

UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN



INFORME DE TRABAJO MONOGRÁFICO

Desarrollo de un Controlador Inteligente para Aires
Acondicionados Utilizando Lógica Difusa y Comunicación
Infrarroja IR

Presentado por:

Br. Josué Enrique Delgado Rocha

TUTOR:

Ing. Álvaro Gaitán

Managua-Nicaragua

Abril, 2019

Dedicatoria

Dedicada a mis padres, Eneida A. Picado-López y Cesar A. Meynard, cuya comprensión y ayuda hicieron posible la realización de este proyecto.

Josué E. Delgado-Rocha

Agradecimientos

Al profesor Álvaro Gaitán por haber sido un guía y un excelente tutor en esta monografía.

Josué E. Delgado-Rocha

Resumen

En este trabajo monográfico se llevó cabo el desarrollo de un controlador inteligente para aires acondicionados. Este dispositivo se encarga de la configuración automática del termostato, el encendido y apagado del aire acondicionado, basándose en la humedad y temperatura interior, utilizando un sistema fuzzy para la toma de decisiones, y enviando los comandos de configuración correspondientes mediante una comunicación infrarroja. La finalidad de este dispositivo es lograr una reducción en el consumo energético de este aparato.

En este documento se describe el proceso de desarrollo del controlador inteligente, como se encuentra constituido y su funcionalidad. Además, también se describe el desarrollo de un medidor IoT utilizado para el monitoreo del consumo energético en un aparato eléctrico. El objetivo de este último dispositivo, es realizar una evaluación entre el consumo energético del aire acondicionado, utilizando el controlador desarrollado y sin este, y de esta manera cuantificar el ahorro obtenido. Para ello, también se describen las pruebas realizadas, el lugar, duración y como fueron llevadas cabo. Y luego presentan y analizan los resultados obtenidos.

Contenido

Introducción	1
Objetivos.....	2
Justificación	3
Capítulo I:	5
Marco Teórico.....	5
1.1 El aire acondicionado.....	5
1.2 Comunicación infrarroja IR.....	7
1.3 Confort térmico	10
1.4 Medición de temperatura y humedad.....	11
1.4.1 Medición de temperatura.	12
1.4.2 Medición de humedad relativa.	13
1.5 Sensores de presencia	14
1.5.1 Sensores PIR.....	14
1.5.2 Sensores tipo radar.....	15
1.5.3 Sensores térmicos.	16
1.6 Control fuzzy.....	17
1.6.1 Universo de discurso y conjuntos fuzzy.	18
1.6.2 Funciones de pertenencia y fuzzificación.....	18
1.6.3 Reglas Fuzzy y operadores lógicos.	19
1.6.4 Defuzzificación.....	22
1.7 Potencia y energía	23
1.8 Método de selección cualitativo	23
1.9 Módulo ESP01	25
1.10 Elementos del sistema de monitoreo energético (SME).	26
1.10.1 Hardware SME.....	26
1.10.2 Software SME.	28

Capítulo II: Análisis y presentación de resultados	31
2.1 Diseño de controlador Inteligente.....	31
2.1.1 Diseño funcional del controlador.....	31
2.1.2 Sistema de medición.....	32
2.1.2.1 Medición de Humedad y Temperatura interior.....	32
2.1.2.2 Detección de presencia.....	36
2.1.3 Interfaces de comunicación.....	41
2.1.3.1 Interfaz infrarroja IR.....	41
2.1.3.2 Interfaz bluetooth.....	43
2.1.4 Indicadores LEDS.....	44
2.1.5 Sistema de control	45
2.1.5.1 Microcontrolador.....	45
2.1.5.2 Modelos de programa.....	48
2.1.5.3 Algoritmo fuzzy.....	51
2.1.6 Fuente alimentación.....	55
2.1.7 Diseño en PCB	56
2.1.8 Diseño de APP	58
2.2 Diseño de medidor IoT.....	60
2.2.1 Diseño funcional medidor.....	60
2.2.2 Sistema de medición.....	60
2.2.2.1 Diseño funcional.....	61
2.2.2.2 Sensor de corriente.....	61
2.2.2.3 Sensor de voltaje.....	62
2.2.2.4 Senales de salida, sensor de corriente y voltaje.....	63
2.2.2.5 Circuito offset.....	64
2.2.2.6 Circuito adaptador de voltaje para sensor de corriente.....	64
2.2.2.7 Circuito adaptador de voltaje para sensor de voltaje.....	65

2.2.2.8	Circuito medidor de voltaje pico.....	66
2.2.2.9	Circuito de desfasamiento.....	67
2.2.3	Sistema de comunicación.....	70
2.2.3.1	Comunicación controlador-ESP.....	71
2.2.3.2	Modelo de programa ESP01.....	72
2.2.3.1	Comunicación ESP-servidor web.....	72
2.2.4	Sistema de señalización.....	74
2.2.5	Sistema de control.....	76
2.2.5.1	Microcontrolador.....	76
2.2.5.2	Modelo de programa.....	76
2.2.6	Sitio web.....	78
2.2.7	Sistema de alimentacion.....	80
2.2.8	Diseno en PCB.....	81
2.2.9	Calibracion del dispositivo.....	83
2.2.9.1	Calibración del sensor de voltaje.....	83
2.2.9.2	Calibración del sensor de corriente.....	84
2.3	Pruebas y resultados.....	85
2.3.1	Montaje de los dispositivos.....	85
2.3.2	Descripción de las pruebas.....	87
2.3.3	Presentación de resultados.....	88
Capítulo III: Conclusiones y recomendaciones.....		90
3.1	Conclusiones.....	90
3.2	Recomendaciones.....	91
Referencias.....		92

Lista de abreviaturas

AA	<i>Aire Acondicionado</i>
LED	<i>Light Emitting Diode</i>
IR	<i>Infrared Radiation</i>
PIR	<i>Passive Infrared</i>
RF	<i>Radio Frecuencia</i>
MCU	<i>Microcontroller Unit</i>
PCB	<i>Printed Circuit Board</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
RDA	<i>Received Data Available</i>
APP	<i>Aplication</i>
IOT	<i>Internet of Things</i>
SME	<i>Sistema de Monitoreo Energético</i>

Lista de Símbolos

A	Área de las placas de un condensador	mm ²
C	Capacitancia	F
f _c	Frecuencia de corte de un filtro	Hz
f _p	Factor de potencia	-
G	Ganancia de un circuito	-
Hr%	Humedad relativa	%
I _c	Corriente de colector en transistor	mA
I _L	Corriente de Línea en una carga	A
I _{led}	Corriente de polarización para un led	mA
I _{Lrms}	Valor efectivo de la Corriente I _L	Arms
I _m	Corriente máxima	A
I _{otc}	Corriente de salida sensor de corriente	mA
K	Banda K de 18 a 27GHz	GHZ
Me	Memoria estimada	Byte
Mt	Memoria total	Byte
N	Relación de espiras de un transformador	-
P	Potencia promedio	W
R	Resistencia eléctrica	Ω
RXO	Pin receptor UART	-
S1	Salida 1 sistema medición del medidor IOT	V
S2	Salida 2 sistema de medición del medidor IOT	V
S3	Salida 3 sistema de medición del medidor IOT	V
S _a	Señal a, resultante del proceso de conversión en medidor IOT	V
S _b	Señal b, resultante del proceso de conversión en medidor IOT	V
T	Periodo de un señal 60Hz	Hz
T°C	Temperatura	°C
T _{df}	Tiempo de desfasamiento entre dos señales	S
t _{off}	Tiempo en bajo de un señal	S
t _{on}	Tiempo en alto de un señal	s
TXO	Pin transmisor UART	-
V _{be}	Voltaje base emisor	V
V _{ce}	Voltaje colector emisor	V
VDD	Voltaje de alimentación	V
V _i	Voltaje de entrada	V
V _L	Voltaje de línea de una carga	V
V _{led}	Voltaje de polarización de un LED	V
V _{Lrms}	Valor efectivo de la Corriente V _L	V
V _m	Voltaje máximo	V
V _o	Voltaje de salida	V
V _{offset}	Voltaje de offset	V

V_{otc}	<i>Voltaje de salida sensor corriente</i>	V
V_{ref}	<i>Voltaje de referencia</i>	V
W	<i>Consumo energetico</i>	J
β	<i>Ganancia de corriente</i>	-
Δt	<i>Intervalo de tiempo</i>	s
ϵ	<i>Permitividad electrica</i>	-
θ	<i>Angulo de desfasamiento</i>	-
μ	<i>Valor de pertenencia</i>	-

Introducción

Día con día incrementa la necesidad de disminuir el consumo energético de los equipos eléctricos que utilizamos cotidianamente, esto debido a que una parte de la energía eléctrica que utilizan estos, proviene de combustible fósiles, los cuales en su generación traen consigo problemas ambientales. Aunque dicho consumo energético está asociado a amplia variedad de aparatos eléctricos, una parte se debe a la utilización de equipos de climatización para proporcionar condiciones confort térmico a las personas en una habitación. Solo en Nicaragua el consumo energético vinculado con los aires acondicionados o **AA**, representa el 20% del consumo energético nacional, según **Mendieta (2014)**.

Dentro del consumo energético utilizado para la climatización, existe un recargo producido por el usuario debido a como este configura el termostato del AA, **Ahmed et al. (2016)**. En muchos de los casos, el usuario selecciona una temperatura muy por debajo de la requerida para el confort térmico, y luego es necesario realizar una reconfiguración de este mismo cuando la temperatura desciende demasiado, lo mismo ocurre en la otra dirección, cuando la temperatura incrementa. Este vaivén de la configuración de la temperatura en el termostato, en combinación con tiempo de respuesta del usuario o si el usuario no realiza estas configuraciones, se produce el recargo energético ya mencionado. Cabe señalar que definir normas de cómo utilizar este aparato, o que temperatura debe ser programa en el termostato, no es suficiente, ya que está a disposición del usuario acatarlas o no.

En este trabajo monográfico se propuso el desarrollo de un dispositivo electrónico, denominado controlador inteligente para aires acondicionados, que se encargue del control de un AA. Este dispositivo realiza la configuración del termostato, el encendido y apagado del aparato, utilizando comunicación infrarroja, como lo hace el control remoto del equipo, pero basándose en la humedad y temperatura interior, y utilizando un sistema fuzzy para la toma decisiones. Esto con el fin de reducir el consumo energético utilizado para la climatización.

Objetivos

Objetivo General

Desarrollar un controlador inteligente para la configuración automática de un aire acondicionado a partir de las condiciones en un ambiente interior.

Objetivos Específicos

- Diseñar los subsistemas de medición, comunicación y control del controlador inteligente a fin de proporcionar una guía para su próxima implementación.
- Realizar un estudio técnico de las tecnológicas disponibles para la selección de los componentes que se utilizaran en la implementación cada uno de los subsistemas.
- Verificar el funcionamiento, de manera individual y mediante simulación, de cada uno de los subsistemas para corregir posibles errores en la etapa de diseño.
- Verificar la efectividad del hardware, de manera individual, mediante la implementación de los subsistemas a nivel de breadboard.
- Realizar un test de evaluación en el que se integren todos subsistemas a nivel breadboard para verificar su funcionamiento de forma conjunta y corregir errores.
- Diseñar las PCBs requeridas para en controlador inteligente.
- Desarrollar un sistema de adquisición para el monitoreo del consumo energético y de las condiciones interiores, desde un entorno web que permita su almacenamiento en una base datos.
- Realizar la evaluación del sistema instalando en un aire acondicionado para verificar funcionamiento.

Justificación

En la época en la vivimos se requiere de un nuevo modelo basado en el ahorro energético, para contribuir en la reducción el consumo de combustibles fósiles en la generación de electricidad, evitando la emisión de contaminantes hacia la atmosfera. Si bien es cierto que el consumo de energía se debe a un gran número de aparatos eléctricos que utilizamos cotidianamente, una parte de este se debe al uso de equipos de climatización. Según **(Yang et al., 2003; Pérez-Lombard et al., 2008; Morosan et al., 2010)** citados por **Castilla et al., (2010)**, se considera que el consumo de energía en los edificios representa el 40% del consumo energético mundial, y que más de la mitad de este está asociado a sistemas de climatización. En lo que respecta a Nicaragua, se utilizan aires acondicionados, o AA, principalmente en oficinas de gobierno central, ministerios, alcaldías, entes regulados y no regulados, institutos técnicos y universidades, según **Mendieta (2014)**. Este mismo, partiendo del balance energético nacional, indica que el 20% del consumo energético nacional es utilizado en para este fin.

Dentro del consumo energético utilizado para la climatización, existe un recargo producido por usuario debido a como este configura el termostato del AA, **Ahmed et al. (2016)**. En muchos de los casos, el usuario selecciona una temperatura muy por debajo de la requerida para el confort térmico, y luego es necesario realizar una reconfiguración de este mismo cuando la temperatura desciende demasiado, lo mismo ocurre en la otra dirección, cuando la temperatura incrementa. Este vaivén de la configuración de la temperatura en el termostato, en combinación con tiempo de respuesta del usuario o si el usurario no realiza estas configuraciones, producen el recargo energético ya mencionado.

Una solución a esta situación, se expresa en el **decreto presidencial No. 2-2008** de ordenamiento del uso de energía, el que define normas y estatutos para minimizar el consumo energético, en las que se incluye medidas para regular el uso de los AA. Sin embargo, estas medidas no han tenido un buen resultado, ya que pues según informes oficiales, en 2016 el consumo energético asociado a este sector, descrito anteriormente, aumentó en un 10.5%, a pesar de las

capacitaciones y de la asistencia brindada a los usuarios, por el Ministerio de Energía y Minas para el uso de los AA.

Ante dicha situación y la importancia que tiene disminuir el consumo energético de los AA, se propuso desarrollar de un Controlador Inteligente para Aires Acondicionados Utilizando Lógica Difusa y Comunicación Infrarroja IR. Dicho dispositivo se encarga de la configuración automática del termostato, el encendido y apagado del aire acondicionado, basándose en la humedad y temperatura interior, utilizando un sistema fuzzy para la toma de decisiones, y enviando los comandos de configuración correspondientes mediante una comunicación infrarroja. La finalidad de este dispositivo es lograr una reducción en el consumo energético de este aparato.

Capítulo I:

Marco Teórico

1.1 El aire acondicionado

El aire acondicionado (AA) es un aparato utilizado para climatizar el interior de las habitaciones. Los más comunes son los de tipo Split y de los de tipo ventana, debido a su facilidad de instalación y su relativo bajo costo. Un sistema de AA no genera aire frío, sino que **extrae el calor del aire** de la estancia que se quiera climatizar **Arbanat (2016)**. **Arbanat** también explica que un aire acondicionado es un circuito frigorífico, a través del cual circula un gas refrigerante que cambia su estado de gas a líquido a medida que atraviesa los distintos componentes del mismo. Es gracias al cambio de estado del frigorífico que se logra extraer el calor sobrante del aire de una habitación, **Arbanat (2016)**. La Fig. 1-1 muestra el ciclo de refrigeración de un AA.

En el proceso de refrigeración intervienen múltiples elementos, es posible listar los más básicos. A continuación los elementos básicos de un AA señalados por **Arbanat (2016)**.

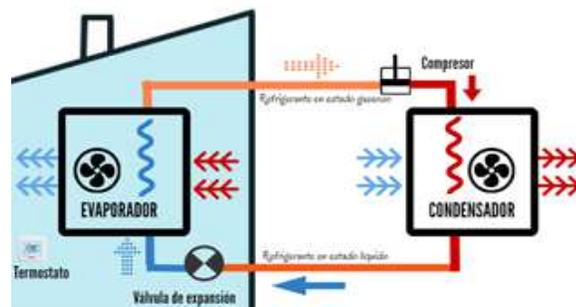


Figura 1-1. Ciclo de refrigeración de un A.A. Fuente: Arbanat (2016).

-A- Unidad interior: La unidad interior también llamada Split contiene el evaporador, donde ocurre el proceso de extracción del aire caliente, que cede su calor al gas refrigerante. Dentro de la unidad interior, un ventilador distribuye el

flujo de aire refrigerado a la estancia. Esta unidad interior cuenta con sensores de temperatura conectados al termostato **Arbanat (2016)**.

-B- Válvula de expansión: La válvula de expansión libera de la presión al gas refrigerante, que al atravesarla pasa de estado líquido a estado gaseoso **Arbanat (2016)**.

-C- Unidad exterior: La unidad exterior alberga el condensador y el compresor donde el gas refrigerante pasa de gas a líquido. Desde esta unidad se expulsa el aire caliente, extraído del interior, hacia el exterior **Arbanat (2016)**.

-D- Compresor: Genera una fuerza para comprimir el gas que llega del evaporador en estado gaseoso. Esta presión aumenta la temperatura del gas que vuelve a su estado líquido y se calienta.

Para fines de este trabajo monográfico, el compresor es el elemento de más interés, ya que es el elemento que consume más energía según **(Arbanat, 2016; Whitman & Johnson 2000)**. La forma en la que es controlado el compresor, marca la diferencia entre un equipo de climatización de alta y baja eficiencia.

En un AA de alta eficiencia la velocidad del compresor se regula de acuerdo con la temperatura deseada y la del interior de la habitación. En cambio, en un AA convencional esto no ocurre, la velocidad del compresor no es regulada, lo hace que estos equipos sean ineficientes energéticamente, según **Whitman & Johnson (2000)**. Sin embargo, los AA de alta eficiencia tienen un mayor costo, por lo cual los AA son más utilizados. En este caso nos centraremos en estudiar este tipo de AA.

Las unidades AA convencionales cuentan con una tarjeta electrónica, esta incorpora un sistema de control, uno de comunicaciones y uno de mediciones. En estos equipos no se realizan la regulación de la velocidad del compresor. El sistema de control se limita a habilitar o deshabilitar la alimentación del compresor, en dependencia de la temperatura programada por el usuario **Whitman & Johnson (2000)**. El control ON-OFF del compresor es llevado a cabo por el sistema de control mediante contactores. En la Fig. 1-2 se muestra la alimentación de compresor de un AA convencional.

El sistema de control opera de acuerdo a una configuración que realiza el usuario a través del control remoto. Este sistema recibe las configuraciones mediante una interfaz de comunicación infrarroja. Algunas de estas configuraciones son la temperatura de apagado del compresor, el modo de operación (COOL/FAN), y la velocidad del FAN. En la Tabla 1-1 se describen estas configuraciones.

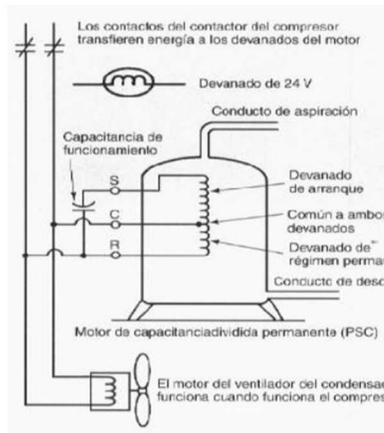


Figura 1-2. Compresor y ventilador exterior cableados en paralelo. Fuente: Whitman & Johnson (2000).

Tabla 1-1
Configuraciones de un aire acondicionado

Configuración	Opciones	Descripción
Modo Cool	Cool	Pone en operación el compresor
Modo Fan	Fan	Deshabilita el compresor y habilita únicamente el ventilador interior
Temperatura	17-30°C	Controla la temperatura de encendido y apagado del compresor.
FAM modo	Automático, bajo, medio, alto	Controla la velocidad de operación del ventilador interior.
Power	ON, OFF	Enciende o paga el equipo

1.2 Comunicación infrarroja IR

Un AA cuenta con su propio sistema de control, el cual puede ser configurado por el usuario utilizando el control remoto del AA. Cuando el usuario presiona uno de los botones de este dispositivo, se envía una trama de bytes, mediante comunicación Infrarroja IR, que contiene la información de la configuración seleccionada. En

esta configuración se puede encender o apagar el AA, modificar la velocidad del fan, cambiar la temperatura de operación del AA, etc.

En este caso, las configuraciones del AA serán realizadas por el controlador inteligente, basándose en su programación. Por lo cual, este dispositivo debe contar con una interfaz IR que le permita enviar mensajes al AA.

Una interfaz de comunicación IR, realiza una conversión de señales eléctricas a señales de luz, que contienen el mensaje de configuración y que el AA puede interpretar. El elemento que permite realizar esta conversión es el LED-IR, el cual emite luz con una longitud de onda de 940nm, según **(Clavijo, 2011; Altium, 2017)**. En la Fig. 1-3 se muestra un ejemplo una interfaz IR.

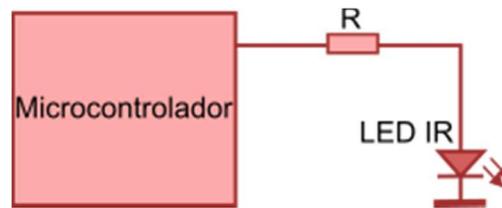


Figura 1-3. Esquema básico de interfaz de comunicación IR. Fuente: Autor.

En principio para la comunicación IR se requiere de una señal portadora de una frecuencia de 38KHz, la cual es modulada en amplitud para la codificación de un 0 o 1 binario **(Clavijo, 2011; Altium, 2017)**. Esta codificación se realiza a partir de un periodo fundamental T, típicamente entre 560 a 600us, en donde, la amplitud de la portadora es modificada, según el símbolo que se desea transmitir, lo indica **(Clavijo, 2011; Altium, 2017)**. En la Fig. 1-4 se muestra la codificación de un “0” y “1” en la portadora 38KHz.

Un mensaje de configuración contiene información del modo y temperatura de operación, y velocidad del fan. Estos mensajes poseen una estructura que puede variar según la marca del AA. En la Fig. 1-5 se muestra la estructura de un mensaje de configuración para un AA de la marca confort-Start, el equipo con el que se llevó cabo las pruebas. En esta figura, sinc, centr y stop, son señales que indican el principio, el centro y el final, respectivamente, de un mensaje de configuración. En la Fig. 1-6 se muestran con detalle estas señales.

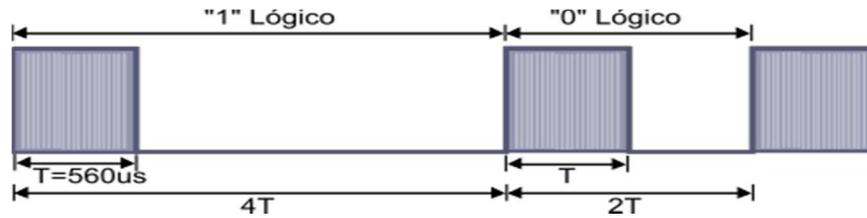


Figura 1-4. Modulación y Codificación de la portadora de 38KHz. Fuente: Autor.

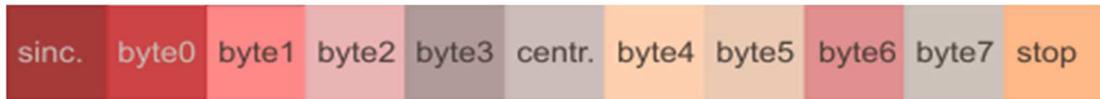


Figura 1-5. Protocolo IR, AA confort-start. Fuente: Autor.

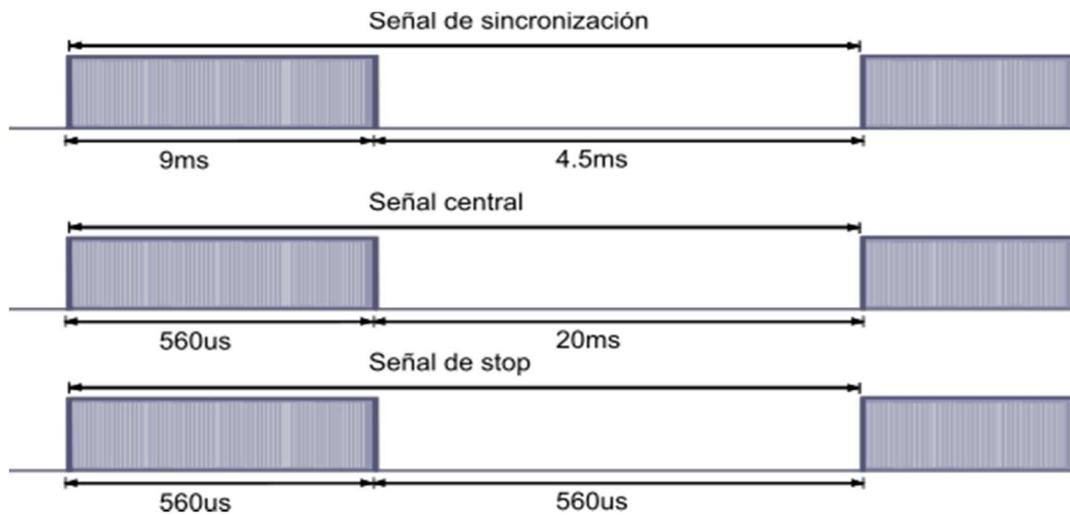


Figura 1-6. Señal de sincronización, de centro y final de mensaje. Fuente: Autor.

Por otro lado, el valor de los bytes del mensaje de configuración, mostrado en la Fig.1-5, cambia de acuerdo a la configuración de encendido o pagado, temperatura, y velocidad del fan. Sin embargo, algunos de estos poseen valores constantes sin importan la configuración seleccionada. En la Tabla 1-2 se muestran los bytes con valores fijos en un mensaje de configuración.

Tabla 1-2
Bytes protocolo IR

Byte	Byte2	Byte3	Byte4	Byte5	Byte6
Valor	20h	50h	01h	21h	00h

En lo que respecta a byte0, byte1 y byte7, estos si presentan dependencia a la configuración seleccionada. A continuación se describe esta dependencia.

El valor de **byte0** depende del ON-OFF del AA y de la velocidad del fan, AUTO, LOW, MED, HIGH, que sean seleccionados. En la Tabla 1-3 se muestran los valores de **byte0** con sus respectivas dependencias.

Tabla 1-3
Valores de *Byte0*

Comando	ON				OFF			
Comando	LOW	AUTO	MED	HIGH	LOW	AUTO	MED	HIGH
Valor	48h	58h	68h	78h	40h	50h	60h	70h

Luego se encuentra **byte1**, este es utilizado para la configuración de la temperatura. Su valor se rige de acuerdo a la Ec. (1-1), donde, **T°C** es la temperatura en grados Celsius dentro del rango $15^{\circ}\text{C} < \text{T}^{\circ}\text{C} < 31^{\circ}\text{C}$.

$$\text{Byte1} = T^{\circ}\text{C} - 16^{\circ}\text{C} \quad (1-1)$$

Por su parte, **byte7** presenta una dependencia al estado ON-OFF y a la temperatura seleccionados. En las ecuaciones (1-2) y (1-3) se describe el comportamiento de **byte7**. Donde *bits_temp*, se define en la Ec. (1-4), mientras que **bits_onoff**, tiene los valores de la Tabla 1-4.

$$\text{Byte7} = X \ll 4 \quad (1-2)$$

$$X = 4 + \text{bits}_{\text{onoff}} + \text{bits}_{\text{temp}} \quad (1-3)$$

$$\text{bits}_{\text{temp}} = T^{\circ}\text{C} - 16^{\circ}\text{C} + 1 \quad (1-4)$$

Tabla 1-4
Bits_OnOff

Comando	ON	OFF
Valor	00h	08h

1.3 Confort térmico

Uno de los factores a tomar en consideración en la climatización es el confort térmico, pues lo que se busca es mejorar la eficiencia de los aires condicionados, sin alterar la comodidad de los usuarios. Según la mayoría de los estándares internacionales (ISO7730, 2009; ASHRAE55, 2004) el confort térmico se puede definir como: “Aquella condición de la mente que proporciona satisfacción con el ambiente térmico”. En el control de climatización se busca la comodidad de las

personas, sin embargo, debe existir, de acuerdo con **Alcalá et al (2002)**, un equilibrio entre los límites estipulados de climatización para el confort térmico y el consumo energético.

Chávez (2002) señala que el hombre tiene un eficaz sistema regulador de temperatura, el cual asegura que la temperatura del centro del cuerpo se mantenga aproximadamente a 37°C. Sin embargo, nuestro cuerpo no percibe directamente la temperatura dentro de una habitación, sino más bien la transferencia de calor entre nuestro cuerpo y el ambiente (**Chávez, 2002**). En este proceso transferencia de calor, intervienen muchos factores, pero **Chávez** resalta: la velocidad de circulación, la humedad y la temperatura propia del aire. Si estos factores no se encuentran dentro valores adecuados, el sistema interior de cuerpo se desequilibra y entramos en disconfort (**Rosón, 2016**).

Por otro lado, la climatización fuera de los límites de confort térmico representa un recargo energético, pues se consume energía fuera de la necesaria. Por lo cual, es de vital importancia mantener la temperatura interior lo más cercano posible a los límites establecidos por la norma, tomando en consideración también la humedad relativa de este. Los estándares internacionales (**ISO7730, 2009; ASHRAE55, 2004**) definen los límites para la humedad y la temperatura para establecer condiciones de confort. En la Fig. 1-7 se muestra la zona de confort definida por estos estándares.

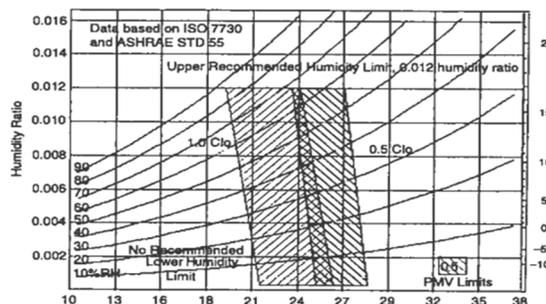


Figura 1-7. Zona de confort estándar. Fuente: ASHRAE55 (2004).

1.4 Medición de temperatura y humedad

Se ha mencionado con anterioridad que las variables de interés para determinar la configuración más apropiada del aire acondicionado son la temperatura y la humedad relativa, debido su relación con el confort térmico.

1.4.1 Medición de temperatura.

Existen una gran cantidad de sensores para realizar la medición de temperatura. Existen cámaras infrarrojas, termómetros digitales, termopares, termistores y circuitos integrados. Cada uno de ellos con rangos de operación distintos, y con aplicaciones diferentes. Es necesario tener en consideración las características a lo que se le quiere medir la temperatura, es decir, este puede ser un líquido, un gas, solido, etc, Conocer estas características es necesario para determinar la naturaleza del sensor que se utilizará para realizar la medición. En este caso, nos interesa la temperatura de ambiente interior, lo que limita bastante el rango de operación del sensor, presión requerida, y la robustez del sensor.

Según los datos históricos climáticos de la ciudad de Managua, reportados por **INETER (2012)** entre 1942-2017, la temperatura máxima promedio es de 33.5°C, en la ciudad de Managua. Además Según la mayoría de los estándares internacionales (**ISO7730, 2009; ASHRAE55, 2004**), la temperatura interior debe oscilar entre 23 y 26°C para establecer condiciones de confort. Por tal razón, el sensor de temperatura debe ser capaz de medir al menos 40°C.

Es posible encontrar una diversidad circuitos integrados diseñados para la medición de temperatura en ambientes interiores, ya calibrados, con rangos de medición bastante parecidos entre los -10°C hasta los 80°C, con presiones de medición similares entre los ± 0.1 y los ± 0.4 °C. La Tabla 1-5 muestra más detalles de este tipo de sensores.

Tabla 1-5
Sensores de temperatura para ambientes interiores

Temperatura	Rango de Medición	Precisión	Tipo de salida
TSYS01	-5 a 50°C	± 0.1	I2C
TMP116	-10 a 85°C	± 0.2	I2C
LMT87LPG	-50 a 150°C	± 0.4	analógica
MCP9700	-40 a 125°C	± 0.4	analógica

1.4.2 Medición de humedad relativa.

De acuerdo a **Michalski (2009)**, la humedad del aire se refiere a la cantidad de vapor de agua presente en el aire. **Michalski** también señala que la cantidad de vapor de agua que puede absorber el aire depende de su temperatura. De manera que, para una temperatura específica, el aire solo puede contener una cantidad máxima de vapor de agua, antes de que este se empiece a condensarse en forma líquida. El porcentaje de este valor máximo que se alcanza, para esa temperatura específica, es llamado humedad relativa (**Michalski, 2009**).

Existen varios tipos de sensores de humedad, de acuerdo al principio físico que siguen para realizar la cuantificación de la misma. Existen mecánicos, basados en sales higroscópicas, basados en la conductividad eléctrica, según su capacidad eléctrica, y también los hay infrarrojos y resistivos. La Tabla 1-6 muestra una breve descripción de estos.

Tabla 1-6
Sensores de temperatura para ambientes interiores

Tipo	Descripción
Mecánicos	Este tipo de sensores aprovecha los cambios en las dimensiones de ciertos materiales en presencia de humedad.
Sales- Higroscópicas	Deducen el valor de la humedad en el ambiente a partir de una molécula cristalina que tiene mucha afinidad con la absorción de agua.
Conductivos	Disponen de una superficie no conductora sobre la cual se posicionan dos rejillas de oro entrelazadas, sin tocarse. La corriente que se establezca debido una diferencia de potencia entre estas dos rejillas será proporcional a la cantidad de moléculas de agua presentes entre ellas.
Capacitivos	Se basan sencillamente en el cambio de la capacidad que sufre un condensador en presencia de humedad.
Infrarrojos	Se basan el hecho que las moléculas pueden absorber la energía de las ondas electromagnéticas. De manera que con una fuente de luz, con la longitud de onda adecuada para ser absorbida por el agua, emitiendo radiación hacia una muestra, esta absorberá parte de esta radiación. La diferencia entre radiación emitida y la resultante luego de la absorción, es debida a la cantidad de agua contenida en la muestra.

A pesar de las diferentes tecnologías de sensores que existen para medir la humedad relativa, los más difundidos en la industria y meteorología, debido a su

fácil producción, bajo costo y fidelidad, según **Glaría & Kouro (2001)**, son los de tipo capacitivo. En el mercado actual es posible encontrar circuitos integrados IC para realizar la medición de la humedad relativa del tipo capacitivo, como los sensores de la serie HIH-4000 desarrollados por HONEYWELL INTERNATIONAL, que cuentan con un elemento capacitivo con polímero termoenduresible ajustado con láser con acondicionador de señal incorporado en el chip. Debido a su amplia disponibilidad nos centraremos en explicar más en detalle el funcionamiento de los sensores capacitivos.

Los sensores capacitivos se basan en el cambio que sufre la capacidad (C en Farad) de un condensador al variar la constante dieléctrica del mismo, (**Glaría & Kouro 2001**). La capacidad de un capacitor se encuentra determinada como se muestra en la Ec. (1-5). Donde C, es el valor de la capacidad; ϵ , es la permitividad del dieléctrico (constante dieléctrica); A, el área de las placas del condensador; y D la distancia entre las placas del condensador.

$$C = \epsilon \cdot \frac{A}{D} \quad (1-5)$$

Una molécula de agua está compuesta por 2 átomos de hidrogeno y una de oxígeno. Las moléculas de agua tienen una disposición neta de sus cargas, geoméricamente hablando, similar a las de un dipolo eléctrico. De manera que, según **Glaría & Kouro**, en presencia de un campo eléctrico, las moléculas de agua se alinean en mayor proporción que las de aire, por tal razón, la constante dieléctrica del agua es mayor que la del aire.

Los sensores de humedad toman muestras del aire circundante, a partir de la cual, ante la presencia de una mezcla gaseosa que contenga vapor de agua, se produce una variación de C_x proporcional a la cantidad de moléculas de agua presentes en el aire, estableciendo así una métrica para medir la humedad (**Glaría & Kouro, 2011**).

1.5 Sensores de presencia

1.5.1 Sensores PIR.

Este tipo de sensores son los más utilizados para aplicaciones de control de iluminación y climatización. El sensor infrarrojo pasivo (PIR) detecta cambios en

el flujo de calor causado por el movimiento del cuerpo humano. Según **Tsai & Young (2013)**, la radiación emitida nuestro cuerpo se mueve en el rango de los infrarrojos con una longitud de onda máxima de $9.4\mu\text{m}$. Los materiales piroeléctricos del sensor PIR detectan esta radiación y generan energía en respuesta. Además, **Tsai & Young** señalan que cuando la radiación infrarroja impacta en los elementos piroeléctricos su temperatura aumenta un poco, dilatándolos y reorganizando su estructura cristalina, creando perturbaciones en su estructura, que permiten detectar la presencia de un cuerpo humano.

Además, **Tsai & Young** añaden que los detectores PIR cuentan con una lente de Fresnel para enfocar la radiación hacia los elementos del sensor, así como también elementos duales para restar la temperatura ambiente y reducir el número de alarmas falsas.

Este tipo de tecnología es muy adecuada para detectar movimiento humano en un rango amplio en una zona de detección grande, pero no es muy efectiva para movimientos pequeños y tampoco es capaz de distinguir entre objetos que se acercan y se alejan, así que únicamente detecta correctamente movimientos horizontales.

1.5.2 Sensores tipo radar.

Este tipo de sensores se basan el efecto Doppler para determinar la presencia de personas. Estos irradian una señal de referencia a determinada frecuencia dentro de la banda K. según el efecto Doppler, la frecuencia de esta señal se verá afectada debido al movimiento de una persona, cuando esta se acerque o aleje del detector, tal como lo describe **Kiss (2015)**. **Kiss** también señala que este tipo de sensores pueden detectar movimientos ligeros del cuerpo humano. La banda K, tiene permitido su uso prácticamente en todo el mundo, y esta tiene rango de frecuencia entre 18 a 27GHz.

Existen ciertos módulos detectores de movimiento integrados que se basan en la tecnología de efecto Doppler con microondas a 24GHz. Uno de ellos es el NJR4265 de New Japan Radio Corporation. Este módulo integra la antena, el circuito de RF, el amplificador FI, la unidad MCU y el regulador de tensión en un tamaño reducido de 14x20.4x8.8mm. El NJR4265 J1 es un sensor de movimiento

inteligente diseñado para detectar objetos con movimientos a velocidad baja. El rango máximo de detección es de 10m en $\pm 35^\circ$, con una velocidad de aproximación de 0.25 a 1m/s. Este sensor es mostrado la Fig. 1-8.

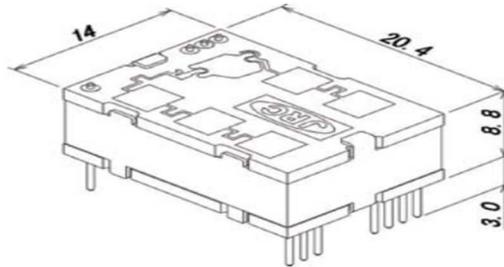


Figura 1-8. Sensor NRJ4265. Fuente: Kiss (2015).

1.5.3 Sensores térmicos.

Los sensores térmicos son esencia de cámaras térmicas de baja resolución. Un ejemplo de este tipo de sensores es el Grid-Eye de Panasonic, el cual es un sensor infrarrojo de tipo termopila, que detecta la cantidad de rayos infrarrojos. El dispositivo Grid-Eye es capaz de detectar la temperatura y los gradientes de temperatura sin contacto a lo largo de toda la superficie indicada, dentro de un ángulo de visión de 60° gracias a una lente de silicio integrada, ofreciendo una imagen térmica del entorno observado de 8x8 pixeles, **Kiss (2015)**. La detección de cuerpos humanos es posible gracias a la medición de temperatura sin necesidad de contacto, ya sea que estén en movimiento o en reposo, **Kiss, (2015)**. La Fig. 1-9, muestra el sensor Grid-Eye y una ilustración de la imagen térmica que proporciona a partir de las mediciones realizadas.

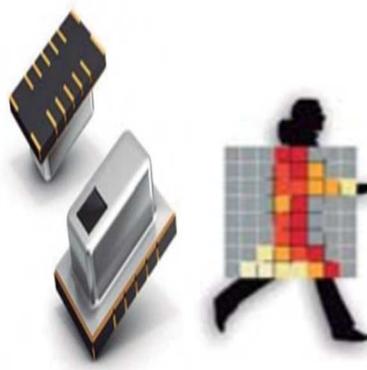


Figura 1-9. Sensor Grid-Eye. Fuente: Kiss (2015). Tomado de Kiss (2015).

1.6 Control fuzzy.

Un controlador es un elemento dentro de un sistema que se encarga de gobernar, dirigir o coordinar un proceso, por ejemplo, un llenado de un tanque, o calefacción de una habitación. El controlador es capaz de controlar un proceso alterando el estado de este, para ello el controlador inyecta o extrae energía de sistema físico (proceso) que se desea controlar a través del elemento final de control, **Katsuhiko (2010)**. Por otro lado, para el proceso adopte el comportamiento esperado, el controlador necesita conocer el estado actual de este, es decir, se requiere de una retroalimentación, como lo señala **Katsuhiko (2010)**.

Además de la retroalimentación en el sistema, también se requiere de un algoritmo para la toma de decisiones. En el caso del control convencional, se encuentran varios métodos, PID, Robust-Control, Adaptive-Control, Stochastic-Control, etc., Sin embargo, estas alternativas se enfocan en la modelación y el uso de modelos descritos mediante ecuaciones diferenciales, lo que los vuelve muy difícil de implementar, y en algunos casos imposibles, **(Katsuhiko, 2010; Passino & Yurkovich, 1998)**.

Opuesto a lo anterior, el control fuzzy se enfoca en obtener una comprensión intuitiva de cómo controlar un proceso, basándose en la experiencia del operador o usuario. El control fuzzy provee una metodología sistemáticamente organizada para presentar, manipular e implementar el conocimiento emperico humano acerca de cómo controlar un proceso, según lo siguiere **Passino & Yurkovich (1998)**.

Existen dos arquitecturas comúnmente utilizadas de sistemas fuzzy, la Mamdani y la Takagi-Sugeno. El sistema fuzzy desarrollado para este proyecto, está basado en una arquitectura Mamdani, la cual consiste principalmente de cuatro elementos: fuzzificadores, mecanismo de inferencia, base de reglas fuzzy, y defuzzificadores. En la Fig. 1-10 se muestra la arquitectura de este tipo sistema, y en lo que resta de esta sección, se realizará una descripción de las características, elementos y funcionalidad de un sistema fuzzy tipo Mamdani.

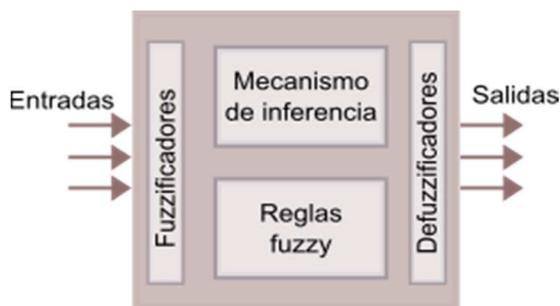


Figura 1-10. Sistema fuzzy tipo Mamdani.
Fuente: Autor.

1.6.1 Universo de discurso y conjuntos fuzzy.

Un universo de discurso es el rango dinámico de una variable física dentro de un proceso en particular, que puede ser la humedad, la temperatura, la velocidad, etc., esto según **(Passino & Yurkovich, 1998; Maguiña, 2010)**. Por otro lado, **Passino** describe que los conjuntos fuzzy corresponden a segmentos de un universo de discurso, donde cada segmento es caracterizado por un valor lingüístico a través de una etiqueta, tales como bajo, medio, alto, etc. En la Fig. 1-11 se ilustra el universo de discurso para la temperatura con los conjuntos fuzzy baja, media y alta.



Figura 1-11. Universo de discurso de la temperatura, con sus respectivos conjuntos fuzzy. Fuente: Autor.

1.6.2 Funciones de pertenencia y fuzzificación.

Se dice que los conjuntos fuzzy son una extensión de los conjuntos clásicos. Esto se debe a que el valor específico o elemento de una variable, dentro de conjunto clásico, pertenece o no pertenece a este. Sin embargo, en los conjuntos fuzzy, al elemento de la variable cuestión se le asignan un grado de pertenencia, de manera que este puede pertenecer total o parcialmente a un conjunto. En cualquier caso, la pertenencia de un elemento siempre tiene un valor entre 0 y 1.

El grado de pertenencia de un elemento dentro de un conjunto fuzzy se encuentra descrito mediante funciones de pertenencia, de las cuales existen diferentes tipos, pero las más utilizadas son las triangulares y las trapezoidales. En la Fig. 1-12 se ilustran este tipo de funciones. Por un lado, una función de tipo

triangular, se encuentra definida mediante la Ec. (1-6), mientras que una de tipo trapezoidal, se define según la Ec. (1-7), esto según **Passino & Yurkovich (1998)**.

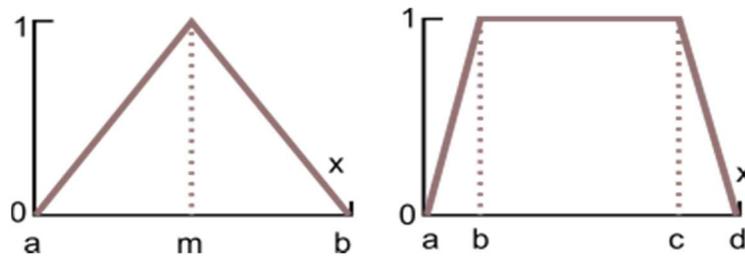


Figura 1-12. Funciones de pertenencia de tipo triangular y trapezoidal. Fuente: Autor.

$$\mu_A(x) = \begin{cases} 0, & \text{si } x \leq a \\ \frac{x-a}{m-a}, & \text{si } x \in (a, m] \\ \frac{b-x}{b-m}, & \text{si } x \in (m, b) \\ 1 & \text{si } x \geq b \end{cases} \quad (1-6)$$

$$\mu_A(x) = \begin{cases} 0, & \text{si } (x \leq a) \vee (x \geq d) \\ \frac{x-a}{b-a}, & \text{si } x \in (a, b] \\ 1, & \text{si } x \in (b, c) \\ \frac{d-x}{d-c}, & \text{si } x \in (c, d) \end{cases} \quad (1-7)$$

El procedimiento de determinar y asignar el grado pertenencia a un elemento dentro de un conjunto fuzzy, es denominado fuzzificación, **Passino & Yurkovich (1998)**. Considere un elemento x_i del universo X , con los conjuntos fuzzy Bajo, Medio y Alto. También considérense las funciones de pertenencia triangulares, ilustradas en la Fig. 1-13. El grado de pertenencia del elemento x_i , en los conjuntos Bajo, Medio, y Alto, correspondería a μ_{BAJO} , μ_{MEDIO} , μ_{ALTO} , respectivamente.

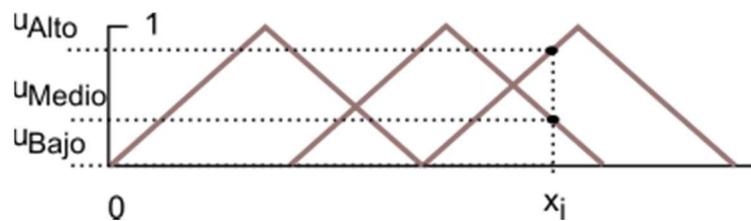
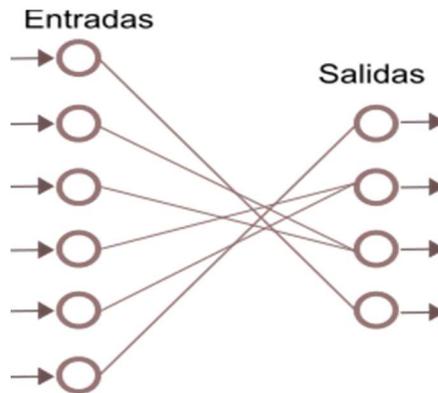


Figura 1-13. Proceso de fuzzificación. Fuente: Autor.

1.6.3 Reglas Fuzzy y operadores lógicos.

Uno los aspectos más importantes de los sistemas fuzzy, es que estos toman sus decisiones basándose en una base o conjunto de reglas. Cada regla le permite al desarrollador, definir la respuesta del sistema, ante las entradas que se

presenten en una situación específica. La decisión que se expresa en cada regla, esta basa en la experiencia del usuario u operario del proceso que se desea controlar, según lo plantea **Passino & Yurkovich (1998)**. La base reglas permite interconectar los conjuntos de entrada con los de salida, como se muestra en la Fig. 1-14, donde cada línea representa una regla.



*Figura 1-14. Mapeo de conjuntos de entrada y salida, (Base de reglas).
Fuente: Autor.*

En un sistema fuzzy, cada regla posee un antecedente y un consecuente. El antecedente corresponde a la situación que desea ser evaluar, y el consecuente corresponde a la respuesta que debe darse ante dicha situación. Un antecedente puede estar constituido por una conjunto fuzzy o por la combinación de dos más conjuntos fuzzy de entrada. Esta combinación, se lleva a cabo mediante operadores lógicos OR o AND. De igual forma, un consecuente puede estar constituido por un conjunto fuzzy o por combinaciones de conjuntos fuzzy de salida. A continuación, en la Fig. 1-15 se ilustra la estructura de las reglas fuzzy.

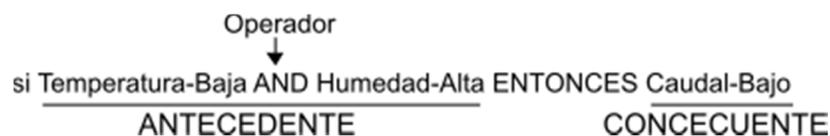


Figura 1-15. Estructura de reglas fuzzy. Fuente: Autor.

La inferencia es procedimiento en el que se evalúan las reglas de un sistema fuzzy, y se determinan cuales están activas, según **Passino & Yurkovich (1998)**. Se dice que una regla esta activa, si los elementos de entrada, cumplen con la condición descrita en el antecedente de la regla en cuestión.

Por otro lado, con respecto a las implicaciones fuzzy, primeramente debe recordarse que una regla permite realizar una conexión entre conjuntos fuzzy de entra y salida. Manera que cada regla tiene una implicación fuzzy. Una implicación corresponde a una modificación en la función de membrecía del conjunto de salida, debido a los valores de pertenencia de entrada.

Considérese la siguiente regla fuzzy.

si Temperatura-alta AND Humedad-baja ENTONCES Caudal-Bajo

La implicación fuzzy correspondiente a esta regla, es ilustrada en la Fig. 1-16. Como se aprecia, la función de membrecía del conjunto fuzzy de salida (caudal-bajo), sufrió una transformación triangular-a-trapezoidal, donde la función resultante tiene una altura máxima μ_{baja}^H . La altura o valor máximo de la función resultante, μ_{salida} , depende del operador utilizado en el antecedente de la regla, si se utiliza el operador AND, μ_{salida} está dado por la Ec. (1-8), en cambio, al utilizar el operador OR, μ_{salida} está dado por la Ec. (1-9).

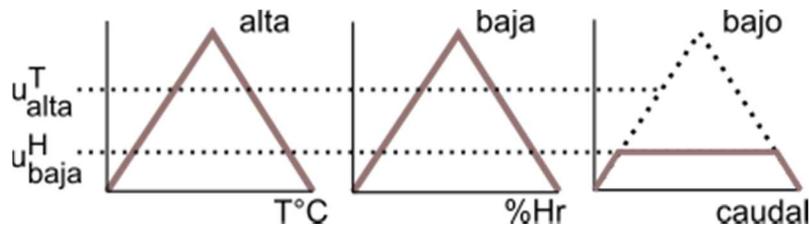


Figura 1-16. Implicación fuzzy. Fuente: Autor.

$$\mu_{salida} = \min(\mu_A(x), \mu_B(x)) \quad (1-8)$$

$$\mu_{salida} = \max(\mu_A(x), \mu_B(x)) \quad (1-9)$$

En el proceso de inferencia todas reglas de sistema fuzzy son evaluadas, y aquellas activas tienen una implicación fuzzy que modifican los conjuntos fuzzy de salida. En la Fig. 1-17, se ilustra el resultado del proceso de inferencia y sus implicaciones sobre los conjuntos fuzzy de salida (caudal).

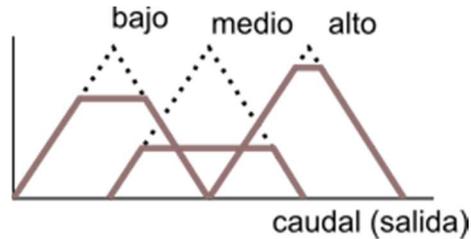


Figura 1-17. Funciones de membresía de los conjuntos fuzzy de salida luego del proceso de inferencia. Fuente: Autor.

1.6.4 Defuzzificación.

La defuzzificación es un proceso matemático usado para convertir los conjuntos fuzzy implicados en la salida en un número real, según Passino & Yurkovich (1998). Dentro de los cuales, existen diversos métodos, método por centro de gravedad, método por media máximo, método del máximo más chico, por ejemplo. Sin embargo, el más utilizado es el método por centro de gravedad.

En este método evalúa del centro de gravedad de la superficie descrita por las funciones de pertenencia resultantes del proceso de inferencia, véase el ejemplo en la Fig. 1-18. El número real de salida es la componente equis (x) del centro de gravedad de la superficie, la cual se calcula según la Ec. (1-10).

$$y_q^{Crisp} = \frac{\sum_{i=1}^R b_i \int \mu_{B_i}(y_q) dy_q}{\sum_{i=1}^R \int \mu_{B_i}(y_q) dy_q} \quad (1-10)$$

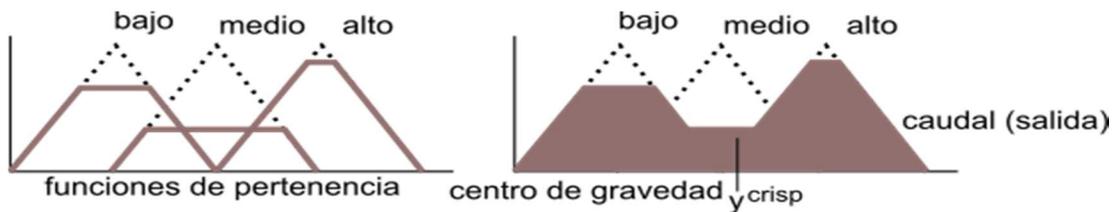


Figura 1-18. Método del centro de gravedad. Fuente: Autor.

Donde R es el número de reglas, b_i es el centro del área de la función de membresía del conjunto fuzzy implicado B_i y

$$\int \mu_{B_i}(y_q) dy_q$$

Denota el área bajo la función de membresía $\mu_{B_i}(y_q)$.

1.7 Potencia y energía

Es importante para este proyecto que el lector tenga en claro las diferencia entre potencia y energía. La energía se define como la capacidad que tiene la materia de producir un trabajo en forma de movimiento, luz, calor, etc., en cambio, la potencia es la rapidez con la que una carga consume energía, según **Young & Freedman (2009)**.

Los electrodomésticos que utilizamos transforman la energía eléctrica en otro tipo diferente de energía para realizar una función determinada, que puede ser calor, luz, acústica u otra. Estos aparatos toman la energía del servicio eléctrico comercial, lo que significa que se alimentan de corriente alterna AC. En este caso, es necesario especificar, como una carga régimen senoidal consume energía.

La energía utilizada por una carga, depende de la rapidez con la que se consume la energía, denominada potencia promedio, **P**, y el tiempo en el que la carga consume esta energía, **Δt**, (**Charles & Sadiku, 2006; Irwin & Nelms, 2011**).

La potencia promedio en régimen senoidal es determinada según la Ec. (1-11), donde **I_m** es amplitud de la corriente a través de la carga, mientras que **V_m** es la amplitud del voltaje de alimentación. El tercer elemento de esta ecuación, **Cos(θ)**, es denominado factor de potencia, en el que θ es el ángulo de desfaseamiento entre **I_m** y **V_m**.

$$P = \frac{1}{2} V_m I_m \cos(\theta_v - \theta_i) [W] \quad (1-11)$$

Por otro lado, la energía utilizada por una carga es determinada según la Ec. (1-12).

$$W = P \cdot \Delta t \quad (1-12)$$

Reemplazando **P**.

$$W = \frac{1}{2} V_m I_m \cos(\theta_v - \theta_i) \cdot \Delta t \quad (1-13)$$

1.8 Método de selección cualitativo

El método de selección cualitativo es un procedimiento sistemático utilizado para la toma de decisiones. En este se consideran un conjunto de opciones posibles, que son evaluadas de acuerdo a un conjunto de criterios. Cada opción

es calificada de acuerdo con cada criterio, y como resultado, la opción con un mayor puntaje es seleccionada, según **Baca (2013)**. A continuación se describe este método.

Primeramente, se elabora una lista de las opciones posibles, como la mostrada en la Tabla 1-7.

Tabla 1-7

Lista de opciones

Ítem	Opción
1	Opción-A
2	Opción-B
3	Opción-D

Segundo, se definen los criterios que deben ser tomados en cuenta durante la decisión. Donde a cada criterio se le asigna un valor relevancia. Este valor expresa la importancia de un criterio de respecto a los otros. Además, debe tenerse en consideración que la relevancia o peso de un criterio, es un número entre cero y uno, de manera que la suma de los pesos, sea igual a uno. Considérese el siguiente ejemplo en la Tabla 1-8.

Tabla 1-8

Criterios con sus respectivos pesos.

Ítem	Peso
Criterio 1	0.8
Criterio 2	0.3
Total	1

El tercer paso, consiste en la elaboración de una matriz de selección, como la mostrada en la Tabla 1-9. En esta matriz, se exponen las opciones y los criterios establecidos. Luego, cada opción es calificada en la medida que cumple con cada criterio. Seguido de esto, se lleva a cabo ponderaciones de las calificaciones, esto consiste en realizar el producto de la calificación asignada a una opción, y el peso del criterio en cuestión, véase la Tabla 1-9. Como paso final, se realiza la sumatoria de las ponderaciones, obteniendo un puntaje final por opción. La opción con mayor puntaje es la seleccionada.

Tabla 1-9
Matriz de selección

Opciones	Criterios	Criterio 1	Criterio 2	Puntaje final
	Pesos	0.30	0.70	
Opción A	calificación	8.00	8.00	
	ponderación	2.40	5.60	8.00
Opción B	calificación	7.00	6.00	
	ponderación	2.10	4.20	6.30
Opción C	calificación	5.00	8.00	
	ponderación	1.50	5.60	7.10

1.9 Módulo ESP01

El ESP01 es un módulo de comunicación WIFI basado en el esp8266EX el cual es un sistema en chip o SoC, **Hernández (s.f. b)**. La ESP8266EX integra los switches de antena, un balun RF, amplificador de recepción de bajo ruido, filtros, un MCU de 32bits, etc., entre otros elementos, así lo señala **Espressif Systems (2015)**. Este también indica que la ESP8266EX cumple con el estándar 802.11 b/g/n y trabaja en la banda 2.5GHz [4].

Por otro lado, **Hernández (s.f. b)** resalta que el esp8266EX, es el elemento más importante del módulo ESP01, pero que además este último incluye una memoria Flash, una antena WIFI, LEDs, y pines de conexión, véase la Fig. 1-19.

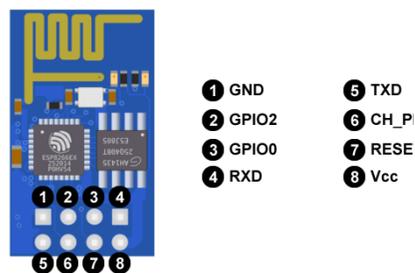


Figura 1-19. Módulo ESP-01.
Fuente: Espressif Systems (2015).

Martin (s.f.) señala que Arduino permite la programación o la creación de un firmware para el esp01, como si se tratara de una tarjeta más de Arduino. Lo único que debe realizarse es la instalación del plugin ESP8266 en el IDE Arduino. Dicho plugin incluye la librería ESP8266Wifi. Dicha librería, ofrece una larga lista

de funciones que permiten gestionar los recursos del módulo esp-01 para aplicaciones IoT.

La librería permite establecer una conexión WIFI y conocer el estado de la misma. Además, provee diversos modos de operar el esp8266EX, denominados clases. Este módulo permite operar como una estación, un punto de acceso, un cliente, cliente seguro, o como servidor.

En la clase Cliente que puede acceder a los servicios proporcionados por los servidores para enviar, recibir y procesar datos, lo explica **Grokhov (2017)**. Véase la Fig. 1-20. En esta clase, el ESP01 y el servidor establecen una comunicación TCP para intercambiar información. Si la conexión es exitosa, es posible enviar mensajes HTTP y recibir mensajes HTTP.



Figura 1-20. ESP8266 operando como cliente. Fuente: Grokhov (2017).

1.10 Elementos del sistema de monitoreo energético (SME).

Esta sección tiene el objetivo describir las características de los elementos de software y de hardware, utilizados en la implementación de un sistema web para el registro de consumo de energía de un AA. Primeramente, se indican los elementos de hardware, y luego los de software.

1.10.1 Hardware SME.

Desde el punto de vista de hardware, este sistema se encuentra integrado por un medidor de consumo de energía o medidor-CE, un ordenador-servidor, un ordenador-cliente y un router, mostrados en la Fig. 1-21. Estos dispositivos se encuentran interconectados a través de una red WIFI.

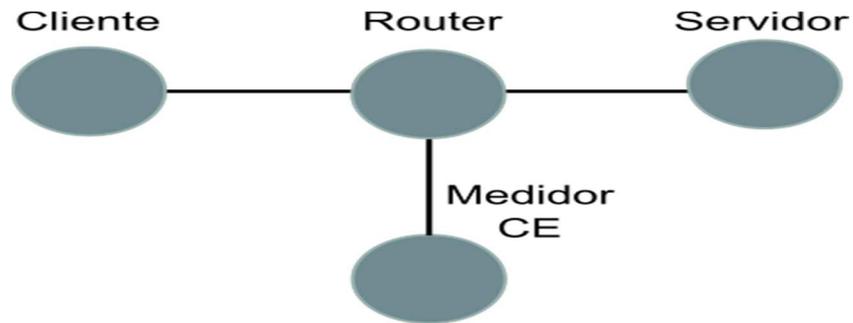


Figura 1-21. Elementos de hardware del sistema de monitoreo. Fuente: Autor.

El **medidor-CE** realiza mediciones del voltaje y corriente utilizados por una carga, para determinar su potencia y la energía utiliza en periodo de tiempo específico. Este dispositivo posee una interfaz WIFI que le permite enviar las mediciones de energía hacia el servidor, donde se almacenan. En el capítulo III se profundiza más sobre las características del medidor-CE.

Por otro lado, el servidor es el ordenador en el que se ejecutan las aplicaciones capaces de atender las peticiones del cliente. En el **SME** se utiliza una **Raspberry Pi 2**, la cual es un ordenador de placa única, de 85x54mm **González (2015)**. Este esta placa aloja un chip Broadcom BCM2835 con procesador ARM hasta a 1 GHz de velocidad, GPU VideoCore IV y hasta 512 Mbytes de memoria RAM, según **González (2015)** y **Castro (2014)**. Además, según los mismos, la Raspberry integra puertos USB, HDMI, Ethernet, 40 pines GPIO y un conector para cámara. En la Fig. 1-22 se muestra la Raspberry Pi 2.



Figura 1-22. Raspberry Pi 2. Fuente: González (2015).

Por su parte, el router se encarga del direccionamientos de los datos de los dispositivos conectados en la red. En cambio, el cliente es dispositivo dentro de la red que usuario utiliza para consultar las mediciones, este puede ser un computador personal o un Smartphone.

1.10.2 Software SME.

El primer elemento que destaca, a nivel de software, es el sistema operativo. La raspberry pi 2 utiliza el sistema operativo **Raspbian**, el cual es una distribución de GNU/LINUX, desarrollado para esta tarjeta, según **González (2015)**. Este también describe que esta distribución permite moverse ágilmente en el hardware de la Raspberry Pi, esta también utiliza un entorno de escritorios LXDE y Midori como navegador web predeterminado. Además, esta distribución incluye herramientas de desarrollo, como IDLE para Python, Scratch para programar videojuegos. En la Fig. 1-23 se muestra el logotipo de este sistema operativo.

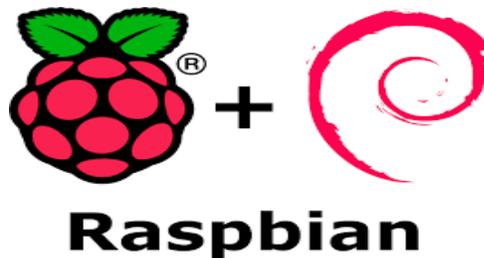


Figura 1-23. Logotipo del sistema operativo Raspbian. Fuente: Long (2016).

Además del sistema operativo Raspbian, el **SME** utiliza diversas aplicaciones para operar, algunas que operan del lado del cliente, y otras del lado del servidor, tal como se muestra en la Fig. 1-24. Del lado del servidor se encuentra el servidor web (Apache), el código desarrollado para cargar y solicitar las mediciones a la base datos (PHP), y la base de datos (MySQL). En cambio, del lado del cliente, el navegador del usuario ejecuta el código del sitio web que le permite consultar las mediciones, en este caso, se utilizan tres lenguajes HTML, CSS y JavaScript. A continuación se describe los elementos mostrados en la Fig. 1-24.

-Apache HTTP Server- es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1, así lo describe **Apache HTTP Server Project (2018)**. Este programa atiende las solicitudes del cliente, GET, POST, etc., a través del puerto TCP 80.

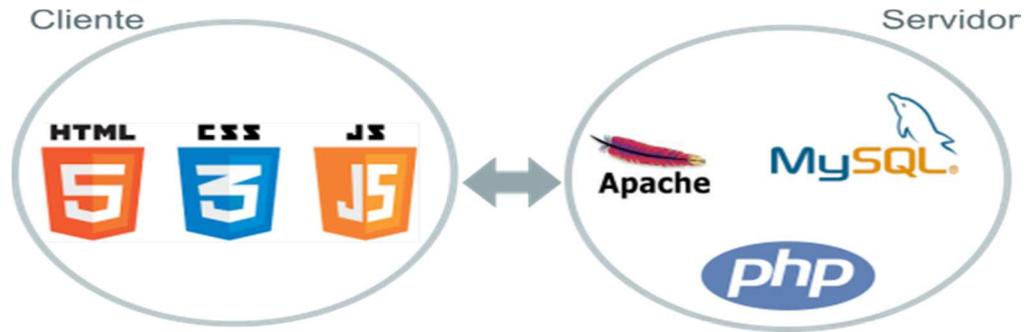


Figura 1-24. Tecnologías de software para el SME. Fuente: Autor.

-**PHP**- “*Hypertext PreProcessor*” es un lenguaje de código abierto utilizado para el desarrollo web y que puede ser incrustado en HTML, esto según **The PHP Group (2018)**. Este es utilizado como lenguaje de backend, es decir, se ejecutado en el servidor, generando un HTML y enviándolo al cliente. Con este lenguaje se pueden recopilar datos de formularios, generar páginas con contenidos dinámicos, enviar y recibir cookies, realizar conexiones con una base de datos, así describe **The PHP Group (2018)**.

-**MySQL**- es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. En otras palabras, MySQL es una aplicación que permite gestionar archivos llamados de bases de datos, según **EsepeStudio (2015)**. Esta aplicación nació como una iniciativa de Software Libre y aún sigue ofreciéndose como tal, para usuarios particulares. MySQL puede es compatible con múltiples plataformas, como PHP, y ofrece una infinidad de aplicaciones que permiten acceder rápidamente a las sentencias del gestor de base de datos.

-**HTML**- “*HyperText Markup Language*” es un lenguaje de marcación de elementos para la creación de documentos de hipertextos, según **Álvarez (2001)**. Este lenguaje permite organizar, estructurar y dar significado al contenido de un sitio web mediante elementos. De los cuales diferentes tipos de elementos de secciones, de agrupación, de ediciones, de contenido incrustado, entre otros, como describe **MDN web docs (2017)**. Sin embargo, cada elemento está conformado por un contenido, etiquetas, y atributos. Las etiquetas indican el inicio y el final de un elemento, y por su parte, los atributos son utilizados para

caracterizar cada elemento, ya sea de forma conjunta con otros elementos, o de forma individual, **Pino (2017)**.

-CSS- "*Cascading Style Sheets*" es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG, según **MDN web docs (2018)**. El cual indica que CSS sirve para dar estilo y diseño a las páginas web, tales como cambiar de tipo de letra, color, tamaño, definir el espaciado del contenido, mostrar múltiples columnas, añadir animaciones y otras funcionalidades.

-JavaScript- Es el lenguaje interpretado orientado a objetos desarrollado por Netscape que se utiliza en millones de páginas web y aplicaciones de servidor en todo el mundo, **MDN web docs (2015)**. **MDN** también resalta el hecho que JavaScript no es "Java interpretativo", sino que este utiliza una sintaxis similar Java y C++ con la intención de reducir el número de nuevos conceptos necesarios para aprender el lenguaje, pero que es un lenguaje de programación dinámico que soporta construcción de objetos basado en prototipos.

Capítulo II: Análisis y presentación de resultados

2.1 Diseño de controlador Inteligente.

2.1.1 Diseño funcional del controlador.

El diseño funcional representa el punto de partida en el proceso de diseño de un sistema electrónico. Aquí se muestran los elementos que integran dicho sistema. Cabe señalar que estos elementos fueron definidos según los requerimientos establecidos. Para lo cual se tomaron en consideración las normas ISO7730 y ASHRAE55 que definen los parámetros que intervienen en el confort térmico, en las que se indica que la humedad y temperatura son las variables con más influencia en el confort térmico. Además, para establecer los requerimientos, también se tomaron en consideración las entrevistas realizadas a los responsables de los laboratorios de electrónica de la FEC-UNI, Véase anexo 1 y anexo 2. Estos proporcionaron información sustancial sobre sus hábitos en el uso de los AA, y de las necesidades presentadas con estos equipos. En la Tabla 2-1 se muestra la lista de requerimientos para este dispositivo.

Partiendo de estos requerimientos se llevó a cabo el diseño funcional del dispositivo. Los elementos definidos son un sistema de control, uno de comunicaciones, uno de medición, uno de señalización y uno de alimentación, tal como se muestra en la Fig. 2-1. A lo largo de este capítulo se profundizarán los aspectos, y características de cada uno de estos elementos.

Tabla 2-1

Requerimientos del controlador inteligente

Ítem	Requerimientos
1	Realizar la medición de la humedad y temperatura interior
2	Detectar presencia de personas
3	Determinar el tiempo de climatización necesario
4	Enviar las ordenes requeridas mediante comunicación IR
5	Indicar visualmente al usuario la operatividad de dispositivo
6	Permitir la conectividad del dispositivo con un Smartphone

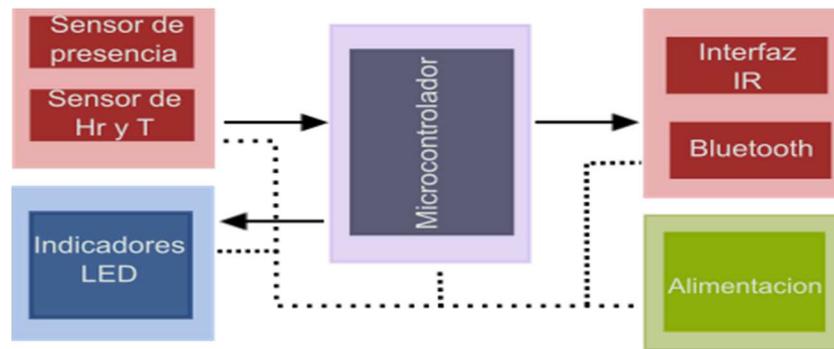


Figura 2-1. *Diseño funcional del controlador inteligente. Fuente: Autor.*

2.1.2 Sistema de medición.

Un sistema de medición puede estar conformado por diversos elementos, sensores, circuitos de acondicionamientos, etc. El sistema de medición se encarga proporcionar señales eléctricas al controlador para medición de las variables físicas requeridas, con las cuales controlador puede tomar sus decisiones. En este caso, las variables de interés son la humedad, la temperatura interior, y la presencia de personas. A continuación, se analizan las tecnologías disponibles para la medición de cada variable, se lleva a cabo la selección de la más adecuada, y se realiza el diseño de los circuitos de acondicionamiento, según sea necesario.

2.1.2.1 *Medición de Humedad y Temperatura interior.*

En el mercado actual es posible encontrar diferentes circuitos integrado IC para realizar la medición de la humedad y temperatura interior de una habitación. Los

hay individuales, duales, con salida analógica o digital. Sin embargo, nos centraremos en evaluar únicamente sensores duales y con salida digital, ya que esto permite realizar un diseño más sencillo, sin necesidad circuitos complejos de acondicionamiento, y además de que este tipo de sensores ya vienen calibrados. A continuación una lista de este tipo de sensores Tabla 2-2.

La selección se realizó utilizando el método cualitativo descrito en la sección 1.8, para lo cual se definieron los criterios mostrados en la Tabla 2-3.

Tabla 2-3

Criterios de selección.

Ítem	Criterios	Peso
1	Exactitud	0.10
2	Rango de operación	0.10
3	Tipo de salida	0.30
4	Tamaño	0.20
5	Costo	0.30
	Total	1.00

En el caso de la exactitud y rango de operación, son los criterios con menor peso de relevancia, con 0.1 para ser precisos. Esto debe a que los sensores evaluados son muy similares en exactitud y rango de operación. Al asignarles un menor peso a estos criterios, se espera que la selección recaiga sobre las diferencias de los sensores, y no sobre sus similitudes.

Por su parte, el tipo de salida y costo son los criterios con mayor peso a los anteriores, 0.3. El tipo de salida debe tomar en consideración en la selección del sensor, ya que algunos pueden estar basados en un protocolo estándar como SPI o I2C, o basados en un protocolo propio del fabricante, o también requerir al tipo acondicionamiento en su salida, por lo cual es un criterio delicado. En cambio, con respecto al costo, en este proyecto se cuenta con presupuesto limitado, por lo es necesario reducir costos.

Por ultimo tenemos el tamaño del sensor, el cual también es un tema delicado. Debe tomarse en cuenta los encapsulados de los sensores considerados, en algunos casos muy pequeños, lo incrementa la complejidad a la hora de creación

de la PCB. A este criterio se le asignó un peso de relevancia de 0.2, véase la Tabla 2-3.

Luego, con los criterios establecidos con sus respectivos pesos de relevancia, y continuando con el método cualitativo, se procedió a la construcción de la matriz de selección, véase la Tabla 2-4. En esta se evaluó cada sensor respecto con cada criterio.

Tabla 2-4
Matriz de selección del sensor de humedad y temperatura.

Sensores	Criterio	Exactitud	Rango de operación	Tipo de salida	Tamaño	Costo	Puntaje final
	Peso	0.1	0.1	0.3	0.2	0.3	
SHT11	Calificación	8	8	4	8	4	5.6
	Ponderación	0.8	0.8	1.2	1.6	1.2	
SI7005	Calificación	8	8	7	7	6	6.9
	Ponderación	0.8	0.8	2.1	1.4	1.8	
SI7022	Calificación	8	8	8	3	8	7
	Ponderación	0.8	0.8	2.4	0.6	2.4	
DHT11	Calificación	4	8	6	9	7	6.9
	Ponderación	0.4	0.8	1.8	1.8	2.1	
DHT22	Calificación	8	8	6	9	7	7.3
	Ponderación	0.8	0.8	1.8	1.8	2.1	

En el caso de la exactitud, debido a que la exactitud de estos sensores es apropiada para la aplicación, y ya que son muy similares entre ellos, se decidió asignar el mismo puntaje a todos los sensores. Así mismo en con respecto al rango de operación de los sensores. Las opciones consideradas cumplen con el rango definido en la sección 1.4, también se les asigno el mismo puntaje a todas las opciones.

En lo que se refiera al tipo salida, se les asignó un mayor puntaje a aquellos sensores que proyectan una menor complejidad para realizar medición desde un microcontrolador. La complejidad se encuentra en que, en algunos casos, se debe desarrollar un conjunto de funciones de código para obtener la medición. De manera que en la asignación del puntaje de acuerdo a criterios del desarrollador respecto a que tan complejo puede ser la implementación de esas funciones.

En tamaño y costo, el tamaño de algunos de los sensores considerados puede complicar su implementación en PCB, recibiendo un menor puntaje aquellos que son demasiado pequeños para ser soldados de manera habitual, y un mayor puntaje en caso contrario. En lo que se refiere a costo, se asignó un mayor puntaje a aquellos sensores con menor costo.

Como resultado final, se llegó a la conclusión que el DHT22 es el sensor más adecuado para el controlador inteligente, ya que obtuvo el mayor puntaje, 7.3, véase la Tabla 2-4.

El DHT22 utiliza un single-bus para la transmisión serial de los datos, es decir, nos es requerido de un periférico especial como una UART, como en caso de otros sensores, únicamente se requiere un pin digital. Además, este sensor envía los datos de acuerdo al protocolo mostrado en la Fig. 2-2, en el que se indica que el controlador inicia la comunicación a través de una señal inicial, y el sensor responde con una trama de bytes, cinco en total. Los primeros dos corresponde la humedad, los siguiente dos a la temperatura, y el ultimo corresponde a un byte de paridad. La humedad se calcula según la Ec. (2-1), mientras que la temperatura, según la Ec. (2-2).

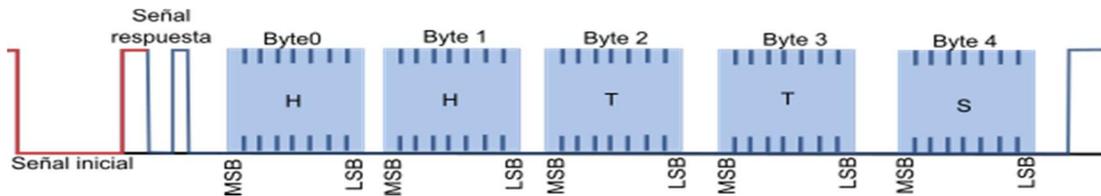


Figura 2-2. Protocolo de comunicación del DHT22. Fuente: Autor.

$$Hr\% = \frac{Byte1 + byte0 \ll 8}{10} \quad (2-1)$$

$$T^{\circ}C = \frac{Byte3 + byte2 \ll 8}{10} \quad (2-2)$$

Por otro lado, para realizar el montaje de conexión con el controlador, en la Fig. 2-3 se muestra el diagrama de conexión del DHT22, donde SDA, es la señal de datos hacia el controlador, y R es un valor sugerido por el fabricante.

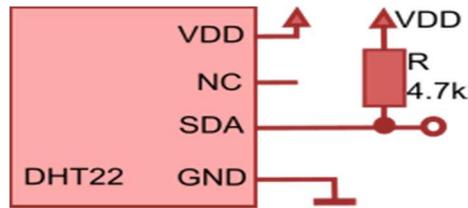


Figura 2-3. Diagrama de conexión del DHT22. Fuente: Autor.

2.1.2.2 Detección de presencia.

En el mercado existen sensores basados en diferentes tecnologías para detectar la presencia de personas en un espacio determinado, entre estas resaltan los siguientes tipos: los PIR, los de radar o de microondas, y los térmicos. La selección entre una tecnología y otra depende propiamente la aplicación en particular. Es necesario tomar en consideración la actividad que vaya a realizarse en el espacio que desea monitorear. Además debe tenerse en cuenta las características geométricas del sitio, donde se instalara el sensor. A continuación se realiza un análisis de las diferentes tecnologías disponibles para sensores de presencia.

Primeramente, los sensores PIR, este tipo de sensor es el más utilizado para detección de presencia, sin embargo, también es el menos efectivo. Se recomienda revisar la sección 1.5. Este sensor no es capaz de ver a través de obstáculos, tan poco responde a movimientos de baja velocidad, ni tampoco movimientos radiales. Debido a estas razones los sensores PIR están destinados para zonas de paso como escaleras.

Luego, están los sensores de tipo radar, que son una tecnología más avanzada que los de tipo PIR. Estos sensores son capaces de ver a través de esquinas y objetos, además de que pueden detectar movimientos ligeros del cuerpo humano. Estos detectores son utilizados en oficinas y salas de reuniones.

Por ultimo tenemos los sensores térmicos, este tipo sensor en simples palabras son cámaras térmicas de baja resolución. Esto significa que la detección de presencia se lleva cabo a partir del análisis de una imagen térmica de la habitación. Por lo cual, para la utilización de este tipo de sensor, se requiere de técnicas de procesamiento de imagen, y una considerable capacidad de cómputo.

Luego de realizar este análisis, se concluyó que las posibles tecnologías de detección de presencia para este proyecto son, los de tipo PIR y los de microondas. Los térmicos, en este caso, no fueron considerados por la cantidad procesamiento requerido en su utilización. Más tarde, se inició la elaboración de una lista de opciones, disponibles en el mercado, con sus respectivas características, para continuar con la selección mediante el método cualitativo, descrito en la sección 1.8. La lista elaborada es mostrada en la Tabla 2-5.

También se elaboró una lista de criterios de selección, de acuerdo con el método cualitativo, y además se les asignó sus correspondientes pesos de relevancia, véase la Tabla 2-6. La asignación de estos pesos se realizó siguiendo la lógica.

Tabla 2-5

Lista de sensores de detección presencia

Dispositivo	Tecnología	Alimentación	Tipo de salida	tamaño cm	Angulode visión	alcance	precio U\$
DT7450	Dual TEC	7.6-16Vdc	bool (24V)	11.9 x 7.1 x 4.2	90	15	69
DT6360STC	Dual TEC	10-14.5Vdc	bool (24V)	12.7 x 12.7 x 5.8	360	15.2	119
NJR4265RF	Microonda	3.3-5Vdc	bool (TTL)	1.4x2.0x0.8	70	10	18.75
DT7235	Dual TEC	7.5-16Vdc	bool (24V)	11.9 x 7.1 x 4.2	90	11	13.34

Tabla 2-6

Criterios de selección

Ítem	Criterio	Peso
1	Alimentación	0.2
2	Tipo de salida	0.2
3	Cobertura	0.3
4	Precio	0.3

Los criterios con mayor relevancia, en este caso, son la cobertura y precio, esto debido a limitaciones de presupuesto, y a las necesidades de cobertura, por eso decidió otorgarles un peso de 0.3 a ambos criterios. En cambio, los criterios de alimentación y tipo de salida, aunque son aspecto a tener en consideración, son en menor medida, por eso se les asigno un peso por igual de 0.2.

Continuando con el método de selección cualitativo, se elaboró la matriz de selección en la que se evalúa cada opción, respecto a cada criterio, y se obtiene un puntaje final como resultado. La matriz de selección es mostrada en la Tabla 2-7, y el proceso de calificación se describe a continuación.

En caso de costo, se asignó un mayor puntaje a aquellas opciones con el menor costo, y menor en caso contrario. En cambio, se les dio una mayor calificación a los sensores con mayor cobertura, que a aquellos con menor alcance.

Por otro lado, los sensores considerados requieren algún tipo de acondicionamiento en su salida, de manera que, en dependencia de la complejidad del circuito de acondicionamiento, se asignó puntaje mayor o menor al tipo de salida. Mientras más complejo sea el circuito de acondicionamiento, menor el puntaje asignado.

El criterio de alimentación fue calificado de la siguiente manera. Dentro las opciones consideradas, algunos de los sensores operan a 24V, consecuentemente esto implica un aumento en la complejidad de la fuente de alimentación de todo el dispositivo en desarrollo, ya que se requerirían dos voltajes de alimentación, 24V para sensores, y 5V para el microcontrolador. Así que los sensores a 24V obtuvieron un menor puntaje, respecto a aquellos que operan a 5V. Como resultado final, se llegó a la conclusión que el NJR4265RF es el sensor más adecuado para esta aplicación, ya que obtuvo el mayor puntaje de 7.6, véase la Tabla 2-7.

Tabla 2-7
Matriz de selección para el sensor de presencia

Sensores	Criterio	Alimentación	Tipo de salida	Cobertura	Costo	Puntaje final
		0.2	0.2	0.3	0.3	
DT7450	Calificación	5	6	8	4	5.8
	Ponderación	1	1.2	2.4	1.2	
DT360STC	Calificación	5	6	9	3	5.8
	Ponderación	1	1.2	2.7	0.9	
NJR4265RF	Calificación	9	5	7	9	7.6
	Ponderación	1.8	1	2.1	2.7	
DT7235	Calificación	6	6	7	9	7.2
	Ponderación	1.2	1.2	2.1	2.7	

El NJR4265 posee dos salidas digitales. Cuando el sensor detecta un movimiento, estas salidas se activan produciendo un pulso, como el mostrado en la Fig. 2-4. Sin embargo, la durabilidad de estos pulsos es muy corta, por lo que se decidió diseñar un circuito para ampliar su durabilidad. La salida de este circuito indicara si en los últimos 10s se ha detectado un movimiento

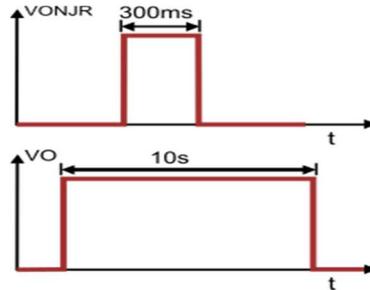


Figura 2-4. Pulsos de entrada y salida del circuito detección de presencia. Fuente: Autor.

El circuito utilizado es mostrado en la Fig. 2-5. En esta figura se observan las conexiones del NJR4265, donde, POT1 es un potenciómetro que permite ajustar el umbral de detección. Por otro lado, la durabilidad del pulso es modificada gracias al temporizador 555 mono-astable, el cual es activado por señales de salida del NJR4265 a través de la compuerta NOR, implementada con transistores, véase nuevamente la Fig. 2-5.

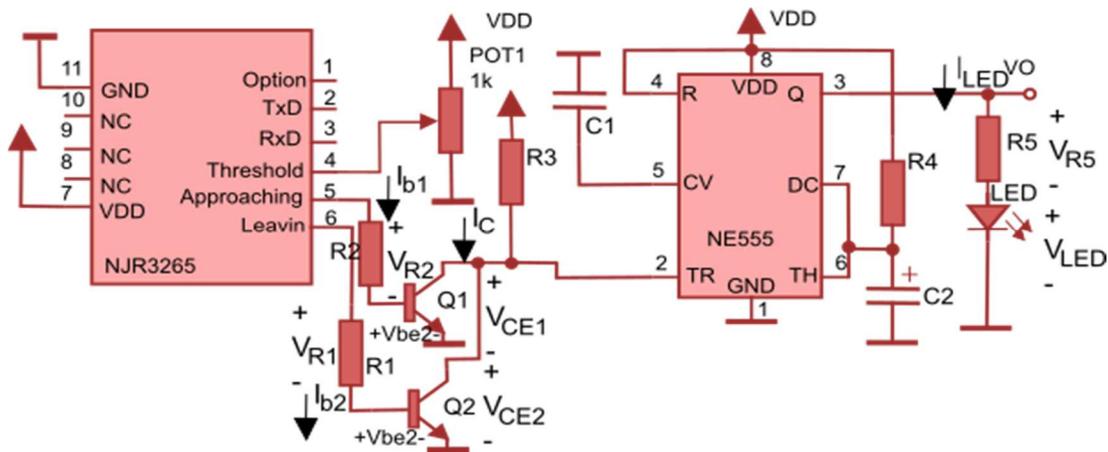


Figura 2-5. Diagrama esquemático para el circuito detector de presencia. Fuente: Autor.

Las resistencias R1, R2 y R3 y los transistores Q1 y Q2, representan una compuerta NOR. Tanto Q1 como Q2 son de propósito general, para ese caso se

seleccionó el modelo 2n3904. Los parámetros de saturación para el 2n3904 son mostrados en la Tabla 2-8, mientras que R1 y R2 se calculan según la Ec. (2-3). Mientras que R3, siendo una resistencia Pull-Up, suele dársele un valor de 10kΩ.

$$R1 = \frac{V_{DD} - V_{be}}{I_C} \cdot \beta \quad (2-3)$$

Resolviendo para Ec. (2-3).

$$R1 = \frac{5V - 0.68V}{10mA} \cdot 10 \approx 4.7k\Omega \quad (2-4)$$

$$R1 = R2 \approx 4.7k\Omega \quad (2-5)$$

Tabla 2-8
Parámetros de saturación para el 2n3904

Característica	Símbolo	Valor requerido	Unid.
Corriente de colector	I _C	10	mA
Voltaje colector-emisor	V _{CE}	0.3	V
Ganancia de corriente	β	10	-
Voltaje base-emisor	V _{BE}	0.68	V

Por otro lado, la duración del pulso de salida, está determinada por la resistencia R4 y el capacitor C2, como se indica en la Ec. (2-6). Así que teniendo en cuenta un tiempo estimado de 10s, y debido a su disponibilidad en el mercado local, se utilizó una resistencia de 100kΩ, para R4, y un capacitor de 100μF, para C4. Por lo cual, el tiempo de la temporización queda definido según la Ec. (2-7).

$$t_{on} = 1.1 \cdot R_4 C_2 \quad (2-6)$$

$$t_{on} = 1.1 \cdot 100\mu F \cdot 100k\Omega = 11s \quad (2-7)$$

Para finalizar con este análisis, tenemos que R5 es utilizada para polarización del LED. El voltaje típico en el LED, V_{LED}, es de 1.7V, y la corriente requerida, I_{LED}, es 10mA, de manera que utilizando la ley Ohm, R5 se calcula según la Ec. (2-8), véase nuevamente la Fig. 2-5.

$$R_5 = \frac{V_{DD} - V_{LED}}{I_{LED}} \quad (2-8)$$

Resolviendo para Ec. (2-8).

$$R_5 = \frac{5V - 1.7V}{10mA} = 330\Omega \quad (2-9)$$

2.1.3 Interfaces de comunicación.

2.1.3.1 Interfaz infrarroja IR.

Esta interface le permite al controlador enviar instrucciones al AA para su funcionamiento, encenderlo, apagarlo, configurar la velocidad del fan, etc. Dichas instrucciones son dictadas de acuerdo al protocolo de comunicación del fabricante del AA, véase la sección 1.3. Sin embargo, en esta sección nos enfocaremos en el diseño de esta interfaz.

Las instrucciones son enviadas mediante una trama de bytes que contiene la información para la configuración del AA, donde, cada bit de esta trama es codificado, modulando una señal portadora 36kHz, como se muestra en la Fig. 2-6. Las señales datos y portadora, son generadas por un microcontrolador, mientras que la señal modulación es el resultado de mezclar ambas, cabe recordar que la señal resultante es una señal de luz infrarroja.

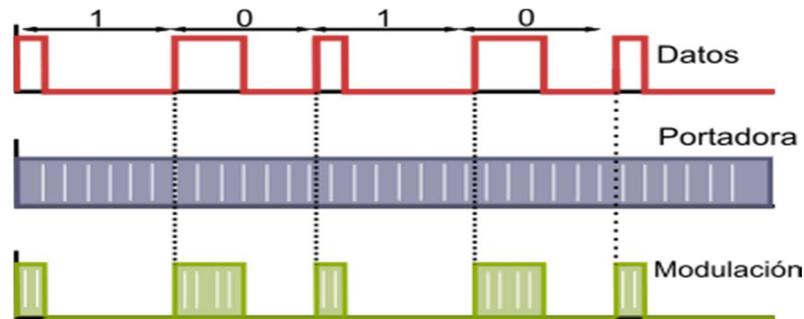


Figura 2-6. Diseño esquemático de la interfaz IR. Fuente: Autor.

El circuito para generar la señal infrarroja “modulación”, mostrado en la Fig. 2-7, utiliza una compuerta AND para mezclar las señales de datos y portadora, y producir una única señal. Esta compuerta está implementada con transistores, lo que permite el control del encendido y apagado del LED-IR, y así producir la señal infrarroja “modulación”. A continuación se realiza un análisis de este circuito.

Primeramente, este circuito representa un driver de control para un LED-IR, por lo que es necesario tener en consideración las características de polarización de un LED-IR, ya que la corriente a través del LED-IR, influye directamente en el alcance de la señal emitida, pero también debe tomarse en consideración el voltaje

de encendido del LED-IR. Las características eléctricas del LED-IR, son mostradas en la Tabla 2-9.

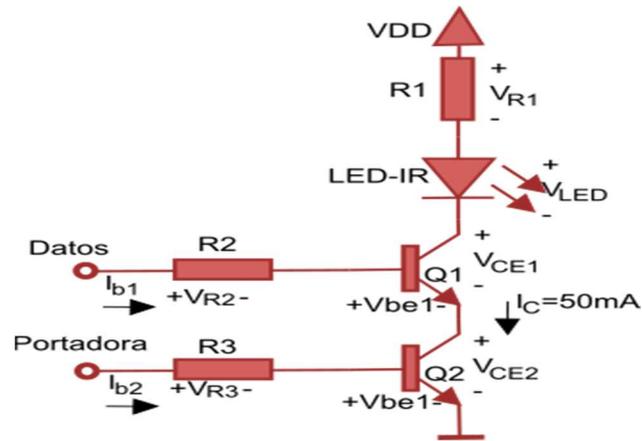


Figura 2-7. Diseño esquemático de la interfaz IR.
Fuente: Autor

Tabla 2-9
Características eléctricas para un LED-IR

Característica	Símbolo	Valor	Unid.
Voltaje de polarización den directa	V_{LED}	1.4	V
Corriente de encendido	I_D	50	mA

Continuando con el diseño de la interfaz IR, Q1 y Q2 son transistores de propósito general. Para este circuito se decidió utilizar el transistor de propósito de general 2n3904, debido a que cumple con las características eléctricas requeridas para este diseño, véase la Tabla 2-10, además de que es fácil de encontrarlo en el mercado local.

Tabla 2-10
Características requeridas de los transistores Q1 y Q2

Característica	Símbolo	Valor requerido	Valor Max.	Unid.
Corriente de colector	I_c	50	200	mA
Frecuencia de conmutación	f_c	38	3,100	kHz

La ecuaciones que gobiernan la interfaz IR mostrada en la Fig. 2-7, son mostradas a continuación.

$$VDD = I_{b1}R_2 + V_{be} \quad (2-10)$$

$$I_{b1} = \frac{I_C}{\beta} \quad (2-11)$$

$$V_{DD} = I_C R_1 + V_{LED} + V_{CE1} + V_{CE2} \quad (2-12)$$

$$R_2 = R_3 \quad (2-13)$$

Despejando.

$$R_2 = \frac{V_{DD} - V_{be}}{I_C} \cdot \beta \quad (2-14)$$

$$R_1 = \frac{V_{DD} - V_{LED} - V_{CE1} - V_{CE2}}{I_C} \quad (2-15)$$

Para resolver a Ec. (2-14) y a Ec (2-15), deben considerarse las características de saturación del 2n3904 para una corriente $I_C=50mA$, estas se muestran en la Tabla 2-11. La solución a estas ecuaciones (2-14) y (2-15), se indican en la Ec. (2-16) y Ec. (2-18), respectivamente.

Tabla 2-11
Característica de saturación para el 2n3904, $I_C=10mA$

Característica	Símbolo	Valor	Unid.
Voltaje de saturación entre colector y emisor. $I_C=50mA$	V_{CE}	0.2	V
Voltaje de saturación entre base y emisor. $I_C=50mA$	V_{BE}	0.7	V
Ganancia de corriente en saturación. $I_C=50mA$	β	10	-

$$R_2 = \frac{5.0V - 0.7V}{50mA} \cdot 10 = 860\Omega \quad (2-16)$$

$$R_2 = R_3 = 860\Omega \quad (2-17)$$

$$R_1 = \frac{5V - 1.4V - 0.2V - 0.2V}{50mA} = 64\Omega \quad (2-18)$$

2.1.3.2 Interfaz bluetooth.

Esta interfaz permite el intercambio de datos entre el controlador inteligente y un Smartphone utilizando Bluetooth. Para esta aplicación se utilizó un HC-05, el cual es un módulo Bluetooth SPP (Serial Port Protocol), diseñado para transferir datos serial a través de una conexión inalámbrica. Este módulo posee dos modos de operación, uno como esclavo y uno como maestro. Un dispositivo maestro siempre inicia la conexión, en cambio, uno esclavo espera a un dispositivo maestro para inicie la conexión. Generalmente, se utiliza el HC05 en modo esclavo

cuando se necesita comunicarse con una PC o celular, ya que estos se comportan como dispositivos maestros.

Los datos enviados por Smartphone, y recibidos por el HC05, son retransmitidos por el HC05 hacia el controlador mediante el protocolo RS232, y viceversa. De esta manera, el Smartphone y controlador intercambian mensajes, como se ilustra en la Fig. 2-9. El diagrama de esquemático de conexión del HC05 se muestra en la Fig. 2-8, donde TXO y RXO son las líneas de datos hacia el controlador.

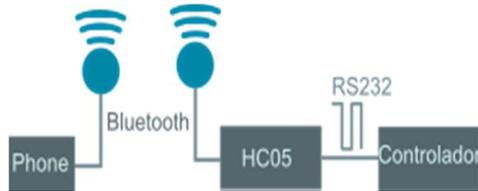


Figura 2-9. Comunicación Smartphone-Controlador. Fuente: Autor.

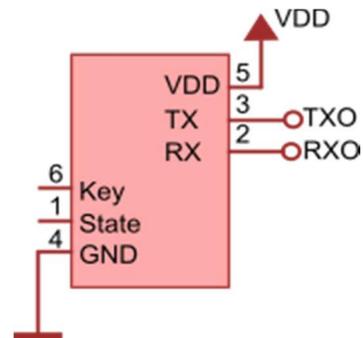


Figura 2-8. Diagrama esquemático del HC05. Fuente: Autor.

2.1.4 Indicadores LEDs.

Los indicadores LEDs, o sistema de señalización, son periféricos que proporcionan información al usuario acerca de la operatividad del dispositivo. En este caso, indican cuando se establece una conexión con un dispositivo Bluetooth, y también muestran información sobre la temporización de la climatización. El circuito de los indicadores LEDs es mostrado en la Fig. 2-10, donde señal-Bluetooth y señal-temporización, son las señales generadas por el controlador para controlar los LEDs. A continuación se realiza un análisis de este circuito.

Primeramente, el circuito mostrado en la Fig. 2-10 cuenta dos mallas eléctricas simétricas, una para el LED1 y otra para el LED2, y son simétricas ya que R1 y R2 poseen el valor de resistencia. Además, los LEDs utilizados para este diseño son de color verde, por lo que se debe considerar las características de polarización para este tipo LEDs en la determinación de los valores de R1 y R2, véase la Tabla 2-12.

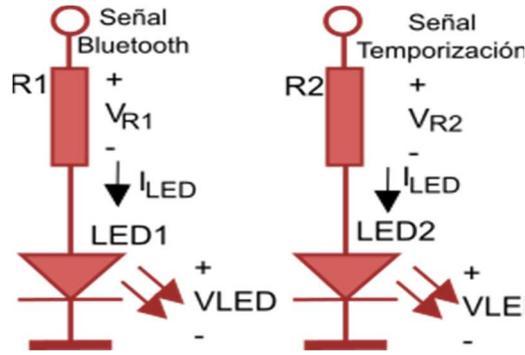


Figura 2-10. Circuito de indicadores LEDs.
Fuente: Autor.

Tabla 2-12
Características eléctricas para un LED-VERDE

Característica	Símbolo	Valor	Unid.
Voltaje de polarización den directa	V_{LED}	1.7	V
Corriente de encendido	I_{LED}	10	mA

Realizando un análisis en una las mallas, y simplificando los cálculos, R1 y R2 pueden determinarse según la Ec. (2-19).

$$R1 = R2 = \frac{V_{DD} - V_{LED}}{I_{LED}} \quad (2-19)$$

Resolviendo.

$$R1 = R2 = \frac{5.0V - 1.7V}{10mA} = 330\Omega \quad (2-20)$$

2.1.5 Sistema de control

2.1.5.1 Microcontrolador.

El microcontrolador constituye el cerebro dentro sistema, este es capaz de obtener información del entorno, procesarla, y tomar decisiones basándose en su programación. Mientras tanto, los sistemas de medición, de comunicación, y de señalización, son interfaces entre el controlador y el medio. Sin embargo, el microcontrolador debe contar con ciertos periféricos internos para utilizar las interfaces externas, En la Tabla 2-13 se muestran los periféricos internos requeridos para el microcontrolador.

Tabla 2-13

Periféricos de Entrada y salida del microcontrolador.

ítem	Tipo	I/O	Descripción
1	Bool	I	Sensor de presencia.
2	Bool	I	Sensor de temperatura y Humedad.
3	Bool	O	Señal de datos IR.
4	PWM	O	Señal portadora dato IR.
5	Bool	O	LED-verde, Indicador de temporización.
6	Bool	O	LED-verde, indicador Bluetooth.
7	UART	O	RX.
8	UART	I	TX.

Según esta tabla, para el controlador inteligente se requiere que el microcontrolador posea 5 I/O digitales, un módulo PWM y uno UART. Sin embargo, también deben considerarse los requerimientos de memoria para el dispositivo. En la Tabla 2-14 se describen los requerimientos de memoria por función de programa, basándose en trabajos anteriores realizados por el desarrollador.

Tabla 2-14

Requerimientos de memoria por función.

Función	ROM (bytes)	RAM (bytes)
Fuzzy	3,872	180
Protocolo IR	390	12
Main	1,494	10
DHT22	464	11
Total	6,220	213

Aunque se desconoce la memoria exacta requerida, esta estimación es un punto de partida para encontrar un microcontrolador adecuado. Sin embargo, deben tomarse precauciones respecto a la memoria del microcontrolador, es mejor que sobre a que haga falta, pero en una medida razonable. Por lo tanto, se recomienda que la memoria estimada, únicamente represente el 40% de la memoria total microcontrolador seleccionado, lo que permite determinar la memoria total, M_T , según la Ec. (2-21). El resultado de la estimación es mostrado en la Tabla 2-15.

$$M_T = \frac{1}{0.4} M_E \quad (2-21)$$

Tabla 2-15
Memoria total requerida

ROM (bytes)	RAM (bytes)
15,550	532

Otros aspectos considerados en la selección del microcontrolador para el controlador inteligente, fue que existiera un modelo en CCS y PROTEUS, y que fuera soportado por el PICkit2, ya que estas son las herramientas utilizadas para el desarrollo y evaluación del programa. Como resultado, se elaboró una lista de opciones de microcontroladores, la que muestra en la Tabla 2-16, y a continuación se realiza la selección microcontrolador, mediante el método cualitativo.

Tabla 2-16
Lista de opciones para microcontrolador

Dispositivo	PROTEUS	CSS	PICKIT	CCP/PWM	7 > IO	UART	Nº pines	ROM (bytes)	RAM (bytes)	Precio
16f1455	SI	SI	SI	SI	SI	SI	14	16k	367	\$1.72
16f1509	SI	SI	SI	SI	SI	SI	20	16k	512	\$1.66
18f14k22	SI	SI	SI	SI	SI	SI	20	16k	512	\$2.27
18f14k50	SI	SI	SI	SI	SI	SI	20	16k	768	\$2.30

Primeramente, se definieron los criterios para evaluar cada opción y seleccionar la más adecuada. Los criterios considerados son el número de pines, memoria, y precio. La asignación de pesos de relevancia se realizó de la siguiente manera.

En lo que se refiere a número de pines y precio, las opciones consideradas, son bastante similares, por lo tanto, se consideró que estos son los criterios menos relevantes, y se les asignó un peso de 0.2 a ambos. Por otro lado, debido a su importancia a los parámetros de memoria se les asignó un mayor peso de relevancia, 0.3, para ambos. La Tabla 2-17 muestra los criterios con sus respectivos pesos.

Tabla 2-17
Criterios de selección para controlador

Ítem	Criterios	Peso
1	Numero de pines	0.2
2	ROM	0.3
3	RAM	0.3
4	Precio	0.2

Continuando con el método cualitativo, se elaboró una matriz de selección, presentada en la Tabla 2-18, en la que se evaluó cada opción respecto a cada criterio, calificando cada criterio de la siguiente manera.

Tabla 2-18
Matriz de selección de controlador

Dispositivo	Criterio	Numero de pines	ROM	RAM	Costo	Puntaje final
	Peso	0.2	0.3	0.25	0.2	
16f1455	Calificación	8	9	6	7	7.20
	Ponderación	1.6	2.7	1.5	1.4	
16f1509	Calificación	7	9	7	7	7.25
	Ponderación	1.4	2.7	1.75	1.4	
18f14k22	Calificación	7	9	7	7	7.25
	Ponderación	1.4	2.7	1.75	1.4	
18f14k50	Calificación	7	9	8	6	7.3
	Ponderación	1.4	2.7	2	1.2	

Al número de pines se le asignó una mejor calificación a los dispositivos con menor número de pines. Por otro lado, al costo se dio un mayor puntaje a las opciones con menor costo. En lo que se refiere a memoria, los dispositivos que se adaptan mejor a los requerimientos establecidos obtuvieron una mayor calificación. Como resultado final, el dispositivo con el mayor puntaje es el 18f14k50, y según método es el dispositivo más adecuado para esta aplicación.

2.1.5.2 Modelos de programa.

El modelo programa describe el comportamiento del controlador. En este caso, debido a su complejidad, el programa es descrito mediante dos modelos, una para

la rutina principal (main), y otra para rutina de interrupción de recepción de datos, INT_RDA, utilizada para comunicación con el dispositivo Bluetooth.

La rutina principal se encuentra constituida por cuatro estados: Evaluando, Climatizando, No-climatizando, y stop. El modelo para la rutina principal es mostrado en la Fig. 2-11, y se describen a continuación.

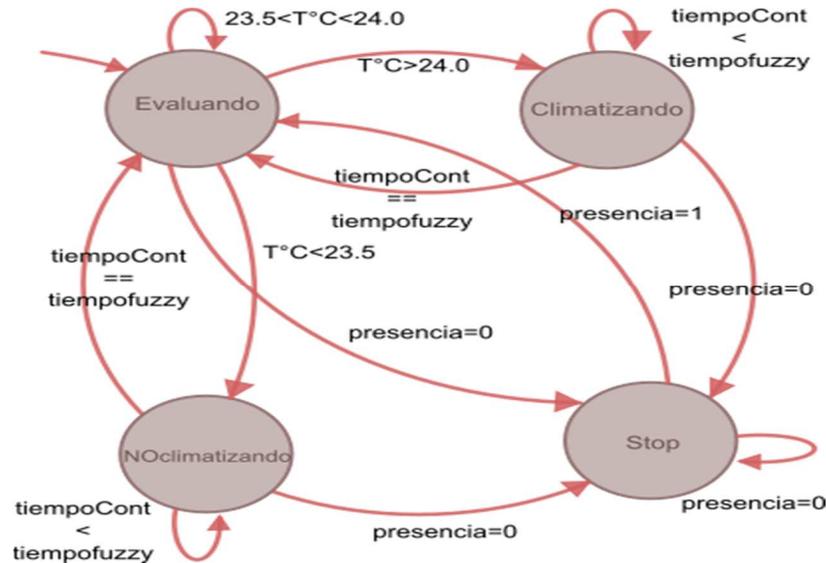


Figura 2-11. Modelo de programa (main). Fuente: Autor.

-Evaluando- Durante este estado el controlador monitorea la humedad y la temperatura. Además determina si es necesario o no la climatización de la habitación, para esto el controlador posee un intervalo de decisión, si la temperatura es mayor a los 24°C, entonces es necesario climatizar, en cambio, si es inferior a los 23.5°C, no es necesario la climatización.

Luego, en el caso que se necesite climatizar, en este estado se determina el tiempo de dicha climatización, basándose en el algoritmo Fuzzy desarrollado, el cual se describe en la siguiente sección. Sin embargo, si no es requerida una climatización, también se determina el tiempo de espera para volver a climatizar la habitación.

-Climatizando- En este estado, se realiza activación del AA, y la temporización del tiempo de climatización, determinado según el algoritmo Fuzzy.

-Esperando- Durante este estado, se lleva cabo la desactivación del AA, y la temporización del tiempo espera, determinado según el algoritmo Fuzzy.

-Stop- Sin importar el estado en el que se encuentre, si la presencia es cero, el controlador pone en espera la climatización hasta la detección nuevamente de presencia. Véase, nuevamente la Fig. 2-11.

Por otro lado, en lo respecta a la rutina de interrupción RDA, ella atiende a una lista de solicitudes enviadas por usuario, mediante el dispositivo Bluetooth. Esta rutina es ejecutada una vez que el dispositivo recibe un nuevo dato a través de su módulo UART. La instrucción o comando, es un número hexadecimal que identifica un grupo de tareas dentro la rutina. Una vez que se recibe una nueva instrucción, se identifica y se ejecutan las tareas correspondientes. La lista de instrucción es mostrada en la Tabla 2-19 con su respectivo valor. Mientras que el modelo para esta rutina es muestra en la Fig. 2-12.

Tabla 2-19
Instrucciones de la rutina de interrupción RDA

Comando o instrucción	Valor
UpdateTemp	0x03
UpdateHr	0x07
ButtonON	0x0B
ButtonOFF	0x0D

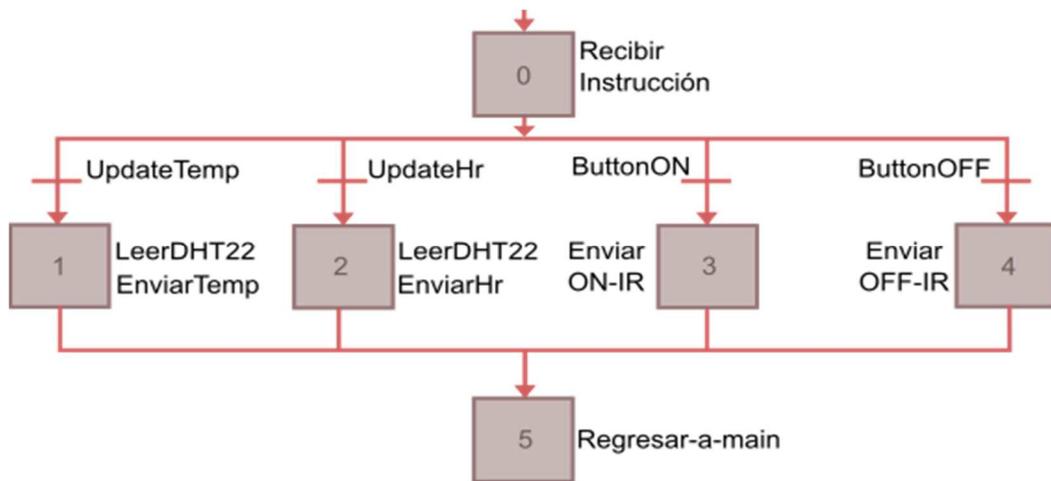


Figura 2-12. Modelo de la rutina de interrupción RDA del controlador. Fuente: Autor.

-UpdateTemp- Esta instrucción es utilizada cuando el usuario quiere actualizar la temperatura mostrada en la APP. Una vez recibida se realiza una medición con el

DHT22, y se envía hacia el dispositivo Bluetooth el valor correspondiente de la medición de la temperatura.

-UpdateHr- Esta instrucción se utiliza cuando el usuario quiere actualizar la humedad mostrada en la APP. Una vez recibida se realiza una medición con el DHT22, y se envía hacia el dispositivo Bluetooth el valor correspondiente de la medición de la humedad.

-ButtonON- Cuando se recibe esta instrucción, se envía el comando-IR de Encender, hacia el aire acondicionado.

-ButtonOFF- se envía el comando-IR de Apagar, hacia el aire acondicionado.

2.1.5.3 Algoritmo fuzzy.

En esta sección se realizará una descripción del sistema fuzzy desarrollado, el cual permite determinar los tiempos de encendido y apagado de un AA para el control de la climatización en una habitación, basándose en humedad y temperatura interior. Primeramente, se presentan las variables de entrada y salida del sistema fuzzy, luego se determinan y definen los universos de discursos para cada variable, y se procede a establecer las funciones de pertenencia con sus respectivas etiquetas lingüísticas, y luego se concluirá con la base reglas desarrolladas.

Cabe señalar que este sistema está basado en la arquitectura Mamdani, la cual consiste principalmente de cuatro elementos: fuzzificadores, mecanismo de inferencia, base de reglas fuzzy, y defuzzificadores. Se recomienda revisar la sección 1.6.

En el control de la climatización de una habitación deben tomarse en consideración la humedad y la temperatura, ya que estas están relacionadas con el confort térmico. La utilización de un algoritmo fuzzy para el control de los tiempos de encendido y apagado de un AA, tiene como finalidad conseguir un ahorro energético, pero manteniendo siempre las condiciones de confort térmico necesario. Por lo tanto, las variables de entrada para el controlador fuzzy son la humedad y la temperatura. Por otro lado, como se ha dicho antes, el controlador se encargará de los tiempos de encendido y apagado del AA, por lo que estas son las

variables de salida del controlador. La Fig. 2-13, ilustra al controlador fuzzy con sus entradas y salidas.

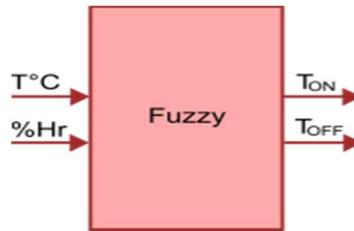


Figura 2-13. Controlador fuzzy.
Fuente: Autor.

Una vez definidas las variables entra y salida del controlador fuzzy, también es necesario definir su rango dinámico, es decir, el universo de discurso para cada variable. En el caso de la temperatura, el universo de discurso queda establecido, por el rango de temperatura en un AA, es decir, de 16°C a 30°C. Mientras tanto, en el caso de la humedad relativa, esta se expresa en términos de porcentaje, por lo tanto, su rango dinámico inicia de 0 hasta 100%. Los universos de discurso para la temperatura y la humedad, son mostrados en la Fig. 2-14 y en la Fig. 2-15, respectivamente.



Figura 2-14. Universo de discurso de la temperatura. Fuente: Autor.

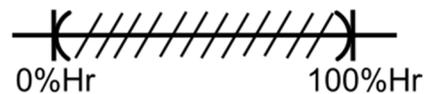


Figura 2-15. Universo de discurso de la humedad relativa. Fuente: Autor.

En lo que se refiere a los tiempos de encendido, T_{ON} , y apagado, T_{OFF} , la definición de sus respectivos universos resultó más compleja, pues fue necesario realizar la observación de la climatización habitual realizada para AA, y estimar los valores de T_{ONMAX} y T_{OFFMAX} . Esto se llevó a cabo durante un par de días, en los que se configuró el AA a una temperatura mínima de 16°C, y luego una vez establecida, se reconfiguró el AA a una temperatura máxima (30°C). Con cada observación se tomó nota del tiempo que demoraba la temperatura en pasar de un extremo a otro. Un ejemplo de las observaciones realizadas es mostrado en la Fig. 2-16. Como resultado de estas observaciones se elaboró la Tabla 2-20 y Tabla 2-21 que contienen información del tiempo requerido para calentar y enfriar la habitación.

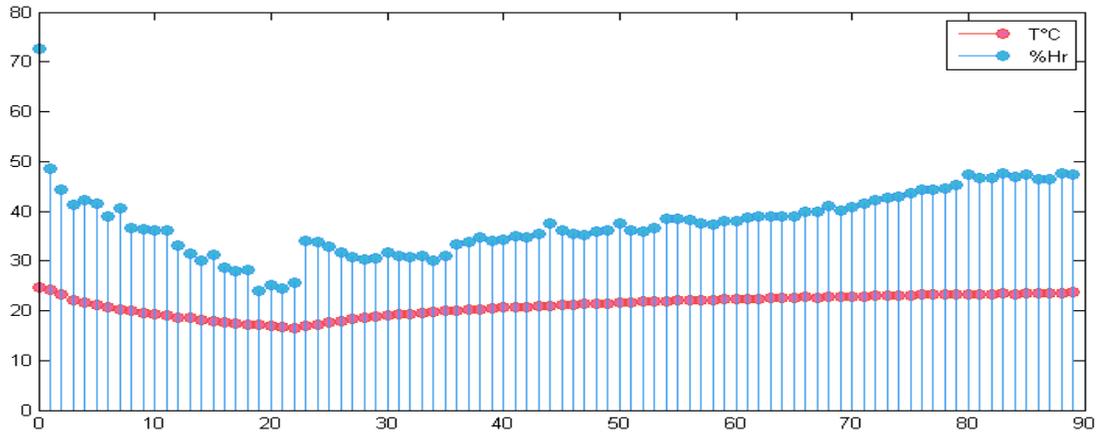


Figura 2-16. Monitoreo de Humedad y temperatura para la estimación de Ton y Toff. Fuente: Autor.

Tabla 2-20

Tiempo de calentamiento.

Exp.	T _O °C	T _F °C	T _E (min)
1	22.80	25.39	112.00
2	17.10	24.10	82.00
3	20.89	24.39	64.00
4	17.89	23.50	53.00
5	18.70	24.50	78.00

Tabla 2-21

Tiempo de enfriamiento.

Exp.	T _O °C	T _F °C	T _E (min)
1	25.20	17.00	50.00
2	24.80	17.10	34.00
3	24.39	17.89	27.00
4	23.50	18.70	22.00
5	24.50	17.39	30.00

Con la información proporcionada por estas tablas se estableció el universo de discurso para el tiempo de encendido, T_{ON}, y el tiempo de apagado, T_{OFF}. Estos son mostrados en la Fig. 2-17 y Fig. 2-18, respectivamente.

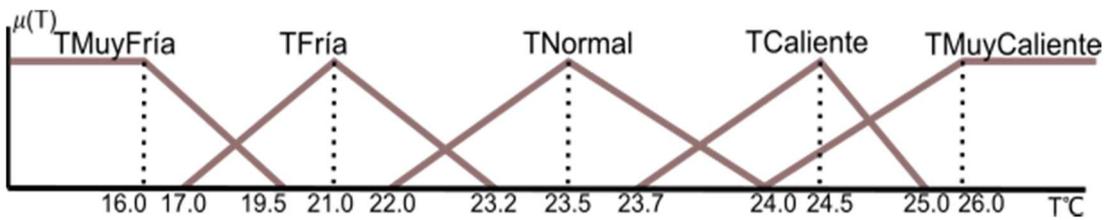


Figura 2-17. Conjuntos fuzzy de la Temperatura. Fuente: Autor.

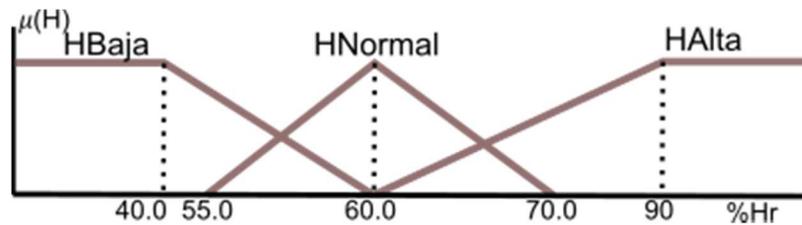


Figura 2-18. Conjuntos fuzzy de la Humedad relativa. Fuente: Autor.

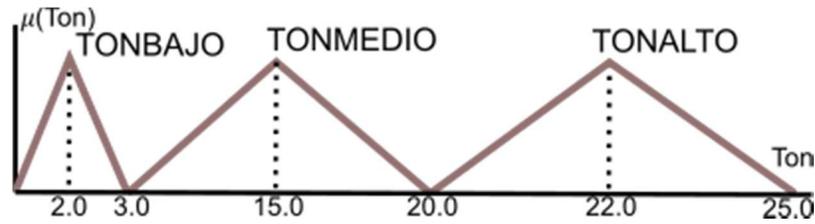


Figura 2-19. Conjuntos fuzzy del tiempo de encendido. Fuente: Autor.

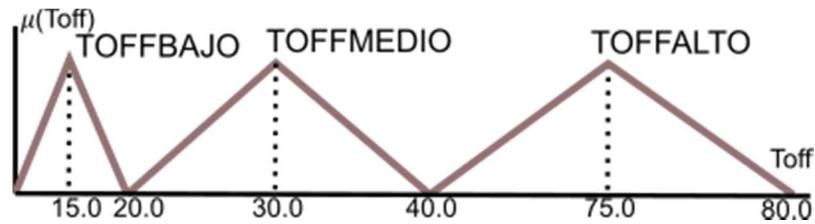


Figura 2-20. Conjuntos fuzzy del tiempo de apagado. Fuente: Autor.

Para finalizar con el diseño, se presenta la base reglas fuzzy desarrollada. Estas reglas correlacionan los conjuntos fuzzy de entrada y con los de salida, y consecuentemente también definen la repuesta del controlador fuzzy. En este caso, la base reglas se expresa mediante un mapeo de los conjuntos de entrada y salida en las figuras 2-21 y 2-22.

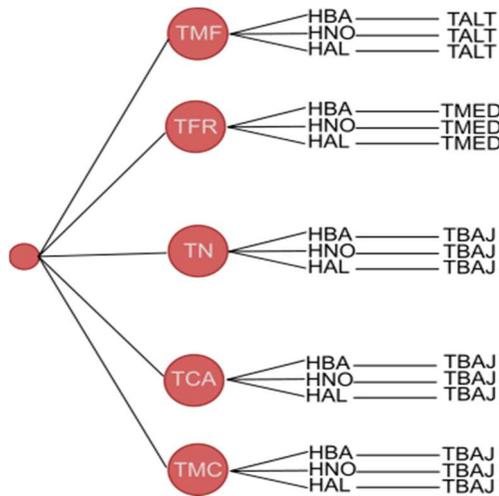


Figura 2-22. Mapeo de reglas tiempo de encendido. Fuente: Autor.

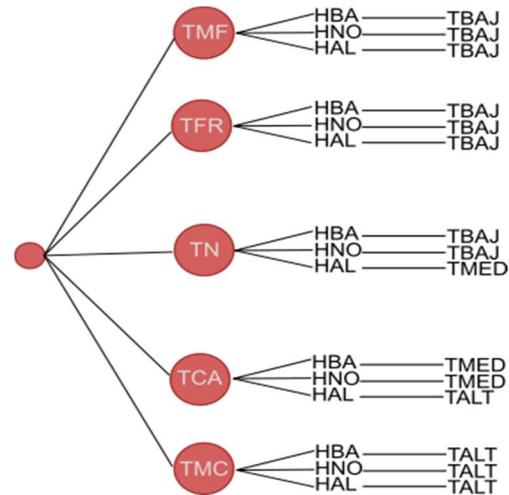


Figura 2-21. Mapeo de reglas tiempo de apagado. Fuente: Autor.

En estas figuras, los términos TMF, TFR, TN, TCA, TMC, son abreviaciones de las etiquetas lingüísticas TMuyFria, TFria, TNormal, TCaliente y TMuyCaliente, respectivamente. Similarmente, los términos HBA, HNO y HAL, representan a las

etiquetas HBAJA, HNORMAL, HALTA. Mientras que TBAJ, TMED, TALT, son utilizadas para representar a las etiquetas TBAJO, TMEDIO, TALTO.

Cada rama del árbol en las figuras de arriba, representa una regla del sistema fuzzy. Cada una de estas, tiene la estructura **SI (antecedente) ENTONCES (consecuente)**, donde antecedente y consecuente son también conjuntos fuzzy, ya sea puros o resultado de operar con ellos. A continuación, se presenta un ejemplo de una regla fuzzy.

SI TMuyFria AND HBaja ENTONCES TonBajo

Para este ejemplo, el antecedente se obtiene como resultado de la operación AND entre los conjuntos fuzzy TMuyFria y HBaja. Mientras que el consecuente lo constituye el conjunto fuzzy TonBajo.

Por otro lado, en lo que respecta al método de defuzzificación, se empleó el método de centro de masa, para lo cual se recomienda al lector ver la sección 1.7.

2.1.6 Fuente alimentación

La fuente de alimentación es el elemento dentro del sistema que se encarga suministrar lo niveles de voltaje y corriente requeridos por todo el sistema. En la Tabla 2-22, se realiza una descripción de los requerimientos de corriente y de voltaje por cada elemento. Según la Tabla, se requiere de una corriente máxima de 176mA y un voltaje de 5V.

Debido a su bajo costo, y simplicidad de diseño, se decidió utilizar una adaptador 120VAC-DC5V estándar como fuente de alimentación. Este adaptador entrega 5V y hasta 600mA, lo cual es más que suficiente para esta aplicación. Este adaptador es mostrado en la Fig. 2-23.

Tabla 2-22
Niveles de corriente y voltaje

Elemento	Voltaje	Corriente
Microcontrolador	5V	4mA
LEDs (señalización)	5V	20mA
LED-IR	5V	50mA
DHT22	5V	2mA
HC05	5V	40mA
NJR3265	5V	60mA
Total		176mA



Figura 2-23. Adaptador 120AC-5VDC.
Fuente: Autor.

2.1.7 Diseño en PCB

Una vez finalizado el diseño de cada uno de los circuitos requeridos en el controlador inteligente, se llevó a cabo la integración de estos en una PCB. Para este sistema fueron utilizadas dos PCB, una que contiene el sistema de control (18F14k50), el de medición (DHT22), el de señalización, la alimentación y el modulo Bluetooth, y otra PCB que se diseñó únicamente para circuito de detección de presencia, con el objetivo de poder ajustar el ángulo de enfoque del NJR3265.

En la Fig. 2-24 se muestra el diagrama esquemático de PCB1, mientras que en la Fig. 2-25 y en la Fig. 2-26, se muestran el Top view y Botton view de PCB1, respectivamente. Por otro lado, en la Fig. 2-27 se muestra el diagrama esquemático de PCB2, y en la Fig. 2-28 y en la Fig. 2-29, se muestran el Top view y el Botton view de PCB2.

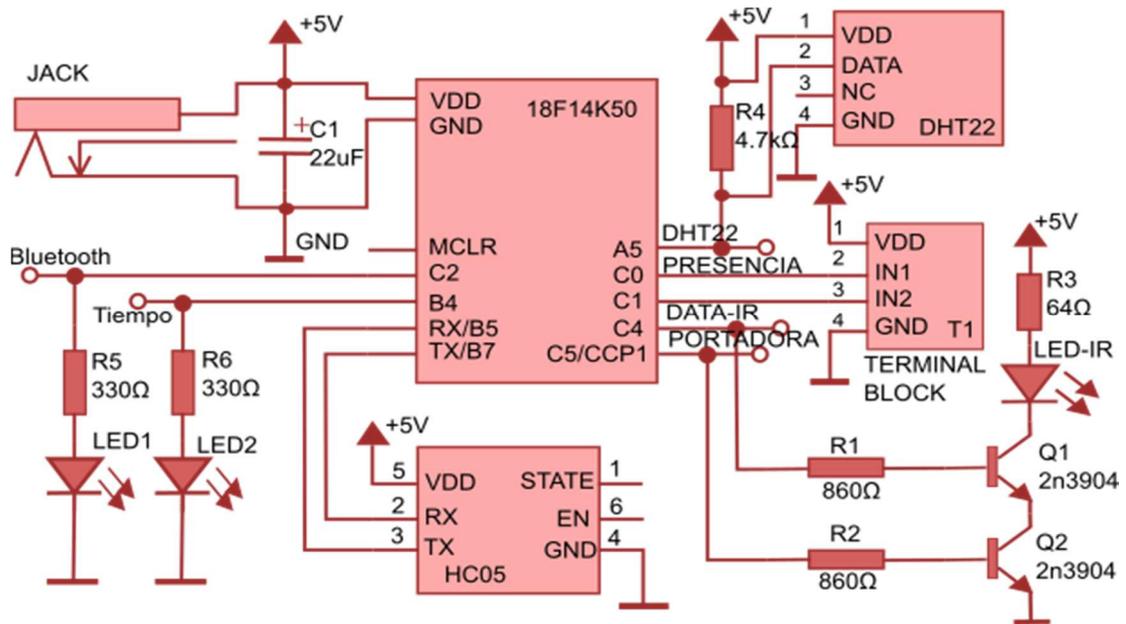


Figura 2-24. Diagrama esquemático del controlador inteligente (PCB1). Fuente: Autor.

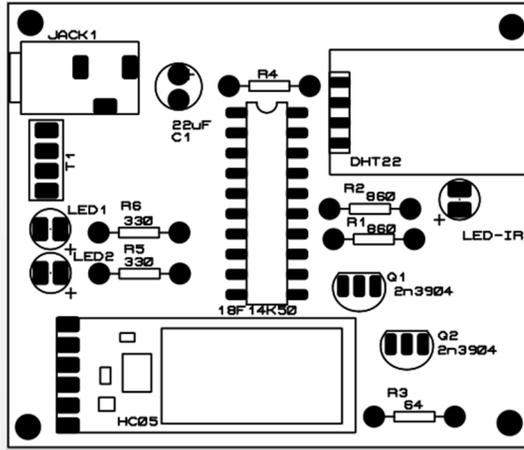


Figura 2-26. PCB1, Top view. Fuente: Autor.

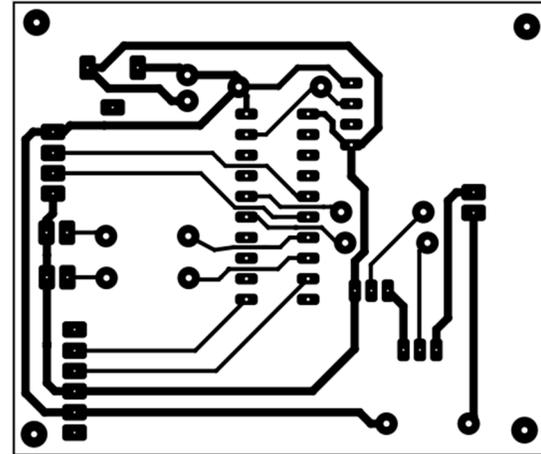


Figura 2-25. PCB1, Bottom view. Fuente: Autor.

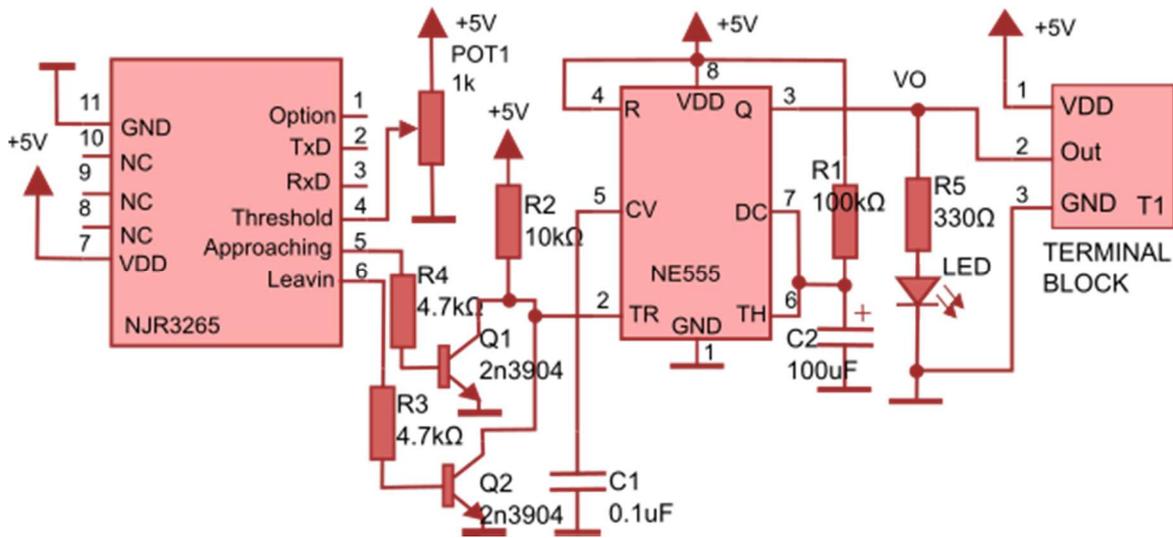


Figura 2-27. Diagrama esquemático del controlador inteligente (PCB2). Fuente: Autor.

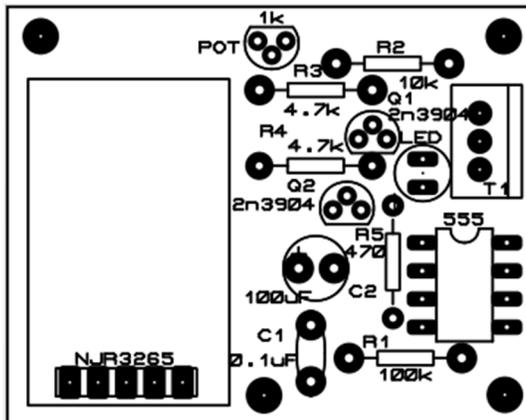


Figura 2-28. PCB2, Top view. Fuente: Autor.

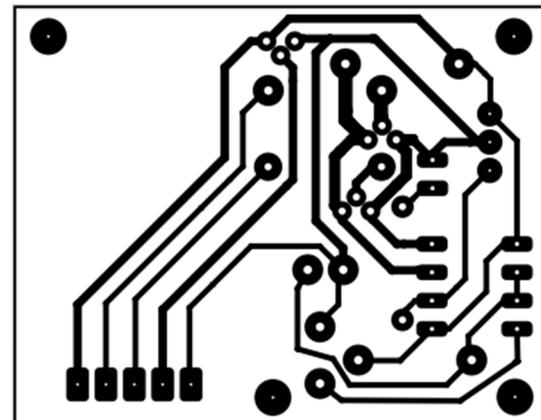


Figura 2-29. PCB2, Bottom view. Fuente: Autor.

2.1.8 Diseño de APP

Una App es una herramienta que le permite al usuario efectuar un conjunto de tareas de cualquier tipo. En este caso en concreto, la App MyController, tiene como objetivo que el usuario, pueda verificar la conectividad del controlador inteligente con su AA, así como también la verificar la funcionalidad del sensor de humedad y temperatura. Para cumplir con estas funciones, esta App contiene dos actividades: la primera de habilitación y conexión del Bluetooth, y la segunda de menú de comandos y visualización de mediciones. Los layouts de estas actividades son mostrados en la Fig. (2-28) y la Fig. (2-29).

-Actividad 1, habilitación y conexión bluetooth- Esta actividad se ejecuta una vez que usuario inicia la app, en ella se le solicita al usuario la habilitación del bluetooth, lo cual se realiza a través del button-switch de encendido y apagado mostrado en la Fig. 2-30. Luego de la habilitación del bluetooth, la app carga la lista de dispositivos vinculados, donde el usuario puede seleccionar el dispositivo quiere conectarse, el este caso el HC-05, véase la Fig. 2-31.

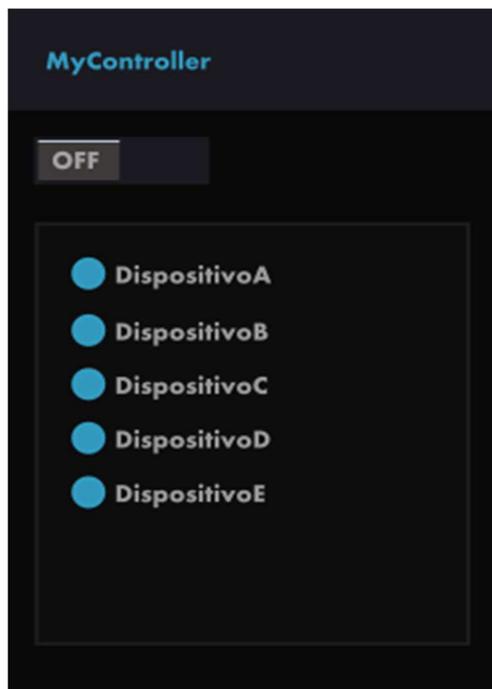


Figura 2-31. Layout-APP actividad 2, habilitación y conexión de bluetooth. Fuente: Autor.

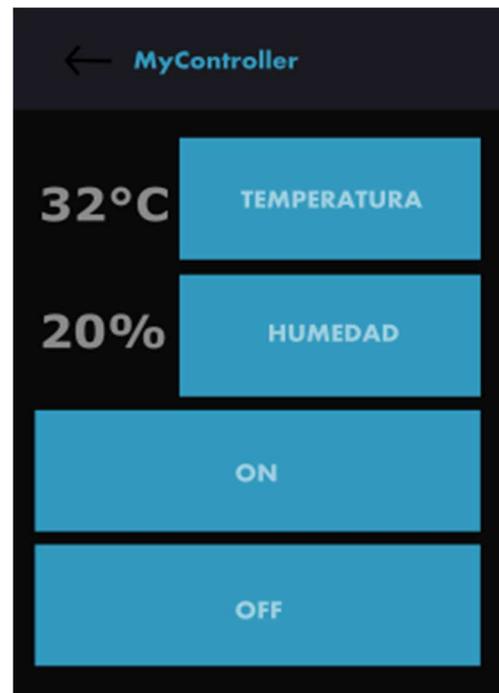


Figura 2-30. Layout-APP actividad 1, menú de comandos y visualización de mediciones. Fuente: Autor.

-Actividad 2, menú de comandos y visualización de mediciones- Con la selección del dispositivo bluetooth, la app ejecuta la segunda actividad, en esta se establece una conexión entre la app y el controlador inteligente. Luego se presenta un menú mediante botones, que usuario puede utilizar para encender o apagar el AA, solicitar las mediciones de la humedad o de la temperatura, y observarlas en pantalla, véase nuevamente la Fig. 2-32. A continuación se proporciona un modelo de esta aplicación que describe su funcionamiento.

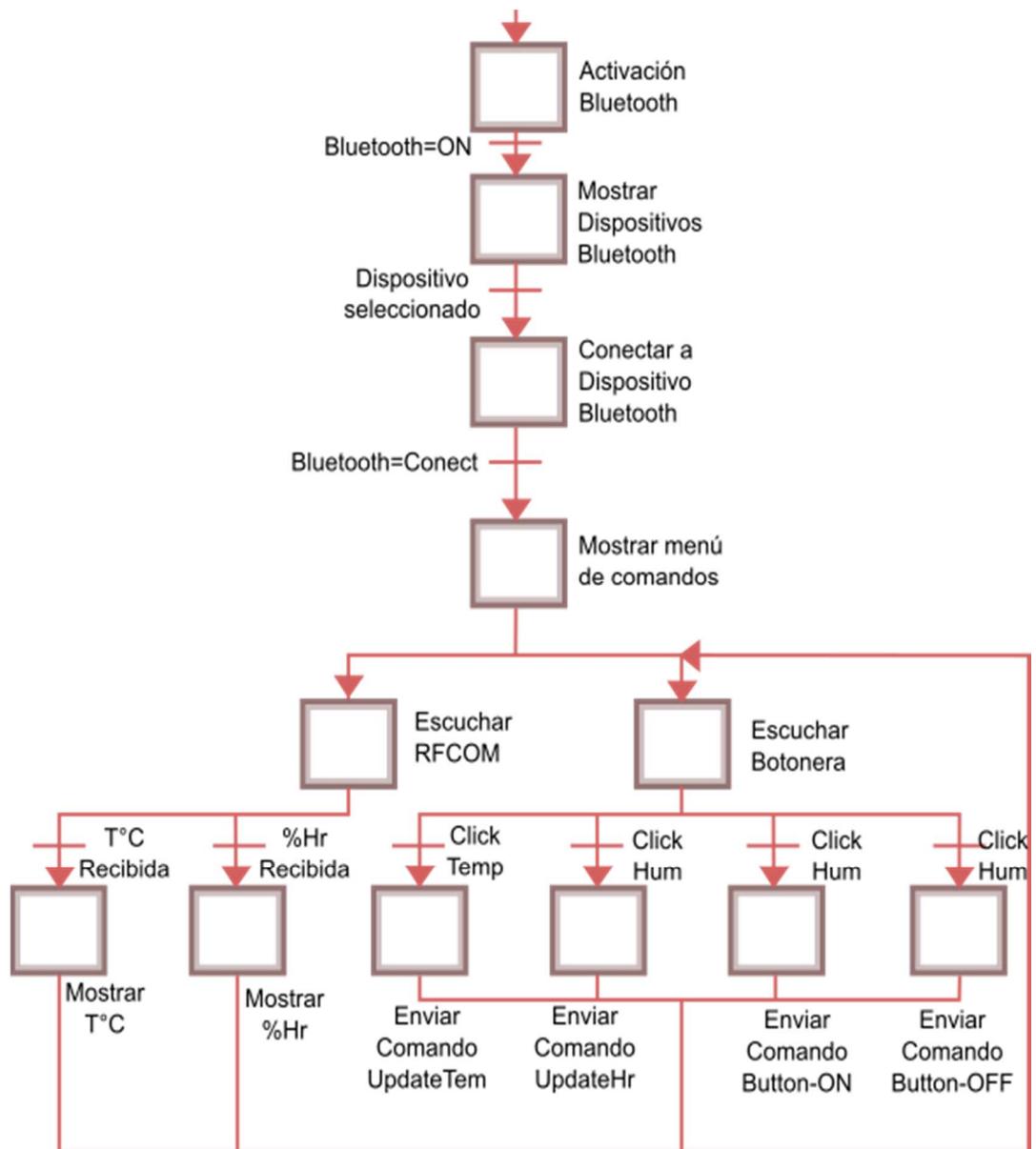


Figura 2-32. Modelo de la App, MyController. Fuente: Autor.

2.2 Diseño de medidor IoT.

2.2.1 Diseño funcional medidor.

El diseño funcional consiste en definir, a grandes rasgos, los elementos que conforman el sistema, según Paul (1993). Cada elemento está destinado a cumplir tareas específicas. Este diseño preliminar es utilizado por el desarrollador como guía en el resto del proceso de diseño. Para realizarlo es necesario definir primeramente los requerimientos del dispositivo.

En este caso, se requiere realizar la medición de la corriente, el voltaje y el factor de potencia en una carga eléctrica, periódicamente. Con estas mediciones, el dispositivo debe determinar la potencia promedio y la energía utilizada en ese periodo. Luego, el registro de la energía utilizada debe ser almacenado en una base de datos.

Los elementos que funcionales que integran al medidor IoT son un sistema de medición, un controlador, un módulo WIFI, un sistema de señalización y uno alimentación. El diseño funcional realizado se muestra en la Fig. 2-33. En el resto de este capítulo se describirán cada uno de estos elementos.

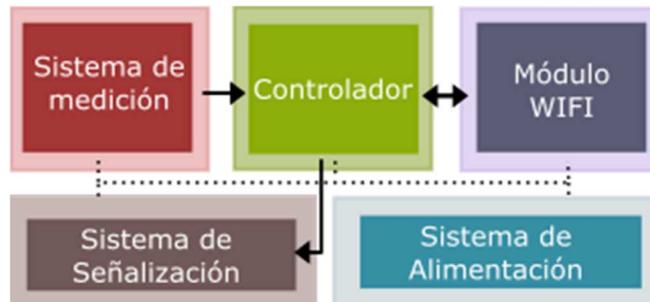


Figura 2-33. *Diseño funcional de medidor de consumo energético. Fuente: Autor.*

2.2.2 Sistema de medición.

El sistema de medición lo constituyen sensores y circuitos de acondicionamiento, que pueden ser etapas de amplificación, filtros, etc. El sistema de medición realiza la transformación de las señales físicas que desean medirse a señales eléctricas, y las acondiciona a niveles adecuados para el controlador pueda realizar las mediciones. En caso del medidor de consumo energético IoT,

este debe realizar las mediciones de corriente y voltaje en una carga para determinar su potencia y consumo energético, se recomienda al lector revisar la sección 1.7 de potencia y energía. El rango de medición de corriente y de voltaje para este dispositivo se muestra en la Tabla 2-23.

Tabla 2-23
Valores de corriente y voltaje
requeridos para medir.

Variable	Valor requerido
Voltaje-AC	250Vrms
Corriente-AC	20Arms

2.2.2.1 *Diseño funcional.*

Esta sección tiene la finalidad de presentar los elementos que integran al sistema de medición y requeridos por el medidor de consumo energético. En la sección 1.7, describe que las variables requeridas para la determinación del consumo energético de una carga o aparato eléctrico, son el voltaje, V_L , corriente, I_L , y factor de potencia, donde este último, es determinado mediante el retardo, θ , entre la corriente y el voltaje utilizados por la carga, por lo que las señales de entrada para este sistema de medición son únicamente el voltaje AC y corriente AC.

Las magnitudes de estas variables deben ser acondicionadas a rangos adecuados para que puedan ser medidos por un microcontrolador, por lo que se requieren circuitos acondicionamiento. Pero también, se requiere de un circuito para la medición del desfase entre señales de entrada. A continuación, en la Fig. 2-34 se presenta el diseño funcional del sistema de medición, donde, **S1**, **S2** y **S3**, son las señales que el microcontrolador utilizará para la determinación de V_L , I_L y θ , respectivamente.

2.2.2.2 *Sensor de corriente.*

En el caso de la medición de corriente se utilizó el sensor SCT013-000, mostrado en la Fig. 2-35, este es un transformador de corriente no invasivo, lo que significa que el SCT013-000 no interrumpen la corriente de la carga, sino que realiza la medición a partir del campo magnético generado por I_L en la línea de

alimentación. Este sensor permite realizar mediciones de hasta 100Arms, la salida es de tipo corriente, denominada I_{OTC} , la cual se encuentra relacionada con I_L , mediante la Ec. (2-22).

$$I_L = 2000 \cdot I_{OTC} \quad (2-22)$$

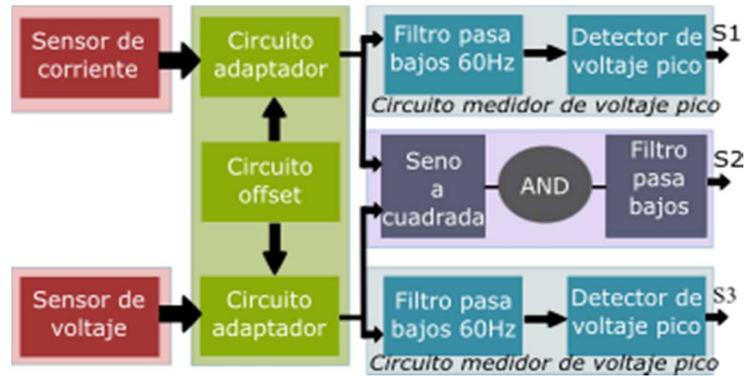


Figura 2-34. Diseño funcional de sistema de medición. Fuente: Autor.



Figura 2-35. Sensor de corriente, SCT013-000. Fuente: Autor.

2.2.2.3 Sensor de voltaje.

Por otro lado, para la medición del voltaje en la carga, V_L , se utilizó un transformador de voltaje. Este dispositivo permite reducir la tensión en un circuito eléctrico. Esta capacidad depende de la relación de espiras de embobinado, N , que relaciona el voltaje en el primario y el secundario, según la Ec. (2-23). El transformador utilizado tiene una relación de esperas de 17.6, y es mostrado en la Fig. 2-36.

$$N = \frac{V_2}{V_1} \quad (2-23)$$



Figura 2-36. Transformador para la medición de voltaje.
Fuente: Autor.

2.2.2.4 Señales de salida, sensor de corriente y voltaje.

Las señales de salida proporcionadas por sensores de corriente y voltaje, I_{out} y V_{out} , respectivamente, son de tipo sinusoidales, con oscilaciones en 0V, tal como se ilustra en la Fig. 2-37. Sin embargo, el módulo ADC de un microcontrolador solo permite la medición dentro del rango de 0V a VDD, es decir, únicamente valores positivos y un voltaje máximo de VDD. Debido a esto es necesario la incorporación de un offset a las señales de salida de los sensores de corriente y voltaje. Además, otro aspecto de estas señales de salida es que su rango dinámico no se ajusta completamente al rango de medición del microcontrolador. También es necesario realizar un ajuste en la amplitud de estas señales. Los ajustes mencionados ilustrados en la Fig. 2-38, son realizados por los circuitos adaptadores y el circuito de offset, mostrados anteriormente en la Fig. 2-34.

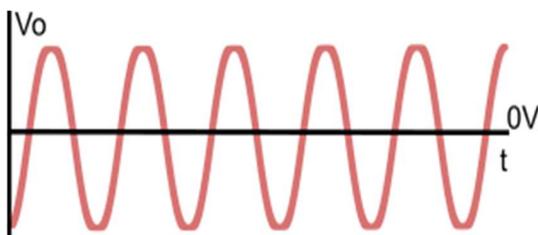


Figura 2-38. Salida de los sensores de corriente y voltaje. Fuente: Autor.

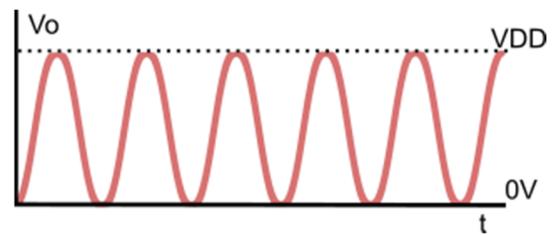


Figura 2-37. Ajuste a las señales de salida de los sensores de corriente y voltaje. Fuente: Autor.

2.2.2.5 Circuito offset.

Como se ha mencionado anteriormente el rango en el que deben oscilar las señales de salida de los sensores es de 0V a VDD. El voltaje de alimentación utilizado para este dispositivo es 3.3V. Para lograr una simetría en las oscilaciones dentro del rango de 0V a 3.3V, el voltaje de offset, V_{offset} , debe tener un valor de 1.65, tal como se ilustra en la Fig. 2-39. En lo que respecta a la amplitud de estas oscilaciones, la magnitud máxima seleccionada es 1.5V, sobre el offset.

El circuito offset se encarga de establecer un voltaje de referencia de 1.65V. El circuito utilizado se muestra en la Fig. 2-40, y lo constituye un divisor de voltaje entre R1 y R2, y un buffer o seguidor de voltaje. Los valores de R1 y R2 son valores genéricos de 10k, y para C1 se utilizó capacitores de 22uF.

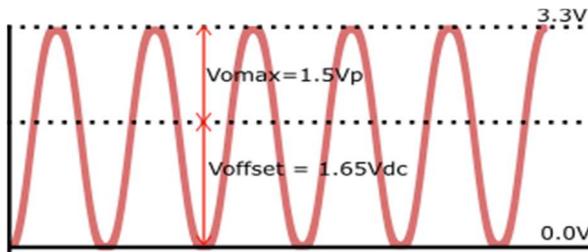


Figura 2-40. Incorporación de un offset a la señales de salida de los sensores de corriente y voltaje. Fuente: Autor.

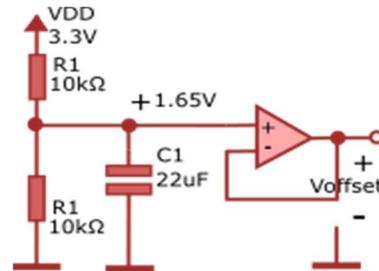


Figura 2-39. Circuito Offset. Fuente: Autor.

2.2.2.6 Circuito adaptador de voltaje para sensor de corriente.

El SCT013-000 tiene una salida de tipo corriente, la cual debe ser convertida en voltaje para poder ser medida por un microcontrolador. Esto se logra colocando una resistencia en sus terminales salida. El voltaje de salida, V_{O1} , se relaciona con la corriente de salida, I_{OTC} , mediante la ley de Ohm, como se indica en la Ec. (2-24).

$$R_{OTC} = \frac{V_{O1}}{I_{OTC}} \quad (2-24)$$

El valor de R_{OTC} limita la amplitud de V_{O1} , es necesario encontrar un valor adecuado para esta resistencia. Este valor depende de la amplitud máxima del voltaje de salida, V_{O1max} , y de la corriente máxima de medición, I_{Lmax} . Anteriormente se mencionó que el valor máximo de V_{O1} es 1.5V. Por otro lado, la I_{Lmax} establecida es 20Arms o 28.28Ap, tal como se indica en la Tabla 2-23,

mostrada anteriormente. Retomando la Ec. (2-22), despejándola y sustituyendo los valores, I_{OTCmax} se calcula según la Ec. (2-25). Por otro lado, (2-24) se resuelve en la Ec. (2-26) como sigue.

$$I_{OTC_{MAX}} = \frac{28.28Ap}{2000} = 14.14mA_p \quad (2-25)$$

$$R_{OTC} = \frac{1.5V_p}{14.14A_p} = 106\Omega \quad (2-26)$$

Por otro lado, la incorporación del offset de voltaje a V_{O1} , se obtiene al colocar el voltaje de referencia, V_{offset} , en una de las terminales de salida del sensor, como se indica en la Fig. 2-41.

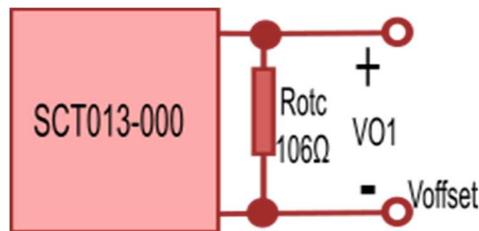


Figura 2-41. Circuito adaptador de voltaje para el sensor de corriente.

Fuente: Autor.

2.2.2.7 Circuito adaptador de voltaje para sensor de voltaje.

El sensor utilizado para la medición de voltaje de carga, V_L , es un transformador de voltaje, el cual reduce la amplitud del voltaje de entrada a niveles apropiados para la seguridad del usuario. Sin embargo, no los adecuados para un microcontrolador. Esto se debe a que, con una relación de espiras N de 17.6, y un V_{LMAX} de 250Vrms o 353Vp, el voltaje de salida del transformador, V_{O2MAX} , tiene un valor de 20Vp, tal como se indica en la Ec. (2-27) a continuación.

$$V_{O2MAX} = \frac{353V_p}{17.6} = 20V_p \quad (2-27)$$

Debido a este nivel de tensión, es requerido colocar un divisor de voltaje en la salida transformador para reducir el voltaje salida a un valor apropiado, en este caso 1.5V. Debido a esto el divisor de voltaje debe tener una ganancia de 0.075, según se indica la Ec. (2-28), donde V_o es el voltaje de salida del divisor de voltaje, y V_i el voltaje de salida.

$$G = \frac{V_o}{V_i} = \frac{20V_p}{1.5V_p} = 0.075 \quad (2-28)$$

Por otro lado, la ganancia, **G**, del divisor de voltaje se expresa la Ec. (2-29).

$$\frac{1}{G} = \left(1 + \frac{R1}{R2}\right) \quad (2-29)$$

Esta ganancia depende de los valores de **R1** y **R2**. Por lo tanto, para cumplir con la relación, se utilizó un valor de 100k para **R1**, mientras que para **R2** se utilizó un valor de 7.5k. En lo que refiere al offset de voltaje, este es incorporado a la señal de salida del transformador, al colocar un voltaje de referencia, **V_{OFFSET}**, en una de sus terminales, como se muestra Fig. 2-42.

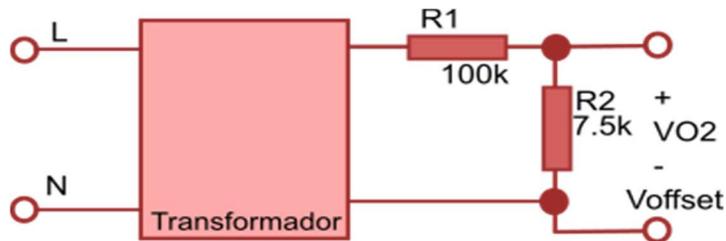


Figura 2-42. Circuito adaptador de voltaje para el sensor de voltaje. Fuente: Autor.

2.2.2.8 Circuito medidor de voltaje pico.

Este circuito se encarga de tomar las señales sinusoidales de salida de los circuitos adaptadores de voltaje, y proporcionar un voltaje en DC correspondiente a la amplitud de la onda sinusoidal de entrada, como se ilustra en la Fig. 2-43. Para realizar esta tarea el circuito medidor de voltaje pico cuenta con tres etapas: un seguidor de voltaje, un filtro pasa-bajo, y una tercera etapa que captura el voltaje máximo de entrada en la salida. El esquema de este circuito se muestra a continuación en la Fig. 2-44.

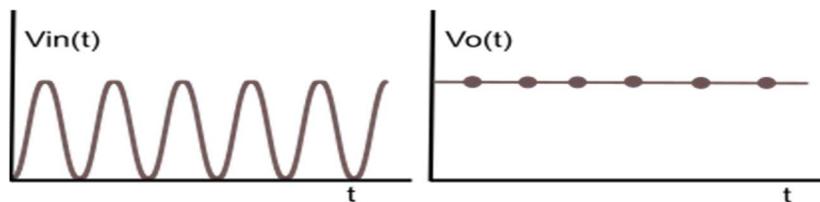


Figura 2-43. Entrada y salida del circuito medidor de voltaje pico. Fuente: Autor.

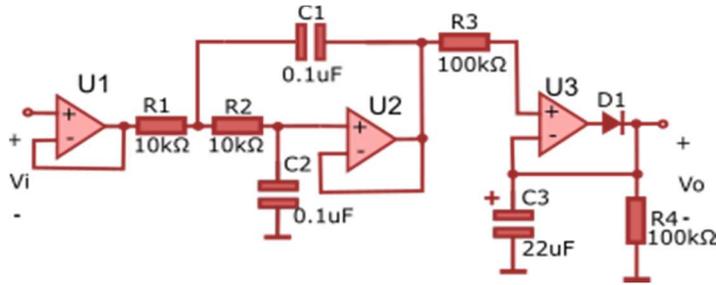


Figura 2-44. Circuito medidor de voltaje pico. Fuente: Autor.

Primeramente, **U1** es únicamente utilizado como seguidor de voltaje. Mientras que, **U2**, es utilizado como filtro pasa-bajo en conjunto con **R1**, **R2**, **C1**, y **C2**. Donde, estos últimos definen la respuesta en frecuencia del filtro, es decir, su frecuencia de corte f_c . Esta frecuencia de determina según la Ec. (2-30), su valor fue determinado de forma experimental, con el requerimiento que la ganancia de voltaje del filtro, para una frecuencia de 60Hz, fuera unitaria. Como resultado, se determinó que f_c debe ser 160Hz.

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (2-30)$$

$$160\text{Hz} = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (2-31)$$

Para cumplir con la relación expresa en la Ec. (2-31), se seleccionó un valor 10k, para **R1** y **R2**, y un valor de 0.1μF, para **C1** y **C2**.

Por último, en la última etapa, se encuentra el captador de voltaje pico **U3**. Este circuito permite almacenar el voltaje pico de la señal de entrada en el capacitor de salida **C3**. Los resistores **R3** y **R4**, permiten la carga y descarga del capacitos C3. Además, estos poseen el mismo valor resistivo para garantizar una ganancia unitaria. Véase nuevamente la Fig. 3-12. Asi también las demás etapas del circuito medidor de voltaje. Esto permite relacionar el voltaje de entrada y de salida utilizan la Ec. (2-32)

$$V_o = \text{MAX}(V_i(t)) \quad (2-32)$$

2.2.2.9 Circuito de desfaseamiento.

El desfaseamiento se define como el retardo que existe entre dos señales del mismo periodo. El desfaseamiento, θ , entre el voltaje y corriente de una carga, es importante para el consumo energético, ya que representa un factor en la forma en

la que carga consume energía, conocido como factor de potencia, f_p . Véase la sección 1.8.

Este circuito proporciona un voltaje en DC, V_{OF} , que permite determinar el retardo entre dos señales sinusoidales. Esto se lleva a cabo mediante tres pasos: primeramente, las señales sinusoidales de entrada son convertidas en señales cuadradas; luego, estas señales cuadradas son mezcladas en una única señal, y más tarde dicha señal es filtrada para obtener un valor en DC como salida.

En el proceso de conversión, cada señal sinusoidal es transformada en una señal cuadrada. Para lograrlo el voltaje de entrada es comparado con un voltaje de referencia, V_{REF} , el cual corresponde con el eje de oscilación de la onda sinusoidal, en este caso 1.65V. De manera que, durante el semiciclo de positivo de la onda, la salida permanece en alto, mientras que durante el semiciclo negativo, la salida permanece en bajo. Este proceso es ilustrado en la Fig. 2-45.

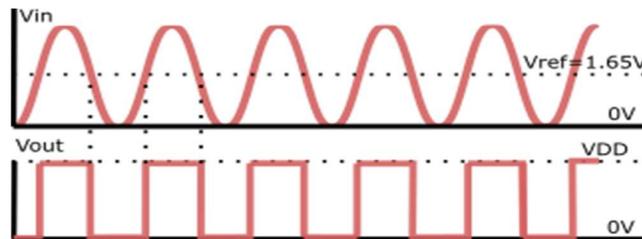


Figura 2-45. Proceso de conversión de una onda sinusoidal a cuadrada. Fuente: Autor.

Luego del proceso de conversión, las señales cuadradas generadas por en la conversión, son mezcladas en una única señal. Esta mezcla es realizada mediante una operación AND entre ambas señales. La señal resultante contiene información sobre retardo entre ambas señales. Para ilustrar el proceso de mezclado véase la Fig. 2-46.

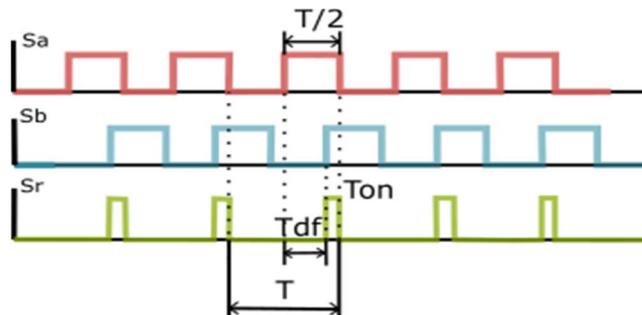


Figura 2-46. Proceso de conversión de una onda sinusoidal a cuadrada. Fuente: Autor.

En la figura anterior, la señales **S_a** y **S_b**, corresponden a la señales de entrada, mientras que **S_r** es la señal resultante. Utilizando a **S_r** es posible determinar el retardo entre ambas señales de entrada, **T_{df}**, utilizando las ecuaciones (2-33) y (2-34), En estas donde, **T**, es el periodo de la señal de 60Hz, y **T_{ON}** es el tiempo en alto de la señal resultante.

$$T_{DF} = \frac{T}{2} - T_{ON} \quad (2-33)$$

$$\theta = \left(\frac{T_{DF}}{\frac{T}{2}} \right) \quad (2-34)$$

En estas ecuaciones es necesario determinar **T_{ON}** para encontrar **T_{df}**. Debido a esto se utiliza un filtro pasa bajo con frecuencia de corte **f_c = 0**. Este proporciona un voltaje de salida DC, **V_o**, que se relaciona con **T_{ON}**, como se indica en la Ec. (2-35), donde T es periodo de la señal de 60Hz, y VDD es voltaje de alimentación del sistema, es decir, 3.3V.

$$T_{ON} = \left(\frac{T}{V_{DD}} \right) \cdot V_o \quad (2-35)$$

En la Fig. 2-47 se muestra el diagrama esquemático del circuito medidor de desfase. En este, **U1** y **U2** son comparadores de voltaje, y realizan función de conversión de sinusoidal-cuadrada. Por otro lado, la mezcla de ambas señales cuadradas resultante, es llevada a cabo mediante la compuerta AND, implementada con diodos D1 y D2, y la resistencia R5. Mientras, la última etapa **U3**, constituye el filtro pasa-bajo.

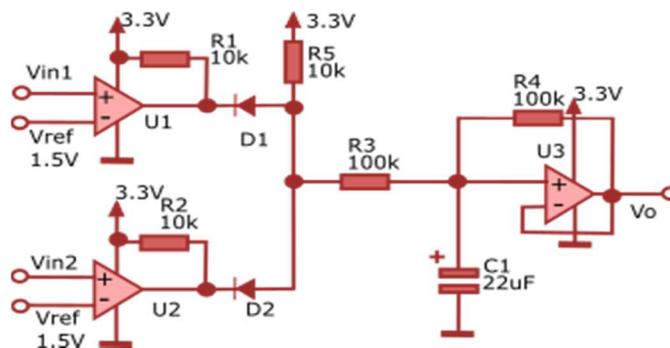


Figura 2-47. Diagrama esquemático medidor de desfase. Fuente: Autor.

Luego de describir el diseño de cada una de las etapas de sistema de medición, a continuación se muestra su diseño total en la Fig. 2-48.

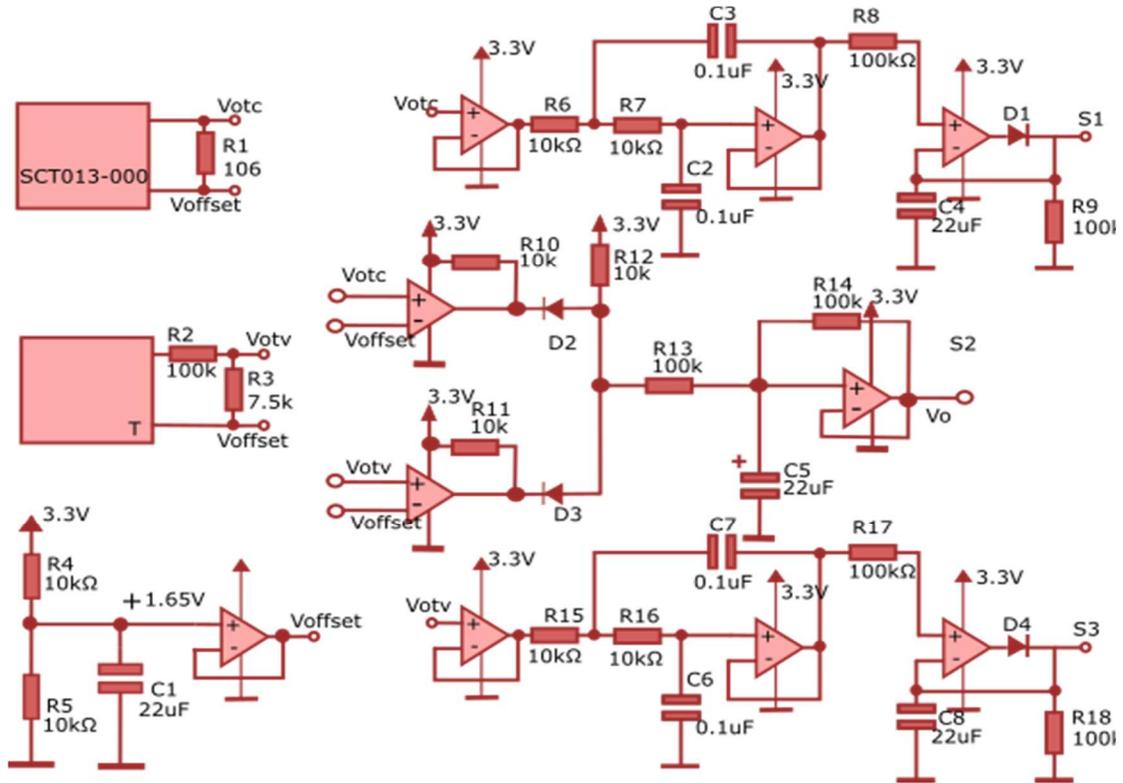


Figura 2-48. Diseño total del sistema de medición. Fuente: Autor.

Utilizando el esquemático anterior, y simplificando los cálculos, los valores de I_L , V_L y θ , se calculan de acuerdo con las siguientes ecuaciones.

$$I_{Lrms} = \frac{2000}{106\sqrt{2}}(S_1 - V_{offset}) \quad (2-36)$$

$$\theta = \left(1 - 2 \frac{S_2 - V_{offset}}{VDD}\right) \quad (2-37)$$

$$V_{Lrms} = \frac{17.6}{\sqrt{2}} \left(1 + \frac{100}{7.5}\right) (S_3 - V_{offset}) \quad (2-38)$$

2.2.3 Sistema de comunicación.

Un sistema de comunicación es un conjunto dispositivos interconectados a través de un medio o canal que intercambia información. En este caso, lo integran el controlador, el módulo WIFI y el servidor WEB.

El módulo WIFI es una interfaz de comunicación que le permite a medidor de consumo energético conectarse a una red inalámbrica y enviar las mediciones realizadas a un servidor WEB. El dispositivo utilizado para esta aplicación es el ESP01. El cual es un módulo de comunicación WIFI basado en el esp8266EX, que es un sistema en chip o SoC. Se recomienda ver la sección 1.9. Sin embargo,

en esta sección se describe cómo se lleva a cabo la transferencia de datos del microcontrolador hacia el servidor web.

En este sistema el ESP01 opera como un adaptador de protocolos, es decir, toma los datos de entrada en un formato y transforma en otro. En este caso, los datos enviados por el microcontrolador, correspondientes a las mediciones e instrucciones que debe seguir el ESP01, son recibidos en un formato RS232, y adaptados a protocolos de red. Esto se ilustra en la Fig. 2-49.



Figura 2-49. Comunicación controlador-servidor WEB. Fuente: Autor.

El módulo ESP01 es reprogramable, es decir, que es posible cargar un nuevo programa al dispositivo, utilizando el IDE de Arduino. Esto permite definir un conjunto de tareas específicas, mediante rutinas de programa, que la ESP01 puede ejecutar cuando se lo solicite. Las tareas requeridas para el medidor de consumo energético son tres: conexión WIFI, estado de conexión WIFI, y envío de datos (kwh). A continuación, se describe a profundidad, como se dictan las instrucciones a la ESP, y como este dispositivo las lleva cabo.

2.2.3.1 Comunicación controlador-ESP.

La comunicación controlador-ESP se realiza mediante un intercambio mensajes RS232, utilizando una trama de bytes de solicitud y una de respuesta. En la Fig. 2-50 se muestra la estructura de mensajes de la comunicación controlador-ESP. En estos mensajes, el controlador indica la tarea que debe realizar la ESP median un byte llamado **comando**. Cada comando es interpretado por la ESP para ejecutar una rutina específica de programa. El byte comando puede tener los valores mostrados en la Tabla 2-24.



Figura 2-50. Protocolo de comunicación controlador-ESP. Fuente: Autor.

Tabla 2-24

Lista de comandos del controlador

Comando	Valor
<i>Conect_wifi</i>	0xA1
<i>Estatus_wifi</i>	0xA9
<i>Load_energ</i>	0xCA

En lo que respecta a **dato1** y **dato2** son utilizados por el controlador para cargar la medición realizada. El **dato1** corresponde con la parte entera de la medición, mientras que **dato2** a la parte decimal. El valor de la medición puede ser determinado según la ecuación siguiente.

$$v(kwh) = dato1 + \frac{dato2}{100} \quad (2-39)$$

Por otro lado, el byte **respuesta** es utilizado por la ESP para indicarle al controlador, si la instrucción enviada fue ejecutada, o no. Este byte puede adoptar dos valores: **ACKOK** y **NOACK**. En caso que ocurra un error en la instrucción, la ESP retorna NOACK, en cambio, si no ocurre ninguna anomalía, retorna **ACKOK**.

2.2.3.2 Modelo de programa ESP01.

Como se ha mencionado anteriormente, el ESP01 es programado para realizar un conjunto determinado tareas, indicadas por el controlador, según la Tabla 2-24. Este dispositivo espera una instrucción, y luego la lleva a cabo. En la Fig. 2-51 se muestra el modelo de programa de la ESP01.

2.2.3.1 Comunicación ESP-servidor web.

La comunicación entre el módulo ESP y servidor WEB, es muy compleja, esto debido a que la información es enviada a través de una red de computadoras, la cual puede ser caracterizada en distintos niveles o capas según el modelo TCP/IP.

La Fig. 2-52 muestra un datagrama que describe los protocolos por capa, utilizados en la comunicación ESP-Servidor.

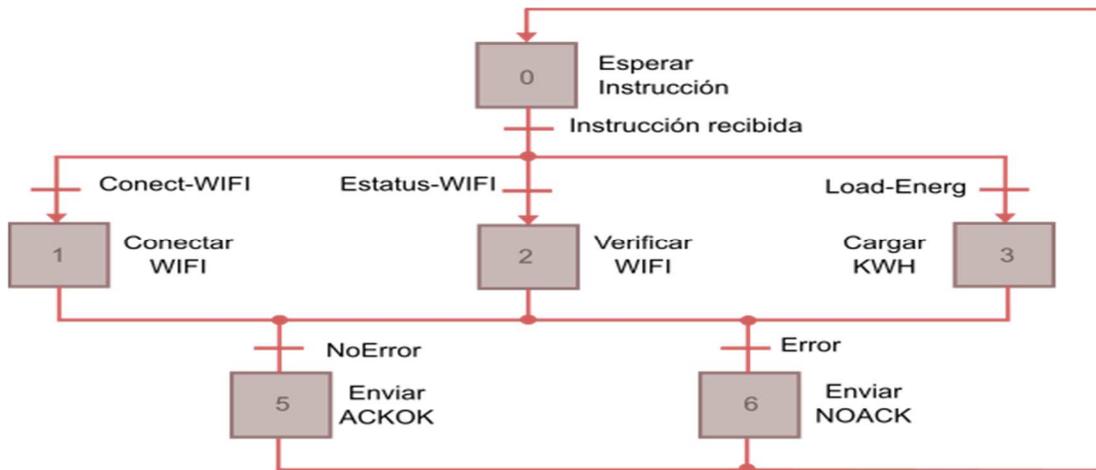


Figura 2-51. Modelo de Programa módulo ESP01. Fuente: Autor.

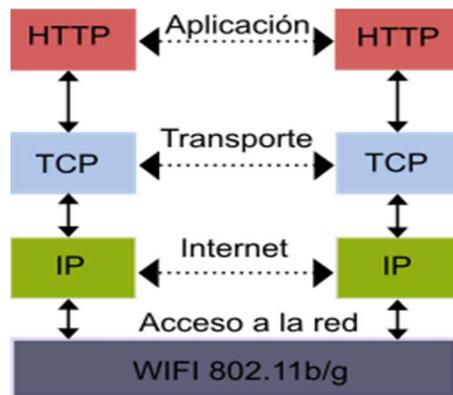


Figura 2-52. Datagrama entre la ESP y el servidor. Fuente: Autor.

En la capa de acceso a la red vemos que ambos dispositivos utilizan el protocolo WIFI 802.11b/g/n que trabaja en la banda de frecuencia de 2.4GHz.

Por su parte en la capa de internet, se utiliza el protocolo IP para direccionar los datos. En lo que se refiere al servidor, este es configurado con una IP estática dentro de la red local, específicamente **192.168.0.5**. Por otro lado, al módulo ESP se le asigna una dirección IP de manera automática cada vez que este accede a la red local (**192.168.0.x**).

Con respecto a la capa de transporte, las mediciones son enviadas utilizando el protocolo TCP, lo que garantiza que los datos serán entregados a su destino sin errores y en mismo orden que fueron entregados.

En la capa más alta del modelo TCP/IP, la capa de aplicación, la ESP y el servidor utilizan el Protocolo de Transferencia de Hipertexto o HTTP para comunicarse. Las mediciones realizadas por el controlador, y recibidas por la ESP, son enviadas mediante mensajes HTTP hacia el servidor utilizando el método POST. Cada vez que la ESP recibe la instrucción de cargar datos al servidor, esta crea un mensaje HTTP-POST, como el que se muestra en la Fig. 2-53 a continuación.

The diagram shows an HTTP-POST message with the following structure and annotations:

```

POST http://192.168.0.5/load_data.php HTTP/1.0
Host: 192.168.0.5
Accept: */*
Content-Length: 3
Content-Type: application/x-www-form-urlencoded
data1 & data2
  
```

Annotations in the diagram:

- URL:** A bracket above the first line points to the entire URL path: `http://192.168.0.5/load_data.php`.
- HOST NAME:** An arrow points from the text "HOST NAME" to the IP address `192.168.0.5` in the Host header.
- script code:** An arrow points from the text "script code" to the filename `load_data.php` in the URL path.
- Bytes a enviar (mediciones):** An arrow points from the text "Bytes a enviar (mediciones)" to the data `data1 & data2` in the body of the message.

Figura 2-53. Estructura de mensaje HTTP-POST generado por la ESP01.
Fuente: Autor.

El mensaje anterior, contiene la IP del servidor (**192.168.0.5**), los datos que desean enviarse al servidor (**data1** y **data2**), y script code o código de aplicación. Este último se encarga de recibir los datos recibidos mediante POST, y almacenarlos en una base datos. A continuación en la Fig. 3-22, se muestra un ejemplo de un script code PHP utilizado para recibir datos mediante POST y guardarlos en una base datos.

2.2.4 Sistema de señalización.

El sistema de señalización son indicadores LEDs que proporcionan información al usuario acerca de la operatividad del dispositivo. En este caso, es necesario que el dispositivo indique el estado de la conexión WIFI, y también el instante en se transmite un dato. El circuito para los indicadores LEDs, es mostrado en la Fig. 2-54, donde señal-WIFI y señal-DATA, son las señales generadas por el controlador para controlar los LEDs. A continuación se realiza un análisis de este circuito.

```

<?php
-----Buffer-----
$integer_part = 0x00;
$decimal_part = 0x00;
-----Recepcion a traves POST-----
$integer_part = $_POST ['t_interpart'];
$decimal_part = $_POST ['t_decimpart'];
-----Conversion to float-----
$datLoad = $integer_part + $decimal_part/100;
-----Conexion a mysql-----
$mysqli = new mysqli($host, $user, $passwd, $Bdbname);
-----Almacenamiento en BD-----
$intrs=mysqli_query($mysqli,"INSERT INTO $nametable ($col1, $col2) VALUES (NULL, $datLoad)");
-----cerramos las conexion con BD-----
mysqli_close($mysqli);
?>

```

Figura 2-54. Ejemplo scrip code php para recepción de datos mediante POST. Fuente: Autor.

Primeramente, el circuito mostrado en la Fig. 2-55 cuenta dos mallas eléctricas simétricas, una para el LED1 y otra para el LED2, lo que significa que R1 y R2 poseen el valor de resistencia. Además, los LEDs utilizados para este diseño son de color verde, por lo que se debe considerar las características de polarización para este tipo LEDs en la determinación de los valores de R1 y R2, véase la Tabla 2-25.

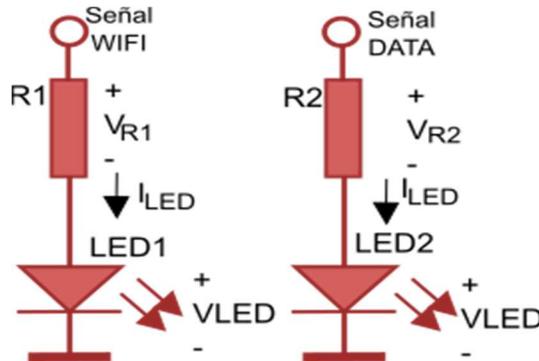


Figura 2-55. Circuito de indicadores LEDs.
Fuente: Autor.

Tabla 2-25
Características eléctricas para un LED-VERDE

Característica	Símbolo	Valor	Unid.
Voltaje de polarización den directa	V_{LED}	1.7	V
Corriente de encendido	I_{LED}	10	mA

Realizando un análisis de las caídas de voltaje en circuito de la Fig. 2-55, el valor de R2 y R2 puede calcular según la Ec. (2-40).

$$R2 = R1 = \frac{V_{DD} - V_{LED}}{I_{LED}} \quad (2-40)$$

Resolviendo.

$$R2 = R1 = \frac{5.0V - 1.7V}{10mA} = 160\Omega \quad (2-41)$$

2.2.5 Sistema de control.

2.2.5.1 Microcontrolador.

El microcontrolador es el cerebro dentro del sistema que se encargará de coordinar las tareas de medición y de envío de datos hacia el servidor según su programación. Este debe realizar mediciones de las salidas de voltaje del sistema de medición, mostrar indicaciones de la operatividad del dispositivo y enviar las mediciones en formato RS232. En la Tabla 2-26 se describen los recursos necesarios para el microcontrolador.

Tabla 2-26

Periféricos de entrada y salida del microcontrolador.

ítem	Tipo	I/O	Descripción
1	Analog	I	Medición de VL.
2	Analog	I	Medición de IL.
3	Analog	I	Medición de FP.
4	Bool	O	LED-verde, indicador de WIFI.
5	Bool	O	LED-verde, Indicador de datos.
7	UART	O	RX.
8	UART	I	TX.

Con respecto a que dispositivo se utilizó, en la sección anterior, en el diseño del controlador inteligente, se decidió utilizar en microcontrolador PIC18f14k50. Por lo tanto, con el objetivo de simplificar el diseño del medidor, y que este dispositivo cuenta con los recursos requeridos, se decidió también utilizar el PIC18f14k50 para el medidor de consumo.

2.2.5.2 Modelo de programa.

Dentro de la programación del microcontrolador, se realiza las mediciones de **IL**, **VL** y **θ** para determinar la potencia promedio. Sin embargo, estas no se realizan de forma continua, sino de manera discreta. Esto significa que el controlador considera un pequeño intervalo de tiempo, **Δt**, entre una y otra medición de potencia promedio, **P_i**, como lo indica la Ec. (2-42).

$$P_i = \frac{1}{2} V_{Li} \cdot I_{Li} \cdot \cos \theta_i \quad [W] \quad (2-42)$$

Asociada a esta potencia P_i , se encuentra un consumo de energía e_i dentro del mismo intervalo de tiempo Δt , como lo indica la Ec. (2-43).

$$e_i = P_i \cdot \Delta t \quad [J] \quad (2-43)$$

Utilizando estos valores discretos, el consumo total de energía es estimado sumando el aporte individual de cada e_i , como se muestra en la Ec. (2-44).

$$W = \sum_0^n e_i \quad [J] \quad (2-44)$$

En la Fig. 2-56 se ilustra cómo se lleva a cabo la estimación del consumo de energía. La curva mostrada representa la potencia promedio consumida por una carga a lo largo del tiempo. El área de bajo de esta curva es la energía utilizada. Esta área (energía), puede ser estimada sumando las áreas, A_i , de los rectángulos mostrados. Observe las similitudes entre las ecuaciones (2-43) y (2-45).

$$A_i = P_i \cdot \Delta t \quad [J] \quad (2-45)$$

$$A = \sum_0^n A_i \quad [J] \quad (2-46)$$

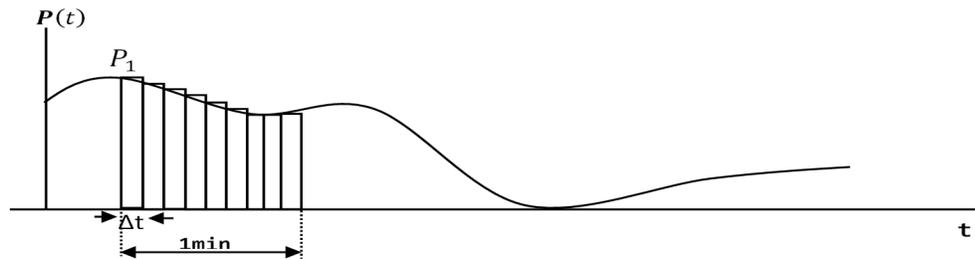


Figura 2-56. Consumo de energía de una carga. Fuente: Autor.

De manera que el controlador realiza la estimación del consumo energético cada Δt seg. Este consumo de energía es acumulado y luego de un 1min este es guardado en la base datos. En la Fig. 2-57, se muestra un modelo del programa del controlador del medidor de consumo. Donde Δt es 3seg.

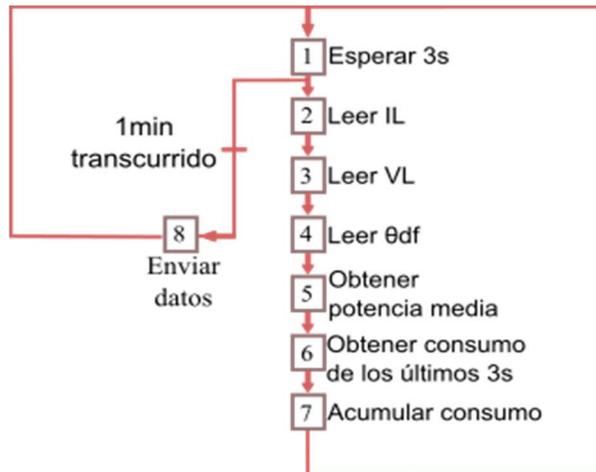


Figura 2-57. Modelo de programa del microcontrolador. Fuente: Autor.

2.2.6 Sitio web.

Este sitio web tiene la finalidad que usuario pueda consultar las mediciones del consumo energético de un AA, llevadas a cabo durante la fase de experimentación y prueba del controlador inteligente. Las mediciones realizadas se utilizaran para comparar la eficiencia energética del AA, con el uso del controlador inteligente y sin este.

El medidor IoT es el dispositivo que se encarga de realizar las mediciones del consumo energético. Este envía las mediciones a un servidor web, y luego este las carga a una base de datos. Con este sitio el usuario puede consultar estas mediciones. En esta sección se describe la su diseño y funcionalidad-Se recomienda al lector revisar la sección 1.10 sobre las tecnologías utilizadas en la implementación de este sistema web.

Este sitio web se encuentra constituido por tres páginas: Home, Gráficos-Main, Gráficos-n. En la Fig. 2-58 se muestra como el usuario puede navegar a través sitio web. Primeramente, Home presenta una descripción del sitio, su finalidad, es dar información sobre el autor de este tema monográfico, véase la Fig. 2-59. Por otro lado, Gráficos-Main carga un conjunto gráficos que describen el consumo energético por día, especificando además, si ese consumo se realizó utilizando el controlador o no, Véase la Fig. 2-60. Además de esto, Gráficos-Main carga gráficos estadísticos que comparan el consumo realizado con el controlador

inteligente y sin este. Por último, Gráficos-n representa a un conjunto de páginas web, cada una ellas carga el consumo energético de un día n, véase la Fig. 2-61.

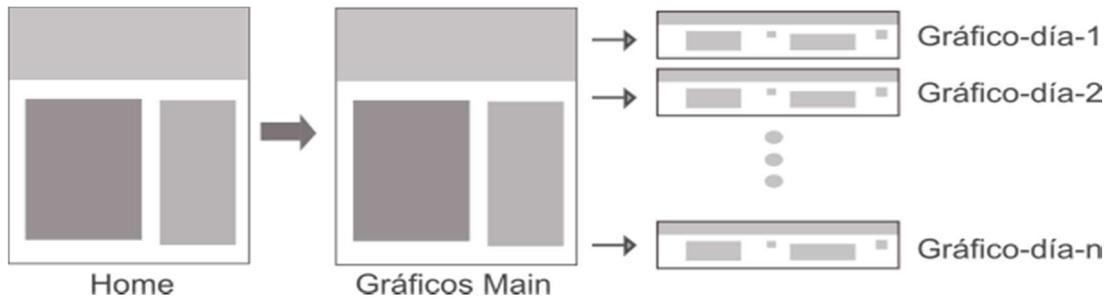


Figura 2-58. Modelo del sitio web. Fuente: Autor.



Figura 2-59. Home del sitio web. Fuente: Autor.

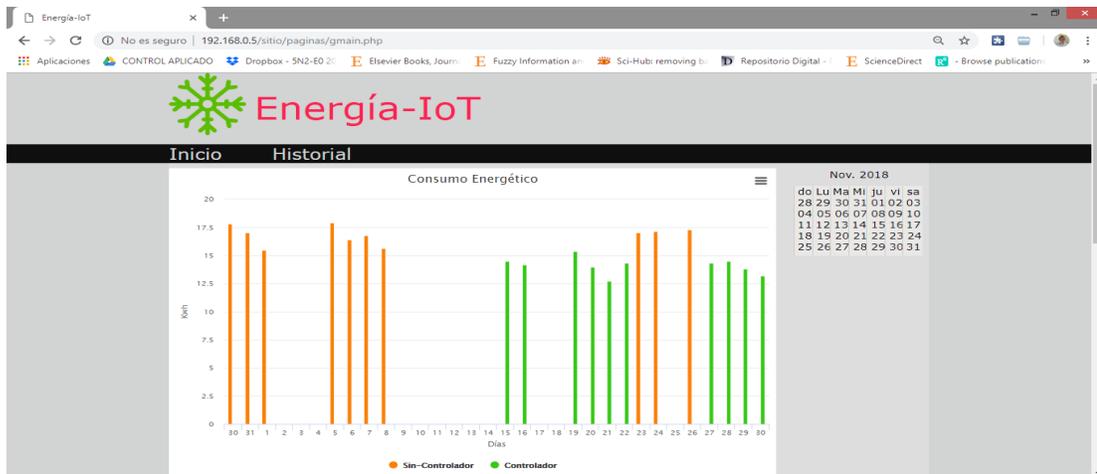


Figura 2-60. Gráficos-Main. Fuente: Autor.

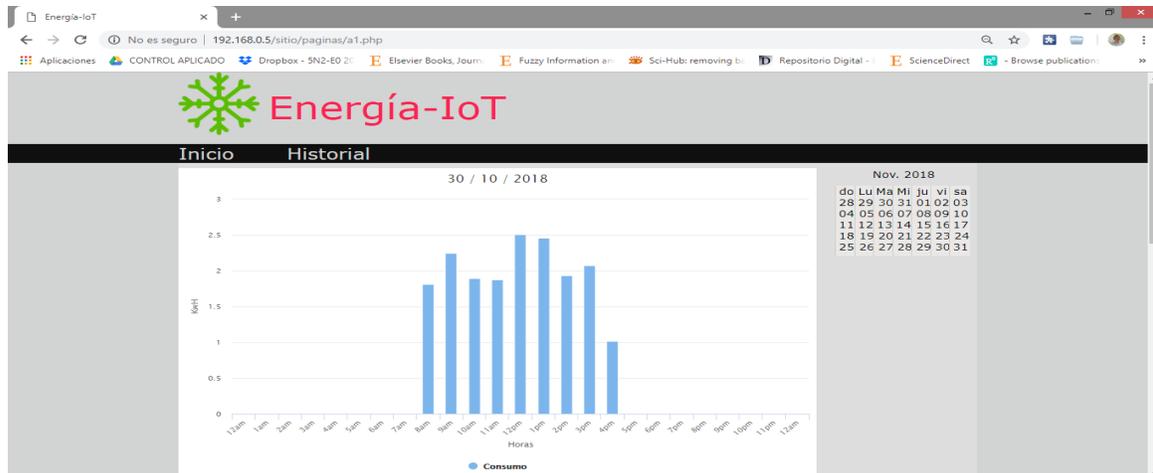


Figura 2-61. Ejemplo de gráficos-n. Fuente: Autor.

2.2.7 Sistema de alimentacion

La fuente de alimentación es el elemento dentro del sistema que se encarga de suministrar los niveles de voltaje y corriente requeridos. En la Tabla 2-27 se realiza una descripción de los niveles de corriente y de voltaje necesarios. Según la Tabla se requiere de una corriente máxima de 84mA y un voltaje de 3.3V.

Debido a su bajo costo, y simplicidad de diseño, se decidió utilizar un adaptador 120VAC-DC5V estándar como fuente de alimentación. Este adaptador entrega 5V y hasta 600mA, lo cual es más que suficiente para esta aplicación. Este adaptador es mostrado en la Fig. 2-62. Sin embargo, se requiere de 3.3V para la operatividad de este sistema, debido a esto se incorporó un regulador de voltaje de 3.3V al sistema de alimentación. El diagrama esquemático de esta fuente es mostrado en la Fig. 2-63.

Tabla 2-27
Niveles de corriente y voltaje necesarios.

Elemento	Voltaje	Corriente
Microcontrolador	3.3V	4mA
LEDs (señalización)	3.3V	20mA
ESP01	3.3V	60mA
Total	3.3V	84mA



Figura 2-62. Adaptador 120AC-5VDC.
Fuente: Autor.

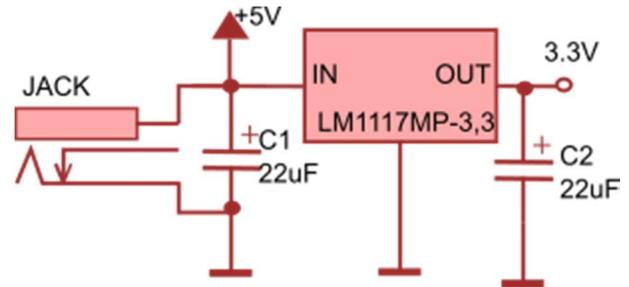


Figura 2-63. Esquemático fuente de alimentación.
Fuente: Autor.

2.2.8 Diseño en PCB.

Una vez finalizado el diseño de cada uno de los circuitos requeridos en el medidor de consumo, se llevó a cabo la integración de estos en una PCB. En este caso se utilizó únicamente una PCB. Esta contiene el sistema de control (18F14k50), circuitos de medición, señalización, y alimentación, pero también el módulo ESP01. A continuación, en la Fig. 2-64 se muestra el diseño total del medidor de consumo energético.

Por otro lado, el diseño de la PCB para el medidor de consumo es mostrado en las siguientes figuras. El Top-View es mostrado en la Fig. 2-65, mientras que el Bottom-View, es mostrado en la Fig. 2-66.

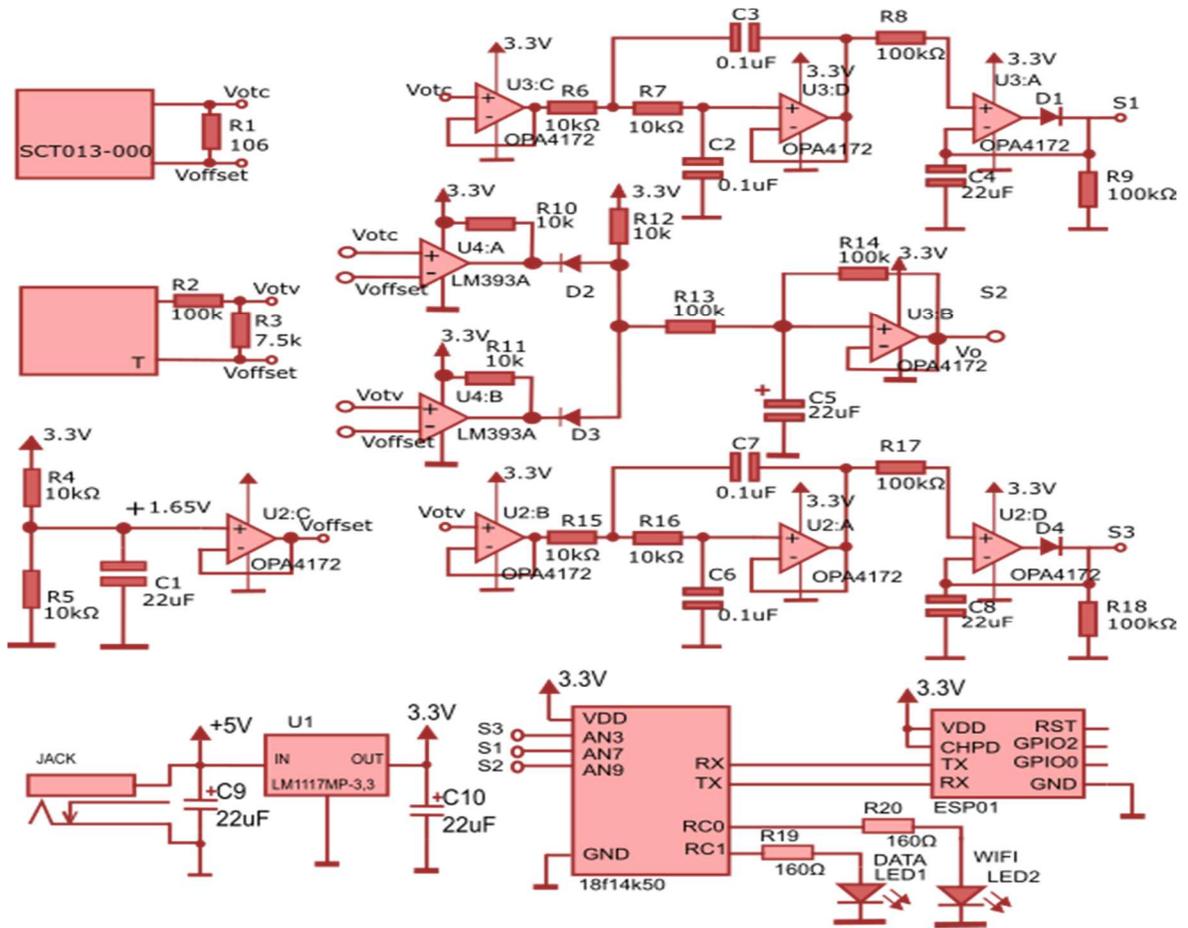


Figura 2-64. Diseño total del medidor de consumo energético. Fuente: Autor.

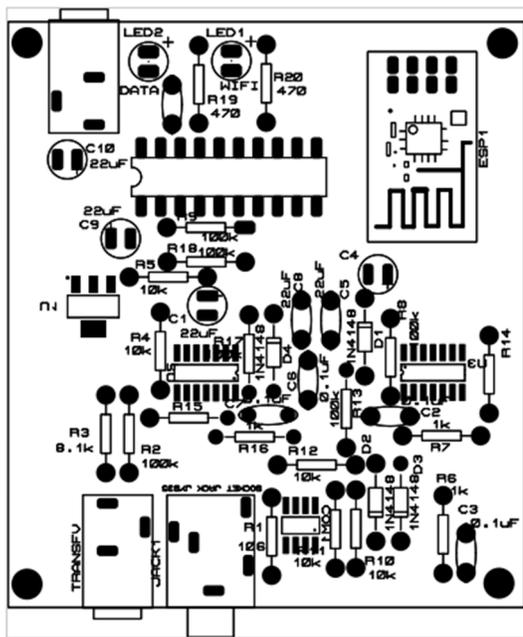


Figura 2-66. PCB Top-View. Fuente: Autor.

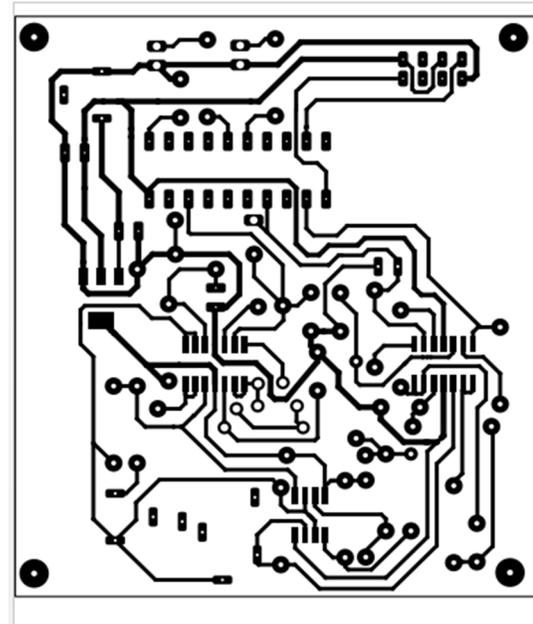


Figura 2-65. PCB Bottom-View. Fuente: Autor

2.2.9 Calibración del dispositivo.

Debido al error o alteraciones en los circuitos de acondicionamiento, fue necesario realizar modificaciones en las ganancias de calibración de los circuitos de acondicionamiento.

2.2.9.1 Calibración del sensor de voltaje.

La relación entre el voltaje de la línea, V_L , y el voltaje de salida del circuito de acondicionamiento, y leído por el controlador, S_3 , se encuentra determinada por la relación de espiras del transformador, N , y las resistencias del divisor de voltaje, R_2 y R_3 , utilizados para adaptar los niveles de voltaje al microcontrolador. Tal como lo describen las siguientes ecuaciones, se recomienda al lector ver nuevamente la Fig. 2-64.

$$V_{Lrms} = \frac{G_1 G_2}{\sqrt{2}} (S_3 - V_{offset}) \quad (2-47)$$

$$G_1 = N \quad (2-48)$$

$$G_2 = 1 + \frac{R_2}{R_3} \quad (2-49)$$

$$G = G_1 G_2 = 234.66 \quad (2-50)$$

$$V_{Lrms} = \frac{234.66}{\sqrt{2}} (S_3 - V_{offset}) \quad (2-51)$$

En la práctica los valores N , R_3 y R_2 , varían del valor esperado, debido a esto es necesario realizar un ajuste al valor de G , dentro de la programación del PIC. Esto con la finalidad corregir el error entre el voltaje medido con el dispositivo desarrollado, y el valor real. La Tabla 2-28, muestra el comportamiento del error, respecto al valor de G . Las mediciones realizadas con dispositivo desarrollado, fueron comparadas con las del multímetro YF-8030A de la marca YFE.

Tabla 2-28
Ajuste de G para la reducción del error en la medición de V_L .

G	VLrms(real)	VLrms(medido)	Error	%Error
234.66	213.50	182.58	30.92	0.14
300.00	213.50	230.70	-17.20	-0.08
280.00	213.50	213.81	-0.31	0.001

Como resultado, se decidió utilizar 280 para el valor de G, debido a que con este valor, el error es despreciable.

2.2.9.2 Calibración del sensor de corriente.

La relación entre la corriente de la línea, I_L , y el voltaje de salida del circuito de acondicionamiento, y leído por el controlador, S_1 , se encuentra determinada por la relación de espiras del transformador de corriente, N , y la resistencia R_1 , utilizados para adaptar los niveles de voltaje al microcontrolador. Tal como lo describen las siguientes ecuaciones, se recomienda al lector ver nuevamente la Fig. 2-64.

$$I_{Lrms} = \frac{G1}{G2\sqrt{2}} (S_1 - V_{offset}) \quad (2-52)$$

$$G1 = N \quad (2-53)$$

$$G2 = R1 \quad (2-54)$$

$$G = \frac{N}{R1} = 18.86 \quad (2-55)$$

$$I_{Lrms} = \frac{18.86}{\sqrt{2}} (S_1 - V_{offset}) \quad (2-56)$$

En la práctica los valores N y R_1 , varían del valor esperado, debido a esto es necesario realizar un ajuste al valor de G , dentro de la programación del PIC. Esto con la finalidad corregir el error entre la corriente medida con el dispositivo desarrollado, y el valor real. La Tabla 2-29, muestra el comportamiento del error, respecto al valor de G . Las mediciones realizadas con dispositivo desarrollado, fueron comparadas con las del multímetro **YF-8030A** de la marca **YFE**, y de esta manera determinar un valor adecuado para G .

Tabla 2-29

Ajuste de G para la reducción del error en la medición de I_L .

G	$I_{Lrms}(\text{real})$	$I_{Lrms}(\text{medido})$	Error	%Error
18.60	17.1	15.32	1.78	10.43%
21.00	17.1	16.41	0.69	4.02%
22.00	17.1	16.95	0.15	0.88%

Como resultado, se decidió utilizar 22.00 como valor para G , debido a que con este valor, el error es despreciable.

2.3 Pruebas y resultados.

2.3.1 Montaje de los dispositivos.

En esta fase del proyecto se realizó el montaje de los dispositivos, el medidor de consumo energético y controlador inteligente para aires acondicionados. Este montaje fue llevado a cabo en el laboratorio de circuitos eléctricos de la facultad de electrotecnia y computación de la UNI. Esto con el objetivo de realizar mediciones del consumo energético del aire acondicionado en este local que permitan comparar el consumo energético utilizado con el uso del controlador inteligente y sin este.

En las siguientes figuras se muestra el montaje de los dispositivos. Primeramente las figuras 2-67 y 2-68, muestran el montaje del medidor energético IoT. Este requiere la conexión de transformador utilizado para medición del voltaje en las fases de alimentación del AA (220vrms). Además, se requiere de la conexión de SCT-00013, un transformador de corriente no invasivo. Luego, estos sensores son conectados a la tarjeta del medidor, y únicamente se requiere la alimentación de este dispositivo para que este empiece a realizar mediciones.



Figura 2-68. Montaje de medidor energético IoT a la alimentación del aire acondicionado (Toma A).

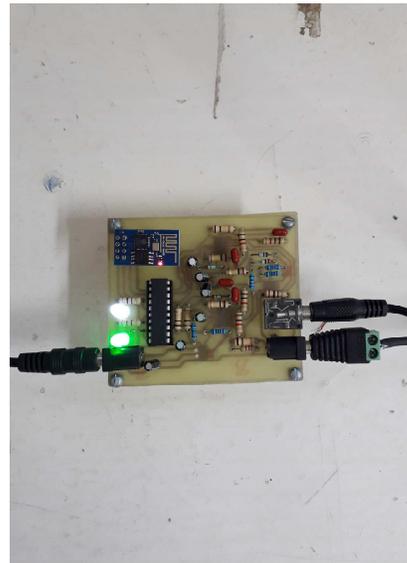


Figura 2-67. Montaje de medidor energético IoT a la alimentación del aire acondicionado (Toma B).

En lo que respecta al controlador inteligente, su montaje se realiza directamente en la unidad de climatización, sobre el receptor infrarrojo integrado en esta unidad. En este caso, este dispositivo requiere de la conexión de la tarjeta del sensor de presencia y su respectiva alimentación. En las figuras 2-69 y 2-70 se muestra el montaje de este dispositivo.



Figura 2-70. Montaje del controlador inteligente en el aire acondicionado del laboratorio de circuitos de FEC-UNI (Vista A). Fuente: Autor.



Figura 2-69. Montaje del controlador inteligente en el aire acondicionado del laboratorio de circuitos de FEC-UNI. (Vista B). Fuente: Autor.



Figura 2-72. Montaje del controlador inteligente en el aire acondicionado del laboratorio de circuitos de FEC-UNI. (Vista C). Fuente: Autor.

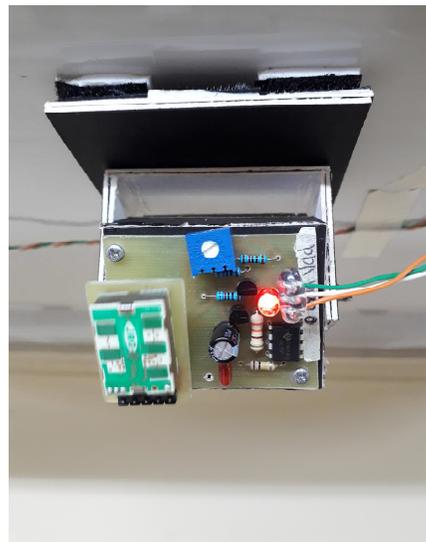


Figura 2-71. Montaje del controlador inteligente en el aire acondicionado del laboratorio de circuitos de FEC-UNI. (Vista D). Fuente: Autor.

2.3.2 Descripción de las pruebas.

Por otro lado, las mediciones del consumo energético fueron llevadas a cabo en el mes de noviembre del año 2018. Las cuales no se realizaron de forma continua, sino de forma discreta en ciertos días del mes noviembre, esto debido a que algunos días fueron utilizados para realizar ajuste a los dispositivos y razones administrativas de la universidad. Sin embargo, se realizaron mediciones durante 20 días, 10 sin el uso de controlador inteligente, y 10 con el uso de este, esto para existiera una simetría en número de días. En cada uno de estos días se observó el consumo de energía de las 8am a las 4pm. A continuación, en la Fig. 2-73 se presenta un calendario de los días en los que realizaron mediciones.

Noviembre del 2018

do	lu	ma	mi	ju	vi	sá
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	1

■ Controlador ■ Sin-Controlador

Figura 2-73. Calendario de mediciones de consumo energético. Fuente: Autor.

El medidor IoT realiza las mediciones de la energía utilizada y luego las envía al servidor web, el cual se encarga de almacenarlas en la base de datos-se recomienda al lector revisar la sección 1.11, donde encontrara más información sobre el funcionamiento del sistema web. En total se utilizaron dos base datos y 20 tablas, 10 por cada tipo de prueba, es decir, con o sin controlador. La primera nombrada “consu_antes”, utilizada para las mediciones sin controlador, y la segunda nombrada “consu_despues”, utilizada para las mediciones con el controlador inteligente. En la Fig. 2-74 se muestra la estructura de las bases utilizadas. El usuario puede consultar estas bases de datos mediante el sitio web desarrollado, véase la sección 1.11.

```

|--consu_antes:
|--dia1( id (int), fecha (timestamp), kjulios (float))
|--dia2( id (int), fecha (timestamp), kjulios (float))
|--dia3( id (int), fecha (timestamp), kjulios (float))
|--dia4( id (int), fecha (timestamp), kjulios (float))
|--dia5( id (int), fecha (timestamp), kjulios (float))
|--dia6( id (int), fecha (timestamp), kjulios (float))
|--dia7( id (int), fecha (timestamp), kjulios (float))
|--dia8( id (int), fecha (timestamp), kjulios (float))
|--dia9( id (int), fecha (timestamp), kjulios (float))
|--dia10( id (int), fecha (timestamp), kjulios (float))
|--consu_despues:
|--dia1( id (int), fecha (timestamp), kjulios (float))
|--dia2( id (int), fecha (timestamp), kjulios (float))
|--dia3( id (int), fecha (timestamp), kjulios (float))
|--dia4( id (int), fecha (timestamp), kjulios (float))
|--dia5( id (int), fecha (timestamp), kjulios (float))
|--dia6( id (int), fecha (timestamp), kjulios (float))
|--dia7( id (int), fecha (timestamp), kjulios (float))
|--dia8( id (int), fecha (timestamp), kjulios (float))
|--dia9( id (int), fecha (timestamp), kjulios (float))
|--dia10( id (int), fecha (timestamp), kjulios (float))

```

Figura 2-74. Estructura de las bases de datos utilizadas durante las mediciones. Fuente: Autor.

En la figura anterior “id” es un número que identifica cada muestra o medición dentro de una tabla. Por su parte, “fecha”, contiene la hora y fecha en la que es carga cada una de las mediciones. Por último, en lo que respecta a “kjulios” contiene el valor de la muestra en kilojulios, esto debido a que kwh es una unidad muy grande para el consumo energético del aire acondicionado en pequeños intervalos de tiempo.

2.3.3 Presentación de resultados.

En la Tabla 2-30 se presentan el consumo energético total por día obtenido durante la fase de pruebas. En esta tabla se especifica en la fecha en la que se llevó a cabo la medición, y el respectivo valor de la medición. Por otro lado, en la Fig. 2-75 también se muestra el consumo de energía, pero además se especifica en cuales se utilizó el controlador inteligente y en cuáles no.

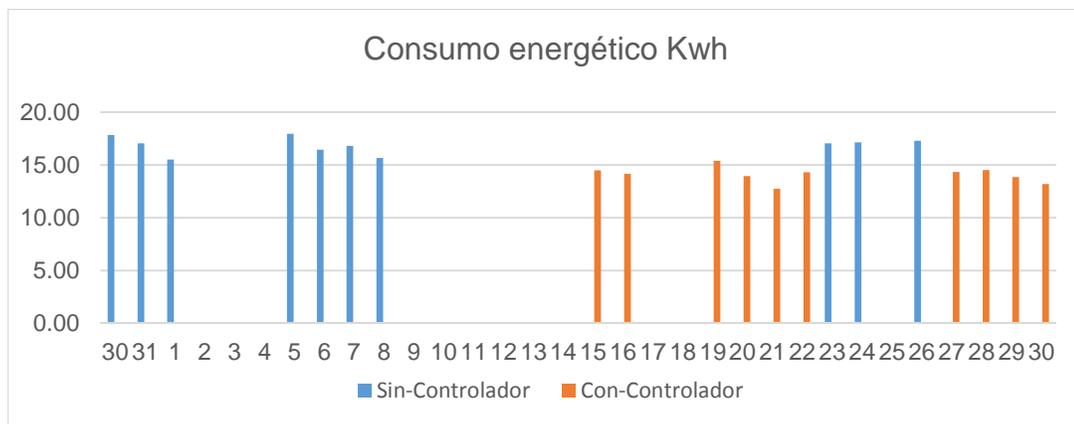


Figura 2-75. Consumo energético, Noviembre 2018. Fuente: Autor.

Tabla 2-30
Mediciones realizadas, Noviembre 2018.

Fecha	Kwh	Fecha	Kwh
30/10/2018	17.82	15/11/2018	14.47
31/10/2018	17.04	16/11/2018	14.16
01/11/2018	15.49	17/11/2018	-
02/11/2018	-	18/11/2018	-
03/11/2018	-	19/11/2018	15.39
04/11/2018	-	20/11/2018	13.95
05/11/2018	17.94	21/11/2018	12.74
06/11/2018	16.42	22/11/2018	14.31
07/11/2018	16.80	23/11/2018	-
08/11/2018	15.65	24/11/2018	-
09/11/2018	-	25/11/2018	-
10/11/2018	-	26/11/2018	-
11/11/2018	-	27/11/2018	14.32
12/11/2018	-	28/11/2018	14.50
13/11/2018	-	29/11/2018	13.84
14/11/2018	-	30/11/2018	13.17

En total, el consumo de energía utilizado por el aire acondicionado en los 10 días de prueba sin controlador fue de 168.60kwh. Mientras que con el controlador fue de 140.85kwh, como lo indica la Tabla 2-31. Esto significa que hubo un ahorro de 27.75kwh, es decir, un ahorro de 16.46% de energía.

Tabla 2-31
Mediciones realizadas, Noviembre 2018.
(Resumen)

kwh Sin-Controlador	kwh Con-controlador
168.60	140.85

Capítulo III: Conclusiones y recomendaciones

3.1 Conclusiones

Este trabajo monográfico cumplió con las expectativas planteadas en sus objetivos. Se desarrolló un controlador inteligente para la configuración automática de un aire acondicionado a partir de las condiciones en un ambiente interior. En el proceso se realizó un diseño preliminar de los subsistemas y se llevó cabo un estudio técnico de las tecnologías para su implementación, se realizaron simulaciones y pruebas a nivel de breadboard, se realizó la integración de los subsistemas y se implementaron sus PCBs correspondientes. También se desarrolló un medidor de consumo energético IoT para evaluar la efectividad del controlador inteligente.

En lo que respecta a los resultados obtenidos de las pruebas realizadas, estos fueron satisfactorios. Tanto el controlador inteligente, como el medidor energético IoT cumplieron su funcionalidad. El controlador inteligente permite la configuración automática del AA, basándose en un algoritmo fuzzy, y partiendo de las condiciones interiores en la habitación. Por su parte, el medidor energético IoT, realiza las mediciones de corriente y voltaje en una carga para determinar el consumo energía utilizada, consumo que luego es almacenado en una base de datos.

Las mediciones obtenidas proyectan que este tipo de controladores inteligentes para AA, pueden tener un impacto positivo en materia de ahorro energético, esto debido a que se logró disminuir el consumo energético en un 16.46% durante las pruebas. A gran escala, la incorporación de este dispositivo en las diferentes instancias de la universidad, como por ejemplo los laboratorios de la FEC-UNI, su impacto su impacto sería mayor.

Aunque los resultados de este prototipo son satisfactorios, deja abierta la posibilidad a nuevas investigaciones de este tipo, y también mejoras al sistema, a continuación se sugieren algunas en la sección de recomendaciones.

3.2 Recomendaciones

- Mejorar el sistema de detección de presencia. Se pueden utilizar sensores térmicos, estos permiten la detección de presencia de personas en reposo. Sin embargo, debe tomarse en consideración que la utiliza de este tipo de sensores requiere de conocimientos en reconocimiento de imágenes.
- Se puede incorporar un RTC al controlador para que el usuario pueda definir y configurar los horarios en los se puede utilizar el AA.
- Para tener una mejor estimación del ahorro energético, es necesario ampliar el tiempo de observación utilizando el controlador y sin este.

Referencias

- Alcalá, R., Cordon, O. & Herrera, F. (2002). Algoritmos genéticos para el ajuste de parámetros y selección de reglas en el control difuso de un sistema de climatización. Recuperado de <https://www.researchgate.net/publication/228543752>
- Altium. (2017). Infrared Communication Concepts. NEC infrared transmission protocol. Recuperado de: <http://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol>
- Álvarez, M. (2001). Que es HTML. Disponible en: <https://desarrolloweb.com/articulos/que-es-html.html>
- Apache HTTP Server Project. (2018). Acerca de Apache. Disponible en: http://httpd.apache.org/ABOUT_APACHE.html
- Arnabat, I. (2016). Cómo funciona el aire acondicionado? Infografía. CALORYFRIO. Recuperado de: <https://www.caloryfrio.com/aire-acondicionado/como-funciona-el-aire-acondicionado-infografia.html>
- Baca, G. (2013). Evaluación de proyectos. Disponible en: <https://ianemartinez.files.wordpress.com/2012/09/evaluacion-de-proyectos-gabriel-baca-urbina-corregido.pdf>
- Castro, A. (2014). “¿Qué es Raspberry Pi, dónde comprarla y cómo usarla?”. Disponible en: <https://computerhoy.com/noticias/hardware/que-es-raspberry-pi-donde-comprarla-como-usarla-8614>
- Charles, A. & Sadiku, M. (2006). Fundamentos de circuitos eléctricos. México D.F., México: McGraw-Hill.
- Chávez, F. (2002). Zona variable de confort Térmico. (Tesis doctoral, Escuela Politécnica de Cataluña, Cataluña, España). Obtenido de: <https://upcommons.upc.edu/handle/2117/93416>
- Clavijo, J. (2011). Diseño y simulacion de sistemas microcontrolados en lenguaje C. Colombia.
- EsepeStudio. (2005). “¿Qué es MySQL?”. Disponible en: <https://www.espestudio.com/noticias/que-es-mysql>
- Espressif Systems. (2015). ESP8266EX datasheet. Disponible en: https://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf
- Glaría, J. & Kouro, S. (2001). Automatización industrial: sensores de humedad. Universidad Técnica Federico Santa María, departamento de electrónica. Disponible en: <http://ingeborda.com.ar/biblioteca/Biblioteca%20Internet/Articulos%20Tecnicos%20de%20Consulta/Instalaciones%20Electricas%20Industriales/Sensores%20de%20Humedad.pdf>
- González, C. (2015). Aplicaciones orientadas a la domótica con Raspberry Pi. Escuela técnica superior de ingeniería, universidad de Sevilla. Tesis de grado. Disponible en: <http://bibing.us.es/proyectos/abreproy/90330/fichero/Memoria+TFG.pdf>
- Grokhotkov, I. (2017). ESP8266WiFi library. Disponible en: <https://arduino-esp8266.readthedocs.io/en/latest/esp8266wifi/readme.html>
- Hernandes, L. (s.f. b). Guía para configurar un ESP-01, el módulo WiFi basado en ESP8266. Promarfacil.com, disponible en: <https://programarfacil.com/podcast/como-configurar-esp01-wifi-esp8266/>
- Hernández, L. (s.f. a) NodeMCU tutorial paso a paso desde cero. programarfacil.com, disponible en: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>
- INETER. (2012). Clima de Mangua. Datos climáticos: 1942 – 2017. Disponible en: <https://www.tutiempo.net/clima/ws-787410.html>
- Irwin, J. & Nelms, R. (2011). Basic engineering circuit analysis. U.S.A: John Wiley & Sons, Inc.
- Katsuhiko, O. (2010). Ingeniería de control Moderna. Madrid, España: PEARSON-EDUCACION.
- Kiss, Z. (2015). Métodos de detección de movimiento: comparativa de la tecnología sensores PIR de Nicera, WAVEEYE de NJRC y GrIDEYE de Panasonic. Revista Española de electrónica. 733, 50-55. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=5515813>
- Maguiña, R. (2010). Sistema de inferencia basada en lógica borrosa: fundamentos y caso de estudio. Revista de investigación de sistemas e ingeniería, 7(1), 91-104. Disponible en: <http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/viewFile/3270/2733>
- Martin, G. (s.f.). Cómo programar NodeMCU con el IDE de Arduino. Programarfacil.com, disponible en: <https://programarfacil.com/esp8266/como-programar-nodemcu-ide-arduino/>
- MDN web docs. (2015c). “¿Qué es JavaScript?”. Disponible en:

- MDN web docs. (2017a). Lista de elementos HTML5. Disponible en: https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos
- MDN web docs. (2018b). CSS. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
- Michalski, S. (2009). Humedad relativa incorrecta. Canadian Conservation Institute. Disponible en: http://www.cncr.cl/611/articles-56474_recurso_10.pdf
- Passino, K. & Yurkovich, S. (1998). Fuzzy control. Menlo Park, California: Addison Wesley.
- Paul, J. (1993). Embedded real-time systems, a specification and design methodology. Nantes, France: IRESTE.
- Pino, J. (f.s.). "¿Qué es HTML?". Disponible en: <https://devcode.la/blog/que-es-html/>
- Rosón, V. (2016). La humedad relativa y sus implicaciones sobre la salud y el confort. Mundo HVACR. Disponible en: <https://www.mundohvacr.com.mx/mundo/2016/01/la-humedad-relativa-y-sus-implicaciones-sobre-la-salud-y-el-confort/>
- The PHP Group. (2018). "¿Qué es PHP?". Disponible en: <http://php.net/manual/es/intro-what-is.php>
- Tsai, C. & Young, M. (2013). Pyroelectric infrared sensor-based thermometer for monitoring indoor objects. Review of scientific instruments. 74 (12) ,5267–5273. doi:10.1063/1.1626005.
- Whitman, W. & Johnson, W. (2000). Tecnología de refrigeración y aire acondicionado. Disponible en: <https://es.scribd.com/doc/161210339/Whitman-Tecnologia-de-La-Refrigeracion-y-Aire-Acondicionado-Fundamentos-Tomo-1>
- Young, H. & Freedman, R. (2009). Física universitaria volume 1. Mexico: Pearson Education.
- Ahamed, U., Bin, Z. & Bin, I. (2016). Fuzzy logic controller design for intelligent air-conditioning system. 2016 2da International Conference on Control Science and Systems Engineering. IEEE, Singapur, Singapur. DOI: 10.1109/CCSSE.2016.7784388
- Castilla, M., Álvarez, J., Berenguel, M., Pérez, M., Rodríguez, F. & Guzmán, J. (2010). Técnicas de control del confort en edificios. Revista Iberoamericana de automática e informática, 7(3), 5-24. DOI: 10.4995/RIAI.2010.03.01
- Mendieta, M. (2014). La energía Solar: una alternativa para el desarrollo sostenible de los diversos sectores de Nicaragua. (Tesis de maestría, Universidad Centro Americana, Managua, Nicaragua). Recuperado de: <http://repositorio.uca.edu.ni/id/eprint/2331>
- Presidencia de la republica de Nicaragua. (30 de enero de 2008). Ordenamiento del uso de la energía. [DECRETO No. 2-2008]. DO: La Gaceta No. 22 del 31 de enero del 2008. Recuperado de: <http://www.mem.gob.ni/wp-content/uploads/2017/05/3-decreto-2-2008-ordenamiento-del-suso-de-la-energia.pdf>

Anexos

Entrevista Responsables de laboratorios de electrónica FEC-UNI

Nombres y apellido: Ulises Fajal Rivas Garcia

Responsable de laboratorio de Simulación FEC.

Fecha: 8 / 03 / 2018

¿Has experimentado condiciones de discomfort, ya sea frío o calor, en el ambiente climatizado aun cuando ya has configurado el aire acondicionado?

- a) SI
- b) NO

¿Cuál consideras que sea la causa de ese discomfort?

Configuración de una temperatura muy baja en el aire acondicionado

¿Cuántas veces realizas la configuración de la temperatura del aire acondicionado?

- a) 1
- b) 2
- c) 3
- d) Más de tres

¿Consideras necesario la incorporación de un dispositivo que se encargue de la configuración automática del aire acondicionado?

- a) SI
- b) NO

Si respondiste SI a la pregunta anterior, ¿Cuáles consideras que deben ser las características y funcionalidades que debe tener dicho dispositivo?

- 1- Debe ser pequeño y fácil instalación.
- 2- Debe encargarse de la temporización de la climatización.
- 3- El dispositivo debe permitir consultar las mediciones realizadas utilizándolo un Smartphone.

Entrevista Responsables de laboratorios de electrónica FEC-UNI

Nombres y apellido: Genes Tatiana Alvarez.

Responsable de laboratorio de Circuitos Elec. FEC.

Fecha: 9 / 03 / 2018

¿Has experimentado condiciones de discomfort, ya sea frío o calor, en el ambiente climatizado aun cuando ya has configurado el aire acondicionado?

- a) SI
- b) NO

¿Cuál consideras que sea la causa de ese discomfort?

Una mala configuración en el termostato.

¿Cuántas veces realizas la configuración de la temperatura del aire acondicionado?

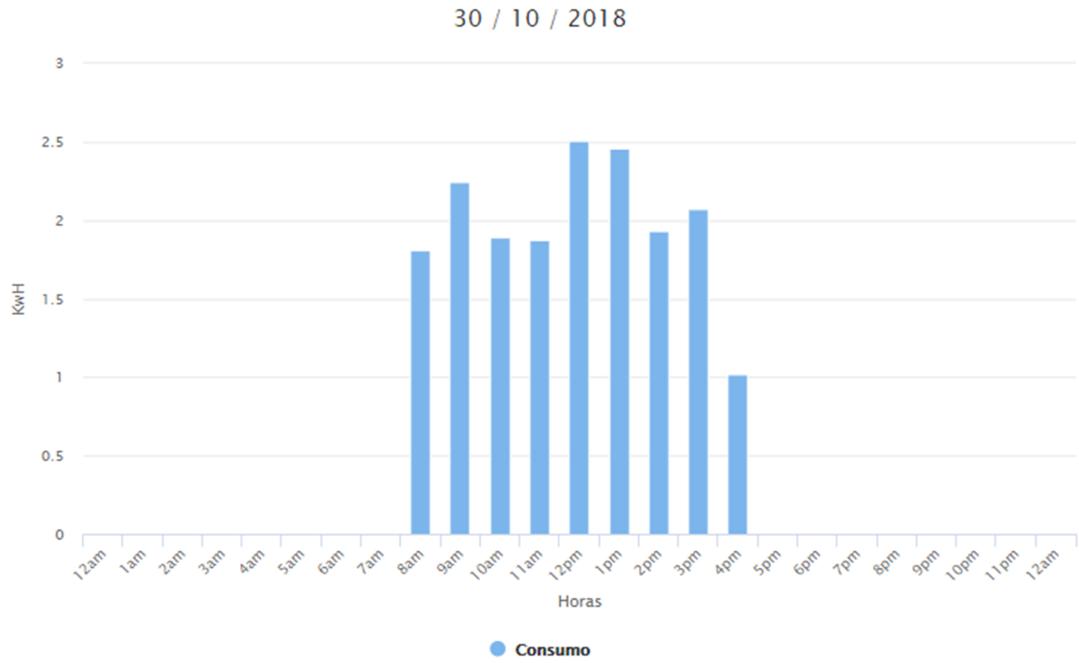
- a) 1
- b) 2
- c) 3
- d) Más de tres

¿Consideras necesario la incorporación de un dispositivo que se encargue de la configuración automática del aire acondicionado?

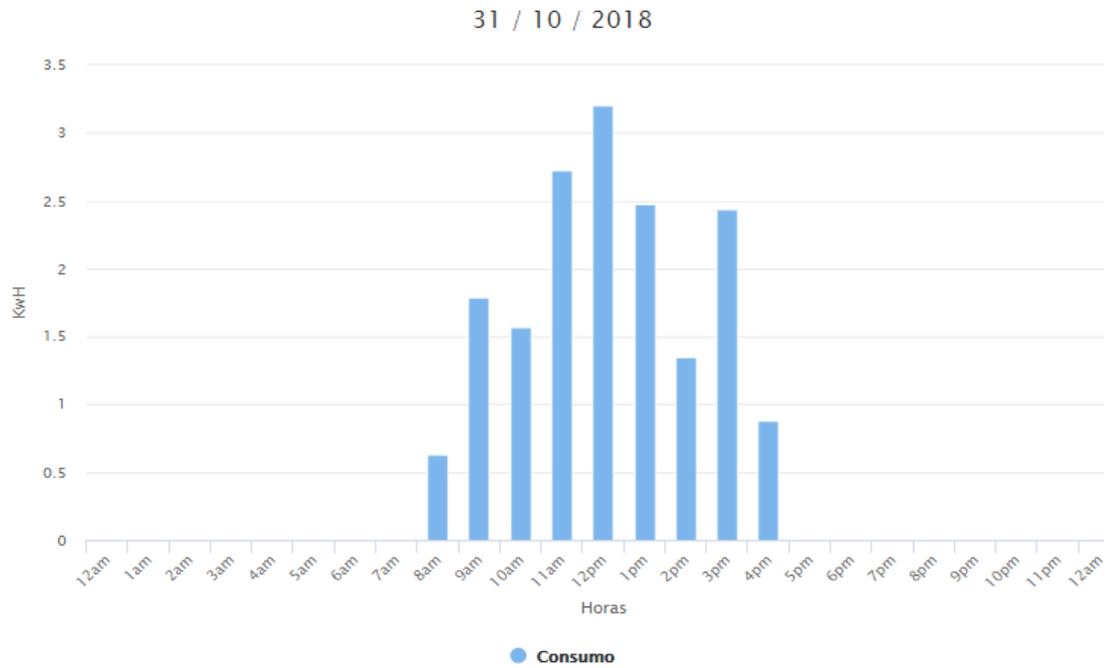
- a) SI
- b) NO

Si respondiste SI a la pregunta anterior, ¿Cuáles consideras que deben ser las características y funcionalidades que debe tener dicho dispositivo?

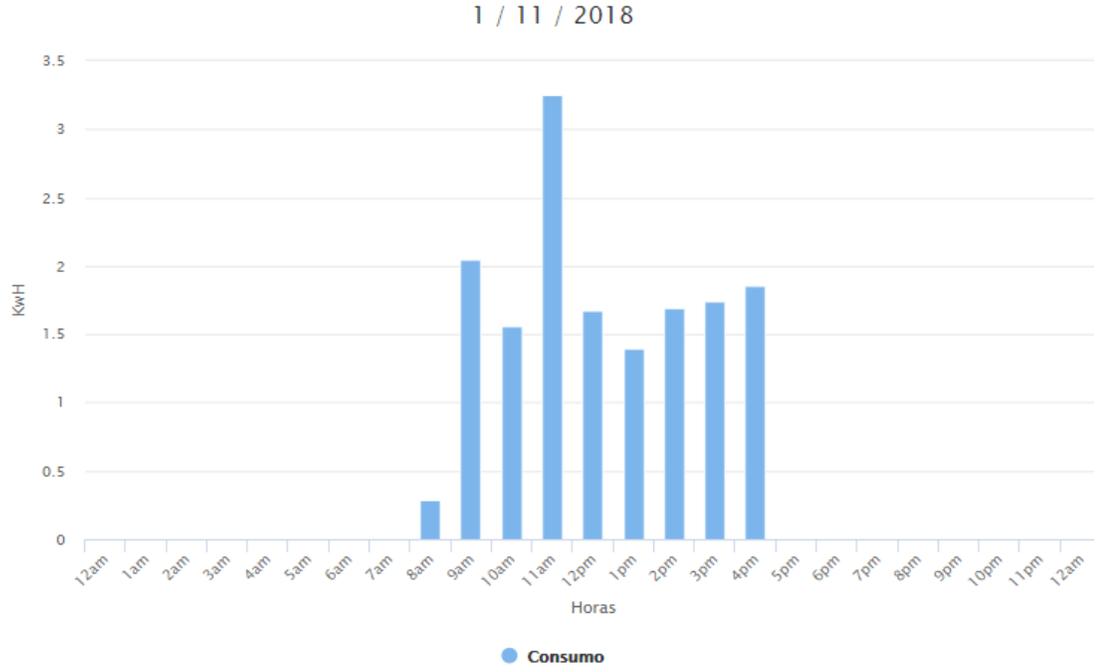
- 1- debe realizar el control del aire acondicionado
como el control remoto, es decir, mediante comunicación infrarroja.
- 2- es necesario la incorporación de sensores de
presencia
- 3- No debe utilizar baterías.



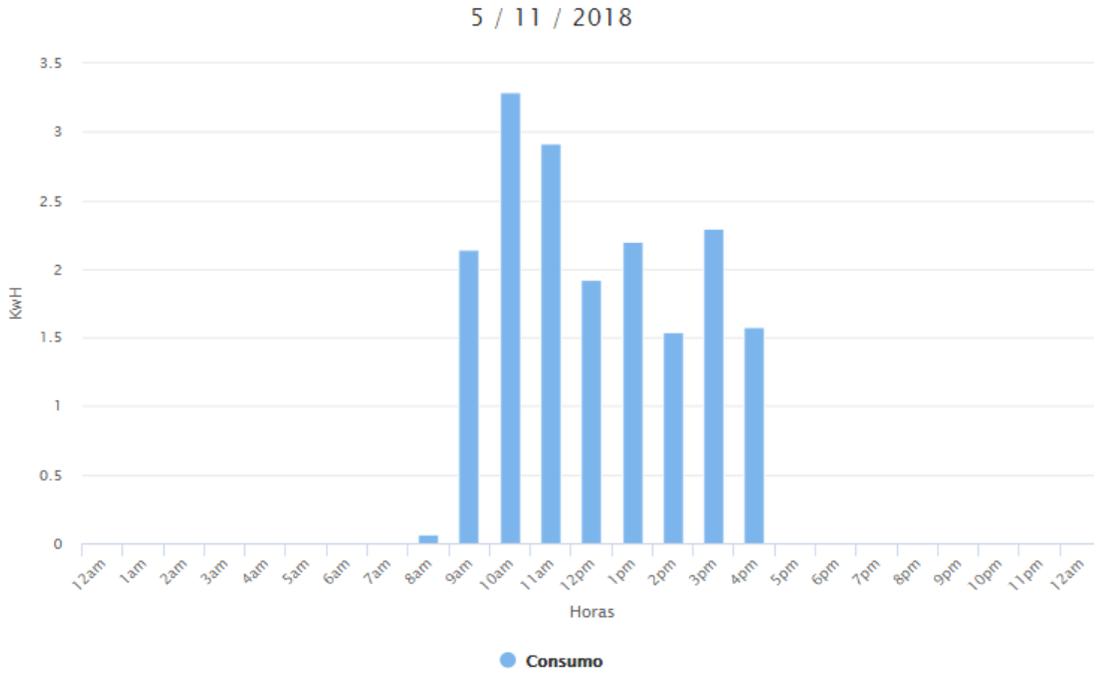
Anexo 3. Consumo energético del AA 30/10/2018, (sin controlador).



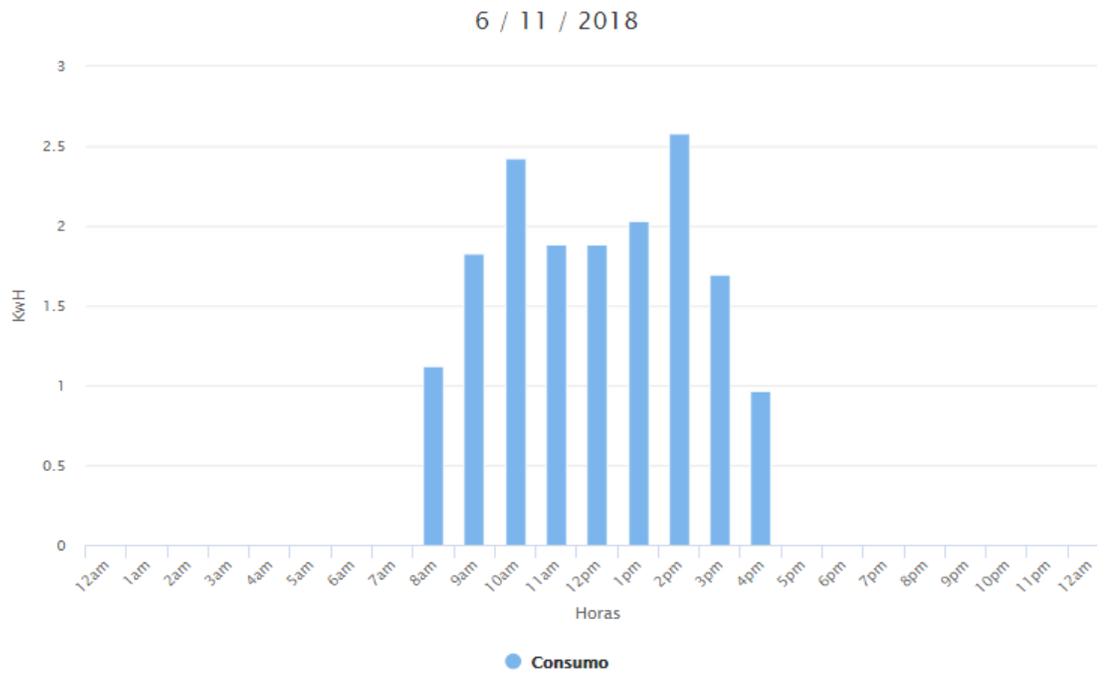
Anexo 4. Consumo energético del AA 31/10/2018, (sin controlador).



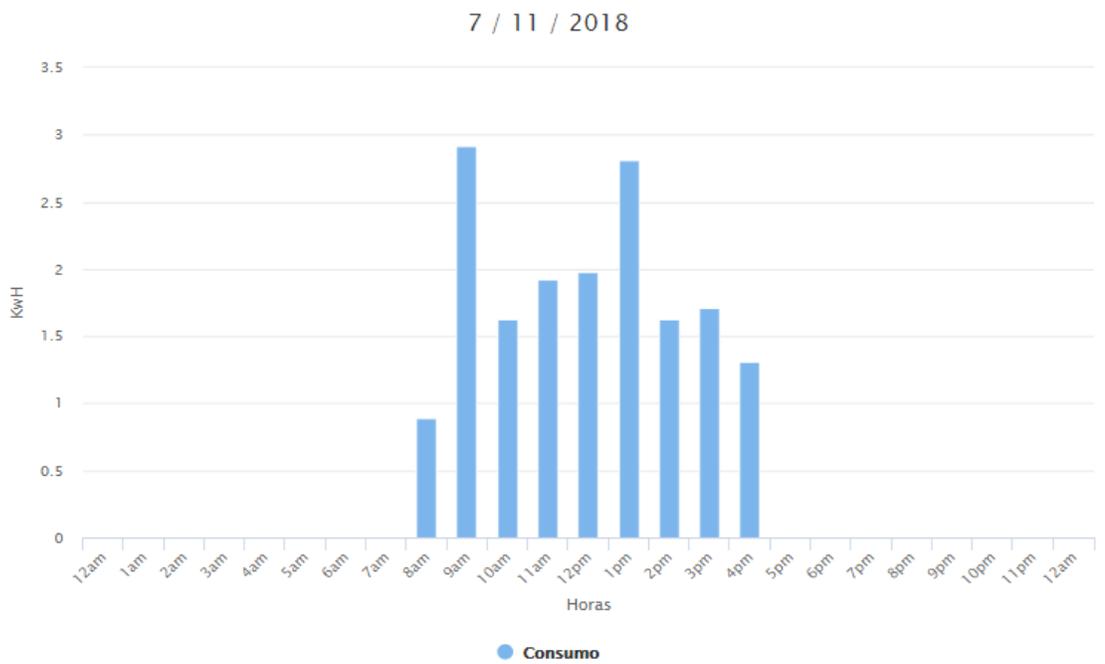
Anexo 5. Consumo energético del AA 01/11/2018, (sin controlador).



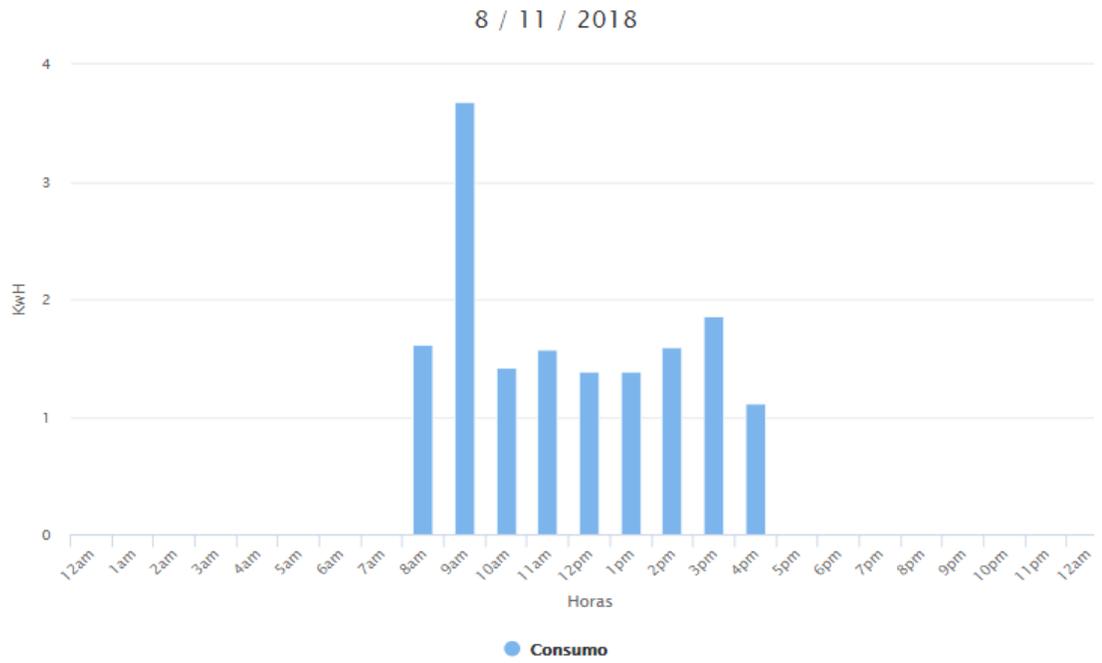
Anexo 6. Consumo energético del AA 05/11/2018, (sin controlador).



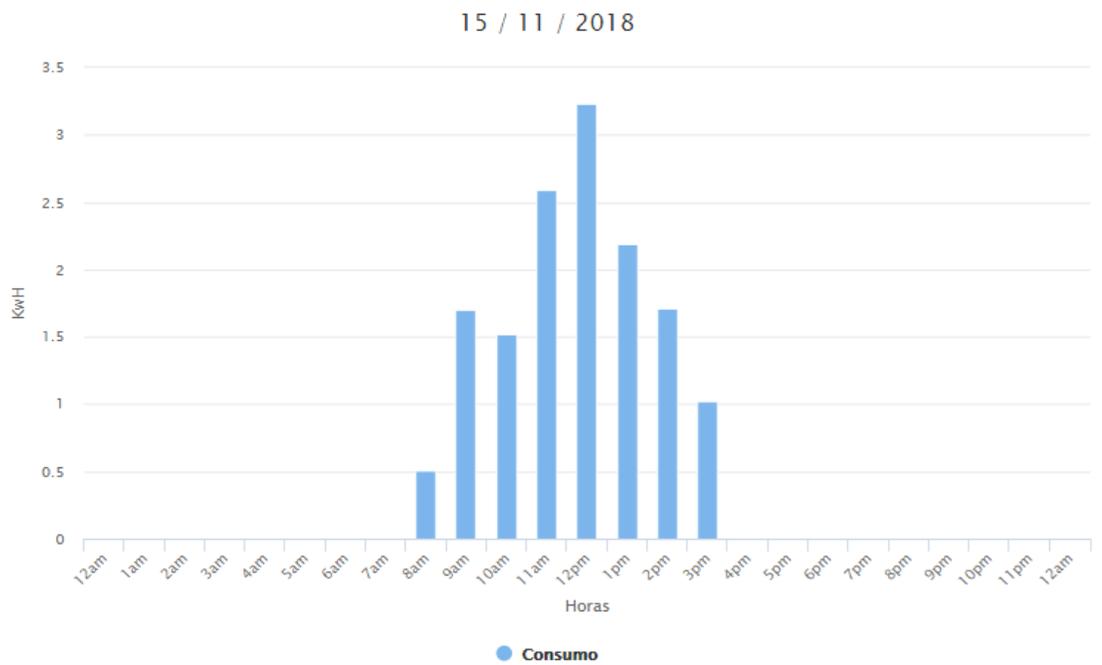
Anexo 7. Consumo energético del AA 06/11/2018, (sin controlador).



Anexo 8. Consumo energético del AA 07/11/2018, (sin controlador).

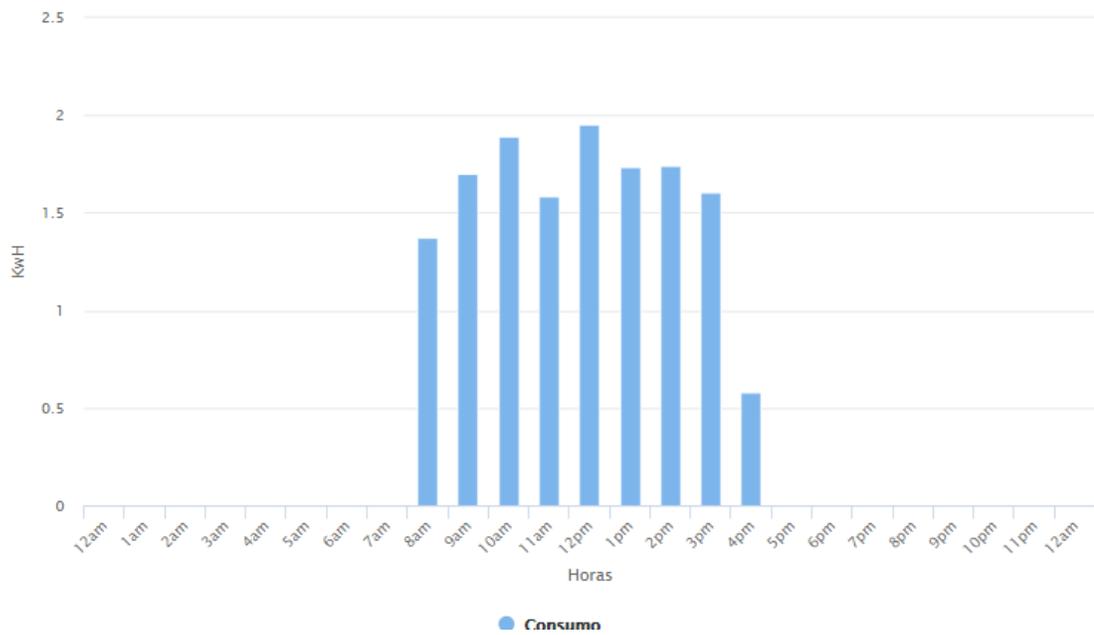


Anexo 9. Consumo energético del AA 08/11/2018, (sin controlador).



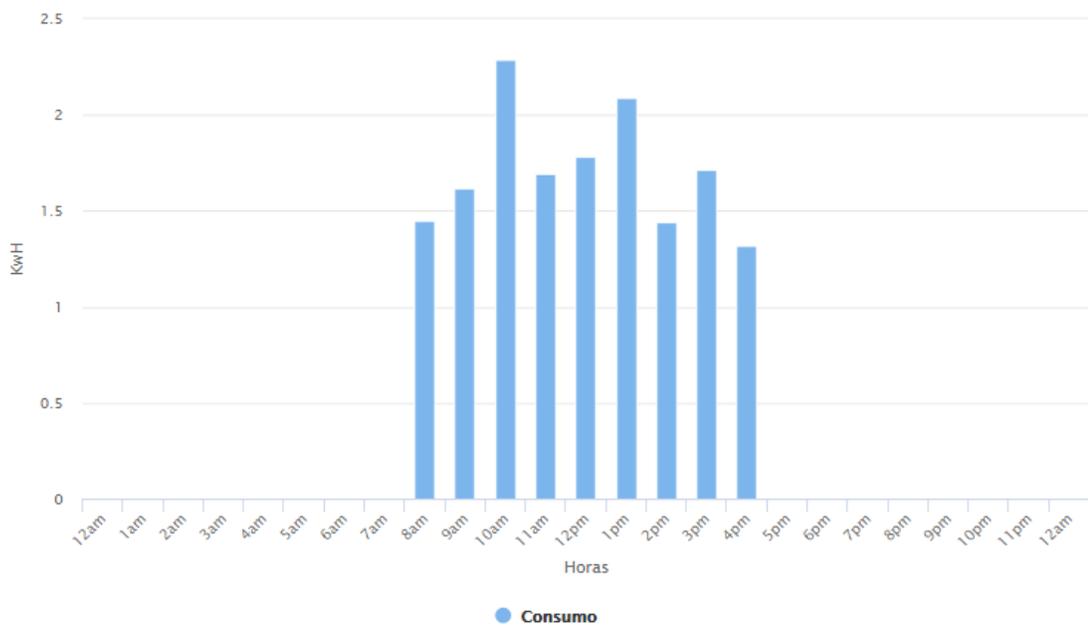
Anexo 10. Consumo energético del AA 15/11/2018, (Con controlador).

16 / 11 / 2018

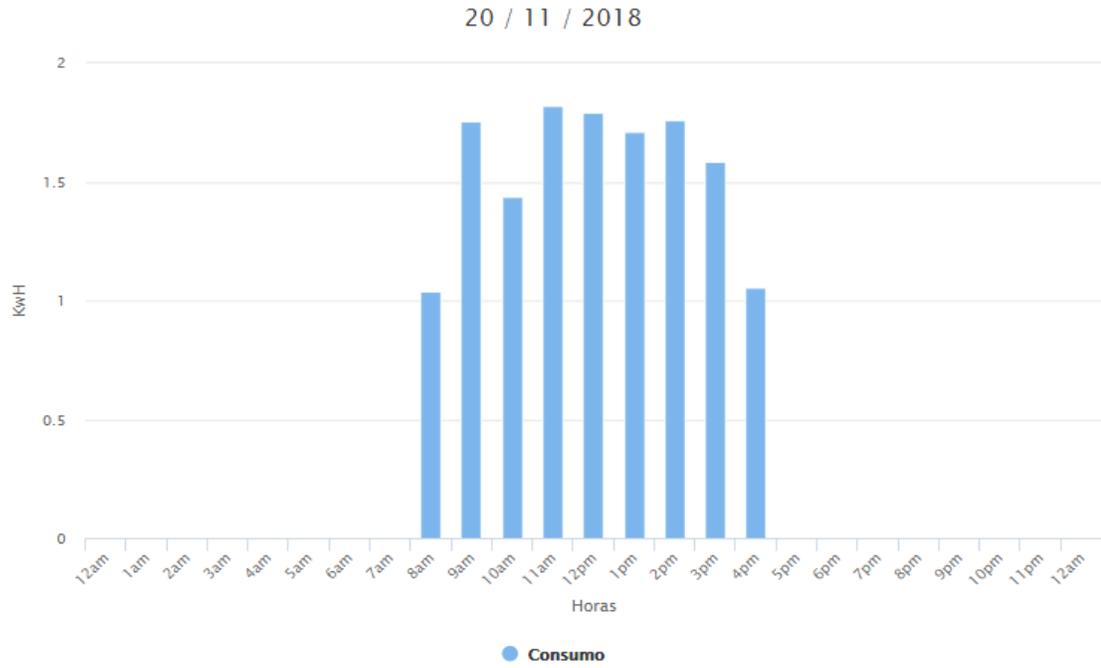


Anexo 11. Consumo energético del AA 16/11/2018, (Con controlador).

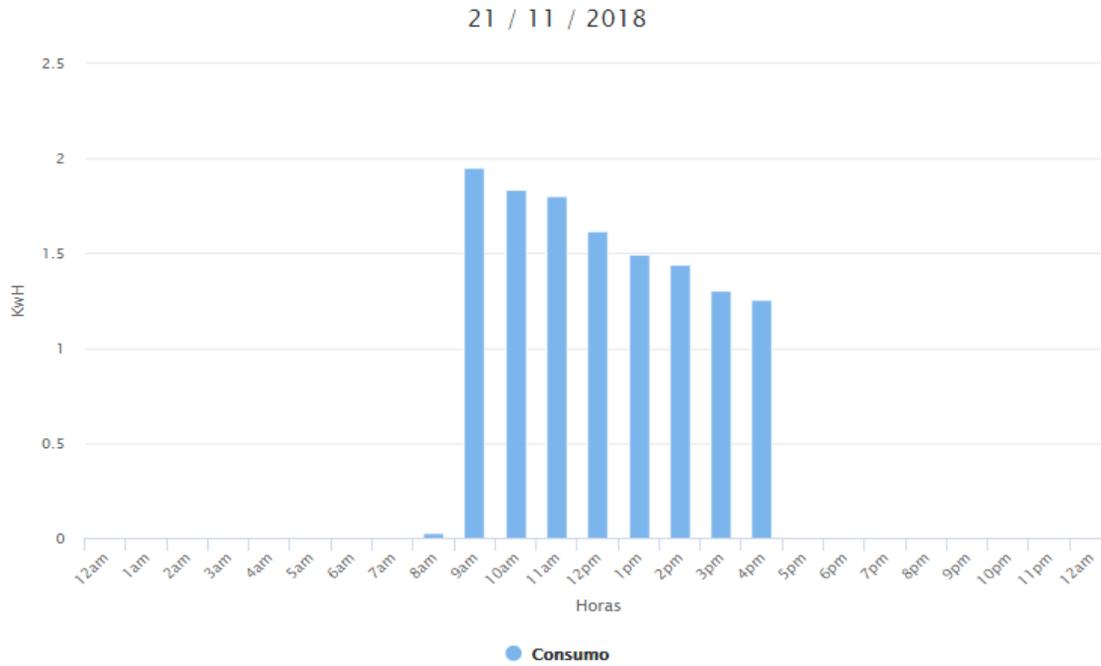
19 / 11 / 2018



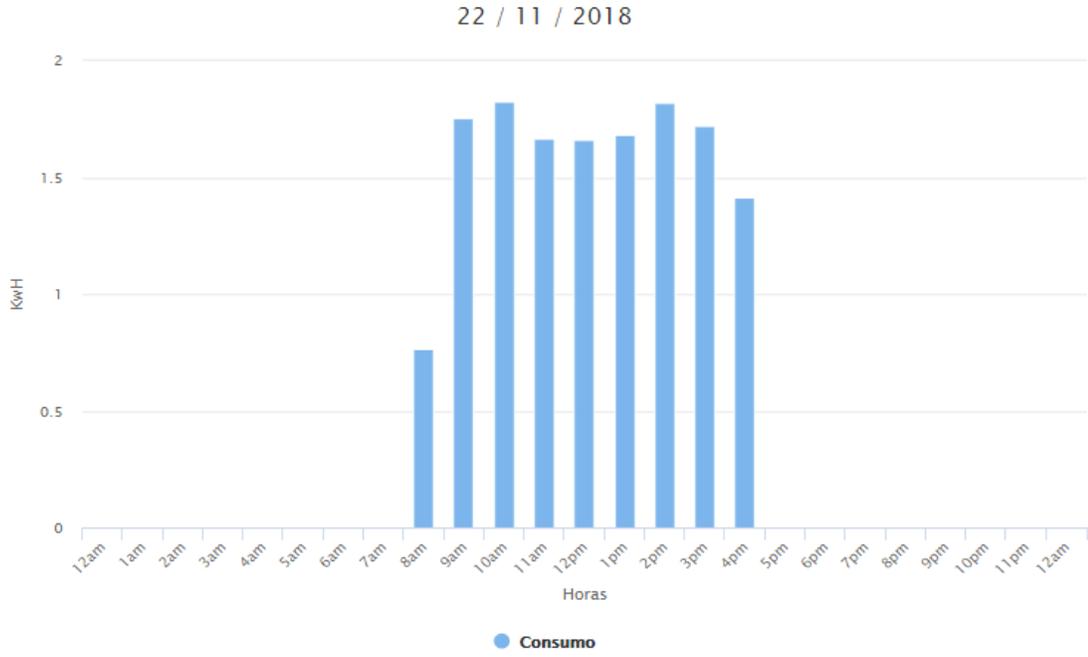
Anexo 12. Consumo energético del AA 19/11/2018, (Con controlador).



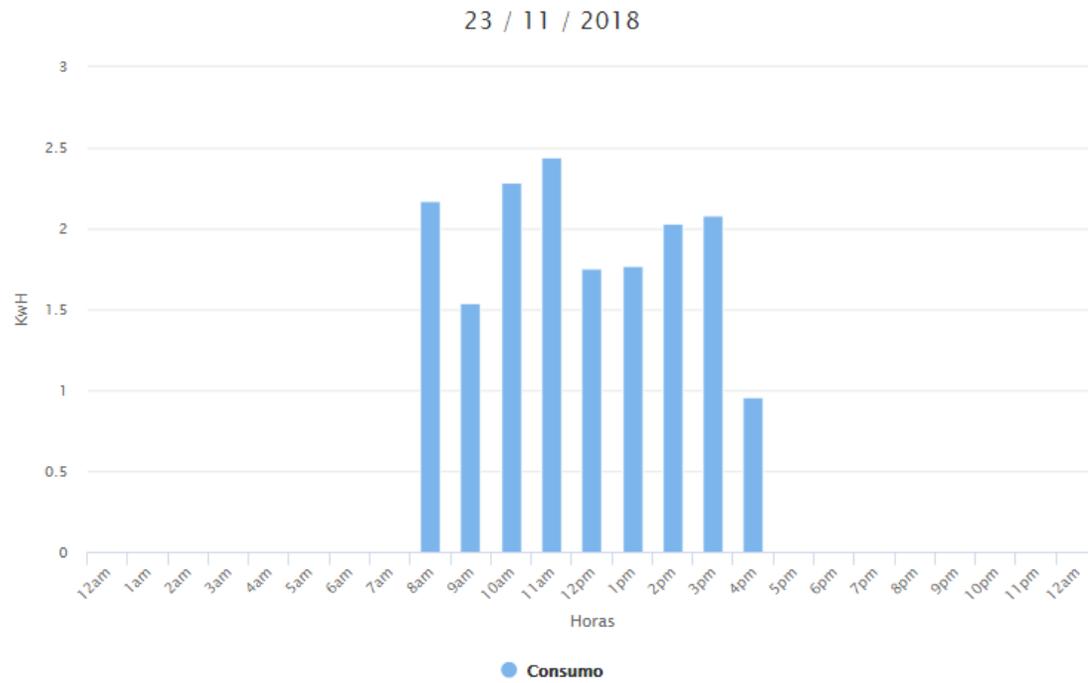
Anexo 13. Consumo energético del AA 20/11/2018, (Con controlador).



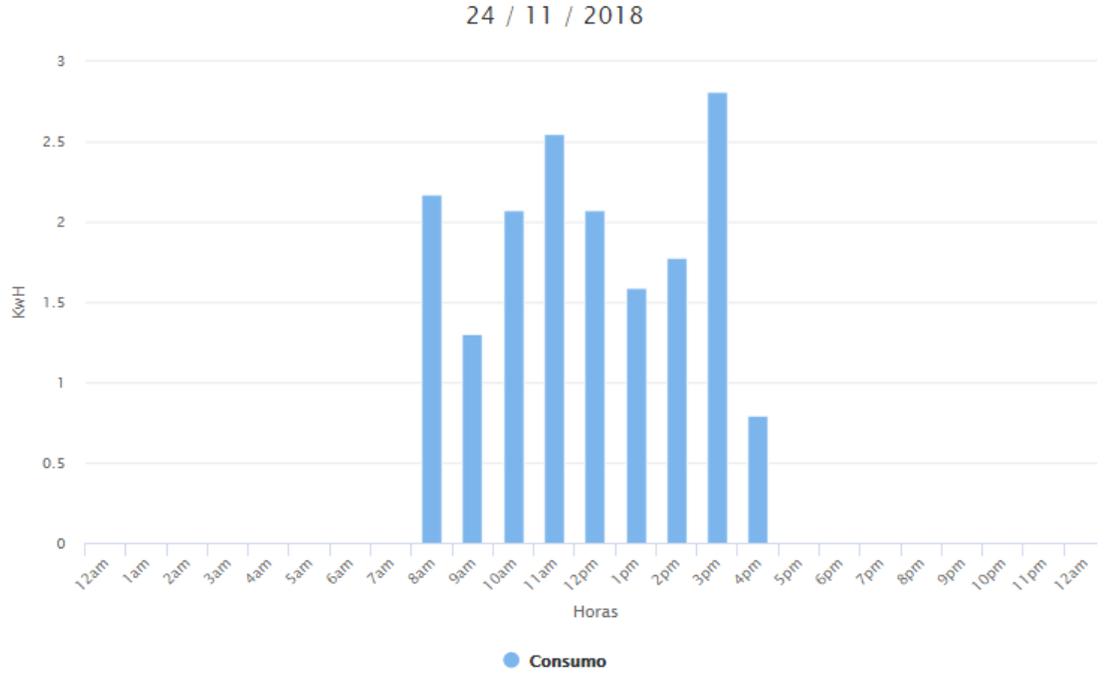
Anexo 14. Consumo energético del AA 21/11/2018, (Con controlador).



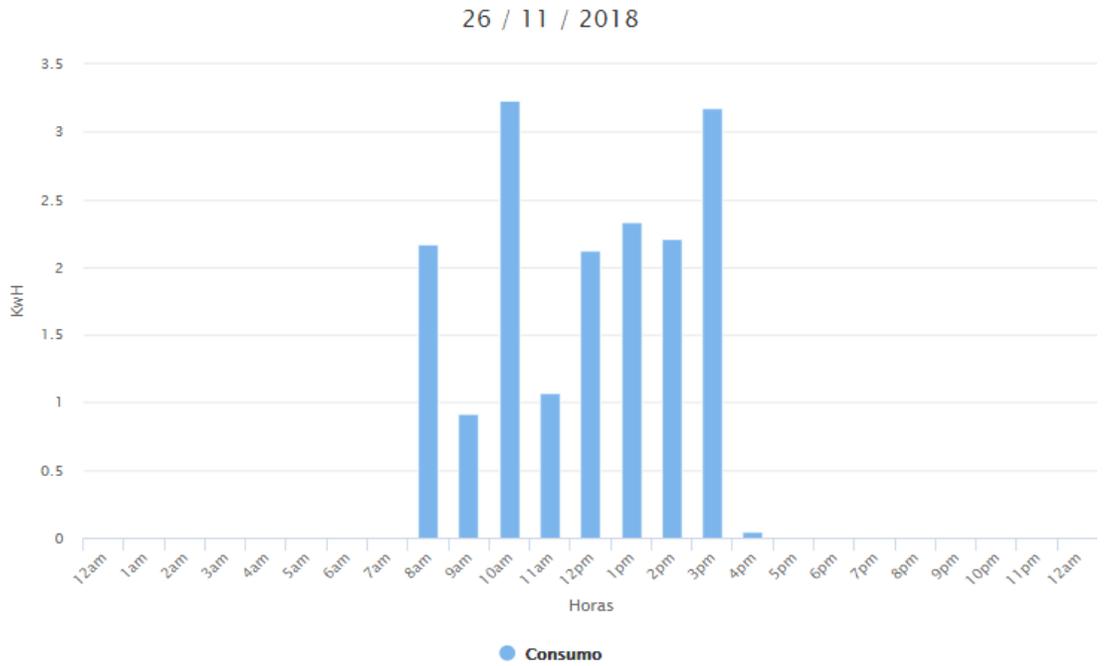
Anexo 15. Consumo energético del AA 22/11/2018, (Con controlador).



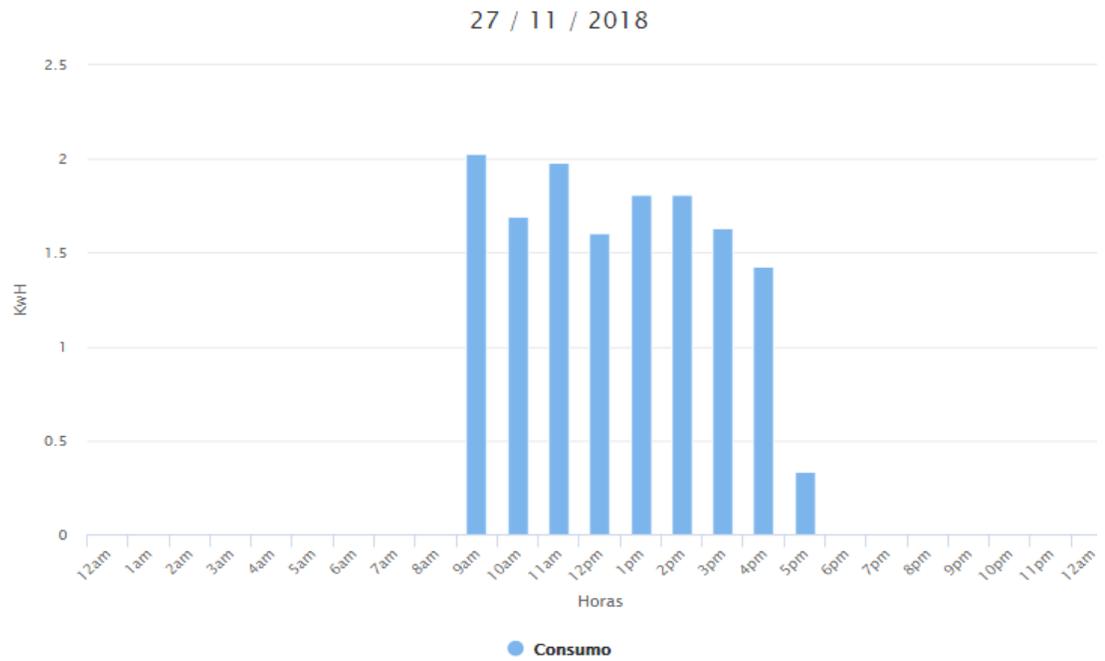
Anexo 16. Consumo energético del AA 23/11/2018, (sin controlador).



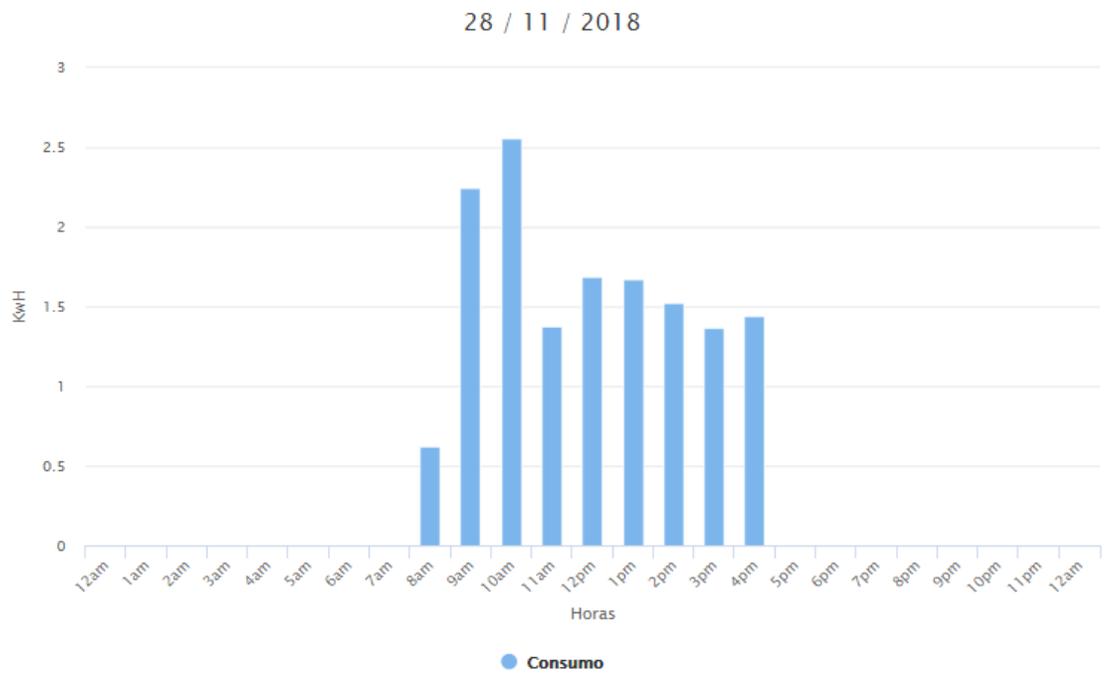
Anexo 17. Consumo energético del AA 24/11/2018, (sin controlador).



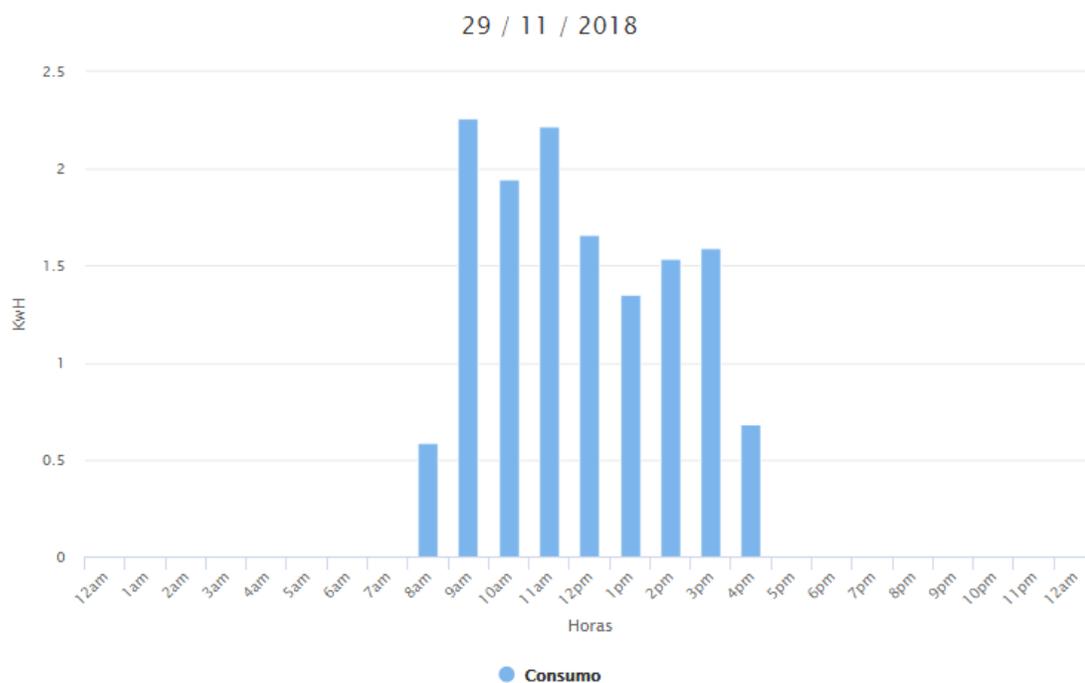
Anexo 18. Consumo energético del AA 26/11/2018, (sin controlador).



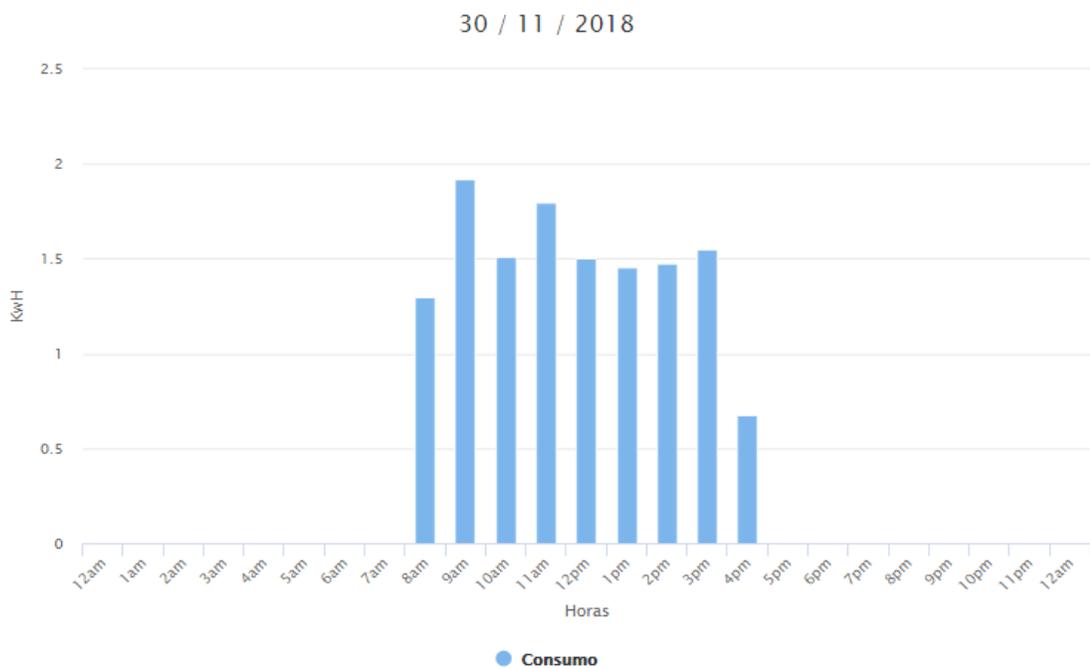
Anexo 19. Consumo energético del AA 27/11/2018, (Con controlador).



Anexo 20. Consumo energético del AA 28/11/2018, (Con controlador).



Anexo 21. Consumo energético del AA 29/11/2018, (Con controlador).



Anexo 22. Consumo energético del AA 30/11/2018, (Con controlador).

```

#define Infrarojo PIN_C4
#define LED_TEMPO PIN_B4
#define LED_RS232 PIN_C2
#define DHT11_1 PIN_A5

#include <18f14k50.h>
#fuses INTRC_IO,NOMCLR
#use delay(clock=4MHZ)
#use rs232(baud=9600, xmit=PIN_B7, rcv=PIN_B5)

#include <.\confortStar.c>
#include <.\DHT22.c>
#include <.\packageByte.c>
#include <.\myfuzzy.c>//controlador fuzzy
#include <.\temporizacion.c>//temporizaciones

//-----
float dhtdat[2];
int despack[2];
#int_rda
void serial_isr(){
    intby=getc();
    if(intby == updateTemp) {
        XleerDHT22(dhtdat);
        package(dhtdat[1],despack);
        putAppdat(0x91,despack[0],despack[1]);
    }
    else if(intby == updateHr) {
        XleerDHT22(dhtdat);
        package(dhtdat[0],despack);
        putAppdat(0x93,despack[0],despack[1]);
    }
    else if(intby == updaFuzz) putAppdat(0xA1,34,78);
    else if(intby == butonON) configurarAire(ON_AIRE,16, FAM_MED);
    else if(intby == butonOFF) configurarAire(OFF_AIRE,16, FAM_MED);
}
//-----

//-----
enum estados{
    Evaluando,
    RuningONN,
    RuningOFF,
    Stop
}estado;
//-----

void main(void){
    output_low(Infrarojo);
    output_low(LED_TEMPO);
    setup_timer_2(T2_DIV_BY_1,26,1);
    setup_ccp1(CCP_PWM);
    set_pwm1_duty(14);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);

    int1 presencia = 1;

    float Tpe = 0.00;//temperatura
    float Hre = 0.00;//humedad relativa

    float otrm[2]; // buffer de datos Hdt22
    float timeBuf = 0.00;//buffer de tiempo

    int16 TsTon = 0x00;//tiempo Ton en segundos
    int16 TsTFF = 0x00;//tiempo Toff en segundos
    int16 cSeg = 0x00;//contador de segundos

    estado = Evaluando;//estado inicial
#define Tmaxv 24.00
#define Tminv 23.50
    delay_ms(1000);

```

```

for(;;){
switch(estado){
case Evaluando:
cSeg = 0x00;//conteo en cero
delay_ms(2000);
XleerDHT22(otrm);
Hre = otrm[0];
Tpe = otrm[1];
if(!presencia) estado = Stop;//presencia cero, Stop
else if(Tpe > Tmaxv){
estado = RuningONN;
timeBuf = fuzzyTon(Tpe,Hre);
TsTon = minToSegConverter(timeBuf);
configurarAire(ON_AIRE,16, FAM_MED);
delay_ms(1500);
configurarAire(ON_AIRE,16, FAM_MED);
delay_ms(1500);
configurarAire(ON_AIRE,16, FAM_MED);
}
else if(Tpe < Tminv){
estado = RuningOFF;
timeBuf = fuzzyToff(Tpe,Hre);
TsTFF = minToSegConverter(timeBuf);
configurarAire(OFF_AIRE,16, FAM_MED);
delay_ms(1500);
configurarAire(OFF_AIRE,16, FAM_MED);
delay_ms(1500);
configurarAire(OFF_AIRE,16, FAM_MED);
}
else estado = Evaluando;
break;
case RuningONN:
delay_1segTON();
cSeg++;
XleerDHT22(otrm);
Hre = otrm[0];
Tpe = otrm[1];
if(!presencia) estado = Stop;//presencia cero, Stop
else if(Tpe < Tminv || cSeg > TsTon) estado = Evaluando;//
else estado = RuningONN;//
break;
case RuningOFF:
delay_1segTOFF();
cSeg++;
XleerDHT22(otrm);
Hre = otrm[0];
Tpe = otrm[1];
if(!presencia) estado = Stop;//presencia cero, Stop
else if(Tpe > Tmaxv || cSeg > TsTFF) estado = Evaluando;//
else estado = RuningOFF;//
break;
case Stop:
if(presencia) estado = Evaluando;//presencia, Evaluar
else estado = Stop;//no presencia, Stop
break;
}
}
}

```

```

#include <18f14k50.h>
#define adc=10
#define fuses INTRC_IO,NOMCLR
#define use delay(clock=4MHZ)
#define use rs232(baud=9600, xmit=PIN_B7, rcv=PIN_B5)
#include <.\ACpower.c> //lib medicion de powermean
#include <.\packageByte.c> // convert float num to two-byte
#include <.\esp8266.c> //driver de comunicacion con esp8266

//-----
#define int_rda
void serial_isr(){
    intby=getc();
    if(intby == byteInit){
        mibuff[0]=intby;
        index++;
    }
    else if(mibuff[0]==byteInit && index==1){
        miBuff[1]=intby;
        index++;
    }
    else if(miBuff[0]==byteInit && index==2 && intby==byteStop){
        respEsp=miBuff[1];
        index=0;
    }
}

//-----
void main(void){
    float potr =0.00,
        enegyTotal=0.00,
        energyMuestra=0.00;
    int g[2];
    int numMuestras=0x00;
    respEsp = 0x00;
    conf_power();
    output_low(led_wifi);
    output_low(led_load);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    conf_power();
    delay_ms(500);
    conect_wifi();
    delay_ms(1000);
    package(0.00,g);
    load_energy(g[0],g[1]);
    for(;;){
        if(numMuestras==100){
            enegyTotal = enegyTotal / 6.00;
            package(enegydTotal,g);
            load_energy(g[0],g[1]);
            energyMuestra = 0.00;
            enegyTotal = 0.00;
            numMuestras = 0x00;
        }
        else{
            delay_ms(3000);
            potr=measurement_power();//KW
            energyMuestra = 3*potr;//kJulios
            enegyTotal = enegyTotal + energyMuestra;//KJulios
            numMuestras++;
        }
    }
}
}

```

Anexo 25. Código fuente del microcontrolador para el medidor IoT, rutina MAIN parte B.

```

#include <ESP8266WiFi.h>
const char* ssid = "Josue2";
const char* password = "micontrasena";
const char* host = "192.168.0.5";
const int httpPort = 80;
//-----Archivos en el servidor-----
const char* url_crt = "http://192.168.0.5/LabWebSite/esp/put_server_currt.php";
const char* url_vol = "http://192.168.0.5/LabWebSite/esp/put_server_voltg.php";
const char* url_pow = "http://192.168.0.5/LabWebSite/esp/put_server_power.php";
const char* url_fpw = "http://192.168.0.5/LabWebSite/esp/put_server_factp.php";
const char* url_consu = "http://192.168.0.5/sitio/esp/put_consumo.php";
//-----keywords-----
#define byteInit 0xAF
#define byteStop 0xAA
#define conectWifi 0xA1
#define wifiOk 0xA3
#define wifiNok 0xA7
#define verWifi 0xA9
#define loadCurt 0xB2
#define loadvolt 0xB4
#define loadfact 0xB6
#define loadpowe 0xB8
#define loadconsumo 0xC2
#define loadACK 0xBA
#define loadNoack 0xBB

//-----
String formatting_POS(String datos,String url){
    String format_post = "POST " + url + " HTTP/1.0\r\n" +
        "Host: " + host + "\r\n" +
        "Accept: *" + "/" + "*\r\n" +
        "Content-Length: " + datos.length() + "\r\n" +
        "Content-Type: application/x-www-form-urlencoded\r\n" +
        "\r\n" + datos;
    return format_post;
}

//-----
String packing_floatnum_POS(int b_msb, int b_lsb,String url_p){
    String s_byte0 = String(b_msb),
        s_byte1 = String(b_lsb);
    String e_byte0 = "t_interpart=",
        e_byte1 = "t_decimpart=";
    String data= e_byte0 + s_byte0 + "&" +
        e_byte1 + s_byte1;
    String pack_data = formatting_POS(data,url_p);
    return pack_data;
}

//-----
void load_server(int byte0, int byte1, String url_1){
    WiFiClient client;
    if (client.connect(host, httpPort)) {
        client.print(packing_floatnum_POS(byte0,byte1,url_1));
        client.stop();
        return;
    }
}

//-----
//-----funciones de comunicacion esp8266 to pic-----
void clear_buff(void); //limpiar buffer de recepcion
void write_reponse(void); //da una respuesta al PIC sobre la instruccion solicitada
void conec_wifi(void); //conectar a la red WIFI
void load_volt(int pint,int pdec); //subir valor de voltaje
void load_curt(int pint,int pdec); //subir valor de corriente
void load_fctp(int pint,int pdec); //subir valor factpower
void load_powe(int pint,int pdec); //subir valor power
//-----variables comunicacion esp8266 to PIC-----
int inxdex=0; //contador de desplazamiento,buffer de recepcion
int byteIntro; //ultimo byte recibido
int byte0, //byte de sincronizacion
byte1, //byte de instruccion
byte2, //data0
byte3, //data1
byte4, //data2
byte5, //data3
byte6; //byte stop

```

```

void clear_buff(){
  byte0    = 0x00;
  byte1    = 0x00;
  byte2    = 0x00;
  byte3    = 0x00;
  byte4    = 0x00;
  byte5    = 0x00;
  byte6    = 0x00;
  byteIntro = 0x00;
  inxdex   = 0x00;
}
//-----
void write_reponse(int r){
  Serial.write(byteInit);
  Serial.write(r);
  Serial.write(byteStop);
}
//-----
void conec_wifi(){
  //wifiOk
  //wifiNok
  int timeOutWifi=0;
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED && timeOutWifi <= 30) {
    delay(1000);
    timeOutWifi++;
  }
  if(WiFi.status() == WL_CONNECTED) {
    write_reponse(wifiOk);
  }
  else {
    write_reponse(wifiNok);
  }
  clear_buff();
}
//-----
void sta_wifi(){
  delay(3000);
  write_reponse(wifiOk);
  clear_buff();
}
void load_volt(int pint,int pdec){
  delay(500);
  load_server(pint,pdec, url_vol);
  write_reponse(loadACK);
  clear_buff();
}
void load_curt(int pint,int pdec){
  delay(500);
  load_server(pint,pdec, url_crt);
  write_reponse(loadACK);
  clear_buff();
}
void load_fctp(int pint,int pdec){
  delay(500);
  load_server(pint,pdec, url_fpw);
  write_reponse(loadACK);
  clear_buff();
}
void load_powe(int pint,int pdec){
  delay(500);
  load_server(pint,pdec, url_pow);
  write_reponse(loadACK);
  clear_buff();
}
void load_consux(int pint,int pdec){
  delay(500);
  load_server(pint,pdec, url_consux);
  write_reponse(loadACK);
  clear_buff();
}
}

```

Anexo 27. Código fuente del ESP01, parte B.

```

//-----
void setup() {
  Serial.begin(9600);
}
//-----

void loop() {
while (Serial.available()) {
  byteIntro = (int)Serial.read();
  if(byteIntro == byteInit){
    byte0 = byteIntro;
    index++;
  }
  else if(byte0 == byteInit && index == 1){
    byte1 = byteIntro;
    index++;
  }
  else if(byte0 == byteInit && index == 2){
    byte2 = byteIntro;
    index++;
  }
  else if(byte0 == byteInit && index == 3){
    byte3 = byteIntro;
    index++;
  }
  else if(byte0 == byteInit && index == 4){
    byte4 = byteIntro;
    index++;
  }
  else if(byte0 == byteInit && index == 5){
    byte5 = byteIntro;
    index++;
  }
  else if(byte0 == byteInit && index == 6 && byteIntro == byteStop){
    index = 0x00;
    if(byte1 == conecWifi) conec_wifi();
    else if(byte1 == verWifi) sta_wifi();
    else if(byte1 == loadCurt) load_curt(byte2-0x01,byte3-0x01);//integer, decimal
    else if(byte1 == loadvolt) load_volt(byte2-0x01,byte3-0x01);//integer, decimal
    else if(byte1 == loadfact) load_fctp(byte2-0x01,byte3-0x01);//integer, decimal
    else if(byte1 == loadpove) load_powe(byte2-0x01,byte3-0x01);//integer, decimal
    else if(byte1 == loadconsumo) load_consux(byte2-0x01,byte3-0x01);//integer, de
    break;
  }
}
}
//nada
}

```

Anexo 28. Código fuente del ESP01, parte C.