

**Sistema de Monitoreo de Temperatura para la conservación de la cadena de frio en plantas  
Industriales**

Javier Antonio Hernández Arango

Asesor

Diego Fernando Nava Cuevas

Universidad Nacional Abierta y a Distancia UNAD  
Escuela de Ciencias Básicas de la Tecnología e Ingeniería

Ingeniería Electrónica

2025

## **Agradecimiento**

Agradezco a Dios por darme la vida, la fuerza y la oportunidad de seguir aprendiendo y creciendo cada día. A la Universidad Nacional Abierta y a Distancia por brindarme el espacio y los recursos necesarios para mi formación y desarrollo profesional. A mis docentes, especialmente a aquellos que han acompañado de cerca de mi proceso, guiándome con su conocimiento, paciencia y dedicación. Su apoyo y enseñanzas han sido fundamentales en este camino.

También quiero expresar mi más profundo agradecimiento a mi familia, cuyo apoyo incondicional, amor y comprensión han sido mi mayor fortaleza. Gracias por estar siempre a mi lado, impulsándome a seguir adelante y alcanzar mis metas.

A todos los que, de alguna manera, han contribuido a la culminación de este proyecto, mi más sincera gratitud.

## **Dedicatoria**

A mis padres, por enseñarme desde niño que el esfuerzo, la constancia y la honestidad son las bases para alcanzar cualquier meta en la vida.

A mi hija, por su ternura, su apoyo y por ser una fuente constante de motivación para seguir adelante.

Y en especial, a mi esposa, por su amor incondicional, su apoyo constante y su compañía en cada paso de este camino. Gracias por estar a mi lado en los momentos difíciles, por creer en mí siempre, y por ser mi fuerza.

## Resumen

Este proyecto desarrolla un sistema de monitoreo que detecta en tiempo real las pérdidas de temperatura en la cadena de frío en plantas industriales. El sistema identifica variaciones en la temperatura de las cavas de conservación, garantizando condiciones óptimas para los productos que requieren almacenamiento controlado y contribuyendo a su adecuada conservación.

El sistema genera una alarma cuando la cadena de frío se encuentra fuera de los parámetros establecidos y envía una notificación por correo electrónico al operador. Además, permite un monitoreo continuo a través de gráficas en la plataforma ThingSpeak, lo que facilita la visualización en tiempo real del comportamiento de la temperatura. Esta información le permite al operador intervenir de forma oportuna para evitar pérdidas significativas o daños en los productos sensibles a las variaciones térmicas.

El enfoque principal del proyecto es garantizar que los productos almacenados en los cuartos fríos o cavas mantengan una cadena de frío confiable.

La implementación de este sistema demuestra que el uso de tecnología IoT en la industria alimentaria permite un monitoreo en tiempo real de las condiciones de conservación, optimiza la gestión de la cadena de frío, previene pérdidas por variaciones de temperatura, mejora la eficiencia operativa, asegura el cumplimiento de normativas de seguridad alimentaria y contribuye a la reducción del desperdicio, manteniendo una alta productividad y rentabilidad.

**Palabras clave:** Sensores, Monitoreo, Conservación, Supervisión, Control, Comunicación Vía WI-FI, Control IOT, Sistema Embebido, Programación.

## **Abstract**

This project develops a monitoring system that detects temperature losses in the cold chain of industrial plants in real time. The system identifies temperature variations in the storage chambers, ensuring optimal conditions for products that require controlled storage and contributing to their proper preservation.

The system triggers an alarm when the cold chain deviates from the established parameters and sends an email notification to the operator. Additionally, it allows continuous monitoring through graphs on the ThingSpeak platform, making it easier to visualize temperature behavior in real time. This information enables the operator to intervene promptly to prevent significant losses or damage to temperature-sensitive products.

The focus of the project is to ensure that products stored in cold rooms or chambers maintain a reliable cold chain.

The implementation of this system demonstrates that the use of IoT technology in the food industry enables real-time monitoring of storage conditions, optimizes cold chain management, prevents losses due to temperature fluctuations, improves operational efficiency, ensures compliance with food safety regulations, and contributes to waste reduction—maintaining high productivity and profitability.

**Keywords:** Sensors, Monitoring, Conservation, Supervision, Control, Communication  
Via WI-FI, IOT Control, Embedded System, Programming.

## Tabla de Contenido

Introducción .....	24
Planteamiento Del Problema.....	27
Justificación .....	29
Objetivos .....	31
Objetivo General .....	31
Objetivos Específicos.....	31
Marco Teórico .....	32
Marco conceptual.....	33
Antecedentes .....	63
Antecedes internacionales.....	63
Antecedentes locales .....	66
Diseño metodológico.....	69
Ingeniería Del Proyecto.....	72
Cuarto frio implementado a escala.....	72
Descripción del código Arduino IDE.....	90
Funcionamiento del sistema.....	104
Análisis e Interpretación de Resultados .....	110

Impactos .....	112
Verificación de objetivos .....	113
Conclusiones .....	115
Propuestas de Mejora del Sistema de Monitoreo Temperatura .....	116
Conclusiones .....	120
Referencias Bibliográficas .....	121

## Listado de Tablas

<b>Tabla 1</b> <i>Características DHT11</i> .....	51
<b>Tabla 2</b> <i>Oferta comercial Prototipo Vs Cava 5 toneladas</i> .....	119
<b>Tabla 3</b> <i>Cotización Cableado-cava 5 toneladas</i> .....	120

## Listado de Figuras

<b>Figura 1</b> <i>Cuarto Frio de Conservación</i> .....	33
<b>Figura 2</b> <i>Cuarto Frio de Congelación</i> .....	34
<b>Figura 3</b> <i>Túnel de Congelación</i> .....	35
<b>Figura 4</b> <i>Abatidor de Temperatura</i> .....	35
<b>Figura 5</b> <i>Versión original del NodeMCU</i> .....	39
<b>Figura 6</b> <i>Versión del NodeMCU es la v1.0 V2</i> .....	40
<b>Figura 7</b> <i>Diferencia de la versión v1.0 v2</i> .....	40
<b>Figura 8</b> <i>Tercera generación”, la versión 1.0 V3</i> .....	41
<b>Figura 9</b> <i>Convertor serial CH340G en lugar del CP2102</i> .....	42
<b>Figura 10</b> <i>Resumen de las versiones</i> .....	42
<b>Figura 11</b> <i>Diferencias entre V2 y V3</i> .....	43
<b>Figura 12</b> <i>Resumen con los pines y sus funciones (Llamas, 2022)</i> .....	43
<b>Figura 13</b> <i>Arduino IDE</i> .....	44
<b>Figura 14</b> <i>Conexión de tres dispositivos a un bus de comunicación I2C</i> .....	45
<b>Figura 15</b> <i>Modulo I2C</i> .....	48
<b>Figura 16</b> <i>Pantalla LCD 16X2</i> .....	49
<b>Figura 17</b> <i>Bus de datos</i> .....	50
<b>Figura 18</b> <i>Sensor DHT11</i> .....	51
<b>Figura 19</b> <i>Buzzer</i> .....	53
<b>Figura 20</b> <i>Diodo Emisor de Luz</i> .....	53
<b>Figura 21</b> <i>Protoboard</i> .....	54
<b>Figura 22</b> <i>ThingSpeak</i> .....	57
<b>Figura 23</b> <i>MATLAB Analysis</i> .....	59
<b>Figura 24</b> <i>Application TimeControl</i> .....	60

<b>Figura 25</b> <i>React</i> .....	60
<b>Figura 26</b> <i>Enviando datos a Thingspeak</i> .....	62
<b>Figura 27</b> <i>Diagrama de conexión del dispositivo</i> .....	64
<b>Figura 28</b> <i>Sistema calefacción</i> .....	65
<b>Figura 29</b> <i>Cuarto Frio a escala vista al interior</i> .....	72
<b>Figura 30</b> <i>Cuarto Frio parte frontal (Autoria, 2025)</i> .....	73
<b>Figura 31</b> <i>Cuarto Frio parte lateral (Autoria, 2025)</i> .....	73
<b>Figura 32</b> <i>Diagrama del comportamiento de los componentes del cuarto frio</i> .....	74
<b>Figura 33</b> <i>Diagrama de componentes (Autoria, 2025)</i> .....	75
<b>Figura 34</b> <i>Grafica de comportamiento</i> .....	77
<b>Figura 35</b> <i>Monitoreo Temperatura Análisis Matlab</i> .....	78
<b>Figura 36</b> <i>Monitoreo Humedad análisis Matlab</i> .....	78
<b>Figura 37</b> <i>NodeMcu Esp8266</i> .....	79
<b>Figura 38</b> <i>Supervisión de datos en Thingspeak</i> .....	80
<b>Figura 39</b> <i>Matlab Analysis app</i> .....	81
<b>Figura 40</b> <i>TimeControl</i> .....	82
<b>Figura 41</b> <i>TimeControl</i> .....	82
<b>Figura 42</b> <i>Ajuste temperaturas Alta y Baja</i> .....	83
<b>Figura 43</b> <i>Función React Temperatura Alta (Autoria, 2025)</i> .....	84
<b>Figura 44</b> <i>Función React Temperatura Alta</i> .....	84
<b>Figura 45</b> <i>Función React Temperatura Baja (Autoria, 2025)</i> .....	85
<b>Figura 46</b> <i>Función React Temperatura Baja</i> .....	86
<b>Figura 47</b> <i>Diagrama de flujo</i> .....	87
<b>Figura 48</b> <i>Diagrama de conexión de tarjeta NodeMcu y componentes</i> .....	89
<b>Figura 49</b> <i>Función para enviar datos a thingspeak</i> .....	90

<b>Figura 50</b> <i>Análisis Matlab</i> .....	91
<b>Figura 51</b> <i>Configuración TimeControl (Autoria, 2025)</i> .....	91
<b>Figura 52</b> <i>Configuración 2 TimeControl</i> .....	92
<b>Figura 53</b> <i>Configuración Temperatura Baja y Alta</i> .....	93
<b>Figura 54</b> <i>Configuración Temperatura Alta (Autoria, 2025)</i> .....	93
<b>Figura 55</b> <i>Configuración Temperatura Baja</i> .....	94
<b>Figura 56</b> <i>Alerta diaria 6:30am</i> .....	95
<b>Figura 57</b> <i>Alerta Temperatura fuera de especificación</i> .....	95
<b>Figura 58</b> <i>Evidencias alertas de correo</i> .....	96
<b>Figura 59</b> <i>Evidencia alerta de correo 2</i> .....	96
<b>Figura 60</b> <i>Evidencia alerta correo 3</i> .....	97
<b>Figura 61</b> <i>Función de alertas locales visuales y sonoras</i> .....	98
<b>Figura 63</b> <i>Función de alertas locales visuales y sonoras</i> .....	99
<b>Figura 64</b> <i>Función de alertas locales visuales y sonoras</i> .....	99
<b>Figura 65</b> <i>Configuración I2C</i> .....	100
<b>Figura 66</b> <i>Configuración Pantalla LCD 16X2</i> .....	100
<b>Figura 67</b> <i>Función Sensor de temperatura DHT11</i> .....	101
<b>Figura 68</b> <i>Función Sensor de temperatura DHT11</i> .....	101
<b>Figura 69</b> <i>Librerías de Arduino IDE empleadas en el Código</i> .....	102
<b>Figura 70</b> <i>Configuración Wifi</i> .....	102
<b>Figura 71</b> <i>Configuración Wifi</i> .....	102
<b>Figura 72</b> <i>Instalación sensor temperatura cuarto frio</i> .....	103
<b>Figura 73</b> <i>Arquitectura del Software</i> .....	104
<b>Figura 74</b> <i>Monitoreo pantalla LCD temper en 24.5 grados Celsius</i> .....	105
<b>Figura 75</b> <i>Monitoreo pantalla LCD temper en 28.9 grados Celsius</i> .....	105
<b>Figura 76</b> <i>Monitoreo local pantalla LCD temperatura en 17.5 grados Celsius</i> .....	106

<b>Figura 77</b> <i>Cuarto frio en escala</i> .....	107
<b>Figura 78</b> <i>Parte interna del cuarto frio</i> .....	108
<b>Figura 79</b> <i>Indicador luminoso entre parámetros</i> .....	108
<b>Figura 80</b> <i>Indicador luminoso temperatura alta fuera de parámetros</i> .....	109
<b>Figura 81</b> <i>Indicador luminoso temperatura baja fuera de parámetros</i> .....	109

## Lista de Apéndices

<b>Apéndice A</b> <i>Código principal Arduino</i> .....	126
<b>Apéndice B</b> <i>Código alerta temperatura análisis Matlab</i> .....	130

Garantizar la conservación adecuada de los productos en las plantas industriales es una tarea esencial que va más allá de la eficiencia operativa; representa un compromiso con la calidad, la seguridad y el bienestar de los consumidores. La gestión de la temperatura dentro de la cadena de frío constituye un desafío constante, ya que cualquier desviación puede traducirse en pérdidas económicas significativas y en riesgos para la inocuidad de los productos.

Este proyecto nace de la necesidad de mejorar los procesos de monitoreo de temperatura en tiempo real, utilizando tecnología IoT para ofrecer soluciones más eficientes y precisas. A través de sensores y plataformas de telemetría, buscamos no solo detectar variaciones de temperatura, sino generar también alertas inmediatas que permitan intervenciones oportunas, evitando daños en los productos y optimizando la operación en las plantas industriales.

Más allá de su aplicación técnica, este trabajo es un reflejo del potencial de la automatización y la innovación en la industria. Es un esfuerzo por integrar la tecnología con la responsabilidad de preservar la calidad de los productos, minimizando el desperdicio y mejorando los estándares de seguridad. Con este sistema, se busca marcar una diferencia tangible en la gestión de la cadena de frío, contribuyendo a la sostenibilidad y la eficiencia del sector.

En este proyecto, se implementa un sistema de monitoreo basado en tecnología IoT para detectar y gestionar en tiempo real las variaciones de temperatura en la cadena de frío. Utilizando sensores DHT11, el sistema mide la temperatura y la humedad del entorno, mostrando la información en una pantalla LCD y enviándola a la plataforma ThingSpeak a través de conexión WiFi.

Para garantizar una respuesta rápida ante fluctuaciones térmicas, el sistema incorpora un mecanismo de alertas visuales y sonoras: que son un Led verde indica que la temperatura está dentro del rango óptimo, Un Led rojo y un timbre avisan cuando la temperatura supera los valores establecidos y finalmente tenemos un Led amarillo y una alerta sonora notifican cuando la temperatura es inferior al umbral permitido.

Este prototipo permite el monitoreo en tiempo real facilitando la toma de decisiones de forma inmediata al enviar notificaciones a la nube, esto asegura la conservación de los productos sensibles a la temperatura y evitando pérdidas en la cadena de frío. Su integración con tecnologías IoT refuerza la eficiencia del control de temperatura, permitiendo una supervisión remota y la optimización de procesos industriales.

## **Planteamiento del Problema**

En la mayoría de las plantas industriales que requieren conservación en frío, se ha identificado una necesidad crítica: la falta de un sistema confiable que permita monitorear la temperatura en tiempo real y generar alertas automáticas ante cualquier desviación. Los sistemas actuales, en su mayoría, carecen de conectividad o capacidad de notificación inmediata, lo que impide a los técnicos de mantenimiento reaccionar de forma oportuna. Esta situación ha generado pérdidas económicas significativas, afectando directamente la calidad, la seguridad de los productos almacenados y la eficiencia operativa.

Además del daño a las materias primas o productos en proceso, la falta de monitoreo eficiente impacta negativamente en la productividad. En muchos casos, se deben detener las líneas de producción hasta resolver el problema, lo que puede extenderse por más de una hora. Si no hay insumos disponibles en stock, se depende del área de compras para adquirir nueva materia prima, provocando retrasos aún mayores en la operación.

Actualmente, en muchas instalaciones industriales se utilizan cavas de distintos tamaños —desde pequeñas cámaras de 2.000 kilogramos hasta grandes cavas de 50.000 kilogramos— que son monitoreadas con sensores básicos sin comunicación inalámbrica o con simples termómetros digitales sin conectividad. Este tipo de monitoreo impide actuar con rapidez y limita la trazabilidad de las condiciones de conservación.

Para abordar esta problemática, se propone el desarrollo de un sistema de monitoreo en tiempo real que utilice tecnologías IoT para detectar variaciones de temperatura en las cavas de conservación que permita visualizar en tiempo real las gráficas de comportamiento el cual permite llevar una trazabilidad de la temperatura y que a su vez envíe notificaciones por correo electrónico al personal encargado. Esto permitirá a los técnicos de mantenimiento actuar de forma inmediata y evitar daños mayores en los productos almacenados.

El objetivo principal de este sistema es brindar a las plantas industriales una herramienta

eficiente y accesible para garantizar la continuidad y confiabilidad de la cadena de frío, prevenir pérdidas materiales y mantener la calidad de los productos durante todo el proceso de almacenamiento.

En contexto a esto: ¿De qué manera la implementación de un sistema de monitoreo en tiempo real, basado en tecnología IoT, puede mejorar la conservación de productos en las cavas de las plantas industriales, reduciendo las pérdidas económicas y optimizando la eficiencia operativa?

## **Justificación**

La necesidad de desarrollar un sistema de monitoreo de temperatura en la cadena de frío se fundamenta en diversos aspectos sociales, científicos y económicos que evidencian su importancia y beneficios potenciales.

Nos encontramos en una sociedad donde la transformación y conservación adecuada de los productos es crucial para garantizar la seguridad alimentaria y la salud pública. La Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) señala que la pérdida y desperdicio de productos representan un desafío global, ya que afecta tanto la disponibilidad de productos como la sostenibilidad del sistema productivo (FAO, El estado de la seguridad alimentaria y la nutrición en el mundo, 2023).

Uno de los principales factores que inciden en la pérdida de productos es la temperatura inadecuada en las cavas de conservación, lo que puede provocar la proliferación de microorganismos patógenos y la contaminación de los productos almacenados. Según la Organización Mundial de la Salud (OMS), las enfermedades transmitidas por productos afectan a más de 600 millones de personas anualmente, muchas de ellas causadas por una mala conservación de los productos perecederos (OMS, 2022). Implementar un sistema de monitoreo en tiempo real contribuirá a garantizar la calidad y seguridad de los productos alimenticios, protegiendo la salud de los consumidores y reduciendo el riesgo de contaminación.

Desde el punto de vista tecnológico, el desarrollo de un sistema de monitoreo de temperatura implica la aplicación de principios científicos relacionados con la física y la tecnología de sensores. Este proyecto permitirá investigar y aplicar estos conocimientos en el diseño de un sistema eficiente y preciso que pueda detectar cambios mínimos en la temperatura y generar alertas oportunas. Además, un estudio de la FAO resalta la importancia de la digitalización en la industria alimentaria para mejorar la eficiencia en la conservación de

productos y minimizar las pérdidas económicas (FAO, fao, 2021)

En términos económicos, Los sistemas de refrigeración industriales trabajan de forma continua y están sometidos a un desgaste constante. Un fallo inesperado en un compresor, un ventilador o una fuga de gas refrigerante puede hacer que la temperatura se eleve rápidamente y provoque pérdidas económicas significativas. (Solaz, 2025). Al implementar un sistema de monitoreo eficaz, se reducirán las pérdidas económicas asociadas con la degradación de los productos y se optimizará la gestión de inventarios, mejorando así la rentabilidad y competitividad de la empresa en el mercado.

Este proyecto tiene impactos positivos en distintos niveles:

Consumidores: Se les garantiza productos de alta calidad y frescura, reduciendo riesgos sanitarios.

Empresa: Disminución de pérdidas económicas y mejora en la eficiencia operativa.

Operarios y personal de mantenimiento: Se proporciona una herramienta eficaz para la detección y prevención de problemas en la cadena de frío, reduciendo tiempos de respuesta ante fallas.

El sistema de monitoreo propuesto se basará en tecnologías IoT, utilizando sensores de temperatura conectados a una plataforma de monitoreo centralizada. Esto permitirá la supervisión continua de las condiciones de almacenamiento, detectando y alertando sobre cualquier variación no deseada. Además, el sistema podrá generar informes detallados y registrar históricos de temperatura para análisis posterior, garantizando un control completo sobre el proceso de conservación de productos.

Con la implementación de esta solución, se espera fortalecer la eficiencia en la industria alimentaria, alineándose con las tendencias de automatización y digitalización que están transformando el sector (ObschatkoMedina, 2020).

## **Objetivos**

### **Objetivo General**

Implementar un sistema de monitoreo de temperatura en las cavas de conservación de frío de plantas industriales mediante sensores y una plataforma centralizada para seguimiento en tiempo real.

### **Objetivos Específicos**

Diseñar la infraestructura de sensores necesaria para monitorear la temperatura en las cavas de conservación de materias primas en plantas industriales.

Desarrollar un sistema de alerta temprana que detecte y notifique automáticamente las variaciones no deseadas de temperatura en la cadena de frío, tanto a nivel local como remoto.

Implementar una plataforma centralizada para el monitoreo y análisis en tiempo real de la temperatura.

Evaluar la efectividad del sistema de monitoreo mediante pruebas piloto en plantas Industriales, con el fin de validar su funcionamiento y ajustar posibles fallos o mejoras requeridas.

## **Marco Teórico**

### **Seguridad Alimentaria**

La urbanización está aumentando en muchos países y en este informe se muestra que está cambiando los sistemas agroalimentarios de formas que ya no podemos entender si consideramos las zonas rurales y las zonas urbanas separadamente. La pauta cambiante de las aglomeraciones de población en el continuo rural-urbano y su interfaz como lugar de intercambio e interacciones socioeconómicas está remodelando los sistemas agroalimentarios, y a su vez, está siendo remodelada por estos, lo que tiene implicaciones sobre la disponibilidad y asequibilidad de las dietas saludables y, por ende, sobre la seguridad alimentaria y la nutrición (FAO, fao, 2023).

### **Principios de Conservación de los Productos**

La conservación de productos consiste en bloquear la acción de los agentes (microorganismos o enzimas) que pueden alterar las características originarias (aspecto, olor y sabor). Estos agentes pueden ser ajenos a los productos (microorganismos del entorno: como bacterias, mohos y levaduras) o estar en su interior, como las enzimas que naturalmente están presentes en ellos.

Para controlar el deterioro de los productos se aplican técnicas de conservación, es decir tecnologías de barrera.

Este deterioro puede ser causado por microorganismos y/o por una variedad de reacciones fisicoquímicas que ocurren después de la cosecha. Sin embargo, la prioridad de cualquier proceso de conservación es minimizar la probabilidad de ocurrencia y de crecimiento de microorganismos deteriorativos y patógenos (Reynes, 2024).

### **Procesos de Descomposición de Productos**

Los alimentos se descomponen por dos causas: por fenómenos vitales y fenómenos no vitales. Los principales causantes de dicha descomposición por fenómenos vitales son los microorganismos, así como las enzimas presentes en los alimentos. Las enzimas son compuestos

de tipo biológico gracias a las cuales se catalizan reacciones químicas específicas. Los microorganismos y las enzimas producen la descomposición interviniendo en procesos físicos y químicos de transformación de las sustancias que componen los alimentos. (Releemat, 2022)

### ***Marco Conceptual***

**Qué es un Cuarto Frio.** Es un espacio refrigerado diseñado para conservar productos perecederos en condiciones óptimas de temperatura y humedad. Estas instalaciones son fundamentales en la industria de la refrigeración y desempeñan un papel vital en la preservación de productos y otros productos sensibles al calor, se utiliza en industrias como la alimentaria, farmacéutica y floral (refrigeracionaldama, 2025).

**Cuarto Frío de Conservación.** Este cuarto frío es el más común y conocido, ya que es el más utilizado en los comercios de productos, supermercados y restaurantes. Su temperatura de trabajo oscila entre 0°C y 18°C (RefriClimas, 2024).

### **Figura 1**

*Cuarto Frio de Conservación*



*Nota.* Imagen de Cuarto Frio de Conservación. Tomado de (RefriClimas, 2024).

**Cuarto Frío de Congelación.** Este tipo de cámaras frigoríficas trabajan con temperaturas menores a 0°C. Están diseñadas para productos que solo se pueden conservar congelados. Un ejemplo de productos para almacenamiento congelado sería cualquier tipo de carne, o el hielo de consumo. Las temperaturas de trabajo oscilan entre 0°C y -20°C.

## Figura 2

*Cuarto Frío de Congelación*



*Nota.* Imagen de un Cuarto Frío de Congelación. Tomado de (*automatismosguillen, 2024*).

Túnel de Congelación. tienen características similares a las cámaras de congelación, con una gran diferencia: El proceso de congelación es acelerado y en movimiento. Son equipos diseñados para congelar a través de chorros de aire frío, como parte de un proceso productivo. El producto entra por uno de sus extremos, al “Túnel”, donde se congela a medida que va pasando. Su temperatura de trabajo puede estar entre los 0°C y los -30°C.

### Figura 3

#### *Túnel de Congelación*



*Nota.* Imagen de un Túnel de Congelación. Tomado de (TGM, 2021)

**Abatidor de Temperatura.** El abatidor de temperatura se utiliza cuando se necesita un “Congelamiento rápido” del producto. Son equipos que pueden trabajar con temperaturas desde 0°C hasta -45°C, siempre con el objetivo de congelar en menos de dos horas, es muy utilizado en la industria alimenticia.

### Figura 4

#### *Abatidor de Temperatura*



*Nota.* La Imagen muestra un Abatidor de Temperatura. Tomado de (deldivel, 2024)

**Cadena de Frío.** La cadena de frío consiste en mantener la temperatura controlada que necesita

un alimento para alcanzar la fecha de consumo preferente indicada en su envase en las mejores condiciones para su consumo (Findus, 2024).

La cadena de frío debe mantenerse a lo largo de todas las etapas por las que pasa el producto ultracongelado desde su producción y almacenamiento, a su transporte y distribución, hasta que sea cocinado por el consumidor (Findus, 2024).

Si la cadena de frío de los productos se interrumpe, puede ponerse en juego tanto la calidad del producto como la seguridad de las personas que lo consumiesen (Findus, 2024).

En el caso de tener una cadena de frío que se mantiene intacta durante la producción, transporte, almacenamiento y venta, se garantiza al consumidor que el producto que recibe se ha mantenido en un rango de temperatura de seguridad en el que los microorganismos, especialmente los más perjudiciales para la salud si es que existieran, han detenido su actividad.

Además, una temperatura de conservación adecuada preservará las características del alimento tanto organolépticas como nutricionales (hannacolombia, 2024).

**La Conservación de los Productos.** La conservación de los productos es un proceso fundamental para garantizar la frescura, la calidad y la seguridad de los productos que consumimos. A lo largo de la historia, el ser humano ha desarrollado diferentes técnicas y métodos para preservar los productos, evitando su deterioro y prolongando su vida útil (Cortes, 2025).

**Sensor de Temperatura.** Según la empresa GSL industrias un sensor de temperatura es un dispositivo diseñado para medir la temperatura por medio de señales eléctricas, transformando la información recibida en datos electrónicos que pueden ser captados por los usuarios para generar la supervisión y registro de una variable (Industrias, 2021).

**Telemetría.** La telemetría permite la medición de datos y el control de procesos y/o equipos a

distancia, monitoreados desde un centro de control, para asegurar la eficiencia y seguridad del funcionamiento del sistema (evaluandoerp, 2022).

**Equipos Cadena de Frío.** Existen diferentes equipos para la conservación de productos o productos médicos (vacunas), los cuales deben estar siempre monitorizados para evitar que sufran daño, por tal motivo a continuación se describen los diferentes equipos:

**Cámaras Frías o Cuarto Frío.** Es un equipo de refrigeración que se utiliza para la conservación y almacenamiento de productos en grandes cantidades (Consorempresarial, 2020).

**Internet of Things (IOT).** El internet de las cosas (Internet of Things, IoT) es una red de objetos físicos conectados a Internet, los cuales logran una interacción mediante sistemas embebidos, redes de comunicaciones y aplicaciones en la nube. Permite la comunicación entre los objetos, capturar, almacenar y recuperar datos, interactuar con usuarios, sistemas y aplicaciones que generan un entorno más inteligente, autónomo y conectado (Muñoz, 2019).

**Sistema de Monitoreo de Cadena de Frío.** Son dispositivos en tiempo real que han tenido un enorme éxito en el mercado y son un gran aliado para llevar a cabo el monitoreo de la cadena de frío debido a los beneficios directos que aportan. uno de ellos es:

Enviar una alerta en tiempo real si se sobrepasa la temperatura máxima. Esto permite que se pueda contactar al operador de la unidad refrigerada para validar que la refrigeración esté trabajando adecuadamente (Danahé San Juan, 2021).

**Automatización.** La automatización es la aplicación de tecnología, programas, robótica o procesos para lograr resultados con una intervención humana mínima.

El software y las tecnologías de automatización se emplean en una amplia gama de industrias, desde finanzas hasta atención médica, servicios públicos y defensa, y prácticamente en todas partes intermedias. La automatización se puede emplear en todos los

aspectos de las funciones empresariales, y las organizaciones que la emplean con mayor eficacia pueden obtener un beneficio competitivo significativo (IBM, 2022).

**Señales Digitales.** La señal digital es una forma de representar una señal analógica mediante un código binario. Esto permite que la señal analógica sea procesada por un dispositivo digital, como un ordenador. Las señales digitales pueden ser transmitidas a través de cables, ondas radio o incluso a través del vacío (por ejemplo, en las comunicaciones por satélite) (Tecnologicos, 2023).

**NodeMCU.** NodeMCU es una placa de desarrollo basado en el ESP12E el cuál es probablemente el módulo más popular que integra el SoC ESP8266. Sin embargo, pese a la popularidad de las placas NodeMCU, también existe mucha confusión respecto a la terminología (Lua, Lolin, versiones) y, en ocasiones, se mezclan o incluso se usan como sinónimos.

El firmware NodeMCU fue creado poco después de aparecer el ESP8266, el 30 de diciembre de 2013. Unos meses después, en octubre de 2014 se publicó la primera versión del firmware NodeMCU en Github. Dos meses más tarde se publicaba la primera placa de desarrollo NodeMCU, denominada devkit v0.9, siendo también Open Hardware.

El firmware NodeMCU podía grabarse en un ESP8266, tras lo cual podíamos programarlo con el lenguaje script Lua. La programación el Lua permitía la conexión y programación del ESP8266 de una forma mucho más sencilla que las herramientas oficiales proporcionadas por Espressif. (Llamas, 2022).

NodeMCU es un nombre que recoge tanto un firmware Open Source y como a una placa de desarrollo basados en el ESP8266.

En principio el nombre NodeMCU se refería principalmente al firmware. Actualmente esto se ha invertido y cuando hablamos de NodeMCU normalmente nos referimos a la placa de desarrollo.

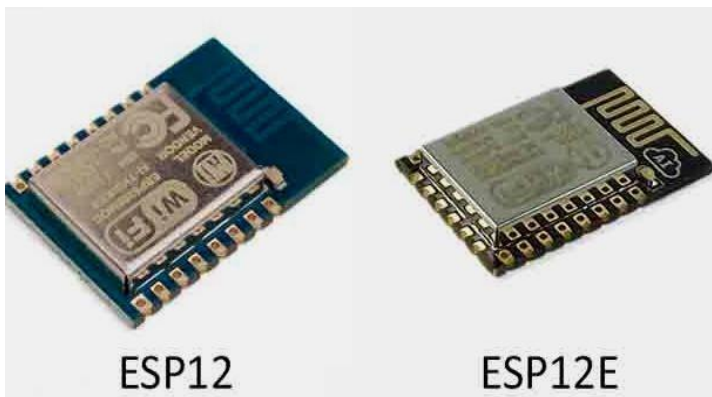
Por otro lado, tenemos tres versiones de la placa NodeMCU, que veremos a continuación. Las designaciones de las versiones también contribuyen a generar confusión. Pero, aunque parece un poco

complicado, en realidad es sencillo si sigues la evolución desde el principio.

La versión original del NodeMCU se denominó devkit v0.9, y montaba un ESP12 junto a 4MB de flash (recordemos que la memoria en el ESP8266 es externa y se conecta por SPI). El ESP12 es similar al ESP12E, pero carece de una hilera de pines por lo que dispone de menos GPIO. (Llamas, 2022).

### Figura 5

*Versión original del NodeMCU*



*Nota.* Versión original del NodeMCU. Tomado de (Llamas, 2022).

La siguiente versión del NodeMCU es la v1.0 V2. De forma resumida Amica, una compañía creada por el alemán Gerwin Janssen, fabricó su propia versión mejorada de la v0.9.

Al equipo original de NodeMCU le gustó y **la** declararon versión “oficial” de NodeMCU.

## Figura 6

Versión del NodeMCU es la v1.0 V2

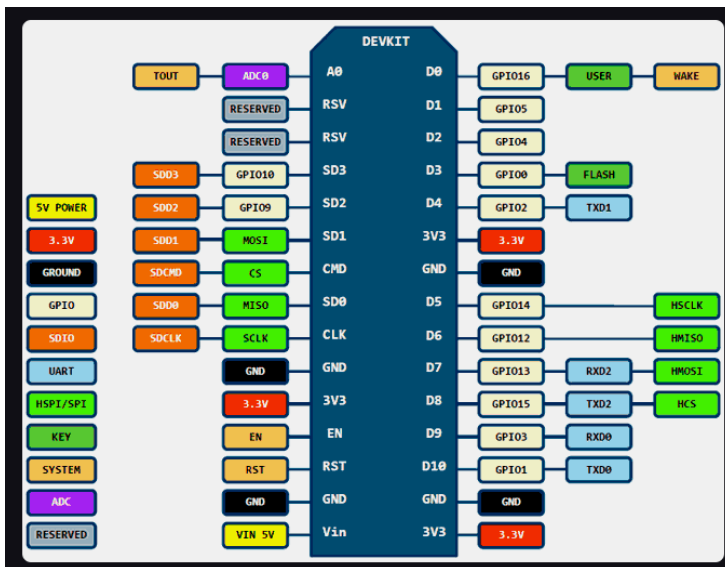


Nota. Imagen de la Versión del NodeMCU es la v1.0 V2. Tomado de. (Llamas, 2022).

La principal diferencia de la versión v1.0 v2 es que monta un ESP12E en lugar de un ESP12, por lo que tiene más pines disponibles que el modelo original. Además, es más estrecha que la versión 0.9, tapando sólo 8 hileras de una breadboard. Esto deja una hilera adicional a cada lado para realizar conexiones (Llamas, 2022).

## Figura 7

Diferencia de la versión v1.0 v2



Nota. La figura muestra la Diferencia de la versión v1.0 v2. Tomado de (Llamas, 2022)

Los tres fabricantes, Amica, Lolin/Wemos y DOIT/SmartArduino fabrican, o han fabricado en algún momento, esta versión v1.0 V2.

Llegamos a lo que a veces se denomina “tercera generación”, la versión 1.0 V3. Básicamente el fabricante Lolin/Wemos decidió crear su propio diseño “mejorado” con unos cuantos cambios menores (Llamas, 2022).

### **Figura 8**

*Tercera generación”, la Versión 1.0 V3*

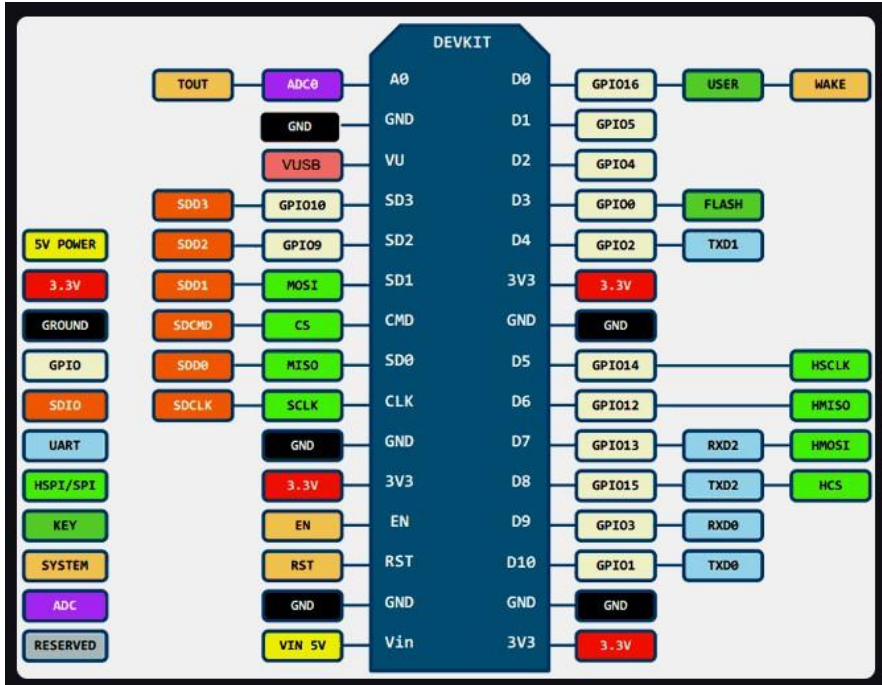


*Nota.* La figura muestra la Tercera generación”, la versión 1.0 V3. Tomado de (Llamas, 2022).

El cambio principal es que la V3 monta un conversor serial CH340G en lugar del CP2102. El fabricante asegura que hace que el puerto USB sea más robusto. Por otro lado, reusaron los dos pines reservados de la V2 para sacar un GND y un VUSB (Llamas, 2022).

**Figura 9**

*Conversor serial CH340G en lugar del CP2102*



*Nota.* Imagen de la Conversor serial CH340G en lugar del CP2102. Tomado de (Llamas, 2022).

**Figura 10**

*Resumen de las Versiones*

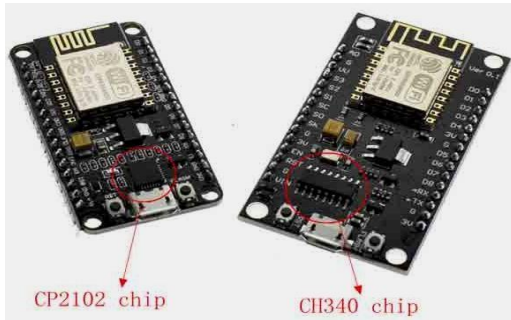
Generación	Versión	Módulo	Anchura	Comentario
Primera	0.9	ESP12	10 pines	La original, no está a la venta
Segunda	1.0	V2	ESP12E	Versión "oficial"
Tercera	1.0	V3	ESP12E	Versión de Wemos/Lolin. Conversor CH340G

*Nota.* Resumen de las versiones. Tomado de. (Llamas, 2022).

La mejor forma de diferencias entre V2 y V3 de un vistazo es fijarnos en el conversor serial que monta. El CP2102 que es cuadrado, o el CH340G que es más alargado (Llamas, 2022).

## Figura 11

### Diferencias entre V2 y V3



Nota. Diferencias entre V2 y V3. Tomado de. (Llamas, 2022).

Imagen del resumen con los pines, sus funciones, y algunas de sus peculiaridades (Llamas, 2022)

## Figura 12

### Resumen con los Pines y sus Funciones (Llamas, 2022)

Pin	GPIO	Input	Output	Comentarios
D0	GPIO16	No interrupciones	No PWM No I2C	HIGH durante boot Resistencia Pull-Down Conectar a RST para Wake-Up
D1	GPIO5	OK	OK	SCL (I2C) (frecuentemente)
D2	GPIO4	OK	OK	SDA (I2C) (frecuentemente)
D3	GPIO0	Pulled Up	OK	Boot falla si pulled LOW Conectado a botón FLASH
D4	GPIO2	Pulled Up	OK	HIGH durante boot Boot falla si pulled LOW Built-in LED
D5	GPIO14	OK	OK	TX1
D6	GPIO12	OK	OK	SLCK (SPI)
D7	GPIO13	OK	OK	MISO (SPI)
D8	GPIO15	Pulled GND	OK	CS (SPI) LOW durante boot Boot falla si pulled HIGH No tiene Pull-Up
RX	GPIO3	OK	RX	HIGH durante boot No usable si se usa UART
TX	GPIO1	TX	OK	HIGH durante boot Boot falla si pulled LOW Debug output en boot No usable si se usa UART
A0	ADC0	Analog input	NO	

Nota. La imagen muestra los pines y sus funciones. Tomado de. (Llamas, 2022)

**Arduino IDE.** Es un entorno de desarrollo y en él se realiza la programación de cada una de las placas de Arduino. Tiene como base el entorno de Processing al igual que un lenguaje de programación fundamentado en Wiring.

Con esta labor la compañía ha buscado facilitar el acceso y uso no solo de la electrónica sino también de la programación de sistemas embebidos. Además, es necesario considerar que toda la plataforma presentada por la compañía se libera bajo la licencia de código abierto y es lo que da total libertad de acceso (tuelectronica, 2021).

### Figura 13

#### *Arduino IDE*

The image shows a screenshot of the Arduino IDE software. The window title is "Blink Arduino 1.6.7". The menu bar includes "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar contains icons for opening files, saving, and running. The main text area displays the "Blink" example code. The code includes a multi-line comment explaining the function, a setup function that initializes pin 13 as an output, and a loop function that toggles the LED on and off every second. The status bar at the bottom indicates the board type: "Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4".

```
Blink Arduino 1.6.7
Archivo Editar Programa Herramientas Ayuda
Blink
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://www.arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
}
```

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

*Nota.* Arduino IDE. Tomado de. (tuelectronica, 2021)

**Modulo I2C.** I2C es un puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA y SCL. Además, el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/s. También es conocido como IIC o TWI – Two Wire Interfaz.

El protocolo I2C es uno de los más utilizados para comunicarse con sensores digitales, ya que a diferencia del puerto Serial, su arquitectura permite tener una confirmación de los datos recibidos, dentro de la misma trama, entre otras ventajas.

La conexión de tantos dispositivos al mismo bus es una de las principales ventajas. Además, si comparamos a I2C con otro protocolo serial, como Serial TTL, este incluye más bits en su trama de comunicación que permite enviar mensajes más completos y detallados.

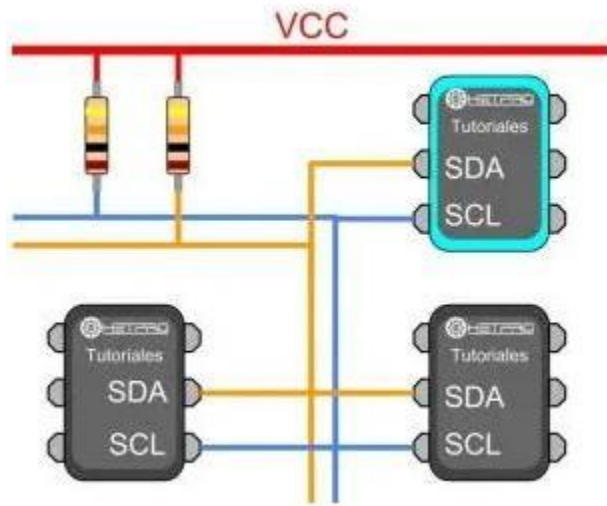
Los mensajes que se envían mediante un puerto I2C, incluye además del byte de información, una dirección tanto del registro como del sensor. Para la información que se envía siempre existe una confirmación de recepción por parte del dispositivo. Por esta razón es bueno diferenciar a los distintos elementos involucrados en este tipo de comunicación (hetpro, 2022)

**I2C – Esquema de Comunicación y Elementos.** Siempre que hablamos de una comunicación oral, se entiende que es entre dos o más personas. Como consecuencia podemos también indicar que en una comunicación digital existen distintos dispositivos o elementos. En el caso de I2C se diferencian dos elementos básicos, un MAESTRO y un ESCLAVO. La Figura-1, muestra una conexión típica de tres dispositivos,

El bus consiste en dos líneas llamadas, Serial DATA – SDA y Serial CLOCK – SCL. Es decir, Datos Seriales y Reloj Serial. En particular al bus se le conectan dos resistencias en arreglo pull-up, de entre 2.2K y 10K. (hetpro, 2022).

**Figura 14**

*Conexión de Tres Dispositivos a un Bus de Comunicación I2C*



*Nota.* Conexión de tres dispositivos a un bus de comunicación I2C. Tomado de. (hetpro, 2022)

El Maestro I2C se encarga de controlar al cable de reloj, por sus siglas en inglés llamada SCL – Serial CLock. Además, el MAESTRO se encarga de iniciar y parar la comunicación. La información binaria serial se envía sólo por la línea o cable de datos seriales, en inglés se llama SDA – Serial DATA. Dos Maestros no pueden hacer uso de un mismo puerto I2C. Puede funcionar de dos maneras, como maestro-transmisor o maestro-receptor (hetpro, 2022).

Sus funciones principales son:

Iniciar la comunicación – S

Enviar 7 bits de dirección – ADDR

Generar 1 bit de Lectura o Escritura – R/W

Enviar 8 bits de dirección de memoria

Transmitir 8 bits de datos –

Confirmar la recepción de datos – ACK – ACKnowledged

Generar confirmación de No-recepción, NACK – No-ACKnowledged

El ESCLAVO I2C, generalmente suele ser un sensor. Este elemento suministra de la

información de interés al MAESTRO. Puede actuar de dos formas: esclavo-transmisor o

esclavo-receptor. Un dispositivo I2C esclavo, no puede generar a la señal SCL (hetpro, 2022). Sus funciones principales son:

Enviar información en paquetes de 8 bits.

Enviar confirmaciones de recepción, llamadas ACK

**I2C – Bits de la Trama del Puerto.** El protocolo de comunicación I2C se refiere al conjunto de bits que son necesarios para enviar uno o varios bytes de información. En lo particular, para este protocolo existen los siguientes bits importantes:

Inicio ó Start – S

Parada – P

Confirmación – ACK

NoConfirmación – NACK

Lectura-/Escritura – L/W

7 bits para la dirección del dispositivo esclavo/maestro

8 bits de dirección (para algunos sensores pueden ser 16 bits)

8 bits de datos

El conjunto de estos bits y su orden va formando distintas tramas de comunicación. Existen distintos modos de comunicación dependiendo del arreglo de estos bits. Tanto el maestro como el esclavo pueden o no generar los bits anteriores, según los modos de comunicación (hetpro, 2022).

El puerto I2C está disponible si las dos líneas, SDA y SCL están en un nivel lógico alto.

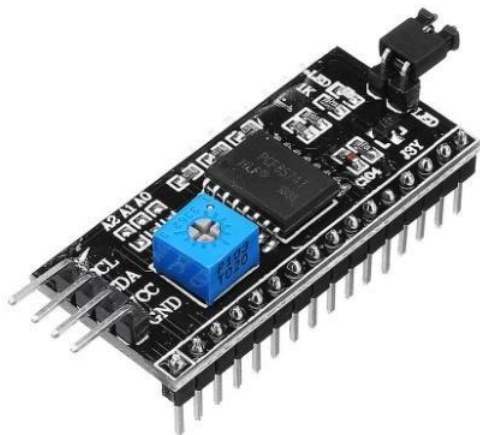
**I2C – Modos de Comunicación.** Los modos de comunicación en I2C se refieren a las distintas tramas que pueden formarse en el bus. Estas tramas o modos dependen, por ejemplo, si queremos leer al sensor esclavo, o si lo queremos configurar. Existen principalmente dos modos de comunicación:

Maestro-Transmisor y Esclavo-Receptor. Este modo se usa cuando se desea configurar un registro del esclavo I2C.

Maestro-Receptor Y Esclavo-Transmisor. Se usa cuando queremos leer información del sensor I2C (hetpro, 2022).

### **Figura 15**

*Modulo I2C*



*Nota.* La imagen muestra el Módulo I2C. Tomado de. (osakaelectronics, 2022).

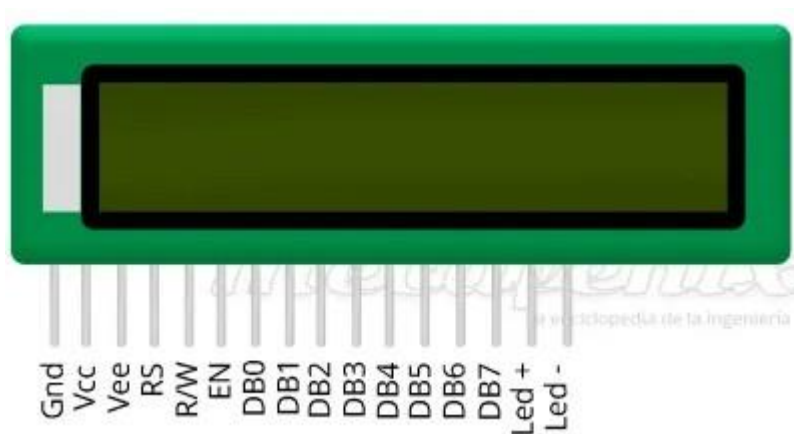
**Pantalla LCD 16X2.** El LCD (Liquid Crystal Display) o pantalla de cristal líquido es un dispositivo empleado para la visualización de contenidos o información de una forma gráfica, mediante caracteres, símbolos o pequeños dibujos dependiendo del modelo.

En este caso vamos a emplear un LCD de 16×2, esto quiere decir que dispone de 2 filas de 16 caracteres cada una. Los píxeles de cada símbolo o carácter varían en función de cada modelo.

El LCD consta de 16 terminales las cuales podemos dividir en pines de alimentación, control y bus de datos bidireccional. Por lo general podemos encontrar además en su estructura los pines de ánodo de led backlight y cátodo de led backlight (Mecanifex, 2022).

### Figura 16

*Pantalla LCD 16X2*



*Nota.* La imagen muestra la Pantalla LCD 16X2. Tomado de. (Mecanifex, 2022).

**Terminales de Alimentación.** Vss: Gnd, Vdd: +5 voltios, Vee: corresponde al pin de contraste, lo regularemos con un potenciómetro de 10K conectado a Vdd.

**Pines de Control.** RS: Corresponde al pin de selección de registro de control de datos (0) o registro de datos (1). Es decir, el pin RS funciona paralelamente a los pines del bus de datos. Cuando RS es 0 el dato presente en el bus pertenece a un registro de control/instrucción. y cuando RS es 1 el dato

presente en el bus de datos pertenece a un registro de datos o un carácter.

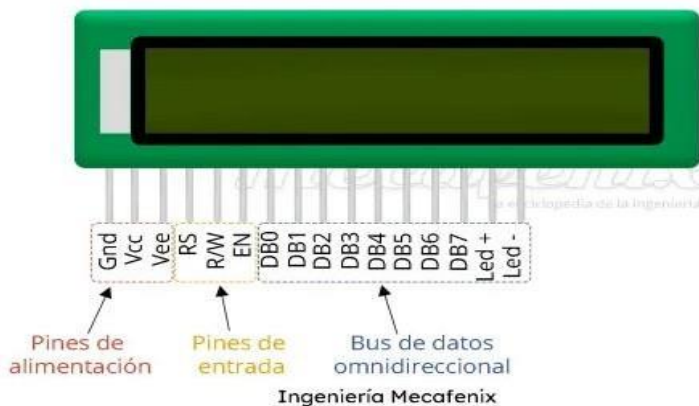
RW: Corresponde al pin de Escritura (0) o de Lectura (1). Nos permite escribir un dato en la pantalla o leer un dato desde la pantalla (Mecanifex, 2022).

E: Corresponde al pin Enable o de habilitación. Si E (0) esto quiere decir que el LCD no está activado para recibir datos, pero si E (1) se encuentra activo y podemos escribir o leer desde el LCD.

**Bus de Datos.** El Bus de datos bidireccional comprende desde los pines D0 a D7. Para realizar la comunicación con el LCD podemos hacerlo utilizando los 8 bits del bus de datos (D0 a D7) o empleando los 4 bits más significativos del bus de datos (D4 a D7) (Mecanifex, 2022).

### Figura 17

#### *Bus de Datos*



*Nota.* La imagen muestra la Bus de datos (Mecanifex, 2022)

**Sensor DHT11.** El DHT11 es un sensor simple que mide temperatura y humedad, todo en uno. Así no tendrás que comprar dos sensores por separado. Su precio es de unos 2€, por lo que es bastante barato, aunque también puedes encontrarlo montado en un módulo (montado en una PCB para facilitar su uso) como suele ser habitual en este tipo de componentes electrónicos para Arduino. En el caso de la placa, incluye una resistencia pull-up de 5 kilo ohmios y un LED que nos avisa del funcionamiento.

El DHT11 es ideal para proyectos educativos y prototipos por su simplicidad, pero no es

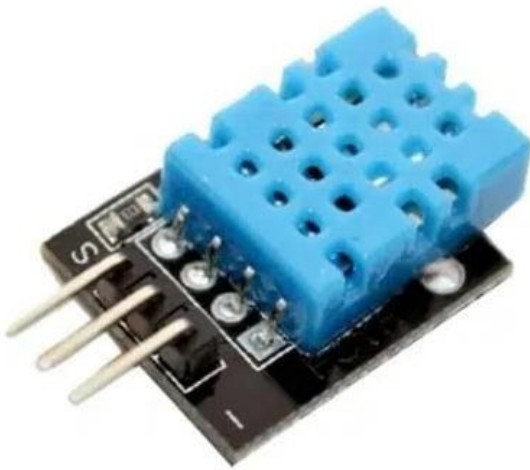
adecuado para ambientes industriales o con alta humedad continua.

Para uso en producción real o conservación de cadena de frío, se recomienda usar sensores más precisos como DHT22, SHT31, BME280 o PT100 con transmisor analógico

DHT11 tiene alta fiabilidad y estabilidad debido a su señal digital calibrada. Además, si te fijas en su Datasheet, verás que tiene características interesantes como verás en próximos apartados (Isaac, 2025).

### Figura 18

*Sensor DHT11*



*Nota.* Sensor DHT11. Tomado de. (Isaac, 2025).

### Tabla 1

*Características DHT11*

Parámetro	Valor / Rango	Observaciones técnicas
-----------	---------------	------------------------

Parámetro	Valor / Rango	Observaciones técnicas
<b>Alimentación (Vcc)</b>	3.0 V a 5.5 V DC	Compatible con 3.3 V y 5 V microcontroladores
<b>Consumo de corriente</b>	2.5 mA en operación / 0.5 mA en reposo	Bajo consumo, ideal para sistemas autónomos
<b>Tipo de salida</b>	Digital de un solo cable (protocolo propietario)	Comunicación por pin digital única
<b>Rango de temperatura medible</b>	0 °C a 50 °C	No apto para ambientes extremos
<b>Precisión de temperatura</b>	±2 °C a 25 °C	Varía con la temperatura ambiente
<b>Resolución de temperatura</b>	8 bits – incrementos de 1 °C	No detecta decimales
<b>Rango de humedad relativa</b>	20 % RH a 90 % RH	No mide valores fuera de este rango
<b>Precisión de humedad</b>	±5 % RH entre 0–50 °C	Se recomienda calibración adicional si se usa en ambientes críticos
<b>Resolución de humedad</b>	1 % RH	Lecturas en pasos enteros
<b>Frecuencia de muestreo</b>	1 Hz (una lectura por segundo)	No apto para monitoreo de alta frecuencia
<b>Tiempo de respuesta típico</b>	<5 segundos	Bajo condiciones ideales de ventilación
<b>Dimensiones del encapsulado</b>	15.5 × 12 × 5.5 mm (aprox.)	Tipo PCB de 3 pines (Vcc, GND, Data)
<b>Compatibilidad</b>	Arduino, ESP8266, ESP32, Raspberry Pi, PIC, STM32	Con librerías disponibles
<b>Temperatura de operación segura</b>	–10 °C a 60 °C (aunque mide solo desde 0 °C)	Sensor puede sobrevivir fuera de rango por cortos periodos

*Nota.* Características del sensor de temperatura y humedad DHT11. Tomado de (Isaac, 2025).

Características del sensor DHT11. Isaac. (2025). <https://www.hwlibre.com/dht11/>

**Buzzer.** Un zumbador o mejor conocido como Buzzer (en inglés) es un pequeño transductor capaz de convertir la energía eléctrica en sonido. Para hacerlos funcionar solo basta

conectar el positivo con el + y la tierra o negativo con el – de una batería o cualquier fuente de corriente directa (Mecafenix, 2025).

### Figura 19

*Buzzer*

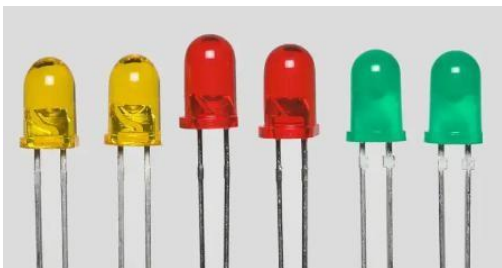


*Nota.* La imagen muestra el Buzzer. Tomado de. (Mecafenix, 2025).

**Diodo Emisor de Luz (LED).** Un Diodo Emisor de Luz (LED) es un dispositivo semiconductor que emite luz cuando una corriente eléctrica fluye a través de él. Cuando la corriente pasa a través de un LED, los electrones se recombinan con huecos emitiendo luz en el proceso. Los Leds permiten que la corriente fluya en dirección directa y bloquean la corriente en dirección inversa (J.L, 2025).

### Figura 20

*Diodo Emisor de Luz*



*Nota.* La imagen muestra un Diodo Emisor de Luz (LED). Tomado de. (J.L, 2025).

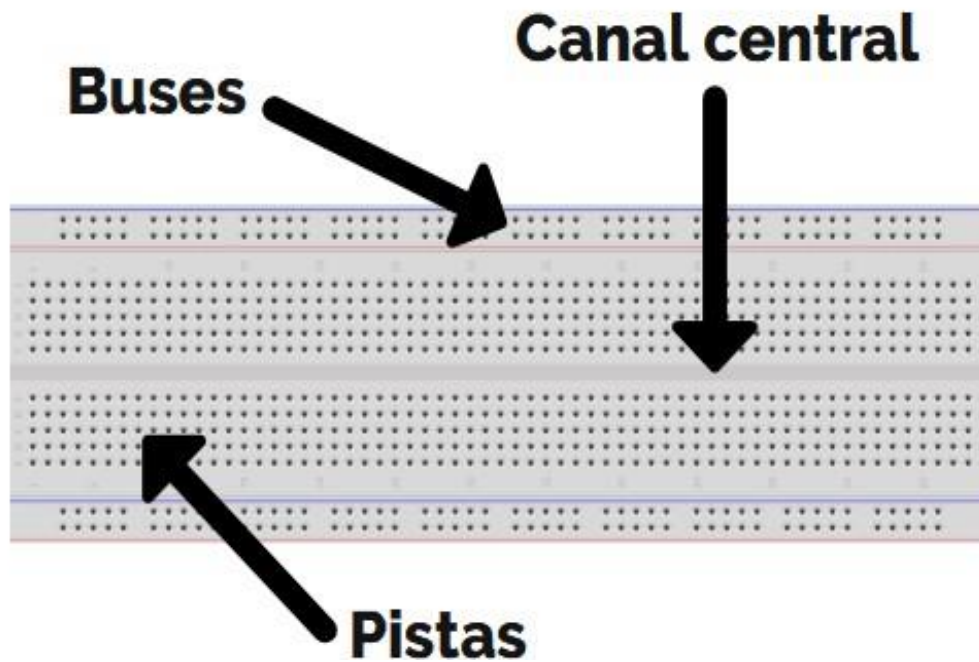
**Protoboard.** El Protoboard es una placa de pruebas que permite interconectar elementos electrónicos sin la necesidad de soldar componentes. Logrando así que se facilite el armado de circuitos o sistemas electrónicos

La placa de Protoboard se emplea normalmente para realizar pruebas experimentales de circuitos

electrónicos. Si la prueba es satisfactoria, el circuito se diseña en una placa de cobre y se suelda para evitar el riesgo de que se desconecte cualquier componente. Si la prueba no es satisfactoria, es fácil cambiar las conexiones y reemplazar los componentes (mielectronicafacil, 2021).

**Figura 21**

*Protoboard*



*Nota.* La imagen se evidencia un Protoboard. Tomado de. (mielectronicafacil, 2021).

**Cable Jumper.** Los Jumpers, también conocidos como puentes o puentes de conexión, son elementos fundamentales en el ámbito de la electrónica y la informática. Estos pequeños dispositivos, generalmente en forma de cables o conectores, se utilizan para establecer conexiones temporales entre puntos específicos de un circuito electrónico. Su función principal es la de cerrar o abrir circuitos, permitiendo así la modificación de la configuración de un sistema o la realización de pruebas y diagnósticos (zonagreen, 2023).

**Cable de Datos.** El cable de datos es, esencialmente, cualquier tipo de medio capaz de transportar una señal binaria de comunicación eléctrica. Los datos binarios son una serie de unos y ceros

que se envían como señal eléctrica entre dos lugares (techlandia, 2022).

**Proteus.** Proteus es una aplicación para la ejecución de proyectos de construcción de equipos electrónicos en todas sus etapas: diseño del esquema electrónico, programación del software, construcción de la placa de circuito impreso, simulación de todo el conjunto, depuración de errores, documentación y construcción.

Sin la utilización de la suite Proteus, el proceso para construir un equipo electrónico basado en un microprocesador se compone de cinco etapas. Sólo al final del proceso somos capaces de detectar los errores y cualquier problema exige volver a ejecutar el ciclo completo.

Con Proteus las fases de prueba no suponen la necesidad de volver a construir nuevos prototipos, con el ahorro de costos y tiempo que ello supone.

Los diferentes módulos que componen Proteus se pueden adquirir de forma independiente añadiendo nuevas funcionalidades a medida que aumentan nuestras necesidades de desarrollo y producción (hubor, 2021).

**Thingspeak.** ThingSpeak es una API y aplicación de código abierto para el Internet de las Cosas que permite almacenar y recopilar datos de objetos conectados a través del protocolo HTTP a través de Internet o de una red local, con ThingSpeak, el usuario puede crear aplicaciones de registro de datos de sensores, aplicaciones de seguimiento de ubicación y una red social para los objetos conectados, con actualizaciones de estado (descubrearduino, 2024).

Thingspeak es un software de código abierto (Open Source) gratuito mantenido por una comunidad de desarrolladores. Podemos descargarlo de su repositorio e instalarlo en nuestro propio servidor si así lo deseamos. Thingspeak.com es un sitio de Internet que presta servicios basados en Thingspeak y ofrece distintos planes de tarifas, incluyendo uno gratuito que contiene algunas limitaciones (tolocka, 2023).

Además de almacenar la información recibida desde nuestros dispositivos, en Thingspeak.com (desde ahora le llamaré directamente Thingspeak) podemos visualizar estos valores de distintas maneras, mantener la información privada o compartirla, hacer distintos tipos de análisis empleando el software matemático Matlab (que está integrado) y establecer alarmas para que el resultado de este análisis nos alerte por mail de alguna situación particular (tolocka, 2023).

### **Funciones ThingSpeak.**

- API abierto
- Recolección de datos en tiempo real
- Datos de geolocalización
- Procesamiento de datos
- Visualización de datos
- Mensajes de estado del circuito
- Plugin

ThingSpeak puede integrarse con Arduino, Raspberry Pi, ioBridge / RealTime.io, Electric Imp, aplicaciones móviles/Web, redes sociales y análisis de datos con MATLAB.

ThingSpeak te permite agregar, visualizar y analizar flujos de datos en directo en la nube.

Algunas de las capacidades clave de ThingSpeak incluyen la capacidad de:

Configurar fácilmente los dispositivos para que envíen datos a ThingSpeak utilizando los protocolos de IO más populares.

- Visualizar los datos de los sensores en tiempo real.
- Agregar datos bajo demanda de fuentes de terceros.
- Utilizar la potencia de MATLAB para dar sentido a tus datos de IO.
- Ejecutar análisis de IO automáticamente en base a horarios o eventos.

- Prototipo y construcción de sistemas de IO sin necesidad de crear servidores o

desarrollar software web.

Actúa automáticamente sobre tus datos y comuníquese utilizando servicios de terceros como Twilio o Twitter.

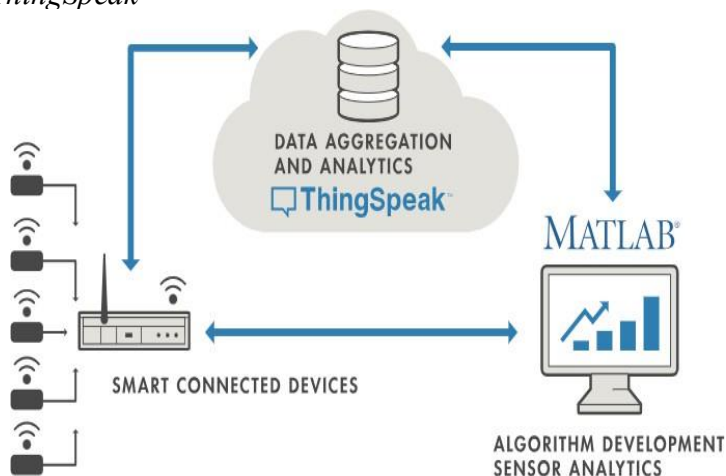
La Internet de los objetos (IO) es un sistema de «cosas conectadas». Las cosas generalmente comprenden un sistema operativo integrado y la capacidad de comunicarse con Internet o con las cosas vecinas.

Uno de los elementos clave de un sistema genérico de IO que tiende un puente entre las diversas «cosas» es un servicio de IO. Una implicación interesante de las «cosas» que comprenden los sistemas de IO es que las cosas por sí solas no pueden hacer nada. Como mínimo, deben tener la capacidad de conectarse con otras «cosas». Pero el poder real de la IO se aprovecha cuando las cosas se conectan a un «servicio», ya sea directamente o a través de otras «cosas».

En estos sistemas, el servicio desempeña el papel de un administrador invisible al proporcionar capacidades que van desde la simple recopilación y supervisión de datos hasta el análisis de datos complejos (descubrearduino, 2024).

## Figura 22

*ThingSpeak*



*Nota.* En la imagen se muestra el diagrama de ThingSpeak. Tomado de. (*thingspeak, 2021*).

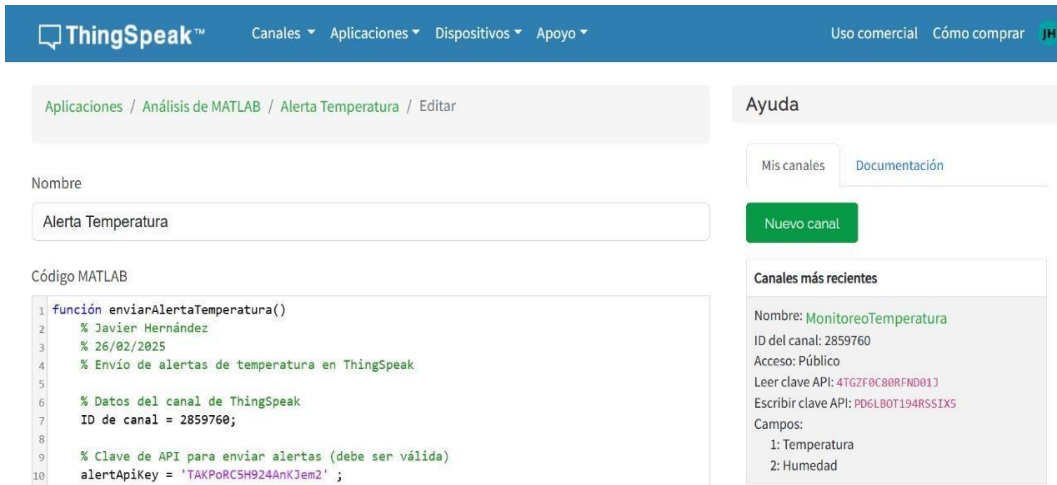
**MATLAB Análisis.** El MATLAB Analysis app para preparar, filtrar y analizar datos, como calcular la humedad promedio, calcular el punto de rocío y eliminar datos atípicos de un canal Thingspeak™. ThingSpeak le permite utilizar MATLAB para analizar y visualizar sus datos. Las aplicaciones de análisis y visualización proporcionan un código de plantilla para ayudarle con operaciones básicas con datos históricos o en vivo. Estos tutoriales lo llevan paso a paso a través de cada plantilla de código. Utilice tutoriales de plantillas de código para comenzar con ThingSpeak. Utilice las aplicaciones para convertir de una unidad a otra, comparar datos diferentes o MATLAB Visualizaciones para permitir la codificación modular.

Además de las funciones integradas MATLAB, también puede utilizar funciones de las cajas de herramientas mencionadas en Acceda a MATLAB Cajas de herramientas complementarias (thingspeak, 2021).similares en el mismo gráfico o visualizar la distribución estadística de sus datos.

Puede Agregar funciones a scripts (MATLAB) en las aplicaciones MATLAB Análisis.

**Figura 23**

*MATLAB Analysis*



*Nota.* En la imagen se evidencia el código del MATLAB Analysis. Tomado de. (thingspeak, 2021).

**Aplicación TimeControl.** La aplicación TimeControl funciona con otras aplicaciones

ThingSpeak™: MATLAB Analysis, ThingHTTP, ThingTweet o TalkBack para realizar una acción en un momento específico o en un horario regular. Puedes usar TimeControl con:

ThingHTTP para comunicarse con dispositivos, sitios web o servicios web

MATLAB® Análisis para actuar sobre tus datos.

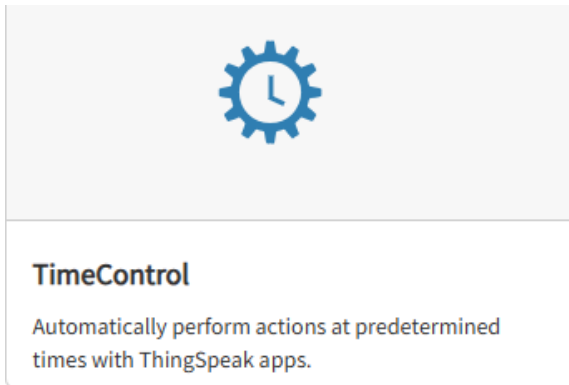
ThingTweet para enviar alertas a través de Twitter®.

TalkBack para poner en cola comandos para un dispositivo.

Por ejemplo, puede realizar una solicitud ThingHTTP que llame a alguien a través de Twilio®, controle un dispositivo o se conecte a un termostato que acepte solicitudes HTTP (thingspeak, 2021).

## Figura 24

### *Application TimeControl*

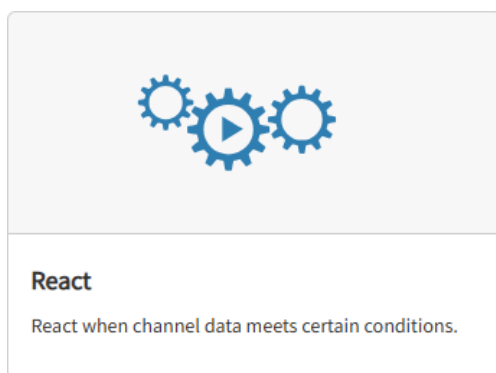


*Nota.* En la imagen se evidencia la Application TimeControl. Tomado de. (*thingspeak, 2021*).

**React.** React funciona con las aplicaciones de análisis ThingHTTP, ThingTweet y MATLAB® para realizar acciones cuando los datos del canal cumplen una determinada condición. Por ejemplo, puede hacer que una aplicación móvil informe su latitud y longitud a un canal ThingSpeak™. Cuando su posición esté dentro de una cierta distancia de su casa, haga que ThingHTTP encienda las luces de su sala de estar (*thingspeak, 2021*).

## Figura 25

### *React*



*Nota.* Se evidencia el sistema React. Tomado de. (*thingspeak, 2021*).

**Envío de Correo por ThingSpeak.** Para enviar la información a Thingspeak desde nuestros dispositivos podemos emplear dos protocolos: HTTP y MQTT. Veremos a continuación que significa

esto.

**El Protocolo HTTP.** Cuando accedemos a cualquier página en la Web lo hacemos empleando el protocolo HTTP. En ese caso nosotros (o el navegador que usemos para acceder) es el cliente, que le solicita información al servidor, la cual generalmente es una página, que puede contener muchos elementos diferentes. Esto se llama solicitud (request). Cuando el servidor recibe la solicitud la analiza y si puede, responde enviando la información solicitada. Eso se denomina respuesta (response). Si algo falló o el servidor no tiene la información solicitada, responderá con un mensaje de error.

El protocolo HTTP es viejo como la misma Web. Fue creado a comienzos de la década de 1990 para definir la forma en que viajaría la información entre los servidores Web y los usuarios.

El protocolo HTTP reconoce dos partes que tienen un rol bien diferenciado: El servidor, que contiene la información y el cliente, que pretende acceder a dicha información.

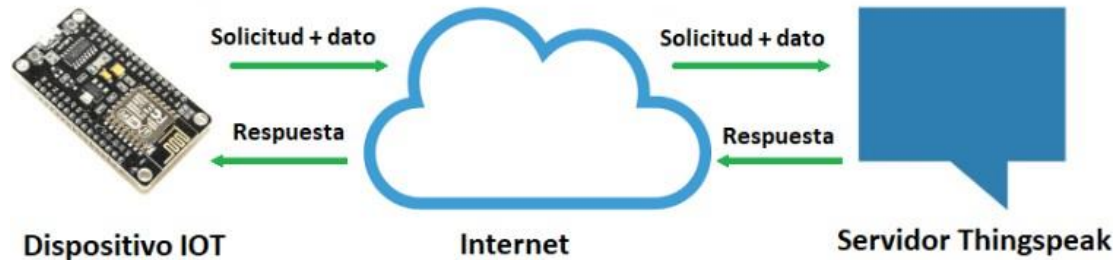
Por esta razón se dice que tiene o responde a una estructura cliente-servidor (tolocka, 2023).

Cuando usamos Thingspeak para enviar datos desde nuestro dispositivo con el protocolo HTTP, el dispositivo es el cliente y Thingspeak el servidor. Si queremos enviar un dato, como un valor leído desde un sensor, debemos hacer una solicitud a una URL que nos provee Thingspeak agregando como información adicional ese valor.

Para enviar esa información dentro de la solicitud, el protocolo HTTP tiene dos métodos: GET y POST (tolocka, 2023).

**Figura 26**

*Enviando datos a Thingspeak*



*Nota.* En la figura se evidencia el sistema Enviando datos a Thingspeak. Tomado de. (tolocka, 2023).

**Los Métodos GET y POST.** GET y POST nos permiten enviar información desde el cliente al servidor dentro de la solicitud. Sirven al mismo propósito, pero tienen características diferentes:

GET: ¿En el método GET, la información enviada se incluye junto a la dirección o URL luego de un símbolo? Si se envían varios datos, se deben separar con el símbolo & (tolocka, 2023).

## **Antecedentes**

A continuación, se presentan ejemplos de sistemas similares desarrollados en Colombia y en otros países, destacando sus características, aplicaciones y aprendizajes relevantes.

### **Antecedes Internacionales**

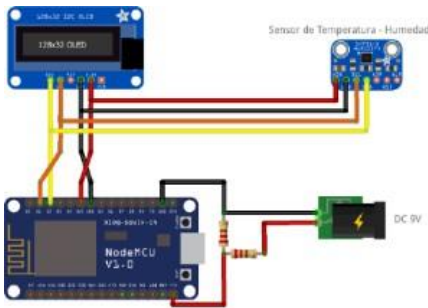
#### ***Diseño e Implementación de Termohigrómetro Basado en Tecnología de Internet de las Cosas para El Almacenamiento y Transporte de Productos***

El presente trabajo tiene por objeto diseñar e implementar un dispositivo IOT para el monitoreo de la temperatura y humedad relativa a lo largo de la cadena de suministros alimentaria en las etapas de almacenamiento y transporte. Para tal fin se usó el sensor de temperatura y humedad SHT31, microcontrolador ESP8266 en placa de desarrollo NodeMCU® v1.0, display OLED®, pulsadores y demás materiales requeridos. Para el diseño del dispositivo se diferenciaron dos tipos de implementación: en almacenamiento y en transporte, puesto que para transporte se requieren adicionalmente dispositivos con conexión GSM-WiFi además de módulo de lectura y escritura de memoria MicroSD®, mientras que en almacenamiento se requiere únicamente conexión WiFi disponible. Los datos son enviados a servidores especializados, los cuales a partir de una suscripción activa permiten gestionar la información, para obtener reportes en tablas, gráficos, etc. en formatos digitales.

En cumplimiento del objetivo principal se consiguió implementar el termohigrómetro basado en la tecnología IOT, para la cadena de suministros de productos en las etapas de almacenamiento y transporte, asegurando el menor costo posible sin que la funcionalidad esperada no se vea afectada por ningún motivo (Pedro Nicoyani Mamani O, 2020).

## Figura 27

### Diagrama de Conexión del Dispositivo



Nota. Diagrama de conexión del dispositivo. Tomado de. (Pedro Nicoyani Mamani O, 2020).

## Diseño e Implementación de un Sistema de Control de Temperatura para un Galpón de Pollos de la Avícola Florián e Hijos S.A.C

El presente proyecto se realizó con el objetivo de diseñar un sistema de control de temperatura en un galpón de pollos de la Avícola Florián e Hijos S.A.C. Es por ello, que se utilizó dos sistemas en el espacio de investigación, los cuales fueron de ventilación y calefacción además de un Control Proporcional Integral Derivativo (PID) encargado de mantener la temperatura adecuada para las aves. El proyecto se apoyó en estudios anteriores para la determinación de la temperatura adecuada según el día de desarrollo de los pollos. Se logró tener un diagnóstico inicial de la situación inicial de automatización en la avícola, de esta manera se empleó nuevos aparatos para la correcta recolección de datos y funcionalidad de los sistemas y programador lógico programable.

La investigación concluye con el análisis de tiempo en que el PID logra regularizar la temperatura ante perturbaciones que se puedan presentar, de igual manera se consigue conocer el comportamiento y la variación que a este le toma para dicho comportamiento.

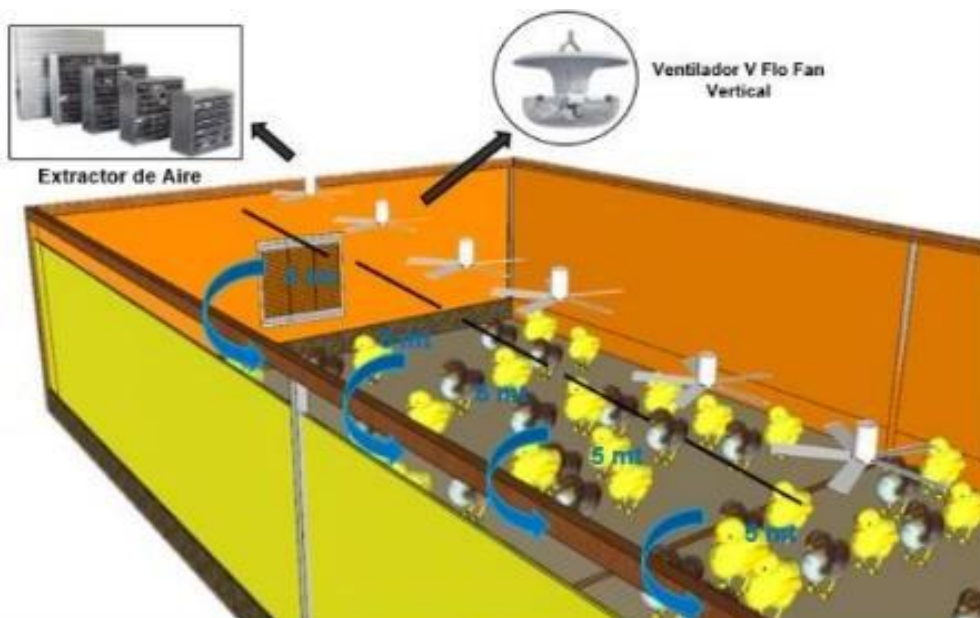
Se diseñó y un sistema de control de temperatura para un galpón de pollos en la avícola Florián e hijos S.A.C teniendo en cuenta los rangos de temperatura adecuada según los días de crecimiento de los

pollos que se encuentran en la (tabla 1) controlando las condiciones

medioambientales de los pollitos con ayuda de un sistema de ventilación y calefacción guiándonos de la tabla 1 donde los rangos de temperatura a seguir son de 28 °C a 32 °C generado en el ambiente de 32 °C para pollitos de edad de que si sobrepasa se encenderá los ventiladores y evitar también que la temperatura baje a menos de 28 °C para ello se encenderá el calentador para aumentar la temperatura. Al final de la construcción del prototipo con circuito de control de temperatura, se verificó que no había necesidad de encender y apagar el calentador o ventilador, sino que se encendían y apagaban automáticamente según el rango de temperatura (León, Gutiérrez H, Jacinto Paredes, & Miranda Mego, 2021).

### Figura 28

#### *Sistema Calefacción*



*Nota.* La imagen muestra un Sistema calefacción. Tomado de. (León, Gutiérrez H, Jacinto Paredes, & Miranda Mego, 2021).

### **Diseño e Implementación de un Sistema de Riego Automatizado y Controlado de Forma**

### **Inalámbrica para una Finca Ubicada en el Sector Popular de Balerio Estacio**

En mayo de 2013 los estudiantes de la facultad de ingeniería de la universidad politécnica Salesiana de Ecuador, Juan Carlos Vasconez Cuzco y Felipe de Jesús Chamba Tenemaza Realizan un sistema capaz de controlar y monitorear de forma inalámbrica En la banda de Frecuencia libre de 2.4GHz, haciendo uso del módulo Wi-Fi Xbee.

Este proyecto buscaba Reducir el tiempo y la necesidad que el agricultor este siempre en constante interacción con los cultivos, estar atento a si falta Riego a este, monitoreando de forma autónoma o automatizada el nivel de humedad en el suelo y enviando esa información hacia el Microcontrolador.

Este proyecto se creó por la necesidades presentes en la explotación del suelo y sobre todo por la constancia de permanecer en lugares donde se necesita riego constante y controlado, con el fin de que los productos en el agro no se dañen o peor aún que se pierdan en su totalidad, nace la necesidad de automatizar un sector a través de un sistema de riego, el mismo que controla las diferentes áreas de siembra el cual tienen varios sensores que interactúan con un microcontrolador que tomara las decisiones de riego en el caso de surgir la necesidad. (Vascones Cuzco & Chamba Tenemaza, 2013).

## **Ntecedentes Locales**

### ***Monitoreo de Temperatura en el Aula de Clase Basada en Arduino***

En la Institución Educativa General Santander, ubicada en el municipio de Soacha, Cundinamarca, se diseñó e implementó una estación meteorológica portátil con sensor de temperatura para el aula de clase.

El sistema, programado con Arduino, permitió a los usuarios obtener información meteorológica en tiempo real. Los datos recopilados revelaron que la temperatura en las aulas oscila entre 24°C y 29°C. La investigación, además de resumir las condiciones ambientales actuales y los problemas existentes en las aulas, evidencia cómo la integración de sensores al sistema permite el monitoreo y cómputo del nivel

de temperatura. Al crear un ambiente de aprendizaje con una temperatura adecuada, se puede fomentar el éxito académico y el desarrollo personal de los estudiantes (Pinzón & Rodríguez González, 2024).

## **Sistema de Monitoreo de Temperaturas y Presiones para el Área de Generación de Frío de la Planta Procesadora de Colanta en San Pedro de los Milagros**

La cooperativa Colanta es líder en el sector agroindustrial, especialmente en la pasteurización de leche y en el procesamiento de derivados lácteos. Entre sus diferentes sedes en toda Colombia, se encuentra en San Pedro de los Milagros la más grande procesadora de derivados lácteos y leche en polvo. A nivel interno se encuentran diferentes áreas, encargadas del correcto funcionamiento de la maquinaria y de la calidad de los productos. Una de estas áreas fundamentales es la de montajes y mantenimiento, dentro de la cual se encuentra el área de servicios que se encarga de todo lo relacionado con generación de vapor, distribución de aire comprimido y garantizar las temperaturas de enfriamiento. Es importante que continuamente se supervise el funcionamiento de la maquinaria empleada, tales como elementos sensores, por este motivo se busca implementar un sistema de supervisión completo, en el cual se pueda visualizar en todo momento desde una estación de trabajo, los valores de las variables más importantes.

Este proyecto implementa una solución para el monitoreo de temperatura y presión de los compresores usados para la generación de frío, que determinan la temperatura en cavas de maduración y almacenamiento basándose en el desarrollo de un sistema SCADA con un PLC SLC5/03 de la familia Allen Bradley y los softwares RSlogix500 y RSview32. Cooperativa Colanta La cooperativa Colanta es líder en agronegocios, especialmente con pasteurización de leche y procesamiento de lácteos. Tiene sedes en toda Colombia, en San Pedro de los Milagros es el mayor fabricante de lácteos y leche en polvo. Internamente, hay diferentes áreas, que son responsables del correcto funcionamiento de las máquinas y la calidad del producto. Una de estas áreas es el área de instalación y mantenimiento, que

incluye el área de servicio, responsable de la generación de vapor, distribución de aire comprimido y garantiza las temperaturas de enfriamiento. Es importante monitorear continuamente el correcto funcionamiento de las máquinas empleadas, como sensores, válvulas automáticas y los diferentes tipos de control utilizados. Este proyecto, por lo tanto, busca implementar un sistema de monitoreo; los valores de las variables más importantes para registrarlos y sus variaciones a lo largo del día. Este proyecto implementa una solución para la monitorización de la temperatura y la presión de los compresores utilizados para la generación de frío, que determinan la temperatura de las bodegas de maduración y almacenamiento, basada en el desarrollo de un sistema SCADA con un PLC SLC5/03 de la familia Allen Bradley y los softwares RSlogix500 y RSview32 (Arcila, 2023).

## **Diseño Metodológico**

Para el diseño e implementación de este proyecto se desglosa en cuatro fases, las cuales se describen a continuación:

### **Fase de Investigación y Recolección de Información**

Para el desarrollo del sistema de monitoreo de temperatura en la cadena de frío, se llevó a cabo un proceso de investigación enfocado en las condiciones térmicas requeridas para la conservación de los productos. Se analizaron factores ambientales que pueden influir en la estabilidad de la temperatura dentro de las cavas de almacenamiento y se estudiaron los estándares de control aplicables en la industria.

Durante esta etapa, también se evaluaron las tecnologías y protocolos de comunicación adecuados para la transmisión de datos en tiempo real, priorizando el uso del protocolo I2C para la interconexión eficiente de los dispositivos. Se seleccionaron los componentes clave para la implementación del sistema, incluyendo:

Proteus, para la simulación del circuito y validación del diseño.

NodeMCU ESP8266, como microcontrolador principal para la adquisición y envío de datos.

Sensor de temperatura DHT11, para la medición precisa de la temperatura.

Protocolo I2C, para la comunicación eficiente entre sensores y el microcontrolador.

ThingSpeak, como plataforma para la visualización remota y el envío de alertas por correo electrónico.

### **Fase de análisis y Definición de Parámetros**

En esta etapa, se establecieron los valores de referencia para el monitoreo de temperatura en la conservación de los productos y se definieron los umbrales críticos que activarían las alertas.

También se analizaron los beneficios del protocolo I2C, el cual permite una comunicación más

rápida y estable entre múltiples dispositivos, facilitando la transmisión eficiente de los datos capturados por los sensores hacia el NodeMCU ESP8266 y posteriormente a ThingSpeak.

### **Fase de Diseño e Implementación del Sistema**

Con base en el análisis previo, el sistema se desarrolló en cinco etapas:

La primera etapa es el diseño del circuito en Proteus, integrando componentes clave como el sensor de temperatura DHT11, el NodeMCU ESP8266, una pantalla LCD para la visualización de datos, y un sistema de alertas compuesto por Leds de colores (rojo, amarillo y verde) y un Buzzer. Cada uno de estos elementos desempeña un papel fundamental en el proyecto, permitiendo la medición, transmisión y señalización de la temperatura en la cadena de frío.

Para la segunda etapa se configuró la comunicación entre los dispositivos mediante el protocolo I2C, asegurando una transmisión estable y eficiente de los datos del sensor de temperatura hacia el NodeMCU ESP8266. Este protocolo facilita la integración de múltiples sensores y dispositivos en un solo bus de datos, optimizando el rendimiento del sistema.

En la tercera etapa se desarrolló el código en Arduino IDE, configurando las funciones necesarias para la lectura del sensor de temperatura DHT11, el procesamiento de datos y la comunicación a través de I2C. Además, se implementó la conexión a internet mediante Wi-Fi, permitiendo el envío de los datos a la plataforma ThingSpeak utilizando el protocolo HTTP.

En la cuarta etapa se creó un canal en ThingSpeak para almacenar y visualizar los datos en tiempo real. Se generaron claves API para la transmisión segura de la información desde el NodeMCU ESP8266. También se configuraron notificaciones por correo electrónico utilizando el análisis de Matlab, la Aplicación TimeControl para realizar una acción en un momento específico o en un horario regular, y el React para realizar acciones cuando los datos del canal cumplen una determinada condición de modo que, si la temperatura excede los umbrales establecidos, se envíe una alerta automática a los

técnicos de mantenimiento por correo electrónico.

Por último, en la quinta etapa, se llevó a cabo el montaje a escala de un cuarto frío, donde se ajustaron las temperaturas de manera controlada para comprobar que el sistema generara alertas cuando los valores estuvieran fuera de los rangos establecidos. incluyendo el NodeMCU ESP8266, el sensor de temperatura DHT11, los indicadores LED, un Buzzer para alertas acústicas y el sistema de alimentación de energía. Finalmente, se realizaron pruebas de conexión y funcionamiento para asegurarse de que cada componente operara correctamente y que el sistema respondiera de manera adecuada a las variaciones de temperatura.

### **Fase de Pruebas y Validación**

Se verifica el correcto desempeño del sistema, para esto se realizaron pruebas funcionales bajo diferentes condiciones ambientales. Se evaluó la precisión de la medición de temperatura, la estabilidad de la comunicación vía I2C, y la correcta transmisión de datos a ThingSpeak.

Se emplearon herramientas estadísticas, como estadística descriptiva y control estadístico de procesos (CEP), para analizar la información recopilada, detectar posibles anomalías y validar la confiabilidad del sistema.

## Ingeniería del Proyecto

### Cuarto Frio Implementado a Escala

Con el fin de llevar a cabo la realización del proyecto se elabora un montaje de un cuarto frio a escala utilizando materiales como Icopor, papel bond, papel fomi e impresión de imágenes con el fin de realizar simulaciones de temperaturas bajas y altas para que nos generen las alertas mediante la tecnología IOT y nos puedan llegar las notificaciones a los correos.

Esta estructura permitió recrear condiciones reales y verificar que el sistema respondiera adecuadamente a los cambios en las variables ambientales, garantizando así su funcionalidad y precisión en la detección de desviaciones, evidencia la estructura del cuarto frio a escala en la figura 29 y 30.

### Figura 29

*Cuarto Frio a Escala Vista al Interior*



*Nota.* En la imagen se evidencia el Cuarto Frio a escala vista al interior

**Figura 30**

*Cuarto Frio Parte Frontal (Autoria, 2025)*



*Nota.* La foto muestra el Cuarto Frio parte frontal

**Figura 31**

*Cuarto Frio Parte Lateral (Autoria, 2025)*

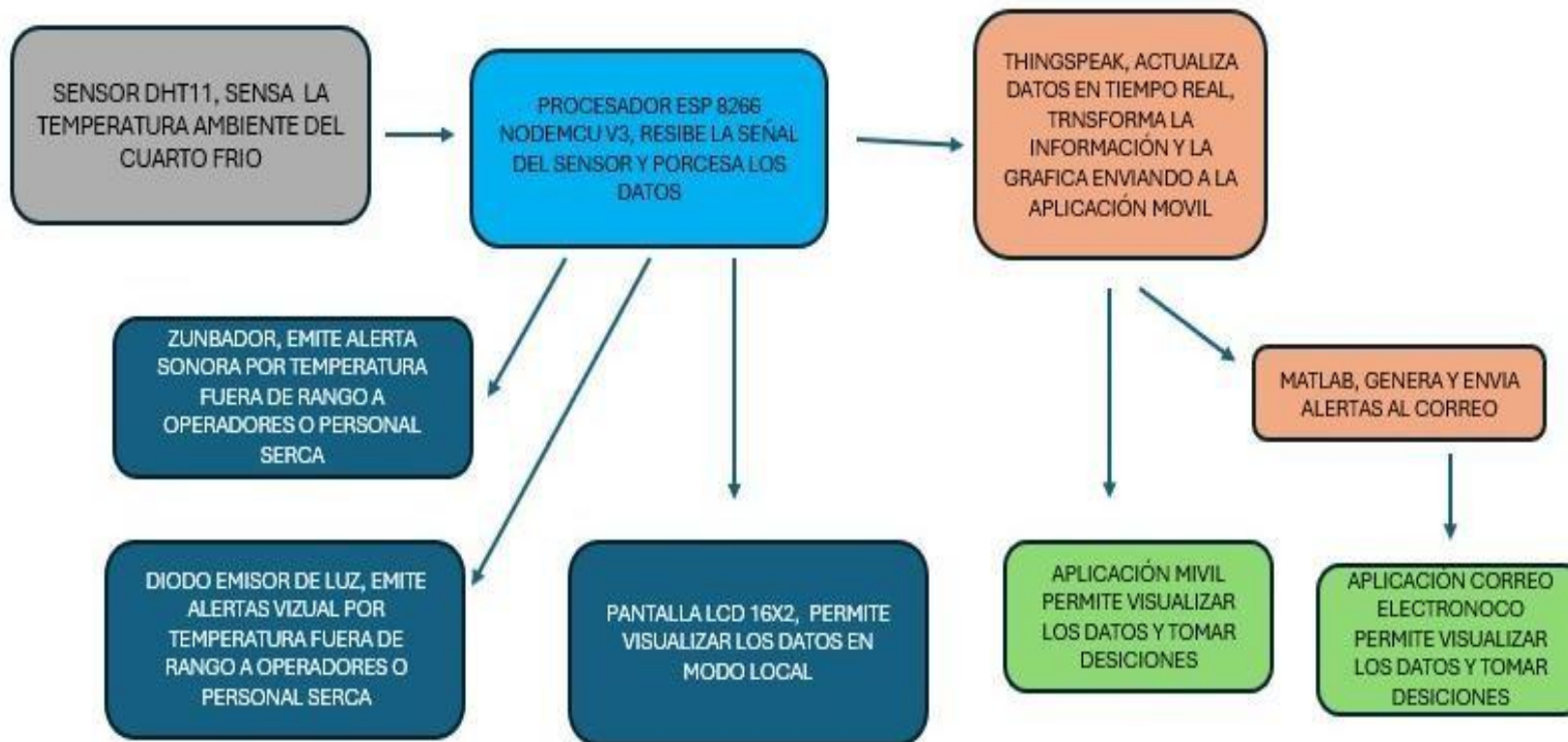


*Nota.* La foto muestra el Cuarto Frio parte lateral

## Diagrama de Comportamientos

Figura 32

Diagrama del Comportamiento de Los Componentes del Cuarto Frio

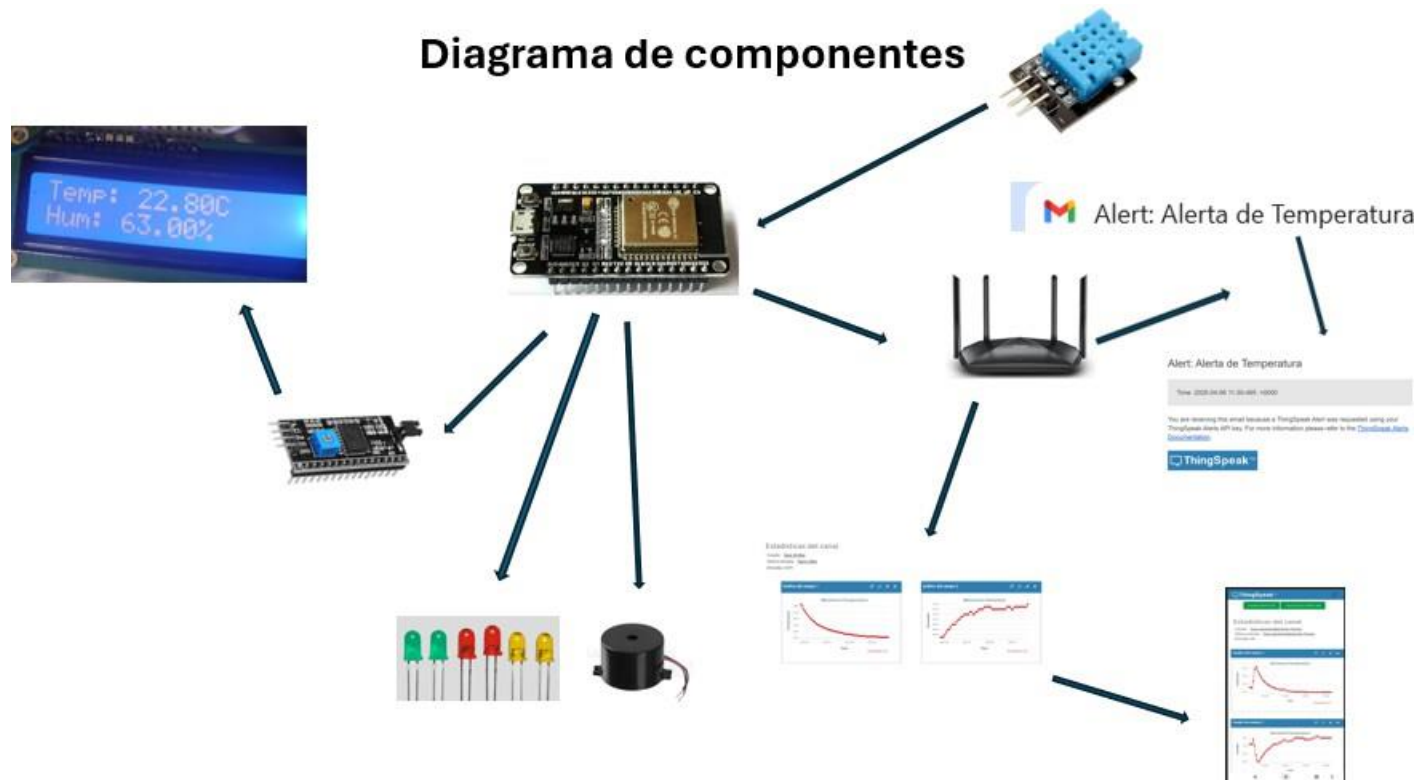


Nota. Diagrama del comportamiento de los componentes del cuarto frio

## Diagrama de Componentes

Figura 33

Diagrama de Componentes (Autoria, 2025)



Nota. Se evidencia el Diagrama de Componentes

## **Sistema de Alertas por Desfase de Temperatura**

Para el diseño del sistema de visualización de la temperatura ambiente en el cuarto frío, se utiliza el sensor DHT11, el cual permite medir la temperatura con un tiempo de respuesta aproximado de 2 segundos. Este sensor cuenta con un pin digital para el envío de datos y dos pines de alimentación (positivo y negativo), lo que facilita su conexión y ayuda a reducir el ruido en las mediciones.

La temperatura del cuarto frío puede verse afectada por diversos factores, como la carga térmica, la dirección del flujo de aire del evaporador y el nivel de hermeticidad del recinto. Debido a estas variables, se hace necesario mantener un monitoreo continuo y en tiempo real de las condiciones térmicas dentro del cuarto frío, con el fin de asegurar un control preciso y confiable.

Por lo anterior se implementó un sistema de monitoreo en tiempo real usando un sensor DHT11, una plataforma IoT (ThingSpeak) y MATLAB para el análisis de datos de temperatura y humedad en una cava. A continuación, se presenta el análisis estadístico y gráfico de los datos registrados durante una ventana de tiempo crítica.

Se puede evidenciar lo siguiente en la figura 34 cómo evoluciona la temperatura con línea de umbral a 28° C, con anotaciones de picos y valles, de igual manera en esta figura se muestra la evolución de la humedad mostrando zonas críticas y finalmente la relación entre la temperatura y la humedad con un comportamiento inverso o directo.

### **Estadísticas Relevantes**

#### ***Temperatura***

- Media: 22.5 °C
- Máxima: 31.2 °C
- Mínima: 10.8 °C
- Desviación estándar: 4.3 °C

## Humedad

- Media: 63.7 %
- Máxima: 97.5 %
- Mínima: 39.2 %
- Desviación estándar: 15.1 %

**Figura 35**

### Grafica de Comportamiento

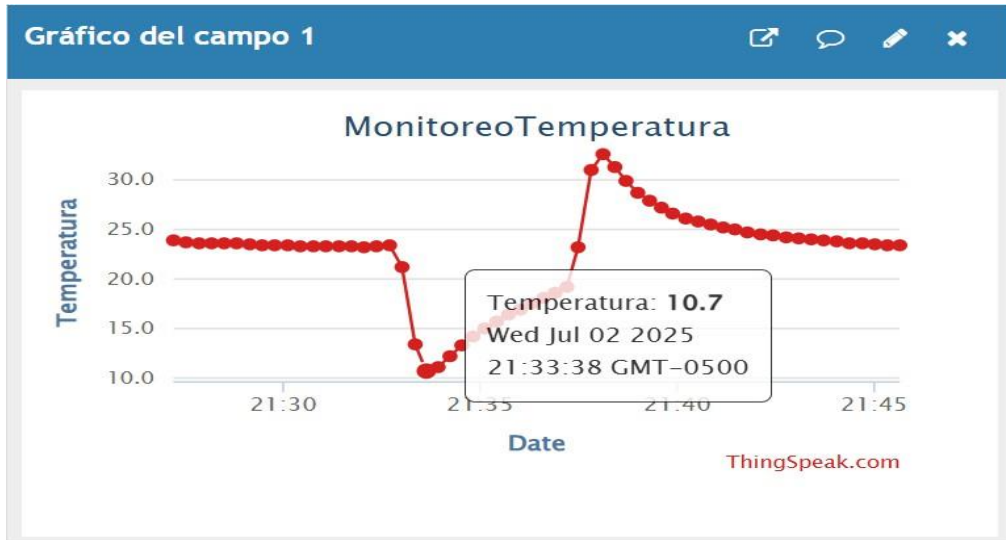


*Nota.* La Grafica muestra el comportamiento del sistema

El análisis de Matlab también nos permite visualizar las gráficas donde se observa la hora y fecha de los eventos:

**Figura 36**

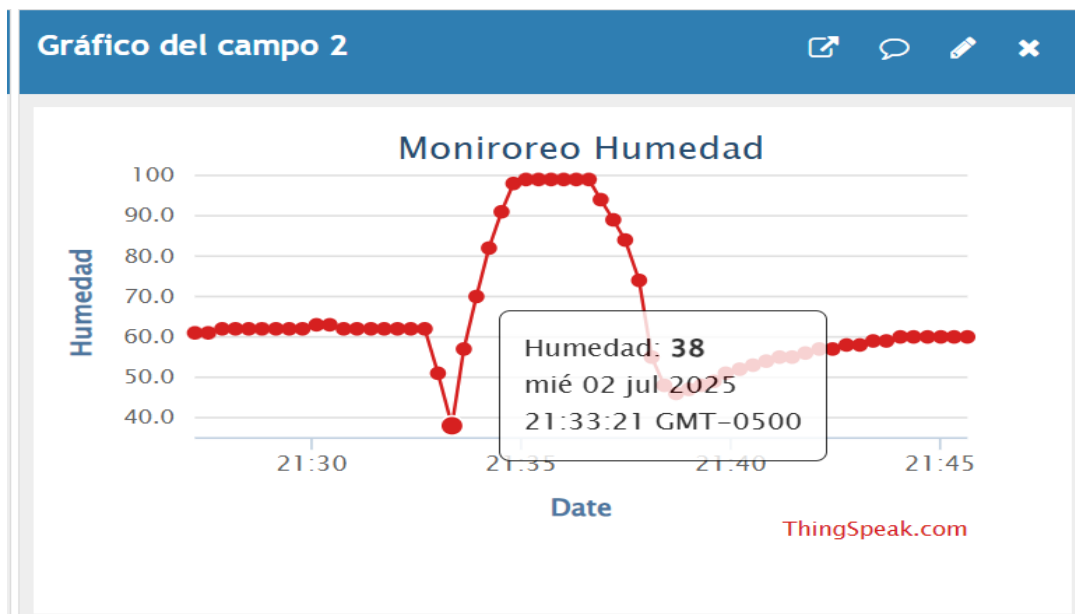
*Monitoreo Temperatura Análisis Matlab*



*Nota.* La grafica se aprecia el Monitoreo Temperatura Análisis Matlab

**Figura 37**

*Monitoreo Humedad análisis Matlab*



*Nota.* La grafica muestra el Monitoreo Humedad análisis Matlab

## Resultados

Durante el monitoreo se detectó una caída súbita de temperatura seguida de un pico abrupto, lo cual puede estar relacionado con la apertura de la cava o un cambio en el sistema de refrigeración. La humedad respondió de manera inversa, alcanzando niveles superiores al 90 %, posiblemente por condensación en respuesta al choque térmico.

Se evidencian valores que superan los umbrales definidos para conservación segura (por ejemplo, temperatura  $> 28\text{ }^{\circ}\text{C}$ ), lo cual activa el sistema de alerta y notificación.

## Sistema de Comunicación

Se implementa un sistema de comunicación que permita el monitoreo de las variables físicas como en este caso es la temperatura del cuarto frío y que estas variables se puedan enviar con el uso de la telemetria por que se usó el módulo ESP 8266 NodeMcu el cual se puede programar con el IDE de Arduino y permitió la comunicación a Thingspeak por medio del WIFI para la visualización de los datos, usando la plataforma de Matlab que permite generar las alertas en tiempo real al correo electrónico, la forma física de este componente se visualiza en la siguiente figura 36.

### NodeMcu Esp8266

#### Figura 38

*NodeMcu Esp8266*



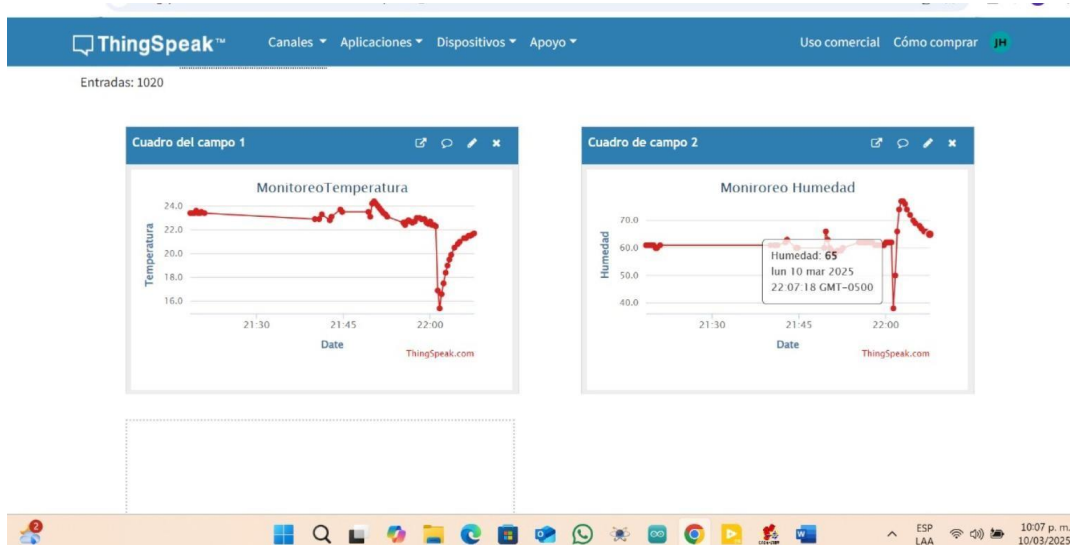
*Nota.* La imagen muestra el NodeMcu Esp8266

## Comunicación con Thingspeak

Para la comunicación de Thingspeak se realiza una serie de instrucciones desde el IDE de Arduino al Nodemcu ESP 8266 nos permitió la visualización de los datos en tiempo real de las magnitudes físicas de temperatura del cuarto frío en la siguiente imagen se puede evidenciar el comportamiento de la temperatura del cuarto frío, como se muestra en las figuras 37.

### Figura 39

#### *Supervisión de Datos en Thingspeak*



*Nota.* Se muestran las gráficas de Supervisión de datos en Thingspeak

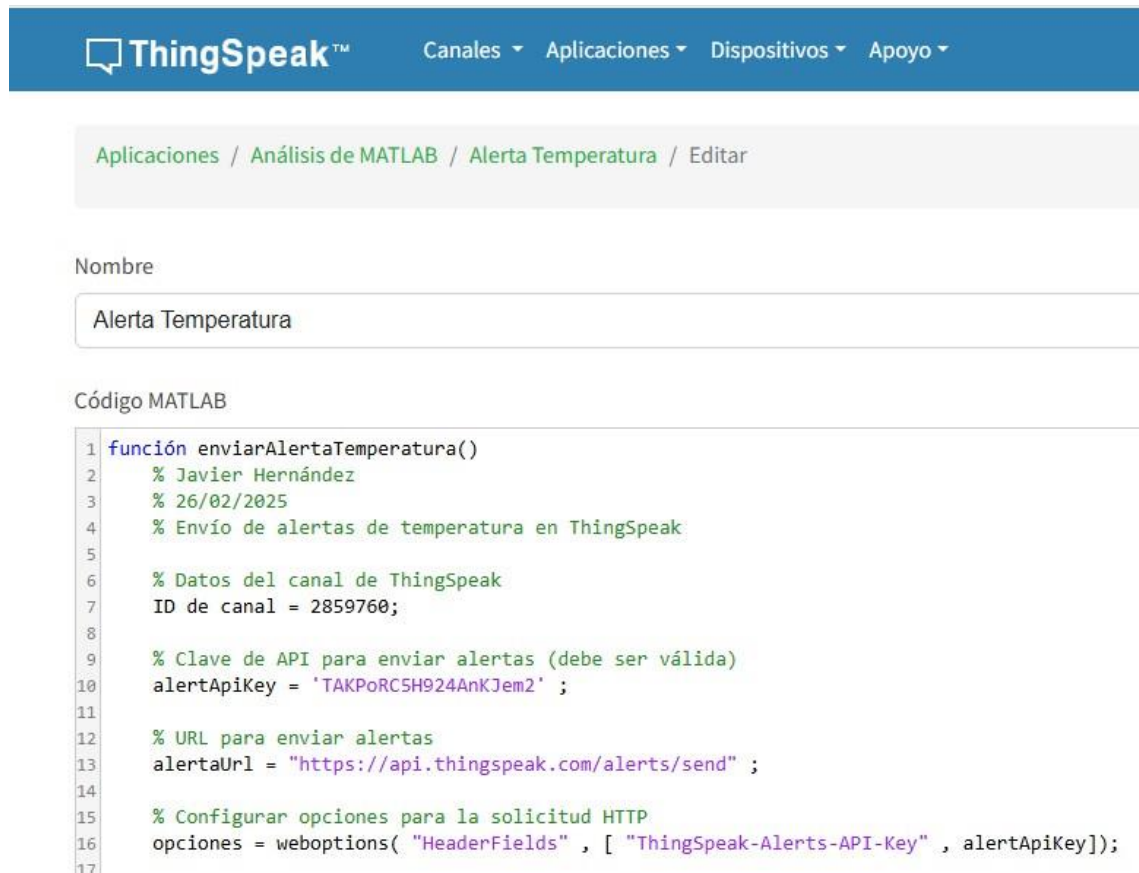
## MATLAB Analysis app

A través de la aplicación MATLAB Analysis se configura una alerta de temperatura que permite filtrar y analizar los datos obtenidos desde un canal de ThingSpeak. Esta herramienta posibilita el cálculo de la temperatura promedio y la eliminación de datos atípicos, mejorando así la precisión del monitoreo.

La alerta se configura mediante un código personalizado que utiliza la clave API del canal y la función `alertUrl` de ThingSpeak, lo que permite establecer límites de temperatura definidos. Cuando se detecta una variación fuera de los parámetros establecidos, el sistema envía automáticamente una alerta en tiempo real al correo electrónico configurado, garantizando una respuesta oportuna y eficiente ante posibles fallas en la cadena de frío, En la siguiente figura 38 se puede evidenciar la aplicación.

**Figura 40**

*Matlab Analysis app*



*Nota.* Código Matlab Analysis app. Tomado de. (*thingspeak, 2021*).

## **TimeControl**

La aplicación TimeControl trabaja en conjunto con otras herramientas de ThingSpeak, como MATLAB Analysis, para facilitar el control de la temperatura. En este proyecto, se configuró TimeControl con una frecuencia diaria para enviar alertas programadas a las 6:30 a.m., permitiendo al personal iniciar la jornada con información actualizada del sistema.

Sin embargo, si se presenta alguna anomalía fuera de los parámetros establecidos, el sistema no espera a la alerta programada: gracias a la integración con la función React, se genera una notificación automática en tiempo real, lo que permite una respuesta inmediata ante cualquier situación crítica, La configuración se realizó como se muestra en las figuras 39 y 40.

## Figura 41

### TimeControl

The screenshot shows the configuration interface for a TimeControl application. The breadcrumb trail is 'Aplicaciones / Control de tiempo / controlar la temperatura / Editar'. The configuration fields are as follows:

- Nombre:** control temperatura
- Huso horario:** UTC (editar)
- Frecuencia:**  Una vez,  Periódico
- Reaparición:**  Semana,  Día,  Hora,  Minuto
- Tiempo:** 6, 30, soy (selected)
- Tiempo difuso:** ± 0 minutos
- Acción:** Análisis de MATLAB
- Código a ejecutar:** Alerta Temperatura

A green button labeled 'Ahorre tiempo y controle' is located at the bottom of the configuration form.

*Nota.* Figura con la configuración del TimeControl

## Figura 42

### TimeControl

The screenshot shows the 'Editar TimeControl' page. A green button labeled 'Editar TimeControl' is at the top. Below it is a table summarizing the configuration:

Nombre:	controlar la temperatura
Frecuencia:	A diario
Huso horario:	UTC (editar)
Última ejecución:	
Lanzarse sobre:	07/04/2025 6:30 a. m.
Tiempo difuso:	± 0 minutos
Análisis MATLAB:	Alerta Temperatura
Creado:	01/03/2025 00:02

*Nota.* Configuración del TimeControl

## Función React

La función React de ThingSpeak permite activar alertas automáticas cuando los datos del canal cumplen ciertas condiciones previamente definidas. En este proyecto, se configuraron dos alertas: una para detectar temperaturas altas y otra para identificar temperaturas bajas, ambas fuera del rango establecido.

Para fines de demostración, se estableció que la alerta por temperatura alta se active cuando el valor supere los 27 °C, y la alerta por temperatura baja, cuando el valor sea inferior a 20 °C. Estos rangos fueron ajustados intencionalmente para que los cambios se simulen más rápidamente y la funcionalidad del sistema pueda demostrarse en menos tiempo.

En un entorno real de conservación de productos, los rangos de temperatura suelen ser más estrictos, ubicándose entre 1 °C y 11 °C, La configuración se realizó como se muestra en las figuras de la 41 – 45.

### Figura 43

#### *Ajuste Temperaturas Alta y Baja*

Nombre	Creado	Última carrera
<input checked="" type="checkbox"/> Alerta Temperatura Alta <input type="button" value="Vista"/> <input type="button" value="Editar"/>	14 de marzo de 2025	2025-04-02 15:27
<input checked="" type="checkbox"/> Alerta Temperatura Baja <input type="button" value="Vista"/> <input type="button" value="Editar"/>	14 de marzo de 2025	2025-04-02 15:27

*Nota.* Ajuste temperaturas Alta y Baja

## Figura 44

### *Función React Temperatura Alta (Autoria, 2025)*

The screenshot shows the configuration interface for a React function in ThingSpeak. The page title is 'Alerta Temperatura Alta' and it is under the 'Reaccionar' (React) section. The configuration includes:

- Nombre de React:** Alerta Temperatura Alta
- Tipo de condición:** Numérico
- Frecuencia de prueba:** Sobre la inserción de datos
- Condición:** Si canal: MonitoreoTemperatura (2859760); campo: 1 (Temperatura); es mayor o igual a: 27
- Acción:** Análisis de MATLAB
- Código a ejecutar:** Alerta Temperatura
- Opciones:**  Ejecutar la acción solo la primera vez que se cumpla la condición;  Ejecutar la acción cada vez que se cumpla la condición

*Nota.* Función React Temperatura Alta

## Figura 45

### *Función React Temperatura Alta*

The screenshot shows the summary page for the React function 'Alerta Temperatura Alta'. It includes an 'Edit React' button and the following details:

Name:	Alerta Temperatura Alta
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2025-04-02 15:27
Channel:	MonitoreoTemperatura
Condition:	Field 1 (Temperatura) is greater than or equal to 27
MATLAB Analysis:	Alerta Temperatura
Run:	Each time the condition is met
Created:	2025-03-14 1:16 am

*Nota.* Función React Temperatura Alta

**Figura 46**

*Función React Temperatura Baja (Autoria, 2025)*

The image shows the configuration page for a React function in ThingSpeak. The page has a blue header with the ThingSpeak logo and navigation links: Canales, Aplicaciones, Dispositivos, and Apoyo. Below the header is a breadcrumb trail: Aplicaciones / Reaccionar / Alerta Temperatura Baja / Editar. The main content area contains several configuration fields:

- Nombre de React:** Alerta Temperatura Baja
- Tipo de condición:** Numérico
- Frecuencia de prueba:** Sobre la inserción de datos
- Condición:**
  - Si canal: MonitoreoTemperatura (2859760)
  - campo: 1 (Temperatura)
  - es menor o igual a: 20
- Acción:** Análisis de MATLAB
- Código a ejecutar:** Alerta Temperatura

*Nota.* Función React Temperatura Baja

## Figura 47

### *Función React Temperatura Baja*



The screenshot shows the ThingSpeak web interface. At the top, there is a blue navigation bar with the ThingSpeak logo and menu items: 'Canales', 'Aplicaciones', 'Dispositivos', and 'Apoy'. Below the navigation bar, a breadcrumb trail reads 'Aplicaciones / Reaccionar / Alerta Temperatura Baja'. A green button labeled 'Editar Reaccionar' is positioned above a table of configuration details.

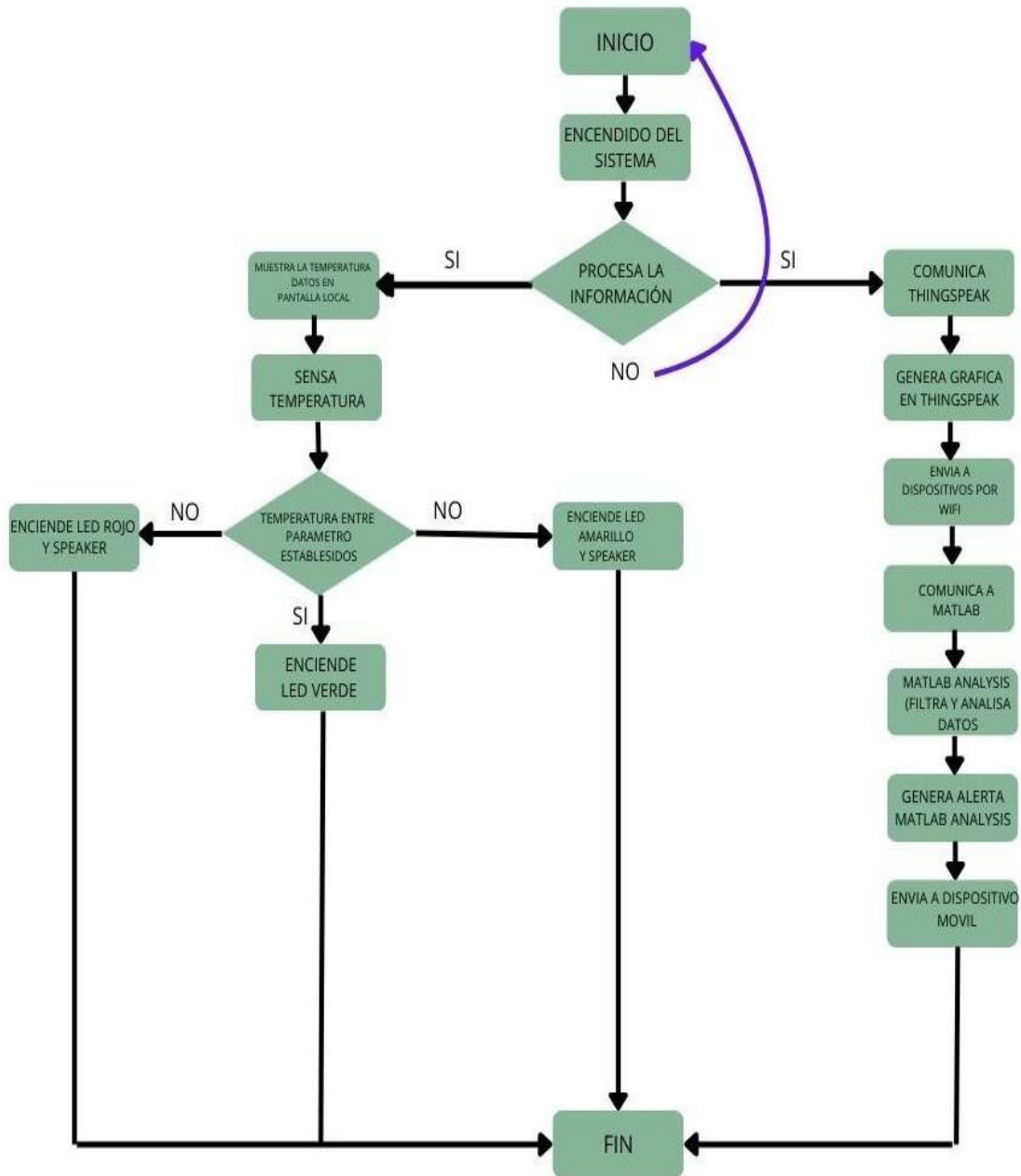
Nombre:	Alerta Temperatura Baja
Tipo de condición:	Numérico
Frecuencia de prueba:	Sobre la inserción de datos
Última ejecución:	2025-04-02 15:27
Canal:	<a href="#">Monitoreo de temperatura</a>
Condición:	El campo 1 (Temperatura) es menor o igual a 20
Análisis MATLAB:	<a href="#">Alerta Temperatura</a>
Correr:	Sólo la primera vez que se cumple la condición
Creado:	14/03/2025 1:17 a. m.

*Nota.* Función React Temperatura Baja

# Diagrama de Flujo Funcionamiento de la Aplicación

Figura 48

Diagrama de Flujo



Nota. Diagrama de flujo

## Componentes y Conexiones

Para realizar un sistema de automatización se pueden usar diferentes tipos de conexiones en diversos voltajes y corrientes, como también las señales en digitales y analógicas, en este caso se hicieron conexiones entre la tarjeta NodeMcu ESP8266 que permite una comunicación a internet usando el protocolo Serial configurado a 115200 baudios para una comunicación con el IDE de Arduino y una comunicación a una pantalla de monitoreo y visualización local con el protocolo I2C.

Se realiza un paso a paso, se inicia con un sensor DHT11 alimentado a un voltaje de 3V con una conexión a tierra y una señal digital, este elemento tiene la función de sensar la temperatura del ambiente el cuarto frío.

Para la visualización y monitoreo local se utiliza una pantalla LCD de 16\*2 dieciséis columnas por 2 líneas para los caracteres acoplado a un módulo I2C para un uso mínimo de terminales digitales alimentado a un voltaje de 3VDC una conexión a tierra y los pines digitales de comunicación CDA y SCL.

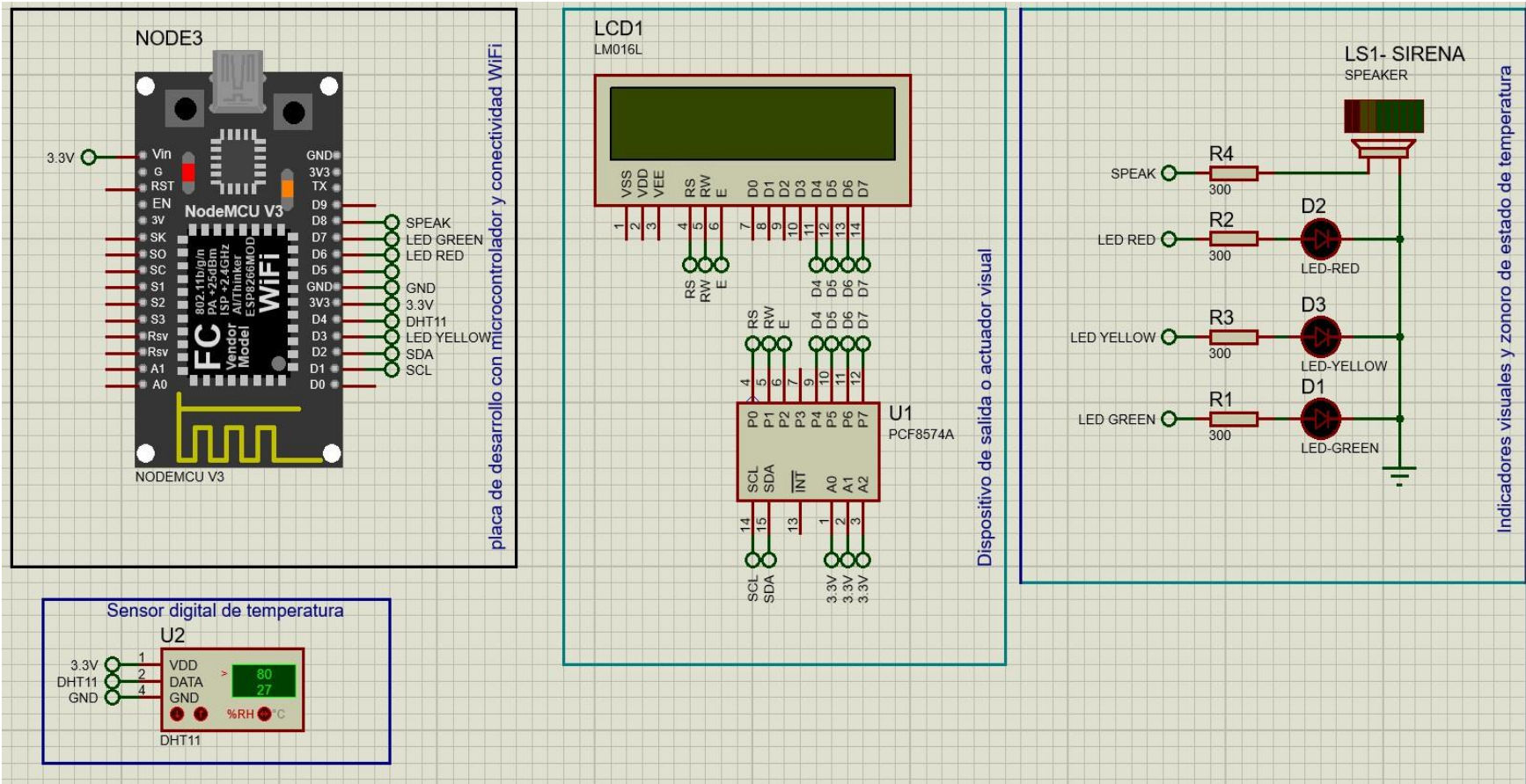
Para el procesamiento de los datos y la comunicación se usa una placa de desarrollo NodeMcu v3 de microcontrolador y conexión a WIFI lo que la hace ideal para este proyecto por sus diferentes conexiones digitales y analógicas que permiten acoplar diferentes componentes, esta tarjeta cuenta con una alimentación de voltaje de 3.3VDC y una puesta a tierra lo que la hace versátil al momento de su implementación.

En la generación de las alertas se utilizan diodos emisores de LUZ, estos leds se implementan en diferentes colores dependiendo del tipo de función y con una conexión a terminales digitales con su respectiva resistencias a 300 ohmios des de la tarjeta de desarrollo y una puesta a tierra, en este caso se utilizó el led de color rojo para indicar cuando la temperatura es alta y fuera de especificación poniendo en riesgo el producto, se implementa un led de color amarillo para indicar que la temperatura es baja por fuera de especificaciones poniendo en riesgo el producto, se implementa un led verde para indicar cuando la temperatura está dentro de parámetros establecidos y el producto está en la temperatura de conservación adecuada.

# Diagrama de Conexión de Tarjeta Nodemcu y Componentes

Figura 49

Diagrama de Conexión de Tarjeta Nodemcu y Componentes



Nota. Diagrama de conexión de tarjeta NodeMcu y componentes

## Descripción del Código Arduino IDE

### Comunicación de Modulo NodeMCU con Thingspeak

Para establecer la comunicación entre el módulo NodeMCU ESP8266 V3 y ThingSpeak, se desarrolló un código en el IDE de Arduino que permite la transmisión de datos a través de internet. Gracias a esta conexión, ThingSpeak recibe la información, la analiza y genera gráficas que permiten realizar una trazabilidad del comportamiento de la temperatura en la cava de refrigeración. Esto facilita la toma de decisiones por parte de los técnicos encargados de las intervenciones. En la Figura 48 se muestra la función utilizada para enviar los datos a ThingSpeak.

### Figura 50

#### *Función para Enviar Datos a Thingspeak*

```
void enviarDatosTS(float temp, float hum) {  
  if (WiFi.status() == WL_CONNECTED) {  
    WiFiClient client;  
    if (client.connect(server, 80)) {  
      Serial.println("Conectado a ThingSpeak");  
  
      String postStr = "api_key=" + apiKey;  
      postStr += "&field1=" + String(temp);  
      postStr += "&field2=" + String(hum);  
  
      client.print("POST /update HTTP/1.1\n");  
      client.print("Host: api.thingspeak.com\n");  
      client.print("Connection: close\n");  
      client.print("Content-Type: application/x-www-form-urlencoded\n");  
      client.print("Content-Length: " + String(postStr.length()) + "\n\n");  
      client.print(postStr);  
  
      Serial.println("Datos enviados a ThingSpeak!");  
    } else {  
      Serial.println("Error al conectar con ThingSpeak");  
    }  
    client.stop();  
  } else {  
    Serial.println("No hay conexión WiFi, intentando reconectar...");  
    conectarWiFi();  
  }  
}
```

*Nota.* Función para enviar datos a thingspeak

### Comunicación de Aplicación Análisis Matlab con el Correo

Para enviar los datos por correo electrónico, se configuró la aplicación MATLAB Analysis, la cual es compatible con la plataforma ThingSpeak. Esta configuración se realiza utilizando una clave API de alertas, que se debe ingresar directamente en el código de MATLAB para permitir el envío automático de notificaciones por correo, como se muestra en la siguiente figura número 50

## Figura 51

### Análisis Matlab



```
Código MATLAB
1 función enviarAlertaTemperatura()
2 % Javier Hernández
3 % 26/02/2025
4 % Envío de alertas de temperatura en ThingSpeak
5
6 % Datos del canal de ThingSpeak
7 ID de canal = 2859760;
```

*Nota.* Análisis Matlab

## Configuración Aplicación TimeControl para Alerta al Correo Electrónico

Luego de configurar la aplicación MATLAB Analysis, se realizaron los ajustes necesarios para que las alertas se enviaran automáticamente por correo electrónico. Para ello, se utilizó la aplicación TimeControl, donde se programó el envío diario de las alertas a las 6:30 a.m. Esta configuración se puede observar en la Figura 51 y 52.

## Figura 52

### Configuración TimeControl (Autoria, 2025)



Notify me via email if this MATLAB Analysis fails when triggered by TimeControl or React.

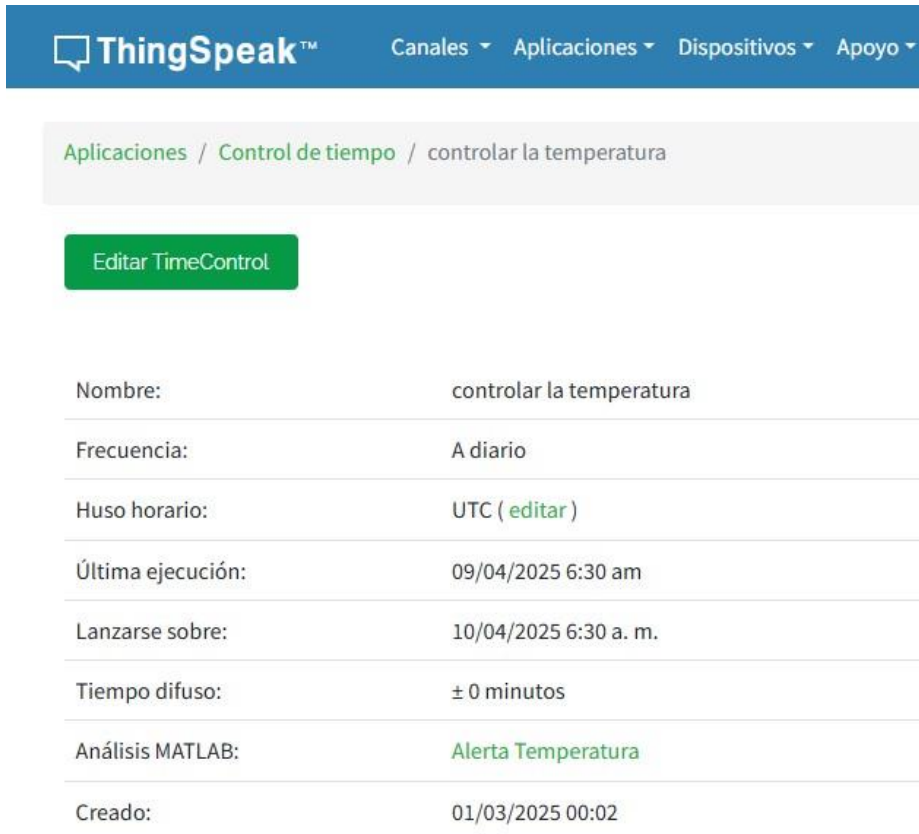
**TimeControl**

Name	Recurrence	Last Ran	Run At
<input checked="" type="checkbox"/> control temperatura	Daily	2025-04-09 6:30 am	2025-04-10 6:30 am

*Nota.* Configuración TimeControl

## Figura 53

### Configuración 2 TimeControl



The screenshot shows the ThingSpeak interface for configuring a TimeControl application. The breadcrumb trail is 'Aplicaciones / Control de tiempo / controlar la temperatura'. A green button labeled 'Editar TimeControl' is visible. Below it is a table of configuration parameters:

Nombre:	controlar la temperatura
Frecuencia:	A diario
Huso horario:	UTC ( <a href="#">editar</a> )
Última ejecución:	09/04/2025 6:30 am
Lanzarse sobre:	10/04/2025 6:30 a. m.
Tiempo difuso:	± 0 minutos
Análisis MATLAB:	<a href="#">Alerta Temperatura</a>
Creado:	01/03/2025 00:02

*Nota.* Configuración 2 TimeControl

### Configuración Aplicación React envió Alerta Correo

Luego de configurar las aplicaciones MATLAB Analysis y TimeControl, se procedió a realizar la configuración de la aplicación React, la cual permite el envío automático de alertas por correo electrónico cada vez que la temperatura supere los 27 °C o descienda por debajo de los 20 °C. En las Figuras 52, 53 y 54 se puede observar la configuración correspondiente para cada uno de estos umbrales de temperatura.

**Figura 54**

*Configuración Temperatura Baja y Alta*

The screenshot shows the ThingSpeak web interface. At the top, there is a blue navigation bar with the ThingSpeak logo and menu items: Canales, Aplicaciones, Dispositivos, and Apoyo. Below the navigation bar, there is a breadcrumb trail: Aplicaciones / Reaccionar. A green button labeled 'Nueva reacción' is visible. The main content area contains a table with two rows of alert configurations.

Nombre	Creado	Última carrera
<input checked="" type="checkbox"/> Alerta Temperatura Alta <a href="#">Vista</a> <a href="#">Editar</a>	14 de marzo de 2025	2025-04-02 15:27
<input checked="" type="checkbox"/> Alerta Temperatura Baja <a href="#">Vista</a> <a href="#">Editar</a>	14 de marzo de 2025	2025-04-02 15:27

*Nota.* Configuración Temperatura Baja y Alta

**Figura 55**

*Configuración Temperatura Alta (Autoria, 2025)*

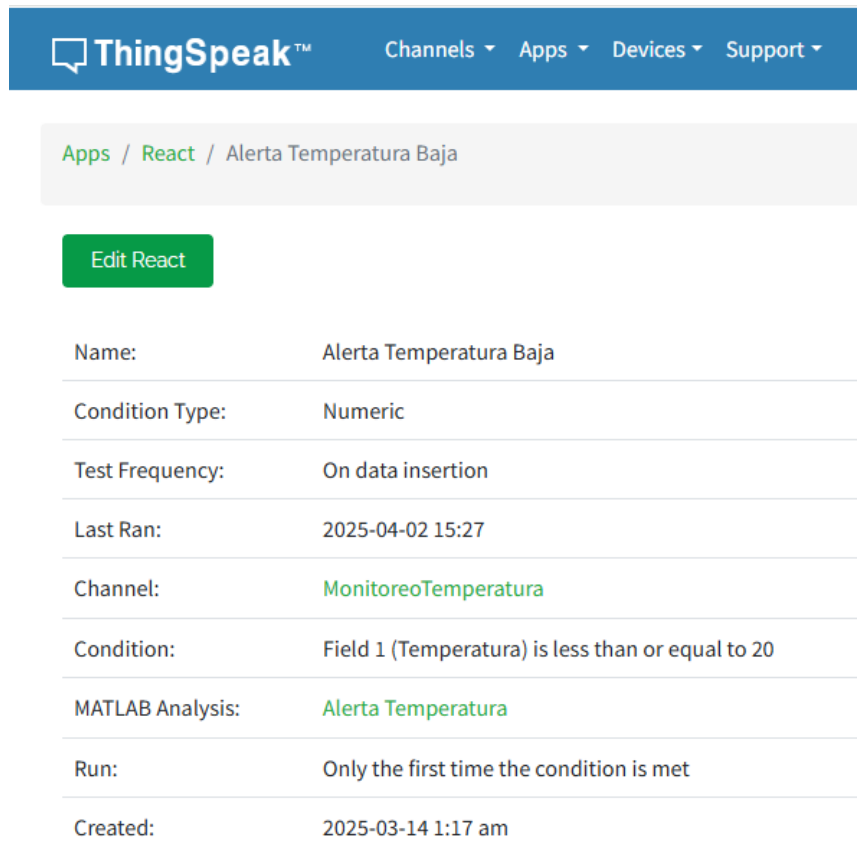
The screenshot shows the configuration details for a high temperature alert in the ThingSpeak interface. The navigation bar includes the ThingSpeak logo and menu items: Channels, Apps, Devices, and Support. The breadcrumb trail is: Apps / React / Alerta Temperatura Alta. A green button labeled 'Edit React' is visible. The configuration details are as follows:

Name:	Alerta Temperatura Alta
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2025-04-02 15:27
Channel:	MonitoreoTemperatura
Condition:	Field 1 (Temperatura) is greater than or equal to 27
MATLAB Analysis:	Alerta Temperatura
Run:	Each time the condition is met
Created:	2025-03-14 1:16 am

*Nota.* Configuración Temperatura Alta

## Figura 56

### Configuración Temperatura Baja



The screenshot shows the ThingSpeak web interface. At the top, there is a blue navigation bar with the ThingSpeak logo and links for Channels, Apps, Devices, and Support. Below this, a breadcrumb trail indicates the current location: Apps / React / Alerta Temperatura Baja. A green 'Edit React' button is visible. The main content area displays the configuration details for the 'Alerta Temperatura Baja' React app in a table-like format.

Name:	Alerta Temperatura Baja
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2025-04-02 15:27
Channel:	MonitoreoTemperatura
Condition:	Field 1 (Temperatura) is less than or equal to 20
MATLAB Analysis:	Alerta Temperatura
Run:	Only the first time the condition is met
Created:	2025-03-14 1:17 am

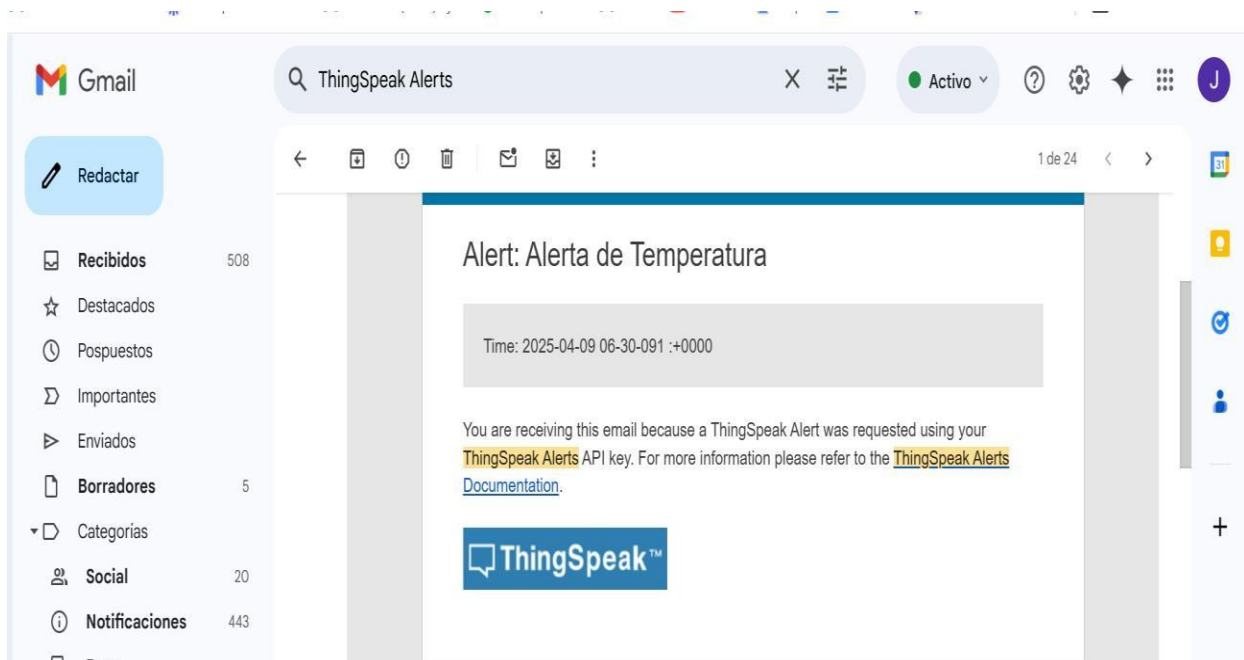
*Nota.* Muestra la Configuración de la Temperatura Baja

### Alertas Correo Electrónico

Después de configurar las aplicaciones MATLAB Analysis, TimeControl y React, el sistema queda habilitado para enviar alertas al correo electrónico, ya sea de forma diaria a las 6:30 a.m. o cuando la temperatura se encuentre fuera de los rangos establecidos. Esta funcionalidad se puede evidenciar en las Figuras desde la 55 hasta la figura 59.

**Figura 57**

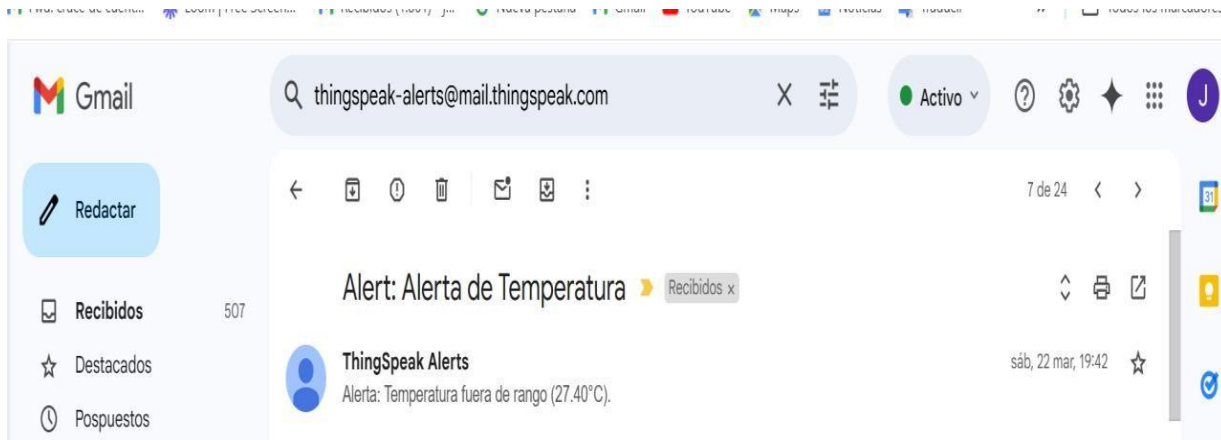
*Alerta Diaria 6:30 am*



*Nota.* Se evidencia la Alerta diaria 6:30am

**Figura 58**

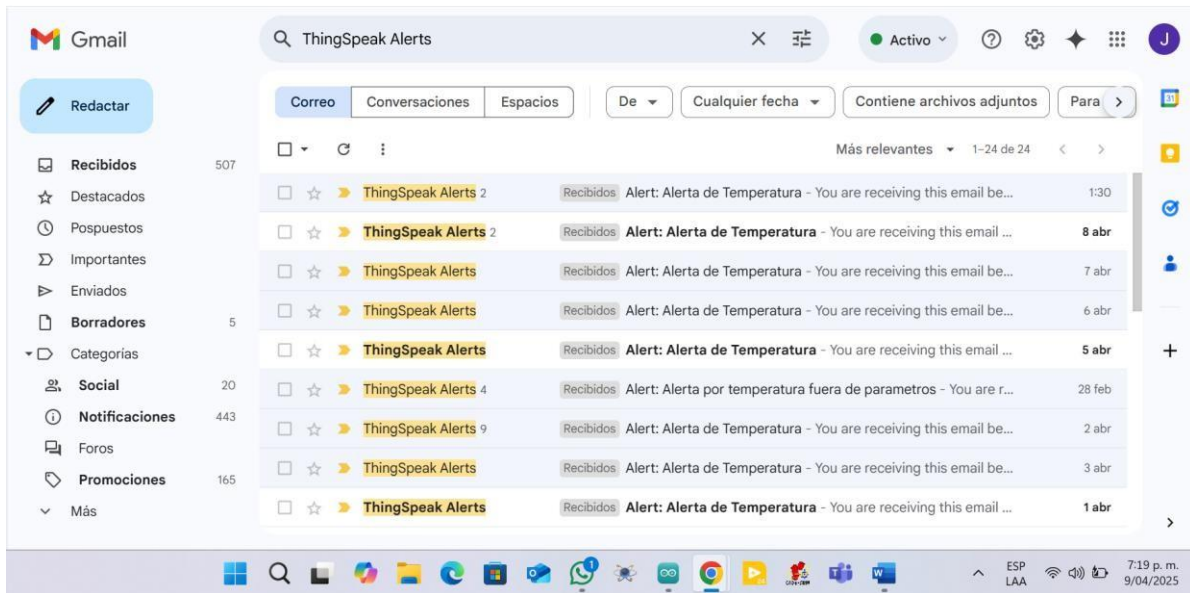
*Alerta Temperatura Fuera de Especificación*



*Nota.* La imagen muestra la Alerta Temperatura fuera de especificación

**Figura 59**

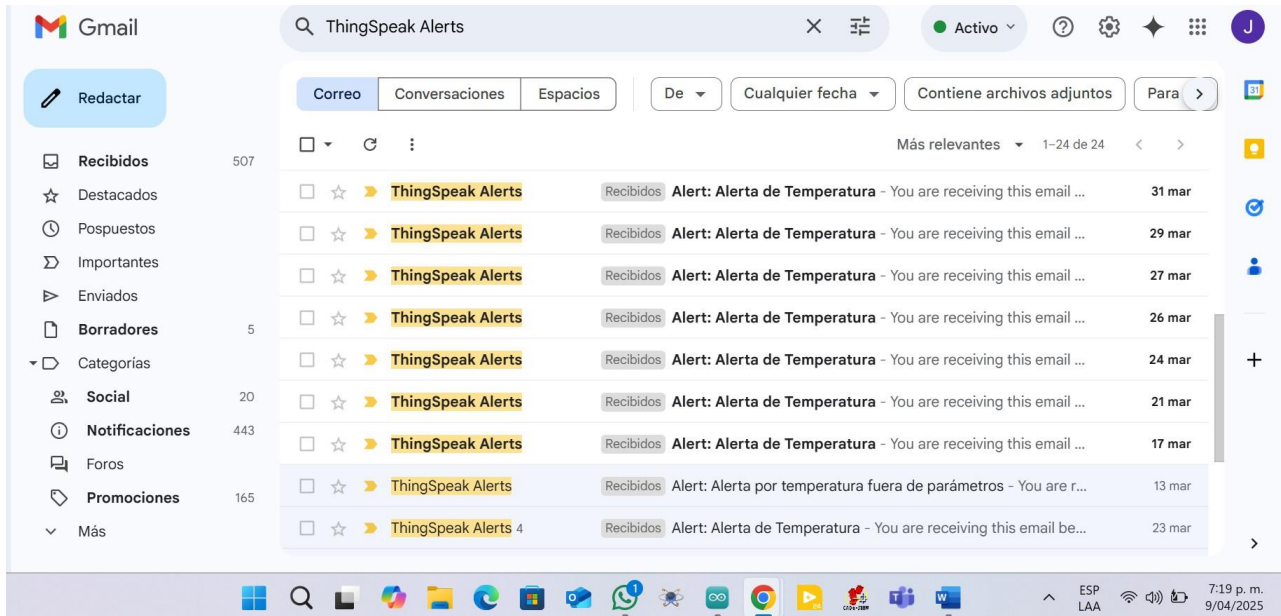
*Evidencias Alertas de Correo*



*Nota.* Se Evidencia las alertas de correo

**Figura 60**

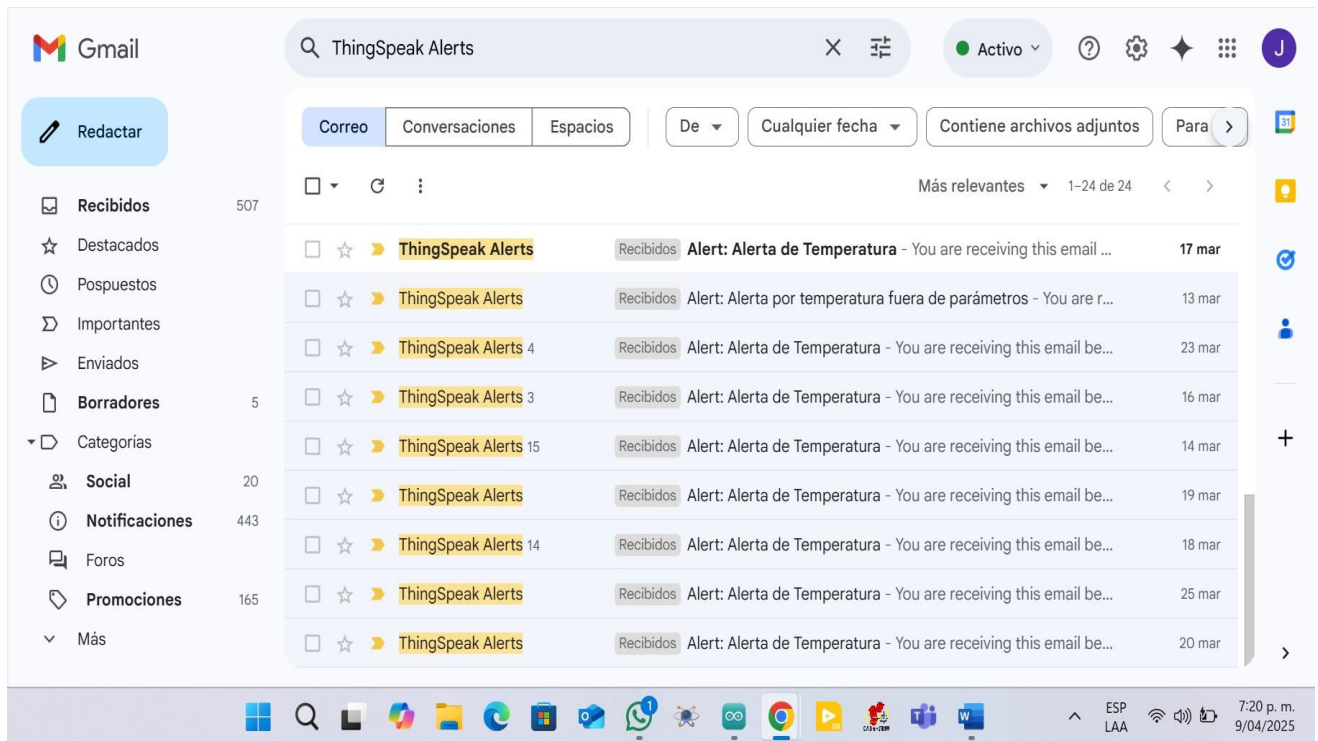
*Evidencia Alerta De Correo 2*



*Nota.* Se está evidenciando las alertas de correo 2

## Figura 61

### Evidencia Alerta Correo 3



*Nota.* Evidencia alerta correo 3

## Función de Alertas Locales Visuales y Sonoras

Para implementar la función de alertas locales, visuales y sonoras, se desarrolló un código en el IDE de Arduino que permite generar respuestas inmediatas ante variaciones en la temperatura del cuarto frío.

Cuando la temperatura ambiente supera los 27 °C, se activa un piloto luminoso de color rojo, acompañado del sonido de una sirena o speaker. Esta doble señal (visual y sonora) advierte sobre una anomalía en la cadena de frío que requiere una intervención técnica inmediata.

Si la temperatura desciende por debajo de los 20 °C, se activa un piloto luminoso de color amarillo, también acompañado por la alarma sonora, indicando nuevamente una condición fuera de los rangos establecidos.

En el caso de que la temperatura se mantenga entre los 20 °C y 27 °C, se enciende un piloto

luminoso de color verde, sin activar la alerta sonora. Esto indica que el sistema se encuentra funcionando dentro de los parámetros normales definidos por el usuario.

De esta forma, se da solución al sistema de alertas locales, facilitando una respuesta rápida ante cualquier desviación térmica. El código correspondiente implementado en el IDE de Arduino se muestra en las Figuras 60, 61, 62 y 63.

## Figura 62

*Función de Alertas Locales Visuales y Sonoras*

```
#define LED1 13 // D7= GPIO13 - color Verde
#define LED2 12 // D6= GPIO12 - color Rojo
#define LED3 0 // D3= GPI15 - color Amarillo
#define SPEAKER 15 // D8= GPIO14 - Buzzer
// Variables
float temp, hume;
unsigned long previousMillis = 0;
const unsigned long interval = 5000; // Intervalo de actualización (5 segundos)
int contconexion = 0;
```

*Nota.* Función de alertas locales visuales y sonoras

## Figura 62

*Función de Alertas Locales Visuales y Sonoras*

```
void loop() {
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        temp = dht.readTemperature(); // Leer temperatura en °C
        hume = dht.readHumidity(); // Leer humedad
```

*Nota.* Función de alertas locales visuales y sonoras

## Figura 63

### *Función de Alertas Locales Visuales y Sonoras*

```
pinMode(LED1, OUTPUT);
pinMode(LED2, OUTPUT);
pinMode(LED3, OUTPUT);
pinMode(SPEAKER, OUTPUT);
```

*Nota.* Función de alertas locales visuales y sonoras

## Figura 64

### *Función de Alertas Locales Visuales y Sonoras*

```
// Función para controlar LED y buzzer según temperatura
void controlarAlertas(float temp) {
  if (temp >= 20 && temp <= 27) {
    digitalWrite(LED1, HIGH); // Enciende LED verde
    digitalWrite(LED2, LOW); // Apaga LED rojo
    digitalWrite(LED3, LOW); // Apaga LED amarillo
    digitalWrite(SPEAKER, LOW); // Apaga el buzzer
  } else if (temp > 27) {
    digitalWrite(LED1, LOW); // Apaga LED verde
    digitalWrite(LED2, HIGH); // Enciende LED rojo
    digitalWrite(LED3, LOW); // Apaga LED amarillo
    tone(SPEAKER, 50); // Buzzer enciende
    delay(1500);
    //noTone(SPEAKER);
  } else if (temp < 20) {
    digitalWrite(LED1, LOW); // Apaga LED verde
    digitalWrite(LED2, LOW); // Apaga LED rojo
    digitalWrite(LED3, HIGH); // Enciende LED amarillo
    tone(SPEAKER, 1500); // Buzzer enciende
    delay(1500);
  }
}
```

*Nota.* Función de alertas locales visuales y sonoras

## Función Visualización de Datos de Temperatura de Cuarto Frio

Para la visualización de los datos locales se configuro en el IDE de Arduino el módulo I2C y la pantalla LCD 16X2 con el que se visualiza el dato de la temperatura en tiempo real. Esta configuración se puede visualizar en las figuras 64 y 65.

## Figura 65

### Configuración I2C

```
// Configuración de LCD I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); // Dirección 0x27 para LCD I2C
```

*Nota.* Muestra la Configuración I2C

## Figura 66

### Configuración Pantalla LCD 16X2

```
// Mostrar datos en la LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(temp);
lcd.print(" C");

lcd.setCursor(0, 1);
lcd.print("Hum: ");
lcd.print(hume);
lcd.print(" %");

// Mostrar datos en el monitor serie
Serial.print("Temp: ");
Serial.print(temp);
Serial.print(" °C | Hum: ");
Serial.print(hume);
Serial.println(" %");

// Control de LED según temperatura
controlarAlertas(temp);

// Enviar datos a ThingSpeak
enviarDatosTS(temp, hume);
}
```

*Nota.* Configuración Pantalla LCD 16X2

## **Función Sensor de Temperatura DHT11**

Para sensar la temperatura ambiente al interior del cuarto frio se implementa una función en IDE de Arduino que permite que el sensor DHT11 tome los datos de la sensación térmica al interior del cuarto y los transfiera a la tarjeta de desarrollo de Nodemcu 8266 v3 para su análisis y proceso, esta configuración se puede evidenciar en la figura 66y 67.

### **Figura 67**

*Función Sensor de Temperatura DHT11*

```
DHT dht(2, DHT11); // DHT11 conectado al pin GPIO2 (D4)
```

*Nota 67.* Función Sensor de temperatura DHT11

### **Figura 68** *Función Sensor de temperatura DHT11*

```
if (isnan(temp) || isnan(hume)) {  
    Serial.println("Error al leer el sensor DHT11");  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print("Error DHT11");  
    return;  
}
```

*Nota.* Función Sensor de temperatura DHT11

## **Librerías de Arduino IDE empleadas en el Código**

Se crea una función para llamar las librerías de IDE Arduino que permiten la configuración del código estas librerías usadas se pueden evidencias en la figura 68.

## Figura 69

*Librerías de Arduino IDE empleadas en el Código*

```
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <ESP8266WiFi.h>
```

*Nota 69.* Librerías de Arduino IDE empleadas en el Código

## Función de IDE Arduino empleadas en el Código para señal WIFI

Se crea la función de IDE de Arduino que permite la conectividad de la señal WIFI con los componentes empleados en el envío de datos, la configuración se puede evidenciar en la figura 70 y 71.

## Figura 70

*Configuración Wifi*

```
// Configuración WiFi
const char* ssid = "DEICY"; // Reemplaza con tu SSID
const char* password = "123456789"; // Reemplaza con tu contraseña
```

*Nota 70.* Configuración Wifi

## Figura 71 Configuración Wifi

```
// Función para conectar a WiFi
void conectarWiFi() {
  WiFi.begin(ssid, password);
  Serial.print("Conectando a WiFi");

  while (WiFi.status() != WL_CONNECTED && contconexion < 50) {
    delay(500);
    Serial.print(".");
    contconexion++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWiFi conectado!");
    Serial.print("Dirección IP: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("\nError de conexión WiFi");
  }
}
```

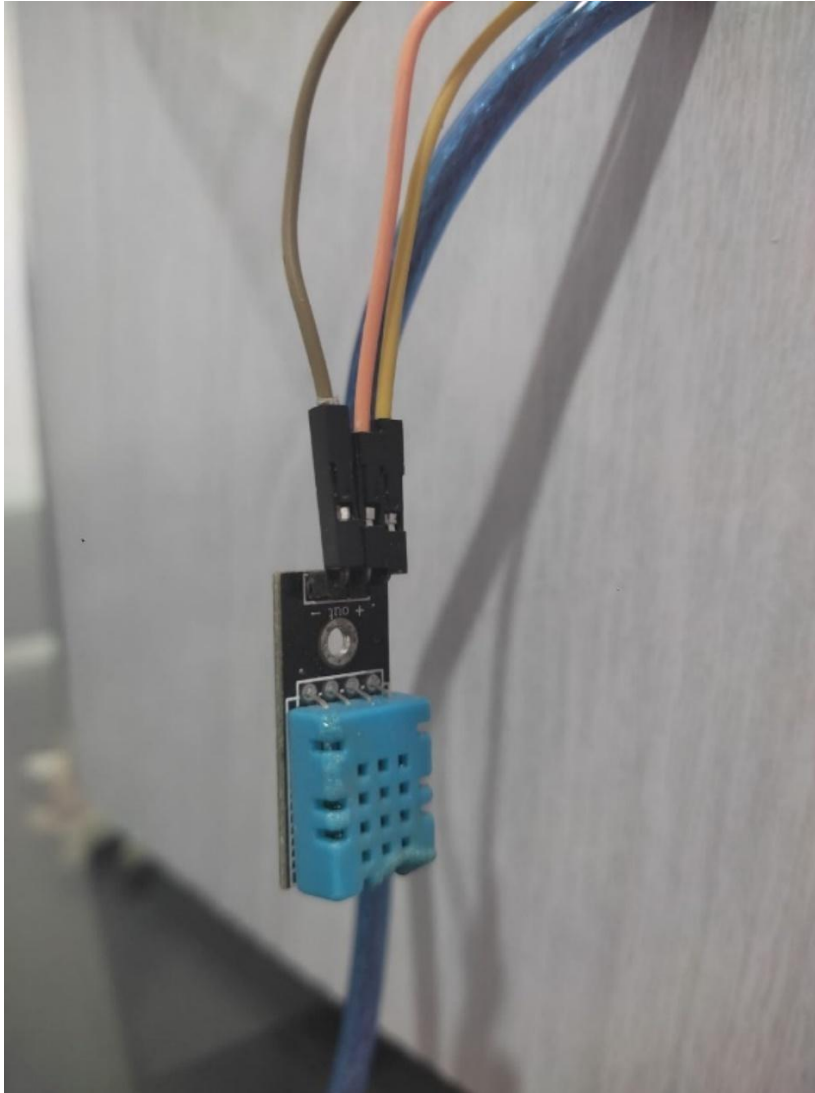
*Nota.* Configuración Wifi

## Implementación del Sensor DHT11 en el Sistema de Monitoreo

El sensor DHT11 se instala en la parte exterior del cuarto frío en escala con el fin de poder tener acceso para la simulación de los cambios de temperatura.

### Figura 72

*Instalación Sensor Temperatura Cuarto Frio*



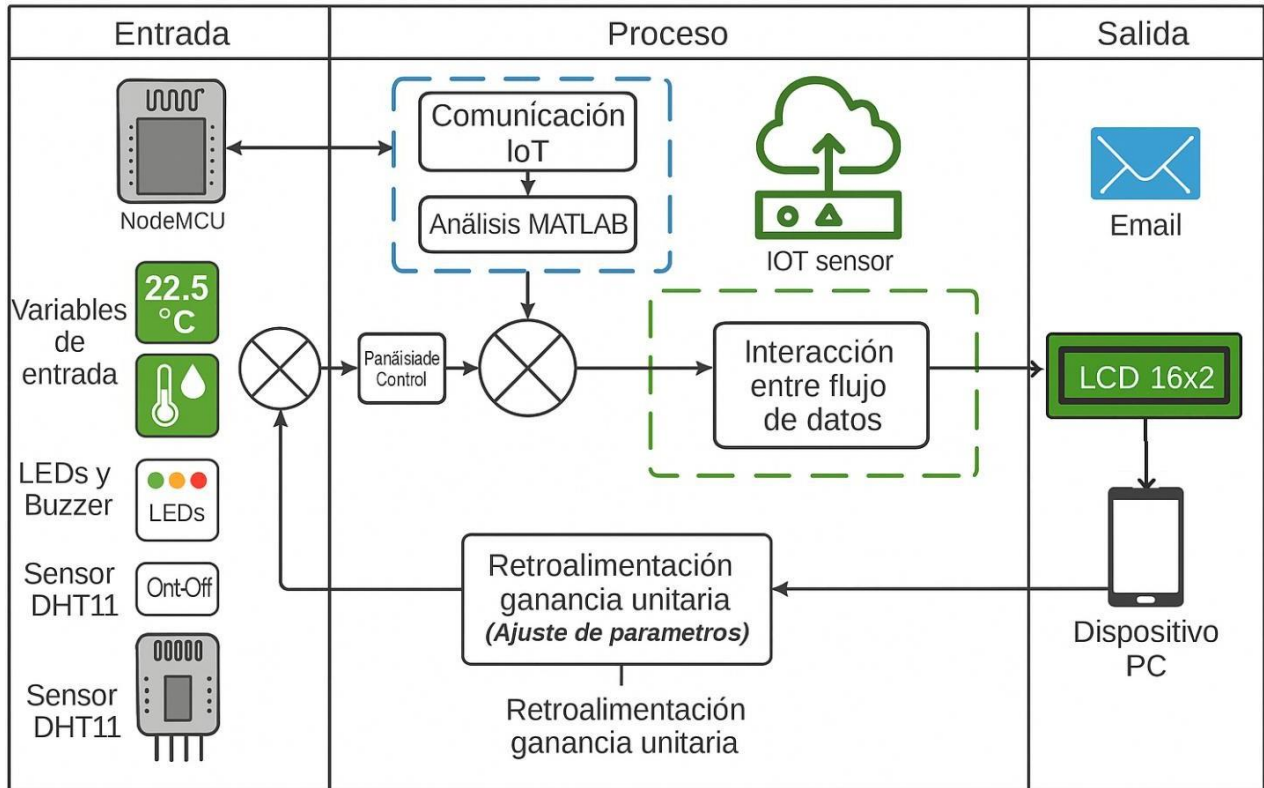
*Nota.* Instalación sensor temperatura cuarto frio DHT11

## Funcionamiento del Sistema

### Arquitectura del Software

Figura 73

### Arquitectura del Software



Nota. Arquitectura del Software

### Visualización en Pantalla LCD

La pantalla LCD permite visualizar en tiempo real los datos de temperatura del cuarto frío, lo que facilita el monitoreo local, como se muestra en el siguiente como se muestra en las siguientes imágenes.

**Figura 74**

*Monitoreo Pantalla LCD Temper en 24.5 Grados Celsius*



*Nota. Monitoreo pantalla LCD temper en 24.5 grados Celsius*

**Figura 75**

*Monitoreo Pantalla LCD temper en 28.9 Grados Celsius*



*Nota. Monitoreo pantalla LCD temper en 28.9 grados Celsius*

## Figura 76

### *Monitoreo Local Pantalla LCD Temperatura En 17.5 Grados Celsius*



*Nota.* Monitoreo local pantalla LCD temperatura en 17.5 grados Celsius

## Monitoreo

El sistema de monitoreo se fundamenta en tecnologías IoT, utilizando módulos WiFi para la transmisión de datos hacia la plataforma en la nube ThingSpeak. Desde allí, los datos pueden ser consultados posteriormente a través de una aplicación móvil desarrollada para tal fin.

## Sistema Implementado

El sistema desarrollado permite monitorear en tiempo real la temperatura dentro de un cuarto frío utilizando un sensor DHT11, un microcontrolador NodeMCU ESP8266 y una pantalla LCD para visualizar los datos localmente. Además, gracias a su conexión a internet, los datos se envían a la plataforma ThingSpeak, donde pueden ser consultados desde cualquier lugar. El sistema también cuenta con alertas visuales y sonoras, así como notificaciones por correo electrónico, lo que garantiza una respuesta rápida ante cualquier desviación térmica.

**Figura 77**

*Cuarto frio en Escala*



*Nota.* Cuarto frio en escala

**Figura 78**

*Parte Interna Del Cuarto Frio*



*Nota.* Parte interna del cuarto frio

**Figura 79**

*Indicador Luminoso entre Parámetros*



*Nota .* Indicador luminoso entre parámetros

**Figura 80**

*Indicador Luminoso Temperatura Alta Fuera De Parámetros*



*Nota.* Indicador luminoso temperatura alta fuera de parámetros

**Figura 81**

*Indicador Luminoso Temperatura Baja Fuera De Parámetros*



*Nota.* Indicador luminoso temperatura baja fuera de parámetros

**Enlace del Video**

<https://youtu.be/L-RdYvtGW8>

## **Análisis e Interpretación de Resultados**

El sistema desarrollado para el monitoreo de temperatura en cuartos fríos mostró un funcionamiento estable, preciso y acorde con los objetivos propuestos desde la fase inicial del proyecto. Durante las pruebas realizadas en el modelo a escala, se simularon distintas condiciones térmicas, tanto por encima como por debajo de los valores definidos como óptimos. En todos los casos, el sistema reaccionó adecuadamente: encendiendo los indicadores visuales (LED rojo, LED amarillo, o en caso de valores óptimos LED Verde, en los casos de los LED amarillo o rojo se activó el Buzzer como alerta sonora, y envió notificaciones automáticas a través de correo electrónico a los usuarios registrados.

Lo más gratificante fue ver cómo, al simular condiciones reales, el sistema reaccionaba de forma inmediata y precisa. Gracias a la plataforma ThingSpeak, se pudo visualizar en tiempo real el comportamiento de la temperatura, y también acceder al historial de los datos desde cualquier lugar, lo que brinda un gran valor en términos de control, trazabilidad y toma de decisiones.

Esta herramienta no solo aporta comodidad, sino que también ofrece confianza al saber que la información está siempre disponible al alcance de la mano.

Un aspecto que vale la pena resaltar es cómo el sistema ayuda a reducir la dependencia del monitoreo manual. Al contar con alertas automáticas, los técnicos o encargados pueden actuar de manera oportuna, sin necesidad de estar verificando constantemente el estado de las cavas. Esto mejora la eficiencia operativa y reduce el riesgo de perder productos por fallas en la temperatura.

Durante todo el proceso, desde la conexión de los sensores hasta el envío de datos por WiFi y la configuración de las alertas, se pudo evidenciar que cada componente cumplió su función correctamente. La comunicación entre el sensor, el microcontrolador, la pantalla LCD y la nube fue estable, lo que dio como resultado un sistema confiable y sencillo de utilizar.

En resumen, el resultado final fue muy positivo. Este proyecto demostró que, con tecnologías accesibles como Arduino, IoT y plataformas en la nube, es posible desarrollar soluciones reales que

aporten al sector industrial. Más allá del logro técnico, queda la satisfacción de haber construido una herramienta útil, funcional y con gran potencial para ser aplicada en escenarios reales.

## Impactos

Desarrollar este proyecto no solo significó diseñar un sistema electrónico funcional, sino también responder a una necesidad real dentro del entorno industrial: proteger la calidad de los productos que consumimos día a día. Detrás de cada sensor, cada línea de código y cada prueba realizada, hay un propósito mucho más grande: contribuir al bienestar de las personas, optimizar procesos que impactan directamente en la economía y demostrar cómo la tecnología puede convertirse en una herramienta poderosa para resolver problemas cotidianos. Este trabajo no solo deja un sistema implementado, sino una huella en distintos niveles que vale la pena mencionar:

La implementación de este sistema demuestra que es posible aplicar tecnología IoT de bajo costo para resolver problemas complejos en la industria. El uso de sensores DHT11, NodeMCU ESP8266, Arduino IDE y la plataforma ThingSpeak permitió desarrollar una solución integral, capaz de monitorear y alertar en tiempo real. Esta experiencia técnica sienta las bases para futuras mejoras y aplicaciones similares en otras áreas como control de humedad, presión, calidad del aire o incluso procesos de refrigeración más complejos.

Uno de los beneficios más directos del proyecto es la prevención de pérdidas económicas ocasionadas por fallas en la cadena de frío. Reducir el desperdicio de alimentos o insumos debido a fallas de temperatura tiene un efecto directo en la sostenibilidad. Menos productos dañados significa menos residuos, menor consumo de recursos para reprocesar o reemplazar mercancía y una disminución en la huella ambiental de la industria. Este sistema contribuye a una producción más responsable, alineada con los objetivos globales de sostenibilidad y eficiencia energética. Esto se traduce en una mejor gestión de inventarios, menores costos operativos y una mayor rentabilidad para las empresas que dependen de procesos de refrigeración.

Mantener la temperatura adecuada en los productos almacenados no solo mejora la eficiencia industrial, sino que también protege la salud de los consumidores. Este sistema aporta a la seguridad alimentaria, al asegurar que los productos lleguen al usuario final en condiciones óptimas.

## **Verificación de Objetivos**

Se logró cumplir el objetivo principal del proyecto: implementar un sistema que permite monitorear en tiempo real la temperatura de las cavas de conservación.

Gracias al uso de sensores DHT11, el módulo NodeMCU ESP8266 y la plataforma ThingSpeak, fue posible crear una solución práctica y funcional. El sistema no solo registra la información, sino que también alerta de manera automática cuando hay cambios que podrían afectar la cadena de frío, ayudando así a tomar decisiones rápidas y acertadas.

### **Infraestructura de Sensores**

Para dar inicio al sistema, se diseñó y ensambló una infraestructura con todos los elementos necesarios para medir la temperatura con precisión. Se eligió el sensor DHT11 por su confiabilidad y facilidad de uso, y se integró con otros componentes como una pantalla LCD, Leds y un Buzzer. Todo esto fue montado primero en simulación con Proteus y luego llevado al prototipo físico, lo que permitió tener una base sólida para el funcionamiento del sistema.

### **Sistema de alertas**

Uno de los puntos clave del proyecto fue desarrollar un sistema de alertas que avisara cuando la temperatura saliera de los valores establecidos. Esto se logró integrando indicadores visuales (Leds de colores) y sonoros (Buzzer) para alertas locales. Además, se configuraron notificaciones automáticas por correo electrónico gracias al uso de herramientas como MATLAB, TimeControl y React desde ThingSpeak. De esta manera, se asegura que tanto el personal en sitio como el encargado remoto estén siempre informados.

### **Monitoreo Centralizado**

También se cumplió con éxito el objetivo de tener una plataforma donde se pudiera consultar la información en todo momento.

ThingSpeak fue la herramienta elegida para centralizar los datos, permitiendo ver gráficas, analizar la información en tiempo real y acceder desde cualquier lugar con conexión a internet. Esto

brinda una ventaja enorme para la supervisión remota y la trazabilidad del comportamiento térmico.

### **Validación del Sistema**

Se llevaron a cabo pruebas en un entorno simulado a escala, que permitió verificar que todos los componentes del sistema funcionaran correctamente. En estas pruebas se ajustaron los parámetros de temperatura para observar cómo reaccionaba el sistema ante diferentes escenarios. Aunque no se implementó directamente en una planta industrial real, esta etapa demostró que el sistema está preparado para ser escalado y validado en campo más adelante. Esto queda como una próxima fase del proyecto.

### **Aportes Realizados en El Proyecto**

El sistema de monitoreo de temperatura en cuartos fríos desarrollado en este proyecto permite a los técnicos de mantenimiento supervisar en tiempo real las condiciones térmicas de conservación de productos sensibles, tanto de forma local como se puede apreciar en la pantalla LCD, o de forma remota como se visualiza en la plataforma ThingSpeak. Esta visualización en tiempo real brinda la tranquilidad de saber que los productos almacenados están dentro de los rangos seguros, minimizando el riesgo de pérdida por descomposición o daño térmico.

Gracias a su sistema de alertas visuales, sonoras y por correo electrónico, el proyecto contribuye a que el personal técnico pueda actuar de forma oportuna ante cualquier desviación en la temperatura, sin necesidad de una vigilancia constante presencial. Esto representa un ahorro significativo en tiempo, mejora la eficiencia del mantenimiento y reduce la dependencia de la supervisión manual.

Además, el sistema lleva un registro histórico de los datos, lo que permite realizar análisis posteriores y tomar decisiones informadas para optimizar los procesos de refrigeración. De esta manera, el proyecto aporta al cumplimiento de estándares de calidad y seguridad, facilitando el trabajo del equipo técnico y asegurando la conservación adecuada de los productos

## Conclusiones

La implementación del sistema de monitoreo de temperatura basado en tecnología IoT demostró ser una solución eficaz para preservar la calidad de los productos sensibles a las variaciones térmicas. Esto representa un avance importante en la garantía de seguridad alimentaria y reducción de desperdicio, especialmente en la industria de procesamiento de alimentos.

De igual manera el sistema desarrollado permite detectar desviaciones de temperatura de forma inmediata y emitir alertas visuales, sonoras y por correo electrónico. Esta capacidad de reacción rápida brinda a los técnicos de mantenimiento una herramienta valiosa para actuar de manera preventiva, minimizando pérdidas económicas y riesgos operativos.

Se logró construir un prototipo funcional a escala utilizando componentes accesibles como el sensor DHT11, el módulo NodeMCU y la plataforma ThingSpeak. Esto demuestra que es posible implementar soluciones tecnológicas de bajo costo, pero de alto impacto, en entornos industriales reales.

La correcta interacción entre los distintos elementos del sistema como sensores, microcontrolador, red Wi-Fi, pantalla LCD y plataforma en la nube evidenció una integración sólida y coherente, que facilita tanto la supervisión local como remota del entorno de conservación.

El proyecto sienta las bases para futuras mejoras, como la inclusión de sensores más precisos, monitoreo multizona, integración con aplicaciones móviles, o la aplicación de algoritmos de predicción mediante inteligencia artificial para anticipar fallas.

Esta experiencia fortaleció competencias en diseño electrónico, automatización, programación embebida y comunicación de datos, representando no solo un logro académico, sino también una contribución directa a la transformación digital de los procesos industriales.

## **Propuestas de Mejora del Sistema de Monitoreo Temperatura**

A lo largo del desarrollo del presente proyecto se lograron importantes avances en la implementación de un sistema de monitoreo de temperatura confiable y funcional. Sin embargo, como en todo proceso de innovación tecnológica, siempre existen oportunidades para seguir mejorando.

Las experiencias adquiridas durante las pruebas, así como la reflexión sobre los desafíos enfrentados, permitieron identificar varias áreas en las que el sistema puede evolucionar para ofrecer un mayor nivel de precisión, usabilidad, escalabilidad y adaptabilidad a entornos reales e industriales.

A continuación, se presentan las principales propuestas de mejora, tanto desde el punto de vista técnico como desde el impacto práctico y humano del sistema en su entorno de aplicación, que ayuda a una mejor interpretación y validación del sistema frente a estándares de calidad o normativas del sector.

### **Profundización en el Análisis De Resultados**

Incorporar gráficas más detalladas del comportamiento térmico mediante herramientas como MATLAB, Excel o Grafana.

Añadir métricas estadísticas: media, desviación estándar, valores extremos y porcentaje de tiempo fuera del rango.

Comparar los datos a lo largo del día para identificar fluctuaciones por horas, días o eventos externos.

### **Escalabilidad del sistema**

Adaptar la solución para el monitoreo de múltiples cavas en paralelo desde una misma plataforma.

Implementar una base de datos SQL (local o en la nube) para almacenar y consultar los datos de cada cava de forma independiente.

## **Mejora de la Precisión y Distribución de Sensores**

Al mejorar el sensor permite la reducción del riesgo de zonas frías o calientes no detectadas y mejor conservación del producto.

Sustituir el sensor DHT11 por modelos más precisos como el DHT22, SHT31 o PT100 con transmisor analógico, que permiten mayor confiabilidad.

Instalar varios sensores distribuidos estratégicamente dentro de la cava para detectar diferencias térmicas internas y garantizar una conservación homogénea.

## **Diversificación de Canales de Alerta**

La capacidad de respuesta es muy importante ante cualquier emergencia, incluso si el correo electrónico no está disponible para esto se debe ampliar los métodos de notificación como incluir mensajes SMS o incluso llamadas automáticas en casos críticos.

Usar plataformas como Twilio o servicios API para integración con celulares del equipo técnico o supervisor.

## **Interfaz para Usuarios no Técnicos**

Es importante que el sistema cuente con una interfaz de usuarios para tener mejor accesibilidad del sistema a operarios y personal sin formación técnica en IoT o programación, por lo anterior se mencionan algunas mejoras:

Desarrollar una interfaz HTML5 simple y visual con colores de alerta, gráficos e indicadores.

Alternativamente, crear una app móvil básica (con MIT App Inventor o Kodular) que permita al usuario:

Ver temperatura actual.

Consultar historial.

Recibir alertas.

## **Seguridad de La Información Transmitida**

La protección de la integridad y confidencialidad de los datos críticos de la cadena de frío es muy importante, por esto, se Implementan medidas de seguridad como:

Transmisión cifrada por HTTPS o TLS.

Uso de API Key y autenticación para evitar accesos no autorizados.

Rutinas automáticas de reconexión del sistema ante caídas de red.

## **Sistemas de Respaldo Energético Y De Datos**

Para garantizar la continuidad operativa incluso en eventos adversos como apagones o fallas de red es recomendable integrar un sistema de alimentación ininterrumpida (UPS) que permita continuar el monitoreo en caso de cortes eléctricos.

Habilitar una bitácora local en memoria interna o tarjeta SD para registrar datos durante caídas de red y sincronizarlos posteriormente.

Estas mejoras permitirán que el sistema evolucione de un prototipo funcional a una solución robusta, escalable y confiable, adaptada a los estándares de la industria y con un impacto positivo en la conservación de productos sensibles a la temperatura.

**Tabla 2***Oferta comercial Prototipo Vs Cava 5 toneladas*

Ítem / Descripción	Versión Prototipo a Escala	Versión Real – Cava 5 Toneladas
1. Sensores de temperatura	DHT11 (bajo costo, uso educativo) \$ 10.000	PT100 + transmisor 4–20 mA (precisión industrial) \$ 274.990 (PT100+transmisor)
2. Controlador de temperatura	No aplica	MT-512Ri Plus (controla compresor y deshielo automático) \$ 160.000
3. Microcontrolador IoT	NodeMCU ESP8266 \$ 30.035	ESP32 (mayor procesamiento y estabilidad) \$ 30.035
4. Alerta visual y sonora local	LED rojo/amarillo/verde + buzzer \$ 30.000 (LED buzzer)	Torre de señal + sirena industrial 24 VDC \$ 204.550
5. Pantalla local	LCD 16x2 I2C \$ 40.000	Se mantiene o se sustituye por dashboard móvil/web
6. Comunicación a la nube (Wi-Fi)	ThingSpeak básico \$ gratis	ThingSpeak + MQTT + servidor o integración opcional \$ 100.000 (licencia)
7. Alerta por correo electrónico	Incluida (ThingSpeak + MATLAB + TimeControl + React)	Incluida (configurable con múltiples destinatarios)
8. Alerta por SMS o WhatsApp	No incluida	Opcional con Twilio (COP \$150.000/año)
9. Visualización web o móvil	Gráficas en ThingSpeak \$gratis	Dashboard HTML5 o Grafana personalizable \$ 400.000
10. Almacenamiento de datos históricos	Gratis por 3 meses (ThingSpeak free)	Gratis + extensible con backup y exportación CSV
11. Cableado	Jumpers y protoboard	Cables industriales: blindados, THHN, canaletas IP65
12. Protección eléctrica y caja	Protoboard abierta sin caja	Caja estanca IP65 + riel DIN + borneras + UPS \$Incluido
13. Respaldo eléctrico (UPS)	No incluido	UPS interactiva 650 VA (30 min autonomía) \$409.900 (APC BV650)
14. Instalación en sitio	No aplica o 1 día en maqueta	2–3 días con instalación, prueba, validación y montaje
15. Desarrollo software (IoT + alertas)	Básico funcional	Completo: conexión nube, alertas, ajustes personalizados \$ 2.000.000
16. Capacitación técnica y manuales	2 h básicas con manual PDF \$ 200.000	3 h + manual técnico, fichas impresas y digitales \$ 400.000
17. Garantía de hardware/software/instalación	3 a 6 meses	6 meses (opcional extensión a 12 meses)

Ítem / Descripción	Versión Prototipo a Escala	Versión Real – Cava 5 Toneladas
18. Soporte técnico	Soporte remoto básico (WhatsApp/Email)	Soporte remoto + opcional presencial postventa
19. Almacenamiento adicional de datos (backup/exportación)	Manual desde ThingSpeak	Incluido + automatización mensual + integración Drive
Costo total estimado (COP)	COP \$1.860.000	COP \$6.130.500

*Nota.* Oferta comercial Prototipo Vs Cava 5 toneladas.

### **Tabla 3**

#### *Cotización Cableado-cava 5 Toneladas*

Cable / Conector	Longitud	Precio U.	Subtotal
Cable blindado 2×0.75 mm <sup>2</sup> (sensor)	10 m	COP 5.000/m	COP 50.000
Cable 4-20 mA / RS485 2×0.5 mm <sup>2</sup>	15 m	COP 3.500/m	COP 52.500
Cable alimentación 220 V 3×1.5 mm <sup>2</sup> THHN	12 m	COP 2.250/m	COP 27.000
Terminales y conectores (faston, espadín, blindaje)	–	–	COP 25.000

Canaletas / tubos conduit ½" + fijación	-	-	COP 40.000
---	---	---	------------

---

Subtotal cableado	-	-	COP 194.500
-------------------	---	---	-------------

---

*Nota.* Cotización Cableado-cava 5 toneladas

## Conclusiones

La implementación del sistema de monitoreo de temperatura basado en tecnología IoT demostró ser una solución eficaz para preservar la calidad de los productos sensibles a las variaciones térmicas. Esto representa un avance importante en la garantía de seguridad alimentaria y reducción de desperdicio, especialmente en la industria de procesamiento de alimentos.

De igual manera el sistema desarrollado permite detectar desviaciones de temperatura de forma inmediata y emitir alertas visuales, sonoras y por correo electrónico. Esta capacidad de reacción rápida brinda a los técnicos de mantenimiento una herramienta valiosa para actuar de manera preventiva, minimizando pérdidas económicas y riesgos operativos.

Se logró construir un prototipo funcional a escala utilizando componentes accesibles como el sensor DHT11, el módulo NodeMCU y la plataforma ThingSpeak. Esto demuestra que es posible implementar soluciones tecnológicas de bajo costo, pero de alto impacto, en entornos industriales reales.

La correcta interacción entre los distintos elementos del sistema como sensores, microcontrolador, red Wi-Fi, pantalla LCD y plataforma en la nube evidenció una integración sólida y coherente, que facilita tanto la supervisión local como remota del entorno de conservación.

El proyecto sienta las bases para futuras mejoras, como la inclusión de sensores más precisos, monitoreo multizona, integración con aplicaciones móviles, o la aplicación de algoritmos de predicción mediante inteligencia artificial para anticipar fallas.

Esta experiencia fortaleció competencias en diseño electrónico, automatización, programación embebida y comunicación de datos, representando no solo un logro académico, sino también una contribución directa a la transformación digital de los procesos industriales.

## Referencias Bibliográficas

- Arcila, O. L. (Enero de 2023). *researchgate*. Obtenido de Sistema de monitoreo de temperaturas y presiones para el área de generación de frío de la planta procesadora de Colanta en San Pedro de los Milagros: [https://www.researchgate.net/publication/30757745\\_](https://www.researchgate.net/publication/30757745_)
- Automatismo, G. (01 de 01 de 2022). *automatismosguillen*. Obtenido de Camaras de congelacion y mantenimiento: <https://www.automatismosguillen.com/es/producto/camaras-de-congelacion-y-mantenimiento/>
- automatismosguillen. (18 de 01 de 2024). *automatismosguillen*. Obtenido de Cámaras frigoríficas: <https://www.automatismosguillen.com/es/producto/camaras-de-congelacion-y-mantenimiento/>
- Autoria, P. (02 de marzo de 2025). Javier Hernandez Arango. La Ceja, Antioquia, Colombia.
- Consoempresarial. (29 de Enero de 2020). *Consoempresarial*. Obtenido de Cuarto Frío: <https://www.confoempresarial.com/simulador-de-cuarto-frio/>
- Cortes, D. (15 de 01 de 2025). *Cesuma*. Obtenido de Técnicas de conservación de alimentos: <https://www.cesuma.mx/blog/tecnicas-de-conservacion-de-alimentos.html>
- Danahé San Juan. (09 de 10 de 2021). *mundohvacr*. Obtenido de Control y monitoreo o cómo asegurar la cadena de frío: <https://www.mundohvacr.com/2021/10/control-y-monitoreo-o-como-asegurar-la-cadena-de-frio/>
- deldivel. (15 de 01 de 2024). *deldivel*. Obtenido de abatidor de temperatura: <https://www.deldivel.es/abatidores-de-temperatura>
- descubrearduino. (15 de junio de 2024). */descubrearduino*. Obtenido de Thingspeak: <https://descubrearduino.com/>
- evaluandoerp. (02 de 05 de 2022). *evaluandoerp*. Obtenido de Cuáles son los distintos sistemas de

producción industrial: <https://www.evaluandoerp.com/sistema-de-gestion/conceptos-basicos/cuales-los-distintos-sistemas-produccion-industrial/>

FAO. (20 de 01 de 2021). *fao*. Obtenido de Economía Agroalimentaria: <https://www.fao.org/agrifood-economics/publications/detail/es/c/1457039/>

FAO. (01 de 05 de 2023). *El estado de la seguridad alimentaria y la nutrición en el mundo*.

Obtenido de fao: [www.fao.org](http://www.fao.org)

FAO. (04 de 09 de 2023). *fao*. Obtenido de Organización de las Naciones Unidas para la Alimentación y la Agricultura: <https://www.fao.org/>

Findus. (18 de 01 de 2024). *Findus*. Obtenido de Qué es la cadena de frío de los alimentos y cómo no interrumpirla: <https://www.findus.es/alimentacion-y-nutricion/congelados/cadena-de-frio-que-es-y-como-no-interrumpirla>

hannacolombia. (01 de 04 de 2024). *hannacolombia*. Obtenido de La Cadena de Frío, clave en la Seguridad Alimentaria: [https://www.hannacolombia.com/blog/post/18/la-cadena-frio-clave-en-la-seguridad-alimentaria?srsltid=AfmBOooICPzgSUOydxJOWXKjBCn9rj0anC27VOYU\\_k0U0fX0bYW0go7o](https://www.hannacolombia.com/blog/post/18/la-cadena-frio-clave-en-la-seguridad-alimentaria?srsltid=AfmBOooICPzgSUOydxJOWXKjBCn9rj0anC27VOYU_k0U0fX0bYW0go7o)

hetpro. (15 de 01 de 2022). *hetpro*. Obtenido de Modulo I2C: [https://hetpro-store.com/TUTORIALES/i2c/?srsltid=AfmBOooBO7RIFdR5fiLJzTPif5aoExdb3xGccelBTBUj2e5DP0pxpt\\_U](https://hetpro-store.com/TUTORIALES/i2c/?srsltid=AfmBOooBO7RIFdR5fiLJzTPif5aoExdb3xGccelBTBUj2e5DP0pxpt_U)

hubor. (03 de 02 de 2021). *hubor*. Obtenido de Proteus: <https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/2-proteus.html>

IBM. (10 de 11 de 2022). *ibm*. Obtenido de Qué es la automatización: <https://www.ibm.com/mx-es/topics/automation>

Industrias, G. (26 de Agosto de 2021). *Induatrias GSL*. Obtenido de Qué es un sensor de temperatura industrial: <https://industriasgsl.com/blogs/automatizacion/sensor-de-temperatura-industrial#:~:text=%C2%BFQu%C3%A9%20es%20un%20sensor%20de,y%20registro%20d>

e%20una%20variable.

Isaac. (2025). *hwlibre*. Obtenido de DHT11 todo sobre el sensor para medir temperatura y humedad:

<https://www.hwlibre.com/dht11/>

J.L, B. (16 de 04 de 2025). *electronicaonline*. Obtenido de Diodo Emisor de Luz:

<https://electronicaonline.net/componentes-electronicos/diodo/que-es-un-led/>

León, R. A., Gutiérrez H, S. K., Jacinto Paredes, J. d., & Miranda Mego, K. M. (12 de 2021). *researchgate*.

Obtenido de Diseño e implementación de un sistema de control de temperatura:

[https://www.researchgate.net/publication/369027455\\_Disenio\\_e\\_implementacion\\_de\\_un\\_sistema\\_de\\_control\\_de\\_temperatura\\_para\\_un\\_galpon\\_de\\_pollos\\_de\\_la\\_Avicola\\_Florian\\_e\\_Hijos\\_SAC](https://www.researchgate.net/publication/369027455_Disenio_e_implementacion_de_un_sistema_de_control_de_temperatura_para_un_galpon_de_pollos_de_la_Avicola_Florian_e_Hijos_SAC)

[ema\\_de\\_control\\_de\\_temperatura\\_para\\_un\\_galpon\\_de\\_pollos\\_de\\_la\\_Avicola\\_Florian\\_e\\_Hijos\\_SAC](https://www.researchgate.net/publication/369027455_Disenio_e_implementacion_de_un_sistema_de_control_de_temperatura_para_un_galpon_de_pollos_de_la_Avicola_Florian_e_Hijos_SAC)

Llamas, L. (2022). *Luis Llamas*. Obtenido de Nodemcu ESP8266: <https://www.luisllamas.es/esp8266-nodemcu/>

Mecafenix, I. (2025). *Ingenieria Mecafenix*. Obtenido de que es el Buzzer:

<https://www.ingmecafenix.com/electronica/componentes/el-buzzer/>

Mecanifex, I. (2022). *Ingeniería Mecanifex*. Obtenido de Pantalla LCD:

<https://www.ingmecafenix.com/electronica/componentes/lcd/>

mielectronicafacil. (2021). *mielectronicafacil*. Obtenido de Estructura de la placa protoboard:

<https://mielectronicafacil.com/instrumentacion/protoboard/#estructura-de-la-placa-protoboard>

Muñoz, O. Q. (2019). *perlego*. Obtenido de Internet de las cosas:

<https://www.perlego.com/es/book/2913651/internet-de-las-cosas-iot-pdf>

ObschatkoMedina, R. d. (2020). *revistaspa*. Obtenido de Seguridad alimentaria en tiempos de cambio global

. Revista de Políticas Agroalimentarias: [www.revistaspa.org](http://www.revistaspa.org)

OMS. (2022). *who*. Obtenido de Enfermedades transmitidas por los alimentos: [www.who.int](http://www.who.int)

osakaelectronics. (15 de 04 de 2022). *osakaelectronics*. Obtenido de Modulo I2C:

[https://osakaelectronicsltda.com/modulos-robotica/modulos-varios/modulo-i2c-lcd-](https://osakaelectronicsltda.com/modulos-robotica/modulos-varios/modulo-i2c-lcd-arduino.html?srsltid=AfmBOorWwYhsnM3s8tf51nwxnm2wpAZ-AcSdKmODK-GqvCdqqC4Z7LvH)

[arduino.html?srsltid=AfmBOorWwYhsnM3s8tf51nwxnm2wpAZ-AcSdKmODK-](https://osakaelectronicsltda.com/modulos-robotica/modulos-varios/modulo-i2c-lcd-arduino.html?srsltid=AfmBOorWwYhsnM3s8tf51nwxnm2wpAZ-AcSdKmODK-GqvCdqqC4Z7LvH)

[GqvCdqqC4Z7LvH](https://osakaelectronicsltda.com/modulos-robotica/modulos-varios/modulo-i2c-lcd-arduino.html?srsltid=AfmBOorWwYhsnM3s8tf51nwxnm2wpAZ-AcSdKmODK-GqvCdqqC4Z7LvH)

Pedro Nicoyani Mamani O, J. M. (20 de Diciembre de 2020). *researchgate*. Obtenido de Internet de

las cosas almacen y transporte de alim:

[https://www.researchgate.net/publication/347967771\\_Diseño\\_e\\_implementación\\_de\\_termohigrometro\\_basado\\_en\\_tecnología\\_de\\_Internet\\_de\\_las\\_cosas\\_para\\_el\\_almacenamiento\\_y\\_transporte\\_de\\_alimentos](https://www.researchgate.net/publication/347967771_Diseño_e_implementación_de_termohigrometro_basado_en_tecnología_de_Internet_de_las_cosas_para_el_almacenamiento_y_transporte_de_alimentos)

Pinzón, J. E., & Rodríguez González, J. (Julio de 2024). *researchgate*. Obtenido de Monitoreo de temperatura basado en arduino:

[https://www.researchgate.net/publication/382509945\\_Monitoreo\\_de\\_temperatura\\_en\\_el\\_aula\\_de\\_clase\\_basada\\_en\\_Arduino](https://www.researchgate.net/publication/382509945_Monitoreo_de_temperatura_en_el_aula_de_clase_basada_en_Arduino)

RefriClimas. (28 de abril de 2024). *refri-climas*. Obtenido de cámara frigorífica: <https://www.refri-climas.com/camara-frigorifica/>

refrigeracionaldama. (2025). *refrigeracionaldama*. Obtenido de Refrigeración y aires acondicionados Almama: <https://www.refrigeracionaldama.com/que-es-un-cuarto-frio#:~:text=En%20resumen%2C%20un%20cuarto%20fi%C3%ADo%20es%20un%20espacio,perecederos%20en%20condiciones%20%C3%B3ptimas%20de%20temperatura%20y%20humedad.>

RelemaT. (12 de Noviembre de 2022). *Por qué se descomponen los alimentos*. Obtenido de relemaT: <https://www.relemaT.es/blog/sabias-que/por-que-se-descomponen-los-alimentos>

Reynes, D. M. (2024). *Ferroice*. Obtenido de los principios de la conservación de alimentos: <https://www.ferroice.com/blog/2017/06/13/los-principios-de-la-conservación-de-alimentos/>

Solaz, A. (04 de 02 de 2025). *fracttal*. Obtenido de Principales amenazas para la cadena de frío y sus consecuencias: <https://www.fracttal.com/es/blog/como-proteger-la-cadena-del-frío-con-iot>

techlandia. (2022). *techlandia*. Obtenido de cable de datos: [https://techlandia.com/definición-cables-datos-hechos\\_88761/](https://techlandia.com/definición-cables-datos-hechos_88761/)

Tecnológicos, M. (2023). *muytecnológicos*. Obtenido de señal digital: <https://www.muytecnológicos.com/diccionario-tecnológico/senal-digital>

TGM. (2021). *TGM*. Obtenido de equipos de transporte banda flexible:

<https://www.tmgburgos.es/productos/equipos-de-transporte/cintas-transportadoras-de-banda-flexible/>

thingspeak. (2021). *thingspeak*. Obtenido de que es thingspeak: <https://thingspeak.mathworks.com/> tolocka, p. (2023). *profetolocka*. Obtenido de publicando datos en thingspeak:

[https://www.profetolocka.com.ar/2021/01/06/micropython-publicando-datos-en-thingspeak- 1/](https://www.profetolocka.com.ar/2021/01/06/micropython-publicando-datos-en-thingspeak-1/)

tuelectronica. (2021). *tuelectronica*. Obtenido de que es arduino: <https://tuelectronica.es/que-es- arduino-ide/>

zonagreen. (2023). *zonagreen*. Obtenido de que es un jumper: <https://zonagreen.com.mx/que-es-un- jumper/>

## Apéndice A

### *Código Principal Arduino*

```
#include <LiquidCrystal_I2C.h>
#include <DHT.h>
#include <ESP8266WiFi.h>

// Configuración WiFi
const char* ssid = "DEICY"; // Reemplaza con tu SSID
const char* password = "39193330*V"; // Reemplaza con tu contraseña

// Configuración de ThingSpeak
const char* server = "api.thingspeak.com";
String apiKey = "PD6LBOT194RSSIX5";

// Configuración de LCD I2C
LiquidCrystal_I2C lcd(0x27, 16, 2); // Dirección 0x27 para LCD I2C
DHT dht(2, DHT11); // DHT11 conectado al pin GPIO2 (D4)

#define LED1 13 // D7= GPIO13 - color Verde
#define LED2 12 // D6= GPIO12 - color Rojo
#define LED3 0 // D3= GPI15 - color Amarillo
#define SPEAKER 15 // D8= GPIO14 - Buzzer
// Variables
float temp, hume;
unsigned long previousMillis = 0;
const unsigned long interval = 5000; // Intervalo de actualización (5 segundos)
int contconexion = 0;

// Función para conectar a WiFi
void conectarWiFi() {
  WiFi.begin(ssid, password);
  Serial.print("Conectando a WiFi");

  while (WiFi.status() != WL_CONNECTED && contconexion < 50) {
    delay(500);
    Serial.print(".");
    contconexion++;
  }

  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("\nWiFi conectado!");
    Serial.print("Dirección IP: ");
    Serial.println(WiFi.localIP());
  } else {
    Serial.println("\nError de conexión WiFi");
  }
}

// Función para enviar datos a ThingSpeak
```

```

void enviarDatosTS(float temp, float hum) {
  if (WiFi.status() == WL_CONNECTED) {
    WiFiClient client;
    if (client.connect(server, 80)) {
      Serial.println("Conectado a ThingSpeak");

      String postStr = "api_key=" + apiKey;
      postStr += "&field1=" + String(temp);
      postStr += "&field2=" + String(hum);

      client.print("POST /update HTTP/1.1\n");
      client.print("Host: api.thingspeak.com\n");
      client.print("Connection: close\n");
      client.print("Content-Type: application/x-www-form-urlencoded\n");
      client.print("Content-Length: " + String(postStr.length()) + "\n\n");
      client.print(postStr);

      Serial.println("Datos enviados a ThingSpeak!");
    } else {
      Serial.println("Error al conectar con ThingSpeak");
    }
    client.stop();
  } else {
    Serial.println("No hay conexión WiFi, intentando reconectar...");
    conectarWiFi();
  }
}

```

// Función para controlar LED y buzzer según temperatura

```

void controlarAlertas(float temp) {
  if (temp >= 20 && temp <= 27) {
    digitalWrite(LED1, HIGH); // Enciende LED verde
    digitalWrite(LED2, LOW); // Apaga LED rojo
    digitalWrite(LED3, LOW); // Apaga LED amarillo
    digitalWrite(SPEAKER, LOW); // Apaga el buzzer
  } else if (temp > 27) {
    digitalWrite(LED1, LOW); // Apaga LED verde
    digitalWrite(LED2, HIGH); // Enciende LED rojo
    digitalWrite(LED3, LOW); // Apaga LED amarillo
    tone(SPEAKER, 50); // Buzzer enciende
    delay(1500);
    //noTone(SPEAKER);
  } else if (temp < 20) {
    digitalWrite(LED1, LOW); // Apaga LED verde
    digitalWrite(LED2, LOW); // Apaga LED rojo
    digitalWrite(LED3, HIGH); // Enciende LED amarillo
    tone(SPEAKER, 1500); // Buzzer enciende
    delay(1500);
    //noTone(SPEAKER);
  }
}

```

```

}

void setup() {
  Serial.begin(115200);
  Serial.println("\nIniciando ESP8266...");

  lcd.init(); // Inicializa la pantalla LCD
  lcd.backlight(); // Enciende la luz de fondo del LCD
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
  pinMode(SPEAKER, OUTPUT);

  // Configuración inicial de los LEDs y el buzzer
  digitalWrite(LED1, HIGH); // LED verde encendido
  digitalWrite(LED2, LOW); // LED rojo apagado
  digitalWrite(LED3, LOW); // LED amarillo apagado
  digitalWrite(SPEAKER, LOW); // Buzzer apagado

  dht.begin(); // Inicia el sensor DHT11
  conectarWiFi(); // Conectar a la red WiFi

  delay(2000); // Espera inicial
}

void loop() {
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    temp = dht.readTemperature(); // Leer temperatura en °C
    hume = dht.readHumidity(); // Leer humedad

    if (isnan(temp) || isnan(hume)) {
      Serial.println("Error al leer el sensor DHT11");
      lcd.clear();
      lcd.setCursor(0, 0);
      lcd.print("Error DHT11");
      return;
    }

    // Mostrar datos en la LCD
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
    lcd.print(temp);
    lcd.print(" C");
  }
}

```

```
lcd.setCursor(0, 1);  
lcd.print("Hum: ");  
lcd.print(hume);  
lcd.print(" %");  
  
// Mostrar datos en el monitor serie  
Serial.print("Temp: ");  
Serial.print(temp);  
Serial.print(" °C | Hum: ");  
Serial.print(hume);  
Serial.println(" %");  
  
// Control de LED según temperatura  
controlarAlertas(temp);  
  
// Enviar datos a ThingSpeak  
enviarDatosTS(temp, hume);  
}  
  
delay(2000); // Pequeño retraso para evitar bloqueos  
}
```

## Apéndice B

### *Código Alerta Temperatura Análisis Matlab*

```
función enviarAlertaTemperatura()
    % Javier Hernández
    % 26/02/2025
    % Envío de alertas de temperatura en ThingSpeak

    % Datos del canal de ThingSpeak
    ID de canal = 2859760;

    % Clave de API para enviar alertas (debe ser válida)
    alertApiKey = 'TAKPoRC5H924AnKJem2' ;

    % URL para enviar alertas
    alertaUrl = "https://api.thingspeak.com/alerts/send" ;

    % Configurar opciones para la solicitud HTTP
    opciones = weboptions( "HeaderFields" , [ "ThingSpeak-Alerts-API-Key" ,
alertApiKey]);

    % Definir los límites de temperatura
    limiteAlto = 27;
    limiteBajo = 20;

    % Leer el último valor de temperatura (última medición)
    lastValue = thingSpeakRead(channelID, 'NumPoints' , 1, 'Fields' , 1)

    % Verificar si se obtuvo un dato válido
    si está vacío (último valor)
        fprintf( "No hay mediciones de temperatura disponibles.\n" );
        devolver ; % Salir de la función si no hay datos
    fin

    % Carga el estado anterior de alerta (global)
    alerta_globalEnviada ;
    si está vacío(alertaEnviada)
        alertaEnviada = false;
    fin

    % Inicializar mensaje de alerta
    cuerpoAlerta = "" ;

    % Verificar si la temperatura está fuera de rango y no se ha enviado alerta
    if (últimoValor > límiteAlto || últimoValor < límiteBajo) && ~alertaEnviada
        alertBody = sprintf( "Alerta: Temperatura fuera de rango (%.2f°C).." ,
lastValue);
```

```
        alertaEnviada = true;    % Marcar que se envió una alerta
    fin

    % Si la temperatura vuelve a la normalidad, resetear la variable
    if (ultimoValor <= limiteAlto && ultimoValor >= limiteBajo)
        alertaEnviada = false;
    fin

    % Enviar la alerta si es necesario
    si ~está vacío(alertBody)
        intentar
            webwrite(alertUrl, "body" , alertBody, "subject" , "Alerta de
Temperatura" , options);
            fprintf( "Alerta enviada: %s\n" , alertBody);
            capturar alguna excepción
            fprintf( "Error al enviar alerta: %s\n" , someException.message);
        fin
    demás
        fprintf( "Temperatura dentro del rango, no se envió alerta.\n" );
    fin
fin
```