ISSN 2591-4219

Diseño de Sistema para el Monitoreo y Registro de Eventos en Frigoríficos Farmacéuticos

Juan E. Quel [a *], Nicolás E. Bernal [a], Ricardo A. Korpys [b], Axel A. Skrauba [c]

[a] *Facultad de Ingeniería, Universidad Nacional de Misiones (UNaM), Oberá, Misiones, Argentina.*
[b] *FI-UNaM, Oberá, Misiones, Argentina.*
[c] *GIDE, IMAM UNaM-CONICET, Facultad de Ingeniería, Oberá, Misiones, Argentina.*
e-mails: juanq99@live.com, bernal.e.nicolas@gmail.com, korpys@fio.unam.edu.ar, axel.skrauba@fio.unam.edu.ar

Resumen

Este trabajo trata sobre el diseño y construcción de un sistema de monitoreo de equipos frigoríficos destinados para el almacenamiento y conservación de productos farmacéuticos. El proyecto es un trabajo en colaboración entre la Facultad de Ingeniería y el Servicio Médico Asistencial Universidad Nacional de Misiones (SMAUNaM). La motivación del proyecto es proveer una herramienta que facilite la visualización y control de las variables de temperatura y humedad relativa para asegurar el cuidado y calidad de los productos farmacéuticos, cumpliendo con las normas exigidas por los organismos ANMAT e IRAM. Para ello, se emplean alarmas visuales y auditivas que informan la ocurrencia de irregularidades en las condiciones de almacenamiento. Además, se permite la consulta y configuración remota de los equipos por medio de un bot enmarcado en el servicio de mensajería libre y gratuita Telegram. El sistema se basa en el microcontrolador *ESP8266*, que ejecuta el programa donde se gestionan las interacciones de los usuarios, las procesa y releva las variables de los frigoríficos mediante sensores *AHT15*, gestiona los eventos irregulares y los registra fehacientemente Los resultados parciales obtenidos en esta primera etapa, demuestran un funcionamiento adecuado del prototipo en relación con los objetivos principales de gestión y configuración remota de los equipos refrigerantes.

Palabras Clave *– ANMAT, Bot, ESP8266, Humedad Relativa, I2C, IRAM, Monitoreo, Temperatura, Telegram.*

1. Introducción

EEl adecuado control de la temperatura y humedad en frigoríficos de farmacias es fundamental para garantizar la calidad y seguridad de los medicamentos almacenados. Si estas condiciones ambientales no se mantienen dentro de ciertos rangos, pueden producirse alteraciones en los principios activos y excipientes, afectando la eficacia terapéutica y dando lugar a riesgos para la salud de los pacientes. Las normativas de ANMAT e IRAM establecen que los frigoríficos farmacéuticos deben operar con temperaturas entre 2°C y 8°C y humedad relativa menor al 60% [1] [2] [3].

En este trabajo se presenta el diseño de un sistema de monitoreo remoto de temperatura y humedad para los frigoríficos de las farmacias de SMAUNaM. El objetivo es alertar ante cualquier desvío de los parámetros que pudiera comprometer la calidad de los medicamentos refrigerados.

La solución propuesta consiste en un dispositivo electrónico con sensores de temperatura, humedad y apertura de puerta. Los datos son procesados por un microcontrolador y transmitidos vía internet a un bot de Telegram. De esta forma, el personal farmacéutico puede consultar el estado del frigorífico en cualquier momento y lugar, recibiendo notificaciones ante cualquier anomalía. El sistema también detecta si la puerta ha quedado mal cerrada, evitando aumentos indeseados de temperatura. Se ha previsto una autonomía de 24 horas ante cortes eléctricos.

*Autor en correspondencia.

Para los usuarios, esto se traduce en la tranquilidad de un monitoreo permanente y la garantía de conservación adecuada de los medicamentos. Desde el punto de vista técnico, el sistema ha sido diseñado de forma modular y escalable para facilitar futuras implementaciones en otras farmacias.

En este artículo se describe el circuito electrónico desarrollado, el algoritmo implementado en el microcontrolador, los cálculos para la selección de componentes y los resultados experimentales obtenidos con un prototipo en condiciones reales..

2. Circuito propuesto

En la Fig. 1 se presenta el diagrama de bloques del sistema de monitoreo, ofreciendo una representación de su estructura y funcionalidad. En el centro del esquema se encuentra el *µC*, actuando como núcleo al gestionar y coordinar todas las funciones del sistema. Para ello se utiliza el *µC ESP8266* [4], en la versión *NodeMCU V3.0*, que cuenta con integración nativa para la conexión *WiFi* necesaria para este proyecto. Además, se encarga de procesar los datos provenientes de los sensores y enviar las señales apropiadas a los componentes de salida.

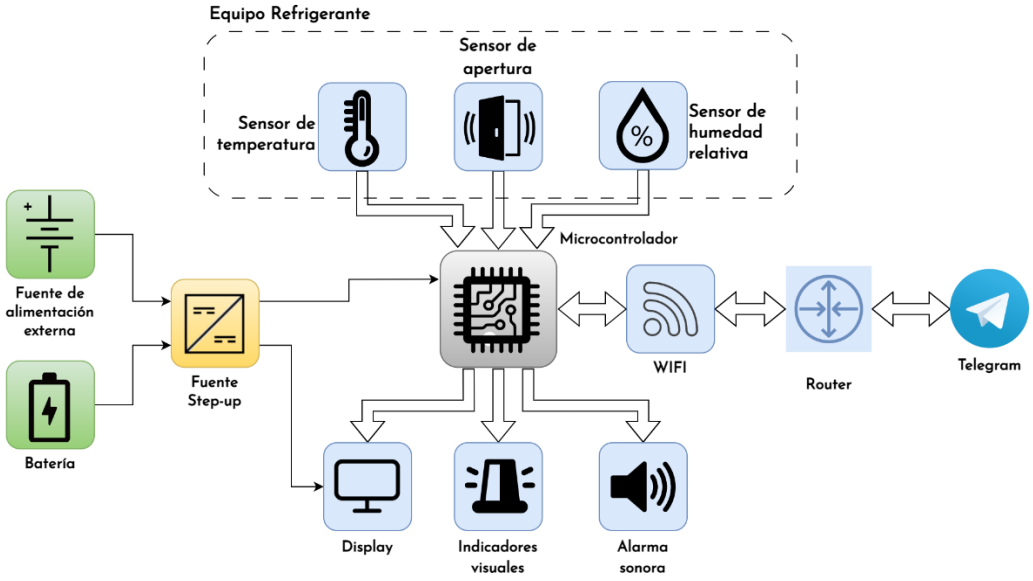


Fig. 1. Diagrama de bloques general del Sistema de monitoreo.

El sistema es alimentado por una fuente de alimentación externa de 5 V o por sistema de respaldo conformado por una batería del tipo *Li-Ion* y su respectivo módulo de carga, capaz de lograr una autonomía de 24 h. Asimismo, se utiliza una fuente *step up* para elevar el voltaje de la batería al nivel requerido por el sistema. El sistema cuenta con dos sensores *AHT15* [5], utilizados para medir la temperatura y humedad en el recinto interno del frigorífico. Además, se ha incorpora un *display*, un componente visual que muestra información relevante del sistema, como la temperatura y humedad actual, así como el estado de la conexión a internet. Además, el sistema incluye varios *LEDs* que proporcionan información adicional al usuario, como la indicación de la fuente de alimentación utilizada (externa o batería), el estado de la conexión a internet y un *LED* específico para señalar situaciones de alarma. También se implementa un *buzzer* que emite señales sonoras en caso de que

se produzca alguna alarma. El sistema posee dos pulsadores que puede usar el usuario, uno de ellos se usa para reiniciar el sistema si es necesario. El otro pulsador, multifunción, tiene dos propósitos: apagar el *buzzer* de alarma cuando se presenten anomalías y para configurar la conexión a una nueva red *WiFi*. Cuando el sistema funciona con la batería, este pulsador enciende el *display* y muestra la información actual del sistema por unos segundos, para posteriormente apagarse y contribuir al ahorro de energía, extendiendo la autonomía de las baterías.

A continuación, se presenta los tópicos principales, involucrados en el desarrollo y diseño del *hardware* de la primera versión del sistema de monitoreo que, a diferencia de la versión final, no cuenta aún con el *display* y utiliza solamente un sensor *AHT15*. Esta etapa es crucial ya que permite cumplimentar el objetivo de realizar pruebas y tener resultados experimentales, con miras a un producto final robusto y confiable.

2.1. *Comunicación I2C*

Los dispositivos esclavos, conectados mediante el protocolo Circuito Inter-Integrado (*I2C*, del inglés *Inter-Integrated Circuit*), son dos sensores *AHT15* y el *display*. En la Fig. 2 se ilustra la conexión de los dispositivos esclavos con el *µC*, empleado como dispositivo maestro. Al emplear el protocolo de comunicación *I2C*, surgen desafíos relacionados con la transmisión de datos entre el *µC* y los sensores debido al aumento de la capacitancia del bus a medida que su longitud se incrementa. Esta capacitancia provoca un aumento en los tiempos de subida de la señal de datos (*SDA*) y en la señal de reloj (*SCL*), lo que restringe tanto la frecuencia de transmisión como la cantidad de dispositivos conectados al bus. Generalmente, se recomienda un valor máximo de capacitancia de 400 pF para el protocolo *I2C* a fin de operar con tiempos de subida reducidos y a la frecuencia nominal. Sin embargo, en algunas aplicaciones, el valor máximo de capacitancia se supera. Para afrontar este problema, existen diversas técnicas disponibles. Una de ellas consiste en disminuir la frecuencia de la línea *SCL* para reducir la velocidad de transmisión de datos [6]. De esta manera, se busca cumplir con los tiempos de retención mínimos de los estados lógicos en alto y bajo de las líneas del bus. Esta técnica resulta favorable, ya que solo requiere modificaciones de *software*, sin necesidad de alteraciones en el *hardware*.

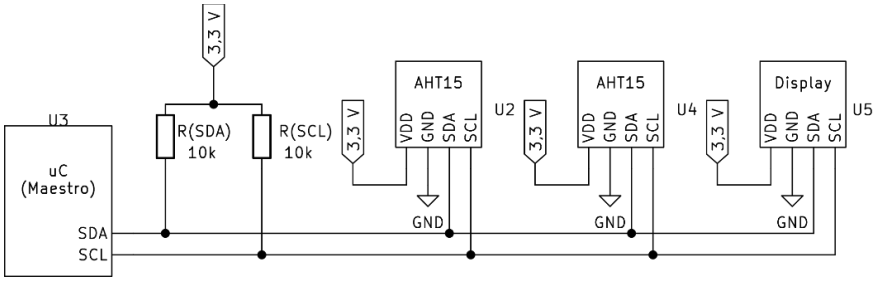


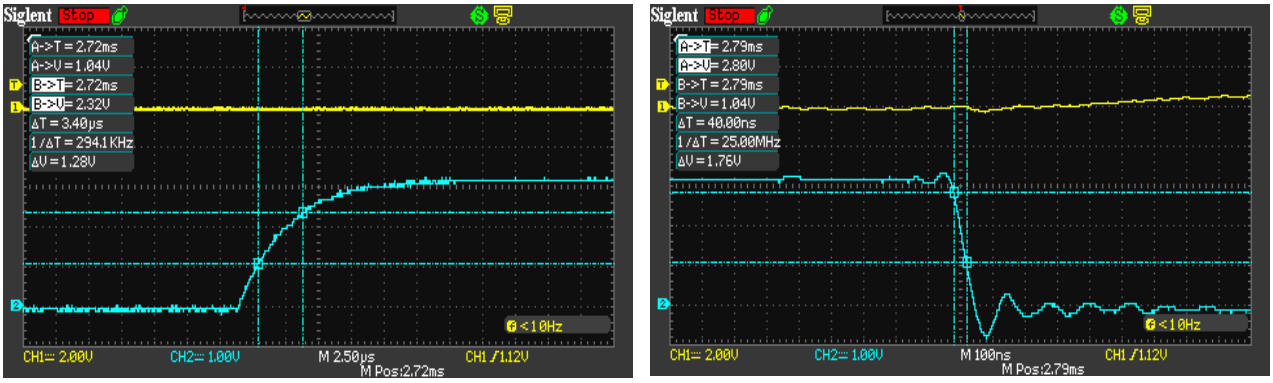
Fig. 2. Conexión de los dispositivos *I2C*.

Para comprobar esta técnica, se realizó una prueba utilizando resistencias *pull-up* de 10 kΩ y una frecuencia de *clock* (f_{SCL}) de 7 kHz, considerando una longitud de 5 metros para el bus *I2C*. El dispositivo esclavo utilizado en esta prueba es un sensor *AHT15*. Con los resultados de esta prueba

se determina la frecuencia máxima que es posible operar en la línea *SCL* mediante la siguiente ecuación:

$$f_{SCL} = \frac{1}{t_{HIGH\,(\min)} + t_{\mathrm{LOW(min)}} + t_{\mathrm{r(actual)}} + t_{\mathrm{f(actual)}}} \qquad (1)$$

Donde $t_{HIGH\,(\min)}$ y $t_{\mathrm{LOW(min)}}$ son los tiempos de establecimiento del dispositivo más limitante conectado al bus. Los tiempos $t_{\mathrm{r(actual)}}$ y $t_{\mathrm{f(actual)}}$ representan los tiempos de subida y bajada, respectivamente, determinados por la constante RC que posea el bus.



(a) **(b)**

Fig. 3. Capturas de la señal *SCL* (azul): (a) Tiempo de subida; (b) Tiempo de bajada.

En la Fig. 3, se aprecia que los tiempos de subida y bajada de la señal SCL son de 3,4 *µs* y 40 *ns*, respectivamente. Siendo los tiempos de $t_{HIGH\,(\min)}$ y $t_{\mathrm{LOW(min)}}$, para el sensor de *AHT15*, de 4,7 µs y 4 µs, respectivamente *[5]*, y reemplazando estos valores en (1) da como resultado una frecuencia de operación de hasta 82,3 kHz, siendo un poco menor a la frecuencia de operación para una transmisión estándar de *I2C* (100 kHz).

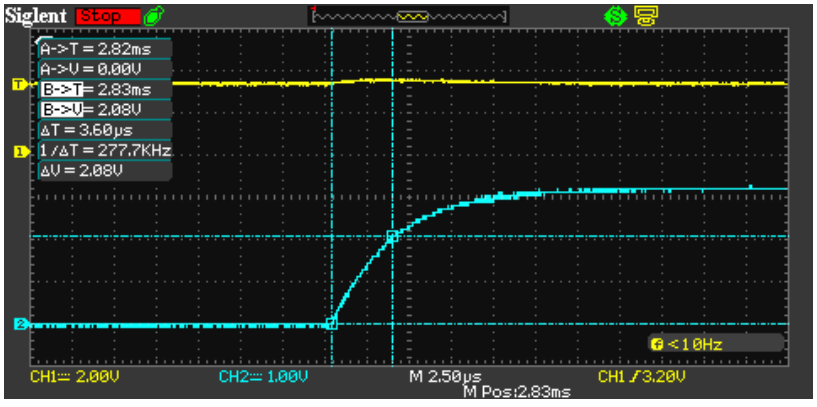


Fig. 4. Constante de tiempo τ medida de la señal *SCL*.

Con la Fig. 4, se determina la constante de tiempo (τ) que posee el bus de comunicación, que es de 3,6 µs. Dado que se usan resistencias *pull up* son de 10 kΩ, la capacidad del bus se obtiene mediante la siguiente expresión:

$$C_b = \frac{\tau}{R_{pu}} = \frac{3,6\mu s}{10k\Omega} = 360\,pF \tag{2}$$

2.2. *Configuración Load Sharing*

Al tener un circuito que funciona con una batería recargable como fuente de alimentación de respaldo, es necesario agregar una etapa de carga para la batería. Para ello, se utiliza una configuración de carga compartida o *load sharing* ubicada anterior a la fuente *step-up*, que eleva la tensión a 5 V.

El término carga compartida hace referencia a que dos o más fuentes alimentan una sola carga. Para sincronizar ambas fuentes, se usan dos diodos; la principal es la fuente de alimentación de la red eléctrica y, ante la ausencia, funciona con la batería. El módulo de carga implementa el circuito integrado *TP4056* [7] que se encarga de llevar a cabo el algoritmo de carga para la batería de Li-Ion en forma automática. Además, la salida de este módulo es monitoreada por el circuito integrado *DW01-P* que tiene como propósito principal incorporar protecciones adicionales como: protecciones contra sobrecarga, contra sobredescarga y contra sobrecorriente.

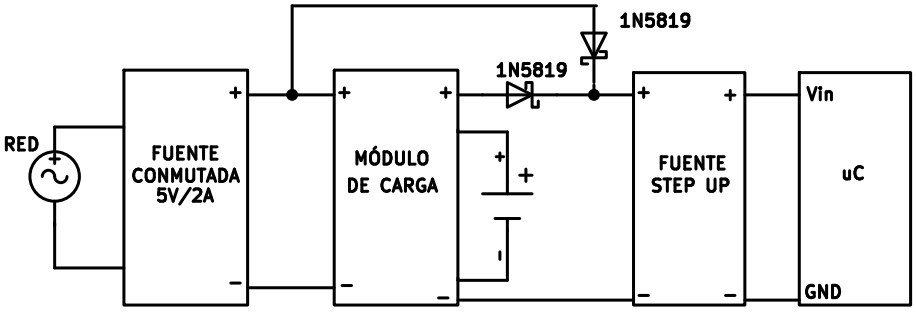


Fig. 5. Diagrama de conexión *Load Sharing*.

Se seleccionaron dos diodos *1N5819* los cuales poseen una corriente máxima de directa de 1 A, que es mayor a la demandada por el circuito; un voltaje de inversa máxima de 40 V y voltaje máximo en directa de 0,6 V siendo este pequeño valor de caída de tensión adecuado para esta aplicación ya que la fuente *step-up* funciona hasta un voltaje mínimo de 2 V de entrada.

El sistema se alimenta mediante una fuente *step-up* que eleva la tensión de la batería de reserva. Esta fuente es necesaria por las características de la placa de desarrollo *ESP826 NodeMCU V.3*, que incorpora un regulador de voltaje de 3,3 V para el microcontrolador *ESP8266*. El regulador requiere un voltaje de entrada de por lo menos 1 V superior a la de salida para funcionar correctamente. Sin embargo, la batería de *Li-Ion* solo proporciona un voltaje máximo de 4,2 V al final del periodo de

carga, lo cual no cumple con el requisito del regulador. La fuente *step-up* también garantiza una tensión constante para los componentes del sistema, evitando las variaciones de la batería durante la descarga hasta que el circuito de protección la desconecta al llegar a 3 V

3. Software implementado en el sistema

En esta sección se presenta el desarrollo del software que se implementó en el sistema para monitorear los equipos refrigerantes mediante el *µC ESP8266*.

3.1. *Entorno de desarrollo utilizado*

El editor de código *Visual Studio Code* (*VSCode*) [8] fue la herramienta de desarrollo integrada que se empleó para desarrollar el software. *VSCode* permite escribir, compilar y depurar código en diferentes lenguajes de programación, como *C++*, que se usó en este proyecto por su amplia aplicación en la programación de sistemas embebidos. Además, *VSCode* admite la instalación de extensiones que simplifican la implementación de programas en sistemas embebidos. Una de estas extensiones es *PlatformIO* [9], un complemento que facilita la creación de proyectos para diferentes plataformas embebidas, entre ellas el *ESP8266* de *ESPRESSIF*, que se utilizó para este proyecto.

3.2. Estructura del programa

El software se diseñó utilizando el enfoque de la programación orientada a objetos en el lenguaje *C++*, que brinda una mejor organización del programa y simplificación de los mecanismos de actualización de datos y estados asociados a los equipos de refrigeración que se monitorean o al propio sistema.

Para ello, en el código se definen tres estructuras de datos *Struct* donde se almacena información que es utilizado en forma constante en distintas partes parte del programa:

- Estado: Esta estructura contiene la información necesaria para el funcionamiento y el control del sistema, como el estado de alarmas y el estado de conexión de la red *WiFi* configurada, entre otros aspectos
- Equipo: Esta estructura almacena la información y procedimientos de muestreo de cada uno de los equipos refrigerantes vinculados al sistema.
- Eventos: Posee una serie de valores enteros que funcionan como contadores para gestionar la ejecución de las distintas tareas del sistema de monitoreo de manera óptima y organizada.

Posteriormente, el código se estructura según la arquitectura del *framework* de Arduino, que consiste en dos funciones principales: *setup* y *loop*. La función *setup* se ejecuta solo una vez al iniciar el programa y realiza las siguientes tareas: configuración de las entradas y salidas digitales (GPIO), definición de las fuentes de las rutinas de servicio a la interrupción (RSI), inicialización y configuración de la interfaz del bot de *Telegram* y verificación del funcionamiento de los sensores de temperatura y humedad. La función *loop* se ejecuta de forma repetitiva y contiene los algoritmos para llevar a cabo las siguientes tareas:

- ❖ Muestreo de los parámetros de temperatura y humedad.

- ❖ Consultas al servidor y atención a acciones del usuario vía Telegram.
- ❖ Envío de mensajes a usuarios para alertar ante la existencia de anomalías del sistema.
- ❖ Configuración para conexión a nueva red *WiFi*.

A continuación, se procede a detallar la configuración de los puertos de entrada y salida digitales utilizadas en el μC y el funcionamiento de las rutinas mencionadas anteriormente, acompañado de sus respectivos diagramas de flujo que ilustran el funcionamiento del sistema.

3.3. Configuración de E/S digitales

El μC *ESP8266*, posee 17 pines de entrada y salida de propósito general (GPIO0-GPIO15). Sin embargo, no todos estos pines son adecuados para su uso, ya que algunos tienen funciones específicas que pueden interferir con el funcionamiento normal del dispositivo [10], quedando disponibles solamente 11 pines GPIO, que además requieren ciertas precauciones para ser utilizados debido a que algunas tienen estados lógicos predefinidos durante el proceso de inicio del μC que, en el caso de ser alterados en forma externa, el μC no podrá iniciar su funcionamiento. Al asignar los pines para los distintos componentes seleccionados se tuvieron en cuenta estas restricciones. La Tabla 1, resume la asignación de pines realizada.

Tabla 1. Asignación y modo de operación de los GPIO.

Componentes	GPIO	Modo de operación
Sensores de temperatura y humedad (AHT15)	D1 (SDA) y D2 (SCL)	E/S
Display (OLED 128x64 pixeles)		
LED de Alarma	D3	Salida
LED de WiFi	D4	Salida
Buzzer	D6	Salida
Pulsador	D5	Entrada
Pin para detección de Red	RX	Entrada

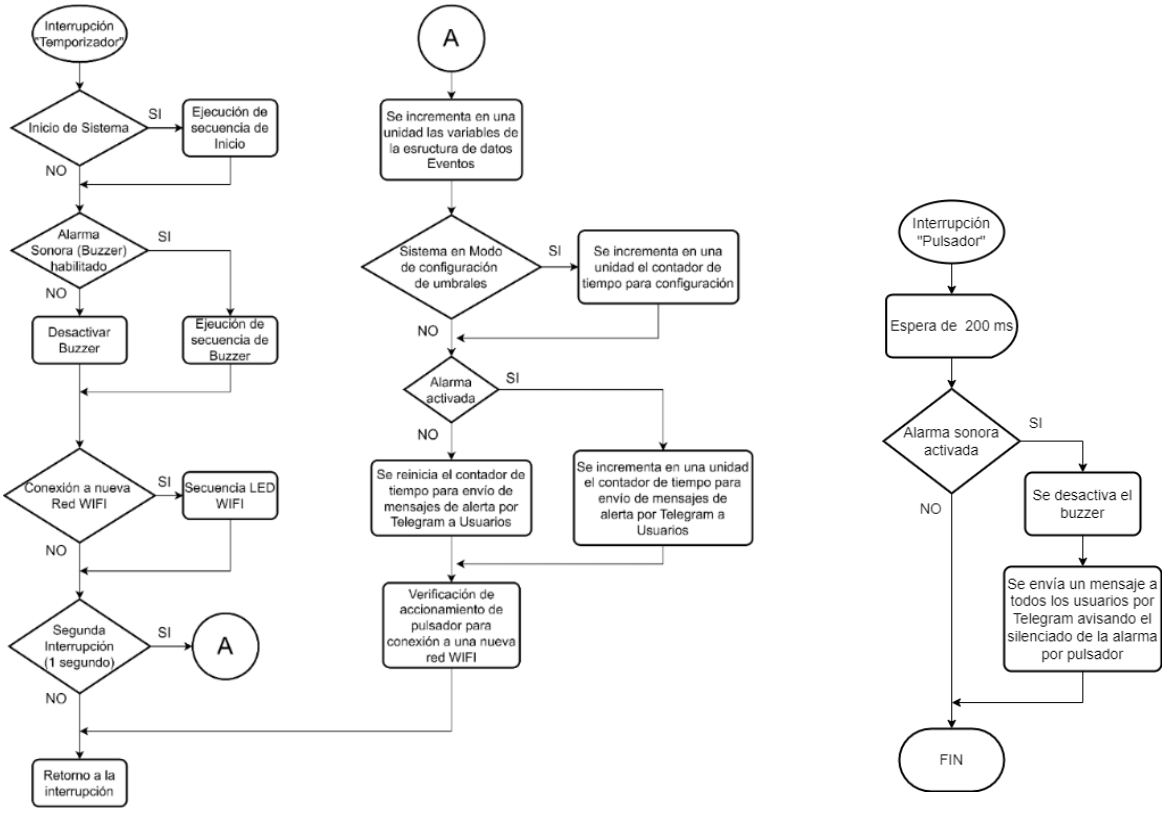
3.4. *RSI:* Temporizador

Esta interrupción se acciona mediante software dado por un temporizador/contador del μC, el cual tiene como finalidad ser utilizado como base de tiempo del sistema. El *ESP8266* cuenta con dos temporizadores/contadores, uno de 32 bits (timer0) y el restante de 23 bits (timer1). Sin embargo, el módulo timer0 se emplea internamente para gestionar las funciones relacionadas con la conectividad *WiFi*, por lo que no se recomienda su uso en el programa. Por esta razón, se hace uso del módulo timer1, ya que se puede configurar y utilizar libremente para generar interrupciones periódicas o eventos de control. El mismo se configura en el modo de operación normal, donde el registro contador (*FRC1*) se lo carga con un valor inicial y se decrementa hasta cero, momento en el que se produce una interrupción por desbordamiento. La frecuencia de las interrupciones está definida por siguiente expresión:

$$t_{TIMER1} = \frac{FREC1 \times PR}{f_{OSC}} \tag{3}$$

Donde *PR* es el valor de *prescaler* y f_{osc} es la frecuencia del oscilador utilizado por el μC que, para el *ESP8266*, es de 80 MHz. Estableciendo una base de tiempo del sistema de 500 ms, implica un valor de *PR* de 16 ciclos de reloj. Sustituyendo estos datos en (3), se deduce que el valor del contador que se debe recargar, tras cada desbordamiento, es de $2{,}5 \times 10^6$.

En la Fig. 6.a se presenta el diagrama de flujo de la rutina ejecutada de manera secuencial por cada interrupción del timer1. Las funciones principales que realiza esta rutina es encargarse de la secuencia accionamiento de los avisos visual (LEDs) y sonoros (*buzzer*) y, también, en gestionar las habilitaciones de eventos a ser ejecutados en la rutina principal.



(a) **(b)**

Fig. 6. Diagrama de flujo de las RSI: (a) Temporizador; (b) Pulsador.

Por cada interrupción, se verifica la ejecución de la secuencia del *buzzer* de la alarma sonora, la cual se desactiva automáticamente al restablecerse el funcionamiento normal o por intervención del usuario. Posteriormente, incrementa en una unidad los contadores de la estructura de datos "Eventos" cada dos interrupciones (es decir, cada 1 segundo). Además, se encarga de llevar el tiempo para cancelar los procesos de configuración de los umbrales de temperatura por parte del usuario y el

tiempo para el envío de mensajes de alerta a los usuarios, por medio de *Telegram* cuando se presentan anomalías en alguno de los equipos refrigerantes

También, la rutina se encarga de detectar el estado activo del pulsador durante un periodo de 5 segundos para activar el modo de configuración de la red *WiFi* y controlar el LED indicador del estado de la conexión *WiFi* para alertar dicho procedimiento de configuración.

3.5. *RSI:* Pulsador

Esta interrupción se acciona por hardware dado por accionamiento del pulsador que posee el sistema. Dado que se encuentra conectado al pin D5 del μC mediante una resistencia *pull-up*, se configura el modo de interrupción externa por flanco ascendente. El diagrama de flujo de la Fig. 6.b muestra la rutina ejecutada después de la interrupción externa, la cual comienza con un retardo de 200 ms para evitar falsas interrupciones causadas por el efecto rebote del pulsador.

El objetivo de esta rutina es silenciar la alarma sonora si la temperatura de alguno de los equipos refrigerantes está fuera del rango establecido. Además, da inicio al tiempo de espera para verificar el estado activo del pulsador, realizado posteriormente por RSI del temporizador, para la configuración de la red *WiFi* del sistema.

3.6. *Control y monitoreo continuo*

Teniendo en cuenta que las variaciones térmicas e higrométricas en un recinto refrigerado son lentas, se estableció una frecuencia de muestreo de 5 segundos. El diagrama de flujo que describe el muestreo de temperatura y humedad se presenta en la Fig. 7.a. Inicia realizando el muestreo de los parámetros para cada uno de los equipos refrigerantes mediante los sensores por el protocolo de comunicación *I2C* y verifica que estos valores estén dentro de los límites definidos para dicho equipo.

Si la temperatura se halla fuera del rango preestablecido, se activan alarmas visuales y sonoras. Asimismo, se envían mensajes de alerta a los usuarios que integran el sistema con la información detallada del equipo afectado y el límite de temperatura excedido. El sistema envía estos mensajes de alerta, mientras persista la anomalía, con una frecuencia de dos minutos en un total de 5 veces a todos los usuarios. El envío de estos mensajes se cancela cuando algún usuario solicita el silenciamiento de las alarmas, que también detiene la alarma sonora, quedando activa la alarma visual (LED) para indicar la continuidad del problema.

3.7. *Rutina de atención a solicitudes del bot Telegram*

La comunicación con el bot de Telegram, que permite el monitoreo remoto de las variables de temperatura y humedad y el envío de avisos de alerta, requiere la instalación de la librería *CBot*. Esta librería es una clase especializada en el manejo de bots de Telegram para plataformas de la marca *ESPRESSIF*, en particular para los modelos *ESP8266* y *ESP32*. Después de incluir esta librería, se conecta el bot a la API de Telegram Bot usando el *Token* del bot, que es un código alfanumérico único que se obtiene al crear el bot.

En la Fig. 7.b se presenta la rutina para la atención de solicitudes por parte de los usuarios habilitados por medio del bot de Telegram.

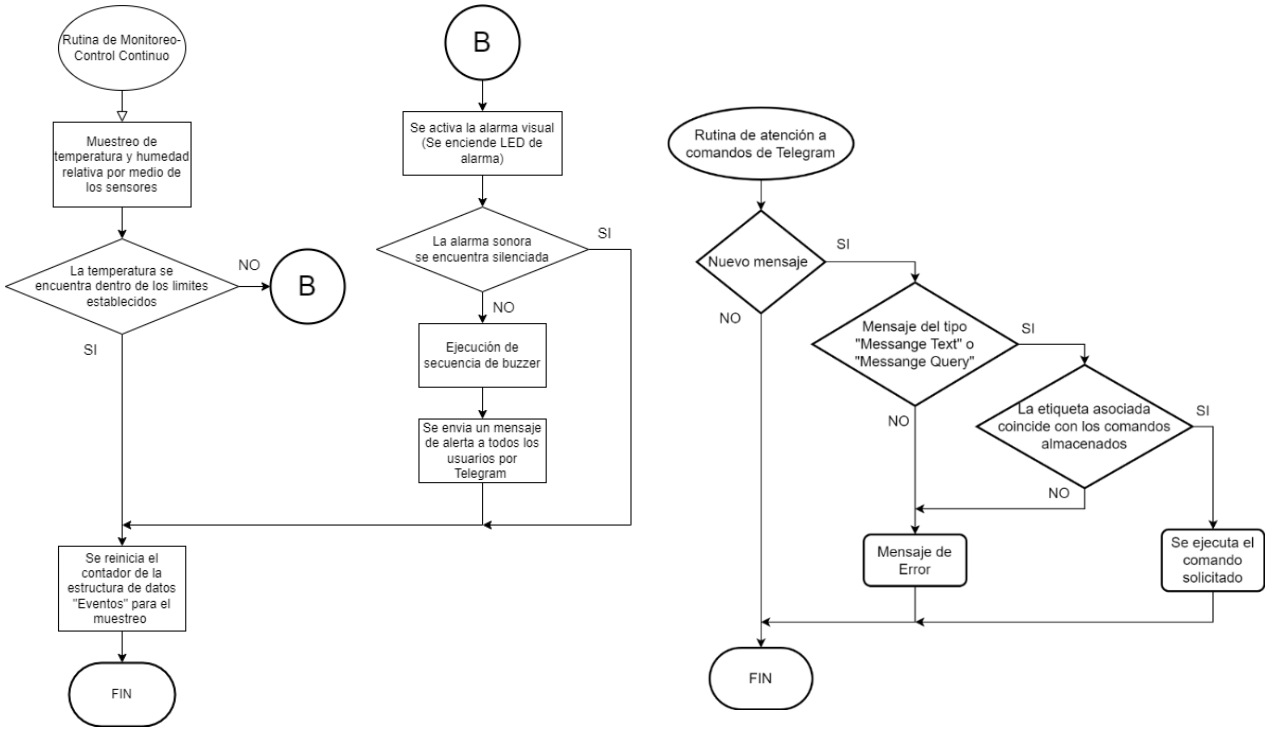


Fig. 7. Diagramas de flujo: (a) Rutina de muestreo continuo de temperatura y humedad relativa; (b) Rutina de atención de comandos por Telegram.

La secuencia inicia realizando una consulta al servidor para verificar la llegada de un nuevo mensaje por parte de algún usuario. La interfaz del bot está diseñada para que el usuario interactúe principalmente con botones de consulta (*Message Query*), salvo en algunos casos donde se requiere ingresar mensajes de texto (*Message Text*) para configurar los límites de temperatura o introducir comandos especiales. Si el mensaje recibido no corresponde a ninguno de estos tipos, el sistema envía un mensaje de error al usuario indicando que el ingreso es inválido. Cada mensaje válido tiene asociado una etiqueta en forma de comando, que el sistema compara con los comandos almacenados y ejecuta la acción solicitada. Los comandos disponibles se indican en la Tabla 2.

Tabla 2. Comandos disponibles del Sistema.

Comando	Descripción
/teclado	Despliega el menú principal del sistema.
/datos	Vinculado al botón de “Consultar Temperatura y Humedad Relativa” del menú principal. Envía un mensaje al usuario solicitante con la información actual de temperatura y HR de todos los equipos refrigerantes.
/alarm_off	Vinculado al botón de “Silenciar Alarmas” del menú principal. Silencia la alarma sonora (buzzer) si la misma se encuentra activada y envía un mensaje a todos los usuarios notificando el silenciado de la alarma y el usuario que lo realizó. Si no está activada, el sistema envía un mensaje al usuario que solicitó el comando informando esa situación.
/config	Vinculado al botón “Config. de Umbrales” del menú principal. Despliega el menú para realizar la configuración de umbrales de los equipos refrigerantes.
/set_umbral_f1	Vinculado al botón “Configurar frigorífico” del menú de configuración. Despliega el menú de umbrales para definir los límites de temperatura del equipo refrigerante un

Comando	Descripción
/set_umbral_f2	Vinculado al botón “Configurar frigorífico 2” del menú de configuración. Realiza lo mismo que el comando “/set_umbral_f1” pero para el equipo refrigerante dos.
/umbrales	Vinculado al botón “Consultar umbrales actuales” del menú de configuración. El sistema envía un mensaje al usuario que solicitó el comando con la información de los valores actuales de umbral superior y umbral inferior de todos los equipos
/umbral_sup	Vinculado al botón “Umbral Superior” del menú de umbrales para definir los límites de temperatura
/umbral_inf	Vinculado al botón “Umbral Inferior” del menú de umbrales para definir los límites de temperatura. Funciona de manera análoga al comando de “/umbral_sup” pero para el caso de umbral inferior.
/test	Comando utilizado para realizar la secuencia de prueba para comprobar el correcto funcionamiento del sistema, en lo que respecta a la alarma visuales y sonoras, como así también la comprobación del pulsador

4. Resultados

4.1. *Hardware del sistema*

La Fig. 8 presenta el esquema del circuito correspondiente de la primera versión del sistema construido, indicándose los componentes utilizados y los valores adoptados.

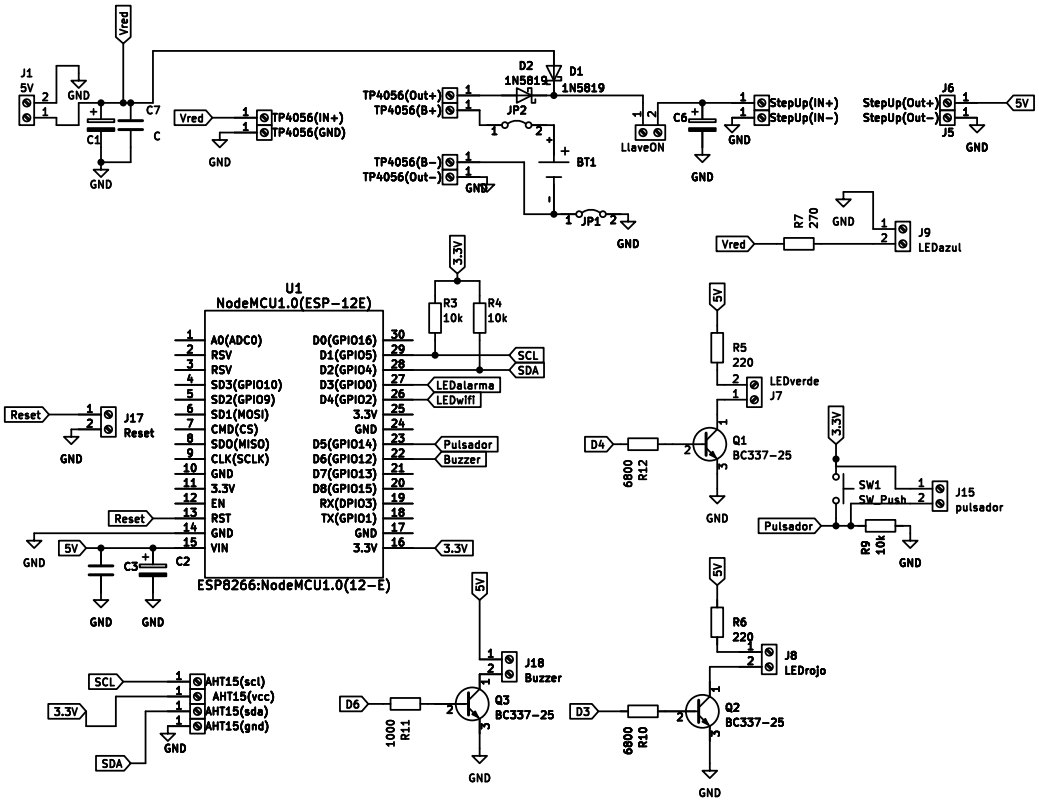


Fig. 8. Esquemático de la primera versión del Sistema.

La placa de circuito impreso (PCB) se diseñó en el paquete de software libre *KiCad* y se ilustra en la Fig. 9, desarrollado en base al esquema anteriormente mencionado.

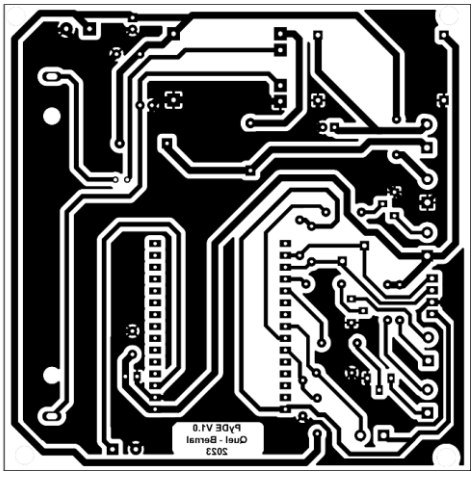
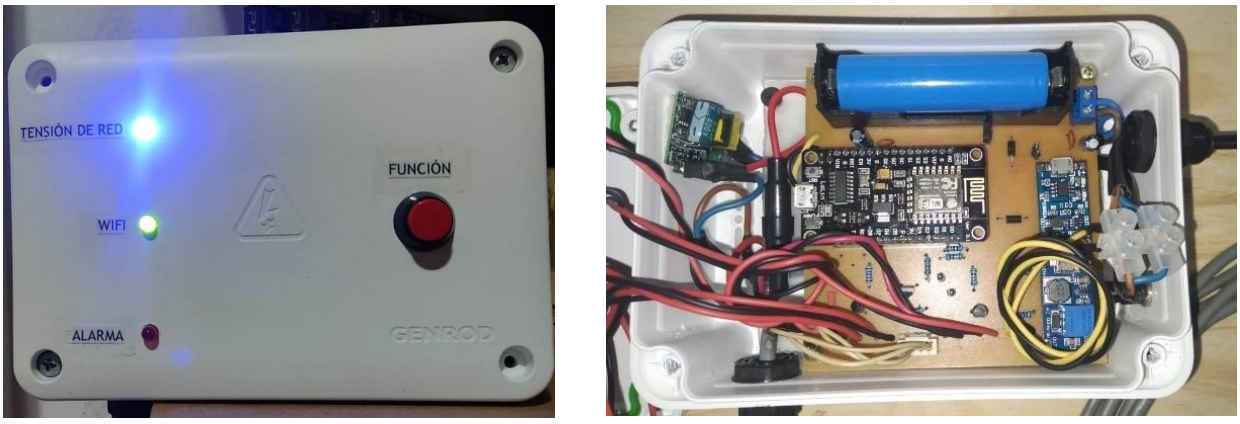


Fig. 9. Circuito impreso de la primera versión del sistema.

El PCB posee un tamaño de 10 cm × 10 cm, en el que se tuvieron en cuenta los encapsulados correspondientes a cada componente para lograr la correcta disposición física de los mismos. Seguidamente, se exponen fotografías del gabinete que contiene todas las partes del sistema.

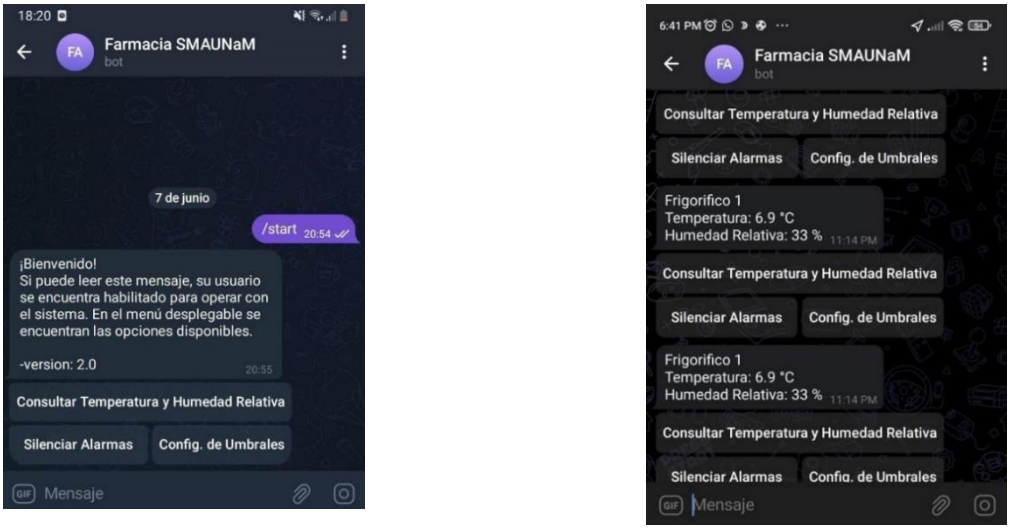


(a) **(b)**

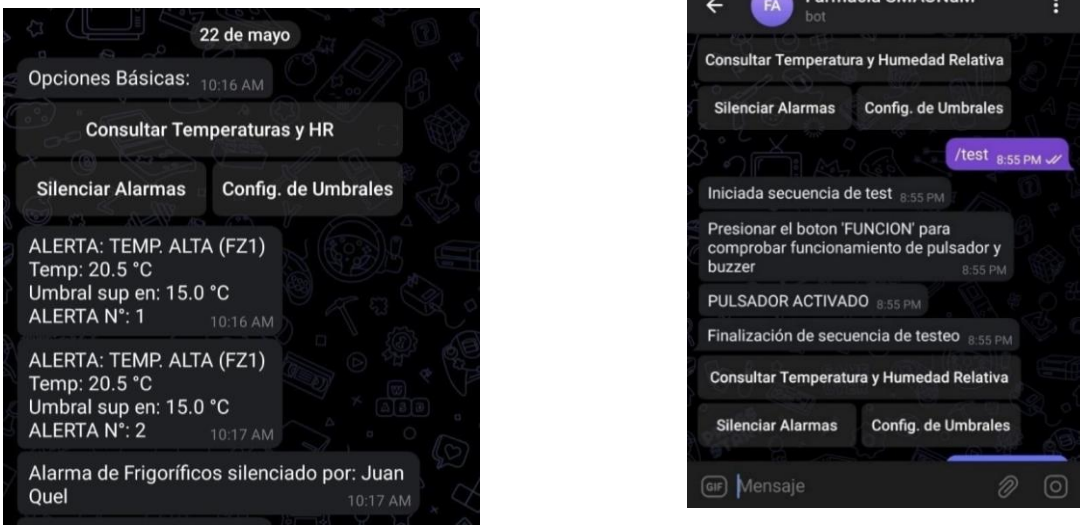
Fig. 10. Ilustraciones del gabinete.

4.2. *Interfaz de usuario (bot Telegram)*

En conjunto al montaje del sistema, se configuró una interfaz de Telegram acorde a las consideraciones de diseño de esta primera versión. Donde los usuarios registrados tienen la posibilidad de consultar la temperatura y humedad de un frigorífico, silenciar la alarma, cambiar los umbrales de temperatura máxima y mínima, o iniciar una secuencia de testeo del sistema. En la Fig. 11 se ilustran algunas de las interacciones y respuestas realizadas en el bot.



(a) **(b)**



(c) **(d)**

Fig. 11. Pantallas del bot de Telegram: (a) Pantalla de inicio; (b) Menú principal y mensaje de información del equipo refrigerante; (c) Mensajes de alerta; (d) Secuencia para comprobar el funcionamiento del sistema.

5. Conclusiones

Los resultados parciales obtenidos en esta primera etapa en el desarrollo del sistema de monitoreo de temperatura y humedad relativa para el almacenamiento de productos farmacéuticos muestran que se vienen logrando con los objetivos planteados al principio del proyecto.

La elaboración de la primera versión del sistema, que trata de una escala pequeña del producto final usando solo un sensor de temperatura y humedad relativa, permitió comprobar el funcionamiento de gran parte del sistema, ya sea en cuestiones referidas al hardware y al software. En lo respecta al hardware, se logró comprobar el correcto funcionamiento del circuito "*load sharing"*, en el que se observó una correcta transición de la fuente de alimentación por la fuente de respaldo, conformado por una *Li-Ion*, sin causar la interrupción del funcionamiento del sistema. Además, se logró comprobar el adecuado intercambio de información a larga distancia del sensor *AHT15* con el *µC* mediante el protocolo de comunicación *I2C*. Determinando así, la efectividad de la técnica de reducir la frecuencia de la línea del reloj para hacer frente a la capacidad contenida en el bus de transmisión. En lo que respecta al software, se comprobó el adecuado monitoreo de las variables de interés a través del bot de Telegram, realizándose la adecuada atención de acciones y el envío de información solicitadas por los usuarios, así como el efectivo accionamiento de las alertas visuales y sonoras del sistema y el envío de mensajes de alertas a los usuarios habilitados ante la presencia de anomalías en alguno de los equipos refrigerantes.

Los resultados obtenidos fueron satisfactorios y validados para esta primera etapa, dejando una buena base para el desarrollo de la segunda etapa del proyecto, en la que se implementarán los componentes restantes para completar la versión final del sistema de monitoreo.

6. Referencias

[1] ANMAT, *Reglamento Técnico Mercosur sobre Buenas Prácticas de Distribución de Productos Farmacéuticos,* 2005.

[2] IRAM, *37018-1:1998 "Medicamentos - Conservación de la cadena de frío en su distribución - Almacenamiento",* 1998.

[3] IRAM, *80101:2001 "Análisis clínicos Reactivos para diagnóstico in vitro. Conservación de la cadena de frío. Almacenamiento, transporte y distribución",* 2001.

[4] ESPRESSIF, «ESP8266EX Datasheet Rev. 7.0,» 2023.

[5] ASAIR, «AHT15 Technical Manual Rev. 1.0,» 2018.

[6] NXP Semiconductors, «2C-bus specification and user manual,» NXP Semiconductors, Eindhoven, 2021.

[7] Top Power ASIC, «A Standalone Linear Li-lon Battery Charger with Thermal Regulation Rev. 2.0».

[8] Microsoft, «Visual Studio Code,» [En línea]. Available: https://code.visualstudio.com/.

[9] PlatformIO, [En línea]. Available: https://platformio.org/.

[10] Random Nerds Tutorial, «ESP8266 Pinout Reference: Which GPIO pins should you use?,» 6 Mayo 2019. [En línea]. Available: https://randomnerdtutorials.com/esp8266-pinout-reference-gpios/. [Último acceso: 16 Junio 2023].