

**IMPLEMENTACIÓN DE UN SISTEMA DE INTERNET DE LAS
COSAS BASADO EN ESP32 PARA EL MONITOREO REMOTO DE
LA SALA DE COMPRESORES DE ARTYTECMO**



YEISON ANDRÉS ORREGO UCHIMA

Autor

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍA MECÁNICA, MECATRÓNICA
E INDUSTRIAL
INGENIERÍA MECATRÓNICA
PAMPLONA
2022**

**IMPLEMENTACIÓN DE UN SISTEMA DE INTERNET DE LAS
COSAS BASADO EN ESP32 PARA EL MONITOREO REMOTO DE
LA SALA DE COMPRESORES DE ARTYTECMO**

YEISON ANDRÉS ORREGO UCHIMA

Autor

Trabajo de grado para optar por el título de Ingeniero en Mecatrónica

Director

OSCAR J. SUAREZ

**Ingeniero Electrónico
Especialista en Educación en Estrategias de Aprendizaje
Magister en Controles Industriales
Doctor en Ciencias con Énfasis en Control Automático**

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍA MECÁNICA, MECATRÓNICA
E INDUSTRIAL
INGENIERÍA MECATRÓNICA
PAMPLONA
2022**

*A Flora Uchima, que con amor y paciencia me dio todas las
herramientas para no desistir en el camino hacia el cumplimiento
de mis objetivos.*

AGRADECIMIENTOS

Llegado a este punto observo el pasado con profundo agradecimiento a todas esas personas que con sus acciones me motivaron, enseñaron y apoyaron. Quiero enfatizar mi agradecimiento a la Gobernación del Meta por haber creído en mis capacidades y haberme brindado su apoyo a lo largo de mi formación; a mis profesores Oswal Vera, Cesar Peña y Diego Bugallo por haberme formado con sus enseñanzas, en especial al Dr. Oscar J. Suarez por darme orientación y motivación en el desarrollo del proyecto; a Felipe Gamarra por asesorarme continuamente y darme esos consejos que como joven necesité muchas veces; a mis compañeros del programa por haber hecho amenos estos años; a mis compañeros del área de Mantenimiento Industrial en ARTYTECMO por sus periódicas capacitaciones; finalmente a Karen que me vio crecer, pero no se quedó para ver los resultados.

Gracias a todos los que me acompañaron y no alcanzo a mencionar, soy la suma de un sinfín de acontecimientos que compartí con ustedes. Nada de lo que hice hubiera sido posible sin la coincidencia de haberlos encontrado en el proceso, gracias Dios por permitirlo.

CONTENIDO

INTRODUCCIÓN	1
1. DESCRIPCIÓN GENERAL DEL PROYECTO.....	2
1.1. Planteamiento del problema	2
1.2. Justificación	3
1.2.1. Beneficio económico	3
1.2.2. Beneficio de disponibilidad	3
1.2.3. Beneficio de innovación	3
1.3. Delimitación	3
1.3.1. Objetivo general.....	3
1.3.2. Objetivos específicos	3
1.3.3. Acotaciones.....	4
2. FUNDAMENTO TEÓRICO	5
2.1. Marco teórico.....	5
2.1.1. Internet de las Cosas	5
2.1.1.1. Definición	5
2.1.1.2. Arquitectura del Internet de las Cosas	5
2.1.2. Hardware para IoT	7
2.1.2.1. ESP32	7
2.1.2.2. Raspberry Pi.....	8
2.1.2.3. Arduino	8
2.1.3. Protocolos de comunicación IoT	9
2.1.3.1. MQTT	9
2.1.3.2. CoAP.....	9
2.1.3.3. HTTP	10
2.1.4. Desarrollo web.....	10
2.1.4.1. HTML	10
2.1.4.2. CSS	11
2.1.4.3. JavaScript.....	11
2.1.5. Plataformas IoT	11
2.1.5.1. Node RED.....	11
2.1.5.2. ThingSpeak	11
2.1.5.3. Arduino IoT Cloud	12
2.1.6. Aplicaciones de IoT	12
2.1.6.1. Agricultura y criaderos	12
2.1.6.2. Salud y bienestar.....	12
2.1.6.3. Casas y edificios	13
2.1.6.4. Otras aplicaciones.....	13
2.1.7. Sala de compresores	13
2.1.8. Acerca de ARTYTECMO	14
2.2. Estado del arte.....	14
2.2.1. Antecedentes nacionales	14
2.2.2. Antecedentes internacionales.....	15
3. MAQUETACIÓN DEL SISTEMA.....	17
3.1. Selección del modelo de arquitectura	17

3.2. Maquetación de la capa de percepción	18
3.3. Maquetación de la capa de transporte.....	19
3.3.1. Elección de protocolo de comunicación	20
3.3.2. Elección de medio inalámbrico	20
3.3.3. Elección de sistema embebido.....	21
3.4. Maquetación de la capa de procesamiento	22
3.5. Diagrama general del sistema.....	23
4. DESARROLLO FÍSICO	25
4.1. Reconocimiento del entorno	25
4.2. Elección de instrumentación.....	27
4.3. Diseño de circuitos	30
5. DESARROLLO INFORMÁTICO	33
5.1. Bróker MQTT	33
5.1.1. Máquina virtual.....	33
5.1.2. Enlace externo	34
5.1.3. Prueba de comunicación MQTT	34
5.2. Desarrollo de la página web	35
5.2.1. Planeación.....	36
5.2.2. Estructura.....	36
5.2.3. Apariencia.....	38
5.3. Conexión Web - MQTT.....	40
5.4. Código ESP32.....	42
6. RESULTADOS	44
6.1. Implementación de hardware.....	44
6.2. Validación del sistema IoT	47
6.2.1. Prueba de conexión.....	47
6.2.2. Prueba de la instrumentación.....	48
6.2.3. Prueba de visualización de datos	49
6.2.4. Prueba de funcionamiento integral	51
6.3. Guía de usuario	53
CONCLUSIONES	54
TRABAJOS FUTUROS.....	56
REFERENCIAS	57
ANEXOS	60
Anexo A: Apariencias del aplicativo.....	60
Anexo B. Repositorio de códigos del sistema	61
Anexo C: Guía de usuario.....	62

LISTA DE FIGURAS

<i>Figura 1. Arquitectura de tres capas.</i>	<i>6</i>
<i>Figura 2. Arquitectura de cinco capas.</i>	<i>6</i>
<i>Figura 3. Tarjeta ESP32 DevKitC V1.</i>	<i>7</i>
<i>Figura 4. Tarjeta Raspberry Pi 3.</i>	<i>8</i>
<i>Figura 5. Tarjeta Arduino Yun.</i>	<i>8</i>
<i>Figura 6. Elementos de preparación del aire.</i>	<i>13</i>
<i>Figura 7. Esquema general del sistema IoT propuesto.</i>	<i>23</i>
<i>Figura 8. Sala de compresores de ARTYTECMO.</i>	<i>25</i>
<i>Figura 9. Diagrama P&ID de la sala de compresores.</i>	<i>26</i>
<i>Figura 10. Distribución física de la sala de compresores.</i>	<i>27</i>
<i>Figura 11. Conversor análogo 0-5V a 0-3.3V.</i>	<i>30</i>
<i>Figura 12. Diagrama del circuito para sensor de presión.</i>	<i>31</i>
<i>Figura 13. Diagrama del circuito para sensor de humedad.</i>	<i>31</i>
<i>Figura 14. Diagrama del circuito para actuadores.</i>	<i>32</i>
<i>Figura 15. Diagrama del circuito sensor de voltaje.</i>	<i>32</i>
<i>Figura 16. Diagrama del circuito sensor de paro.</i>	<i>32</i>
<i>Figura 17. Página web para pruebas.</i>	<i>35</i>
<i>Figura 18. Diagrama general de la página web.</i>	<i>36</i>
<i>Figura 19. Estructura de archivos de la página web.</i>	<i>37</i>
<i>Figura 20. Ventana de inicio de sesión VL.</i>	<i>38</i>
<i>Figura 21. Ventana de acceso denegado VS.</i>	<i>38</i>
<i>Figura 22. Ventana de Dashboard VL.</i>	<i>39</i>
<i>Figura 23. Ventana de Documentación con menú desplegado VL.</i>	<i>40</i>
<i>Figura 24. Esquema general del código para la ESP32.</i>	<i>42</i>
<i>Figura 25. Gabinete central.</i>	<i>44</i>
<i>Figura 26. Tarjeta para compresor principal.</i>	<i>45</i>
<i>Figura 27. Tarjeta para compresor secundario.</i>	<i>45</i>
<i>Figura 28. Instalación sensor de presión en compresor secundario.</i>	<i>46</i>
<i>Figura 29. Instalación sensor de humedad.</i>	<i>46</i>
<i>Figura 30. Conexión de ESP32 como cliente MQTT lograda.</i>	<i>47</i>
<i>Figura 31. Conexión de página como cliente MQTT lograda.</i>	<i>48</i>
<i>Figura 32. Publicaciones de la ESP32.</i>	<i>49</i>
<i>Figura 33. Datos recibidos en la página web.</i>	<i>50</i>
<i>Figura 34. Visualización de datos en la página web.</i>	<i>50</i>
<i>Figura 35. Paquetes capturados con Wireshark.</i>	<i>52</i>

LISTA DE TABLAS

<i>Tabla 1. Matriz de elección de arquitectura.</i>	<i>18</i>
<i>Tabla 2. Comparación de protocolos de transporte.....</i>	<i>20</i>
<i>Tabla 3. Comparación de medios inalámbricos.....</i>	<i>21</i>
<i>Tabla 4. Comparación de sistemas embebidos.</i>	<i>22</i>
<i>Tabla 5. Parámetros a monitorear y manipular.</i>	<i>28</i>
<i>Tabla 6. Comparación de sensores de humedad.....</i>	<i>28</i>
<i>Tabla 7. Comparación de sensores de presión.....</i>	<i>29</i>
<i>Tabla 8. Temas del sistema.</i>	<i>41</i>
<i>Tabla 9. Matriz de evaluación para estado normal.</i>	<i>51</i>
<i>Tabla 10. Matriz de evaluación para estado especial.....</i>	<i>52</i>

RESUMEN

Este trabajo se enmarca en el desarrollo de una práctica empresarial ejecutada en la empresa de corte textil ARTYTECMO. Esta empresa cuenta con una red neumática fundamental para sus procesos de corte textil y mantenimiento, la mayoría de máquinas usan la fuerza del aire comprimido para efectuar sus movimientos y los técnicos de mantenimiento lo emplean para la limpieza de los equipos. Esta red posee carencias en la visualización y accionamiento, provocando problemas frecuentes por no detectar a tiempo presiones bajas, excesos de humedad en la red y reinicio de compresores en apagones eléctricos.

A fin de mitigar los daños ocasionados por los inconvenientes expuestos se implementa un sistema de Internet de las Cosas de bajo costo aplicado a la sala de suministro de aire comprimido para monitorear de forma remota las variables más importantes. Este documento plasma el paso a paso para lograr dicha implementación.

Primeramente, se trazan los parámetros generales del sistema como la arquitectura y los componentes de cada capa, seleccionando el protocolo de comunicación MQTT, la tarjeta de desarrollo ESP32, la conexión a la red vía wifi, entre otros. Luego se identifica la composición y distribución de la sala de compresores analizando la instrumentación y diseñando los circuitos necesarios para realizar la adquisición de datos asociados a variables tales como voltaje, humedad y presión; también los actuadores que permiten la activación remota de los compresores y válvulas de alivio.

Seguidamente se desarrolla una página web de visualización que le permita al usuario el monitoreo y la activación de la sala de compresores desde cualquier dispositivo conectado a la red local de la empresa. Posteriormente, se efectúa la conexión entre el aplicativo de visualización y el circuito de accionamientos y sensores a través de la tarjeta de desarrollo, este paso completa la implementación de Internet de las Cosas en la sala de compresores. Por último, se realizan las validaciones necesarias para comprobar el correcto funcionamiento del sistema implementado. También se crea una guía de usuario que da las directrices para emplear debidamente la herramienta creada.

Palabras claves:

Internet de las Cosas, mantenimiento, monitoreo, ESP32.

ABSTRACT

This work has as context the development of a corporate internship executed in the textile cutting company ARTYTECMO. This company has a pneumatic network that is essential for its textile cutting and maintenance processes. Most of the machines use the force of compressed air to carry out their movements and maintenance technicians use it to clean the equipment. This network has shortcomings in the visualization and operation, causing frequent problems by not detecting low pressures in time, excessive humidity in the system, and restarting compressors in electrical outages.

To mitigate the damage caused by the above-mentioned drawbacks, a low-cost Internet of Things system applied to the compressed air supply room is implemented to remotely monitor the most important variables. This document shows the step by step to achieve such implementation.

First, the general parameters of the system are outlined, such as the architecture and components of each layer, selecting the MQTT communication protocol, the ESP32 development board, and the network connection via wifi, among others. Then the composition and distribution of the compressor room are identified, analyzing the instrumentation and designing the necessary circuits to perform the acquisition of data associated with variables such as voltage, humidity, and pressure; also, the actuators that allow remote activation of compressors and relief valves.

Next, a visualization page is developed to allow the user to monitor and activate the compressor room from any device connected to the company's local network. Subsequently, the connection between the visualization application and the circuit drives and sensors through the development board is made, this step completes the implementation of the Internet of Things in the compressor room. Finally, the necessary validations are performed to verify the correct operation of the implemented system. A user guide is also created to provide guidelines for the correct use of the tool created.

Keywords:

ESP32, Internet of Things, maintenance, monitoring.

INTRODUCCIÓN

La revolución digital permitió la creación de lazos entre individuos y sociedades con mayor velocidad y eficacia, permitió el tratamiento de grandes cantidades de datos y proporcionó programas y aplicaciones con fines específicos. Por otro lado, la industria se expandió y aparecieron diversos objetos con funciones concretas que nos facilitan la producción automatizada y con esto la vida cotidiana. La forma de ver los procesos lucrativos cambió hacia un enfoque donde se tiene un aspecto físico y uno informático, el primero constituye lo tangible, lo que recolecta valores y modifica variables; el segundo corresponde a una representación digital del primer aspecto, que permite la toma de decisiones de forma más acertada. La integración de estos objetos con la digitalización comprende un nuevo reto que apunta a la interconexión total de personas y cosas en una red, esto es Internet de las Cosas.

La expresión “Internet de las Cosas” proviene del término en inglés *Internet of Things (IoT)* y se emplea para describir una visión tecnológica y novedosa de la forma como se relacionan los objetos y personas en su día a día. En la literatura se encuentran definiciones como la de Miraz, que define el IoT como una interconexión de “cosas” que se utilizan para comunicar y censar toda la información que existe en el mundo real; esto puede verse como un compuesto conectado a toda persona o cosa, en todo momento o lugar y todo servicio o red [1]. A pesar de que el IoT se orienta a conectar todos los objetos, en la etapa actual se aprecian divisiones relacionadas con la aplicación, siendo una muy común el Internet Industrial de las Cosas (IIoT), cuyo enfoque es la conexión de cualquier proceso o subproceso industrial a internet, obteniendo beneficios como la reducción de costos mediante el monitoreo de las variables principales.

En busca de contribuir a la expansión del internet de las cosas mejorando procesos industriales, este trabajo implementa un sistema IoT basado en la tarjeta de desarrollo ESP32 para el monitoreo remoto de la sala de compresores de ARTYTECMO, empresa dedicada al corte textil. Las variables manipuladas son voltaje, humedad y presión. El sistema permite a los auxiliares de mantenimiento accionar de forma remota los compresores y válvulas de alivio, además de verificar estados como encendido, apagado y energizado en los mismos. Esta implementación es novedosa en la empresa y los beneficios demostrados plantan precedentes para futuras ampliaciones de la red IoT.

Para plasmar de forma ordenada el desarrollo del sistema, este documento se organiza en seis capítulos. El Capítulo 1 describe de forma general el proyecto, allí se define el problema, se plantean objetivos y se limita el alcance. El Capítulo 2 fundamenta teóricamente la investigación y define las tecnologías que se emplean. El Capítulo 3 describe la estructura general del proyecto y muestra la elección de los componentes de cada capa. El Capítulo 4 desarrolla la implementación física, mostrando los circuitos necesarios para la recolección de datos y modificación de estados. El Capítulo 5 contiene el desarrollo informático que aborda la creación del servidor local, la página web y la programación de la tarjeta ESP32. El último capítulo muestra los resultados obtenidos, se realizan las verificaciones necesarias y se incluye una guía de usuario. Finalmente, se concluye acerca del trabajo realizado y se plantean trabajos futuros.

1. DESCRIPCIÓN GENERAL DEL PROYECTO

1.1. Planteamiento del problema

El aire comprimido presenta ventajas importantes, puede proporcionar movimiento controlado a un costo muy bajo, sus sistemas son de fácil diseño e implementación y proporciona factores de seguridad elevados por las bajas presiones que emplea cotidianamente a nivel industrial [2]. Lo anterior permite que sea utilizada con frecuencia en industrias de todo tipo, incluyendo la de corte textil; en esta industria es común el empleo de actuadores lineales neumáticos para el uso de elevadores, alternadores de bandas transportadoras y secuencias de movimiento rutinarias en máquinas específicas como cortadoras automáticas de cuchilla lineal asistidas por computador y cortadoras manuales con cuchilla tipo disco para la creación de sesgos [3].

Por otro lado, el aire comprimido es utilizado con frecuencia en los mantenimientos preventivos industriales, elimina de una forma rápida y eficaz material particulado que se aloja en las superficies de los equipos por operación normal; el aire a presión actúa de forma inmediata y no deja residuos, convirtiéndose en un método indispensable de limpieza para industrias con polución elevada como la de confección textil.

ARTYTECMO maneja aire comprimido para la operación y mantenimiento de sus equipos, dispone para su generación una sala de suministro clásica equipada con dos compresores de tornillo, un tanque acumulador y un sistema de preparación para evitar contaminación y humedad excesiva en el aire. Este sistema de preparación no indica el porcentaje de humedad, se debe conectar de forma manual y carece de un sistema de alarmas. Estas deficiencias conllevan daños graves al efectuar limpiezas de tarjetas electrónicas con aire que deja residuos líquidos; esto puede generar cortocircuito o deterioro en los componentes.

La sala de compresores tampoco alerta a los auxiliares de mantenimiento cuando el compresor principal y el de respaldo no encienden o no entregan la presión suficiente. En este escenario el depósito pierde su presión rápidamente por el trabajo normal de la maquinaria si lanzar alerta alguna. Esto es extremadamente peligroso porque las cortadoras automáticas sin presión aterrizan de forma brusca el afilador que contiene una cuchilla que se quiebra de forma violenta por el impacto, generando riesgo para el operario, daño en la producción y afectaciones en la máquina.

En el intento de contribuir con una solución a las dos problemáticas expuestas aparece la siguiente cuestión ¿Es posible mitigar los daños ocasionados por humedad y errores de activación de compresores implementando un sistema de monitoreo y accionamiento remoto basado en Internet de las Cosas en la sala de suministro de aire comprimido de la empresa ARTYTECMO?

1.2. Justificación

Concluir este proyecto con la implementación de Internet de las Cosas (IoT) en la sala de suministro de aire comprimido de ARTYTECMO representa para la empresa y sus colaboradores tres beneficios claros a nivel económico, de disponibilidad del equipo de mantenimiento y de innovación.

1.2.1. Beneficio económico

El sistema propuesto evitará daños en la maquinaria de la empresa al garantizar aire comprimido con rangos de humedad permitidos, disminuyendo con esto los costos asociados a reparaciones o cambios de elementos como motores eléctricos o tarjetas electrónicas, que suelen ser los componentes de las máquinas más valorizados en el mercado.

1.2.2. Beneficio de disponibilidad

La sala de suministro se encuentra ubicada en el extremo opuesto a la base del personal de mantenimiento, esto hace que su desplazamiento hasta allí para poner en marcha los sistemas, consuma tiempos en el que se pueden adelantar mantenimientos preventivos de otros equipos, suministrando una mayor confiabilidad a la empresa. El sistema propuesto permitirá al personal monitorear y accionar desde su base la sala, creando disponibilidad para otras tareas.

1.2.3. Beneficio de innovación

La implementación de IoT en la sala de suministro establece un precedente que permitirá a la empresa observar los beneficios que trae la industria 4.0, incentivando la implementación en más operaciones del proceso productivo, reduciendo costos y optimizando los recursos.

1.3. Delimitación

1.3.1. Objetivo general

Implementar un sistema de Internet de las Cosas basado en ESP32 para monitorear la sala de compresores de ARTYTECMO.

1.3.2. Objetivos específicos

- Acondicionar la sala de compresores con la instrumentación necesaria para monitorear las variables del proceso, desarrollando los circuitos de acondicionamiento requeridos para el acople a la tarjeta ESP32.

- Crear una página de visualización albergada en un servidor local para acceder a la información recopilada de los equipos de la sala de compresores.
- Establecer la conexión entre el sistema embebido y el aplicativo mediante el protocolo MQTT.
- Validar el correcto funcionamiento del sistema implementado, evaluando condiciones normales y especiales.
- Generar una guía de usuario para operar de forma adecuada el sistema desarrollado.

1.3.3. Acotaciones

Algunos detalles del desarrollo de esta aplicación se verán censurados por políticas de privacidad de la empresa. La limitación de recursos económicos no permite la vinculación de una base de datos robusta para almacenar los datos capturados.

2. FUNDAMENTO TEÓRICO

2.1. Marco teórico

2.1.1. Internet de las Cosas

El pilar fundamental del presente desarrollo es el Internet de las Cosas, es por esto que se deben plantar bases teóricas fuertes que permitan el correcto desarrollo de las etapas posteriores. En este apartado se presentan los conceptos más importantes para entender los sistemas IoT y percibir su alcance.

2.1.1.1. Definición

La expresión “Internet de las Cosas” se emplea para describir una visión tecnológica y novedosa de la forma como se relacionan los objetos y personas en su día a día. En la literatura son variadas las definiciones que se encuentran, pero todas van encaminadas hacia la misma dirección, difiriendo en pequeños detalles. Teniendo así definiciones como la de Miraz, que define el IoT como una interconexión de “cosas” que se utilizan para comunicar y censar toda la información que existe en el mundo real; esto puede verse como un compuesto conectado a toda persona o cosa, en todo momento o lugar y todo servicio o red [4]. También hay definiciones más delimitadas como la de Xu y He, que otorgan propiedades específicas al IoT, como conectividad global, dinamismo, protocolos de configuración interpretables, inteligencia, atributos físicos únicos e integración perfecta [5].

En relación con la propiedad otorgada al IoT de conectividad global, existe un gran debate, porque es posible encontrar referencias importantes como la ITU (*International Telecommunication Union*) que se mantienen en una idea más clásica, defendiendo que el Internet de las Cosas es aquel que abarca un conjunto de tecnologías interdisciplinarias de conectividad inalámbrica, sensórica e identificación por radiofrecuencia [6]. Dando con lo anterior la categoría de sistemas IoT a aquellos que se desarrollan únicamente para redes locales, que llegan a ser más seguras que las expuestas en internet.

2.1.1.2. Arquitectura del Internet de las Cosas

Al ser el IoT una tecnología relativamente nueva y en constante desarrollo, no ha sido posible determinar una arquitectura que represente un consenso unificado. Diferentes autores han realizado propuestas, pero es difícil rastrear el punto exacto donde se mencionan por primera vez. Sin embargo, al respecto se han elaborado revisiones bibliográficas importantes, entre ellas se destaca la desarrollada por Wu y compañía en [7], allí se indican las dos arquitecturas de mayor peso en la actualidad, las de tres y cinco capas.

Como se muestra en la Figura 1 la arquitectura de tres capas se compone en orden ascendente de: la capa de percepción, donde se ubican los sensores y actuadores del sistema usados para recopilar y modificar la información de los objetos; la capa de red, que funciona

como el cerebro del IoT porque sus funciones principales son procesar y transmitir la información recopilada en la capa anterior; la capa de aplicación, donde se proporcionan los servicios de interacción necesarios para vincular los procesos sociales e industriales.



*Figura 1. Arquitectura de tres capas.
Fuente: Adaptación de [7].*

Como se observa en la Figura 2 la arquitectura de cinco capas se diferencia de la anterior porque la capa de red se divide en capa de transporte y capa de procesamiento, además se agrega una capa superior llamada capa de negocios. La capa de percepción se mantiene con las mismas funciones y elementos mencionados para la arquitectura anterior. La capa de transporte se encarga de llevar los datos digitalizados en la capa anterior a la siguiente capa para procesamiento, para esto emplea diversas tecnologías como el wifi, redes 4G, redes de áreas local (LAN), *Zigbee*, entre otras. La capa de procesamiento se encarga de administrar y tratar los datos; se emplea computación en la nube, bases de datos, computación ubicua y procesamiento inteligente. La siguiente capa es la de aplicación y se mantiene con la misma designación dada en la arquitectura anterior. Finalmente, la capa de negocios se dedica a potenciar los beneficios tanto económicos como sociales que pueden obtenerse del IoT, esto incluye la seguridad, investigación y lanzamiento de nuevos proyectos, enfoque y aplicaciones.



*Figura 2. Arquitectura de cinco capas.
Fuente: Adaptación de [7].*

La arquitectura de cinco capas es la más empleada en sistemas IoT desarrollados para fines académicos o de pequeña envergadura; incluso para estos fines la capa superior referente a los negocios pasa a un segundo plano, como se demuestra en [8] y [9]. A continuación, se detallan las capas de una arquitectura muy empleada a nivel investigativo, la de cinco capas modificada.

2.1.2. Hardware para IoT

La capa de percepción debe contar con sensores y actuadores para recopilar y modificar la información, por lo regular esta instrumentación es dirigida por una tarjeta de control maestra, que cuenta con los protocolos necesarios para ejecutar el transporte de los datos hacia el procesamiento, que en algunos casos también es realizado por el mismo sistema embebido. Es por esto que se debe contar con una tarjeta potente y adecuada para cada estudio. Como las aplicaciones a nivel investigativo son de alcance limitado, en estas se emplean con frecuencia sistemas embebidos porque representan una inversión de menor costo sin afectar el desempeño de las tareas que se ejecutan. A continuación, se mencionan los más empleados.

2.1.2.1. ESP32

Son un conjunto de tarjetas de desarrollo, módulos y microcontroladores muy empleados en aplicaciones IoT por su versatilidad y costo. Son varias las opciones disponibles en esta familia y se diferencian en capacidad de almacenamiento, cantidad de pines para conexión de periféricos, protocolos de comunicación, memoria flash y tamaño de encapsulado. A modo de ejemplo, la tarjeta ESP32 DevKitC mostrada en la Figura 3 posee módulos para comunicación UART y SPI, Bluetooth, wifi, 34 pines de propósito general para la conexión de periféricos analógicos o digitales, posibilidad de encriptación en memoria flash, entre otros. Se puede programar en varios entornos de desarrollo libres y existe variedad de librerías que facilitan el manejo de dispositivos especiales [10].



Figura 3. Tarjeta ESP32 DevKitC V1.

Fuente: <https://www.espressif.com/en/products/devkits>

2.1.2.2. Raspberry Pi

Raspberry es un computador compacto, ligero y económico que posee sistema operativo propio y puede interactuar con dispositivos de hardware como servomotores, dispositivos de visualización y sensores. Existen varios modelos con características diferentes, el más empleado para aplicaciones IoT es la Raspberry Pi 3 mostrada en la Figura 4, se puede configurar como servidor o base de datos, cuenta con 40 pines para conexión de periféricos digitales, posee Bluetooth, wifi, puerto Ethernet RJ45, ranura para memoria micro SD, entre otros [11]. Debido a su pequeño tamaño, es óptima para instalación en proyectos o tableros de control.



Figura 4. Tarjeta Raspberry Pi 3.

Fuente: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>

2.1.2.3. Arduino

Arduino es una plataforma para el desarrollo de prototipos basada en hardware y software libre que pone a disponibilidad del público en general el mundo de la electrónica. Su entorno de desarrollo es muy amigable y de fácil mecanización, lo que permite el aprendizaje rápido, así no se tengan fundamentos de programación [11]. Las placas Arduino se pueden modificar libremente, esto ha propiciado un auge en el desarrollo de proyectos de escuela con enfoque en la robótica e Internet de las Cosas.

Se encuentran disponibles variados modelos, entre ellos Arduino Uno, Arduino Mega, Arduino Yun, Arduino ADK y Arduino Pro Mini. Se destaca el Arduino Yun mostrado en la Figura 5 para aplicaciones IoT porque permite conexión mediante wifi o cable de red, cuenta con canales PWM, pines digitales y analógicos, memoria EEPROM de 1KB, memoria flash de 32KB, entre otros. Además, se puede vincular a Arduino IoT Cloud de forma sencilla, constituyendo una herramienta IoT de fácil implementación.



Figura 5. Tarjeta Arduino Yun.

Fuente: <https://docs.arduino.cc/retired/boards/arduino-yun>

2.1.3. Protocolos de comunicación IoT

Denominado así aquel protocolo que permite el intercambio bidireccional de datos para aplicaciones IoT, normalmente entre el aplicativo web o de visualización y la tarjeta de desarrollo y/o la instrumentación. Para cubrir esta necesidad se presentan tres opciones bastante empleadas a nivel industrial e investigativo, estos son: CoAP, MQTT y HTTP.

2.1.3.1. MQTT

Quizá el protocolo más empleado en los sistemas IoT es el MQTT (*Message Queuing Telemetry Transport*) porque es ligero y consume poco ancho de banda, gracias a la reducción del encabezado de cada mensaje. Este protocolo fue especialmente diseñado para la recopilación de información de sensores consumiendo pocos recursos. Los datos recopilados son sensibles y se debe verificar la seguridad en cualquier protocolo, los desarrolladores de MQTT partieron de este hecho para incluir cifrado SSL (*Secure Sockets Layer*), sin embargo, cuando los mensajes se cifran el coste informático-computacional aumenta y deja de ser simple, perdiendo MQTT su mayor ventaja.

La arquitectura de este protocolo sigue una topología en estrella, donde el nodo principal recibe el nombre de bróker y los elementos que se someten a él son clientes. Los clientes no interactúan entre sí, es el bróker el que direcciona los mensajes mediante el concepto de publicación/suscripción. La publicación puede ser realizada por los clientes en cualquier momento, pero debe estar dirigido a un tema (*topic*); el cliente solo debe asegurarse de que el mensaje le llegue al bróker. La suscripción hace que cobre sentido la publicación, cada cliente envía al bróker el tema al que desea suscribirse, al inicio de la comunicación, para recibir la información relacionada con el tema [12].

2.1.3.2. CoAP

Este es un protocolo de transferencia de documentos diseñado para la transmisión en dispositivos de capacidad restringida como microcontroladores, por esto es que los paquetes son más pequeños que en otros protocolos. Se caracteriza por permitir el intercambio de mensajes de forma asíncrona y requerir de manera obligatoria un servidor para cada comunicación con el cliente (comunicación M2M) lo que puede convertir una tarea con demasiados clientes en una ardua labor, sin embargo, se pueden programar mensajes para envío automático ante cualquier variación de la variable, compensando de esta manera la sobrecarga de estar cambiando un mismo servidor entre varios clientes. Trabaja con la modalidad requerimientos y respuestas, similar a HTTP, de hecho, no es la única característica que comparten, también lo hacen con las capacidades de proxy y la forma de administrar la caché; esta y otras similitudes llevan a la comunidad a pensar que CoAP es la versión mini de HTTP. A pesar de lo anterior hay diferencias significativas, una de ellas es que CoAP trabaja sobre UDP (Protocolo de Datagramas de Usuario) en lugar de TCP (Protocolo de Control de Transmisión) como lo hace HTTP.

Este protocolo maneja dos calidades de servicio que se pueden definir tanto para los requerimientos como las respuestas, se puede elegir que el mensaje sea confirmado o no, en

el primer caso el cliente informa sobre la recepción del mensaje y en el segundo no sucede. Para finalizar este breve resumen sobre protocolo se mencionan los métodos que se pueden utilizar, estos son GET, POST, PUT y DELETE, el primero permite la obtención de la información que contenga el requerimiento, si es hallado se obtiene un código de confirmación; el siguiente es utilizado para decirle al servidor que cree un nuevo recurso; el tercero se utiliza como actualizador de estados de los recursos existentes; el último elimina los recursos seleccionados y libera espacio en el servidor para nuevas interacciones [13].

2.1.3.3. HTTP

Este importante protocolo es el empleado por la *World Wide Web*, es decir, está presente en la mayoría de búsquedas a través de internet. Funciona a través de una conexión cliente – servidor y por su tiempo en desarrollo posee diversas librerías que se emplean para aumentar el alcance de las aplicaciones. Aplicado a sistemas IoT este protocolo no resulta tan conveniente por la cantidad de ancho de banda que requiere, ya que los clientes son los que siempre deben iniciar el contacto, haciendo que los actuadores cuestionen todo el tiempo al servidor por alguna variación en las variables, sobrecargando el sistema con tráfico innecesario. No obstante, hay aplicaciones de gran envergadura en los que es útil utilizar este protocolo en compañía de herramientas potenciadoras porque HTTP proporciona transporte de datos, pero no otorga formatos de visualización, permitiendo contener en sus mensajes diversos lenguajes de programación como JavaScript, JavaScript Object Notation, XML, entre otros [14].

Una ventaja asociada a este protocolo es la seguridad que otorga cuando se emplea la Seguridad en la Capa de Transporte (TLS) porque permite un encriptado de alto nivel que no se logra con otros protocolos más livianos como los mencionados anteriormente. Una desventaja es el tamaño de los paquetes que envía, el encabezado de estos contiene demasiada información para ser implementado en microcontroladores de bajo nivel computacional.

2.1.4. Desarrollo web

En la arquitectura de cinco capas, la etapa de procesamiento incluye por lo general métodos de visualización e interacción, donde el usuario puede controlar y monitorear los objetos. Al estar en una red local o global, la forma más sencilla, amigable y utilizada es por medio de páginas web que pueden ser visitadas desde computadores y celulares. De ahí que el dominio en el desarrollo web se vuelva un punto importante para la implementación de Internet de las Cosas. Próximamente, se mencionan algunos conceptos clave para el desarrollo web.

2.1.4.1. HTML

El término Lenguaje de Etiquetas de Hipertexto proviene de la expresión en inglés *HyperText Markup Language* y es el concepto más básico de la web, ya que ordena y da significancia al contenido de la página. Este lenguaje emplea etiquetas para marcar el texto, logrando diferenciar entre encabezados, títulos, imágenes, entre otros [15].

2.1.4.2. CSS

Cascading Style Sheets (CSS), en español hojas de estilo en cascada, apareció como solución a la personalización compleja que poseía el desarrollo único en HTML, donde se debía modificar línea a línea cualquier cambio en la apariencia. Este lenguaje hace más fácil el trabajo de los desarrolladores del *front-end*, porque no deben modificar el esqueleto de la página web (HTML) para realizar cambios dinámicos en la apariencia [16]. Es importante aclarar que CSS no constituye un lenguaje de programación como tal, es de hecho un documento que interactúa con las páginas web basándose en otros lenguajes.

2.1.4.3. JavaScript

JavaScript es un lenguaje de programación que puede correr sin necesidad de un compilador en las páginas web, lo que resulta en gran ventaja para el rendimiento. Con este lenguaje se pueden crear animaciones que interactúen con las acciones del usuario, dando más dinamismo a las páginas web [17]. Es utilizado normalmente del lado del cliente y se escribe con una sintaxis similar a la de C++ y Java lo que facilita a los programadores su adaptación. También posee una amplia gama de librerías que permiten su interacción del lado del servidor y que hacen más rápida la programación.

2.1.5. Plataformas IoT

Como el desarrollo web implica gran demanda de conocimiento, se han desarrollado plataformas que facilitan de forma drástica la implementación de sistemas IoT, la mayoría se enfocan en la visualización y control de los datos a través de paneles interactivos y amigables. El desarrollador solo debe constituir la primera capa del sistema y vincularla a la plataforma preestablecida. Algunas incluso ofrecen servicios de tratamiento y análisis de datos. A continuación, se listan algunas de las más sencillas y potentes.

2.1.5.1. Node RED

Node RED es una herramienta que permite conectar el mundo real con servicios en línea de forma sencilla, basándose en arquitecturas del Internet de las Cosas. Se programa de forma gráfica empleando bloques que se van conectando mediante hilos, esto compone redes que le dan el nombre a la plataforma. La plataforma se construyó basándose en Node Js que es una adaptación de JavaScript que funciona del lado del servidor, su modo bloqueante y de eventos la hace ideal para trabajar con objetos. El almacenamiento de las redes o flujos creados se logra empleando JSON (JavaScript Object Notation). Todo lo anterior permite que Node RED se pueda ejecutar en tarjetas como la Raspberry Pi [18].

2.1.5.2. ThingSpeak

Esta es una plataforma libre desarrollada por MathWorks, la misma empresa que desarrolló Matlab y Simulink, grandes herramientas empleadas en la ingeniería. ThingSpeak permite almacenar datos en la nube y crear aplicativos para visualización y control de

variables relacionadas con objetos. Además, ofrece servicios especiales como la posibilidad de agregar complementos propios para extender las aplicaciones, soportando HTML, CSS y JavaScript para esto. Se puede vincular de forma sencilla a tarjetas de desarrollo económicas como Arduino, Raspberry Pi, IOBridge y ESP32 [19].

2.1.5.3. Arduino IoT Cloud

Esta es la plataforma fue desarrollada por Arduino para integrar sus placas con el Internet de las Cosas, aunque se pueden vincular otras tarjetas. Siguiendo la filosofía de Arduino, esta plataforma busca que los desarrolladores se enfoquen más en la capa de negocios y aplicación que en las inferiores que refieren al fundamento tecnológico. De la misma forma que las anteriores, permite la creación de paneles de visualización y control muy agradables sin tener que pasar por la engorrosa codificación de una página web tradicional [20].

2.1.6. Aplicaciones de IoT

La capa de aplicación aparece en las dos arquitecturas principales porque es de suma importancia, es la que tiene contacto directamente con los usuarios finales. Entiéndase por contacto la interacción accesible. Las demás capas son transcendentales y también entran en contacto con el usuario, pero de forma superficial. En otras palabras, el usuario no debe conocer a fondo el funcionamiento de las demás capas, solo se interesa por la repercusión que tengan estas en su vida. Por lo anterior es importante mencionar algunas de las aplicaciones más conocidas para visualizar el impacto que puede tener la implementación de sistemas IoT.

2.1.6.1. Agricultura y criaderos

La seguridad alimentaria en los últimos años se ha convertido en una preocupación de envergadura mundial. En la actualidad se requiere producir más alimentos y de mejor calidad, empleando menos recursos, optimización en su máxima expresión. La problemática ha llevado a la comunidad científica a buscar nuevos métodos, refugiándose en tecnologías emergentes y de gran impacto como el Internet de las Cosas. Son muchos los trabajos que aplican IoT a diferentes dominios de la agricultura y la cría de animales de consumo, revisiones bibliográficas como la de Farooq y compañía en [21] detallan estos avances mostrando los beneficios obtenidos y retos por afrontar.

2.1.6.2. Salud y bienestar

Por sus características, los sistemas IoT se han aplicado en el ámbito de la sanidad para mejorar y personalizar los servicios que se prestan. Se fantasea con implantar chips en cada persona para llevar registro de sus situaciones médicas permitiendo mejorar los tiempos de atención y evitar errores de diagnóstico. Mientras ese avance llega, se han logrado progresos en dominios como la monitorización de signos vitales, niveles de hormonas y enzimas, el tratamiento de algunas enfermedades y aspectos logísticos en centros de atención médica. Selvaraj ha recopilado algunos de estos avances en su revisión bibliográfica [22].

2.1.6.3. Casas y edificios

La aplicación de sistemas IoT a casas y edificios es tal vez una de las aplicaciones más conocidas, se ha trabajado desde hace tiempo bajo el nombre de domótica. Los beneficios de incorporar en los hogares sensores y actuadores saltan a la vista, se previenen accidentes relacionados con incendios y fugas de gas, se puede optimizar el uso de la energía y se otorga confort al poder controlar todos los electrodomésticos, luces, puertas y objetos en general desde la palma de la mano o incluso por comandos de voz. Estos sistemas aún tienen un inconveniente significativo y es la seguridad relacionada con el acceso, la tecnología empleada con frecuencia es limitada y no permite la encriptación de los datos que se mueven en la red [23].

2.1.6.4. Otras aplicaciones

Las aplicaciones del Internet de las Cosas son muy extensas porque el concepto mismo busca abarcar toda la realidad conocida, por ende, no es posible citar todas las aplicaciones a detalle. Se considera que las expuestas son las de mayor relevancia para la cotidianidad. Otras aplicaciones más específicas son mencionadas por Khaled en [24].

2.1.7. Sala de compresores

También llamado cuarto de generación de aire comprimido, hace referencia al lugar donde se ubican todos los elementos necesarios para generar y filtrar el aire comprimido que se empleará en diversos procesos. Normalmente, estas salas se componen como en la Figura 6 y se pueden encontrar compresores, acumuladores, depósitos de condensados, secadores, unidades de mantenimiento y dispositivos para la lectura de variables en los equipos. De los anteriores el más importante es el compresor porque es el elemento donde se origina el fluido que da funciones a los demás componentes; de este son varios los tipos existentes, el compresor de pistón, de tornillo, centrífugo, de paletas, de desplazamiento, entre otros.

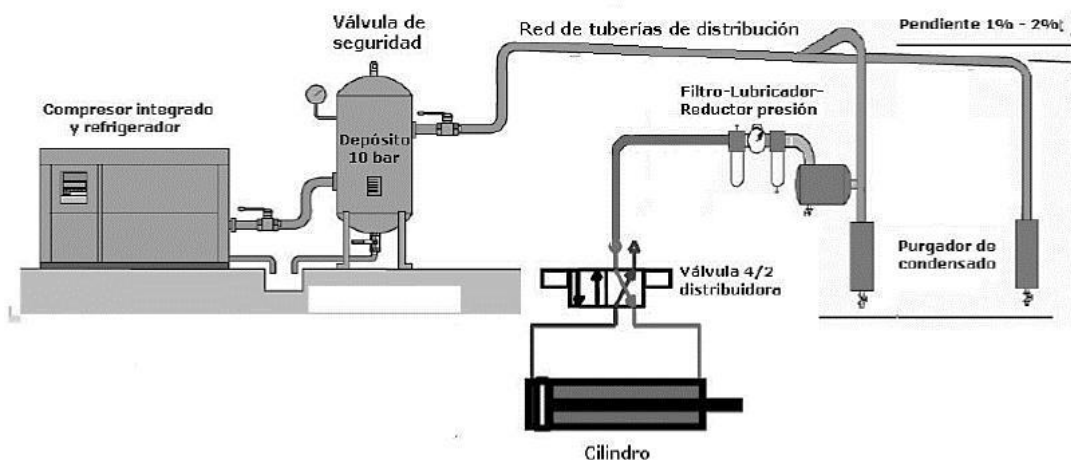


Figura 6. Elementos de preparación del aire.

Fuente: Adaptación de [2].

Profundizando en el compresor de tornillo porque es el tipo que se tiene en la planta de ARTYTECMO, se sabe que es muy empleado en industrias que no disponen de un gran terreno porque se puede ubicar en cualquier lugar de la planta sin causar molestias al personal por ruido emitido. Este compresor, como explica Fernández en [25] tiene un acabado de gran precisión en la forma de dos tornillos helicoidales que engranan para impulsar el aire de un extremo a otro. La hermeticidad se asegura en las cámaras de trabajo con perfiles cicloidales, mejorando el rendimiento mecánico y la dinámica del rotor. El funcionamiento normalmente se divide en tres etapas, la aspiración, el desplazamiento, la compresión y el escape. En la primera etapa se llena progresivamente la cámara de trabajo para que en la segunda este se desplace a volumen constante hasta la entrada de los tornillos; en la tercera etapa el tornillo conductor empieza a desplazar el fluido reduciendo su volumen y aumentando su presión hasta la cuarta etapa donde ocurre el escape.

2.1.8. Acerca de ARTYTECMO

Arte y Tecnología de Moda, comúnmente llamada ARTYTECMO, es una sociedad por acciones simplificadas constituida en el año 2014 como una subdivisión de la empresa Línea Directa para lograr una mejor distribución en sus tareas. Línea Directa es una empresa que comercializa ropa por catálogo en Colombia y Perú, bajo marcas reconocidas localmente como Pacifika, Carmel, Loguin y Real Human. Las prendas que comercializan llevan un proceso de Calidad, Corte, Confección, Desgaste, Estampado y nuevamente calidad. Con el crecimiento de la empresa fue necesaria la entrada de ARTYTECMO para tomar los procesos relacionados al corte. Hoy en día esta empresa posee el corte textil más grande de Colombia, contando con cortadoras automáticas CNC, extendedoras automáticas, cortadoras de sesgos y de extremos; maquinaria que emplea neumática, electrónica y control.

2.2. Estado del arte

Desde hace más de una década, el Internet de las Cosas ha estimulado el interés de investigadores alrededor del mundo que buscan desarrollar nuevos métodos, combinaciones, tecnologías y aplicaciones. Para sentar buenas bases en esta investigación es importante hacer una revisión sobre los avances que se han tenido en la aplicación de IoT a sistemas como el que aquí se trata, monitoreo y control remoto de una sala de compresores o que en su defecto empleen la misma instrumentación o tecnología. También son útiles las investigaciones referentes a la calidad del aire comprimido y su eficiencia energética.

2.2.1. Antecedentes nacionales

A nivel nacional destacan trabajos como el realizado por Ríos en [26], allí se plantea un prototipo para el monitoreo de condiciones ambientales (especialmente el aire) a bajo costo basado en tecnologías IoT. En la capa de percepción emplea sensores especiales que requieren etapas de acondicionamiento, están acoplados a un Arduino Nano que se comunica mediante un módulo LoRa a una interfaz desarrollada en la plataforma IoT Ubidots. Este trabajo es referente porque indica el paso a paso para crear sistemas IoT de baja complejidad y costo, pero con alto impacto en la sociedad.

Otro análisis importante es realizado por Efro en [27]. En este trabajo se evalúa la eficiencia energética del sistema de aire comprimido, que es el componente principal de la planta de producción de oxígeno medicinal de la clínica privada de alta complejidad de Barranquilla. Se realiza una toma de datos de forma clásica empleando los instrumentos que dispone la planta para ello, luego se realiza el análisis bajo los criterios de la Norma ISO 50001 para finalmente realizar las modificaciones respectivas que concluyen con un aumento en la eficiencia energética y una reducción en los costos de producción. Este trabajo contiene una directriz para la evaluación de eficiencias que puede ser implementada en aplicaciones IoT.

Trabajos de mejora en el sistema de aire comprimido de empresas textiles también son referentes para el presente desarrollo, es por esto que el trabajo de Martínez en [28] se presenta; él realiza el análisis necesario para demostrar la obsolescencia del compresor de la red neumática, empleando un analizador de redes Kaeser. Encuentra la solución reemplazando el compresor principal por uno con variador de velocidad que represente una disminución del consumo energético en el proceso. También realiza un análisis económico donde demuestra el tiempo de retorno que tendrá la inversión en el nuevo equipo. Este trabajo demuestra que algunas mejoras requieren cambios desde la base.

Adicionalmente, se puede presentar el trabajo de Rodríguez y compañía en [29], ellos desarrollan un sistema de internet de las cosas que permite monitorear y controlar una sala de compresores. Aplican control a un regulador de presión y monitorean todos los transductores de la planta, empleando una Arduino Mega y una plataforma de IoT gratuita (ThingSpeak). También desarrollan un aplicativo web para trasladar el monitoreo a un dispositivo móvil. De este trabajo se destaca el bajo presupuesto que emplearon, demuestran que no es necesaria una inversión fuerte para un desarrollo robusto.

Para finalizar, en el contexto colombiano aparece el trabajo desarrollado por Londoño en [30], allí plantea un sistema para el monitoreo remoto de fallas y alertas tempranas en el sistema de aire comprimido de la compañía colombiana de galletas Noel S.A.S. Las variables intervenidas son la presión, la humedad relativa, la temperatura del agua de los compresores y el flujo del aire; estos datos se procesan en un PLC S71200 de marca SIEMENS. Londoño concluye su trabajo con una reducción del 90% en los paros de producción ocasionados por fallas de la torre de enfriamiento. El trabajo desarrollado emplea netamente automatización industrial y las alarmas solo son visibles en el tablero de control, lo que puede representar inconvenientes para el personal encargado del mantenimiento.

2.2.2. Antecedentes internacionales

En el panorama internacional el interés por el eje transversal de esta investigación se mantiene, es bien sabido que el aire comprimido se emplea en diversas industrias y no analizar su eficiencia energética repercute negativamente. Esto motivó a Cobeñas en [31] para plantear una solución basada en el criterio de verificación de la norma ISO 50001, diseñando un sistema de monitoreo y medición de las variables de caudal y presión en la línea de aire comprimido. Para esto, eligió tarjetas de desarrollo de bajo costo como el Arduino Nano y el NodeMCU para realizar una comunicación con un servidor local desarrollado con Apache que posee una base de datos MySQL. Todo el proceso se visualiza

desde una página web creada con ayuda de Bootstrap. Este planteamiento propone otra solución para implementar sistemas IoT que requieren más conocimiento técnico, pero garantizan mejores resultados.

Otra investigación con buenos resultados es la desarrollada por Escalona y Bohórquez en [32], en esta se automatiza el proceso de generación de aire comprimido de la cervecería nacional en Cumbayá incluyendo visualización online. Se emplea un autómata Kaesser, una red híbrida Profinet y el software InTouch para el diseño del aplicativo online. La implementación de este sistema garantiza una producción de aire comprimido de calidad y confiable que facilita la provisión de aire a todas las áreas de la empresa, reduciendo los tiempos de parada y el presupuesto destinado a mantenimiento.

Se puede adicionar el artículo elaborado por Murty y compañía en [33]. En este trabajo se desarrolla un sistema de monitoreo para compresores empleando el módulo ESP8266 y sensores como MEMS para detectar anomalías en la desalineación, la no linealidad, las vibraciones y la temperatura en los compresores. Para la visualización se emplea la plataforma Ubidots recopilando todos los datos necesarios para ser analizados matemáticamente en el dominio de frecuencia y de tiempo. Se concluye que es altamente beneficioso actualizar los sistemas actuales de IoT a unos más avanzados que reduzcan el cableado para mejorar la facilidad del montaje.

Mourtzis y compañía realizan otro aporte importante en [34], allí presentan el diseño y desarrollo de un marco de trabajo para la monitorización remota de sistemas de refrigeración, especialmente el compresor, que junto con el condensador forman el punto crítico del sistema. Se emplea una red de sensores inalámbricos con algoritmos inteligentes para el mantenimiento preventivo. El mayor aporte radica en el desarrollo de la red de sensores que minimiza los componentes que se emplean, convirtiéndose en una opción excelente para las pequeñas industrias.

Para cerrar este apartado se menciona el trabajo desarrollado por Ochoa y compañía en [9], que se enfoca en la creación de un sistema IoT para el monitoreo de la calidad del aire. Emplean sensores de concentración de dióxido de carbono, temperatura, humedad y densidad de polvo, acoplándolos a una Raspberry que funciona como bróker en el protocolo MQTT, se desarrolla también una aplicación que utiliza NodeJS, Express y SocketIO para representar los datos capturados y ejecutar órdenes de cambio. Este trabajo representa un camino opcional para desarrollar el sistema IoT cambiando las capas de percepción y evidentemente la de aplicación.

3. MAQUETACIÓN DEL SISTEMA

En la actualidad se han desarrollado diversas tecnologías, aplicativos y protocolos que permiten la creación de sistemas IoT, esta diversidad constituye demasiadas combinaciones de software y hardware que se pueden realizar para implementar estos sistemas. Para dar continuidad al desarrollo aquí planteado es necesario analizar algunas de las opciones más conocidas para realizar una elección de tecnologías que permita la planeación de la estructura general que tendrá el sistema y trazar así una ruta eficaz que lleve a la culminación de la propuesta de manera óptima. A continuación, se realizan los análisis y comparaciones necesarias que conducen a la elección de las tecnologías a emplear en cada etapa del sistema.

3.1. Selección del modelo de arquitectura

Como se mencionó en el capítulo anterior, existen varios modelos para describir la arquitectura de un sistema IoT, todos poseen el mismo enfoque y las mismas características, pero se diferencian en la división que se realiza para describir la ruta que tienen los datos desde su recepción en el mundo real hasta su disposición final en el mundo digital. Estas divisiones conforman diferentes capas en una estructura jerárquica y continua, donde el nivel más bajo es en el que ocurre la captura de los datos en el mundo físico y el nivel más alto comprende la disposición final que se le da a estos.

La disposición final es interpretada de diversas formas y esto constituye otro punto diferencial. Hay autores que defienden que el proceso realizado en un sistema IoT tiene como fin una aplicación específica donde los usuarios monitoreen, interpreten y controlen la información relacionada con los objetos conectados al sistema. Otros, en cambio, llevan la disposición final a un nivel más alto, donde se entiende que el tratamiento de datos se emplea para generar un beneficio económico, esta característica es innata de todos los procesos que se realizan en sociedad y por ende tiene más aceptación.

En resumen, los modelos de arquitectura se diferencian en incluir o no el modelo de negocio que se maneja en el espacio donde se establece el sistema. Entre los modelos ampliamente aceptados en la industria, se encuentra uno que da como disposición final la obtención de un beneficio económico y otra cuya última capa es la aplicación que tiene el conjunto de herramientas tecnológicas involucradas, estas son la arquitectura de cinco y tres capas respectivamente. Los detalles de estas dos arquitecturas se encuentran en el capítulo anterior. Definir la arquitectura es primordial porque con esto se establecen los apartados que se deben ir cubriendo a medida que se avanza en la investigación, para realizar esto se opta por la matriz de elección mostrada en la Tabla 1.

Tabla 1. Matriz de elección de arquitectura.

Requerimiento del sistema	Peso	5 Capas		3 Capas	
		Valor	Aporte	Valor	Aporte
Aquirir datos de sensores	5	5	25	5	25
Activar actuadores	5	5	25	5	25
Enviar datos a un servidor	4	5	20	5	20
Consultar datos de forma remota	4	5	20	5	20
Diferenciar el transporte de datos del procesamiento	2	5	10	1	2
Aportar a una aplicación específica	5	5	25	5	25
Incluir un beneficio económico	4	5	20	1	4
		Total	145	Total	121

Esta matriz muestra los requerimientos generales del sistema; en la columna “Peso” se asocia un valor de 1 a 5 de acuerdo al nivel de importancia, siendo 1 el nivel más bajo y 5 el nivel más alto. Las columnas “Valor” indican el beneficio que aporta a cada requerimiento el tipo de arquitectura, este se evalúa con la escala mencionada anteriormente. En las columnas “Aporte” se registra el resultado de multiplicar el “Peso” de cada requerimiento por el “Valor” que brinda cada arquitectura. Así se obtiene una representación numérica de los beneficios que aporta cada modelo. Los resultados de la matriz sugieren que es más apropiado el modelo de arquitectura de cinco capas. La diferencia radica en la importancia que tiene la diferenciación del transporte de los datos y la inclusión del beneficio económico. Además de esto, se elige la arquitectura de cinco capas porque resulta de comprensión más sencilla al tener los pasos más diferenciados y no dejar por fuera un factor tan importante como el beneficio económico que es intrínseco a toda actividad industrial.

Aunque la capa superior de la arquitectura elegida marca la diferencia, se debe aclarar que este documento no profundiza en esta por el contexto propio que posee el proyecto. Recordando con esto que, aunque está presente, los indicadores que reflejan los beneficios económicos de la implementación del sistema IoT son analizados en dependencias administrativas y no del área de ingeniería y mantenimiento. Las demás capas, que es donde sucede la ingeniería, representan la máxima de este trabajo y la maquetación de cada una se da a continuación.

3.2. Maquetación de la capa de percepción

En esta capa se encuentran todo tipo de sensores y actuadores, relacionados directamente con la aplicación, que por la situación global la mayoría de investigaciones se enfocan en los mismos temas: monitoreo de condiciones ambientales, optimización energética, optimización de recursos naturales, mantenimiento preventivo, inventarios,

control de calidad, etc. Todas ellas aplicaciones que requieren de sensores de humedad, presión, temperatura, distancia, luz, posición, velocidad, sonido y demás, también de actuadores que permitan las modificaciones de estas variables como contactores, electroválvulas, motores, electroimanes, entre otros.

La aplicación aquí tratada no es la excepción y requiere traductores que permitan ir de fenómenos físicos a información digital y viceversa. Estos elementos son elegidos con base en factores como las variables que intervienen en el proceso, el tipo de señal eléctrica que manejan, el índice de protección, el costo del componente y los niveles de tratamiento requerido para sus señales. En esta instancia de maquetación solo es posible mencionar las variables que se identificaron durante el planteamiento del problema, los demás factores son tratados más adelante.

En la sala de compresores se identificaron de manera visual tres variables fundamentales que deben ser captadas por los sensores: presión, humedad y voltaje; las dos primeras asociadas al aire comprimido que allí se genera y la última relacionada con el estado de los compresores (encendido y apagado). En cuanto a los actuadores, se identificó que se requieren elementos que permitan la activación y desactivación de motores y solenoides. Aunque esta observación parece superficial, permite dar orientación a la búsqueda de información para hacerla más específica e ir dando avances desde lo macro hacia lo micro de todo el sistema.

3.3. Maquetación de la capa de transporte

Antes de iniciar este apartado, es pertinente aclarar que pueden existir confusiones con el lugar que ocupan algunos protocolos y tecnologías en el modelo de arquitectura de cinco capas en IoT y el que tienen estos mismos en el clásico modelo de interconexión de sistemas abiertos (OSI), se sigue tratando de los mismos protocolos, pero en una interpretación de flujo de datos diferente. Dicho esto, en la capa de transporte se cuenta con protocolos de comunicación como el CoAP (Protocolo de Aplicación Restringida), MQTT y HTTP (Protocolo de Transferencia de Hipertexto), que permiten el envío de datos desde y hacia un servidor web o local. Estos protocolos requieren de un dispositivo que envíe los datos receptados por los sensores y reciba las instrucciones para modificar las variables a través de los actuadores; por facilidad se utilizan sistemas embebidos que permiten la conexión de módulos para envío de datos por medio de Bluetooth, wifi y ZigBee. Esto genera una cantidad de opciones significativa que abarca tarjetas de desarrollo de familias como Raspberry, Arduino, FPGA (Field Programable Gate Array), BeagleBone, ESP, ATTINY e incluso circuitos de diseño propio con microcontroladores menos conocidos.

En relación con la información suscitada en el párrafo previo, se debe realizar la elección de tres tecnologías: protocolo de comunicación, medio inalámbrico de envío y sistema embebido.

3.3.1. Elección de protocolo de comunicación

En el segundo capítulo se introdujeron los tres principales protocolos para esta capa, corresponde aquí realizar la comparación entre ellos para definir el más adecuado en la aplicación que se desarrolla. Esta comparación se realiza a través de la Tabla 2 donde se exponen las principales características de cada protocolo.

Tabla 2. Comparación de protocolos de transporte.

	Protocolo		
Característica	HTTP	MQTT	CoAP
Patrón	Cliente/Servidor	Publicar/Subscribir	Cliente/Servidor
Complejidad	Alta	Baja	Baja
Tamaño de los mensajes	Grande, encabezado detallado basado en texto	Pequeño, encabezados binarios	Pequeño, encabezado codificado
Niveles de servicio	Uno	Tres	Dos
Librerías	Alto nivel de desarrollo	Variedad de aplicaciones basadas en C	Variedad dependiente de aplicación
Licencia	Abierta	Abierta	Abierta
Topología	1 a 1	Estrella	1 a 1
Seguridad	TLS incorporado	Disponible SSL/TLS	Disponible DTLS
Tranporte	TCP	TCP	UDP

El análisis realizado sobre la Tabla 2 descarta de manera efectiva dos de los protocolos. El HTTP queda fuera por su nivel de complejidad alto, su gran requerimiento de ancho de banda causado por el tamaño de sus encabezados, su único nivel de servicio, su poco soporte en la comunidad y su topología que aumenta el nivel de consumo de ancho de banda cuando son varios los elementos enlazados al servidor. Los dos restantes poseen muchas similitudes y se enlistan como buenas opciones para ser implementadas en un sistema embebido de poca capacidad, comparten el nivel de complejidad, el tamaño de los encabezados, las librerías, la licencia, la capacidad de seguridad; sin embargo, hay dos diferencias que llevan la inclinación a favor de MQTT, estas son: la topología, que en estrella es mucho más beneficiosa para el tipo de sistema a implementar, y el patrón que posee un solo dispositivo encargado de direccionar los mensajes a través de las publicaciones y suscripciones. Por lo tanto, el protocolo de comunicación a emplear en la capa de transporte es MQTT.

3.3.2. Elección de medio inalámbrico

La principal función que debe cumplir este es la comunicación entre el sistema embebido y el servidor donde se aloje la página web, de antemano se ha descartado una conexión alámbrica porque implica una ruta demasiado larga y por normativa se deberían ubicar repetidores cada 100m y emplear un cable UTP CAT6 haciendo que los costos

aumenten drásticamente, además que la conexión debe ser aprobada por una auditoria que puede tardar meses. Así pues, entre las opciones restantes que no involucren cableado y sean comunes se tiene wifi, Bluetooth y ZigBee. Lee y compañía en [35] realizan acertadas descripciones de cada una de estas tecnologías, mencionan que: Bluetooth es el estándar IEEE 802.15.1 y consiste en una red inalámbrica de área personal diseñada para dispositivos de pequeña capacidad informática como los teclados y ratones de computador, contempla una comunicación entre un maestro y uno o varios esclavos que esperan en estado de reposo la señal del maestro, esto para consumir menos energía. ZigBee es el estándar IEEE 802.15.4 y define las especificaciones para redes de área personal de velocidad baja que soporten dispositivos sencillos, se diferencia del anterior por la velocidad y la forma en que se organizan los elementos, en ZigBee existe una red autoorganizada que funciona siguiendo jerarquías, hay elementos de función completa y de función reducida. Finalmente, wifi incluye las normas IEEE 802.11.a/b/g para redes de área local, más grandes y robustas que las de área personal, su principal característica es que al conectarse a un punto de acceso se permite la conexión a internet a velocidades de banda ancha. Con la información proporcionada por Lee se puede realizar la comparación presentada en la Tabla 3.

Tabla 3. Comparación de medios inalámbricos.

	Bluetooth	ZigBee	WiFi
Banda de frecuencia	2.4GHz	2.4GHz	2.4G; 5GHz
Velocidad máxima de señal	1Mb/s	250Kb/s	54Mb/s
Distancia nominal	10m	10m	100m
Numero de canales RF	79	16	14
Célula básica	Piconet	Estrella	BSS
Protección de datos	16 Bit CRC	16 Bit CRC	32 Bit CRC

Fuente: Adaptación de [35].

Teniendo en cuenta la comparación realizada en la Tabla 3, especialmente en la distancia nominal y las velocidades de señal, se decide emplear wifi como medio inalámbrico. Esta decisión se ve reforzada por el hecho de que emplear Bluetooth o Zigbee requiere además del nodo de conexión un nodo de recepción, en cambio, empleando wifi el nodo de recepción es el punto de acceso que ya se tiene en ARTYTECMO.

3.3.3. Elección de sistema embebido

En el capítulo correspondiente al referente teórico se mencionan tres familias de sistemas embebidos muy potentes y altamente empleados en el desarrollo de aplicativos IoT, aquí se comparan las tarjetas Arduino Uno, ESP32 DevKit V1 y Raspberry Pi 3 y se realiza la selección basándose en criterios tecnológicos y económicos. Para ver de forma más clara las características de las tarjetas se ubica la información en la Tabla 4.

Tabla 4. Comparación de sistemas embebidos.

	ESP32 DevKit V1	Arduino Uno	Raspberry Pi 3
GPIOs	34	14	8
Arquitectura	32bits	8bits	32bits
WiFi	Sí	No	Sí
Bluetooth	Sí	No	Sí
RAM	520KB	2KB	256MB
FLASH	15Mb	32Kb	SD Card
DAC	2	0	1
ADC	18	6	0
Interfaces	SPI, I2C, UART, 12S, CAN	SPI, I2C, UART	SPI, I2C, UART
Tamaño	5x2.3x0.8cm	8x5.5x2.5	8.5x5.3cm
Precio	32.000,00	42.000,00	600.000,00

Dar un vistazo a la Tabla 4 sugiere que el Arduino Uno no cumple con el requerimiento indirectamente planteado en la sección anterior, poseer wifi. Además, sus memorias RAM y FLASH se quedan cortas para la recepción y transmisión de información. Estos dos factores lo descartan de forma automática. Analizando las dos tarjetas de desarrollo restantes se encuentran bastantes similitudes, ambas ostentan una arquitectura de 32 bits, poseen wifi y Bluetooth; sin embargo, se observa que las capacidades de la Raspberry Pi 3 están orientadas a otro tipo de aplicaciones, posee memoria suficiente para correr servidores en su interior y, en cambio, no posee tantos GPIOs para conexión de sensores y actuadores, todo esto acompañado de la garrafal diferencia de precio hace que la balanza se incline a favor de la ESP32 DevKit V1.

3.4. Maquetación de la capa de procesamiento

En la capa de procesamiento se encuentran varias herramientas. Las plataformas IoT desarrolladas por compañías como MathWorks, Arduino, Amazon y Node RED, revolucionan la implementación de estos sistemas porque ofrecen paquetes muy completos para el procesamiento y observación de datos. Pero si se desea mayor personalización de los paneles de visualización y control en el tratamiento de datos, se deben desarrollar los aplicativos webs con las herramientas clásicas como HTML, CSS y JavaScript. Además, se deben tener en cuenta que algunas aplicaciones requieren de bases de datos que se gestionan normalmente en MySQL, MongoDB, Oracle, entre otras. Para finalizar, es importante mencionar que aparecen algunas herramientas extra si el sistema se desarrolla de forma local porque se debe tratar la creación de un servidor, esto normalmente se realiza con máquinas virtuales que seccionan una parte de un computador y permite que el servidor se ejecute en segundo plano.

Si bien la elección de plataformas ya creadas facilita enormemente el trabajo, por temas netamente legales no se pueden emplear en ARTYTECMO. Se debe recurrir entonces a las herramientas clásicas para la creación del aplicativo web, esto sugiere el empleo de

HTML en su quinta versión, CSS en su tercera versión y JavaScript, además de una herramienta que permita la creación de una base de datos con licencia libre.

Para cubrir los requerimientos anteriores se presentan varias opciones que agrupan tecnologías compatibles entre sí, orientadas al desarrollo de aplicaciones web. La primera opción es LAMP, que corresponde a un acrónimo de Linux (sistema operativo), Apache (servidor HTTP), MySQL (sistema de gestión de bases de datos) y PHP (lenguaje de programación de uso general orientado al desarrollo web), todos componentes de uso libre y licencia abierta. Una segunda opción es LEMP, donde la L vuelve a ser Linux, la E es para Nginx (servidor web), la M para MariaDB (gestor de base de datos) y finalmente P para PHP o Perl (lenguaje de programación orientado al desarrollo web). En el tercer lugar se encuentra MEAN, acrónimo de MongoDB (gestor de base de datos de NoSQL), Express (entorno de trabajo para programadores NodeJs), Angular (desarrollo web del lado del cliente) y NodeJs (entorno de ejecución del conjunto). Finalmente, aparece XAMPP, donde la X se refiere a cualquier plataforma o sistema operativo, la A para Apache, la M para MariaDB y la doble P para PHP.

Las opciones expuestas pueden ser utilizadas de forma indiferente porque logran los mismos resultados y consumen a grandes rasgos los mismos recursos, lo que deja como factor decisivo el conocimiento que tiene el desarrollador acerca de cada tecnología. Dicho esto, se menciona que el conjunto que aquí se emplea es XAMPP con una ligera modificación en el gestor de base de datos, en lugar de MariaDB se utiliza phpMyAdmin que es una herramienta basada en PHP orientada a la administración de MySQL. Con Apache se crea un servidor local para albergar la página web y sus complementos, esta condición es esencial para la seguridad del proyecto y fue impuesta por las directrices de ARTYTECMO para evitar la creación de puntos de acceso débiles en la red de la empresa.

3.5. Diagrama general del sistema

Todas las elecciones realizadas a lo largo del presente capítulo constituyen el esquema general que tendrá el sistema, este se condensa en la Figura 7 y permite visualizar las tecnologías empleadas.

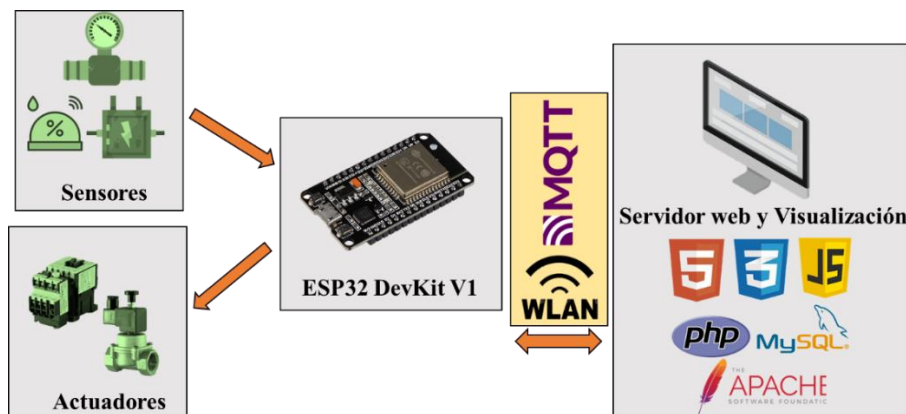


Figura 7. Esquema general del sistema IoT propuesto.

A modo de resumen, en la capa de percepción se cuenta con sensores de humedad, presión y voltaje, y contactores como actuadores que se acoplan en paralelo a los sistemas manuales que ya poseen los equipos de la sala de compresores. En la capa de transporte será la tarjeta ESP32 DevKit V1 la encargada de direccionar los datos de los sensores y enviar las acciones a los actuadores por medio de conexiones cableadas; a su vez la tarjeta se comunicará mediante el protocolo MQTT con el servidor que aloja la página web que permite la interacción del usuario con la sala. Esta última se realiza con herramientas tradicionales de desarrollo web como HTML, CSS, JavaScript y PHP. Los detalles más específicos del sistema se dan más adelante.

4. DESARROLLO FÍSICO

Este capítulo desarrolla el componente físico del sistema IoT que se implementa. Inicia con la identificación plena de los componentes que se deben intervenir y la realización de planos, luego se realiza la selección de sensores y actuadores, trazando los circuitos necesarios para su adaptación a la tarjeta de desarrollo.

4.1. Reconocimiento del entorno

El primer paso es realizar un reconocimiento de la sala de compresores para identificar los detalles que se deben tener en cuenta para las etapas posteriores, por esto se realiza un levantamiento de información que inicia con una visita a la zona que se muestra en la Figura 8. Se identifica allí que la sala se compone de un compresor principal, uno secundario y un tanque acumulador.



Figura 8. Sala de compresores de ARTYTECMO.

El compresor principal es de la marca Ingersoll Rand, serie R 4-11, funciona con un tornillo rotativo acoplado a un variador de frecuencia que controla el motor trifásico de 230V, brindando un trabajo silencioso y un arranque sin recargos a la red eléctrica. Puede trabajar a una velocidad máxima de 3500rpm produciendo en su tanque de 80gal una presión de hasta 150PSI, pero está configurado para dar de forma constante 125PSI. Posee un panel de control donde se puede visualizar el tiempo de operación (fundamental para realizar los mantenimientos), la presión de trabajo y realizar diversas configuraciones, incluyendo la vinculación a una red inalámbrica para visualizar estos datos de forma remota, esto puede significar un avance importante en el desarrollo de la investigación, pero se descarta porque solo se consigue desde la página oficial de la marca y para compartir la información se requiere la renovación de una licencia por parte de ARTYTECMO. Para finalizar se menciona que el compresor implementa un sistema de enfriamiento para el aire e incluye un sistema automático para el alivio de condensados.

Pensando en las variables que se identificaron en la etapa de maquetación, se enfoca la observación hacia las opciones disponibles para la obtención de variables y estados. La tarjeta que integra el panel de control dispone de borneras para encender y apagar el compresor de forma remota. El paro de emergencia cuenta con un contacto normalmente abierto utilizable en la lectura de su estado. Se identifica una intervención sencilla de las fases para leer de forma digital si está o no activa. Para la adaptación de sensores de presión se puede realizar una derivación de la salida del compresor.

En cuanto al compresor secundario, se tiene uno de la marca ya mencionada, esta vez de la serie UP6. Posee características similares al compresor principal, ambos cuentan con la misma capacidad en su tanque, funcionan al mismo nivel de voltaje y a la misma velocidad máxima. Las diferencias radican en que el compresor secundario carece de variador de frecuencia, sistema automático de alivio de condensados, panel de control inteligente con acceso a red inalámbrica y sistema de refrigeración para el aire. Para suplir el alivio de condensados se dispone de una válvula temporizada que se configura in situ.

El tanque acumulador posee una capacidad de 1000L y acumula en su interior el aire a una presión de 125PSI, es indispensable en el proceso porque garantiza flujo constante en toda la red neumática. Cuenta con una válvula de alivio similar a la del compresor secundario y un manómetro análogo.

Los dispositivos mencionados anteriormente se configuran como muestra el diagrama P&ID plasmado en la Figura 9. Este diagrama se realiza según la norma ISA S5.1 y puede ser consultada para la correcta interpretación de los elementos, pero a modo explicativo se aclara que las líneas segmentadas representan señales eléctricas, las líneas intersecadas con una L son conductos por donde fluye un líquido y las líneas intersecadas con líneas paralelas son para conductos neumáticos.

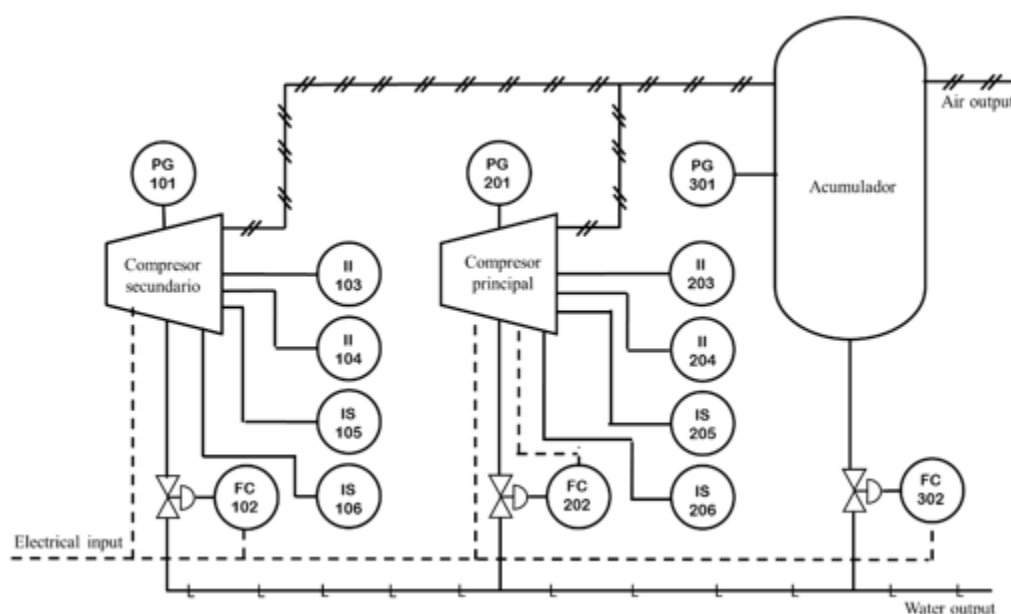


Figura 9. Diagrama P&ID de la sala de compresores.

Otro factor importante para la instalación de la instrumentación es la distribución física que posee la sala, esto nos dará las distancias del cableado en cada sección. Una representación física de la sala con las medidas más importantes se ilustra en la Figura 10, allí se representa con un rectángulo naranja una caja de distribución eléctrica que alimenta ambos compresores, las líneas verdes son las acometidas eléctricas, las líneas azules representan los ductos neumáticos.

Analizar la Figura 10 permite determinar la ubicación del tablero principal más conveniente, esta se da posicionándolo de forma alineada con el compresor principal en la pared donde se encuentra la distribución eléctrica antes mencionada.

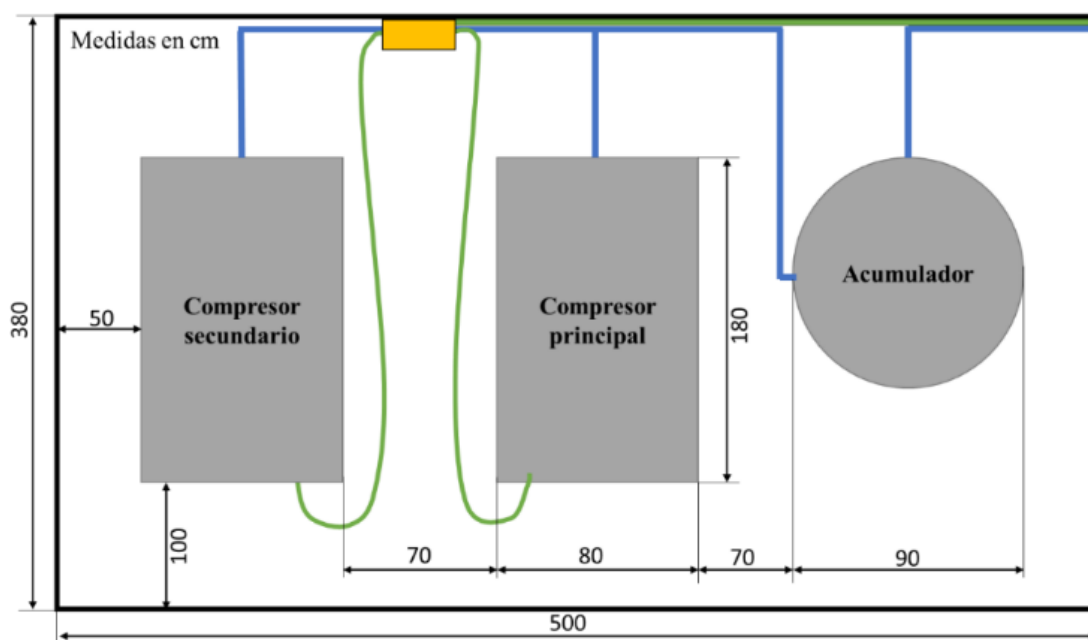


Figura 10. Distribución física de la sala de compresores.

4.2. Elección de instrumentación

En la fase de maquetación se identificaron las variables principales que se desean intervenir, sin embargo, no se profundiza en ese apartado sobre los factores que se pueden monitorear y manipular desde la página de visualización porque no se había realizado la inmersión en la sala. Con la ejecución del reconocimiento del entorno se logra la identificación detallada de todas las variables que se monitorean y manipulan desde la página web, esta información se condensa en la Tabla 5. Los mismos valores del compresor principal se repiten para el compresor secundario, pero no tabulan para evitar redundancias.

Tabla 5. Parámetros a monitorear y manipular.

Equipo	Variable	Instrumento requerido	Función general	Funcion remota
Acumulador	Humedad	Sensor de humedad	Registrar la humedad de la red en la salida del acumulador	Graficar humedad
	Voltaje	Relé optoacoplado	Conmutar de forma paralela el pulsador de prueba de la electroválvula temporizada	Aliviar condensados
	Presión	Sensor de presión	Registrar la presión del tanque acumulador	Graficar presión
Compresor principal	Presión	Sensor de presión	Registrar la presión del compresor principal	Graficar presión
	Voltaje	Relé optoacoplado	Conmutar las bornas de encendido de forma remota	Encender compresor
			Conmutar las bornas de apagado de forma remota	Apagar compresor
		Transformador de voltaje (Fuente)	Otorgar una salida con voltaje bajo cuando detecte una entrada con voltaje alto	Detectar alimentación eléctrica
				Detectar encendido del compresor
		Contacto normalmente abierto	Permitir el paso de la corriente cuando se active	Detectar pulsación del paro de emergencia

Para cumplir los requerimientos se debe elegir el dispositivo adecuado según lo estipulado en la tercera columna, la tarea reduce su complejidad porque algunos se repiten varias veces. Así pues, se da inicio con el sensor de humedad, se busca que este sea resistente a la presión, se pueda adaptar fácilmente al tanque acumulador y dé una medida relativa en porcentaje. Se seleccionaron dos sensores de humedad, la Tabla 6 compara sus características.

Tabla 6. Comparación de sensores de humedad.

Característica	Sensor	
	SHT20	AM2305B
Voltaje de alimentación	5V	3.3-5V
Señal de salida	Analoga (1 linea)	Analoga (1 linea)
Resolución	3%	0.1%
Distancia de transmisión	10m	20m
Tiempo de entrega (días)	5	20
Protección	Metálica	PVC
Costo	120.000,00	99.000,00

Realizada la comparación entre los dos sensores de humedad presentados, se opta por elegir el sensor SHT20 porque posee una mejor protección y resistencia a la presión, un tiempo de entrega mejor y es de fácil adaptación a la tarjeta ESP32.

El relé opto acoplado es un circuito de fácil construcción, pero para esta aplicación se ha elegido una estructura modular para facilitar el mantenimiento del sistema y el reemplazo de componentes averiados. Se usa el mismo tipo de módulo en aplicaciones donde se requieren contactores de 5V y 120V por lo que se toma el tope más alto como requisito. También es necesario que la activación se realice con una señal tradicional de 5 o 3.3V. Entre las opciones que aparecen no se encuentran diferencias significativas porque la mayoría son de marca genérica, sin embargo, hay un factor fundamental y es el número de canales por módulo, se encuentran opciones de uno, dos, cuatro, ocho, doce y hasta veinticuatro canales; se prefiere de un solo canal porque es más sencillo su reemplazo en caso de fallas.

Por su parte, el sensor de presión debe tener capacidad para medir 150PSI, una adaptación sencilla tanto a los compresores como al acumulador, una señal de salida de fácil acople a la tarjeta ESP32 y una precisión mínima del 5%. Se seleccionaron en una primera instancia dos sensores, estos se comparan en la Tabla 7.

Tabla 7. Comparación de sensores de presión.

	Sensor	
Característica	MPS20N0040D	USP-G41
Voltaje de alimentación	3.3-5V	5V
Señal de salida	I2C (2 líneas)	Analoga (1 línea)
Resolución	3%	1%
Distancia de transmisión	10m	30m
Tiempo de entrega (días)	3	3
Valor máximo	150PSI	1.2MPa
Protección	Matálica	Matálica
Costo	52.000,00	78.000,00

Presentadas las características más importantes de los sensores de presión seleccionados, se prefiere elegir el sensor USP-G41 porque posee un cuerpo tipo sonda y es más efectiva la medida de esta forma y no realizando una derivación como se debería en caso de elegir la otra opción. Además, emplea menos pines para la interpretación de los datos.

Para que la tarjeta de desarrollo pueda interpretar que el compresor está energizado, encendido o apagado, se eligió intervenir la línea de voltaje correspondiente. Hacer una toma muestra queda descartado de forma automática porque los niveles de voltaje que se manejan son destructivos para la ESP32, por esto es necesario implementar una etapa de acondicionamiento. Se puede pensar en un sensor de corriente de tipo herradura, pero no es posible porque se registran valores muy diferentes en ambos compresores debido a que solo uno tiene variador y el objetivo es unificar los circuitos empleados en ambos compresores.

Indagando un poco más se opta por emplear un transformador de voltaje con una configuración *pull up* en su salida, mientras no exista voltaje en las terminales de interés, la tarjeta de desarrollo recibe un cero lógico, pero cuando se energiza o enciende el compresor se transforma el voltaje de ingreso a una señal estándar de 5V.

El contacto normalmente abierto viene por defecto en el botón tipo seta, por lo que pierde sentido realizar una selección para este apartado. Dicho esto, queda realizada la selección de todos los componentes necesarios para el acondicionamiento de la sala de compresores.

4.3. Diseño de circuitos

Haber elegido circuitos modulares para la implementación de la instrumentación reduce en gran cuantía la complejidad de este apartado, solo se debe realizar la aclaración de las conexiones entre los módulos y mencionar que el nivel máximo de voltaje que trabaja la ESP32 en sus entradas análogas, digitales y salidas digitales es de 3.3V. Se adiciona con esto la necesidad de incluir un convertor bidireccional de nivel lógico porque la mayoría de instrumentación disponible en el mercado y elegida anteriormente trabaja 5V, este convertor también es modular y se adquiere de forma sencilla hasta de ocho canales. El módulo mencionado funciona perfectamente para entradas y salidas digitales, pero para interpretar los datos análogos ofrecidos por los sensores es inservible, para esta labor se identifican dos soluciones viables, la primera es la implementación de circuito mostrado en la Figura 11.

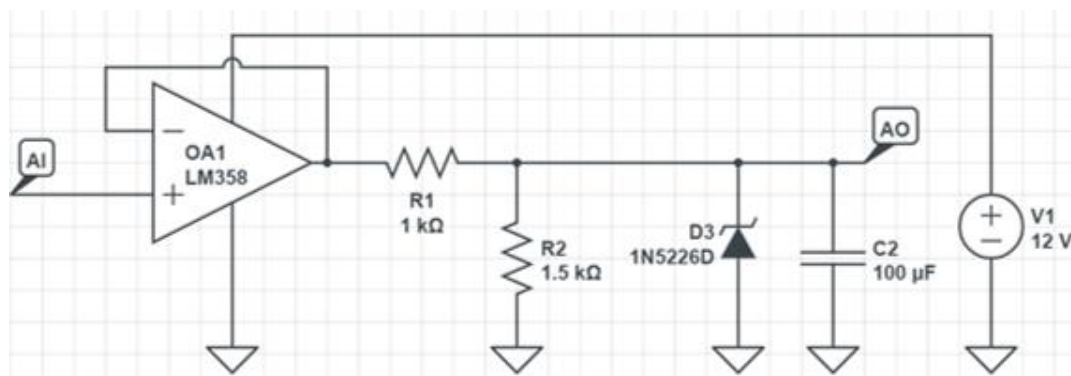


Figura 11. Conversor análogo 0-5V a 0-3.3V.

La segunda opción es utilizar un salto intermedio con el fin de continuar la metodología modular, es decir, emplear dos módulos concatenados, el primero convierte la señal de voltaje análoga de 0-5V a una señal de corriente estándar de 4-20mA, el segundo convierte esta señal de corriente en una de voltaje de 0-3.3V. Al existir más dispositivos las pérdidas de señal son mayores, pero es tan poco significativo que se elige la opción con escala por la facilidad de implementación que conlleva.

Con la información anterior se completa todo lo necesario para generar los diagramas de los circuitos; en total son cinco tipos que se repiten en varios escenarios, incluso para

manipular diferentes componentes de la sala de compresores, este fue un objetivo desde el inicio. La Figura 12 muestra el circuito de acondicionamiento para el sensor de presión. Este circuito se requiere tres veces para medir la presión en el tanque acumulador, el compresor principal y el compresor secundario. Aunque no es explícito, se da por entendido que la tierra de la fuente está conectada a la tierra de la ESP32 para generar una sola referencia.

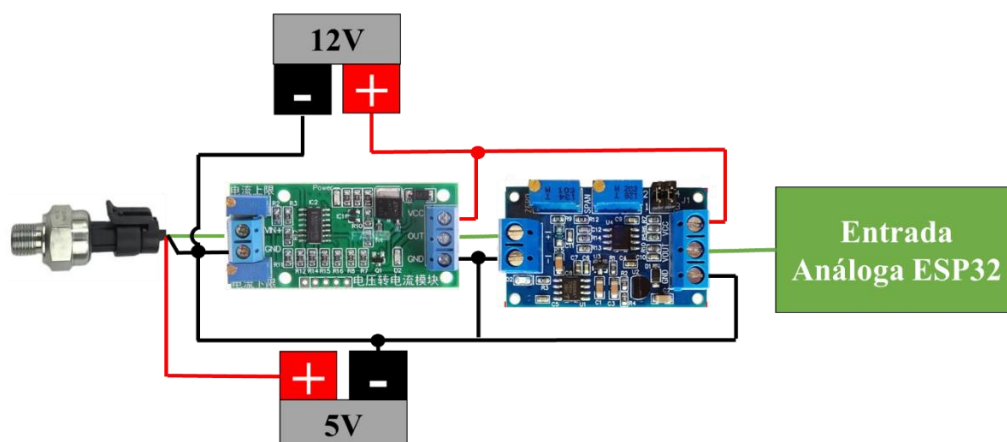


Figura 12. Diagrama del circuito para sensor de presión.

Continuando se tiene la Figura 13, allí se muestra el circuito para el sensor de presión. Este circuito se requiere una sola vez para medir la presión en el tanque acumulador, donde se encuentra la humedad más alta de toda la red. Los datos se adquieren mediante el protocolo I²C (Inter-Integrated Circuit). Se debe adicionar que el sensor también registra la temperatura, pero esta no se tiene en cuenta porque su variación es muy baja.

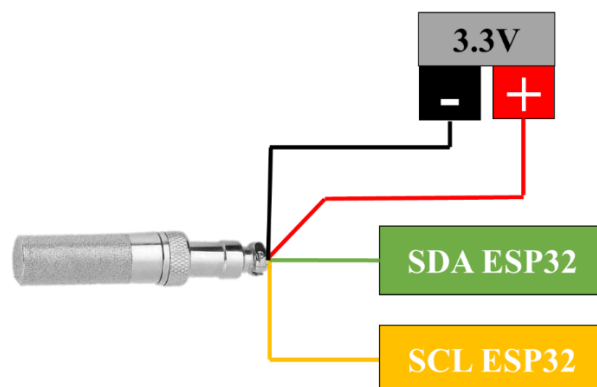


Figura 13. Diagrama del circuito para sensor de humedad.

La siguiente es la Figura 14, el circuito allí mostrado es el empleado para accionar de forma remota, el encendido y apagado de los compresores, además de elementos como la válvula de alivio del acumulador. La técnica ocupada para lograr el primer cometido es ubicar el contacto normalmente abierto del relé en paralelo al pulsador que ejecuta la acción que se desea controlar desde la página web. El circuito se implementa cinco veces en total.

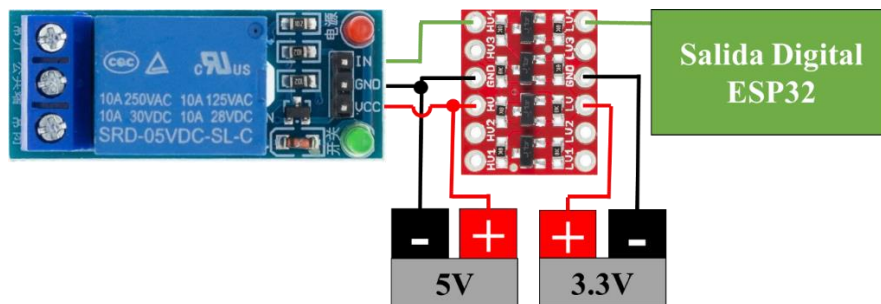


Figura 14. Diagrama del circuito para actuadores.

El siguiente circuito se muestra en la Figura 15, este es el sensor de alto voltaje más ligero que se pudo configurar entre las opciones disponibles. Lo que se percibe como una caja negra es una fuente de voltaje que transforma 240VAC en 5VDC. Este último voltaje es muy alto y debe bajar a 3.3V para ser leído por la tarjeta de desarrollo. La entrada digital a la que se conecta cuenta con un *pull down* que evita errores de lectura cuando la fuente tiene 0VAC en sus pines de entrada.

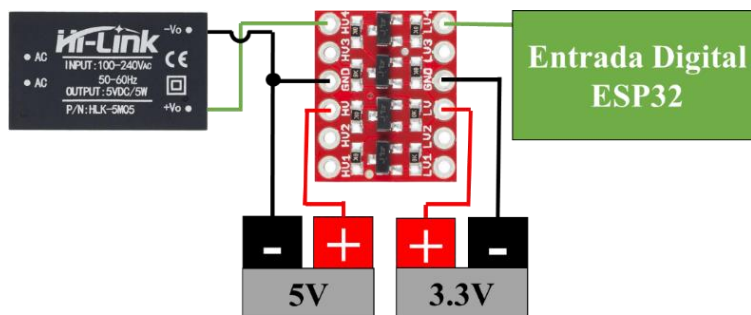


Figura 15. Diagrama del circuito sensor de voltaje.

Para finalizar se menciona el circuito de la Figura 16, empleado para detectar si el paro de emergencia de los compresores ha sido pulsado. Su principio de funcionamiento es muy sencillo, consiste en la capacidad del botón tipo seta de modificar mecánicamente un contacto normalmente abierto para dejar fluir una señal de 5V que es detectada por un pin digital con configuración *pull down* para evitar lecturas erróneas.

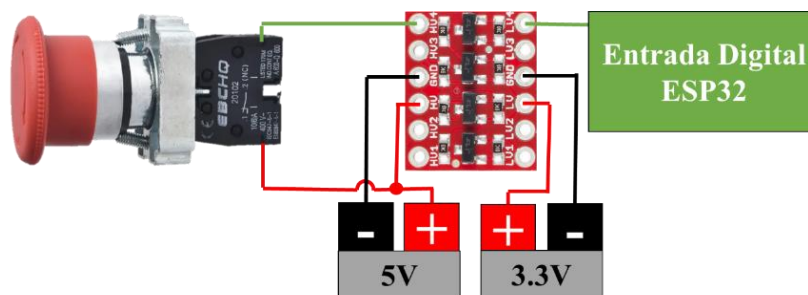


Figura 16. Diagrama del circuito sensor de paro.

5. DESARROLLO INFORMÁTICO

En este capítulo se abarcan todos los detalles de los componentes informáticos que integran el sistema IoT, es decir, lo relacionado con el software. Se siguen directrices y se aplican las tecnologías planteadas en la maquetación para lograr las conexiones dentro del sistema.

5.1. Bróker MQTT

Para realizar la comunicación entre la tarjeta de desarrollo y la página web mediante el protocolo MQTT, debe existir un bróker que se encarga del direccionamiento de los paquetes de datos mediante las publicaciones y suscripciones a los diferentes temas que se manejan, estos son conocidos como *tópicos*. La dirección física de este bróker puede ser almacenada al interior del sistema embebido elegido, sin embargo, esto representa un sobrecargo para la operación. Es mucho más sensato realizar la instalación en un equipo más potente como un computador, el mismo que se utiliza por defecto para monitorear la sala de compresores. A continuación, los detalles para obtener la operatividad del bróker.

5.1.1. Máquina virtual

Para evitar inconvenientes con el sistema operativo del computador que recibe al bróker, se decide emplear una máquina virtual; esta simula un sistema de computación independiente que posee todas las capacidades necesarias para la ejecución de programas. Son varias las opciones que se encuentran para este tipo de simuladores: Parallels, EMU, Windows Virtual PC, VirtualBox, vmWare, entre otras; mismas que ofrecen diversas alternativas en cuanto al sistema operativo, la cantidad de comunicaciones y el nivel de interacción que manejan. El nivel de interacción hace referencia al nivel de conocimiento que requiere el usuario para poder navegar y configurar apartados específicos, por mencionar un ejemplo, se citan los sistemas operativos Windows y Linux, el primero posee un nivel de interacción alto, lo que conlleva un nivel de conocimiento bajo por parte del usuario; caso contrario Linux que requiere mayor pericia por parte del usuario para modificar la mayoría de propiedades del sistema.

La máquina seleccionada para implementar en esta tarea es VirtualBox. Esta máquina desarrollada por Oracle Corporation, permite la instalación de diversos sistemas operativos, incluyendo MAC OS, la creación de hasta 32 CPUs virtuales, la variación de resolución de pantalla, el soporte a puertos USB del sistema padre y ofrece clonación de máquinas reales o virtuales. Además, brinda una licencia de uso abierto para evaluación y desarrollos menores.

La instalación se realiza desde la página oficial para un sistema amd64. La configuración que se realiza incluye un tamaño de disco duro de 20GB y una memoria RAM de 1GB. El sistema operativo que se instala es Ubuntu, una conocida, elegante y ligera derivación del sistema de distribución de Linux Ubuntu; es gratuito, fácil de utilizar, se encuentra disponible en español y tiene una gran comunidad que da respaldo a nuevos

desarrollos. Lo anterior nos otorga una superficie para configurar un servidor local que soporte al bróker.

5.1.2. Enlace externo

Es fundamental crear un punto de acceso remoto para que la tarjeta de desarrollo pueda interactuar con el bróker, para esto se utiliza el protocolo de comunicación SSH (*Secure Shell*) mediante el programa PuTTY, que realiza conexiones seguras y encriptadas a través del puerto 22. Este software es de uso libre y es compatible con Ubuntu. Es importante aclarar que, para permitir este acceso, desde la máquina virtual se instalan los servicios de SSH, empleando en la terminal de comandos con permiso de superusuario el comando: *apt-get install openssh-server*. Para realizar la configuración del punto de acceso, inicialmente se debe añadir un nombre a la sesión y agregar la dirección IP del dispositivo que será consultado, en este caso la dirección de la máquina virtual antes creada.

Posteriormente, se carga el punto de acceso desde PuTTY, se ingresa con las credenciales de la máquina virtual y se otorgan permisos de superusuario para realizar una serie de instalaciones y configuraciones. Se inicia instalando Mosquitto con el comando *apt-get install mosquitto*. Mosquitto es un programa de licencia abierta que funciona como agente de comunicaciones del protocolo MQTT, es el encargado de administrar los datos de los temas, en otras palabras, Mosquitto es el bróker del sistema. Al ser un software de bajos requerimientos informáticos es perfecto para ejecutarse en nuestra máquina virtual, dejando de lado en la competencia agentes como Mosca, EMQTT o ActiveMQ.

Seguidamente, se debe realizar la instalación del complemento para los clientes, esto se realiza mediante el comando *apt-get install mosquitto-clients*. De esta forma se habilitan las herramientas para realizar el test inicial de funcionamiento de la comunicación MQTT a través de un cliente virtual. Para realizar la prueba se debe tener una terminal suscrita y una que publique, para obtener la primera se registra desde PuTTY un nuevo tema usando el comando *mosquitto_sub -h localhost -t "topic/Probe"*, donde Probe es el nombre que recibe el tema. Para obtener el publicador se abre una ventana nueva en PuTTY y se ejecuta el comando *mosquitto_pub -h localhost -t "topic/Probe" -m "Primer test"*, esto publica en el tema Probe el mensaje "Primer test". Si la comunicación es un éxito, el mensaje se debe reflejar de forma inmediata en la ventana del suscriptor.

Al ejecutar la iteración anterior se obtienen resultados buenos, significando esto que el bróker se está ejecutando de forma correcta en la máquina virtual y que el acceso remoto a través de PuTTY está listo para comunicaciones externas.

5.1.3. Prueba de comunicación MQTT

La finalidad de este apartado es asegurarse de que el bróker puede direccionar mensajes desde una página web, empleando para ello el protocolo WebSocket, mismo que permite la comunicación bidireccional entre un servidor y un cliente, el primero será nuestra máquina virtual y el segundo una página web de prueba. Los pasos que siguen se ejecutan en la aplicación final del sistema IoT y son indispensables para verificar su funcionamiento.

Permitir la comunicación WebSocket requiere la intervención del bróker, se crea en sus directorios un archivo con el nombre *websockets.conf* y allí se depositan los puertos que empleará cada protocolo, se elige el 1883 para el MQTT y el puerto 9001 para WebSocket. También se hace necesaria la desactivación de la conexión IPv6 para evitar interferencias y retardos de conexión a la IPv4, esto se realiza en el directorio *sysctl.conf*.

Realizadas las configuraciones del protocolo WebSocket se abre la terminal de comandos de la máquina virtual y se realiza la instalación de Apache mediante el comando *apt-get install apache2*, este complemento otorga a la máquina virtual servicios de HTTP permitiendo la visualización de los archivos referenciados como HTML desde un navegador como página web. También es necesario instalar Git aplicando el comando *apt-get install git*, esto permite la clonación de repositorios GitHub. La obtención de la página de prueba se hace clonando el repositorio HiveMQ, este otorga una página web que permite conexiones WebSocket.

Después de realizar la clonación se accede al navegador de preferencia desde cualquier dispositivo conectado a la red wifi del computador anfitrión de la máquina virtual y se ingresa a la página de HiveMQ obteniendo la visual de la Figura 17. En el apartado de conexión se ingresan los datos de la máquina virtual y se conecta, cuando el estado cambia a conectado se agrega un nuevo tema en el apartado de suscripciones. Luego en publicaciones se escribe el tema en el que se realizará la publicación y el mensaje. Si todo está en orden al pulsar el botón publicar se debe reflejar el mensaje en el apartado de Mensajes.

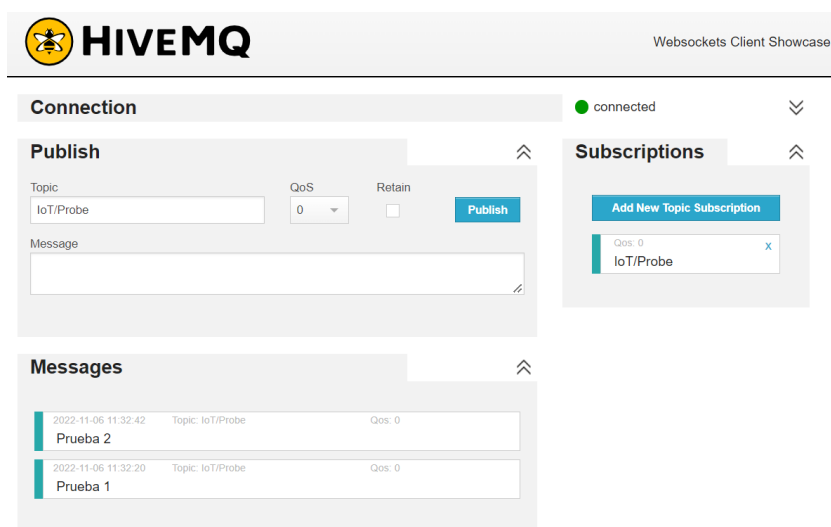


Figura 17. Página web para pruebas.

5.2. Desarrollo de la página web

Este apartado contempla la construcción del aplicativo de visualización, referenciado algunas veces como página web, incluye la planeación, la estructura que posee y finalmente la apariencia y navegación final del visualizador.

5.2.1. Planeación

La importancia de la página web es notablemente alta, es el componente del sistema con el que el usuario entra en contacto y desde donde se pueden visualizar todas las variables de la sala de compresores. La funcionalidad de la página web se discutió con el equipo de mantenimiento de ARTYTECMO quienes serán los usuarios constantes del aplicativo, los resultados de estas discusiones se plasman en la Figura 18. Se decidió realizar un inicio de sesión restringido porque, aunque el aplicativo se ubica principalmente en un computador de la oficina de Mantenimiento Industrial, es posible acceder a la página desde cualquier dispositivo conectado a la red wifi local de la empresa. El tablero de mando e instrumentos, de ahora en adelante Dashboard, es de fundamental inclusión, es la razón de ser de la página web. Se agregaron también apartados para soporte técnico brindado por el desarrollador del proyecto y acceso a documentos de interés como manuales y lista de partes de los compresores. Los apartados de visualización de datos y las entradas del menú llevan a otras páginas que poseen características especiales según corresponda el caso.

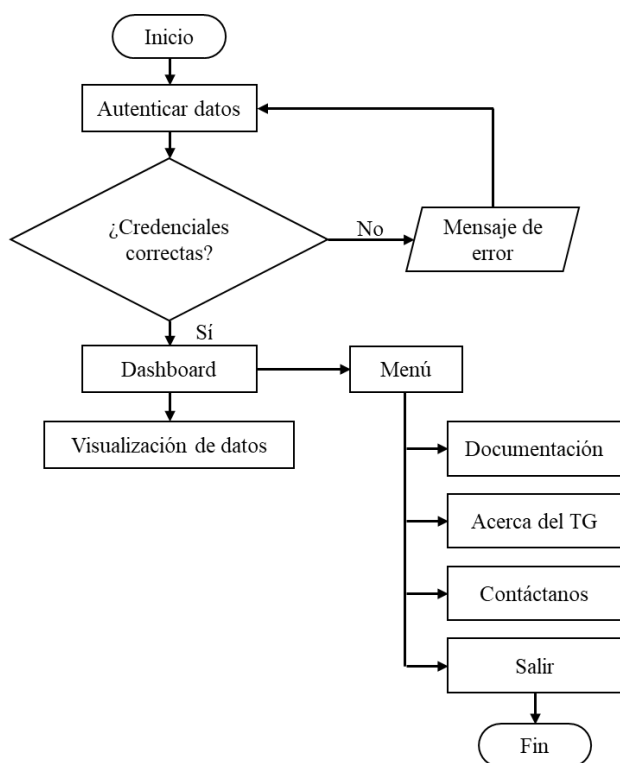


Figura 18. Diagrama general de la página web.

5.2.2. Estructura

Para el desarrollo de la página propuesta se emplea el conjunto XAMPP que ofrece todas las herramientas necesarias para el soporte local del aplicativo. La instalación del conjunto deja un repositorio en el disco principal del computador anfitrión de la máquina virtual con nombre “xampp” y dentro de este en la carpeta “htdocs” se almacenan todos los

archivos que conforman el aplicativo, la estructura y jerarquía de estos se visualiza en la Figura 19.

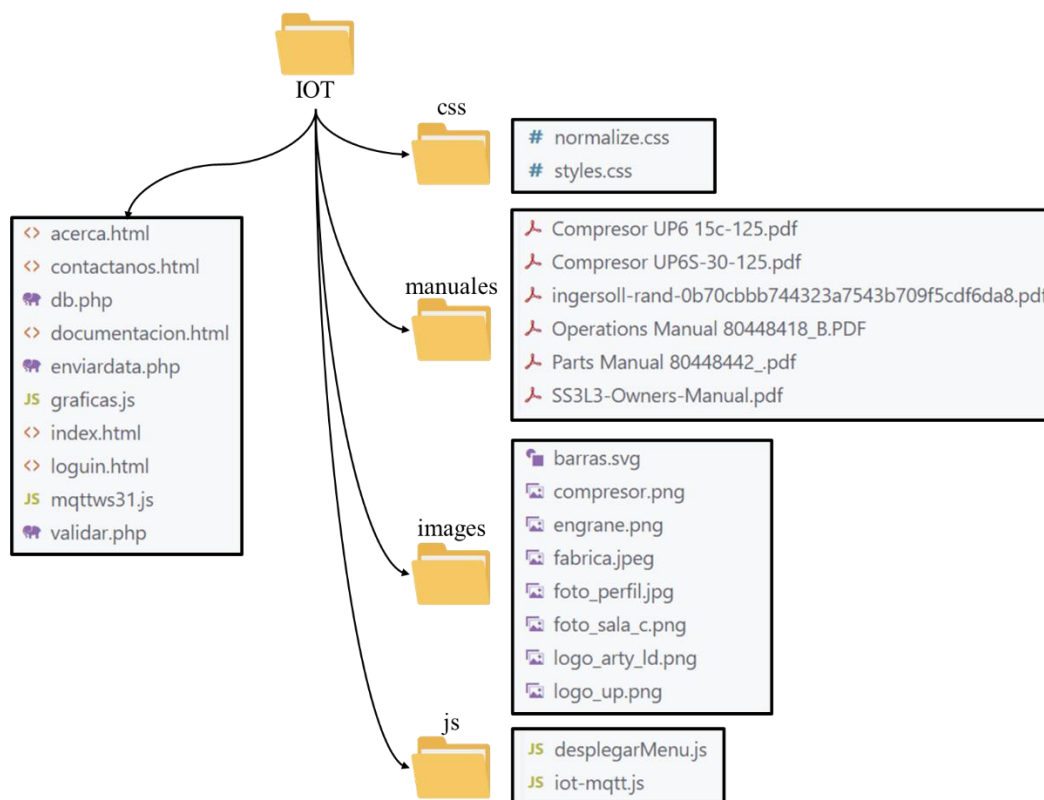


Figura 19. Estructura de archivos de la página web.

La carpeta de manuales almacena todos los archivos que podrán ser consultados en la opción del menú “Documentación”. La carpeta imágenes contiene las figuras que se muestran en la página, incluyendo los iconos de la pestaña superior cuando se está en el navegador, los fondos y elementos del encabezado. La carpeta css contiene un archivo de normalización que quita cualquier formato que incluya por defecto el navegador que abra la página, esto se emplea para tener mayor control de la organización de los elementos; también contiene el archivo de estilos que da apariencia a las clases y direcciones que usan los archivos HTML mencionados más adelante. Por su lado, la carpeta js contiene un archivo que da dinamismo al menú desplegable y otro en el que indican los detalles para realizar la conexión al bróker, publicar los valores de los botones y recibir datos de las gráficas, sobre este archivo se profundiza al final del capítulo.

De forma externa se encuentran los archivos principales a los que se direcciona navegando en el menú y algunos complementarios que permiten la creación de gráficas mediante la librería *ChartJs*, la escritura y lectura de la base de datos mediante PHP para inicio de sesión y registro de datos de contacto. Se aclara en esta parte que los datos de sensores y demás estados simplemente se visualizan y no se registran en ningún lugar porque no se tiene la capacidad física para dar tratamiento al gran volumen de datos que se generan.

5.2.3. Apariencia

Los estilos del aplicativo se realizan en base los colores corporativos de la empresa, se puntualiza en formas degradadas, transparencias opacas y carencia de esquinas. También se ajusta el diseño para la visualización en pantallas de resolución menor como la de un celular, técnica llamada comúnmente Diseño Responsivo. Se alternan vistas desde pantalla de computador (VL por Vista Laptop) y celular (VS por Vista Smartphone) en los diferentes apartados a fin de tener una mejor distribución en el documento.

Cuando el usuario ingresa a la página web se encuentra el entorno mostrado en la Figura 20. Debe digitar el usuario y la contraseña para acceder a la ventana principal, en caso de pasar la validación aparece una notificación en la parte superior como muestra la Figura 21.



Figura 20. Ventana de inicio de sesión VL.



Figura 21. Ventana de acceso denegado VS.

En caso de ser autenticado se da ingreso a la ventana del Dashboard que es la página principal, allí se pueden observar 4 secciones tal como refleja la Figura 22, la primera para la humedad de la red, la segunda para los valores asociados al acumulador, y las dos últimas para los datos asociados a los compresores principal y secundario.

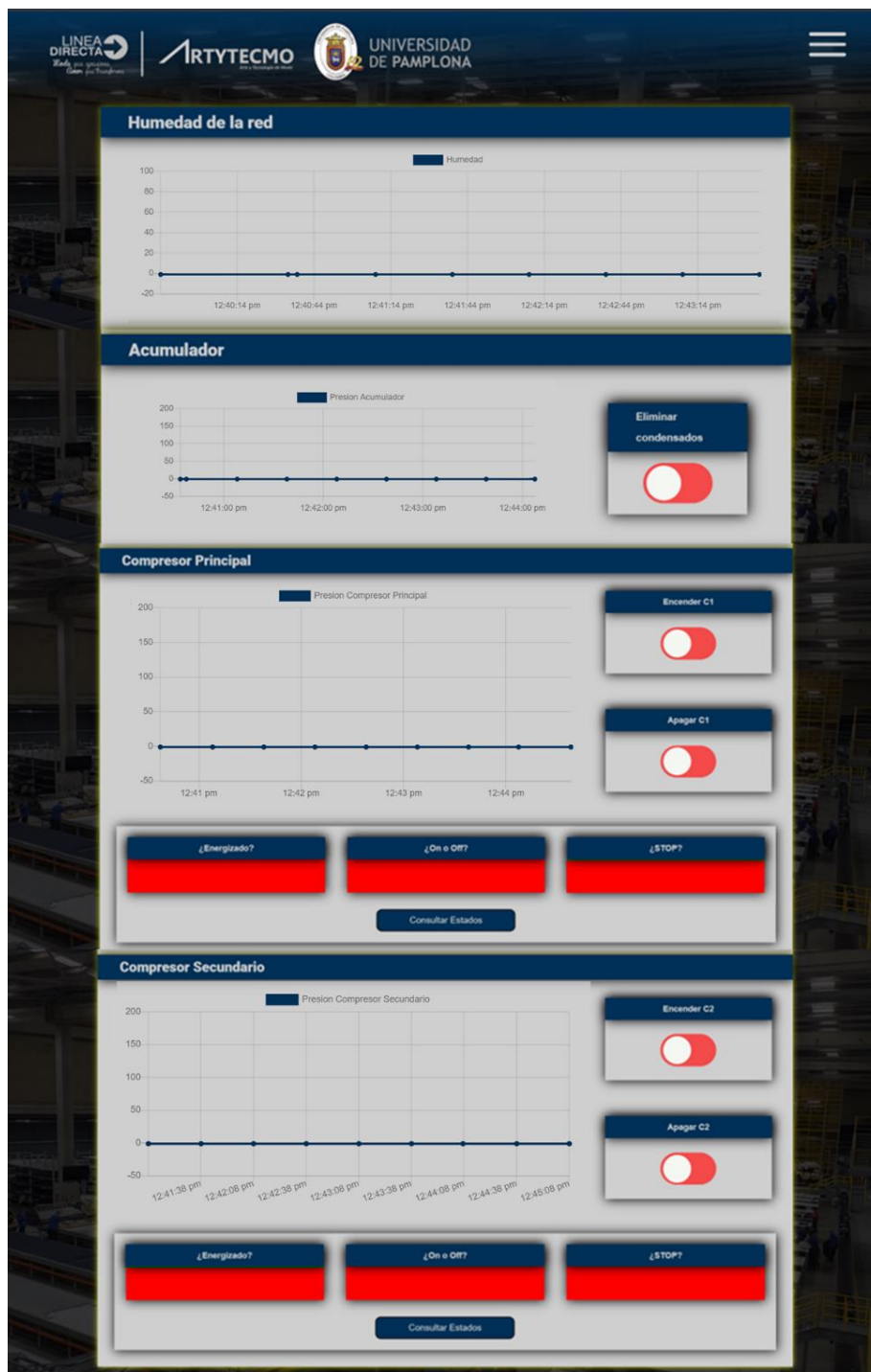


Figura 22. Ventana de Dashboard VL.

Seleccionando el menú desplegable de la parte superior aparecen las opciones que se han mencionado con anterioridad, navegar entre ellas es muy intuitivo. En la Figura 23 se muestra la apariencia de la opción “Documentación” con el menú desplegado en el lateral.

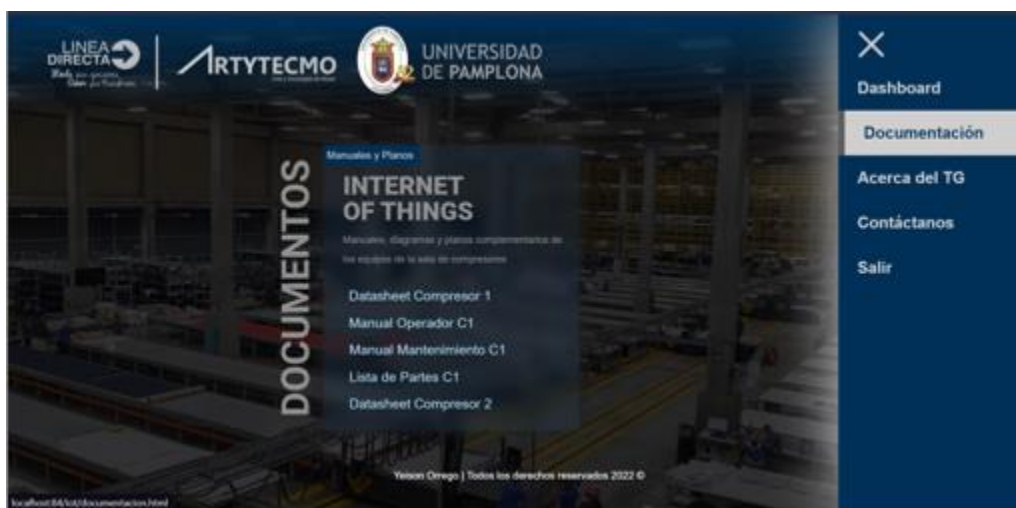


Figura 23. Ventana de Documentación con menú desplegado VL.

Las apariencias de las demás opciones se ilustran en el apartado de Anexos con el fin de dar mayor orden al documento. Cuando se selecciona Acerca del TG se muestra información general del trabajo desarrollado aquí y se explica su objetivo. En el apartado de Contáctanos se despliega un formulario donde se debe ingresar el celular y el correo para que el desarrollador del sistema de soporte técnico si se requiere, esta información se almacena en una base de datos administrada con phpMyAdmin y no es posible ingresar valores repetidos para evitar saturar la base de datos que tiene capacidades muy restringidas; cuando los datos se registran con éxito se da una notificación de color verde y cuando los datos no se han podido registrar o se intenta un registro doble se notifica de color rojo. Finalmente, la opción salir cierra a sesión y redirige a la página de inicio de sesión.

5.3. Conexión Web - MQTT

Aunque cada uno de los documentos mostrados en la estructura de archivos de la página web es importante y cumplen funciones específicas, hay algunos que pueden ser eliminados y la página perdería propiedades de apariencia o dinamismo, pero seguiría funcionando. No obstante, existe uno que cumple la función principal de este trabajo y no puede ser eliminado porque se perdería la esencia, esta función es la de enlazar la página web con el bróker para escribir y leer mensajes en los diferentes temas.

El archivo que por nombre lleva `iot-mqtt.js` es un código escrito en JavaScript que emplea la librería `Paho-MQTT` para configurar la página web desarrollada como un cliente del bróker establecido en la máquina virtual. Esta librería se incluye en el archivo `mqttws31.js` y es llamada únicamente en el código HTML de la página que contiene el Dashboard.

El código se estructura con un encabezado y diez funciones que entran en diversos momentos según se interactúe en la página. En el encabezado se da una dirección aleatoria para ser usada como identificador del cliente y se genera la solicitud al bróker con el identificador, la dirección IP de la máquina virtual (contenedora del bróker) y el puerto de la conexión WebSocket. También se realiza la inicialización de variables que almacenan los valores de los sensores y algunos estados.

De las funciones mencionadas siete (son de interacción) se ejecutan cuando se conmutan los interruptores o se pulsan los botones del Dashboard, estas funciones establecen cambios de estado en variables y si es necesario publican mensajes. Las funciones restantes poseen un nivel más alto en la jerarquía. La primera y más importante es la función *onConnect* que se ejecuta cuando el cliente es aceptado por el bróker, se encarga de generar las suscripciones a los quince temas que manejan, la Tabla 8 muestra para qué es utilizado cada uno y su rango de valores.

Tabla 8. Temas del sistema.

Tema (<i>topic</i>)	Descripción	Valor
IoT/Hum	Humedad de la red	0-100 (Porcentaje)
IoT/Ptk	Presion del acumulador	0-150PSI
IoT/AlvTk	Accionamiento de la valcula de alivio del acumulador	ON - OFF
IoT/Pr1	Presion del compresor principal	0-150PSI
IoT/OnC1	Accionamiento para encender el compresor principal	ON - OFF
IoT/OffC1	Accionamiento para apagar el compresor principal	ON - OFF
IoT/EnerC1	Registro de la conexión a red eléctrica del compresor principal	ON - OFF
IoT/OnOffC1	Registro del estado del compresor principal (encendido o apagado)	ON - OFF
IoT/ParoC1	Registro del paro de emergencia del compresor principal	ON - OFF
IoT/Pr2	Presion del compresor secundario	0-150PSI
IoT/OnC2	Accionamiento para encender el compresor secundario	ON - OFF
IoT/OffC2	Accionamiento para apagar el compresor secundario	ON - OFF
IoT/EnerC2	Registro de la conexión a red eléctrica del compresor secundario	ON - OFF
IoT/OnOffC2	Registro del estado del compresor secundario (encendido o apagado)	ON - OFF
IoT/ParoC2	Registro del paro de emergencia del compresor secundario	ON - OFF

Por su parte, la función *onConnectionLost* informa cuando la petición de cliente fue rechazada por el bróker. Finalmente, la función *onMessageArrived* establece lo que se debe hacer cada vez que llegue un mensaje con dirección a algún tema en el que se está suscrito, está dividida en una estructura condicional que verifica el tema destino y guarda el mensaje recibido en variables que son requeridas por las siete funciones de interacción.

5.4. Código ESP32

Completar el esquema de comunicación MQTT requiere un cliente adicional, este será la tarjeta de desarrollo ESP32, que además debe interpretar los mensajes que recibe para accionar los sistemas que están disponibles en la sala de compresores y a su vez leer los sensores e informar los cambios al bróker para que se puedan ver reflejados en la página web. Para realizar la programación del sistema embebido se emplea *PlatformIO* porque admite su integración al entorno *Visual Studio Code* que es donde se desarrolla toda la página web; para el desarrollador tener todas las herramientas en un mismo entorno es mucho más ameno y eficiente.

PlatformIO es un entorno de desarrollo integrado (IDE siglas de *Integrated Development Environment*) de licencia abierta, que se enfoca en la programación C/C++ y se orienta a dispositivos de hardware; proporciona depurador, detección de puerto automático, asistente inteligente de escritura y gestor de librerías integradas. Lo anterior lo hace perfecto para la aplicación que se desarrolla aquí.

Elegida la herramienta de desarrollo, se procede a realizar el código que se estructura de la forma que ilustra la Figura 24. Allí se observa que cuando se da inicio al programa, energizando la tarjeta de desarrollo, se realiza la inclusión de las librerías Arduino, wifi, *PubSubClient*, *stdlib*; la primera es la que permite la programación semejante a la utilizada por Arduino y otorga el uso de comandos como *digitalWrite*, *pinMode*, *Serial.println*, entre otros, se incluye porque el desarrollador está familiarizado al entorno Arduino y no con otros empleados para el mismo propósito como *MicroPython*. La segunda es la librería que otorga el control del módulo wifi integrado en la tarjeta de desarrollo, las configuraciones que se elaboran son esenciales para el sistema IoT. La siguiente librería es la encargada de envestir de propiedades de cliente MQTT a la tarjeta ESP32. Finalmente, la librería *stdlib* habilita prototipos de funciones generales de C para la gestión de variables.

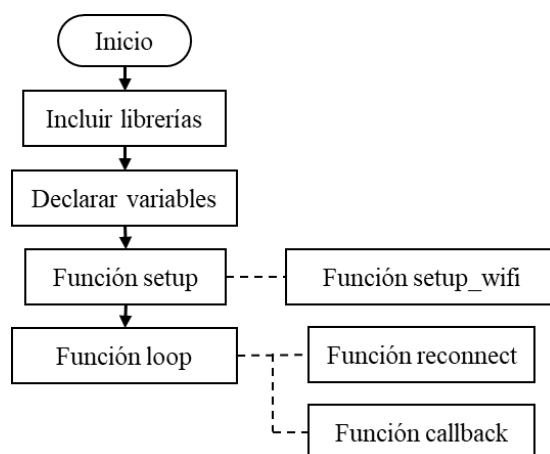


Figura 24. Esquema general del código para la ESP32.

En la declaración de variables se establecen los datos de la red wifi, la dirección del bróker, la creación del cliente MQTT y algunos registros para guardar los estados principales

de la sala de compresores. La función *setup* define algunos pines como salidas digitales, la velocidad en 115200 baudios y configura los primeros cinco temas correspondientes a las salidas digitales que manipula la tarjeta, además de esto llama a la función *setup_wifi* que configura la conexión a la red y se ejecuta hasta que el resultado es positivo. Por su parte, la función *loop* establece las actividades que se realizarán de forma permanente, iniciando por la conformación de la conexión al bróker, en caso de encontrarse desconectado realiza el llamado a la función *reconnect* que intenta la conexión hasta que sea positivo el resultado, para luego se suscribe a los cinco temas configurados para las salidas digitales. Cuando se ha realizado la conexión al bróker, la función *loop* continua con una revisión que ocurre cada diez segundos, en esta se realizan las lecturas de sensores y estados que se informan al bróker para ser publicados en la página web. Finalmente, la función *callback* se ejecuta como una interrupción del ciclo *loop* cada vez que se recibe un mensaje en los temas donde se ha suscrito la tarjeta de desarrollo. Se realiza una inspección del mensaje recibido descifrando el tema de interés y realizando la acción específica en los pines correspondientes.

Todas estas funciones y declaraciones conllevan el correcto funcionamiento de la tarjeta ESP32 DevKit V1 como cliente de la conexión MQTT, logrando la lectura de los botones situados en la página web y la escritura de los valores leídos en los pines de la tarjeta para ser mostrados en el Dashboard del visualizador.

6. RESULTADOS

En este capítulo se plasman los resultados de la investigación, se incluyen imágenes de los montajes realizados en la sala de compresores. Se realiza la validación del sistema mediante matrices de evaluación creadas especialmente para este propósito. Al final se elabora una guía de usuario que permite a los auxiliares de mantenimiento el correcto uso de la herramienta desarrollada.

6.1. Implementación de hardware

Si bien en apartados anteriores se trató la selección de instrumentación y el diseño de los circuitos, es tema del apartado actual la implementación física que se realizó en la sala de compresores. Esta se compone del gabinete central que se muestra en la Figura 25, contiene la fuente de alimentación del sistema, la tarjeta ESP32, los módulos de conversión de voltaje a corriente y de corriente a voltaje, los conversores lógicos bidireccionales y el módulo de relé opto acoplado que activa la válvula de alivio del tanque acumulador. Desde esta caja central se distribuyen líneas para ambos compresores y el tanque principal, en total son veinticuatro dedicadas a alimentación y señales de información. El gabinete cuenta con un totalizador que interrumpe la alimentación de todo el sistema. La instalación de la ESP32 sobre pines hembra permite su desacople sencillo para actualizaciones del código que contiene.



Figura 25. Gabinete central.

Dentro de los tableros propios de los compresores se hizo necesaria la implementación de una tarjeta con cuatro elementos para evitar el viaje por el mismo conducto de señales de alto y bajo voltaje porque esto puede ocasionar interferencias en las señales de control. La tarjeta implementada dentro del compresor principal se muestra en la Figura 26 y la hecha para el compresor secundario en la Figura 27, esta contiene los sensores de voltaje que permiten al usuario ver desde la página web el estado de los compresores,

también contiene los módulos relé que se emplean para el accionamiento del motor principal del compresor.



Figura 26. Tarjeta para compresor principal.

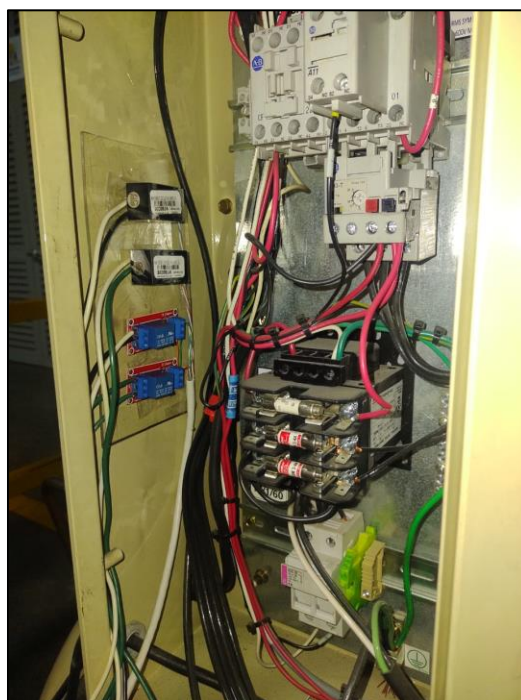


Figura 27. Tarjeta para compresor secundario.

Además de la tarjeta mostrada, cada compresor se equipa con un sensor de presión, para esto fue necesario realizar una derivación de la línea del manómetro análogo que tiene cada equipo. La Figura 28 muestra la instalación del sensor de presión realizada en el compresor secundario. De forma similar se hizo para el compresor secundario y el tanque acumulador.



Figura 28. Instalación sensor de presión en compresor secundario.

Para el tanque acumulador fue necesario realizar una segunda derivación para poder realizar la conexión del sensor de humedad, esta conexión se hizo de manera análoga al procedimiento anterior. La Figura 29 plasma el resultado de esta instalación.



Figura 29. Instalación sensor de humedad.

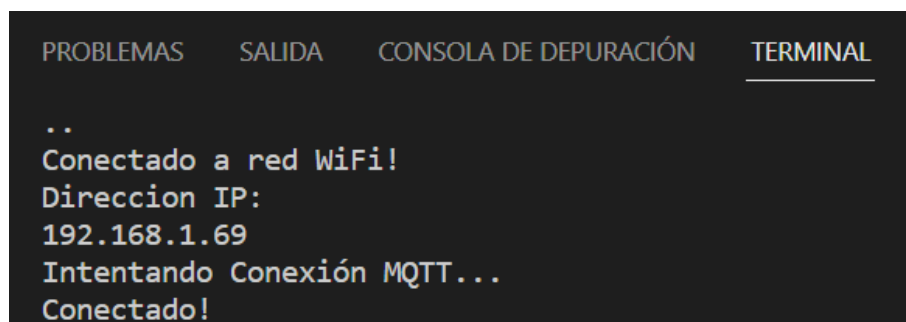
6.2. Validación del sistema IoT

Validar un proceso de tanto tacto de forma escrita conlleva una complejidad alta, por esto se han elaborado una serie de pruebas que tienen como fin ir demostrando paulatinamente el funcionamiento de los principales componentes del sistema.

6.2.1. Prueba de conexión

La primera prueba consistió en verificar que ocurra la conexión entre los componentes. Es necesario reiterar que la estructura de una comunicación MQTT se basa en una topología estrella cuyo nodo principal es el bróker, que en este trabajo se configuró en la máquina virtual. Fue fundamental iniciar el nodo principal para verificar las conexiones, esto se realizó lanzando la máquina desde el acceso directo en el escritorio del computador principal. Al abrir la máquina se tuvo que acceder con las credenciales; en una operación normal se deja trabajando la máquina en segundo plano, pero para la comprobación fue preciso acceder a la terminal de comandos y ejecutar la orden *ifconfig*, esto permitió verificar la dirección IP, por defecto es dinámica. Fue necesario pasar la dirección IP a una estática porque se incluye en el código de la ESP32 y si se deja variable es imposible lograr la conexión. Esta configuración se hizo desde los ajustes de red de la máquina virtual y solo es necesario hacerlo una vez.

Al energizar la ESP32 se inicia la conexión wifi y luego se conecta al bróker, para visualizar que ocurrieran las conexiones fue conveniente conectar la tarjeta a un monitor serial, por comodidad se utilizó el monitor que ofrece PlatformIO. En la Figura 30 se plasma lo que ocurrió cuando se energizó la tarjeta, allí se muestran dos intentos de conexión a la red, estos se representan con puntos, luego se logró la conexión y se mostró la dirección IP que le fue otorgada por el enrutador a la tarjeta. Establecido el puente inalámbrico entre la ESP32 y el bróker se realizó la conexión MQTT de forma positiva. Los códigos que permiten esto están disponible en el apartado de anexos.



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

..
Conectado a red WiFi!
Direccion IP:
192.168.1.69
Intentando Conexión MQTT...
Conectado!
```

Figura 30. Conexión de ESP32 como cliente MQTT lograda.

Del lado del navegador ocurre algo similar, pero la conexión wifi es intrínseca al proceso, dejando únicamente la tarea de conexión al bróker. Por su naturaleza las páginas web no pueden soportar el protocolo MQTT, por esto se emplea WebSocket para recubrir los mensajes del primer protocolo y enviarlos en la capa TCP, esto lo realiza la librería Paho y

la habilitación de la máquina virtual para soportar WebSocket. Con esto claro, la conexión al bróker es ejecutada por el código JavaScript *iot-mqtt.js*. La conexión se refleja en la consola del inspector de la página web, la Figura 31 muestra el mensaje de conexión exitosa. Cuando no ocurre la conexión se informa de manera similar y se intenta cada 10s.

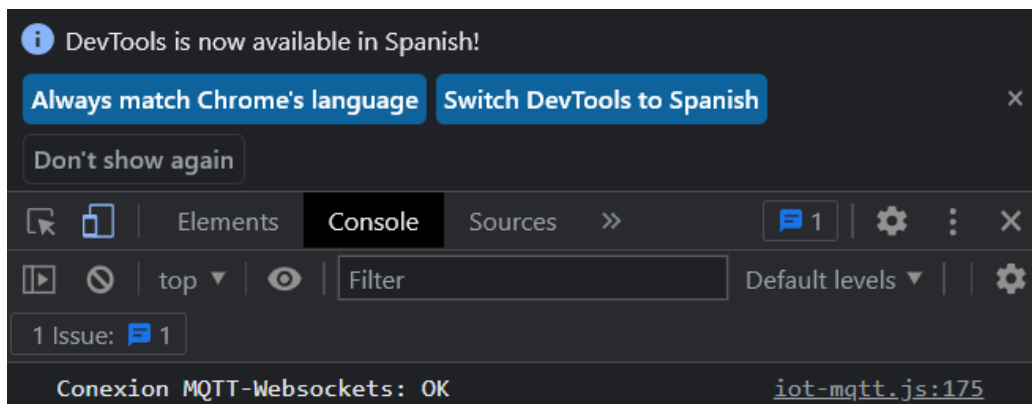


Figura 31. Conexión de página como cliente MQTT lograda.

Las verificaciones realizadas demuestran que ambos clientes se conectan de forma exitosa al bróker y quedan habilitados para la suscripción y publicación de mensajes en los temas correspondientes.

6.2.2. Prueba de la instrumentación

La siguiente prueba se realizó para verificar que la instrumentación está censando de forma correcta los valores de los sensores y que además está publicando la información en los temas correspondientes. Esta parte del sistema funciona como sigue: inicialmente la tarjeta se suscribe a los tópicos que se relacionan con modificación de variables (actuadores), donde se recibe información externa. Entiéndase por externa la información que no es capturada por la propia tarjeta. Luego de la suscripción empieza la publicación continua de los valores que adquiere la tarjeta, estos son los sensores de humedad, presión y voltaje. La información externa que se recibe llega como interrupción y se descifra inmediatamente, luego de ejecutar la acción ordenada se continúa con la publicación de los valores censados.

Para verificar que el funcionamiento es correcto se conecta la tarjeta ESP32 al monitor serie de PlatformIO. La Figura 32 muestra la información publicada por la tarjeta. Allí se evidencia que la primera acción fue la suscripción a cinco temas y luego inició la publicación de los diez temas restantes, como se esperaba. Los paquetes se van publicando cada 10s, iniciando con el valor de la humedad en red a través del tema "IoT/Hum" y finalizando con el valor del estado del paro de emergencia del compresor secundario a través del tema "IoT/ParoC2".

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

..
Conectado a red WiFi!
Direccion IP:
192.168.1.69
Intentando Conexión MQTT...
Conectado!
Suscripción a IoT/AlvTk: OK
Suscripción a IoT/OnC1: OK
Suscripción a IoT/OffC1: OK
Suscripción a IoT/OnC2: OK
Suscripción a IoT/OffC2: OK
Enviando: [IoT/Hum]62.0
Enviando: [IoT/Ptk]15.0
Enviando: [IoT/Pr1]0.0
Enviando: [IoT/Pr2]41.0
Enviando: [IoT/EnerC1] OFF
Enviando: [IoT/OnOffC1] OFF
Enviando: [IoT/ParoC1] OFF
Enviando: [IoT/EnerC2] OFF
Enviando: [IoT/OnOffC1] OFF
Enviando: [IoT/ParoC2] ON
Enviando: [IoT/Hum]62.0
Enviando: [IoT/Ptk]16.0
Enviando: [IoT/Pr1]0.0
Enviando: [IoT/Pr2]40.0
Enviando: [IoT/EnerC1] OFF
Enviando: [IoT/OnOffC1] OFF
Enviando: [IoT/ParoC1] OFF
Enviando: [IoT/EnerC2] OFF
Enviando: [IoT/OnOffC1] OFF
Enviando: [IoT/ParoC2] ON
```

Figura 32. Publicaciones de la ESP32.

Cuando el sistema opera con normalidad, no es necesaria la conexión al monitor serie, los eventos se siguen actualizando de forma automática. Que se haya realizado la suscripción y publicación como se esperaba comprueba que la instrumentación cumple su función.

6.2.3. Prueba de visualización de datos

La siguiente prueba se realizó para verificar que la página web puede recibir los mensajes de los tópicos a los que se suscribe, generar publicaciones y reflejar en gráficas y contenedores los valores de los sensores. Inicialmente, se verificó la recepción de los mensajes, para esto se ingresó a la consola del inspector de la página web y se confirmó que las publicaciones hechas por la ESP32 llegan a la página web. La Figura 33 muestra las publicaciones realizadas por la tarjeta.

Luego se verificó que la información se plasma de forma correcta en las gráficas y contenedores de estado, para esto se ingresó a la página web en la opción Dashboard. La Figura 34 muestra la visualización de los datos.

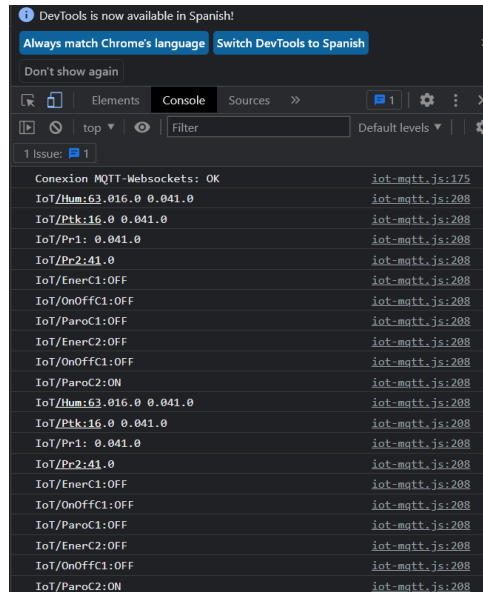


Figura 33. Datos recibidos en la página web.

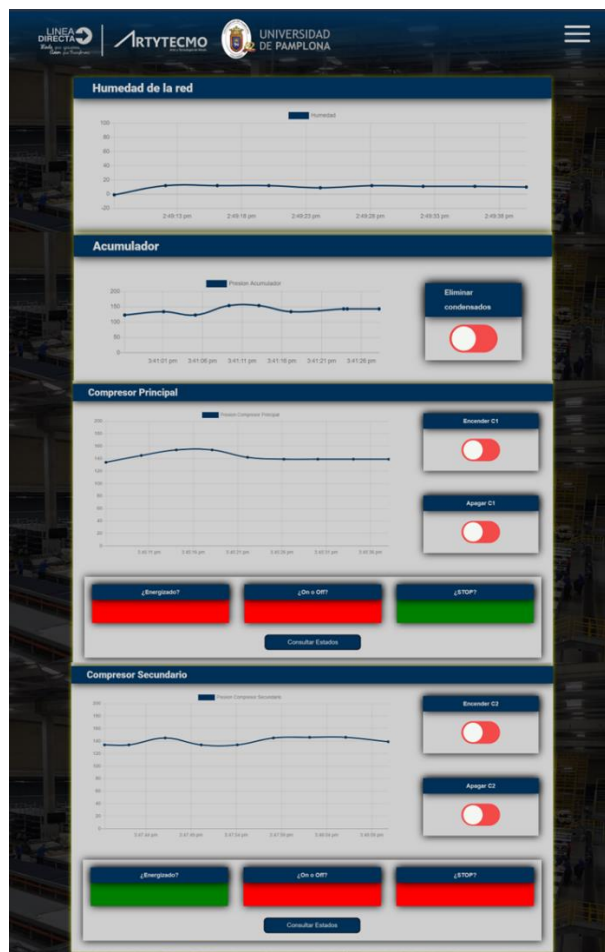


Figura 34. Visualización de datos en la página web.

6.2.4. Prueba de funcionamiento integral

Para verificar que el sistema en su totalidad funciona correctamente, se requirió la división en dos estados de trabajo, uno normal y otro especial. Para la condición de funcionamiento normal se diseñó una matriz que asocia preguntas a ciertos criterios que se deben realizar de forma secuencial, si todo el sistema funciona de forma correcta, las respuestas a estas preguntas deben estar en su totalidad marcadas con un sí, en caso contrario el sistema posee un error en algún punto y debe consultarse la guía de usuario para resolverlo. La prueba consistió en evaluar la matriz creada, los resultados obtenidos se plasman en la Tabla 9. Para comprobar encendido y apagado de compresores se alternaron el principal y secundario para no afectar la producción. Los resultados son satisfactorios y evidencian el correcto funcionamiento del sistema en un estado normal.

Tabla 9. Matriz de evaluación para estado normal.

Criterio	Pregunta	Respuesta
Accionar totalizador del gabinete central	¿Inicia ventilador de fuente y enciende led en ESP32?	Sí
Abrir pagina web desde dispositivo conectado a la red local	¿Se encuentra la dirección buscada?	Sí
Inicar sesión	¿Se accede al Dashboard?	Sí
Abrir consola del navegador	¿Ocurre la conexión MQTT?	Sí
Inspección del Dashboard	¿Pasados 10s se registra el primer valor en las graficas de presion y humedad?	Sí
Prueba de alivio de condensados	¿Se activa la válvula de alivio al mover el interruptor de la página web?	Sí
Prueba de encendio de compresor	¿Se enciende el compresor al mover el interruptor de la página web?	Sí
Prueba de apagado de compresor	¿Se apaga el compresor al mover el interruptor de la página web?	Sí
Prueba de consulta de estados	¿Se modifican los indicadores al consultar los estados?	Sí

El funcionamiento especial del sistema ocurre ante un apagón eléctrico, esta es una condición rara y corta porque se cuenta con un grupo electrógeno de respaldo y un sistema de alimentación ininterrumpida para los equipos principales (cortadoras automáticas), los demás equipos se ven afectados por la conmutación que acontece cuando sale la electrificadora y entra la planta o viceversa. Los compresores se apagaban durante este intercambio y requerían reinicio manual, obligando a los auxiliares de mantenimiento a hacer presencia en la sala de compresores, ocupando un tiempo importante en el desplazamiento.

Lo anterior hace importante evaluar el comportamiento de lo implementado para determinar si se logró una mejora en este aspecto. Para ello se realizó la Tabla 10. Allí se plasman los criterios que se siguieron para realizar la evaluación, se anotaron las observaciones para discutir la eficacia del sistema. Se aclara que esta tabla solo se utiliza en

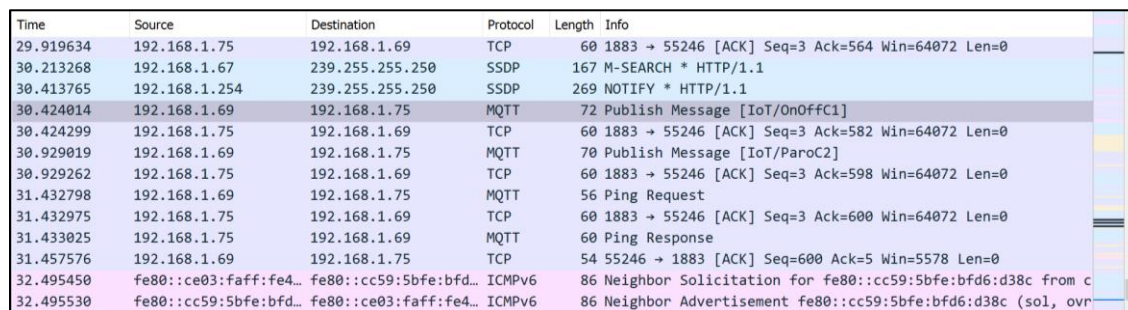
este apartado y no se requiere en la operación cotidiana del proceso o cuando ocurra un apagón.

Tabla 10. Matriz de evaluación para estado especial.

Criterio	Comportamiento observado
Actualización de página web 30s después de reconexion	La página recarga adecuadamente mostrando directamente el Dashboard.
Eventos en las gráficas de variables	Después de 10s de actualizada la página ocurre la actualización automática de los valores en las gráficas.
Actualización de estados (inicial)	El botón consultar estado responde y en ambos compresores evidencia un estado energizado, motor principal apagado y botón de paro sin activación.
Activación de compresores	Los botones encender compresor ponen en marcha ambos compresores, se observa que inmediatamente entra el compresor principal a funcionar para recuperar la carga perdida.
Actualización de estados (final)	Al consultar estados de nuevo, se actualiza con estados energizado, encendido y sin paro.

Para poder realizar las observaciones registradas en la Tabla 10 se esperó a que el fallo eléctrico ocurriera. Estas observaciones son positivas y demuestran la mejora conseguida con el sistema, todo el reinicio de los compresores se efectúa desde la oficina de Mantenimiento Industrial.

Como agregado en la validación integral, se realizó una inspección de los paquetes que son enviados vía wifi empleando Wireshark, un analizador de protocolos manejado normalmente para analizar datos en redes de comunicaciones. Una parte de la captura realizada se muestra en la Figura 35. De los paquetes capturados solo interesa los relacionados con la dirección IP 192.168.1.75 que corresponde al bróker, se puede identificar que unos se envían empleando el protocolo MQTT y otros el TCP, estos últimos son los encapsulados con WebSocket.



Time	Source	Destination	Protocol	Length	Info
29.919634	192.168.1.75	192.168.1.69	TCP	60	1883 → 55246 [ACK] Seq=3 Ack=564 Win=64072 Len=0
30.213268	192.168.1.67	239.255.255.250	SSDP	167	M-SEARCH * HTTP/1.1
30.413765	192.168.1.254	239.255.255.250	SSDP	269	NOTIFY * HTTP/1.1
30.424014	192.168.1.69	192.168.1.75	MQTT	72	Publish Message [IoT/OnOffC1]
30.424299	192.168.1.75	192.168.1.69	TCP	60	1883 → 55246 [ACK] Seq=3 Ack=582 Win=64072 Len=0
30.929019	192.168.1.69	192.168.1.75	MQTT	70	Publish Message [IoT/ParoC2]
30.929262	192.168.1.75	192.168.1.69	TCP	60	1883 → 55246 [ACK] Seq=3 Ack=598 Win=64072 Len=0
31.432798	192.168.1.69	192.168.1.75	MQTT	56	Ping Request
31.432975	192.168.1.75	192.168.1.69	TCP	60	1883 → 55246 [ACK] Seq=3 Ack=600 Win=64072 Len=0
31.433025	192.168.1.75	192.168.1.69	MQTT	60	Ping Response
31.457576	192.168.1.69	192.168.1.75	TCP	54	55246 → 1883 [ACK] Seq=600 Ack=5 Win=5578 Len=0
32.495450	fe80::ce03:faff:fe4...	fe80::cc59:5bfe:bfd...	ICMPv6	86	Neighbor Solicitation for fe80::cc59:5bfe:bfd6:d38c from c
32.495530	fe80::cc59:5bfe:bfd...	fe80::ce03:faff:fe4...	ICMPv6	86	Neighbor Advertisement fe80::cc59:5bfe:bfd6:d38c (sol, ovr

Figura 35. Paquetes capturados con Wireshark.

6.3. Guía de usuario

La guía de usuario constituye un documento que permite al personal que utilice el sistema entender la finalidad de este, aprender a manipular los elementos que lo componen y arreglar problemas comunes que se presenten en la interacción con el sistema IoT. Para crearla se inició una búsqueda de guías similares para afianzar una estructura generalizada que sirva como modelo, sin embargo, a nivel investigativo, trabajos similares no contemplan la creación de una guía de usuario porque el resultado de la investigación, llámese artículo o tesis, contiene la información necesaria para manipularlo. A consideración propia, los usuarios finales no requieren el conocimiento total del proyecto, solo deben entender el funcionamiento y conceptos generales.

Otras guías de usuario de productos tecnológicos más sofisticados incluyen elementos como tabla de contenido, iniciación en el sistema, funciones principales y solución de errores. Estos parámetros son los tenidos en cuenta para la redacción de la guía de usuario disponible en anexos. Con el equipo de Mantenimiento Industrial de ARTYTECMO se discutió una primera versión que presentaba algunos errores, estos se corrigieron para obtener el resultado final mostrado en este documento.

CONCLUSIONES

Durante la investigación se encontró que la implementación de nuevas tecnologías de la información aporta beneficios destacables en los procesos industriales. De forma más específica, tecnologías como el Internet de las Cosas revolucionan la manera en que se opera y/o administran procesos productivos. Poner en marcha estas herramientas, por su novedad, puede representar una inversión alta si el número de variables censadas es grande y se requiere tratamiento de datos o registro en bases de datos. Sin embargo, es posible implementar sistemas IoT eficientes con requerimientos bajos, esto se demostró en este documento.

El sistema que se implementó garantiza monitoreo y activación remota de la sala de compresores. Mediante la primera característica se redujeron los daños ocasionados por presiones bajas o exceso de humedad en la red. Con la segunda se redujeron los tiempos de intervención en sala, generando mayor disponibilidad del personal de mantenimiento. Este desarrollo comprendió pequeños logros que sumaron para que la función final fuera exitosa, a continuación se mencionan.

El acondicionamiento de la sala de compresores se logró realizar empleando componentes modulares de bajo costo, fácil instalación y reemplazo, realizando para ello un reconocimiento dimensional y funcional de la sala con el que se pudo determinar qué componentes se requerían y cuáles desempeñarían una mejor función. Dentro de esta selección fue fundamental la comparación realizada entre tarjetas de desarrollo comunes para definir que la mejor opción para el cumplimiento del objetivo central es la ESP32. Con respecto a esta tarjeta, vale la pena mencionar que su característica de voltaje bajo (3.3V) representó complicaciones para la implementación porque hizo necesaria la instalación de módulos adicionales para convertir los rangos, afectando en mayor medida las señales analógicas.

Con la creación de una página web se logró realizar la visualización de los datos obtenidos y modificar estados de la sala relacionados con la activación de la válvula de alivio del tanque acumulador y los motores principales. Almacenarla en un servidor local garantizó una capa de seguridad fuerte respaldada por el equipo de Soporte y Sistemas de ARTYTECMO. Denegar el acceso a los usuarios que no posean las credenciales consiguió que únicamente el personal capacitado accediera a la página web y por ende a la manipulación de una zona tan importante. Incluir los manuales, listas de partes y hojas de datos de los compresores en un apartado de la página hizo más cómoda las consultas, tradicionalmente lo hacían desde la biblioteca de manuales ubicada en la oficina de Mantenimiento Industrial. Por otro lado, se menciona que la actualización de los valores censados es cada 10s, este tiempo se puede reducir, pero no se recomienda porque el costo computacional es muy alto y la información no varía con tanta velocidad.

Haber realizado la conexión a través del protocolo MQTT, facilitó en gran medida la transferencia de la masiva cantidad de datos a través de un sistema embebido con capacidades de procesamiento restringidas como lo es la ESP32. Asimismo, no hubiera sido posible la comunicación entre el bróker y la página web sin implementar el recubrimiento con WebSocket.

La guía de usuario desarrollada permite a los auxiliares de Mantenimiento de la empresa familiarizarse de forma sencilla y rápida con el sistema. Se consiguió plasmar el paso a paso para iniciar el sistema desde cero, explicar las funciones principales y dar solución a los errores más comunes presentados durante quince días de seguimiento.

TRABAJOS FUTUROS

Para maximizar la herramienta a futuro se pueden presentar los resultados obtenidos al Jefe de Área para conseguir la aprobación y apoyo para implementar una base de datos segura donde puedan ser almacenados los datos recolectados, esto con el fin de realizar análisis posteriores que permitan determinar el consumo energético de los compresores y encontrar medidas para reducirlo.

Implementar un sistema de alarmas mediante mensajes de texto sobre Telegram o WhatsApp que informe de primera mano a los auxiliares si se detectó una caída de presión, un incremento excesivo en la humedad o si los compresores perdieron su alimentación y deber ser reiniciados.

Finalmente, se recomienda a futuro implementar una pantalla en el tablero central que haga las veces de monitor serial para darle mayor autonomía al sistema y eliminar la presencia de un computador para solucionar errores de comunicación entre la tarjeta y el bróker.

REFERENCIAS

- [1] M. Miraz, *et al.* “A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)”. En *IEEE Internet Technologies and Applications*, pp. 219–224, 2015.
- [2] A. Solé. *Neumática e hidráulica*. Ed 2, Marcombo, 2011.
- [3] C. Gómez. “Automatización del corte en el sector textil”. Tesis de pregrado, Universidad EAFIT, 2019.
- [4] M. Miraz, *et al.* “A Review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT)”. En *IEEE Internet Technologies and Applications*, pp. 219–224, 2015.
- [5] L. Xu y W. He. “Internet of things in industries: A survey”. En *IEEE Transactions on Industrial Informatics*, pp. 2233–2243, 2014.
- [6] ITU. ITU Internet Report 2005: The Internet of Things. [En línea]. Disponible en: www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf 2005.
- [7] M. Wu, T. Lu, F. Ling, J. Sun y H. Du. “Research on the architecture of Internet of Things”. En *3rd international conference on advanced computer theory and engineering*, vol. 5, pp. 465-484, 2010.
- [8] R. Peña, G. Chanchí y M. Campo. “Sistema IoT para la monitorización de niveles de ruido en zonas aledañas al aeropuerto de Cartagena de Indias”. En *Revista Ibérica de Sistemas e Tecnologías de Informação*, pp. 257-271, 2021.
- [9] A. Ochoa, L. Cangrejo y T. Delgado. “Alternativa Open Source en la implementación de un sistema IoT para la medición de la calidad del aire”. En *Revista Cubana de Ciencias Informáticas*, vol. 12, pp. 189-204, 2018.
- [10] M. Cepeda y J. Hernández. “Diseño de sistema domótico basado en IoT utilizando microcontrolador ESP32 mediante protocolo de comunicación MQTT, para paciente con diplegia braquial”. Tesis de maestría, Tecnológico Nacional de México, 2020.
- [11] C. Bell. *Beginning IoT Projects*. Ed 1, Apress, 2021.
- [12] A. Lliso. “Uso de MQTT para el control de dispositivos de IoT”. Tesis doctoral, Universidad Politécnica de Valencia, 2021.
- [13] Z. Shelby, K. Hartke y C. Bormann. *The constrained application protocol (CoAP)*. Ed 1, 2014.
- [14] B. Lee, T. Fielding y H. Frystyk. *Hypertext transfer protocol--HTTP/1.0*. Ed 1, 1996.
- [15] Equipo Vértice. *Diseño básico de páginas web en HTML*. Ed 1, Vértice, 2009.

- [16] Á. Arias, M. Arias y A. Durango. *Curso de Desarrollo Web*. Ed 2, IT Campus Academy, 2016.
- [17] J. Pérez. *Introducción a JavaScript*. Ed 1, Libros Web, 2019.
- [18] Node RED. Acerca de nosotros. [En línea]. Disponible en: <https://nodered.org/> 2022
- [19] ThingSpeak. Acerca de nosotros. [En línea]. Disponible en: <https://thingspeak.com/2022>.
- [20] Arduino IoT Cloud. Acerca de nosotros. [En línea]. Disponible en: <https://docs.arduino.cc/arduino-cloud/> 2022.
- [21] M. Farooq, S. Riaz, A. Abid, T. Umer y Y. Zikria. “Role of IoT technology in agriculture: A systematic literature review”. En *Electronics*, vol. 9, pp. 319, 2020.
- [22] S. Selvaraj y S. Sundaravaradhan. “Challenges and opportunities in IoT healthcare systems: a systematic review”. En *SN Applied Sciences*, vol. 2, pp. 1-8, 2020.
- [23] T. Song, R. Li, B. Mei, J. Yu, X. Xing y X. Cheng. “A privacy preserving communication protocol for IoT applications in smart homes”. En *IEEE Internet of Things Journal*, vol. 4, pp. 1844-1852, 2017.
- [24] K. Mohamed. “The era of Internet of Things: towards a smart world”. En *The Era of Internet of Things*, pp. 93-108, 2019.
- [25] P. Díez. *Compresores*. Ed 1, Universidad de Cantabria, 2000.
- [26] R. Ríos. “Prototipo de monitoreo de calidad de condiciones ambientales a bajo costo con tecnologías IOT”. Tesis de pregrado, Universidad de la Salle, 2020.
- [27] J. Efro. “Evaluación de la eficiencia energética del sistema de aire comprimido en una clínica provada de alta complejidad ubicada en la ciudad de Barranquilla”. Tesis de pregrado, Universidad de la Costa, 2019.
- [28] J. Rojas. “Optimización de sistema de aire comprimido en CREATUM SA”. Tesis de pregrado, Instituto Tecnológico de Medellín, 2018.
- [29] L. Sotelo, L. Londoño, A. Botero y F. Hurtado. “Sistema de monitoreo y control remoto usando IOT para un regulador de presión”. En *Scientia et technica*, vol. 22, pp. 391-397, 2017.
- [30] J. Londoño. “Implementación del monitoreo remoto de fallas y alertas tempranas en el sistema de aire comprimido de la compañía de galletas Noel SAS”. Tesis de pregrado, Instituto Tecnológico de Medellín, 2017.
- [31] J. Cobeñas. “Diseño de un sistema de monitoreo y supervisión de eficiencia energética del consumo de aire comprimido en cilindros neumáticos aplicando tecnología IIoT bajo el estándar ISO 50001”. Tesis de pregrado, Universidad Nacional de Piura, 2020.

- [32] A. Bohórquez. “Sistema automático para control de proceso y generación de aire comprimido con visualización online en cervecería nacional planta Cumbayá”. Tesis de pregrado, Universidad de Israel, 2020.
- [33] A. Murty, M. Satyanarayana y I. Devi. “Compressor Health Monitoring using IOT”. En *International Journal of Mechanical and Production Engineering Research and Development*, vol. 8, pp. 117-124, 2019.
- [34] D. Mourtzis, J. Angelopoulos y N. Panopoulos. “Design and development of an IoT enabled platform for remote monitoring and predictive maintenance of industrial equipment”. En *Procedia Manufacturing*, vol. 54, pp. 166-171, 2021.
- [35] J. Lee, Y. Su y C. Shen. “A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and wifi”. En *IECON*, pp. 46-51, 2007.

ANEXOS

Anexo A: Apariencias del aplicativo



Figura 1 Anexo A. Vista desde el celular de “Acerca del TG”.

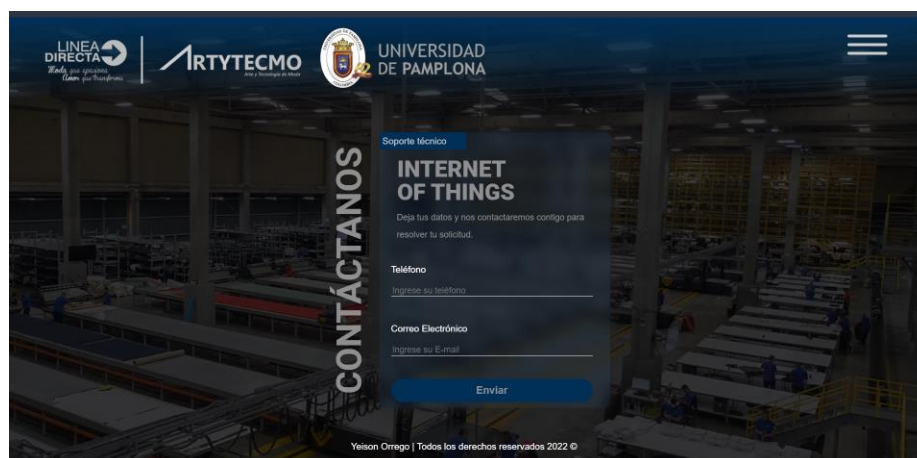


Figura 2 Anexo A. Vista desde el computador de “Contáctanos”.

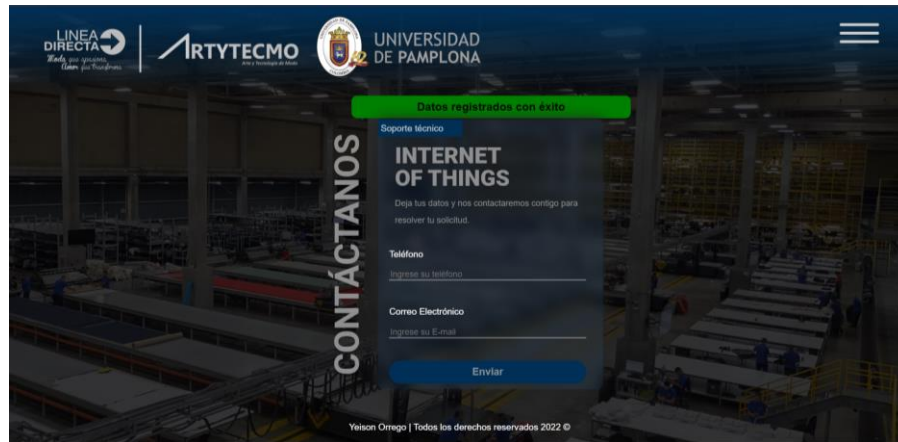


Figura 3 Anexo A. Vista desde el computador de “Contáctanos” cuando el registro de datos fue exitoso.

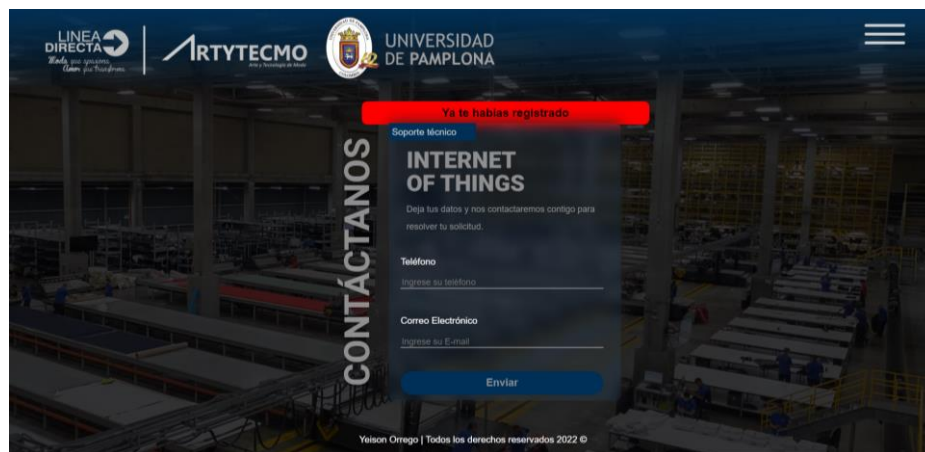


Figura 4 Anexo A. Vista desde el computador de “Contáctanos” cuando se intenta un registro de datos ya almacenados.

Anexo B. Repositorio de códigos del sistema

Por comodidad, los códigos pertinentes al sistema se han alojado en un repositorio GitHub, se debe tener en cuenta que al ser publico no es posible poner la ultima versión, pero una versión media sirve a los revisores para entender la estructura y funcionamiento del sistema.

Enlace del repositorio: <https://github.com/Uchimatron/Trabajo-de-Grado-IoT.git>



Sistema IoT Sala de Compresores ARTYTECMO

Guía del usuario



UNIVERSIDAD
DE PAMPLONA

TABLA DE CONTENIDO

INTRODUCCIÓN	1
ARRANQUE DEL SISTEMA	2
FUNCIONES PRINCIPALES	5
Graficas Real Time	5
Interruptores	5
Contenedores de estado	5
Documentación	6
Acerca del TG	6
Contáctanos	6
Salir	6
SOLUCIÓN DE ERRORES	7
No se actualizan los datos	7
Web sin conexión MQTT	7
Máquina virtual con IP diferente	7
ESP32 sin conexión MQTT	7

INTRODUCCIÓN

El sistema IoT implementado en la sala de compresores constituye una herramienta útil que mejora los tiempos de intervención en operaciones rutinarias como el reinicio de los compresores luego de apagones eléctricos y la desconexión que se realiza finalizando semana. También permite realizar un monitoreo constante sobre las dos variables más importantes de la sala de compresores, la presión y la humedad; con esto se evitan problemas por bajas presiones o exceso de humedad en el aire comprimido que se emplea para la limpieza de equipos durante los mantenimientos preventivos que se realizan a diario.

Entender el funcionamiento de cada componente del sistema IoT y solucionar errores comunes le permitirá sacar el máximo provecho de la herramienta, esta Guía lo llevará de la mano a través de tres secciones que le permitirán, como usuario técnico, afianzar sus conocimientos del sistema de forma sencilla.



UNIVERSIDAD
DE PAMPLONA

ARRANQUE DEL SISTEMA

Para iniciar el sistema encienda el computador principal, este es el que se sitúa junto a la ventana contigua al taller. Ingrese con las credenciales de Mantenimiento Industrial, las encuentra escritas en el tablero principal con la referencia “Datos PC”. Espere a que se estabilice el computador y diríjase al escritorio, allí encontrará un icono similar al que muestra la Figura 1.



Figura 1. Icono de máquina virtual.

Pulse dos veces sobre el icono y espere a que arranque la máquina virtual, esto puede llegar a tardar hasta un minuto. Cuando aparezca la ventana que se muestra en la Figura 2 diligencie el recuadro con la contraseña definida para la máquina virtual, si la desconoce consúltela con el Coordinador de Mantenimiento de la sede.

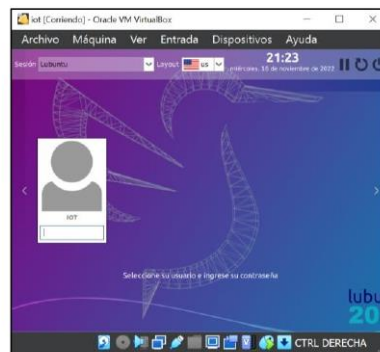


Figura 2. Inicio de la máquina virtual.

Una vez validada la contraseña, ubíquese en el escritorio número uno de la máquina virtual y realice la siguiente combinación de teclas: Ctrl + Alt + T, esto le abrirá la caja de comandos donde deberá digitar *ifconfig* y presionar Enter para verificar que la dirección IP sea 192.168.1.75. Si es correcto cierre la caja de comandos y minimice la ventana de la máquina, está listo para el siguiente paso.

En caso de que la dirección sea diferente imite los siguientes pasos. Vaya al escritorio número uno. De clic derecho sobre el icono de Conexión, lo encuentra en la esquina inferior derecha, se asemeja a un conector RJ45. En el menú que se despliega seleccione “Editar las conexiones...”. Despliegue el menú de Ethernet y de doble clic en “Conexión Cableada 1”.

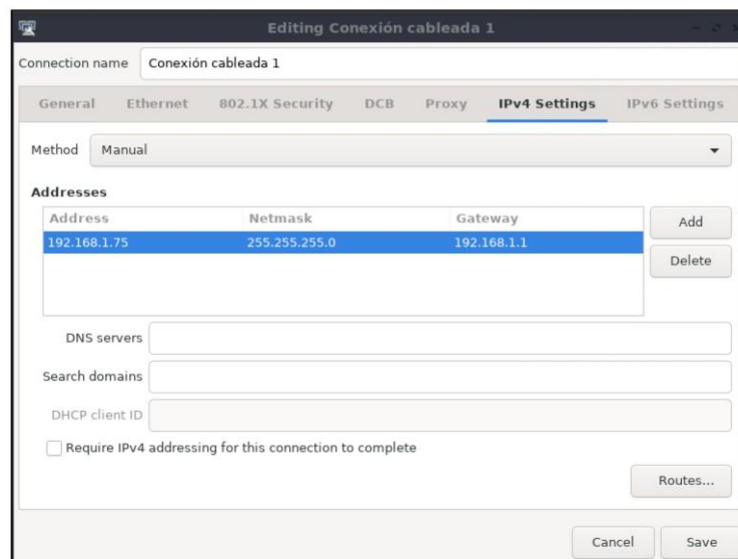


Figura 3. Modificación de IP a estática.

Diríjase a las opciones de configuración IPv4 y en direcciones agregue una nueva con dirección 192.168.1.75, máscara de red 255.255.255.0 y Gateway 192.168.1.1. Guarde las modificaciones y cierre las ventanas abiertas. Reinicie la conexión cableada y minimice la ventana de la máquina.

Lo realizado hasta el momento habilitó el bróker de la comunicación, ahora corresponde habilitar los clientes. Para la página web, diríjase al navegador de confianza y copie la dirección <http://localhost:84/iot/login.html>, esto le abre la página web para su inicio de sesión. Ingrese las credenciales correspondientes para pasar al Dashboard. Abra el inspector de la página web y luego la consola, asegúrese de que el mensaje “Conexión MQTT-WebSocket: OK” aparezca en primera línea, de lo contrario diríjase a la solución de problemas.

Para habilitar la ESP32 debe dirigirse hasta la sala de compresores y activar el interruptor del tablero central, lo ideal es que este interruptor esté siempre activado para que el sistema cumpla su función.

Actualice la página web en el Dashboard y empiece la monitorización de variables, si lo requiere encienda los compresores.

FUNCIONES PRINCIPALES

La pagina web está equipada con una serie de elementos que le permitirán obtener información y modificar parámetros de la sala de compresores, a continuación, conocerá los detalles de los más importantes.

Gráficas Real Time

Podrá encontrar cuatro de estas en el apartado Dashboard, tres para presión y una para humedad. Se refrescan de forma automática cada 10s y brindan los datos relacionados con las variables mencionadas. Si desea conocer el valor exacto ubique el cursor sobre el extremo derecho de la curva, en un recuadro emergente observará la fecha del dato y el valor.

Interruptores

Encontrará cinco de estos en el Dashboard, su función es modificar parámetros de la sala de compresores. Su activación recomendada es de 20s, a modo de ejemplo, suponga que desea encender el compresor principal, diríjase a “Encender C1” y cámbiele el estado a ON, espere 20s y regréselo al estado original, esto se traduce en una activación segura del motor principal del compresor.

Contenedores de estado

Encontrará seis de estos en el Dashboard, su función es capturar el estado asociado, puede estar relacionado con la energía, el encendido o el paro de emergencia, para refrescar la información debe pulsar el botón “Consultar Estados”, si se torna de color verde quiere decir que la característica asociada está activa y se permanece en rojo está desactivada.



UNIVERSIDAD
DE PAMPLONA

Documentación

Esta opción esta disponible en el menú desplegable, al abrirla se accede a una ventana donde podrá abrir los principales documentos asociados a la sal de compresores como listas de partes, hojas de datos y manuales.

Acerca del TG

Esta opción le permite acceder a información general relacionada con la motivación de desarrollar esta implementación, el autor y la institución a la que pertenece.

Contáctanos

En esta función puede registrar datos de contacto para que el desarrollador se ponga en contacto con usted y lo oriente en caso de que la presente Guía no sea suficiente para resolver sus problemas. Asegúrese de no registrar datos repetidos porque le saldrá un error.

Salir

Con esta función el inicio de sesión se elimina y se regresa a la página de validación de credenciales, normalmente no debe usarla, pero se recomienda si por algún evento especial hay personal no capacitado con acceso al computador principal.



UNIVERSIDAD
DE PAMPLONA

SOLUCIÓN DE ERRORES

Por la estructura del sistema los errores son reducidos, sin embargo, en las siguientes líneas encontrará la solución a los problemas que puede llegar a tener.

No se actualizan los datos

Si está utilizando el sistema y de forma inesperada los datos de las graficas dejan de actualizarse revise su conexión de internet, asegúrese de que está en la red local de la empresa y no con datos móviles, luego refresque la página. Si el problema persiste vaya al siguiente error.

Web sin conexión MQTT

Si la pagina no actualiza las gráficas y la solución anterior no funcionó, abra la consola del inspector y verifique la desconexión del bróker, espere a que ocurran dos reintentos, si persiste el problema el bróker ha perdido su conexión. Revise que la máquina virtual sigue funcionando en segundo plano y está en el escritorio número uno. Si el problema persiste debe reconfigurar la dirección original del bróker.

Máquina virtual con IP diferente

Si sospecha que el bróker cambio su dirección IP debe dirigirse a la máquina virtual y ejecutar los pasos mencionados en el segundo párrafo de la página 3. Esto debe reestablecer su conexión.

ESP32 sin conexión MQTT

Si su página web presenta conexión MQTT, pero aun así los valores no se actualizan, debe abrir la consola del inspector y verificar que los mensajes están



UNIVERSIDAD
DE PAMPLONA

llegando, de lo contrario el problema lo posee la tarjeta. Diríjase a la sala de compresores y reiníciela. Si continúa el error, debe apagar el panel principal y retirar la tarjeta para cargarle la memoria que está disponible en el disco Copérnico en la carpeta IOT_Compresores, recuerde que esta memoria se abre con PlatformIO o en su defecto el IDE de Arduino. Antes de cargarla asegúrese que los datos de la red wifi estén actualizados. Vuelva a ubicar la tarjeta en el tablero y reinicie la pagina web, esto debe solucionar su problema.