

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

Escuela de Ingeniería y Ciencias



Sistema de administración de energía utilizando Internet de las Cosas (IoT)

Tesis presentada por

David Lizárraga Osuna

sometida a la

Escuela de Ingeniería y Ciencias

como un requisito parcial para obtener el grado académico de

Maestro en Ciencias

en

Ingeniería Energética

Monterrey Nuevo León, 15 de Mayo de 2018

Reconocimientos

A mi familia, quienes siempre me han apoyado a conseguir mis objetivos.

A Mailin Yuleisy Echeverri Molina, por su amor y comprensión que me ayudaron a superar momentos difíciles.

A mis asesores, el Dr. Armando Rafael Llamas Terrés, el MC. Jesús Antonio Baez Moreno, y el MC. Luis Enrique Camargo Reyes, por su asesoría, disposición y comentarios de mejora.

Al Instituto Tecnológico y de Estudios Superiores, por la oportunidad que me brindó para lograr realizar mis estudios.

Sistema de administración de energía utilizando Internet de las Cosas (IoT)

por

David Lizárraga Osuna

Resumen

Se desarrolló un sistema de administración de energía utilizando Internet de las Cosas (IoT), el cual gestiona los equipos de aire acondicionado en instalaciones comerciales y edificios, pudiendo lograr ahorros en energía y demanda. Además, es completamente configurable y funciona en empresas con cualquier número de sucursales y aires acondicionados.

El sistema está compuesto por un dispositivo central, que cuenta con una interfaz web, desde el cual se puede monitorear y controlar los equipos de aire acondicionado; por módulos remotos, los cuales envían y reciben información de manera inalámbrica vía WiFi, y son los encargados de medir la temperatura y humedad, así como controlar a los aires acondicionados; y finalmente por gateways que sirven como puente de comunicación entre el dispositivo central y los módulos remotos en caso de que existan múltiples redes.

Para el desarrollo del sistema se utilizó solamente hardware y software de código abierto. Dentro del hardware utilizado, destacan el Raspberry Pi 3 Model B como dispositivo central, dos Arduino Megas con sus respectivos Ethernet Shields como gateway, y NodemCU como módulo remoto, utilizando DHT22 para las mediciones de temperatura y humedad, así como relevadores de pulso para controlar los aires acondicionados.

Se usaron varios protocolos de comunicación, tal como el UDP, utilizado para la comunicación entre el dispositivo central con los gateways, así como la comunicación entre los gateways y los módulos remotos. Modbus RTU, protocolo que utilizan los

Arduinos Mega de los gateways para transferir información entre ellos; y el protocolo I2C, utilizado por un Arduino para poder leer y escribir sobre un RTC con el objetivo de mantener la hora actualizada.

Los dispositivos fueron programados con diversos lenguajes, destacando PHP, HTML5, CSS3 y JavaScript, utilizados para la programación de la interfaz web. El lenguaje Python con el cual se desarrolló la programación para que el dispositivo central se pudiera comunicar con los gateways, y el lenguaje Arduino, usado para programar los Arduinos Mega y los NodemCU.

Se probaron 2 prototipos, el primero de una sucursal y un equipo de aire acondicionado simulado con focos, con el objetivo de probar el sistema de control. El segundo de 3 sucursales y 4 equipos de aire acondicionado, cuya prueba duró 110 días, con el objetivo de ver la calidad de la comunicación entre los dispositivos, logrando arriba del 99.6% de paquetes UDP satisfactorios y arriba de 98.4% de paquetes enviados por serial con Modbus.

Lista de Figuras

FIGURA 1 CONSUMO FINAL DE ELECTRICIDAD EN EL MUNDO POR SECTORES EN EL 2013 [1]...	16
FIGURA 2 CÁLCULO DE CFE DE LA DEMANDA FACTURABLE.....	22
FIGURA 3 EJEMPLO DE SISTEMA DE CONTROL DE LAZO CERRADO [9]	25
FIGURA 4 ZONA DE CONFORT [13].....	28
FIGURA 5 AIRE ACONDICIONADO TIPO PAQUETE DE INTERIORES [7].....	30
FIGURA 6 AIRE ACONDICIONADO TIPO PAQUETE DE TECHO [7].....	30
FIGURA 7 AIRE ACONDICIONADO TIPO PAQUETE SPLIT [7]	31
FIGURA 8 DISTRIBUCIÓN DE ARTÍCULOS PUBLICADOS (2008-2015)	33
FIGURA 9 ESTRUCTURA DE LA COMUNICACIÓN MODBUS RTU.....	40
FIGURA 10 DIAGRAMA DE CONEXIÓN DEL BUS I2C [38]	40
FIGURA 11 ENERGÍA QUE SE DEJÓ DE CONSUMIR EN UNA ESCUELA DE MICHIGAN CON UN SISTEMA DE ADMINISTRACIÓN DE ENERGÍA EN AIRES ACONDICIONADOS [2]	41
FIGURA 12 ENERGÍA QUE SE DEJÓ DE CONSUMIR EN UNOS EDIFICIOS GUBERNAMENTALES DE FLORIDA CON UN SISTEMA DE ADMINISTRACIÓN DE ENERGÍA EN AIRES ACONDICIONADOS [2]	42
FIGURA 13 APAGANDO LOS COMPRESORES DOS HORAS ANTES DEL HORARIO DE SALIDA [41].	43
FIGURA 14 ARQUITECTURA DEL SISTEMA	46
FIGURA 15 COMPONENTES DEL SISTEMA	46
FIGURA 16 RASPBERRY PI 3 MODEL B [43]	48
FIGURA 17 ARDUINO MEGA 2560 [44].....	49
FIGURA 18 ETHERNET SHIELD W5100 [45]	50
FIGURA 19 CONEXIÓN DEL ARDUINO MEGA 2560 A CONVERTIDOR RS485 A TTL [46]	50
FIGURA 20 CONVERTIDOR RS485 A TTL [46].....	51
FIGURA 21 RTC DS3231 [47]	51
FIGURA 22 NODEMCU v1.0 [48].....	52
FIGURA 23 DHT-22 [49].....	53
FIGURA 24 MCP23017 [50].....	53
FIGURA 25 ULN2803A [51].....	54
FIGURA 26 FUNCIONAMIENTO DEL RELEVADOR TQ2SS-L2-3V	54
FIGURA 27 TQ2SS-L2-3V [52]	55

FIGURA 28 LÓGICA DE CONTROL DE ACUERDO CON SETPOINTS Y TEMPERATURA.....	56
FIGURA 29 CONTROL DEL MÓDULO REMOTO	57
FIGURA 30 DIAGRAMA DE FLUJO DE CÓDIGO EN PYTHON DEL RASPBERRY PI	58
FIGURA 31 MAPA DE LA INTERFAZ WEB	59
FIGURA 32 DIAGRAMA DE FLUJO DE CÓDIGO DEL ARDUINO MEGA A	63
FIGURA 33 DIAGRAMA DE FLUJO DE CÓDIGO DEL ARDUINO MEGA B	67
FIGURA 34 DIAGRAMA DE FLUJO DE CÓDIGO DEL ESP8266	69
FIGURA 35 FLUJO DE DATOS PARA PRUEBAS DE CONFIABILIDAD	72
FIGURA 36 PÁGINA DE INICIO DEL SISTEMA DE ADMINISTRACIÓN DE ENERGÍA	75
FIGURA 37 PÁGINA PARA AGREGAR SUCURSAL.....	75
FIGURA 38 OPCIONES DE MÓDULOS REMOTOS POR EQUIPO DE AIRE ACONDICIONADO	76
FIGURA 39 ERROR AL QUERER DAR DE ALTA UN TERMOSTATO Y UN RELEVADOR EN UN MISMO AIRE ACONDICIONADO	77
FIGURA 40 PÁGINA DE INICIO DEL GATEWAY	77
FIGURA 41 PÁGINA DE CONFIGURACIÓN DEL GATEWAY.....	78
FIGURA 42 RED GENERADA POR EL MÓDULO REMOTO	79
FIGURA 43 PÁGINA DE INICIO DEL MÓDULO REMOTO.....	80
FIGURA 44 PÁGINA DE CONFIGURACIÓN DEL MÓDULO REMOTO	81
FIGURA 45 PÁGINA DE INFORMACIÓN DE LA RED DEL MÓDULO REMOTO.....	82
FIGURA 46 PÁGINA DE INICIO	83
FIGURA 47 LIGA HACIA LOS HISTÓRICOS DE LA SUCURSAL	83
FIGURA 48 DATOS EN TIEMPO REAL DE LA SUCURSAL 1	84
FIGURA 49 PÁGINA DE DATOS HISTÓRICOS DE LA SUCURSAL 1	85
FIGURA 50 DATOS HISTÓRICOS DEL MES DE JULIO DE LA SUCURSAL 1	86
FIGURA 51 DATOS HISTÓRICOS DEL DÍA 10 DE JULIO DEL 2017 DE LA SUCURSAL 1	87
FIGURA 52 DESCARGA DE DATOS HISTÓRICOS	87
FIGURA 53 DATOS HISTÓRICOS DESCARGADOS	88
FIGURA 54 LIGA A LA CONFIGURACIÓN GLOBAL.....	88
FIGURA 55 LIGA A CONFIGURACIÓN DE SUCURSAL	89
FIGURA 56 SELECCIÓN DE HORA EN LA PÁGINA DE CONFIGURACIÓN	89
FIGURA 57 PÁGINA DE CONFIGURACIÓN.....	90

FIGURA 58 SIMULACIÓN DE SISTEMA DE UNA SUCURSAL CON UN EQUIPO DE AIRE ACONDICIONADO	91
FIGURA 59 DISPOSITIVO CENTRAL.....	92
FIGURA 60 DOS ARDUINOS MEGA QUE COMPONEN AL GATEWAY	92
FIGURA 61 MÓDULO REMOTO	93

Lista de Tablas

TABLA 1 PERÍODOS DURANTE EL DÍA EN LA TARIFA HM EN EL NORTE DE MÉXICO [4]	24
TABLA 2 ESCALA DE SENSACIÓN TÉRMICA DE SIETE PUNTOS [11]	26
TABLA 3 TASA METABÓLICA PARA VARIAS ACTIVIDADES [12].....	26
TABLA 4 VALORES DE AISLAMIENTO POR VESTIMENTAS [7].....	27
TABLA 5 LISTA DE DISPOSITIVOS UTILIZADOS	47
TABLA 6 CARACTERÍSTICAS DE UN RASPBERRY PI 3 MODEL B [43]	47
TABLA 7 CARACTERÍSTICAS DE UN ARDUINO MEGA 2560 [44]	49
TABLA 8 CARACTERÍSTICAS DEL NODEMCU V1.0 [48].....	52
TABLA 9 CÓDIGO DE ESTADO POR DISPOSITIVO	73

Contenido

RESUMEN.....	5
LISTA DE FIGURAS.....	7
LISTA DE TABLAS.....	10
CONTENIDO.....	11
1 INTRODUCCIÓN.....	15
1.1 JUSTIFICACIÓN.....	15
1.2 OBJETIVO GENERAL.....	17
1.3 OBJETIVOS ESPECÍFICOS.....	17
1.4 HIPÓTESIS.....	18
1.5 METODOLOGÍA.....	18
1.6 ESTRUCTURA DE TESIS.....	19
2 MARCO TEÓRICO.....	21
2.1 TARIFAS ELÉCTRICAS EN MÉXICO.....	21
2.1.1 TARIFAS EN BAJA TENSIÓN DE CFE.....	22
2.1.2 TARIFA OM DE CFE.....	23
2.1.3 TARIFA HM DE CFE.....	23
2.2 CONTROL EN SISTEMAS DE AIRE ACONDICIONADO.....	24
2.3 CONFORT TÉRMICO.....	25
2.4 TIPOS DE AIRES ACONDICIONADOS.....	28
2.4.1 AIRE ACONDICIONADO TIPO PAQUETE.....	29
2.5 INTERNET DE LAS COSAS (IoT).....	31
2.6 PLATAFORMAS ELECTRÓNICAS DE CÓDIGO ABIERTO.....	35
2.6.1 ARDUINO.....	35
2.6.2 RASPBERRY PI.....	36
2.7 LENGUAJES DE PROGRAMACIÓN.....	36
2.7.1 LENGUAJE ARDUINO.....	37
2.7.2 PYTHON.....	37
2.7.3 PHP.....	38
2.7.4 MYSQL.....	38
2.7.5 JAVASCRIPT.....	38

2.7.6 HTML	38
2.7.7 CSS	39
2.8 PROTOCOLOS DE COMUNICACIÓN	39
2.8.1 MODBUS RTU	39
2.8.2 I2C	40
2.8.3 UDP	40
2.9 ANTECEDENTES	41
3 METODOLOGÍA.....	45
3.1 ARQUITECTURA DEL SISTEMA.....	45
3.2 HARDWARE	47
3.2.1 RASPBERRY PI 3 MODEL B.....	47
3.2.2 ARDUINO MEGA 2560	48
3.2.3 ETHERNET SHIELD W5100.....	49
3.2.4 CONVERTIDOR RS485 A TTL	50
3.2.5 RTC DS3231	51
3.2.6 NODEMCU v1.0	51
3.2.7 DHT-22	52
3.2.8 MCP23017	53
3.2.9 ULN2803A	53
3.2.10 TQ2SS-L2-3V.....	54
3.3 CONTROL DEL SISTEMA.....	55
3.4 PROGRAMACIÓN.....	57
3.4.1 ALGORITMO RASPBERRY PI	57
3.4.1.1 INICIO	59
3.4.1.2 AGREGAR SUCURSAL.....	59
3.4.1.3 CONFIGURACIÓN.....	60
3.4.1.4 SUCURSAL	60
3.4.1.5 DATOS HISTÓRICOS	60
3.4.1.6 ELIMINAR SUCURSAL.....	60
3.4.1.7 UDP	61
3.4.1.8 GET_CONF.....	61

3.4.1.9 MAIN_ALERT	61
3.4.2 ALGORITMO ARDUINO MEGA A.....	61
3.4.2.1 STARTETHERNET	62
3.4.2.2 MODBUS_UPDATE.....	62
3.4.2.3 CHECKER.....	62
3.4.2.4 SENDUDP.....	62
3.4.2.5 RECEIVEUDP	64
3.4.2.6 GETTIME.....	64
3.4.2.7 UPDATEEEPROM	64
3.4.2.8 READEEPROM.....	64
3.4.2.9 RESETER.....	65
3.4.2.10 WDT	65
3.4.3 ALGORITMO ARDUINO MEGA B.....	65
3.4.3.1 GETSTATE	65
3.4.3.2 RECEIVEUDP	66
3.4.3.3 ALERTA.....	66
3.4.3.4 GETTEMP	66
3.4.3.5 WEBSERVER.....	68
3.4.4 ALGORITMO ESP8266.....	68
3.4.4.1 INICIO	68
3.4.4.2 CONFIGURACIÓN.....	68
3.4.4.3 INFORMACIÓN.....	70
3.4.4.4 CÓDIGO PRINCIPAL DEL ESP8266.....	70
3.5 PRUEBAS DE CONFIABILIDAD.....	71
4 RESULTADOS.....	74
4.1 GUÍA DE INSTALACIÓN.....	74
4.2 CONFIGURACIÓN DEL DISPOSITIVO CENTRAL	74
4.3 CONFIGURACIÓN DEL GATEWAY	77
4.4 CONFIGURACIÓN DEL MÓDULO REMOTO.....	78
4.5 VISUALIZACIÓN DE DATOS	82
4.6 CONTROL DEL SISTEMA.....	88

4.7	DIAGNÓSTICO DE COMUNICACIÓN.....	92
5	CONCLUSIONES	94
5.7	RESULTADOS Y CONCLUSIONES	94
5.8	CONTRIBUCIONES	94
5.9	TRABAJO FUTURO.....	96
6	REFERENCIAS	97

1 Introducción

En este primer capítulo se presenta la situación actual del consumo de electricidad en el mundo y el impacto que los sistemas de aire acondicionado tienen en este. Se mencionan algunos beneficios de los sistemas de administración de energía en edificios, así como la importancia de comenzar a implementar dichos sistemas con hardware y software de código abierto. Más adelante se presenta el objetivo general, los objetivos específicos, la hipótesis de esta investigación, las etapas de la metodología y la estructura de la tesis.

1.1 Justificación

Los sistemas de aire acondicionado son los equipos que contribuyen en mayor medida al consumo de energía y a la demanda máxima en la mayoría de los edificios residenciales, comerciales y de servicios, incluso, en algunos edificios donde el control de la humedad toma un papel importante como, por ejemplo, hospitales, museos, laboratorios y centros de datos, estos sistemas representan el 70% del consumo de energía eléctrica y hasta el 85% de la factura eléctrica total para usuarios en media tensión.

Por otra parte, el consumo final de energía eléctrica en el mundo en el 2013 fue de 17,867 TWh, como se muestra en la Figura 1, entre el sector residencial y el sector comercial conforman el 51%, atribuyéndole, como se mencionó anteriormente, gran parte a los sistemas de acondicionamiento de aire [1].

Es común encontrar edificios en los cuales, no se tienen ajustes de temperatura de acuerdo con la ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers), y que se mantienen acondicionados espacios no ocupados. Mediante un sistema de administración de energía como el propuesto, es posible estandarizar los ajustes de temperatura, manejar diferentes setpoints para los horarios de ocupación y NO-ocupación, entre otras técnicas propuestas por la ASHRAE tal como

el apagado y el encendido inteligente [2]. Con la utilización del sistema propuesto, es posible lograr ahorros significativos en energía y en facturación.

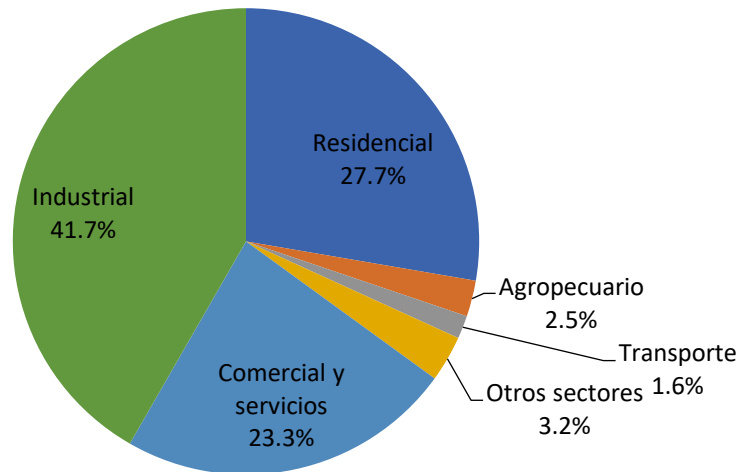


Figura 1 Consumo final de electricidad en el mundo por sectores en el 2013 [1]

De acuerdo con la ASHRAE, el ahorro con un control apropiado del sistema de aire acondicionado en un edificio ya existente es entre el 10% y el 15% del costo total anual por energía [2].

A pesar de que sistemas de administración de energía que realizan este tipo de control ya existen en el mercado, su implementación se realiza principalmente en empresas en las cuales la facturación mensual es de cientos de miles de pesos, sin embargo, no son asequibles para las micro, pequeñas y medianas empresas con facturación menor, y en muchos casos, el costo de la energía eléctrica en éstas, representa un porcentaje importante de los costos de operación [3]. Al desarrollar un sistema con hardware y software de código abierto, libre de licencias y con componentes de bajo costo, se podría llegar a implementar en cualquier tipo empresa que cuente con sistemas de aire acondicionado, sin importar el tamaño de esta.

1.2 Objetivo General

Desarrollar un sistema de administración de energía con hardware y software de código abierto completamente configurable, capaz de utilizar alguna de las técnicas de control de equipos de aire acondicionado propuestas por la ASHRAE, así como monitorear y registrar temperatura y humedad en múltiples zonas.

1.3 Objetivos específicos

- Diseñar la arquitectura del sistema que se adapte a los espacios físicos de las empresas y edificios modernos.
- Adquirir plataformas electrónicas de código abierto a bajo costo, que satisfagan el diseño de la arquitectura del sistema.
- Desarrollar un método para el control de sistemas de aire acondicionado (tipo paquete), que se adapte al hardware disponible.
- Diseñar el algoritmo a utilizar en las plataformas electrónicas a partir de los métodos desarrollados.
- Programar las plataformas electrónicas con el algoritmo e integrar todos los componentes del sistema, que permita la transferencia de datos y el control de los equipos de aire acondicionado.
- Desarrollar una interfaz gráfica amigable que permita a los usuarios monitorear el sistema y definir tanto los ajustes de temperatura, como los horarios de operación de los equipos de aire acondicionado.
- Observar el comportamiento del sistema desarrollado en operación, para determinar su confiabilidad de acuerdo con la cantidad de datos transferidos satisfactoriamente.

1.4 Hipótesis

Es posible desarrollar un sistema de administración de energía con hardware y software de código abierto, que permita controlar la operación de los equipos de aire acondicionado de acuerdo con la ocupación y a la hora del día, para de esta manera, lograr una reducción en la facturación de energía eléctrica, y que, además, sea confiable y fácil de configurar por parte de los usuarios, para instalaciones residenciales, micro, pequeñas y medianas empresas.

1.5 Metodología

Para desarrollar el sistema de administración de energía, lo primero que se ideó fue la arquitectura del sistema, la cual debía contar con ciertas características, tal como, replicabilidad, fácil instalación, que todos sus componentes y programas sean tanto de hardware como software de código abierto y que se pueda aplicar en empresas que sus sucursales no se encuentren cercanas unas de las otras. Se pensó que un medio ideal para la transmisión de datos es el WiFi, ya que se puede encontrar en la mayoría de los edificios de la actualidad. Con esto, se definió que la arquitectura del sistema debía tener tres principales componentes: un dispositivo central que canalice toda la información y sea el portal entre los usuarios y el resto del sistema; módulos remotos que se encarguen de controlar los equipos de aire acondicionado, y de medir parámetros como la temperatura y humedad; y gateways que actúen como puentes de comunicación entre el dispositivo central y los módulos remotos, preparando la información de toda una zona (una sucursal, por ejemplo), enviándosela al dispositivo central y viceversa. Con dichos gateways, puede existir comunicación entre los módulos remotos y el central, aunque estos se encuentren en diferentes redes.

Se determinó que el mejor candidato para el dispositivo central es un Raspberry Pi 3 Model B, debido a su gran capacidad de procesamiento, siendo una computadora del tamaño de una tarjeta de crédito. En él se desarrolló la interfaz, la cual es una página web creada en HTML5, CSS3, JavaScript y PHP, mientras que la base de datos donde

se maneja toda la información es gestionada por MySQL. La comunicación entre el Raspberry Pi y los gateways está basada en el protocolo UDP y escrita en Python.

Los componentes elegidos para constituir al gateway son dos Arduinos Mega, con Ethernet Shield cada uno. Un Arduino Mega (llamado A) se conecta a la red del dispositivo central, el otro (llamado B) se conecta a la red de los módulos remotos; la comunicación entre los Arduinos es por Modbus RTU RS485, por lo que cada uno requiere de un convertidor. En caso de que el edificio o empresa en donde se quiera instalar el sistema no sea muy grande (menor a 10 aires acondicionados), y exista una red WiFi en toda la zona, se puede omitir el gateway.

El ESP8266 NodemCU v1.0 fue seleccionado como módulo remoto, ya que este es un microcontrolador el cual tiene un módulo integrado que le permite conectarse a redes vía WiFi. A él se le conectan sensores de temperatura y humedad DHT22, y relevadores de pulso que le permiten controlar los equipos de aire acondicionado. El método de control que se ideó para implementarlo en el sistema consiste en interceptar las señales de control de los termostatos a los equipos de aire acondicionado, manipulándolos con los relevadores del módulo remoto.

El sistema de administración de energía se programó para que identifique fallas en cada uno de los componentes y los registre, con el fin de generar datos históricos que permitan calcular la confiabilidad del sistema, de acuerdo con el porcentaje de errores que se produzcan al realizar las pruebas.

1.6 Estructura de tesis

Esta tesis está segmentada en cinco capítulos:

- **Introducción:** aquí se aborda la situación del consumo de energía eléctrica de los equipos de aires acondicionados en edificios, así como los beneficios de los sistemas de administración de energía.

- Marco teórico: se explican fundamentos básicos relacionados al proyecto de investigación, tal como zona de confort, Internet de las Cosas (IoT), hardware y software de código abierto, y algunos antecedentes.
- Metodología: se detallan los pasos que se siguieron para el desarrollo del sistema de administración de energía, desde el diseño de la arquitectura del sistema, hasta la programación de cada uno de los componentes.
- Resultados: aquí se muestra el prototipo del sistema de administración de energía ya integrado, la guía de instalación y configuración, así como sus interfaces y los resultados de confiabilidad.
- Conclusiones: se presentan las contribuciones de este proyecto de investigación y el trabajo a futuro.

2 Marco teórico

En este capítulo se presentarán algunos conceptos básicos necesarios para comprender el desarrollo de este estudio y los beneficios que puede generar. Algunos ejemplos son, las tarifas eléctricas en México, la zona de confort propuesta por la ASHRAE, la explicación de en qué consisten los aires acondicionados tipo paquete, qué son las plataformas de código abierto y dos ejemplos, Arduino y Raspberry Pi; los lenguajes de programación utilizados para el desarrollo de esta investigación; y, por último, los protocolos de comunicación que utilizan los dispositivos de la misma.

Al final del capítulo se presentan cuatro trabajos similares al que se está proponiendo, dos de ellos son casos de estudios de la ASHRAE, donde generan ahorros a partir de un sistema de control de aires acondicionados, el cual utiliza el método de *horarios de ocupación*. El tercero consiste en un sistema de administración de energía aplicado en la Universidad Politécnica de Valencia, que controla miles de minisplits, logrando ahorros en energía y demanda. Por último, se muestra una plataforma de *Casa Inteligente* desarrollada con dispositivos de código abierto.

2.1 Tarifas eléctricas en México

Las tarifas eléctricas de CFE se componen de tres elementos, cargo por energía, cargo por demanda y penalización o bonificación por factor de potencia. El cargo por energía es simplemente el producto del costo unitario de la energía multiplicado por la cantidad consumida durante el período de facturación [4].

El cargo por demanda está en función de las demandas máximas registradas en los distintos períodos horarios, las cuales se determinan mensualmente por medio de instrumentos de medición, que indican la demanda media en kilowatts, durante cualquier intervalo de 15 minutos del periodo en el cual el consumo de energía eléctrica sea mayor que en cualquier otro intervalo de 15 minutos en el período correspondiente. Los intervalos de demandas de 15 minutos están formados por tres sub intervalos de 5 minutos [4], tal y como se muestra en la Figura 2.

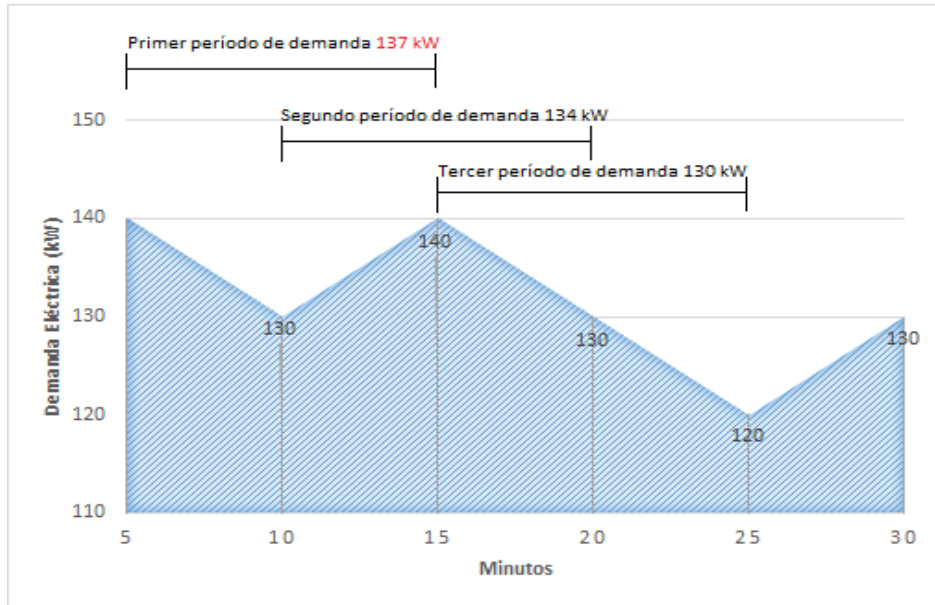


Figura 2 Cálculo de CFE de la demanda facturable

Para el caso del factor de potencia, este se calcula con un promedio mensual y genera un recargo en caso de que sea menor a 90%, o una bonificación en caso de que sea mayor a 90%, de acuerdo a las siguientes ecuaciones [4].

$$\text{Recargo} = \frac{3}{5} * \left(\frac{0.9}{fp} - 1 \right)$$

$$\text{Bonificación} = \frac{1}{4} * \left(1 - \frac{0.9}{fp} \right)$$

2.1.1 Tarifas en baja tensión de CFE

Entre las tarifas en baja tensión de CFE se encuentran las residenciales, tal como las *tarifas domésticas 1, 1A, 1B, 1C, 1D, 1E, 1F* y *de alto consumo (DAC)*, y las comerciales como las *tarifas 2 y 3*. Todas ellas tienen en común que solo cobran por la energía consumida [5], por lo que cualquier implementación de sistemas de eficiencia energética en estos esquemas tarifarios, solo lograría ahorro por energía.

2.1.2 Tarifa OM de CFE

La tarifa OM (Ordinaria para servicio general en media tensión), es una tarifa en media tensión en donde hay un cargo por energía y otro por demanda, además de realizar un recargo o bonificación por factor de potencia. La demanda facturable resulta del periodo quinceminutal más alto del mes, por lo que al igual que en las tarifas de baja tensión, un sistema de ahorro de energía implementado en instalaciones con este esquema tarifario difícilmente lograría un ahorro por demanda [6].

2.1.3 Tarifa HM de CFE

La tarifa HM (Tarifa horaria para servicio general en media tensión) es una tarifa en media tensión, que está dividida en períodos a lo largo del día. Estos periodos, para el norte de México se muestran en la Tabla 1. Para la componente de energía existen tres costos por kWh diferentes, que pertenecen a los horarios base, intermedio y punta. Para la demanda hay un único cargo por kW de demanda facturable, sin embargo, para calcular esta se utiliza la siguiente ecuación [4]:

$$DF = DP + FRI \times \max(DI - DP, 0) + FRB \times \max(DB - DPI, 0)$$

Donde DF es la demanda facturable, DP es la demanda máxima en período punta, DI es la demanda máxima en horario intermedio, DB es la demanda máxima en período base, DPI es la demanda máxima en los horarios de punta e intermedio, y, FRI y FRB son factores de reducción que dependen de la región tarifaria, siendo 0.30 y 0.15 para la mayor parte del país. Analizando la ecuación anterior, es evidente que el horario que representa mayor peso para el costo por demanda es la punta, ya que, en el mejor de los escenarios, este representa el 70% de la demanda facturable [4]. Esto convierte al horario punta en un área de oportunidad para proyectos de ahorro energético.

Verano			
Día de la semana	Base	Intermedio	Punta
Lunes a viernes	0:00 - 6:00	6:00 - 20:00 22:00 - 24:00	20:00 - 22:00
Sábado	0:00 - 7:00	7:00 - 24:00	
Domingo y festivos	0:00 - 19:00	19:00 - 24:00	
No verano			
Día de la semana	Base	Intermedio	Punta
Lunes a viernes	0:00 - 6:00	6:00 - 18:00 22:00 - 24:00	18:00 - 22:00
Sábado	0:00 - 8:00	8:00 - 19:00 21:00 - 24:00	19:00 - 21:00
Domingo y festivos	0:00 - 18:00	19:00 - 24:00	

Tabla 1 Períodos durante el día en la tarifa HM en el norte de México [4]

2.2 Control en sistemas de aire acondicionado

Un sistema de control de aire acondicionado, modula principalmente la capacidad del equipo para mantener un parámetro predeterminado, típicamente temperatura, humedad relativa, presión, entalpía, flujo de aire, concentración de contaminantes, entre otros; con el fin de preservar el confort térmico del lugar acondicionado, proveer una cantidad y calidad adecuada de aire, garantizar la seguridad del equipo y que sea lo más eficiente posible en términos energéticos [7].

Dicho sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado en un proceso. Los esquemas de control más básicos son el de lazo abierto y lazo cerrado. En los primeros, la salida no tiene efecto sobre la acción de control, en cambio, los de lazo cerrado (véase Figura 3) se retroalimentan para convertir la respuesta relativamente insensible a las perturbaciones externas y variaciones internas en los parámetros del sistema [8].

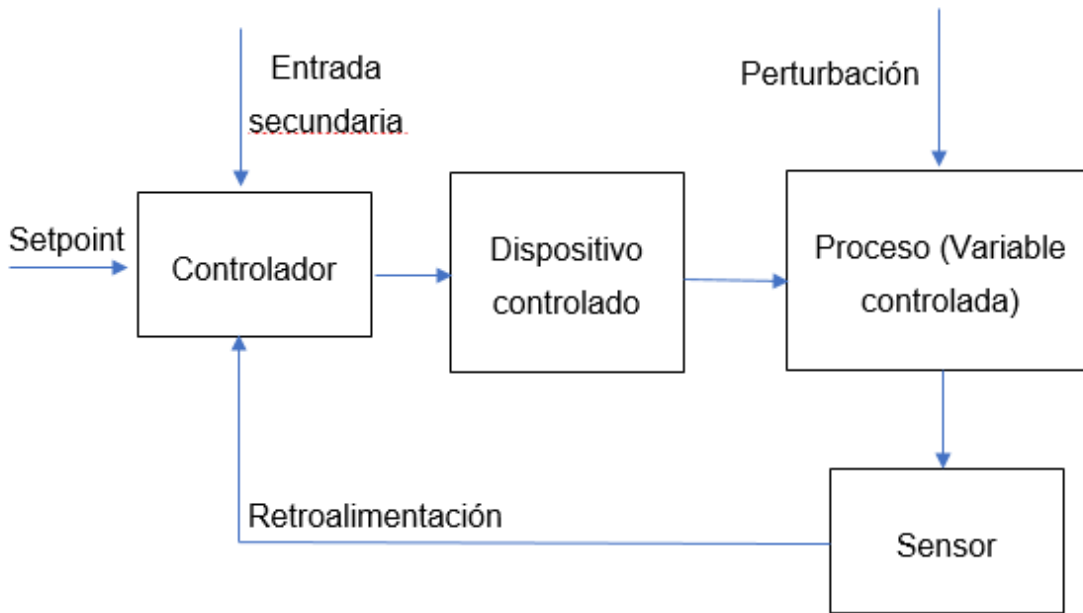


Figura 3 Ejemplo de sistema de control de lazo cerrado [9]

Todo sistema de control tiene tres componentes básicos: los sensores, que son los responsables de medir las variables del sistema; el controlador, que es el cerebro del sistema y el que toma las decisiones en base a las mediciones recibidas de los sensores; y los actuadores, encargados de realizar la acción resultante de la decisión del controlador [10].

Existen distintos modos de control, estos describen cómo el controlador realizará las acciones correctivas para manipular la variable controlada. Hay distintas clasificaciones del modo de control, pero sólo se mencionará el control de *dos posiciones*, ya que es el que se utiliza en la presente investigación. Este utiliza sólo dos estados de operación, el máximo y el mínimo. Algunos ejemplos son el encender y apagar ventiladores y compresores [8].

2.3 Confort térmico

El confort térmico es el estado en que una persona se siente satisfecha con un entorno climático. En otras palabras, se puede decir que es la sensación térmica de no estar ni

frío ni caliente, definido por la escala de siete puntos propuesta por la ASHRAE que se muestran en la Tabla 1 [11].

-3	Frío
-2	Fresco
-1	Ligeramente fresco
0	Neutral
+1	Ligeramente cálido
+2	Cálido
+3	Caliente

Tabla 2 Escala de sensación térmica de siete puntos [11]

Algunos parámetros que afectan el confort térmico son la temperatura, humedad relativa, velocidad del aire, calidad del aire, tasa metabólica y el aislamiento de la vestimenta. La tasa metabólica, es la energía liberada por unidad de área de la superficie del cuerpo y depende principalmente de la intensidad de las actividades físicas realizadas por el cuerpo humano, su unidad de medida es el *met*, que equivale a 58.24 W/m² [12]. En la Tabla 3 se muestra la tasa metabólica para varias actividades.

Actividad	Tasa metabólica (<i>met</i>)
Dormir	0.7
Estar sentado y quieto	1.0
Leer sentado	1.0
Escribir	1.1
Dar clases	1.6
Cocinar	1.6 – 2.0
Limpiar la casa	2.0 - 3.4
Caminar a 3.2 km/h	2.0
Caminar a 6.4 km/h	3.8
Bailar	2.4 - 4.4
Jugar tennis	3.6 – 4.0
Jugar basketball	5.0 - 7.6

Tabla 3 Tasa metabólica para varias actividades [12]

En cuanto al nivel de aislamiento de la vestimenta, este se mide con la unidad *clo*, que equivale a $0.16 \text{ m}^2 \cdot \text{°C}/\text{W}$ o $0.88 \text{ h} \cdot \text{ft}^2 \cdot \text{°F}/\text{BTU}$. En la Tabla 4 se muestran los valores de aislamiento para algunos conjuntos de vestimenta.

Vestimenta	Valor de aislamiento (<i>clo</i>)
Shorts y camisa de manga corta	0.41
Pantalón ajustado y camisa de manga corta	0.50
Pantalón ajustado y camisa de manga larga	0.62
Pantalón ajustado, camisa de manga larga y saco de traje	0.96
Pantalón suelto, camisa de manga larga y suéter de manga larga	1.01
Ropa deportiva de manga larga	0.77

Tabla 4 Valores de aislamiento por vestimentas [7]

Una vez definidos la tasa metabólica y el valor de aislamiento por vestimenta, se puede proceder a analizar la gráfica de la zona de confort propuesta por la ASHRAE en el estándar 55, con el objetivo de mostrar las condiciones ideales que deberían darse dentro de los espacios acondicionados tanto en invierno como en verano (Figura 4). El eje curvo del lado izquierdo es la temperatura de bulbo húmedo, las curvas diagonales corresponden a la humedad relativa, el eje derecho es la proporción de masa de vapor de agua por masa de aire seco; y por último, el eje inferior corresponde a la temperatura operativa, que en la práctica, considerando que los coeficientes de radiación y convección son aproximadamente iguales, se puede tomar la temperatura operativa como el promedio de la temperatura seca del aire y la temperatura de los elementos del lugar habitado [13].

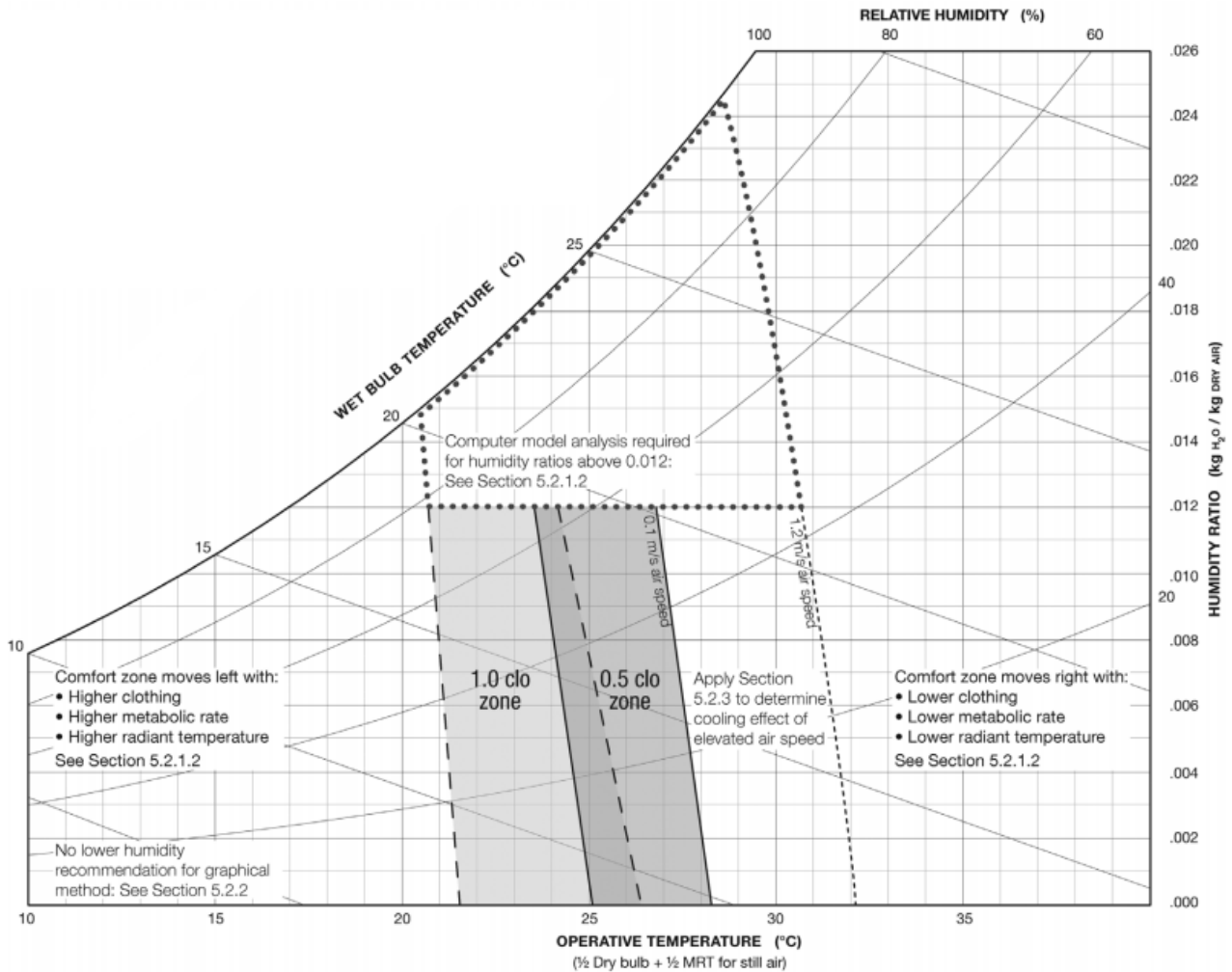


Figura 4 Zona de confort [13]

Cabe destacar que la zona de confort (zonas grises) mostrada en la Figura 4 está basada en una velocidad del viento no mayor a 0.10 m/s. Además, se muestran dos zonas, con 1.0 y 0.5 *clo* que corresponden al nivel de aislamiento de las vestimentas en invierno y verano respectivamente [13].

2.4 Tipos de aires acondicionados

Los aires acondicionados adecúan el aire del exterior, lo transportan y lo introducen en el interior. Controlan y mantienen la temperatura, humedad, movimiento y limpieza del aire; y el diferencial de presión del espacio, con el fin de proveer las condiciones

necesarias dentro del espacio acondicionado, ya sea mejorar el confort y la salud de los ocupantes, refrigeración de productos delicados, entre otros [14].

Según [14], los aires acondicionados se pueden clasificar de acuerdo a sus características de construcción y operación. Algunos tipos de aire acondicionado se listan a continuación:

- Aire acondicionado para habitación individual.
- Aire acondicionado de enfriamiento por evaporación.
- Aire acondicionado basado en desecantes.
- Aire acondicionado con almacenamiento térmico.
- Aire acondicionado de paquete.

Solo se abordará con mayor detalle los aires acondicionados tipo paquete, ya que son los que serán controlados con el sistema presentado en esta tesis.

2.4.1 Aire acondicionado tipo paquete

Estos sistemas pueden estar contenidos en una sola unidad de paquete, o dividido en dos unidades. Los paquetes de una sola unidad contienen ventiladores, filtros, evaporador de expansión directa, compresores, condensadores y otros accesorios, los cuales pueden estar tanto en el interior del espacio acondicionado (véase Figura 5), como en el exterior, comúnmente en el techo (véase Figura 6). En un sistema dividido o *Split*, una parte está en el interior, y se compone principalmente del sistema de control, ventiladores, filtro, y evaporador; mientras que la otra parte se encuentra en el exterior, y contiene el compresor y el condensador (véase Figura 7). También son caracterizados por venir integrados de fábrica, y estar listos para usar después de su instalación [15].

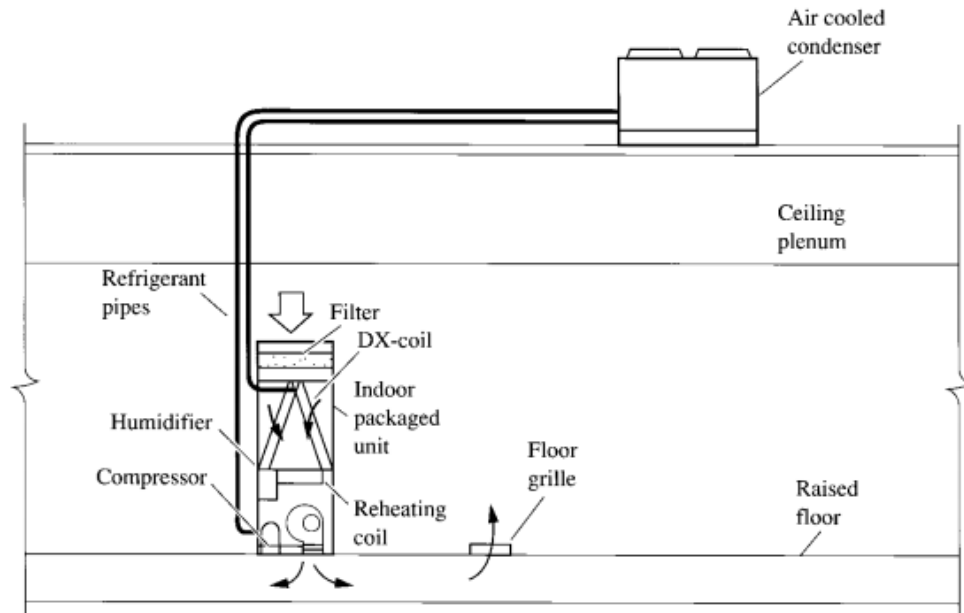


Figura 5 Aire acondicionado tipo paquete de interiores [7]

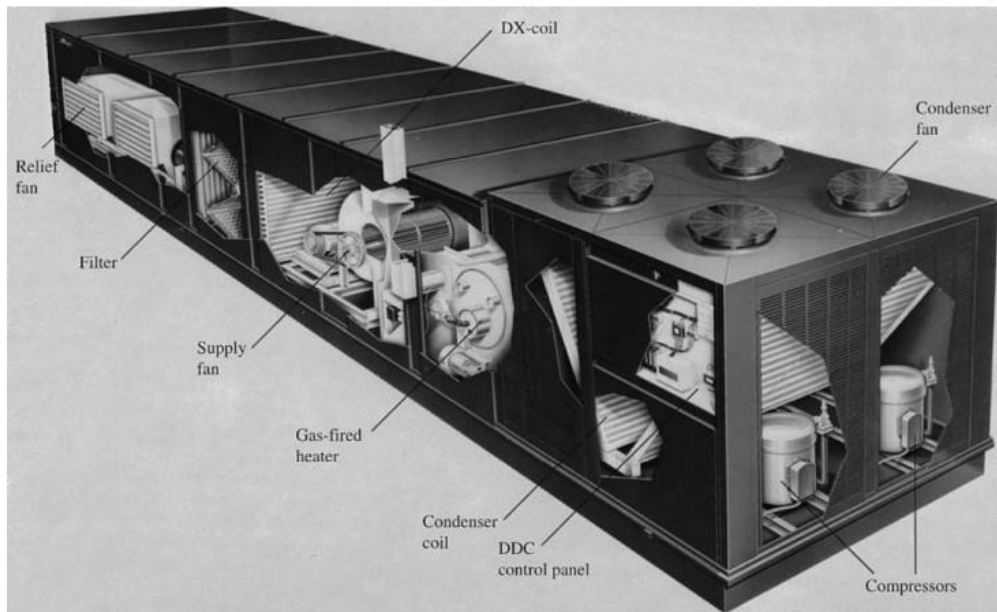


Figura 6 Aire acondicionado tipo paquete de techo [7]

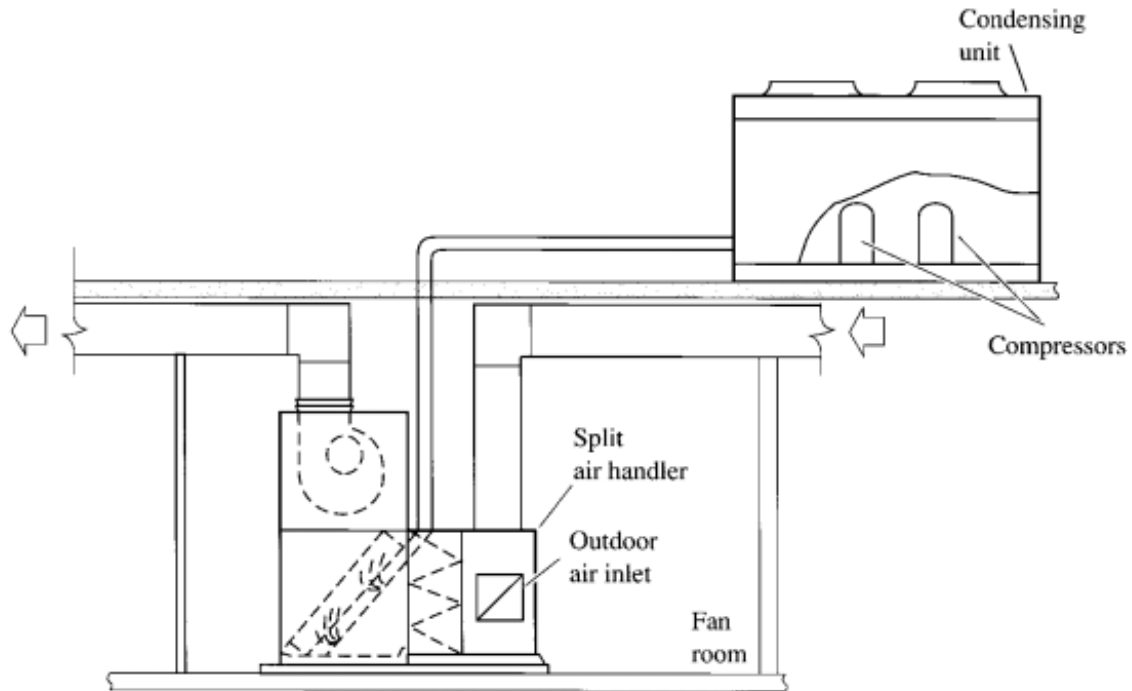


Figura 7 Aire acondicionado tipo paquete Split [7]

Los aires acondicionados de tipo paquete pueden servir tanto para una simple habitación como para varias. Los equipos que solamente consisten en un gran sistema unitario son llamados centrales, los cuales cuentan con ductos de suministro para la distribución del aire [16].

2.5 Internet de las cosas (IoT)

Cada vez es más estrecha la relación entre las personas y el contenido multimedia. Para ello se crean máquinas inteligentes que permitan dicha interacción. Esto se conoce comúnmente como sistemas ciber físicos llamados en inglés como cyber-physical systems (CPS) [17], los cuales conectan los dispositivos computacionales con el mundo físico. El CPS lo componen cuatro tecnologías según [18]:

1. Automatización de procesos
2. Robótica avanzada

3. Vehículos autónomos y semi autónomos
4. Internet de las cosas

El término de internet de las cosas surge en 1999 y se define como una infraestructura de red global dinámica con capacidades de autoconfiguración basada en protocolos de comunicación estándar e interoperables donde las cosas físicas y virtuales tienen identidades, atributos físicos y personalidades virtuales, utilizan interfaces inteligentes y se integran perfectamente en la red de información [19]. Una definición corta lo define como: una red mundial de objetos únicos direccionables interconectados [20].

En los términos más sencillos, el IoT parece contemplar una sociedad donde todos los miembros tengan acceso a un entorno de Internet, caracterizado por la autoconfiguración, la auto gestión, la tecnología inteligente en cualquier momento y en cualquier lugar [19].

Al referirse a la palabra “cosas” se hace alusión a todo aquello que puede ser leído, reconocido, localizado, dirigido y controlado sin esfuerzo, usando internet. Internet permite a las tecnologías digitales de cualquier tipo compartir información y comunicarse. Los beneficios potenciales de esta integración son enormes. El IoT busca aplicar este mismo proceso a objetos físicos. En lugar de utilizar interfaces complicadas, muchas de estas aplicaciones requieren sólo un sensor simple que proporcione información pasiva y continua sobre el objeto [21].

Según Mishra et al [18], el IoT es el que genera mayores valores económicos de las cuatro tecnologías antes expuestas, con un impacto estimado en 36 trillones USD en EE.UU. Su relevancia también trasciende, al punto de considerar a las industrias digitales como responsables del 20% del producto interno bruto en Estados Unidos, dejando el restante 80% para agricultura, construcción, manufactura, energía, etc.

Empresas como Intel, Microsoft, AT&T y DHL, están invirtiendo en el IoT, generando una variedad de informes que calculan un enorme crecimiento futuro. Se estima que para 2019, el IoT será el mayor mercado de dispositivos en el mundo. Igualmente, los estudios sobre el tema han crecido exponencialmente en los últimos años (Figura 8), lo cual indica un mayor interés con el paso del tiempo [18].

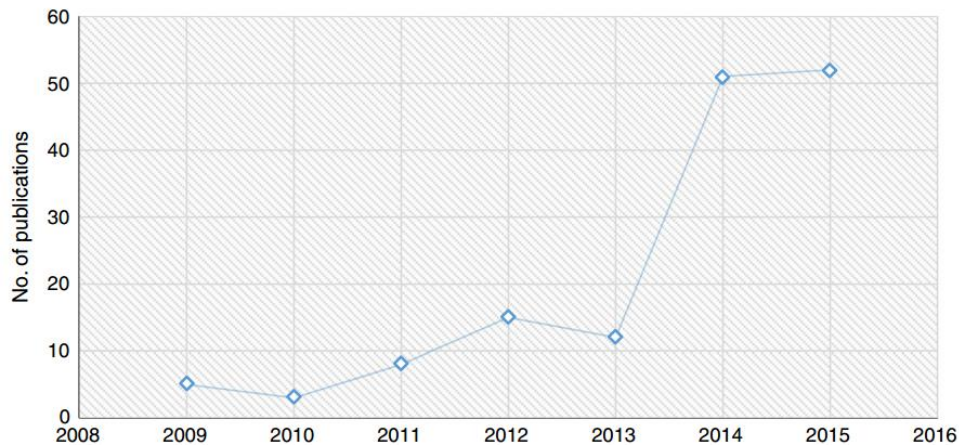


Figura 8 Distribución de artículos publicados (2008-2015)

Las aplicaciones concernientes al IoT pueden clasificarse en cuatro: el área industrial, de la salud, en entornos inteligentes; social y personal. En las aplicaciones para entornos inteligentes se han desarrollado avances que ayudan a las personas. Polycarpou et al. (2013) [22] identificaron la aplicación en sistemas de estacionamiento. Puede ayudar a los conductores a encontrar un estacionamiento según su conveniencia y preferencia, ahorrando así tiempo y combustible, al tiempo que reduce el nivel de huella de carbono. Los vehículos con sensores móviles obtienen información detallada relacionada con la densidad del tráfico o las condiciones superficiales de la carretera en comparación con los sensores fijos que se utilizaron anteriormente [23].

Entre otros usos también está la seguridad (por ejemplo, control de acceso, atención de seguridad para personas mayores, informes de tiempo para el cuidado en el hogar); seguimiento y rastreo (por ejemplo, gestión de flotas, logística para el transporte de mercancías); pagos de salud (por ejemplo, atención domiciliaria), control remoto y

mantenimiento (por ejemplo, hogares inteligentes, vigilancia ambiental); y medición (por ejemplo, redes eléctricas inteligentes) [20].

Existen siete factores de valor con los que el IoT puede contribuir a los procesos de negocio, que de acuerdo a Nolin y Olson (2016), son:

1. Disparador de proximidad manual simplificada: Las cosas inteligentes pueden comunicar su número de identificación a otras cosas en su proximidad. Cuando las cosas inteligentes se mueven manualmente cerca unas de otras, este disparador de proximidad permite transacciones instantáneas, pagos o comprobaciones con validez automáticamente.
2. Disparador automático de proximidad: este es un servicio que se puede activar de forma automática, sin intervención humana, cuando dos cosas entran en proximidad con otra. Por ejemplo, cuando el coche y la llave están lo suficientemente cerca, la puerta se desbloquea y se abre.
3. Activación automática del sensor: con este servicio, las cosas pueden detectar los datos de su entorno y esto desencadena diversas actividades. Esto puede aliviar el trabajo activo de las personas en la detección de diversos procesos a su alrededor.
4. Seguridad automática del producto: el IoT puede usarse para rastrear productos certificados por marcas y obstruir todo negocio de falsificaciones. Aquí reside un potencial de aniquilación total del mercado negro.
5. Retroalimentación simple y directa por parte del usuario: el IoT también proporcionará retroalimentación simple al usuario para afirmar o tranquilizarnos que un cierto disparador ha sido activado. Esto puede ser en forma de un pitido al pasar una tarjeta.
6. Amplia información de los usuarios: las personas pueden recibir una amplia retroalimentación a través de un dispositivo práctico, como el teléfono móvil. Esto conduce a la "comodidad profunda del producto porque la información individualizada está a mano exactamente en el punto de la decisión".

7. Mente que cambia la retroalimentación: hay un enorme potencial cuando la tecnología puede influir en el comportamiento de los usuarios. El IoT puede ser usado para hacer que la gente se comporte de una manera más "racional".

2.6 Plataformas electrónicas de código abierto

Las plataformas electrónicas de código abierto son objetos físicos para los cuales su documentación y diseños relevantes son accesibles de manera gratuita. De acuerdo con la *“Open Source Hardware Association”*, todos tienen la libertad de modificar y vender tales diseños abiertos, otorgando a los usuarios control total de la tecnología [24].

Los procesos de diseño y desarrollo de plataformas electrónicas de código abierto son eficientes e innovadores porque la comunidad se compone de muchas personas alrededor del mundo, quienes la mayoría son voluntarios que colaboran en diferentes proyectos, asimismo, la información circula libremente, las personas comparten ideas y aprenden unos de los otros [24].

2.6.1 Arduino

Arduino es una plataforma electrónica de código abierto, hecho para ser fácil, tanto de programar como de utilizar su hardware. Las tarjetas Arduino son capaces de leer datos de entrada, como sensores, y convertirlos en salidas, como mover un motor. Los Arduinos se pueden programar utilizando el lenguaje de programación Arduino, con el Arduino Software (IDE) [25].

A través de los años, Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos, hasta complejos instrumentos científicos. Inicialmente nació como una herramienta para hacer prototipos rápidamente, enfocado a estudiantes sin conocimientos previos en programación, sin embargo, tan pronto alcanzó una audiencia mayor, las tarjetas comenzaron a cambiar adaptándose a las nuevas necesidades y retos [25].

Algunas de las ventajas que representa utilizar Arduino es que al ser una plataforma de código abierto lo hace económico e incluso se podría ensamblar de manera manual, su programación se puede extender y distribuir a través de librerías escritas en C++, además de que el Arduino IDE puede correr en la mayoría de los sistemas operativos [25].

2.6.2 Raspberry Pi

El Raspberry Pi es una computadora de código abierto del tamaño de una tarjeta de crédito, que originalmente fue creado como un dispositivo de bajo costo que ayude a mejorar las habilidades de programación y el entendimiento del hardware a un nivel pre-universitario. Pero debido a su tamaño y su precio asequible, fue adoptado rápidamente por desarrolladores para proyectos que requerían más que lo básico de un microcontrolador como Arduino [26].

Un Raspberry Pi es más lento que una computadora de escritorio o laptop moderna, pero, aun así, es una computadora Linux completa, que puede proveer lo mismo a un bajo consumo de energía. Hoy en día muchas distribuciones de Linux tienen versiones optimizadas para el Raspberry Pi, sin embargo, las dos opciones más populares son el Raspbian, que está basado en el sistema operativo Debian; y Pidora, basado en Fedora [27].

2.7 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas, como microcontroladores o computadoras [28]. Existen muchos lenguajes de programación, sin embargo, sólo se mencionarán algunos que son los que se utilizaron para este trabajo, tal como, Arduino, Python, PHP y JavaScript, además, se describirán los códigos HTML, CSS y el administrador de base de datos MySQL.

2.7.1 Lenguaje Arduino

El lenguaje de programación Arduino fue desarrollado para programar las tarjetas Arduino, es un lenguaje de alto nivel y está basado en C y C++, por lo que soporta todas las funciones de C y algunas de C++ [29].

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos funciones que encierran bloques que contienen declaraciones o instrucciones. La primera, llamada *setup()* es la parte encargada de recoger la configuración, debe contener la declaración de las variables y es la primera función en ejecutar el programa, el cual, sólo lo hace una vez. La otra parte es la función *loop()*, que contiene el código que se ejecutará continuamente, es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo. También se pueden declarar más funciones que contienen un conjunto de instrucciones que son ejecutadas cuando se llama a la función, con el objetivo de reducir el tamaño del código [30].

2.7.2 Python

Python es un lenguaje de programación interpretado orientado a objetos y de código abierto, caracterizado por su simple sintaxis que lo hace altamente legible y coherente. Python comúnmente se utiliza para la creación de códigos de mantenimiento y administración de sistemas operativos, ya que puede buscar archivos y direcciones, ejecutar otros programas y realizar procesamientos paralelos. Otras aplicaciones son la creación de interfaces gráficas, realizar tareas de redes, tanto en modo cliente como servidor, programación de base de datos, programación numérica y científica [31].

La mayoría de los programas escritos en Python no necesitan modificarse al cambiarse de plataforma computacional, por ejemplo, para pasar programas entre Linux y Windows basta con copiar y pegar el código entre las máquinas, esta es una de las razones por las cuales los desarrolladores de Raspberry Pi recomiendan Python para la programación de dicho dispositivo [31].

2.7.3 PHP

PHP es el motor detrás de millones de aplicaciones web dinámicas, ya que cuenta con una sintaxis legible, soporta diferentes sistemas operativos y servidores web, además, es una herramienta para el procesamiento de formularios HTML y crear páginas web, puede comunicarse con una multitud de base de datos y es capaz de utilizar numerosos protocolos de red. Estas características lo han convertido en un lenguaje ideal para el desarrollo web del lado del servidor [32].

2.7.4 MySQL

El sistema de administración de base de datos MySQL es popular por muchas razones, es rápido, fácil de configurar, usar y administrar. Se puede utilizar en una gran variedad de distribuciones de Linux y Windows, y los programas basados en MySQL se pueden escribir en muchos lenguajes. Combinar SQL con un lenguaje de programación genera un marco muy flexible para emitir consultas, procesar resultados e incrementar la capacidad de realizar operaciones complejas con la base de datos [33].

2.7.5 JavaScript

JavaScript es un lenguaje de programación que opera en el lado del cliente, haciendo las páginas web interactivas. Soporta la mayoría de los buscadores web, y se pueden escribir los programas en cualquier editor de texto. Normalmente se utiliza cuando se requiere ejecutar un código después de que la página ha cargado, como la actualización de datos visualizados en tiempo real [34].

2.7.6 HTML

HyperText Markup Language (HTML) es un estándar que define una estructura básica del código para la elaboración de páginas web, con el objetivo de reducir el costo y complejidad del desarrollo web. Fue desarrollado por el *World Wide Web Consortium*

(W3C), que es una comunidad internacional que desarrolla estándares para asegurar el crecimiento de la Web a largo plazo [35].

Una página web es puramente un archivo de texto, y recae en el navegador web, interpretar el código contenido en el archivo para mostrarlo en pantalla. Al ser un estándar, permite que cualquier página HTML, pueda ser interpretada de la misma manera por cualquier navegador web actualizado [35].

2.7.7 CSS

Las páginas web consiste en un código HTML que describe el contenido de la página, y un código CSS (Cascading Style Sheets), que le dicen al navegador cómo debe mostrarse el contenido, por ejemplo, los tipos de letra, los colores, sombras, gradientes, esquinas redondeadas, etc [35].

2.8 Protocolos de comunicación

Los protocolos de comunicaciones son sistemas de reglas que permite que dos o más máquinas se comuniquen para transmitir información físicamente [36]. Para el proyecto se emplearon tres principales protocolos de comunicación: Modbus RTU, I²C y UDP; los cuales se explican a continuación.

2.8.1 Modbus RTU

Modbus fue publicado en 1979 por la compañía Modicon, es un protocolo de comunicación serial y es el protocolo estándar en el sector industrial. Modbus permite que hasta 247 dispositivos esclavos se conecten a un único maestro dentro del mismo bus [37]. En la Figura 9 se muestra la estructura básica de la comunicación con Modbus RTU.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1-T2-T3-T4*	8 BITS	8 BITS	$n \times 8$ BITS	16 BITS	T1-T2-T3-T4*

Figura 9 Estructura de la comunicación Modbus RTU

2.8.2 I²C

I²C (*Inter Integrated Circuit*) es un protocolo de comunicación serial, que es comúnmente utilizado debido a su simplicidad. El bus I²C consiste en dos líneas, SDA y SCL. El primero es el que lleva la información de un dispositivo a otro, el segundo es el encargado de la sincronización de todos los dispositivos conectados al bus. En la Figura 10 se muestra el diagrama de conexión del bus I²C [38].

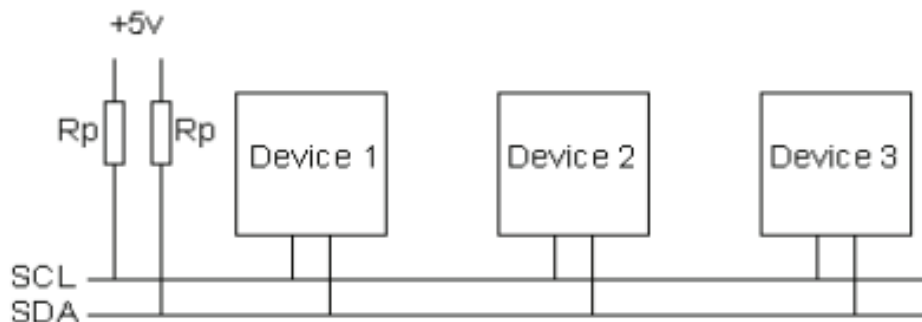


Figura 10 Diagrama de conexión del bus I2C [38]

2.8.3 UDP

El protocolo de comunicación UDP (User Datagram Protocol) es parte del modelo TCP/IP. Se caracteriza por no tener un flujo de control de datos, ya que, al haber un error, el paquete es desechado, por lo que no hay garantía de que todos los datos que se envían sean recibidos, así como el orden en el que llegan. En la comunicación UDP, quien inicia el proceso de intercambio de datos, es llamado cliente, mientras que, quien espera el intercambio de datos se le conoce como servidor [39].

2.9 Antecedentes

La ASHRAE en 2009 publicó dos estudios donde aplicaron el método de *horarios de ocupación* [2], el cual consiste en programar los equipos de aire acondicionado para que trabajen automáticamente en las horas en las cuales el lugar acondicionado está ocupado, y que apaguen cuando no lo está [40].

El primer estudio es una pequeña escuela localizada en Michigan, donde el sistema de aire acondicionado utiliza gas natural para calentar las aulas. El proyecto logró durante su primer año de haberse implementado un ahorro de 70,000 USD debido a la reducción de la energía consumida tal como se puede observar en la Figura 11. El ahorro energético se mantuvo casi estable durante los siguientes tres años, pero al haber un incremento en el gas natural, los ahorros se dispararon al doble, entre más costoso es el gas natural mayor es el ahorro que se genera [2].

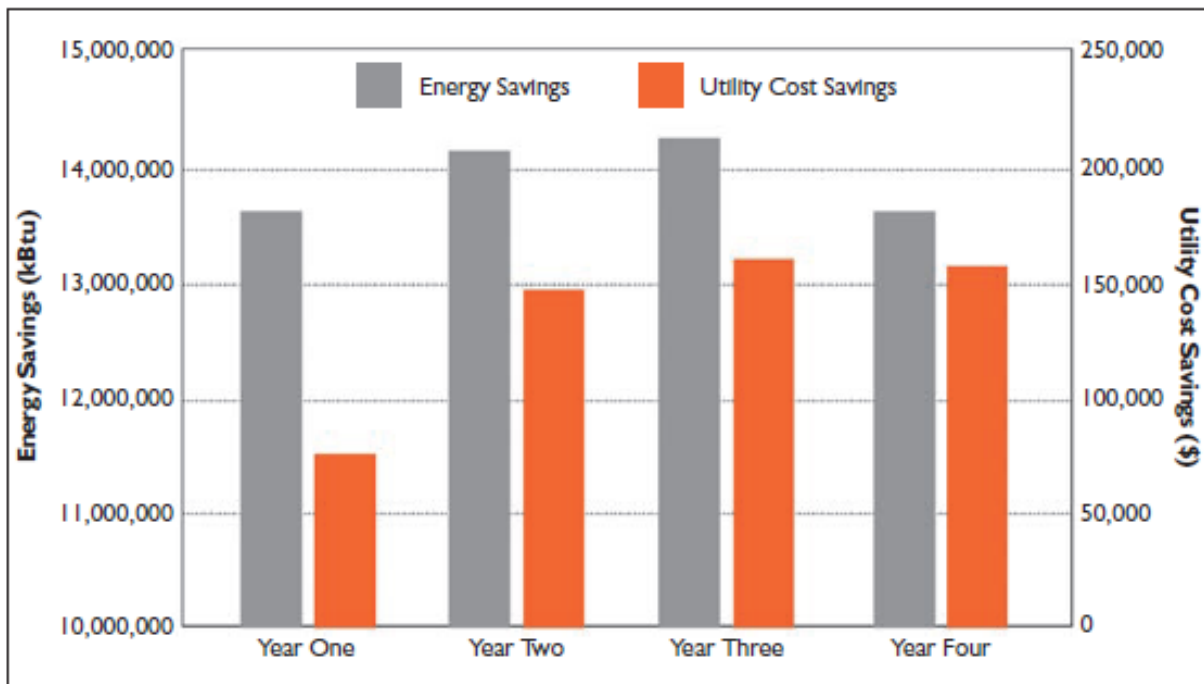


Figura 11 Energía que se dejó de consumir en una escuela de Michigan con un sistema de administración de energía en aires acondicionados [2]

El segundo caso publicado por la ASHRAE fue el de unas oficinas gubernamentales en Florida. Antes de que el proyecto se instalara, los costos por los servicios energéticos eran de 108,000 USD anuales. Durante el primer año se lograron ahorros de casi 15,000 USD, y el segundo de 20,000 USD (Figura 12) [2].

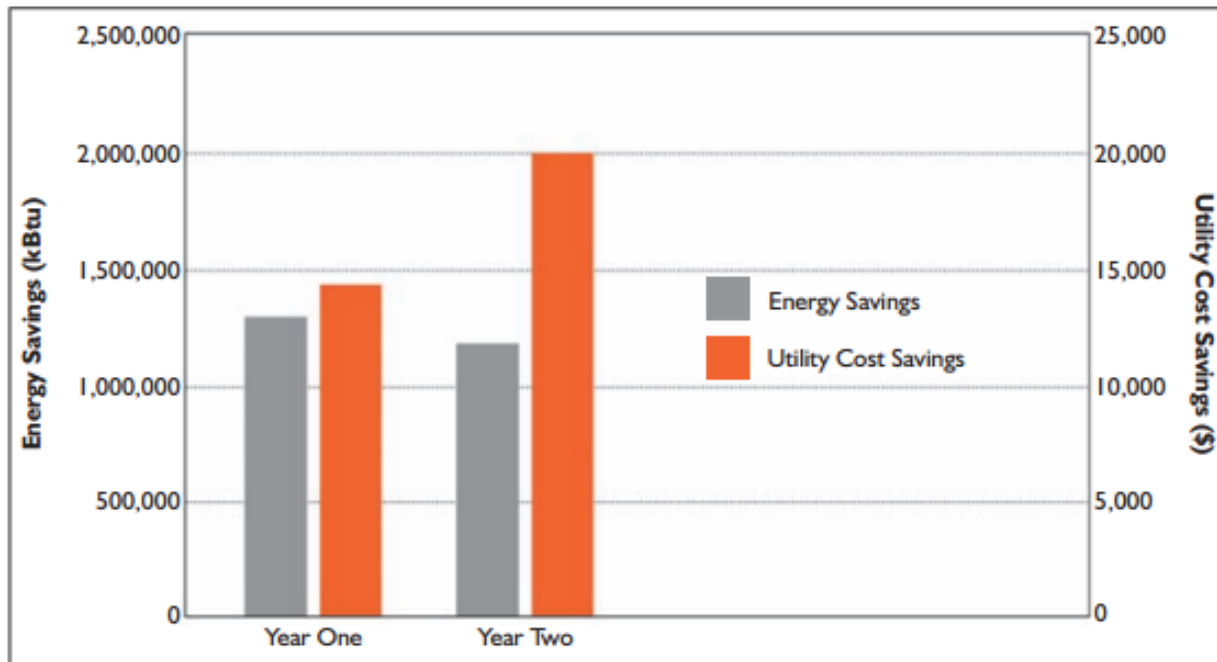


Figura 12 Energía que se dejó de consumir en unos edificios gubernamentales de Florida con un sistema de administración de energía en aires acondicionados [2]

Otro caso de estudio es la Universidad Politécnica de Valencia, España, donde se instaló un sistema de administración de energía que consiste en el control del sistema de aire acondicionado del campus, el cual está compuesto de muchos minisplits distribuidos en todos los salones y oficinas. Cada minisplit tiene instalado un Controlador Lógico Programable (PLC), que cuenta con un sensor de temperatura y tiene comunicación con un servidor central, que es el que toma las decisiones y retroalimenta al PLC para tomar la acción de encendido o apagado. Además, el sistema cuenta con medidores de energía eléctrica que permite obtener la carga del edificio, así como calcular la demanda y los ahorros energéticos. Mediante una interfaz web, se puede gestionar el sistema para programar por horarios los aires acondicionados del campus [41].

Durante los meses de noviembre y diciembre, los cuales son los meses del año en que las condiciones climáticas se mantienen más estables en Valencia y los climas trabajan en modo calefacción, se realizaron varias pruebas, tal como ajustar el setpoint, hacer precalentamiento antes de las horas laborales, y apagar los compresores horas antes de que termine el horario laboral. Esta última prueba se muestra en la Figura 13, donde dejaron solamente la ventilación desde dos horas antes del horario de salida (a las 21 horas) [41]. Se puede observar que, debido a la inercia térmica del edificio, no se vio significativamente afectada la temperatura.

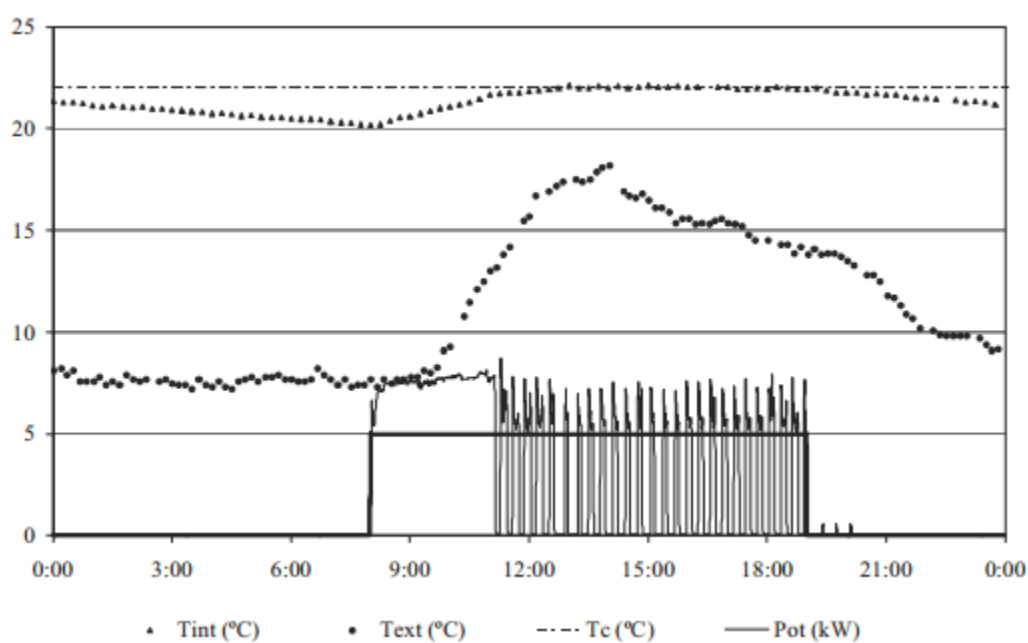


Figura 13 Apagando los compresores dos horas antes del horario de salida [41]

Con este sistema se ha obtenido una disminución en el consumo de energía eléctrica, así como reducción de la demanda máxima desplazando el pico de demanda manipulando los compresores de los minisplits, sin afectar el confort ni la calidad del aire interior, consiguiendo ahorros económicos significativos [41].

Por otro lado, las plataformas de código abierto están tomando terreno en los sistemas de control de edificios, como el desarrollado por Medilla Kusriyanto y Bambang Dwi Putra, que es una plataforma de *Casa Inteligente* que consiste en un Arduino Mega 2560

utilizado como servidor web y actuador, donde los usuarios pueden controlar la iluminación del hogar desde una interfaz amigable, mediante cualquier dispositivo que tenga acceso a la red de área local (LAN) [42].

3 Metodología

En este capítulo se describen las etapas del desarrollo del sistema de administración de energía presentado en esta investigación. Inicialmente se muestra el diseño y la arquitectura del sistema el cual se adapta al espacio físico de las empresas y edificios modernos, así como los componentes del mismo. Después se explica la metodología para poder manipular los equipos de aire acondicionados, y la programación de cada uno de los elementos del sistema para poder lograr su integración, permitiendo la transferencia de datos, visualización y control. Finalmente se describirá en qué consisten y cómo fueron desarrolladas las pruebas de confiabilidad que determinarán la robustez del sistema.

3.1 Arquitectura del sistema

Los principales componentes del sistema de administración de energía desarrollado son: un equipo central único en todo el sistema que contiene una interfaz web y una base de datos, los módulos remotos, que son sensores y controladores de los equipos de aire acondicionado; y un gateway que permite la comunicación entre el dispositivo central y los módulos remotos. Estos últimos dos componentes se ubican en cada una de las sucursales, tal como se puede ver en la Figura 14.

En la Figura 15 se muestran los componentes la arquitectura del sistema, donde el dispositivo central es un Raspberry Pi, el gateway son dos Arduinos Mega 2560 comunicados por Modbus RTU, y los módulos remotos son NodemCU, los cuales tienen un microcontrolador que es capaz de conectarse a la red vía Wifi. Estos últimos, cuentan con sensores y relevadores, con el fin de estar registrando la temperatura y la humedad de cada espacio refrigerado de las sucursales, y a su vez, controlar las señales del equipo de aire acondicionado.

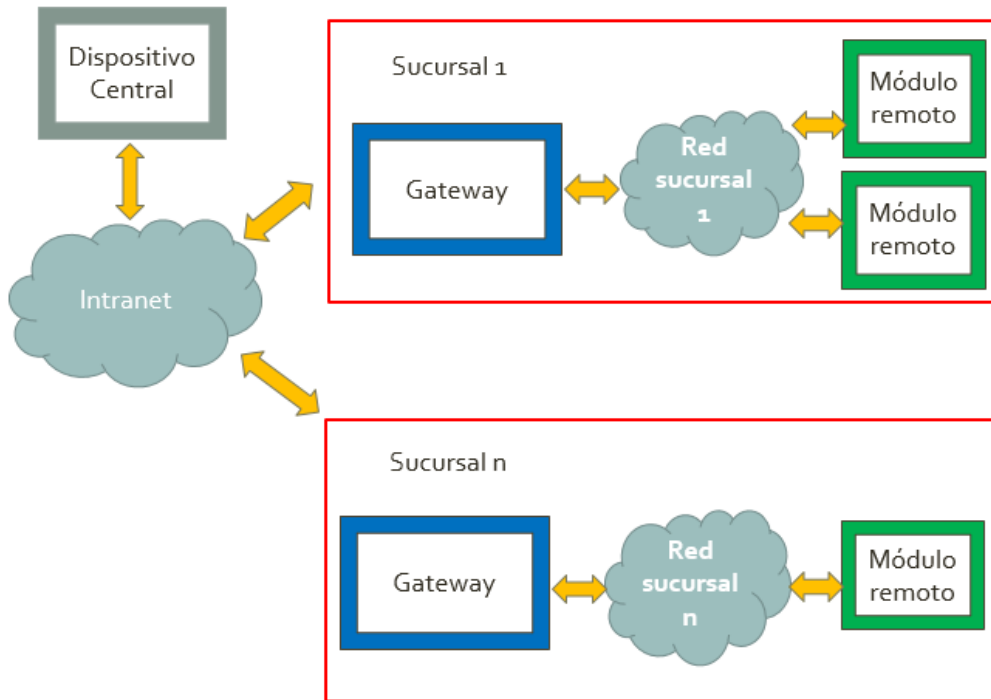


Figura 14 Arquitectura del sistema

Arquitectura del sistema

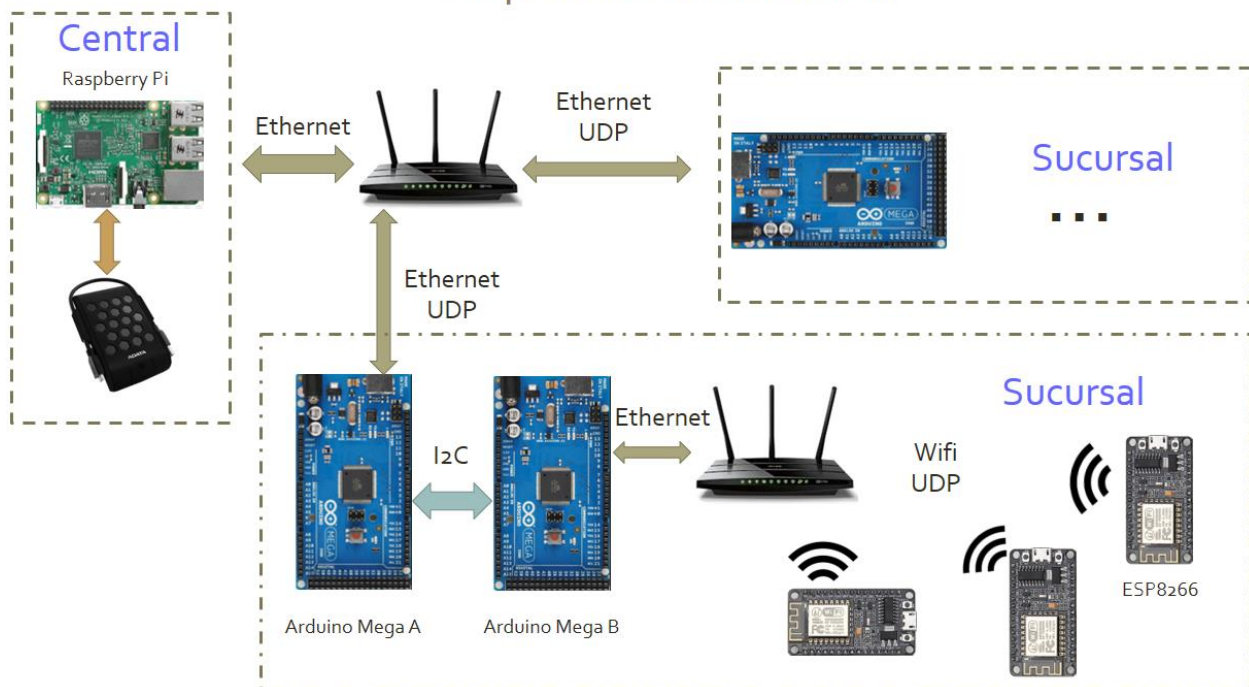


Figura 15 Componentes del sistema

3.2 Hardware

La lista de dispositivos que componen al sistema de administración de bajo costo se muestra en la Tabla 5:

Raspberry Pi 3 Model B
Arduino Mega 2560
Ethernet Shield W5100
Convertidor RS485 a TTL
RTC DS3231
NodemCU v1.0
DHT-22
MCP23017
ULN2803A
TQ2SS-L2-3V
Disco Duro (Opcional)

Tabla 5 Lista de dispositivos utilizados

A continuación, se describirán a detalle cada uno de los componentes listados.

3.2.1 Raspberry Pi 3 Model B

En todo el sistema solamente se necesita un Raspberry Pi 3 Model B, ya que éste fue seleccionado como dispositivo central, debido a su gran potencia computacional. En la Tabla 6 muestran las características de un Raspberry Pi 3 Model B.

CPU de 64-bit quad-core ARMv8 a 1.2 GHz
LAN inalámbrica 802.11n
1 GB RAM
40 GPIO
Puerto Ethernet
Espacio para tarjeta Micro SD

Tabla 6 Características de un Raspberry Pi 3 Model B [43]

Para poder utilizar un Raspberry Pi 3 Model B es necesario contar con una tarjeta Micro SD, a la cual se le carga el sistema operativo, los programas y los scripts. Adicional a esta, se puede agregar un disco duro conectado a cualquiera de sus puertos USB. Esta práctica solo se recomienda para sistemas que cuenten con 30 o más sucursales. Otro aspecto que cabe destacar, es que a pesar de que este Raspberry Pi puede conectarse a la red vía Wifi, se recomienda conectarlo directamente con un cable Ethernet para mayor estabilidad de conexión. En la Figura 16 se muestra un Raspberry Pi 3 Model B.

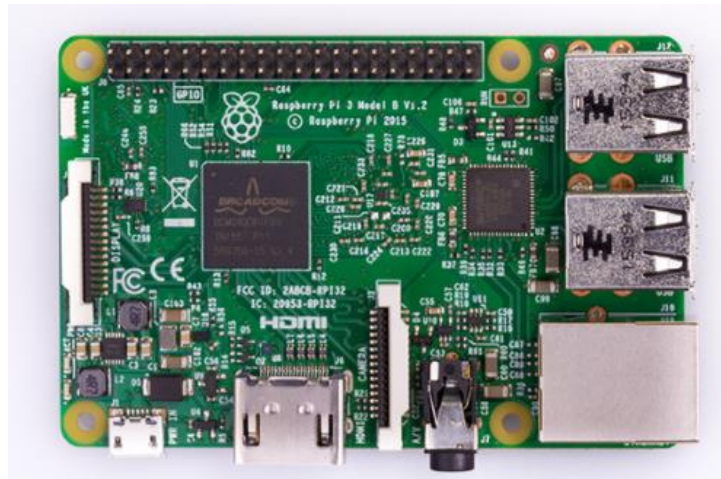


Figura 16 Raspberry Pi 3 Model B [43]

3.2.2 Arduino Mega 2560

Por cada sucursal se requieren dos Arduinos Mega 2560. Ambos, con sus respectivos Ethernet Shields W5100, cumplen con la función de gateway, el cual tiene la capacidad de interconectar al dispositivo central con los módulos remotos, aunque se encuentren en diferentes redes. Uno de los Arduinos, tiene conectado un RTC DS3231 al cual se comunica por medio del protocolo I²C. El Arduino Mega 2560 cuenta con las características listadas en la Tabla 7.

Los Arduinos Mega se comunican entre sí mediante Modbus RTU, por lo que ambos necesitan un convertidor RS485 a TTL. Esto con la finalidad de separar ambas redes con

un alto grado de seguridad mediante la comunicación serial. Uno de los Arduinos, tiene conectado un RTC DS3231 al cual se comunica por medio del protocolo I²C.

Microcontrolador Atmega 2560
8 KB SRAM
256 KB de Memoria Flash
4 KB EEPROM
54 pines digitales
16 pines analógicos

Tabla 7 Características de un Arduino Mega 2560 [44]

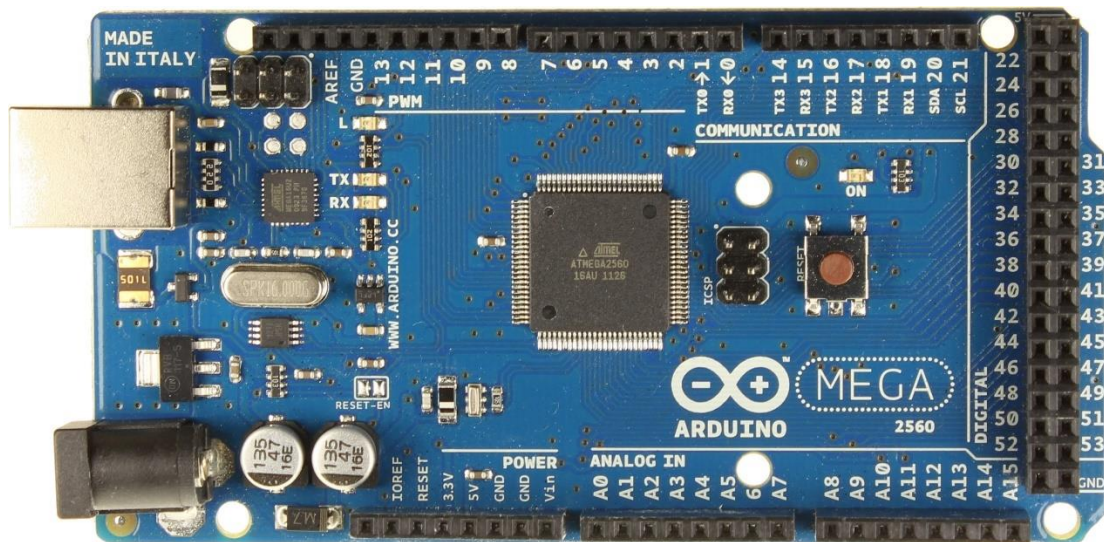


Figura 17 Arduino Mega 2560 [44]

3.2.3 Ethernet Shield W5100

El Ethernet Shield W5100 es utilizado para poder dar conectividad a los Arduinos Mega 2560 a las redes. Una vez conectados, estos pueden comunicarse con el dispositivo central o con los módulos remotos a través del protocolo de comunicación UDP, por lo que se requieren de dos por sucursal. En la Figura 18 se puede ver el Ethernet Shield.

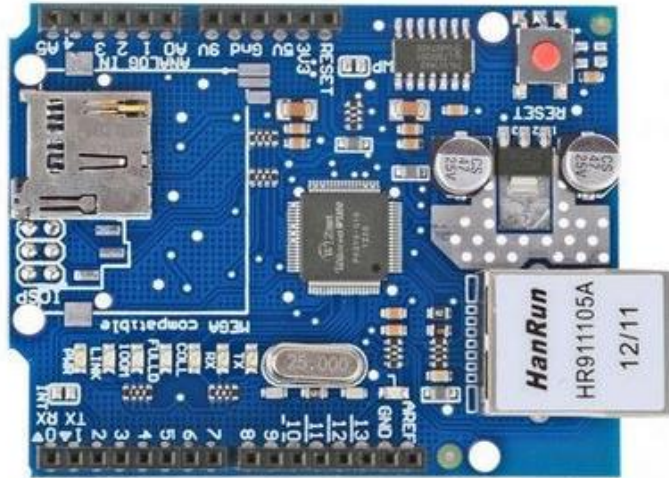


Figura 18 Ethernet Shield W5100 [45]

3.2.4 Convertidor RS485 a TTL

Los convertidores RS485 a TTL son indispensables para que los Arduinos Mega 2560 se puedan comunicar por Modbus RTU, es necesario dos de estos módulos por sucursal. Se necesita agregar algunas resistencias y capacitores para lograr un funcionamiento óptimo, tal como lo muestra la Figura 19. En la Figura 20 se muestra el Convertidor RS485 a TTL.

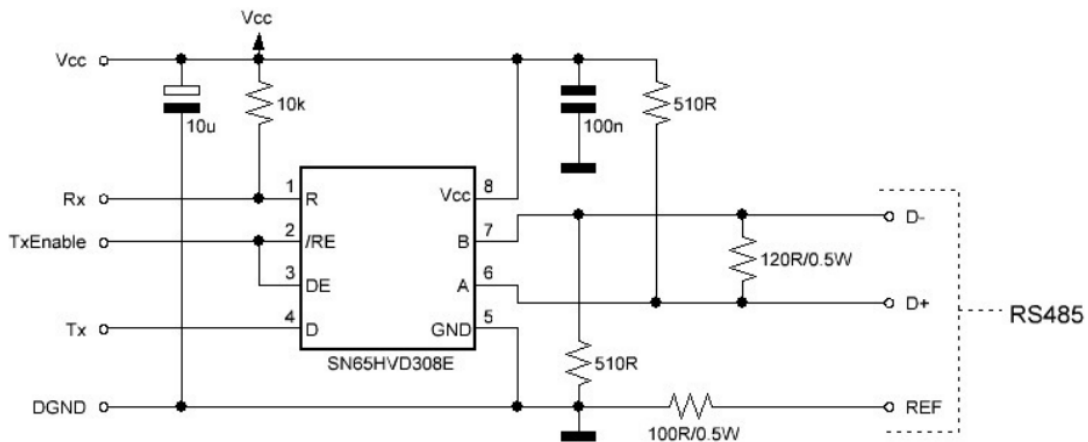


Figura 19 Conexión del Arduino Mega 2560 a Convertidor RS485 a TTL [46]

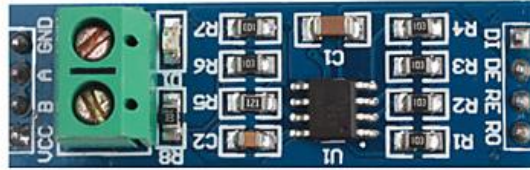


Figura 20 Convertidor RS485 a TTL [46]

3.2.5 RTC DS3231

El RTC (Real Time Clock) DS3231 es utilizado por uno de los Arduinos Mega 2560 con el fin de tener la fecha y hora exacta en todo momento para poder ordenar a los módulos remotos el estado de los equipos de aire acondicionado de acuerdo con la *ocupación por horarios* configurada desde el dispositivo central. En la Figura 21 se muestra dicho RTC.

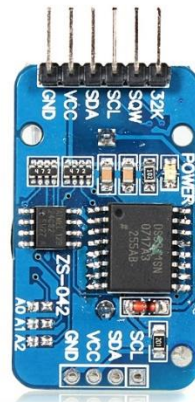


Figura 21 RTC DS3231 [47]

3.2.6 NodemCU v1.0

El NodemCU es el cerebro de los módulos remotos, a ellos se les conectan los sensores de temperatura y humedad DHT-22 y el módulo de expansión de puertos digitales MCP23017. Los NodemCU son los que contienen el código de control por setpoint y temperatura de los equipos de aire acondicionado. Por cada aire acondicionado se requieren entre uno y dos NodemCU.

Por su pequeño tamaño, bajo costo, bajo consumo de energía y su capacidad de conectarse a internet de manera inalámbrica son los motivos por los cuales el NodemCU fue seleccionado. En Tabla 8 se enlistan las características del NodemCU.

32-bit MCU de baja energía
LAN inalámbrica 802.11 b/g/n
Protocolos TCP/IP integrados
Unidad de alimentación de energía integrada
WiFi 2.4 GHz, soporta WPA/WPA2

Tabla 8 Características del NodemCU v1.0 [48]

En la Figura 22 se puede observar la pequeña y versátil forma del NodemCU v1.0.

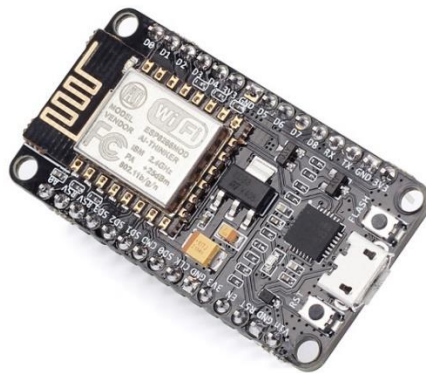


Figura 22 NodemCU v1.0 [48]

3.2.7 DHT-22

El DHT-22 es un sensor de temperatura y humedad cuyo rango de medición está entre 40 y 125 °C, con una precisión de 0.5 °C; y miden humedad relativa entre 0 y 100% con una precisión del 2-5% [49]. Hay un DHT-22 por cada módulo remoto del sistema. En la Figura 23 se puede ver el sensor de temperatura y humedad.

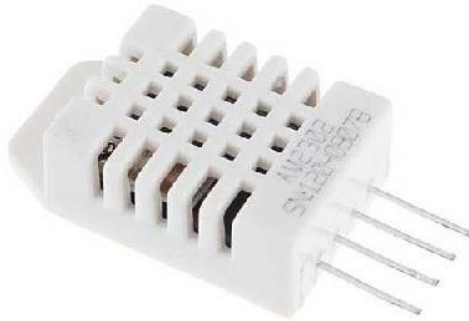


Figura 23 DHT-22 [49]

3.2.8 MCP23017

El MCP23017 es un circuito integrado expensor de puertos digitales, otorgando 16 puertos extras. Es utilizado en el módulo remoto para incrementar los puertos del NodemCU, los cuales son necesarios para poder controlar los relevadores TQ2SS-L2-3V mediante ULN2803A. El NodemCU se comunica con el MCP23017 con el protocolo de comunicación I²C. En la Figura 24 se puede ver el MCP23017.



Figura 24 MCP23017 [50]

3.2.9 ULN2803A

El ULN2803A es una matriz de transistores Darlington de 50V, 500mA, consiste en ocho transistores NPN que cuentan con salidas de alto voltaje con diodos de cátodo común para conmutar cargas inductivas [51]. En el módulo remoto, estas cargas inductivas son los relevadores TQ2SS-L2-3V. Debido a que solo cuenta con 8 transistores, y se

requieren de 10, se necesitan dos ULN2803A por cada módulo remoto. En la Figura 25 se muestra el ULN2803A.



Figura 25 ULN2803A [51]

3.2.10 TQ2SS-L2-3V

El TQ2SS-L2-3V es un relevador de pulsos que puede cerrar o abrir circuitos dependiendo del lado del relevador en que se aplica un voltaje. Este relevador funciona cuando se le aplica a la bobina 3 V y 66.7 mA [52], y su lógica se muestra en la Figura 26, donde el cuadro negro representa el lado del relevador pulsado. Por cada módulo remoto se necesitan 5 relevadores TQ2SS-L2-3V, uno para cada línea de control de los aires acondicionados. La Figura 27 muestra un TQ2SS-L2-3V.

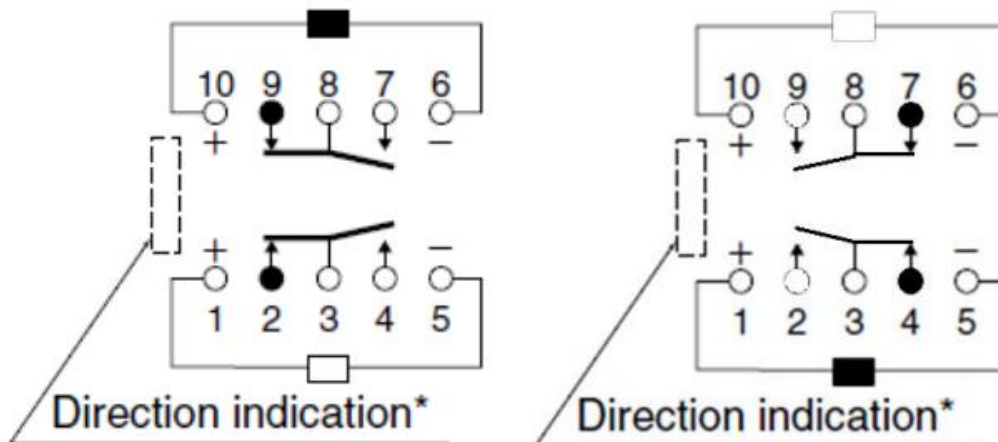


Figura 26 Funcionamiento del relevador TQ2SS-L2-3V



Figura 27 TQ2SS-L2-3V [52]

3.3 Control del sistema

El sistema hará control de los equipos de aire acondicionado con el modo de *dos posiciones*, bajo los criterios de *horarios de ocupación* y setpoints de temperatura, fijados por un administrador, quien es el usuario final. El administrador podrá configurar en qué días y horarios funcionarán los ventiladores, los compresores o calefactores. Estos horarios se pueden cambiar con facilidad desde la interfaz web del sistema en el dispositivo central, quien se encargará de distribuir la información hacia los gateways, que, a su vez, ellos envían el estado del aire acondicionado (apagado, ventilador, o compresores) a los dispositivos remotos.

Una vez que los módulos remotos conocen el estado del aire acondicionado, controlan el sistema de acuerdo con los setpoints de refrigeración y de calefacción, según la temperatura del interior, tal como se puede observar en la Figura 28.

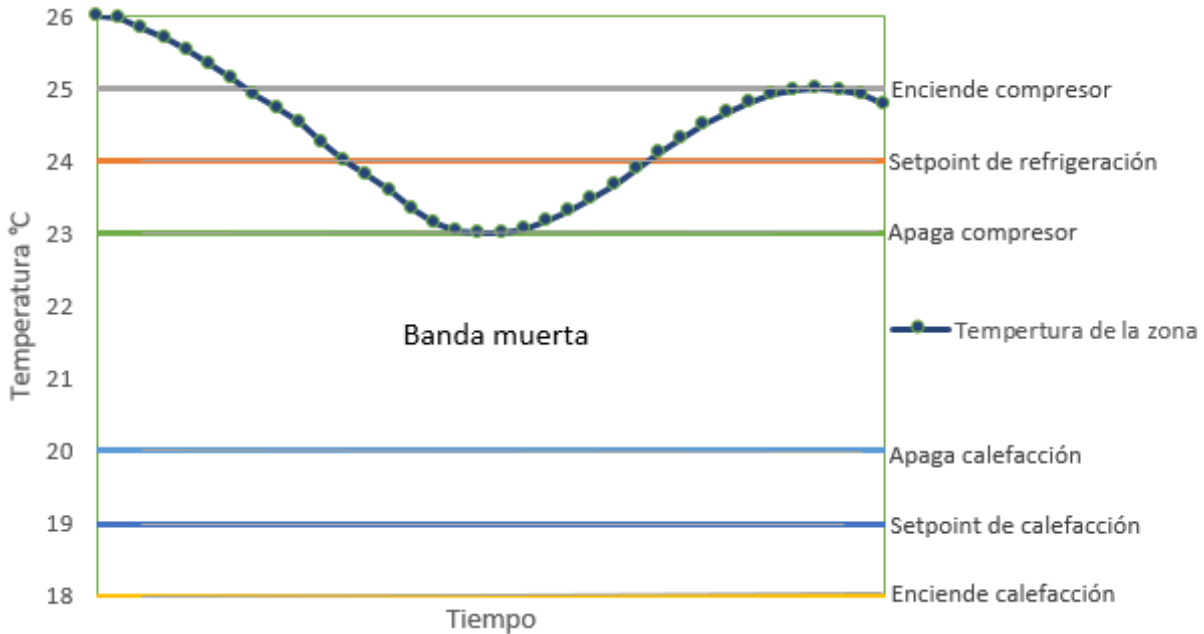


Figura 28 Lógica de control de acuerdo con Setpoints y temperatura

La Figura 28 muestra un ejemplo donde se definieron aleatoriamente los setpoints de refrigeración y de calefacción, siendo 24 y 19 °C respectivamente. Ya que la mayoría de los aires acondicionados tipo paquete cuentan con dos compresores y dos calefactores, cuando la temperatura sea igual o mayor a 27 °C, el módulo remoto encenderá dos compresores, si la temperatura está entre 25 y 27 °C, solo encenderá uno, y en caso de llegar a 23 °C, lo apagará. Por otro lado, si la temperatura es de 16 °C o menor, el módulo remoto encenderá los dos calefactores, si está entre 16°C y 18 °C, prenderá solo uno, y los apagará al alcanzar los 20 °C. En caso de que la temperatura se encuentre entre los 20 y 23 °C, los módulos remotos no permitirán que ni los compresores ni los calefactores entren en operación. Para esta situación en particular, la temperatura inicial de la zona refrigerada era de 26 °C, por lo que encendió el compresor, hasta llegar a los 23 °C, temperatura a la cual se apaga el compresor mientras la temperatura vuelve a subir a 25 °C donde prende de nuevo y así se mantiene oscilando.

Los módulos remotos cuentan con cinco relevadores de pulso, que permiten abrir y cerrar circuitos de control a un bajo consumo de energía. Con esto controlan las señales que el termostato original del sistema de aire acondicionado envía al equipo. Estas señales son:

ventilador, compresor 1, compresor 2, calefactor 1, calefactor 2. Bajo este esquema, los módulos remotos se pueden instalar sin tener que eliminar el termostato original (véase Figura 29), lo que le otorga cierto control a las personas que se encuentren en el área refrigerada, sin embargo, este control está restringido a las operaciones que involucren ahorro de energía, como, por ejemplo, aumentar el setpoint de refrigeración, o disminuir el setpoint de calefacción.

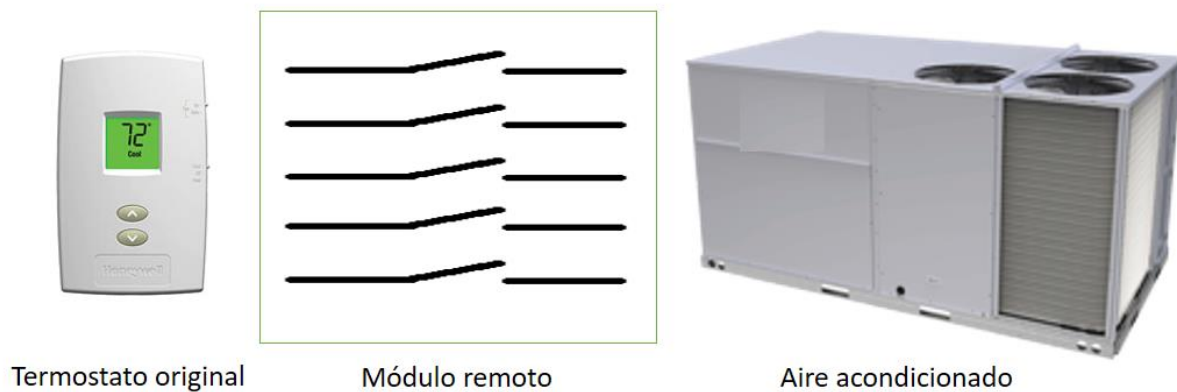


Figura 29 Control del módulo remoto

3.4 Programación

La programación del sistema se puede dividir en seis partes principales: la interfaz web, la comunicación del dispositivo central hacia los gateways, la comunicación de los gateways hacia el dispositivo central, la comunicación de los gateways con los módulos remotos, la comunicación serial interna de los gateways, y los módulos remotos. Los primeros dos se pueden encontrar en el algoritmo del Raspberry Pi, el tercero en la programación del Arduino Mega A, el cuarto en el Arduino Mega B, el quinto en los códigos del Arduino Mega A y B; y el último en la programación del ESP8266.

3.4.1 Algoritmo Raspberry Pi

El dispositivo central es un Raspberry Pi 3 Model B. Dentro de él, se manejan dos códigos principales, la interfaz web y la comunicación con los gateways. El primero está

programado principalmente en PHP, y apoyándose con HTML, CSS3 y JavaScript, el mapa de la interfaz se muestra en la Figura 31. El segundo está programado completamente en Python, y su diagrama de flujo se puede ver en la Figura 30. Ambos códigos tienen acceso a la base de datos de MySQL para escribir y leer datos.

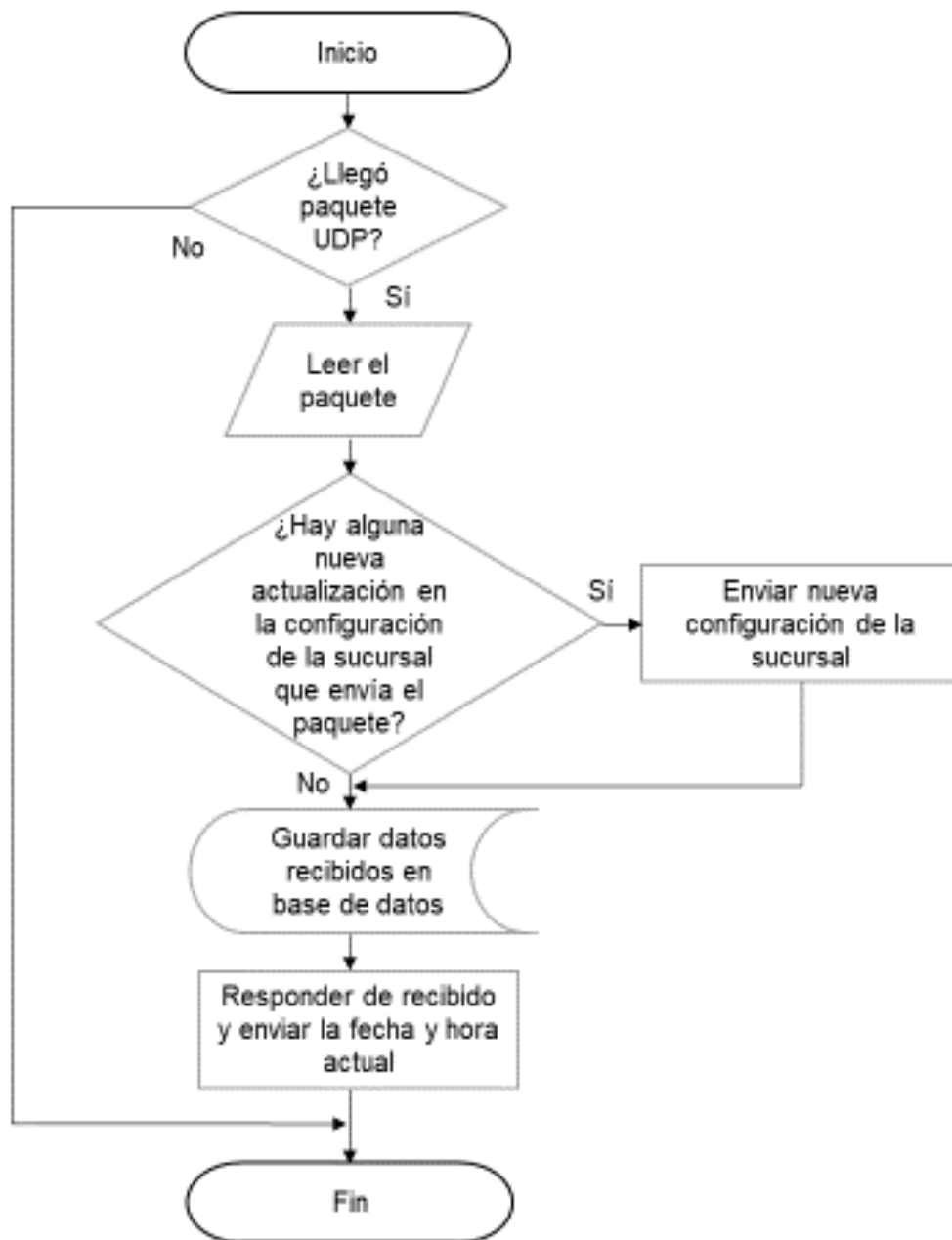


Figura 30 Diagrama de flujo de código en Python del Raspberry Pi

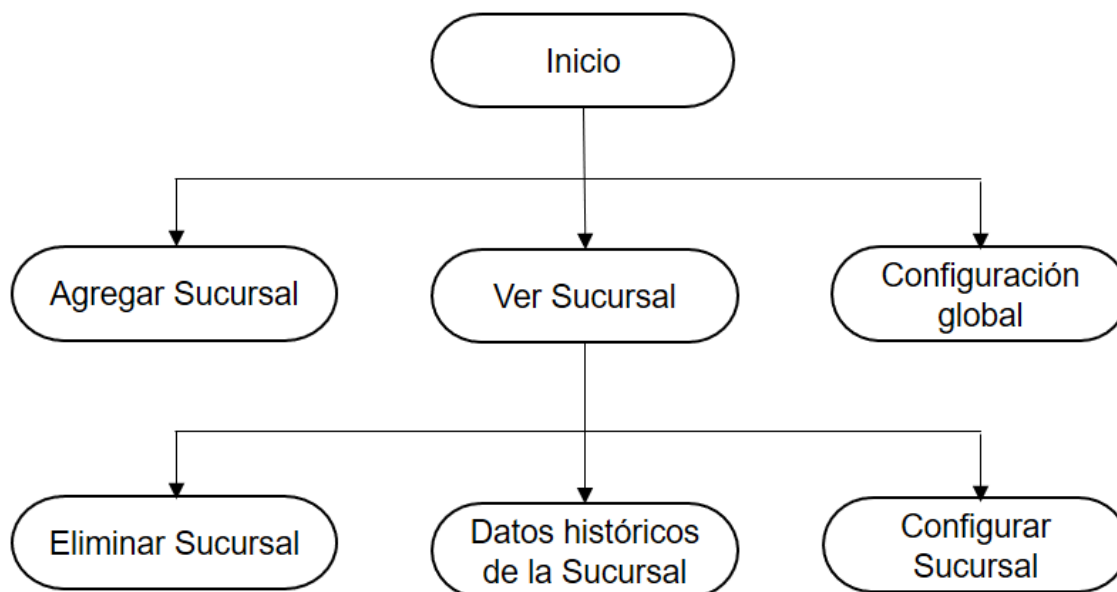


Figura 31 Mapa de la interfaz Web

A continuación, se describirá cada una de las páginas que componen a la interfaz web del sistema, así como cada función del código en Python.

3.4.1.1 Inicio

Es la página principal del sistema de administración de energía, la cual accede a la base de datos y descarga la lista completa de sucursales existentes para su selección, donde dando clic en una de ellas lo enviará a la página *Sucursal*. Además, desde esta página se tiene acceso a las páginas *Agregar Sucursal* y *Configuración Global*.

3.4.1.2 Agregar Sucursal

Aquí se pueden crear nuevas sucursales. Es necesario dar el número de identificación de la sucursal a crear, el nombre, número de aires y los tipos de módulos remotos de cada aire (Termostato, Relevador y Sensor). Una vez dada de alta la información, se guarda en la base de datos y crea nuevas tablas dentro de ella, donde se ingresarán todos los registros de temperatura, humedad y alarmas de dicha sucursal.

3.4.1.3 Configuración

A esta página se accede enviando con el método POST la ID de la sucursal que se quiere configurar, en caso de no enviar nada, se desplegará el modo *Configuración Global*, donde cualquier cambio que se realice, aplicará para todas las sucursales. Los parámetros que se pueden configurar desde esta página son: setpoint de refrigeración, setpoint de calefacción y los horarios de operación.

3.4.1.4 Sucursal

Para poder visualizar contenido en esta página es necesario enviar la ID de la sucursal que se quiere consultar con el método POST. La información que se despliega es una tabla con la temperatura y humedad de cada zona acondicionada en la sucursal, así como una gráfica de cada una de ellas. Tanto la tabla como las gráficas, se actualizan automáticamente cada que se ingresan nuevos datos en la base de datos. En caso de que no se envíe la ID, o está no exista, se redireccionará a la página de *Inicio*.

3.4.1.5 Datos históricos

Al igual que en la página *Sucursal*, se debe enviar una ID de sucursal válida sino será redireccionado a la página *Inicio*. Aquí se muestran por default todos los datos graficados de temperatura y humedad registrados en cada una de las zonas acondicionadas de la sucursal consultada, a su vez, se puede ver solamente un día, un mes, o un año. Además de solo visualizar las gráficas, se pueden descargar los datos brutos de lo que se está viendo.

3.4.1.6 Eliminar Sucursal

Esta página elimina la sucursal y todos sus datos. Debido a que una vez eliminada la sucursal no se puede recuperar la información, la página genera varias ventanas de alerta, con el fin de evitar que el usuario cometa el error de eliminar algo que no quería.

3.4.1.7 UDP

La función UDP es la principal en cuanto a la comunicación entre el Raspberry Pi y los gateway. Esta función está en la espera de paquetes recibidos cada cinco minutos por parte de los gateway, los cuales contienen la información del estado de la sucursal en la que se encuentran. Una vez recibido estos paquetes, la función se encarga de decodificar y cerciorarse de que la información no esté corrupta. Una vez aprobada la validación, se verifica si hubo cambios en la configuración de la sucursal que envía la información, en caso de ser así, esta es enviada con la función *send_conf*, y en caso de no encontrar cambios recientes, guarda la información recibida en la base de datos.

3.4.1.8 Get_conf

Esta función es llamada cuando se detecta que hubo cambios recientes en la configuración de una sucursal. Lo que hace es enviar toda la información de la configuración de la sucursal almacenada en la base de datos, tal como, los setpoints de refrigeración y calefacción, horarios de operación, número de aires acondicionados, y tipo de módulos remotos por equipo.

3.4.1.9 Main_alert

Revisa continuamente que se reciba información cada cinco minutos de cada una de las sucursales dadas de alta en el sistema de administración de energía. En caso de que alguna de ellas haya fallado, se activará la alerta y se guardará en un registro con la hora en que sucede dicho evento, con la finalidad de llevar un control de diagnóstico, como se describirá en la sección 3.5.

3.4.2 Algoritmo Arduino Mega A

El Arduino Mega A es el componente del gateway que está conectado a la misma red que el dispositivo central, y es el puente de comunicación entre el Arduino Mega B y el Raspberry Pi, utilizando Modbus RTU como maestro y UDP respectivamente, tal como

lo describe el diagrama de flujo de la Figura 32. A continuación, se muestran las funciones programadas en este dispositivo.

3.4.2.1 startEthernet

Es llamada al inicio de todo el código y cada que se detecte una pérdida en la conexión con la red, y lo que hace es inicializar dicha conexión.

3.4.2.2 Modbus_update

Inicializa una consulta al Arduino Mega B por Modbus RTU, ya sea para leer la información de los sensores de los módulos remotos, o para escribir la configuración de la sucursal. Esta función es llamada constantemente para mantener la información intercambiada lo más actualizada posible.

3.4.2.3 Checker

Esta es otra de las funciones de diagnóstico, está monitoreando la comunicación por Modbus RTU con el Arduino Mega B, con el objetivo de detectar fallas. En caso de ser así, el Arduino Mega A, envía la alerta al dispositivo central, y este la guarda en la base de datos.

3.4.2.4 sendUDP

Es usada cada cinco minutos, y es la función encargada de enviar la información más actualizada al dispositivo central por medio de UDP. Los paquetes UDP enviados se componen de la temperatura, humedad relativa, alerta de cada sensor y alerta de los módulos remotos de cada uno de los equipos de aire acondicionado de la sucursal, así como la alerta del Arduino Mega B que se obtiene con la función *checker de la sección 3.4.2.3*.

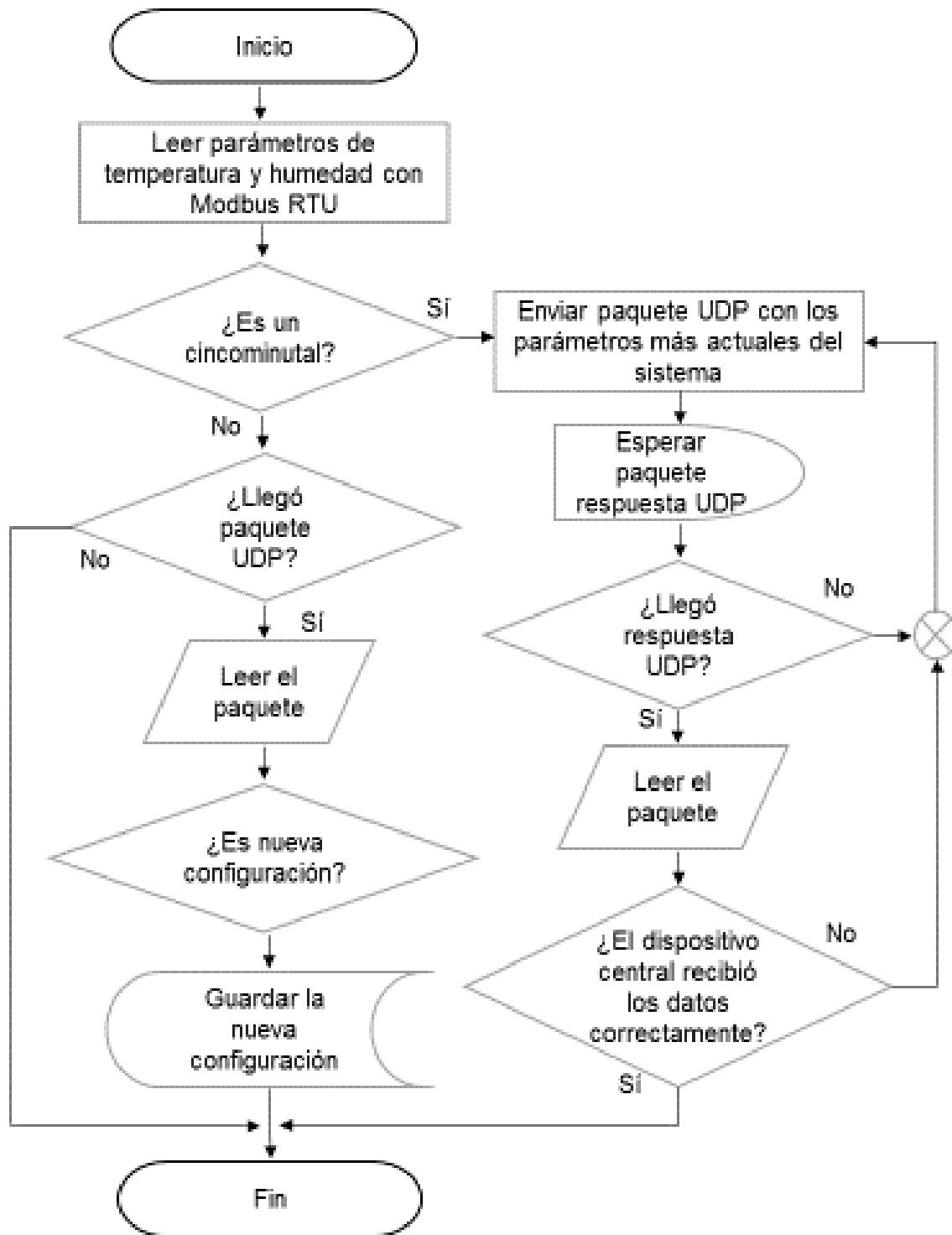


Figura 32 Diagrama de flujo de código del Arduino Mega A

3.4.2.5 receiveUDP

Una vez enviado el paquete UDP al dispositivo central con la función *sendUDP* de la sección 3.4.2.4, se espera una respuesta por parte del Raspberry Pi. Esta es la función encargada de recibir y procesar dicha respuesta. Si la respuesta no llega en cinco segundos, se vuelve a llamar a la función *sendUDP*, caso contrario, se interpreta el mensaje, y si el contenido es un "1", significa que el mensaje fue recibido satisfactoriamente; si el contenido es un "2", significa que el dispositivo central detectó una configuración reciente y la envía en este paquete, por lo que se procede a decodificarlo y posteriormente llama a la función *updateeeprom* de la sección 3.4.2.7.

3.4.2.6 Gettime

Es llamada cada hora para sincronizar la fecha y hora del Gateway con la del dispositivo central, posteriormente, es ingresada al RTC para tener acceso a ella en caso de algún apagón o anomalía.

3.4.2.7 updateeeprom

Esta función guarda los datos más recientes de la configuración en la memoria EEPROM, para que se pueda retener más allá de apagones o reinicios del sistema. La información que se almacena en la memoria EEPROM son la dirección IP del dispositivo central, el número de sucursal en la que se encuentra dicho gateway, los setpoints de refrigeración y calefacción, los horarios de operación, el número de aires acondicionados y los tipos de módulos remotos de cada uno de ellos.

3.4.2.8 readeeprom

Esta función se llama cada que el Arduino Mega inicializa el sistema. Lee toda la configuración de la memoria EEPROM para poder operar el sistema de la sucursal de acorde a lo fijado por el usuario en el dispositivo central.

3.4.2.9 reseter

Es la encargada de reiniciar al Arduino Mega todos los días a las 03:02:30 am, con el fin de mantener estabilidad en el dispositivo.

3.4.2.10 wdt

Es un *watchdog* que está monitoreando partes del código, estas partes son: el ciclo principal (*void loop*), la comunicación por Modbus RTU y la comunicación por UDP. Consiste en un contador incremental por segundo que se reinicia al ejecutarse las partes del código antes mencionadas, en caso de que el contador llegue a cierto valor, el Arduino Mega es reiniciado. Los valores definidos son: 60 segundos para el ciclo principal, 750 segundos para la comunicación por Modbus RTU y 900 segundos para la comunicación UDP.

3.4.3 Algoritmo Arduino Mega B

El Arduino Mega B es el componente restante del gateway, el cual está conectado a la red de los módulos remotos, recibiendo de ellos los datos de temperatura y humedad de cada zona acondicionada por medio de UDP. A su vez, es el esclavo en la comunicación por Modbus RTU, dejando a disposición del Arduino Mega A toda la información que recibe de los módulos remotos, tal como lo describe el diagrama de flujo de la Figura 33. A continuación, se muestran las funciones programadas en este dispositivo, sin embargo, algunas de ellas son las mismas que en el Arduino Mega B, como, *startEthernet*, *gettime*, *updateeprom*, *readeprom*, *reseter* y *wdt*, por lo que serán omitidas en la explicación.

3.4.3.1 getstate

Esta función se encarga de obtener el estado del aire acondicionado (apagado, ventilador o refrigeración/calefacción) de acuerdo al horario de operación en la configuración y la fecha y hora actual.

3.4.3.2 receiveUDP

Al igual que la función *receiveUDP* de la sección 3.4.2.5, ésta se encarga de recibir y decodificar todos los paquetes recibidos por UDP, sin embargo, en el Arduino Mega B, son recibidos por parte de los módulos remotos. Los módulos remotos envían paquetes cada cinco o quince minutos, dependiendo si son alimentados de la red eléctrica o con una batería, y pueden llegar en cualquier momento. Los datos que contienen los paquetes son la temperatura, humedad y alerta del sensor del módulo remoto. Una vez que llega un paquete y se decodifica exitosamente, el Arduino Mega B envía los setpoints de refrigeración y calefacción, así como el estado de operación obtenido con la función *getstate* de la sección 3.4.3.1.

3.4.3.3 alerta

Esta función está monitoreando constantemente los paquetes recibidos de los módulos remotos, por lo que, al detectar alguna falla, genera una alerta que lo pone a disposición en los registros Modbus para que el Arduino Mega A pueda consultarlos y transmitirlos al dispositivo central. Aunque la comunicación UDP se encuentre bien, se puede generar una alerta en caso de que el módulo remoto haya detectado falla en el sensor. Las alertas se explicarán más a detalle en la sección 3.5.

3.4.3.4 gettemp

Por cada aire acondicionado puede haber hasta dos módulos remotos, una combinación de relevador y sensor de temperatura, o termostato y sensor de temperatura. En caso de que un equipo tenga la segunda combinación, ambos módulos remotos están midiendo tanto temperatura como humedad, por lo que esta función calcula un promedio de ambas mediciones, el cual se pone a disposición en los registros Modbus para que pueda ser consultado.

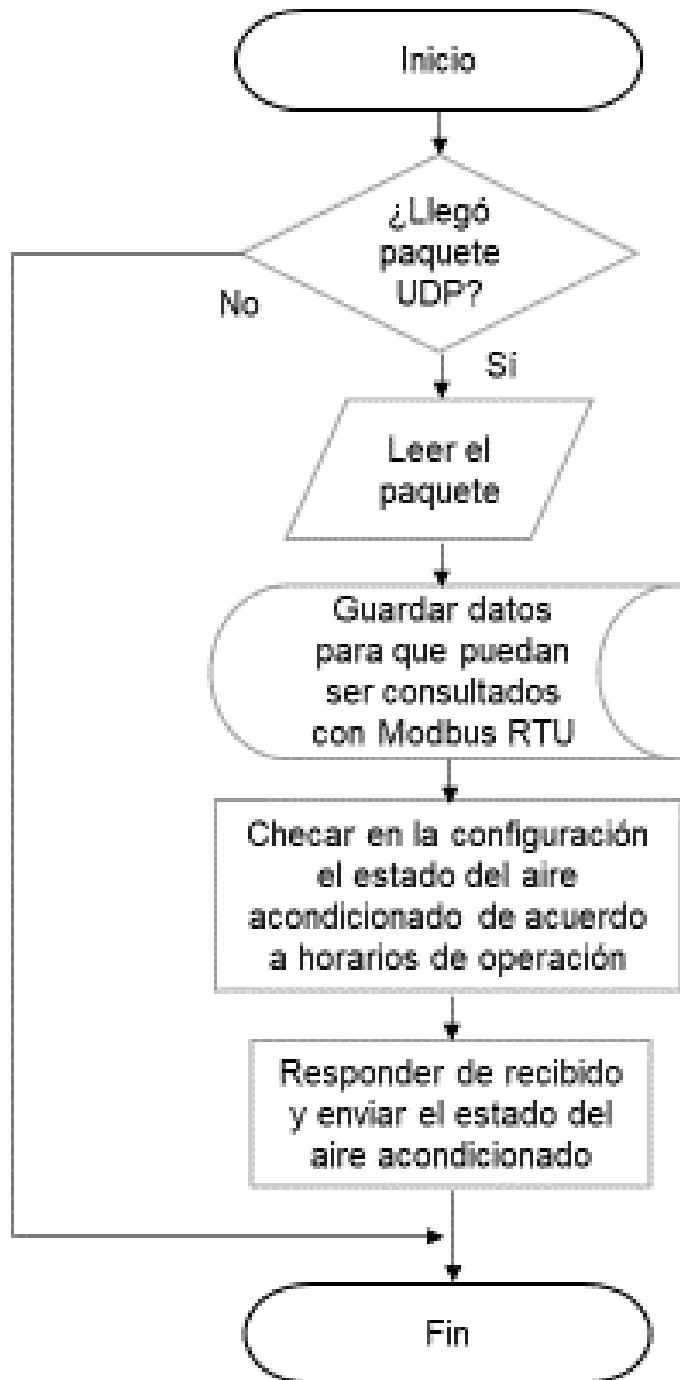


Figura 33 Diagrama de flujo de código del Arduino Mega B

3.4.3.5 webservice

Se puede acceder a la página de configuración del gateway a través del Arduino Mega B desde la red de los módulos remotos. En esta página de configuración se puede ingresar la dirección IP del dispositivo central, así como asignar el número de sucursal en la cual se encuentra el gateway.

3.4.4 Algoritmo ESP8266

Los ESP8266 son el cerebro de los módulos remotos, los cuales se encargan de medir temperatura y humedad y/o controlar los equipos de aire acondicionado. Los módulos remotos también cuentan con un pequeño webservice para configurarlo. En la Figura 34 se muestra el diagrama de flujo del código del ESP8266 y en las siguientes secciones se describen los códigos programados en este dispositivo.

3.4.4.1 Inicio

Esta es la página principal del webservice del módulo remoto, y solamente cuenta con dos botones, *configuración* e *información*, este webservice fue desarrollado en base a la librería *Web_Config* de John Lassen [53]. A esta página se puede acceder conectándose al WiFi del módulo remoto y escribiendo la dirección *192.168.4.1*, en la sección 4.1 se darán más detalles de este tema.

3.4.4.2 Configuración

Esta es la segunda página del webservice, y es donde se configuran los módulos remotos. Los parámetros que se tienen que dar de alta en esta página son: Nombre del módulo remoto, número de aire, tipo de dispositivo (sensor, relevador o termostato), tipo de alimentación (red eléctrica o batería), IP del Gateway, y, por último, nombre y contraseña de la red a la cual se debe conectar.

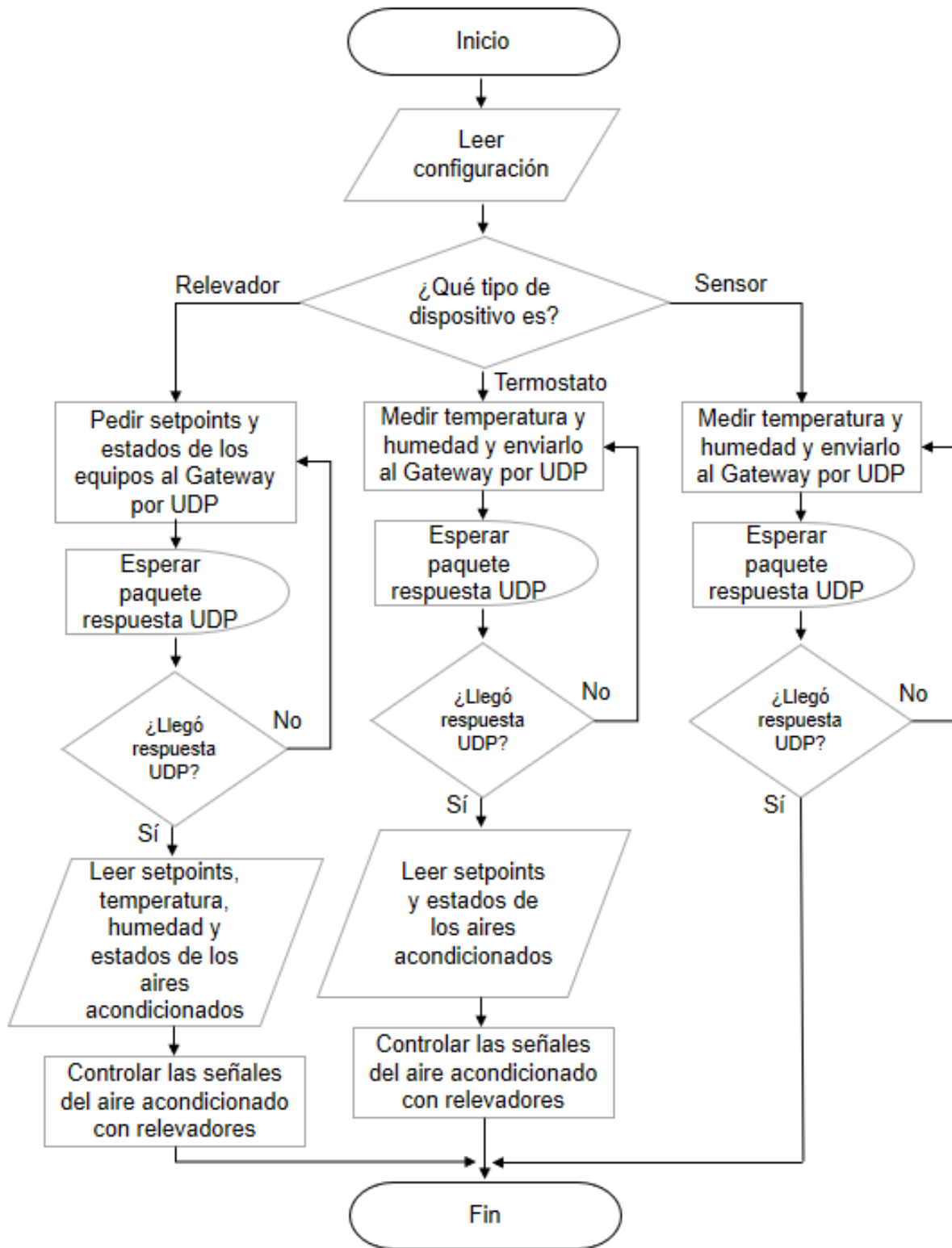


Figura 34 Diagrama de flujo de código del ESP8266

3.4.4.3 Información

Es la tercera página del webserver, y sirve para cerciorarse de que la configuración se realizó correctamente, ya que muestra si el módulo remoto se ha conectado a la red predefinida, así como su dirección IP y MAC.

3.4.4.4 Código principal del ESP8266

A diferencia del Raspberry Pi, y los Arduinos Mega A y B, el ESP8266 está escrito en un solo código sin funciones (además de las páginas del webserver), tal como se muestra en la Figura 34.

Inicialmente el módulo remoto despierta, carga la configuración de su memoria EEPROM, y revisa si el botón de configuración está siendo oprimido, y de ser así, inicia la señal WiFi para que puedan acceder a su webserver y configurarlo, en este modo dura cinco minutos, ya que después continua con el resto del código. En caso de no haber estado oprimido dicho botón, no inicia el webserver y continua.

Lo siguiente que hace es conectarse a la red especificada en la configuración, si lo logra, continua con el código, y en caso de que no, vuelve a dormirse por otro periodo de tiempo. Los periodos de tiempo en que se duerme el módulo remoto están ligados al tipo de alimentación que tiene, ya que, si está conectado a la red eléctrica, dura cinco minutos dormido, y si está alimentado con batería dura quince minutos, esto con el objetivo de ahorrar energía.

Una vez conectado a la red, verifica qué tipo de dispositivo es, si es termostato o sensor, mide la temperatura y humedad, en caso de que falle al hacer esto, generará la alarma del sensor, la cual indica lo sucedido. Después envía los datos medidos o la alarma al gateway por UDP, y se espera unos cuantos segundos para recibir la respuesta, y si esta no llega, vuelve enviar otro paquete UDP, hasta un máximo de cinco. En caso de que el módulo remoto sea de tipo relevador, enviará el paquete UDP pidiendo los datos necesarios para el control del aire acondicionado. Cuando se recibe la respuesta UDP,

se decodifica y se obtienen los setpoints de refrigeración y de calefacción, así como el estado de operación del equipo (apagado, ventilador o refrigeración/calefacción).

Con la información recibida del gateway, y en caso de ser termostato o relevador, se procede con el control del aire acondicionado de acuerdo al estado de operación y a la lógica expuesta en la sección 3.3. Finalmente, el módulo remoto se vuelve a dormir por otro periodo.

3.5 Pruebas de confiabilidad

El sistema de administración de energía desarrollado en esta investigación está programado de tal manera que sea sencillo identificar cuando uno de los componentes del sistema está presentando fallas, dejando un registro del evento con el fin de calcular estadísticas de confiabilidad.

En la figura Figura 35 se muestra el flujo de datos relacionados a la prueba de confiabilidad, donde al final, el Raspberry Pi recibe toda la información del estado del sistema, donde se almacenan en la base de datos.

Toda la información acerca del estado del sistema almacenada, no podrá ser visualizada por el usuario en la interfaz gráfica, por lo que tiene que ser consultada directamente con alguna herramienta de base de datos, tal como *“phpMyAdmin”*.

La información es almacenada cada cinco minutos con un código especial, el cual se muestra en la Tabla 9, donde si en un cincominutal se registraron solamente ceros, significa que el sistema no presentó ninguna falla.

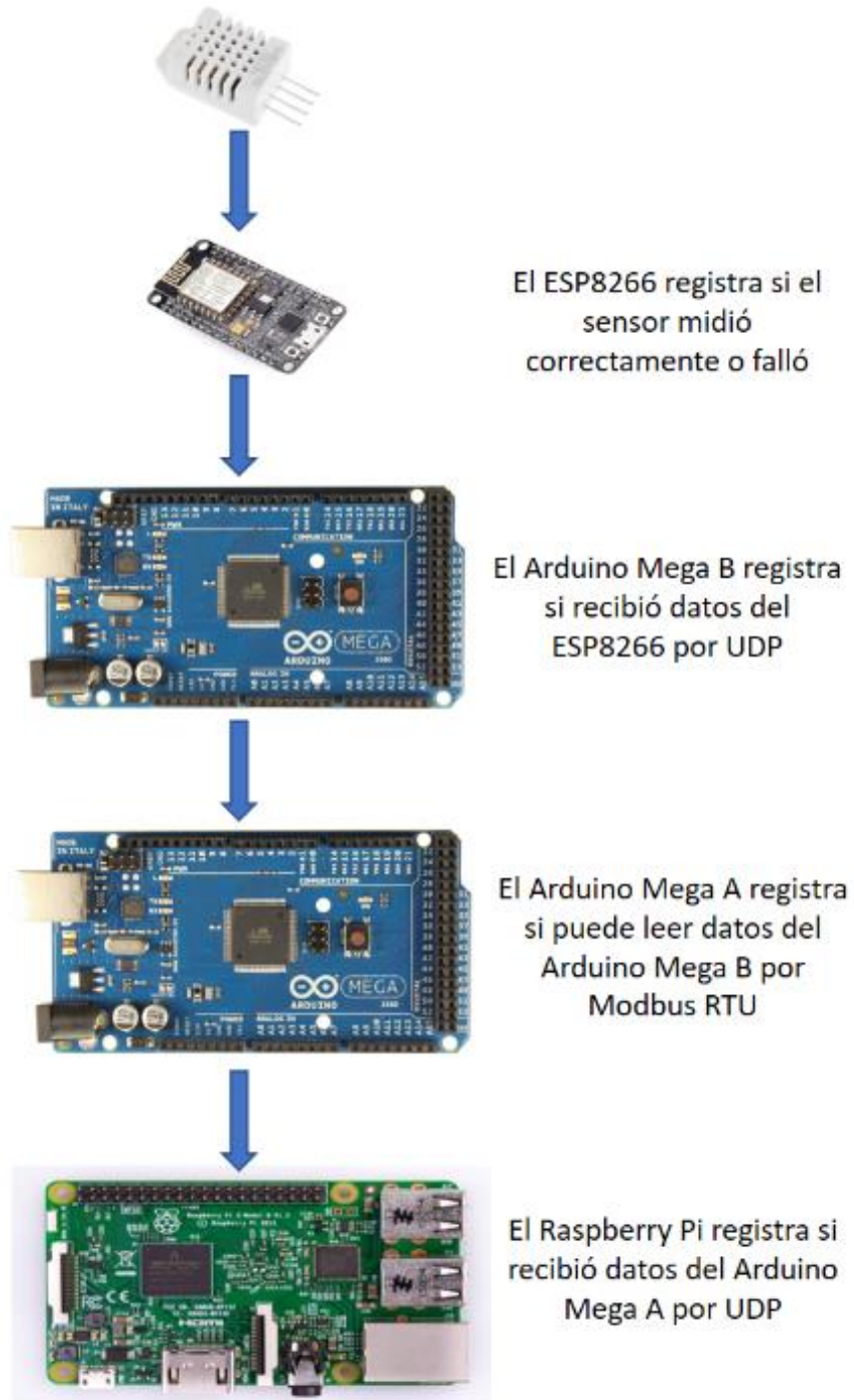


Figura 35 Flujo de datos para pruebas de confiabilidad

Dispositivo	Código de estado
ESP8266	0: Funciona correctamente 1: El ESP8266 no se comunicó con el Arduino Mega B 2: Falló el sensor
Arduino Mega B	0: El Arduino Mega A pudo leer los registros del Arduino Mega B correctamente 1: Error en la comunicación Modbus RTU
Arduino Mega A	0: Funciona correctamente 1: El Arduino Mega A no se comunicó con el Raspberry Pi

Tabla 9 Código de estado por dispositivo

4 Resultados

4.1 Guía de instalación

Como ya se mencionó previamente, el sistema de administración de energía cuenta con tres componentes principales, el dispositivo central, el gateway y el módulo remoto, por lo que, al querer instalar dicho sistema, se deben configurar cada uno de los componentes de acuerdo con las características de la zona donde se instalará. En todo el sistema solo habrá un dispositivo central, sin embargo, si hay más de una zona o “sucursal” debe haber un gateway por cada una de ellas; mientras que puede haber entre uno y dos módulos remotos por equipo de aire acondicionado que se quiera controlar. Otro aspecto que cabe destacar es que el límite por sucursal es de 9 aires acondicionados. A continuación, se describen los pasos para instalar cada uno de estos dispositivos.

4.2 Configuración del dispositivo central

Para realizar la instalación del sistema de administración de energía, lo primero que se debe hacer es conectar el dispositivo central a una fuente de poder y a una red. Una vez conectado, ingresar al webserver desde una computadora, Tablet o celular que tenga acceso a la misma red y tenga un buscador web, en él, se tiene que ingresar la dirección IP del dispositivo central. El buscador web lo direccionará hacia el webserver y se verá una pantalla como la que se puede observar en la Figura 36.

Aquí se tiene que dar clic en “*Agregar Sucursal*”, que los llevará a la página que se muestra en la Figura 37. Primero se deben llenar tres campos: número de identificación de sucursal, el cual no se puede repetir para otras sucursales; nombre de la sucursal; y número de aires de la sucursal.

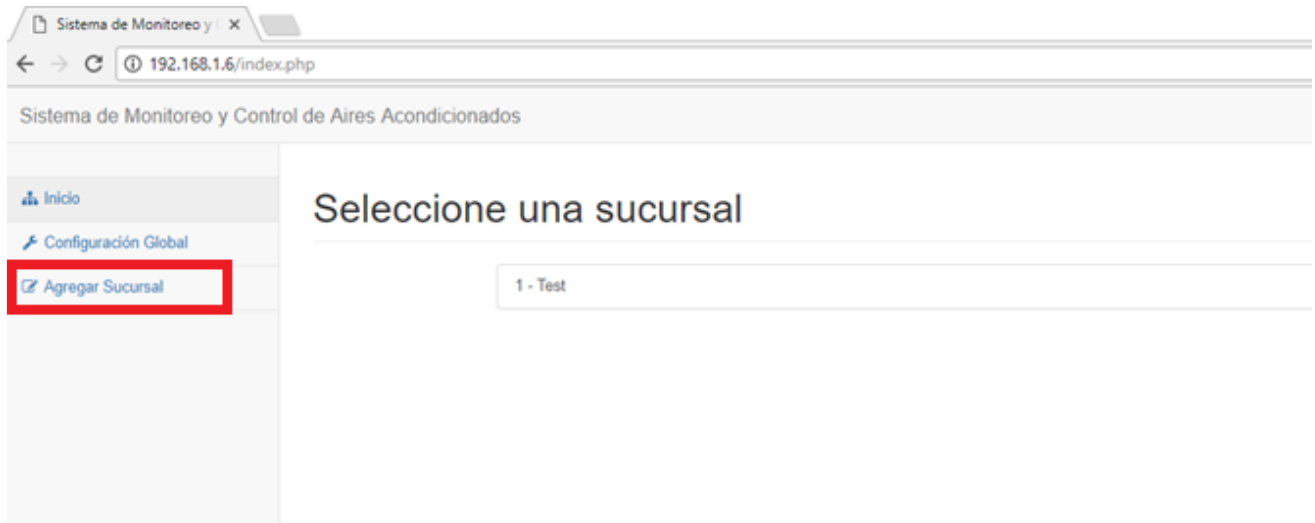


Figura 36 Página de inicio del sistema de administración de energía

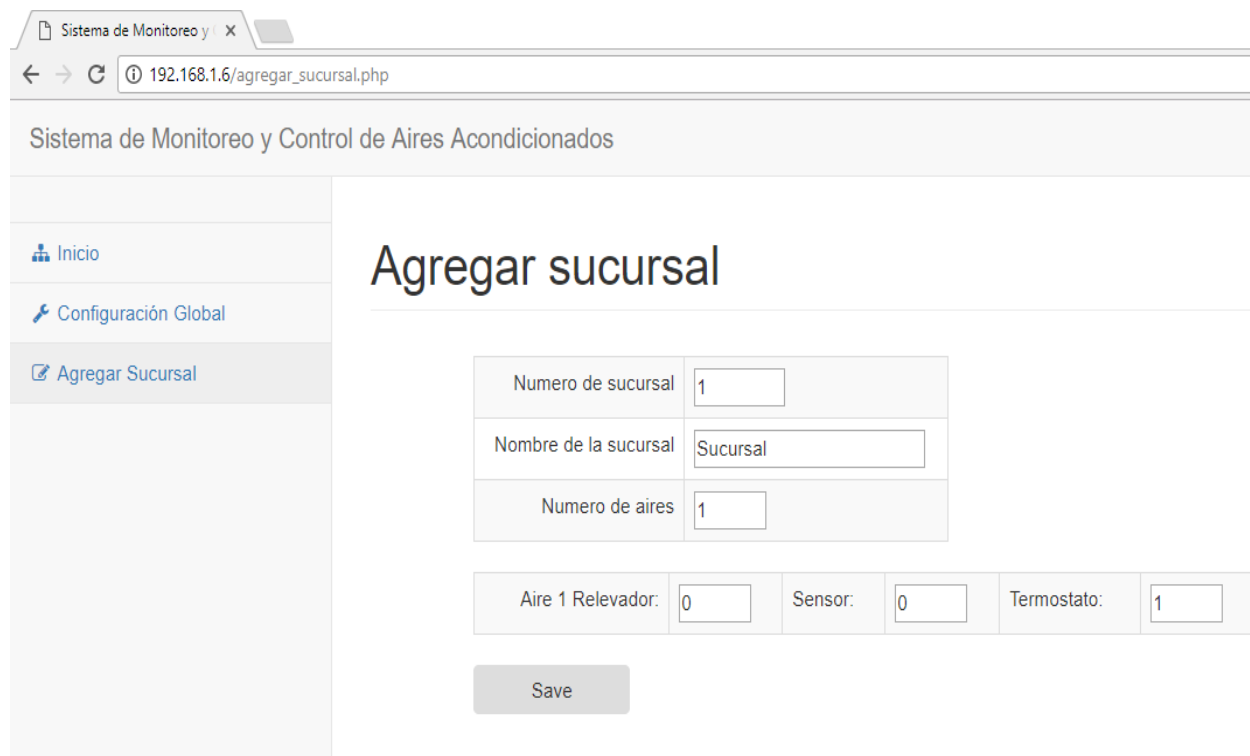


Figura 37 Página para agregar sucursal

Una vez definido el número de aires que tiene la sucursal, se debe de especificar qué tipos de módulos remotos habrá por equipo de aire acondicionado, las opciones son: termostato, termostato – sensor, relevador – sensor (Figura 38). Como es de esperarse, se puede tener un solo módulo remoto de tipo termostato controlando un aire acondicionado, sin embargo, se puede agregar otro módulo remoto tipo sensor, por lo que las temperaturas medidas por cada uno se promediarán. Caso contrario, no se puede tener un módulo remoto de tipo relevador sin uno de tipo sensor, ya que el relevador está diseñado para controlar fuera de la zona acondicionada, y es necesario conocer su temperatura para poder realizar un control apropiado.

Numero de sucursal	1				
Nombre de la sucursal	Sucursal				
Numero de aires	3				
Aire 1 Relevador:	0	Sensor:	0	Termostato:	1
Aire 2 Relevador:	0	Sensor:	1	Termostato:	1
Aire 3 Relevador:	1	Sensor:	1	Termostato:	0

Save

Figura 38 Opciones de módulos remotos por equipo de aire acondicionado

El sistema no permite dar de alta un equipo de aire acondicionado que tenga un módulo remoto tipo termostato y otro tipo relevador, ya que es innecesario y puede provocar conflicto en el sistema. La alerta que surge en el sistema al querer hacer lo anterior se muestra Figura 39.

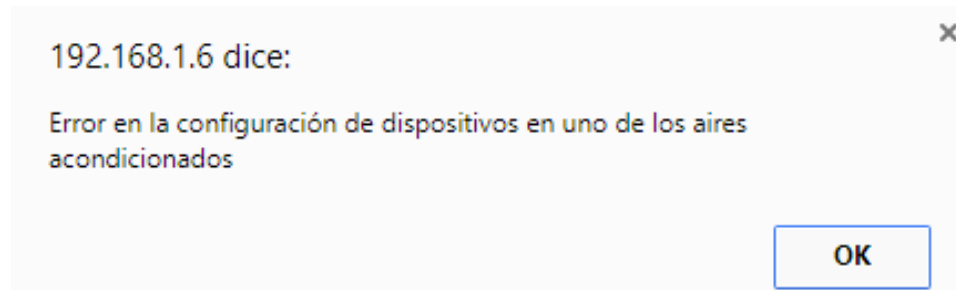
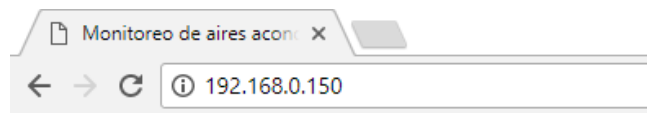


Figura 39 Error al querer dar de alta un termostato y un relevador en un mismo aire acondicionado

4.3 Configuración del gateway

Los gateway tienen la capacidad de conectarse a una o dos redes diferentes; para configurarlo es necesario conectar la computadora, Tablet o celular a la red a la cual se conectarán los módulos remotos, una vez ahí, se debe ingresar la dirección IP del gateway y aparecerá la página que se muestra en la Figura 40.



Configuración actual:

IP del dispositivo central:

192.168.1.6

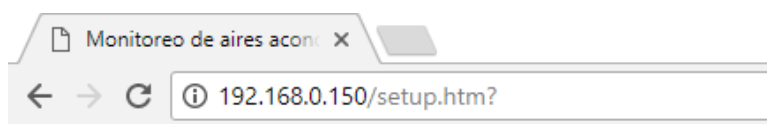
Numero de equipo de sucursal:

1

Configurar

Figura 40 Página de inicio del gateway

En la página de *inicio* del gateway se despliega la dirección IP del dispositivo central que se tiene registrada, así como el número de identificación de la sucursal, si se desean cambiar estos datos, se debe dar clic en *configurar*, lo que redireccionará a la página de configuración que se observa en la Figura 41.



Ingrese los siguientes datos:

IP del dispositivo central:

 . . .

Numero de equipo de sucursal:

Figura 41 Página de configuración del gateway

Una vez que se ingresó la información, se debe dar clic en aceptar, en caso de que no se quiera configurar, tiene que dar clic en regresar o cerrar la página.

4.4 Configuración del módulo remoto

Los módulos remotos se conectan a la red vía WiFi, pero, además, tienen la capacidad de crear su propia red WiFi, por medio de la cual los usuarios se pueden conectar para configurarlo. Para que el módulo remoto entre en modo configuración, es necesario dar clic en el botón “*configuración*” y “*reset*” al mismo tiempo; y checar las redes disponibles

desde la computadora, Tablet o celular, hasta encontrar una red que comience con las iniciales “ESP” (véase Figura 42), la cual corresponde a la red del módulo remoto; la contraseña dicha red es “12345678”.

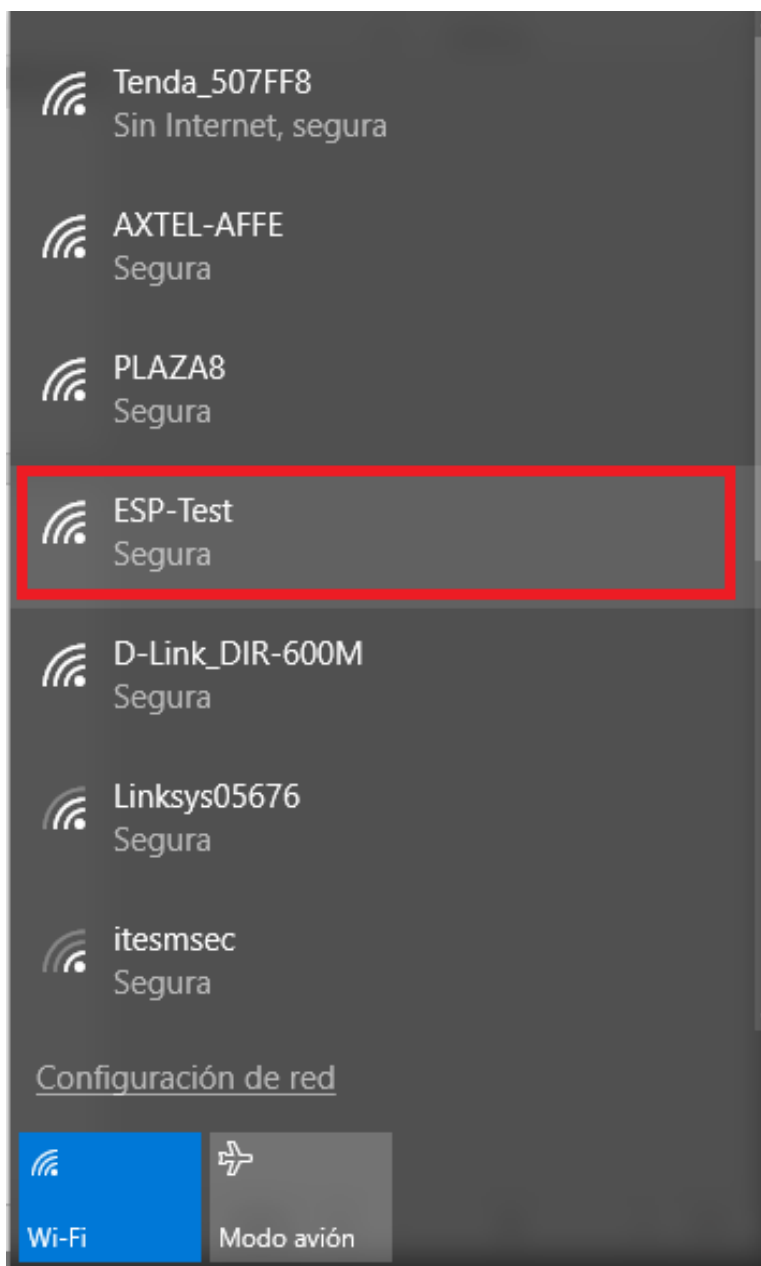


Figura 42 Red generada por el módulo remoto

Una vez conectados a la red, debe ingresar a la dirección IP “192.168.4.1” desde un buscador web, con esto se ingresará a la página de inicio del módulo remoto, la cual se muestra en la Figura 43.

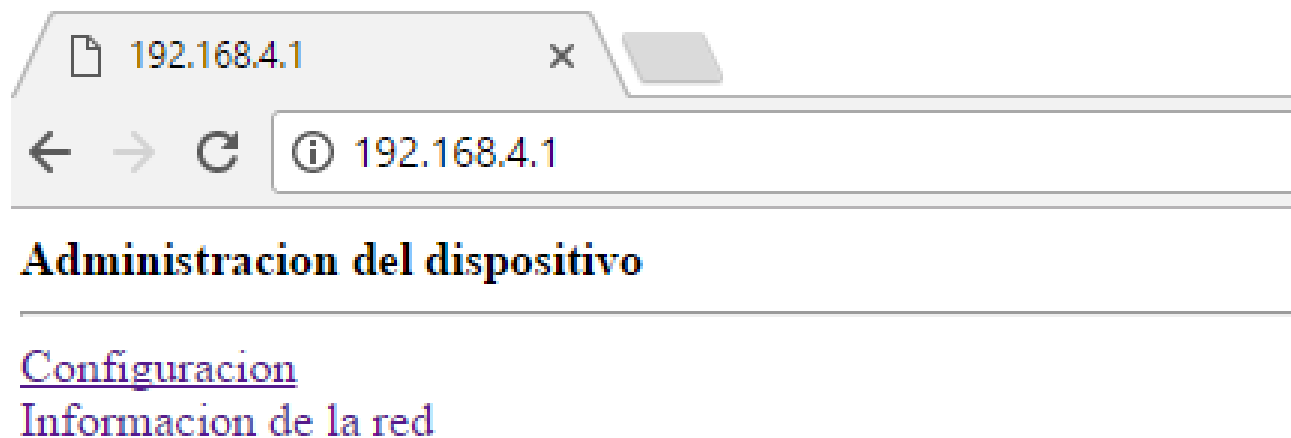


Figura 43 Página de inicio del módulo remoto

En la página de inicio hay dos opciones, “*Configuración*” e “*Información de la red*”, y al dar clic en la primera liga se accede a la página de configuración del módulo remoto (véase Figura 44). Aquí se debe ingresar el nombre que se le quiera dar al módulo remoto, el cual se mostrará en la SSID del mismo; el número de aire acondicionado que está gestionando dentro de la sucursal, el tipo de módulo remoto (sensor de temperatura, termostato o relevador); el tipo de alimentación, ya sea baterías o conectado a la red eléctrica (fuente externa); la dirección IP del gateway al que enviará la información; el SSID y la contraseña de la red a la cual se conectará, y si lo hará con DHCP o con direcciones estáticas, además muestra las redes que están a su alcance así como su intensidad de señal. Una vez que hayan guardado la información, el módulo remoto se reiniciará.

192.168.4.1/config.html x

← → ↻ 192.168.4.1/config.html

< **Configuración**

Nombre del dispositivo

Numero de aire

Tipo de dispositivo:

Termostato
 Sensor de temperatura
 Relevador ID:

Fuente de alimentacion:

Externa
 Baterias

IP del dispositivo central

Conectarse al router con las siguientes características:

SSID:

Contraseña:

DHCP:

IP:

Netmask:

Gateway:

Estado de la conexión:
Conectado

Redes:
Se encontraron 10 redes

Name	Quality	Enc
AXTEL_01A750	66%	*
INFINITUMqvdI	60%	*
AXTEL_XTREMO-1193	74%	*
AXTEL_XTREMO-097C	60%	*
TP-LINK_6408A9	92%	*
DIRECT-F9-HP OfficeJet Pro 7740	86%	*
NRGY	92%	*
La Pitzota Tec	34%	*
INFINITUMEC4649	26%	*
image_box	18%	*

Figura 44 Página de configuración del módulo remoto

Para corroborar que el módulo remoto está conectado a la red es necesario dar clic en la segunda liga del menú de inicio “*Información de la red*” (véase Figura 45). En esta página se puede ver el SSID, netmask, gateway de la red a la cual está conectada el módulo remoto, así como las direcciones IP y Mac de éste.

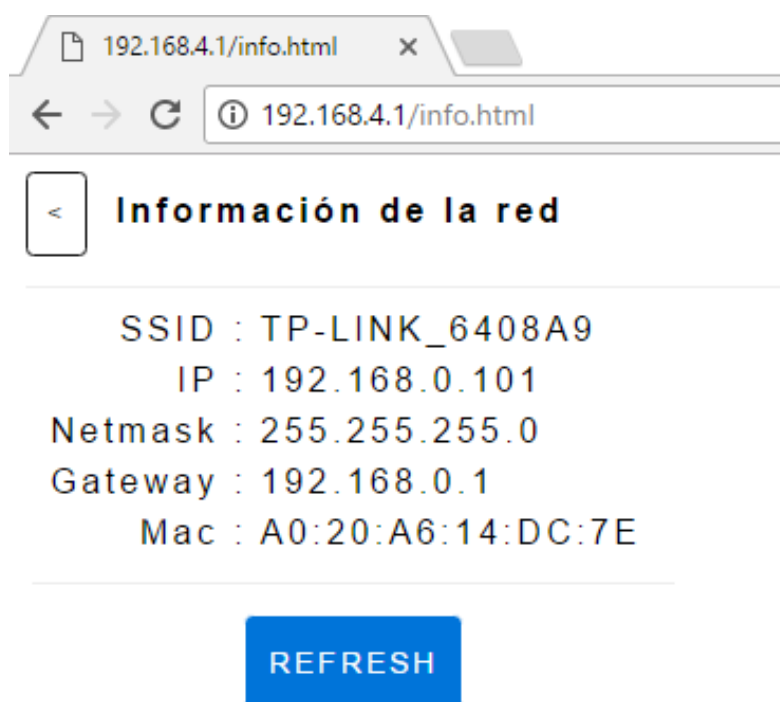


Figura 45 Página de información de la red del módulo remoto

4.5 Visualización de datos

Para visualizar los datos del sistema de administración de energía es necesario ingresar al webserver, en cuya página de *inicio* despliega todas las sucursales que están dadas de alta, tal como se muestra en la Figura 46. En el caso de prueba, solo se manejó una sucursal llamada “*test*”. Para ver los datos en tiempo real de la sucursal, se debe dar clic en ella, redireccionando a la página “*sucursal.php*” (Figura 48).

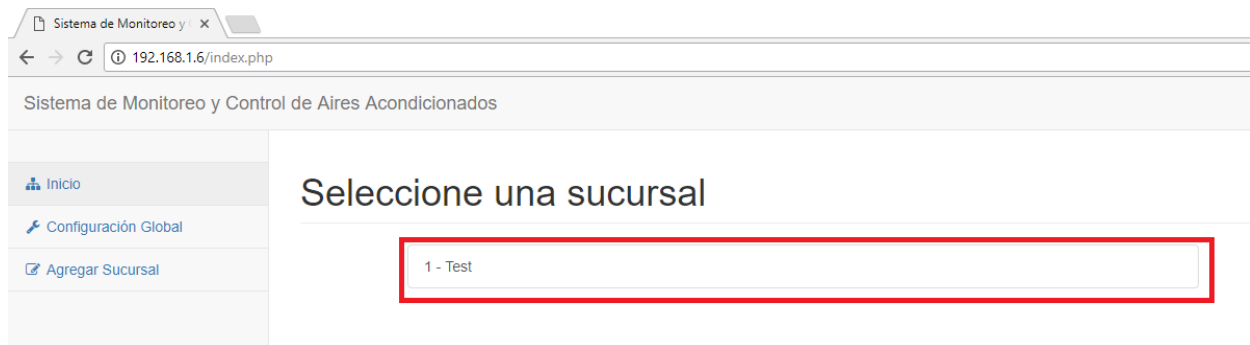


Figura 46 Página de inicio

Para la prueba se utilizaron cuatro módulos remotos para tres equipos de aire acondicionado, es por eso que, en la tabla de la Figura 48 aparecen tres renglones, siendo cada uno de ellos un equipo diferente, así como una gráfica de la temperatura y humedad del último día para cada caso. Debido a que los módulos remotos estuvieron en el mismo espacio, los datos que se muestran son bastante parecidos. Arriba de la tabla se muestra la última actualización, y a pesar de que se considera que los datos son en tiempo real, estos se actualizan cada cinco minutos.

Además de poder visualizar los datos en tiempo real, es posible ver los históricos registrados en la vida del sistema, dando clic en “*Históricos de la Sucursal*”, tal como se ve en la Figura 47.

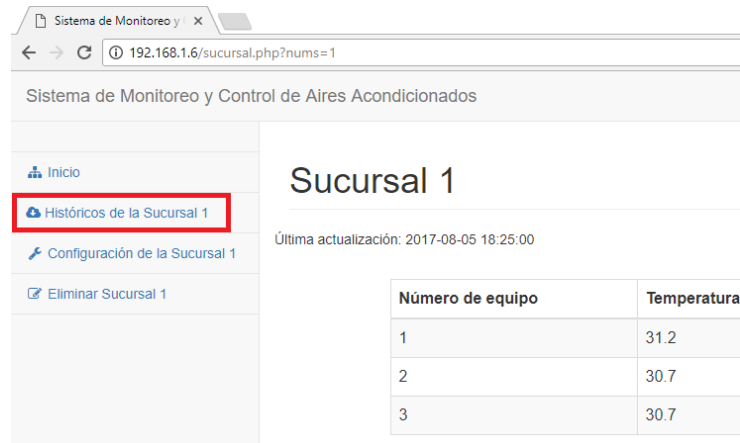


Figura 47 Liga hacia los históricos de la sucursal

- [Inicio](#)
- [Históricos de la Sucursal 1](#)
- [Configuración de la Sucursal 1](#)
- [Eliminar Sucursal 1](#)

Sucursal 1

Última actualización: 2017-08-05 17:35:00

Número de equipo	Temperatura	Humedad
1	31.1	63.9
2	30.6	65.6
3	30.6	65.6

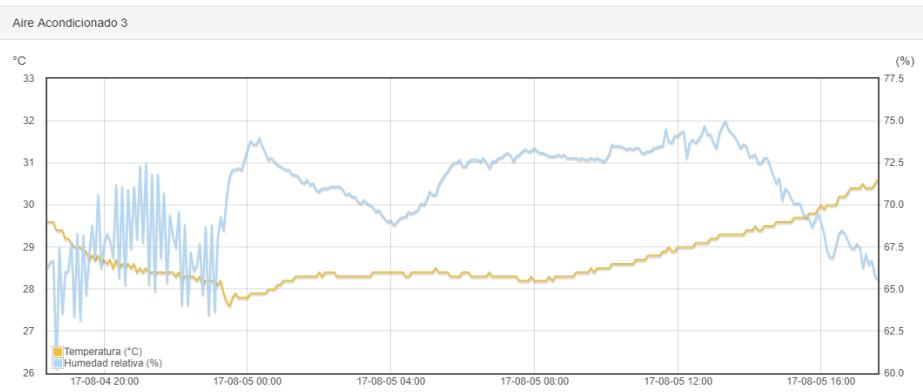
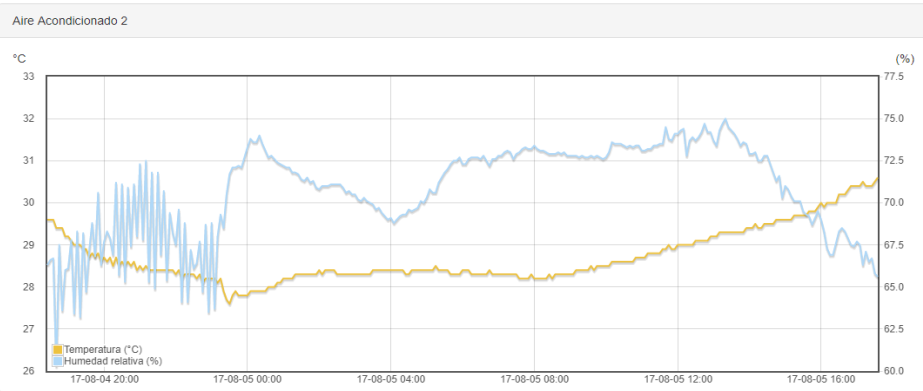
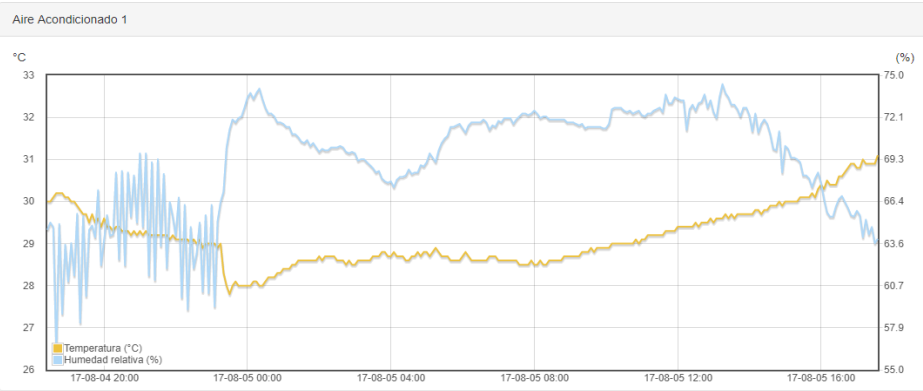


Figura 48 Datos en tiempo real de la sucursal 1

- Inicio
- Sucursal 1
- Configuración de la Sucursal 1
- Eliminar Sucursal 1

Datos históricos de la sucursal 1

Día: Mes: Año:

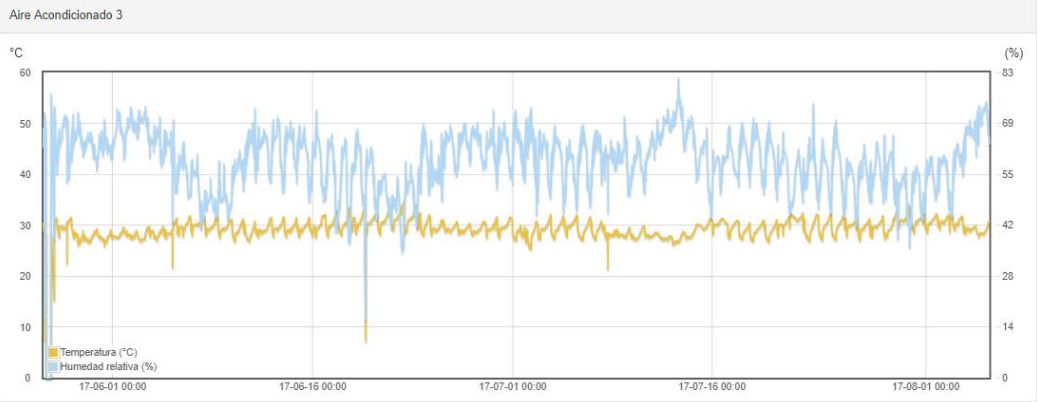
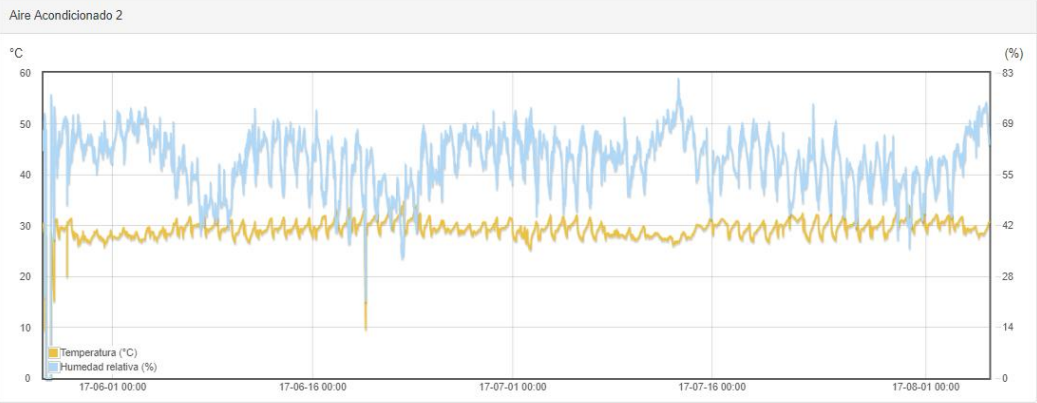
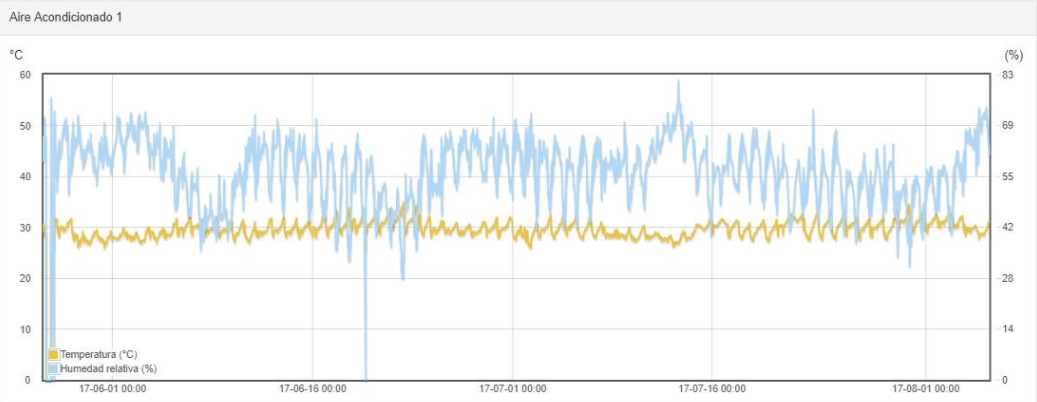


Figura 49 Página de datos históricos de la sucursal 1

Al inicio, se mostrarán todos los datos registrados (Figura 49), sin embargo, es posible visualizar todo un año, mes o día en específico. Por ejemplo, en la Figura 50 se seleccionó Julio del 2017, y al dar clic en “Mostrar” las gráficas muestran todos los datos de dicho mes. Así mismo, al seleccionar un día, como por ejemplo el 10, aparecen los datos del día 10 de Julio del 2017, como se ve en la Figura 51.



Figura 50 Datos históricos del mes de Julio de la sucursal 1

Además de visualizar los datos históricos de temperatura y humedad de los aires acondicionados de las sucursales en forma de gráficos, es posible descargar dichos datos en formato “.CSV” (Separado por comas), el cual puede manipularse con cualquier procesador de texto u hojas de cálculo como Microsoft Excel.

Al seleccionar el día que se quiere descargar, se debe dar clic en “Descargar”, donde aparecerá un cuadro para ingresar el nombre con el cual se desea guardar el archivo y la dirección en la computadora, Tablet o celular, tal como se muestra en la Figura 52. En la Figura 53 se pueden ver los datos descargados.



Figura 51 Datos históricos del día 10 de Julio del 2017 de la sucursal 1

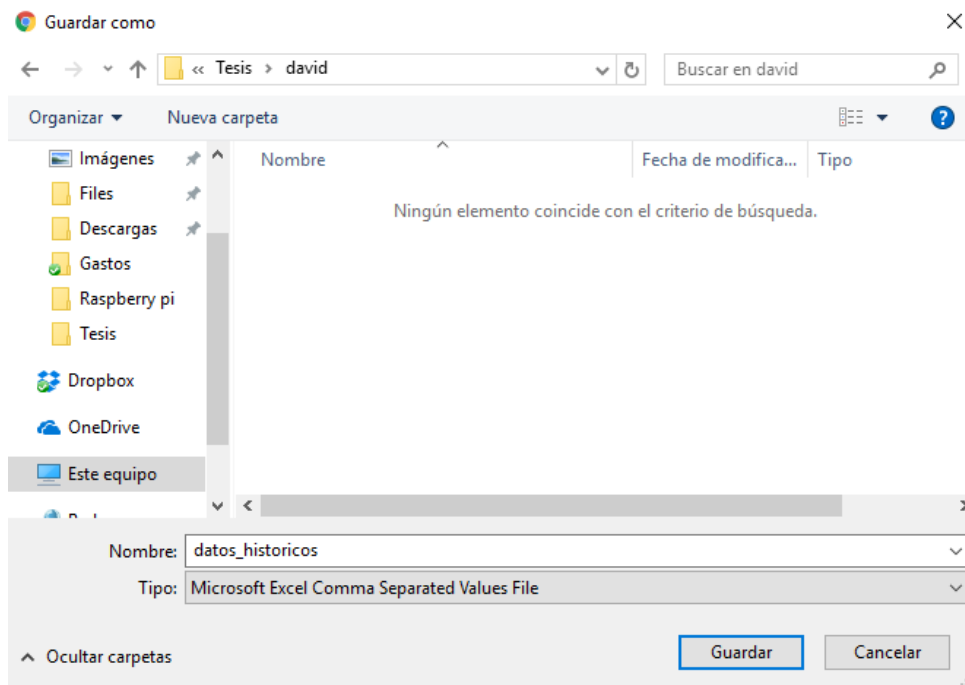


Figura 52 Descarga de datos históricos

The screenshot shows an Excel spreadsheet titled 'datos_historicos'. The data is organized in columns: A (Timestamp), B (Temperatura_1), C (Humedad_1), D (Temperatura_2), E (Humedad_2), F (Temperatura_3), and G (Humedad_3). The rows represent time intervals from 7/10/2017 0:00 to 7/10/2017 1:40. The values for temperature and humidity fluctuate between approximately 27.5 and 28.4 for temperature and 54.4 and 56.1 for humidity.

Timestamp	Temperatura_1	Humedad_1	Temperatura_2	Humedad_2	Temperatura_3	Humedad_3
7/10/2017 0:00	28.4	54	28.2	56.1	28.2	56.1
7/10/2017 0:05	28.4	52.9	28.2	55.5	28.2	55.5
7/10/2017 0:10	28.4	52.7	28.2	55.2	28.2	55.2
7/10/2017 0:15	28.4	53.3	28.2	55.2	28.2	55.2
7/10/2017 0:20	28.4	53.1	28.2	55.3	28.2	55.3
7/10/2017 0:25	28.3	52.6	28.1	54.8	28.1	54.8
7/10/2017 0:30	28	54	27.9	55.7	27.9	55.7
7/10/2017 0:35	27.9	54	27.8	55.9	27.8	55.9
7/10/2017 0:40	28	54.7	27.8	56.2	27.8	56.2
7/10/2017 0:45	27.8	54.3	27.7	56	27.7	56
7/10/2017 0:50	27.8	54.5	27.7	56.3	27.7	56.3
7/10/2017 0:55	28	54.8	27.8	56.1	27.8	56.1
7/10/2017 1:00	27.8	54.1	27.7	56	27.7	56
7/10/2017 1:05	27.8	54.7	27.7	56.7	27.7	56.7
7/10/2017 1:10	27.7	54.5	27.6	56.6	27.6	56.6
7/10/2017 1:15	27.7	55.1	27.6	56.7	27.6	56.7
7/10/2017 1:20	27.7	55.6	27.6	57.2	27.6	57.2
7/10/2017 1:25	27.7	55.2	27.6	57	27.6	57
7/10/2017 1:30	27.7	55.7	27.6	57.5	27.6	57.5
7/10/2017 1:35	27.7	56.1	27.6	57.7	27.6	57.7
7/10/2017 1:40	27.6	55.4	27.5	57.1	27.5	57.1

Figura 53 Datos históricos descargados

4.6 Control del sistema

El control del sistema de aires acondicionados se realiza por sucursal, lo que significa que todos los equipos dados de alta para una sucursal funcionarán con el mismo setpoint y mismos horarios. También es posible hacer una configuración global (que afecta a todas las sucursales). Para entrar en el modo de configuración global, es necesario estar en la página de inicio y dar clic en “*Configuración Global*”, como se muestra en la Figura 54.

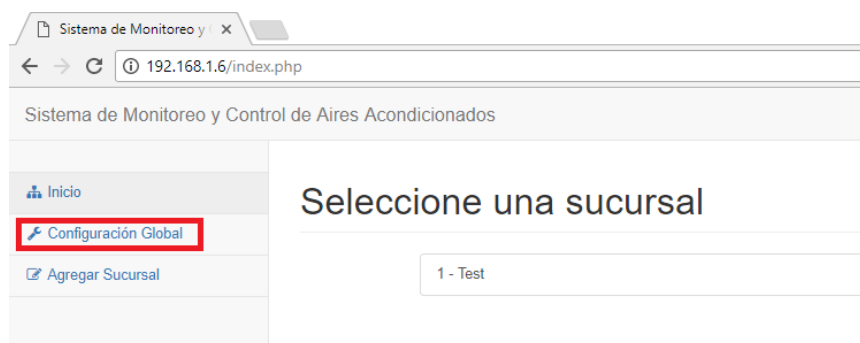


Figura 54 Liga a la Configuración Global

Para configurar solamente una sucursal, es necesario seleccionar primero la sucursal a configurar, una vez estando en la página de dicha sucursal, seleccionar “*Configuración de la Sucursal X*”, tal como se ve en la Figura 55.



Figura 55 Liga a Configuración de Sucursal

Al ingresar a la página de configuración, se verá lo que se muestra en la Figura 57, donde se puede asignar un setpoint de refrigeración, y otro de calefacción, además de los estados de los aires acondicionados por hora y para los siete días de la semana. Si solo se desea cambiar el estado de una hora específica, en la pestaña de “*Mostrar*” se puede ingresar la hora deseada y automáticamente la tabla desplegará dicha hora, como se puede observar en la Figura 56.

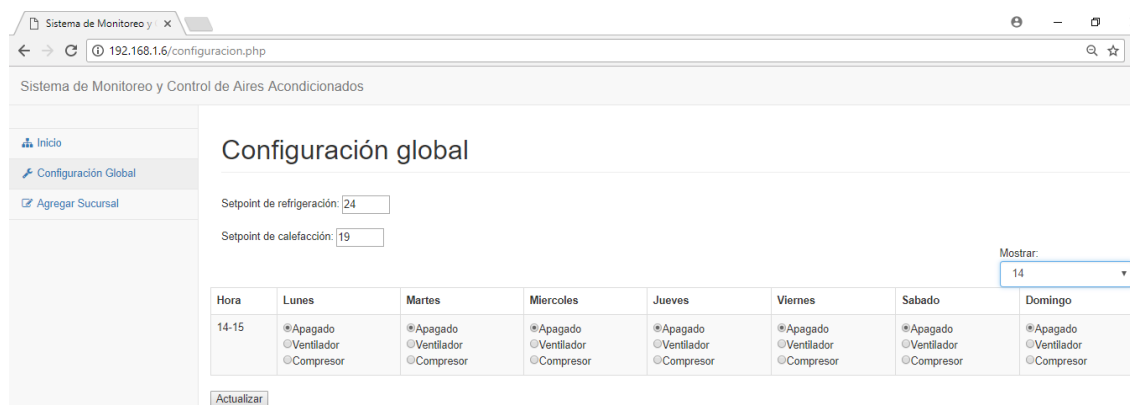


Figura 56 Selección de hora en la página de configuración

Una vez asignados los cambios deseados, es necesario dar clic en “Actualizar”, los cuales tendrán efecto aproximadamente un minuto después de haber terminado la configuración.

Se desarrolló un prototipo de una sucursal con un solo equipo de aire acondicionado (simulado con focos), para probar el control del sistema. Se registró que todos los cambios y configuraciones realizadas en el sistema se ejecutaron de manera correcta. En la **¡Error! No se encuentra el origen de la referencia.** se muestra dicho prototipo.



Figura 58 Simulación de sistema de una sucursal con un equipo de aire acondicionado

4.7 Diagnóstico de comunicación

La prueba para determinar la cantidad de paquetes de comunicación enviados y recibidos satisfactoriamente se realizó en el prototipo con un dispositivo central (Figura 59), una sucursal con un gateway (Figura 60), tres aires acondicionados y cuatro módulos remotos (Figura 61). Se analizó la comunicación entre el Arduino Mega A y el Raspberry Pi por UDP (gateway y dispositivo central), la comunicación entre el Arduino Mega A y el Arduino Mega B por Modbus RTU, y las comunicaciones entre el Arduino Mega B y los ESP8266 por UDP (gateway y módulos remotos).



Figura 59 Dispositivo central

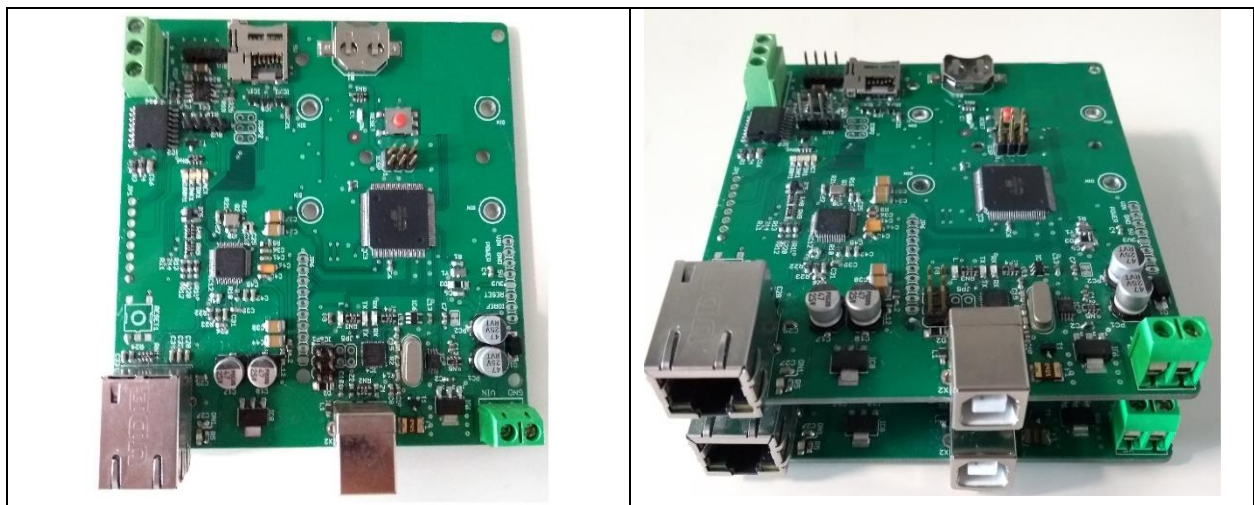


Figura 60 Dos Arduinos Mega que componen al Gateway

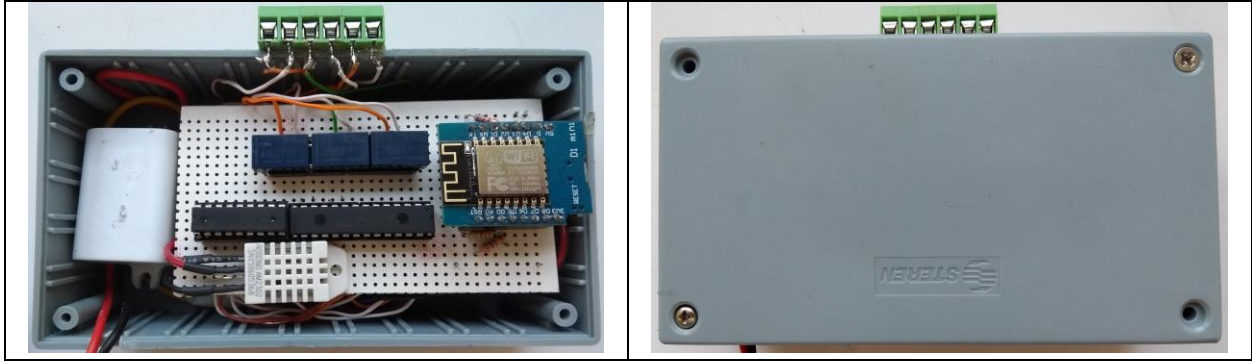


Figura 61 Módulo remoto

Cabe destacar que el prototipo estaba instalado en un solo sitio, por lo que las distancias entre los módulos remotos y el modem eran mínimas. Es posible que, en una aplicación real, al aumentar la distancia entre estos, los paquetes enviados y recibidos satisfactoriamente se reduzcan.

En la comunicación entre el Arduino Mega A y el Raspberry Pi por UDP, fallaron 72 paquetes de 33600, dando un total de 99.79% de paquetes satisfactorios. En la comunicación entre el Arduino Mega A y el Arduino Mega B por Modbus RTU, fallaron 518, dando un total de 98.46% de paquetes satisfactorios. En cuanto a la comunicación entre el Arduino Mega B y los cuatro ESP8266, el total de paquetes satisfactorios fue de 99.62%.

5 Conclusiones

5.7 Resultados y conclusiones

Los sistemas de aire acondicionado representan un alto consumo energético, generando la demanda máxima en zonas residenciales y comerciales. Estos sistemas permiten preservar el confort térmico de un lugar, brindando cierta cantidad y calidad de aire. Lo cual los convierte en sistemas necesarios en diferentes ambientes.

En muchas ocasiones, los sistemas de aire acondicionado operan más de lo necesario debido a una falta de ajustes de temperatura que puedan controlarlo. Con respecto a esto, se encontró que, mediante el control adecuado de los equipos de aire acondicionado, se pueden lograr ahorros de hasta el 15% del costo total anual por energía.

Ya existen sistemas de administración de energía que ayuden a controlar el consumo, sin embargo, estos sistemas actuales, son accesibles para empresas grandes, que gracias a su poder adquisitivo pueden optar por esta opción. Por otra parte, las pequeñas y medianas empresas presentan serias dificultades para adquirirlos.

Utilizando hardware y software de código abierto, se pueden desarrollar sistemas de administración de aires acondicionados a bajo costo, sin tener que disminuir la confiabilidad y robustez del equipo.

5.8 Contribuciones

Se desarrolló un sistema de gestión y control de aires acondicionados: replicable, de fácil instalación, de hardware y software de código abierto, aplicable en empresas con sucursales alejadas.

La arquitectura del sistema cuenta con tres componentes principales: un dispositivo central; módulos remotos y gateways. El primero permite procesar toda la información y

conecta a los usuarios con el resto del sistema mediante una interfaz gráfica; los segundos controlan los aires acondicionados; y los gateways sirven de puente de comunicación entre los dos anteriores.

Se propuso como dispositivo central un Raspberry Pi 3 Model B, en el cual, se desarrolló la interfaz gráfica creada en HTML5, CSS3, JavaScript y PHP; en el mismo Raspberry Pi, se encuentra la base de datos que guarda toda la información y es administrada por MySQL. La comunicación entre el Raspberry Pi y los gateways está basada en el protocolo UDP y está escrita en Python.

Dos Arduinos Megas con Ethernet Shields constituyen el gateway. La comunicación entre los Arduinos es por Modbus RTU. Este dispositivo se puede omitir en caso de que el sistema de aire acondicionado tenga menos de diez equipos, y todos se encuentren dentro de la misma red WiFi.

Los módulos remotos son ESP8266 NodemCU, los cuales son microcontroladores que cuentan con un integrado que permite la conexión WiFi. Los módulos remotos tienen DHT22 que son sensores de temperatura y humedad, además de relevadores de pulso que permiten manipular los equipos de aire acondicionado a un bajo consumo de energía, interceptando las señales de control de los termostatos. Los módulos remotos pueden funcionar como termostatos, o solamente como sensores de temperatura o controlador.

Se realizaron dos prototipos para las pruebas experimentales. El primer prototipo es un sistema de un aire acondicionado, que solo tiene un dispositivo central y un módulo remoto. El segundo prototipo es un sistema con tres aires acondicionados, que cuenta con el dispositivo central, un gateway y cuatro módulos remotos.

El primer prototipo fue utilizado para probar el sistema de control, ya que se simulan los estados de los equipos de aire acondicionado (compresor, ventilador o apagado) con focos incandescentes. Resultando que los módulos remotos son capaces de controlar a los equipos de aire acondicionado con el método de control propuesto y, además, los

usuarios pueden cambiar el estado de éstos, así como sus setpoints de calefacción y enfriamiento fácilmente desde la interfaz web.

El segundo prototipo se utilizó para probar la confiabilidad de la comunicación entre todos los dispositivos. Los resultados obtenidos fueron que, la comunicación UDP entre el dispositivo central y el gateway se enviaron y recibieron satisfactoriamente el 99.79% de los paquetes. La comunicación por Modbus RTU interna en el gateway los paquetes satisfactorios fueron del 98.46%. Finalmente, los paquetes enviados y recibidos satisfactoriamente entre el gateway y los módulos remotos fueron del 99.62%.

5.9 Trabajo futuro

El sistema de administración de energía desarrollado se puede mejorar agregándole algoritmos de “Machine Learning” o “Inteligencia artificial”, con el objetivo de que pueda realizar técnicas como el encendido y apagado inteligente de manera automática, y que, además, con el tiempo, aprenda a hacerlo de la manera más óptima, tomando en cuenta las temperaturas interiores y exteriores. Con estos algoritmos, también podría lograr un setpoint ideal, donde encuentre el balance entre el consumo energético y una retroalimentación por parte de los habitantes del edificio. Para conseguir lo anterior, no es necesario agregar o cambiar el hardware del sistema, ya que, el Raspberry Pi 3 cuenta con la capacidad de procesamiento suficiente para lograrlo.

Otra de las mejoras en la que se puede trabajar, es en cambiar el dispositivo central por un servidor dedicado, ya que esto vencería la limitación de tener que operar bajo un intranet, porque mientras los gateway tengan acceso a internet, ya sea por ethernet, WiFi o GSM, podrían comunicarse con el servidor central.

6 Referencias

- [1] (2014). *Balance Nacional de Energía 2014*.
- [2] J. Murphy and N. Maldeis, "Using Time-of-Day Scheduling To Save Energy," (in English), *Ashrae Journal*, Article vol. 51, no. 5, pp. 42-+, May 2009.
- [3] S. Pomar Fernández, J. A. Rangel Magdaleno, and R. E. Franco Zesati, "La influencia de las barreras a la innovación que limitan la competitividad y el crecimiento de las pymes manufactureras. (Spanish)," *Administracion y Organizaciones*, Article vol. 17, no. 33, pp. 33-57, 12// 2014.
- [4] C. F. d. Electricidad. (2017, 06/03/2017). *Tarifas HM - CFE*. Available: http://app.cfe.gob.mx/Aplicaciones/CCFE/Tarifas/Tarifas/Tarifas_industria.asp?Tarifa=HM
- [5] C. F. d. Electricidad. (2017, 06/06/2017). *Tarifas BT - CFE*. Available: http://app.cfe.gob.mx/Aplicaciones/CCFE/Tarifas/Tarifas/tarifas_casa.asp
- [6] C. F. d. Electricidad. (2017, 06/03/2017). *Tarifas OM - CFE*. Available: http://app.cfe.gob.mx/Aplicaciones/CCFE/Tarifas/Tarifas/Tarifas_industria.asp?Tarifa=OM
- [7] S. K. Wang, *Handbook of air conditioning and refrigeration*, 2nd ed. New York: McGraw-Hill, 2000.
- [8] K. Ogata, *Modern control engineering*, 5th ed. (Prentice-Hall electrical engineering series Instrumentation and controls series). Boston: Prentice-Hall, 2010, pp. x, 894 p.
- [9] D. R. Coughanowr, *Process systems analysis and control*, 2nd ed. (McGraw-Hill chemical engineering series). New York: McGraw-Hill, 1991, pp. xvii, 566 p.
- [10] C. A. Smith and A. B. Corripio, *Principles and practice of automatic process control*. New York : J. Wiley, c1997.
- 2nd ed., 1997.
- [11] American Society of Heating Refrigerating and Air-Conditioning Engineers., *2003 ASHRAE handbook : heating, ventilating, and air-conditioning applications*, Inch-Pound ed. Atlanta, Ga.: ASHRAE, 2003.
- [12] American Society of Heating Refrigerating and Air-Conditioning Engineers., "ASHRAE handbook. Fundamentals," Inch-pound ed. Atlanta, Ga.: American Society of Heating, Refrigerating, and Air-Conditioning Engineers, 1985, p. v.
- [13] ASHRAE (Firm), *Standard 55-2013 user's manual : ANSI/ASHRAE standard 55-2013, thermal environmental conditions for human occupancy*. Atlanta, GA: ASHRAE Research, 2016, p. pages cm.
- [14] American Society of Heating Refrigerating and Air-Conditioning Engineers., *ASHRAE pocket guide for air conditioning, heating, ventilation, refrigeration*, SI ed. Atlanta, GA: American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2009, pp. ix, 262 p.
- [15] American Society of Heating Refrigerating and Air-Conditioning Engineers., "ASHRAE handbook. Heating, ventilating, and air-conditioning systems and equipment," Inch-pound ed. Atlanta, GA: American Society of Heating, Refrigerating and Air Conditioning Engineers, 1992, p. v.
- [16] American Society of Heating Refrigerating and Air-Conditioning Engineers., *2010 ASHRAE Handbook : Refrigeration*, Inch-pound ed. Atlanta, GA: ASHRAE, 2010.

- [17] R. Poovendran, "Cyber–Physical Systems: Close Encounters Between Two Parallel Worlds [Point of View]," *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1363-1366, 2010.
- [18] M. Deepa, G. Angappa, J. C. Stephen, P. Thanos, D. Rameshwar, and W. Samuel, "Vision, applications and future challenges of Internet of Things: A bibliometric study of the recent literature," *Industrial Management & Data Systems*, vol. 116, no. 7, pp. 1331-1355, 2016/08/08 2016.
- [19] N. Jan and O. Nasrine, "The Internet of Things and convenience," *Internet Research*, vol. 26, no. 2, pp. 360-376, 2016/04/04 2016.
- [20] A. Per and M. Lars-Gunnar, "Service innovations enabled by the "internet of things"," *IMP Journal*, vol. 9, no. 1, pp. 85-106, 2015/04/13 2015.
- [21] F. Peter, "Through the looking glass: thinking through the internet of things," *Library Hi Tech News*, vol. 32, no. 5, pp. 4-7, 2015/07/06 2015.
- [22] E. Polycarpou, L. Lambrinos, and E. Protopapadakis, "Smart parking solutions for urban areas," in *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2013, pp. 1-6.
- [23] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32-39, 2011.
- [24] M. Moritz, T. Redlich, P. P. Grames, and J. P. Wulfsberg, "Value creation in open-source hardware communities: Case study of Open Source Ecology," in *2016 Portland International Conference on Management of Engineering and Technology (PICMET)*, 2016, pp. 2368-2375.
- [25] Arduino. (2017, 04/06/2017). *Arduino - Introduction*. Available: <https://www.arduino.cc/en/Guide/Introduction>
- [26] @Raspberry_Pi. (2017, 06/04/2017). *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi*. Available: <https://www.raspberrypi.org/>
- [27] @opensourceway. (2017, 04/06/2017). *What is a Raspberry Pi?* Available: <https://opensource.com/resources/what-raspberry-pi>
- [28] K. C. Loudon, *Lenguajes de programación: principios y práctica*, 2a ed. (no. Book, Whole). México, D.F: Thomson, 2004.
- [29] Arduino. (2017, 04/06/2017). *Arduino - Language Reference*. Available: <https://www.arduino.cc/en/Reference/HomePage>
- [30] Arduino. (2017, 04/06/2017). *Arduino Playground - Structure*. Available: <http://playground.arduino.cc/ArduinoNotebookTraduccion/Structure>
- [31] M. Lutz, *Learning Python*, Fifth edition. ed. Beijing: O'Reilly, 2013, pp. I, 1540 pages.
- [32] D. Sklar and A. Trachtenberg, *PHP cookbook*, Third edition. ed. Sebastopol, CA: O'Reilly Media Inc., 2014, pp. xxi, 793 pages.
- [33] P. DuBois, *MySQL cookbook*, Third edition. ed. Beijing ; Sebastopol, CA: O'Reilly, 2014, pp. xxv, 837 pages.
- [34] S. Powers, *JavaScript cookbook : programming the web*, Second edition. ed. Sebastopol, CA: O'Reilly Media, 2015, pp. xvii, 611 pages.
- [35] R. Clark, *Beginning HTML5 and CSS3 : the Web evolved : next generation Web standards* (The expert's voice in web development). Berkeley, Calif.

New York: Apress ;

Distributed to the book trade worldwide by Springer Science+Business Media, 2012, pp. xx, 600 p.

- [36] D. Dragomir, L. Gheorghe, S. Costea, and A. Radovici, "A Survey on Secure Communication Protocols for IoT Systems," in *2016 International Workshop on Secure Internet of Things (SloT)*, 2016, pp. 47-62.
- [37] Y. Kuang, "Communication between PLC and Arduino Based on Modbus Protocol," in *2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control*, 2014, pp. 370-373.
- [38] B. V. Vishakh, M. K. Khwaja, and C. M. Vidhyapathi, "Comprehensive automated device for hotel management using I2C protocol," in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2015, pp. 1-4.
- [39] P. Miller, *TCP/IP : the ultimate protocol guide*. Boca Raton, Fla: BrownWalker Press, 2009.
- [40] *Energy Standard for Buildings Except Low-Rise Residential Buildings*, 2013.
- [41] G. E. Escrivá, A. Marrec, M. A. Ortega, I. V. Salazar, and C. Álvarez-Bel, "Evaluation of control strategies in HVAC split systems," in *2009 IEEE PES/IAS Conference on Sustainable Alternative Energy (SAE)*, 2009, pp. 1-8.
- [42] M. Kusriyanto and B. D. Putra, "Smart home using local area network (LAN) based arduino mega 2560," in *2016 2nd International Conference on Wireless and Telematics (ICWT)*, 2016, pp. 127-131.
- [43] @Raspberry_Pi. (2017, 11/06/2017). *Raspberry Pi 3 Model B - Raspberry Pi*. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- [44] Arduino. (2017, 11/06/2017). *Arduino - ArduinoBoardMega2560*. Available: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [45] Arduino. (2017, 11/06/2017). *ethernet shield w5100 - Bing*. Available: https://www.bing.com/search?q=ethernet+shield+w5100&form=PRMXES&https_msn=1&refig=692d36b7da754e3f802468d545c252d8&pq=ethernet+shield+w&sc=1-17&sp=1&q=AS&sk=
- [46] Arduino. (2017, 06/11/2017). *Arduino Playground - ModbusMaster Library*. Available: <http://playground.arduino.cc/Code/ModbusMaster>
- [47] Arduino. (Time, 06/11/2017). *Arduino Playground - Time*. Available: <http://playground.arduino.cc/Code/time>
- [48] (2017, 06/11/2017). *NodeMCU Documentation*. Available: <https://nodemcu.readthedocs.io/en/master/#nodemcu-documentation>
- [49] L. Aosong Electronics Co. (2017, 06/11/2017). *DHT-22*. Available: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [50] Microchip. (2017, 06/11/2017). *MCP23017*. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf>
- [51] T. Instruments. (2017, 06/11/2017). *ULN2803A*. Available: <http://www.ti.com/lit/ds/symlink/uln2803a.pdf>
- [52] Panasonic. (2017, 06/11/2017). *TQ Relays*. Available: http://www.mouser.com/ds/2/315/mech_eng_tq-1075833.pdf
- [53] L. John. (2017, 25/06/2017). *ESP 8266 Arduino IDE WebConfig*. Available: <https://www.john-lassen.de/en/projects/esp-8266-arduino-ide-webconfig>