

Universidad Tecnológica Nacional

Proyecto Final

**MONITOREO REMOTO DE EQUIPOS DE FRÍO
MEDIANTE IOT**

Autor:

- Cendra Sebastian Andrés

Director:

- Ing. Sergio Burgos

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero Electrónico
en la*

Facultad Regional Paraná

Diciembre de 2022

Declaración de autoría:

Yo declaro que el Proyecto Final “Monitoreo Remoto de Equipos de Frio Mediante IoT” y el trabajo realizado es propio. Declaro:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firma:

-
-
-

Fecha:

Agradecimientos:

A mi docente director, Sergio Burgos, por el tiempo invertido y su paciencia. Además por motivarme, marcarme un norte, durante todo el proceso de realización del proyecto final. A la Universidad Tecnológica Nacional, Facultad Regional Paraná por ser el motor que impulsa a generaciones de futuros profesionales, desde el primer día. A todos los docentes de la institución, tanto a los que están como los que ya no están en medio nuestro. Cada uno aportó su granito de arena, algunos tuvieron una influencia y motivación extra en mi desarrollo como profesional.

A mi esposa, Jorgelina Becker, por ser un pilar importante en mi vida personal y todo el apoyo incondicional que me dio desde el día en que nos conocemos.

A mi familia, por ser quienes depositaron su confianza en mí a la hora de encarar este proyecto, y en especial a mi padre por darme el ejemplo a seguir como futuro ingeniero. Me dieron su apoyo emocional y económico.

A Andrés Reno, un compañero de trabajo que incondicionalmente me brindó su amistad y todos los días me motivaba para poder finalizar esta etapa.

A Benjamín y Mariano López, dos personas que me brindaron sus conocimientos sobre equipos de frío desinteresadamente. Mariano me facilitó además uno de sus clientes para las pruebas de campo del presente proyecto.

A Andres Tapari, quien me facilitó su taller y herramientas para la construcción de un circuito impreso.

Mis amigos de la infancia también son parte, y quiero agradecerles a ellos por estar conmigo a pesar de las distancias.

A todos y cada uno de ellos dedico este proyecto final.

Sebastian Andrés Cendra

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

MONITOREO REMOTO DE EQUIPOS DE FRÍO
MEDIANTE IOT

Sebastian Andrés Cendra

Abstract:

In order to fulfill the academic requirements of this university career, the developed work consisted of the design, calculation, programming, and field tests of an Internet of the Things device, capable of collecting relevant data from a refrigeration equipment (the temperature of the chamber and the current of the engines). These data were later displayed in data management tool that monitors and arranges them visually, better known as Dashboard.

The system can be divided into six main blocks. The first one consisted of the data acquisition, which was performed by means of sensors properly selected. The second one was the conditioning of the previously acquired data. Thirdly, the data were processed using a ESP32 microcontroller. This microcontroller has the necessary characteristics to connect to the internet by WiFi. This way, the processed data were published in a Broker through MQTT protocol. The sixth block was in charge of connecting to the MQTT Broker, collecting the data published in it by the microcontroller and display them in a user interface called Dashboard.

Keywords: WiFi, Internet of Things, MQTT, Monitoring System, ESP32.

Resumen:

El trabajo desarrollado para concluir con los requerimientos académicos de esta carrera universitaria, consistió en el diseño, calculo, programación y pruebas de campo de un dispositivo de Internet de las Cosas, capaz de capturar datos relevantes de un equipo de frío (temperaturas y corrientes de los motores) para luego presentarlos en una herramienta de gestión de la información que monitoriza, y los muestra de manera visual. Mejor conocido como Dashboard.

El sistema se puede dividir en seis bloques principales. El primero de ellos fue la adquisición de los datos, el cual se realizó mediante sensores apropiadamente seleccionados. El segundo fue el acondicionamiento de los datos adquiridos previamente. En tercer lugar se realizó el procesamiento de los datos en un microcontrolador ESP32. El mismo tiene las características necesarias para conectarse a internet mediante WiFi. De esta forma se publicaron los datos procesados en un Bróker mediante protocolo MQTT. El Sexto bloque fue encargado de conectarse al Bróker MQTT, adquirir los datos publicados en el mismo por el microcontrolador y mostrarlos en una interfaz usuario denominado Dashboard.

Palabras Clave: WiFi, Internet de las Cosas, MQTT, Sistema de Monitoreo, ESP32.

Reconocimientos:

Al Ingeniero Sergio Burgos por su dedicación y ayuda a realizar este proyecto. A la Universidad Tecnológica Nacional - Facultad Regional Paraná por todos los conocimientos transmitidos a lo largo de todos estos años.

Índice:

Capítulo 1: Introducción	1
1.1 Descripción del Proyecto.....	1
1.2 Fundamentación.	1
1.3 Diagrama de bloques (general).....	1
1.4 Estudio de Mercado:.....	2
1.5 Target.....	2
1.6 Pruebas de concepto. ¿Es un producto útil?	2
1.7 Pruebas de producto. Competencia. ¿Qué hay en el mercado? ¿En qué se diferencia de lo que voy a hacer?.....	2
1.8 Competencia en el mercado.	2
1.9 Análisis de recursos.....	6
1.10 Análisis de costos.	6
1.11 Ciclo de vida del producto.....	6
Capítulo 2: Desarrollo	7
2.1 Marco Teórico.	7
2.1.1 Conceptos Básicos de la Termodinámica	7
2.1.1.1 Masa, fuerza, peso, densidad y volumen específico.....	7
2.1.1.2 Presión, absoluta manométrica y de vacío.	7
2.1.1.3 Trabajo, potencia y energía.	8
2.1.1.4 Calor y temperatura.	8
2.1.1.5 Primera Ley Termodinámica, ecuación de la energía.....	9
2.1.1.6 Líquidos, vapores y cambio de estado de los mismos.....	9
2.1.1.7 Refrigeración por evaporación.	10
2.1.1.8 Ley de los gases perfectos.	10
2.1.1.9 Usos de la refrigeración.....	10
2.1.1.10 Método de refrigeración.	11
2.1.1.11 Equipo de refrigeración.	11
2.1.1.12 Refrigeración por vaporización de un líquido.....	11
2.1.1.13 Proceso de Expansión.....	11
2.1.1.14 Proceso de Evaporación.	12
2.1.1.15 Recuperación del refrigerante.	12
2.1.1.16 El sistema de refrigeración por compresión de vapor.	12

2.1.1.17	Proceso de Expansión (A-B).	12
2.1.1.18	Proceso Evaporación (B-C).	13
2.1.1.19	Proceso de Compresión (C-D).	13
2.1.1.20	Proceso de Condensación (D-A).	13
2.1.2	Transductores y Sensores.	14
2.1.2.1	Sensores de Temperatura.	14
2.1.2.2	Termistores.	14
2.1.2.3	Transformadores.	15
2.1.2.4	Acondicionamiento de señal.	16
2.1.3	Internet de las Cosas (IoT).	17
2.1.3.1	Áreas de desarrollo más fuertes de IoT.	17
2.1.3.2	Características de los dispositivos IoT.	19
2.1.3.3	Placas de desarrollo de microcontroladores.	19
2.1.3.4	Estándares y tecnologías de red.	19
2.1.3.5	Protocolos de red de IoT.	20
2.1.3.6	Plataformas IoT para agilizar el desarrollo de aplicaciones IoT.	21
2.1.4	Protocolo MQTT.	21
2.1.4.1	Patrón de publicación/suscripción de MQTT.	22
2.1.4.2	Escalabilidad.	22
2.1.4.3	Filtrado de Mensajes.	23
2.1.4.4	Principales Características.	23
2.1.4.5	Cliente y Bróker MQTT.	23
2.1.4.6	Conexión MQTT.	24
2.1.4.7	Publicar.	24
2.1.4.8	Suscribir.	25
2.1.4.9	Cancelar suscripción.	26
2.1.5	NodeRed y Dashboard.	26
2.1.5.1	Conceptos.	26
2.1.5.2	Dashboard.	28
2.2	Desarrollo.	30
2.2.1	Diagrama de bloques general del proyecto.	30
2.2.2	Estudio Equipo de Frío.	30
2.2.3	Investigación de componentes.	31
2.2.3.1	Sensor de temperatura.	31

2.2.3.2	DS18B20.....	31
2.2.3.3	Sensor de corriente.....	36
2.2.3.4	SCT-013.....	36
2.2.3.5	ESP32.....	39
2.3	Propuestas de circuitos o esquemas.....	40
2.3.1	Pruebas, experiencias.....	46
2.3.2	Pruebas Sensor de Temperatura.....	46
2.3.3	Pruebas Sensor Corriente.....	48
2.4	Desarrollo del software o firmware.....	52
2.4.1	Visual Studio Code.....	52
2.4.2	Diagrama en bloques programa principal.....	55
2.4.3	Inicializar LEDs.....	55
2.4.4	Conectar a la red WiFi, ESP Touch.....	56
2.4.5	Conexión a Bróker MQTT (HiveMQ).....	59
2.4.6	Inicializar sensores temperatura.....	64
2.4.7	Inicializar sensores de corriente.....	64
2.4.8	Lectura sensores temperatura.....	65
2.4.9	Lectura sensores corriente.....	66
2.4.10	RSSI.....	67
2.4.11	Funciones buscar sensores e invertir direcciones.....	68
2.5	Interfaz HMI (Dashboard).....	68
2.5.1	IBM Cloud, servicios y NodeRed.....	69
2.6	Diseño completo.....	76
2.7	Instalación y modo de uso.....	79
2.7.1	Paso 1. Estudio Cámara Frigorífica.....	79
2.7.2	Paso 2. Instalación.....	81
2.7.3	Paso 3. Puesta en funcionamiento.....	83
2.7.4	Paso 4. Dashboard e identificación de sensores de temperatura.....	84
2.8	Clase del instrumento.....	87
Capítulo 3: Resultados.....		92
Capítulo 4: Análisis de Costos.....		93
4.1	Costos Materiales.....	93
4.2	Mano de Obra.....	93
4.3	Servicios IBM Cloud y HiveMQ.....	94

4.4	Valor de Venta en el Mercado.....	95
	Capítulo 5: Discusión y Conclusión.....	97
	Capítulo 6: Literatura Citada.....	98

Lista de Figuras:

Figura 1 - Diagrama de bloques, general	1
Figura 2 - Control Electrónico Refrigeración ERC112C	2
Figura 3 - Data Logger DXT-2412 WiFi - Tensión Corriente Potencia	3
Figura 4 - Data Logger WiFi DTX1720 - Registrador de Datos Temperatura	5
Figura 5 - Relación entre las presiones absoluta, manométrica y de vacío	8
Figura 6 - Formas de energía	8
Figura 7 - Flujo de calor entre dos cuerpos	9
Figura 8 - Cambios de estado del agua	10
Figura 9 - Refrigeración obtenida por la vaporización de un líquido	11
Figura 10 - Sistema Básico de refrigeración por compresión de vapor	13
Figura 11 - PTC y NTC	14
Figura 12 - Diagrama de un transformador	15
Figura 13 - Arquitectura MQTT	22
Figura 14 - Modelo MQTT	24
Figura 15 - Publicación MQTT	25
Figura 16 - Suscripción MQTT	26
Figura 17 - Interfaz NodeRed	28
Figura 18 - Ejemplo Dashboard	28
Figura 19 - Nodos disponibles en la paleta Dashboard	29
Figura 20 - Diagrama en bloques general	30
Figura 21 - DS18B20 Topología del Bus	32
Figura 22 - DS18B20 Pulsos inicialización	33
Figura 23 - DS18B20 Pulsos de Escritura y Lectura	33
Figura 24 - DS18B20 Presentación formato sumergible	34
Figura 25 - DS18B20 Presentación formato TO-92	34
Figura 26 - DS18B20 Pines	35
Figura 27 - Presentación SCT-013	36
Figura 28 - SCT-013 Terminales de conexión	38
Figura 29 - SCT-013 Circuito Interno	38
Figura 30 - PINOUT NODEMCU-ESP32	40
Figura 31 - Esquemático Principal	41
Figura 32 - Fuente Alimentación	41
Figura 33 - Pines ESP32	42
Figura 34 - Sensores Temperatura	43
Figura 35 - Adaptación Señal Sensor Corriente	44
Figura 36 - Sensores Corriente	45
Figura 37 - Detección Presencia Sensor	45
Figura 38 - LEDs Indicadores	46
Figura 39 - Conexión DS18B20 al ESP32	47
Figura 40 - Pruebas Sensor Temperatura	48
Figura 41 - Señal Continua, con Offset y Muestreada	49
Figura 42 - Gráfica Muestras ADC Sensor Corriente	51

Figura 43 - Grafica Voltajes Instantáneos Sensor Corriente	51
Figura 44 - Señal Osciloscopio	52
Figura 45 - Plataforma Visual Studio Code	53
Figura 46 - PlatformIO IDE	53
Figura 47 - Nuevo Proyecto	54
Figura 48 - Campos Proyecto Nuevo	54
Figura 49 - Programa Principal ESP32	55
Figura 50 - Diagrama Conexión a Red WiFi	56
Figura 51 - PlayStore: EspTouch	58
Figura 52 - EspTouch Interfaz	58
Figura 53 - SmartConfig y EspTouch	59
Figura 54 - HiveMQ Ingreso	60
Figura 55 - HiveMQ Nuevo Cluster	60
Figura 56 - HiveMQ Credenciales Cluster	61
Figura 57 - Configuración MQTT	63
Figura 58 - Inicialización Sensores Temperatura	64
Figura 59 - Inicialización Sensores Corriente	65
Figura 60 - Lectura Sensores Temperatura	65
Figura 61 - Lectura Sensores Corriente	66
Figura 62 - IBM Catalogo Servicios	69
Figura 63 - IBM Configuración NodeRed	69
Figura 64 - IBM Detalles Aplicación	70
Figura 65 - IBM Automatización de la implementación	70
Figura 66 - IBM Clúster	71
Figura 67 - IBM Despliegue finalizado	71
Figura 68 - NodeRed Instalación	72
Figura 69 - NodeRed Propiedades Nodo MQTT	72
Figura 70 - NodeRed Nuevo Bróker MQTT	73
Figura 71 - NodeRed Programa Completo	74
Figura 72 - NodeRed Programa del Nodo Function	74
Figura 73 - NodeRed Dashboard	75
Figura 74 - NodeRed Sensores Temperatura	75
Figura 75 - Diseño PCB	76
Figura 76 - PCB Grabado Laser	77
Figura 77 - PCB Impreso	77
Figura 78 - PCB Pasado por Percloruro Férrico	77
Figura 79 - PCB Componentes Soldados	78
Figura 80 - PCB Instalado en Gabinete Abierto	78
Figura 81 - PCB Instalado en Gabinete Cerrado	79
Figura 82 - Diagrama de Conexiones	82
Figura 83 - Instalación Sensores Corriente	82
Figura 84 - Instalación Sensores Temperatura	82
Figura 85 - Puesta en funcionamiento	83
Figura 86 - Dispositivo Conectado a Internet y Bróker MQTT	84

Figura 87 - Dashboard menú	84
Figura 88 - Dashboard Menú Sensores Temperatura	85
Figura 89 - Dashboard Estado WiFi y Sensores	85
Figura 90 - Dashboard Realtime e Histórico Temperatura	86
Figura 91 - Dashboard Realtime e Histórico Corriente	86
Figura 92 - Resultados Detalle Histórico Temperatura	87
Figura 93 - Resultados Detalle Histórico Corrientes	87
Figura 94 - Medición Corriente Motor Evaporador	88
Figura 95 - Corriente SCT-013-005 vs Instrumento Patron	88
Figura 96 - Medición Corriente Compresor	89
Figura 97 - Corriente SCT-013-010 vs Instrumento Patrón	90
Figura 98 - Temperaturas DS18B20 vs Instrumento Patrón	91
Figura 99 - Testo 115i	91
Figura 100 - Ingresos Costos Utilidad Balance	96

Lista de Tablas

Tabla 1 - Características ERC112C	3
Tabla 2 - Análisis de costos	6
Tabla 3 - Modelos OSI y TCP/IP	19
Tabla 4 - Modelos TCP/IP e IoT	20
Tabla 5 - Características Equipo Frío	30
Tabla 6 - Opciones Sensores de Temperatura	31
Tabla 7- DS18B20 Formato Registro de Temperatura	34
Tabla 8 - DS18B20 Resistencias PullUp	35
Tabla 9 - Opciones Sensores Corriente	36
Tabla 10 - Modelos SCT-013	38
Tabla 11 - Selección Pines ESP32	42
Tabla 12 - Atenuación ADC	49
Tabla 13 - Muestras ADC Sensor Corriente	50
Tabla 14 - Voltajes Instantáneos Sensor Corriente	50
Tabla 15 - Tópicos MQTT	62
Tabla 16- Cobertura RSSI	67
Tabla 17 - Casos y Condiciones Sensores Viejos y Nuevos	68
Tabla 18 - Características Eléctricas Compresor	80
Tabla 19 - Motor Forzador Evaporador	81
Tabla 20 - Corriente SCT-013-005 vs Instrumento Patron	88
Tabla 21 - Corriente SCT-013-010 vs Instrumento Patron	89
Tabla 22 - Temperaturas DS18B20 vs Instrumento Patrón	90
Tabla 23 - Costo Materiales	93
Tabla 24 - Costo, Precio Venta, Utilidad, Balance	95

Lista de Abreviaciones

1-Wire	Protocolo de comunicaciones en serie
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
A/D	Analógico/Digital
AMQP	Advanced Message Queuing Protocol
Bit	Digito en sistema binario
Bps	Bits per second
CDMA	Code Division Multiple Access
CoAP	Constrained Application Protocol
DDoS	Denial of Service
ESP32	Familia de chips SoC desarrollado por Espressif Systems
ESP8266	Chip de bajo costo WiFi
GPIO	General Purpose Input Output
GSM	Global System for Mobile Communications
HTTPS	Hypertext Transfer Protocol Secure
I2C	Inter-Integrated Circuit
I2S	Integrated Interchip Sound
IA	Inteligencia Artificial
IDE	Integrated Drive Electronics
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IPv6	Internet Protocol version 6
ISO	Open System Interconnection
LTE	Long Term Evolution
MHz	Mega Hercios
MQTT	Message Queuing Telemetry Transport
PCB	Printed Circuit Board
PTD	Platinum resistance temperature detector
PWM	Pulse Width Modulation
QoS	Quality of Service
RMS	Root Mean Square
RPL	Routing Protocol for Low power and Lossy Networks
RSSI	Received Signal Strength Indicator
RTD	Resistance temperature detector
SD	Secure Digital
SoC	System on Chip
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
SSID	Service Set Identifier
TCP-IP	Protocolo comunicaciones e Internet
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
UI	User Interface

USB	Universal Serial Bus
WiFi	Wireless Fidelity
XMPP	Extensible Messaging and Presence Protocol

Dedicado a:

Mi Familia y Amigos.

Capítulo 1: Introducción

1.1 Descripción del Proyecto.

El objetivo de este proyecto es realizar un sistema de monitoreo para equipos de frío (cámaras frigoríficas), mediante Internet de las Cosas (IoT).

Se trata de un dispositivo encargado de medir la temperatura tanto en el ambiente como la temperatura de evaporación en la unidad de frío. Además la corriente eficaz en los motores del compresor, evaporador y condensador de la unidad. El dispositivo debe reportar los datos sensados mediante enlace inalámbrico a un servidor de datos para luego realizar monitoreo en tiempo real. Asimismo se propone realizar el sistema de gestión mediante enlace inalámbrico y levantar un servicio en la nube para realizar el monitoreo desde cualquier lugar con acceso a internet.

1.2 Fundamentación.

La necesidad en la industria de tener los dispositivos conectados a internet, cada vez se está haciendo más cotidiano. La posibilidad de tener registro del estado de las variables importantes de un equipo o un conjunto de equipos de frío y a través de internet, nos permitirá tener mejores herramientas a la hora de realizar diagnóstico de fallas y/o mantenimiento. De esta forma disminuir los tiempos de traslado e inspección del técnico a la hora de realizar un diagnóstico de un equipo. Poder tener una perspectiva general del funcionamiento del equipo antes de trasladarse al lugar donde se encuentra el mismo.

Durante el transcurso de este proyecto, quiero poner en práctica la mayor cantidad de los conocimientos adquiridos a lo largo de la carrera.

1.3 Diagrama de bloques (general).

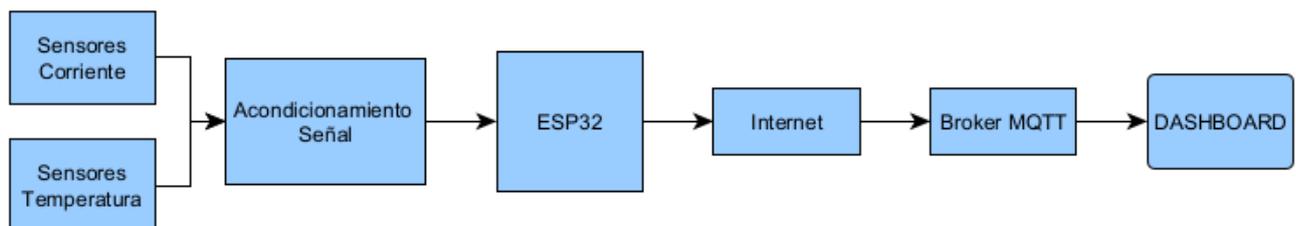


Figura 1 - Diagrama de bloques, general

Observamos en primer lugar el sensado de dos variables elegidas apropiadamente, las temperaturas del evaporador y ambiente. Asimismo las corrientes eficaces de los tres motores que intervienen en el equipo, compresor, evaporador y condensador.

Una vez capturadas estas variables físicas, acondicioné su señal mediante circuitos apropiados para el ingreso a un microcontrolador. El mismo está en un esquema ordenador, el cual proporcionó una conexión física WiFi para poder conectarlo a internet. Los datos fueron procesados para interactuar con un Bróker MQTT, el cual se encargó de gestionar las conexiones entre el ESP32 y el Dashboard. El Dashboard se conectó a un Bróker MQTT, recibió los datos que fueron proporcionados por el ESP32 al Bróker y los mostró en una interfaz HMI.

1.4 Estudio de Mercado:

1.5 Target.

Al tipo de cliente que apunta este producto es al técnico de soporte de equipos de frío, instaladores y dueños de cadenas de sistemas de refrigeración. Donde los mismos desean monitorear el estado de sus equipos y por consiguiente garantizar la calidad de los productos que tienen almacenados en los mismos.

1.6 Pruebas de concepto. ¿Es un producto útil?

Desde el punto de vista del usuario final, es una herramienta útil a la hora de controlar el estado y calidad de los productos almacenados en los equipos donde se instale este dispositivo. La forma de realizar mediciones de una forma continua garantiza tener una mejor comprensión del funcionamiento del equipo y poder detectar posibles fallas a tiempo.

1.7 Pruebas de producto. Competencia. ¿Qué hay en el mercado? ¿En qué se diferencia de lo que voy a hacer?

En el mercado existen equipos que sensan solamente la temperatura en el interior o, en el interior y el exterior del equipo. Sin tener en consideración la corriente de los motores de la cámara frigorífica. En caso de falla, la notificación/alarma es mediante un display la cual se encuentra ubicada en el equipo instalado. Por otro lado existen equipos que solamente sensan una corriente y no tienen en cuenta la temperatura. La posibilidad de conexión a internet existe pero es otro tipo de equipos los que realizan esta tarea, es necesario integrar un equipo adicional, el cual encarece mucho el producto final.

1.8 Competencia en el mercado.

Control Electrónico Refrigeración ERC 112C



Figura 2 - Control Electrónico Refrigeración ERC112C

Fuente: https://assets.danfoss.com/photos/preview/latest/145760/ID107325684990-0101_preview.jpg?w=300&h=300

Color de pantalla	Azul
Digital inputs (DI)	1 Uds.
Entrada analógica	4 Uds.
Frecuencia [Hz]	50/60 Hz
Grupo de productos	Controladores de temperatura
Nombre de producto	Control de refrigera. electrónico
Reparable	No
Salidas digitales (DO) [Uds.]	4 Uds.
Tensión de alimentación [V] c.a. [máx.]	240 V
Tensión de alimentación [V] c.a. [mín.]	100 V
Tipo	ERC 112C
Tipo de controlador	Refrigeración comercial
Área de aplicación	Expositores con puerta de cristal

Tabla 1 - Características ERC112C



Figura 3 - Data Logger DXT-2412 WiFi - Tensión Corriente Potencia

Fuente: https://http2.mlstatic.com/D_NQ_NP_737454-MLA52370540258_112022-O.webp

Registrador Wi-Fi de tensión, corriente y potencia que permite almacenar y darle seguimiento a sus respectivas variaciones de forma remota (App).

- Sensor corriente: 30A (Pinza Efecto Hall) (Cant.:1)
- Sensor tensión (Interno): Sensa la tensión de entrada del equipo.
- Cálculo Potencia: La plataforma calculará la potencia de acuerdo a las variables de tensión y corriente sensadas.
- Transmisión de datos: Wi-Fi 2.4GHz
- Tiempo de transmisión: 1 minuto
- Alertas: Sí - Notificación a través de la App
- Gráficos: Sí - Smartphone ó Página web
- Exportar datos: Sí - Desde la PC (Archivo .xls)

Toda la información se transmite a través de la red Wi-Fi y puede visualizarse a través de una aplicación de teléfono inteligente o navegador. Además los datos serán reflejados en gráficos para poder analizar las variaciones y tomar decisiones con información confiable.

Este Data Logger Wi-Fi es un equipo versátil adecuado para una amplia cartera de rubros, farmacéuticos, almacenamiento, gabinetes, laboratorios.

DATOS TECNICOS:

- Sensor de corriente - SCT013 30A/1V
- Tamaño de apertura 13mm x 13mm
- Cantidad de sensores: 1
- Longitud sensor: 1mts
- Intervalo de medición del datalogger: 1 minuto
- Memoria del servidor: 3 meses
- Alimentación: 100-240VAC 50Hz
- Tamaño del equipo: 85x74x32mm.



Figura 4 - Data Logger WiFi DTX1720 - Registrador de Datos Temperatura

Fuente: https://http2.mlstatic.com/D_NQ_NP_620228-MLA50824941734_072022-O.webp

Registrador Wi-Fi de temperatura que permite almacenar y darle seguimiento a las variaciones de temperatura en las zonas que se deseen. Cuenta con tres sensores basados en la tecnología de Internet de las cosas (IoT). Los mismos registran temperatura.

Toda la información se transmite a través de la red Wi-Fi y podrá visualizarse a través de una aplicación de teléfono inteligente o navegador. Además los datos son reflejados en gráficos para poder analizar las variaciones y tomar decisiones con información confiable.

Este Datalogger Wi-Fi es un equipo versátil adecuado para una amplia cartera de rubros, farmacéuticos, almacenamiento, gabinetes, laboratorios.

DATOS TECNICOS:

- Temperatura de funcionamiento: -55 a 125 C
- Exactitud: $\pm 0,5$ C (de -10 a +85 C)
- Resolución: 0,1 C
- Sensor: DS18B20 (Sumergible)
- Cantidad de sensores: 3
- Longitud sensor: 2mts
- Intervalo de medición: 1 minuto
- Memoria del servidor: 3 meses
- Tipo de fuente: Switching-Entrada: 100-240VAC 50Hz

1.9 Análisis de recursos.

Tengo a disposición algunos recursos físicos, y otros económicos para la realización del proyecto presentado. Respecto del tiempo, lo realizaría en el tiempo que pueda disponer, luego del horario de mi trabajo particular, ya que tengo que cumplir con un horario diario y semanal. Este producto está dirigido a equipos de refrigeración industrial.

1.10 Análisis de costos.

Concepto	Costo USD	Aprox. \$
Componentes electrónicos	\$ 10	\$ 3.200
ESP32	\$ 14	\$ 4.480
Sensores	\$ 40	\$ 12.800
Gabinete	\$ 5	\$ 1.600
Mano de Obra	\$ 50	\$ 16.000
TOTAL	\$ 119	\$ 38.080

Tabla 2 - Análisis de costos

1.11 Ciclo de vida del producto.

Se estima una vida útil del producto a desarrollar en unos 10 años aproximadamente.

Capítulo 2: Desarrollo

2.1 Marco Teórico.

2.1.1 Conceptos Básicos de la Termodinámica

2.1.1.1 Masa, fuerza, peso, densidad y volumen específico.

La masa (m) de un objeto o cuerpo es la cantidad de materia que contiene. Su unidad es el kilogramo.

La fuerza (f) es el empuje o atracción que un cuerpo ejerce sobre otro. Su unidad es el Newton.

El peso de un cuerpo (p) es la fuerza ejercida sobre el mismo por la atracción gravitacional de la tierra. Esta es la diferencia entre la masa, el peso es una fuerza.

La densidad (d) es la masa por unidad de volumen de una sustancia.

$$d = \frac{m}{\text{volumen}}$$

El volumen específico (v) es el recíproco de la densidad.

$$v = \frac{\text{volumen}}{m}$$

2.1.1.2 Presión, absoluta manométrica y de vacío.

La presión (p) se define como la fuerza ejercida por unidad de área.

$$p = \frac{\text{fuerza}}{\text{area}}$$

Un espacio del que se ha evacuado todo el gas o líquido, tiene presión cero. La presión ejercida por un fluido por encima del valor de cero se conoce como presión absoluta.

El aire atmosférico sobre la Tierra ejerce una presión a causa de su peso. Se ha medido la presión que ejerce el aire al nivel del mar, de esta forma se determinó la presión atmosférica absoluta. La presión atmosférica varía además en función de la altitud sobre el nivel del mar. A mayor altura, menor presión.

Los instrumentos utilizados para medir presión, se construyen para medir la diferencia de presión entre la presión de un fluido y la presión atmosférica. La presión por encima de la presión atmosférica es la presión manométrica.

Cuando un fluido ejerce una presión inferior a la presión atmosférica, la diferencia con la presión atmosférica se llama presión de vacío.

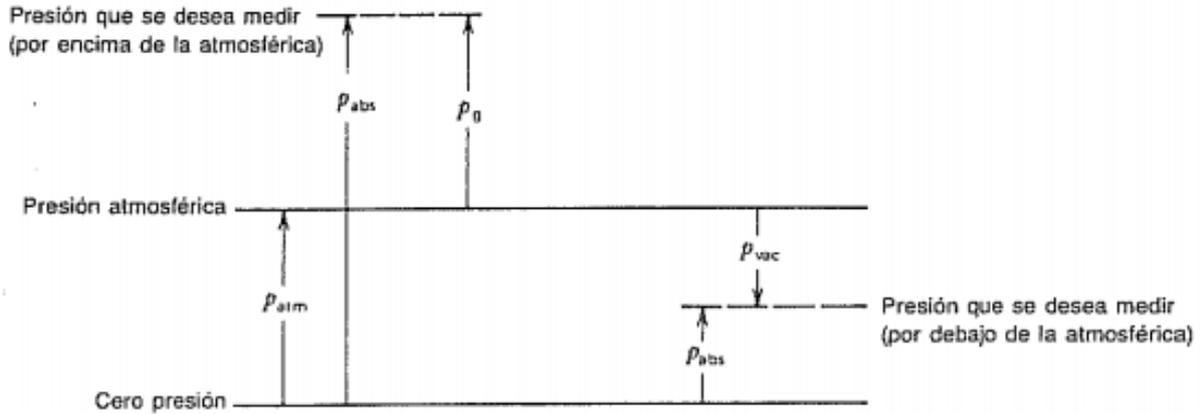


Figura 5 - Relación entre las presiones absoluta, manométrica y de vacío

Fuente: E.G. Pita, Principios y Sistemas de Refrigeración. 1ra ed. México D.F: LIMUSA S.A., 1991 - Pág. 22

2.1.1.3 Trabajo, potencia y energía.

El trabajo es el efecto creado por una fuerza cuando desplaza a un cuerpo de un lugar a otro. Su unidad es el Joule

$$trabajo = fuerza \times desplazamiento$$

La potencia es la rapidez con la que se realiza el trabajo

$$potencia = \frac{trabajo}{tiempo}$$

La energía es la capacidad de realizar un trabajo. El trabajo es una de las formas de energía. Puede existir en varias formas. Algunas formas de energía se almacenan en los cuerpos y otras formas son de transferencia cuando hay movimiento entre los cuerpos. En la Figura 6, podemos ver algunos ejemplos de los distintos tipos de energía almacenada y de transferencia.

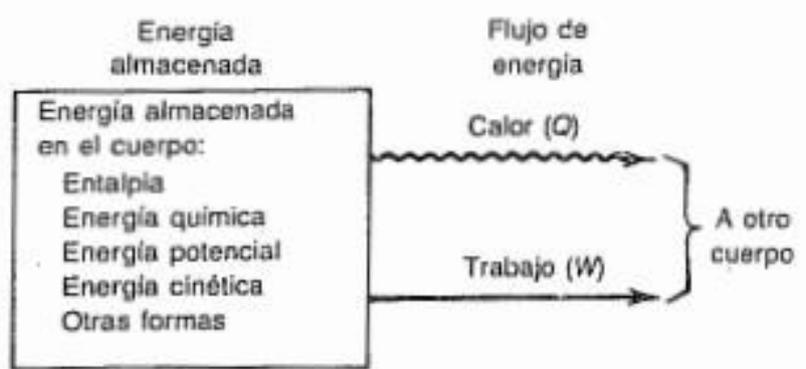


Figura 6 - Formas de energía

Fuente: E.G. Pita, Principios y Sistemas de Refrigeración. 1ra ed. México D.F: LIMUSA S.A., 1991 - Pág. 28

2.1.1.4 Calor y temperatura.

El calor se define como la forma de energía que es transferida de un cuerpo a otro debido a una diferencia de temperatura.

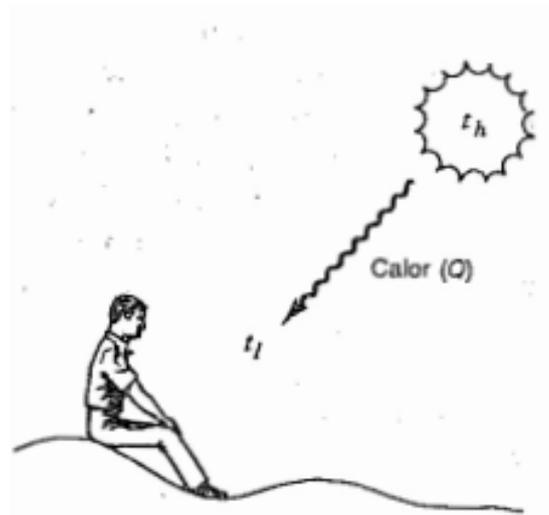


Figura 7 - Flujo de calor entre dos cuerpos

Fuente: E.G. Pita, *Principios y Sistemas de Refrigeración*. 1ra ed. México D.F: LIMUSA S.A., 1991 - Pág. 28

El calor fluye siempre desde un cuerpo cuya temperatura es más alta hacia un cuerpo que tiene una temperatura más baja.

Se puede observar que el calor sólo puede fluir en forma natural de una temperatura más alta a una temperatura más baja. Si no hay diferencia de temperatura, no existe flujo de calor.

La refrigeración es un caso especial en la transferencia de calor, es la transferencia o remoción del calor de un cuerpo para llevarlo a una temperatura deseada más baja. Es por esto que debe existir un cuerpo con temperatura más baja, para que exista la transferencia de calor. Esta condición es la que origina la necesidad de métodos de refrigeración.

2.1.1.5 Primera Ley Termodinámica, ecuación de la energía.

La termodinámica es la rama de la física que trata de la transformación entre el calor y el trabajo. La primera ley de la termodinámica es un principio que trata sobre la conservación de la energía: la energía agregada a un sistema menos la energía removida del sistema es igual al cambio de energía.

$$\text{cambio de energía} = \text{energía agregada} - \text{energía removida}$$

2.1.1.6 Líquidos, vapores y cambio de estado de los mismos.

Las sustancias pueden existir en tres diferentes estados (fases): sólido, líquido o vapor (gas). Los fenómenos que tienen lugar cuando una sustancia cambia de estado, se describen en la Figura 8.

En a, tenemos la condición inicial, un líquido con agua a temperatura ambiente. Puesto que no está tapado, está sujeto a la presión atmosférica.

En b, se agrega calor al agua y se observa que la temperatura del agua aumenta continuamente a medida que se agrega calor.

En c, la temperatura del agua llega a un punto, en el cual no sigue aumentando,

En d, a pesar que se sigue agregando más calor, la temperatura no sube durante un tiempo. Y se observa que el agua cambia de estado líquido a gaseoso. A este fenómeno denominamos ebullición.

En e, toda el agua se ha evaporado. Si se agrega más calor se observará que la temperatura del vapor empezará a aumentar nuevamente, como se ve en f.

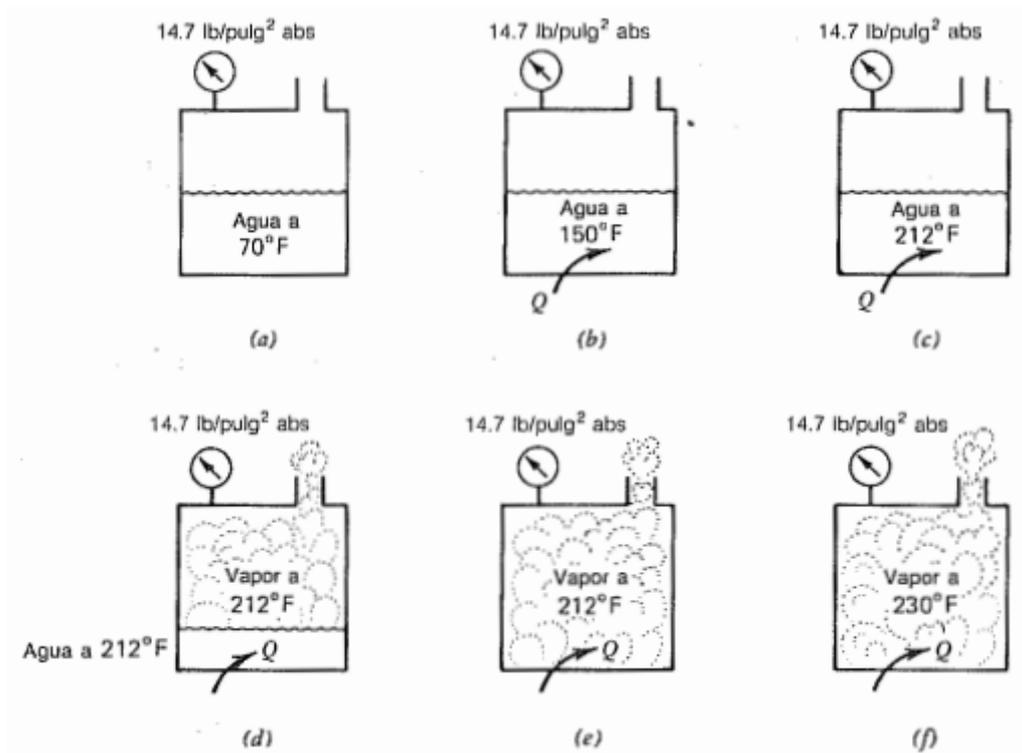


Figura 8 - Cambios de estado del agua

Fuente: E.G. Pita, Principios y Sistemas de Refrigeración. 1ra ed. México D.F: LIMUSA S.A., 1991 - Pág. 37

2.1.1.7 Refrigeración por evaporación.

Cuando una sustancia se encuentra en estado líquido, ocurre que si la presión circundante hace disminuir súbitamente a un valor inferior al de su presión de saturación, el líquido empezará a hervir para pasar al estado gaseoso. La velocidad de las moléculas es suficientemente alta para permitirles que escapen rápidamente a la presión más baja. La ebullición enfriará la sustancia a la temperatura de saturación correspondiente a la presión más baja. Cuando el líquido hierve, absorben su correspondiente calor latente de vaporización de cualquier cuerpo circundante, enfriándolo.

2.1.1.8 Ley de los gases perfectos.

En ciertas condiciones, la presión el volumen y la temperatura de los gases están relacionado por una ecuación llamada, ecuación de los gases perfectos o ideales. Se puede expresar de la siguiente manera:

$$p \times V = m \times R \times T$$

Al reordenar los términos de la ecuación para dos condiciones diferentes de gas 1 y 2, tenemos la siguiente ecuación:

$$\frac{p_2 \times V_2}{T_2} = \frac{p_1 \times V_1}{T_1}$$

2.1.1.9 Usos de la refrigeración.

Existen cuatro clasificaciones principales de los equipos de refrigeración existentes: domestico, comercial, de aire acondicionado e industrial. La primera es utilizada para la preparación y conservación de los alimentos, fabricación de hielo y para enfriar bebidas en el hogar. La refrigeración comercial, es utilizada en tiendas de venta, locales de comida e instituciones, con el mismo propósito que el doméstico. La refrigeración de aire acondicionado se utiliza para el confort

de las personas y también en la industria para crear temperatura, humedad y limpieza de aire necesaria en los procesos de fabricación. Por último, la refrigeración industrial es necesaria en la industria alimentaria para el procesamiento, preparación y preservación a gran escala. Como aplicaciones principales tenemos las plantas de enfriamiento y congelación de alimentos, cámaras frigoríficas, cervecerías y productos lácteos. Sobre ésta última clasificación desarrollé éste proyecto.

2.1.1.10 Método de refrigeración.

La refrigeración se define como la remoción de calor de una sustancia para llevarla o mantenerla a una temperatura convenientemente baja o inferior a la temperatura ambiente. El método más utilizado para producir la refrigeración mecánica se conoce como el sistema de compresión de vapor. En este sistema, un líquido refrigerante volátil se evapora en un evaporador, este proceso da como resultado una remoción de calor (enfriamiento) de la sustancia que se debe enfriar. Se requiere un compresor y un condensador a fin de mantener el proceso de refrigeración y recuperar el refrigerante para su reutilización.

2.1.1.11 Equipo de refrigeración.

Los principales componentes del equipo son el evaporador, el compresor y el condensador. El equipo puede estar constituido por componentes separados o puede ser del tipo integral.

2.1.1.12 Refrigeración por vaporización de un líquido.

Tenemos la siguiente situación: un tanque lleno de líquido refrigerante R-12, a presión relativamente alta 200 lb/pulg^2 y a una temperatura ambiente 80°F . El refrigerante debe estar en estado líquido. La salida del tanque se conecta en A mediante una válvula, a una tubería B-C. La presión circundante es la atmosférica. Cuando se abre la válvula, el refrigerante fluye a través de la tubería, debido a que la presión en el interior del tanque es mayor. La disposición del sistema provee circulación del fluido por la parte exterior de la tubería. Este fluido es la sustancia que se debe enfriar.

2.1.1.13 Proceso de Expansión.

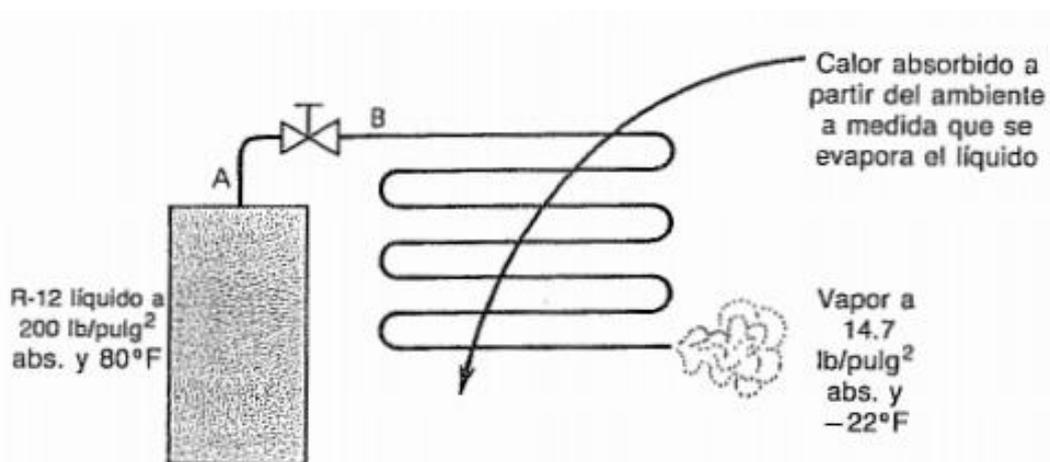


Figura 9 - Refrigeración obtenida por la vaporización de un líquido

Fuente: E.G. Pita, *Principios y Sistemas de Refrigeración*. 1ra ed. México D.F: LIMUSA S.A., 1991 - Pág. 54

Tan pronto como el líquido refrigerante fluye desde el tanque a través de la abertura pequeña de la válvula, su presión disminuye inmediata y bruscamente en B a aproximadamente la presión a la que está expuesto el tubo. Ésta presión es mucho más baja que la presión de saturación del líquido R-12. Esto provocará que el líquido refrigerante empezará a hervir y se convertirá en un gas.

Durante este proceso, se absorbe el calor. Este calor se obtiene del propio refrigerante, que está a una temperatura relativamente elevada al entrar a la válvula. Esto da por resultado que el refrigerante se enfríe a la temperatura de saturación correspondiente a la presión inferior, al salir de la válvula. Así es como se obtiene la baja temperatura necesaria para la refrigeración.

2.1.1.14 Proceso de Evaporación.

La refrigeración realmente útil tiene lugar en la tubería B-C. El fluido circundante que pasa por el exterior del tubo está a una temperatura más elevada que la del refrigerante. Por consiguiente, el calor pasa del fluido al refrigerante a través de las paredes del tubo, enfriándose así el fluido. De esta manera se obtiene la refrigeración deseada. El calor que adquiere el refrigerante líquido hace que éste se vapore al pasar por la tubería. Esta tubería se conoce con el nombre de evaporador.

2.1.1.15 Recuperación del refrigerante.

El sistema anteriormente descrito, es un sistema ineficiente, ya que el líquido refrigerante al llegar al ambiente, no se puede recuperar. Es por eso que debemos recuperar de alguna forma el líquido refrigerante para continuar su reutilización. Para ello se hace uso de dos elementos el compresor y el condensador. La temperatura de saturación (condensación) de los fluidos aumenta con el aumento de la presión. Por consiguiente, debemos elevar la presión y la temperatura del gas hasta un nivel suficientemente elevado, para luego remover el calor del mismo, utilizando un medio de enfriamiento como el aire o agua a una temperatura del medio ambiente natural. Se utiliza un compresor para elevar la presión del refrigerante en estado gaseoso. El gas a alta presión pasa entonces al condensador. El condensador se utiliza para remover el calor del refrigerante, lo cual hace que se condense. El refrigerante se halla entonces en su condición original, listo para su reutilización.

2.1.1.16 El sistema de refrigeración por compresión de vapor.

Los cuatro componentes básicos del sistema son: dispositivo de expansión, evaporador, compresor y condensador.

2.1.1.17 Proceso de Expansión (A-B).

El líquido refrigerante entra al dispositivo de expansión en el punto A. Existen diversos dispositivos de expansión: la válvula de expansión y el tubo capilar. En cualquiera de los casos, el dispositivo de control de flujo tiene una abertura estrecha, lo que da por resultado gran pérdida de presión al fluir el refrigerante a través del mismo. El refrigerante sale por el punto B a una presión menor que en A. Debido a este fenómeno una parte del refrigerante se vaporiza inmediatamente y pasa a estado gaseoso. La porción de líquido que se vaporiza, toma el calor necesario para su evaporación, enfriándola de esta manera. El refrigerante sale de la válvula como una mezcla de líquido y vapor en estado saturado y con una temperatura menor.

2.1.1.18 Proceso Evaporación (B-C).

El refrigerante fluye a través de la tubería del evaporador, de B a C. El aire que se debe enfriar, fluye por el exterior de los tubos. Se halla a una temperatura más elevada que la del refrigerante dentro del evaporador. Por consiguiente, el calor fluye desde el aire hacia el refrigerante, a través de las paredes del tubo.

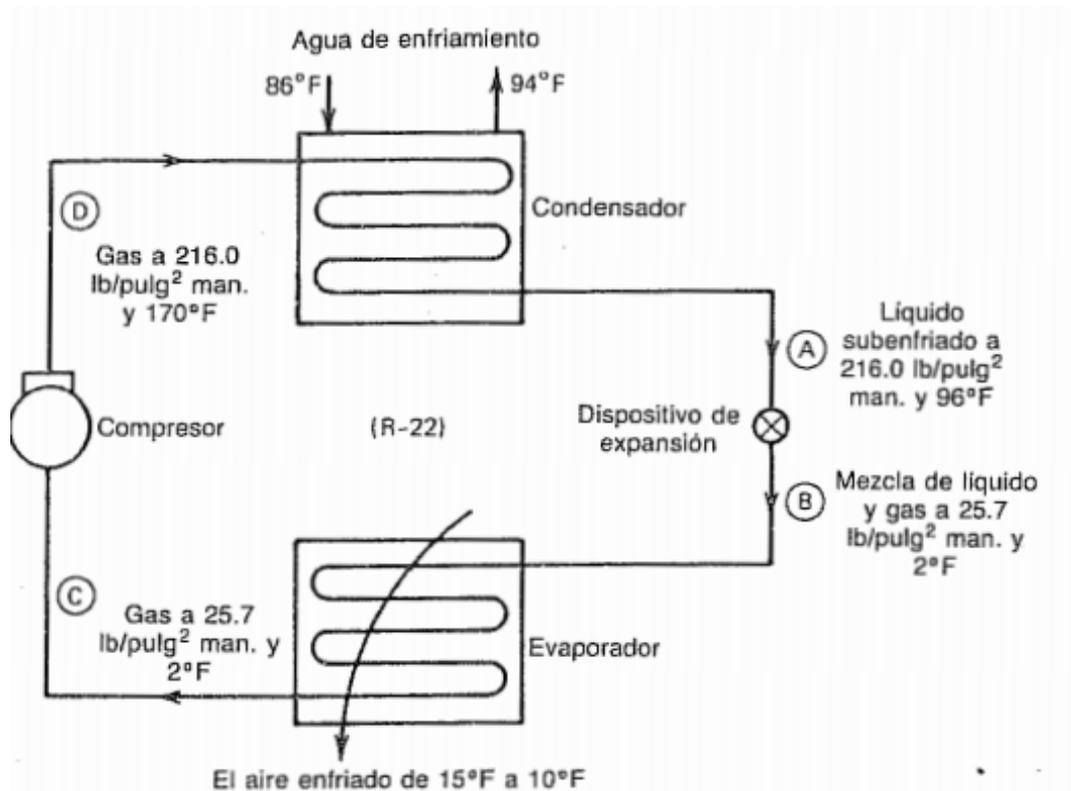


Figura 10 - Sistema Básico de refrigeración por compresión de vapor

Fuente: E.G. Pita, *Principios y Sistemas de Refrigeración*. 1ra ed. México D.F: LIMUSA S.A., 1991 - Pág. 56

2.1.1.19 Proceso de Compresión (C-D).

El compresor hace entrar al vapor por el lado de la succión, para luego comprimirlo a una presión elevada, adecuada para efectuar la condensación. Esta presión es a aproximadamente igual a la cual entro al dispositivo de control de flujo. Se requiere trabajo para comprimir el gas, este trabajo procede de un motor. Este trabajo contribuye a aumentar la energía almacenada del vapor comprimido, resultando un aumento de su temperatura.

2.1.1.20 Proceso de Condensación (D-A).

El gas a alta presión que descarga el compresor, fluye a través de la tubería del condensador. El aire, generalmente es el fluido que fluye por el exterior de la tubería. En el ejemplo de la Figura 10, se utiliza agua. Como el refrigerante está sobrecalentado cuando entra al condensador, primero se enfría hasta que alcanza su temperatura de saturación. La remoción adicional de calor resulta en la condensación gradual del refrigerante, hasta que se licúa en su totalidad. Luego el ciclo vuelve a repetirse con la expansión.

2.1.2 Transductores y Sensores.

Se denomina transductor, a todo dispositivo que convierte una señal de una forma física en una señal correspondiente pero de otra forma física distinta. Es decir es un dispositivo que convierte un tipo de energía en otro.

Existen seis tipos de señales: mecánicas, térmicas, magnéticas, eléctricas, ópticas, y moleculares (químicas). Cualquier dispositivo que convierta una señal de un tipo en una señal de otro tipo debería considerarse un transductor.

En la práctica se consideran transductores a aquellos que ofrecen una señal de salida eléctrica.

Un sensor es un dispositivo que a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida.

Los sensores basados en la variación de la resistencia eléctrica son los más abundantes.

2.1.2.1 Sensores de Temperatura.

Los detectores de temperatura basados en la variación de una resistencia eléctrica, se suelen designar como RTD (Detector de Temperatura Resistivo). Dado que el material empleado con mayor frecuencia para esta finalidad es el platino, se habla de PRT (Termómetro de Resistencia de Platino).

El fundamento de las RTD es la variación de la resistencia de un conductor con la temperatura. En un conductor, el número de electrones disponibles para la conducción no cambia apreciablemente con la temperatura. Pero si ésta aumenta, las vibraciones de los átomos alrededor de sus posiciones de equilibrio son mayores, y así dispersan más eficazmente a los electrones, reduciendo su velocidad media. Esto implica un coeficiente de temperatura positivo, un aumento de la resistencia con la temperatura.

$$R = R_0(1 + \alpha_1 T)$$

Donde R_0 es la resistencia a la temperatura de referencia y T el incremento de temperatura respecto a la de referencia.

Los sensores de platino de película fina son mucho más económicos que los de hilo bobinado y sus prestaciones son casi de igual calidad. Se han aplicado en múltiples casos industriales, en automóviles, electrodomésticos y edificios.

2.1.2.2 Termistores.

Son resistores variables con la temperatura, pero no están basados en conductores, sino en semiconductores. Si su coeficiente de temperatura es negativo, se denominan NTC, mientras que si es positivo se denominan PTC. Su simbología es la siguiente:



Figura 11 - PTC y NTC

El fundamento de los termistores está en la dependencia de la resistencia de los semiconductores con la temperatura. Al aumentar la temperatura lo hace también el número de portadores reduciéndose la resistencia, y de ahí que presenten coeficiente de temperatura negativo (NTC). Esta dependencia varía con la presencia de impurezas, si el dopado es muy intenso, el semiconductor adquiere propiedades metálicas con coeficiente de temperatura positivo (PTC) en un margen de temperaturas limitado.

Para las NTC la fórmula es la siguiente:

$$R_T = R_0 \left[B \left(\frac{1}{T} - \frac{1}{T_0} \right) \right]$$

Donde R_0 es la resistencia a 25 °C u otra temperatura de referencia, y T_0 es la temperatura expresada en Kelvin. El parámetro B es la temperatura característica del material (va de 2000K a 5000K)

2.1.2.3 Transformadores.

Los transformadores son dispositivos diseñados para transferir energía eléctrica de un circuito a otro. Logran esta transferencia usando un campo magnético que intersecta a ambos circuitos. Son capaces de entregar un distinto valor de corriente o voltaje alterno en sus terminales de salida respecto a los valores aplicados a sus terminales de entrada.

El transformador funciona empleando el fenómeno eléctrico de la inductancia mutua. Esta inductancia mutua es el efecto que se presenta cuando el campo magnético de un elemento también influye en otros elementos cercanos. El resultado de ese acoplamiento magnético es que se inducen corrientes y voltajes en los elementos cercanos.

Consiste en dos bobinas (llamadas primario y secundario) enrolladas alrededor de un núcleo común de material magnético.

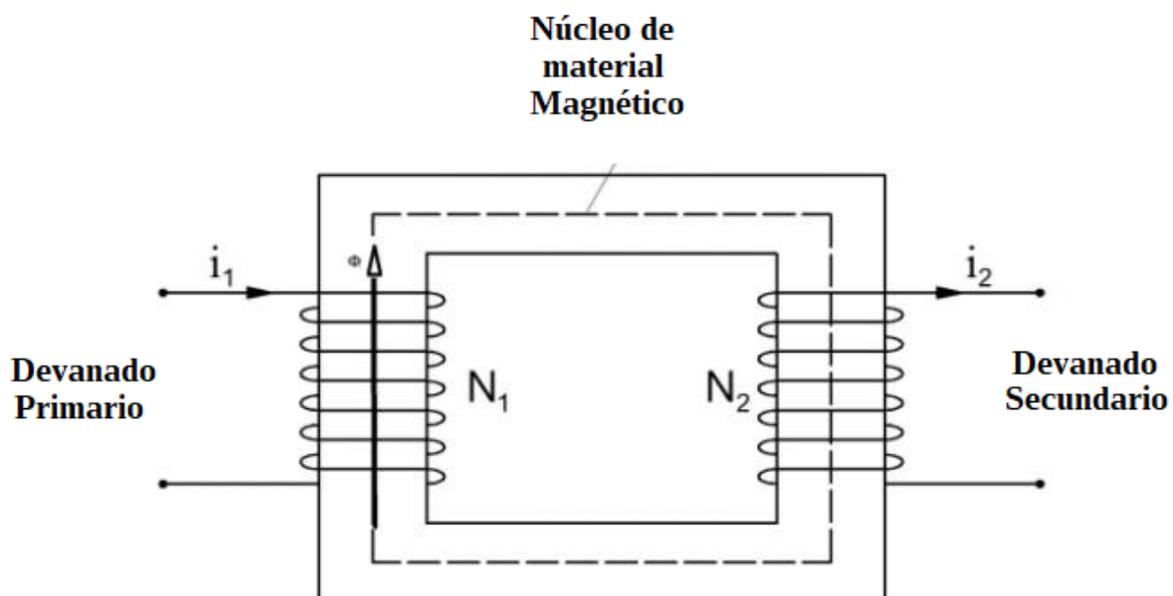


Figura 12 - Diagrama de un transformador

Fuente: <https://media.cheggcdn.com/study/aaf/aaf4bd21-e32d-4cc7-b175-bec473a7b2af/image> (editada)

Si hacemos circular una corriente en el devanado primario, origina un campo magnético que está restringido principalmente al núcleo magnético alrededor del cual está devanado el primario. Si en el mismo núcleo está también enrollada la bobina secundaria, el campo magnético también ligará la bobina secundaria.

Si la corriente en el primario es variable (ca) en lugar de ser fija, el campo magnético en el núcleo también cambiará. Como un campo magnético cambiante es visto por un conductor como un campo magnético en movimiento, las cargas libres en el conductor de la bobina secundaria experimentan una fuerza. Como se pueden mover estas cargas libres se trasladarán bajo la influencia de la fuerza y se producirá una corriente. De este modo, una corriente cambiante en el primario originará un flujo de corriente en el secundario del transformador.

El flujo de corriente que se induce en el secundario también tiene un voltaje asociado con él. La ley de Faraday establece que la magnitud del voltaje inducido por la variación de un flujo magnético en una bobina de N vueltas está dada por:

$$v = NK \frac{d\phi}{dt} = M \frac{di}{dt}$$

En el transformador ideal, todo el flujo magnético creado por la bobina primaria también liga la bobina secundaria. Entonces el voltaje en las terminales del secundario depende de la velocidad de cambio de la corriente en el primario.

$$V_2 = M \frac{di_1}{dt}$$

La relación de número de vueltas del primario al número de vueltas del secundario se define como:

$$\text{relación de vueltas} = \frac{N_p}{N_s}$$

La relación del voltaje a través del primario al voltaje a través del secundario es igual a la relación de vueltas.

$$\frac{V_p}{V_s} = \frac{N_p}{N_s}$$

2.1.2.4 Acondicionamiento de señal.

Los acondicionadores de señal, adaptadores o amplificadores, son los elementos del sistema de medida que ofrecen, a partir de la señal de salida de un sensor electrónico, una señal apta para ser presentada, registrada o que permita un procesamiento posterior mediante un equipo o instrumento estándar. Consisten normalmente en circuitos electrónicos que ofrecen las siguientes funciones: amplificación, filtrado, adaptación de impedancias, modulación y/o demodulación.

Si una de las etapas de tratamiento de la señal de medida es digital, si la salida del sensor es analógica, hará falta un convertidor A/D.

2.1.3 Internet de las Cosas (IoT).

La Cuarta Revolución Industrial, o Industria 4.0, es el cuarto gran cambio industrial desde la Revolución Industrial del siglo XVIII. Le preceden otras tres: la primera tuvo que ver con el carbón y el vapor; la segunda evolucionó con la electricidad y el automóvil; la tercera siguió con la explosión informática llamada Internet de las Cosas (IoT). El IoT es la base de la Industria 4.0 y utiliza una red de dispositivos interconectados para suministrar datos a través de Internet. El IoT se utiliza en una amplia gama de sectores: salud, fabricación, automóvil, comercio minorista y automatización de edificios, entre otros.

El valor de la IoT se reconoce en la captación de grandes datos utilizados para tomar decisiones y mejorar la eficiencia operativa. Con el ritmo de la rápida adopción de IoT, la demanda de desarrolladores cualificados de IoT aumenta al mismo tiempo, ya que IoT transforma nuestra forma de vivir y trabajar.

2.1.3.1 Áreas de desarrollo más fuertes de IoT.

- Hardware.

El núcleo de la IoT son los dispositivos conectados. Entre ellos se encuentran los dispositivos de tipo digital, que se conectan con otros dispositivos (máquina a máquina), y los dispositivos de tipo físico, que tienen circuitos integrados en un sensor, que transmite a un destino específico: un cuadro de mandos, una plataforma de monitorización remota u otro punto central de recepción de datos. Los datos incluyen imágenes, texto y otros parámetros, que sirven para proporcionar información.

El diseño de dispositivos incluye la capacidad de proceso y de almacenamiento. Los dispositivos pueden utilizar un microcontrolador o un Sistema en un Chip (SoC) más detallado, que combina más componentes como una CPU y dispositivos de E/S con un circuito integrado.

- Conectividad.

La conectividad es otro aspecto clave de la IoT, que permite que los dispositivos se comuniquen con otros dispositivos, así como con aplicaciones y servicios que se ejecutan en la nube. El diseño y la gestión de la red son habilidades esenciales dentro de la IO, debido al gran volumen de dispositivos conectados y al impacto que las decisiones de diseño de la red pueden tener a escala.

Además del diseño de la red, los desarrolladores deben tener un conocimiento práctico de los estándares, protocolos y tecnologías de red. Entre ellas se encuentran el WiFi, el Bluetooth de baja energía, el Zigbee, el celular, las tecnologías RFID que se utilizan en aplicaciones de consumo y las tecnologías de redes de área amplia de baja potencia (LPWAN) como LoRa. La LPWAN también incluye SigFox y NB-IoT (IoT de banda estrecha) que ofrecen una conectividad inalámbrica de bajo coste y bajo consumo de energía de largo alcance, que se adaptan mejor a las aplicaciones de IoT industriales y a gran escala.

- Diseño de aplicaciones.

Las aplicaciones web y móviles proporcionan interfaces de usuario para interactuar y consumir datos de los dispositivos IoT. Sin embargo, los dispositivos IoT pueden tener sus propias interfaces de

usuario (UI). Las interfaces basadas en la voz y en los gestos están ganando adeptos en el ámbito de la IoT, sobre todo para la automatización del hogar, mientras que las interfaces de realidad aumentada ofrecen interesantes posibilidades para superponer los datos de la IoT al mundo físico. En consecuencia, las habilidades de diseño de la interfaz de usuario y la experiencia del usuario son algunas de las habilidades más populares en la IoT en este momento.

Las aplicaciones web y móviles se desarrollan utilizando lenguajes de alto nivel, con Java, Swift y Node.js entre los principales lenguajes para el desarrollo de aplicaciones de IoT.

- Seguridad.

La ciberseguridad es primordial en cualquier debate sobre el IoT. Los dispositivos IoT son bastante vulnerables a los compromisos de seguridad.

La tecnología operativa -los dispositivos IoT y sus redes- corre un gran riesgo de seguridad. Las organizaciones que utilizan IoT deberían emplear políticas distintas para la tecnología operativa, diferentes de las de otras redes dentro de la organización. Esto debería incluir simulacros, respuesta a incidentes, planes de recuperación de desastres y políticas específicas para ataques de ransomware.

La seguridad debe estar incorporada en cada paso del diseño del sistema, no añadida como una idea a posteriori. Las cuestiones críticas que están estrechamente relacionadas con la seguridad incluyen la ética de los datos, la privacidad y la responsabilidad.

Las consideraciones de seguridad del IoT deben incluir, como mínimo

- Acceso a los puntos finales
- Encriptación de datos cuando sea necesario
- Autenticación adecuada

A medida que millones de nuevos dispositivos entran en el panorama de IoT cada día, las puertas de entrada de los ataques aumentan. Los dispositivos IoT se han utilizado para lanzar ataques de denegación de servicio distribuido (DDoS), así como otros ataques graves y dañinos.

- Datos e inteligencia artificial (IA).

La Inteligencia Artificial (IA) se ha convertido en una parte intrínseca de las redes de IoT. Esto se debe a la proliferación de datos, a la aplicación de algoritmos mejores y adaptados a dichos datos, y a la mejora de la potencia y las capacidades de almacenamiento de los dispositivos.

La analítica inteligente de big data implica la aplicación de técnicas de computación cognitiva extraídas de la minería de datos, el modelado, la estadística, el aprendizaje automático y la IA. Estas técnicas pueden aplicarse en tiempo real a los flujos de datos de los sensores para realizar análisis predictivos o para tomar decisiones de forma autónoma en respuesta a los datos entrantes, y también pueden aplicarse a los datos históricos para identificar patrones o anomalías en los datos.

Los desarrolladores de IoT, ahora y en el futuro, requerirán mayores habilidades de aprendizaje automático e IA.

2.1.3.2 Características de los dispositivos IoT.

A medida que se desarrolle el panorama del IoT, se introducirán nuevos dispositivos y plataformas. Hay características clave que son comunes a los dispositivos IoT y que proporcionan una base de comparación a la hora de seleccionar el hardware y el software para configurar una nueva red IoT o para desarrollar y ampliar una ya existente.

Los dispositivos IoT pueden caracterizarse por sus capacidades:

- Adquisición y control de datos
- Procesamiento y almacenamiento de datos
- Conectividad
- Gestión de la energía

2.1.3.3 Placas de desarrollo de microcontroladores.

Un microcontrolador es un SoC que proporciona capacidades de procesamiento y almacenamiento de datos. Los microcontroladores contienen un núcleo de procesador (o núcleos), memoria (RAM) y memoria de sólo lectura programable y borrable (EPROM) para almacenar los programas personalizados que se ejecutan en el microcontrolador. Las placas de desarrollo de microcontroladores son placas de circuito impreso con circuitos adicionales que soportan el microcontrolador para hacer más cómodo el prototipo y la programación del chip.

Los sensores y actuadores se conectan al microcontrolador a través de pines digitales o analógicos de entrada/salida de propósito general (GPIO) o a través de un bus de hardware.

2.1.3.4 Estándares y tecnologías de red.

El modelo de Interconexión de Sistemas Abiertos (OSI) es un modelo abstracto estándar de la ISO que consiste en una pila de siete capas de protocolo. De arriba abajo, son: aplicación, presentación, sesión, transporte, red, enlace de datos y física. El TCP/IP, o conjunto de protocolos de Internet. Ofrece una implementación concreta y simplificada de estas capas en el modelo OSI.

Modelo OSI	Modelo TCP/IP
7 Aplicación	Aplicación
6 Presentación	
5 Sesión	
4 Transporte	Transporte
3 Red	Red
2 Enlace de datos	Red de acceso y física
1 Física	

Tabla 3 - Modelos OSI y TCP/IP

2.1.3.5 Protocolos de red de IoT.

En la siguiente tabla se muestran algunos de los protocolos de red que se han adoptado ampliamente en el ámbito de la IoT y dónde encajan dentro de las capas TCP/IP.

Modelo TCP/IP	Modelos IoT
Aplicación	HTTPS, XMPP, CoAP, MQTT, AMQP
Transporte	UDP, TCP
Red	IPv6, 6LoWPAN, RPL
Red de acceso y física	IEEE 802.15.4 WiFi (802.11 a/b/g/n) Ethernet (802.3) GSM,CDMA, LTE

Tabla 4 - Modelos TCP/IP e IoT

Las tecnologías IoT implementadas en el desarrollo fueron las siguientes:

- Red de acceso y física: WiFi.

WiFi es una red inalámbrica estándar basada en las especificaciones IEEE 802.11a/b/g/n. 802.11n ofrece el mayor rendimiento de datos, pero a costa de un elevado consumo de energía, por lo que los dispositivos IoT podrían utilizar únicamente 802.11b o g por razones de conservación de la energía. Aunque el WiFi se adopta en muchos prototipos y dispositivos IoT de la generación actual, a medida que se generaliza la disponibilidad de soluciones de mayor alcance y menor consumo, es probable que el WiFi sea sustituido por alternativas de menor consumo.

- Red: IPv4.

En la capa de Internet, los dispositivos se identifican mediante direcciones IP. El IPv6 se utiliza normalmente para las aplicaciones de IoT sobre el direccionamiento IPv4 heredado. IPv4 está limitado a direcciones de 32 bits, que sólo proporcionan unos 4.300 millones de direcciones en total, lo que es menos que el número actual de dispositivos IoT que están conectados, mientras que IPv6 utiliza 128 bits, por lo que proporciona 2.128 direcciones (unos $3,4 \times 10^{38}$ o 340 billones de billones). En la práctica, no todos los dispositivos IoT necesitan direcciones públicas.

- Transporte: TCP.
- Aplicación: MQTT.

Message Queue Telemetry Transport (MQTT) es un protocolo de mensajería basado en la publicación/suscripción que fue diseñado para su uso en situaciones de bajo ancho de banda, particularmente para sensores y dispositivos móviles en redes poco fiables.

2.1.3.6 Plataformas IoT para agilizar el desarrollo de aplicaciones IoT.

Las plataformas de IoT proporcionan un middleware para conectar y gestionar los dispositivos de hardware y los datos que recogen con las aplicaciones móviles y web orientadas al usuario. Estas plataformas IoT suelen estar diseñadas para establecer una comunicación segura entre dispositivos, aplicaciones y servicios. También facilitan y agilizan el desarrollo y la implantación de aplicaciones IoT. Las plataformas IoT están diseñadas para ser escalables y fiables. Por ejemplo, implementan la tolerancia a fallos para que sus sistemas de IoT sigan funcionando frente a los fallos de los dispositivos, la red o los servicios en la nube. Al aprovechar los estándares y proporcionar APIs y SDKs bien documentados, las plataformas de IoT le permiten integrar dispositivos, aplicaciones y servicios heterogéneos en su sistema de IoT, proporcionándole la flexibilidad de personalizar y sustituir componentes a medida que cambian los requisitos.

2.1.4 Protocolo MQTT.

MQTT es un protocolo de transporte de mensajería de publicación/suscripción cliente-servidor. Es ligero, abierto, sencillo y está diseñado para ser fácil de implementar. Estas características lo hacen ideal para su uso en muchas situaciones, incluso en entornos limitados, como la comunicación en contextos de Máquina a Máquina (M2M) e Internet de las Cosas (IoT), donde se requiere una pequeña huella de código y/o el ancho de banda de la red es escaso.

Es un protocolo muy ligero y binario, y debido a su mínima sobrecarga de paquetes, MQTT sobresale en la transferencia de datos por cable en comparación con protocolos como HTTP. Otro aspecto importante del protocolo es que MQTT es extremadamente fácil de implementar en el lado del cliente. La facilidad de uso fue una de las principales preocupaciones en el desarrollo de MQTT y hace que sea un ajuste perfecto para los dispositivos con recursos limitados hoy en día.

Fue inventado en 1999 por Andy Stanford-Clark (IBM) y Arlen Nipper (Arcom, ahora Cirrus Link). Necesitaban un protocolo que permitiera una pérdida mínima de batería y un ancho de banda mínimo para conectar con oleoductos vía satélite. Los dos inventores especificaron varios requisitos para el futuro protocolo:

- Implementación sencilla
- Entrega de datos con calidad de servicio
- Ligereza y eficiencia del ancho de banda
- Agnóstico de datos
- Conocimiento continuo de la sesión

Estos objetivos siguen siendo el núcleo de MQTT. Sin embargo, el enfoque principal del protocolo ha cambiado de los sistemas embebidos propietarios a los casos de uso abiertos del Internet de las Cosas (IoT).

2.1.4.1 Patrón de publicación/suscripción de MQTT.

El modelo MQTT de publicación/suscripción (también conocido como pub/sub) ofrece una alternativa a la arquitectura tradicional cliente-servidor. En el modelo cliente-servidor, un cliente se comunica directamente con un endpoint. El modelo pub/sub desacopla el cliente que envía un mensaje (el publicador) del cliente o clientes que reciben los mensajes (los suscriptores). Los editores y los suscriptores nunca se ponen en contacto directamente. De hecho, ni siquiera saben que el otro existe. La conexión entre ellos es gestionada por un tercer componente (el Bróker). El trabajo del Bróker es filtrar todos los mensajes entrantes y distribuirlos correctamente a los suscriptores.

El aspecto más importante de pub/sub es la disociación entre el editor del mensaje y el receptor (suscriptor). Este desacoplamiento tiene varias dimensiones:

- Desacoplamiento espacial: El editor y el suscriptor no necesitan conocerse (por ejemplo, no hay intercambio de dirección IP y puerto).
- Desacoplamiento temporal: El editor y el suscriptor no necesitan funcionar al mismo tiempo.
- Desacoplamiento de la sincronización: No es necesario interrumpir las operaciones de ambos componentes durante la publicación o la recepción.

En resumen, el modelo MQTT pub/sub elimina la comunicación directa entre el editor del mensaje y el receptor/suscriptor. La actividad de filtrado del Bróker permite controlar qué cliente/suscriptor recibe qué mensaje. El desacoplamiento tiene tres dimensiones: espacio, tiempo y sincronización.

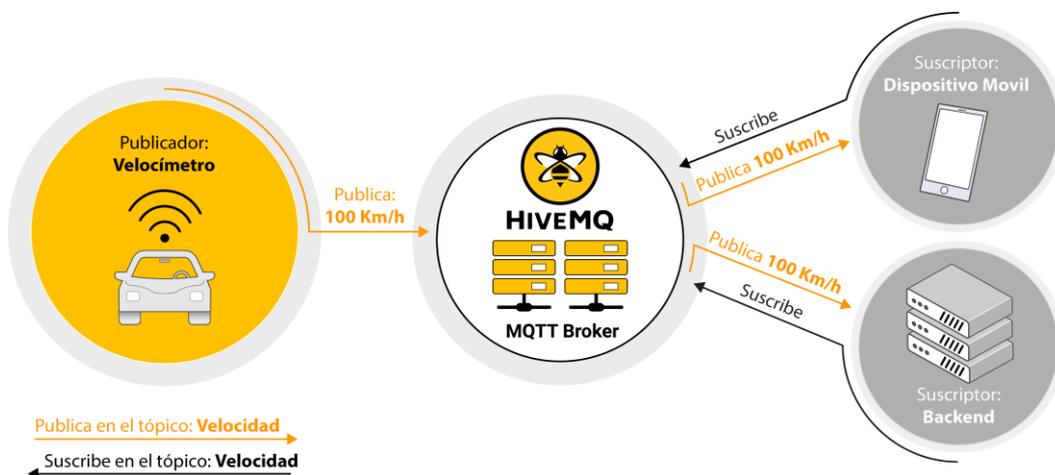


Figura 13 - Arquitectura MQTT

Fuente: <https://www.hivemq.com/img/svg/mqtt-publish-subscribe.svg> (editada)

2.1.4.2 Escalabilidad.

El protocolo MQTT se adapta mejor que el enfoque tradicional cliente-servidor. Esto se debe a que las operaciones en el Bróker pueden ser altamente paralelas y los mensajes pueden ser procesados de una manera dirigida por eventos. El almacenamiento en caché de los mensajes y el enrutamiento inteligente de los mismos suelen ser factores decisivos para mejorar la escalabilidad. Sin embargo, escalar a millones de conexiones es un reto.

2.1.4.3 Filtrado de Mensajes.

Está claro que el Bróker MQTT juega un papel fundamental en el proceso pub/sub. Pero, ¿cómo consigue el Bróker filtrar todos los mensajes para que cada suscriptor reciba sólo los mensajes que le interesan? Como verás, el Bróker tiene varias opciones de filtrado:

OPCIÓN 1: FILTRADO BASADO EN EL ASUNTO.

Este filtrado se basa en el asunto o tema que forma parte de cada mensaje. El cliente receptor se suscribe al Bróker por temas de interés. A partir de ahí, el Bróker se asegura de que el cliente receptor reciba todos los mensajes publicados en los temas suscritos. En general, los temas son cadenas con una estructura jerárquica que permite el filtrado basado en un número limitado de expresiones.

OPCIÓN 2: FILTRADO BASADO EN EL CONTENIDO.

En el filtrado basado en el contenido, el intermediario filtra el mensaje basándose en un lenguaje de filtrado de contenido específico. Los clientes receptores se suscriben a las consultas de filtrado de los mensajes que les interesan. Una desventaja importante de este método es que el contenido del mensaje debe conocerse de antemano y no puede cifrarse ni modificarse fácilmente.

OPCIÓN 3: FILTRADO BASADO EN TIPOS.

Cuando se utilizan lenguajes orientados a objetos, el filtrado basado en el tipo/clase de un mensaje (evento) es una práctica habitual. Por ejemplo, un suscriptor puede escuchar todos los mensajes de tipo Excepción o cualquier subtipo.

2.1.4.4 Principales Características.

MQTT incorpora todos los aspectos de pub/sub que hemos mencionado:

- MQTT desvincula espacialmente al publicador y al suscriptor. Para publicar o recibir mensajes, los editores y suscriptores sólo necesitan conocer el nombre de host/IP y el puerto del Bróker.
- MQTT desacopla por tiempo. Aunque la mayoría de los casos de uso de MQTT entregan los mensajes en tiempo casi real, si se desea, el Bróker puede almacenar los mensajes para los clientes que no están en línea. (Se deben cumplir dos condiciones para almacenar los mensajes: que el cliente se haya conectado con una sesión persistente y se haya suscrito a un tema con una calidad de servicio superior a 0).
- MQTT funciona de forma asíncrona. Dado que la mayoría de las bibliotecas de clientes funcionan de forma asíncrona y se basan en callbacks o un modelo similar, las tareas no se bloquean mientras se espera un mensaje o se publica un mensaje. En ciertos casos de uso, la sincronización es deseable y posible. Para esperar un determinado mensaje, algunas bibliotecas tienen APIs síncronas. Pero el flujo suele ser asíncrono.

2.1.4.5 Cliente y Bróker MQTT.

Debido a que MQTT desacopla el publicador del suscriptor, las conexiones del cliente siempre son manejadas por un Bróker. Antes de entrar en los detalles de estas conexiones, defino los conceptos de cliente y Bróker.

Cuando hablo de un cliente, casi siempre me referí a un cliente MQTT. Tanto los editores como los suscriptores son clientes MQTT. Las etiquetas de publicador y suscriptor se refieren a si el cliente está publicando mensajes o está suscrito para recibirlos (la funcionalidad de publicar y suscribir también se puede implementar en el mismo cliente MQTT). Un cliente MQTT es cualquier dispositivo (desde un microcontrolador hasta un servidor completo) que ejecuta una biblioteca MQTT y se conecta a un Bróker MQTT a través de una red. Por ejemplo, Android, Arduino, C, C++, C#, Go, iOS, Java, JavaScript y .NET. Puedes ver una lista completa en la wiki de MQTT.

El Bróker es responsable de recibir todos los mensajes, filtrarlos, determinar quién está suscrito a cada mensaje y enviar el mensaje a estos clientes suscritos. El Bróker también mantiene los datos de sesión de todos los clientes que tienen sesiones persistentes, incluyendo las suscripciones y los mensajes perdidos. Otra responsabilidad del Bróker es la autenticación y autorización de los clientes.

El Bróker es el eje central por el que deben pasar todos los mensajes. Por lo tanto, es importante que el Bróker sea altamente escalable, integrable en los sistemas backend, fácil de supervisar y (por supuesto) resistente a los fallos.

2.1.4.6 Conexión MQTT.

El protocolo MQTT se basa en TCP/IP. Tanto el cliente como el Bróker necesitan tener una pila TCP/IP.

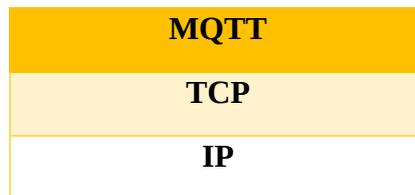


Figura 14 - Modelo MQTT

La conexión MQTT es siempre entre un cliente y el Bróker. Los clientes nunca se conectan entre sí directamente. Para iniciar una conexión, el cliente envía un mensaje CONNECT al Bróker. El Bróker responde con un mensaje CONNACK y un código de estado. Una vez establecida la conexión, el Bróker la mantiene abierta hasta que el cliente envía un comando de desconexión o la conexión se rompe.

2.1.4.7 Publicar.

Un cliente MQTT puede publicar mensajes tan pronto como se conecta a un Bróker. MQTT utiliza el filtrado basado en temas de los mensajes en el Bróker. Cada mensaje debe contener un tema que el Bróker puede utilizar para reenviar el mensaje a los clientes interesados. Normalmente, cada mensaje tiene una carga útil que contiene los datos a transmitir en formato de bytes. MQTT es independiente de los datos. El caso de uso del cliente determina cómo se estructura la carga útil. El cliente emisor (Publisher) decide si quiere enviar datos binarios, datos de texto o incluso XML o JSON completos.

En la Figura 15, PacketID identifica de forma exclusiva un mensaje mientras fluye entre el cliente y el agente. El identificador del paquete sólo es relevante para los niveles de QoS superiores a cero. La biblioteca del cliente y/o el Bróker son responsables de establecer este identificador interno de MQTT.

PUBLICACION	
contiene:	Ejemplo:
PacketID	9213
topicName	“topico/3”
qos	2
retainFlag	falso
payload	“temperatura: 19.2”
dupFlag	falso

Figura 15 - Publicación MQTT

Fuente: https://www.hivemq.com/img/blog/publish_packet.png (editada)

topicName (nombre del tópico) es una cadena simple estructurada jerárquicamente con barras inclinadas como delimitadores. Por ejemplo, "mi casa/sala de estar/temperatura".

QoS Este número indica el nivel de calidad de servicio (QoS) del mensaje. Hay tres niveles: 0, 1 y 2. El nivel de servicio determina qué tipo de garantía tiene un mensaje para llegar al destinatario previsto (cliente o Bróker).

Retain Flag Esta bandera define si el mensaje es guardado por el Bróker como el último valor bueno conocido para un tema específico. Cuando un nuevo cliente se suscribe a un tema, recibe el último mensaje retenido en ese tema.

Payload es el contenido real del mensaje. MQTT es independiente de los datos. Es posible enviar imágenes, texto en cualquier codificación, datos encriptados y prácticamente todos los datos en binario.

Bandera DUP La bandera indica que el mensaje es un duplicado y fue reenviado porque el destinatario previsto (cliente o Bróker) no acusó recibo del mensaje original.

Cuando un cliente envía un mensaje a un Bróker MQTT para su publicación, el Bróker lee el mensaje, acusa recibo del mismo (según el nivel de QoS) y lo procesa. El procesamiento por parte del Bróker incluye determinar qué clientes se han suscrito al tema y enviarles el mensaje.

2.1.4.8 Suscribir.

Publicar un mensaje no tiene sentido si nadie lo recibe. En otras palabras, si no hay clientes que se suscriban a los temas de los mensajes. Para recibir mensajes sobre temas de interés, el cliente envía un mensaje SUBSCRIBE al Bróker MQTT. Este mensaje de suscripción es muy simple (Figura 16), contiene un identificador de paquete único y una lista de suscripciones.

El PacketID identifica de forma única un mensaje mientras fluye entre el cliente y el Bróker. La biblioteca del cliente y/o el Bróker son responsables de establecer este identificador interno de MQTT.

SUSCRIPCIÓN	
contiene:	Ejemplo:
PacketID	9213
qos1	1
topic1	“topico/1”
qos2	0
topic2	“topico/2”
...	...

Figura 16 - Suscripción MQTT

Fuente: https://www.hivemq.com/img/blog/subscribe_packet.png (editada)

Lista de suscripciones. Un mensaje SUBSCRIBE puede contener múltiples suscripciones para un cliente. Cada suscripción se compone de un tema y un nivel de calidad de servicio (QoS). El tema en el mensaje de suscripción puede contener comodines que hacen posible suscribirse a un patrón de temas en lugar de un tema específico. Si hay suscripciones superpuestas para un cliente, el Bróker entrega el mensaje que tiene el mayor nivel de QoS para ese tema.

2.1.4.9 Cancelar suscripción.

La contraparte del mensaje SUBSCRIBE es el mensaje UNSUBSCRIBE. Este mensaje elimina las suscripciones existentes de un cliente en el Bróker. El mensaje UNSUBSCRIBE es similar al mensaje SUBSCRIBE y tiene un identificador de paquete y una lista de temas.

2.1.5 NodeRed y Dashboard.

Node-RED es una herramienta de programación para conectar dispositivos de hardware, API y servicios en línea de formas nuevas e interesantes.

Ofrece un editor basado en el navegador que facilita el cableado de flujos utilizando la amplia gama de nodos de la paleta que pueden desplegarse en su tiempo de ejecución con un solo clic.

2.1.5.1 Conceptos.

- Nodo. Un nodo es el bloque básico de un flujo (Figura 17).

Los nodos se activan al recibir un mensaje del nodo anterior en un flujo, o al esperar algún evento externo, como una solicitud HTTP entrante, un temporizador o un cambio de hardware GPIO. Procesan ese mensaje, o evento, y luego pueden enviar un mensaje a los siguientes nodos del flujo.

Un nodo puede tener como máximo un puerto de entrada y tantos puertos de salida como necesite.

- Nodo de configuración. Un nodo de configuración *config* es un tipo especial de nodo que contiene una configuración reutilizable que puede ser compartida por los nodos regulares de un flujo.

Por ejemplo, los nodos *mqtt in* y *mqtt out* utilizan un nodo MQTT Bróker *config* para representar una conexión compartida con un bróker MQTT.

- Flujo. Un flujo se representa como una pestaña dentro del espacio de trabajo del editor y es la principal forma de organizar los nodos.

El término "flujo" también se utiliza para describir informalmente un conjunto de nodos conectados. Así, un flujo (pestaña) puede contener múltiples flujos (conjuntos de nodos conectados).

- Contexto. El contexto es una forma de almacenar información que puede ser compartida entre nodos sin utilizar los mensajes que pasan por un flujo.

Hay tres tipos de contexto;

- Nodo - sólo visible para el nodo que estableció el valor
- Flujo - visible para todos los nodos del mismo flujo (o pestaña en el editor)
- Global - visible para todos los nodos

Por defecto, Node-RED utiliza un almacén de contextos en memoria para que los valores no se guarden en los reinicios. Puede configurarse para utilizar un almacén basado en un sistema de archivos para que los valores sean persistentes. También es posible conectar complementos de almacenamiento alternativos.

- Mensaje. Los mensajes son lo que pasa entre los nodos de un flujo. Son objetos JavaScript simples que pueden tener cualquier conjunto de propiedades. A menudo se les denomina *msg* dentro del editor.

Por convención, tienen una propiedad *payload* que contiene la información más útil.

- Subflujo. Un subflujo es una colección de nodos que se colapsan en un solo nodo en el área de trabajo.

Pueden utilizarse para reducir la complejidad visual de un flujo, o para empaquetar un grupo de nodos como un componente reutilizable utilizado en múltiples lugares.

- Cables. Los cables conectan los nodos y representan cómo pasan los mensajes a través del flujo.
- Paleta. La Paleta se encuentra a la izquierda del editor y enumera los nodos que están disponibles para usar en los flujos.

Se pueden instalar nodos adicionales en la paleta utilizando la línea de comandos o el Gestor de Paleta.

- Espacio de trabajo. El Espacio de Trabajo es el área principal donde se desarrollan los flujos arrastrando nodos de la paleta y conectándolos entre sí.

El espacio de trabajo tiene una fila de pestañas a lo largo de la parte superior; una para cada flujo y cualquier subflujo que se haya abierto.

- Barra lateral. Contiene paneles que proporcionan una serie de herramientas útiles dentro del editor. Entre ellos se incluyen paneles para ver más información y ayuda sobre un nodo, para ver el mensaje de depuración y para ver los nodos de configuración del flujo.

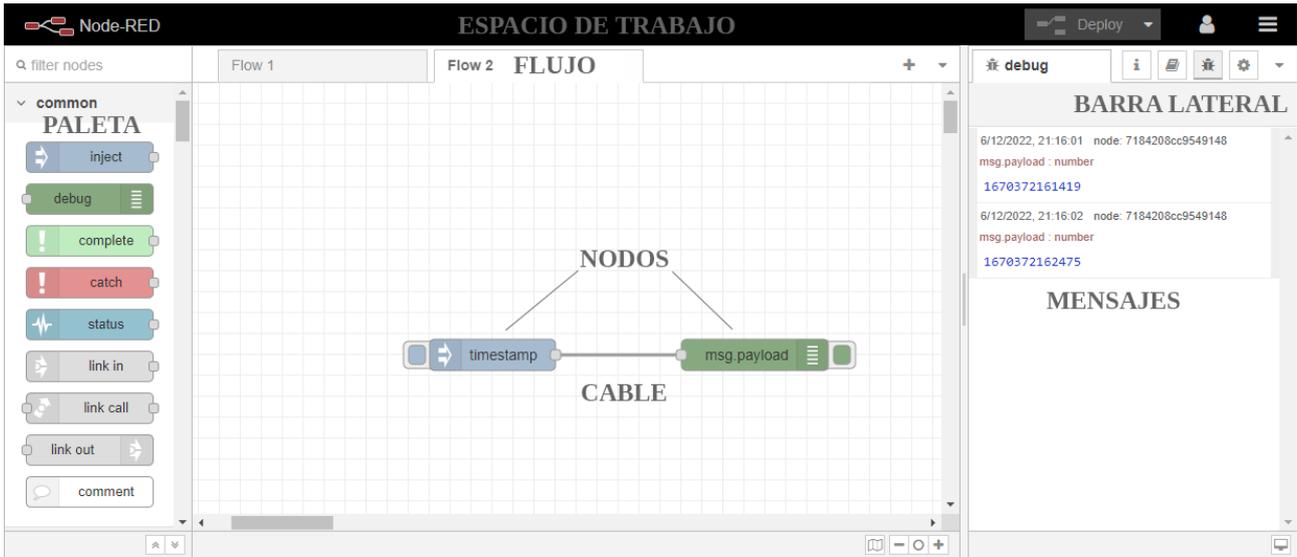


Figura 17 - Interfaz NodeRed

2.1.5.2 Dashboard.

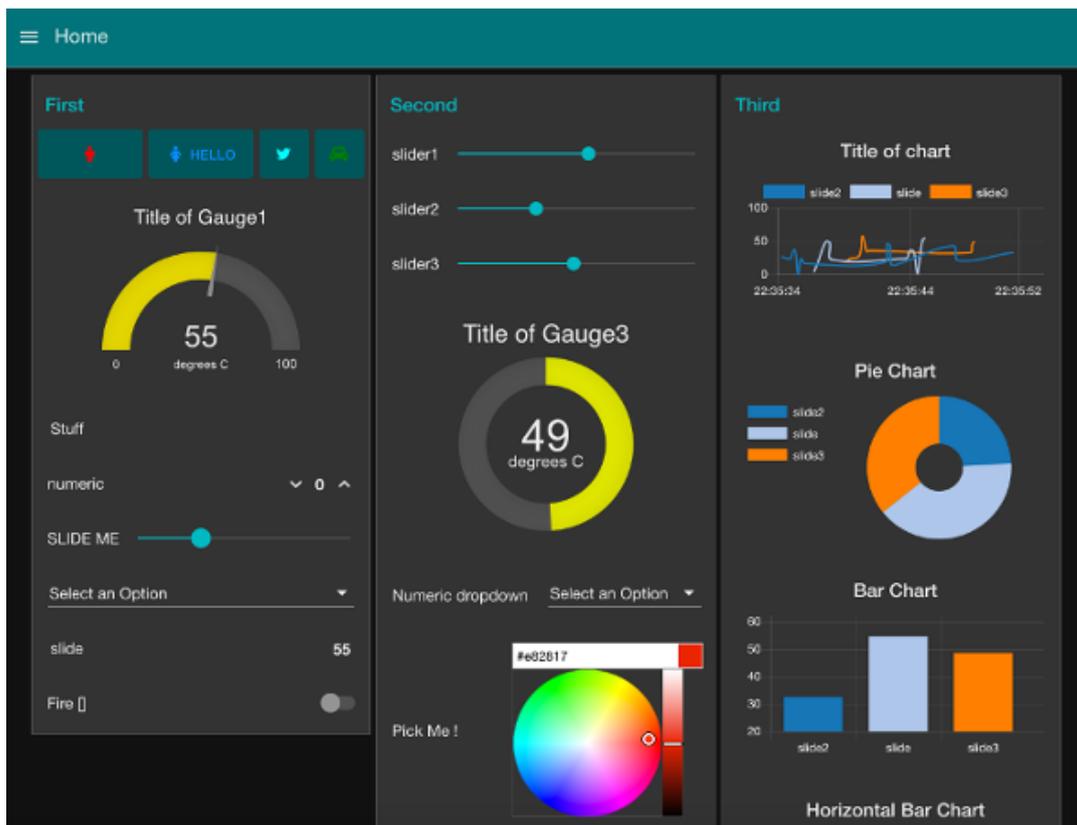


Figura 18 - Ejemplo Dashboard

Fuente: <http://nodered.org/images/Dashboarde.png>

En los sistemas informáticos empresariales, un Dashboard es un tipo de interfaz gráfica de usuario que suele ofrecer vistas rápidas de los indicadores clave de rendimiento (KPI) relevantes para un objetivo o proceso empresarial concreto. En otros usos, Dashboard es otro nombre para "informe de progreso" o "informe" y se considera una forma de visualización de datos. Al proporcionar esta visión general, los empresarios pueden ahorrar tiempo y mejorar su toma de decisiones utilizando cuadros

de mando.

El Dashboard suele ser accesible mediante un navegador web y suele estar vinculado a fuentes de datos que se actualizan periódicamente (Figura 18).

El módulo node-red-Dashboard proporciona un conjunto de nodos para crear rápidamente un panel de datos en vivo (Figura 19).

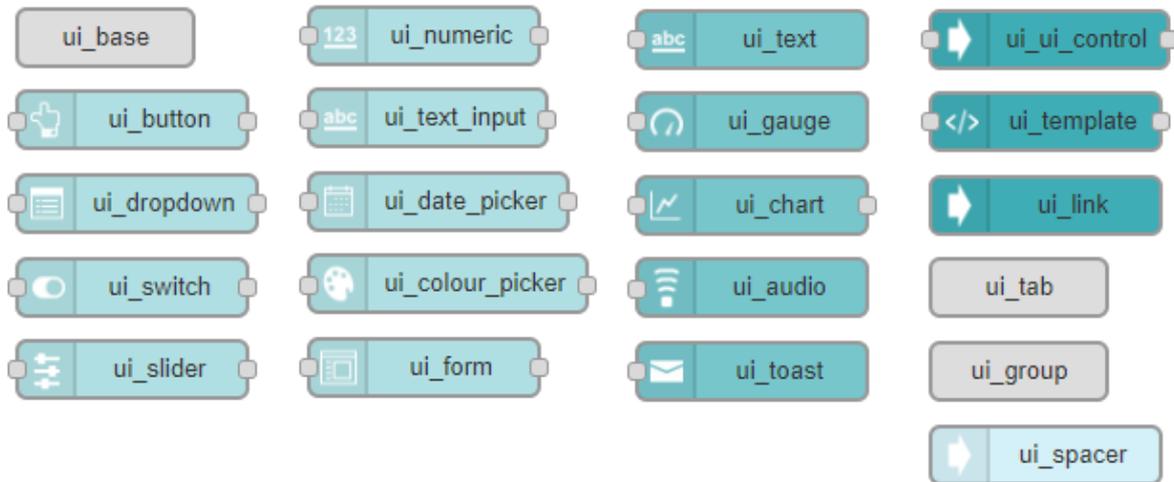


Figura 19 - Nodos disponibles en la paleta Dashboard

2.2 Desarrollo.

2.2.1 Diagrama de bloques general del proyecto.

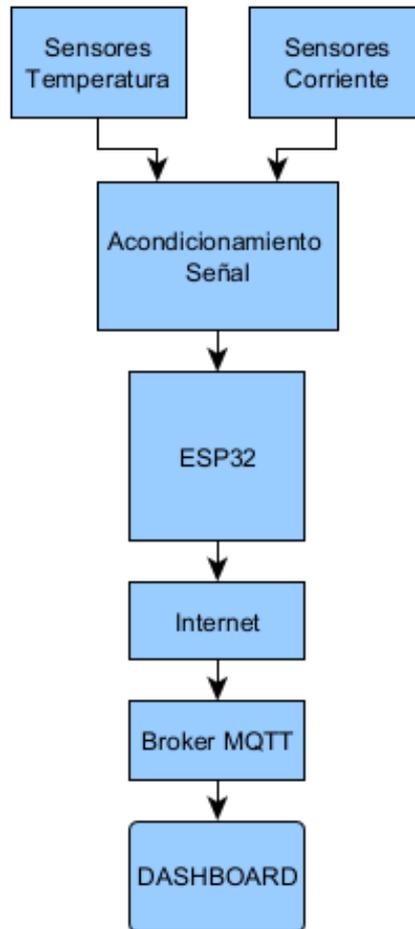


Figura 20 - Diagrama en bloques general

2.2.2 Estudio Equipo de Frío.

El diseño del proyecto final, está dimensionado con las características eléctricas de una cámara frigorífica convencional de envasados al vacío. Las características de la cámara del diseño:

Concepto	Valor Nominal Funcionamiento	Observaciones
Motor Compresor	9.9A	
Motores Evaporador	3.6A	3 motores forzadores
Motores Condensador	3.6A	3 motores forzadores
Interior Cámara	3 °C	
Exterior Cámara	Temperatura Ambiente	

Tabla 5 - Características Equipo Frío

2.2.3 Investigación de componentes.

Para la obtención de las variables físicas de temperatura y corriente, se estudiaron varios tipos de sensores.

2.2.3.1 Sensor de temperatura.

Entre los sensores de temperatura que se pueden conseguir, podemos ver la Tabla 6.

Modelo	Rango Temp (min-Max)	Resolución	Comunicación	Alimentación	Costo
TC72	-55°C a 125°C	10bit	SPI	2.65V-5.5V	\$1147
TMP125	-40°C a 125°C	10bit	SPI	2.7V-5.5V	\$1938
ADT7320	-40°C a 150°C	16bit	SPI	2.7V-5.5V	\$6033
ADT7518	-40°C a 120°C	10bit	SPI/I2C	2.7V-5.5V	\$2789
MAX6627	0°C a 125°C	12bit	SPI	3V-5.5V	\$4156
AHT10	-40°C a 85°C	9bit	I2C	2V-5.5V	\$988
DS18B20	-55°C a 125°C	12bit	1WIRE	3V-5.5V	\$910
MAX31865	-55°C a 125°C	15bit	SPI	3V-3.3V	\$8979

Tabla 6 - Opciones Sensores de Temperatura

2.2.3.2 DS18B20.

El sensor elegido por prestaciones y precio es el DS18B20. Es un sensor económico y el rango de temperatura que puede medir se adecúa a las condiciones de temperatura del sistema. Las características del mismo las podemos obtener de la hoja de datos del fabricante. A continuación enumero las más básicas.

El sensor DS18B20 es un termómetro digital de resolución configurable entre 9-12bits de resolución. Además posee funciones de alarma configurables con valores superior e inferior. El protocolo de comunicación es 1-Wire, que como dice su nombre requiere un solo cable para comunicación (y la conexión de tierra) para comunicarse con un microprocesador. En adición el sensor DS18B20 puede tomar energía directamente del terminal de datos, eliminando la existencia de una fuente externa. Una de las características principales es que cada sensor posee una dirección única de 64bits asignado desde fábrica.

Características.

- Protocolo de comunicación 1-Wire, requiere un solo pin para comunicación.
- Capacidad multipunto simplifica la distribución de aplicaciones de detección de temperatura.
- No requiere componentes externos.
- Puede tomar energía de la línea de datos. La fuente de alimentación es de 3.0V a 5.5V.

- No requiere energía de reserva.
- Las mediciones de temperatura oscilan entre los:
 - -55°C hacia los $+125^{\circ}\text{C}$. Con error $\pm 2^{\circ}\text{C}$
 - -30°C hacia los $+100^{\circ}\text{C}$. Con error $\pm 1^{\circ}\text{C}$
 - -10°C hacia los $+85^{\circ}\text{C}$. Con error $\pm 0.5^{\circ}\text{C}$
- La resolución es programable entre 9-12bits.
- La conversión con 12bits de resolución tiene una demora de 750ms
- Alarmas configurables por el usuario.

La conexión típica entre un microcontrolador y los sensores 1-Wire es la siguiente:

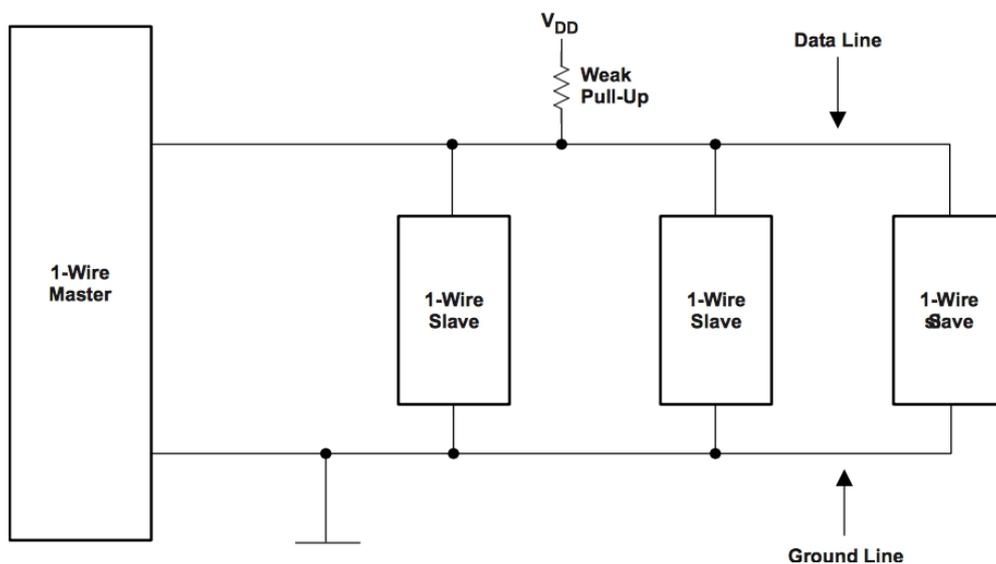


Figura 21 - DS18B20 Topología del Bus

Fuente: https://www.geekfactory.mx/wp-content/uploads/2015/04/ds18B20_one_wire_topologia_bus.png

Funcionamiento.

El esclavo envía un pulso de 480 microsegundos y luego queda en estado alto el bus, el esclavo (o esclavos) responden llevando el bus a estado bajo (pulso de presencia). Esta operación sincroniza el bus e indica a los esclavos que el maestro va a enviar información.

Escribir bit 0 - El maestro envía un pulso de estado bajo de 60 microsegundos, luego libera el bus.

Escribir bit 1 - El maestro envía un pulso en estado bajo de 6 microsegundos, luego libera el bus.

Leer bit (desde esclavo) - El maestro genera un pulso bajo de 6 microsegundos y luego libera el bus. Si el esclavo desea transmitir un "1", el esclavo "deja flotar" el bus, si desea enviar un "0", mantiene un estado lógico bajo.

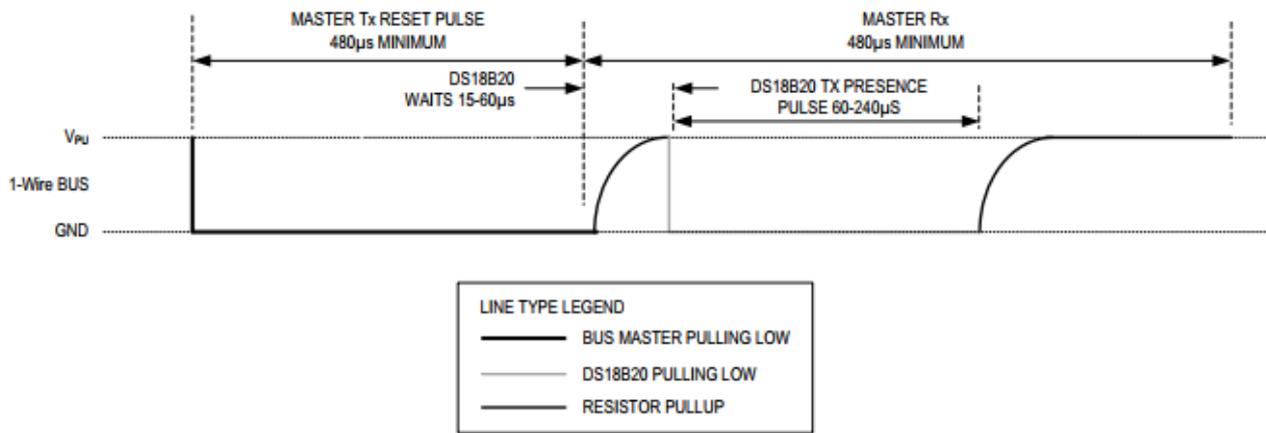


Figura 22 - DS18B20 Pulsos inicialización

Fuente: <https://pdf1.alldatasheet.es/datasheet-pdf/download/227472/DALLAS/DS18B20.html> - Pág. 15

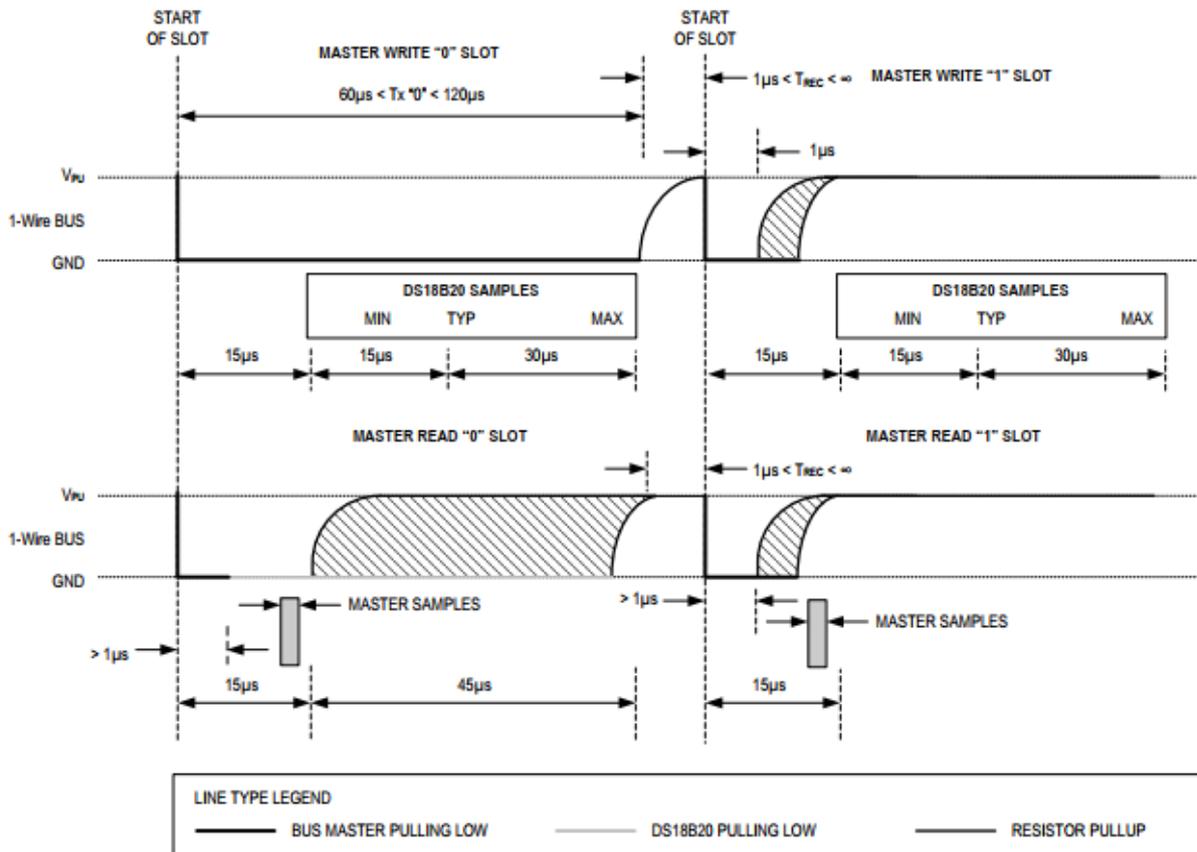


Figura 23 - DS18B20 Pulsos de Escritura y Lectura

Fuente: <https://pdf1.alldatasheet.es/datasheet-pdf/download/227472/DALLAS/DS18B20.html> - Pág. 16

Aquí, los bits de signo ‘S’ son 0 si la temperatura es positiva o 1 si esta es negativa.

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2 ³	2 ²	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	2 ⁸	2 ⁵	2 ⁴

S = SIGN

Tabla 7- DS18B20 Formato Registro de Temperatura

Fuente: <https://pdf1.alldatasheet.es/datasheet-pdf/download/227472/DALLAS/DS18B20.html> - Pág. 6

Presentaciones del sensor y pinout.



Figura 24 - DS18B20 Presentación formato sumergible

Fuente: <https://www.geekfactory.mx/wp-content/uploads/2013/06/ds18b20-sensor-de-temperatura-sumergible.jpg>

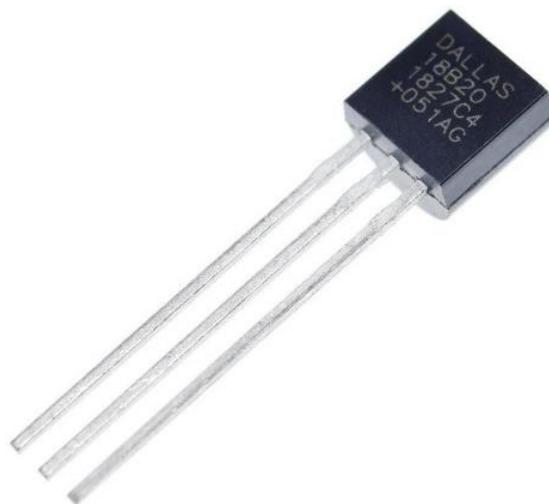


Figura 25 - DS18B20 Presentación formato TO-92

Fuente: <https://geekbotelectronics.com/wp-content/uploads/2021/06/DS18B20-Temperatura.jpg>

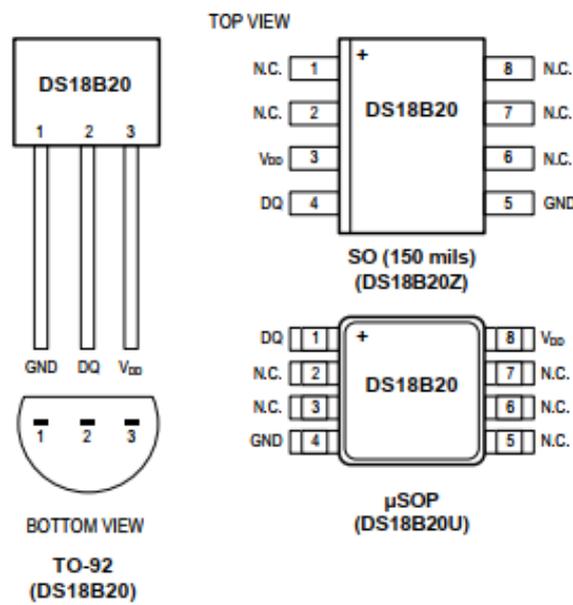


Figura 26 - DS18B20 Pines

Fuente: <https://pdf1.alldatasheet.es/datasheet-pdf/download/227472/DALLAS/DS18B20.html> - Pág. 1

Entre las principales ventajas que presenta el sensor elegido, trabaja dentro del rango de temperaturas que requiere el sistema, además es un sensor digital. Necesita solamente una resistencia de pull-up para polarizarlo (ver Tabla 8). Al ser un sensor digital, su lectura entrega directamente la temperatura en grados centígrados.

Los colores de los terminales y su función son los siguientes: GND Negro, VDD Rojo, Dato Amarillo o Blanco.

Las resistencias de Pull-Up están en función de la cantidad de metros de cable de datos.

RESISTENCIA PULL-UP	DISTANCIA DEL CABLE (METROS)
4,7 kΩ	De 0 m a 5 m
3,3 kΩ	De 5 m a 10 m
2,2 kΩ	De 10 m a 20 m
1,2 kΩ	De 20 m a 50 m

Tabla 8 - DS18B20 Resistencias PullUp

2.2.3.3 Sensor de corriente.

Los sensores de corriente comerciales que se pueden conseguir, son los siguientes:

Código	Rango (min-Max)	Resolución	Comunicación	Alimentación	Costo
LTC2947	-30A a 30A	10bit	SPI	2.7V-5.5V	\$9399
TLI4970	-25A a 25A	16bit	SPI	3.1V-3.5V	\$2733
INA219	0 a 26V	9-12bit	I2C	3V-5.5V	\$1514
LTC4151	7V a 80V	12bit	I2C	7V-80V	\$4824
INA3221	0 a 26V	13bit	I2C	2.7V-5.5V	\$2662
MAX34408	2.5V a 13.2V	8bit	I2C	2.7V-3.3V	\$3218
ACS712	-30A a 30A	-	Analógica	2.7V-5.5V	\$1014
SCT-013	5A /100A	-	Analógica	-	\$3839

Tabla 9 - Opciones Sensores Corriente

2.2.3.4 SCT-013.

Luego de analizar y estudiar cada uno de los sensores listados anteriormente, el que mejor se adecúa al diseño de este proyecto es el SCT-013. Si bien no es un sensor digital, el formato del sensor elegido permite medir corriente, sin tener que intervenir en el circuito eléctrico del equipo de frío, contrariamente al resto de los modelos estudiados. Se trata de un transformador de corriente, el cual permite medir corrientes en un rango acotado de tensión en la salida. Este valor de tensión se puede ingresar directamente en un microcontrolador con simples circuitos de adaptación de señal.



Figura 27 - Presentación SCT-013

Características.

- Ventana núcleo de 13mm x 13mm
- 1m de cable, con terminal macho plug de 3.5mm
- Salida de corriente o tensión (puede ser por resistencia interna o externa)
- Núcleo de ferrita.
- Rango de salida 0-1V

Funcionamiento.

Consiste en un transformador de corriente, el cual es similar a un transformador de tensión y está basado en los mismos principios de funcionamiento. Sin embargo tienen objetivos diferentes y, en consecuencia, están diseñados y construidos de forma distinta.

Un transformador de corriente busca generar una intensidad en el secundario que sea proporcional a la intensidad que atraviesa el primario. Para ello se desea que el primario esté formado un número de espiras reducido.

Se puede emplear esto para construir un sensor de corriente no invasivo. En un sensor se corriente el núcleo ferromagnético puede estar dividido de forma que pueda abrirse y arrollar un conductor.

De esta forma, se tiene un transformador en el que:

- El cable por el que circula la intensidad a medir constituye un devanado primario
- La “pinza” es el núcleo magnético
- El devanado secundario está integrado como parte de la sonda.

Cuando la corriente alterna circula por el conductor se genera un flujo magnético en el núcleo ferromagnético, que a su vez genera una corriente eléctrica en el devanado secundario.

La relación de transformación de intensidad depende de la relación entre el número de espiras.

El primario generalmente está formado por una única espira formada por el conductor a medir. Aunque es posible enrollar el conductor haciendo que pase más de una vez por el interior de la “pinza”. El número de espiras del secundario, integrado en la sonda, varía 1000-2000 según modelos del SCT-013.

A diferencia de los transformadores de tensión, en un transformador de intensidad el circuito secundario nunca debería estar abierto, porque las corrientes inducidas podrían llegar a dañar el componente. Por ese motivo, los sensores de SCT-130 disponen de protecciones (resistencia burden en los sensores de salida por tensión, o diodos de protección en los sensores de salida por corriente).

Pines de conexión.

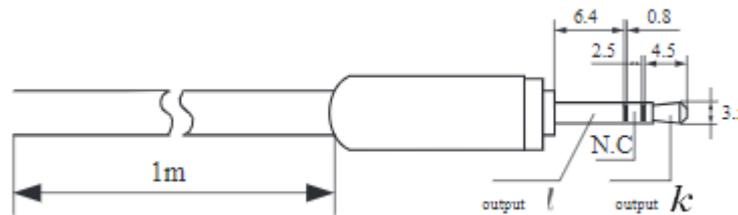


Figura 28 - SCT-013 Terminales de conexión

Fuente: <https://pdf1.alldatasheet.es/datasheet-pdf/view/1160229/YHDC/SCT013-005.html>

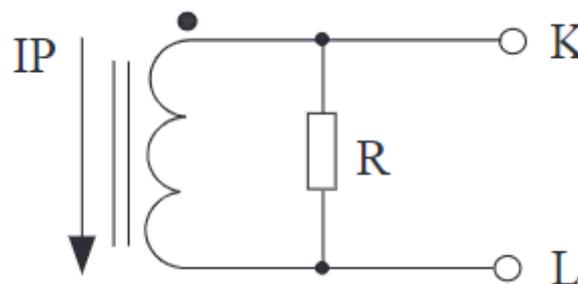


Figura 29 - SCT-013 Circuito Interno

Fuente: <https://pdf1.alldatasheet.es/datasheet-pdf/view/1160229/YHDC/SCT013-005.html>

En la Tabla 10 expongo todos los modelos que vienen comercialmente del SCT-013. El rango dinámico de salida puede ser de 0-50mA. En este caso, es necesario agregar una resistencia externa para convertir la corriente de salida en un valor de tensión para poder trabajar con un microcontrolador.

El resto de los modelos, tiene un rango dinámico de 0 a 1 V. Elegí dos modelos con este rango, uno para medir la corriente del compresor de la cámara frigorífica (Escala 10A). Y dos para medir la corriente de los motores del evaporador y condensador (Escala 5A).

Modelo	SCT-013-000	SCT-013-005	SCT-013-010	SCT-013-015	SCT-013-020
Corriente Entrada	0-100A	0-5A	0-10A	0-15A	0-20A
Salida	0-50mA	0-1V	0-1V	0-1V	0-1V
Modelo	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-000V
Corriente Entrada	0-25A	0-30A	0-50A	0-60A	0-100A
Salida	0-1V	0-1V	0-1V	0-1V	0-1V

Tabla 10 - Modelos SCT-013

La cámara elegida para el estudio y diseño del sistema de monitoreo tiene un motor en el compresor que tiene un consumo de 9.9A de corriente nominal (Tabla 5). Tanto el evaporador como el condensador tienen 3 motores de 1.2A cada uno, es decir un consumo de 3.6A de corriente nominal para el evaporador y el condensador. Los modelos elegidos para este diseño fueron el SCT-013-010 y el SCT-013-005.

2.2.3.5 ESP32.

ESP32 es la denominación de una familia de chips SoC de bajo costo y consumo de energía, con tecnología WiFi y Bluetooth de modo dual integrada. El ESP32 emplea un microprocesador Tensilica Xtensa LX6 en sus variantes de simple y doble núcleo e incluye interruptores de antena, balun de radiofrecuencia, amplificador de potencia, amplificador receptor de bajo ruido, filtros, y módulos de administración de energía. El ESP32 fue creado y desarrollado por Espressif Systems y es fabricado por TSMC utilizando su proceso de 40 nm. Es un sucesor de otro SoC, el ESP8266.

Características:

- Procesador:
 - CPU: microprocesador de 32-bit Xtensa LX6 de doble núcleo (o de un solo núcleo), operando a 160 o 240 MHz y rindiendo hasta 600 DMIPS
 - Co-procesador de ultra baja energía (ULP)
- Memoria: 520 KiB SRAM
- Conectividad inalámbrica:
 - WiFi: 802.11 b/g/n
 - Bluetooth: v4.2 BR/EDR y BLE
- Interfaces periféricas:
 - 12-bit SAR ADC de hasta 18 canales
 - 2 × 8-bit DACs
 - 10 × sensores de tacto (sensores capacitivos GPIOs)
 - 4 × SPI
 - 2 × interfaces I²S
 - 2 × interfaces I²C
 - 3 × UART
 - Controlador host SD/SDIO/CE-ATA/MMC/eMMC
 - Controlador esclavo SDIO/SPI
 - Interfaz Ethernet MAC con DMA dedicado y soporte para el protocolo IEEE 1588 Precision Time Protocol
 - Bus CAN 2.0
 - Controlador remoto infrarrojo (TX/RX, hasta 8 canales)
 - Motor PWM
 - LED PWM (hasta 16 canales)
 - Sensor de efecto Hall
 - Pre-amplificador analógico de ultra baja potencia

Node MCU ESP32.

El módulo elegido para la realización del presente proyecto, es el Node MCU ESP32, el cual es una plataforma completa de desarrollo, módulo SoC (system on chip), basado en el microcontrolador Tensilica de 32 bits, funcionando a una frecuencia de reloj de 240 Mhz, con una memoria Flash de 4MB, y 500 KB de SRAM. Tiene conectividad inalámbrica WiFi, estándar 802.11 b/g/n, y Bluetooth integrado BLE.

NodeMCU ESP32 mejora las capacidades del ESP8266, con un procesador de doble núcleo de 32 bits. Además incluye internamente una gran cantidad de periféricos para la conexión con sensores

táctiles capacitivos, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

El ESP32 tiene el protocolo TCP-IP integrado, permitiendo su acceso a internet a través de un router WiFi, con el cual puede conectarse en forma inalámbrica, gracias a su antena integrada al módulo.

NodeMCU ESP32 integra una interfaz USB-Serial CP2102 y, durante la fase de desarrollo de aplicaciones, puede operarse desde el puerto USB de la computadora con un cable micro USB, el cual permite la comunicación serial a una velocidad de 9600 hasta 921,600 bps, y proporciona además la alimentación de 5v al módulo.

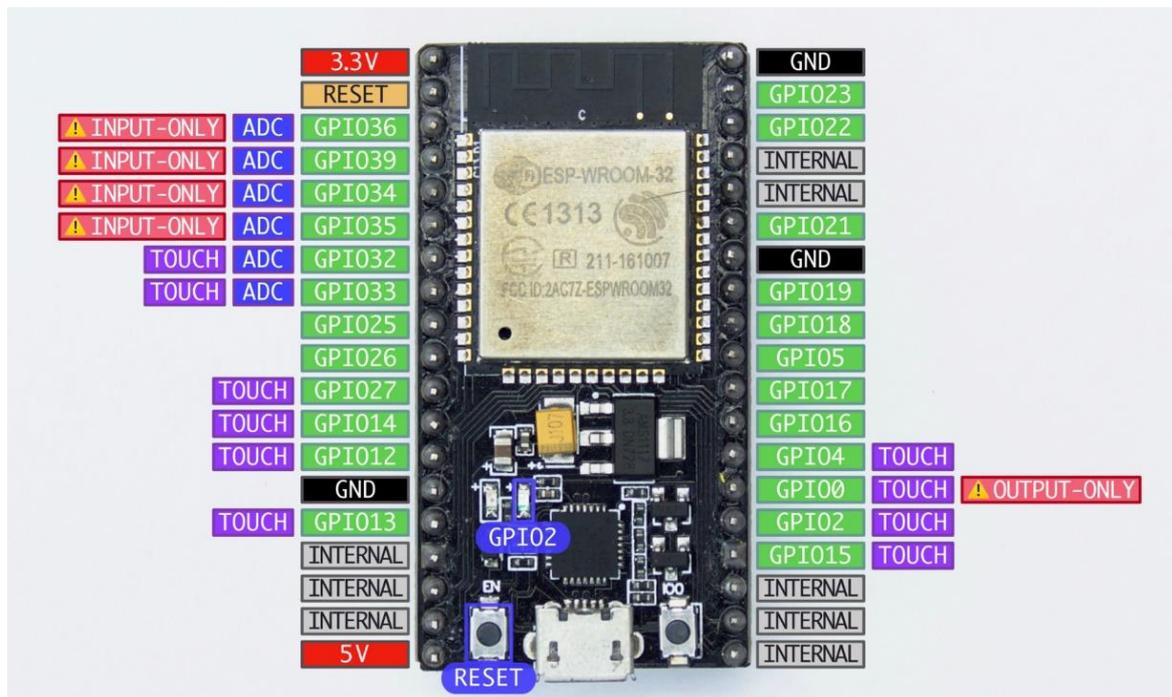


Figura 30 - PINOUT NODEMCU-ESP32

Fuente: https://esphome.io/_images/nodemcu_esp32-full.jpg

El módulo ESP32 cuenta con 30 pines, de los cuales 25 son de los llamados GPIO (General Purpose Input Output). Puede funcionar de manera totalmente autónoma como servidor web y recibir comandos remotos para operar los pines mencionados, ya sea para activar dispositivos o enviar status de sensores digitales o analógicos. Incluye también pines con las funciones PWM, 1wire, I2C, SPI, UART, y convertidores A/D de 12 bits.

Su voltaje de alimentación es de 5v, el cual es reducido a 3.3v por medio de un regulador on board. El consumo promedio es de 100 ma. La corriente aumenta durante la transmisión y recepción WiFi, de tal manera que, si se alimenta con fuente externa, se recomienda una capacidad de 300 ma. También puede alimentarse directamente a través del conector micro USB.

2.3 Propuestas de circuitos o esquemas.

Habiendo elegido los materiales para el desarrollo del presente proyecto, el circuito esquemático diseñado es el siguiente:

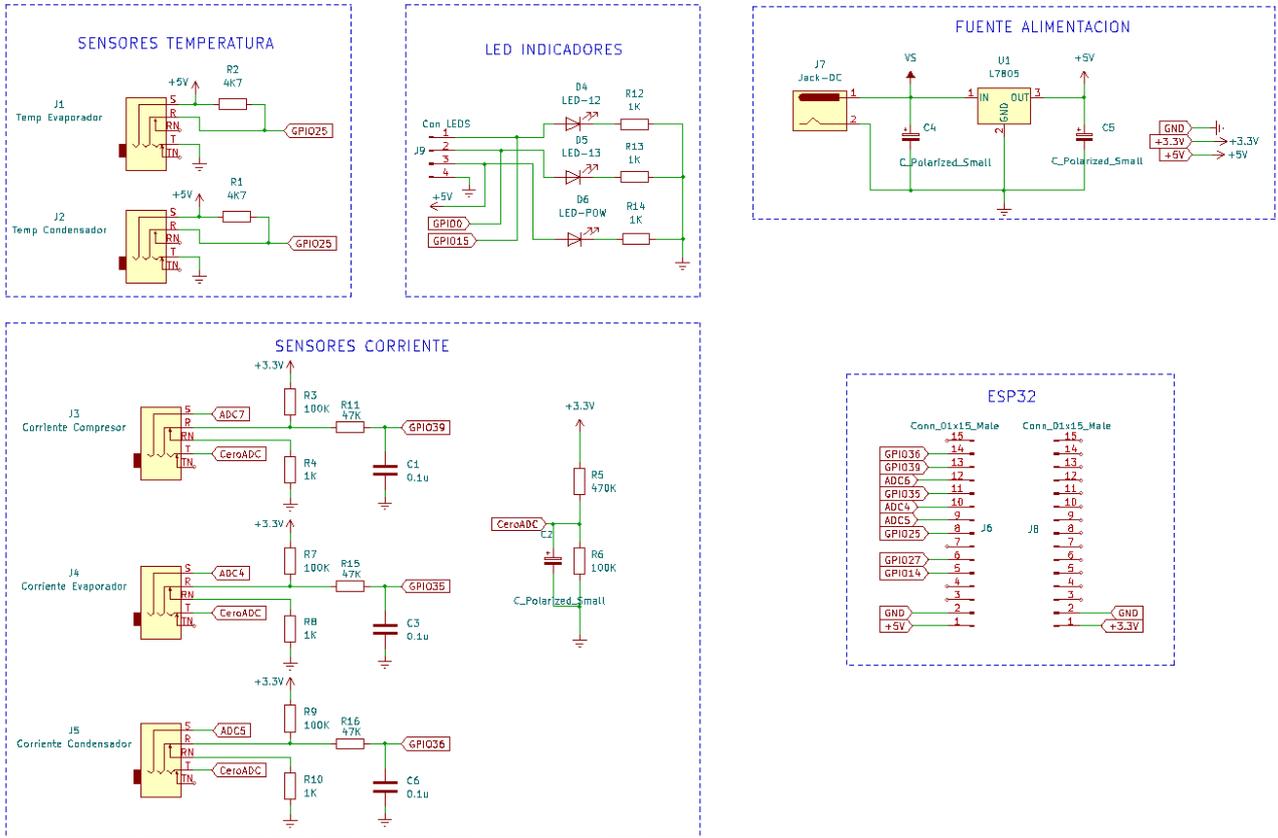


Figura 31 - Esquemático Principal

Dentro de este esquema existen cinco bloques, los cuales comprenden: la fuente de alimentación, los sensores de temperatura, los sensores de corriente, el ESP32 y por ultimo unos LEDs indicadores.

En la fuente de alimentación (Figura 32), tiene una entrada de 12V de corriente continua y un regulador 7805, el cual estabiliza el voltaje de entrada a una salida fija de +5V. Los dos capacitores funcionan como filtros. Este voltaje de +5V, alimenta todo el circuito. El voltaje de +3.3, necesario en algunas etapas del circuito, lo obtuve del ESP32 (PIN1).

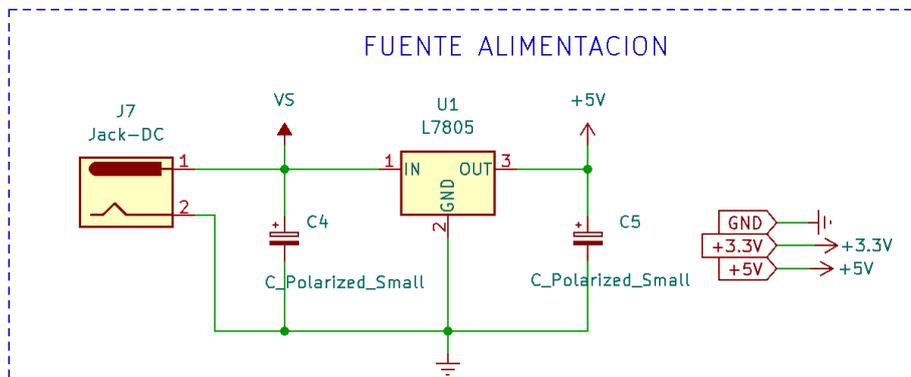


Figura 32 - Fuente Alimentación

El ESP32, como explico anteriormente cumple la función de adquisición y procesamiento de todos los datos tomados por los sensores. Se conecta a internet mediante WiFi y mediante protocolo MQTT, se comunica con una interfaz HMI. Los pines elegidos para la realización de cada función se encuentran en la Tabla 11 y la Figura 33.

Concepto	TIPO	PIN ESP32
Sensor Temperatura Evaporador	Entrada	GPIO25
Sensor Temperatura Ambiente	Entrada	GPIO25
Sensor Corriente Compresor	Entrada	GPIO34/ADC6
Sensor Corriente Evaporador	Entrada	GPIO32/ADC4
Sensor Corriente Condensador	Entrada	GPIO33/ADC5
Contacto Sensor Corriente Condensador	Entrada	GPIO39
Contacto Sensor Corriente Evaporador	Entrada	GPIO35
Contacto Sensor Corriente Compresor	Entrada	GPIO36
LED Indicador Broker MQTT Conectado	Salida	GPIO27
LED Indicador WiFi Conectado	Salida	GPIO14

Tabla 11 - Selección Pines ESP32

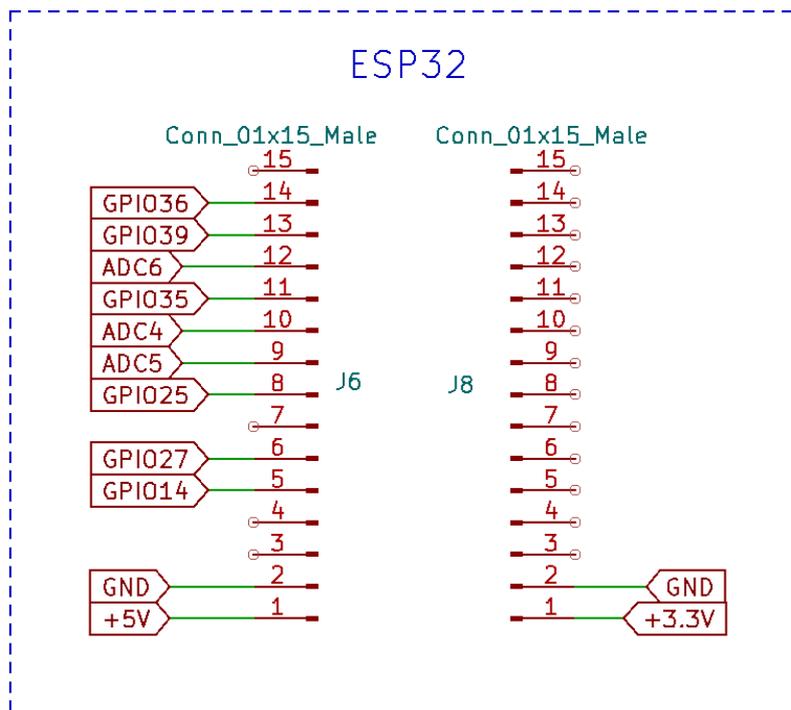


Figura 33 - Pines ESP32

Los sensores de temperatura son dos DS18B20. Uno que recibe valores de la temperatura del evaporador del equipo de frío y el otro que recibe valores de temperatura ambiente. Como había mencionado anteriormente el protocolo de comunicación es 1-Wire. El circuito de polarización de estos sensores es una resistencia de 4.7Ohms (Figura 34), dado por el fabricante entre los terminales de +5V y Datos.

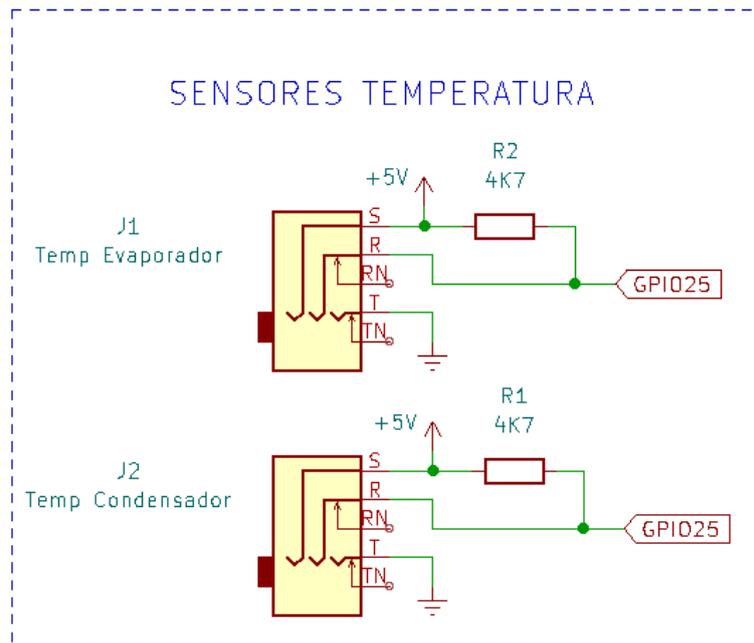


Figura 34 - Sensores Temperatura

Los sensores de corriente son tres transformadores SCT-013. Uno de 10A, el cual toma valores de corriente del motor compresor. Y dos de 5A, los cuales leen valores de corriente de los motores del evaporador y el condensador respectivamente. Luego por software las corrientes instantáneas se traducen en valores de corriente eficaz.

Como habíamos visto, los valores de tensión que entregan los transformadores son de 1V alterno. Es decir que tengo una parte de la señal que es positiva y otra negativa. Las entradas analógicas de la mayoría de los microprocesadores, trabajan con voltajes positivos. El ESP32, trabaja con señales de 0-3.3V en sus pines de entrada analógicos. Tiene varios modos de configuración del ingreso de amplitud señal en sus pines analógicos (Ver Tabla 12). Durante la configuración de los pines analógicos, definí un rango de entrada de 3.3V, utilizando de esta forma el rango máximo permitido para el voltaje de entrada en el terminal del ADC. Para adaptar la señal que entrega el sensor y utilizarla en el microcontrolador, agregué una señal de offset (una constante de tensión) y de esta forma todos los valores de tensión fueron positivos. Para la adaptación de la señal usé un divisor resistivo (Figura 35).

Antes de calcular el divisor resistivo, calculé los valores de tensión pico que entrega el sensor de corriente. El voltaje rms de salida es 1V.

$$V_{RMS} = 1V$$

$$V_P = V_{RMS} * \sqrt{2}$$

$$V_P = 1,414V$$

$$V_{PP} = V_P * 2$$

$$V_{PP} = 2.828$$

Los rangos recomendados por el fabricante son lecturas por encima de los 100mV. De esta forma El

voltaje en el punto medio del divisor resistivo es:

$$V_{offset} = \frac{V_{PP}}{2} + 100mV$$

$$V_{offset} = \frac{2.828V}{2} + 100mV$$

$$V_{offset} = 1.514V$$

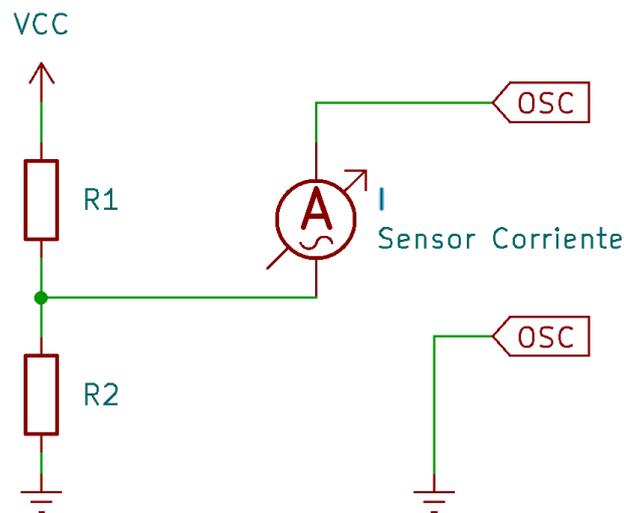


Figura 35 - Adaptación Señal Sensor Corriente

El cálculo del divisor resistivo es el siguiente:

$$V_{offset} = \frac{V_{3.3V} * R_2}{R_1 + R_2}$$

Definí R2=33KΩ, y despejando R1 nos queda

$$R_1 = \frac{R_2 * (V_{3.3V} - V_{offset})}{V_{offset}}$$

$$R_1 = \frac{33K * (3.3V - 1.514V)}{1.514V}$$

$$R_1 = 38.92K$$

La resistencia normalizada que se aproxima más al valor obtenido es **R₁= 39KΩ**.

Cuando Vin del sensor sea 1Vrms, tengo 1414mV de semiciclo positivo y 1414mV de semiciclo negativo. El voltaje máximo en la entrada del ESP es:

$$V_{inmax} = V_{offset} + V_p$$

$$V_{inmax} = 1514mV + 1414mV$$

$$V_{inmax} = 2928mV$$

De esta forma queda formado el divisor resistivo para el ingreso de la señal alterna proveniente de los SCT-013. Agregué además un capacitor de 0.1uF, para filtrar armónicos.

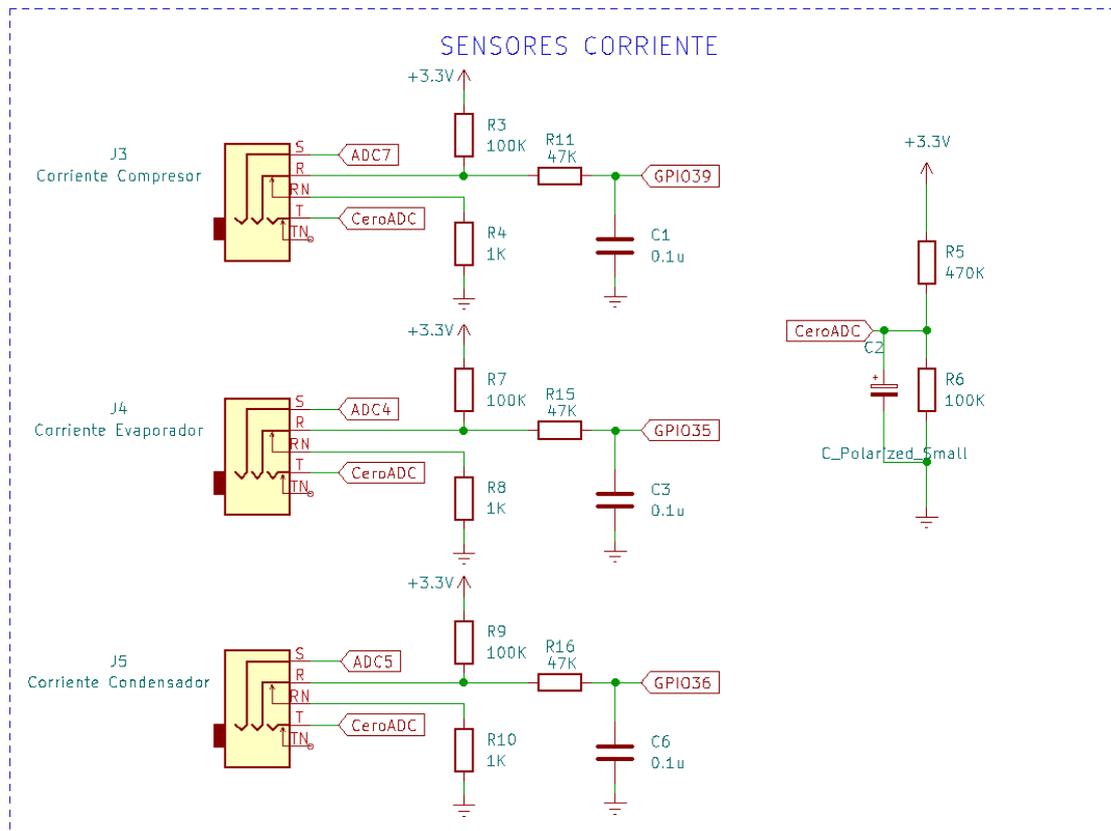


Figura 36 - Sensores Corriente

Los jacks de audio, utilizados para la conexión de los sensores, tienen un par de contactos normal cerrado. Utilicé un par de ellos en el diseño para indicar la presencia de sensor. Para ello requerí de unos pocos componentes pasivos.

Cuando se inserta un sensor (miniplug 3.5mm), el interruptor se abre. El circuito de la Figura 37 muestra cómo deben conectarse los componentes pasivos para realizar la detección de presencia del sensor.

Cuando el sensor no está conectado, R y RN están cerrados. El nivel de tensión que llega a GPIO ESP es el que queda determinado por el divisor resistivo R1 y R2 principalmente. La caída que aporta R3, está dado por la corriente que circula por la misma, la cual es despreciable. Es decir que a GPIO ESP llegan:

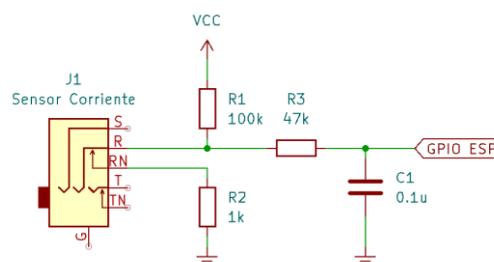


Figura 37 - Detección Presencia Sensor

$$V_{GPIO\ ESP} = \frac{V_{3.3V} * R_2}{R_1 + R_2}$$

$$V_{GPIO\ ESP} = \frac{V_{3.3V} * 1k}{100k + 1k}$$

$$V_{GPIO\ ESP} = 0.03V$$

En este caso el valor que ingresa al ESP es un cero lógico.

Cuando se conecta el sensor en el Jack, el contacto NC abre el circuito y la corriente circula por R1 y R3. Como el consumo de corriente del microcontrolador es muy baja (del orden de los micro amperes), la caída de voltaje que produce la corriente es despreciable. En este caso el valor que ingresa al ESP es un estado lógico alto.

De esta forma detecto presencia en los sensores de corriente. Es un dato que luego envío a la interfaz HMI para mostrar esta información.

Por ultimo instalé tres LEDs indicadores, uno de color rojo para indicar presencia de corriente y +5V en todo el circuito. Y dos de color verde, los cuales fueron encargados de indicar presencia de Internet en la placa (GPIO13) y conexión con el Bróker MQTT (GPIO12).

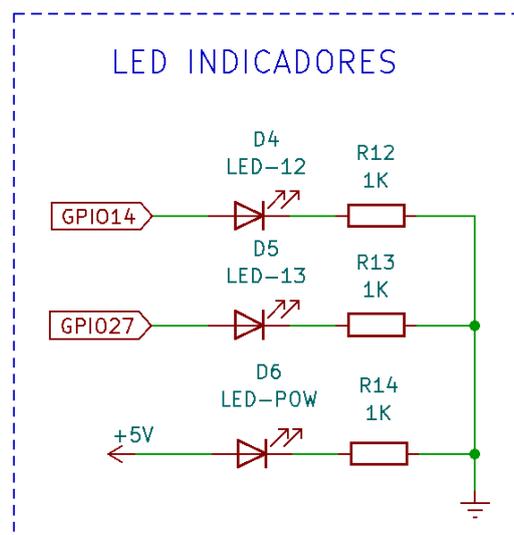


Figura 38 - LEDs Indicadores

2.3.1 Pruebas, experiencias.

Se realizaron pruebas de medición con los sensores para comprender su funcionamiento y manejo de protocolos/librerías.

2.3.2 Pruebas Sensor de Temperatura.

En primer lugar hice pruebas con los sensores de temperatura. Se conectó un DS18B20 al microcontrolador y se probó el protocolo 1-Wire (Figura 39). Se configuró y se capturaron datos de la temperatura ambiente.

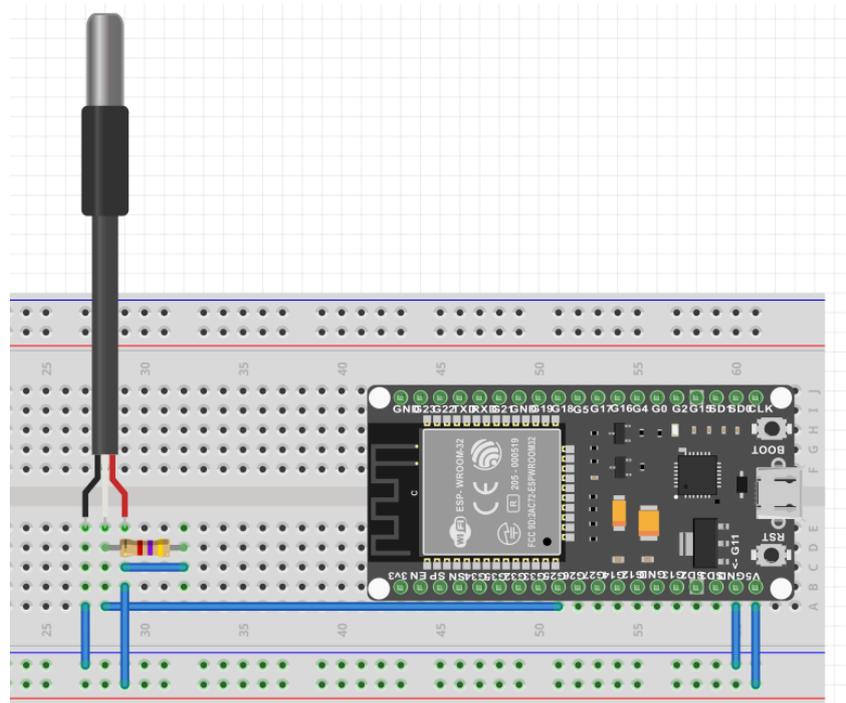


Figura 39 - Conexión DS18B20 al ESP32

Las pruebas se realizaron tomando la temperatura ambiente. Los resultados fueron exitosas, con resultados idénticos a la temperatura ambiente. Hubo algunos inconvenientes durante la integración al código principal.

```
#include <stdio.h>
#include "ds18b20.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "driver/gpio.h"

// Temp Sensors are on GPIO25
#define TEMP_BUS 25
DeviceAddress tempSensors[1];
search(tempSensors [0],true);

void app_main(void){
    ds18b20_init(TEMP);
    getTempAddresses(tempSensors);
    ds18b20_setResolution(tempSensors,1,10);
    while (1) {
        ds18b20_requestTemperatures();
        float temp = ds18b20_getTempC((DeviceAddress *)tempSensors[0]);
        ESP_LOGI("Msj: ", "Temperatura Sensor: %.2fC \n", temp);
        vTaskDelay(1000 / portTICK_PERIOD_MS);
    }
}
```

La librería y el protocolo 1-Wire, durante el proceso de obtención de la temperatura necesita un tiempo muerto muy alto y eso producía errores en las mediciones durante la ejecución del resto del programa, por distintas interrupciones que ocurren durante la ejecución de la misma. La solución fue volver a tomar la temperatura nuevamente en caso de error.

```
[0;32mI (13791) Msj: Temperatura Sensor: 31.00C
[0m
[0;32mI (14891) Msj: Temperatura Sensor: 31.00C
[0m
[0;32mI (15991) Msj: Temperatura Sensor: 31.00C
[0m
[0;32mI (17091) Msj: Temperatura Sensor: 31.00C
[0m
[0;32mI (18191) Msj: Temperatura Sensor: 31.00C
[0m
[0;32mI (19291) Msj: Temperatura Sensor: 31.00C
[0m
```

Figura 40 - Pruebas Sensor Temperatura

2.3.3 Pruebas Sensor Corriente.

Es necesario considerar el teorema del muestreo de Nyquist Shannon para la digitalización de la señal analógica continua (Figura 41 a). La corriente continua de la línea, en nuestro país es de 50Hz. El teorema del muestreo define que la frecuencia mínima a la que se debe realizar el muestreo es del doble de la frecuencia mayor de la señal que quiero muestrear. Es decir al menos 100Hz. Opte por tomar cien muestras en un período, es decir que el muestreo se realizó a una frecuencia de 5KHz, o si lo pasamos a período una muestra cada 200us (Figura 41 b).

La señal que entregan los sensores SCT-013 es 1V para una corriente de entrada rms de 10A (modelo SCT-013-010).

Para la generación de la señal de muestreo creé dos bucles anidados. Uno para generar las muestras a 5KHz, y la otra para el período de 20ms en el que ocurre un ciclo de la señal de línea.

Por otro lado está la cuantificación de la señal, esto es la asignación de niveles en amplitud. La cantidad de niveles de cuantificación está determinada por los conversores A/D que posee el ESP32, que como ya vimos son de 12bits. Es posible configurar los conversores A/D entre 9 y 12 bits. Elegí la más alta para poder tener la mejor resolución (Ver Tabla 12). Es decir con $V_{in}=3.3V$, tenemos $2^{12}=4096$ bits. Calculando el incremento por bit obtuve el siguiente resultado:

$$V_{1bit} = \frac{3.3V}{4096}$$

$$V_{1bit} = 805.66\mu V$$

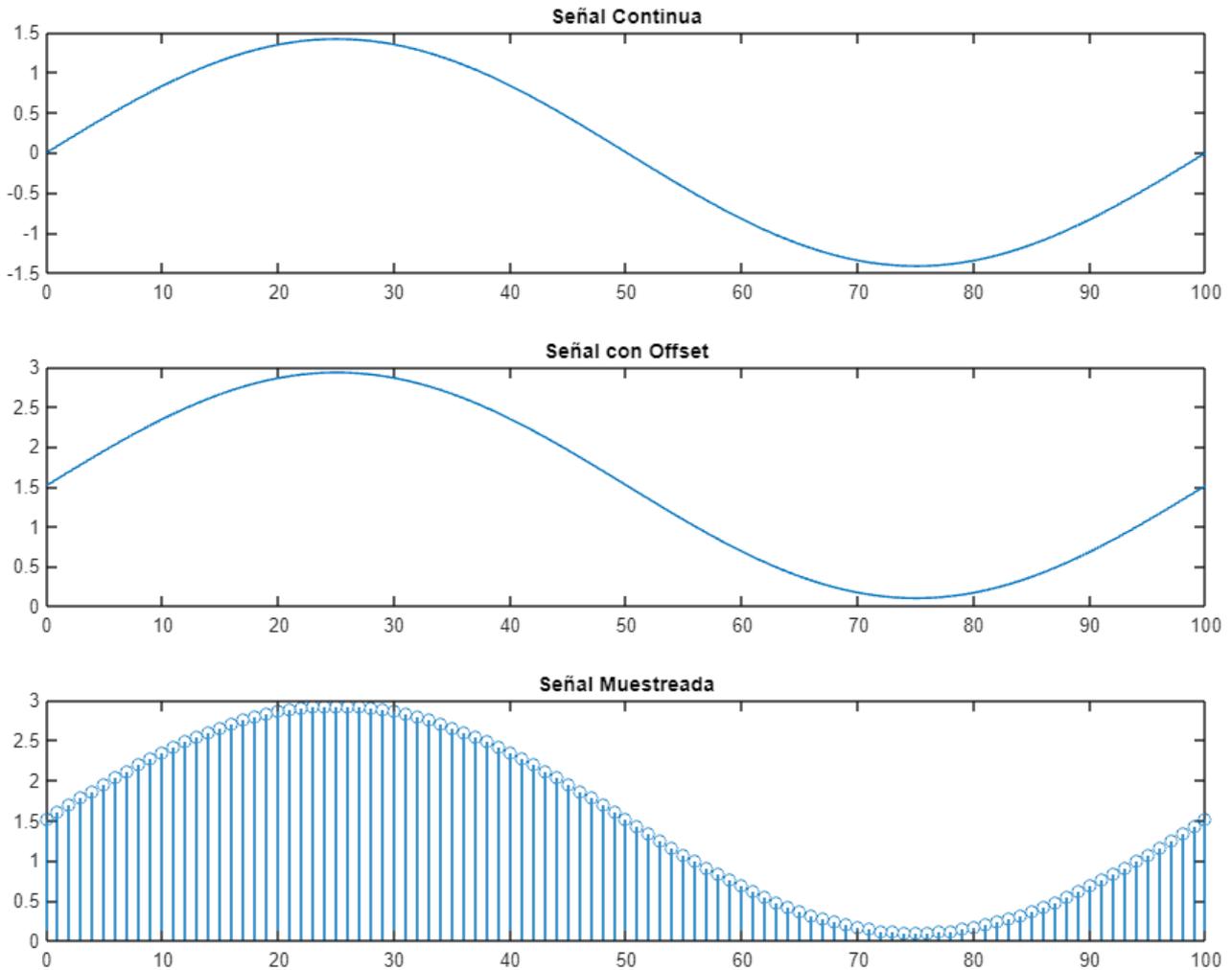


Figura 41 - Señal Continua, con Offset y Muestreada

La entrada a fondo de escala por defecto en el ADC es de 1.1V y se puede configurar hasta una entrada a fondo de escala de 3.3V. En la siguiente tabla se muestran las distintas atenuaciones con los valores máximos de entrada para cada atenuación configurable.

Cuando VDD es 3.3V en el ESP32:

Atenuación	Configuración ESP32	Rango máximo Voltaje	Rango Recomendado
0db	ADC_ATTEN_DB_0	1.1V	100-950mV
2.5db	ADC_ATTEN_DB_2_5	1.5V	100-1250mV
6db	ADC_ATTEN_DB_6	2.2V	150-1750mV
11db	ADC_ATTEN_DB_11	3.3V	150-2450mV

Tabla 12 - Atenuación ADC

Las pruebas se realizaron midiendo el consumo de una plancha de 650W con la pinza de 5A. Calculé la corriente rms que debió arrojar el sensor:

$$I_{rms} = \frac{P}{V_{rms}} = \frac{650W}{220V} = 2.95A$$

n	bits								
1	1200	21	641	41	1497	61	2683	81	2577
2	1178	22	667	42	1600	62	2729	82	2470
3	1152	23	704	43	1651	63	2762	83	2410
4	1115	24	690	44	1691	64	2759	84	2394
5	992	25	705	45	1843	65	2749	85	2287
6	945	26	730	46	1876	66	2832	86	2222
7	909	27	711	47	1895	67	2790	87	2118
8	880	28	752	48	1987	68	2794	88	2127
9	878	29	769	49	2015	69	2768	89	2066
10	770	30	802	50	2131	70	2771	90	1990
11	705	31	848	51	2162	71	2687	91	1874
12	712	32	917	52	2239	72	2730	92	1856
13	661	33	990	53	2305	73	2731	93	1745
14	606	34	1044	54	2314	74	2742	94	1725
15	650	35	1079	55	2464	75	2669	95	1663
16	593	36	1150	56	2423	76	2701	96	1582
17	599	37	1264	57	2512	77	2656	97	1531
18	627	38	1337	58	2532	78	2646	98	1408
19	622	39	1431	59	2623	79	2672	99	1385
20	624	40	1482	60	2672	80	2625		

Tabla 13 - Muestras ADC Sensor Corriente

n	Vin	n	Vin	n	Vin	n	Vin	n	Vin
1	-0,419900	21	-0,895050	41	-0,167450	61	0,839800	81	0,749700
2	-0,438600	22	-0,872950	42	-0,079900	62	0,878900	82	0,658750
3	-0,460700	23	-0,841500	43	-0,036550	63	0,906950	83	0,607750
4	-0,492150	24	-0,853400	44	-0,002550	64	0,904400	84	0,594150
5	-0,596700	25	-0,840650	45	0,125800	65	0,895900	85	0,503200
6	-0,636650	26	-0,819400	46	0,153850	66	0,966450	86	0,447950
7	-0,667250	27	-0,835550	47	0,170000	67	0,930750	87	0,359550
8	-0,691900	28	-0,800700	48	0,248200	68	0,934150	88	0,367200
9	-0,693600	29	-0,786250	49	0,272000	69	0,912050	89	0,315350
10	-0,785400	30	-0,758200	50	0,370600	70	0,914600	90	0,250750
11	-0,840650	31	-0,719100	51	0,396950	71	0,843200	91	0,152150
12	-0,834700	32	-0,660450	52	0,462400	72	0,879750	92	0,136850
13	-0,878050	33	-0,598400	53	0,518500	73	0,880600	93	0,042500
14	-0,924800	34	-0,552500	54	0,526150	74	0,889950	94	0,025500
15	-0,887400	35	-0,522750	55	0,653650	75	0,827900	95	-0,026350
16	-0,935850	36	-0,462400	56	0,618800	76	0,855100	96	-0,095200
17	-0,930750	37	-0,365500	57	0,694450	77	0,816850	97	-0,138550
18	-0,906950	38	-0,303450	58	0,711450	78	0,808350	98	-0,243100
19	-0,911200	39	-0,223550	59	0,788800	79	0,830450	99	-0,262650
20	-0,909500	40	-0,180200	60	0,830450	80	0,790500		

Tabla 14 - Voltajes Instantáneos Sensor Corriente

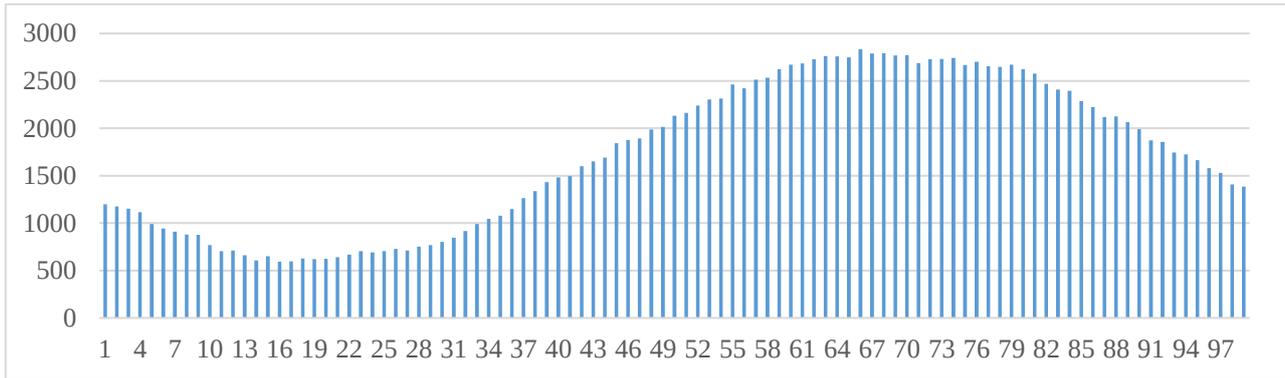


Figura 42 - Gráfica Muestras ADC Sensor Corriente

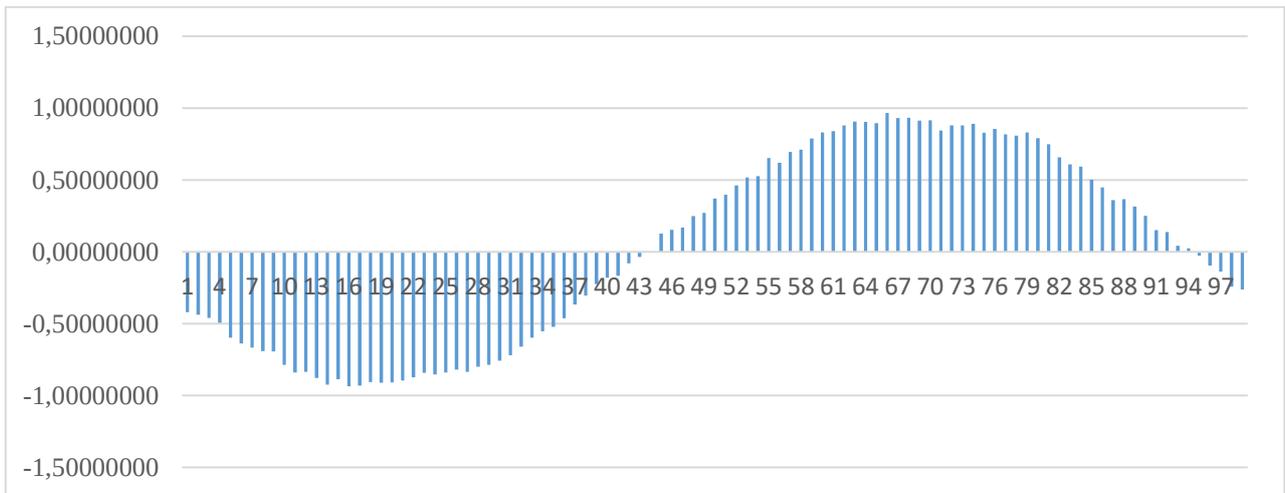


Figura 43 - Grafica Voltajes Instantáneos Sensor Corriente

La señal a la entrada fue de 1.78Vpp. Como el sensor utilizado en la medición fue de 5A, la corriente Irms fue:

$$I_{rms} = ESCALA\ PINZA * V_{rms}$$

$$I_{rms} = ESCALA\ PINZA * \sqrt{\sum_0^n \frac{(i_k)^2}{n-1}}$$

$$I_{rms} = 5 * 0.663161$$

$$I_{rms} = \mathbf{3.3156A}$$

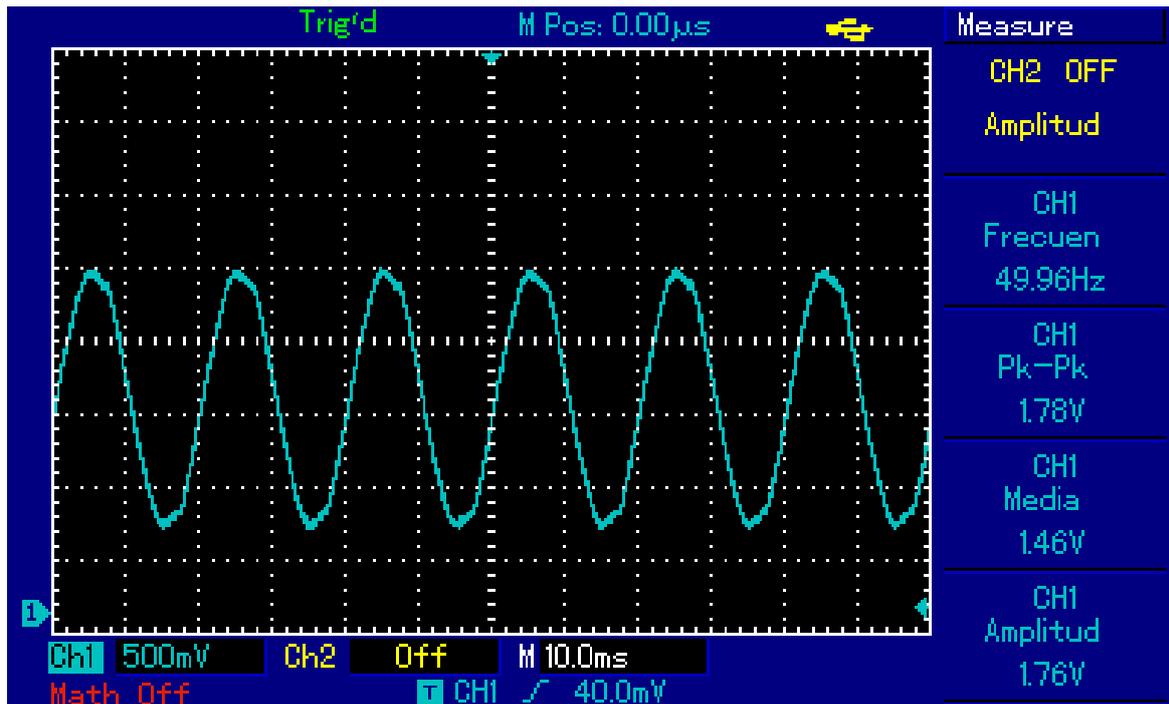


Figura 44 - Señal Osciloscopio

Cálculo realizado con las mediciones del osciloscopio.

$$I_{rms} = ESCALA\ PINZA * V_{rms}$$

$$I_{rms} = ESCALA\ PINZA * \frac{V_{pp}}{2 * \sqrt{2}}$$

$$I_{rms} = 5 * \frac{1.78V}{2 * \sqrt{2}}$$

$$I_{rms} = 3.15A$$

Con una diferencia de 0.16A, las mediciones logradas con el sensor SCT-013, fueron muy buenas.

2.4 Desarrollo del software o firmware.

2.4.1 Visual Studio Code.

El desarrollo del firmware del ESP32 se realizó en Visual Studio Code (Figura 45), el cual es un entorno de desarrollo integrado en el cual se puede programar con diferentes lenguajes. Dentro de este IDE, instalé la extensión PlatformIO, el cual permite realizar la programación del microcontrolador con framework sugerido por el fabricante: Espressif.

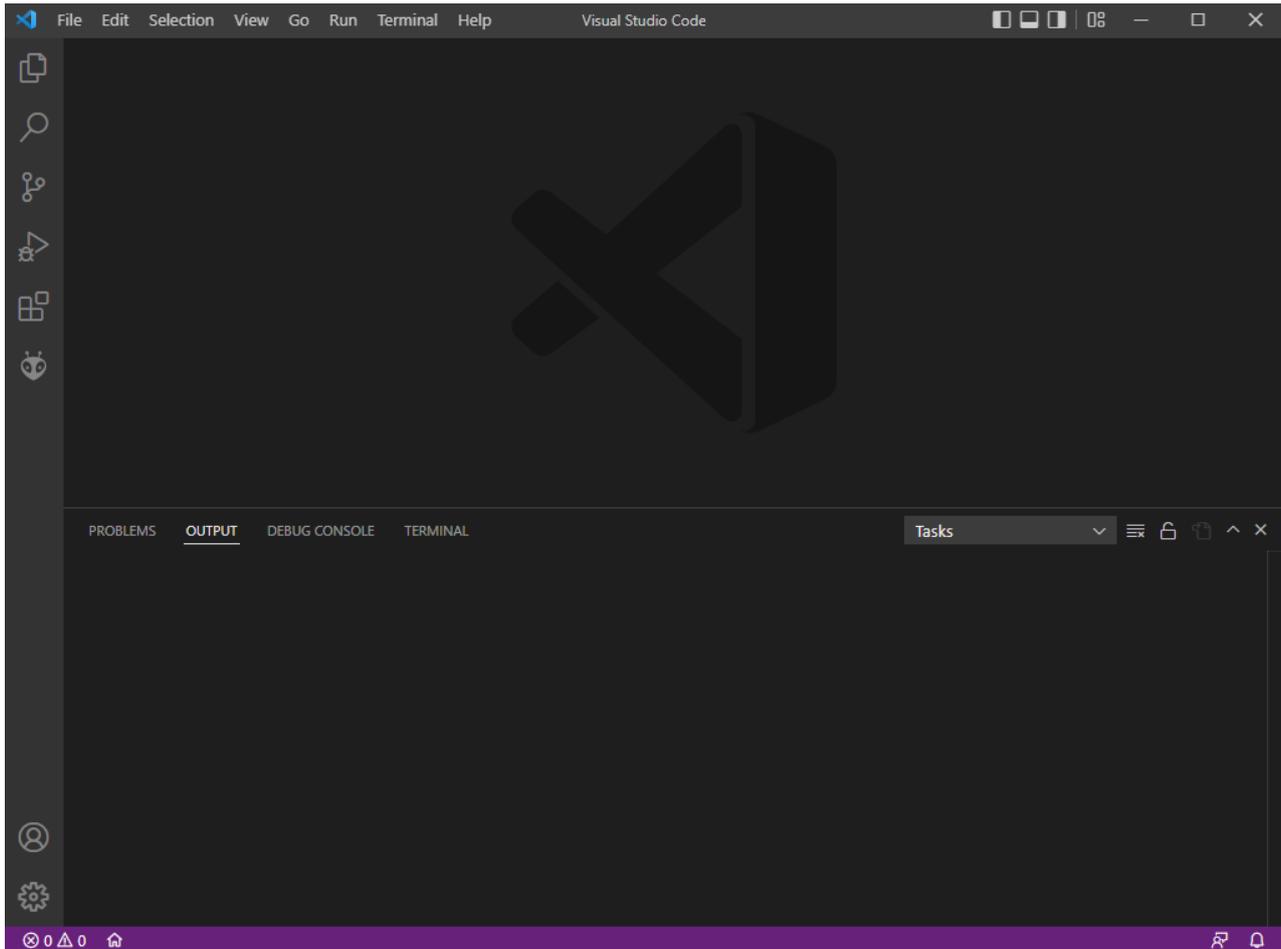


Figura 45 - Plataforma Visual Studio Code

El plugin PlatformIO, se instala en la paleta de extensiones.

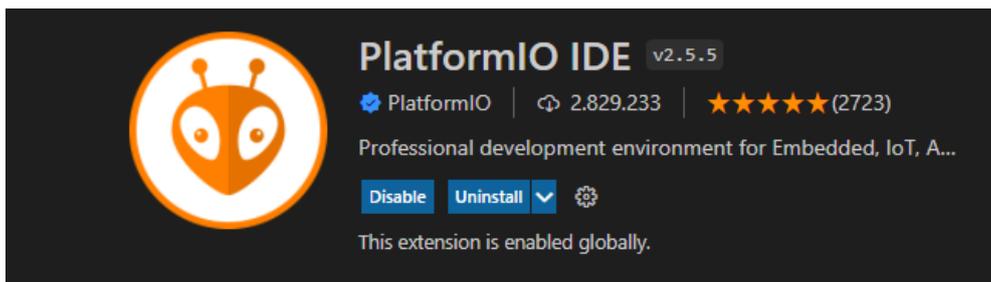


Figura 46 - PlatformIO IDE

Para iniciar un nuevo proyecto, ingresé a la extensión PlatformIO, en el menú PIO Home hice clic en Open. Dentro de esta ventana elegí New Project.



Figura 47 - Nuevo Proyecto

Dentro del proyecto nuevo definí en primer lugar un nombre. En el campo Board, seleccioné NodeMCU-32S. Y el Framework: Espressif IoT Development Framework. Una vez finalizados los campos, toqué en finish se crearon todas las carpetas y archivos del nuevo proyecto.

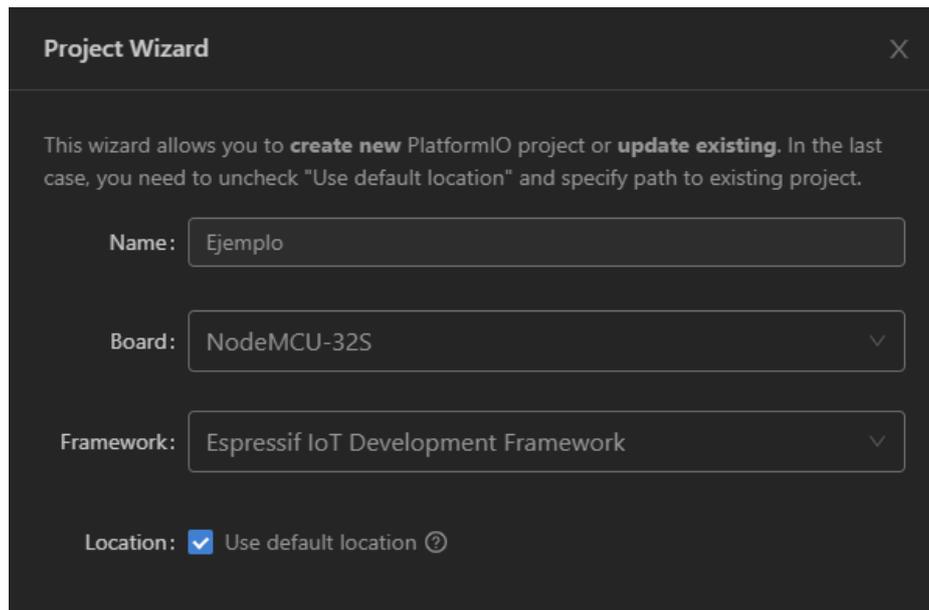


Figura 48 - Campos Proyecto Nuevo

Para empezar a programar el código, ingresé en la carpeta src y al archivo main.c. Dentro de esta carpeta fui programando y segmentando el código para tener un esquema lo más ordenado posible. El archivo main.c, es el que tiene el programa principal. En la carpeta include, agregué las librerías que fueron requeridas por el código, por ejemplo las librerías para el funcionamiento del sensor de temperatura DS18B20. Y también los archivos que contengan las funciones que fui creando a lo largo del desarrollo.

2.4.2 Diagrama en bloques programa principal.

Los bloques principales del funcionamiento del programa del ESP32, los podemos observar en la Figura 49. Cada uno de los bloques se explicará en detalle a lo largo de los siguientes párrafos.

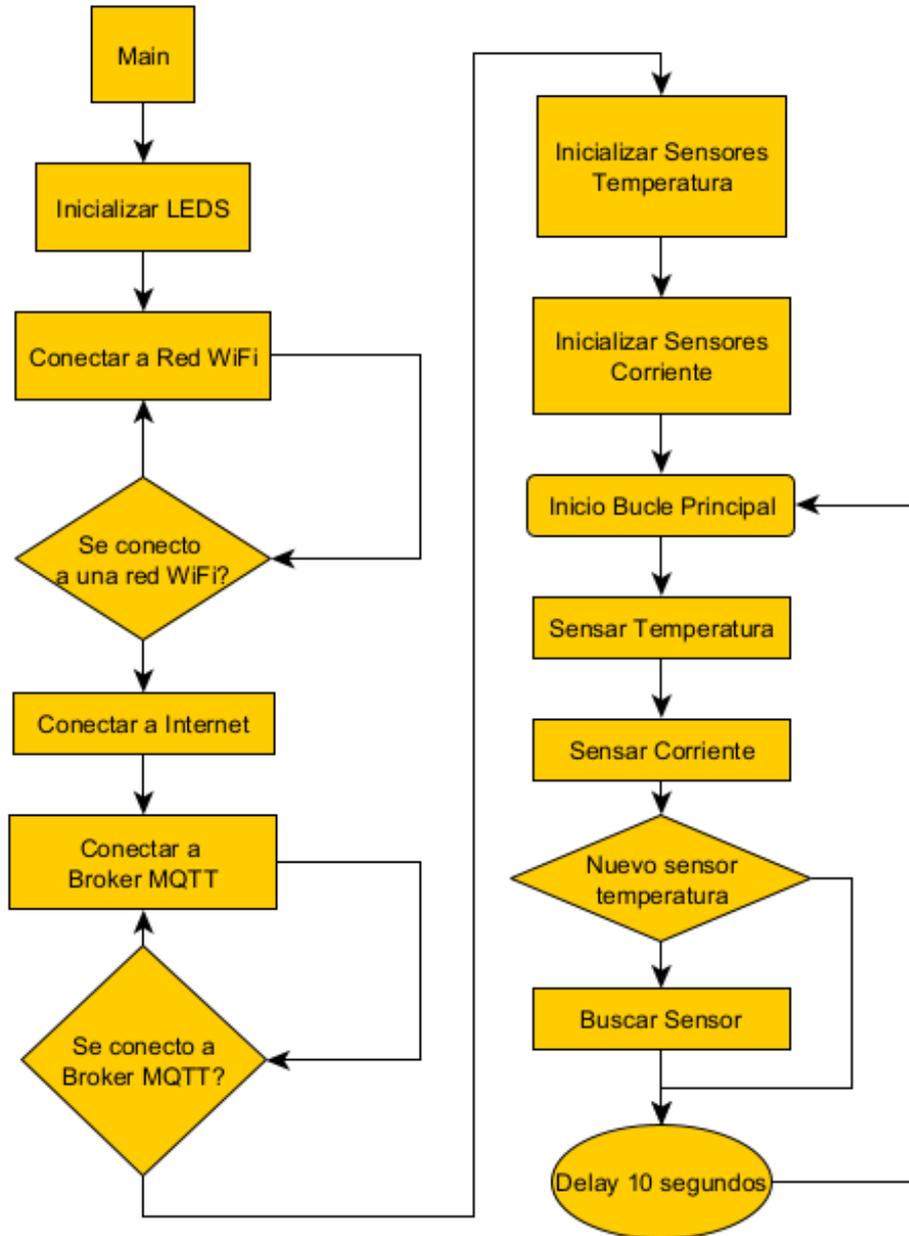


Figura 49 - Programa Principal ESP32

2.4.3 Inicializar LEDs.

En el primer bloque, Inicializar LEDs, configuré los GPIO destinados a la indicación de presencia de tensión, conexión WiFi y conexión al Bróker MQTT. Todos ellos se configuraron como salida. (Ver Tabla 11).

```

esp_err_t init_led(void)
{
    gpio_reset_pin(LEDWIFI);
    gpio_reset_pin(LEDMQTT);
    gpio_set_direction(LEDWIFI, GPIO_MODE_OUTPUT);
    gpio_set_direction(LEDMQTT, GPIO_MODE_OUTPUT);
    return ESP_OK;
}
    
```

2.4.4 Conectar a la red WiFi, ESP Touch.

Una vez configurados los GPIO de los LEDs, inicio la conexión a una red WiFi.

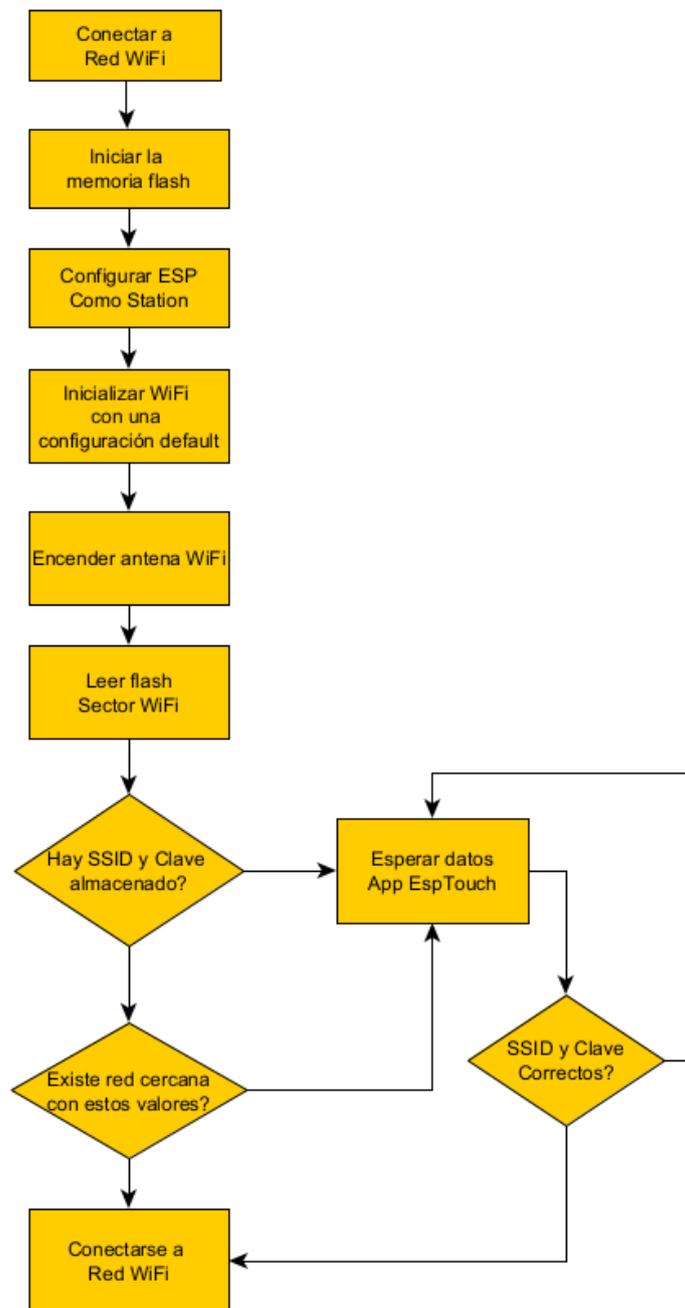


Figura 50 - Diagrama Conexión a Red WiFi

Dentro de las posibilidades que tiene el microcontrolador, existen librerías dedicadas para trabajar con Protocolo de Comunicaciones, tal como es WiFi y MQTT. A continuación muestro algunas de las librerías de las que explicaré alguna de sus funciones en este apartado.

```
#include "esp_wifi.h"
#include "esp_wpa2.h"
#include "nvs_flash.h"
#include "esp_event.h"
#include "esp_smartconfig.h"
```

La memoria flash del ESP tiene un sector dedicado al almacenamiento de SSID (nombre de la red WiFi) y Password. En caso de reset del microcontrolador, se puede consultar este espacio de memoria y realizar la conexión a la red WiFi. Es por eso que en primer lugar inicializo la memoria flash.

El driver WiFi del ESP puede configurarse como Access Point (punto de acceso, es decir puedo conectarme al ESP mediante WiFi). O también puedo configurarlo como Station (el ESP se conecta a una red WiFi). Luego de iniciar la memoria flash, configuré el ESP como Access Point.

Se inicializa el protocolo WiFi con una configuración “default” o por defecto que viene incluida en la librería. Esto lo aconseja el fabricante para garantizar el correcto funcionamiento durante la primera configuración. Luego puede cambiarse su configuración a una específica. Figura 50.

Enciendo el módulo WiFi y procedo a leer la memoria flash. Durante la primera ejecución del programa, no existían datos almacenados en la memoria, entonces el programa llama a la función “Esperar datos de la App EspTouch”. En este punto el programa se quedó en un bucle esperando recibir datos de una App. La App, la descargué del PlayStore (Figura 51) y trabaja con la librería SmartConfig, propia del fabricante Espressif. La aplicación me permitió configurar la red WiFi a la que quise que se conecte el ESP32. Esta app se llama EspTouch: SmartConfig for ESP8, desarrollada por KhoaZero123.

Una consideración antes de utilizar la App, debido a que el ESP32 solamente trabaja en la frecuencia de 2.4GHz, cuando se efectúa la conexión, la antena WiFi debe ser 2.4GHz, sino la app arroja un error.

Conecté un dispositivo móvil a una red WiFi. De ésta conexión, la app toma el SSID de la red WiFi y queda un campo disponible para escribir el Password (Figura 52). Al hacer clic en confirmar, el programa del ESP recibe los datos de SSID y contraseña e intenta conectarse a esa red con los datos enviados desde la app. Si los datos son correctos, se configura el ESP y se conecta a esa red WiFi. Caso contrario, vuelve a esperar nuevos datos desde la App. Figura 50.



Figura 51 - PlayStore: EspTouch

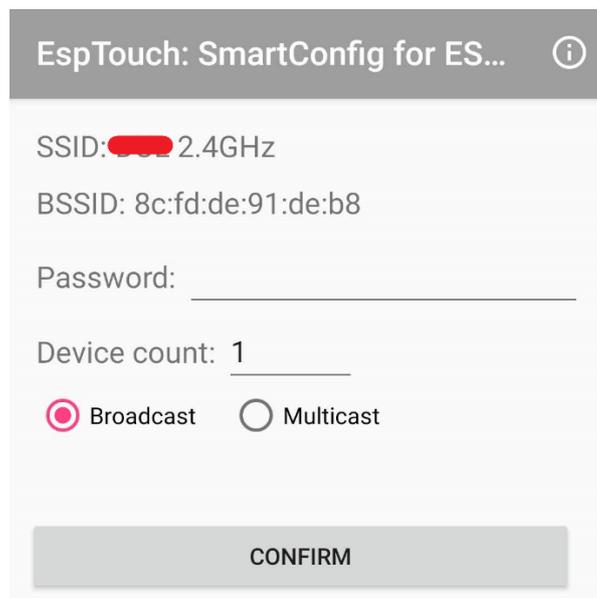


Figura 52 - EspTouch Interfaz

Una vez en el programa principal del ESP, se almacenaron estos en la memoria flash, se configuro el ESP como Station e intento conectarse a la red WiFi con los datos ingresados en la app.

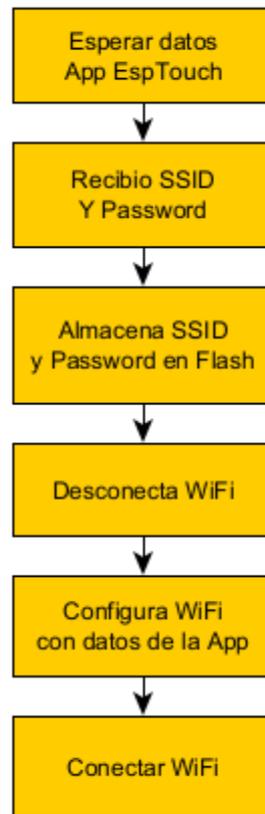


Figura 53 - SmartConfig y EspTouch

En caso de cambio de SSID y/o Password del router, por diferentes motivos (cambio de cable modem, router, configuración, traslado, etc.). El equipo se reinicia con valores guardados en el espacio de memoria flash de SSID y Password. Pero al ser diferentes a los nuevos valores de la red WiFi, el programa queda a la espera de nuevos valores provenientes de la App. Cuando estos valores coinciden con datos correctos del cable modem/router, el ESP se conecta a la red WiFi. Seguidamente el ESP adquirió la IP y Máscara de Red del modem y se conectó a Internet. Figura 50.

El SSID de la red a la cual me conecté, se almacenó también en una variable global, que luego utilicé para publicarlo en un tópico MQTT, el cual se podrá visualizar en la interfaz usuario Dashboard, que explicaré más adelante.

2.4.5 Conexión a Bróker MQTT (HiveMQ).

Una vez conectado a internet, procedí a conectar el Bróker MQTT. Elegí un Bróker MQTT gratuito, HIVEMQ. Figura 54.

El Bróker MQTT de HiveMQ facilitó el movimiento de datos hacia y desde los dispositivos conectados de forma eficiente, rápida y fiable. Permite la conexión de 100 dispositivos y hasta 10Gb de almacenamiento de datos en forma gratuita.

Ingresé a <https://www.hivemq.com/mqtt-cloud-Broker/> y cree una cuenta.

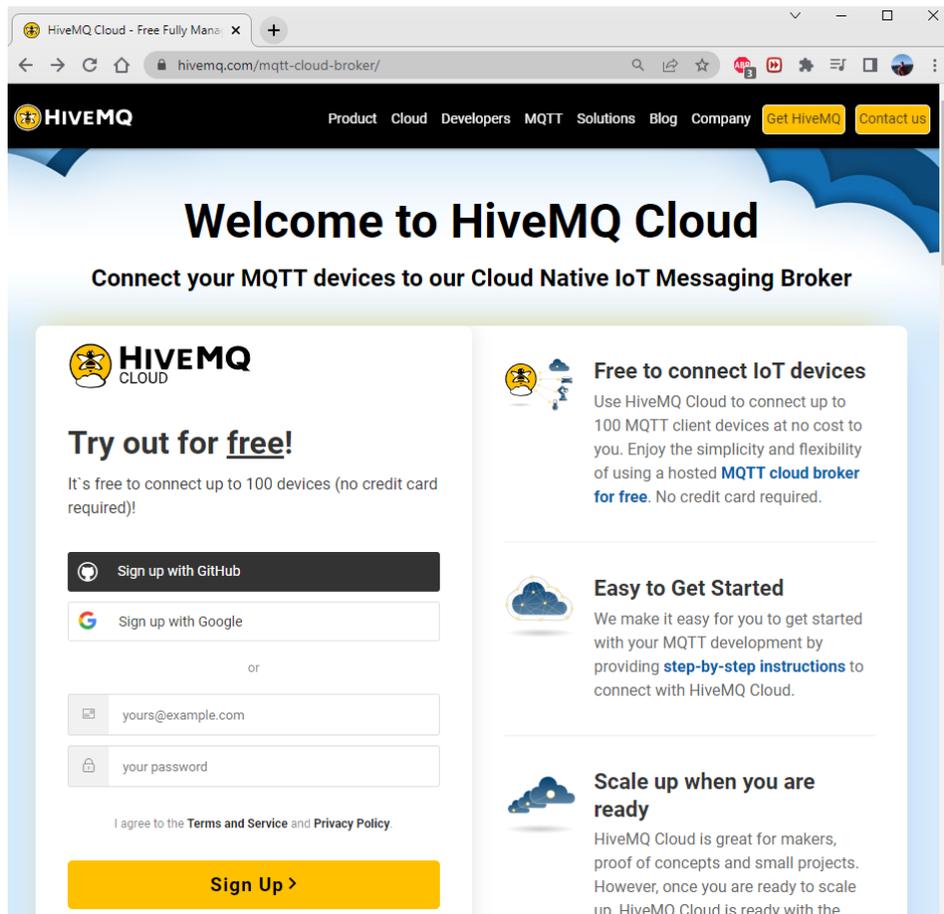


Figura 54 - HiveMQ Ingreso

Una vez dentro de la cuenta, establecí un nuevo Cluster (Figura 55). Elegí la opción gratuita.

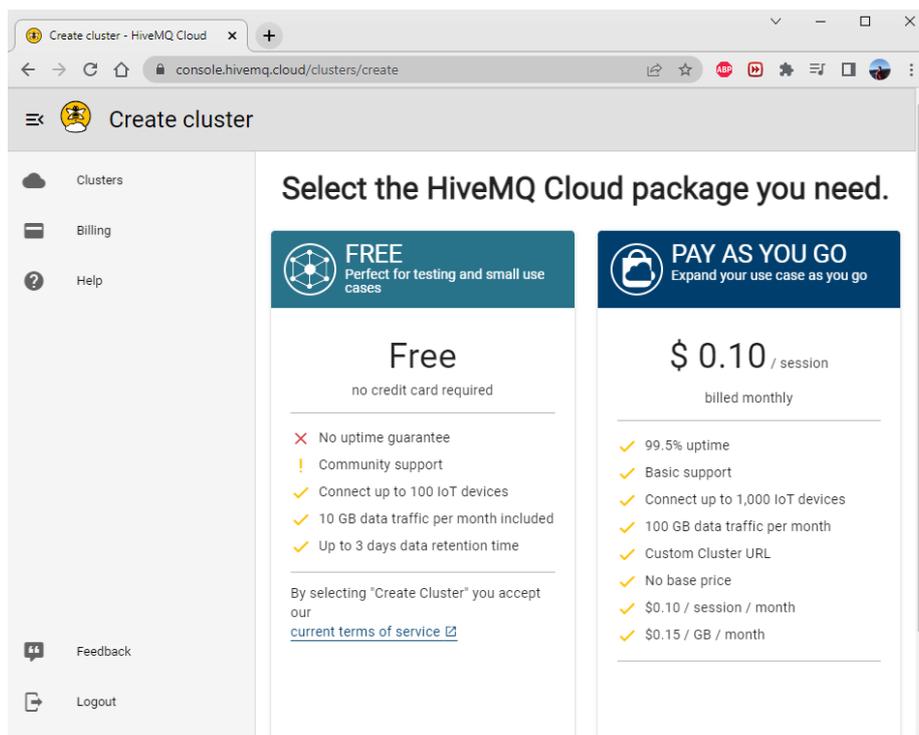


Figura 55 - HiveMQ Nuevo Cluster

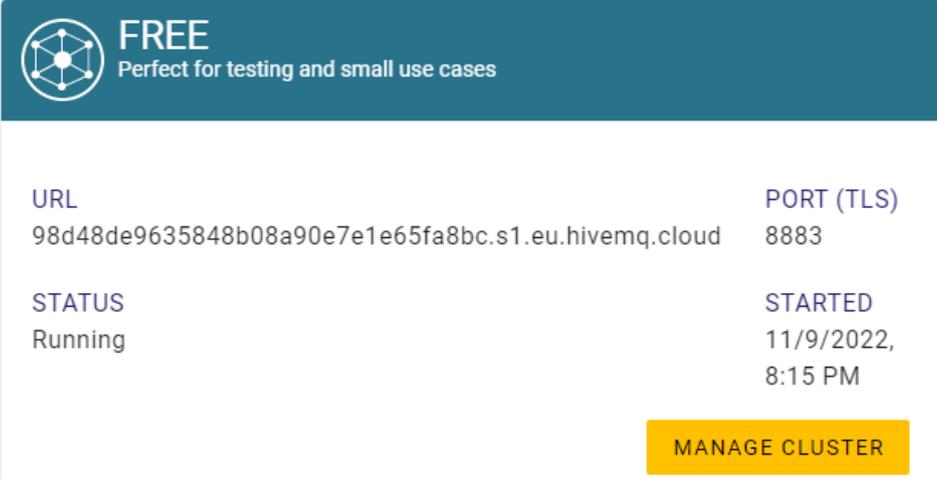
Una vez creado el clúster, tuve acceso a las credenciales para trabajar con un Bróker (Figura 56), cree un usuario y una contraseña para poder suscribir y publicar en los distintos tópicos que definí en el sistema. Además de estas credenciales, necesité un certificado SSL, para darle mayor seguridad a la conexión entre el ESP y el Bróker MQTT.

Un certificado SSL es un certificado digital que autentica la identidad de un sitio web y habilita una conexión cifrada. La sigla SSL significa Secure Sockets Layer (Capa de sockets seguros), un protocolo de seguridad que crea un enlace cifrado entre un servidor web y un navegador web.

Para generar el certificado SSL, ingresé al terminal de comandos (Linux) y escribí la siguiente línea:

```
echo "" | openssl s_client -showcerts -connect  
url.hivemq.cloud:8883 | sed -n "1,/Root/d; /BEGIN/,/END/p" |  
openssl x509 -outform PEM >hivemq.pem
```

El archivo hivemq.pem, tuve que agregar en la carpeta principal del proyecto en Visual Studio Code.



URL	PORT (TLS)
98d48de9635848b08a90e7e1e65fa8bc.s1.eu.hivemq.cloud	8883

STATUS	STARTED
Running	11/9/2022, 8:15 PM

MANAGE CLUSTER

Figura 56 - HiveMQ Credenciales Cluster

Como expliqué anteriormente, el protocolo MQTT trabaja mediante el modelo publicación/suscripción de mensajes. Cada mensaje se escribe en un tópico. Cada tópico tiene un identificador único, el cual sirve para identificar el tipo de dato que se quiere publicar o al cual quiero suscribirme, dependiendo si a esta información la vemos desde el lado del microcontrolador o desde el lado de la interfaz HMI.

Para el diseño de este proyecto, se definieron los tópicos teniendo como criterio las señales de los sensores, el estado de sus conexiones, funciones específicas relacionadas a las direcciones de los sensores de temperatura y algunos parámetros de información general para agregar detalles a la interfaz HMI. Tabla 15.

Tópico	Descripción
direccion/evaporador	Direccion Hexadecimal Temperatura Evaporador
direccion/ambiente	Direccion Hexadecimal Temperatura Ambiente
temperatura/evaporador	Sensor Temperatura Evaporador
temperatura/ambiente	Sensor Temperatura Ambiente
corriente/compresor/valor	Sensor Corriente Compresor
corriente/evaporador/valor	Sensor Corriente Evaporador
corriente/condensador/valor	Sensor Corriente Condensador
corriente/compresor/contacto	Contacto SC Compresor
corriente/evaporador/contacto	Contacto SC Evaporador
corriente/condensador/contacto	Contacto SC Condensador
wifi/ssid	Nombre Red WiFi
wifi/rssi	Potencia Red WiFi
buscar/sensores	Busca direcciones sensores temperatura
invertir/sensores	Invierte ubicación direcciones sensores temperatura

Tabla 15 - Tópicos MQTT

Para realizar la conexión del ESP al Bróker MQTT (Figura 57), cargué en el programa del microcontrolador las credenciales que obtuve de HiveMQ y el certificado SSL que generé con las credenciales. Se inicializó el cliente MQTT y paso seguido, se inicializaron los sensores de temperatura.

```
const esp_mqtt_client_config_t mqtt_cfg = {
    .uri = "mqtt://mqtt.eclipse.org:8883",
    .event_handle = mqtt_event_handler,
    .cert_pem = (const char *)mqtt_eclipse_org_pem_start,
};
esp_mqtt_client_handle_t client = esp_mqtt_client_init(&mqtt_cfg);
esp_mqtt_client_register_event(client, ESP_EVENT_ANY_ID, mqtt_event_handler,
client);
esp_mqtt_client_start(client);
```

Luego del inicio del cliente MQTT y antes de la inicialización de los sensores de temperatura, se publicó el primer mensaje, el ssid de la red wifi en el tópico wifi/ssid, el cual se almacenó previamente (ver 2.4.4 Conectar a la red WiFi, ESP Touch).

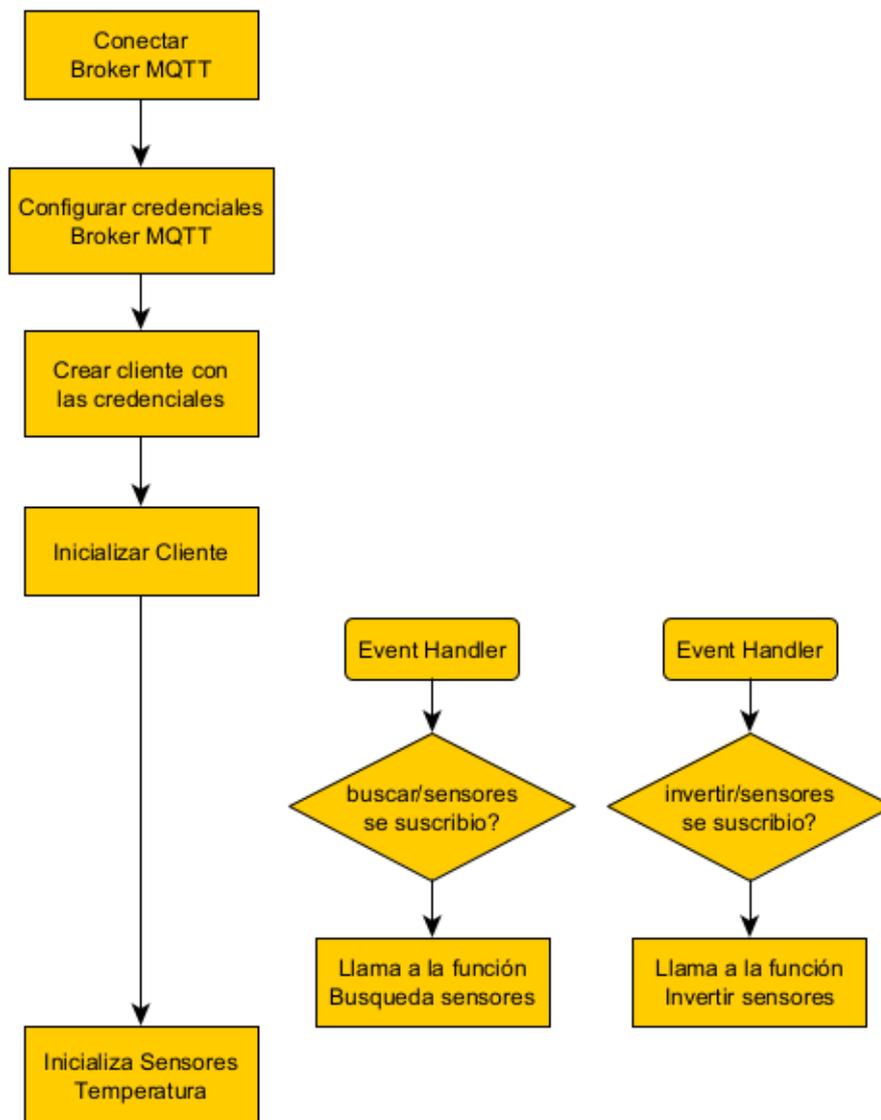


Figura 57 - Configuración MQTT

En paralelo tengo los “Event Handler”, que son interrupciones del programa principal. Los mismos ocurren en el programa en forma asíncrona. Este event handler se ejecuta cuando ocurre una solicitud de suscripción al tópico buscar/sensores o invertir/sensores, proveniente del Dashboard. Cuando se fuera necesario hacer un reemplazo de uno o dos de los sensores, por malfuncionamiento o deterioro, los sensores tienen una dirección única de 64bits, como vimos en el funcionamiento de los DS18B20. Esto va a producir una lectura errónea por el programa hasta que configure las direcciones correctas de los nuevos sensores. Para ello utilizo uso de un botón en la interfaz HMI, el cual al presionarlo realiza una solicitud al bróker MQTT un evento de suscripción al tópico buscar/sensores. Esta solicitud, genera una interrupción en el programa principal y atiende el “Event Handler”. Esta interrupción llama a la función búsqueda de sensores. Asimismo cuando las direcciones no se corresponden con la ubicación del sensor de temperatura del evaporador/ambiente, se puede realizar un intercambio de las direcciones llamando a la función invertir sensores. Estas funciones se invocan desde el panel de control Dashboard.

2.4.6 Inicializar sensores temperatura.

Antes de realizar los procesos de lectura, configuro los sensores definiendo los pines de entrada/salida del ESP, las resoluciones de los A/D, y lectura de direcciones según corresponda en cada caso (Figura 58). Para la inicialización de los sensores de temperatura, hice uso de la librería DS18B20, desarrollada por feelfreelinux. La misma implementa el protocolo 1-Wire para la configuración y captura de datos de los sensores.

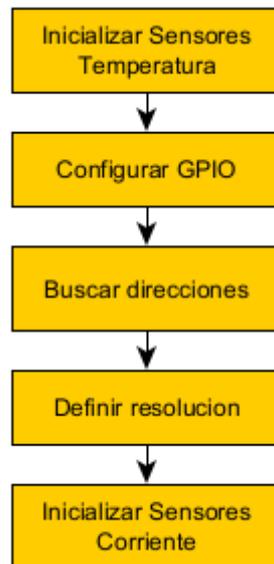


Figura 58 - Inicialización Sensores Temperatura

Lo primero que hace el programa, es definir el GPIO a utilizar para recibir los datos de los sensores 1-Wire. El terminal definido por la Tabla 11 para ésta función es el *GPIO13*. Mediante la función *search ()*, generó los pulsos de sincronismo, leyó las direcciones que tiene cada sensor y lo asignó a una variable tipo *DeviceAddress* definida por la librería. A continuación defino la resolución con la que realizaron las lecturas. A mayor resolución el tiempo que demora en entregar las mediciones son mayores. Utilicé una resolución de 10bits. Una vez configurados los sensores, las variables están en condiciones para realizar lecturas de temperatura.

```

DeviceAddress tempSensors;
ds18b20_init(GPIO25);
search(tempSensors[0],true); //Leemos dirección sensor 1
search(tempSensors[1],true); //Leemos dirección sensor 2
ds18b20_setResolution(tempSensors,2,10);
  
```

2.4.7 Inicializar sensores de corriente.

Para inicializar los sensores de corriente, definí los pines del ESP que fueron los encargados de tomar los valores analógicos, la atenuación y la cantidad de bits de resolución del ADC. .

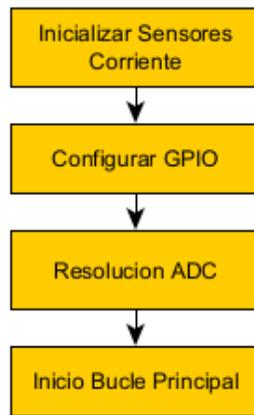


Figura 59 - Inicialización Sensores Corriente

```

adc1_config_channel_atten(ADC1_CHANNEL_7, ADC_ATTEN_DB_11);
adc1_config_channel_atten(ADC1_CHANNEL_4, ADC_ATTEN_DB_11);
adc1_config_channel_atten(ADC1_CHANNEL_5, ADC_ATTEN_DB_11);
adc1_config_width(ADC_WIDTH_BIT_12);
    
```

Los pines definidos para la configuración de los sensores de corriente, se encuentran en la Tabla 11.

La resolución que configuré es la más alta, 12bits. De esta forma tuve mejor precisión de las mediciones realizadas. La atenuación es solo configurable para los canales ADC1. La atenuación elegida fue la de 11db (Tabla 12). De esta forma tengo un mayor rango en los niveles de cuantización.

2.4.8 Lectura sensores temperatura.

Una vez configurados los sensores, el programa entra en el bucle principal, el cual realizó las mediciones de temperatura y corriente, controló si hubo un cambio de sensor mediante la interrupción “búsqueda/sensores”, o la interrupción “invertir/sensores” ejecutó un delay de 10 segundos y volvió a tomar nuevas mediciones. Figura 60.

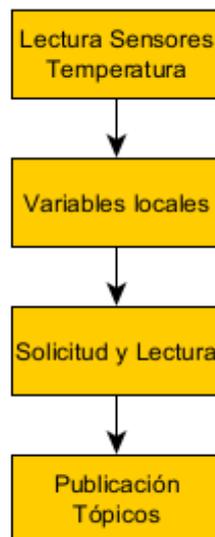


Figura 60 - Lectura Sensores Temperatura

Para la lectura de los sensores de temperatura, llamé a dos funciones que ejecutan los pulsos de lectura mediante el protocolo 1-Wire y devolvió el valor de temperatura en grados centígrados.

```
ds18b20_requestTemperatures();  
temp1 = ds18b20_getTempC((DeviceAddress *)tempSensors[0]);  
temp2 = ds18b20_getTempC((DeviceAddress *)tempSensors[1]);
```

Estos valores se publicaron en los tópicos correspondientes mediante protocolo MQTT. La temperatura del sensor de temperatura del evaporador en el tópico temperatura/evaporador. Ídem para el sensor de temperatura ambiente.

2.4.9 Lectura sensores corriente.

Para la lectura de los sensores de corriente, se ejecutaron dos tareas anidadas, es decir una tarea que se ejecutó dentro de otra tarea. De esta forma controlé dos ciclos diferentes, uno para generar el pulso de muestreo y otro para dar inicio y fin al ciclo del muestreo. Para ello generé un par de contadores como variables locales e inicié timers para la ejecución de las tareas mencionadas. Además establecí tres banderas locales para identificar la presencia de conexión de los sensores mediante el circuito explicado en la Figura 37 - Detección Presencia Sensor.

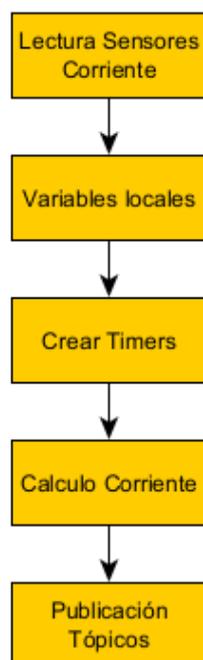


Figura 61 - Lectura Sensores Corriente

Definé las variables locales, inicié el generador de pulso de muestreo a 5KHz al mismo tiempo que inicié el ciclo de muestreo. En cada interrupción el ESP llamó a la función lectura de datos de los pines analógicos y los guardó en un vector. Cuando el ciclo de muestreo llegó a completar un período detuvo las dos tareas y realizó los cálculos matemáticos con los datos obtenidos en los vectores.

La matemática implementada calculó en primer lugar el valor de corriente acorde a la resolución del convertor A/D. Seguidamente aplicó la fórmula de cálculo de V_{rms} para valores discretos de señal, la cual es:

$$V_{rms} = \sqrt{\sum_0^n \frac{(i_k)^2}{n-1}}$$

Para convertir estos valores a corriente, lo multipliqué por el factor de escala del sensor utilizado para realizar la medición (5 para el de 5A y 10 para el de 10A).

```
float irms = 0.0;
float value = 0.0;
int aux = 0; // acumulador
for (int i = 0; i < SAMPLES - 1; i++)
{
    aux = vector_samples[i] - vmed;
    value = aux * 0.000805664;
    irms = irms + value * value;
}
irms = sqrt(irms / (SAMPLES - 1));
```

El ESP leyó el estado de los pines de detección de presencia de sensor. En caso de que el sensor estaba conectado, la bandera que indica su presencia se mostró en estado alto y a la salida mostró el valor medido por el sensor. En el caso desconectado, la bandera se mostró en estado bajo y a la salida devolvía el valor de 0.

Toda esta información se publicó por protocolo MQTT a los tópicos correspondientes por la Tabla 15 - Tópicos MQTT.

2.4.10 RSSI.

En telecomunicaciones, el indicador de intensidad de la señal recibida (Received Signal Strength Indication-RSSI) es una medida de la potencia presente en una señal de radio recibida. Este parámetro resulta de interés en la interfaz usuario o Dashboard, por lo que utilicé este valor para dar la información de la intensidad de la red WiFi a la cual está conectado el dispositivo. Para ello el ESP interpretó el valor rssi de la red a la cual está conectada, Y la publicó al bróker mediante el tópico definido en la Tabla 15.

Intensidad RSSI	Nivel de cobertura
0 a -40	Señal Excelente
-40 a -60	Señal Muy buena
-60 a -70	Señal Media
-70 a -80	Baja Cobertura
Inferior a -80	Sin cobertura

Tabla 16- Cobertura RSSI

En la Tabla 16, observamos una tabla comparativa de los diferentes niveles de rssi y el nivel de cobertura de las mismas.

```
char mqtt[33] = "SSID";
wifi_ap_record_t ap;
esp_wifi_sta_get_ap_info(&ap);
snprintf(mqtt, sizeof(mqtt), "%d", ap.rssi);
esp_mqtt_client_publish(client, "wifi/rssi", mqtt, 0, 0, 0);
```

2.4.11 Funciones buscar sensores e invertir direcciones.

Existen dos tópicos a los cuales suscribí el ESP32 buscar/sensores e invertir/sensores. Cuando se presionaron los botones en la interfaz HMI que llamaron a éstos tópicos, inmediatamente se ejecutaron las funciones definidas para cada caso.

Función buscar sensores.

Para la búsqueda de la dirección de un sensor de temperatura, estudiamos la siguiente tabla.

Caso	Condición Anterior	Condición Nueva Op1	Condición Nueva Op2
Caso 1	Sensor Viejo 1	Sensor Nuevo 1	Sensor Viejo 1
	Sensor Viejo 2	Sensor Viejo 2	Sensor Nuevo 1
Caso 2	Sensor Viejo 1	Sensor Viejo 1	Sensor Nuevo 2
	Sensor Viejo 2	Sensor Nuevo 2	Sensor Viejo 2

Tabla 17 - Casos y Condiciones Sensores Viejos y Nuevos

En esta tabla interpretamos que en el caso 1 y 2 de la condición nueva opción 1, el sensor nuevo se direccionó en la posición donde estaba el anterior. Y en las condiciones nuevas opción 2 para los dos casos, los sensores nuevos se ubicaron en la dirección opuesta. Es decir si el sensor viejo 1 estaba tomando temperatura en el evaporador, la nueva va a estar tomando temperatura ambiente y el otro sensor cambiaría su función. Para ello realicé un algoritmo que me permitió identificar si el nuevo sensor se encuentra en la dirección de reemplazo con el anterior o si cambió su ubicación en el vector. En caso de que estuvieran en posiciones cambiadas de lugar, se realizó la inversión de la ubicación de las direcciones en el vector. Caso contrario el sensor nuevo queda en esa posición y se sigue ejecutando el programa de forma normal.

Función invertir sensores.

Esta función lo que hace es invertir la ubicación de las direcciones en el vector *DeviceAddress tempSensors*.

2.5 Interfaz HMI (Dashboard).

Uno de los aspectos centrales de este proyecto es la interfaz de usuario. Para el desarrollo del mismo utilicé la herramienta Node-Red y un servidor en la nube IBM Cloud para poder acceder al mismo desde internet.

2.5.1 IBM Cloud, servicios y NodeRed.

Una vez creada la cuenta en IBM Cloud, verifiqué el mail y accedí a los distintos servicios que provee este servidor en la nube.

Seguidamente inicié una app NodeRed, la cual pude llamar escribiendo su nombre en la paleta de catálogo.

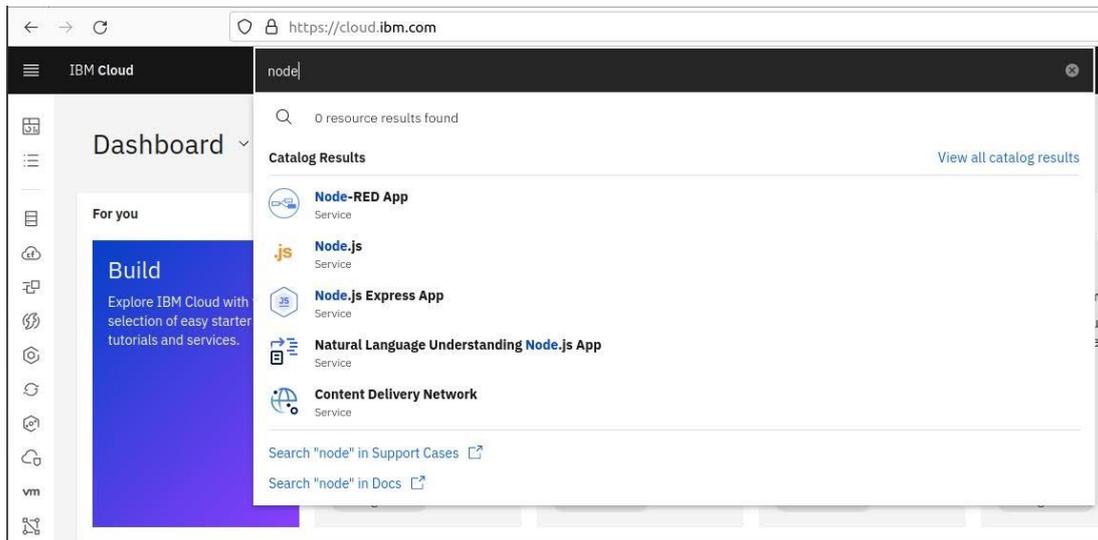


Figura 62 - IBM Catalogo Servicios

Para crear una aplicación NodeRed, ejecuté un servidor en la nube de IBM, instalé un sistema operativo y ejecuté la aplicación.

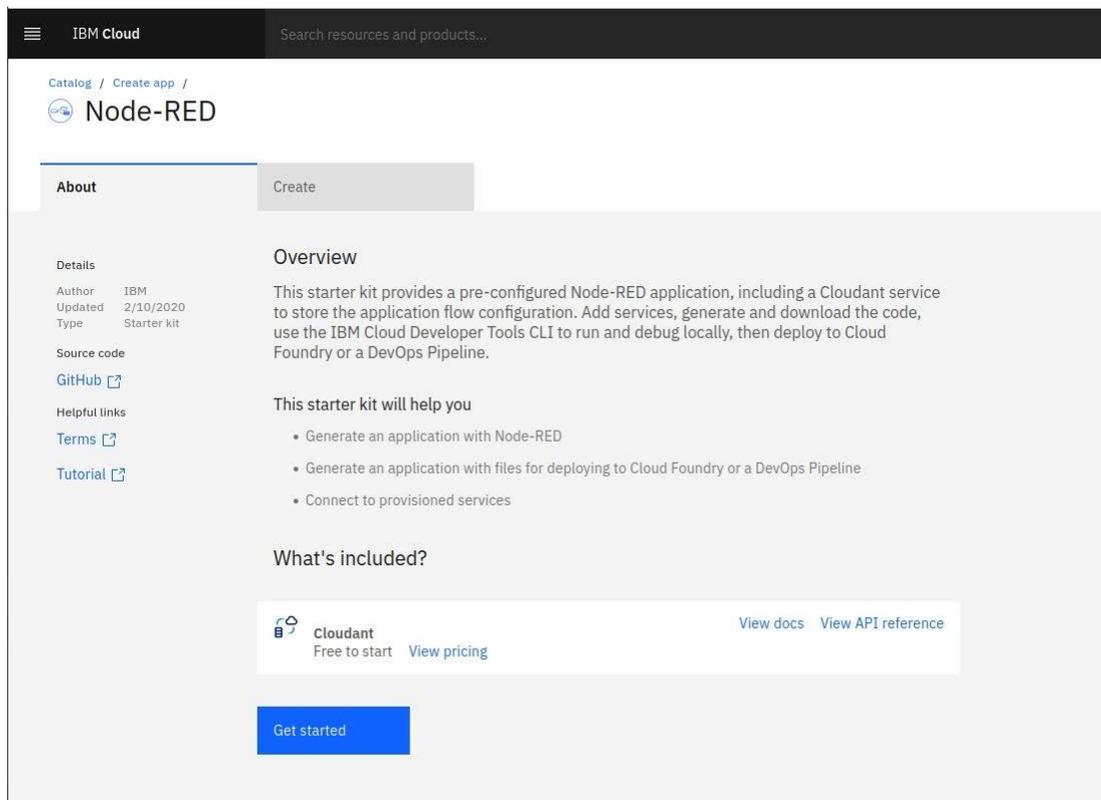


Figura 63 - IBM Configuración NodeRed

Antes de ejecutar el servicio, fue necesario desplegar la aplicación, es decir iniciar el servidor en la nube y ejecutar el sistema operativo. Hice clic en *Deploy your app*.

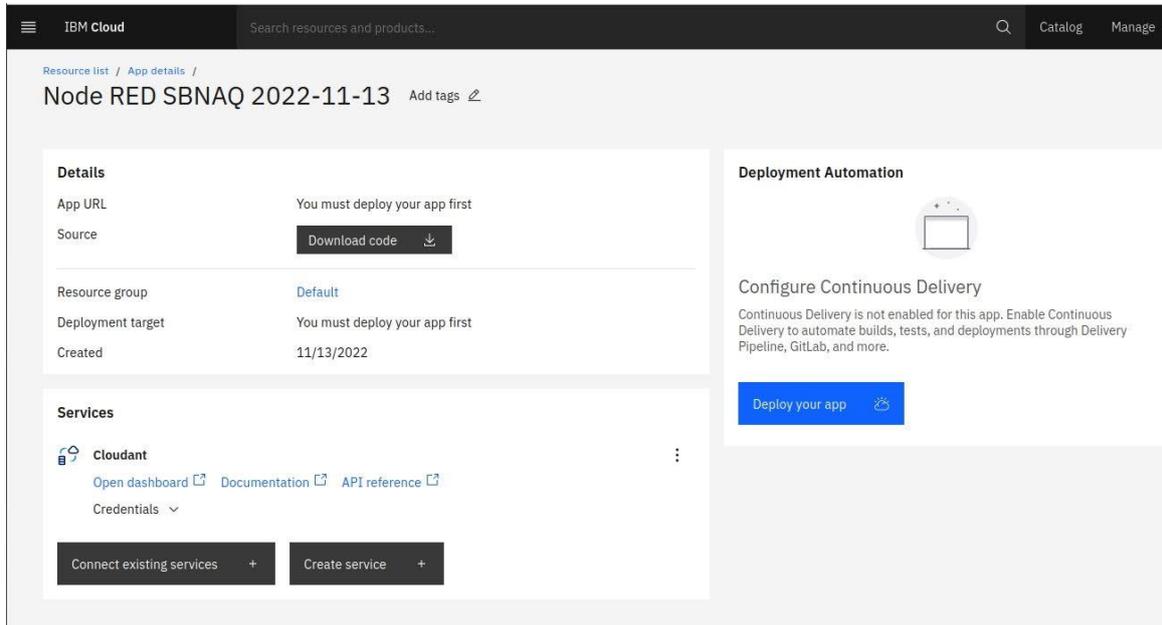


Figura 64 - IBM Detalles Aplicación

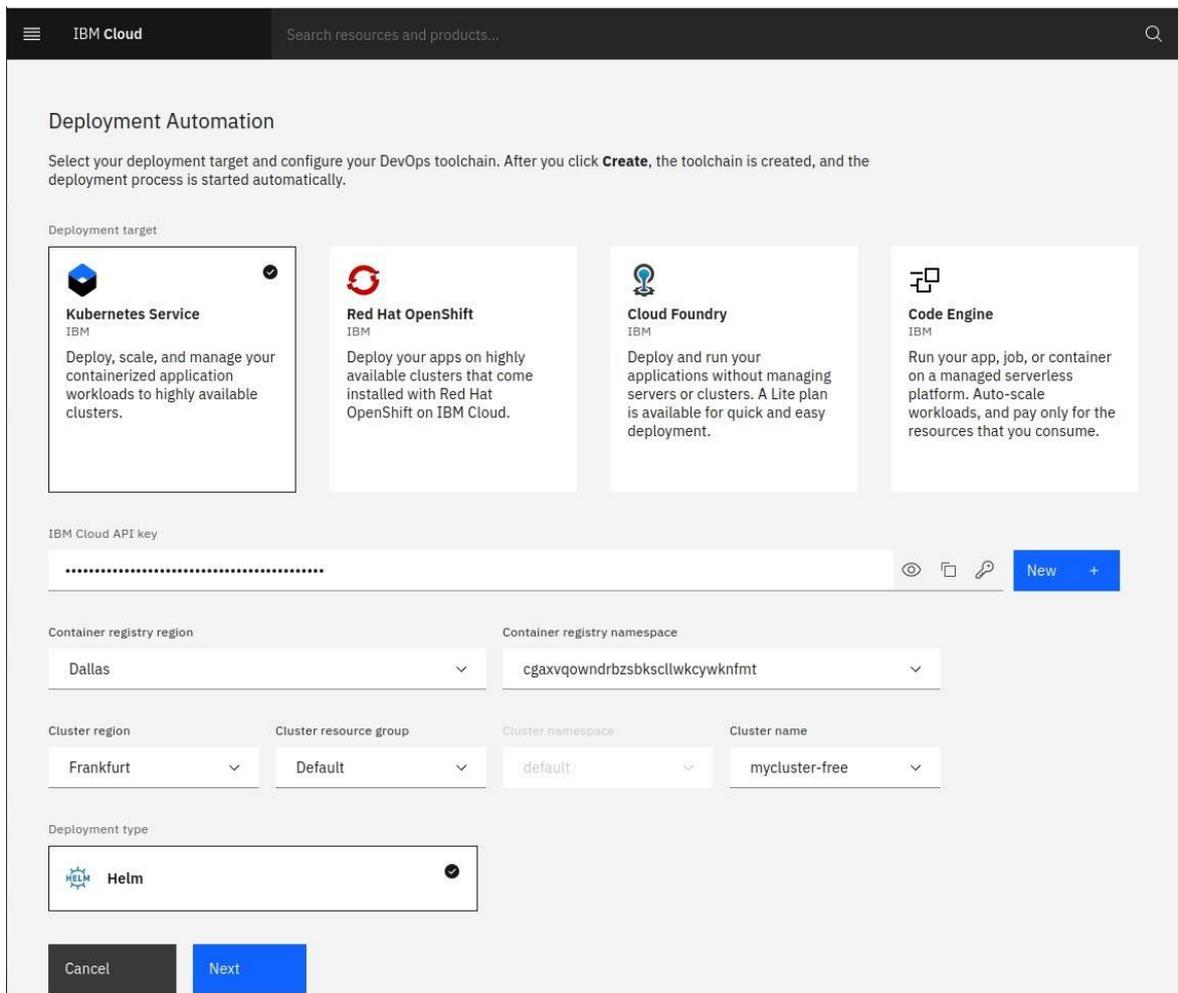


Figura 65 - IBM Automatización de la implementación

La implementación de la aplicación NodeRed puede realizarse a través de cualquiera de los cuatro servicios que se encuentran en el panel de la Figura 65. Elegí *Kubernetes Service*, ya que me permitió ejecutar de forma gratuita la aplicación.

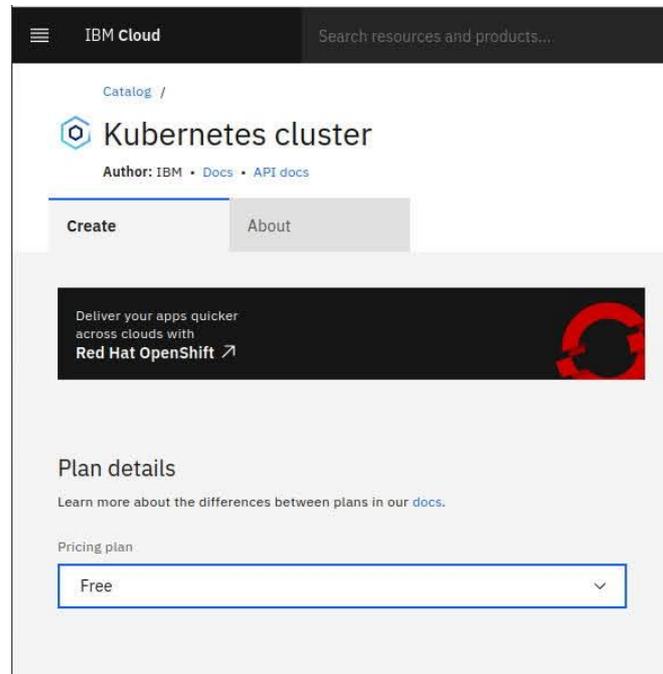


Figura 66 - IBM Clúster

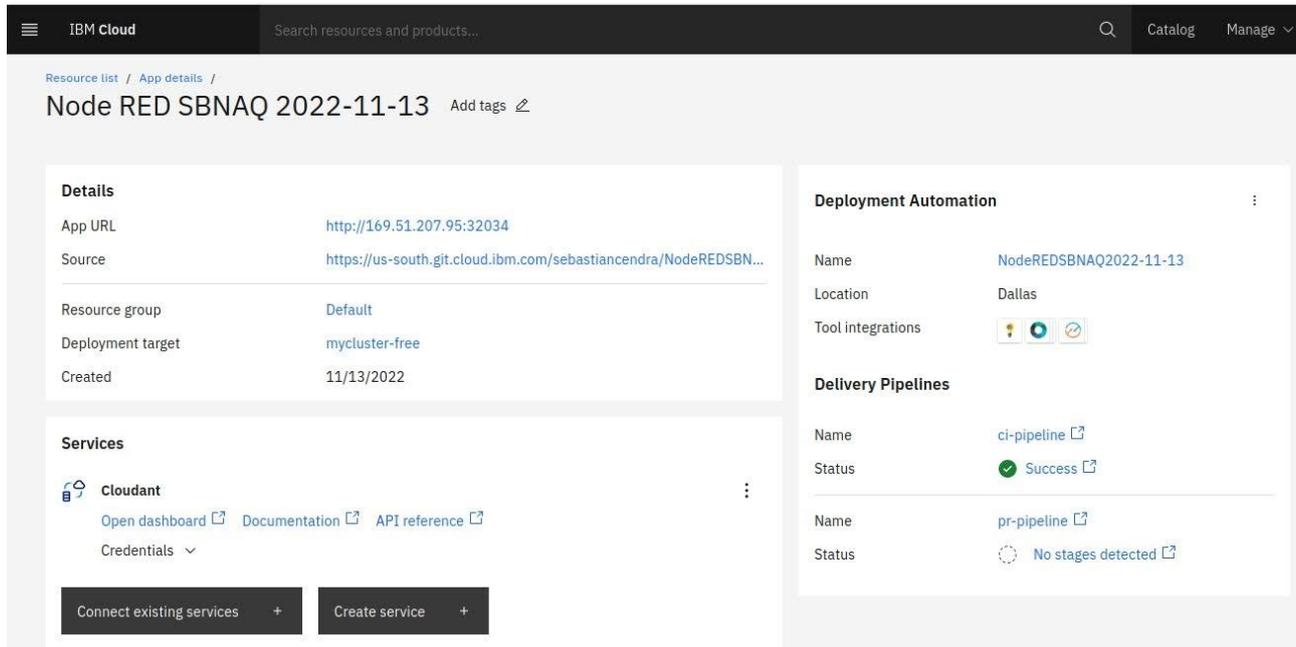


Figura 67 - IBM Despliegue finalizado

Una vez finalizado el despliegue de la aplicación, IBM proporciona una IP pública donde se encontró la aplicación NodeRed.

Ingresé a esa dirección IP, y realicé las configuraciones finales para comenzar a programar en NodeRed. Configuré usuario y contraseña de acceso a la aplicación.

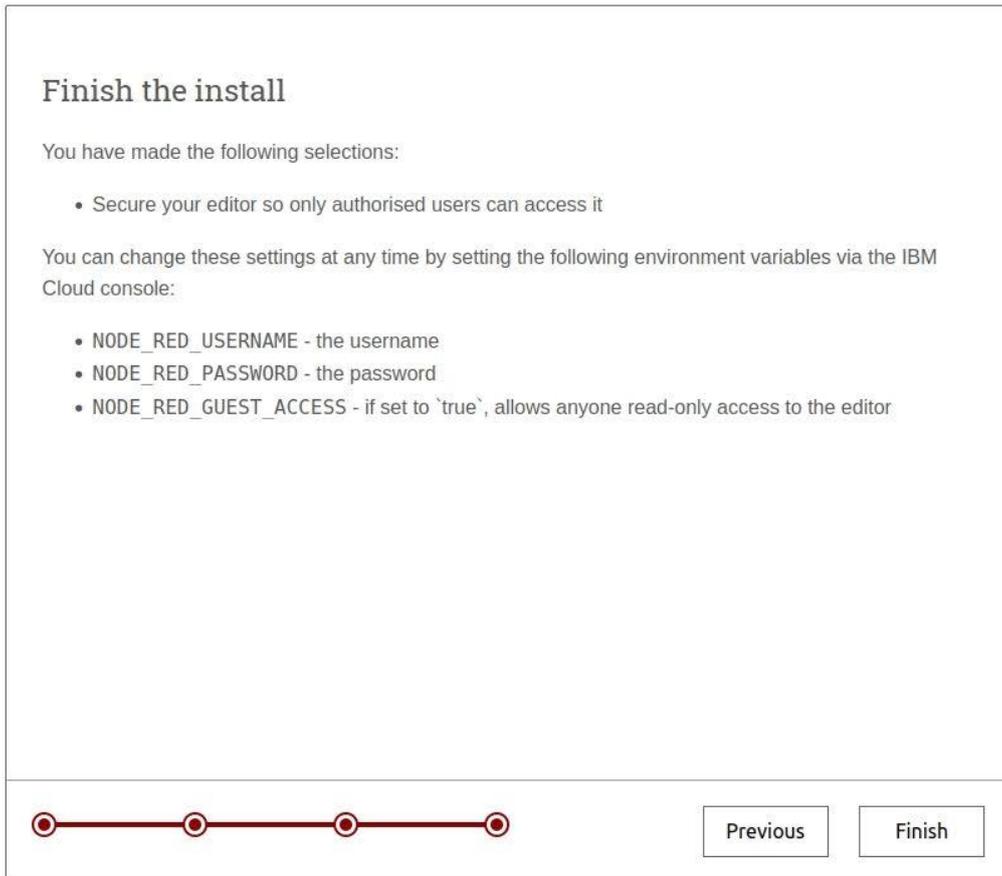


Figura 68 - NodeRed Instalación

Una vez instalado NodeRed, procedí a realizar la programación del Dashboard.

En el espacio de trabajo agregué nodos *mqtt in*, *mqtt out*, *gauge*, *chart*, *LED*, *text*, *button* y *function*. Lo primero que hice fue configurar el Bróker MQTT. Para ello ingresé en las propiedades de un nodo *mqtt in* o *mqtt out*.

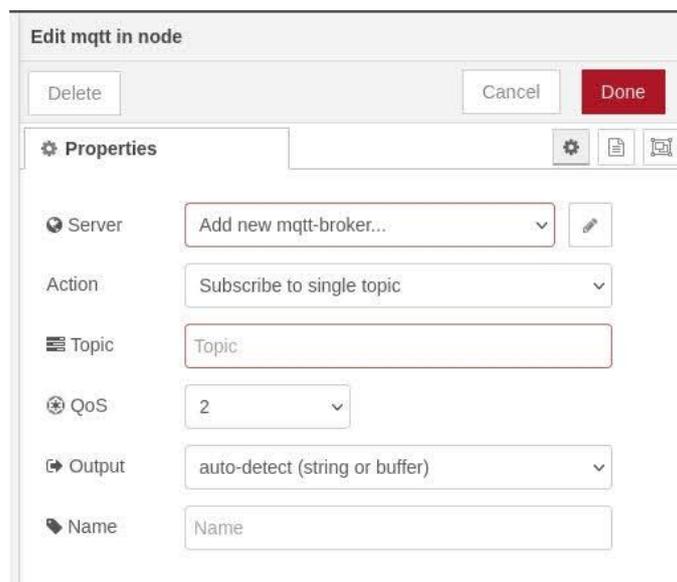


Figura 69 - NodeRed Propiedades Nodo MQTT

Agregué un nuevo Bróker MQTT, definí un nombre (HiveMQ, en este caso) y cargué todas las credenciales que obtuve en HiveMQ (Figura 56 - HiveMQ Credenciales Cluster). En *TLS configuration* cargué además el certificado SSL que generé anteriormente. Con todos estos datos ingresados, pude utilizar esta configuración llamándola con su nombre en cualquier nodo mqtt.

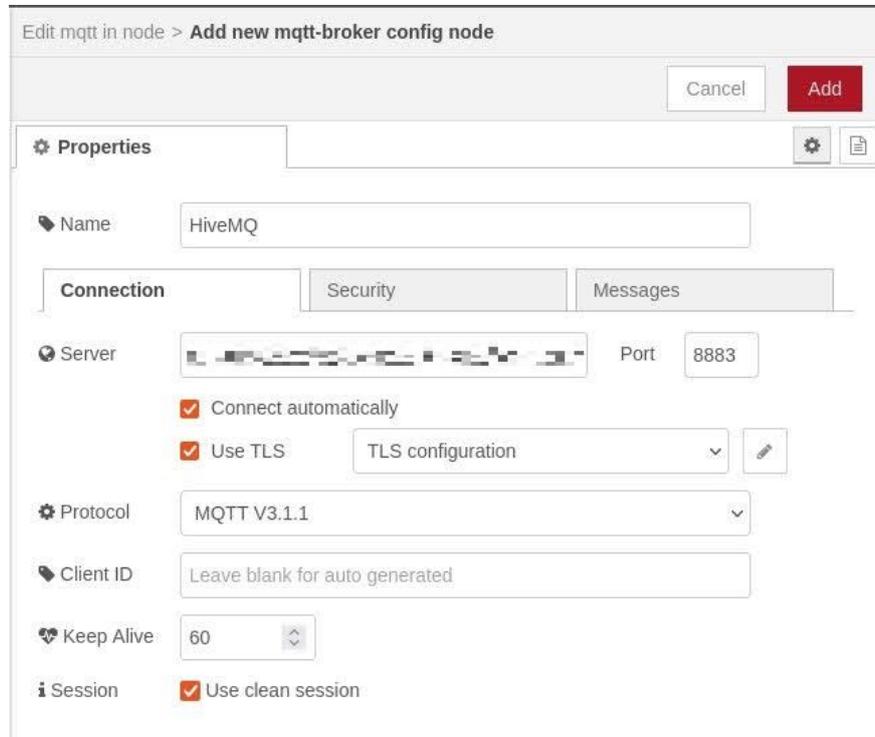


Figura 70 - NodeRed Nuevo Bróker MQTT

En las propiedades cargué el tópicos correspondiente a la función a realizar. Los tópicos elegidos para el diseño se encuentran en la Tabla 15.

Separé la programación del Dashboard en cuatro grupos y tres tabs. Un tab es una ventana en la interfaz usuario. El primer tab (Figura 73) es el Dashboard principal. En él cargué indicadores instantáneos de la señal de los sensores, una gráfica histórica, el estado de conexión de los mismos y el estado de la red WiFi. En el segundo tab (Figura 74) agregué información respecto a los sensores de temperatura. Las direcciones de los mismos, y dos botones para llamar a las funciones de buscar sensores e invertir la ubicación de las direcciones. Y en el tercero, agregué gráficas de los datos históricos pero de un tamaño mayor, es decir con más detalles en su información.

El programa completo podemos verlo en la Figura 71. En el mismo se pueden diferenciar los cuatro grupos que se hicieron para la programación de todo el Dashboard.

En el primer grupo hay cinco nodos *mqtt in*, donde agregué el Bróker *HiveMQ* configurado anteriormente y el tópicos correspondiente a cada sensor. De esta forma cada nodo se suscribió al Bróker MQTT solicitando la información que está publicando el ESP. La información de cada nodo mqtt se reflejó tanto en un *gauge* (indicador en forma de aguja), el cual mostró el valor instantáneo. Y además en un *chart*, el cual almacenó todos los datos que provienen del Bróker y los mostró en un gráfico temporal, que me permitió tener una visualización de los datos históricos.

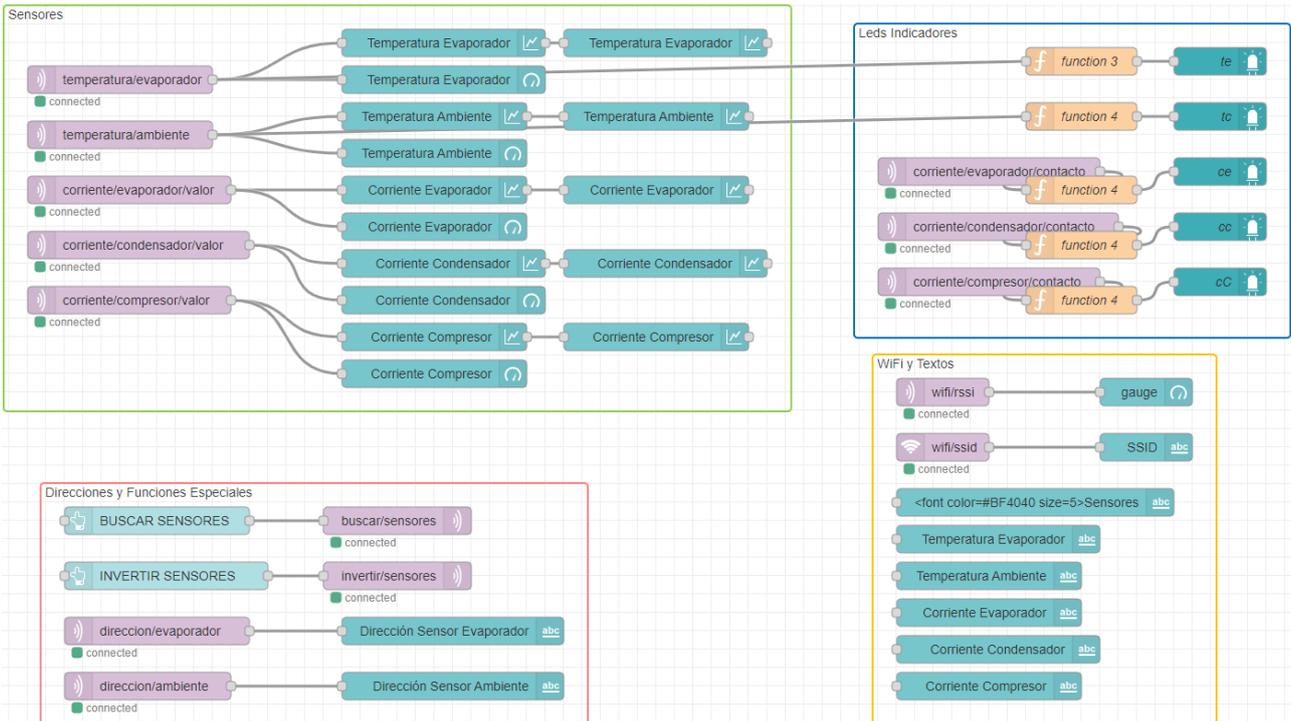


Figura 71 - NodeRed Programa Completo

En el segundo grupo están los LEDs indicadores. Para los sensores de temperatura, no tuve que hacer un detector de presencia, a diferencia de los sensores de corriente, ya que al ser digitales pude detectar si estaban conectados o no. Los valores que muestra el ESP cuando están desconectados es de -222°C, y por hoja de datos el valor más bajo que miden de temperatura es -55°C. Entonces utilicé el nodo *function* para hacer la conversión de los valores que provienen del Bróker y determinar si están o no conectados los sensores de temperatura. Dentro del nodo *function*, se programó una sentencia if.

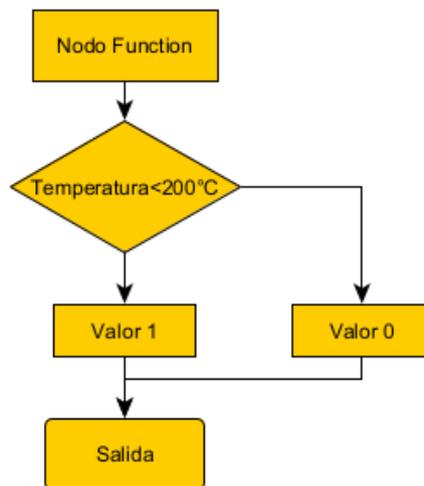


Figura 72 - NodeRed Programa del Nodo Function

Los sensores de corriente, tienen un tópico dedicado a la información del estado de conexión de los mismos provenientes del ESP32. Estos datos se obtienen directamente suscribiéndose a los tópicos designados.

Estos cinco sensores tienen un LED indicador en el Dashboard. Si el estado es 1, el LED es de color verde y en caso contrario el LED es de color rojo. Para acompañar al diseño y estética, agregué unos nodos *text*, donde identifiqué con sus nombres a cada uno de los sensores (grupo WiFi y Textos de Figura 71).



Figura 73 - NodeRed Dashboard

Recién hice mención al tercer grupo *WiFi* y *Textos*, y expliqué los textos que agregué al diseño del grupo de LEDs indicadores. También en esta misma sección agregué la información de SSID y RSSI de la red WiFi para conocer el nombre y estado de la intensidad de la red a la cual está conectado el microcontrolador. Para ello hice uso de dos nodos *mqtt in*, un *gauge* para la intensidad de señal y un *text* para el nombre de la red WiFi. Figura 71.



Figura 74 - NodeRed Sensores Temperatura

Por ultimo está el grupo Direcciones y Funciones Especiales. En el mismo existen dos nodos *mqtt in*, dos nodos *mqtt out*, dos nodos *text* y dos *button*. Mediante los *button*, al presionarlos, genero una publicación al tópico buscar/sensores o invertir/sensores. El cual llama y ejecuta las funciones en el ESP. El retorno de esa información, es decir la dirección y ubicación de los sensores de temperatura, la pude ver reflejada en los siguientes nodos *text*, que reciben la información de los tópicos dirección/evaporador y dirección/ambiente respectivamente. La visualización en pantalla la podemos ver en la Figura 74.

2.6 Diseño completo.

Luego de analizar todos los bloques y el funcionamiento de cada etapa, explico a continuación el diseño final del sistema de monitoreo, la construcción y detalles de armado.

Como vimos antes el circuito planteado en la Figura 31 - Esquemático Principal, tengo el esquema final diseñado para el proyecto. Pase del diagrama esquemático al diseño del PCB, y logre el siguiente resultado:

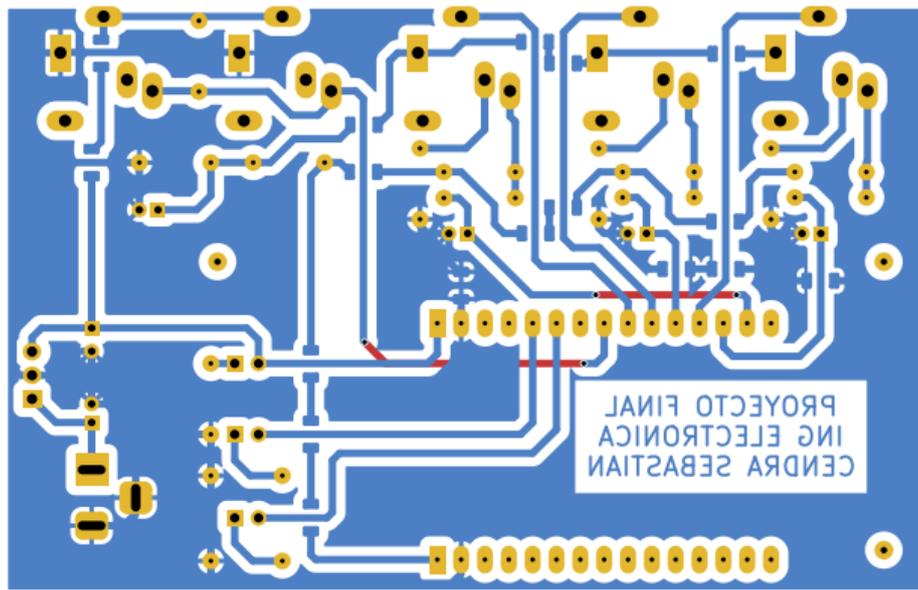


Figura 75 - Diseño PCB

Para la construcción del PCB utilicé una placa de pertinax de 10cm por 10cm (las medidas reales son 10cm por 6.5cm, donde el material sobrante se cortó). El método para la impresión de las pistas que se utilizó fue el grabado láser (Figura 76). Se aplicó una capa de pintura con un aerosol, y mediante un cnc de grabado láser se imprimió el diseño de las pistas (Figura 77). Luego el pertinax se sumergió en ácido percloruro férrico y de esta forma se eliminó el material indeseado, logrando el circuito diseñado (Figura 78).

Seguidamente se soldaron todos los componentes, como se ve en la Figura 79.

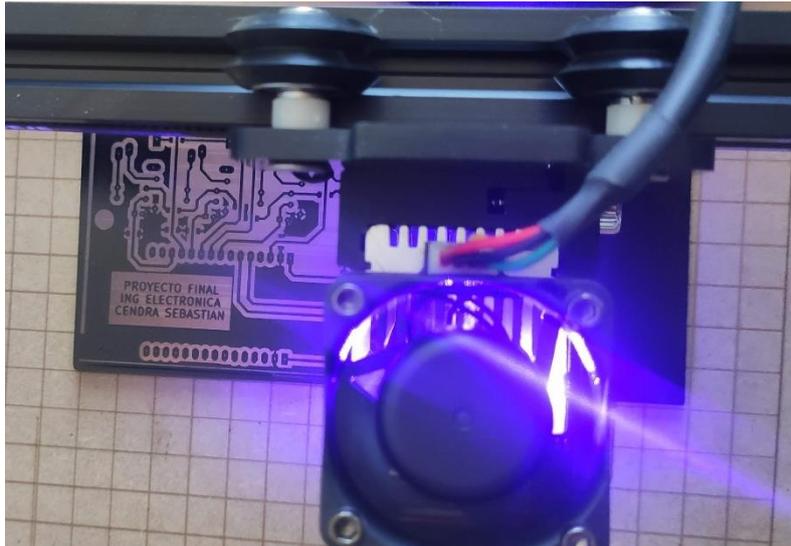


Figura 76 - PCB Grabado Laser

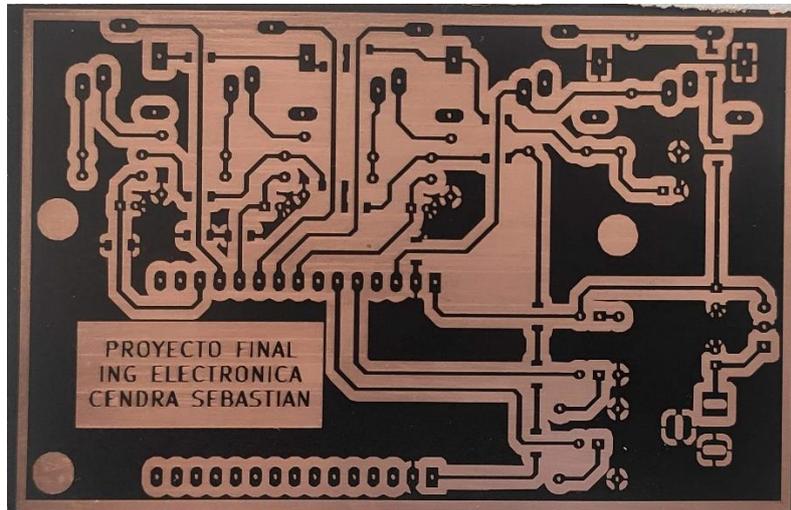


Figura 77 - PCB Impreso

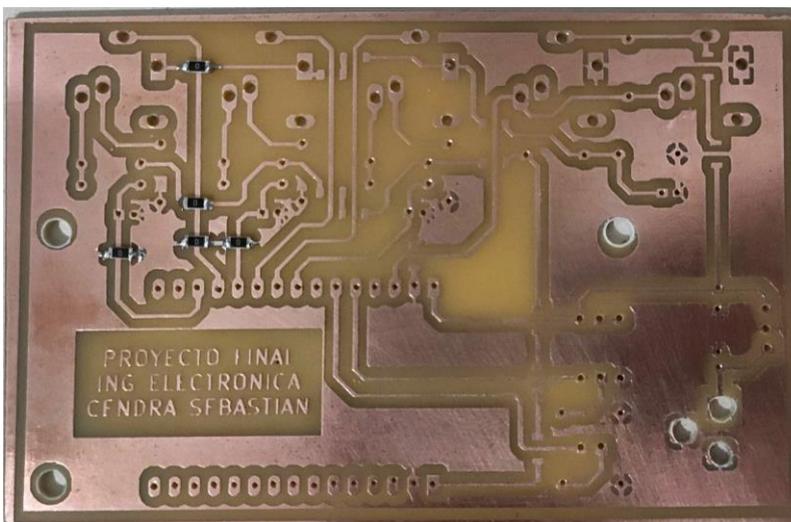


Figura 78 - PCB Pasado por Percloruro Férrico

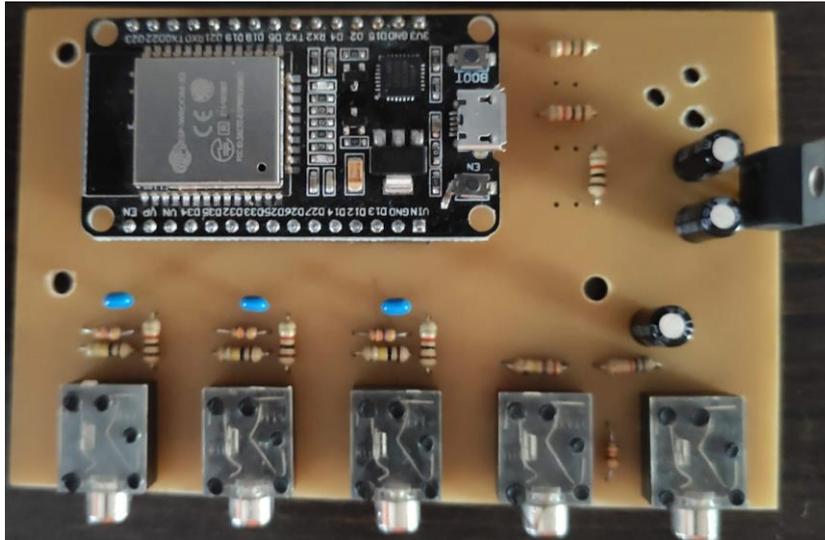


Figura 79 - PCB Componentes Soldados

Fue necesario prescindir de un gabinete donde instalar el PCB. Comercialmente conseguí un gabinete de plástico inyectado de 12cm de largo por 7,5cm de ancho y 5cm de alto. Busqué ubicar el PCB de la mejor manera, de forma tal que las entradas de los sensores quedaran de frente hacia el usuario, los LEDs en la parte superior y la fuente de alimentación en la parte posterior. Luego de su instalación quedo de la siguiente manera.

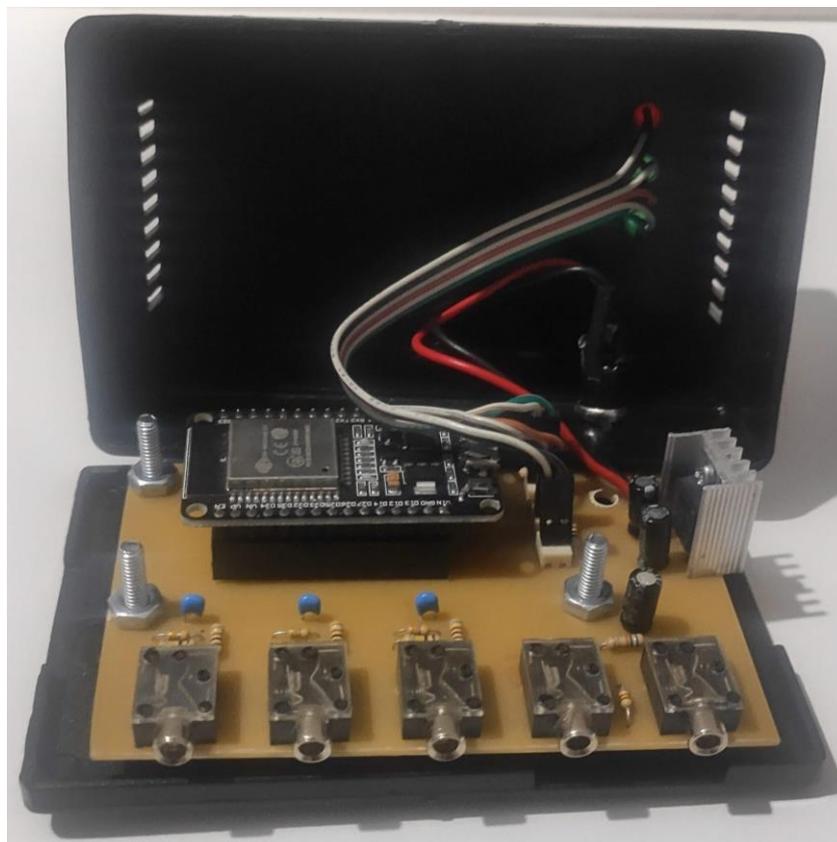


Figura 80 - PCB Instalado en Gabinete Abierto



Figura 81 - PCB Instalado en Gabinete Cerrado

2.7 Instalación y modo de uso.

Para la instalación del sistema de monitoreo de equipos de frío requerí:

- 1 dispositivo de monitoreo de equipos de frío.
- 1 cámara frigorífica.
- 2 Sensores DS18B20
- 2 Sensores SCT-013-005
- 1 Sensor SCT-013-010
- 1 Fuente de alimentación de 12V CC.
- Conexión a internet, mediante WiFi (para el dispositivo)
- Una pc o dispositivo móvil con conexión a internet (para el usuario)

2.7.1 Paso 1. Estudio Cámara Frigorífica.

Se consiguió una cámara frigorífica con características diferentes a las diseñadas para realizar mediciones y pruebas de funcionamiento, pero los parámetros medidos se encontraron dentro del rango de escalas implementados. El equipo sobre la cual se realizaron las pruebas es una cámara de media temperatura conservación de cortes porcinos, vacunos y elaborados. Características:

- Motor compresor Blue Star Q44 de 2HP.

	Modelos	QR44	QR58
Especificaciones	Capacidad Nominal (W)	5770	7900
	COP (W/W)	3.0	3.1
	Nivel sonoro (SPL) (dB) (A)	≤62	≤65
	Desplazamiento (cc/rev)	44	58.2
Datos del Motor	Potencia de Entrada (W)	2060	2680
	Consumo RLA (A)	10.2	13.5
	Consumo LRA (A)	65	70
	Aislacion	F	
	Revoluciones rev/min	≥2860	
	Alimentación	220-240V/1φ/50Hz	
	Resistencia bobina de arranque Ω (25°C)	1.01	0.69
	Resistencia Bobina de trabajo Ω (25°C)	2.88	2.68
	Capacitor de arranque	88-108uF/330V	
	Capacitor de trabajo	35uF/450V	
	Relay de arranque	Pick Up at:180-195V Drop Out at::40-105V	
	Carga de aceite (Cm3)	1500	
Tipo de Aceite	SUNISO 3GS		
Peso Neto (aceite lubricante incluido) Kg	33	34	
Refrigerante	R22		

Tabla 18 - Características Eléctricas Compresor

Fuente: <http://mail4.ansal.com.ar/Documentacion/pdfs/906304.pdf> - Pág.4

La información que necesité saber de la tabla del motor es el consumo RLA. El cual es 10.2A. Como vimos en 2.3 Propuestas de circuitos o esquemas, el divisor resistivo me dió un offset en 1.514V, esto equivale a un voltaje rms máximo de entrada de 1.07V. Traducido a la pinza SCT-013-010, es una entrada máxima de 10.7A. Se podría reemplazar por una pinza de 15A, pero realicé las mediciones con los elementos que había adquirido. Otro dato a considerar que el consumo RLA depende de condiciones normales de presión y temperatura a la que está sometido.

- Motor evaporador: Forzador axial pala reja de 350mm. El modelo instalado es WG-YWF4D-350 monofásico. Vemos en la Tabla 18 que la corriente nominal es de 0.65A.
- Motor condensador: Forzador axial pala reja de 400mm. El modelo instalado es WG-YWF4D-400 monofásico. Vemos en la Tabla 19 que la corriente nominal es de 0.82A.
- Por características constructivas de instalación de la cámara solamente fue posible medir las corrientes de los motores pero no en forma independiente. El contactor que alimenta el compresor, alimenta también el condensador (tiene los motores en serie). Es decir que la corriente sensada en el contactor fue la suma de la corriente del compresor y el condensador.
- Temperatura en el interior de la unidad: es un parámetro de diseño y la misma está definida como $T_{in} > 0^{\circ}\text{C}$. Es decir la mínima es 0°C .

MOTORES ELÉCTRICOS REFRIGERACIÓN



Forzadores de Rotor Externo YWF Series Ø250 a 710mm / 220V & 380V

Motores para Refrigeración

Forzadores de Rotor Externo

YWF Series, Aspirantes, Monofásicos 220V/1F/200V & Trifásicos 380/3F/50Hz

Cod. SAP	Fases	Tensión	Frec.	Polos	Cap. µF	Datos Eléctricos			Caudal m3/h	Tipo	dBA	Peso Kg
						Conex.	Watts	Amp.				
WG-YWF4E-250	1	220V	50Hz	4	2		60	0.23	730	Aspirante	55	3
WG-YWF4E-300	1	220V	50Hz	4	3		75	0.35	1620	Aspirante	58	3.3
WG-YWF4E-350	1	220V	50Hz	4	4		140	0.65	2270	Aspirante	62	4.9
WG-YWF4D-350	3	380V	50Hz	4	NA		150	0.38	2290	Aspirante	62	4.9
WG-YWF4E-400	1	220V	50Hz	4	6		180	0.82	3400	Aspirante	68	7.2
WG-YWF4D-400	3	380V	50Hz	4	NA		180	0.47	3400	Aspirante	68	7.2
WG-YWF4E-450	1	220V	50Hz	4	8		250	1.20	4620	Aspirante	68	7.2
WG-YWF4D-450	3	380V	50Hz	4	NA		250	0.60	4620	Aspirante	68	7.2
WG-YWF4E-500	1	220V	50Hz	4	10		420	1.95	6420	Aspirante	72	9.5
WG-YWF4D-500	1	220V	50Hz	4	NA		450	0.90	6570	Aspirante	72	9.5
WG-YWF4D-630	3	380V	50Hz	4	NA		880	1.60	11600	Aspirante	78	15
WG-YWF4D-710	3	380V	50Hz	6	NA	Triang.	900	1.75	12750	Aspirante	80	25
						Estr.	650	1.20	12750			



Tabla 19 - Motor Forzador Evaporador

Fuente: <https://www.totaline.com.ar/wp-content/uploads/2016/08/41-Flyer-Forzadores-Axiales-YWF-Weiguang.pdf>

2.7.2 Paso 2. Instalación.

El diagrama de conexiones podemos verlo en la Figura 82.

Instalé los sensores de corriente (atención tener cuidado si el equipo está con corriente, para mayor seguridad realizar la instalación con el equipo desenergizado). Abrí las pinzas de los SCT-013 y enlacé un solo conductor (la fase) que alimenta el motor o grupo de motores. Figura 83.

Conecté los sensores de temperatura uno en el interior de la unidad de frío y el otro fuera del equipo. Llevé los cables hacia el dispositivo. Y conecté los jacks en su ubicación correspondiente. Figura 84.

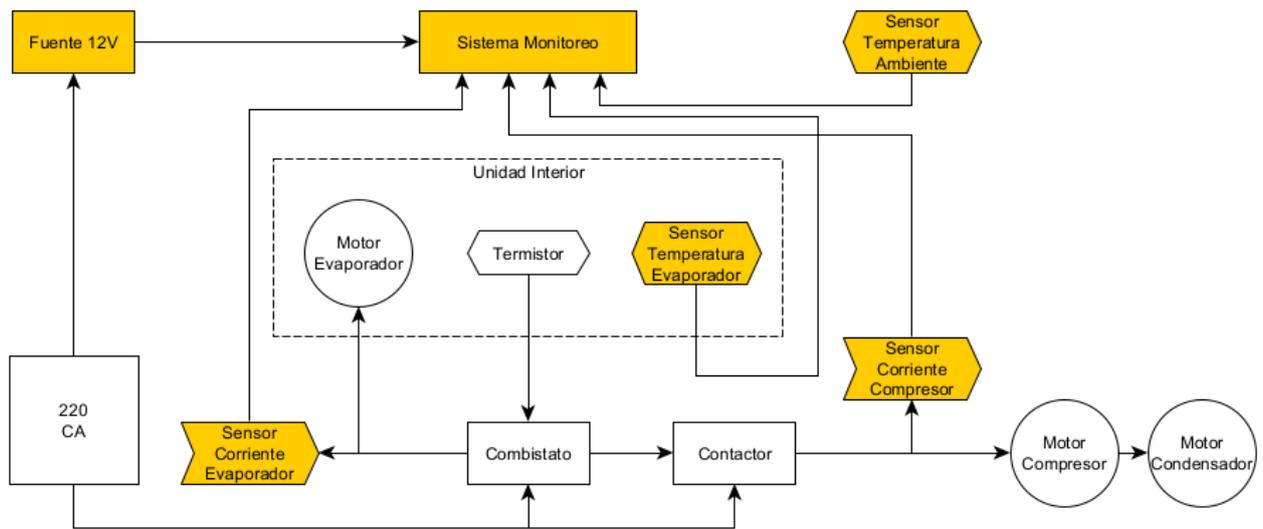


Figura 82 - Diagrama de Conexiones



Figura 83 - Instalación Sensores Corriente



Figura 84 - Instalación Sensores Temperatura

2.7.3 Paso 3. Puesta en funcionamiento.

Una vez instalado el equipo y sus respectivos sensores, procedí a energizar el dispositivo. Conecté el transformador de 12VCC a 220V CA.

El LED Rojo, indica presencia de energía en el proyecto.



Figura 85 - Puesta en funcionamiento

Durante la primera puesta en funcionamiento, el dispositivo debió conectarse a la red WiFi local. El LED verde con la leyenda *WIFI*, estuvo encendiéndose de forma intermitente hasta que conecté el equipo a la red WiFi Figura 85. Para ello en un dispositivo móvil instalé la siguiente aplicación desde el PlayStore: ESPTouch SmartConfig. Figura 51.

Tuve que conectar el dispositivo móvil a la red WiFi local, a la antenna 2.4GHz (no es compatible con antenna 5.8GHz).

Abrí la app, escribí la contraseña de la red WiFi en la que estaba conectado y toqué el botón de confirmar. Figura 52.

Cuando el LED verde *WIFI*, quedó encendido, comenzó la conexión con el Bróker MQTT. Durante este proceso el LED verde *MQTT* se encendió en forma intermitente hasta que finalizó la conexión con el Bróker. Figura 86.

Una vez conectado el Bróker, el LED verde *MQTT*, quedó encendido de forma permanente y los sensores comenzaron a reportar datos al Bróker.



Figura 86 - Dispositivo Conectado a Internet y Bróker MQTT

2.7.4 Paso 4. Dashboard e identificación de sensores de temperatura.

Desde una computadora con conexión a internet, ingresé a la siguiente dirección web: <http://169.51.207.95:32034/ui>, con los códigos de acceso configurados en la Figura 68.

En el primer menú está el Dashboard principal. Al hacer clic en las tres barras horizontales ubicadas en la esquina superior izquierda, se despliega un menú donde podemos elegir entre la vista *Dashboard*, *Sensores Temperatura* e *Históricos*.

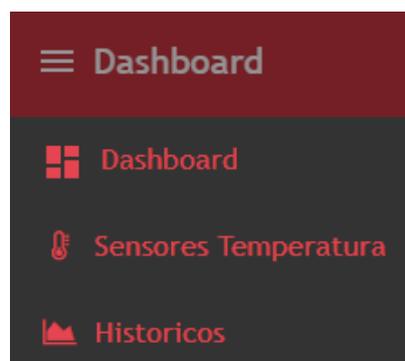


Figura 87 - Dashboard menú

Al ver el estado de las temperaturas del evaporador y ambiente, pude determinar si están correctamente conectados. Ya que la temperatura del evaporador es muy inferior a la ambiente.

En caso que los sensores reportaren la temperatura en forma invertida, se debe ingresar a la vista *Sensores de Temperatura* (Figura 88). Allí presionar el botón *Invertir Sensores* y de esta manera cambian las direcciones de posición. Podemos observarlo en esta misma vista.



Figura 88 - Dashboard Menú Sensores Temperatura

En este punto, ingresé a la vista *Dashboard* y observé la información que entregó el dispositivo a través de los sensores.

En el lado izquierdo y superior pude observar la información correspondiente a la red WiFi (Figura 89). El nombre de la red WiFi (ssid) a la cual estaba conectado el dispositivo y su potencia (rssi). La interpretación de los distintos niveles de señal, la podemos ver en la Tabla 16.



Figura 89 - Dashboard Estado WiFi y Sensores

En la parte central existen dos indicadores temporales (*Realtime Temperatura*) de la temperatura del evaporador, ambiente y dos gráficos históricos (*Histórico Temperatura*). Figura 90.

En la parte izquierda existen tres indicadores temporales (*Realtime Corriente*) de las corrientes del compresor, evaporador y condensador respectivamente. Y además sus correspondientes gráficos históricos (*Histórico Corriente*). Figura 91.

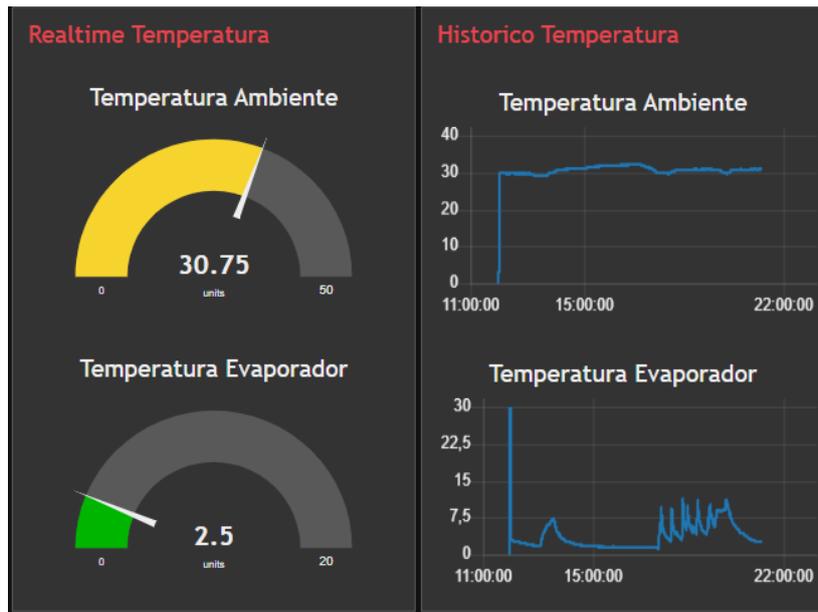


Figura 90 - Dashboard Realtime e Histórico Temperatura

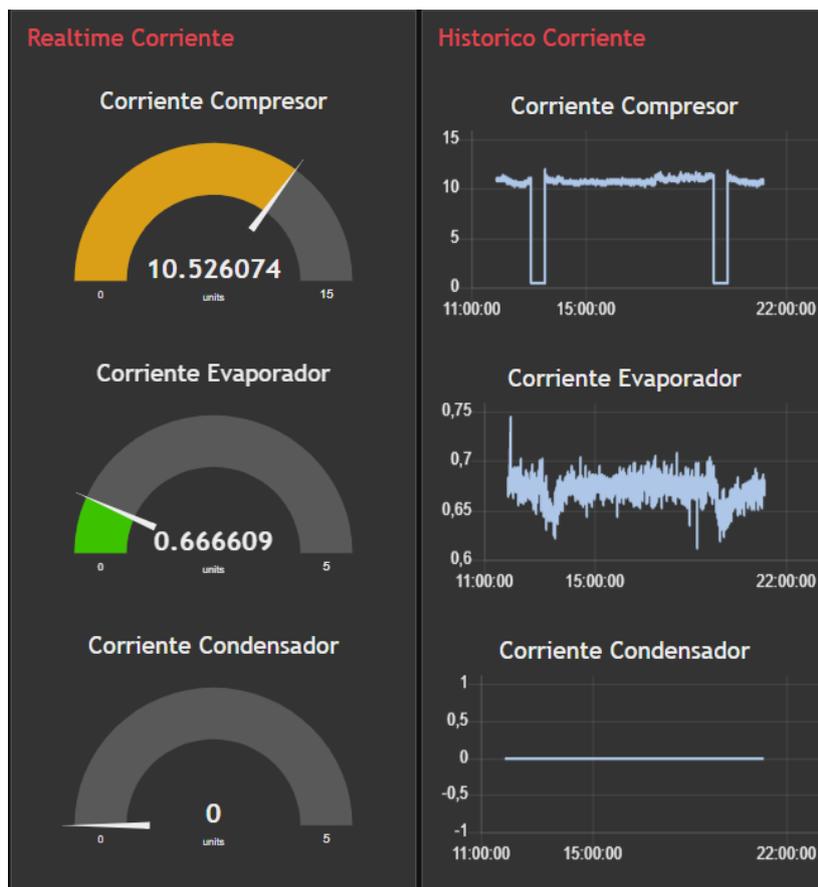


Figura 91 - Dashboard Realtime e Histórico Corriente

Hice además una tercera vista más ampliada de los datos históricos para tener más detalle de la información.

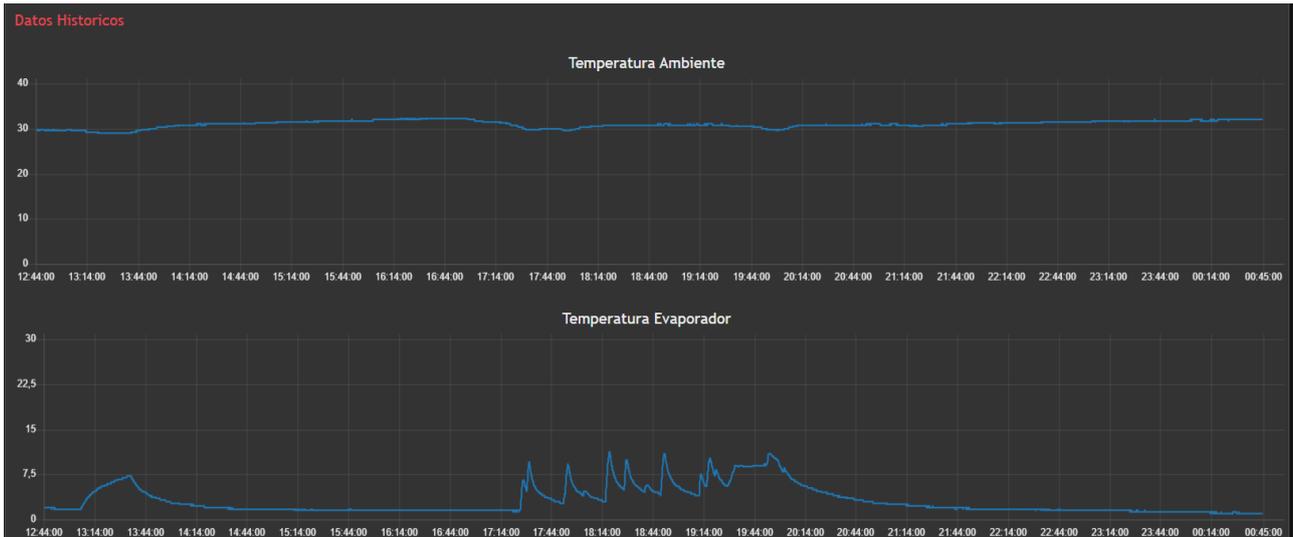


Figura 92 - Resultados Detalle Histórico Temperatura

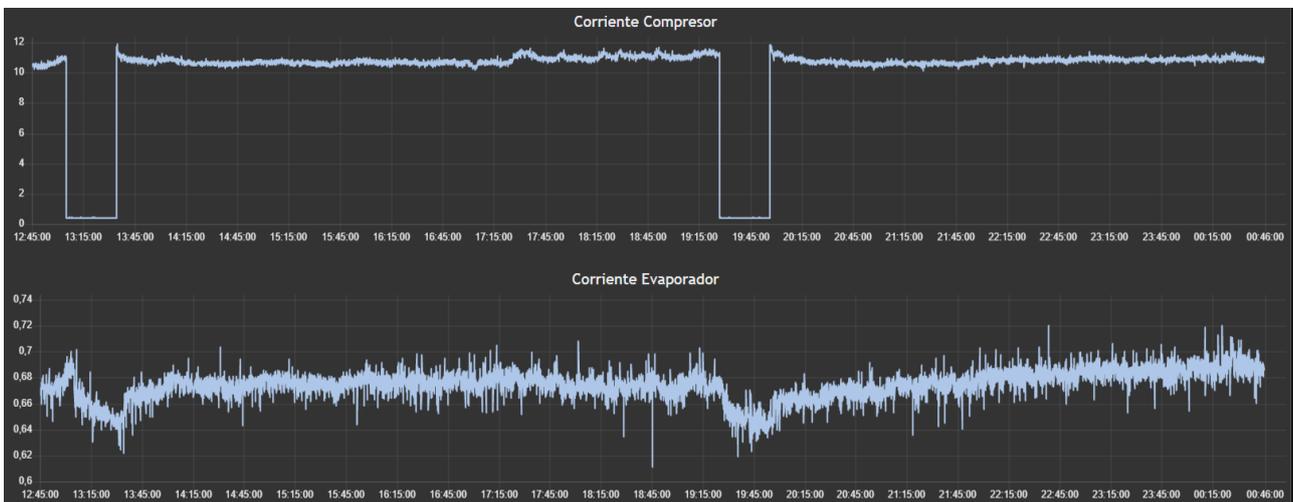


Figura 93 - Resultados Detalle Histórico Corrientes

2.8 Clase del instrumento.

Durante la puesta en funcionamiento del presente proyecto, tomé mediciones con instrumentos que fueron considerados patrón.

Para las mediciones de corriente, utilicé un téster UNIT-T, modelo UT210D, el cual dispone una pinza amperométrica para medir corrientes. Los datos obtenidos fueron los siguientes.



Figura 94 - Medición Corriente Motor Evaporador

Muestra	Corrientes Instrumento Patrón [A]	Corrientes SCT-013-005 [A]	Error [A]
1	0,71	0,8	-0,09
2	0,71	0,81	-0,10
3	0,70	0,79	-0,09
4	0,71	0,8	-0,09
5	0,70	0,82	-0,12
6	0,71	0,8	-0,09
7	0,70	0,81	-0,11
8	0,70	0,79	-0,09
9	0,70	0,8	-0,10
10	0,71	0,8	-0,09

Tabla 20 - Corriente SCT-013-005 vs Instrumento Patrón

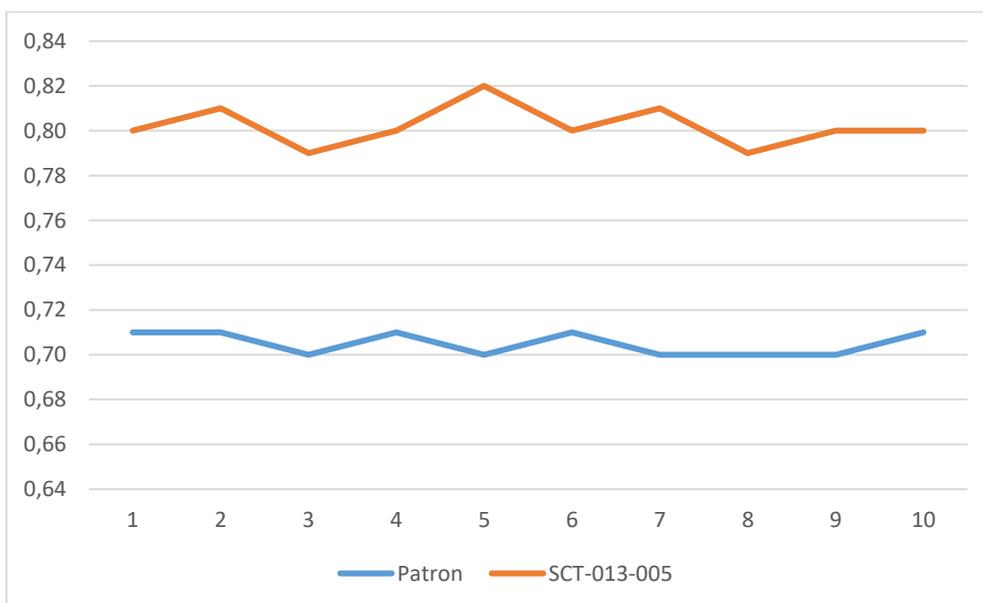


Figura 95 - Corriente SCT-013-005 vs Instrumento Patrón

Las mediciones obtenidas en la Tabla 20, arrojaron resultados con un error máximo de 120mA. Calculando la clase del instrumento:

$$c\% = \pm \frac{\Delta x_{max}}{Alcance} * 100$$

$$c\% = \pm \frac{0.12A}{5A} * 100$$

$$c\% = \pm 2.4\%$$

Por los resultados obtenidos, lo mejor hubiera sido haber utilizado el modelo SCT-013-001 para el motor del evaporador, pero se realizaron mediciones con los materiales diseñados.



Figura 96 - Medición Corriente Compresor

Muestra	Corrientes Instrumento Patrón [A]	Corrientes SCT-013-010 [A]	Error [A]
1	11,08	11,09	-0,01
2	11,12	11,10	0,02
3	11,10	11,12	-0,02
4	11,06	11,08	-0,02
5	11,10	11,12	-0,02
6	11,12	11,07	0,05
7	11,08	11,08	0,00
8	11,09	11,12	-0,03
9	11,00	11,03	-0,03
10	11,12	11,10	0,02

Tabla 21 - Corriente SCT-013-010 vs Instrumento Patrón

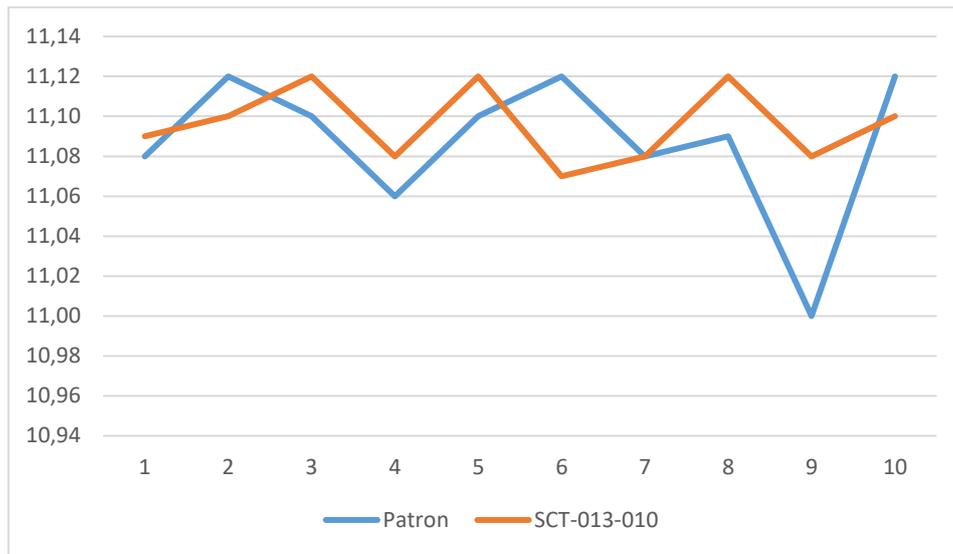


Figura 97 - Corriente SCT-013-010 vs Instrumento Patrón

Las mediciones obtenidas en la Tabla 21, arrojaron resultados con un error máximo de 50mA. Calculando la clase del instrumento:

$$c\% = \pm \frac{\Delta x_{max}}{Alcance} * 100$$

$$c\% = \pm \frac{0.05A}{10A} * 100$$

$$c\% = \pm 0.5\%$$

A medida que las mediciones se acercaron al fondo de escala, los resultados obtenidos con el instrumento patrón se parecieron mucho más a los obtenidos mediante el dispositivo.

Asimismo se realizaron contrastaciones de temperatura con un TESTO 115i (Figura 99). El cual tiene una exactitud de ±1.3°C. La sonda de prueba trabaja con una aplicación dada por el fabricante *testo Smart*. El cual tiene vistas de gráfica o tabla para la presentación de los resultados obtenidos.

Muestra	Temperaturas Instrumento Patrón [°C]	Temperaturas DS18B20 [°C]	Error [°C]
1	3,80	5,5	-1,70
2	3,60	5,25	-1,65
3	3,60	5,25	-1,65
4	3,40	5,5	-2,10
5	25,60	24,5	1,10
6	27,50	25,75	1,75
7	28,20	26,5	1,70
8	28,20	26,5	1,70
9	28,90	27	1,90
10	29,20	27,25	1,95

Tabla 22 - Temperaturas DS18B20 vs Instrumento Patrón

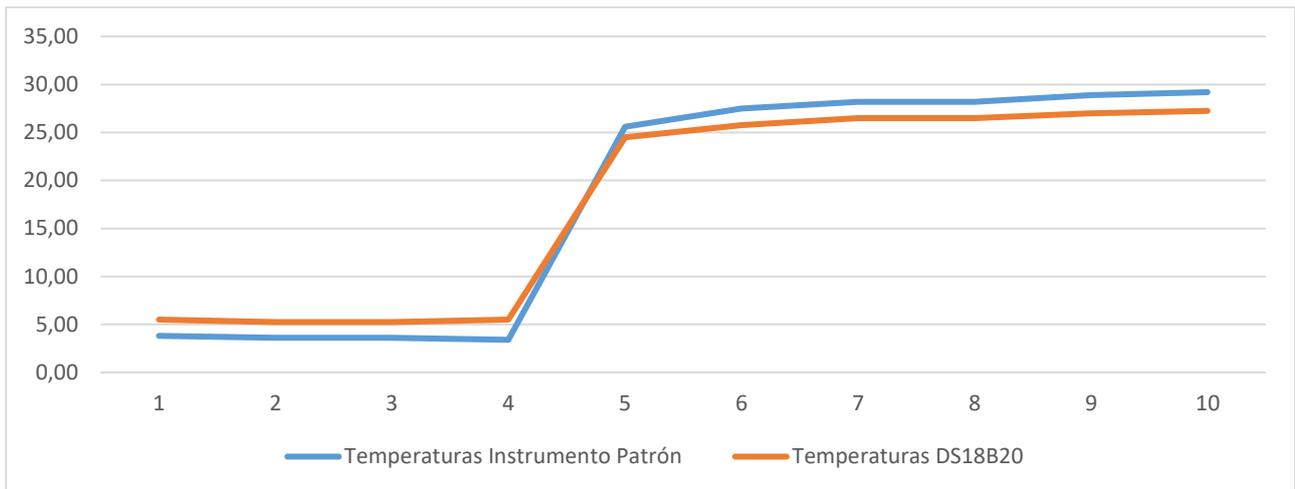


Figura 98 - Temperaturas DS18B20 vs Instrumento Patrón



Figura 99 - Testo 115i

Las mediciones obtenidas en la Figura 97, arrojaron resultados con un error máximo de 2.10°C. Calculando la clase del instrumento:

$$c\% = \pm \frac{\Delta x_{max}}{Alcance} * 100$$

$$c\% = \pm \frac{2.10^{\circ}C}{(100 - (-30))^{\circ}C} * 100$$

$$c\% = \pm 1.61\%$$

Capítulo 3: Resultados

Como resultado fundamental se logró cumplir con los requerimientos fundamentales del planteo del presente proyecto. A lo largo de cada etapa del desarrollo y diseño se realizaron pruebas de campo con los sensores en forma separada, obteniendo los resultados esperados.

Si bien el diseño no plantea un instrumento de precisión de laboratorio, el detalle de la información adquirida y procesada por el microcontrolador es alta. El rango elegido para cada sensor cumple con los requerimientos del sistema.

Las pruebas realizadas en los entornos de HiveMQ e IBM Cloud, me brindaron herramientas para garantizar un producto sólido, con eficiencia y seguridad en el transporte de los datos desde el dispositivo hacia internet y desde ahí hacia la interfaz HMI.

El desarrollo completo fue probado en una cámara frigorífica de un local comercial y se recopilaron datos durante 12 horas de funcionamiento.

El sistema tuvo una conexión con internet estable dentro del lugar donde se instaló, logró una comunicación estable con el Bróker, asimismo los con servicios obtenidos de IBM Cloud, arrojando resultados de forma continua durante todo el proceso.

Un punto que trajo dificultades durante la implementación del firmware del microcontrolador fue la comunicación WiFi. Durante el proceso, logré comprender el funcionamiento del protocolo, los pasos necesarios a seguir para realizar una comunicación exitosa con un cable modem y la asignación de una dirección IP, para identificar el dispositivo en Internet.

Como aprendizaje, puedo decir que hay determinados terminales del ESP que no pueden ser utilizados como lectura de datos o escritura, mientras el microcontrolador está trabajando con el módulo WiFi. El GPIO13 no puede ser utilizado como entrada de datos de protocolo 1-Wire, y GPIO25 y GPIO26 no pueden utilizarse como entradas digitales de señal.

El proyecto final superó las expectativas de diseño, desarrollo e implementación.

Capítulo 4: Análisis de Costos

Listo a continuación todos los materiales y componentes utilizados para el desarrollo de este proyecto. Los costos fueron obtenidos de sitios de venta online en el mes de Noviembre de 2022.

4.1 Costos Materiales.

Concepto	Cantidad	Precio Unit	Sub Total
Nodemcu ESP32	1	\$ 4.285	\$ 4.285
DS18B20	2	\$ 1.245	\$ 2.490
SCT-013-005	2	\$ 4.415	\$ 8.830
SCT-013-010	1	\$ 4.415	\$ 4.415
Resistencias 1K	6	\$ 30	\$ 180
Resistencias 4,7K	1	\$ 30	\$ 30
Resistencias 33K	1	\$ 30	\$ 30
Resistencias 39K	1	\$ 30	\$ 30
Resistencias 47K	3	\$ 30	\$ 90
Resistencias 100K	3	\$ 30	\$ 90
Capacitores Electrolíticos 10uF 50V	1	\$ 25	\$ 25
Capacitores Electrolíticos 1uF 50V	1	\$ 25	\$ 25
Capacitores Cerámicos 0,1uF	4	\$ 36	\$ 144
LED Rojo	1	\$ 12	\$ 12
LED Verde	2	\$ 12	\$ 24
Jack TRS est. hembra PCB de 3,5 mm con interruptor	5	\$ 100	\$ 500
Conector Jack Dc Hueco 2.5mm Para Chasis	1	\$ 220	\$ 220
L7805	1	\$ 300	\$ 300
PCB 10cm x 10cm	1	\$ 250	\$ 250
Fuente 12V 2A	1	\$ 770	\$ 770
Chasis Plástico Inyectado 12 x 7,5 x 5 cm	1	\$ 800	\$ 800
Total			\$ 23.540

Tabla 23 - Costo Materiales

4.2 Mano de Obra.

El cálculo realizado para obtener los costos de mano de obra contempla el tiempo dedicado a la investigación, diseño, desarrollo e implementación del proyecto.

El proyecto se realizó en un lapso de ocho meses de trabajo, con jornadas reducidas de trabajo, ya que tengo que cumplir con una jornada laboral semanal en relación de dependencia. Estimé un promedio de 4horas diarias. Calculando la cantidad de horas trabajadas:

$$\text{Horas trabajadas} = 8[\text{meses}] * 4 \left[\frac{\text{semanas}}{\text{mes}} \right] * 5 \left[\frac{\text{dias}}{\text{semana}} \right] * 4 \left[\frac{\text{horas}}{\text{dia}} \right]$$

$$\text{Horas trabajadas} = 640\text{Horas}$$

El Colegio de Ingenieros de la provincia de Entre Ríos CIEER [16] (Colegio de Ingenieros y Especialistas de Entre Ríos), mediante la RESOLUCION N° 1888/2022 de DIRECTORIO, Acta N°

493/2022 fija un valor mínimo por hora para las actividades realizadas por un ingeniero electrónico en la provincia de Entre Ríos, a la fecha de presentación de este proyecto es de \$.1700 (pesos argentinos).

Calculé el costo de la mano de obra total.

$$M.O = \text{Cant Hs Trabajadas} \times \text{Precio/Hora}$$

$$M.O = 640[\text{Horas}] \times 1700 \left[\frac{\$}{\text{hora}} \right]$$

$$\mathbf{M.O = \$1.088.000}$$

La mano de obra tiene un costo de \$1.088.000 (un millón ochenta y ocho mil pesos argentinos)

4.3 Servicios IBM Cloud y HiveMQ.

Si bien para el presente proyecto se trabajó con servicios en IBM Cloud y HiveMQ, utilicé versiones gratuitas para poder hacer las pruebas de campo. Pero si los dispositivos superan la capacidad mínima, hay que tener en consideración los costos de los mismos.

HiveMQ es gratuito hasta cien dispositivos (tópicos) suscritos al Bróker MQTT, por Cluster (Figura 55) y 10Gb de almacenamiento. En el presente diseño la cantidad de tópicos utilizados fueron catorce (Tabla 15). El siguiente plan tiene un costo de USD0.10/dispositivo mensual. La cantidad de equipos que podemos comercializar utilizando la versión gratuita son:

$$\text{Dispositivos Version Gratuita} = \frac{100[\text{tópicos}]}{14 \left[\frac{\text{tópicos}}{\text{dispositivo}} \right]}$$

$$\mathbf{\text{Dispositivos Version Gratuita} = 7}$$

$$\text{Costo minimo plan Estandar} = 14 \left[\frac{\text{tópicos}}{\text{dispositivo}} \right] * 8[\text{dispositivos}] * \frac{\text{Precio}}{\text{dispositivo}} \text{ mensual}$$

$$\text{Costo minimo plan Estandar} = 14 \left[\frac{\text{tópicos}}{\text{dispositivo}} \right] * 8[\text{dispositivos}] * \text{USD0.10}$$

$$\mathbf{\text{Costo minimo plan Estandar} = \text{USD11.2}(\text{dolares estadounidenses})}$$

Teniendo en consideración el precio del dólar en nuestro país a la fecha 26 Noviembre 2022 de \$320 (pesos argentinos)

$$\mathbf{\text{Costo minimo plan Estandar} = \$3584(\text{pesos argentinos})}$$

El servicio Cloudant de IBM Cloud, para levantar un disco con un sistema operativo tiene un plan gratuito con una capacidad de almacenamiento encriptado de 1Gb. El servicio Kubernetes gratuito tiene una duración de un mes. Eligiendo un servicio Estandar, para ejecutar la aplicación Node-Red con 2 vCPU 4GB RAM Virtual compartido, un disco primario de 25Gb SSD, un disco secundario de 100Gb SSD y 1Gbps la velocidad de la red, el costo promedio por hora es de USD0.11. Además tiene costos por:

- Nodo de trabajador USD0.34/hora
- Equilibrador de carga multizona USD0.02/hora
- Seguimiento de actividad USD1.50/mes
- Registro USD1.50/Gb-mes
- Supervisión USD0.05/Gb-mes

Esto da un coste estimado de USD260.64/mes (doscientos sesenta dólares americanos con sesenta y cuatro centavos al mes). Convirtiendo este valor a pesos argentinos da un total de \$83.404,80 (ochenta y tres mil cuatrocientos cuatro pesos argentinos y ochenta centavos).

Es difícil hacer un cálculo certero sobre el costo de estos servicios, pero son valores que hay que tener en consideración a la hora de ofrecer este producto al mercado, ya que estos servicios son necesarios para el funcionamiento del producto.

4.4 Valor de Venta en el Mercado.

Teniendo en cuenta el costo de los materiales, defino el margen de ganancia que quiero obtener por cada dispositivo de un 41,25%.

$$\text{Valor de Venta en el Mercado} = \frac{\text{Costo Materiales}}{1 - \%utilidad}$$

$$\text{Valor de Venta en el Mercado} = \frac{\$23.540}{1 - 41,25\%}$$

$$\text{Valor de Venta en el Mercado} = \$40.000$$

Calculé a continuación la cantidad de unidades que deben venderse para cubrir los costos de mano de obra del diseño. Haciendo un cálculo de ingresos, costos y utilidad por cada unidad creada y restando su valor a la mano de obra calculada, me da un total de 66 equipos para cubrir los costos de mano de obra del proyecto.

Unid	Costo	% Ganancia	Precio Venta	Ingresos	Costos	Utilidad	Mano de Obra	Balance
1	\$ 23.540	41,25%	\$ 40.068	\$ 40.068	\$ 23.540	\$ 16.528	\$ 1.088.000	-\$ 1.071.472
10	\$ 23.540	41,25%	\$ 40.068	\$ 400.681	\$ 235.400	\$ 165.281	\$ 1.088.000	-\$ 922.719
20	\$ 23.540	41,25%	\$ 40.068	\$ 801.362	\$ 470.800	\$ 330.562	\$ 1.088.000	-\$ 757.438
30	\$ 23.540	41,25%	\$ 40.068	\$ 1.202.043	\$ 706.200	\$ 495.843	\$ 1.088.000	-\$ 592.157
40	\$ 23.540	41,25%	\$ 40.068	\$ 1.602.723	\$ 941.600	\$ 661.123	\$ 1.088.000	-\$ 426.877
50	\$ 23.540	41,25%	\$ 40.068	\$ 2.003.404	\$ 1.177.000	\$ 826.404	\$ 1.088.000	-\$ 261.596
60	\$ 23.540	41,25%	\$ 40.068	\$ 2.404.085	\$ 1.412.400	\$ 991.685	\$ 1.088.000	-\$ 96.315
61	\$ 23.540	41,25%	\$ 40.068	\$ 2.444.153	\$ 1.435.940	\$ 1.008.213	\$ 1.088.000	-\$ 79.787
62	\$ 23.540	41,25%	\$ 40.068	\$ 2.484.221	\$ 1.459.480	\$ 1.024.741	\$ 1.088.000	-\$ 63.259
63	\$ 23.540	41,25%	\$ 40.068	\$ 2.524.289	\$ 1.483.020	\$ 1.041.269	\$ 1.088.000	-\$ 46.731
64	\$ 23.540	41,25%	\$ 40.068	\$ 2.564.357	\$ 1.506.560	\$ 1.057.797	\$ 1.088.000	-\$ 30.203
65	\$ 23.540	41,25%	\$ 40.068	\$ 2.604.426	\$ 1.530.100	\$ 1.074.326	\$ 1.088.000	-\$ 13.674

66	\$ 23.540	41,25%	\$ 40.068	\$ 2.644.494	\$ 1.553.640	\$ 1.090.854	\$ 1.088.000	\$ 2.854
67	\$ 23.540	41,25%	\$ 40.068	\$ 2.684.562	\$ 1.577.180	\$ 1.107.382	\$ 1.088.000	\$ 19.382
68	\$ 23.540	41,25%	\$ 40.068	\$ 2.724.630	\$ 1.600.720	\$ 1.123.910	\$ 1.088.000	\$ 35.910
69	\$ 23.540	41,25%	\$ 40.068	\$ 2.764.698	\$ 1.624.260	\$ 1.140.438	\$ 1.088.000	\$ 52.438
70	\$ 23.540	41,25%	\$ 40.068	\$ 2.804.766	\$ 1.647.800	\$ 1.156.966	\$ 1.088.000	\$ 68.966

Tabla 24 - Costo, Precio Venta, Utilidad, Balance

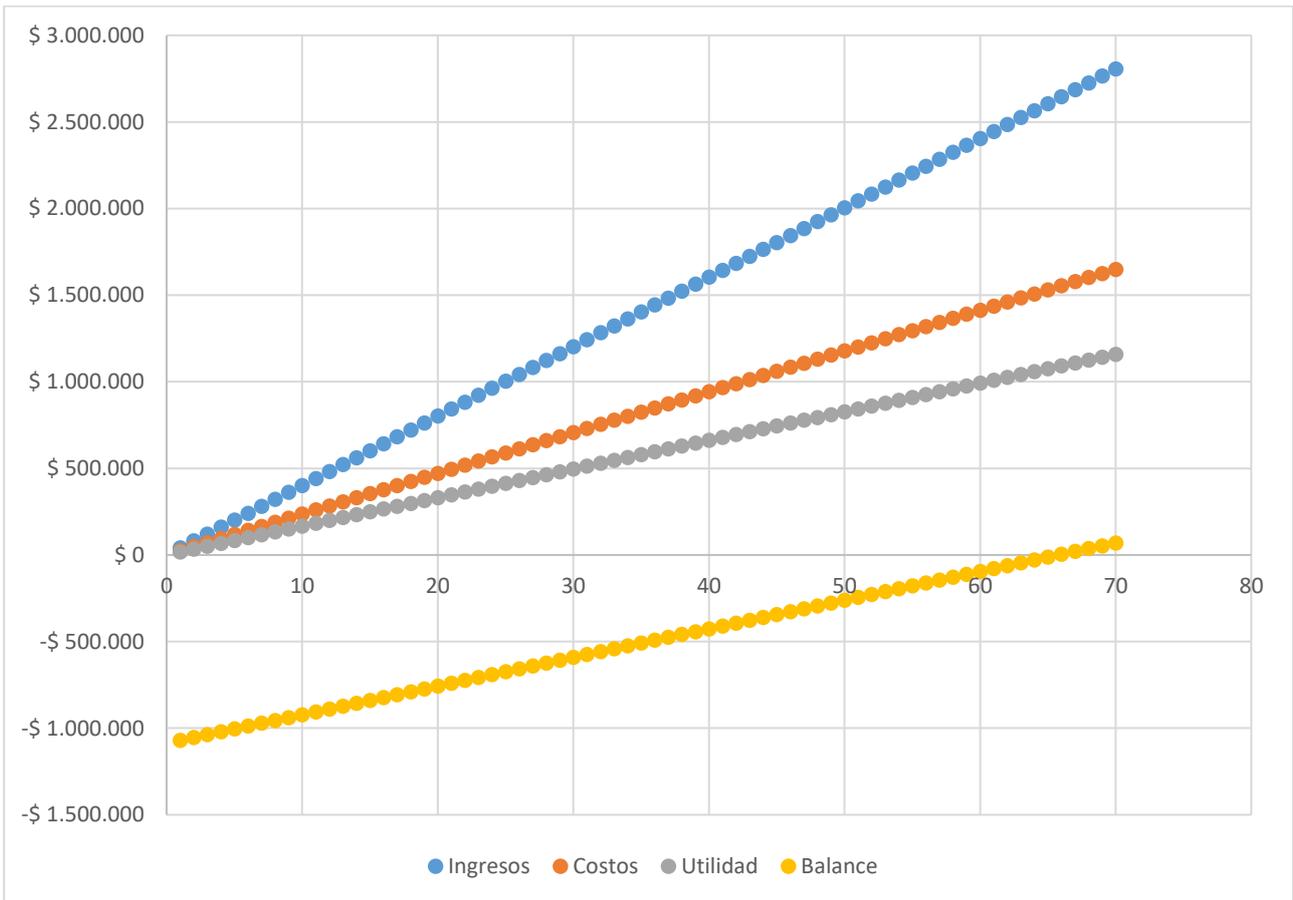


Figura 100 - Ingresos Costos Utilidad Balance

Donde:

$$\text{Ingresos} = \text{Valor Venta} * \text{Unidades}$$

$$\text{Costos} = \text{Costo Materiales} * \text{Unidades}$$

$$\text{Utilidad} = \text{Ingresos} - \text{Costos}$$

$$\text{Balance} = \text{Utilidad} - \text{Mano de Obra}$$

Capítulo 5: Discusión y Conclusión

Los nuevos desarrollos y un nuevo punto de inflexión en la industria manufacturera, es la Industria 4.0. Cada vez van a ser más los dispositivos que se conecten a internet y esto sumado al procesamiento Big Data y la incorporación de la Inteligencia Artificial, proponen los cambios y transformaciones de las tecnologías que hoy conocemos para las próximas décadas.

La incorporación de servicios en la nube, me dieron la posibilidad de trabajar con un sistema muy eficiente, estable, seguro y escalable. Además de las herramientas que proporcionaron pude realizar una interfaz de una manera muy sencilla.

Construí dos diseños hasta llegar a la versión final del proyecto. En la primera solo se contemplé las entradas de los sensores de temperatura y corriente; y los leds indicadores de presencia de voltaje, conexión WiFi y servidor MQTT. En la segunda agregué los indicadores de presencia de sensores de corriente. Los entornos HMI fueron muy diferentes entre la primera versión y la final. En la primera versión solo contemplé las señales de los sensores y su gráfico histórico. En la segunda se agregaron los leds indicadores de presencia de cada sensor. Además incorporé mediante nuevos tópicos la información correspondiente al SSID y RSSI de la red WiFi a la cual está conectado el dispositivo. También agregué las funciones de búsqueda de sensor, en caso de reemplazo e inversión de dirección de sensor de temperatura, ya que no siempre durante el primer escaneo coincidían las direcciones de los sensores con su visualización en el Dashboard. Y por último la tercera visualización de datos históricos en detalle de la interfaz HMI.

Respecto a otros productos que se encuentran comercialmente, el desarrollado en este proyecto tiene la posibilidad de medir las corrientes de los motores y las temperaturas en forma simultánea. Cuando los comerciales se dedican al sensado de temperaturas o corrientes en equipos separados. Estos productos funcionan con una aplicación desarrollada por ellos, lo cual le da ventaja al producto desarrollado en este proyecto que funciona en una página web que se ejecuta en un servidor en la nube. No es necesario instalar ninguna aplicación extra para su uso (excepto la app para realizar la conexión a una red WiFi). Una ventaja de los productos comerciales respecto del diseñado es que tienen un almacenamiento histórico de hasta tres meses.

Como mejoras, pueden agregarse nuevas funciones y sensores (voltaje, presión), según los requerimientos de diseño del cliente.

Asimismo la incorporación del almacenamiento de los datos en una base de datos a fin de tener un control histórico más extenso en el tiempo y poder recurrir a esos datos cuando sea necesario. Además de la incorporación de alertas sobre temperaturas y corrientes críticas.

El presente proyecto pretende tener una mejor comprensión del funcionamiento de una cámara frigorífica y una información más detallada del mismo. No sólo obteniendo datos de las temperaturas, sino también información de su entorno, causante de las variaciones de temperatura.

Capítulo 6: Literatura Citada

- [1] E.G. Pita, *Principios y Sistemas de Refrigeración*. 1ra ed. México D.F: LIMUSA S.A., 1991.
- [2] R.P. Areny, *Sensores y Acondicionadores de Señal*. 4ta ed. Barcelona: Marcombo, 2006.
- [3] A. Gerber, J. Romero (2020, Enero 30). “Key concepts and skills for getting started in IoT” [Online]. Available <https://developer.ibm.com/learningpaths/iot-getting-started-iot-development/concepts-and-skills/>
- [4] A. Gerber, J. Romero (2020, Enero 30). “Choosing the best hardware for your next IoT Project” [Online]. <https://developer.ibm.com/learningpaths/iot-getting-started-iot-development/hardware-guide/>
- [5] A. Gerber, J. Romero (2020, Enero 31). “Connecting all the things in the Internet of Things” [Online]. <https://developer.ibm.com/learningpaths/iot-getting-started-iot-development/networking-guide/>
- [6] A. Gerber, J. Romero (2020, Enero 31). “Streamlining the development of your IoT applications by using an IoT platform” [Online]. <https://developer.ibm.com/learningpaths/iot-getting-started-iot-development/iot-platforms/>
- [7] The HiveMQ Team (2015, Enero 12). “Introducing the MQTT Protocol - MQTT Essentials: Part 1” [Online]. <https://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt/>
- [8] The HiveMQ Team (2015, Enero 19). “Publish & Subscribe - MQTT Essentials: Part 2” [Online]. <https://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe/>
- [9] The HiveMQ Team (2015, Julio 17). “MQTT Client and Broker and MQTT Server and Connection Establishment Explained - MQTT Essentials: Part 3” [Online]. <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>
- [10] The HiveMQ Team (2015, Febrero 2). “MQTT Publish, Subscribe & Unsubscribe - MQTT Essentials: Part 4” [Online]. <https://www.hivemq.com/blog/mqtt-essentials-part-4-mqtt-publish-subscribe-unsubscribe/>
- [11] “DS18B20 Programmable Resolution 1-Wire Digital Thermometer”, MAXIM, Estados Unidos, Datasheet, Octubre 2007.
- [12] “Split Core Current Transformer SCT013-005”, YHDC, China, Datasheet.
- [13] “ESP32 Series”, Espressif Systems, China, Datasheet, Octubre 2022.
- [14] “ESPECIFICACIONES DE COMPRESORES HERMETICOS DE REFRIGERACION”, BlueStar Refrigeration Equipment Co., China, Marzo 2007, Datasheet.
- [15] “Motores Eléctricos Refrigeración”, Weiguang, China, Datasheet.
- [16] CIEER (2022, Septiembre 2). “Resolución N° 1888/2022 de Directorio, Acta N° 493/2022” [Online].

https://www.cieer.org.ar/release/images/documentos/resoluciones/RES%201888_22%20%20Actualizacion%20del%20valor%20de%20la%20unidad%20arancelaria%20Ingenio.pdf