este aspecto. La conclusión de este artículo es que la implementación de estos dispositivos en conjunto mejoran la precisión de la profundidad en un 7.3 % comparado con un 24.4 % en sistemas con 1 solo acelerómetro; ejerciendo la técnica tanto en superficie rígida como en superficie blanda, sin embargo, este trabajo se podría haber complementado con herramientas de software que hicieran una correlación de los datos recopilados por los sensores para realizar un registro de seguimiento por usuario de su nivel de capacitación en la técnica y en la variable más difícil de dominar, la profundidad de la compresión de pecho.

- **Affordable Hi-fidelity VR based CPR simulator with Haptics Feedback [15]**

En este artículo se comenta un experimento realizado sobre 2 grupos de personas con un equipo conocido como “VR-CPR”(Virtual Reality - Cardio Pulmonar Resuscitation), el cual tenía como propósito demostrar con métricas que tipo de resistencia de tórax era más apropiada para la simulación de la reanimación cardio pulmonar, si la lineal o la no lineal.

El maniquí estaba equipado con un resorte lineal colocado en una placa de fuerza. Un desarrollo de software fue el encargado de virtualizar el maniquí y por medio de unas gafas de realidad virtual se podía visualizar el avatar al cual se le realizaría la técnica de RCP. Para simular la resistencia del tórax de manera lineal y no lineal se utilizó la retroalimentación de fuerza que proporcionaba el movimiento de la placa de fuerza ubicada en el pecho del maniquí.

La compresión visual del pecho está relacionada lineal o no linealmente con la fuerza medida desde la placa de fuerza. El objetivo específico de este estudio era probar la hipótesis de que muchas compresiones torácicas lineales o no lineales se pueden percibir en la realidad virtual simplemente manipulando la retroalimentación visual.

Los resultados del experimento arrojaron que el equipo que realizo RCP simulando un tórax no lineal (Grupo 1) supero al equipo que realizo la practica simulando un tórax lineal (Grupo 2) con medidas de media y desviación estándar de 91.66 % - 1.70 % para el grupo 1 y 86.39 % - 3.44 % para el grupo 2 respectivamente. Estas estadísticas se calcularon con base en las variables principales que se miden en la reanimación cardiopulmonar: Frecuencia, profundidad y retroceso del pecho.

▪ **Comparing three CPR feedback devices and standard BLS in a single rescuer scenario: A randomised simulation study [57]**

En este articulo se comenta un estudio practicado a 240 estudiantes de medicina capacitados en soporte vital básico. Para la realización del estudio los estudiantes debieron dividirse en 4 grupos (60 personas por grupo) y a cada grupo le correspondería un dispositivo de retroalimentación visual de diferente fabricante excepto un grupo que realizaría el experimento sin ningún tipo de dispositivo de retroalimentación visual.

Los dispositivos que se utilizaron para este experimento son los siguientes:

1. Zoll PocketCPR
2. Laerdal CPRmeter
3. Iphone app Zoll PocketCPR

Los resultados del estudio arrojaron métricas demostrando que la eficacia de la reanimación cardio pulmonar de 2 grupos que utilizaban retroalimentación con dispositivos del fabricante Zoll Medical no mejoraron significativamente pues los valores de frecuencia y descompresión del pecho estaban fuera de los rangos de las guías de la AHA.

Sin embargo, los resultados para el dispositivo CPRmeter del fabricante Laerdal fueron favorecidos puesto que los valores de las variables se encontraban entre los rangos de las recomendaciones de las guías de la AHA. El grupo que realizo la practica sin ningún tipo de dispositivo de mano de retroalimentación obtuvo resultados aproximados a los del grupo que utilizo el CPRmeter.

En conclusión, para este experimento los estudiantes de medicina que no utilizaron dispositivos de retroalimentación visual de mano demostraron que sus habilidades en soporte vital básico estaban en optimas condiciones al igual que los que usaron el dispositivo de Laerdal CPRmeter, mientras que los 2 grupos que usaron el dispositivo de fabricante Zoll obtuvieron pobres resultados que se le atribuyen a retrasos en las mediciones de los equipos.

- **HeartiSense: A Novel Approach to Enable Effective Basic Life Support Training without an Instructor [30]**

En este artículo se propone un sistema educativo de RCP que permita un aprendizaje más preciso el cual posea una mayor efectividad sin necesidad de que se encuentre un instructor y pueda brindar diferentes escenarios por medio de simulaciones. El método de medición sigue las recomendaciones de las guías de la AHA, por lo que este proyecto busca medir presión, velocidad y profundidad al realizar las compresiones. Los sensores que utilizan para la medición de estas variables son: Sensor de fuerza FSR No.406, potenciómetro lineal y sensor barométrico BMP085 para volumen de aire. Cada uno de estos valores obtenidos son enviados en tiempo real a través de Bluetooth.

El sistema posee un módulo de evaluación automatizada cuando se completa la capacitación, los datos del usuario se agregan y analizan para obtener valores significativos según las recomendaciones de las guías de la AHA. Los resultados que se le muestran al usuario son los siguientes:

1. Calificación general
2. Nivel de logro de capacitación
3. Análisis detallado por elemento (profundidad, velocidad, respiraciones)

Las pruebas de usuario se realizaron con 5 aprendices que habían completado el entrenamiento de RCP y en 5 expertos en rescate de emergencia el cual genera una evaluación del dispositivo arrojando un valor de medición de 95 % de los sensores.

- **Effectiveness of feedback with a smartwatch for high-quality chest compressions during adult cardiac arrest: A randomized controlled simulation study [2]**

El propósito de este artículo es comprobar la efectividad del uso de los relojes inteligentes como dispositivo de retroalimentación en la práctica de RCP. Se hace un estudio con 40 estudiantes de medicina capacitados en RCP para saber qué efectividad posee realizar las compresiones torácicas con un reloj inteligente. Los materiales que utilizaron en este estudio fueron un maniquí de entrenamiento de RCP “SkillReporter, Laerdal, Stavanger, Noruega” y el reloj inteligente “Galaxy Gear Live; Samsung Electronics”.

Para el experimento se realiza una aplicación de retroalimentación para el reloj inteligente en la que se proporciona una retroalimentación visual para la profundidad y tasa de compresión por minuto. La retroalimentación visual se muestra al usuario por pantalla desde el dispositivo utilizando tres colores para describir el nivel de profundidad:

1. Azul: Profundidad de la compresión torácica mayor a 6 cm
2. Rojo: Profundidad de la compresión torácica menor a 5 cm

### 3. Verde: Profundidad de la compresión torácica entre 5 cm y 6 cm

Estudiaron los efectos que conlleva realizar la RCP haciendo un estudio con 40 estudiantes que se dividieron en 2 grupos cada uno de 20 estudiantes. El primer grupo realiza las compresiones torácicas sin ningún tipo de retroalimentación visual solo la que indicaba mecánicamente el maniquí (sonido por contacto entre piezas mecánicas en el interior del maniquí), el otro grupo utilizaba el reloj inteligente. Como conclusión se llegó a que el uso de un sistema de retroalimentación que utiliza un reloj inteligente no mejora la profundidad y descompresión promedio, pero sí mejora la proporción compresiones torácicas por minuto, por lo tanto, un reloj inteligente puede ser un buen dispositivo para proporcionar una retroalimentación de cantidad de compresiones por minuto mientras se realiza la RCP.

## 2.2. Marco Teórico

### Contextualización de las tecnologías para la medición de las principales variables en la reanimación cardiopulmonar

Se han mencionado en los antecedentes diferentes tecnologías para medir la frecuencia y profundidad de las compresiones de pecho implementadas en proyectos científicos alrededor del mundo. A continuación, se realizará una inspección a las variables involucradas en el ejercicio de la reanimación cardiopulmonar exclusivamente para este proyecto y se establecerá una relación entre tipos de sensores y los principios fundamentales físicos bajo los cuales operan.

#### Frecuencia de las compresiones de pecho:

Cuando se realiza la técnica de RCP es importante contar la cantidad de compresiones de pecho que se le realizan al paciente y en cumplimiento a las recomendaciones de las guías de la AHA esta actividad debe oscilar entre 100 y 120 cpm. Debido a que la variable a medir corresponde al conteo de la variación de un movimiento en dos direcciones se debe caracterizar dicho movimiento, Por lo tanto, se esperaría que la señal se pueda caracterizar de forma senoidal donde se logre obtener el periodo y la frecuencia de las compresiones de pecho.

Para cumplir con este objetivo se pueden considerar una serie de sensores que satisfagan las necesidades del problema. Entre los tipos de sensores para lograr medir el conteo del movimiento se pueden considerar los siguientes:

- **Sensores de distancia:** El sensor de distancia En teoría entrega al sistema un dato numérico correspondiente al tiempo que tarda la señal en viajar de ida y regreso hasta el objeto bien sea de tecnología ultrasónica o haz de luz infrarrojo *ToF*. Este sensor ubicado en un maniquí de RCP y funcionando bajo las condiciones de una practica de RCP en su salida arroja lo que teóricamente se aproxima a una señal del tipo senoidal correspondiente a los desplazamientos

de las compresiones de pecho y se puede considerar la idea de un algoritmo que detecte el número de compresiones por medio del desplazamiento en un determinado tiempo.

- **Sensores de fuerza:** Este tipo de sensor comúnmente se encuentra de resistencia variable o piezo eléctrico. Partiendo de la idea de que en el RCP se debe liberar el pecho entre compresiones, resulta preciso considerar una variación significativa en la fuerza sometida al pecho durante cada ejecución de compresión y con base en esto estimar el número de compresiones realizadas.
- **Sensores de movimiento:** Los acelerómetros son sensores que detectan su propio movimiento y con base en esto ejercen una variación de la señal que emiten[24]. Partiendo del estudio realizado en los antecedentes del proyecto, con este tipo de sensor se lograría calcular de manera aproximada el número de compresiones de pecho realizadas durante la reanimación cardiopulmonar.

#### **Profundidad de las compresiones de pecho:**

La profundidad de la compresión del pecho es una variable que se debe tener controlada al igual que la de el numero de compresiones realizadas para garantizar una buena ejecución de la técnica de RCP. En cumplimiento a las recomendaciones de las guías de la AHA, la profundidad del pecho debe ser controlada idealmente hasta 50 mm y como máximo hasta 60 mm. Con base en esto se tiene claridad de que la variable se debe medir en unidades de longitud (m) y a continuación se realizará una inspección de los sensores que pueden aplicar para la medición.

- **Sensores de distancia:** Al igual que en la variable del número de compresiones, los sensores de distancia aplican para medir esta variable puesto que pertenece a la magnitud física para la cual han sido diseñados.
- **Sensores de fuerza:** Entre los antecedentes del proyecto se encuentran investigaciones que involucran al sensor de fuerza para medir la profundidad de las compresiones de pecho. Para este tipo de medición con sensores de fuerza se debe realizar una caracterización del sensor en función a la constante del resorte que incorporan los maniqués de RCP.
- **Sensores de movimiento:** El sensor acelerómetro se puede utilizar para medir distancias basado en estudios científicos mencionados en los antecedentes del proyecto. Existen diferentes métodos probados experimentalmente para calcular distancias con el acelerómetro y se diferencian entre si dependiendo de la dimensión en la cual capturen la variación de la aceleración, bien sea temporal o frecuencial.

#### **Retroceso del pecho:**

La expansión del pecho después de cada compresión es tan importante como mantener en promedio la distancia de la profundidad a 50 mm como recomienda las guías de la AHA. En sí, esta

liberación de pecho es una variable opuesta a la profundidad de las compresiones puesto que se debe garantizar que el tórax retorne completamente a su forma natural entre compresiones.

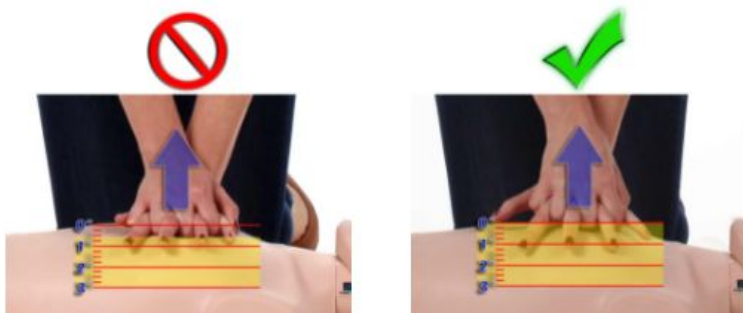


Figura 2.1: Izquierda: Mal retroceso, derecha: Retroceso OK. Tomado de [16]

Para esta variable en teoría aplica cualquiera de los análisis realizados a las 2 anteriores variables puesto que los sensores mencionados también son aplicables debido a que se podrían utilizar cálculos para garantizar el retroceso del pecho sea cercano al inicio de cada compresión.

### 2.2.1. Definiciones

Se agregan todas aquellas bibliografías que dan el soporte conceptual a las ideas expresadas en el planteamiento del problema. A continuación, se enseñan las siguientes definiciones de los conceptos técnicos en los que se fundamenta la investigación:

- **Asociación Americana del Corazón** : [55]

La Asociación Americana del Corazón (AHA, por sus siglas en inglés), fundada por seis cardiólogos en 1924, es una organización sin fines de lucro dedicada a combatir la enfermedad cardiovascular. La AHA proporciona a los profesionales de la salud lineamientos para el tratamiento, basados en estudios científicos, para ayudarles a ofrecer una atención de calidad a sus pacientes y ofrece RCP, con avanzados cursos de formación en primeros auxilios a través de una red de más de cuatro mil centros de formación en 80 países de todo el mundo. La AHA lucha por políticas de salud pública más fuertes e incide por cambios para proteger y mejorar la salud de nuestras comunidades.

- **Guías de la AHA**: [41]

Cada cinco años, y luego de una rigurosa evaluación realizada por los más importantes especialistas en resucitación, la AHA publica las Directrices para RCP y Atención Cardiovascular

de Emergencia (ACE). Desde 1972, estas guías han sido el estándar mundial para el tratamiento de las emergencias cardiovasculares. Se desarrolla con el objetivo de que los profesionales encargados de la reanimación e instructores de la AHA se centren en la investigación de la reanimación y recomendaciones importantes o controvertidas que sean susceptibles a modificar en la práctica o entrenamiento de la reanimación.

Los principales cambios de la actualización de las guías de 2015 para RCP por parte de reanimadores legos fueron los siguientes:

1. La frecuencia recomendada de las compresiones torácicas es de 100 a 120 cpm (compresiones torácicas por minutos, antes era de al menos 100 cpm).
2. Se ha aclarado la recomendación de la profundidad de la compresión torácica para adultos, que es de al menos 5 cm (2 pulgadas), pero no superior a 6 cm (2,4 pulgadas).
3. Se siguen resaltando las características de la RCP de alta calidad: compresiones torácicas con la frecuencia y profundidad adecuadas, permitiendo una descompresión torácica completa tras cada compresión, reduciendo al mínimo las interrupciones en las compresiones y evitando una ventilación excesiva.

■ **Ataque Cardíaco:** [5]

Los ataques cardíacos conocidos popularmente como infartos son emergencias médicas donde el paciente afectado necesita recibir en el menor tiempo posible reanimación cardiopulmonar. Los síntomas de un ataque cardíaco se relacionan con dolores fuertes en el pecho, espalda o brazos, así como también mareos, fatiga, ritmo cardíaco anormal y/o ansiedad. En cualquier sitio se puede presentar un caso de ataque cardíaco sea hogar, trabajo, centro médico, centros deportivos, centros comerciales, centros educativos, entre otros.

■ **Reanimación Cardiopulmonar - RCP:** [48]

Sus siglas significan reanimación cardiopulmonar traducidas al español del término CPR - Cardiopulmonary Resuscitation. Es un procedimiento utilizado en casos de emergencias para salvar vidas cuando una persona ha dejado de respirar y/o su corazón ha detenido las palpitaciones consecuencia de múltiples causas como por ejemplo ataques cardíacos, ahogamiento o fuertes descargas eléctricas.

■ **Microcontrolador:** [51]

Un microcontrolador es un circuito integrado programable capaz de realizar operaciones matemáticas complejas a gran velocidad. Los microcontroladores se componen de una unidad central de procesamiento, unidad de memoria, puertos de entrada y salida de señales digitales y/o analógicas.

- **Interfaz Gráfica:** [21]

La interfaz gráfica de usuario son todos aquellos elementos gráficos que ayudan a la comunicación entre humanos y sistemas. Las interfaces gráficas utilizan un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz.

- **Base de Datos:** [32]

Colección ordenada de datos estructurados propios de un mismo contexto y almacenados de manera sistemática para su utilización. La información contenida en una base de datos se puede actualizar o eliminar.

- **Protocolo MQTT:** [33]

Protocolo de comunicación machine to machine (M2M) utilizado popularmente para el envío de datos entre dispositivos IoT. Sus siglas en inglés significan *Message Queuing Telemetry Transport*. EL protocolo MQTT es ideal para la comunicación bidireccional entre dispositivos a través de la nube donde ocurre el transporte de los mensajes servidor - cliente por medio de publicaciones y suscripciones a los tópicos. Cada vez que un mensaje es publicado en el tópico automáticamente es recibido por todos los dispositivos suscritos al tópico.

- **IoT (Internet of Things):** [29]

El internet de las cosas es un concepto que se entiende como la extensión del internet hacia diferentes dispositivos electrónicos de uso doméstico, industrial, transporte entre otros. Las tecnologías del internet de las cosas brindan la capacidad de monitorear, administrar y controlar dispositivos de forma remota y de crear nuevos conocimientos e información procesable a partir de flujos masivos de datos en tiempo real.

- **Estándar IEEE 802.11:** [28]

Es un estándar para redes inalámbricas definido por el Institute of Electrical and Electronics Engineers (IEEE). se divide en dos capas principales: la capa MAC (Media Access Control) y la capa física o PHY. Estas dos capas permiten hacer una separación funcional del estándar y, lo que es más importante, permite que un único protocolo de datos pueda usarse con distintos métodos de transmisión.

- **Capa de Dispositivos:** [37]

Es la capa inferior de la arquitectura IoT la cual tiene la función de detectar, monitorear, sensor y capturar eventos, niveles y desfases para convertirlas en datos y transmitirlos a un Gateway ubicado en la capa de transporte.

- **Capa de Transporte:** [37]

Está formada por los nodos del ecosistema IoT que intervienen en la comunicación entre algunos dispositivos y la Capa de Red. La transmisión de los datos puede realizarse por medios alámbricos o inalámbricos, generalmente por medios inalámbricos y algunos de estos objetos tienen la capacidad de recibir tareas desde las capas superiores de forma virtual.

- **Capa de Red:** [36]

En la Capa de Red se consideran las diferentes redes que dan soporte a los sistemas IoT. Se caracterizan por transportar y direccionar los datos sensados a las capas encargadas de realizar procesamiento y almacenamiento de los mismos. Está compuesta por todas aquellas redes a través de las cuales se puede acceder a Internet.

- **Capa de Centro de Datos:** [36]

Son utilizadas como herramientas de ayuda, para visualizar y analizar gran cantidad de datos y además, esta capa debe contar con la capacidad de manejar los dispositivos de dichos sistemas y, a su vez, soportar varios de los protocolos que puedan ser empleados en los mismos.

- **Capa de Aplicación:** [36]

La Capa de Aplicaciones es utilizar la información brindada por el resto del sistema IoT para darle valor a la misma. No posee interacción directa con las capas de Centro de Datos.

- **Cifrado:** [27]

Es necesario garantizar la seguridad de los datos y comandos, para ello, es fundamental la Capa de transporte, donde dichos datos y comandos son encriptados para que transiten de manera segura hacia la capa que vayan dirigidos.

- **Procesadores de datos en tiempo real:** [27]

Permite distribuir los datos procedentes de los sensores entre los componentes de las soluciones de IoT pertinentes.

- **TOF (Time of Flight):** [35]

El tiempo de vuelo traducido al español del inglés “Time of Flight” es una técnica utilizada para estimar distancias entre objetos calculando el tiempo transcurrido entre la emisión y la recepción de un haz de luz infrarrojo.

- **Onda Ultrasónica:** [38]

Las señales ultrasónicas son ondas invisibles e inaudibles de alta frecuencia ( $>20\text{KHz}$ ). Los sensores ultrasónicos son detectores de proximidad y medidores de distancia sin fricción, que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. El sensor emite una señal de ultrasonido y cuenta el tiempo que la señal tarda en ir y regresar.

- **Sensor de Fuerza:** [17]

Son elementos para medir fuerza que se aplica sobre las células de carga que conforman el elemento sensitivo. La señal eléctrica generada por el sensor de fuerza en teoría es proporcional a la fuerza aplicada. Existen de diferentes tipos y principios eléctricos como los piezo resistivos.

## 2.3. Marco metodológico

-Para cumplir con el objetivo 1 se realizaron los siguientes procedimientos:

### 2.3.1. Teoría de control aplicado en el sistema: Humano - Maniquí instrumentado

Cuando nos referimos a la teoría de control, el termino *realimentación* se encuentra contenido implícitamente dentro del concepto del objeto de estudio, asumiendo la posición de que el sistema será controlado mediante las correcciones a las acciones de control a través de la observación de la salida de la Planta [39].

La teoría de control aplicada para este modelo en particular relaciona 2 elementos que se encuentran por separado: Humano – Maniquí instrumentado. Estos 2 elementos cuando se integran y entran en funcionamiento logran satisfacer los requisitos para formar un lazo cerrado de control donde el humano realizara unas acciones sobre la planta que, para este caso, será el maniquí de RCP y observara a través de una interfaz gráfica la salida de las señales generadas por la planta como lazo de realimentación. A continuación en la figura 2.2, se expone una ilustración realizada por los autores como ayuda visual para mejorar el entendimiento del concepto.



Figura 2.2: Sistema Humano - Maniquí instrumentado. Fuente: Autores

Ahora bien, si se aplica la teoría de control un poco más detallada al sistema Humano - Maniquí instrumentado, se podría modelar con un sistema en lazo cerrado sin perturbaciones con el propósito de simplificar la explicación del modelo, donde el humano será el encargado de corregir los parámetros y ajustarlos para realizar las acciones de control en la entrada de la planta. A continuación en la figura 2.3, se enseña un diagrama en bloques de un sistema en lazo cerrado sin perturbaciones Humano - maniquí instrumentado:

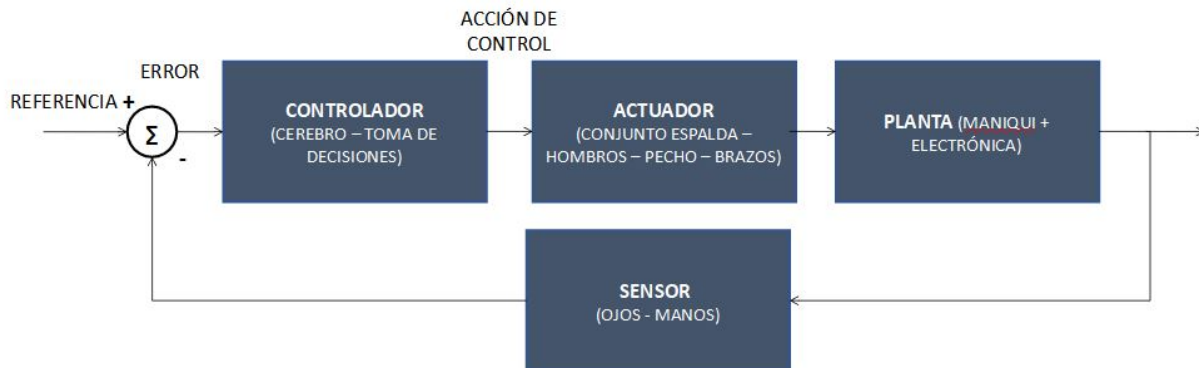


Figura 2.3: Modelo teórico en lazo cerrado Sistema Humano - Maniquí instrumentado. Fuente: Autores

Donde los bloques sensor, controlador y actuador pertenecen al humano debido a que sus ojos y manos en conjunto con la realimentación visual que el dispositivo ofrece funcionan como sensores para alertar a su cerebro de la cantidad de fuerza que esta aplicando con su tren superior (Espalda, hombros, pecho, brazos) sobre el torso del maniquí. El bloque planta pertenece al maniquí instrumentado con la electrónica para la realimentación visual que le sirve al humano para tomar las decisiones en el ejercicio de la técnica de la RCP.

### 2.3.2. Definición de requerimientos

Para dar cumplimiento a cada uno de los objetivos específicos y los alcances propuestos se deben considerar todos los requerimientos funcionales que se pueden extraer del análisis en bloques realizado de la teoría de realimentación para el diseño del prototipo; la elección de componentes y la ejecución del modelo a escala.

Continuando nuevamente con el análisis en bloques realizado en la subsección 2.3.1 se logra contextualizar así el modelo de un dispositivo de realimentación visual para un maniquí de RCP pero, tenemos que esto corresponde a un 50 % del diseño del modelo propuesto, puesto que el porcentaje restante corresponde a la gestión de los datos generados por el dispositivo. La propuesta para el modelo incluye un almacenamiento de los datos en la nube para el monitoreo de las actividades realizadas en el maniquí durante una sesión de reanimación cardiopulmonar.

Con base en las recomendaciones de las guías de la AHA, se tiene que se deben realizar entre 100 y 120 compresiones de pecho por minuto durante la práctica de la RCP, entonces se podría considerar como una sesión de RCP los primeros 60 segundos durante los cuales se deben garantizar el control entre sus rangos de tolerancia de las 3 variables que este proyecto planteo en un principio

medir: Frecuencia de las compresiones, retroceso y profundidad.

A continuación, en la siguiente subsección se plantea un modelo en bloques más generalizado que involucra el sistema de realimentación visual con la gestión de los datos en la nube como propuesta de solución al problema planteado en este trabajo.

### 2.3.3. Modelo en bloques teórico propuesto para la solución a la pregunta problema

Se presenta la concepción de un modelo en bloques como solución a la pregunta problema definida en el primer capítulo de este trabajo. Para esto fue necesario tomar en cuenta la información recopilada en los antecedentes y bases teóricas relacionadas con modelos para la medición de variables en el ejercicio de la reanimación cardiopulmonar más los modelos para el internet de las cosas. Gracias a esto se logró encontrar fundamentos básicos de funcionamiento de estos sistemas y agregar un diseño nuevo como alternativa a lo consultado en el marco teórico. Ver figura 2.4.

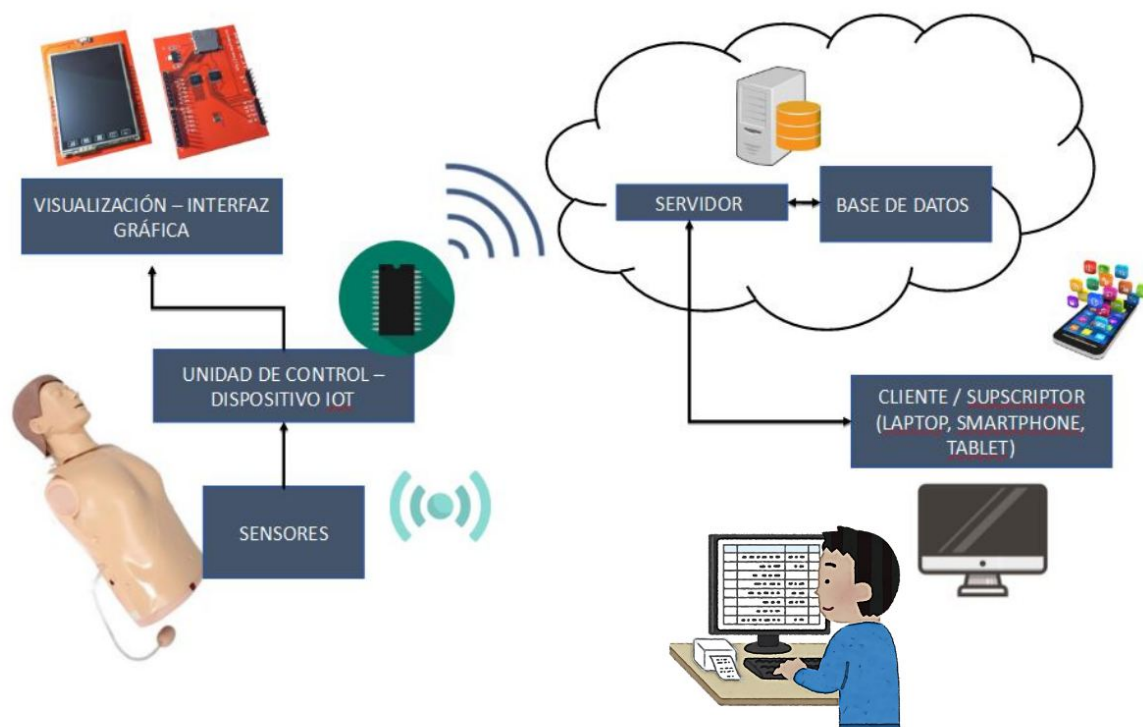


Figura 2.4: Modelo en bloques general del sistema. Fuente: Autores

Si bien se enseña en la figura 2.4, el modelo propuesto involucra el dispositivo de realimentación

visual con la gestión de los datos generados por el sensor para ser transmitidos a través de internet por un procesador especial diseñado para soluciones del internet de las cosas el cual se conectara con un servidor alojado en la nube que cuenta con una arquitectura de software para publicar contenido y posteriormente ser consultado. Para realizar las consultas a los datos publicados se debe considerar la parametrización de un módulo especial para almacenar los datos del sensor y una vez con esto se podrán realizar análisis de datos y visualizaciones.

A partir de esta subsección se expone el proceso metodológico que involucra la etapa de diseño del modelo. Se ha realizado desde los antecedentes una documentación bibliográfica de los componentes e implementos a utilizar para construir el modelo a escala de la solución.

Por medio de la concepción general del sistema descrito en la subsección 2.3.3, se procede a realizar un modelo en bloques más específico que relacione directamente tecnologías a utilizar mediante procesos de selección de componentes electrónicos y software, pero nuevamente para lograr esta actividad primero se realiza la selección de componentes a continuación.

### 2.3.4. Proceso de selección de componentes de Hardware y Software

#### 2.3.4.1. Sensores

La selección del sensor o el grupo de sensores con los cuales se realizara la medición de las 3 variables básicas de la reanimación cardiopulmonar, hacen parte fundamental para la implementación del sistema y por tanto requirió el análisis realizado en la sección 2.2 del marco teórico para elegir instrumentos entre diferentes métodos de medición de estas 3 variables: frecuencia de las compresiones de pecho, el retroceso y la profundidad.

Con el propósito de llevar a cabo la medición de las variables requeridas y con esto lograr una trazabilidad y monitoreo de la práctica de RCP, es necesario un dispositivo capaz detectar cambios de desplazamientos cortos de manera rápida que ayude a calcular la profundidad y velocidad de las compresiones torácicas en menos de 500 milisegundos. Para la aplicación requerida es recomendable utilizar sensores de distancia que puedan ser utilizados en microcontroladores para una fácil integración.

Para llevar a cabo la selección del sensor, se toman las medidas de las dimensiones internas del maniquí que se encuentra en la universidad. La distancia interna desde la espalda hasta el esternón del maniquí es de 14cm aproximadamente (Ver figura 2.5) y según las recomendaciones de las guías de la AHA las compresiones torácicas son de 5 a 6cm, es decir, colocando de ejemplo a un estudiante haciendo las compresiones torácicas, el recorrido que tendría en condiciones ideales seria de 14 a 8cm aproximadamente, sin embargo, eso es en condiciones ideales cuando el estudiante tiene el conocimiento de la técnica de RCP pero, para un estudiante que aún no sabe la técnica de RCP, es

necesario medir un recorrido más amplio como garantía, es decir, que el recorrido caracterizado sea de 14 a 6cm aproximadamente, aunque estas condiciones teóricas planteadas difícilmente aplicarían para el maniquí puesto que mecánicamente el torso tiene una limitante de fábrica que impide que el recorrido de la compresión sea más amplio que la recomendación de las guías de la AHA.

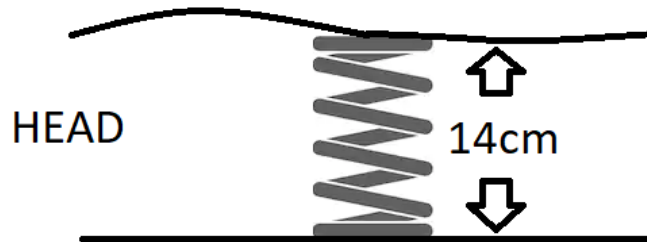


Figura 2.5: Bosquejo vista lateral izquierda maniquí de RCP

Para cumplir con este logro se mostrarán a continuación distintos modelos de instrumentos que cumplen el requerimiento mencionado y se realizará un proceso de selección al alcance de los objetivos planteados.

#### **Sensor ToF VL6180x:[12]**

El sensor VL6180x es capaz de medir la distancia de forma precisa ya que combina las capacidades de medición de luz ambiental y rango de proximidad en un mismo conjunto. Una de sus ventajas es que, al funcionar con láser, tiene un rango mucho más preciso en comparación a otros sensores de distancia ya que no posee ángulo de medición. Puede medir hasta 20cm de distancia por mediciones de tiempo de vuelo. A diferencia de los sensores ópticos más simples que utilizan la intensidad de la luz reflejada para detectar objetos, el VL6180x utiliza la tecnología FlightSense para medir con precisión cuánto tiempo tardan los pulsos emitidos de luz láser infrarroja en llegar al objeto más cercano y reflejarse en un detector, lo que lo convierte en esencialmente un sensor LIDAR de corto alcance. Ver figura 2.6

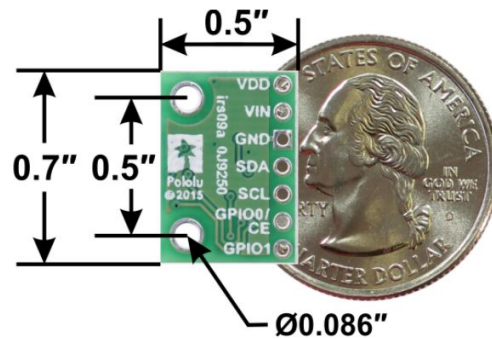


Figura 2.6: Sensor de distancia de tiempo de vuelo VL6180x vista superior con dimensiones

### Sensor Ultrasónico HY-SRF05 y HC-SR04[52][53]

El sensor HY-SR05 capaz de medir distancias por ultrasonidos detectando objetos y calculando su distancia en un rango de 2 a 450 cm. Se destaca el bajo costo, precisión y disponibilidad. Para medir la distancia con este módulo, el microcontrolador envía un pulso de 10 micro segundos el cual dispara una serie de ocho pulsos cortos de ultrasonido que emite el transmisor ultrasónico. Los pulsos de sonido alcanzan al objetivo y vuelven reflejados; viajan a 343 metros por segundo aproximadamente. Por otra parte, el HC-SR04 es un homólogo del HY-SR05 y su rango de movimiento es de 2 a 400 cm, menor en comparación al HY-SRF05 así como también su precisión es baja en comparación con el primero.

La distancia al objetivo es el tiempo que transcurre entre la transmisión y la recepción de los pulsos ultrasónicos, multiplicado por 343 metros por segundo y dividido por dos (para contar la ida y el regreso). Ver figuras 2.7 y 3.1

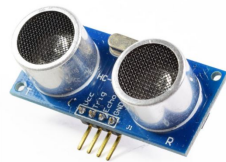


Figura 2.7: Sensor Ultrasónico HC-SR04 vista frontal



Figura 2.8: Sensor Ultrasónico HY-SRF05 vista frontal

### Sensor de distancia infrarrojo SHARP GP2Y0D815[13]

Este sensor infrarrojo es capaz de medir distancias detectando objetos dentro de un rango de 0.5 a 15 cm, posee tres dispositivos: Un diodo emisor de infrarrojos *IRE*D, un detector sensitivo

de posición *PSD* y un circuito procesador de señales. El termino SHARP *Agudo* es porque tiene un rango de visión muy reducido, esto porque la luz que emite es puntual. Tiene como ventaja no afectar la luz ambiental o el Sol, además el circuito procesador de señales filtra y elimina cualquier fuente de luz diferente a la emitida por el sensor. Ver figura 2.9

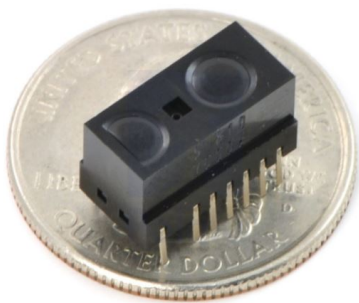


Figura 2.9: Sensor distancia SHARP GP2Y0D815

Para comenzar con la selección del sensor, se puede elegir una de las variables que entre sus métodos de medición se ajusta más a los instrumentos anteriormente descritos para los cuales aplica su comportamiento físico: La profundidad.

La medición de la profundidad en el ejercicio de la reanimación cardiopulmonar se debe manejar en milímetros para garantizar una resolución apropiada en el dispositivo de visualización del movimiento y de esta manera el usuario puede sentir más confianza en el momento que realiza trabajo sobre el maniquí de RCP. Ahora por otra parte, en seguimiento a las recomendaciones de las guías de la AHA, la profundidad de las compresiones debe ser idealmente partiendo desde 0 hasta 50 mm y a lo máximo 60 mm para garantizar que el movimiento de la compresión no lesione a un paciente en la práctica real. Se debe tener otra consideración más con respecto a la medición de la profundidad de la compresión y es la frecuencia de cambio de la variable en función del tiempo, es por esto que el sensor a elegir debe tener la característica de ajustar en fracción de segundos su salida en función del cambio de longitud al cual sea sometido.

Una vez con todo esto descrito se realiza una tabla comparativa entre sensores para medir distancias con base en sus principios de funcionamiento, frecuencia de trabajo, disponibilidad, tamaño, costos, voltajes de alimentación y otros parámetros técnicos.

Nombre	Frecuencia pulsación	Precisión	Rango medida	Tamaño	Angulo medición	Voltaje Alimentación	Moneda	Costo	Disponibilidad
HC-SR04	40KHz	3 mm	2cm - 400cm	40mm x 20mm x 15mm	15°	5VDC	COP	\$10.000	Nacional
HY-SRF05	40KHz	2 mm	2cm - 450cm	40mm x 20mm x 15mm	15°	5VDC	COP	\$12.000	Nacional
VL6180X	33MHz	1 mm	0.5cm - 10cm	4.40mm x 2.40mm x 1 mm	0°	2.8VDC	COP	\$40.000	Nacional
SHARP GP2Y0D815Z0F	38KHz	2mm	0 - 15cm	13.6mm x 7mm x 7.95mm	0°	6.2VDC	COP	\$29.900	Nacional

Tabla 2.1: Selección de sensores de distancia

Según el análisis realizado en el cuadro 2.1 de los distintos sensores se determina que todos cumplen con el principio fundamental requerido en este apartado el cual es medir el desplazamiento de las compresiones torácicas. Ahora, los criterios que se tuvieron en cuenta para la selección del componente fueron el rango de medición, tamaño, costo, disponibilidad local, alimentación y operatividad. Para agregar a este apartado, la alimentación del instrumento debe ser común a las fuentes de alimentación que se encuentran en el mercado local lo cual permite ser integrado con mayor facilidad al sistema propuesto, a la vez no generando costos económicos adicionales.

Como conclusión y en función del sistema propuesto, el mejor componente para el alcance de nuestro proyecto es el sensor ultrasónico HY-SRF05. Este sensor se utilizará debido a que se encuentra en el rango de medición requerido (min 5cm, Max 13cm), se puede utilizar en cualquier superficie sin afectarle la luz ambiental o artificial, tiene un bajo voltaje de alimentación, se acopla fácilmente con la mayoría de los microcontroladores, el precio es muy bajo para el presupuesto de un prototipo funcional y finalmente su frecuencia de trabajo es tan rápida como para responder con su precisión de 2mm a los cambios del sistema.

Para continuar con el proceso de selección de sensores fundamentados en la variable física a medir en la práctica de la RCP, se realizará un análisis y se evaluará si el sensor elegido para medir profundidad aplica para medir las 2 variables restantes.

Primero, comenzaremos analizando la frecuencia de las compresiones de pecho. Tomando como punto de partida las recomendaciones de las guías de la AHA, las compresiones de pecho deben estar en un rango entre 100 a 120 por minuto pero, para que las compresiones sean efectivas, deben realizarse en función a la profundidad ideal que la guía recomienda: 50 mm a lo menos.

En la sección 2.2 de las bases teóricas se comenta acerca de medir las compresiones de pecho con instrumentos de resistencia variable como los *Force Sensing Resistor-FSR*, que se conectan a

un circuito y varían la caída de tensión en función a la fuerza aplicada sobre la celda resistiva. Considerar usar este sensor implicaría realizar una calibración de las compresiones de pecho en función de la constante del resorte del maniquí de RCP.

Por otra parte, se tienen los sensores para medir movimientos en un sistema: los acelerómetros. Este tipo de sensor se podría utilizar para medir la frecuencia de las compresiones de pecho haciendo uso de la integración numérica en tiempo discreto para lograr obtener velocidad y posición a partir de la señal de aceleración.

Otra manera de realizar este cálculo es con el sensor de distancia con el cual se mide la profundidad de las compresiones partiendo del concepto teórico en donde una compresión de pecho es efectiva siempre y cuando se logre a lo menos la profundidad recomendada en las guías de la AHA que sería 5 cm por tanto, resultaría óptimo la utilización del concepto con el mismo sensor para medir 2 de las 3 variables.

Por último, la variable que resta por analizar su comportamiento físico y asignar un sensor que interprete su salida es la variable del retroceso del pecho. Las guías de la AHA recomiendan que siempre debe garantizarse el completo retroceso del pecho tras cada compresión y este concepto se puede interpretar como aproximar la medición de la variable que mide la profundidad hasta 0 cm.

Nuevamente citando la sección de las bases teóricas de este trabajo, se deja comentado en el análisis para la variable de retroceso que una de las formas para obtener su medida era con sensores de celda resistiva que al aplicar fuerza sobre ellos varía la caída de tensión en un circuito. Un sensor *FSR* para este propósito resultaría apropiado para garantizar que el retroceso del pecho se está realizando de manera correcta asumiendo que la fuerza ejercida sobre la celda resistiva es en teoría nula cuando se permite la liberación de la caja torácica.

Por otra parte, otro modo de lograr el objetivo para la liberación de la caja torácica es utilizando el valor cuantitativo de la profundidad que arroja el sensor de distancia. Para este caso se validaría que el valor de profundidad del sensor se aproxime a 0 cm. El propósito de saber la medida del retroceso es garantizar que el usuario permita que la caja torácica en este caso del maniquí de RCP retorne a su forma natural para nuevamente aplicar una compresión y resulte efectiva en seguimiento a los lineamientos de las guías de la AHA.

Finalmente para darle conclusión a esta sección se decide realizar la implementación utilizando el sensor de distancia ultrasónico HY-SRF05 para medir las 3 variables. Esta decisión estuvo basada en los siguientes criterios: Precio, rango de medición, tamaño, disponibilidad (En caso de necesitar un reemplazo/repuesto de carácter urgente), Objeto detectable (Detección no afectada por materiales y/o colores de objeto). El precio del sensor se encuentra entre los más bajos y además se encuentra con disponibilidad a nivel nacional, su tamaño es adecuado a pesar de ser más grande que los sensores *ToF*. El rango de medición del sensor se ajusta a los requerimientos y a las dimen-

siones internas del maniquí y su principio de funcionamiento por ultrasonido no se ve afectado por la oscuridad ni el color del material al interior del maniquí.

#### 2.3.4.2. Microcontrolador

Una vez teniendo el tipo de sensor que se implementara se procede a elegir el tipo de microcontrolador que se utilizara. Para esta actividad se tuvieron en cuenta los siguientes puntos:

- El microcontrolador deberá estar constantemente realizando operaciones para obtener los valores de las medidas del sensor en la pantalla de visualización y además deberá enviar datos en un corto tiempo, por tanto, la velocidad del procesador es un rubro importante.
- El microcontrolador deberá tener los suficientes puertos GPIO para satisfacer los requerimientos (Al menos 10).
- El microcontrolador debe tener la capacidad de enviar datos de manera inalámbrica hacia un servidor en la nube.
- El voltaje de alimentación debe encontrarse en valores comunes o también podría considerarse la utilización de una batería externa.
- El microcontrolador debe ser compatible con un lenguaje de programación de desarrollo libre y que permita la integración de hardware y software.
- La disponibilidad, el precio y el tamaño son criterios que definen su elección

Teniendo en cuenta los puntos anteriormente descritos se realiza una matriz de selección con algunos microcontroladores en el mercado:

Nombre	Procesador	Memoria Ram	WiFi	IDE	Puertos GPIO	Voltaje Alimentación	Moneda	Costo	Disponibilidad
Arduino UNO	ATmega328P	2 kB	No	Micropython Arduino IDE	+10	5VDC USB 7-12 VDC	COP	\$36.000	Nacional
Arduino Nano	ATmega328 16MHz	2 kB	No	Micropython Arduino IDE	-10	7-12 VDC	COP	\$14.000	Nacional
ESP32	LX6 32bit Dual Core a 160 MHz (hasta 240 MHz)	520 kB	802.11 b/g/n (Hasta 20dBm) WEP, WPA	Micropython Arduino IDE	+10	5VDC	COP	\$48.000	Nacional
ESP8266	Tensilica LX106 32 bit a 80MHz (Hasta 160MHz)	80 kB (40kB disponibles)	802.11 b/g/n (Hasta 20dBm) WEP, WPA	Micropython Arduino IDE	-10	5VDC	COP	\$22.000	Nacional
Raspberry Pi 3 Model B+	Broadcom BCM2837B0 Cortex-A53	1 GB LPDDR2 SDRAM	Doble banda 2.4 GHz a 5GHz 802.11 b/g/n/ac	C, Python	+10	5VDC	COP	\$196.000	Nacional

Tabla 2.2: Selección de microcontroladores

Haciendo análisis a la tabla 2.2, se realiza la elección del microcontrolador en función a los siguientes criterios: Costo, disponibilidad, Velocidad de procesamiento, conectividad WiFi; es finalmente el ESP32. El costo del microcontrolador es el más alto entre los dispositivos que comparten el IDE de Arduino pero sus características como la conectividad WiFi, la disponibilidad del controlador a nivel nacional, la disponibilidad de más de 10 puertos GPIO, su velocidad de procesamiento, sus 2 núcleos y la alimentación lo hacen la opción más versátil y robusta para el propósito de este trabajo.

### 2.3.4.3. Visualización / Interfaz de usuario

Una vez habiendo elegido sensor y microcontrolador resta por seleccionar la interfaz de usuario.

La interfaz de usuario es parte esencial en el funcionamiento del sistema de cara al usuario final puesto que es la parte visible con la cual se logrará la realimentación visual de las acciones realizadas durante la práctica de la RCP.

Para continuar con este procedimiento se realizará una pequeña lista de condiciones que debe cumplir la interfaz ó siendo más específicos a partir de este momento, la pantalla de visualización deberá cumplir con las siguientes condiciones como mínimo:

- Garantizar la visualización de las 3 variables de la reanimación cardiopulmonar con objetos interactivos que le resulten efectivo al usuario en el proceso de realimentación visual.

- Voltaje de alimentación común que permita ser conectado a la fuente de los otros dispositivos: Microcontrolador y sensores.
- La pantalla debe permitir ser programada en software de desarrollo libre y ajustarse al controlador del ESP32.

Teniendo en cuenta nuevamente los puntos anteriormente descritos se realiza una matriz de selección con algunas pantallas en el mercado:

Nombre	Tamaño	Color	FPS	IDE	Disponibilidad	Voltaje Alimentación	Moneda	Costo
TFT ILI9341	2.2 Pulgadas	Sí	60	Micropython Arduino IDE	Nacional	3.3VDC	COP	\$58.000
LCD OLED 0.96	0.96 Pulgadas	No	30	Micropython Arduino IDE	Nacional	3.3 - 5VDC	COP	\$16.000
LCD QC2004A	20x4 Pixeles	No	30	Micropython Arduino IDE	Nacional	5VDC	COP	\$29.000

Tabla 2.3: Selección de pantalla para visualización

Haciendo nuevamente análisis a la tabla 2.3, se realiza la elección de la pantalla para la visualización de la realimentación al usuario en función de los siguientes criterios: Costo, disponibilidad, FPS - *Frames Per Second*, tamaño e IDE; es finalmente la TFT ILI9341. El costo de la pantalla se encuentra entre el promedio en comparación con sus homologas pero, sus 60 FPS, su tamaño de 320x240 pixeles y sencilla integración con el controlador del ESP32 lo hacen la elección ideal para el desarrollo de este trabajo.

#### 2.3.4.4. Servidor IoT

En función de los objetivos planteados para este proyecto es necesario un entorno de desarrollo integrado para la interfaz de usuario de la plataforma IoT.

Se debe lograr la comunicación entre el microcontrolador y un servidor IoT en la nube para enviar los datos generados por el sensor hacia una base de datos donde se pueda almacenar y posteriormente consultar para realizar observaciones y análisis con el propósito de generar en detalle el comportamiento de las compresiones torácicas y sacar conclusiones durante de la práctica.

Con base en el argumento dado en el anterior párrafo, se realiza una lista de requerimientos que debe ofrecer el servidor para este trabajo:

- El servidor IoT en lo posible debe poseer un lenguaje de programación de desarrollo libre que permita programar algunas funciones de análisis de datos.

- El servidor IoT debe permitir la integración con un motor de base de datos para almacenar la data enviada por el microcontrolador.
- El servidor IoT debe permitir crear dashboard donde el usuario pueda realizar análisis de visualizaciones con los datos.

Se hace un proceso de selección a continuación para la interfaz del usuario IoT teniendo en cuenta los requerimientos planteados anteriormente, estas son las opciones:

### **NODE-RED**[31]

Node-red es una herramienta de programación la cual sirve para conectar distintos dispositivos de hardware y una interfaz de programación de aplicaciones, diseñada para agilizar procesos IoT. Como función principal podremos recibir los datos que se encuentran e integrarlos a una interfaz gráfica. Muestra visualmente las relaciones y funciones, y permite que usuario programe sin tener que utilizar lenguajes de programación como HTML, Java, entre otros. Node-Red es un editor de flujo basado en el navegador donde conectando una serie de nodos entre sí conseguimos que se comuniquen entre ellos.

### **ThingSpeak**[10]

ThingSpeak es una Plataforma de análisis IoT de código abierto creada por Mathworks, los responsables del software Matlab. diseñada para conectar distintos dispositivos de hardware la cual puede guardar y almacenar datos con protocolos de comunicación MQTT y HTTP. Dentro de la Interfaz de ThingSpeak se puede escribir y ejecutar código de MATLAB para realizar trazabilidad y visualizaciones, enviar notificaciones mediante redes sociales como Twitter.

### **HTML**[23]

*HyperText Markup Language* Es el Lenguaje con el que se define el contenido y la estructura de las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y los diversos elementos que la componen: imágenes, listas, vídeos etc. Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, dichas etiquetas se utilizan para describir algo que se quiere representar en una página web. Tiene una estructura clara y es fácilmente comprensible, está dividido en dos bloques `<head>` y `<body>`.

*Hipertexto* se refiere a enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web.

Cabe resaltar que hay más plataformas IoT importantes como lo son: Web Services IoT Platform, Microsoft Azure IoT HUB, IBM Watson IoT Platform, Cisco IoT Cloud Connect, Google Cloud IoT y Sofia2 IoT Platform. Pero estas tienen un costo por el número de dispositivos y/o pago

de almacenamiento y tráfico de datos.

Luego de analizar las opciones planteadas anteriormente, se elige a Node- Red como la interfaz de usuario IoT. Ya que esta ya es una herramienta que simplifica mucho el trabajo, y que facilita el desarrollo de páginas web y comunicaciones para aplicaciones IoT, a diferencia de HTML que se debe tener un nivel de conocimiento importante para el desarrollo de la interfaz. Node Red ofrece una interfaz de usuario tipo dashboard donde se pueden utilizar indicadores para la visualización de la data almacenada a comparación de ThingSpeak que posee una dashboard la cual tiene un número limitado de mensajes al día.

#### 2.3.4.5. Protocolo de comunicación

Para la plataforma IoT escogida es necesario un protocolo de comunicación el cual sea un puente que comunique la interfaz IoT y el microcontrolador, por lo que se plantea los siguientes protocolos de comunicación:

##### **HTTP *Protocolo de Transferencia de Hipertexto***[56]

Es el protocolo de transmisión de información de la World Wide Web. Es un protocolo que sigue un modelo unidireccional, cliente-servidor, el cliente solo puede recibir información en caso de realizar una petición, nunca de forma pasiva. HTTP usa el Puerto 80, que es el puerto desde el que el servidor web acepta peticiones. La mayoría de los recursos son archivos (imágenes, etc.) pero puede incluir otro tipo de datos como *scripts*. Es un protocolo asíncrono por lo que no es ideal para redes de altas latencias, sin embargo, su estructura cliente-servidor, junto con la capacidad para usar cabeceras, permite a este protocolo evolucionar con las nuevas y futuras aplicaciones en Internet.

##### **MQTT *Message Queue Telemetry Transport***[56]

Es un protocolo de mensajería asíncrona, usado para la comunicación *machine-to-machine*: M2M, en el ámbito de IoT. Tiene una estructura Cliente/Servidor basado en publicaciones y suscripciones a los denominados *tópicos*. Cada vez que un mensaje es publicado será recibido por el resto de dispositivos adheridos a un tópico del protocolo. Trabaja perfectamente en dispositivos de baja potencia, frecuentemente utilizados en aplicaciones IoT. El protocolo MQTT funciona sobre TCP/IP o sobre otros protocolos de red con soporte bidireccional y sin pérdidas de datos. Está optimizado para trabajar con un gran número de clientes conectados de forma simultánea. se utiliza en una amplia variedad de industrias, como la automotriz, la fabricación, las telecomunicaciones, el petróleo y el gas.

Realizando una comparación de los protocolos enunciados anteriormente y basado en proyectos que requerían protocolos de comunicación en IoT, se ha considerado que el protocolo MQTT es

el mejor para este tipo de aplicaciones, debido a su facilidad de implementación, su versatilidad y capacidad para trabajar con elementos de poca potencia, además de su estructura de publicación y suscripción es bidireccional por lo que a diferencia del protocolo HTTP, el intercambio de información puede ir en ambas direcciones entre los diferentes elementos que integran el sistema. Es de los protocolos más utilizado para realizar aplicaciones IoT.

Finalmente para esta sección se concluye que el servidor IoT a utilizar es el bróker MQTT Mosquitto en conjunto con la plataforma de programación por flujos y visualización de datos NODE-RED basada en node-js.[22]

El protocolo MQTT en el bróker Mosquitto funciona sobre TCP/IP con soporte bidireccional y sin pérdida de datos. Desde el IDE de Arduino se puede programar el microcontrolador para conectarse con el bróker y transmitir datos del sensor.

La plataforma de programación por flujos NODE-RED se integra con el bróker MQTT y permite conectarse con el microcontrolador para recibir datos a través de la red WiFi. Esta plataforma entre sus funcionalidades tiene la integración con motor de base de datos MySQL que la hace útil para almacenar datos del sensor. Adicional a todo esto NODE-RED tiene una interfaz de usuario tipo dashboard donde se pueden utilizar indicadores para la visualización de la data almacenada.

### 2.3.5. Modelo en bloques detallado del sistema propuesto

Después de haber realizado la selección de cada componente por subsistemas, A continuación se enseñará el diseño del diagrama en bloques en específico del sistema de realimentación visual con monitoreo y trazabilidad para un maniquí de RCP

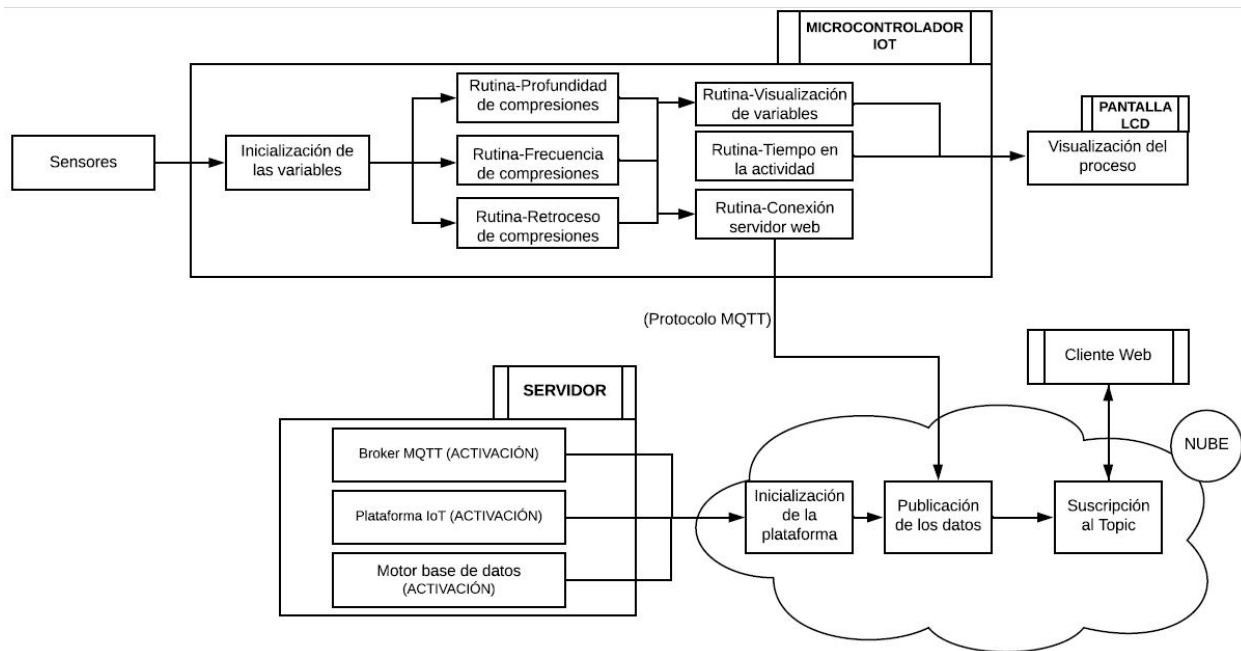


Figura 2.10: Diagrama en bloques específico del sistema

En la figura 2.10 se enseña el diagrama en bloques en específico del sistema propuesto. Con ayuda de este diagrama se tiene un concepto más específico de la solución final a implementar.

Además del diagrama en bloques se presenta un diagrama de flujo del algoritmo del sistema como paso necesario en el proceso de diseño con el objetivo de tener claridad en las entradas y salidas que se desea tener en el sistema.

2.3.6. Diagrama de flujo del algoritmo del sistema

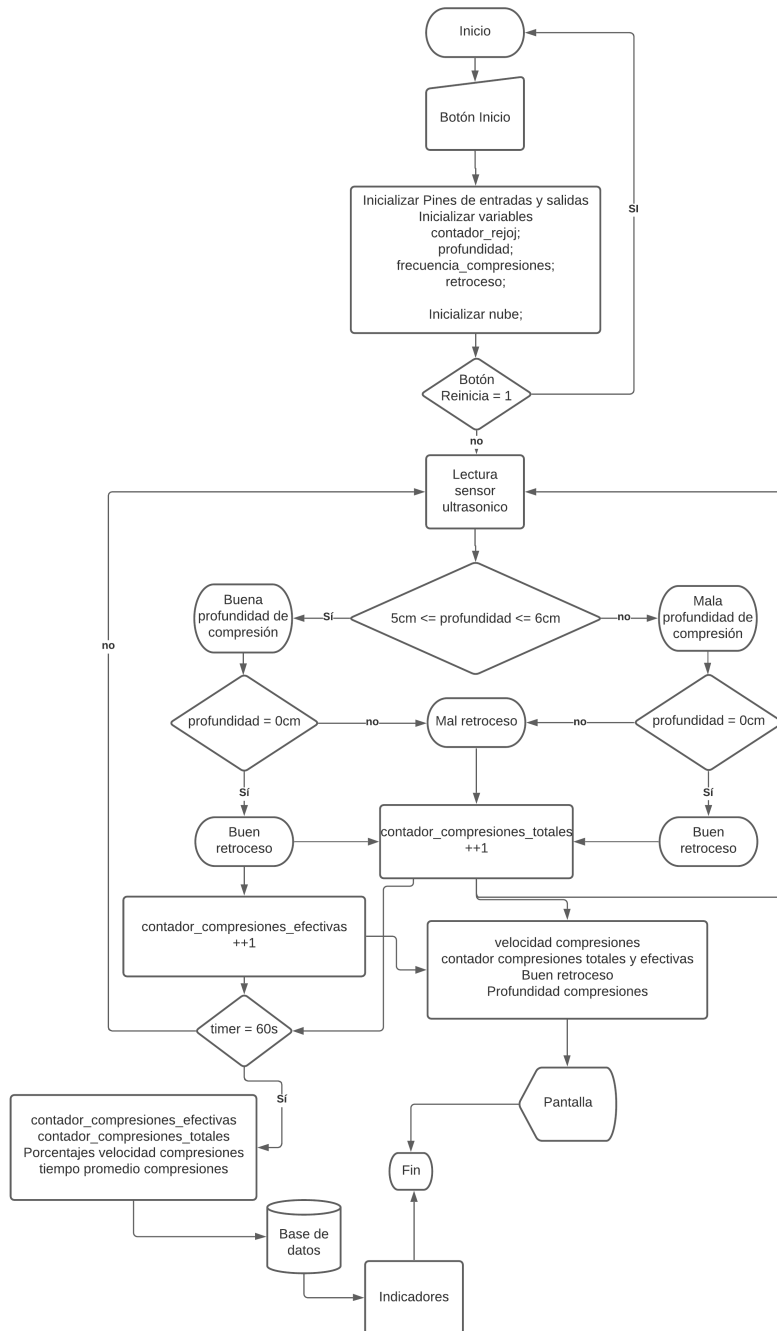


Figura 2.11: Diagrama de flujo del algoritmo del sistema

### 2.3.6.1. Diseño en pseudocódigo para la programación de las rutinas en el microcontrolador

Se tiene que el sensor ultrasónico dispara una onda de ultrasonido hacia un objeto de frente y la señal que rebota es recibida por el receptor. Para realizar la estimación de la distancia al objeto se tiene que la velocidad del sonido es teóricamente 343,2 m/s a 20°C. Ahora, la señal de salida que arroja el sensor es recibida en el microprocesador y por medio de una librería de desarrollo libre parametrizada para este sensor en el IDE de Arduino se puede obtener en una variable el tiempo en milisegundos que tarda la señal en viajar de ida y regreso desde el transmisor hasta el receptor del sensor respectivamente. A continuación, lo que se realiza para estimar la distancia es tomar la velocidad del sonido, multiplicar por la variable que tiene el tiempo en milisegundos y dividir entre 2 el resultado debido a que la señal es de ida y regreso. La expresión se puede detallar en la siguiente fórmula:

$$\text{Distancia(m)} = \frac{(\text{Velocidad del sonido } \frac{m}{s}) \times (\text{Tiempo(s)})}{2} \quad (2.1)$$

#### Algoritmo profundidad

Ahora teniendo la ecuación para la distancia teórica estimada del sensor se tiene que la medida tiene su inicio o punto de partida de 0mm desde el transmisor y la distancia final corresponde al punto del objeto donde rebota la señal. Para invertir este comportamiento de la señal de modo que el punto de partida estuviera en el objeto y no en el sensor se utilizó el siguiente algoritmo descrito en pseudocódigo. Ver figura 2.12

```
1 Distancia de partida = 0
2 Variable para almacenar diferencia = 0
3
4 Función Calcular Distancia () {
5
6   Si Distancia de partida es igual que 0 {
7     Distancia de partida = Distancia actual sensor ultrasónico
8   }
```

```
9
10 Variable para almacenar diferencia = Distancia de partida - ...
    Distancia actual sensor ultrasónico
11 }
```

Figura 2.12: Pseudocódigo para estimar la profundidad con el sensor HY-SRF05

Donde se asigna a la variable *Distancia de partida* el valor de la distancia que calcula el sensor cuando hace el disparo de la señal en el interior del maniquí sin ser sometido a perturbaciones. Luego este valor capturado como la distancia máxima se utiliza para hacer diferencias con los cambios de distancia cuando se realizan las compresiones de pecho.

Una vez se obtiene el método para medir la distancia desde el punto opuesto al sensor se define este paso como el cálculo de la profundidad y se empiezan a formular los demás métodos para realizar los cálculos de la frecuencia de las compresiones y el retroceso.

### Frecuencia de las compresiones

Para medir las frecuencia de las compresiones de pecho en segundos se diseña la siguiente formula:

$$\text{Tiempo promedio compresiones de pecho (s)} = \frac{\text{Tiempo compresiones de pecho (s)}}{\text{Numero de compresiones acumuladas}} \quad (2.2)$$

En la anterior formula se divide el tiempo transcurrido desde el inicio de la práctica (Se establece como tiempo inicial de la práctica el momento en que el usuario realiza la primer compresión de pecho) entre el número de compresiones acumuladas para obtener un tiempo medio entre compresiones en segundos. El tiempo medio entre compresiones ideal con base en las recomendaciones de las guías de la AHA debe hallarse en el rango entre 500 a 600 milisegundos para garantizar que al cumplirse un minuto el valor del número de compresiones se encuentre en el rango de 100 a 120.

### Numero de compresiones

Para calcular el numero de compresiones se realiza el siguiente algoritmo en pseudocódigo:

```
1 Variable estado 1 = 0
2 Variable estado 2 = 0
3 Contador compresiones = 0
4
5 Función Número de compresiones de pecho {
6   Si la distancia actual del sensor < 1mm {
7     Variable estado 1 = 1
8   }
9   Si la distancia actual del sensor > 49mm {
10    Variable estado 2 = 1
11
12    Si Variable estado 1 es igual a 1 y variable estado 2 es ...
13      igual a 1 {
14        Contador compresiones = Contador compresiones + 1
15      }
16 }
```

Figura 2.13: Pseudocódigo para estimar las compresiones acumuladas con el sensor HY-SRF05

Una vez teniendo el número de compresiones en una variable acumulativa *Contador compresiones*, Se tiene que esta variable debe funcionar como divisor para la ecuación de la frecuencia de las compresiones de pecho reasignado la variable *Numero de compresiones acumulada = Contador compresiones*.

### Retroceso

Para calcular el retroceso se tiene que, en seguimiento a las recomendaciones de las guías de la AHA, esta variable se debe controlar como la expansión de la caja torácica después de cada compresión de pecho. Si bien la descripción de las guías de la AHA y de otras bibliografías consultadas y relacionadas en este trabajo en el capítulo de referencias bibliográficas se refieren al retroceso como una variable cualitativa y no le asignan un valor cuantitativo en específico, se puede asumir la variable retroceso como la aproximación a 0mm de la distancia de la compresión. En consecuencia a las ideas definidas se puede formular para la variable retroceso el siguiente algoritmo en pseudocódigo:

```
1 ...Se debe haber realizado al menos 1 compresión de pecho
2
3 Si la distancia actual del sensor < 1mm {
4   Retroceso completo
5 }
```

Figura 2.14: Pseudocódigo para estimar el retroceso del tórax con el sensor HY-SRF05

Para que el valor de esta variable sea estable desde el principio de las compresiones se propone parametrizar un valor de *offset* en el algoritmo para calcular la distancia. Este valor de *offset* se utiliza en el sistema debido a que cuando un usuario apoya las manos sobre el pecho del maniquí pero no realiza ninguna acción de compresión, el peso de las manos más los brazos y el tren superior generan un valor de profundidad definido para este trabajo como *error de profundidad natural*.

El *error de profundidad natural* puede ser suprimido parametrizando un valor de *offset* en el algoritmo para el cálculo de la distancia.

### 2.3.6.2. Diseño de los indicadores para la interfaz gráfica

Primero se realizó un bosquejo de la disposición de los indicadores y sus formas de manera manual. A continuación, un bosquejo realizado por los autores de este trabajo:

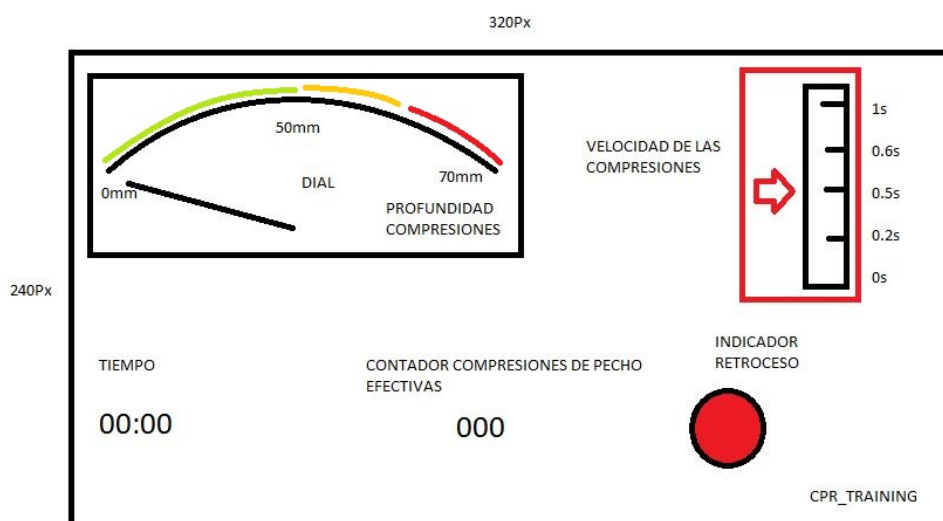


Figura 2.15: Indicadores Prototipo CPR-TRAINING #1. Fuente: Autores

Para el indicador de profundidad se decide hacer un dial en donde la aguja se moverá en un rango de 0 a 70mm. La idea es primero que en el rango entre 0 y 50mm exista un diferenciador de color verde que le informe al usuario que está en la zona de profundidad ideal, después en el rango entre 50 y 60mm informar por un color naranja al usuario que se encuentra en la zona de profundidad máxima y para los valores mayores a 60mm indicar por medio del color rojo que se encuentra en una zona que no debería estar.

Luego para el indicador de velocidad de las compresiones de pecho se diseña un indicador rectangular con una aguja que indica el tiempo en milisegundos entre compresiones.

Para la variable del retroceso se diseña un indicador tipo semáforo de un solo color en donde una circunferencia se rellena de color rojo si se realiza una compresión de pecho ideal (es decir que la profundidad llegó a 50mm como mínimo) y la misma circunferencia se debe de rellenar en negro si se logra un buen retroceso del pecho (es decir que la profundidad se aproxime a 0 mm).



Figura 2.16: Indicador retroceso de 2 estados 1.Rojo: Incompleto 2. Negro: Completo Prototipo CPR-TRAINING #1. Fuente: Autores

Para el contador del número de compresiones efectivas (Compresión efectiva: Realizar compresión como mínimo hasta 50mm y retroceso completo hasta 0mm durante cada ciclo en el ejercicio) se utiliza un indicador de numero entero de 1 byte para tener los números que van desde el 0 hasta 255 y en el indicador se representaría como un numero de 3 cifras (000).

También se decide implementar un temporizador que le indique al usuario la cantidad de tiempo que ha transcurrido desde el inicio de la práctica.

*-Para cumplir con el objetivo 2 se realizaron los siguientes procedimientos:*

### 2.3.7. Integración Hardware y Software

#### 2.3.7.1. Conexión de los dispositivos y fuente de alimentación

Una vez calibrado el sensor ultrasónico se procede a programar en el microcontrolador las rutinas diseñadas para calcular el retroceso, profundidad, velocidad y numero de compresiones de pecho pero para realizar este procedimiento primero se deben conectar los componentes y dar alimentación al circuito.

El microcontrolador ESP32 Versión Devkit V1 entre sus pines de voltaje de salida solo cuenta con 3.3 Voltios máximo y el sensor HY-SRF05 debe alimentarse con al menos 5 VDC. Por otro lado, el ESP32 tiene un pin VIN de 4.5 VDC para ser alimentado desde una fuente externa y no depender del cable de transferencia de datos USB para estar encendido. Debido a las variables para la alimentación de diferentes dispositivos se presenta la necesidad de tener circuitos reguladores de voltaje para alimentar el sensor y el microcontrolador.

En esta etapa se decide utilizar dos módulos reguladores LM2596 (convertidor DC-DC). Este dispositivo es un modulo de bajo costo y una alternativa rápida para integrar convertidores DC-DC con salida entre 1.5 y 35 VDC en un circuito.

El microchip LM2596 es un circuito integrado monolítico capaz de conducir corrientes de hasta de 3 amperios, ideal para alimentar microcontroladores y sensores [25].



Figura 2.17: Modulo regulador - convertidor DC-DC ideal para sensores y microcontroladores [47].

De esta manera ya se tiene la regulación de la alimentación del sensor y del módulo con 2 circuitos LM2596 respectivamente. A continuación se realiza el análisis para la selección de la fuente de alimentación apropiada.

Para realizar una aproximación teórica de la fuente de alimentación que debe acoplarse al sistema se debe tener en cuenta diseñar un modelo teórico del consumo de cada componente pasivo en el circuito.

De acuerdo al argumento del párrafo anterior se realiza la búsqueda del consumo de corriente máxima especificado en las hojas de datos de cada componente:

Componente	Cantidad	Consumo Max. (mA)	Total Consumo (mA)
HY-SR05 (Sensor ultrasónico)	1	40	40
Micro ESP32	1	68	68
Pantalla TFT ILI9341	1	0.1	0.1
Regulador Voltaje LM2596	2	0.08	0.16
Total consumo (mA)			108.26

Tabla 2.4: Consumo en mA teórico de componentes

Teniendo en cuenta que entre los componentes a alimentar se encuentra un microcontrolador y un sensor que requieren de 4.5 y 5 Voltios de voltaje de entrada respectivamente, se busca implementar una fuente de alimentación que garantice un voltaje superior pero regulado con los módulos LM2596.

Teniendo como características una fuente por encima de 110mA y un voltaje superior a 5VDC

se decide implementar un cargador tipo transformador reductor de voltaje de 120VAC a 12VDC @ 1000mA. Este tipo de cargador tiene la robustez para alimentar el prototipo. Además, los reguladores de voltaje que se utilizarán tienen la capacidad de recibir hasta 37 VDC lo que convierte la alimentación del circuito en un elemento práctico porque se pueden conectar a la red eléctrica doméstica, comercial o universitaria.

Una vez se tienen claros los elementos para el consumo y regulación energética se continúa con la conexión entre los dispositivos. Se decide integrar al sistema 2 botones para dar inicio y reinicio al programa.

Con todo esto se presenta a continuación una fotografía del montaje en protoboard del sistema para empezar a realizar las codificaciones de prueba del sistema.

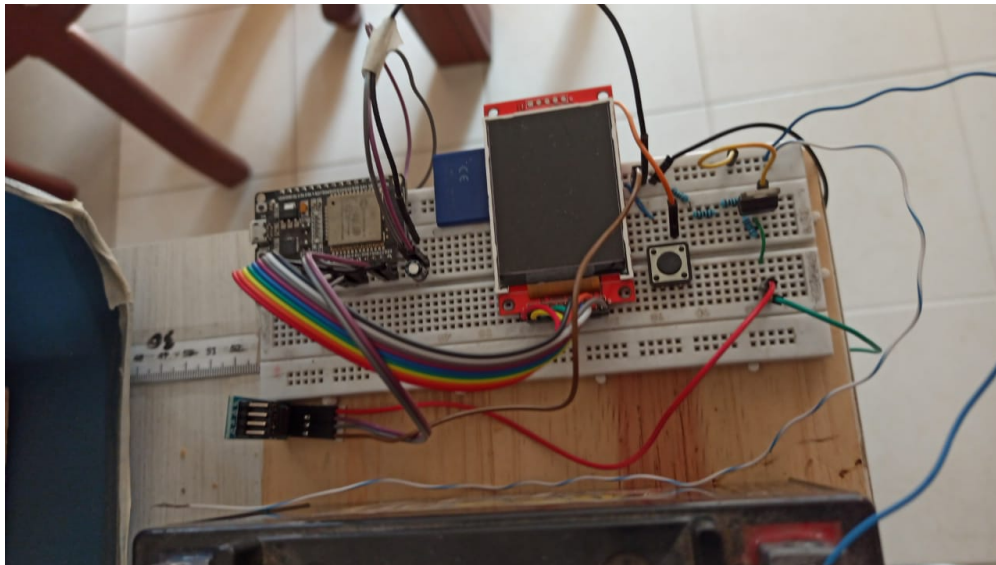


Figura 2.18: Prototipo CPR-TRAINING #1 en protoboard.

# Resultados del proyecto

---

## 3.1. Resultados de calibración sensor de distancia ultrasónico HY-SRF05

Para comenzar con la integración de los componentes de hardware y software del sistema se tuvo en cuenta la calibración del elemento que va medir las variables, el sensor ultrasónico. Para calibrar este sensor primero se consultó uno de los parámetros más importantes a tener en cuenta: La velocidad del sonido en el aire.

Teóricamente la velocidad del sonido en el aire a una temperatura de 20°C es 343,2 m/s aproximadamente [49]. Para ajustar la variable a las condiciones locales donde el sensor va a realizar trabajo se tuvo en cuenta el promedio de la temperatura local. Este valor de temperatura en promedio es de 25°C según el centro meteorológico nacional [11] de modo que ajustando esta variable en la ecuación para determinar la velocidad del sonido en el aire se obtiene el siguiente resultado:

Se tiene que la ecuación para determinar la velocidad del sonido en un gas ideal es:

$$V_{sonido}(m/s) = \sqrt{\frac{\gamma RT}{M}} \quad (3.1)$$

Donde

- Masa molecular aparente aire seco 28,96 g-mol  $\approx 29,0 \times 10^{-3}$  kg-mol
- Constante adiabática aire = 1.4

- Constante gases ideales = 8.31 J/mol.K
- Temperatura del aire: 25°C → En Kelvin ≈ 298,15K

$$345,84(m/s) = \sqrt{\frac{(1,4)(8,31J/mol.K)(298,15K)}{29,0 \times 10^{-3} kg-mol}} \quad (3.2)$$

Una vez se obtiene la velocidad del sonido en el aire para la ubicación donde va a desempeñar trabajo el sensor se determina que en la zona la variación de la temperatura no siempre se mantiene para el valor promedio tomado del centro meteorológico nacional, además el porcentaje de error de la temperatura tomada a 20°C con respecto a la de 25°C es igual a 0.76 % y esto por esto que se define tomar la velocidad del sonido como 343,2 m/s para la calibración del sensor, cálculos y algoritmos.

Con el fin de lograr una aproximación en la salida del sensor ultrasónico que se ajuste a las medidas de la distancia universal, se ha propuesto implementar un banco de calibración con una regla métrica impresa para el sensor.

Experimentalmente se ha extraído la medida interna del maniquí en función del lugar donde será ubicado el sensor para ejercer la medición de profundidad, frecuencia de las compresiones de pecho y retroceso. Esta medida se toma como aproximadamente 14 cm desde la espalda hasta el esternón del maniquí. Ver figura 2.5

Según la hoja de datos del sensor ultrasónico HY-SRF05[58] su rango de detección de objetos se encuentra entre 2 cm y 450 cm con una precisión de 2 mm.

Con base en los datos del párrafo inmediatamente anterior y las condiciones de medición a las que será sometido el sensor se propuso utilizar un rango de calibración de 3 a 14 cm. Se tiene en cuenta la altura del emisor y receptor del sensor donde aproximadamente miden 1.5 cm, Ver figura 3.1. Con estos datos se tiene que a partir de 4.5 cm obtenemos medida linealizada hasta la parte superior del maniquí a 14 cm donde el diferencial de medida es de 9.5 cm y la profundidad máxima del ejercicio de RCP es hasta 6.0 cm, por tanto, el método de calibración se ajusta a la necesidad del problema.

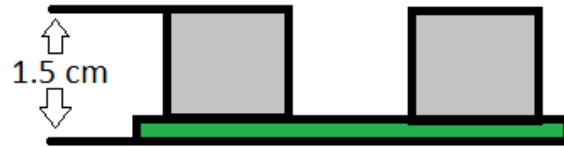


Figura 3.1: Vista superior sensor HY-SRF05 con longitud de su Tx y Rx

A continuación, se presentan unas imágenes que evidencian el montaje para la calibración del sensor.



Figura 3.2: Calibración sensor imagen 1

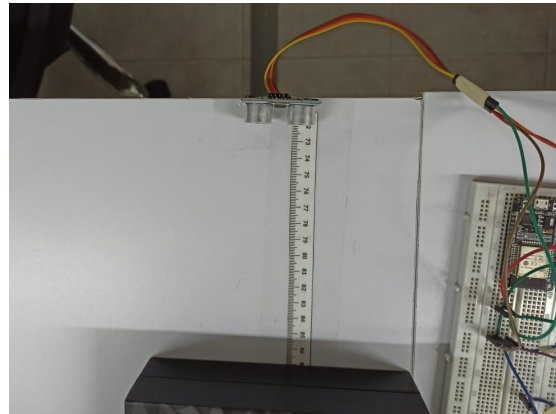


Figura 3.3: Calibración sensor imagen 2

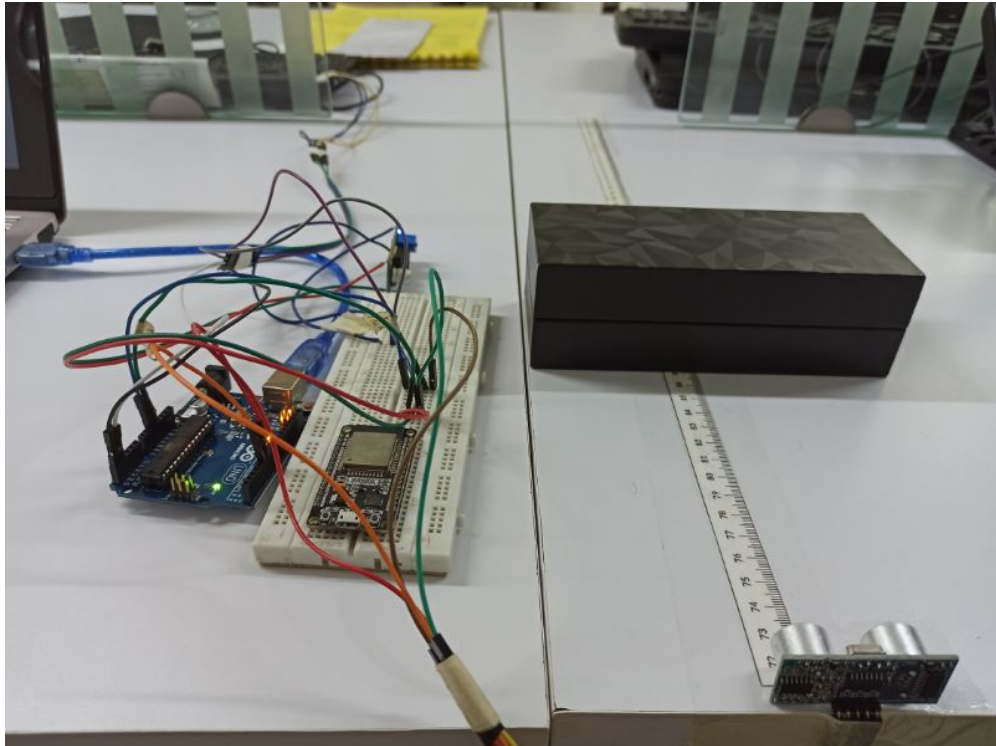


Figura 3.4: Calibración sensor imagen 3

### Procedimiento

Tenemos que la velocidad del sonido para este proceso es  $343.2 \text{ (m/s)}$  a  $20^\circ\text{C}$ .

Ahora bien, el sensor ultrasónico envía una señal desde el pin *trigger* y espera ser recibida en el pin *echo*. El tiempo que tarda en recibir la señal el sensor es el que se utiliza para calcular la distancia, pero debido a que el sensor emite la señal y espera de regreso el pulso se debe tomar la mitad del tiempo para el cálculo. Para este paso entonces se toma la media de la velocidad del sonido como  $171,6 \text{ (m/s)}$  a  $20^\circ\text{C}$ .

Teniendo en cuenta las variables necesarias para realizar el cálculo de la distancia se utiliza una rutina en Matlab para capturar 50 muestras cada 10 mm en el rango de calibración entre 30 mm y 140 mm. De las 50 muestras por cada centímetro medido se realizará un promedio simple para escoger los valores del vector de la variable independiente.

En la siguiente imagen se enseña una matriz en Matlab que contiene las muestras tomadas del sensor. Ver figura 3.5

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	25.7400	34.6632	44.9592	49.0776	63.6636	73.7880	82.1964	87.8592	102.7884	111.5400	120.8064	130.7592	
2	25.9116	35.6928	43.9296	50.1072	63.6636	72.7584	83.0544	88.8888	101.5872	111.5400	120.8064	131.7888	
3	25.9116	35.6928	44.4444	49.0776	62.4624	72.2436	83.2260	88.7172	101.2440	112.5696	121.8360	130.5876	
4	25.7400	35.6928	44.4444	49.0776	63.4920	73.9596	82.1964	87.6876	102.7884	112.5696	120.8064	130.7592	
5	24.7104	35.6928	44.9592	50.1072	63.6636	73.7880	82.0248	87.6876	101.2440	111.5400	120.8064	130.5876	
6	25.9116	35.6928	44.4444	50.1072	63.6636	72.7584	83.0544	88.8888	101.5872	111.3684	120.8064	131.6172	
7	25.9116	34.6632	44.9592	50.1072	62.4624	72.7584	83.0544	88.2024	102.7884	111.5400	120.8064	130.2444	
8	25.9116	35.6928	44.9592	49.0776	63.4920	73.4448	82.0248	87.8592	102.7884	112.5696	120.8064	130.7592	
9	25.9116	35.6928	43.9296	50.1072	63.6636	73.4448	83.2260	88.8888	101.5872	111.5400	120.8064	130.5876	
10	24.7104	35.6928	43.9296	50.1072	63.6636	72.7584	82.0248	88.8888	101.7588	111.5400	121.8360	131.6172	
11	25.9116	35.6928	44.9592	50.1072	62.4624	72.9300	82.1964	88.8888	101.2440	111.5400	120.8064	130.5876	
12	25.9116	35.6928	44.4444	49.0776	62.4624	73.2732	83.0544	87.6876	101.7588	112.5696	120.9780	130.7592	
13	25.7400	35.6928	44.9592	50.1072	63.4920	72.2436	83.2260	88.8888	102.7884	111.5400	121.8360	130.7592	
14	24.8820	35.6928	44.9592	50.1072	63.6636	72.2436	82.0248	88.7172	102.7884	111.5400	120.8064	130.5876	
15	25.9116	35.6928	44.9592	49.0776	62.4624	72.2436	82.1964	87.8592	101.7588	111.5400	120.8064	130.7592	
16	25.9116	35.6928	43.9296	50.1072	63.4920	73.2732	83.2260	88.8888	101.7588	112.5696	120.9780	130.7592	
17	25.9116	35.6928	44.9592	50.1072	63.6636	73.2732	83.0544	88.8888	102.7884	111.5400	122.0076	130.5876	
18	25.9116	34.4916	44.4444	50.1072	63.4920	72.2436	82.1964	87.6876	101.7588	111.5400	120.8064	130.5876	
19	25.9116	35.6928	44.9592	49.0776	62.4624	73.4448	82.0248	87.6876	101.7588	112.5696	120.8064	130.7592	
20	24.7104	35.6928	44.9592	50.1072	63.4920	73.9596	83.0544	88.5456	101.7588	112.5696	120.8064	131.6172	

Figura 3.5: Matriz de muestras del sensor ultrasónico

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	25.7228	35.5384	44.5130	49.8258	63.2140	73.0330	82.5808	88.3225	102.0677	111.9072	121.0913	130.8828	
2													

Figura 3.6: Vector fila con el promedio de cada columna

La variable independiente en este caso corresponde a la salida del sensor y se utilizarán los valores del vector fila de promedio para el procedimiento de linealización.

Se crea un vector con la distancia real en el rango de 30 mm a 140 mm con pasos de 10 mm. El vector distancia real será la variable dependiente en la función.

Después utilizando la herramienta *Curve Fitting Toolbox* de Matlab se realiza el proceso de linealización para aproximar la salida del sensor con las medidas de distancia universales.

Los parámetros en el software se ajustaron para un polinomio de grado 1 con la forma  $f(x) = ax + b$ .

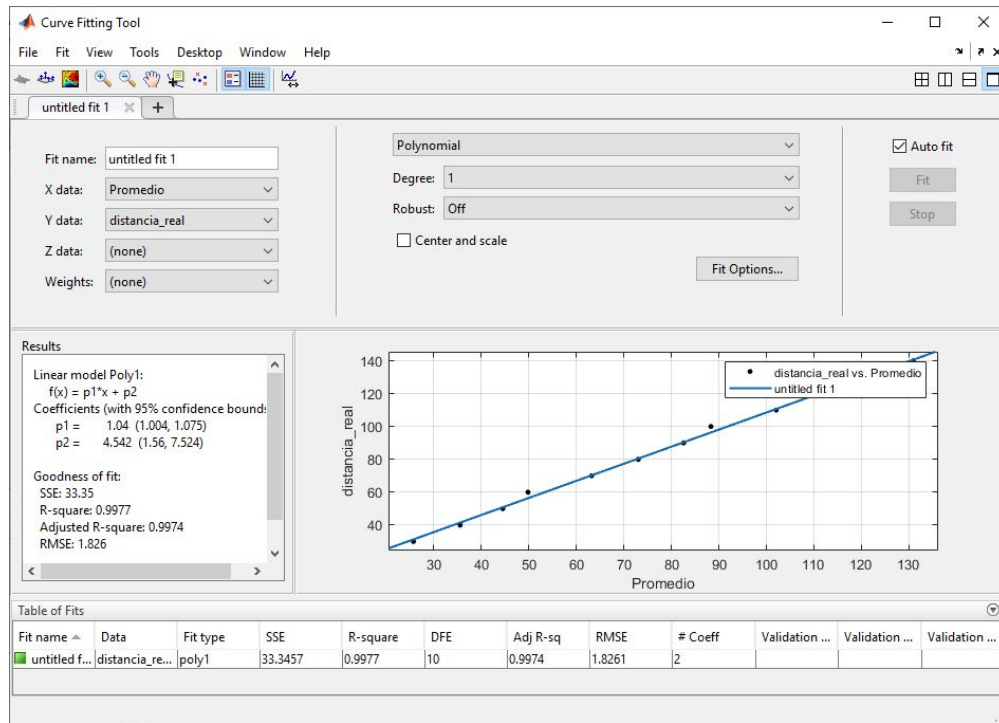


Figura 3.7: Curve Fitting Toolbox

Los valores arrojados por el software son:

- $a = 1,04$
- $b = 4,542$

A continuación se enseñan los valores arrojados para la desviación estándar y la varianza en las figuras 3.8 y 3.9 respectivamente.

The screenshot shows a variable editor window titled 'Variables - Desv\_std'. It displays a 1x12 double array of values representing standard deviation for 13 data points.

1	2	3	4	5	6	7	8	9	10	11	12	13
0.3987	0.3880	0.4964	0.4725	0.5256	0.6094	0.5321	0.5407	0.5728	0.5107	0.4723	0.4600	

Figura 3.8: Desviación estándar datos del sensor ultrasónico

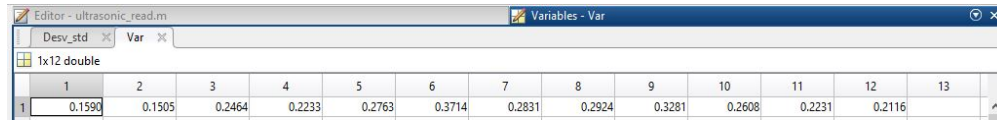


Figura 3.9: Varianza datos del sensor ultrasónico

Con base en los resultados obtenidos se realiza una prueba nuevamente para analizar los resultados de la recta de linealización.

Nuevamente se toman muestras cada 10 mm de la variable de salida del sensor con el ajuste lineal y el resultado es el siguiente:

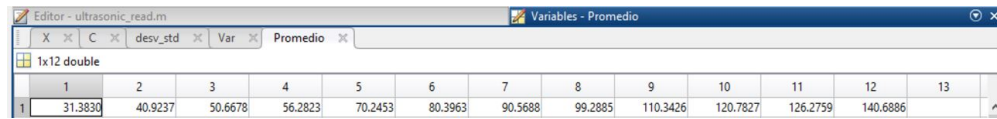


Figura 3.10: Vector fila con el promedio de cada columna despues de la recta de ajuste

Los valores en promedio para cada punto entre 30 mm y 140 mm muestran una aproximación con respecto al vector de la distancia real, sin embargo existen datos que se encuentran significativamente alejados como el de las columnas 4 y 11.

### Porcentajes de error

A continuación se presentan 2 tablas y 2 figuras con los porcentajes de error y grafica de las curvas de los experimentos realizados con el sensor ultrasónico antes y después de aplicar la ecuación para linealización.

Antes de realizar la linealización:

<b>Distancia promedio (mm)</b>	25,722	35,538	44,513	49,825	63,214	73,033	82,580	88,322	102,067	111,907	121,091	130,882
<b>Distancia teórica (mm)</b>	30,00	40,00	50,00	60,00	70,00	80,00	90,00	100,00	110,00	120,00	130,00	140,00
<b>% error</b>	14,257	11,154	10,974	16,957	9,694	8,709	8,244	11,678	7,211	6,744	6,853	6,512

Tabla 3.1: Porcentajes de error antes de realizar linealización

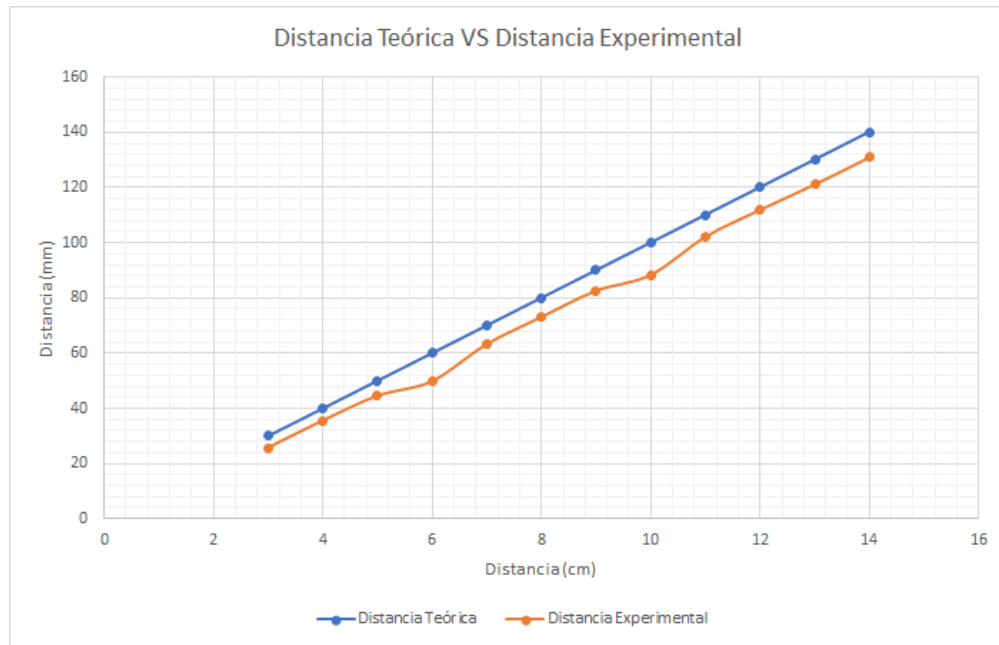


Figura 3.11: Curvas de distancia teórica vs experimental antes de aplicar linealización

Después de realizar la linealización

<b>Distancia promedio (mm)</b>	31,382	40,923	50,667	56,282	70,245	80,396	90,568	99,288	110,342	120,782	126,275	140,688
<b>Distancia teórica (mm)</b>	30,00	40,00	50,00	60,00	70,00	80,00	90,00	100,00	110,00	120,00	130,00	140,00
<b>% error</b>	4,610	2,309	1,336	6,196	0,350	0,495	0,632	0,712	0,311	0,652	2,865	0,492

Tabla 3.2: Porcentajes de error después de realizar linealización

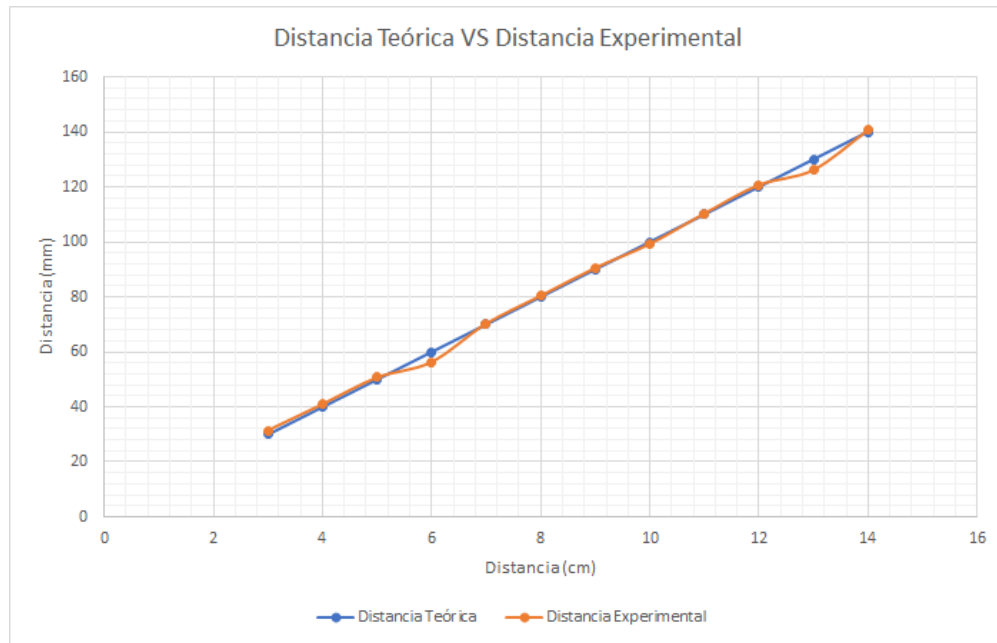


Figura 3.12: Curvas de distancia teórica vs experimental después de aplicar linealización

En función a los resultados obtenidos tenemos que el porcentaje de error en las medidas disminuye significativamente con los coeficientes de ajuste de la recta de linealización. Para los valores que se encuentran en el rango entre 80,0 y 140,0 mm el porcentaje de error es aproximadamente 0% para todos los valores exceptuando un valor atípico: 2,86%. En el rango entre 80,0 y 140,0 mm es donde el sensor funciona arrojando los valores de la profundidad de las compresiones de pecho.

### 3.2. Resultados implementación de rutinas e indicadores

Continuando con la implementación del software en el sistema, se integra el sensor ultrasónico con el ESP32 y se empieza a programar las rutinas diseñadas en pseudocódigo. Para efectos de pruebas se utiliza el monitor serial del IDE (Entorno de desarrollo integrado) de Arduino como *debug*.

Una vez teniendo las variables de salida para la frecuencia de las compresiones de pecho en segundos, el número de compresiones, la profundidad en mm y el retroceso se integra al ESP32 la pantalla LCD ILI9341 para empezar a programar lo que serán los gráficos con los cuales se van a visualizar las variables de la práctica.

Para este paso se utilizaron herramientas de internet como tutoriales audiovisuales y documentación técnica del dispositivo en la página oficial de Arduino. Para realizar el primer diseño de los indicadores en la pantalla se tuvieron en cuenta las librerías de código abierto disponibles en GitHub

[4] para construir figuras en la pantalla LCD y algunos ejemplos de programas de código abierto en GitHub listos para cargar a la pantalla e iniciar los primeros bosquejos.

### 3.3. Prototipo #1 CPR-TRAINING

En esta sección se comparte el resultado de la ejecución del código en el microprocesador y la forma en la cual quedaron dispuestos cada uno de los indicadores en la pantalla LCD.



Figura 3.13: Indicadores implementados en el microcontrolador Prototipo CPR-TRAINING #1.

Los indicadores como el dial, marcador en barra vertical y el reloj se construyeron mediante ejemplos de código abierto de GitHub [4] y Arduino incluido en las librerías para la pantalla TFT. Para este proceso se realiza un análisis a la estructura de los códigos de cada uno de estos indicadores y se realizan cambios para ajustarlo a la necesidad de este proyecto.

Una vez teniendo las variables del sensor representadas en la pantalla TFT se puede continuar con la etapa de pruebas del sensor montado en un maniquí de RCP.

### 3.3.1. Implementación modelo experimental casero de torso RCP para pruebas

Debido al corto tiempo que se tiene con los maniquís de RCP en el edificio guayacanes para realizar pruebas se propone implementar un maniquí de RCP casero para realizar dichas pruebas y efectuar ajustes rápidos en el dispositivo.

Para la implementación de un modelo de maniquí RCP casero se buscaron recursos en internet y en la IEEE relacionados con el tema objeto de estudio y grupos de investigación acerca de electrónica y biomedicina en reanimación cardiopulmonar.

Durante la búsqueda se encuentra la siguiente información en una pagina web con la siguiente URL: <http://reanyma.blogspot.com/2012/01/maniqui-economico-para-rcp.html>: *Maniquí económico para RCP por el Dr. Saúl Drajer Médico Subjefe del comité de atención cardiovascular de emergencia de la fundación interamericana del corazón y delegado ILCOR. Director de la clínica de la esperanza en Buenos aires - Argentina [14].*

La persona creadora de la página Web y el artículo científico [14] comparte la manera de construir un torso de maniquí de RCP con materiales como un tarro plástico de 5 galones para líquidos, un resorte con constante de proporcionalidad aproximada a los que tienen los maniquís de RCP de laboratorios como LAERDAL y elementos varios como herramientas para cortar, atornillar y pulir.

Antes de iniciar con la implementación del maniquí casero de RCP primero se comparte la información vía correo electrónico con la ingeniería biomédica del hospital simulado Valentina Corchuelo y con el coordinador científico del hospital simulado Oscar Echeverry, esto con el objetivo de obtener una realimentación profesional respecto a la idea de implementar un torso de maniquí de RCP Casero.

Luego de compartir la información con los profesionales su respuesta fue positiva y dieron el aval para construir el equipo casero argumentando que era una buena idea para hacer pruebas en el diseño del dispositivo. Con base en esto se les solicita amablemente a los profesionales del hospital simulado las medidas de un resorte de maniquí RCP adulto con el objetivo de enviar a fabricar uno igual para el modelo casero.

La información compartida acerca de un resorte de maniquí RCP adulto es la siguiente:

- Longitud resorte: 16.3 cm
- Diámetro resorte: 5.3 cm
- Espesor alambre: 0.5 cm

(Medidas tomadas con un vernier - pie de rey)

A continuación, imágenes del resorte compartidas por los profesionales:



Figura 3.14: Resorte maniquí RCP Laerdal vista frontal



Figura 3.15: Resorte maniquí RCP Laerdal vista superior

Una vez teniendo la información que compartieron desde el hospital simulado acerca de las dimensiones del resorte se procede a enviar a fabricar uno igual para el modelo casero. Continuando con los otros materiales se logra conseguir un envase plástico de 5 galones con dimensiones aproximadas de volumen similares a las de un torso de maniquí de RCP.

Finalmente teniendo los materiales para el modelo casero el resultado del ensamblaje de las piezas es el siguiente:

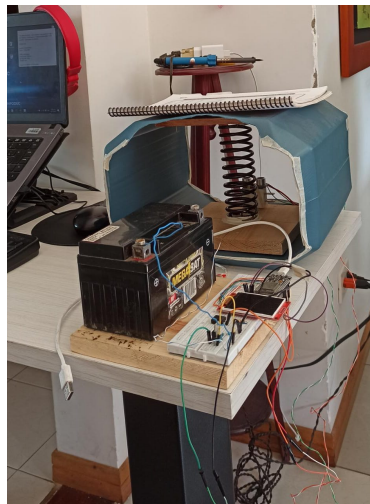


Figura 3.16: Prototipo CPR-TRAINING #1 vista completa



Figura 3.17: Prototipo CPR-TRAINING #1 vista interior

Una vez finalizado el prototipo del sistema #1 se decide programar una reunión de presentación del prototipo al público objetivo de este proyecto que corresponde al equipo profesional y científico del hospital simulado del edificio guayacanes.

### 3.3.2. Socialización virtual prototipo #1 CPR-TRAINING con equipo profesional en Salud Universidad Javeriana Cali

La reunión se programa para la fecha del 21-01-2021 con los asistentes a continuación: Oscar Echeverry - Coordinador científico del hospital simulado, Valentina Corchuelo - Ingeniera Biomédica del hospital simulado, Juan Pablo Navarrete y Juan Felipe Cifuentes estudiantes de Ingeniería electrónica.

Las notas de la reunión se encuentran en el anexo en el documento nombrado: "Pontificia Universidad Javeriana Cali Acta 001" y en el documento se detallan todos los puntos presentados hacia los asistentes del hospital simulado durante la reunión. En el documento también se encuentran cada una de las observaciones brindadas por los profesionales del hospital simulado. Cada una de estas observaciones se tienen en cuenta para realizar ajustes en el sistema y continuar con lo que será el prototipo #2.

## 3.4. Prototipo #2 Sistema de retroalimentación visual CPR-TRAINING

A continuación, se realiza una lista de las observaciones y recomendaciones del documento anexo "Pontificia Universidad Javeriana Cali Acta 001" para mejorar el prototipo del sistema:

1. **Indicador profundidad:** Modificación convención de color para el indicador tipo dial:
  - Rango ideal: Color verde [5cm - 6cm].
  - Rango no ideal: Color rojo [0cm - 5cm) y (6cm - 7 cm].
2. **Indicador profundidad:** Cambiar submúltiplo de la unidad de medida mm ->cm.
3. **Indicador velocidad de compresiones:** Se debe manejar un indicador por colores que resulte mas fácil de entender para el usuario cuando va lento y debe incrementar la velocidad; cuando va bien en velocidad y en teoría la frecuencia de sus pulsaciones se encuentra entre 100 y 120 compresiones por minuto y finalmente cuando va muy rápido y debe bajar el ritmo de velocidad.

La recomendación para este indicador es la siguiente:

- <100 cpm (Color amarillo, lento)

- $>120$  cpm (Color rojo, rápido)
  - $\geq 100$  cpm y  $\leq 120$  cpm (Color verde, ideal)
4. **Indicador número de compresiones efectivas:** El indicador de numero de compresiones efectivas se encuentra bien (Las compresiones efectivas son las que cumplen con profundidad  $\geq 5$ cm y retroceso completo = 0cm durante cada ciclo de compresión de pecho) pero es necesario un indicador que acumule todas las compresiones, efectivas y no efectivas.
  5. **Indicador retroceso:** Este tipo de indicador se puede continuar utilizando como se encuentra hasta el momento puesto que corresponde a una alerta visual cualitativa que informa al usuario cuando esta cumpliendo con la liberación/expansión del pecho. Se recomienda como ajuste cambiar el color a verde (Actualmente rojo).

Una vez teniendo en cuenta las recomendaciones sugeridas por los profesionales del hospital simulado se continua con el ajuste al diseño a nivel de código en el sistema.

Para realizar estos ajustes primero se rediseñan en un dibujo tipo bosquejo los indicadores de la figura 3.13. A continuación se presenta el bosquejo:

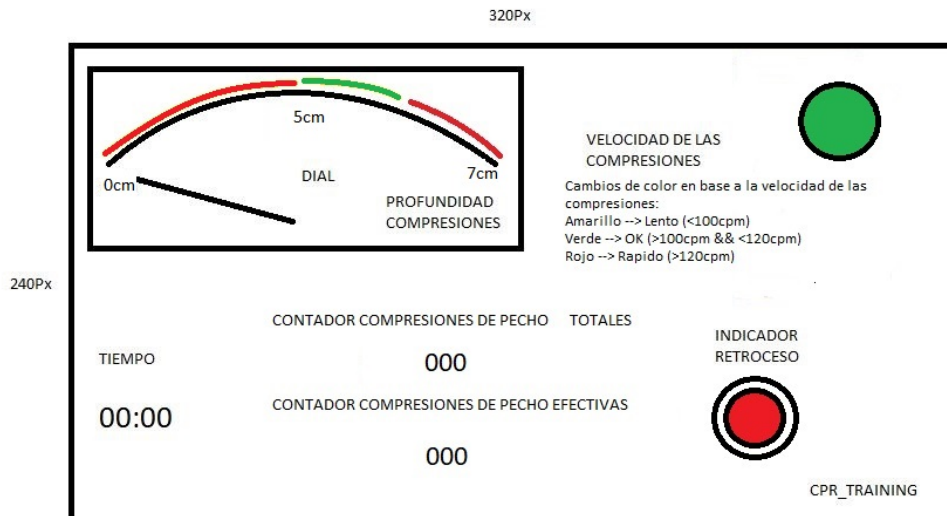


Figura 3.18: Indicadores Prototipo CPR-TRAINING #2.

Finalizando los ajustes en el código para el prototipo #2 del CPR-TRAINING se realizan pruebas en el montaje en protoboard nuevamente y se toma una fotografía de los objetos ajustados en la interfaz gráfica.



Figura 3.19: Indicadores implementados en el microcontrolador Prototipo #2 CPR-TRAINING.

A partir de este punto se decide diseñar e implementar la tarjeta de circuito impreso con ayuda de la herramienta de software Eagle de Autodesk.

Circuito esquemático:

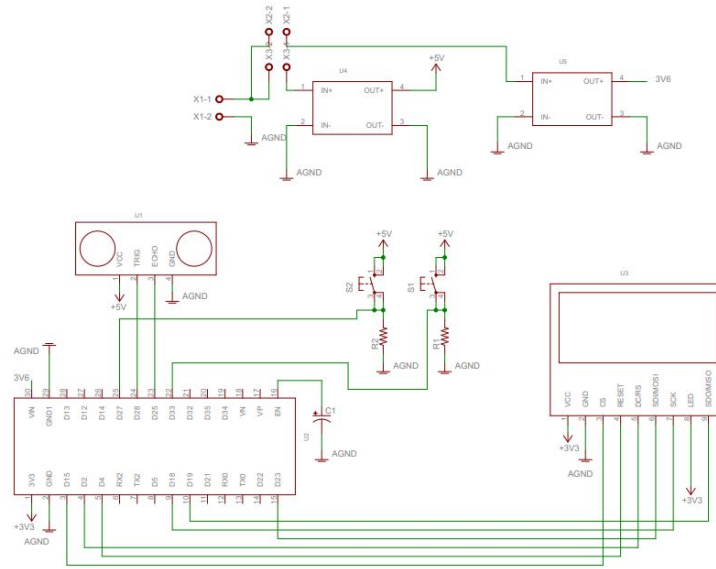


Figura 3.20: Circuito esquemático Prototipo #1 CPR-TRAINING.

Circuito PCB:

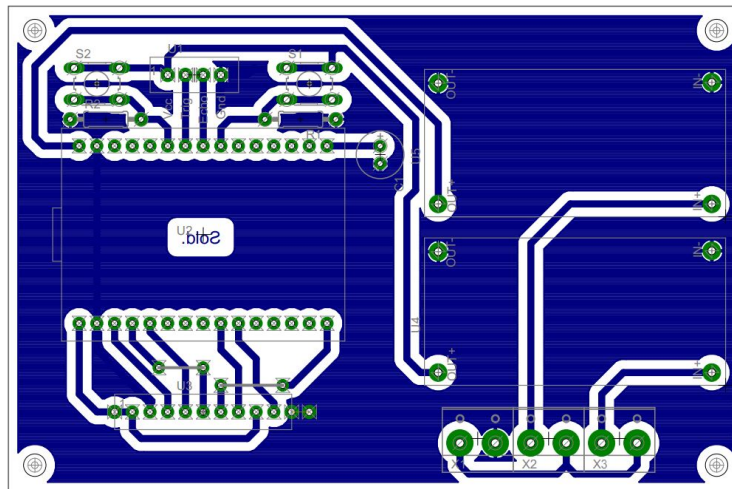


Figura 3.21: Circuito PCB Prototipo #1 CPR-TRAINING.

Construir el circuito impreso y conseguir un objeto en el cual se puede guardar y proteger el circuito electrónico garantiza que se pueda transportar fácilmente hacia el lugar donde se requiere realizar las pruebas y evaluación del sistema que para este caso es el hospital simulado de la uni-

versidad Javeriana Cali.

A continuación, se presenta de manera breve el procedimiento de ensamble de los componentes y el prototipo terminado.

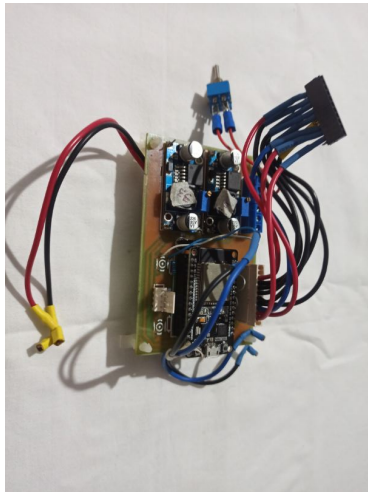


Figura 3.22: Montaje componentes en PCB vista 1

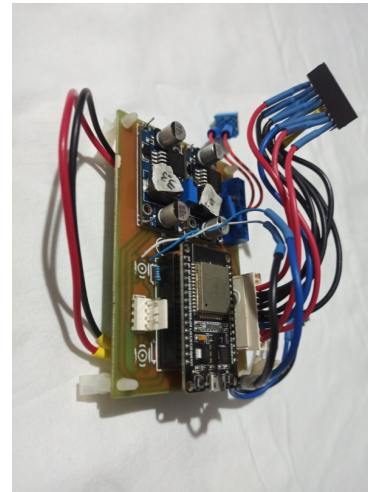


Figura 3.23: Montaje componentes en PCB vista 2

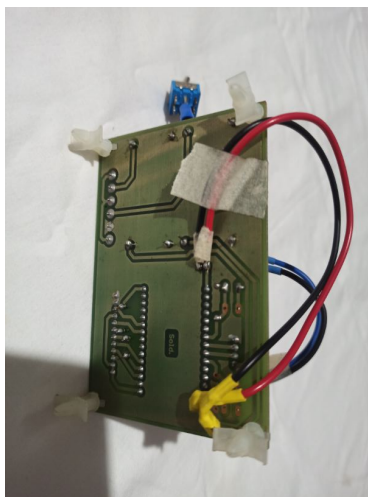


Figura 3.24: Montaje componentes en PCB vista 3



Figura 3.25: Cubierta superior caja del proyecto



Figura 3.26: Alojamiento de componentes

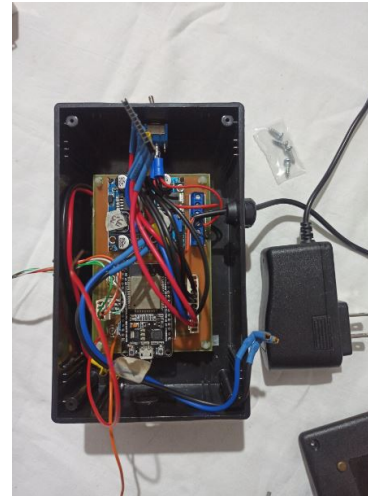


Figura 3.27: Circuito insertado en la caja del proyecto



Figura 3.28: Interruptor ON/OFF parte superior caja del proyecto



Figura 3.29: CPR-TRAINING totalmente armado

Antes de continuar hacia la sección de evaluación del prototipo se presentará la implementación de la plataforma IoT en el servicio Node-Red.

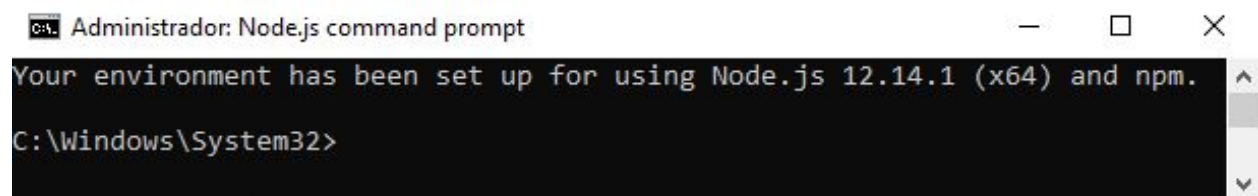
### 3.5. Implementación de indicadores en la plataforma IoT Node-Red

Para hacer uso de los servicios de Node-Red en un sistema operativo Windows se recomienda seguir el tutorial de la página oficial de Node-Red en la siguiente URL: <https://nodered.org/docs/getting-started/windows>.

Si el sistema operativo es diferente de Windows desde la misma URL se puede navegar en el sitio de Node-Red y buscar el sistema operativo que se ajuste a la necesidad.

Siguiendo el paso a paso que recomienda la página oficial de Node-Red se debe instalar en Windows el lenguaje de programación node-js para diseñar e implementar servicios y programas para computadoras. Una vez instalado node-js en el equipo se deben ejecutar algunas sentencias para instalar los paquetes de Node-Red como servidor local en la máquina y de esta manera se trabaja con una aplicación de manera local simulando acceder a la misma a través de internet.

Una vez instalado node-js y los paquetes de Node-Red en la máquina se procede a activar el servidor local de Node-Red a través de la ventana de comandos de node-js como se enseña en las siguientes figuras:



```
Administrador: Node.js command prompt
Your environment has been set up for using Node.js 12.14.1 (x64) and npm.
C:\Windows\System32>
```

Figura 3.30: Ventana de comandos node-js.

```
ca. node-red
Your environment has been set up for using Node.js 12.14.1 (x64) and npm.

C:\Windows\System32>node-red
18 May 01:25:24 - [info]

Welcome to Node-RED
=====

18 May 01:25:24 - [info] Node-RED version: v1.0.3
18 May 01:25:24 - [info] Node.js version: v12.14.1
18 May 01:25:24 - [info] Windows_NT 10.0.19042 x64 LE
18 May 01:25:26 - [info] Loading palette nodes
18 May 01:25:54 - [info] Dashboard version 2.29.0 started at /ui
18 May 01:25:57 - [info] Settings file : C:\Users\Usuario.DESKTOP-581B0EE\.node-red\settings.js
18 May 01:25:57 - [info] Context store : 'default' [module=memory]
18 May 01:25:57 - [info] User directory : C:\Users\Usuario.DESKTOP-581B0EE\.node-red
18 May 01:25:57 - [warn] Projects disabled : editorTheme.projects.enabled=false
18 May 01:25:57 - [info] Flows file : C:\Users\Usuario.DESKTOP-581B0EE\.node-red\flows_CONSULTOR06.json
18 May 01:25:57 - [info] Server now running at http://127.0.0.1:1880/
18 May 01:25:57 - [warn]
```

Figura 3.31: Ejecución del servidor local Node-Red desde la ventana de comandos node-js

Una vez iniciado el servicio local, la maquina entrega una dirección IP 127.0.0.1 y un numero de puerto 1880 es decir, la comunicación se establece con el localhost (Servidor instalado) en el mismo equipo [50]. La ventaja de utilizar este tipo de servicios es trabajar en el proyecto durante la etapa de desarrollo, una vez se finalice una etapa de desarrollo de software y se realicen pruebas se podría decir que el sistema esta listo para salir a internet a etapa productiva, pero en términos generales el objetivo de este proyecto no requiere de salir a internet y exponer el servicio mediante IP publica sin embargo, el microcontrolador para enviar datos al servicio de Node-Red requiere de conexión a internet via WiFi y el servicio para recibir los datos requiere también conectarse a la misma red en donde el microcontrolador se ha conectado previamente. El servidor local toma como punto de acceso a internet la red WiFi a la cual la computadora se conecta.

Después de iniciar el servicio se utiliza la dirección del servidor en el navegador predeterminado del equipo y se accede a la aplicación Node-Red.

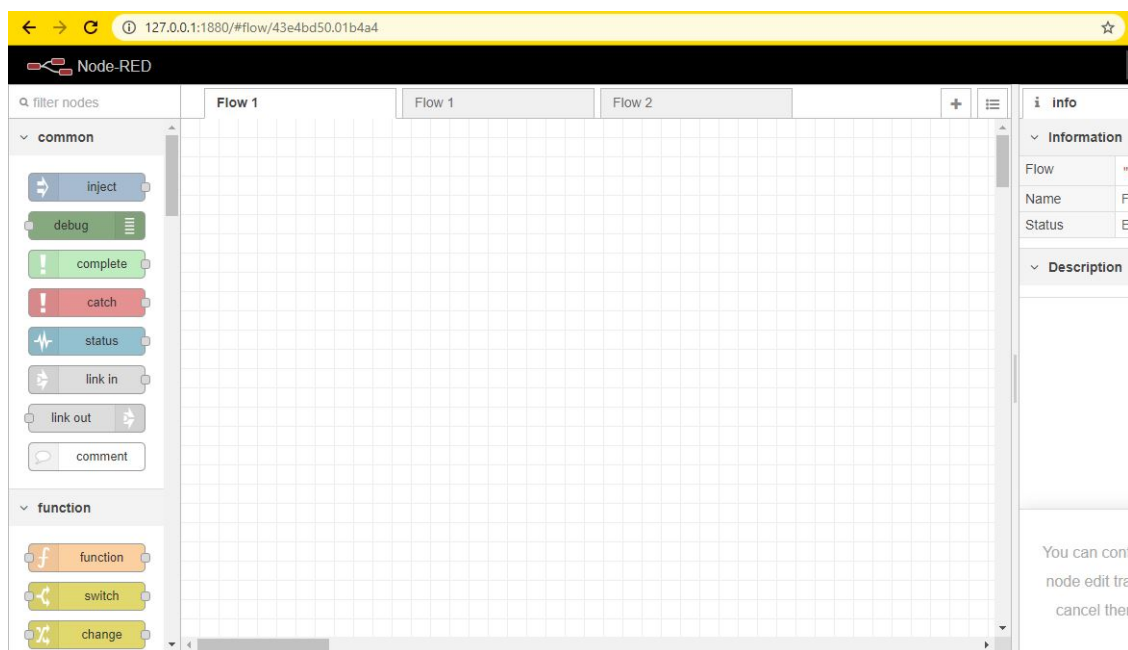


Figura 3.32: Entorno de desarrollo integrado de Node-Red

Para implementar código en Node-Red se debe utilizar el modelo de programación por nodos en el cual el servicio ofrece diferentes objetos llamados nodos con funcionalidades específicas, ejemplo: Nodo de mensaje por correo electrónico, Nodo de notificación vía SMS o Nodo para crear una función con sintaxis JavaScript.

Para este proyecto se definieron en la etapa de diseño los datos que serían entregados por el microcontrolador a la nube del servidor IoT. Estos datos son generados por el sensor, transformados y recolectados en variables durante un tiempo de 1 minuto. También se argumenta que la retroalimentación recibida por parte del equipo del hospital simulado de la universidad en la subsección 3.3.2 se tiene en cuenta y se agrega a la etapa de diseño a partir de la subsección 2.3.6. A continuación, una lista de los datos enviados por el microcontrolador al servicio de Node-Red a través de la conexión a internet del dispositivo.

#### Lista de datos:

1. - Número total de compresiones
2. - Número de compresiones efectivas
3. - Porcentaje del tiempo (1 minuto) en velocidad compresiones baja (<100 cpm)
4. - Porcentaje del tiempo (1 minuto) en velocidad compresiones OK (>100 cpm y <120 cpm)

5. - Porcentaje del tiempo (1 minuto) en velocidad compresiones rapida (>120 cpm)
6. - Tiempo promedio entre compresiones durante toda la actividad en 1 minuto

Estos datos se envían cada minuto más un segundo (61 segundos) concatenados en una sola variable tipo cadena. El servicio recibe el valor de la variable y por medio de una función en JavaScript se recorre la cadena de caracteres y se asignan el respectivo valor de cada variable a cada columna en una tabla de una base de datos MySQL creada para este desarrollo. A continuación, se presenta en detalle la etapa de creación de la base de datos con el propósito de guardar información histórica.

### **3.5.1. Base de datos MySQL para almacenar los datos del prototipo**

Se elige trabajar con una base de datos MySQL debido a que Node-Red tiene un nodo con funciones específicas para esta base de datos y resulta muy práctica de implementar para este trabajo.

Para instalar la base de datos primero se busco información acerca de instalación de manera local con el objetivo de administrar el recurso desde la misma maquina todo con fines de realizar simulaciones, pruebas y toda aquella actividad relacionada con desarrollo de software.

Para este propósito se encontró información acerca de un sistema para la gestión de base de datos MySQL llamado XAMPP[54]. Este gestor de base de datos es distribuido por Apache[18] (Servidor HTTP gratuito y de desarrollo libre a nivel mundial) como software libre y con una gran comunidad alrededor del mundo. Con ayuda de este servidor local y una aplicación para ejecutar consultas SQL se logro construir una base de datos y una tabla con 8 columnas para almacenar los datos enviados por el microcontrolador a través de internet.

A continuación, se adjuntan algunas imágenes del servidor local XAMPP y de la estructura construida en el gestor de base de datos:

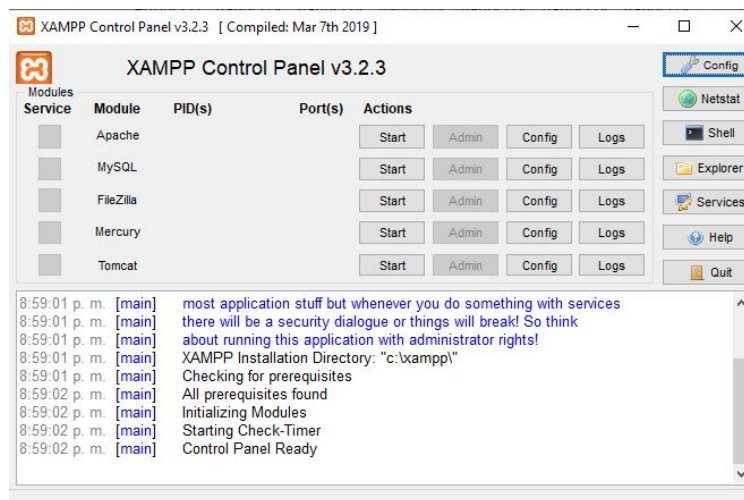
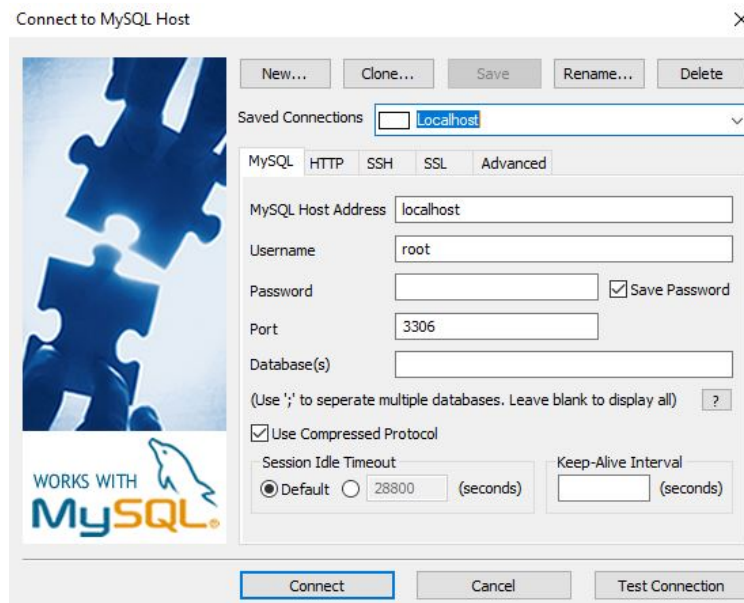


Figura 3.33: Panel de control XAMPP

La interfaz gráfica de usuario (GUI) elegida para el sistema de gestión de la base de datos es SQLyog. En esta interfaz se pueden ejecutar consultas SQL para crear, modificar o eliminar bases de datos, tablas, registros. En la figura 3.34 se presenta la ventana de SQLyog para la conexión al servidor local *localhost*

Figura 3.34: Conexión SQLyog con el *localhost*

Una vez iniciada la conexión se crea una base de datos con nombre *cptraining* y por medio de

la siguiente sentencia SQL se crea una tabla con 8 columnas llamada *entrenamiento\_2*:

Listing 3.1: Sentencia SQL para crear la tabla *entrenamiento\_2*

```
1 CREATE TABLE entrenamiento_2 (  
2     ID INT UNSIGNED NOT NULL AUTO_INCREMENT,  
3     comp_tot INT UNSIGNED NOT NULL,  
4     comp_tot INT UNSIGNED NOT NULL,  
5     porc_t_len FLOAT(4,2),  
6     porc_t_ok FLOAT(4,2),  
7     porc_t_rap FLOAT(4,2),  
8     tiempo_prom FLOAT(4,2),  
9     fecha_reg TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
10    PRIMARY KEY (ID)  
11 );
```

Después de ejecutar la consulta se confirma que la tabla se encuentre en la base de datos. Ver figura 3.35:

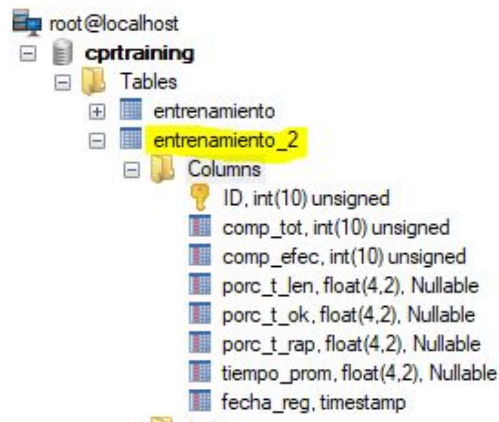


Figura 3.35: Tabla *entrenamiento\_2* creada

Una vez se confirma la existencia de la tabla *entrenamiento\_2* se procede a realizar un resumen de las columnas creadas en la tabla. La primera columna corresponde a la llave primaria que toda tabla de base de datos en un modelo relacional debe tener como identificador de cada registro insertado en la tabla. Esta llave primaria se define como auto incremental y se auto inserta cada vez que se ejecuta una sentencia *INSERT*. El auto incremental se puede definir de 1 en 1, 2 en 2 o en el patrón que el desarrollador desee dar a su modelo.

Luego se definen las columnas 2 hasta la 7 en donde se almacenarán los datos enviados por el microcontrolador. La columna 2 almacena el dato del número de compresiones totales, la columna 3 almacena el dato del numero de compresiones efectivas, la columna 4 almacena el dato del porcentaje de tiempo en velocidad lenta, la columna 5 almacena el dato del porcentaje de tiempo en velocidad ok, la columna 6 almacena el dato del porcentaje de tiempo en velocidad rápida, la columna 7 almacena el dato del tiempo promedio entre compresiones durante toda la actividad (1 minuto) y por ultimo la columna 8 almacena el dato de fecha y hora en el cual el registro fue insertado.

Con base en el diseño planteado se implementaron nodos de entrada del servidor MQTT para recibir los datos del microcontrolador, nodos de funciones JavaScript para las rutinas que se encargan de tomar el dato en cadena de caracteres y desagregar su información para pasar al nodo de base de datos y realizar la inserción del registro en la tabla. También hay otros nodos de funciones JavaScript que se encargan de realizar consulta a los datos de la tabla y nodos para graficar y visualizar en un dashboard mediante indicadores el comportamiento de las variables.

A continuación, se presenta una imagen que evidencia el trabajo realizado en el entorno de desarrollo integrado de Node-Red.

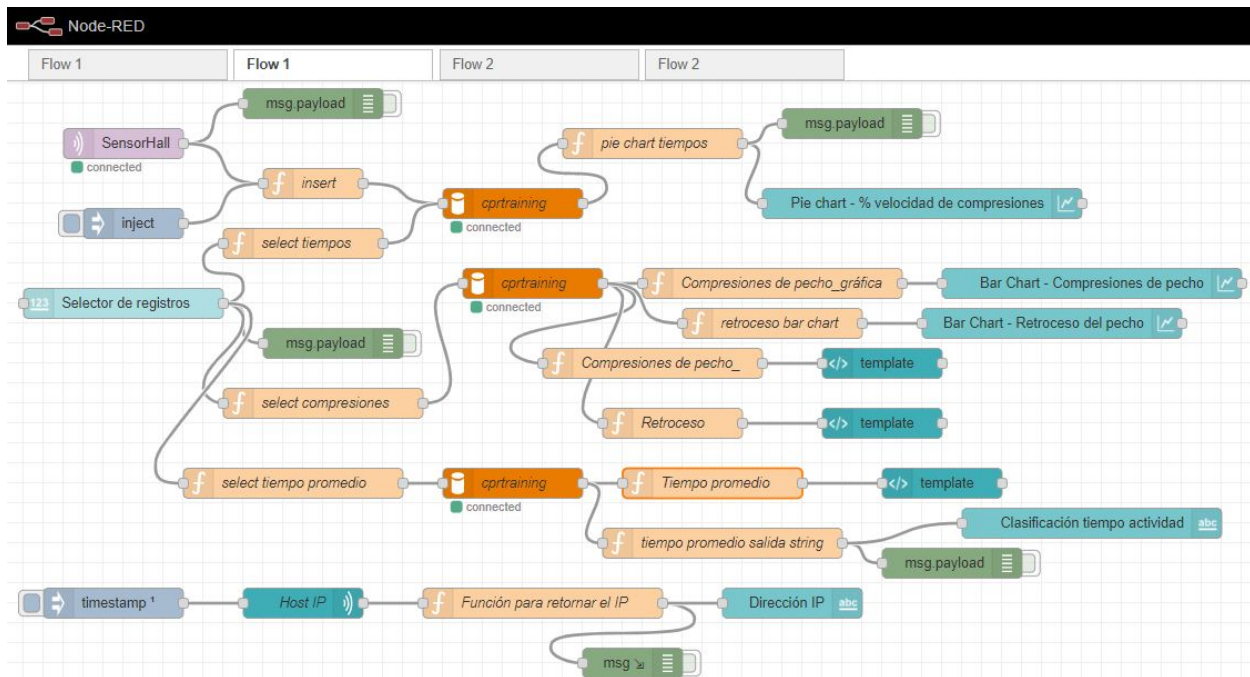
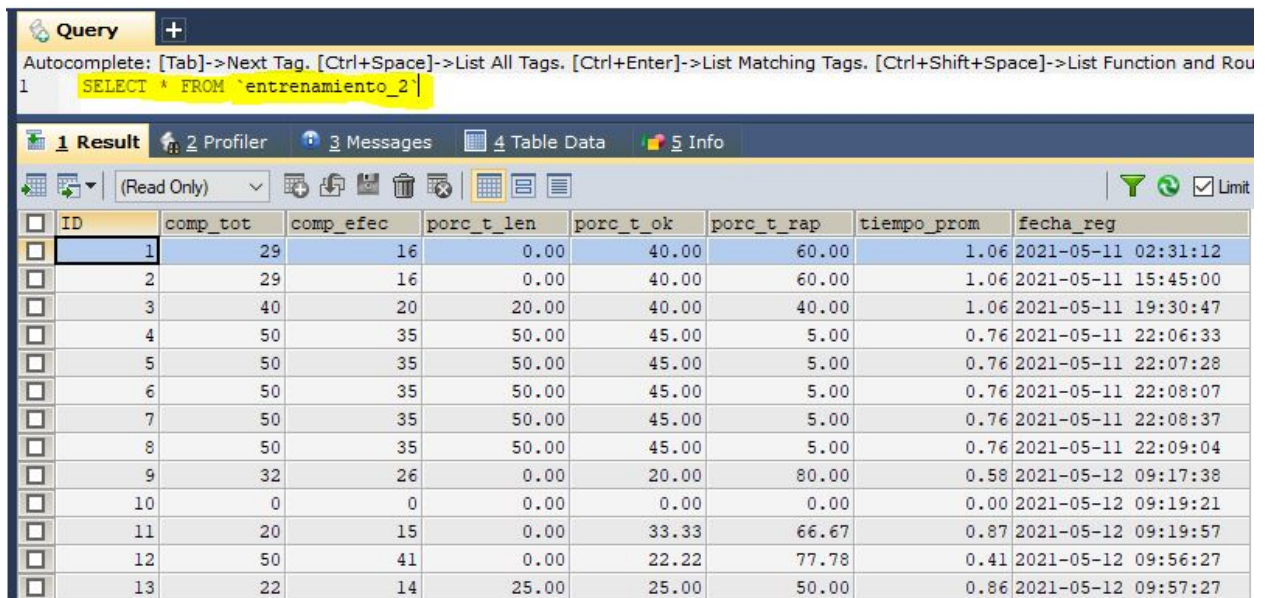


Figura 3.36: Desarrollo del algoritmo en Node-Red para construir indicadores con los datos cuantificados y enviados vía WiFi por el prototipo

En el programa construido en Node-Red de la figura 3.36 se encuentra un nodo en color naranja

con nombre *cprtraining* e icono de un cilindro en color blanco, éste nodo es el que contiene las funcionalidades de la base de datos.

El nodo *cprtraining* se conecta con la base de datos y permite gestionar la información, de manera que se puede agregar, modificar, eliminar y consultar registros. A continuación, se presenta la tabla *entrenamiento\_2* con algunos datos de prueba agregados desde el prototipo funcionando como se planeo desde la etapa de diseño.



The screenshot shows a database query interface. At the top, a query editor contains the SQL statement: `SELECT * FROM `entrenamiento_2``. Below the editor, a toolbar includes options for '1 Result', '2 Profiler', '3 Messages', '4 Table Data', and '5 Info'. The main area displays a table with 13 rows of data. Each row has a checkbox on the left. The table columns are: ID, comp\_tot, comp\_efec, porc\_t\_len, porc\_t\_ok, porc\_t\_rap, tiempo\_prom, and fecha\_reg.

ID	comp_tot	comp_efec	porc_t_len	porc_t_ok	porc_t_rap	tiempo_prom	fecha_reg
<input type="checkbox"/> 1	29	16	0.00	40.00	60.00	1.06	2021-05-11 02:31:12
<input type="checkbox"/> 2	29	16	0.00	40.00	60.00	1.06	2021-05-11 15:45:00
<input type="checkbox"/> 3	40	20	20.00	40.00	40.00	1.06	2021-05-11 19:30:47
<input type="checkbox"/> 4	50	35	50.00	45.00	5.00	0.76	2021-05-11 22:06:33
<input type="checkbox"/> 5	50	35	50.00	45.00	5.00	0.76	2021-05-11 22:07:28
<input type="checkbox"/> 6	50	35	50.00	45.00	5.00	0.76	2021-05-11 22:08:07
<input type="checkbox"/> 7	50	35	50.00	45.00	5.00	0.76	2021-05-11 22:08:37
<input type="checkbox"/> 8	50	35	50.00	45.00	5.00	0.76	2021-05-11 22:09:04
<input type="checkbox"/> 9	32	26	0.00	20.00	80.00	0.58	2021-05-12 09:17:38
<input type="checkbox"/> 10	0	0	0.00	0.00	0.00	0.00	2021-05-12 09:19:21
<input type="checkbox"/> 11	20	15	0.00	33.33	66.67	0.87	2021-05-12 09:19:57
<input type="checkbox"/> 12	50	41	0.00	22.22	77.78	0.41	2021-05-12 09:56:27
<input type="checkbox"/> 13	22	14	25.00	25.00	50.00	0.86	2021-05-12 09:57:27

Figura 3.37: Consulta a la tabla *entrenamiento\_2* con algunos registros de prueba enviados desde el prototipo

Teniendo datos almacenados en la tabla se pueden construir indicadores basados en esta información.

Se presenta a continuación un dashboard con los reportes construidos con base en los datos enviados por el dispositivo a través de la red WiFi.

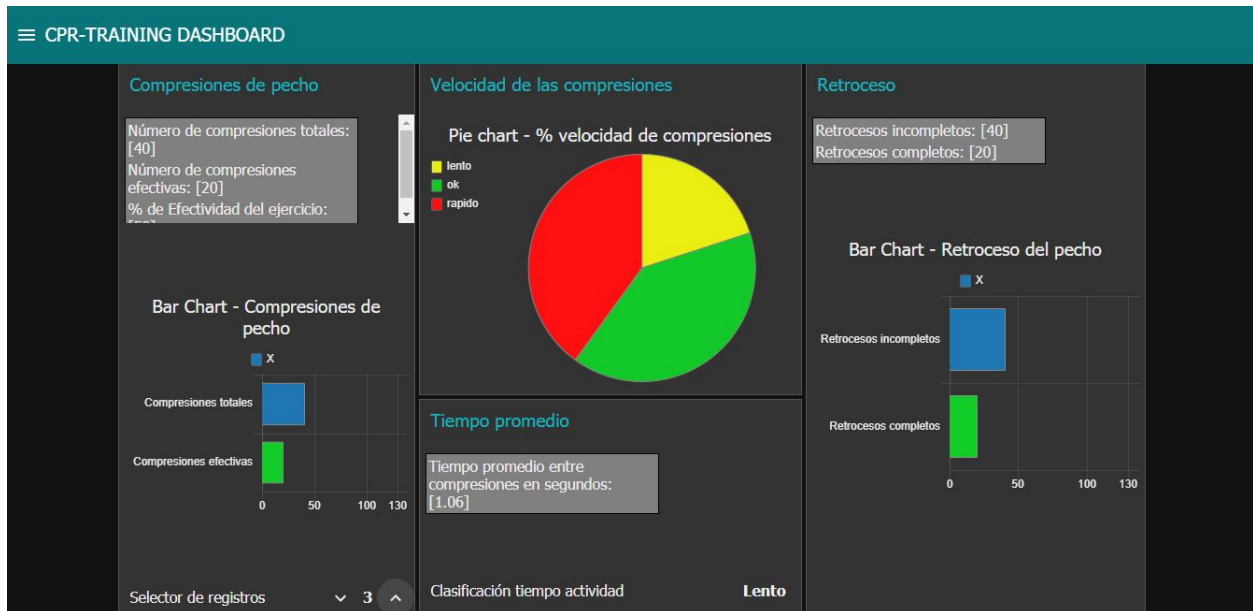


Figura 3.38: DASHBOARD CPR-TRAINING

Se elaboraron 4 grupos para la vista del dashboard:

1. Compresiones de pecho
2. Velocidad de las compresiones
3. Retroceso
4. Tiempo promedio

Cada uno de estos grupos contiene al menos 1 dato de 1 columna de la tabla *entrenamiento\_2*.

El grupo compresiones de pecho realiza una consulta con un *SELECT* a los campos *compresiones totales* y *compresiones efectivas* de la tabla *entrenamiento\_2*. Los datos de cada registro se presentan en indicadores tipo texto y gráfico de barras del grupo compresiones de pecho, además se construye una ecuación para arrojar el porcentaje de efectividad del ejercicio realizado dividiendo el total de compresiones efectivas entre las compresiones totales.

En el mismo grupo compresiones de pecho se encuentra un botón con la funcionalidad de selector de registros. Este botón se puede utilizar para navegar en el dashboard por los diferentes registros de la tabla *entrenamiento\_2*.

Continuando con la explicación del dashboard el grupo velocidad de las compresiones realiza una consulta *SELECT* a los campos *porcentaje velocidad lento*, *porcentaje velocidad ok* y *porcentaje*

velocidad rápido de la tabla *entrenamiento\_2*. Los datos se presentan como una sumatoria del 100 % en una gráfica de torta. Cada registro de estas 3 columnas en la tabla almacena el porcentaje del total que le corresponde.

El grupo retroceso realiza una consulta *SELECT* a los campos compresiones totales y compresiones efectivas. Este grupo presenta una gráfica de barras al igual que la de compresiones totales. En teoría durante el desarrollo del algoritmo para cuantificar estas variables de la subsección 2.3.6 el lograr compresiones efectivas depende directamente de realizar un retroceso del pecho completo, por otra parte, realizar una compresión que no cumpla con la profundidad mínima de 5cm influye en que no se tenga en cuenta el retroceso del pecho como un retroceso completo y se cuantifique como un retroceso incompleto.

Para finalizar, el grupo tiempo promedio arroja un indicador de texto informando el tiempo promedio entre compresiones de pecho en unidad de segundos durante 1 minuto de ejecución de la actividad, además presenta otro indicador de texto que clasifica el tiempo promedio de la actividad entre lento, ok o rápido.

*-Para cumplir con el objetivo 3 se realizaron los siguientes procedimientos:*

### 3.6. Evaluación consumo de corriente del prototipo

Se realiza un análisis práctico del consumo de corriente del prototipo para comparar los resultados con el cálculo teórico realizado en la subsección 2.3.7.1 tabla 2.4. Se procedió a medir el consumo del prototipo electrónico para comparar los 108.26 mA obtenidos teóricamente contra los 136.1 mA que arrojó el multímetro teniendo en cuenta que este consumo de corriente se mide conectado el medidor en serie con el circuito desde la fuente de alimentación con la pantalla LCD conectada, el microprocesador, reguladores de voltaje y componentes pasivos (resistencias, capacitores).

A continuación, se presenta una imagen tomada al multímetro durante la medición de corriente del prototipo.

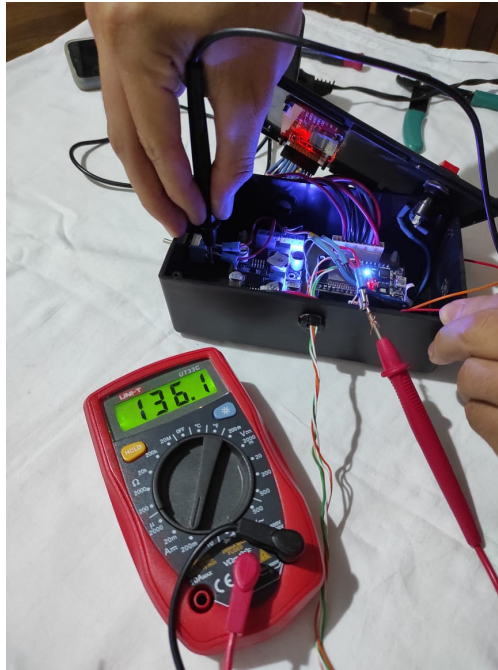


Figura 3.39: Medición consumo de corriente DC prototipo

Se realiza un cálculo para saber el porcentaje de error que se obtuvo entre la medición experimental y el cálculo teórico

$$25,7\% = \left\| \frac{108,26 - 136,1}{108,26} \right\| * 100 \quad (3.3)$$

El valor de porcentaje de error obtenido es de 25.7%. Finalmente, las características de la fuente de alimentación elegida para el prototipo se encuentran sobredimensionadas para ambos valores (Teórico y experimental).

## 3.7. Evaluación del prototipo y resultados finales

### 3.7.1. Evaluación del prototipo en el hospital simulado del campus universitario

Se realiza el desplazamiento del prototipo hacia el campus universitario el día 27/05/2021 02:00 p.m, en donde el coordinador científico Oscar Echeverry y la Ingeniería Biomédica Stephanie Ordoñez del hospital simulado fueron los encargados de evaluar el funcionamiento del sistema.

Para la evaluación del sistema se planteó un cronograma de actividades de la siguiente manera:

1. Conexión del prototipo a la red WiFi de la Universidad e instalación del sensor en el maniquí de pruebas LAERDAL.
2. Inicialización de los servidores de base de datos e IoT.
3. Pruebas iniciales de funcionamiento del sensor montado en el maniquí.
4. Explicación de los desarrolladores acerca del manejo de la herramienta a los evaluadores.
5. Evaluación practica del prototipo por parte de los expertos.
6. Retroalimentación de la actividad.

Se inicia la ejecución de las actividades en el cronograma comenzando por la conexión a la red WiFi de la universidad y la instalación del sensor en el maniquí. La conexión a la red fue exitosa al igual que el ensamble del sensor en el maniquí LAERDAL. Se realizaron pruebas iniciales con el prototipo encendido y el sensor ubicado en el maniquí y funciona de manera igual que en el modelo del torso casero.

A continuación, se realiza la explicación del funcionamiento del sistema que se les dio a los evaluadores.

El prototipo CPR-TRAINING es un dispositivo electrónico que se compone de un módulo electrónico con interfaz gráfica, el elemento sensor y un conector a la red eléctrica domestica 120VAC. Una vez identificadas las partes del dispositivo se debe ubicar el elemento sensor al interior del maniquí de RCP situándolo como un objeto fijo dentro del maniquí preferiblemente en la zona de la espalda frente a alguno de los pectorales en donde se tiene el movimiento de las compresiones de pecho.



Figura 3.40: Ubicación del sensor en el maniquí

Después de ubicar el sensor el maniquí esté debe armarse completamente y se debe garantizar que el usuario no inicie las compresiones antes de encender el dispositivo, esto con el objetivo de permitir que al encender el prototipo el sensor estime la distancia interna del torso del maniquí sin presión externa (La única presión que debe existir en ese instante es la del peso de los elementos de la estructura del maniquí).

Una vez se enciende el sistema el usuario puede ubicarse en posición para hacer reanimaciones y presionar el botón *start* cuando quiera iniciar la actividad de práctica.

### 3.7.2. Resultados de la evaluación

La actividad inicia con un maniquí LAERDAL en el cual el desplazamiento del resorte permite comprimir el pecho a lo mucho 5.4 cm (Dato experimental brindado por los expertos del hospital en función a su experiencia de trabajo con este equipo) debido a que viene configurado de fábrica con estas características, además de que el maniquí incorpora en su interior un elemento sonoro de contacto mecánico que trabaja como freno indicando al usuario que la compresión llegó a los 5cm

de profundidad.

El valor de profundidad arrojado por el CPR-TRAINING oscilaba entre 5cm y 5.4cm a lo máximo cuando se ejercía una compresión efectiva.

Después de evaluar la profundidad se valida el indicador de velocidad de las compresiones. Para probar este indicador el evaluador Oscar Echeverry utiliza una aplicación móvil llamada *Metronomo Beats* disponible en Google play store [45] para llevar la frecuencia de las compresiones al *beat* que la aplicación produce por medio de un sonido.

Primero configura la app en 90 *beats* por segundo y realiza a este ritmo las compresiones de pecho en el maniquí. El resultado es el indicador en color amarillo con el texto *lento* arrojado por pantalla.

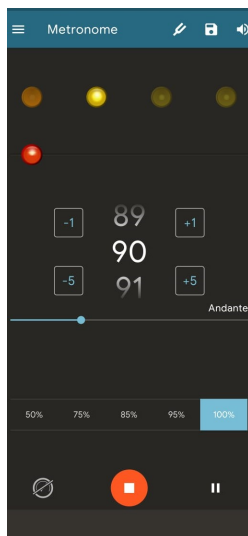


Figura 3.41: *Metronomo beats* a 90 beat/s

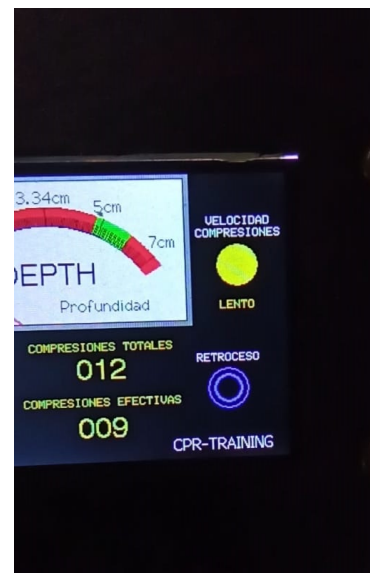


Figura 3.42: Indicador velocidad pantalla: *Lento*

Luego ubica el metrónomo en 105 beats por segundo y el resultado arrojado por pantalla es el indicador en color verde con el texto *ok*.

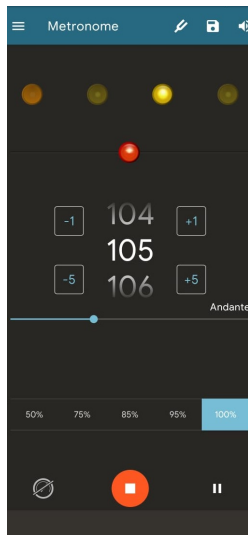


Figura 3.43: *Metronomo beats* a 105 beat/s



Figura 3.44: Indicador velocidad pantalla: *OK*

Finalmente ubica el metrónomo en 125 beats por segundo y el resultado arrojado por pantalla es el indicador en color rojo con el texto rápido.

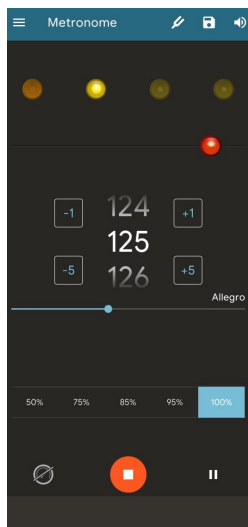


Figura 3.45: *Metronomo beats* a 125 beat/s

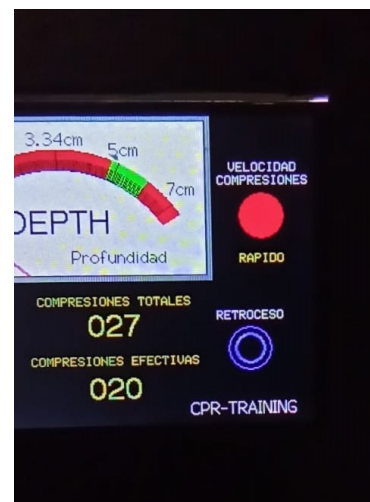


Figura 3.46: Indicador velocidad pantalla: *Rápido*

En esta etapa se logro validar un retraso de aproximadamente 1 segundo en el indicador al cambiar de velocidad.

Después de haber realizado pruebas en el maniquí LAERDAL se ubico el sensor en otro maniquí del laboratorio con marca NASCO. Este maniquí tiene en el laboratorio aproximadamente 20 años y no tiene ningún componente electrónico. Los evaluadores querían ver el desempeño del sistema en un maniquí sin instrumentación electrónica.

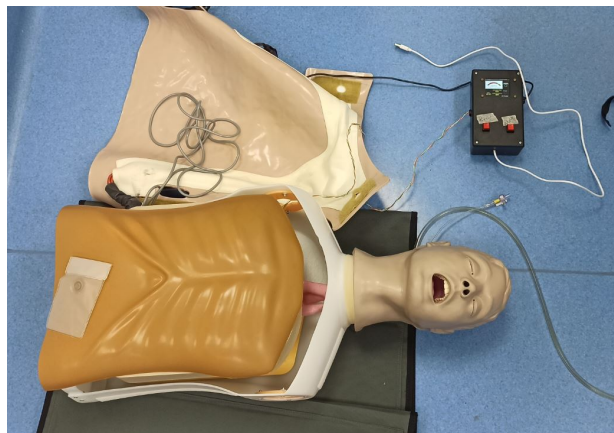


Figura 3.47: Prototipo CPR-TRAINING en maniquí NASCO sin armar

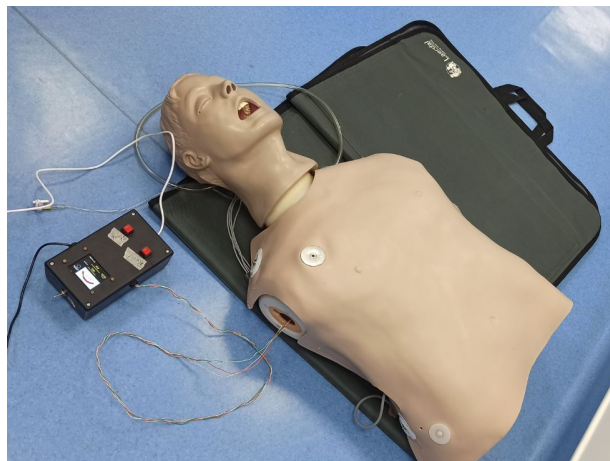


Figura 3.48: Prototipo CPR-TRAINING en maniquí NASCO armado completamente

Con el maniquí NASCO Se realizaron las mismas pruebas en el mismo orden y los resultados fueron más satisfactorios. El maniquí NASCO no tiene en su interior un sistema de freno que le obstaculice desplazar la compresión de los resortes más allá de 5cm permitiendo como máximo hasta 6.5cm. Tampoco incorpora una alarma sonora que indique al usuario que llego al punto máximo de compresión recomendado por las guías de la AHA de manera que resulta útil realizar la practica en este maniquí equipado con el prototipo debido a que ayuda al usuario a sensibilizar sus manos, brazos y hombros al momento de realizar una compresión que debe estar en el rango entre 5cm y 6cm.

El indicador de profundidad del prototipo arrojaba distancias máximas de 6.5cm en el maniquí NASCO.

Finalmente, las pruebas de velocidad de compresiones se realizaron también con la aplicación móvil del metrónomo. Las pruebas para el indicador de retroceso estaban relacionadas directamente con las pruebas de compresiones debido a que es una alarma visual de 2 estados que tiene un punto de partida y uno final en el cual siempre se debe procurar llevarlo al punto de partida (el 0).

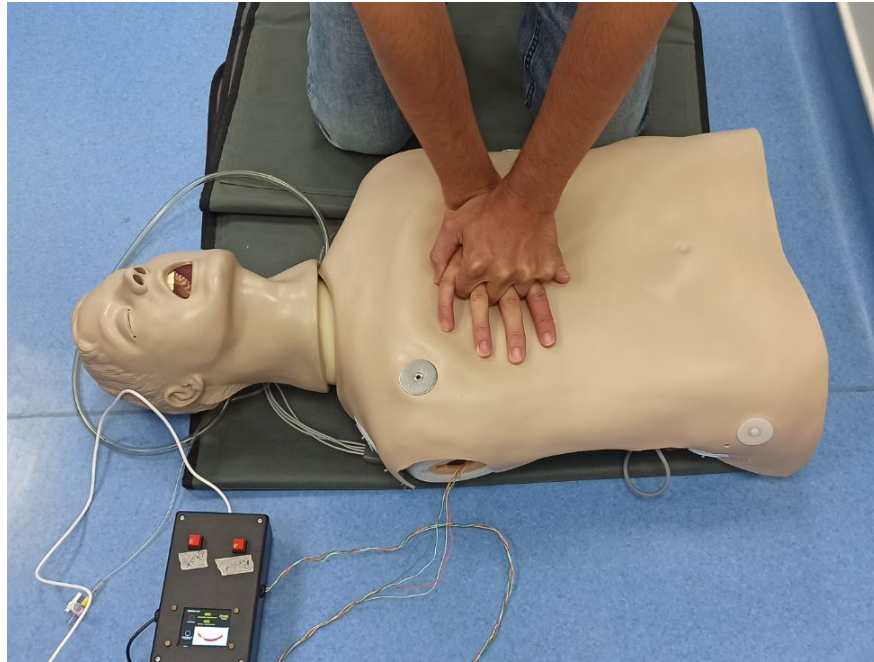


Figura 3.49: Evaluación del prototipo en el maniquí NASCO

Como sugerencia final del coordinador científico aconsejó manejar un código de color diferente del rojo para el indicador de retroceso debido a que en la práctica en las ciencias de la salud el rojo simboliza señales de urgencias en los dispositivos y en este indicador de retroceso solo simbolizaba que una compresión se había realizado y que debía llevarse el pecho hasta 0cm para cumplir con la regla de la efectividad de la compresión de pecho.

Por último, se presentan los datos en el dashboard correspondientes al resultado de la actividad realizada por el evaluador Oscar Echeverry.

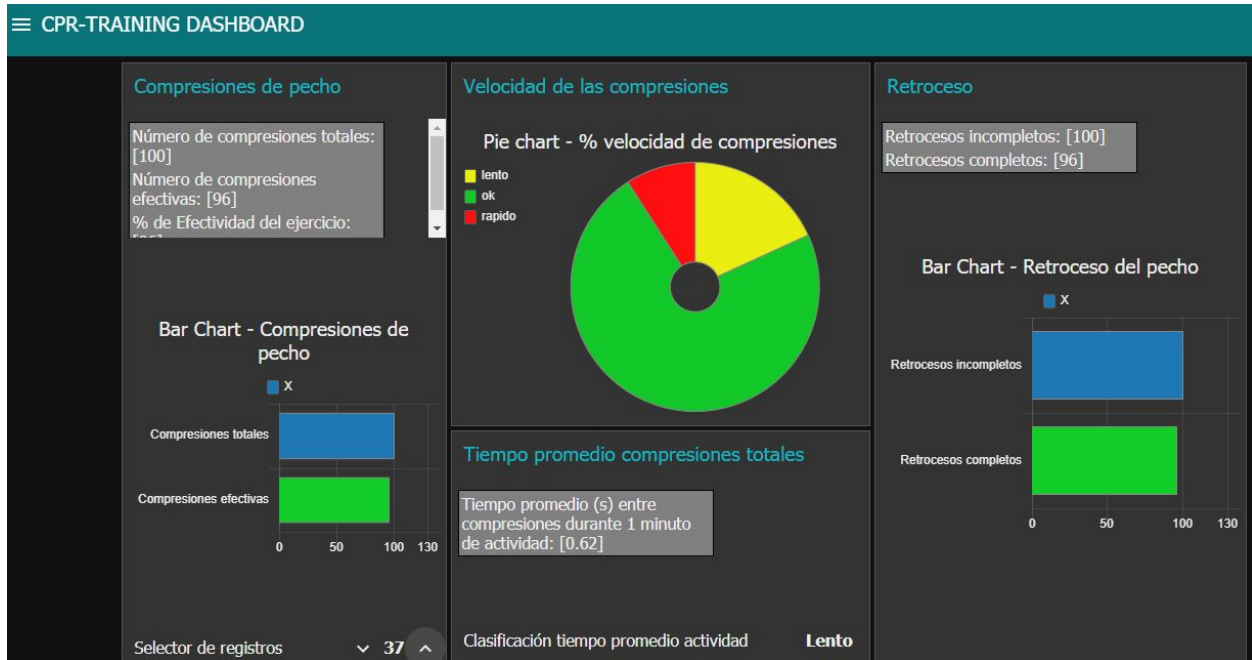


Figura 3.50: Métricas del evaluador Oscar Echeverry en el Dashboard

### 3.7.3. Comparación con otras tecnologías existentes en el hospital simulado

Características	QCPR Leardal	CPR-TRAINING
Mide profundidad, velocidad compresiones, retroceso, numero de compresiones totales y efectivas	Si	Si
Presenta datos en tiempo real	Si	Si
Almacena datos históricos	No	Si
Autonomía fuente de alimentación	Aceptable (Batería AA)	Aceptable
Resistencia al polvo, agua, intemperie	No	No

Tabla 3.3: Comparación entre QCPR Laerdal y CPR-TRAINING

Características	PRESTAN CPR	CPR-TRAINING
Mide profundidad, velocidad compresiones, retroceso, numero de compresiones totales y efectivas	No	Si
Presenta datos en tiempo real	Si	Si
Almacena datos históricos	No	Si
Autonomía fuente de alimentación	Aceptable (Batería AA)	Aceptable
Resistencia al polvo, agua, intemperie	No	No

Tabla 3.4: Comparación entre PRESTAN CPR y CPR-TRAINING

### 3.8. Futuros trabajos

El prototipo al final cumple con los objetivos específicos planteados desde el anteproyecto sin embargo durante la realización del proyecto se lograron identificar diferentes mejoras que podrían darle al prototipo un ajuste para convertirse en dispositivo final. Por otra parte, también surgieron ideas que no se encontraban dentro de los requerimientos de los objetivos pero que podrían servir a otros estudiantes para plantear trabajos de grado relacionados con el tema objeto de estudio. A continuación, se mencionan algunas de estas ideas:

- Para este trabajo se implemento un modelo casero de torso para RCP de bajo costo para hacer pruebas y ajustes rápidos en el prototipo. Este modelo se construyo en función a la arquitectura que tiene un maniquí de RCP en la actualidad en donde se modela la resistencia del cuerpo humano con un resorte. Este modelo plantea que un usuario al practicar RCP en maniquís se encontrara con un objeto de comportamiento lineal, pero al realizar la actividad de RCP en la vida real se encontrara con que el cuerpo humano tiene un comportamiento no lineal.

Para este punto se plantea estudiar la manera de diseñar, implementar y evaluar un modelo de cuerpo humano no lineal para maniquí de RCP.

- Una mejora importante para este proyecto seria construir un modelo relacional con tabla de usuarios que registre los datos de las personas que realicen las practicas de RCP, por ejemplo: Nombres, apellidos, teléfono, correo electrónico, semestre, login y password. Con estos datos se podría guardar los registros de las sesiones de RCP por usuarios y se podría usar los nodos de Node-Red que integran a Telegram como herramienta de mensajería instantánea y correo electrónico de Gmail para enviar notificaciones y avances a los usuarios.
- Una de las funcionalidades que podría añadirse al prototipo es la variable de las ventilaciones. La variable de las ventilaciones entre compresiones de pecho se podría integrar con otros sensores que midan presión absoluta por ejemplo y a partir de allí se agregarían las funciones

que integren la cuantificación de la variable mas los respectivos indicadores en la pantalla TFT y en el dashboard de Node-Red.

- Se podría pensar como mejora tener un sistema de alimentación dual que integre al sistema de alimentación por toma de corriente 120VAC actual. Este sistema de alimentación podría convertir portable a campo abierto el prototipo de manera que en lugares donde no se encuentre un punto de acceso a la red eléctrica domestica 120VAC se pueda utilizar el prototipo con otro tipo de alimentación como baterías.
- Actualmente la funcionalidad del sistema es la obtención de las variables, luego su cuantificación y por último se envían los datos a la base de datos para posteriormente ser publicados en el dashboard de Node-Red. Una mejora importante sería tratar de obtener los resultados en tiempo real, con esto el usuario podría ver su actividad en cualquier momento de la práctica de RCP sin la necesidad de tener la obligación de terminar su sesión de practica y a la vez tendría una retroalimentación más oportuna.
- Pensando en el usuario y cuyo objetivo es que el estudiante logre aprender y mejorar su técnica de RCP de una manera efectiva, se podría agregar en el dashboard mensajes basados en artículos científicos de salud de RCP que le sean de utilidad al usuario para mejorar su técnica dependiendo de los errores que este presentando según los datos que fueron captados durante su práctica.

### 3.9. Conclusiones

- Se logró el desarrollo de un prototipo para maniquí de RCP con sistema de retroalimentación visual, monitoreo de fácil entendimiento y trazabilidad de los datos generados por el estudiante el cual cumple con los requerimientos de las guías de la AHA. El sistema incluye servicios IoT como complemento de información para el usuario final que pueda ser de gran ayuda para su proceso de aprendizaje de la técnica de RCP. Como consecuencia cumple con los requerimientos que se establecieron en el diseño de la solución propuesta.
- Se encontró que para darle solución a este tipo de problemáticas es necesario involucrar a expertos en el tema objeto de estudio que desde su experiencia y aplicación de destrezas en el ámbito laboral apoyen en el proceso de diseño de tal forma que el proyecto construido sea un conjunto de soluciones adoptadas desde los desarrolladores hasta los interesados.
- Durante la planeación y diseño del prototipo del sistema se tuvieron en cuenta diferentes factores para la selección de insumos y componentes como lo son la comparación de tecnologías, sus costos de adquisición, lugares de adquisición (nacional o internacional), garantía de los productos ofrecidos, formas de envío, facturación que ayude a controlar los gastos para realizar balance de cuentas y validar si los gastos se ajustan, están por debajo o sobrepasan el presupuesto planeado en el anteproyecto de este trabajo.

- 
- Se implementó un sensor ultrasónico HY-SRF05 por su diseño y construcción el cual permite una medición muy acertada con un error inferior al 1 % en la salida después de su linealización en el rango de escala métrica decimal que se utiliza en el proyecto. Cabe resaltar que el sensor se clasifica como sensores de distancia a diferencia de otros tipos de sensores (movimiento y fuerza) que se tuvieron en cuenta en la etapa de planeación y diseño del sistema. Se elige trabajar con el sensor de distancia debido a que cuenta con pocos antecedentes de trabajo los cuales hayan dado validez de su funcionamiento y por lo tanto, se puede catalogar un sistema novedoso al utilizar este sensor.
  - Asociar distintas disciplinas en un trabajo de investigación tiene como ventaja discutir diferentes puntos de vista que ayudan al desarrollo de un producto robusto y apropiado para su aplicación práctica. Por consiguiente, el trabajo de diferentes disciplinas con un mismo objetivo ayuda a encontrar soluciones apropiadas al problema planteado descubriendo así una herramienta profesional tan importante como lo es el trabajo en equipo multidisciplinar.

---

## 4.1. Códigos

```
1 //-----  
2 //  Librerias  
3 //-----  
4 #include <TFT_eSPI.h>  
5 #include <SPI.h>  
6 #include <stdio.h>  
7 #include <stdlib.h>  
8 #include <WString.h>  
9 #include <WiFi.h>  
10 #include <PubSubClient.h>  
11  
12 TFT_eSPI tft = TFT_eSPI();      // LLamado a la libreria TFT  
13  
14 #define TFT_GREY 0x5AEB  
15 //-----  
16 //Redes WiFi  
17 //-----  
18 //const char * WIFISSID = "HOME-A48B";  
19 //const char * WIFIPASSWORD = "18166BAE57A35259";
```

```
20
21 //const char* WIFISSID = "HOME-891A";
22 //const char* WIFIPASSWORD = "A560D589A35B9812";
23
24 const char* WIFISSID = "Javeriana";
25 const char* WIFIPASSWORD = "";
26
27 #define PUERTOMQTT 1883
28 //const char* BROKERMQTT = "192.168.1.68";
29 const char* BROKERMQTT = "10.70.0.139";
30 //-----
31 // Variables para controlar el inicio del loop
32
33 int LOOP_STATE_STOPPED = 0;
34 int LOOP_STATE_STARTED = 1;
35
36 int loopState = LOOP_STATE_STOPPED;
37 //-----
38 // Variables boton push
39
40 const int startbuttonPin = 33;    // start button
41 int startbuttonState = 0; //state start button
42
43 const int restartbuttonPin = 27;    // restart button
44 int restartbuttonState = 0; //state restart button
45 //-----
46 // Variables velocidad compresiones
47
48 float contador_porcentaje_len = 0.0;
```

```
49 float contador_porcentaje_ok = 0.0;
50 float contador_porcentaje_rap = 0.0;
51 float contador_final = 0.0;
52 float contador_1 = 0.0;
53 float contador_2 = 0.0;
54 float contador_3 = 0.0;
55 float tiempo_total = 0.0;
56 float t_medio_total = 0.0;
57 float tiempo = 0.0;
58 float t_medio = 0.0;
59 //-----
60 // Variables contador compresiones pecho TOTALES
61
62 uint8_t ccT=00; // compresiones de pecho
63 byte occT = 99;
64 int T_bandera1=0; //boolean control
65 int T_bandera2=0; //boolean control
66
67 uint32_t target_time = 0;
68
69 int cpm = 0;
70 //-----
71 // Variables contador compresiones pecho efectivas
72
73 uint8_t cc=00; // compresiones de pecho
74 byte occ = 99;
75 int bandera1=0; //boolean control
76 int bandera2=0; //boolean control
77 //-----
```

```
78 // Variables reloj Digital
79
80 uint32_t targetTime = 0;           // Proximo segundo
81 uint8_t mm=00;                     // Minutos
82 uint8_t ss=00;                     // Segundos
83
84 byte omm = 99, oss = 99;
85 byte xcolon = 0, xsecs = 0;
86 unsigned int colour = 0;
87 //-----
88 //Variables medidor analógico
89
90 #define M_SIZE 1.0 // Tamaño medidor analógico
91
92 float ltx = 0; // Guarda la coord en X del piso de la aguja
93 //del medidor analogico
94 uint16_t osx = M_SIZE*120, osy = M_SIZE*120; // Saved x & y coords
95 uint32_t updateTime = 0; // tiempo actualización
96
97 int old_analog = -999; // Value last displayed
98 //int old_digital = -999;
99 int value = 0; // medidor analógico bv
100 int old_value = -1; // medidor analógico
101 //-----
102 //Variables sensor Ultrasonido
103 #define triggerEmisor 26 //Pin trigger Ultrasonido
104 #define echoReceptor 25 //Pin echo Ultrasonido
105
106 float distancia;
```

```
107 int temporal=0;
108 int diferencial;
109 //-----
110 //Variables pub-sub client
111
112 String topic_1 = "SensorHall";
113
114 WiFiClient ClienteWifi;
115 PubSubClient ClienteMQTT(ClienteWifi);
116
117 uint32_t ultimoIntentoReconexion;
118 //-----
119 int false_print_time = 1; //Variable para controlar el tiempo ...
    falso por pantalla
120 int false_print_count_cc = 1; //Variable para controlar el ...
    contador de cc falso por pantalla
121
122 //-----
123 void setup(void) {
124 //-----
125 //Pantalla tft
126 tft.init();
127 tft.setRotation(1);
128 tft.fillScreen(TFT_BLACK);
129 analogMeter(); // Dibuja el medidor analógico
130 updateTime = millis(); // Variable temporal medidor ...
    analogico
131 targetTime = millis() + 1000; //Variable temporal reloj ...
    digital
```

```
132 target_time = millis() + 1000;
133
134 Serial.begin(115200); // For debug
135
136 //Botón push
137 pinMode(startbuttonPin, INPUT_PULLUP);
138 pinMode(restartbuttonPin, INPUT_PULLUP);
139
140 //pub-sub client
141 Serial.println("Iniciando programa..."); //Iniciando la ...
    conexión WiFi
142 conectarAlWifi();
143 ClienteMQTT.setServer(BROKERMQTT, PUERTOMQTT); //le decimos ...
    cual es el servidor y el puerto al que se debe conectar
144 ClienteMQTT.setCallback(callback); //le decimos como se llama la ...
    funcion de callback
145
146 //Sensor ultrasonido
147 pinMode(triggerEmisor, OUTPUT); // El emisor emite por lo que es ...
    configurado como salida
148 pinMode(echoReceptor, INPUT); // El receptor recibe por lo que ...
    es configurado como entrada
149
150 //Titulos
151 tft.setTextColor(TFT_YELLOW, TFT_BLACK);
152 tft.drawString("COMPRESIONES EFECTIVAS", 96, 185, 1);
153
154 tft.setTextColor(TFT_YELLOW, TFT_BLACK);
155 tft.drawString("COMPRESIONES TOTALES", 102, 138, 1);
```

```
156
157 //Titulo "Tiempo:"
158 tft.setTextColor(TFT_WHITE);
159 tft.setCursor(25,180);
160 tft.print("TIEMPO:");
161
162 tft.drawString("CPR-TRAINING",220,220,2);
163
164 //Titulo "Retrosceso:"
165 tft.drawString("RETROCESO",242,150,1);
166
167 //Titulo "Velocidad Compresiones:"
168 tft.drawString("VELOCIDAD",253,35,1);
169 tft.drawString("COMPRESIONES",244,45,1);
170
171 //Titulo "Circulo blanco:"
172 tft.drawCircle(278, 75, 17, TFT_WHITE); //Circulo sin color de ...
    relleno (Punto inicial X, Punto inicial Y, Radio del ...
    circulo, Color)
173 tft.drawCircle(268, 180, 17, TFT_BLUE); //Circulo sin color de ...
    relleno (Punto inicial X, Punto inicial Y, Radio del ...
    circulo, Color)
174 }
175 void loop() {
176
177   if(mm == 0 && ss == 59){
178
179     if (!ClienteMQTT.connected()){
180       if (millis() - ultimoIntentoReconexion > 250){
```

```
181     ultimoIntentoReconexion = millis();
182     // Attempt to reconnect
183     if (reconexion()) {
184         ultimoIntentoReconexion = 0;
185     }
186 }
187 } else {
188     ClienteMQTT.loop();
189 }
190
191 if( ClienteMQTT.connected() ){
192     //ClienteMQTT.publish("/Conexion","conectado");
193     publicar();
194 }
195 }
196 if( false_print_time == 1){
197
198     tft.setTextColor(TFT_YELLOW, TFT_BLACK);
199     tft.setCursor(15,190);
200     tft.drawString("00:00",15,190,4);
201 }
202
203 if( false_print_count_cc == 1){
204
205     tft.setTextColor(TFT_YELLOW, TFT_BLACK);
206     tft.setCursor(120,190);
207     tft.drawString("000",140,200,4);
208     tft.drawString("000",140,150,4);
209 }
```

```
210
211  if (digitalRead(startbuttonPin) == HIGH) {
212    if (loopState == LOOP_STATE_STOPPED){
213      loopState = LOOP_STATE_STARTED;
214
215    } else { // if(loopState == LOOP_STATE_STARTED)
216      loopState = LOOP_STATE_STOPPED;
217    }
218  }
219
220  if (loopState == LOOP_STATE_STARTED) {
221    false_print_time = 0; //Arroja en pantalla un 00:00 hasta que ...
      se presiona start
222    false_print_count_cc = 0; //Arroja en pantalla un 000 hasta ...
      que se presiona start
223
224    if (updateTime < millis()) {
225      updateTime = millis() + 25; // Delay to limit speed of update
226
227      calcular_distancia();
228
229      plotNeedle((10.0/7.0)*value, 0); // Toma entre 2 y 12ms en ...
      volver a dibujar la aguja
230
231      compresiones_pecho();
232      reloj();
233    }
234  }
235
```

```
236   if (digitalRead(restartbuttonPin) == HIGH){
237       loopState = LOOP_STATE_STOPPED;
238       reiniciar();
239   }
240 }
241 //-----
242 //                               FUNCIONES
243 //-----
244 //                               FUNCIÓN DISTANCIA SENSOR ULTRASONIDO
245 //-----
246 void calcular_distancia(){
247
248     //Para estabilizar el valor del pin Trig se establece a LOW
249     digitalWrite (triggerEmisor, LOW);
250     delayMicroseconds(2);
251
252     //Se lanzan los 8 pulsos del emisor
253     digitalWrite (triggerEmisor, HIGH);
254     delayMicroseconds(10);
255     digitalWrite (triggerEmisor, LOW);
256     distancia= (0.171600*(pulseIn (echoReceptor, HIGH))); ...
                //centimetros/useg
257
258     /* Velocidad del sonido en el aire es de 0,034 cm/microsegundos
259        ida y vuelta nos interesa solo la mitad (0,034/2 = 0,017) */
260
261     if (temporal==0){
262         temporal=distancia;
263     }
```

```
264 diferencial=temporal-distancia;
265
266 //offset gráfica
267 if (diferencial < 7){
268     diferencial = 0;
269     value = diferencial;
270 } else {
271     value = diferencial;
272 }
273 }
274 //-----
275 // FUNCIÓN CONTADOR COMPRESIONES DE PECHO Y VISUALIZACIÓN
276 //-----
277 void compresiones_pecho() {
278     //-----
279     //Compresiones efectivas
280
281     if (value < 1){
282         bandera1=1;
283     }
284
285     if (value > 49){
286         bandera2=1;
287
288         if (bandera1 == 1 && bandera2 == 1){
289             occ = cc;    // Guarda el ultimo valor para la ...
                actualización del display
290             cc++;        // Contador compresiones
291             bandera1=0;
```

```
292     bandera2=0;
293     }
294 }
295
296 tft.setTextSize(1);
297 tft.setTextColor(TFT_YELLOW, TFT_BLACK);
298 int xpos = 140;
299 int ypos = 200; // Extremo superior izquierdo del recuadro ...
    del contador
300
301 if (occ != cc) { // Contador
302     occ = cc;
303     // Dibuja
304     if (cc < 100) xpos += tft.drawChar('0', xpos, ypos, 4); ...
        // Pone un 0 si las compresiones son <100
305     if (cc < 10) xpos += tft.drawChar('0', xpos, ypos, 4); ...
        // Pone un 0 si las compresiones son <10
306     xpos += tft.drawNumber(cc, xpos, ypos, 4); ...
        // Dibuja la cantidad de compresiones efectivas
307 }
308 //-----
309 //Compresiones TOTALES
310
311 if (value < 10){
312     T_bantera1=1;
313 }
314
315 if (value > 40){
316     T_bantera2=1;
```

```
317
318     if (T_bantera1 == 1 && T_bantera2 == 1){
319         occT = ccT;    // Guarda el ultimo valor para la ...
                        // actualización del display
320         ccT++;        // Contador compresiones
321         cpm++;
322     }
323     T_bantera1=0;
324     T_bantera2=0;
325 }
326
327 tft.setTextSize(1);
328 tft.setTextColor(TFT_YELLOW, TFT_BLACK);
329 int xposss = 140;
330 int yposss = 150; // Extremo superior izquierdo del recuadro ...
                        // del contador
331
332 if (occT != ccT) { // Contador
333     occT = ccT;
334     // Dibuja
335     if (ccT < 100) xposss += tft.drawChar('0', xposss, yposss, ...
                        // 4); // Pone un 0 si las compresiones son <100
336     if (ccT < 10) xposss += tft.drawChar('0', xposss, yposss, ...
                        // 4); // Pone un 0 si las compresiones son <10
337     xposss += tft.drawNumber(ccT, xposss, yposss, 4); ...
                        // Dibuja la cantidad de compresiones efectivas
338 }
339 //-----
340 //Velocidad de las compresiones
```

```
341
342     if (cpm != 0 || cc != 0){
343
344         if (target_time < millis()) {
345             // Set next update for 1 second later
346             target_time = millis() + 1000;
347             tiempo++;
348
349             t_medio = (tiempo/cpm);
350             t_medio_total = (60.0/cc);
351
352             if(cpm > 1){
353
354                 if(t_medio > 0.6){
355
356                     tft.fillCircle(278, 75, 17, TFT_YELLOW); //Circulo ...
357                     // sin color de relleno (Punto inicial X, Punto ...
358                     // inicial Y, Radio del circulo, Color)
359                     tft.fillRect(255, 100, 45, 15, TFT_BLACK);
360                     tft.drawString("LENTO",263,105,1);
361                     contador_1++;
362                 }
363
364                 if(t_medio ≥ 0.5 && t_medio ≤0.6){
365
366                     tft.fillCircle(278, 75, 17, TFT_GREEN); //Circulo ...
367                     // sin color de relleno (Punto inicial X, Punto ...
368                     // inicial Y, Radio del circulo, Color)
369                     tft.fillRect(255, 100, 45, 15, TFT_BLACK);
```

```
366         tft.drawString("OK",273,105,1);
367         contador_2++;
368     }
369
370     if(t_medio < 0.5){
371
372         tft.fillCircle(278, 75, 17, TFT_RED); //Circulo sin ...
            color de relleno (Punto inicial X, Punto inicial ...
            Y, Radio del circulo, Color)
373         tft.fillRect(255, 100, 45, 15, TFT_BLACK);
374         tft.drawString("RAPIDO",261,105,1);
375         contador_3++;
376     }
377
378     tiempo = 0.0;
379     cpm = 0;
380 }
381 }
382 (contador_final = contador_1 + contador_2 + contador_3);
383
384 if(contador_final != 0){
385     contador_porcentaje_len = (contador_1/contador_final)*100;
386     contador_porcentaje_ok = (contador_2/contador_final)*100;
387     contador_porcentaje_rap = (contador_3/contador_final)*100;
388 }
389 }
390 /*
391 Serial.print("lento");
392 Serial.println(contador_porcentaje_len);
```

```
393     Serial.print("");
394     Serial.print("ok");
395     Serial.println(contador_porcentaje_ok);
396     Serial.print("");
397     Serial.print("rapido");
398     Serial.println(contador_porcentaje_rap);
399     Serial.print("");*/
400     //-----
401     //Retrosceso
402
403     if (bandera1 == 0 && bandera2 == 1){
404         tft.fillCircle(268, 180, 12, TFT_BLUE); //Circulo relleno ...
            de color (Punto inicial X, Punto inicial Y, Radio del ...
            circulo, Color)
405     }
406     if (bandera1 == 1 && bandera2 == 1){
407         tft.fillCircle(268, 180, 10, TFT_BLACK); //Circulo relleno ...
            de color (Punto inicial X, Punto inicial Y, Radio del ...
            circulo, Color)
408     }
409 }
410
411 //-----
412 //  FUNCIÓN RELOJ DIGITAL
413 //-----
414
415 void reloj(){
416
417     tft.setTextSize(1);
```

```
418 tft.setTextColor(TFT_YELLOW, TFT_BLACK);
419
420 if (targetTime < millis()) {
421     // Set next update for 1 second later
422     targetTime = millis() + 1000;
423
424     // Ajusta los valores del tiempo adicionando 1 segundo
425     ss++;           // Contador segundos
426     if (ss == 60) { // Condición para saber cuando se cumple 1 ...
427         minuto
428         ss = 0;     // Reinicia los segundos a 0
429         omm = mm;  // Guarda el ultimo minuto para la ...
430         // actualización del display
431         mm++;      // Contador minutos
432         if (mm > 59) { // Condición para saber cuando se cumple 1 ...
433             hora
434             mm = 0;
435         }
436     }
437 }
438
439 // Actualiza el reloj digital
440 int xpos = 15;
441 int ypos = 190; // Extremo superior izquierdo del recuadro ...
442 // del contador
443 int ysecs = 190;
444
445 if (omm != mm) { // Contador de minutos
446     omm = mm;
447     // Dibuja los minutos
```

```
443     if (mm < 10) xpos += tft.drawChar('0', xpos, ypos, 4); // ...
        Pune un 0 si los minutos son <10
444     xpos += tft.drawNumber(mm, xpos, ypos, 4);           // ...
        Dibuja los minutos
445     xsecs = xpos;
446 }
447 if (oss != ss) { // Contador de segundos
448     oss = ss;
449     xpos = xsecs;
450
451     if (ss % 2) { // Flash the colons on/off
452         tft.setTextColor(0x39C4, TFT_BLACK);           // Alterna a ...
            gris el color de ":"
453         xpos += tft.drawChar(':', xsecs, ysecs, 4); // ":"
454         tft.setTextColor(TFT_YELLOW, TFT_BLACK);      // Colores
455     }
456     else {
457         xpos += tft.drawChar(':', xsecs, ysecs, 4); // ":"
458     }
459
460     //Segundos
461     if (ss < 10) xpos += tft.drawChar('0', xpos, ysecs, 4); // ...
        Pune un 0 si los segundos son <10
462     tft.drawNumber(ss, xpos, ysecs, 4);               // ...
        Dibuja los segundos
463 }
464 }
465 }
466 //-----
```

```
467 // FUNCIÓN PARA DIBUJAR EL MEDIDOR ANALÓGICO
468 //-----
469
470 void analogMeter()
471 {
472     // Meter outline
473     tft.fillRect(0, 0, M_SIZE*239, M_SIZE*126, TFT_GREY);
474     tft.fillRect(5, 3, M_SIZE*230, M_SIZE*119, TFT_WHITE);
475     tft.setTextColor(TFT_BLACK); // Color texto
476
477     // Draw ticks every 5 degrees from -50 to +50 degrees (100 deg. ...
        FSD swing)
478     for (int i = -50; i < 51; i += 1) {
479         // Long scale tick length
480         int t1 = 15;
481
482         // Coodinates of tick to draw
483         float sx = cos((i - 90) * 0.0174532925);
484         float sy = sin((i - 90) * 0.0174532925);
485         uint16_t x0 = sx * (M_SIZE*100 + t1) + M_SIZE*120;
486         uint16_t y0 = sy * (M_SIZE*100 + t1) + M_SIZE*140;
487         uint16_t x1 = sx * M_SIZE*100 + M_SIZE*120;
488         uint16_t y1 = sy * M_SIZE*100 + M_SIZE*140;
489
490         // Coordinates of next tick for zone fill
491         float sx2 = cos((i + 5 - 90) * 0.0174532925);
492         float sy2 = sin((i + 5 - 90) * 0.0174532925);
493         int x2 = sx2 * (M_SIZE*100 + t1) + M_SIZE*120;
494         int y2 = sy2 * (M_SIZE*100 + t1) + M_SIZE*140;
```

```
495     int x3 = sx2 * M_SIZE*100 + M_SIZE*120;
496     int y3 = sy2 * M_SIZE*100 + M_SIZE*140;
497
498     // Green zone limits
499     if (i ≥ -50 && i < 22) {
500         tft.fillTriangle(x0, y0, x1, y1, x2, y2, TFT_RED);
501         tft.fillTriangle(x1, y1, x2, y2, x3, y3, TFT_RED);
502     }
503
504     // Orange zone limits
505     if (i ≥ 22 && i < 38) {
506         tft.fillTriangle(x0, y0, x1, y1, x2, y2, TFT_GREEN);
507         tft.fillTriangle(x1, y1, x2, y2, x3, y3, TFT_GREEN);
508     }
509
510     // Red zone limits
511     if (i ≥ 38 && i ≤ 50) {
512         tft.fillTriangle(x0, y0, x1, y1, x2, y2, TFT_RED);
513         tft.fillTriangle(x1, y1, x2, y2, x3, y3, TFT_RED);
514     }
515
516     // Short scale tick length
517     if (i % 25 != 0) t1 = 8;
518
519     // Recalculate coords incase tick lenght changed
520     x0 = sx * (M_SIZE*100 + t1) + M_SIZE*120;
521     y0 = sy * (M_SIZE*100 + t1) + M_SIZE*140;
522     x1 = sx * M_SIZE*100 + M_SIZE*120;
523     y1 = sy * M_SIZE*100 + M_SIZE*140;
```

```
524
525     // Draw tick
526     tft.drawLine(x0, y0, x1, y1, TFT_BLACK);
527
528     // Check if labels should be drawn, with position tweaks
529     if (i % 25 == 0) {
530         // Calculate label positions
531         x0 = sx * (M_SIZE*100 + t1 + 10) + M_SIZE*120;
532         y0 = sy * (M_SIZE*100 + t1 + 10) + M_SIZE*140;
533         switch (i / 25) {
534             case -2: tft.drawCentreString("0cm", x0, y0 - 12, 2); break;
535             case -1: tft.drawCentreString("1.67cm", x0, y0 - 9, 2); ...
                    break;
536             case 0: tft.drawCentreString("3.34cm", x0, y0 - 6, 2); ...
                    break;
537             //case 1: tft.drawCentreString("52.5", x0, y0 - 9, 2); ...
                    break;
538             case 1: tft.drawCentreString("5cm", x0, y0 - 9, 2); break;
539             case 2: tft.drawCentreString("7cm", x0, y0 - 12, 2); break;
540         }
541     }
542
543     // Now draw the arc of the scale
544     sx = cos((i + 5 - 90) * 0.0174532925);
545     sy = sin((i + 5 - 90) * 0.0174532925);
546     x0 = sx * M_SIZE*100 + M_SIZE*120;
547     y0 = sy * M_SIZE*100 + M_SIZE*140;
548     // Draw scale arc, don't draw the last part
549     if (i < 50) tft.drawLine(x0, y0, x1, y1, TFT_BLACK);
```

```
550 }
551
552 tft.drawString("Profundidad", M_SIZE*(15 + 155 - 40), ...
    M_SIZE*(119 - 20), 2); // logo derecha abajo
553 tft.drawCentreString("DEPTH", M_SIZE*120, M_SIZE*70, 4); // ...
    Comment out to avoid font 4
554 tft.drawRect(5, 3, M_SIZE*230, M_SIZE*119, TFT_BLACK); // Draw ...
    bezel line
555
556 plotNeedle(0, 0); // Put meter needle at 0
557 }
558 //-----
559 // FUNCIÓN PARA DIBUJAR LA AGUJA VARIABLE DEL MEDIDOR ANALÓGICO
560 //-----
561 void plotNeedle(int value, byte ms_delay)
562 {
563     int value_print = value/(10.0/7.0);
564     if (value_print < 0) value_print = 0;
565     if (value_print > 70) value_print = 70;
566     tft.setTextColor(TFT_BLACK, TFT_WHITE);
567     char buf[8]; dtostrf(value_print, 4, 0, buf);
568
569     //tft.drawRightString(buf, M_SIZE*40, M_SIZE*(119 - 20), 2);
570
571     if (value < 0) value = 0; // Limit value to emulate needle end ...
        stops
572     if (value > 100) value = 100;
573
574     // Move the needle until new value reached
```

```
575 while (!(value == old_analog)) {
576     if (old_analog < value) old_analog++;
577     else old_analog--;
578
579     if (ms_delay == 0) old_analog = value; // Update immediately ...
        id delay is 0
580
581     float sdeg = map(old_analog, -10, 110, -150, -30); // Map ...
        value to angle
582     // Calcualte tip of needle coords
583     float sx = cos(sdeg * 0.0174532925);
584     float sy = sin(sdeg * 0.0174532925);
585
586     // Calculate x  $\Delta$  of needle start (does not start at pivot point)
587     float tx = tan((sdeg + 90) * 0.0174532925);
588
589     // Erase old needle image
590     // Erase old needle image
591     tft.drawLine(M_SIZE*(120 + 20 * ltx - 1), M_SIZE*(140 - 20), ...
        osx - 1, osy, TFT_WHITE);
592     tft.drawLine(M_SIZE*(120 + 20 * ltx), M_SIZE*(140 - 20), osx, ...
        osy, TFT_WHITE);
593     tft.drawLine(M_SIZE*(120 + 20 * ltx + 1), M_SIZE*(140 - 20), ...
        osx + 1, osy, TFT_WHITE);
594
595     // Re-plot text under needle
596     tft.setTextColor(TFT_BLACK);
597     tft.drawCentreString("DEPTH", M_SIZE*120, M_SIZE*70, 4); // ...
        // Comment out to avoid font 4
```

```
598     tft.drawString("Profundidad", M_SIZE*(15 + 155 - 40), ...
        M_SIZE*(119 - 20), 2); // logo derecha abajo
599
600     // Store new needle end coords for next erase
601     ltx = tx;
602     osx = M_SIZE*(sx * 98 + 120);
603     osy = M_SIZE*(sy * 98 + 140);
604
605     // Draw the needle in the new postion, magenta makes needle a ...
        bit bolder
606     // draws 3 lines to thicken needle
607     tft.drawLine(M_SIZE*(120 + 20 * ltx - 1), M_SIZE*(140 - 20), ...
        osx - 1, osy, TFT_RED);
608     tft.drawLine(M_SIZE*(120 + 20 * ltx), M_SIZE*(140 - 20), osx, ...
        osy, TFT_MAGENTA);
609     tft.drawLine(M_SIZE*(120 + 20 * ltx + 1), M_SIZE*(140 - 20), ...
        osx + 1, osy, TFT_RED);
610
611     // Slow needle down slightly as it approaches new postion
612     if (abs(old_analog - value) < 10) ms_delay += ms_delay / 5;
613
614     // Wait before next update
615     delay(ms_delay);
616 }
617 }
618 //-----
619 //  FUNCIÓN PARA REINICIAR LAS VARIABLES
620 //-----
621 void reiniciar(){
```

```
622 loopState = LOOP_STATE_STOPPED;
623 startbuttonState = 0;
624
625 contador_porcentaje_len = 0.0;
626 contador_porcentaje_ok = 0.0;
627 contador_porcentaje_rap = 0.0;
628 contador_final = 0.0;
629 contador_1 = 0.0;
630 contador_2 = 0.0;
631 contador_3 = 0.0;
632 tiempo_total = 0.0;
633 t_medio_total = 0.0;
634 tiempo = 0.0;
635 t_medio = 0.0;
636
637 false_print_time = 1; //Variable para controlar el tiempo falso ...
    por pantalla
638 false_print_count_cc = 1; //Variable para controlar el contador ...
    de cc falso por pantalla
639
640 ccT=00; // compresiones de pecho
641 occT = 99;
642 T_bandera1=0; //boolean control
643 T_bandera2=0; //boolean control
644 target_time = 0;
645 cpm = 0;
646 cc=00; // compresiones de pecho
647 occ = 99;
648 bandera1=0; //boolean control
```

```
649 bandera2=0; //boolean control
650 targetTime = 0; // Proximo segundo
651 mm=00; // Minutos
652 ss=00; // Segundos
653 omm = 99, oss = 99;
654 xcolon = 0, xsecs = 0;
655 colour = 0;
656 ltx = 0; // Guarda la coord en X del piso de la aguja
657 //del medidor analogico
658 updateTime = 0; // tiempo actualización
659 old_analog = -999; // Value last displayed
660 value = 0; // medidor analógico bv
661 old_value = -1; // medidor analógico
662 temporal=0;
663 tft.drawCircle(278, 75, 17, TFT_WHITE); //Circulo sin color de ...
        relleno (Punto inicial X, Punto inicial Y, Radio del ...
        circulo, Color)
664 tft.fillCircle(278, 75, 16, TFT_BLACK); //Circulo sin color de ...
        relleno (Punto inicial X, Punto inicial Y, Radio del ...
        circulo, Color)
665 tft.drawCircle(268, 180, 17, TFT_BLUE); //Circulo sin color de ...
        relleno (Punto inicial X, Punto inicial Y, Radio del ...
        circulo, Color)
666 tft.fillCircle(268, 180, 16, TFT_BLACK); //Circulo sin color de ...
        relleno (Punto inicial X, Punto inicial Y, Radio del ...
        circulo, Color)
667 tft.fillRect(255, 100, 45, 15, TFT_BLACK);
668 }
669 //-----
```

```
670 // FUNCIÓN PARA CONECTARSE A LA RED WIFI
671 //-----
672 void conectarAlWifi() {
673   Serial.println("Iniciando conexión WiFi");
674   WiFi.begin(WIFISSID, WIFIPASSWORD);
675   Serial.print("Conectando a: ");
676   Serial.println(WIFISSID);
677   while (WiFi.status() != WL_CONNECTED) {
678     delay(500);
679     Serial.print(".");
680   }
681   Serial.println("");
682   Serial.println("WiFi Conectado ");
683   Serial.print("Direccion IP: ");
684   Serial.println(ClienteWifi.localIP());
685 }
686 //-----
687 // FUNCIÓN RECONEXION MQTT
688 //-----
689 boolean reconexion() {
690   Serial.println("Conectando al broker mqtt");    //intentando ...
        conectar al broker
691   if (ClienteMQTT.connect("ESP32Client")) {
692     Serial.println("Conectado");                //publicamos ...
        que estamos conectados
693   ClienteMQTT.publish("/conexion", "Conectado");
694   } else {
695     Serial.print("falló, rc=");
696     Serial.print(ClienteMQTT.state());
```

```
697 }
698 return ClienteMQTT.connected();
699 }
700 //-----
701 //  FUNCIÓN CALLBACK
702 //-----
703 void callback(char* topic, byte* payload, unsigned int length) {
704     String mensaje;
705     for(uint8_t i=0; i <length;i++){
706         mensaje+=char(payload[i]);
707     }
708 }
709 //-----
710 //  FUNCIÓN PARA PUBLICAR DATOS EN EL TOPIC
711 //-----
712 void publicar(){
713
714     char msg [255];
715     snprintf (msg, 255, "%d, %d, %f, %f, %f, %f", ...
716             ccT,cc,contador_porcentaje_len,contador_porcentaje_ok,
717             contador_porcentaje_rap,t_medio_total);
718     delay(1000);
719     ClienteMQTT.publish("SensorHall",msg);
719 }
```

Figura 4.1: Firmware para el microcontrolador ESP32 del prototipo final

# Pontificia Universidad Javeriana Cali

## Facultad de Ingeniería y Ciencias – Carrera Ingeniería Electrónica

### Acta 001 – Reunión para validar la implementación del prototipo y recibir recomendaciones para ajustes y cambios rápidos en el modelo 22/01/2021

Asistentes:

Oscar Javier Echeverry Gómez  
Coordinador Científico  
[oscar.echeverry@javerianacali.edu.co](mailto:oscar.echeverry@javerianacali.edu.co)

Valentina Corchuelo Guzmán  
Ingeniería Biomédica  
[valentina.corchuelo@javerianacali.edu.co](mailto:valentina.corchuelo@javerianacali.edu.co)

Firma: \_\_\_\_\_

Firma: \_\_\_\_\_

Juan Pablo Navarrete Morales  
Estudiante Pregrado Ingeniería Electrónica  
[juan96@javerianacali.edu.co](mailto:juan96@javerianacali.edu.co)

Juan Felipe Cifuentes Agudelo  
Estudiante Pregrado Ingeniería Electrónica  
[felipecifuentes@javerianacali.edu.co](mailto:felipecifuentes@javerianacali.edu.co)

Juan Pablo Navarrete M.

Firma: \_\_\_\_\_

Felipe Cifuentes

Firma: \_\_\_\_\_

### Objetivos

- Presentar a los asistentes del hospital simulado de la universidad Javeriana Cali el prototipo del sistema de retroalimentación visual con monitoreo y trazabilidad implementado hasta la fecha
- Recibir recomendaciones para implementar ajustes al modelo planteado

### Desarrollo de la reunión

Los estudiantes de pregrado de ingeniería electrónica Juan P Navarrete y Juan F Cifuentes realizan la presentación a los asistentes del hospital simulado del prototipo implementado hasta la fecha.

Para contextualizar el modelo textualmente se realiza una lista de los subsistemas presentados a la audiencia:

1. Implementación de hardware (Sensor, microprocesador, interfaz gráfica / Pantalla visualización de indicadores)
2. Modelo casero de un torso para realizar pruebas del sistema de retroalimentación visual

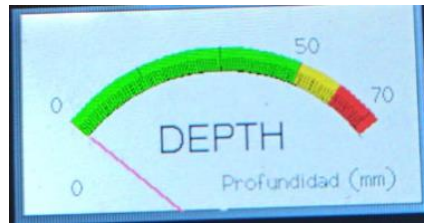
Durante la presentación del sistema se enseña en imágenes la forma como se encuentran representadas las variables en la pantalla de visualización.

### Indicador de profundidad

Para la variable de profundidad se presenta un indicador tipo dial con rangos de medida entre 0mm – 70mm. En el interior del rango también se representan las distancias por colores además de números y se definen los intervalos a continuación:

1. 0mm – 50mm Color verde (actividad por debajo de los parámetros de las guías de la AHA)
2. 50mm – 60 mm Color amarillo (actividad Ok – Rango ideal por parámetros de la guía de la AHA)
3. 60mm – 70mm Color rojo (Rango no ideal)

A continuación, se adjunta imagen del indicador como se presenta en la pantalla de visualización



**Recomendaciones recibidas de la audiencia para este indicador:**

1. Se debe realizar un cambio en el sistema de colores de la siguiente manera:
  - Rango ideal:** Color verde 50mm – 60mm
  - Rango no ideal:** Color Rojo 0mm – 50mm y 60mm – 70mm
2. Se debe realizar cambio en la nomenclatura de la unidad de medida, cambiar de mm a cm puesto que las recomendaciones de las guías de la AHA manejan la unidad de medida en cm.

**Indicador de frecuencia de las compresiones**

Para la variable de las frecuencias de las compresiones se presenta un indicador en barra vertical con aguja de desplazamiento para representar el tiempo ideal entre compresiones. Según las guías de la AHA el numero de compresiones debe estar entre 100 y 120 por minuto. Realizando un calculo con estas variables se tiene que para cumplir con el valor del intervalo inferior se deben realizar compresiones cada 500ms a lo menos y para cumplir con el valor del intervalo superior se deben realizar compresiones cada 600ms.

A continuación, una imagen del indicador en la pantalla



Donde CC es Chest Compressions (Compresiones de pecho). El 0.00 representa el valor del tiempo en segundos y la aguja se mueve según la velocidad de las compresiones. Una frecuencia ideal se encuentra entre 0.5 y 0.6 segundos.

**Recomendaciones recibidas de la audiencia para este indicador:**

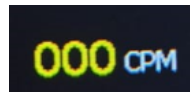
3. Los asistentes del hospital simulado argumentan que el usuario que realiza la practica de RCP en el maniquí espera saber si durante la ejecución del ejercicio va lento, va rápido o se encuentra en el rango de velocidad adecuada. La sugerencia aquí es modificar el indicador para que enseñe si esta por debajo, por encima o dentro del intervalo de frecuencias por minuto según las guías de la AHA.
4. Este indicador se puede representar por un sistema de colores como se describe a continuación:
  - <100 cpm/min (Color amarillo)
  - >120 cpm/min (Color rojo)
  - >100 cpm/min y <120 cpm/min (Color verde)

Se puede representar con una figura geométrica que indique el sistema de colores

**Indicador de numero compresiones efectivas**

Entre los indicadores de la pantalla se encuentra uno que acumula el numero de compresiones efectivas, es decir las realizadas en el rango de 50mm – 60mm.

A continuación, una imagen que representa el indicador:



**Recomendaciones recibidas de la audiencia para este indicador:**

5. El indicador se encuentra bien pero no describe totalmente la variable puesto que solo acumula compresiones efectivas. Se debe manejar un indicador que muestre el total de compresiones efectivas y no. Con el dato de las compresiones efectivas y el total de compresiones se puede realizar un calculo del porcentaje de efectividad.

**Indicador de retroceso**

El indicador de retroceso se representa como un circulo led que se enciende cuando se cumple con la compresión efectiva y se apaga cuando se realiza la liberación del pecho hasta 0cm. Si los valores de efectividad (>5cm) y liberación (= 0cm) no se cumplen entonces el indicador se quedará encendido o apagado hasta que se cumpla la condición.



**Recomendaciones recibidas de la audiencia para este indicador:**

6. Este indicador se puede seguir manejando para alertar cualitativamente acerca de la liberación efectiva del pecho. Se debe cambiar el color a verde para enviar el mensaje de OK.

Para finalizar, en función a las recomendaciones escritas en esta acta se realizarán ajustes en el prototipo para presentar al usuario y continuar con la evaluación del proyecto y plan de pruebas.

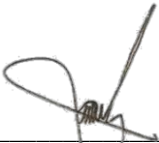
## Pontificia Universidad Javeriana Cali


### Facultad de Ingeniería y Ciencias – Carrera Ingeniería Electrónica

**Acta Reunión 002 – Reunión para evaluar la funcionalidad del prototipo con el coordinador científico y la ingeniería biomédica del hospital simulado de la Universidad Javeriana**  
27/05/2021 Asistentes:

Oscar Javier Echeverry Gómez  
Coordinador Científico  
[oscar.echeverry@javerianacali.edu.co](mailto:oscar.echeverry@javerianacali.edu.co)

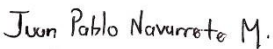
Stephanie Ordoñez Coordinador  
Ingeniera Biomédica  
[stephanie.ordóñez@javerianacali.edu.co](mailto:stephanie.ordóñez@javerianacali.edu.co)

Firma: 

Firma: 

Juan Pablo Navarrete Morales  
Estudiante Pregrado Ingeniería Electrónica  
[juan96@javerianacali.edu.co](mailto:juan96@javerianacali.edu.co)

Juan Felipe Cifuentes Agudelo  
Estudiante Pregrado Ingeniería Electrónica  
[felipecifuentes@javerianacali.edu.co](mailto:felipecifuentes@javerianacali.edu.co)

Firma: 

Firma: 

### Objetivos

- Construir un cronograma de actividades para la visita al hospital simulado.
- Presentar el prototipo del sistema al coordinador científico e ingeniera biomédica del hospital simulado.
- Explicar las funcionalidades del prototipo para realizar las pruebas y evaluaciones en el maniquí.

### Cronograma de actividades

1. Instalación del prototipo CPR-TRAINING en el maniquí del hospital simulado, conexión del prototipo a la red WiFi de la universidad, inicialización servidor de base de datos y servidor IoT. (30 minutos)
2. Demostración y pruebas de funcionamiento (15 minutos)
3. Evaluación del prototipo (60 minutos)
4. Retroalimentación de las actividades (15 minutos)

### Desarrollo de la reunión

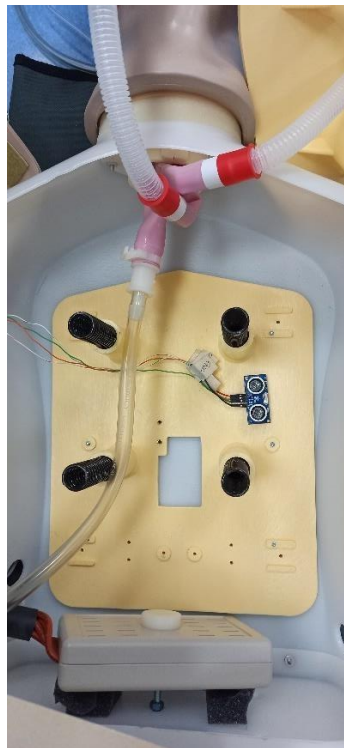
Se realiza encuentro presencial en una sala del hospital simulado de la universidad Javeriana Cali entre el director científico del hospital simulado Oscar Echeverry, la ingeniera biomédica Stephanie Ordoñez y los estudiantes Juan Pablo Navarrete y Juan Felipe Cifuentes.

Los estudiantes llevan al hospital el prototipo funcional del sistema de retroalimentación visual con monitoreo y trazabilidad para ser instalado en un maniquí de practica del hospital.

Cumpliendo con el primer punto del cronograma se realiza la instalación de la red WiFi de la universidad en el prototipo con el objetivo de enviar por internet los datos del ejercicio realizado durante la actividad con el maniquí a la plataforma IoT y finalmente observar las métricas de la practica en el dashboard diseñado para este propósito.

A continuación, se realiza la explicación del funcionamiento del prototipo desde su instalación. Primero se debe ubicar el sensor en la parte interior del maniquí (espalda interna del maniquí), el sensor debe quedar como un objeto fijo. Ahora bien, se realiza la explicación del por qué el sensor debe ir ubicado de esta forma.

Debido a que el sensor es de ultrasonido y su principio de funcionamiento se basa en emitir una señal a una frecuencia de 40KHz que al hacer contacto con un objeto solido es devuelta y recibida por el receptor del sensor debe ir puesto de manera fija en una superficie del maniquí que no presente movimiento, en este caso la espalda del maniquí.



Después se arma el maniquí completamente y una vez con todos sus elementos ajustados se conecta a la red eléctrica el prototipo y se enciende mediante el interruptor ON/OFF.

Cuando se enciende el prototipo se debe validar que el maniquí este completamente armado y que no se le esté ejerciendo fuerza sobre el pecho. Se debe garantizar esto para que el sensor al encenderse envíe la señal y estime la distancia interna entre pecho y espalda con los resortes presionados solo por el peso de los elementos de la estructura del maniquí.

Una vez realizado estos pasos se puede dar inicio al ejercicio presionando el botón *start*.

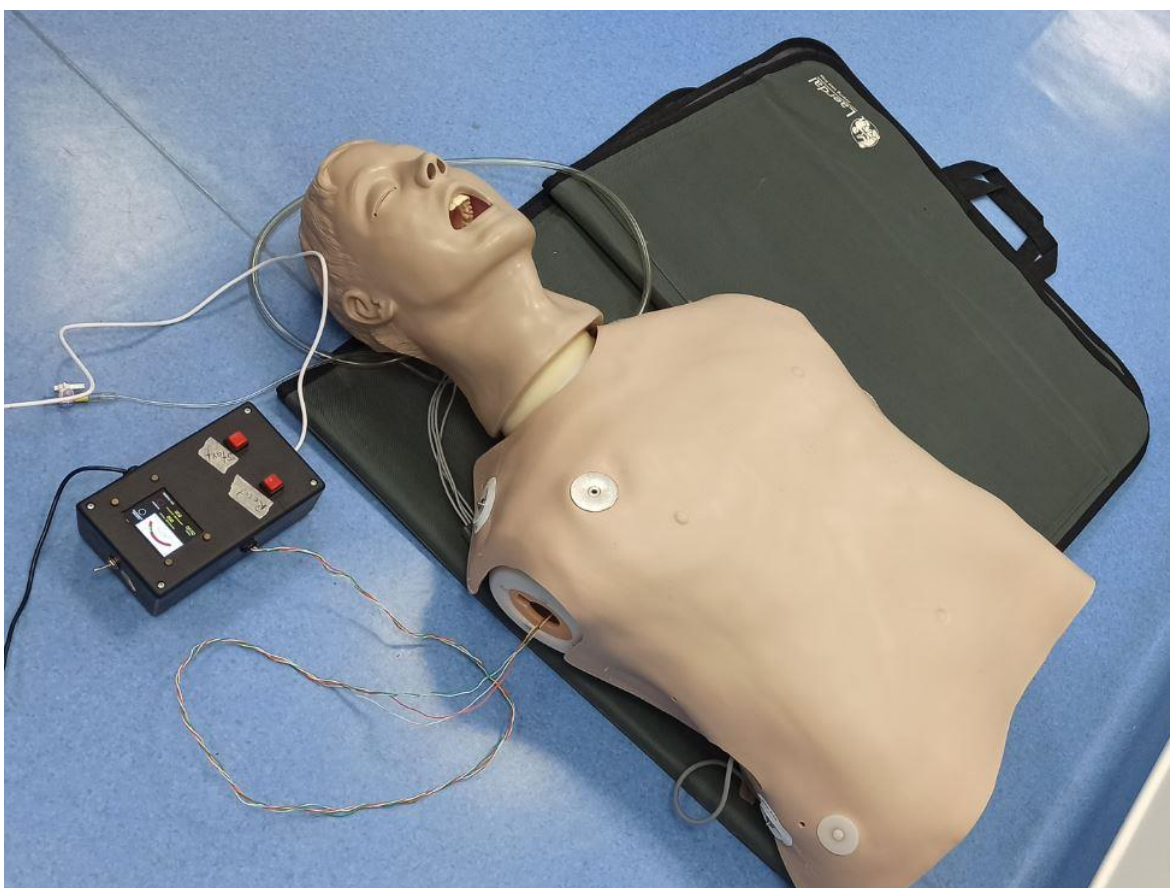


El director científico y la ingeniera biomédica realizan cada uno pruebas en el prototipo, primero en un maniquí LAERDAL y a continuación en un maniquí NASCO. Para ambos maniquís las distancias arrojadas en el indicador de la pantalla del prototipo se encontraban en el rango entre 0cm y 6cm. Estos maniquís no permiten extender la compresión más allá de 6cm.

Los valores arrojados en la pantalla para velocidad de las compresiones, indicador de retroceso, número de compresiones totales y efectivas se ajustaban en cada maniquí a la práctica del movimiento entre compresiones que cada usuario realizaba.

Finalmente se recibe por parte de los directivos del hospital simulado una retroalimentación positiva de la actividad en donde se satisfacen cada uno de los objetivos del proyecto respecto al prototipo funcional presentado.

**MANUAL DE USUARIO  
CPR-TRAINING**



**PONTIFICIA UNIVERSIDAD JAVERIANA CALI  
VALLE DEL CAUCA – COLOMBIA**

**2021**

## COMPONENTES DEL PROTOTIPO

1. Módulo de control electrónico con interfaz gráfica
2. Sensor ultrasónico
3. Conector red eléctrica domestica 120VAC



## 1. INTRODUCCIÓN

- **Objetivo**

Brindar soporte a los usuarios del sistema CPR-TRAINING enseñando de manera textual su forma de funcionamiento, cuidados del producto y líneas de atención al cliente.

- **Requerimientos**

- Fuente de alimentación domestica 120VAC con toma de voltaje convencional fase – neutro.
- Maniquí de RCP LAERDAL, NASCO, PRESTAN u otras marcas.
- Conexión a internet.
- Dirección IP de la red (Se conoce desde la ventana de comandos/terminal con la sentencia *ipconfig* en Windows).
- Parametrizar el nombre de la red, contraseña de acceso y dirección IP en el código fuente del CPR-TRAINING.
- Inicializar servidor base de datos y servidor IoT (Internet Of Things – Internet de las cosas)

## 2. INSTALACIÓN DE PROGRAMAS NECESARIOS.

- **Node.js**

Para la realizar la instalación de Node red es necesario instalar primero Node.js para para ejecutar un programa escrito en JavaScript.

- a) Ingresamos a la página oficial de Node.js

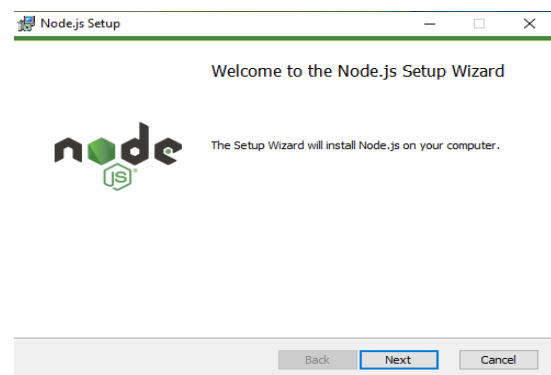
<https://nodejs.org/es/>



(1). Página Oficial node.js

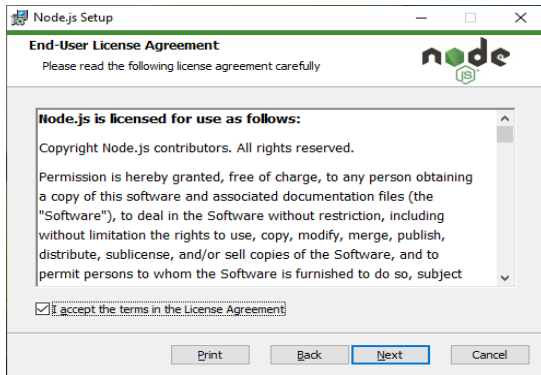
Nota: dependiendo del sistema operativo ingresamos al siguiente link: <https://nodejs.org/es/download/> para descargar el programa.

- b) Descargamos la versión recomendada para la mayoría, luego de su descarga abrimos el programa de instalación.



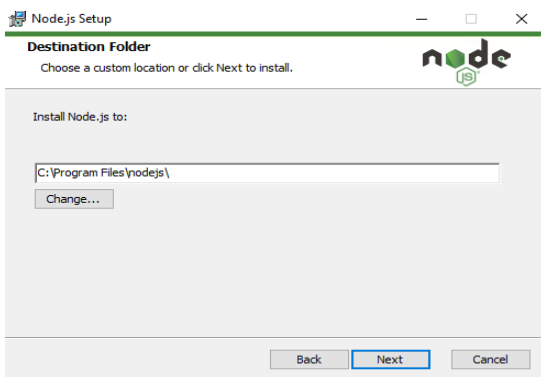
(2). Inicio de instalación

- c) Damos clic en siguiente y aceptamos la licencia del programa.



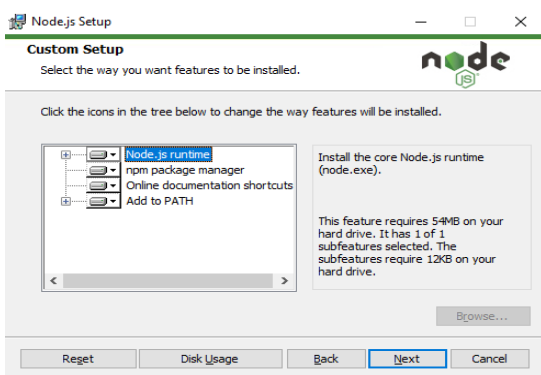
- (3). Licencia del programa.

- d) Se da clic en siguiente y el programa se instalará en una ruta por defecto del sistema operativo.



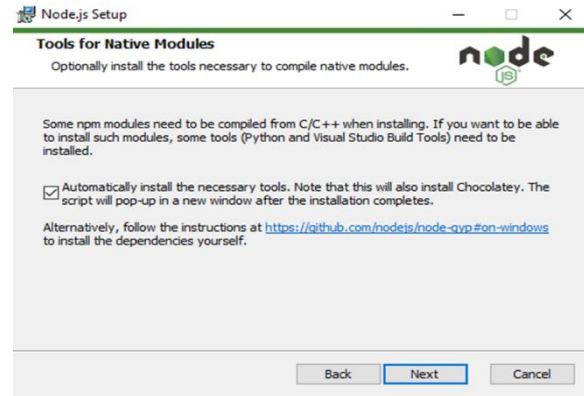
- (4). Ruta de instalación del programa.

- e) Se da clic en siguiente para su instalación completa.



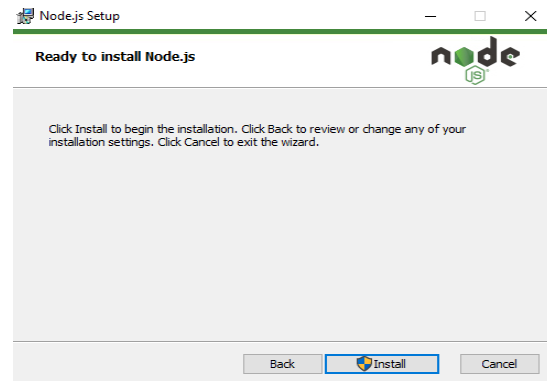
- (5). Instalación completa del programa.

- f) Seleccionamos la opción de instalar automáticamente las herramientas necesarias y posteriormente damos clic en siguiente.



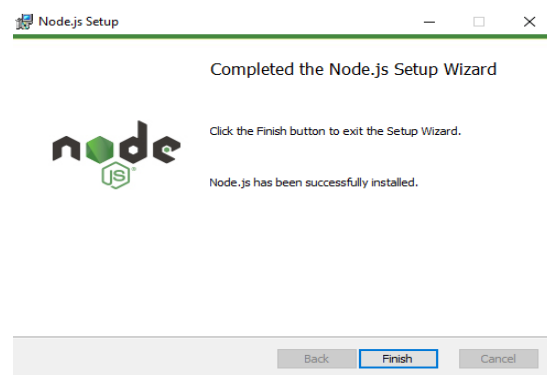
- (6). Herramientas a instalar en el programa.

- g) Luego damos clic en instalar.



- (7). Instalación node.js

- h) Por último, damos clic en finalizar.



- (8). Finalizar programa node.js

- i) Posteriormente luego de su instalación se abrirá automáticamente una pestaña para instalar las herramientas necesarias para el programa Node.js; damos clic con cualquier letra.

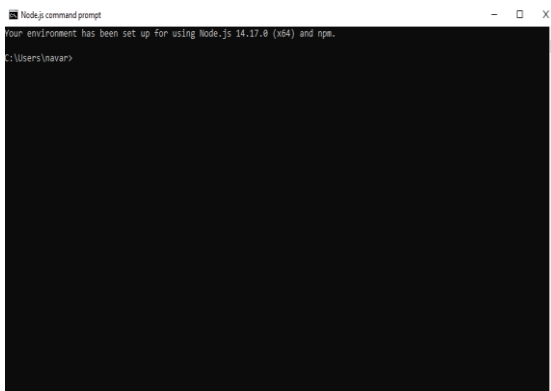


#### (9). Instalación de herramientas node.js

Nota: luego de presionar la tecla empieza la instalación, dependiendo del sistema operativo preguntara si está seguro de su instalación, dar clic en Si.

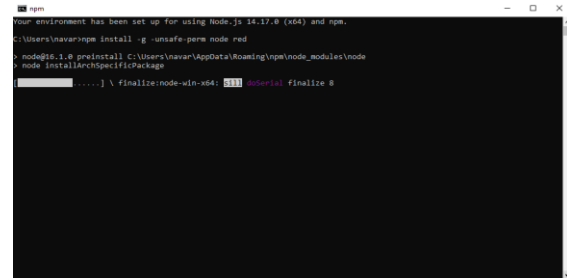
- **Node-Red**

- a) Luego de instalar Node.js abrimos “node.js command prompt”.



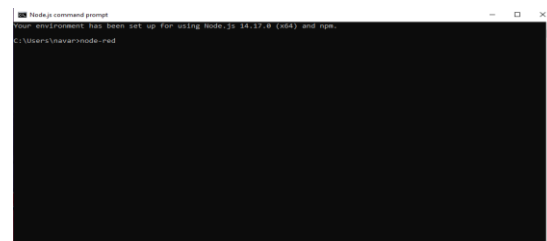
#### (10) Command prompt de node.js

- b) Escribimos en la ventana de comando **npm install -g --unsafe-perm node-red**. Damos clic en la tecla ENTER y automáticamente empezará a instalarse.



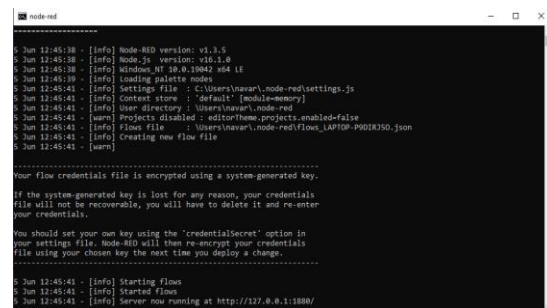
#### (11). Instalación de Node-red

- c) Luego cerramos “node.js command prompt”, reiniciamos el computador y posteriormente abrimos uno nuevo comando y escribimos “node-red” dando clic en la tecla ENTER.



#### (12). Comando para inicializar Node-red.

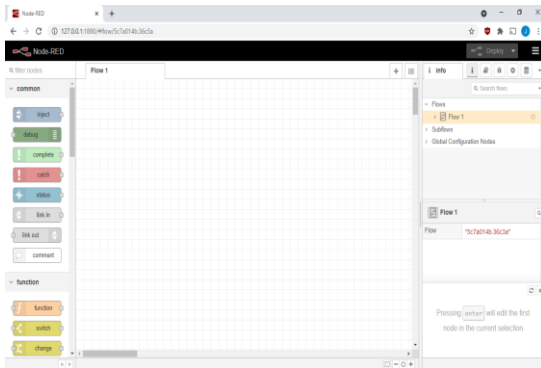
- d) Posteriormente se inicializa automáticamente el programa Node-red.



#### (13). Inicialización de Node-red.

- e) Por último, para abrir el programa Node-red abrimos nuestro navegador preferido y escribimos la siguiente dirección URL:

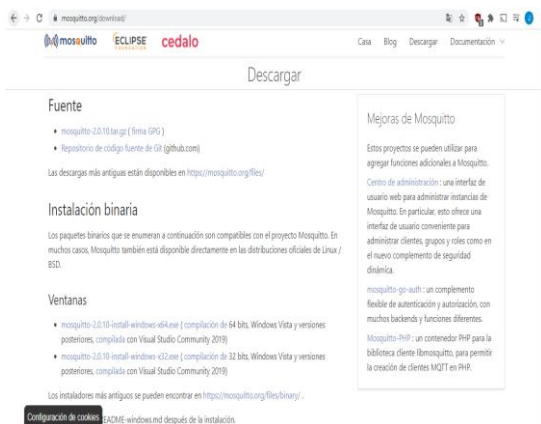
<http://127.0.0.1:1880/>



(14). Dashboard de Node-red.

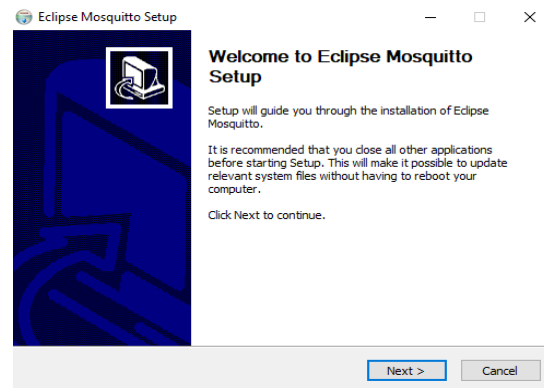
- **Broker Eclipse Mosquitto**

- a) Para instalar el Broker Mosquitto es necesario ir a la página oficial de Eclipse Mosquitto <https://mosquitto.org/> en la opción de descargas <https://mosquitto.org/download/> descargamos la versión de nuestro sistema operativo.



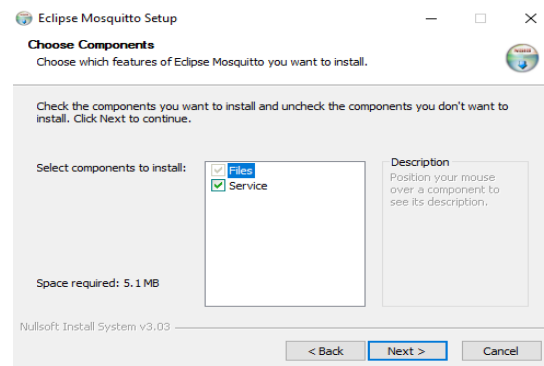
(15). Página oficial Eclipse Mosquitto.

- b) Abrimos el instalador posteriormente damos en siguiente.



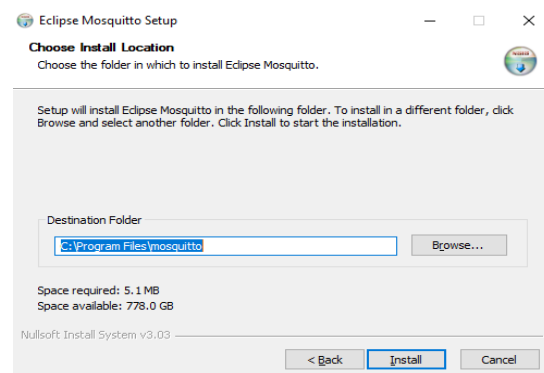
(16). Inicio de instalación de Eclipse Mosquitto.

- c) Instalamos la versión completa.



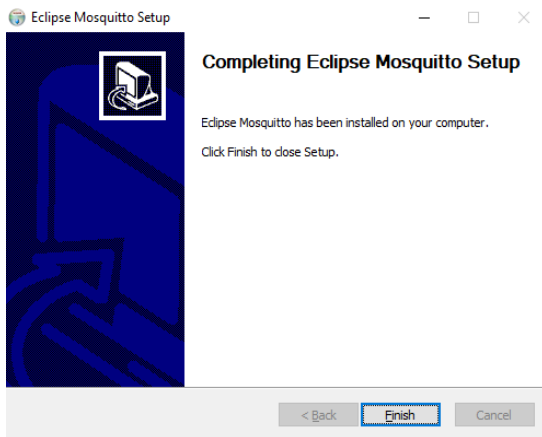
(17). Instalación completa del programa.

- d) Posteriormente ejecutamos la instalación.



(18). Instalación Eclipse Mosquitto

e) Por último, damos en finalizar la instalación.



(19). Finalizar programa Eclipse Mosquitto.

- **XAMPP**

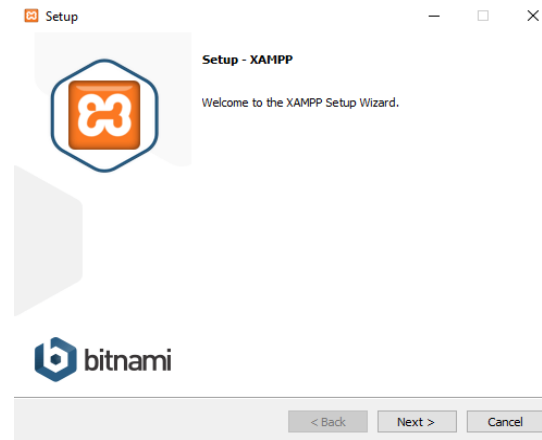
Para la Gestión de base datos es necesaria la Instalación de XAMPP es necesario lo siguientes pasos:

a) Ingresamos a la página oficial de XAMPP  
<https://www.apachefriends.org/es/index.html> y descargamos la versión para nuestro sistema operativo



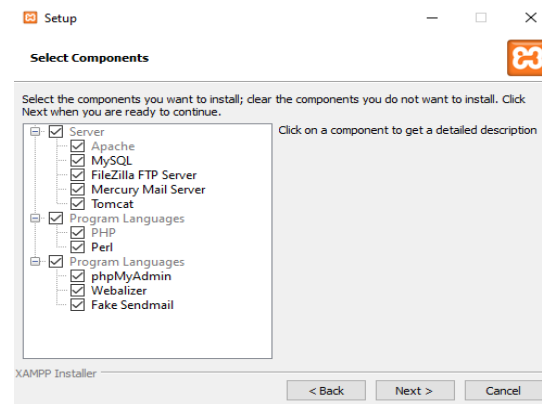
(20). Página Oficial de XAMPP.

b) Luego abrimos el instalador damos clic en siguiente para iniciar con el proceso de instalación.



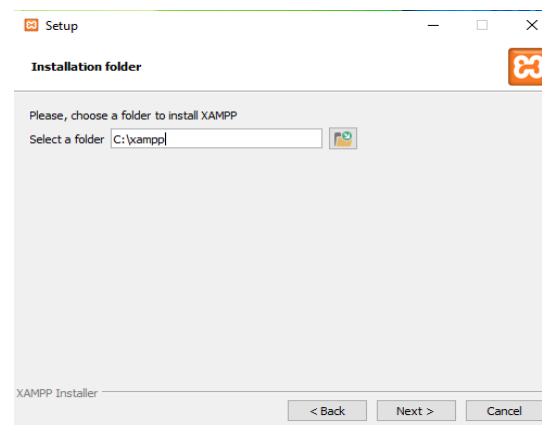
(21). Inicio de instalación del programa.

c) Instalamos la versión completa del programa.



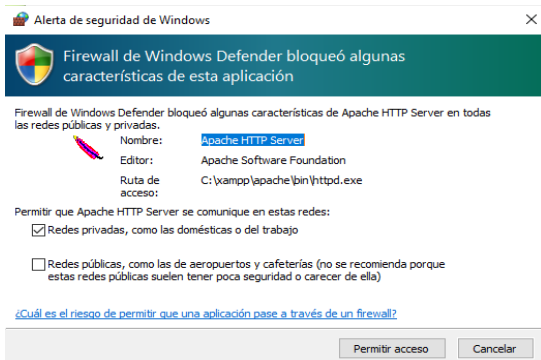
(22). Instalación completa del programa.

d) Damos clic en siguiente para que el programa empieza a instalarse.



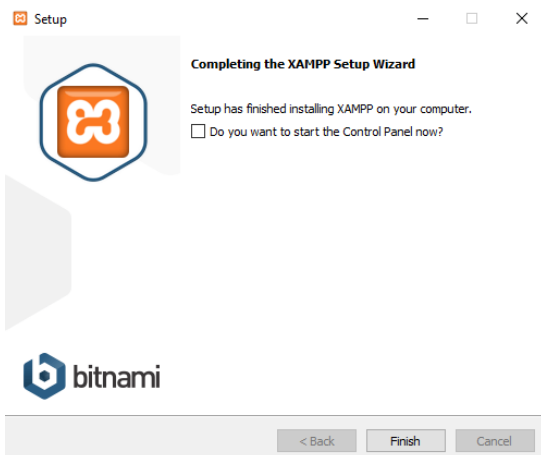
(23) Instalación XAMPP.

- e) Permitimos el acceso de XAMPP para que pueda estar en red.



(24). Permisos de firewall al programa XAMPP.

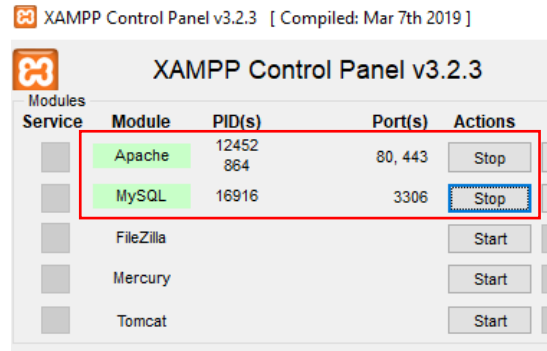
- f) Por último, damos clic en finalizar.



(25). Finalizar programa XAMPP.

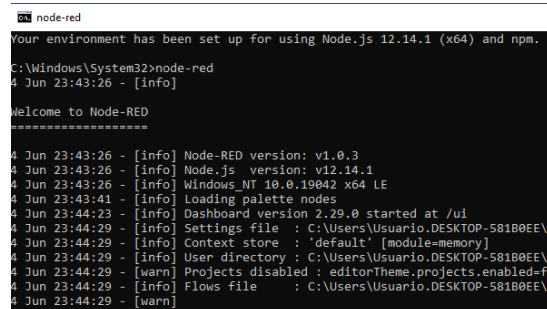
### 3. CONEXIÓN A LA RED WiFi / INICIALIZACIÓN DEL SERVIDOR IOT Y SERVIDOR BASE DE DATOS.

Para inicializar la base de datos se debe tener previamente instalado el programa de apache XAMPP. Una vez se inicia el programa se inicializa el apache y la base de datos MySQL (26).



(26). Panel de control de XAMPP

Después de inicializar el servidor apache se debe realizar la misma acción con el servidor IoT por medio de la ventana de comandos de Node-Js (27) (Se debe haber instalado previamente los recursos de Node-Js en la maquina).



(27). Ventana de comandos de Node-Js

Para inicializar el servicio de Node-Red solo se debe escribir en la ventana de comandos: `node-red` y listo se podrá ingresar a la plataforma IoT por medio de la IP <http://127.0.0.1:1880/>.

Para conectar el dispositivo con los servicios en la nube IoT se debe parametrizar en el software código fuente el nombre de la red WiFi, contraseña y la IP asignada por el DHCP.

```

const char* WIFISSID = "Javeriana";
const char* WIFIPASSWORD = "";

#define PUERTOMQTT 1883
//const char* BROKERMQTT = "192.168.1.68";
const char* BROKERMQTT = "10.70.0.139";
//-----

```

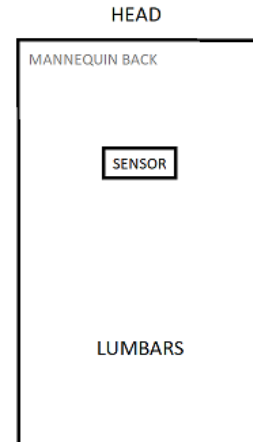
(28). Ejemplo de codificación red WiFi y dirección IP en el software código fuente de CPR-TRAINING.

#### 4. FUNCIONAMIENTO DEL SISTEMA

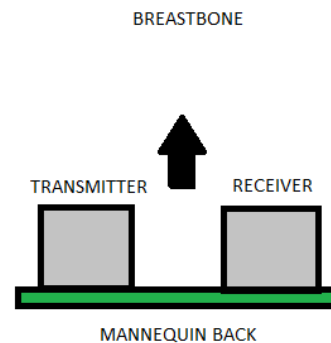
El sistema CPR-TRAINING es un dispositivo capaz de medir las variables principales de la reanimación cardiopulmonar (RCP). Estas variables son a. Profundidad de las compresiones de pecho, b. Velocidad de las compresiones de pecho, c. Retroceso (Expansión del pecho), d. Numero de compresiones de pecho totales y efectivas.

El prototipo se debe conectar a la red eléctrica domestica con toma de voltaje convencional fase-neutro 120VAC. Una vez conectado **no** se debe **encender** el interruptor ON/OFF del prototipo sin antes garantizar que el sensor este ubicado en su posición.

Antes de encender el prototipo se debe garantizar que el sensor ultrasónico se encuentre ubicado en el maniquí de RCP. El sensor es un elemento fijo del sistema y por tanto se debe ubicar en la parte interna de la espalda del maniquí (29) con los elementos de emisión y recepción de la señal apuntando hacia el esternón del maniquí (30).



(29). Vista interna del maniquí con el sensor ubicado en la zona anterior del esternón del maniquí



(30). Vista lateral del sensor con los elementos emisor y receptor apuntando hacia el esternón en la parte posterior del maniquí.

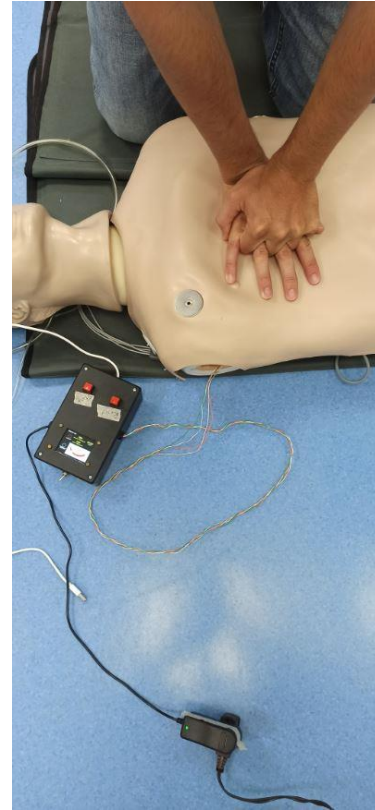
Una vez instalado el sensor en el maniquí se debe garantizar que todas las piezas del maniquí estén en su lugar y se encuentre completamente armado, de esta manera, se puede proceder a encender el sistema mediante el interruptor ON/OFF (31).



(31). Interruptor ON/OFF del dispositivo.

Cuando el sistema se encienda el usuario no debe estar ejerciendo ningún tipo de presión sobre el maniquí, es decir, el maniquí en la parte del esternón solo debe estar ejerciendo presión el peso natural de los elementos de su estructura. Se debe esperar a lo menos 10 segundos después de encendido el dispositivo para garantizar que el sensor se calibre y que el dispositivo se conecte a la red WiFi.

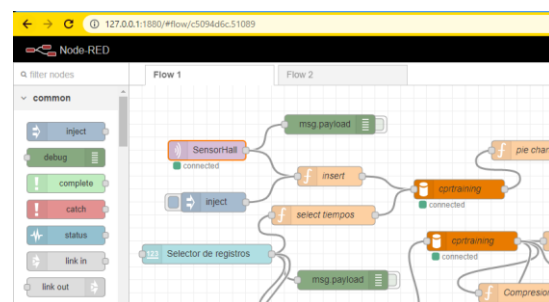
Finalmente se puede hacer uso del sistema presionando el botón *start* y empezando el ejercicio de las compresiones sobre el CPR-TRAINING (32).



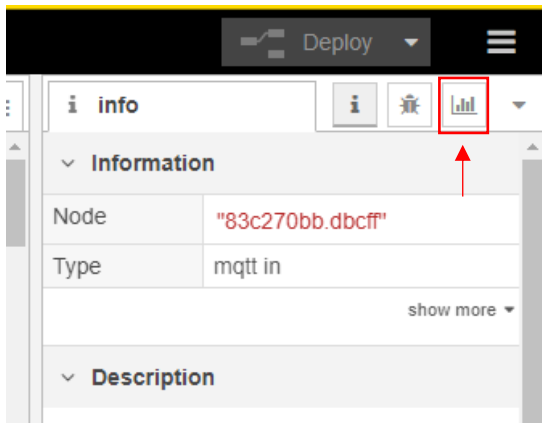
(32). CPR-TRAINING en funcionamiento

El sistema tiene un reloj que al cumplirse un minuto de la sesión después de presionar el botón *start* envía los datos a la plataforma IoT automáticamente.

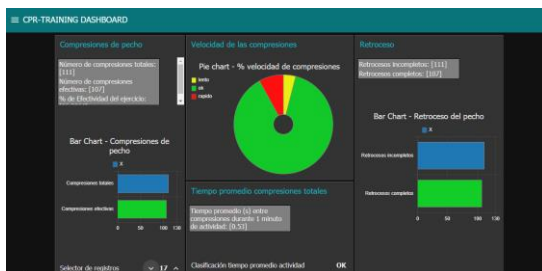
Para consultar los datos se debe ingresar al *localhost* de Node-Red a través de un navegador web por medio de la dirección IP 127.0.0.1;1880 (33). Una vez se ingresa a la plataforma web se pueden visualizar los datos en el dashboard (34)(35).



(33). IDE Node-Red.



(34). Botón para visualizar indicadores en el dashboard en Node-Red.



(35). CPR-TRAINING Dashboard.

## 5. CUIDADOS DEL PROTOTIPO

- No golpear el prototipo ni el sensor
- No sumergir en agua el prototipo ni el sensor
- No conectar a una red eléctrica que sea diferente a la domestica 120VAC
- No estirar los cables del sensor

## 6. LINEAS DE ATENCIÓN

En caso de reparación por daños llamar al +57 315-807-4782 SOPORTE CPR-TRAINING.

Correo:  
soportecprtraining@javerianacali.edu.co

# Bibliografía

- [1] PRESTAN ADULT SERIES 2000 Add-On. <https://www.prestanproducts.com/products/parts-accessories/adult-series-2000-add-on-kit/>, 2021.
- [2] Chiwon Ahn, Juncheol Lee, Jaehoon Oh, Yeongtak Song, Youngjoon Chee, Tae Ho Lim, Hyunggoo Kang, and Hyungoo Shin. Effectiveness of feedback with a smartwatch for high-quality chest compressions during adult cardiac arrest: A randomized controlled simulation study. *PloS one*, 12(4):e0169046, 2017.
- [3] Robert Anderson, Alexandre Sebaldt, Yiqun Lin, and Adam Cheng. Optimal training frequency for acquisition and retention of high-quality cpr skills: A randomized trial. *Resuscitation*, 135:153–161, 2019.
- [4] Arduino, ESP8266 PlatformIO IDE compatible TFT library optimised for the Raspberry Pi Pico (RP2040), STM32, and ESP32 that supports different driver chips. [https://github.com/Bodmer/TFT\\_eSPI](https://github.com/Bodmer/TFT_eSPI), 31 March 2021.
- [5] L Bellows and R Moore. Salud del corazón: tratamiento de la enfermedad cardíaca con dieta, la. *Ficha técnica (Colorado State University Extension). Serie de alimentos y nutrición; n. 9.384*.
- [6] Miriam Bolet Astoviza and María Matilde Socarrás Suárez. Alimentación adecuada para mejorar la salud y evitar enfermedades crónicas. *Revista Cubana de medicina general integral*, 26(2):0–0, 2010.
- [7] María Dolores Lazo Caparros. Nivel de conocimiento y aptitudes de la reanimación cardiopulmonar en trabajadores. *Revista Enfermería del Trabajo*, 7(4):109–116, 2017.
- [8] Dolores Cárdenas Cruz et al. *Análisis de un programa de formación masiva en soporte vital básico para la población general proyecto salvavidas: primera fase*. Number 103-105. Universidad de Granada, 2013.
- [9] Marta Casillas Cabana et al. *La reanimación cardiopulmonar en escolares: estudio observacional de su ejecución*. PhD thesis, Didácticas especiales, 2014. 25-26.
- [10] Mikołaj Chwalisz. thingspeak documentation. *SL: Thingspeak*, 2016.
- [11] IDEAM COLOMBIA. Tiempo y clima. <http://www.ideam.gov.co/web/tiempo-y-clima/clima>, 2020.
- [12] Sensor de distancia de tiempo de vuelo VL6180x vista superior con dimensiones. Disponible en web. <https://www.pololu.com/product/2489>.

- 
- [13] Sensor de distancia infrarrojo SHARP GP2Y0D815. Disponible en web. <http://https://www.pololu.com/product/2466>.
- [14] Saúl Drajer. Maniquí de bajo costo para la reanimación cardiopulmo-nar básica en niños. *Revista Mexicana de Pediatría*, 78(5):210–211, 2011.
- [15] SI Varun Durai, Raj Arjunan, and M Manivannan. Affordable hi-fidelity vr based cpr simulator with haptics feedback. In *2017 IEEE International Symposium on Haptic, Audio and Visual Environments and Games (HAVE)*, pages 1–5. IEEE, 2017.
- [16] 9 CPR Training Errors. <https://www.emssafetyservices.com/2015/03/02/9-cpr-training-errors/>, 2021.
- [17] Acxel Enrique Fernández Galeano. *Estudio de sensores piezoeléctricos en aplicaciones de medición de fuerza*. PhD thesis, 2018.
- [18] Apache Software Foundation. <https://www.apache.org/>, 2021.
- [19] Francis Gohier, Kiran Dellimore, and Cornie Scheffer. Development of a smart backboard system for real-time feedback during cpr chest compression on a soft back support surface. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 346–349. IEEE, 2013.
- [20] J Gómez. Morbimortalidad cardiovascular en el mundo. *Revista colombiana de cardiología*, 19(6):298–9, 2012.
- [21] LIZBETH LUNA GONZÁLEZ et al. El diseño de interfaz gráfica de usuario para publicaciones digitales. 2004.
- [22] Joan Guasch Llobera and Mario Calleja Collado. Monitorización de sensores con arduino utilizando el protocolo mqtt. B.S. thesis, Universitat Politècnica de Catalunya, 2019.
- [23] Raúl Tabarés Gutiérrez. El surgimiento de html5; un nuevo paradigma en los estándares web. *Revista Teknokultura*, 13(1):169–192, 2016.
- [24] Edgar Manríquez Herrejón, Vibraciones Mecánicas, and José Nicolás Ponciano Guzmán. Acelerómetros y vibrómetros.
- [25] Loay F Hussein, Anis Ben Aissa, Islam Abdalla Mohamed, Saed Alruwaili, and Abdalazi Alanzi. Development of a secured vehicle spot detection system using gsm. *International Journal of Interactive Mobile Technologies*, 15(4), 2021.
- [26] iStock RCP en adultos Compresiones de pecho. <https://www.istockphoto.com/es/fotos/mu%C3%B1eco-de-rcp>, 2021 iStock.

- [27] Falguni Jindal, Rishabh Jamar, and Prathamesh Churi. Future and challenges of internet of things. *International Journal of Computer Science & Information Technology (IJCSIT) Vol*, 10, 2018.
- [28] Jahanzeb Khan and Anis Khwaja. *Building secure wireless networks with 802.11*. John Wiley & Sons, 2003.
- [29] Tai-hoon Kim, Carlos Ramos, and Sabah Mohammed. Smart city and iot, 2017.
- [30] Yeram Kwon, Sungwon Lee, Jihoon Jeong, and Wonjoon Kim. Heartisense: a novel approach to enable effective basic life support training without an instructor. In *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*, pages 1699–1704. ACM, 2014.
- [31] Milica Lekić and Gordana Gardašević. Iot sensor integration to node-red platform. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–5. IEEE, 2018.
- [32] Mengchi Liu. Advanced database systems. *Instructor*, 2020.
- [33] Xiangtao Liu, Tianle Zhang, Ning Hu, Peng Zhang, and Yu Zhang. The method of internet of things access and network communication based on mqtt. *Computer Communications*, 153:169–176, 2020.
- [34] Yuanshan Liu, Zitong Huang, Heng Li, Guanghui Zheng, Qin Ling, Wanchun Tang, and Zhengfei Yang. Cpr feedback/prompt device improves the quality of hands-only cpr performed in manikin by laypersons following the 2015 aha guidelines. *The American journal of emergency medicine*, 36(11):1980–1985, 2018.
- [35] Ying Chen Lou, Hengzhou Ding, and Ruben Manuel Velarde. Methods and apparatus for performing exposure estimation using a time-of-flight sensor, May 2 2017. US Patent 9,638,791.
- [36] David Benítez Machado, Caridad Anías Calderón, and Lieter Plasencia Moreno. *PROPUESTA DE ARQUITECTURA PARA INTERNET DE LAS COSAS*. PhD thesis, Tesis de grado, Instituto Superior Politécnico José Antonio Echeverría, 2016. 6 - 10.
- [37] Luca Mainetti, Luigi Patrono, and Antonio Vilei. Evolution of wireless sensor networks towards the internet of things: A survey. In *SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks*, pages 1–6. IEEE, 2011.
- [38] Virginia Martínez Fuentes. Introducción a la plataforma arduino y al sensor ultrasónico hc-sr04: experimentado en una aplicación para medición de distancias. B.S. thesis, 2016.
- [39] Juan Ignacio Mulero Martínez et al. El papel de la realimentación en el control de sistemas dinámicos. 2010.

- [40] Robert W Neumar, Michael Shuster, Clifton W Callaway, Lana M Gent, Dianne L Atkins, Farhan Bhanji, Steven C Brooks, Allan R De Caen, Michael W Donnino, Jose Maria E Ferrer, et al. Part 1: executive summary: 2015 american heart association guidelines update for cardiopulmonary resuscitation and emergency cardiovascular care. *Circulation*, 132(18\_suppl\_2):S315–S367, 2015.
- [41] Katuska Jeanneth Nolivos Mantilla. Estudio de factibilidad para la creación de un centro de entrenamiento filial de la american heart association que dicta cursos de reanimación cardiopulmonar en la ciudad de atuntaqui. B.S. thesis, PUCE, 2016.
- [42] World Health Organization. *Official records of the World Health Organization*. Number 227-228. United Nations, World Health Organization, Interim Commission, 1975.
- [43] Digna Maria Gonzalez Otero. *Feedback systems for the quality of chest compressions during cardiopulmonary resuscitation*. PhD thesis, Universidad del País Vasco-Euskal Herriko Unibertsitatea, 2015. 15-18, 32-46.
- [44] Christopher Plata, Miriam Stolz, Tobias Warnecke, Susanne Steinhauser, Jochen Hinkelbein, Wolfgang A Wetsch, Bernd W Böttiger, and Oliver Spelten. Using a smartphone application (pocketcpr) to determine cpr quality in a bystander cpr scenario a manikin trial. *Resuscitation*, 137:87–93, 2019.
- [45] Google Play Storea pps Metrónomo Beats gratis. [https://play.google.com/store/apps/details?id=com.andymstone.metronome&hl=es\\_CO&gl=US](https://play.google.com/store/apps/details?id=com.andymstone.metronome&hl=es_CO&gl=US), 2021 Google.
- [46] Little Anne QCPR. <https://laerdal.com/us/products/simulation-training/resuscitation-training/little-anne-qcpr/>, 2021 Laerdal Medical.
- [47] Electronic References. <https://sumador.com/products/regulador-ajustable-dc-dc-buck-lm2596>, March 2021.
- [48] Luis Rojas, Andrés Aizman, Juan Pablo Arab, Franco Utili, and Max Andresen MsC. Reanimación cardiopulmonar básica: conocimiento teórico, desempeño práctico y efectividad de las maniobras en médicos generales. *Revista médica de Chile*, 140(1):73–77, 2012.
- [49] Jhon Sebastian Torres Muñoz. Diseño y construcción de una herramienta de apoyo para el estudio de conceptos básicos del sonido mediante realidad virtual inmersiva. 2019.
- [50] Setting up TCP/IP networking. <https://www.ibm.com/docs/en/dop/1.6.0?topic=servers-setting-up-tcpip-networking>, 2021.
- [51] Jorge Valderrama and Ebert Brea. Esp8266: Un microcontrolador para el internet de las cosas. *Universidad Central de Venezuela, Tech. Rep. Accessed: May, 8, 2020*.
- [52] Sensor Ultrasónico HC-SR04 vista superior. Disponible en web. <https://www.naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>.

- 
- [53] Sensor Ultrasónico HY-SRF05 vista superior. Disponible en web. <https://www.ardobot.co/sensor-de-distancia-de-ultrasonido-hc-sr05-hy-srf05.html>.
- [54] XAMPP Official Website. <https://www.apachefriends.org/es/index.html>, 2021.
- [55] Euler Yajahuanca Correa. Revisión crítica: aspectos de la reanimación cardiopulmonar básica del adulto a fortalecer en el enfermero (a) del servicio de emergencia. 2018.
- [56] Tetsuya Yokotani and Yuya Sasaki. Comparison with http and mqtt on required network resources for iot. In *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*, pages 1–6. IEEE, 2016.
- [57] Bernhard Zapletal, Robert Greif, Dominik Stumpf, Franz Josef Nierscher, Sophie Frantal, Moritz Haugk, Kurt Ruetzler, Christoph Schlimp, and Henrik Fischer. Comparing three cpr feedback devices and standard bls in a single rescuer scenario: a randomised simulation study. *Resuscitation*, 85(4):560–566, 2014.
- [58] VA Zhmud, VG Trubin, NO Kondratyev, and KA Kuznetsov. Ultrasonic distance sensor hc-sr04. *Automatics & Software Enginery*. 2017. *N4 (22)*, (7/212):10, 2017.