INSTITUTO TECNOLÓGICO DE CHIHUAHUA

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

SISTEMA NEURO-DIFUSO DE CONTROL INTELIGENTE DE TEMPERATURA

TESIS

PRESENTA:

Ing. Jesús Omar Ponce Chaparro

DIRECTOR DE LA TESIS:

Dr. Pedro Rafael Márquez Gutiérrez

TECNOLÓGICO NACIONAL DE MÉXICO

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

CHIHUAHUA, CHIH. ENERO DEL 2020

“No creo que haya una emoción más intensa para un inventor que ver alguna de sus creaciones funcionando. Esa emoción hace que uno se olvide de comer, de dormir, de todo”.

- Nikola Tesla

A mi esposa…

Por tu apoyo, fortaleza y comprensión, pero sobre todo, por cuidar de nuestra hogar tu sola en mis horas de ausencia.

A mis pequeños hijos…

Perdón por el tiempo que no les dedique por buscar un sueño personal.

A mis padres…

Por su consejo y apoyo incondicional.

A mis maestros y mentores...

Por compartir su conocimiento y brindar su amistad.

SISTEMA NEURO-DIFUSO DE CONTROL INTELIGENTE
DE TEMPERATURA

Ing. Jesús Omar Ponce Chaparro
Maestría en Ingeniería Mecatrónica
División de Estudios de Posgrado e Investigación
Instituto Tecnológico de Chihuahua
Chihuahua, Chih., 2019
Director de tesis: Dr. Pedro Rafael Márquez Gutiérrez

RESUMEN

El control de temperatura en edificaciones es un tema que paulatinamente va tomando fuerza en la actualidad, por lo que cada vez es más común observar determinados aparatos y dispositivos que ofrecen este tipo de funciones. Sin embargo, la eficiencia y usabilidad de estos aparatos no siempre es una experiencia grata.

Este documento presenta el uso de técnicas de inteligencia artificial aplicadas en dispositivos de control de temperatura residencial. Se implementó un sistema de control neuro-difuso en un controlador de temperatura para mejorar su desempeño en cuestión de confort y uso de recursos energéticos. Esto es posible debido a que el sistema cuenta con habilidades de aprendizaje y toma de decisiones, las cuales le permiten gestionar su funcionamiento de acuerdo a sus características más utilizadas.

ABSTRACT

Temperature control in buildings is a subject that is taking matter nowadays, such that is very common to observe devices that offers this kind of features more frequently. Anyway, the efficiency and usability of these devices is not always a pleasure.

This document presents the usage of artificial intelligence techniques applied to residential temperature control devices. The intention was to implement a neural-fuzzy control system in a temperature control device to improve its performance related to comfort and energy savings. This is possible now because the system is able to learn and take its own decisions, which allows to manage its functionality according its most used features.

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABLAS

I. CAPÍTULO I. CONCEPTOS GENERALES

1.1 Introducción

En el afán de mejorar nuestra calidad de vida nos vemos en la necesidad de buscar maneras de controlar, en medida de lo posible, el ambiente que nos rodea.

Para poder lograr esta actividad, es necesario realizar las siguientes funciones:

1- Medir el estado actual.
2- Comparar el estado actual con el estado deseado.
3- Ejecutar una acción para llevar el estado actual al estado deseado.

La manera en la que comúnmente se conoce el estado de un sistema es a través de sensores y/o transductores.
La manera en que se compara la diferencia entre estados y se decide la acción a tomar es a través de un controlador.
La manera en que se ejecuta una acción para cambiar el estado es a través de una planta.

La teoría de control clásica se basa en modelos matemáticos que describen el comportamiento de alguna planta la cual se pretende controlar. Aunque durante los últimos años esta teoría es ampliamente utilizada, el principal problema de este tipo de control reside en que, si se cambia la planta, se deberá modificar también el controlador.

Debido a lo anterior es que se han investigado (y se siguen experimentando) distintas alternativas de control [1], y he aquí donde toman relevancia otros métodos como aquellos basados en lógica difusa. La idea principal del control difuso [2] es construir un modelo basado en un “experto en control” humano, que sea capaz de controlar la planta sin necesidad de un modelo matemático. El “experto en control”

especifica sus acciones de control en forma de reglas lingüísticas. Estas reglas de control son traducidas entonces a la teoría de conjuntos difusos obteniendo un cálculo que pueda simular el comportamiento del “experto en control”.

El definir reglas lingüísticas correctas depende del conocimiento del “experto en control”, pero la traducción de estas reglas a la teoría de conjuntos difusos es un área que no está formalizada, por lo que algunas decisiones se deben tomar de manera arbitraria.

Las redes neuronales ofrecen la posibilidad de corregir esta situación. Una red neuronal artificial puede ser consideradas como un modelo matemático simplificado del cerebro, la cual está diseñada para aprender de la información manejada, pero necesita ser entrenada primero. Quizás su mayor ventaja es su adaptabilidad, ya que pueden automáticamente optimizar su comportamiento para reconocer patrones, tomar decisiones, controlar sistemas, entre otras. La adaptabilidad les permite a las redes neuronales desempeñarse de una manera adecuada incluso cuando el ambiente o sistema a controlar varia en el tiempo.

La combinación de redes neuronales y control difuso concentra las ventajas de ambas aproximaciones en una misma solución. Una red de control neuro-difuso es entonces aquella red que integra los elementos y funciones básicas de control tradicional en funciones de lógica difusa dentro de una estructura que contiene habilidades de aprendizaje.

Aunque actualmente la variedad de sistemas de control y de leyes de control es bastante amplia, este estudio se enfoca en la utilización de métodos de control neuro-difusos que se encuentren orientados a la utilización en sistemas de climatización residencial.

1.2 Propósitos y expectativas

1.2.1 Objetivos

Objetivo general: Construir un sistema de control inteligente basado en técnicas de Inteligencia Artificial para el acondicionamiento de ambientes que tome en consideración variables del entorno como temperatura, humedad, punto de rocío y CO_2, tal que se pueda adaptar a diferentes entornos y usuarios automáticamente.

Objetivos específicos:

1. Definir criterios cualitativos de nivel de confort con base en estándares.
2. Desarrollar un subsistema de aprendizaje basado en técnicas *Deep Learning* de acuerdo a los criterios de confort definidos.
3. Implementar el sistema inteligente de control neuro-difuso basado en los algoritmos desarrollados, tal que:
    a) Aprenda las configuraciones preferidas del usuario y el entorno.
    b) Defina el criterio óptimo de operación del sistema basado y lo ajuste dinámicamente.
    c) Mantenga el ambiente en un nivel confortable de operación y con consumo económico de energía.

1.2.2 Meta

Implementación física de un sistema de control ambiental para el hogar, que sea lo suficientemente inteligente para tomar decisiones acerca del modo óptimo de operación considerando variables del entorno (temperatura, humedad, nivel concentración de gases y partículas) además de preferencias de usuario (aprendizaje de las configuraciones por día y hora previamente utilizadas), lo que conlleva a la maximización del desempeño (confort) reduciendo el consumo de recursos energéticos (gas y electricidad).

1.3 Hipótesis

Es posible incrementar la eficiencia energética y mejorar la experiencia de usuario de un sistema de control de climatización residencial por medio de habilitar al controlador con un algoritmo inteligente que sea capaz de tomar decisiones acerca del modo óptimo de operación del sistema basado en los historiales de configuraciones del usuario y estados ambientales.

1.4 Justificación

La “experiencia de usuario” de las personas al utilizar distintos y nuevos aparatos eléctricos y electrónicos no siempre es satisfactoria, ya que en muchas ocasiones hacer uso óptimo de sus características se requiere de algún tipo de familiarización con la tecnología, lo que puede complicar un tanto las cosas para el usuario y llevarlo a utilizarlos de una manera inadecuada, que ocasiona el consumo ineficiente de los recursos energéticos (como gas y electricidad) o puestas a punto equivocados.

Debido a lo anterior, la frase “productos inteligentes” resuena cada vez más hoy en día. Existen en el mercado diversos tipos de productos que ofrecen algún grado de control sobre las variables previamente mencionadas que, sin embargo, requieren de la intervención del usuario en gran medida para poder operar satisfactoriamente y al gusto del usuario. Si en algún momento el usuario olvida u omite cambiar el estado del sistema (o ignora que una opción se encuentra disponible), este pudiera quedarse funcionando con tales configuraciones indefinidamente llegando a realizar dispendio de recursos, físicos, monetarios y de confort.

He aquí donde toma relevancia este proyecto, ya que se pretende proporcionar al sistema de control con cierto nivel de inteligencia para que pueda aprender las preferencias del usuario dependiendo de los cambios del entorno. Esto permitirá al

sistema auto-ajustarse para consumir de manera óptima los recursos sin comprometer el confort o salubridad del ambiente.

Cabe mencionar que el interés de adentrarse en esta área viene apoyado por el hecho de que actualmente existen empresas donde se diseñan dispositivos de índole similar, y existe un interés en experimentar con técnicas de Inteligencia Artificial. Sin embargo, los productos actuales no cuentan con la funcionalidad necesaria para cubrir ciertas necesidades entrelazadas del cliente como facilidad de uso, ahorro de energía y puesta a punto.

1.5 Metodología utilizada

1.5.1 Tipo de investigación

El presente análisis tiene como objetivo el presentar un proyecto cuya investigación es del tipo experimental, ya que el tema de estudio está orientado a buscar y proporcionar una solución de control de climatización residencial más eficiente, novedosa e inteligente que las que actualmente están disponibles.

Así pues, se pretende realizar una determinación evaluativa acerca de las prestaciones proporcionadas por el objeto de estudio.

1.5.2 Diseño de la investigación

Antes de proseguir con el desarrollo de este apartado, es de importancia el comprender que no existe un método universal para modelar y definir un plan de acción, sino que este se define con base en características y parámetros de interés para acotarse adecuadamente a las necesidades del objeto de estudio.

El proceso de implementación es del tipo recursivo, es decir que algunas consideraciones que fueron definidas inicialmente en el momento de planeación

pueden cambiar, ya sea por factibilidad o complejidad entre otros casos. Lo importante es tener en cuenta y no perder de vista la meta principal, con el fin de dirigir adecuadamente cualquier proceso de retroalimentación, rediseño y re-implementación hacia ella.

Las fases a seguir durante el desarrollo de este proyecto son las siguientes:

1- Definición del alcance.
2- Investigación de soluciones relacionadas previamente implementadas.
3- Elaboración conceptual del modelo operativo.
4- Implementación física de la solución propuesta obtenida.
5- Evaluación y valoración de los resultados.

1.6 Infraestructura

Para poder desarrollar y llevar a cabo este proyecto, algunos dispositivos y equipo serán necesarios, así como documentación, textos especializados y asesoría profesional. El Instituto Tecnológico de Chihuahua cuenta con biblioteca, tanto de libros físicos como virtuales, con gran facilidad para acceder a distintas fuentes de información, así como laboratorios para poder realizar experimentación física para validación de hardware y software.

Además de lo anterior, otros requerimientos para el desarrollo de este proyecto se encuentran listados a continuación:

- Computadora para investigación y acceso de recursos digitales, así como manejo de paquetes computacionales.
- Tarjetas de desarrollo para pruebas, validación e implementación.
- Equipo de medición y pruebas.

El capítulo II presenta la descripción del tema de investigación, así como información sobre el estado del arte realizado previamente.

II. CAPÍTULO II. DESCRIPCION DEL TRABAJO

2.1 Definición del problema de tesis

De acuerdo al tipo de investigación, el punto central de esta investigación es la de diseñar un algoritmo de control con autoaprendizaje y capacidad de decisión para el control de temperatura residencial, así como su implementación física utilizando *hardware* ya existente.

El diseño del algoritmo "inteligente" presenta una serie de retos, entre los cuales destacan:

1- Identificar las necesidades y características de mayor interés por parte de los usuarios de este tipo de sistemas.
2- Analizar diversas técnicas de control neuro-difuso para conocer ventajas y desventajas.
3- Definir cuantitativamente la técnica de control óptima para el objeto de estudio
4- Plantear y desarrollar una posible solución.
5- Implementación física y adquisición de resultados.
6- Evaluar factibilidad de la solución propuesta.

2.2 Antecedentes

2.2.1 Aplicaciones de sistemas difusos

La primera aplicación de lógica difusa se debe a Mamdani [3] de la Universidad de Londres, Reino Unido, quien en 1974 diseño un control difuso experimental para una máquina de vapor. En 1980, una compañía danesa (F.L Smidth & Co. A/S) utilizó teoría difusa en el control de un horno de cemento. Tres años después, Fuji Electric

Co, Ltd., en Japón, implementó control difuso de inyección de químicos en plantas purificadoras de agua.

El primer controlador difuso fue exhibido en 1987 en el segundo congreso de la asociación internacional de sistemas difusos (IFSA, por sus siglas en ingles). Este controlador fue creado por Omron Corp, una compañía japonesa que comenzó su investigación en lógica difusa en 1984, pero no fue sino hasta 1990 cuando Japón comenzó a incluir teoría difusa en productos comerciales. Un ejemplo de ello es el sistema de control crucero automático en los automóviles, ocasionando que controles complejos fueran más eficientes y fáciles de utilizar.

Hoy en día existen bastantes productos en el mercado basados en lógica difusa (como muestra la ***tabla 2.1***.). El dominio más extendido se encuentra en control difuso de diversas características físicas y químicas como temperatura, corriente eléctrica, flujo de líquidos y gases, movimiento de máquinas, etc. También se pueden obtener sistemas difusos aplicando los principios de conjuntos difusos y lógica difusa a otras áreas, por ejemplo, en sistemas basados en conocimiento, bases de datos que guardan y recuperan información, control de patrones los cuales interactúan con señales visuales o de audio difusas, aplicaciones en medicina, economía o problemas de administración que involucren procesamiento de información difusa.

Tabla 2.1 Aplicaciones industriales de controladores basados en lógica difusa [4].

Producto	Compañía
Lavadora	AEG, Sharp, Goldstar
Olla arrocera	Goldstar
Freidora	Tefal
Horno de microondas	Sharp
Rasuradora eléctrica	Sharp
Refrigerador	Whirlpool
Cargador de batería	Bosch
Limpiadora de vacío	Philips, Siemens
Cámara de video	Canon, Sanyo, JVC
Control de transmisión automotriz	GM, Honda, Mazda
Control de clima	Ford
Control de temperatura	NASA
Tarjeta de crédito	GE

Cuando los sistemas difusos se aplican a los problemas apropiados, sus características típicas son respuestas más rápidas y suavizadas que con sistemas convencionales. Lo anterior se traduce en operaciones más eficientes y confortables para tareas como control de temperatura, velocidad crucero, etc. Además, esto conlleva un ahorro de energía, reduce los costos de mantenimiento y prolonga la vida de los equipos.

En sistemas difusos, el describir las reglas de control es normalmente más simple y rápido que en sistemas convencionales, por lo que normalmente se ejecutan de manera más rápida, son más robustos y tienen un costo menor en conjunto, contribuyendo a un mejor desempeño.

En resumen, los métodos convencionales son adecuados para problemas simples, mientras que los sistemas difusos se acomodan mejor a sistemas o aplicaciones complejas que involucran pensamiento humano descriptivo o intuitivo.

Aún y con las ventajas que los sistemas difusos representan, existen algunas limitantes y problemas que deben ser considerados, entre los que se incluyen los siguientes [5]:

- *Estabilidad:* un detalle mayor para el control difuso.
  No hay garantía teórica que un sistema difuso general no se comportará de manera caótica y permanecerá estable, aunque esa posibilidad parece remota desde el punto de vista de la experiencia.

- *Capacidad de aprendizaje:* Los sistemas difusos no cuentan con capacidad de aprendizaje y no tienen memoria.

- *Determinar y ajustar las funciones de pertenencia y reglas difusas adecuadas no siempre es fácil.* Incluso después de pruebas extensivas, es difícil definir la cantidad de funciones de pertenencia que realmente son necesarias.

Debido a las condiciones anteriores, primero se debe realizar un análisis acerca de si un sistema difuso es una opción correcta para un determinado problema. Si el conocimiento que se tiene acerca del comportamiento del sistema puede describirse de forma aproximada o por medio de reglas heurísticas, entonces la utilización de un sistema difuso es factible.

2.2.2 Aplicaciones de redes neuronales artificiales

El estudio de computación cerebral se remonta a trabajos de McCulloch y Pitts en 1943 [6], quienes crearon un modelo informático para redes neuronales basados en las matemáticas y algoritmos denominados lógica de umbral. Años después, Minsky y Papert [7] continuaron con los estudios, sin embargo, la investigación se estancó en 1969, después de descubrir que los ordenadores no tenían suficiente poder de procesamiento para manejar eficazmente el tiempo de ejecución largo requerido por las grandes redes neuronales.

Existen varias clases de problemas que pueden solucionarse más fácilmente a través de redes neuronales que con otro tipo de técnicas. Estas tareas presentan generalmente algún tipo de ambigüedad, como aquellas inherentes al área de reconocimiento de caracteres escritos a mano. Problemas de este tipo son difíciles de solucionar con métodos convencionales, en parte porque los métodos utilizados por el cerebro para comparar patrones no son similares a aquellos seleccionados por un ingeniero diseñando un sistema de reconocimiento.

Los diseñadores de sistemas difusos y sistemas expertos a menudo encuentran difícil la tarea de encontrar descripciones aceptables para las relaciones complejas que gobiernan las clases de inclusión. En sistemas basados en redes neuronales que pueden ser entrenados, estas relaciones se obtienen abstractamente de los datos de entrenamiento. Debido a que las redes neuronales pueden construirse para utilizar de entradas y salidas, pueden ser utilizadas para resolver problemas que requieran

cantidades de variables de entrada considerablemente mayores que las que pudieran ser utilizadas factiblemente por la mayoría de las aproximaciones.

De igual manera, se debe considerar que las redes neuronales no trabajarán adecuadamente para resolver problemas en los cuales no sea posible obtener conjuntos de datos de entrenamiento suficientes [8]. La ***tabla 2.2*** lista algunas de las aplicaciones de redes neuronales [9].

Tabla 2.2 Aplicaciones industriales y comerciales de redes neuronales.

Área	Uso
Industria de telecomunicaciones	Administración de equipo telefónico
Control de sonido y vibración	Control adaptivo para generar ruido y vibración igual y opuesto
Aceleradores de partículas	Cancelar perturbaciones que disminuyan la exactitud de posicionamiento de haces de partículas en colisión
Reconocimiento de caracteres	Equipo comercial para el reconocimiento de patrones y caracteres impresos
Automotriz	Control de frenado y suspensión
Biomedicina	Investigación de secuencias de RNA y DNA, así como clasificación de formas de onda de ECG y EEG
Reconocimiento de voz	Sistemas independientes de reconocimiento de habla y lenguaje

Su característica de auto-optimización les permite a las redes neuronales “diseñarse” a sí mismas, es decir, tienen la capacidad de adaptarse a la aplicación. Debido a ésta característica, con pequeñas modificaciones en los parámetros de operación es posible utilizar la misma red neuronal en una amplia variedad de aplicaciones.

2.2.3 Importancia de métodos neuro-difusos en control de sistemas

Cada técnica inteligente presenta propiedades computacionales particulares (habilidad para aprender, toma de decisiones) que las vuelven adecuadas para ciertos problemas, pero no para otros. Por ejemplo, mientras las redes neuronales son adecuadas para el reconocimiento de patrones, no lo son así al explicar cómo llegaron a determinada decisión. La lógica difusa es adecuada para explicar sus decisiones, pero no para adquirir automáticamente las reglas que utiliza para tomar decisiones. En la ***tabla 2.3*** se presenta una breve comparación entre sistemas difusos y redes neuronales [10].

Tabla 2.3 Propiedades de sistemas difusos y redes neuronales.

Habilidad		Sistema difuso	Red neuronal
Adquisición de conocimiento	*Herramientas*	*Interacción*	*Algoritmos*
	Entrada	*Experto humano*	*Conjuntos de pruebas*
Incertidumbre	*Información*	*Cuantitativo y cualitativo*	*Cuantitativo*
	Cognición	*Toma de decisión*	*Percepción*
Razonamiento	*Mecanismo*	*Búsqueda heurística*	*Computación paralela*
	Velocidad	*Baja*	*Alta*
Adaptación	*Tolerancia a fallas*	*Baja*	*Muy alta*
	Aprendizaje	*Inducción*	*Ajuste de pesos*
Lenguaje natural	*Implementación*	*Explicita*	*Implícita*
	Flexibilidad	*Alta*	*Baja*

En teoría, redes neuronales y sistemas difusos son equivalentes debido a que son convertibles, pero en la práctica cada uno tiene sus ventajas y desventajas.

Mientras que la lógica difusa desempeña un mecanismo de inferencia bajo incertidumbre cognitiva, las redes neuronales ofrecen ventajas como aprendizaje, adaptación, tolerancia a fallas, paralelismo y generalización.

Lo mencionado anteriormente han sido las limitaciones centrales detrás de la creación de sistemas inteligentes híbridos donde dos o más técnicas son combinadas de manera que se sobrepongan a las limitaciones de las técnicas individuales.

Para habilitar a un sistema para que sea capaz de tratar con incertezas de una manera “parecida a los humanos”, se puede incorporar el concepto de lógica difusa dentro de redes neuronales. El sistema hibrido resultante es llamado red neuro-difusa.

Las redes neuronales son utilizadas como sistemas de toma de decisiones en el control de equipos. Aunque la lógica difusa puede codificar el conocimiento experto directamente utilizando reglas con etiquetas lingüísticas, usualmente toma bastante tiempo el diseñar y ajustar las funciones de pertenencia que definen estas etiquetas lingüísticas cuantitativamente. Las técnicas de aprendizaje de las redes neuronales pueden automatizar este proceso y reducir sustancialmente el tiempo y costo de desarrollo, mientras mejoran su desempeño [8].

El uso de sistemas híbridos inteligentes está creciendo rápidamente con aplicaciones exitosas en distintas áreas, incluyendo control de procesos, ingeniería de diseño, comercio financiero, evaluación de crédito, diagnósticos médicos y simulaciones cognitivas [11].

La habilitación de un sistema para que sea capaz de tratar con incertidumbres cognitivas de una manera similar a los seres humanos, consiste incorporar el concepto de lógica difusa en la red neuronal.

En las ***figuras 2.1*** y ***2.2*** se muestran dos posibles modelos de sistemas neuro-difusos [4].

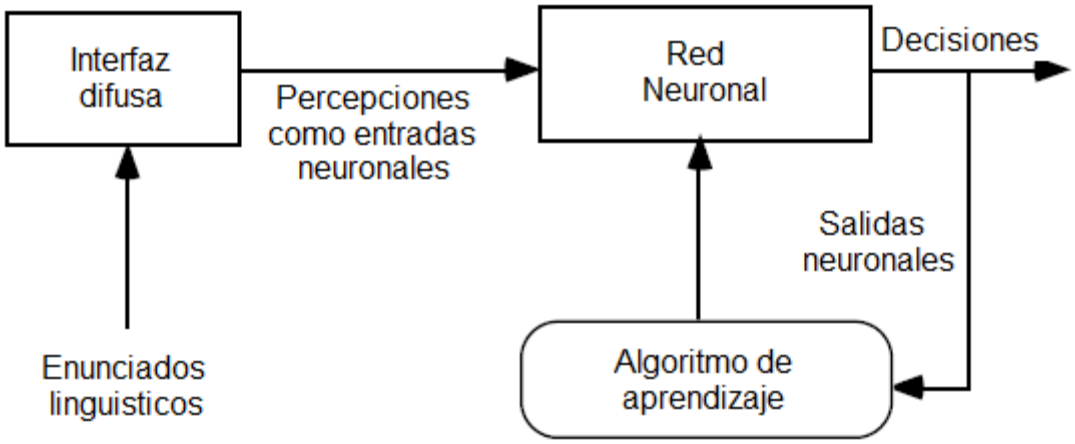

Figura 2.1 La red neuronal puede ser entrenada para obtener las decisiones deseadas.

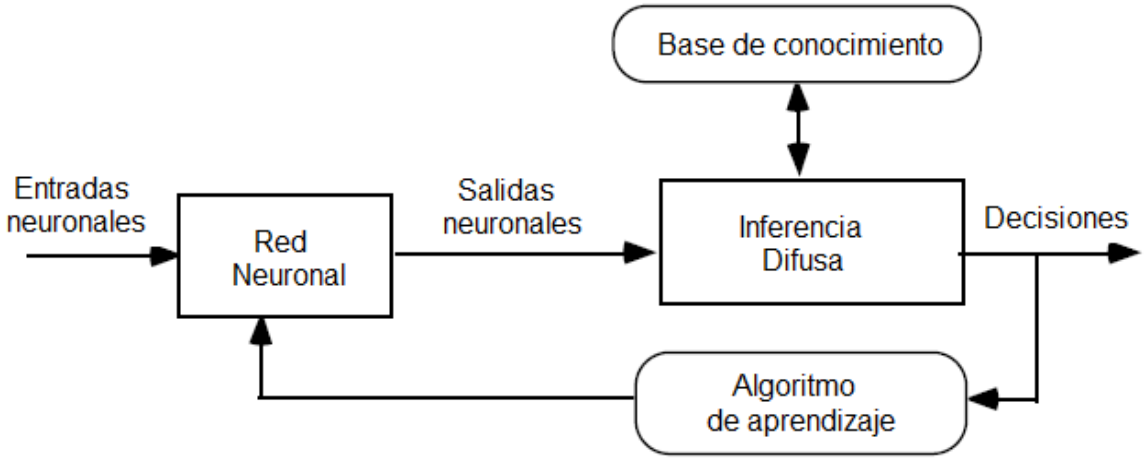

Figura 2.2 Red neuronal multicapa gestiona el mecanismo de inferencia difusa.

Basados en el proceso computacional involucrado en un sistema neuro-difuso, se puede clasificar el sistema como estático o dinámico.

Un sistema neuro-difuso típico es la arquitectura ARIC (por sus siglas en inglés *Approximate Reasoning Based Intelligent Control*), de Hamid R. Berenji [12]. Es un modelo de red neuronal de un controlador difuso, el cual aprende actualizando sus

predicciones del comportamiento físico del sistema y ajusta una base de conocimiento control predefinida.

Debido a las ventajas y prestaciones descritas anteriormente, es que se ha decidido utilizar un sistema de control neuro-difuso para la realización de este proyecto, ya que encaja en mayor medida con las características que se buscan gestionar y controlar.

2.2.4 Métodos de aprendizaje reforzado en el control de sistemas

La idea de aprendizaje reforzado se originó del término “control óptimo” al formularse un problema de diseño de un controlador para minimizar una medida del comportamiento del sistema a través del tiempo. Fue entonces que surgió el concepto de proceso de decisión de Markov, una teoría de aprendizaje reforzado para formular problemas de control óptimo.

El agente de aprendizaje reforzado aprende a actuar para maximizar la recompensa. No requiere de un “maestro” que le enseñe como tomar una acción, sino que toma decisiones a través del método de prueba y error, reconociendo una recompensa del ambiente con el cual interactúa. Aprendizaje reforzado no se trata de un método supervisado ni tampoco de un no supervisado, sino que se trata de una tercera categoría de aprendizaje de máquinas, ya que obtiene una señal de recompensa por una acción la cual desconoce si es correcta o incorrecta. En el contexto de inteligencia artificial, el aprendizaje reforzado le permite al agente detectar comportamientos automáticamente, lo cual no se puede lograrse con los métodos de aprendizaje supervisado o no supervisado.

La ***figura 2.3*** representa un proceso de toma decisiones secuencial. El estado S_t se refiere a una condición especifica en pasos discretos de tiempo $t=0,1,2\ldots$ Respondiendo al ambiente, el agente selecciona una acción determinística o

estocástica A_t, la cual intenta maximizar los retornos futuros y recibe una recompensa R_{t+1} mientras el agente se transfiere al estado S_{t+1}.

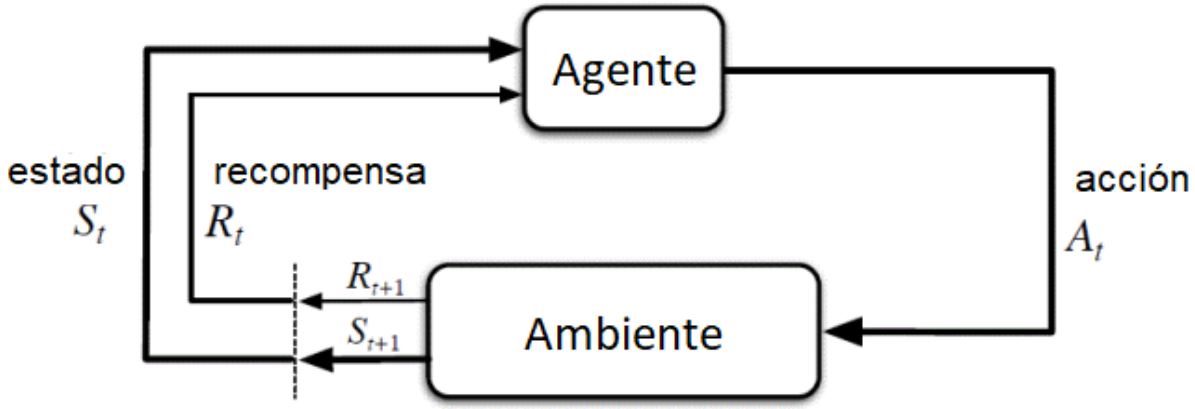

Figura 2.3 Interacción entre un agente y su ambiente en aprendizaje reforzado.

2.3 Estado del arte

2.3.1 Sistemas neuro-difusos en el control de temperatura

Hoy en día es común encontrar sistemas de aire acondicionado en los edificios. Están diseñados para crear un ambiente adecuado para vivir o generar ambientes especiales para lugares industriales. Los principales parámetros de interés en un sistema de aire acondicionado incluyen temperatura, humedad y calidad del aire en algún lugar.

El grado de requerimientos para los sistemas de aire acondicionados van en aumento, y junto con ellos el consumo de energía que se requiere para mantenerlos operando. Es por este motivo que se busca constantemente el mejorar la eficiencia de este tipo de dispositivos.

Recientemente, para el control moderno, se han desarrollado algunos controles adaptativos y de auto-sintonización. Los controles auto-sintonizables han reemplazado los viejos controladores PID de topología una entrada / una salida en muchos lugares.

El desarrollo de este tipo de controles ha sido posible gracias a los avances en los campos de lógica difusa y redes neuronales artificiales.

En los sistemas de lógica difusa, se generan resultados difusos a partir de entradas imprecisas. Pueden manejar de la incerteza de un modelo de pensamiento humano con funciones de pertenencia, mientras que las redes neuronales artificiales actúan como sistemas de procesamiento distribuido y pueden aprender de las relaciones entre las entradas y las salidas, por lo que los modelos pueden ser lineales o no lineales [13].

El uso de sistemas neuro-difusos en el control de temperatura es una práctica que ha tomado fuerza e interés en los últimos años, sobre todo por su habilidad de controlar sistemas no lineales y multi-paramétricos, los cuales pueden ser muy difíciles (si no es que imposible) de modelar matemáticamente [14], aunado a su capacidad de aprendizaje. Este tipo de sistemas permiten mejorar la experiencia de usuario reduciendo la complejidad de utilización, aumentando las funciones y con ellas el confort y maximizando el desempeño funcional.

La implementación de un sistema neuro-difuso en control de temperatura suele basarse en la caracterización de rangos de temperatura por medio de reglas difusas, generando así funciones de pertenencia para describen la operación del sistema.

Como se muestra en la ***figura 2.4***, los diferentes estados del sistema están caracterizados por diferentes rangos de niveles de temperatura, los cuales han sido ajustados a conjuntos difusos. Debido a lo anterior, la funcionalidad y confiabilidad del sistema dependerá de que tan certero se haya definido el criterio de cada rango. Es aquí donde toman importancia las redes neuronales, ya que se utilizan para sintonizar adecuadamente los puntos que definen los límites de cada rango y pueden adecuarse a la mayoría de las situaciones.

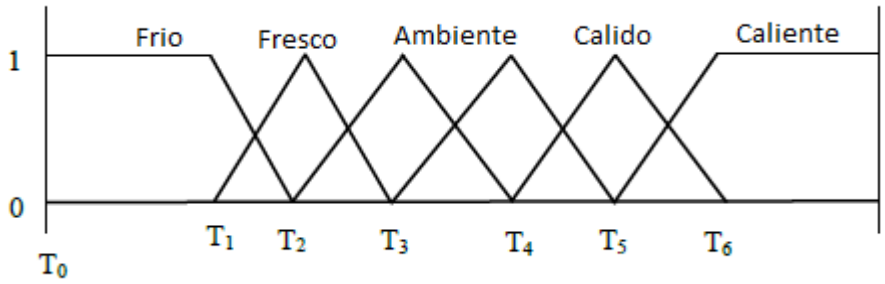

Figura 2.4 Función de pertenencia de la temperatura de entrada.

Este tipo de algoritmos híbridos ofrece la ventaja de modelar los sistemas de una manera sencilla tal que pueda implementarse en la mayoría de computadoras comerciales por medio de instrucciones IF-ELSE (gracias a su componente de lógica difusa), y su proceso de aprendizaje pueda derivarse de la auto-sintonización (gracias a su componente de redes neuronales).

2.3.2 Aprendizaje reforzado en aplicaciones HVAC

El modelar el problema de control HVAC como parte del proceso de decisiones de Markov, permite diseñar una solución la cual puede manejar efectivamente la incerteza ambiental. Así como en la mayoría de problemas de aprendizaje en el mundo real, no se tiene un conocimiento previo de un modelo ambiental completo, distribución de recompensas o transición de probabilidades. Sin embargo, métodos de aprendizaje reforzado como *Q-learning* pueden utilizarse para generar políticas óptimas en ausencia de modelos ambientales completos.

Q-learning pertenece a una colección de algoritmos denominados *métodos de diferencial temporal*. Al no requerir un modelo completo del ambiente, los métodos de diferencia temporal poseen una ventaja significativa y tienen la capacidad de realizar predicciones incrementalmente, sin embargo, puede requerir experiencia significativa con un ambiente dado para obtener un buen resultado.

Un método de aplicación de *Q-learning* a sistemas HVAC consiste en enmarcar el entorno como un proceso de decisión de Markov. La idea es mantener bajo el número de estados y acciones para que el problema recaiga en los límites de manejabilidad [15].

Las acciones de control en un sistema HVAC son ejecutadas en intervalos de tiempo discreto conocidos como épocas. Al final de cada época, el agente de aprendizaje observa el estado actual del entorno y escoge entre realizar o no una acción automatizada de HVAC. Las recompensas obtenidas por el agente de aprendizaje son entonces distribuidas de acuerdo a ciertos escenarios que surjan.

El capítulo III presenta información referente al desarrollo del proyecto y los experimentos realizados para su validación.

III. CAPÍTULO III. METODOS Y MATERIALES UTILIZADOS

3.1 Diseño de la investigación

Durante el desarrollo del proyecto fueron utilizados diferentes técnicas, métodos, aplicaciones y materiales para lograr la implementación de las distintas funcionalidades requeridas, las cuales fueron identificadas como necesidad al momento de realizar la investigación sobre el estado del arte.

Una vez identificadas y listadas todas las necesidades, se plantearon propuestas de solución para cada uno de los problemas asociados con cada punto involucrados, las cuales han sido puestas a prueba y validadas con base en la experimentación. Durante este proceso, algunas de las soluciones han funcionado adecuadamente durante su implementación al primer intento sin mayor problema, mientras otras han tenido que evolucionar desde una concepción inicial para lograr, si no la funcionalidad deseada, una muy cercana a ésta. Es debido a todo lo anteriormente mencionado que el propósito de esta sección es el describir aquellos métodos y materiales que han sido seleccionados y utilizados para la realización del proyecto.

Cada módulo funcional requiere de ciertos componentes específicos de *hardware* y *software* para operar correctamente. Por lo tanto, una manera de presentar consistentemente los métodos y materiales es navegar a través de los distintos módulos que componen el sistema e ir desglosando sus partes, para de esta manera entender a detalle su funcionamiento.

Los módulos que se revisan en este apartado son los siguientes:

- Servidor de Temperaturas.
- Base de datos de temperatura.
- Interface servidor – unidad remota.

- Agente de interfaz gráfica.
- Agente de verificación de temperatura seleccionada.
- Agente de verificación de hora de cambio.
- Agente de verificación de rutina.
- Agente de operación en base a termostato.
- Periférico de adquisición de temperatura ambiente.
- Periférico de accionamiento de cargas.

La figura 3.1 muestra de manera simplificada la interacción entre los componentes base del sistema.

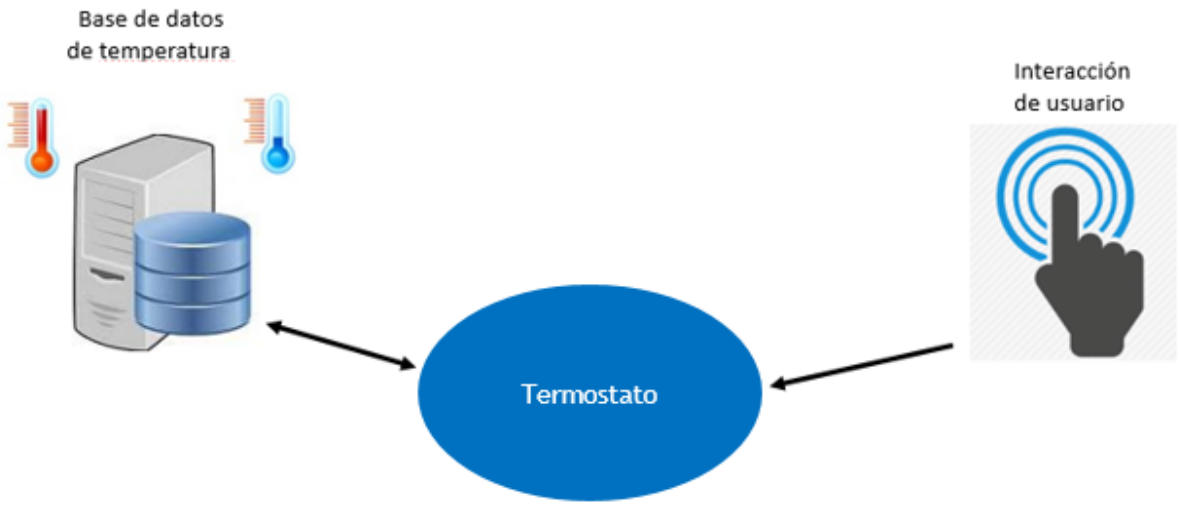

Figura 3.1 Interacción entre los componentes del sistema.

3.2 Experimentos y análisis

En los apartados siguientes se describirá la manera sobre cómo está formado el sistema y la funcionalidad específica de cada uno de sus componentes, lo cual es de suma importancia para comprender la interacción entre los diferentes agentes y como afectan el desempeño total del sistema.

Una vez que los distintos componentes del sistema han sido integrados para un objetivo en común, es necesario evaluar (bajo diferentes condiciones de funcionamiento) el desempeño y resultados de las partes, para determinar de esta manera la efectividad de la solución propuesta al problema descrito previamente en los capítulos I y II.

Las diferentes pruebas y validaciones que se realizan tienen como objetivo monitorear el comportamiento de cada uno de los algoritmos aquí definidos, y de esta manera verificar su desempeño en conjunto.

3.2.1 Servidor de Temperaturas

Incluida en las funcionalidades del proyecto, se encuentra la característica del sistema de disminuir el consumo de energía utilizado durante la operación del equipo. Actualmente existen diferentes métodos y técnicas para lograr este objetivo [15] [16] [17] [18] [19] [20], dentro de los cuales se encuentran las siguientes opciones principalmente:

1- Sensado de espacios: se basa en la utilización de sensores de presencia de personas para determinar cuando algún espacio o área se encuentra ocupado y solo bajo esta condición se enciende el equipo, de otra manera se encuentra siempre apagado.
2- Variación de punto de operación: se basa en la modificación de la temperatura a alcanzar dependiendo de la carga térmica del sistema (cantidad de usuarios en un espacio). Generalmente requiere una caracterización del flujo de personas en un edificio y su distribución térmica para una operación correcta.
3- Encendido por horarios: se basa en configurar el sistema para encender y apagar por determinado tiempo. Se requiere conocer con anterioridad la rutina del usuario en el espacio de interés.

Para la implementación de esta funcionalidad en el proyecto, se optó por diseñar una técnica novedosa la cual pudiera ofrecer alguna ventaja contra la operación normal de un termostato (se define como operación normal aquel escenario en el cual el usuario enciende y apaga el equipo manualmente y este último controla la temperatura alrededor de un nivel seleccionado).

La propuesta para esta funcionalidad contiene dos componentes principales:

1- Comparación estadística del punto de operación por década.
2- Algoritmo de aprendizaje de rutina de usuario.

En este apartado se revisará el componente 1 (comparar estadísticamente el punto de operación). El componente 2 es descrito en apartados posteriores.

La comparación estadística del punto de operación por década consiste en comparar la temperatura de operación seleccionada por el usuario con límites estadísticos, los cuales han sido obtenidos a partir del monitoreo de temperatura en una zona determinada con una ventana de tiempo de 10 años atrás. De esta manera, el sistema es capaz de alertar al usuario en caso de seleccionar una temperatura que se encuentre fuera de límites estadísticos, generando así conciencia sobre el uso que se le está dando al equipo.

Como primera aproximación de solución, se planteó el utilizar un servicio *web* para obtener los datos del historial de temperatura y generar las estadísticas a partir de ellos. Sin embargo, los servicios *web* que fueron evaluados requieren suscripción por el servicio, lo cual se refleja en un costo para la elaboración del proyecto. Debido a la condición anterior, se decidió investigar metodologías alternas como solución para la implementación de esta funcionalidad.

Durante la investigación de métodos alternos se encontraron gran cantidad de servicios de meteorología que ofrecen descargar parte de sus documentos con registros de temperatura. Tomando ventaja de esta oportunidad, se decidió adquirir estos documentos para concentrar la información en un registro maestro.

Una vez obtenida la información, surgió la interrogante acerca de cómo acceder a ella y poder administrarla de manera eficiente, ya que es necesario realizar consultas a los registros de forma recurrente.

Debido a que en proyectos trabajados anteriormente se había tenido de manera similar la necesidad de manejar registros con grandes cantidades de información, se cuenta con un poco experiencia en este ámbito. Estas necesidades fueran previamente solventadas por medio de motores de bases de datos y, tomando en cuenta esta experiencia y los resultados obtenidos, se decidió crear un servidor remoto para montar en él una base de datos, en la cual estarían cargados todos los registros de temperatura que se han obtenido de los diferentes servicios de metrología. Así pues, es posible tener concentrada toda la información fuera de línea, disponible en todo momento, habilitando la posibilidad de acceder a ella de manera recurrente a través de un gestor de búsqueda.

La descripción acerca de los diferentes componentes para montar la base de datos en el servidor, así como el motor de búsqueda utilizado es descrita en apartados posteriores.

3.2.2 Base de datos de temperatura

Un servidor es, en esencia, simplemente un sitio al que se puede acceder para requerir algún tipo de servicio. Sin embargo, para que realmente sea útil, en él se debe alojar algún tipo de sistema el cual se encargue de gestionar los recursos para hacer uso efectivo de la información o de los servicios. En nuestro caso, que requerimos disponibilidad de datos alojados en el servidor (guardar la información y acceder a ella), se requiere crear una base de datos [22].

Existen varios motores de bases de datos de uso libre en el mercado lo cuales son capaces de llevar acabo esta actividad, dentro de los cuales se encuentran principalmente:

1- *Microsoft SQL.*
2- *Microsoft ACCESS.*
3- *MySQL.*
4- *Oracle Berkeley DB.*
5- *SQL lite.*
6- *PostgreSQL.*
7- *Apache CouchDB.*
8- *Hyper SQL.*
9- *Small SQL.*
10- *OpenQM.*

El motor de bases de datos que se decidió utilizar fue *Microsoft Access*. Los parámetros principales considerados para tomar esta decisión son los siguientes:

- Simplicidad (conocimiento previo del entorno y su uso).
- Facilidad de uso (intuitivo).
- Portabilidad (se puede migrar fácilmente de PC).
- Mínima configuración requerida por el PC para actuar como servidor.
- Se incluye como herramienta estándar en la mayoría de las versiones de Microsoft Office.

Una vez habiendo elegido el motor de bases de datos, se prosiguió a definir la estructura de datos requerida para alojar la información. Tomando en cuenta las necesidades del proyecto y el hecho de que el sistema tiene que comparar una temperatura dada (la seleccionada por el usuario) con límites de temperatura estadísticos (obtenidos de los servicios meteorológicos) se planteó el siguiente esquema para alojar la información:

- Número_de_registro.
- Fecha.
- Temperatura_Maxima.
- Temperatura_Minima.

Aunque la información obtenida de los servicios *web* es mayor que la contenida en estos campos, los parámetros que se han mostrado previamente son aquellos que fueron definidos como información de interés para la aplicación. El hacer uso de cantidades mayores de información no tiene ningún valor agregado para el sistema y sin embargo puede demeritar sus capacidades, ya que el resultado se vería reflejado en archivo de mayor tamaño, lo cual puede ocasionar un aumento en los tiempos de consulta.

La información contenida en los registros de la base de datos no se introduce directamente de cómo fue obtenida por medio del servicio *web*, sino que es necesario proporcionarle un formato adecuado y utilizar únicamente aquellos campos de interés que fueron definidos previamente. De esta manera, se reduce el tamaño total que utiliza la información almacenada y se cuenta únicamente con los datos que son útiles para el sistema, lo cual tiene un impacto positivo en la mejora de rendimiento durante la interacción cliente-servidor en los periodos de consulta. Es por ello que estos campos se convirtieron en los de mayor interés para las estadísticas de entrada del sistema. La estructura de los campos contenidos en la base de datos puede observarse en la ***figura 3.2***.

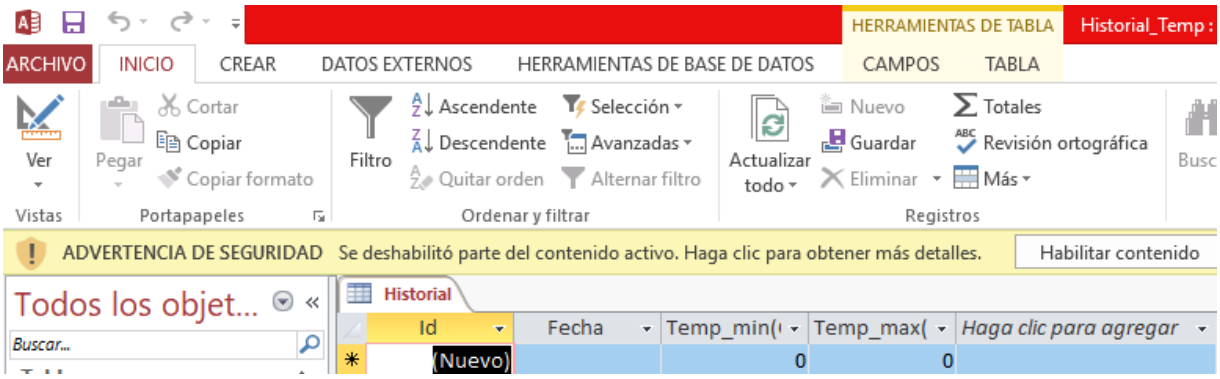

Figura 3.2 Base de datos.

3.2.3 Interface servidor-unidad remota

Como se comentó anteriormente, se decidió que las estadísticas de entrada del sistema fueran alojadas en un servidor externo a la unidad remota (la unidad remota es aquella con la cual tiene interacción el usuario, en este caso, el termostato). Ya que el termostato es quien decidirá el modo de operación del sistema con las características de ambiente configuradas por el usuario, es necesario que el servidor y el termostato se encuentren en constante comunicación. Para ello se definió una estrategia de crear un canal de comunicación.

La comunicación entre sistemas es tema bastante extenso, tal es así que existen una gran diversidad de métodos, protocolos e interfaces para la realización de este tipo de tareas. Dentro de los tipos de comunicación más comúnmente utilizados se encuentran los siguientes:

A- Comunicaciones alámbricas
   1- *RS232.*
   2- *I2C.*
   3- *SPI.*
   4- *PLC.*
   5- *Ethernet.*

B- Comunicaciones inalámbricas
   1- *WIFI.*
   2- *Bluetooth.*
   3- *Zigbee.*
   4- *Zwave.*
   5- *SigFox*.

Como primera aproximación y parte del proceso de validación y pruebas de sistema, se implementó un esquema de comunicación serial RS232. Para llevarlo a cabo, se realizó una topología del tipo servidor-unidad remota de manera provisional, la cual consiste en utilizar la interface USB tanto de la PC servidor como de la unidad remota como un puerto serie (modo emulación UART).

Al utilizar esta topología, la PC detecta a la unidad remota como un periférico serie RS232 y es posible enviar comandos seriales para enviar y recibir información. Esto permite a la unidad remota (termostato) realizar requisiciones de información sobre las estadísticas de temperatura al servidor y obtener de esta manera datos específicos sobre el punto de operación por parte del servidor.

La base de datos que se encuentra montada en el servidor no posee los medios, por si sola, para ejecutar una consulta de información requerida por la unidad remota a través de un puerto serie y regresar la información de la consulta por este mismo medio. Para ello, es necesario utilizar una aplicación intermedia corriendo del lado del servidor que reciba las necesidades del exterior, realice consultas a la base de datos y regrese los resultados de la consulta al exterior. Debido a estas necesidades, se ha realizado una aplicación en C# que realiza las funciones previamente descritas.

La razón de elegir C# para desarrollar la aplicación de gestión es la siguiente:

- Sintaxis es similar a C, el cual se está utilizando para crear el código de la unidad remota y se está buscando comunizar en medida de lo posible.
- Ambiente visual, reduce el tiempo de desarrollo.
- Provee acceso a periféricos, como puertos seriales.
- *Microsoft* provee versiones gratuitas de la herramienta.
- Amplia comunidad de ejemplos y ayuda en internet.
- Se cuenta con conocimiento del lenguaje y las herramientas.

Para el funcionamiento de la interface, primero se debe seleccionar un puerto serie (el numero dependerá del puerto USB donde se conecte la unidad remota) y después se debe presionar el botón "*Open Port*". Una vez hecho esto, se abrirá el canal de comunicación entre ambos agentes y podrá fluir la información de acuerdo a la demanda.

La consulta siempre es iniciada por la unidad remota, la cual envía un mensaje de requisición de información con el formato TYYYYMMDD, en donde:

- T es el prefijo identificador para temperatura.
- YYYY son los dígitos para año.
- MM son dígitos para mes.
- DD son dígitos para día.

Los parámetros Y, M y D son representativos del día en que se está realizando la consulta de información.

La respuesta a las consultas con este tipo de identificadores será un mensaje con el formato "MAXxx, MINyy", donde:

- MAX prefijo para el valor máximo de temperatura.
- MIN prefijo para el valor mínimo de temperatura.
- "xx" y "yy" son los datos que indican el valor numérico de la temperatura para cada prefijo.

La interface cuenta con un campo nombrado "*Data Sniffer*". El cual tiene el propósito de mostrar todos los comandos recibidos, así como la información enviada (es sumamente útil al momento de estar realizando pruebas con el sistema y además puede ser utilizado como bitácora de transferencias). Las ***figuras 3.3*** y ***3.4*** muestran la aplicación de transferencia de datos y la interconexión de la tarjeta de desarrollo con el servidor respectivamente.

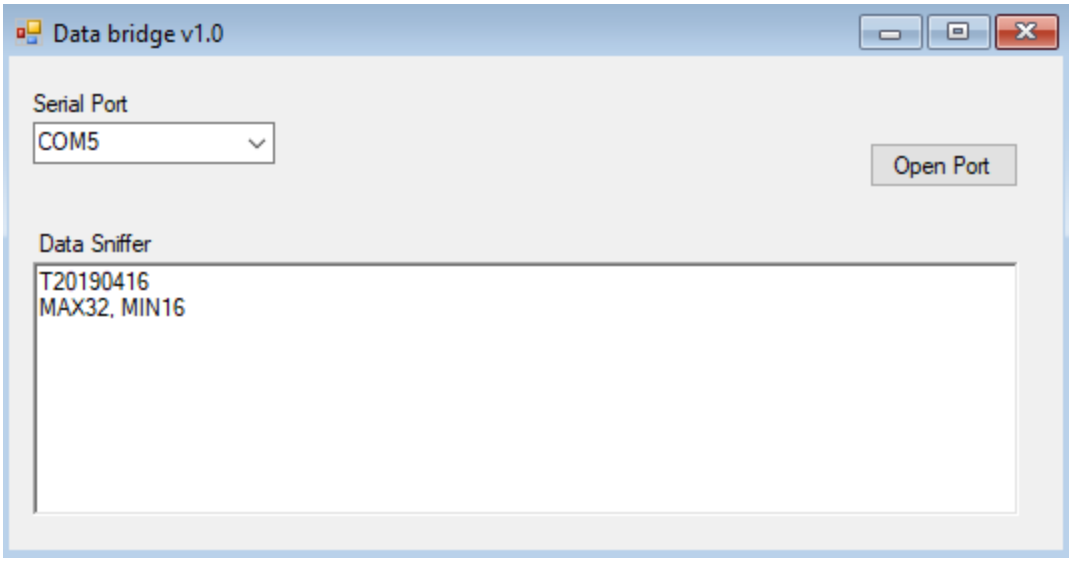

Figura 3.3 Vista de la interface servidor-unidad remota.

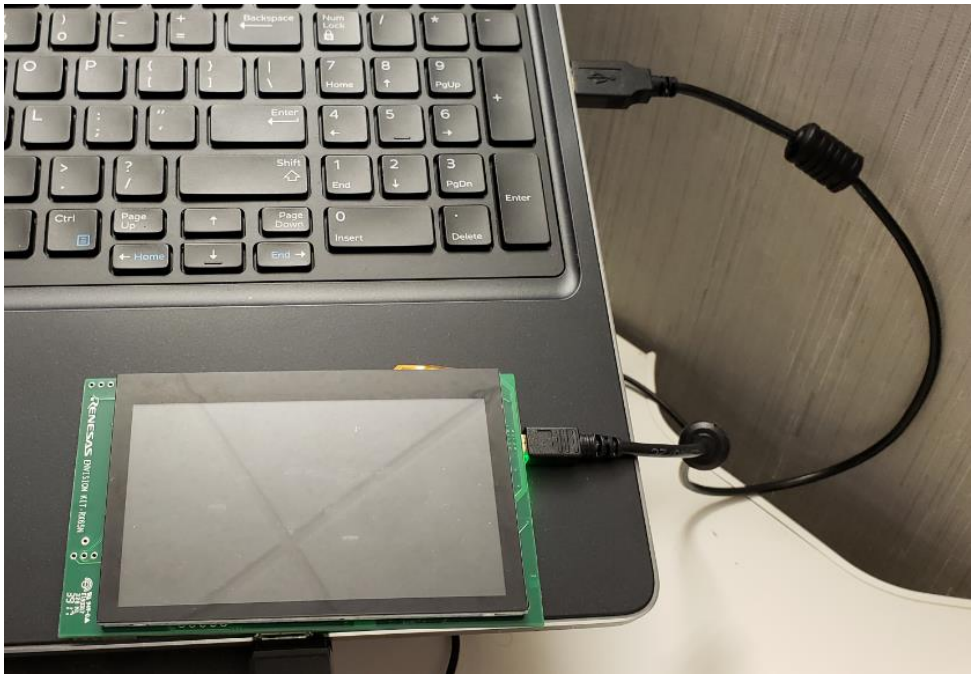

Figura 3.4 Conexión por puerto serie virtual entre tablilla de desarrollo y PC servidor.

La comunicación con esta interface demostró ser funcional y cumple con el objetivo. Sin embargo, en la práctica la instalación del termostato puede realizarse casi en cualquier parte de un edificio. Debido a esta condición, utilizar algún tipo de comunicación alámbrica requiere el tendido de líneas de trasmisión (usualmente cables) desde la compuerta de acceso al servidor hasta el termostato. Si en la instalación se tienen más de un termostato, las líneas de transmisión tendrían que ser enviadas desde el concentrador hasta cada termostato, traduciéndose en una instalación compleja y poco flexible.

Debido a las desventajas previamente mencionadas, se decidió que el método a utilizar para la comunicación entre servidor y unidad remota debía ser inalámbrico.

Para implementar el análogo inalámbrico de la interface, se comenzó por definir el tipo de comunicación adecuado para el sistema. Debido a que la mayoría de los termostatos conectados en el mercado actual utilizan comunicación WIFI, se decidió utilizar un módulo de expansión que permitiera agregar esta funcionalidad a la tablilla de desarrollo, por lo que se prosiguió a investigar dispositivos de comunicación inalámbrica actualmente disponibles en el mercado. Las características que debía reunir el dispositivo son las siguientes:

A- Comunicación WIFI con servidor.
B- Comunicación serial con unidad remota (está característica se limita a los protocolos SPI, I2C o RS232 debido a que son estos los periféricos con los que cuenta la tarjeta de desarrollo).

Durante la investigación, se encontró un dispositivo que ha ganado popularidad en los últimos años para la implementación de proyectos de IoT. Este dispositivo es un microcontrolador con interface WIFI y *stack* de protocolo incluido, cuyo nombre comercial es ESP8266.

Existen diferentes variantes de módulos WIFI que incluyen el IC ESP8266, las cuales varían en el número de terminales disponibles de acuerdo a las funciones que sean requeridas. Un ejemplo de este módulo de muestra en la ***figura 3.5***.

La necesidad de comunicación de este proyecto se basa únicamente en el requerimiento de enviar y recibir datos por la red WIFI. Debido a lo anterior fue seleccionada una versión de módulo de comunicación la cual fuera capaz de ser conectada a través de un puerto de comunicación serial (en este caso, RS232) para que fuera posible realizar la interface de comunicación con la tarjeta de desarrollo Renesas.

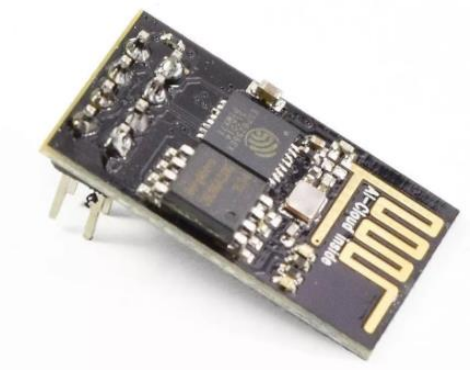

Figura 3.5 Modulo de comunicación WIFI ESP8266.

El utilizar este módulo permite habilitar al sistema remoto para poder obtener información de las estadísticas de temperatura del servidor necesarias para la correcta funcionalidad del algoritmo de operación. Sin embargo, la aplicación utilizada anteriormente (esquema de comunicación alámbrica) no pudo ser utilizada debido a la naturaleza del tipo de conexión, por lo que se debió desarrollar una aplicación similar, pero con la diferencia de actuar como servidor TCP/IP (ya que tanto el servidor como la unidad remota estarán conectados a la misma red inalámbrica). La conexión entre la tarjeta de desarrollo y el modulo WIFI puede observarse en la ***figura 3.6***.

Figura 3.6 Interface entre modulo WIFI y tarjeta de desarrollo.

El desarrollo de la aplicación cliente/servidor inalámbrico basada en protocolo TCP/IP, al igual que la aplicación alámbrica, fue realizado en C# debido al conocimiento previo de este lenguaje para de esta manera reducir el tiempo de desarrollo. La interfaz de usuario de esta aplicación se muestra en la ***figura 3.7***.

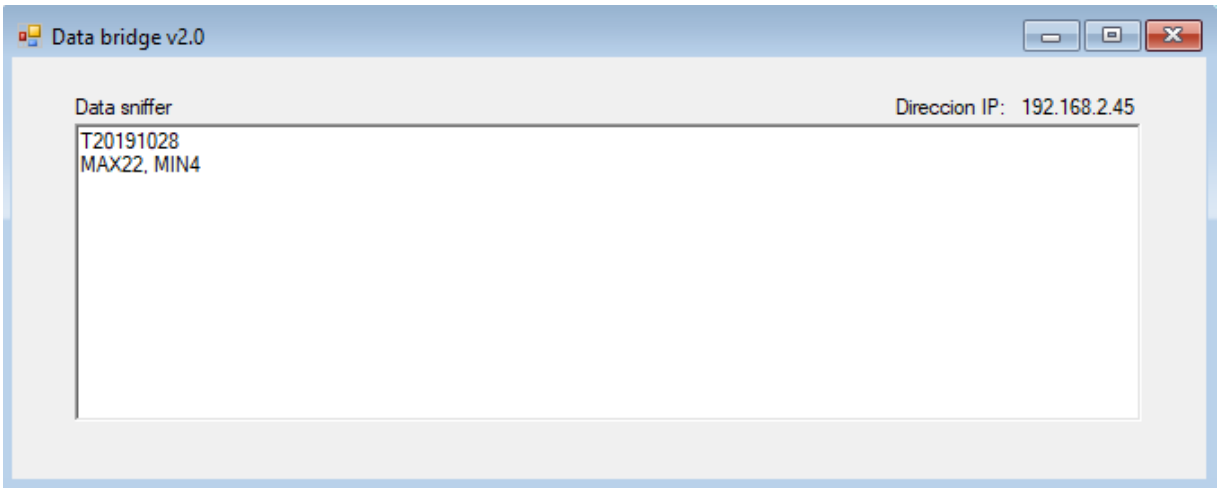

Figura 3.7 Vista de la interface tipo cliente/servidor entre servidor-unidad remota.

Debido a las necesidades de la aplicación, esta última interface fue la seleccionada para ser incluida en el proyecto.

3.2.4 Agente de interfaz gráfica

Una cuestión de suma importancia para la implementación del proyecto es el tema de la interfaz de usuario, ya que ésta permitirá al usuario modificar las características de operación del sistema.

La tarjeta de desarrollo de Renesas cuenta con una pantalla táctil integrada de resolución de 480x270 pixeles, lo que permite desplegar información al usuario y obtener información de su parte.

Cada uno de los pixeles en la pantalla soporta el formato de color RGB565 (Red, Green, Blue), el cual consiste en que el pixel es manejado por un registro de 16 bits, cuyo contenido se encuentra particionado para la configuración de color de la siguiente manera:

- 5 bits para la representación del color rojo (R).
- 6 bits para la representación del color verde (G).
- 5 bits para la representación del color azul (B).

La distribución de color en el registro de control de la pantalla grafica se muestra en la ***figura 3.8***.

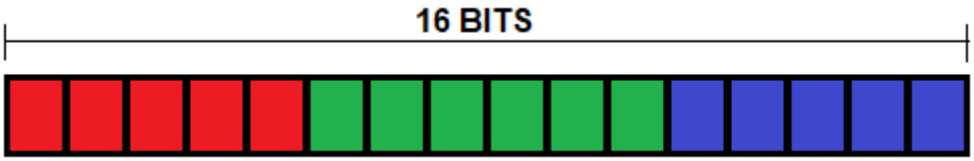

Figura 3.8 Distribución de la información del registro de color de pixeles.

La combinación en el valor de este registro da lugar a un color especifico en la luz emitida por la pantalla en un área determinada (el área dependerá del pixel configurado) la cual es observada por el usuario, como se observa en la ***figura 3.9***.

Para que sea posible mostrar información en pantalla, es necesario configurar el color de encendido de cada pixel por separado y hacer un barrido por determinadas áreas de la pantalla para formar patrones, de tal manera que estos patrones en conjunto tengan algún sentido para el usuario (una imagen, una letra, una figura, entre otros).

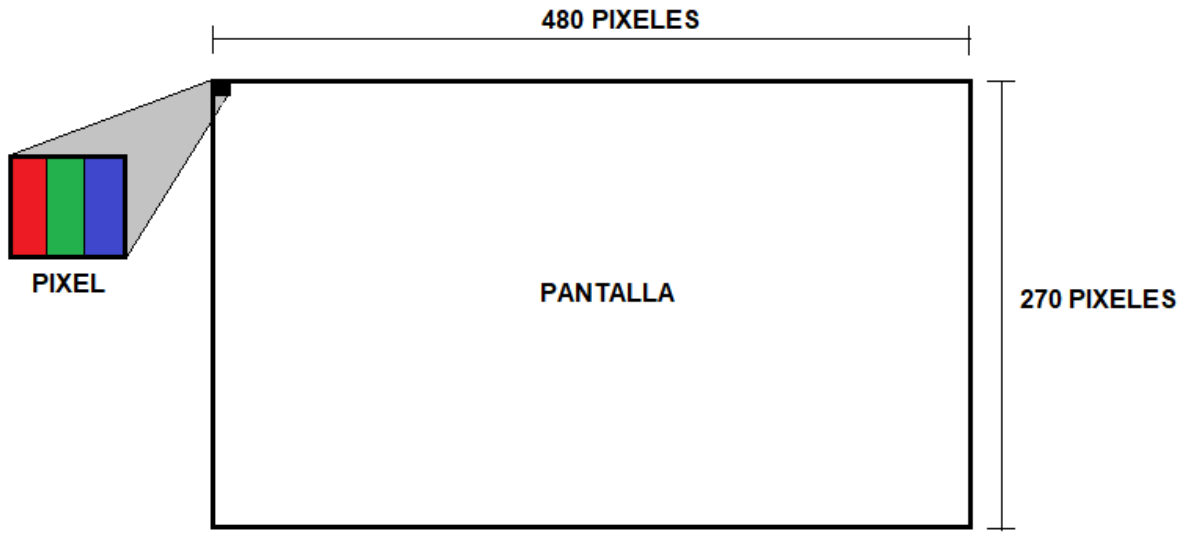

Figura 3.9 Vista de la interface tipo cliente/servidor entre servidor-unidad remota.

Debido que mostrar información requiere configurar pixel por pixel para formar un patrón deseado, es necesario crear una función que se encargue de configurar el registro de color para un pixel dado de acuerdo a la coordenada que ocupa en la pantalla. Esta función se muestra en la ***figura 3.10***.

```
void lcd_pixel(int16_t x, int16_t y, uint32_t colour)//uint32_t colour
{
    uint16_t* address = (uint16_t*)0x00800000;
    address += x;
    address += y * 480;
    *address = (uint16_t)colour;//converted_colour
}
```

Figura 3.10 Función utilizada para configurar los pixeles.

Ya que esta función es utilizada para configurar el valor de color de cada pixel individualmente, tiene que ser llamada el número de veces necesarias de acuerdo a la cantidad de pixeles que se deseen configurar.

Las imágenes utilizadas en la interfaz de usuario son imágenes JPG que se convierten a mapa de bits de 16 bits, de tal manera que se tiene un código de representación de color de 16 bits para cada uno de los pixeles de la imagen a mostrar (un ejemplo de una imagen convertida se muestra en la ***figura 3.11***). Estas imágenes se guardan en la memoria ROM del dispositivo, de tal manera que cuando sea necesario mostrarlas, se realiza un barrido por cada pixel de la imagen para obtener su valor de color el cual es pasado a cada pixel de la pantalla. Al finalizar el barrido de datos tanto de la imagen guardada en memoria como de los registros de la pantalla, esta última mostrara la imagen al usuario.

```
const unsigned short image[129600] ={
        0x0000, 0x0000, 0x0020, 0x0020, 0x0020, 0x0840, 0x0040, 0x0040, 0x0040, 0x0020, 0x0040, 0x0040, 0x0840, 0x0840, 0x0840, 0x0840,
        0x0860, 0x0861, 0x0881, 0x1081, 0x1081, 0x1081, 0x1081, 0x10A1, 0x10A1, 0x10A2, 0x10A2, 0x10C2, 0x10C2, 0x18C2, 0x10C2, 0x10C2,
        0x10A2, 0x10A2, 0x10A2, 0x10A2, 0x10A2, 0x10A2, 0x10A1, 0x10A1, 0x10A1, 0x10A1, 0x10A2, 0x10C2, 0x18C2, 0x10C2, 0x10C2, 0x18C2,
        ..........
        0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800,
        0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800, 0x0800,
        };
```

Figura 3.11 Ejemplo de imagen guardada en memoria.

La ***figura 3.12*** muestra la función de barrido de los registros de memoria y pantalla codificada en C.

```
for(pixy = 0; pixy < 270; pixy++)
{
    for(pixx = 0; pixx < 480; pixx++)
    {
        lcd_pixel(pixx, pixy, image[pix]);
        pix++;
    }
}
```

Figura 3.12 Ejemplo de código para mostrar la imagen desde la memoria a la pantalla.

Al ejecutar el código, la imagen desplegada en pantalla se mostrará tal y como se puede observar en la ***figura 3.13***:

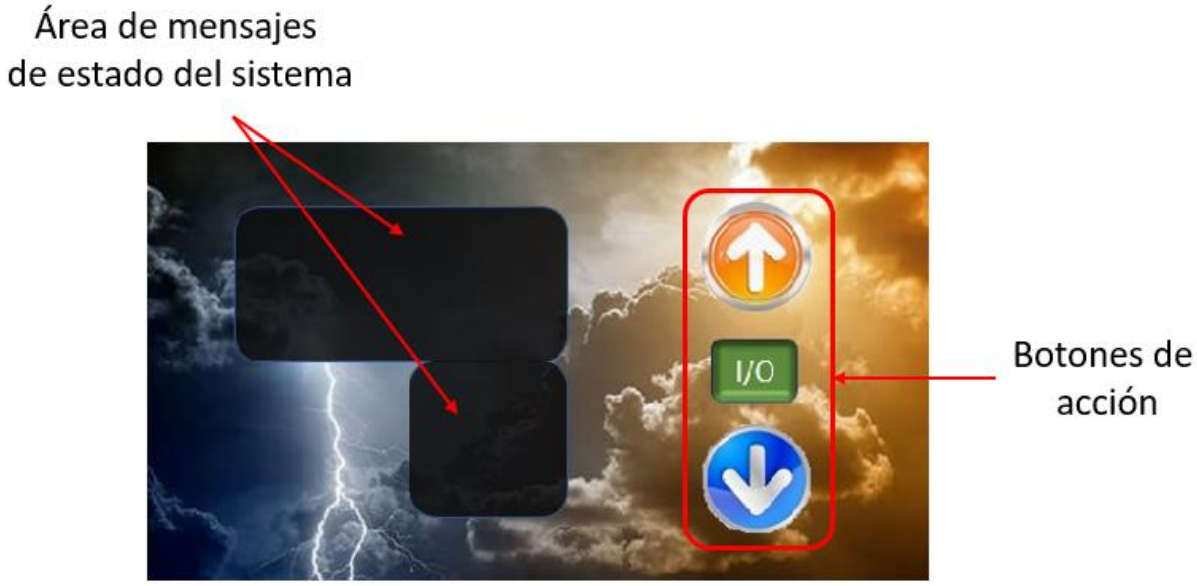

Figura 3.13 Interfaz de usuario.

3.2.5 Agente de verificación de temperatura seleccionada

Como se ha venido comentado anteriormente, parte de la funcionalidad básica del sistema es ser capaz de determinar cuando la temperatura seleccionada por el usuario no se encuentra dentro de un rango definido por los límites estadísticos históricos [23].

La implementación de esta funcionalidad se lleva a cabo por medio del diseño de un algoritmo dedicado el cual tiene como función adquirir la fecha actual, así como la temperatura de interés al momento que el usuario realiza un cambio en la configuración del sistema. Al identificar estos datos, la información es empaquetada en forma de comando con el formato TYYYYMMDD, en donde Y, M y D son representativos del día en que se está realizando la consulta de información:

- T es el prefijo identificador para temperatura.
- YYYY son los dígitos para año.
- MM son dígitos para mes.
- DD son dígitos para día.

De esta manera, la consulta es enviada al servidor para ser atendida, y así recibir de vuelta en la unidad remota una respuesta que contenga los valores máximos y mínimos de temperatura. La información recibida es comparada entonces con el nivel de temperatura configurada por el usuario (punto de operación seleccionado) para así determinar si la selección del usuario se encuentra dentro o fuera de los límites estadísticos históricos.

La interacción entre los límites estadísticos de temperatura y el nivel configurado por el usuario puede observarse en la ***figura 3.14***.

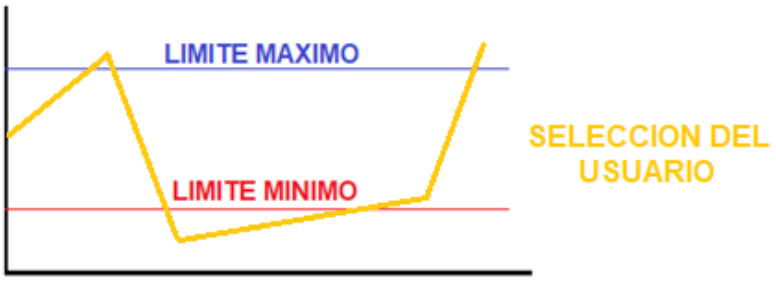

Figura 3.14 Interacción entre los datos estadísticos obtenidos del servidor y la selección del usuario.

Una vez que el sistema ha determinado el punto en que se encuentra la selección del usuario con respecto a los límites estadísticos, existirán dos escenarios:

A- La temperatura se encuentra dentro de los límites estadísticos.
B- La temperatura se encuentra fuera de los límites estadísticos (ya sea por arriba o por debajo de éstos).

A continuación, se menciona la descripción de cada escenario de operación.

3.2.5.1 Temperatura dentro de límites estadísticos

Tan pronto como el sistema determina que la temperatura seleccionada por el usuario se encuentra dentro de los límites estadísticos, el sistema comienza a operar y se detiene una vez que alcanza el punto de operación. No se requiere interacción adicional del usuario en este escenario.

3.2.5.2 Temperatura fuera de límites estadísticos

Si se determina que el punto de operación seleccionado por el usuario se encuentra por encima o por debajo de los límites estadísticos, el sistema muestra la notificación *"Temperatura fuera de límites estadísticos. ¿Desea apagar automáticamente el equipo al alcanzar el límite?"* al usuario, como se muestra en la ***figura 3.15***.

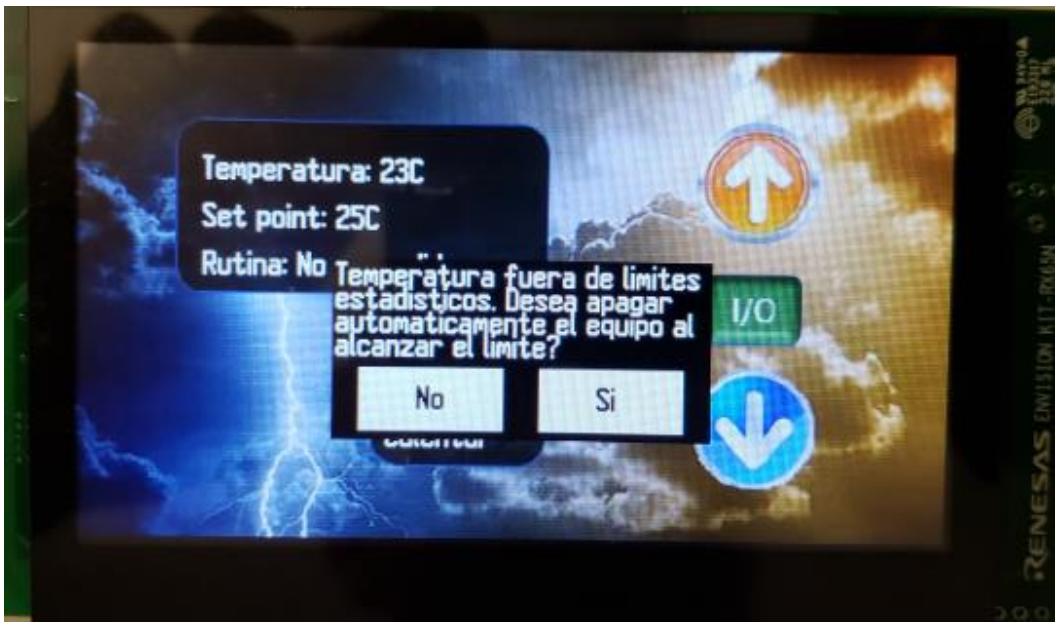

Figura 3.15 Notificación del sistema mostrada al usuario.

Al mismo tiempo que se muestra la notificación, el sistema muestra también dos opciones a seleccionar al usuario:

A- ***Opción No:*** El sistema sigue operando de manera normal incluso al pasar por los límites estadísticos. El sistema se detiene únicamente al alcanzar el punto de operación seleccionado por el usuario o si el usuario interrumpe la operación manualmente.

B- ***Opción Si:*** El sistema se detiene hasta alcanzar el punto de operación estadístico (ya sea el límite máximo en caso del modo “calentar” o el límite mínimo en caso del modo “enfriar”) o si el usuario interrumpe la operación manualmente.

Una vez que se selecciona cualquiera de las opciones en los escenarios descritos anteriormente, el sistema continua con su operación normal. La ***figura 3.16*** muestra un extracto de código del programa principal del sistema.

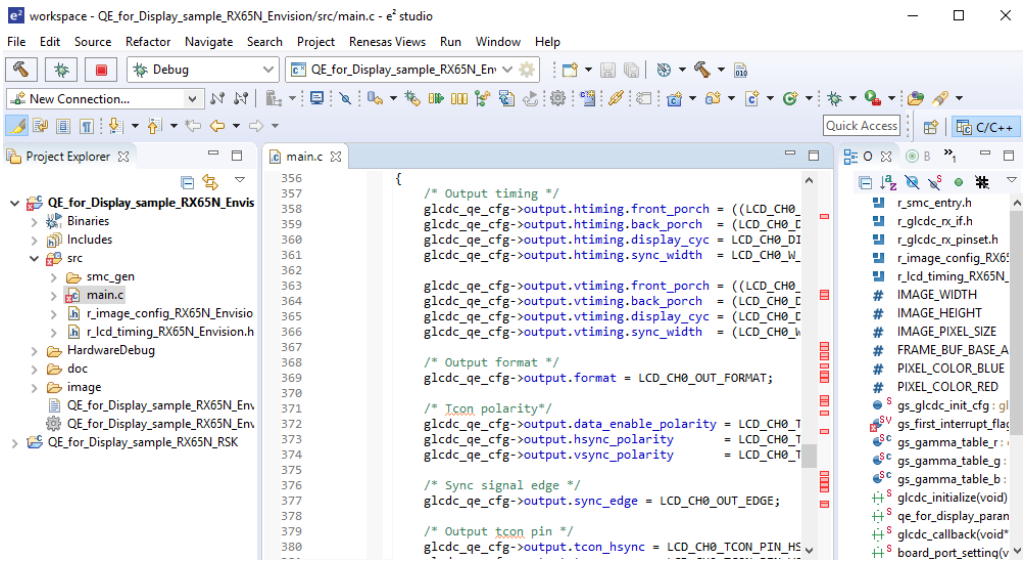

Figura 3.16 Extracto de código del programa en unidad remota.

La finalidad de este algoritmo es crear conciencia en el usuario acerca de los puntos de operación seleccionados para el sistema y de esta manera acortar el tiempo de funcionamiento ya que entre menor sea la brecha entre la temperatura deseada de

operación y la temperatura ambiente actual, el sistema se satisfacer más rápidamente y, por lo tanto, la pérdida de calor también será menor, resultando en una operación que no malgasta los recursos energéticos. La aseveración anterior está basada en la evolución que experimenta el estado de un sistema térmico La respuesta característica en estos sistemas es del tipo exponencial, cuyos límites de operación están determinados por las características del ambiente a controlar, dentro de las cuales se encuentran las siguientes:

- Potencia del equipo de calefacción / refrigeración.
- Tamaño del espacio.
- Aislamiento térmico de la edificación.
- Distribución de las salidas/tomas de aire.
- Carga térmica del sistema.

La ***figura 3.17*** muestra la evolución de un sistema el cual fue encendido durante 5 horas para calentar una edificación cuando se encontraba en 15^{o}C, con un punto de operación seleccionado de 35^{o}C. Puede observarse que el sistema incrementa su temperatura hasta el punto de estabilización de 28.5^{o}C aproximadamente en una hora, pero a partir de ese tiempo el sistema no es capaz de incrementar más la temperatura ambiente, aun cuando está programado para llegar a 35^{o}C. Esta limitante en la respuesta se debe a la combinación de los distintos factores listados anteriormente, por lo cual no existe una configuración genérica y es de suma importancia seleccionar estos parámetros adecuadamente para cada instalación [23] [24].

Una vez que se determina el valor de la temperatura de entrada, se ejecuta un algoritmo cuya función es delimitar el punto de operación del sistema. Esto es necesario para acotar el rango de acción del algoritmo de aprendizaje a ciertos valores de temperatura, ya que de lo contrario sería muy complicado (y poco probable) llegar a establecer una rutina para el usuario.

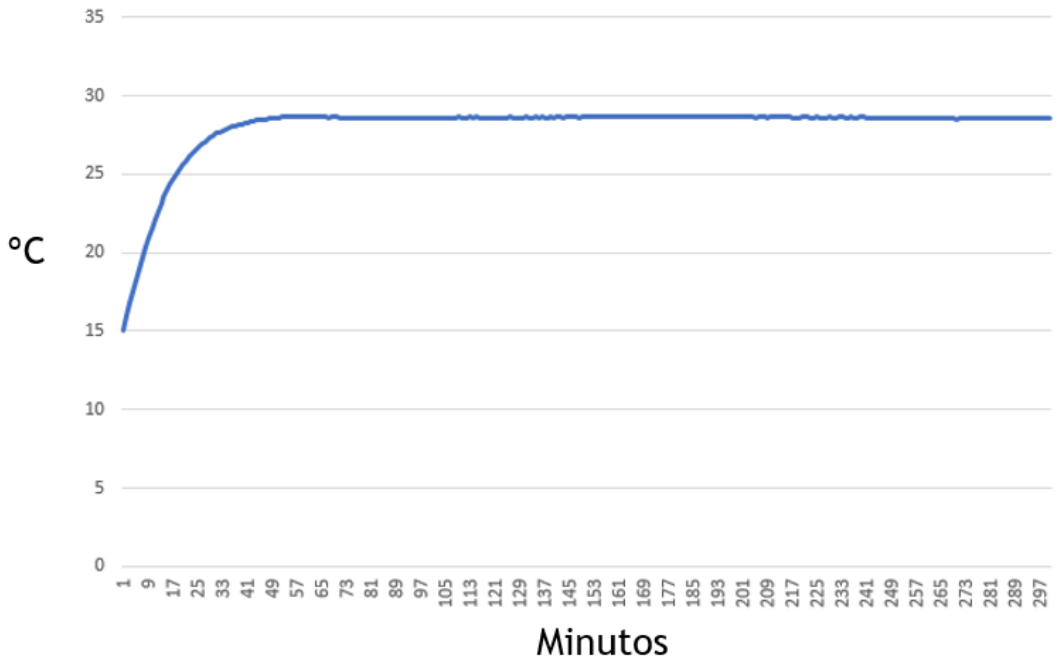

Figura 3.17 Ejemplo de respuesta característica de un sistema térmico.

3.2.5.3 Delimitación de temperatura seleccionada

Para limitar la cantidad de niveles de temperatura que puede seleccionar el usuario y evitar tener un espectro de posibilidades bastante amplio, se decide limitar el rango de acción de los valores de temperatura seleccionados por el usuario. Cabe mencionar que el usuario puede seleccionar cualquier temperatura de interés, sin embargo, el sistema encasilla internamente la temperatura seleccionada a alguna de las clasificaciones previamente definidas.

Para efectos de prueba e implementación rápida, se definen 5 clasificaciones de temperatura, las cuales se describen a continuación:

- Muy frio.
- Frio.
- Ambiente.
- Caliente.
- Muy caliente.

Es posible cambiar la lógica de programación e ingresar mayores clasificaciones al sistema, aunque para efectos de funcionalidad y desempeño, cualquier temperatura que seleccione el usuario se encasilla dentro de estas 5 clasificaciones.

Para definir el criterio de clasificación de temperatura, se realiza una encuesta a 100 personas en la cual se pregunta acerca de los niveles de temperatura que ellos consideran para cada clasificación (muy frío, frío, ambiente, caliente, muy caliente) dentro de una edificación (casa, oficina, etcétera). El objetivo de esta encuesta es determinar las características de operación del sistema en base a la experiencia y gustos de los usuarios potenciales, para de esta manera tener ajustado el sistema a lo que los usuarios denominan punto de *confort*. Conociendo esta información, es posible definir y codificar una función que seccione el espectro la temperatura en clasificaciones de acuerdo a los gustos de los usuarios [24].

Las preguntas que se realizan en la encuesta se encuentran en la *tabla* ***A.2*** de la sección A*péndices.*

En base a los resultados obtenidos durante la realización de la encuesta ***A.2***, se definió que las clasificaciones de temperatura tuvieran una separación de 4ºC, siendo el rango como se muestra a continuación:

- Muy frío – Temperaturas menores a 15.5ºC.
- Frío – Temperaturas entre 15.5ºC y 19.5ºC.
- Ambiente – Temperaturas entre 19.5ºC y 23.5ºC.
- Caliente – Temperaturas entre 23.5ºC y 27.5ºC.
- Muy caliente – Temperaturas mayores a 27.5ºC.

Como puede observarse, para las clasificaciones de los extremos (“Muy frío” y “Muy caliente”) solo existe el límite superior y el límite inferior. Para las categorías intermedias (“Frío”, “Ambiente” y “Caliente”), existe un límite bajo y un límite alto. Si se

grafica el comportamiento de cada una de las clasificaciones [36], se obtienen los diagramas mostrados en las ***figuras 3.18, 3.19, 3.20, 3.21*** y ***3.22*** respectivamente.

$$\mu_{MF}(T) \begin{cases} 1 & T \leq 15 \\ \dfrac{16 - T}{16 - 15} & 15 < T < 16 \\ 0 & T \geq 16 \end{cases}$$

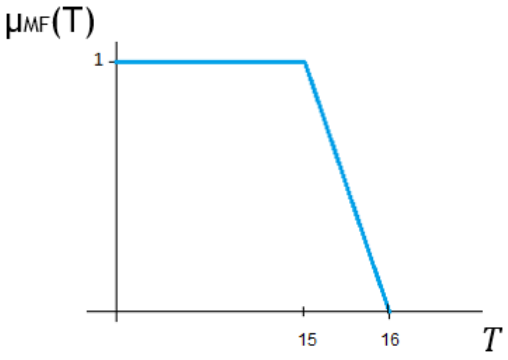

Figura 3.18 Respuesta de la función de pertenencia *"Muy frío".*

Donde:

- μMF(T) es la función de pertenencia "Muy frío".
- T es la temperatura seleccionada por el usuario.

$$\mu_{F}(T) \begin{cases} 0 & T < 15 \ o \ T > 20 \\ \dfrac{T - 15}{16 - 15} & 15 < T < 16 \\ 1 & 16 < T < 19 \\ \dfrac{20 - T}{20 - 19} & 19 < T < 20 \end{cases}$$

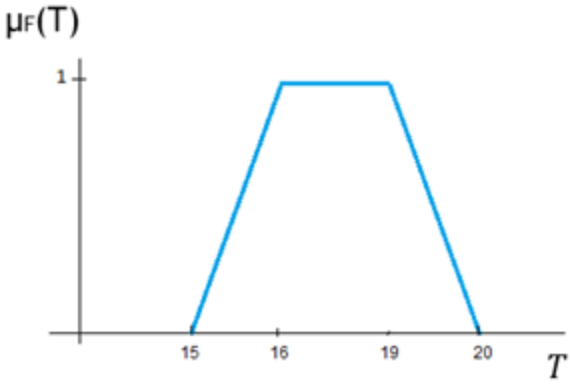

Figura 3.19 Respuesta de la función de pertenencia *"Frío".*

Donde:

- μF(T) es la función de pertenencia "Frío".
- T es la temperatura seleccionada por el usuario.

$$\mu_A(T) \begin{cases} 0 & T < 19 \ o \ T > 24 \\ \frac{T - 19}{20 - 19} & 19 < T < 20 \\ 1 & 20 < T < 23 \\ \frac{24 - T}{24 - 23} & 23 < T < 24 \end{cases}$$

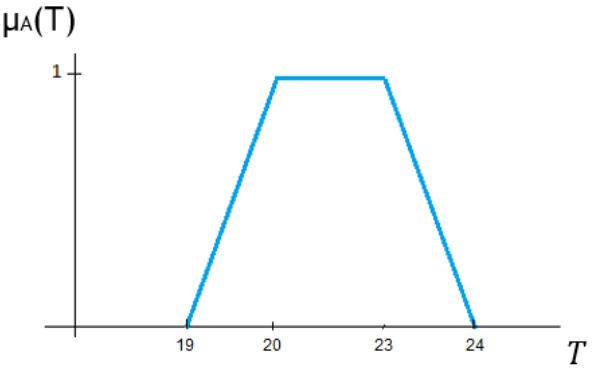

Figura 3.20 Respuesta de la función de pertenencia *"Ambiente".*

Donde:

- μA(T) es la función de pertenencia "Ambiente".
- T es la temperatura seleccionada por el usuario.

$$\mu_C(T) \begin{cases} 0 & T < 23 \ o \ T > 28 \\ \frac{T - 23}{24 - 23} & 23 < T < 24 \\ 1 & 24 < T < 27 \\ \frac{28 - T}{28 - 27} & 27 < T < 28 \end{cases}$$

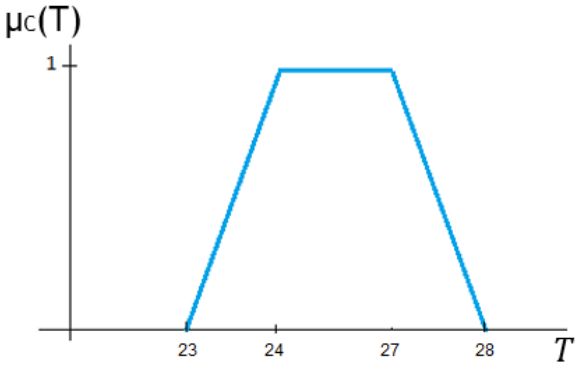

Figura 3.21 Respuesta de la función de pertenencia *"Caliente".*

Donde:

- μC(T) es la función de pertenencia "Caliente".
- T es la temperatura seleccionada por el usuario.

$$\mu_{MC}(T) \begin{cases} 0 & T < 27 \\ \dfrac{T-27}{28-27} & 27 < T < 28 \\ 1 & T > 28 \end{cases}$$

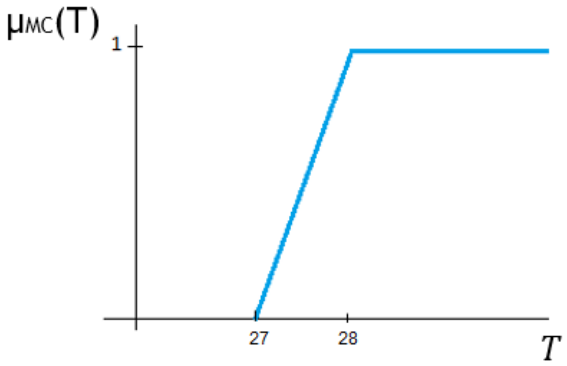

Figura 3.22 Respuesta de la función de pertenencia *"Muy caliente".*

Donde:

- μMC(T) es la función de pertenencia "Muy caliente".
- T es la temperatura seleccionada por el usuario.

Basados en las descripciones anteriores, el diagrama que muestra la interacción entre las diferentes categorías se muestra en la ***figura 3.23***:

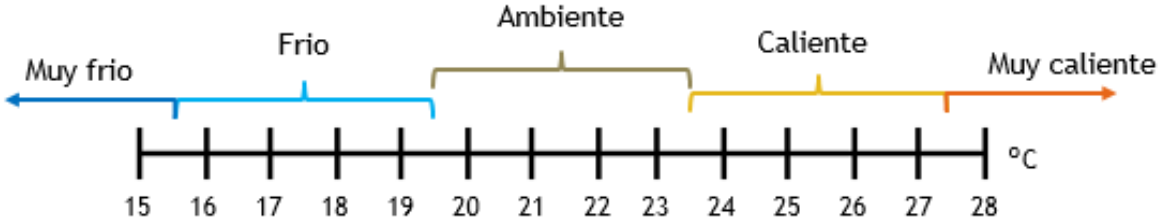

Figura 3.23 Interacción entre funciones de pertenencia de temperatura.

La temperatura media para cada una de las categorías es la siguiente:

- Muy frío – 15ºC.
- Frío – 17.5ºC.
- Ambiente – 21.5ºC.
- Caliente – 25.5ºC.
- Muy caliente – 28ºC.

Estas clasificaciones de temperatura son las que se utilizan como la selección del usuario al momento de que se realiza la interacción entre el usuario y el sistema [25]. Sin embargo, si al momento de realizar el cambio de temperatura el sistema muestra la alerta sobre los límites estadísticos de temperatura al usuario, la temperatura histórica tendrá prioridad si el usuario selecciona la opción de utilizarla como punto de operación del sistema. El extracto de código de la ***figura 3.24*** muestra la codificación en lenguaje C de la función de clasificación de temperatura.

```
int RangoTemp(TempU)
{
    if(TempU < 15.5)
    {
        Rango = 1;
        TempU = 15;
    }
    else if(TempU >= 15.5 && TempU < 19.5)
    {
        Rango = 2;
        TempU = 17.5;
    }
    else if(TempU >= 19.5 && TempU < 23.5)
    {
        Rango = 3;
        TempU = 21.5;
    }
    else if(TempU >= 23.5 && TempU <= 27.5)
    {
        Rango = 4;
        TempU = 25.5;
    }
    else if(TempU > 27.5)
    {
        Rango = 5;
        TempU = 28;
    }

    return Rango;
}
```

F**igura 3.24** Función de clasificación de temperatura por rango.

3.2.6 Agente de verificación de hora de cambio

Como se ha mencionado con anterioridad, el sistema tiene la capacidad de aprender sobre los gustos del usuario acerca de la configuración del sistema de control de aire.

La apuesta del sistema para aprender de la rutina del usuario se basa en encontrar patrones durante los cambios de configuración. Para llevar esta funcionalidad a cabo, se desarrolla un agente el cual está encargado de verificar la hora en que cierta configuración es cambiada [21].

Es posible que el usuario cambie la configuración del sistema no necesariamente a la misma hora siempre [26], sino con diferencia de algunos minutos, lo que en primera instancia lleva a un número de posibilidades muy grande para la capacidad de aprendizaje. Debido a este motivo, es necesario desarrollar una función que “delimite” los períodos de acción u horas en las cuales el sistema es modificado.

Para definir el criterio de delimitación de horas, se realiza una encuesta a 100 personas en la cual se pregunta acerca de las horas en que normalmente se enciende y se apagan los dispositivos de control de aire en su casa. El objetivo de esta encuesta no es conocer la hora exacta en que sucedía esta actividad, sino obtener las fracciones de hora en las cuales suceden estos cambios (múltiplos de 5 minutos, media hora, horas completas, etcétera). Conociendo esta información, es posible realizar una función que delimite las horas en las que se cambia alguna configuración dependiendo el rango en que se realiza el cambio. Las preguntas realizadas en la encuesta se encuentran en la *tabla **A.1*** de la sección A*péndices.*

En base a los resultados que se obtienen en la encuesta anterior, se define que la delimitación de la hora se realiza de la siguiente manera:

- Hora en punto + / - 15 minutos = hora en punto.
- Media hora + / - 15 minutos = media hora.

Como ejemplo, si el usuario enciende o apaga el sistema entre 8:45 am y 9:15 am, la hora de cambio que toma el sistema como configuración es a las 9:00 am. En cambio, si el usuario cambia la configuración entre 9:16 am y 9:44 am, la hora de cambio que toma el sistema como configuración es a las 9:30 am. Esta delimitación permite al sistema trabajar con intervalos finitos, de lo contrario se tiene una combinación bastante amplia de horarios lo cual evita que el sistema pueda encontrar algún patrón en los cambios de preferencia del usuario.

Una vez definidos los requerimientos de delimitación, es necesario el encontrar una función que describa este comportamiento para el modelado y programación de la funcionalidad.

Si se grafica el comportamiento esperado de función delimitadora de hora, se obtiene el diagrama de la ***figura 3.25***.

Figura 3.25 Gráfica de la función delimitadora de horas.

La ***figura 3.25***, describe el caso de delimitación de un rango de horas a una hora en concreto. Es evidente la similitud entre esta gráfica y una función de pertenencia en lógica difusa, por tal motivo se selecciona este método para el modelado de la función del agente verificador de hora de cambio.

La definición de la función de pertenencia genérica para cualquier hora de cambio se puede representar como el diagrama de la ***figura 3.26***.

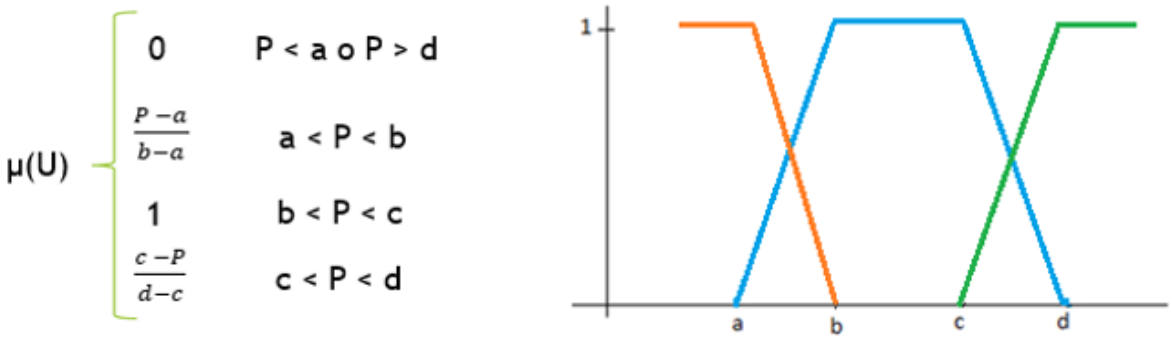

Figura 3.26 Gráfica de la función de pertenencia rutina de usuario.

Donde:

- μ(u) es la función de pertenencia rutina de usuario.
- a, b, c y d son los límites de hora en cada intervalo.
- P es la hora a la que el usuario realizó el cambio.

La gráfica azul representa la hora delimitada por el algoritmo difuso, mientras que las gráficas naranja y verde representan las horas adyacentes a la hora delimitada por el algoritmo. Puede observarse que, mientras más se avanza en el tiempo, va ganando prioridad la siguiente delimitación de hora.

En la figura ***3.27*** se muestra un extracto de código en lenguaje C el cual pertenece a la función codificada del algoritmo difuso de delimitación de hora:

```
void DelimH(int horaU, int minU)
{
    if (minU < 15)
    {
        horaR = horaU;
        minR = 0;
    }
    else if(minU > 45)
    {
        horaR = horaU + 1;
        minR = 0;
    }
    else
    {
        horaR = horaU;
        minR = 30;
    }
}
```

Figura 3.27 Algoritmo difuso de delimitación de hora codificado en C.

3.2.7 Agente de aprendizaje

El enfoque base del sistema se centra en el hecho de la capacidad de aprender las configuraciones realizadas por el usuario y ajustar el comportamiento en base a ellas. Para ello, fue necesario desarrollar un agente que provea la inteligencia necesaria al sistema [27].

El criterio definido para seleccionar el modelo óptimo de operación del sistema se basa en los siguientes parámetros:

A- El sistema define como hábito una operación realizada durante 7 días consecutivos. Esto toma en consideración la temperatura seleccionada, la hora de inicio de operación y la hora de paro de operación.

B- El sistema sobre-escribe el modo de operación con el nuevo gusto del usuario. Sin embargo, el gusto anterior es guardado en la base de datos del historial de cambios. Esto permite al sistema enriquecer su información para definir operaciones por temporada año con año.

Debido a que es posible que el usuario seleccione una misma temperatura de operación alrededor de una hora y no siempre a la misma hora exacta (diferencia de minutos y segundos) [26], es necesario ajustar el parámetro “hora” dentro de un número finito opciones (esta característica ha sido descrita anteriormente en la *sección 3.2.4*).

Una vez que el sistema ha detectado un cambio en su configuración, es necesario verificar si las nuevas configuraciones son parte de una rutina o no. Esta funcionalidad requiere de un algoritmo que tenga capacidad de memoria y decisión con respecto a eventos anteriores.

Durante el capítulo 1 de este documento, se describió la intención de utilizar métodos neuro-difusos para la implementación del proyecto. La componente de lógica difusa del sistema ha sido implementada como algoritmos de delimitación con la finalidad de clasificar y delimitar algunas características de operación, sin embargo, para el requerimiento de aprendizaje del sistema se decide utilizar un esquema diferente, el enfoque de redes neuronales.

Al utilizar una red neuronal en el sistema, se pretende que esta sea capaz de modificar su comportamiento actual y futuro dependiendo de las condiciones de operación que el usuario haya utilizado previamente [29] [30]. Se puede comentar que se busca que el sistema tenga “memoria” y aprenda de sus “experiencias” anteriores.

Para solventar los requerimientos del algoritmo de aprendizaje, se comienza por definir las características y necesidades que deben cumplirse. Por lo tanto, una definición del modelo para la implementación de la red neuronal se basa en las siguientes características:

- Valores de inicio (entradas).
- Modificador de ponderación.
- Valor de ajuste.
- Función de determinación de estado.

Tomando en cuenta los requerimientos anteriores, se observa que se asemejan bastante a las características del modelo del *perceptron*, lo cual se puede observar en la ***figura 3.28***:

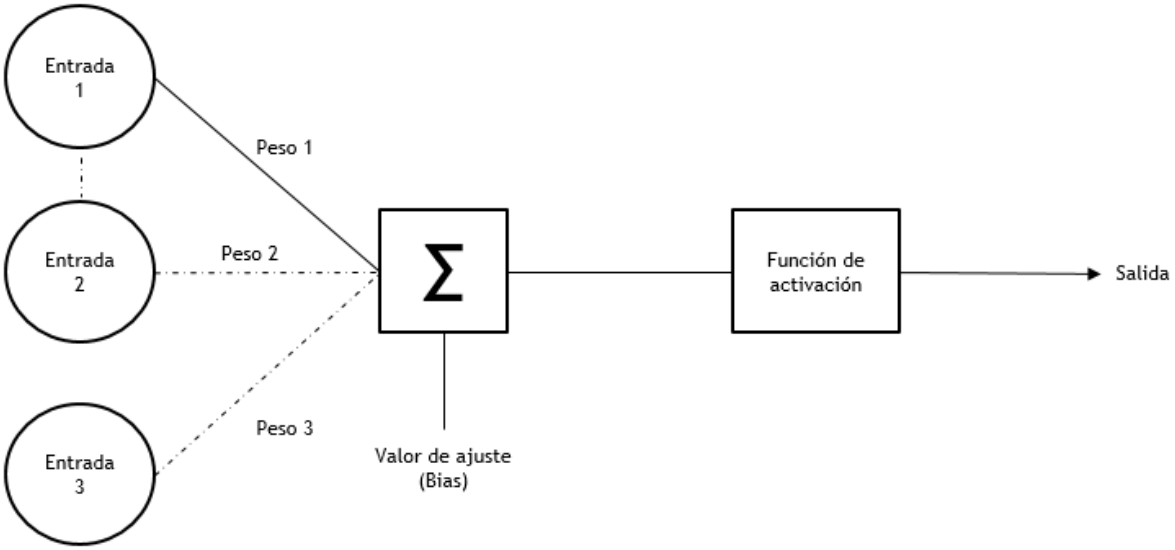

Figura 3.28 Modelo básico de un *perceptron.*

Se procede entonces a relacionar los parámetros del perceptron con los requerimientos de operación, lo que lleva a una primera aproximación en la implementación de la red neuronal [31]. Esta primera aproximación toma en cuenta las siguientes características:

- Cada vez que el usuario cambie la interacción del sistema se toma como la entrada de la red neuronal.
- El peso en el valor de la entrada es definido por la regla que determine una condición de rutina.
- El valor de ajuste debe permitir al perceptron conocer los estados anteriores.
- La función de activación debe indicar si se ha generado una nueva rutina o no, por lo que deberá ser simple.

Para la asignación del valor numérico a los puntos anteriores, se realizan los métodos que se describen a continuación [37].

3.2.7.1 Entrada

Cada ocasión que el usuario realiza una modificación en la temperatura de operación del sistema, este verifica el rango de hora y de temperatura que se ha seleccionado (de acuerdo a lo descrito en las funciones de lógica difusa de secciones anteriores). Si el rango de ambas características (hora y temperatura) de la interacción actual es el mismo que el rango seleccionado en la interacción anterior, el valor de entrada de la red neuronal se define como 1, de lo contrario se define como 0. De esta manera, cada vez que el usuario interacciona con el sistema, es una entrada para el algoritmo de red neuronal que ejecuta la funcionalidad del perceptron.

3.2.7.2 Peso

Debido a que la semana cuenta con 7 días, se opta por que la regla para determinar una rutina sea configurar la misma temperatura a la misma hora durante 7 días de operación consecutivos. De este modo, el valor del peso será determinado por la siguiente relación:

$$Bias = \frac{1}{Numero\ de\ dias} = \frac{1}{7} = 0.15 \qquad (3.1)$$

El valor numérico que se determina para el peso es fijo y modifica el valor de la entrada cada vez que se realiza una interacción por medio del usuario. El resultado de la multiplicación del valor de la entrada por el valor del peso se introduce al punto de suma en el modelo del perceptron.

3.2.7.3 Valor de ajuste (*bias*)

El valor de ajuste fue seleccionado de acuerdo a los siguientes criterios definidos:

- La salida del punto de suma debe ser menor a 1 cada vez que el sistema no detecta una nueva rutina de usuario.
- La salida del punto de suma debe ser mayor a 1 cada vez que el sistema detecta una nueva rutina de usuario.

De acuerdo a lo anterior y las definiciones previamente definidas para la entrada y el peso, la única manera que la salida del punto de suma sea mayor que uno (>1) al determinar la aparición de una nueva rutina, es generar algún tipo de “memoria” en la red neuronal. Por este motivo se decide que el valor de ajuste debe ser ajustable y tiene que depender de la salida de un estado anterior del punto de suma (ya que así el resultado de las interacciones se ira sumando en cada iteración).

La expresión definida para determinar el valor de ajuste en la red neuronal es la siguiente:

$$b(t) = n(t-1) \tag{3.2}$$

Donde:

- *b(t)* es el valor de ajuste actual.
- *n(t-1)* es la salida de un estado anterior del punto de suma.

3.2.7.4 Función de activación

Esta función determina la salida de la red neuronal (perceptron). Existen diferentes tipos de funciones de activación, sin embargo, la función *lineal* es la seleccionada debido a su simplicidad. Esta función se muestra en la figura ***3.29***.

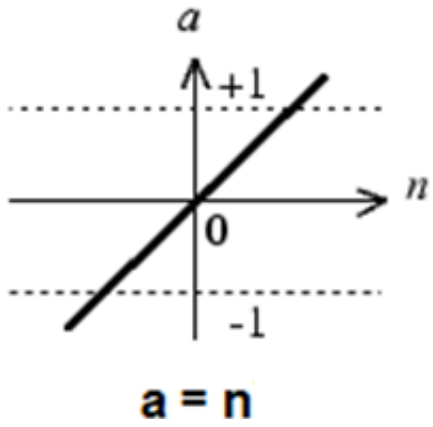

Figura 3.29 Función de activación lineal.

Para la gráfica de la ***figura 3.29***, se entiende que la salida de la función de activación es igual a la entrada de la función (esto es, si a = - 0.5 entonces n = -0.5 y si a = +0.5 entonces n = +0.5).

Con lo anterior descrito, el modelo inicial de perceptron propuesto para la solución del problema del algoritmo de aprendizaje puede ser observado en la figura 3.30.

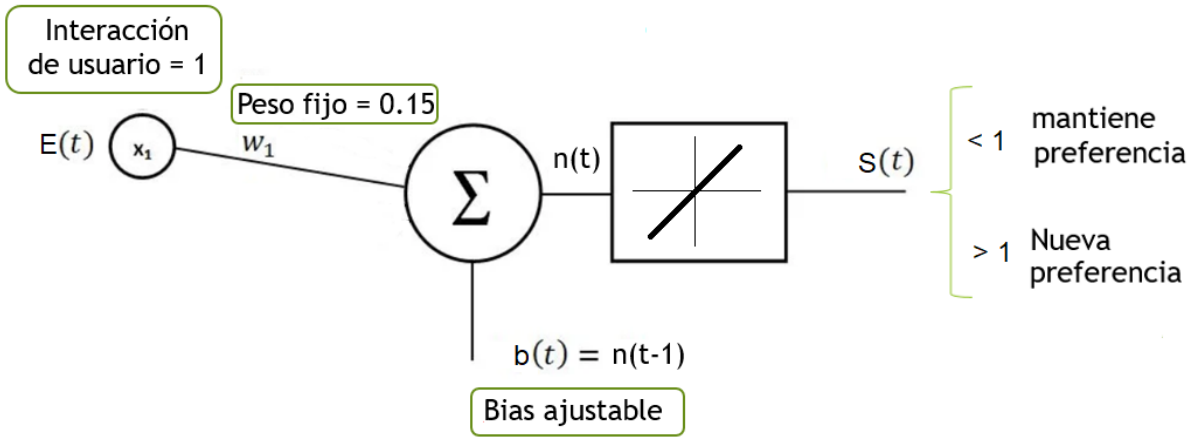

Figura 3.30 Primera aproximación al modelo del perceptron.

Una vez codificada la representación gráfica del perceptron, se prueba su funcionalidad. El desempeño del modelo propuesto es aceptable y cumple con los requerimientos definidos previamente, sin embargo, cuenta con el detalle de que los valores de *n(t)* y *S(t)* son exactamente el mismo en cada evento, lo que lleva a cuestionarse si realmente la función de activación está siendo utilizada de manera adecuada.

Para solucionar el detalle de valores de *n(t)* y *S(t)* iguales, una segunda aproximación es definida. La esencia de funcionamiento está basada en las mismas características, sin embargo, se modifican algunos detalles en las etapas de *entrada*, *valor de ajuste (bias)* y *función de activación* que simplifican el diseño y dejan una tarea específica a cada componente sin llegar a ser redundantes en los datos de cada operación.

Los cambios en los distintos componentes del algoritmo de aprendizaje para la implementación de la segunda aproximación se describen a continuación

3.2.7.5 Entrada modificada

Anteriormente, el valor de la entrada fue definida como un valor fijo cada vez que el usuario tenía una interacción con el sistema y la función del algoritmo de aprendizaje era llamada. En esta ocasión, el cambio consiste en modificar el valor de la entrada, de 1 hasta 7, cada vez que el algoritmo es llamado. El valor de la entrada es actualizado por una función la cual compara la selección del usuario en la interacción actual con los datos más recientes en el sistema. Este procedimiento lleva a dos posibilidades:

1- ***El rango de hora y temperatura de la interacción actual es el mismo que el rango seleccionado en la interacción anterior.*** En este escenario, el valor actual de la entrada es incrementado en 1 (por ejemplo, si el valor actual del registro de conteo es 2 ahora es 3). Cuando el valor del registro

de entrada es 7, se define esta configuración como la nueva rutina de usuario.

2- ***El rango de hora y temperatura de la interacción actual es diferente que el rango seleccionado en la interacción anterior.*** En este escenario, el valor actual de entrada es reiniciado a 1 (por ejemplo, si el valor actual del registro de conteo es 5 ahora es 1). Esto indica que han cambiado las características del ambiente y el usuario desea una nueva configuración, por lo que se empiezan a monitorear las nuevas características de hora y temperatura. De nueva cuenta, esta nueva configuración debe tener 7 interacciones iguales (lo que implica a llevar el contador de entrada a 7) para que sea definida como nueva rutina.

En ambos escenarios, el valor del peso se mantiene fijo a 0.15, siguiendo la definición descrita en la *ecuación 1*.

3.2.7.6 Valor de ajuste (bias) modificado

Debido a que la lógica del valor de entrada se modificó, el valor de ajuste necesita ser re-definido en el nuevo esquema. El nuevo valor de ajuste es seleccionado de acuerdo a los siguientes criterios definidos:

- La salida del punto de suma debe ser menor a 0 cada vez que el sistema no detecta una nueva rutina de usuario.
- La salida del punto de suma debe ser mayor o igual a 0 cada vez que el sistema detecta una nueva rutina de usuario.

De acuerdo al criterio anterior, la siguiente expresión describe la cantidad que debe de tomar el valor de ajuste para que se cumplan los requerimientos:

$$w1\ x\ E(t) + b(t) = 0 \qquad (3.3)$$

Donde:

- b(t) es el valor de ajuste.
- E(t) es el valor de la entrada.
- w1 es el valor del peso.

Considerando que el valor del peso es fijo (0.15) y que el cambio en la función de activación se tiene que dar cuando la salida del punto de suma *n* es mayor o igual a 0 (lo que sucede cuando la entrada E(t) es 7, ya que es el número de repeticiones necesarias para asignar una nueva rutina), la siguiente expresión describe la cantidad que debe de tomar el valor de ajuste para que se cumplan los requerimientos definidos:

$$b(t) = -w1\ x\ E(t)\ =\ -0.15\ x\ 7 = -1.05 \qquad (3.4)$$

De acuerdo a la ***ecuación 3.4***, el valor de ajuste b(t) debe ser de -1.05. Por cuestiones de simplicidad y usabilidad, se decidió utilizar el valor cerrado -1 (esto no impacta a la función de activación debido a que la cantidad inferior más cercana es de 0.9).

3.2.7.7 Función de activación modificada

Con la finalidad de simplificar el resultado arrojado por la red neuronal, una nueva función de activación es seleccionada. El criterio para la selección de la nueva función de activación tomo en cuenta los siguientes requerimientos:

1- Debe determinar cuando una configuración de usuario tiene que ser seleccionada como rutina.

2- Debe acotar el número de resultados de salida posibles que causan los distintos valores de entrada *n(t).*

Debido a los puntos anteriores, se define que una salida con 2 valores es lo más apropiado: un valor para mantener la configuración de rutina sin cambio y otro valor para indicarle al sistema el asignar la nueva configuración definida por el usuario como rutina.

Como ha sido comentado previamente, existen diferentes tipos de funciones de activación, sin embargo, la función *límite* es la seleccionada debido a que provee únicamente dos valores de salida para todo el espectro de valores de entrada, cubriendo así los requerimientos anteriormente mencionados. La ***figura 3.31*** muestra el comportamiento de la función límite para diferentes valores de entrada.

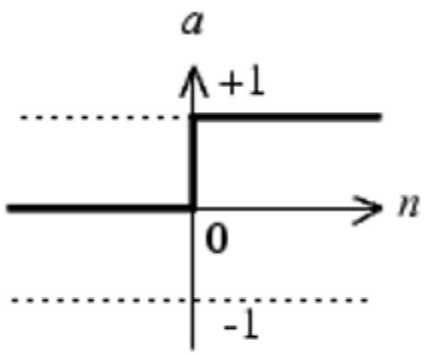

Figura 3.31 Función de activación límite.

En la gráfica de ***la figura 3.31*** se puede observar que mientras la entrada *n* sea menor que 0, el valor devuelto por la función *a* es 0. Cuando el valor de entrada *n* es mayor que o igual que 0, el valor devuelto por la función *a* es 1. Esto se traduce en que la salida del perceptron únicamente realiza la transición a 1 cuando se identifica una nueva rutina a guardar.

Con lo anterior descrito, el modelo modificado del perceptron (segunda aproximación) para la solución del problema del algoritmo de aprendizaje se representa gráficamente como se muestra en la figura 3.32.

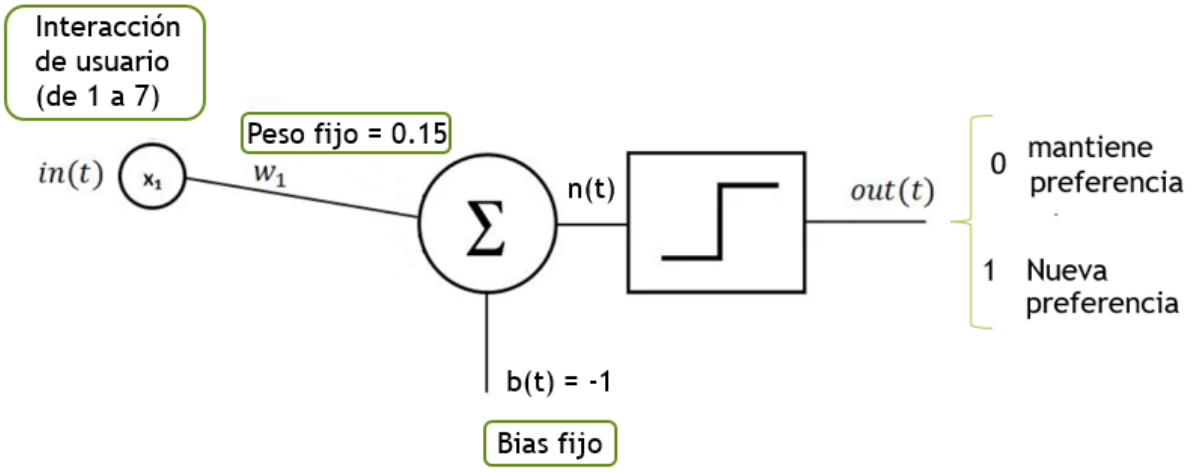

Figura 3.32 Modelo final del perceptron.

Una vez codificada la representación gráfica final del perceptron, es posible probar su comportamiento. Se observa que el desempeño cubre las necesidades que son identificadas durante la definición de los requerimientos. Los resultados que se obtienen durante la validación se presentan en el siguiente capítulo.

El extracto de código dela ***figura 3.33*** muestra la función codificada del algoritmo del aprendizaje en lenguaje C para la verificación de rutina de encendido del sistema.

```
int RutinaEnc(float Entrada)
{
    if((Entrada*0.15) - 1 > 0)
    {
        ActualizarRutinaEnc(horaR, minR);
        Rutina = 1;
    }
    else
    {
        Rutina = 0;
    }

    return Rutina;
}
```

Figura 3.33 Algoritmo de aprendizaje codificado en C.

Como puede observarse en las descripciones anteriores, el algoritmo de aprendizaje aquí presentado requiere de una entrada que puede tomar valores de 1 a 7 para trabajar. El valor de esta entrada se determina por una función, la cual se describe a continuación.

3.2.7.8 Función de ponderación del valor de entrada

El objetivo de esta función es evaluar los resultados de los algoritmos agente *de verificación de temperatura seleccionada* y *agente de verificación de hora de cambio,* así como los parámetros de la última interacción del usuario para determinar el valor de entrada que será enviado al perceptron. Esto se realiza por medio del proceso mostrado en la ***figura 3.34***.

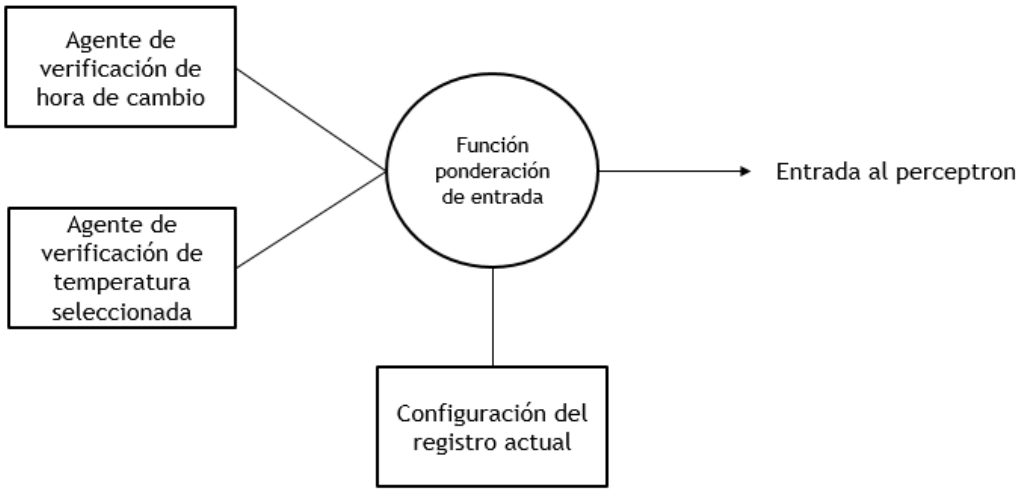

Figura 3.34 Algoritmo de aprendizaje codificado en C.

Cada vez que el usuario realiza un cambio en la configuración del sistema (ya sea encendido o apagado), esta función es llamada. Si la función determina que los parámetros obtenidos durante la interacción actual son iguales que los parámetros de la interacción anterior, se incrementa en 1 el valor del registro de entrada a la red

neuronal y el sistema mantiene los valores de hora y temperatura de la interacción como potencial cambio de rutina. En cambio, si la función determina que los parámetros obtenidos durante la interacción actual son diferentes a los obtenidos en la interacción anterior, la función reinicia a 1 el valor del registro de entrada a la red neuronal y toma los nuevos valores de hora y temperatura como valores potenciales de cambio de rutina.

El extracto de código que muestra la codificación en lenguaje C de la función de verificación de rutina entre los parámetros de la interacción actual y los parámetros de la interacción anterior se muestra en la ***figura 3.35***.

```
void VerifRutina(void)
{
    if((horaU == horaR) && (minU == minR) && (TempU == TempR))
    {
        ContEnt += 1;
        RutinaEnc(ContEnt);
    }
    else
    {
        ContEnt = 1;
        horaR = horaU;
        minR = minU;
        TempR = TempU;
    }
}
```

Figura 3.35 Función de verificación de parámetros de interacción.

3.2.8 Agente de operación en base a termostato

Anteriormente se han descrito los diferentes algoritmos incluidos en la funcionalidad del sistema para la implementación del sistema neuro-difuso de control de temperatura. Cabe mencionar que, aunque el motivo de enfoque del presente estudio es el desarrollo del sistema inteligente, todo el comportamiento se encuentra montado sobre la funcionalidad base del sistema: actuar como termostato.

Para poder llevar a cabo la tarea de funcionamiento base como termostato, los requerimientos que se definen para este propósito son los que enumeran a continuación [28]:

1- Posibilidad de encender/apagar el sistema.
2- Acceso a modificar la temperatura.
3- Mostrar información al usuario.
4- Leer la temperatura ambiente.
5- Activación/desactivación de etapas de calefacción/refrigeración.

La interacción del usuario con los puntos 1, 2 y 3 son contemplados en la interface de usuario presentada en la sección *3.2.4 Agente de interfaz gráfica*, mientras que las descripciones de los puntos 4 y 5 se realiza en las secciones posteriores.

Una vez que el usuario selecciona un punto y un modo de operación (se ha configurado el sistema a cierta temperatura ya sea para calentar o enfriar el ambiente), la lógica de control ejecuta ciertas acciones como leer sensores, consultar el servidor de la base de datos y la activación o desactivación de algunos dispositivos para realizar el control de temperatura ambiente.

La secuencia que realiza el algoritmo de control para que sistema ejecute la operación como termostato es la siguiente:

1- Obtener los parámetros de operación por medio de la interacción del usuario (temperatura, hora y modo de operación).
2- Consulta del historial de temperatura a la base de datos para definir temperatura de operación.
3- Obtener la temperatura ambiente a través del sensor de temperatura.
4- Comparar la temperatura ambiente obtenida contra la temperatura de operación definida.
5- Activar o desactivar las etapas de calefacción o refrigeración (según sea el caso).
6- Repetir los pasos 3, 4 y 5 continuamente hasta que se cumpla la rutina o el usuario apague el sistema manualmente.

El diagrama de la ***figura 3.36*** muestra gráficamente el proceso descrito anteriormente.

Figura 3.36 Proceso de operación como termostato.

La ***figura 3.37*** muestra el código de la función de operación como termostato.

```
void Control(void)
{
       AmbTemp = AdqTemp();
       if(modo == "Calentar" && Estado == 0 && AmbTemp < TempU)
       {
           Won();
           Gon();
           Estado = 1;
       }
       else  if(modo == "Calentar" && Estado == 1 && AmbTemp > (TempU + 2))
       {
           Woff();
           Goff();
           Estado = 0;
       }
       else if(modo == "Enfriar" && Estado == 0 && AmbTemp > TempU)
       {
           Yon();
           Gon();
           Estado = 1;
       }
       else if(modo == "Enfriar" && Estado == 1 && AmbTemp < TempU - 2)
       {
           Yoff();
           Goff();
           Estado = 0;
       }

}
```

Figura 3.37 Función de control de la operación como termostato.

3.2.9 Periférico de adquisición de temperatura ambiente

La esencia del proyecto es un dispositivo de control de temperatura (termostato). Como tal, el dispositivo debe ser capaz de conocer en todo momento la temperatura del ambiente que está controlando. La tarjeta de desarrollo contiene todo lo necesario para la lógica, pero no tiene integrado algún sensor que permita conocer el estado de la variable de interés (temperatura). Sin embargo, cuenta con puertos de comunicación los cuales permiten agregar diferentes periféricos, con diferentes características y funcionalidades, para de esta manera expandir el alcance y posibilidades de la tarjeta de desarrollo.

Las capacidades de expansión que se encuentran en la tablilla de evaluación son las siguientes:

- Puertos GPIO.
- Puerto serie.
- Puerto I^2C.

De las interfaces previamente mencionadas, actualmente dos de ellas se encuentran en uso:

- El puerto serie está siendo utilizado para la conexión con el módulo de expansión WIFI.
- Los diferentes periféricos GPIOs están siendo utilizados por la interface de conexión con el HVAC (módulo de relevadores).

Por los motivos mencionados anteriormente, se decide utilizar un módulo de sensado de temperatura con interface I^2C. Existen diferentes módulos en el mercado que cuentan con estas características (sensores de temperatura I^2C), pero se decide utilizar el módulo de sensado LM75A debido a su facilidad de adquisición y la gran cantidad de material de apoyo que puede encontrarse sobre el en la *web*.

El modulo que incluye sensor LM75A se muestra en la ***figura 3.38***.

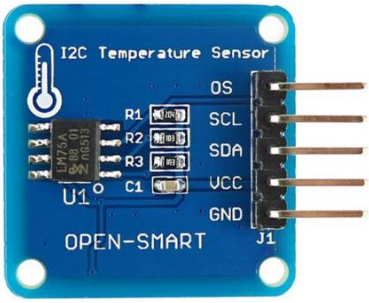

Figura 3.38 Modulo de sensado de temperatura con interface I^2C.

El uso de este módulo habilita al sistema para obtener los datos de temperatura ambiente en cualquier momento. La integración del módulo de temperatura con la tablilla de control puede observarse en la ***figura 3.39***.

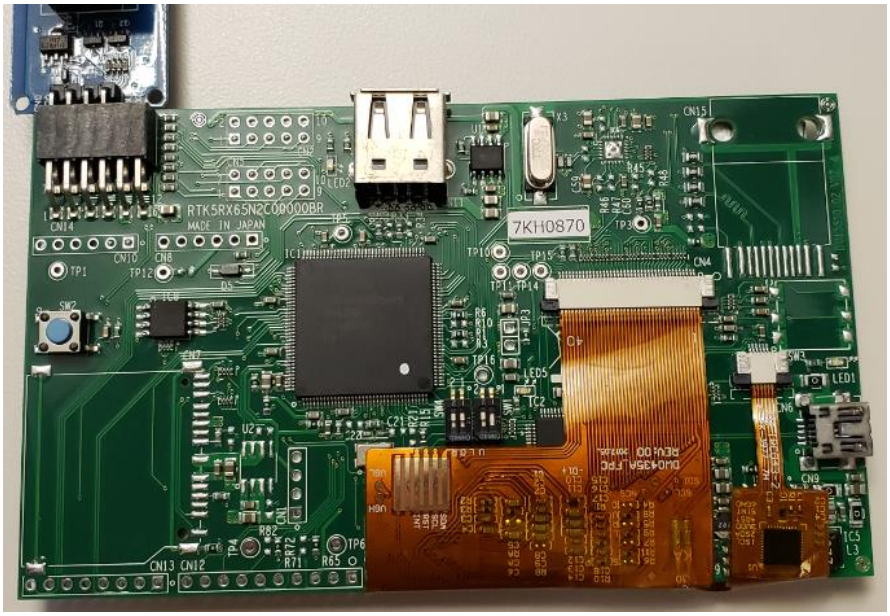

Figura 3.39 Integración del módulo de sensado de temperatura y tarjeta de control.

El extracto de código de la ***figura 3.40*** muestra la codificación en lenguaje C de la función que ejecuta el proceso de lectura del sensor de temperatura por I^2C.

```
float AdqTemp(void)
{
    R_Config_SCI2_Start();
    Temp = R_Config_SCI2_IIC_Master_Send(0x44, 0x02, 2);
    R_Config_SCI2_Stop();

    return Temp;
}
```

Figura 3.40 Procedimiento para leer el sensor de temperatura por I^2C.

3.2.10 Periférico de accionamiento de cargas

Hasta el momento se ha descrito la implementación por *software* de los diferentes algoritmos que componen el sistema, así como algunos componentes de hardware para interactuar con el ambiente y otros dispositivos. En este apartado se describe el funcionamiento y la interacción del módulo utilizado para el control de cargas (HVAC).

Tal y como sucede con el módulo de sensado de temperatura, la tarjeta de desarrollo no cuenta con dispositivos de potencia para realizar el accionamiento de las etapas del HVAC (cargas). Para realizar esta función, es necesario colocar un aditamento a la tarjeta de desarrollo que le permita el control de aparatos externos.

El termostato descrito en este documento está pensado para ser utilizado en sistemas de HVAC *one heat - one cool* convencionales (más información sobre los diferentes dispositivos HVAC se encuentra en el apartado *tipos básicos de sistemas HVAC* en la sección *apéndices*), lo que quiere decir que contara con 3 cargas:

- 1 etapa para calentar.
- 1 etapa para enfriar.
- 1 etapa de abanico.

Para activar cada una de las etapas mencionadas anteriormente, es necesario enviar una señal de 24VAC a la terminal correspondiente del HVAC. Este detalle requiere de una solución en específico, ya que toda la circuitería que actualmente se encuentra en la tablilla de desarrollo es en base a 3.3V.

Existen diferentes componentes en el mercado cuya función permite utilizar diferentes tipos de voltaje en sus terminales de entrada/salida y, de esta manera, funcionar como enlace para diferentes etapas de potencia en un ensamble. Como la mayoría de los termostatos en el mercado contienen relevadores para el accionamiento de cargas, estos dispositivos se seleccionan para incluirse en el proyecto y mantener compatibilidad con los productos actualmente comercializados.

Como se ha venido mencionando, la tarjeta de desarrollo cuenta con algunos periféricos de entrada y salida de datos. Para seleccionar cada una de los 3 relevadores conectados a las cargas, es necesario utilizar 3 GPIOs del puerto de interface de la tarjeta de desarrollo, pero el manejo de los relevadores requiere el uso de un controlador para evitar sobrecargar el puerto de la tarjeta de desarrollo.

Afortunadamente, existen diferentes opciones ya integradas en el mercado (módulo de relevadores con driver) los cuales únicamente necesitan una señal digital (1 o 0 lógico) para la activación de cada canal. Debido a lo anterior, se decide utilizar el módulo de relevadores *Arduino 4 relay module*, el cual ya ofrece la solución completa de 4 relevadores y 4 *drivers* integrados (el cual se muestra en la ***figura 3.41***).

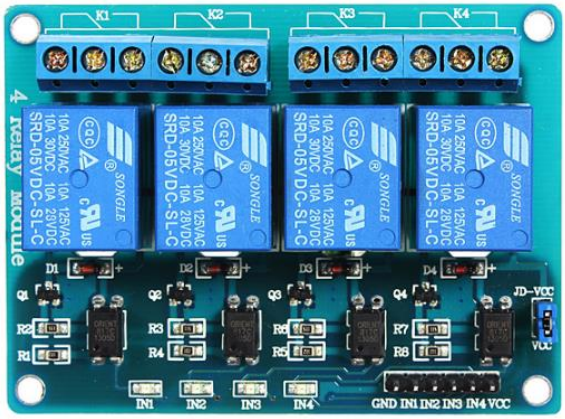

Figura 3.41 Modulo de 4 relevadores y drivers.

La integración del módulo de relevadores y los otros componentes del sistema se muestran en la ***figura 3.42***.

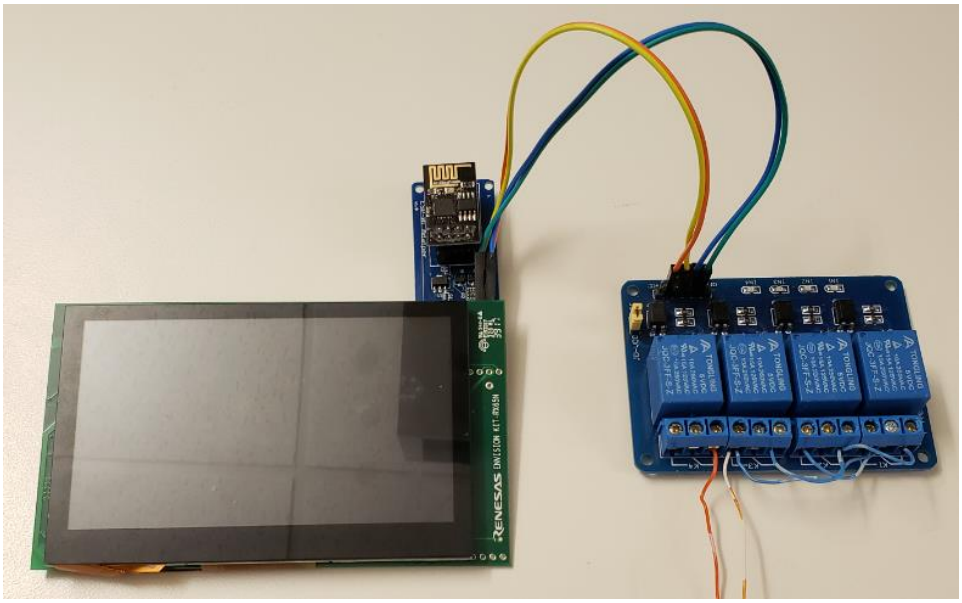

Figura 3.42 Integración del módulo de relevadores y la tarjeta de desarrollo.

3.3 Recolección de información y ordenamiento de datos

Una vez que se definen y se implementan los distintos algoritmos y componentes que integran el sistema, es necesario verificar su funcionamiento para así obtener datos del comportamiento de la aplicación. Esta información es de suma importancia debido a que en base a ésta serán definidas la usabilidad y eficacia de la solución.

El siguiente es un listado de las diferentes validaciones que fueron realizadas y que son descritas en esta sección:

- Pruebas al agente de verificación de temperatura seleccionada.
- Pruebas al agente de verificación de hora de cambio.
- Pruebas al agente de aprendizaje.

- Comparación de desempeño entre un sistema con rutina de aprendizaje en base a las preferencias del usuario y un sistema sin aprendizaje.

Cada una de las pruebas mencionadas anteriormente tiene la finalidad de validar evaluar categóricamente una de las características clave del algoritmo del sistema y al final se encuentra una comparación entre un sistema implementado con las características descritas en este documento y un sistema operado manualmente en todo momento. Con los datos obtenidos por medio de estas validaciones, se determinan las ventajas y desventajas que se obtienen al utilizar esta solución propuesta.

El procedimiento específico relacionado a cada validación, así como los resultados que se obtienen en cada una de las pruebas realizadas, se describen a detalle en los apartados posteriores.

3.3.1 Pruebas al agente de verificación de temperatura seleccionada

Como se describe a detalle en la *sección 3.2.5*, el objetivo de este agente es el comparar la temperatura de operación seleccionada con el usuario contra los límites históricos de temperatura para el día en cuestión. Por lo tanto, las pruebas con este algoritmo consisten en realizar una serie de consultas a la base de datos mediante la selección de diferentes puntos de operación en diferentes días. El resultado esperado es que para cada interacción se delimite la temperatura a uno de los rangos predefinidos y se acepte la temperatura del usuario o que el sistema proporcione una opción basada en los límites estadísticos.

Los diferentes escenarios de prueba, así como el comportamiento que se obtiene por el sistema se muestran en la ***tabla 3.1***.

Tabla 3.1 Listado de pruebas realizadas al algoritmo de verificación de temperatura seleccionada.

PRUEBA	RESULTADO
Fecha: 01/01/2019 **Temperatura seleccionada:** 16.0	**Respuesta del sistema:** Toma 17.5ºC como la selección del usuario **Rango definido:** Frío
Fecha: 31/01/2019 **Temperatura seleccionada:** 24.0	**Respuesta del sistema:** Ofrece temperatura máxima histórica de 19ºC como selección de usuario **Rango definido:** Caliente
Fecha: 01/02/2019 **Temperatura seleccionada:** 2.0	**Respuesta del sistema:** Ofrece temperatura mínima de rango de 15ºC como selección de usuario **Rango definido:** Muy frío
Fecha: 28/02/2019 **Temperatura seleccionada:** 18.0	**Respuesta del sistema:** Toma 17.5ºC como la selección del usuario **Rango definido:** Frio
Fecha: 01/03/2019 **Temperatura seleccionada:** 15.0	**Respuesta del sistema:** Toma 15ºC como la selección del usuario **Rango definido:** Muy frío
Fecha: 31/03/2019 **Temperatura seleccionada:** 25.0	**Respuesta del sistema:** Toma 25.5ºC como la selección del usuario **Rango definido:** Caliente
Fecha: 01/04/2019 **Temperatura seleccionada:** 32.0	**Respuesta del sistema:** Ofrece temperatura máxima de rango de 28ºC como selección de usuario **Rango definido:** Muy caliente
Fecha: 30/4/2019 **Temperatura seleccionada:** 14.0	**Respuesta del sistema:** Ofrece temperatura mínima de rango de 15ºC como selección de usuario **Rango definido:** Muy frío
Fecha: 01/05/2019 **Temperatura seleccionada:** 9.0	**Respuesta del sistema:** Ofrece temperatura mínima de rango de 15ºC como selección de usuario **Rango definido:** Muy frío
Fecha: 31/05/2019 **Temperatura seleccionada:** 30.0	**Respuesta del sistema:** Ofrece temperatura máxima de rango de 28ºC como selección de usuario **Rango definido:** Muy caliente
Fecha: 01/06/2019 **Temperatura seleccionada:** 20.0	**Respuesta del sistema:** Toma 21.5ºC como la selección del usuario **Rango definido:** Ambiente
Fecha: 30/06/2019 **Temperatura seleccionada:** 32.0	**Respuesta del sistema:** Ofrece temperatura máxima histórica de 31ºC como selección de usuario **Rango definido:** Muy caliente

Fecha: 01/07/2019 **Temperatura seleccionada:** 33.0	**Respuesta del sistema:** Ofrece temperatura máxima de rango de 28ºC como selección de usuario **Rango definido:** Muy caliente
Fecha: 31/07/2019 **Temperatura seleccionada:** 15.0	**Respuesta del sistema:** Ofrece temperatura mínima histórica de 16ºC como selección de usuario **Rango definido:**
Fecha: 01/08/2019 **Temperatura seleccionada:** 25.0	**Respuesta del sistema:** Toma 25.5ºC como la selección del usuario **Rango definido:** Caliente
Fecha: 31/08/2019 **Temperatura seleccionada:** 16.0	**Respuesta del sistema:** Ofrece temperatura mínima histórica de 18ºC como selección de usuario **Rango definido:** Frío
Fecha: 01/09/2019 **Temperatura seleccionada:** 34.0	**Respuesta del sistema:** Ofrece temperatura máxima de rango de 28ºC como selección de usuario **Rango definido:**
Fecha: 30/09/2019 **Temperatura seleccionada:** 21.0	**Respuesta del sistema:** Toma 21.5ºC como la selección del usuario **Rango definido:** Ambiente
Fecha: 01/10/2019 **Temperatura seleccionada:** 22.0	**Respuesta del sistema:** Toma 21.5ºC como la selección del usuario **Rango definido:** Ambiente
Fecha: 31/10/2019 **Temperatura seleccionada:** 23.0	**Respuesta del sistema:** Toma 21.5ºC como la selección del usuario **Rango definido:** Ambiente
Fecha: 01/11/2019 **Temperatura seleccionada:** 10.0	**Respuesta del sistema:** Ofrece temperatura mínima de rango de 15ºC como selección de usuario **Rango definido:** Muy frío
Fecha: 30/11/2019 **Temperatura seleccionada:** 20.0	**Respuesta del sistema:** Ofrece temperatura máxima histórica de 19ºC como selección de usuario **Rango definido:** Frío
Fecha: 01/12/2019 **Temperatura seleccionada:** 15.0	**Respuesta del sistema:** Toma temperatura de 15ºC como selección de usuario **Rango definido:** Muy frio
Fecha: 31/12/2019 **Temperatura seleccionada:** 0.0	**Respuesta del sistema:** Toma temperatura de 0ºC como selección de usuario **Rango definido:** Muy frío

De acuerdo a la información anterior, el resultado que se obtiene en cada uno de los escenarios de prueba es de acuerdo a los requerimientos de funcionalidad planteados durante la etapa de definición en la *sección 3.2.5*.

3.3.2 Pruebas al agente de verificación de hora de cambio

Como se comentó en la *sección 3.2.6*, el objetivo de este agente es el obtener y delimitar la hora de las interacciones en las que el usuario modifica el estado del sistema. De ahí que las pruebas en este algoritmo consisten en realizar una serie de cambios de configuración para activar al agente. El resultado esperado es que para cada interacción se delimite la hora de operación dentro de los intervalos establecidos. Las pruebas realizadas se muestran en la ***tabla 3.2***.

Tabla 3.2 Listado de pruebas realizadas al algoritmo de verificación de hora de cambio.

Evento	Hora de cambio	Hora de registro
Hora 1	11:45AM	11:30AM
Hora 2	11:46AM	12:00PM
Hora 3	11:50AM	12:00PM
Hora 4	12:00PM	12:00PM
Hora 5	12:10PM	12:00PM
Hora 6	12:14PM	12:00PM
Hora 7	12:15PM	12:00PM
Hora 8	12:25PM	12:30PM
Hora 9	12:35PM	12:30PM
Hora 10	12:45PM	12:30PM

El resultado que se obtiene en cada uno de los escenarios cumple los requerimientos de funcionalidad que se plantean durante la etapa de definición en la *sección 3.2.6*. Los resultados sobre la eficacia del método en conjunto se muestran posteriormente.

3.3.3 Pruebas al agente de aprendizaje

Como fue descrito a detalle en la sección 3.2.7, el objetivo de este agente es el determinar si la interacción que se está llevando a cabo por parte del usuario tiende a ser rutina o es un cambio contante. Por lo tanto, las pruebas en este algoritmo consisten en realizar una serie de cambios de configuración para que el agente se active. El resultado esperado es que para cada interacción se actualice el valor de los registros de encendido y apagado según el peso determinado por los parámetros utilizados en cada interacción.

Los resultados de las pruebas realizadas al agente de aprendizaje se muestran en las ***tablas 3.3*** y ***3.4***.

Tabla 3.3 Listado de pruebas al algoritmo de aprendizaje con cambios constantes.

DIA#	Hora de encendido	Valor del registro de encendido	Hora de apagado	Valor del registro de apagado
Inicio	NA	0	NA	0
Día 1	8:01AM	0.15	1:09PM	0.15
Día 2	7:55AM	0.3	1:14PM	0.3
Día 3	8:30AM	0.15	1:31PM	0.15
Día 4	8:16AM	0.3	1:20PM	0.3
Día 5	8:44AM	0.45	2:15PM	0.15

Día 6	9:05AM	0.15	2:16PM	0.3
Día 7	9:10AM	0.3	2:14PM	0.45
Día 8	8:00AM	0.15	12:46PM	0.15
Día 9	7:46AM	0.3	12:55PM	0.3
Día 10	8:14AM	0.45	12:10PM	0.45

Tabla 3.4 Cambios mantenidos que permiten el aprendizaje de una rutina.

DIA#	**Hora de encendido**	**Valor del registro de encendido**	**Hora de apagado**	**Valor del registro de apagado**
Inicio	NA	0	NA	0
Día 1	8:46AM	0.15	5:16PM	0.15
Día 2	8:50AM	0.3	5:20PM	0.3
Día 3	8:55AM	0.45	5:25PM	0.45
Día 4	9:00AM	0.6	5:30PM	0.6
Día 5	9:05AM	0.75	5:135PM	0.75
Día 6	9:10AM	0.9	5:40PM	0.9
Día 7	9:14AM	1.05	5:44PM	1.05
Día 8	9:47AM	0.15	5:46PM	0.15
Día 9	10:52AM	0.3	5:50PM	0.3
Día 10	10:54AM	0.45	5:55PM	0.45

Día 11	11:05AM	0.6	6:00PM	0.6
Día 12	11:09AM	0.75	6:05PM	0.75
Día 13	11:11AM	0.9	6:10PM	0.9
Día 14	11:13AM	1.05	6:14PM	1.05

El resultado que se obtiene en cada uno de los escenarios cumple los requerimientos de funcionalidad planteados durante la etapa de definición en la *sección 3.2.7*. Los resultados sobre la eficacia del método en conjunto se muestran posteriormente.

3.3.4 Comparación de desempeño entre un sistema con rutina de aprendizaje en base a las preferencias del usuario y un sistema sin aprendizaje

El objetivo en conjunto de los diferentes algoritmos y agentes utilizados en el sistema es el de proveer la capacidad de aprendizaje para así aumentar la sensación de *confort* por parte del usuario y además minimizar el consumo de recursos energéticos utilizados durante los períodos de operación.

Con la finalidad de determinar el beneficio del sistema, se han monitoreado dos aplicaciones:

A- Un sistema inteligente.

B- Un sistema manual.

En el caso de estudio A, el sistema se opera durante 6 horas consecutivos utilizando el nivel de temperatura sugerido por la aplicación durante la interacción y siguiendo los lineamientos de operación de cada uno de los algoritmos que se presentan en este escrito.

En el caso de estudio *B*, el sistema se opera también durante 6 horas, pero los puntos de operación corresponden a selección del usuario (sin la funcionalidad de los algoritmos descritos en este documento).

Para efectos de análisis y simplificación en la verificación de las soluciones, el resultado sobre el tiempo de operación en cada uno de los escenarios ha sido capturado y graficado, de tal manera que pueda realizarse una comparativa tanto visual como cuantitativa acerca de su desempeño.

La ***figura 3.43*** muestra la evolución en temperatura y desempeño de los sistemas descritos previamente:

- La traza azul representa una selección de usuario de 23ºC durante un día con temperatura histórica máxima de 20ºC, la cual el usuario acepto utilizar como temperatura de operación debido a la sugerencia del sistema inteligente.
- La traza naranja representa una selección de usuario de 23ºC durante un día con temperatura histórica máxima de 20ºC en un sistema sin las capacidades descritas en este documento.

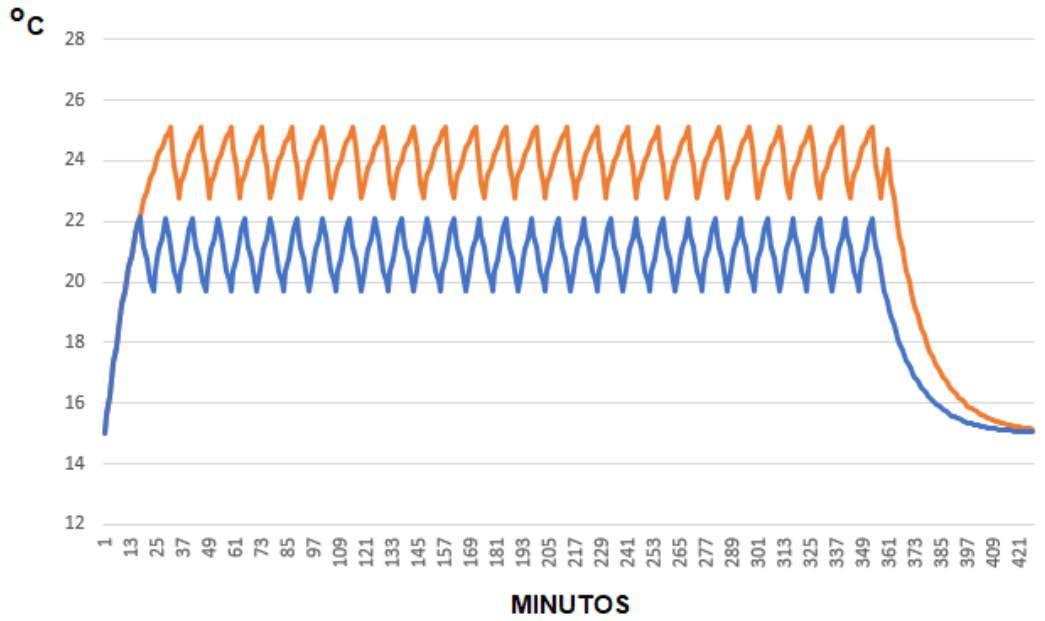

Figura 3.43 Control de temperatura activo durante 6 horas a 20ºC (azul) y 23ºC (naranja).

En la ***figura 3.43*** se observa la diferencia en la operación entre ambas condiciones de temperatura (aquella opción que fue seleccionada libremente por el usuario y aquella opción que fue seleccionada por sugerencia del sistema). Es visiblemente notable que ambos escenarios operan bajo las mismas condiciones ambientales, tal que la temperatura base fue de 15ºC. Al pasar el tiempo, la temperatura adquiere un nivel cada vez mayor hasta llegar al punto de operación configurado para cada caso (20ºC y 23ºC).

Tomando como referencia la cantidad de minutos que cada sistema duro encendido y apagado de acuerdo a la ***figura 3.43***, se pueden obtener los datos de la ***tabla 3.5***.

Tabla 3.5 Tiempos de encendido y apagado en casos A y B.

	Sistema inteligente (Caso A)	**Sistema manual (Caso B)**
Tiempo total encendido (Minutos)	166	262
Tiempo total apagado (Minutos)	260	164

En la tabla anterior se indican la cantidad de minutos que opera el sistema HVAC (separados por tiempo encendido y apagado) mientras se controla la temperatura ambiente. Por simple comparación, se observa un aumento en la cantidad de minutos que el HVAC dura encendido cuando es configurado a una temperatura de operación mayor en el sistema.

IV. CAPÍTULO IV. RESULTADOS Y CONCLUSIONES

4.1 Los objetivos y metas alcanzados

A lo largo de esta tesis se han venido mencionando los motivos que llevaron a desarrollar el proyecto de la manera en que fue implementado, así como una serie de experimentos para validar la funcionalidad de cada componente y su impacto en el sistema.

La búsqueda de un resultado satisfactorio estuvo enfocada en todo momento en los siguientes objetivos (planteados inicialmente en la sección 1.2.1):

a) *Aprenda las configuraciones preferidas del usuario y el entorno* – Esto se logró a través de la implementación de un modelo de red neuronal (perceptrón) que aprende de las interacciones del usuario con el sistema, conociendo así sus preferencias.
b) *Defina el criterio óptimo de operación del sistema basado y lo ajuste dinámicamente* – Esto se logró por medio de la implementación de una base de datos con el histórico de temperatura para cierta zona (en este caso, la ciudad de Chihuahua) el cual informa al usuario sobre la diferencia de la temperatura seleccionada y los límites históricos para cada día en específico, así como la implementación de un sistema difuso para la categorización de la temperatura por rangos de acuerdo a la sensación térmica de los usuarios.
c) *Mantenga el ambiente en un nivel confortable de operación y con consumo económico de energía* – Esto se logró al implementar en conjunto los diferentes algoritmos que conforman el sistema, habilitando así la capacidad del sistema de encenderse automáticamente a cierta temperatura y por determinado tiempo, minimizando así la interacción requerida por parte del usuario.

Bajo el mismo esquema, la meta propuesta fue la implementación de un controlador inteligente de temperatura el cual maximiza del uso eficiente de los recursos energéticos utilizados para la operación [33]. Esta meta se ha cumplido, puesto que el modelo inteligente propuesto fue llevado a la práctica y puesto a prueba, obteniendo resultados satisfactorios los cuales serán mostrados en el siguiente apartado.

4.2 Ventajas del sistema

Durante el planteamiento inicial del proyecto, se definió que la principal ventaja ofrecida radica en dos características:

1- Confort de usuario en su utilización.
2- Ahorro de recursos energéticos.

Como fue descrito en el apartado anterior, diferentes agentes fueron incluidos en el sistema con la finalidad de alcanzar este cometido. A continuación, se muestran los resultados del desempeño obtenido del sistema al someterlo a pruebas de uso.

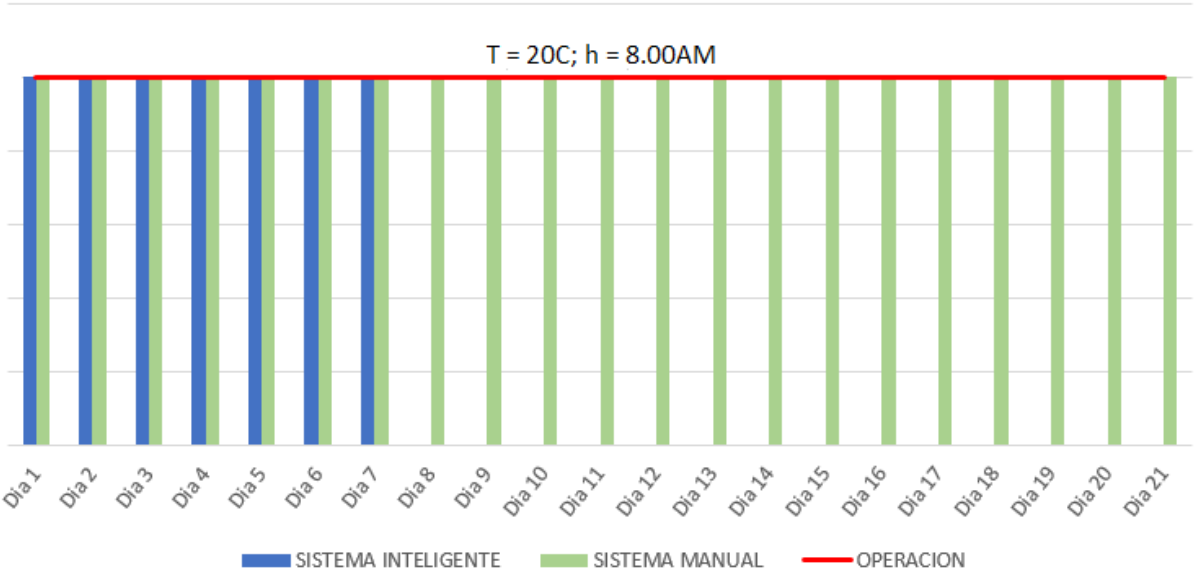

Figura 4.1 Interacción de usuario y operación por 21 días.

En la ***figura 4.1***, se muestra un sistema inteligente (barra azul) el cual fue configurado a la misma hora por 7 días. Puede observarse que el sistema operó por 21 días basado en las características seleccionadas por el usuario durante la etapa de aprendizaje. También puede observarse como un sistema manual (barra verde) operó también por 21 días, pero la interacción del usuario se llevó acabo en cada uno de estos días al no contar el sistema con algún tipo de inteligencia. En ambos casos, los sistemas operaron a una temperatura de 20ºC a partir de las 8AM, sin embargo, el usuario tuvo que interaccionar los 21 días de prueba con el sistema manual mientras solo interaccionó 7 días con el sistema inteligente para obtener el mismo resultado.

Para verificar el desempeño del sistema en cuanto a maximización de utilización de recursos, se midió el tiempo de duración en estado de encendido del HVAC para diferentes temperaturas seleccionadas bajo el mismo esquema de operación. El resultado se puede observar en la ***tabla 4.1***.

Tabla 4.1 Tiempos de encendido y apagado para diferentes temperaturas durante 6 horas.

	18C	20C	23C	25C
Tiempo apagado	283	260	164	109
Tiempo encendido	143	166	262	317

Graficando la información previamente mostrada, se puede observar el comportamiento mostrado en la **figura 4.2**.

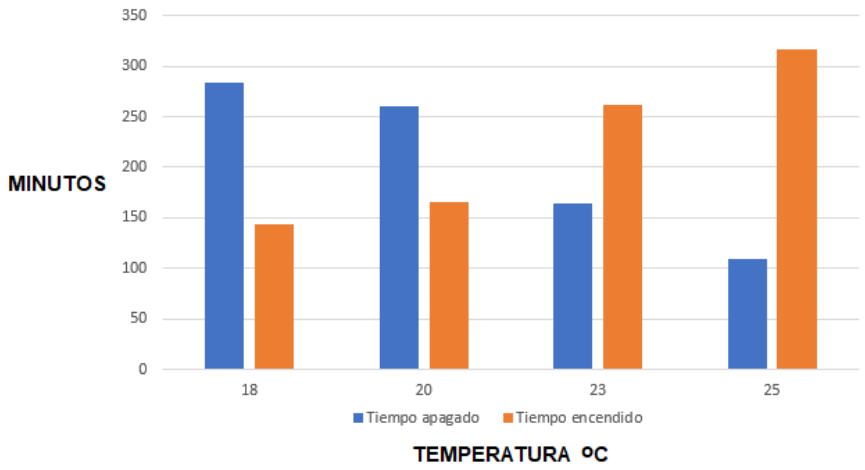

Figura 4.2 Duración del sistema en estados apagado (azul) y encendido (naranja).

De la gráfica anterior puede inferirse que, a mayor temperatura de operación seleccionada mayor es el tiempo que necesita permanecer encendido el sistema para lograr mantener la condición ambiente deseada.

Como ejemplo, tomando de referencia una calefacción de 150000BTU, esta consume aproximadamente 1 litro de gas por hora al configurarse en un esquema de 3 ciclos por hora y 1.5KWh de electricidad. Tomando en cuenta el costo del gas de $10.95 pesos por litro y el costo base del KWH de $0.80 pesos (al 20 de diciembre del 2019), el costo de operación para cada escenario por las 6 horas de trabajo antes mencionadas se muestra en la ***tabla 4.2***.

Tabla 4.2 Costo de operación por 6 horas de control de temperatura.

Temperatura	Costo gas	Costo electricidad	Costo total
18C	26.0975	1.906666667	28.00416667
20C	30.295	2.213333333	32.50833333
23C	47.815	3.493333333	51.30833333
25C	57.8525	4.226666667	62.07916667

Puede observarse como el costo de operación se eleva de forma exponencial al incrementar la temperatura de operación. Por tal motivo, el enfoque seleccionado para la implementación de la característica de minimización de uso de recursos del sistema es una solución factible.

4.3 Mejoras al sistema

Referente al algoritmo de control, la implementación del proyecto estuvo centrada en el control de temperatura ambiente. Sin embargo, el método neuro-difuso aquí presentado no está limitado únicamente a esta aplicación y puede ser utilizado para agregar características al sistema, tales como:

- Control de humedad.
- Control de calidad de aire.

- Reconocimiento de patrones de operación por usuario.
- Detección de niveles de CO2.

Aunque las bases para la implementación del método han sido definidas en este documento, la adición de cada una de estas funcionalidades requiere tratarse como una particularidad y debería realizarse su propio caso de estudio con análisis, experimentos, pruebas y validaciones para asegurar la eficacia del método al utilizarse para el control de estos u otros parámetros.

Referente a la implementación física, ésta se realizó utilizando tarjetas de desarrollo y módulos de expansión. Una mejora latente en este aspecto es el desarrollo de una tablilla propietaria, diseñada a medida para cumplir todos los requerimientos de funcionalidad requeridos, tales como:

- Lógica de control (microcontrolador y memoria).
- Circuitos de comunicación inalámbrica.
- Circuitos de control de cargas.
- Sensores.

Tan solo el desarrollo de un producto de tales características amerita evaluarse en un caso de estudio aparte, para definir y cumplir a detalle cada una de las necesidades.

4.4 Conclusiones

El objetivo principal de esta tesis ha sido la definición e implementación de un modelo neuro-difuso para el control de temperatura ambiente que permita simplificar la utilización de un termostato y maximizar su desempeño, lo cual ha sido realizado a través de los siguientes logros:

- El sistema es capaz de aprender los gustos del usuario, aumentado así la sensación del confort al utilizar el dispositivo.

- El sistema es capaz de ahorrar combustible y energía eléctrica al sugerir operar a niveles estadísticos más bajos.

Actualmente existen algunas investigaciones referentes a la reducción y maximización energética, las cuales abordan la solución de maneras distintas [13], [14], [15] [16] [17] [18] [19] [20] [21] [32] [33]. La solución aquí propuesta busca ser innovadora, de tal manera que pueda buscarse una patente por ella en un futuro.

Se ha mostrado como la utilización adecuada del algoritmo aquí presentado es capaz de reducir los costos de operación en una instalación al momento de controlar la temperatura. Esto es una gran ventaja en la época actual (en la cual el costo de vida se encarece cada vez más y los salarios no aumentan en la misma proporción), ya que esta solución implementada de manera comercial podría apoyar en la economía de los hogares sin sacrificar la comodidad.

Tanto el algoritmo como la implementación presentan áreas de oportunidad y definitivamente pueden ser mejorados. Esto abre la posibilidad a trabajos de estudio posteriores, los cuales pudieran enriquecer el sistema actual o buscar soluciones alternas, soluciones con más funcionalidad y más completas y, porque no decirlo, soluciones que incluso pudieran ayudar a aumentar la eficiencia de este sistema.

BIBLIOGRAFIA

[1] L. A. Zadeh, “Fuzzy sets”, Informat. Control, vol 8, pp. 338-353, 1965.

[2] D. Nauck, F. Klawonn, R. Kruse: “Fuzzy Sets, Fuzzy Controllers, and Neural Networks”. Wissenschaftliche Zeitschrift der Humboldt-Universitat zu Berlin, R. Medizin 41 (4) (1992), 99-120.

[3] C.C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller”. IEEE Trans. Syst. Man Cybern. 20 (1990), Part I: 404-418, Part II: 419-435

[4] R. Fuller, “Neural Fuzzy systems”, 1995

[5] T. Munkata and Y. Jani, Fuzzy systems: An overview, Communications of ACM, 37(1994) 69-76

[6] W.S.McCulloch and W.A.Pitts, “A logical calculus of the ideas imminent in nervous activity”, Bull. Math. Biophys. 5(1943) 115-133.

[7] M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, Mass., 1969).

[8] D. Nauck and F. Klawonn, “Combining neural networks and fuzzy Controllers”, 1993

[9] D.E.Rumelhart, B.Widrow and M.A.Lehr, “Neural Networks: Applications in Industry, Business and Science”, Communications of ACM, 1994

[10] M.M. Gupta and D.H. Rao, “On the principles of fuzzy neural networks, Fuzzy Sets and Systems”, 61(1994) 1-18.

[11] J.J. Buckley and Y. Hayashi, “Fuzzy neural nets and applications, Fuzzy Systems and AI”, 1(1992) 11-41.

[12] H. Berenji, “A reinforced learning based architecture for fuzzy logic control”, 1992

[13] F. Zhang, “Building temperature control with intelligent methods”, 2014

[14] A. N. Isizoh, “Temperature control system using fuzzy logic technique”, 2012

[15] Barret, Enda, “Autonomous HVAC control, a reinforcement learning Approach”, 2015

[16] Adam Nagy, “Deep reinforcement learning for optimal control of space Heating”, 2018

[17] Zhiang Zhang, “A deep reinforcement learning approach to using whole building energy model for HVAC optimal control”, 2018

[18] Guanyu Gao, “Energy efficient thermal comfort control in smart buildings via deep reinforcement learning”, 2019.

[19] Caleb Petrie, “Energy efficiency control methods of HVAC systems for smart campus”, 2018

[20] Jennifer King, “Smart buildings: Using smart technology to save energy in existing buildings”, 2017
[21] K. Venkatesan, “Optimal scheduling of air conditioners for energy Efficiency”, 2018
[22] Tim de Bruin, “The importance of experience replay database composition in deep reinforcement learning”, 2015
[23] Dasa Majcen, “Statistical model of the heating prediction gap in Dutch dwellings: Relative importance of building household and behavioral characteristics”, 2015
[24] Huyen Do, “Data driven evaluation of residential HVAC system efficiency using energy and environmental data”, 2019
[25] Fakhruddin H.N, "Fuzzy logic in HVAC for human comfort",2016
[26] Chin-Chi Cheng, “Artificial intelligence-assisted heating ventilation and air conditioning control and the unmet demand for sensors”, 2019
[27] Yuan Wang, “Along-short term memory recurrent neural network based reinforcement learning controller for office heating ventilation and air conditioning systems”, 2017
[28] Yuanglong Li, “Transforming cooling optimization for green data center via deep reinforcement learning”, 2017
[29] Tianshu Wei, “Deep reinforcement learning for building HVAC control”, 2017
[30] Richard Sutton, “Reinforcement learning: An introduction, second edition”, 2017
[31] Abdul Afram, “Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case of a residential HVAC system”, 2017
[32] Ajay N Bhagwat, “Energy efficiency technologies for heating, ventilation and air conditioning (HVAC)”, 2015
[33] Han Mengjie, “A review of reinforcement learning methodologies on control systems for building energy”, 2018
[34] Hussain, Kazmi, “Gigawatt-hour scale savings on a budget of zero: Deep reinforcement learning based on optimal control of hot water systems”, 2018
[35] Microsoft access, “https://www.quackit.com/microsoft_access/tutorial/”
[36] Martin T. Hagan, “Neural network design”, 2nd edition
[37] Timothy J. Ross, “Fuzzy logic with engineering applications”, 2nd edition
[38] ASHRAE, “Fundamentals of HVAC control systems”, 2011
[39] Encuestas realizadas a empleados de la empresa Resideo de abril a julio 2019
[40] Guía para la obtención del grado de maestro en ingeniería Mecatrónica, Instituto tecnológico de Chihuahua, 2016

APÉNDICES

A Encuestas

Tabla A.1 Encuesta de horas de encendido/apagado de un sistema de calefacción en una vivienda.

PREGUNTA	RESPUESTA
¿A qué hora enciendes la calefacción en tu casa en la mañana entre semana?	No se enciende: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora apagas la calefacción en tu casa en la mañana entre semana?	No se apaga: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora enciendes la calefacción en tu casa en la tarde entre semana?	No se enciende: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora apagas la calefacción en tu casa en la tarde entre semana?	No se apaga: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora enciendes la calefacción en tu casa en la mañana los fines de semana?	No se enciende: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora apagas la calefacción en tu casa en la mañana los fines de semana?	No se apaga: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora enciendes la calefacción en tu casa en la tarde los fines de semana?	No se enciende: Hora en punto: Medias horas: Fracción de hora:
¿A qué hora apagas la calefacción en tu casa en la tarde los fines de semana?	No se apaga: Hora en punto: Medias horas: Fracción de hora:

Tabla A.2 Encuesta sobre sensación térmica dentro de una vivienda.

PREGUNTA	RESPUESTA
¿A qué temperatura dentro de una edificación consideras que el ambiente está muy frio?	Temperatura mínima: Temperatura máxima:
¿A qué temperatura dentro de una edificación consideras que el ambiente está frio?	Temperatura mínima: Temperatura máxima:
¿A qué temperatura dentro de una edificación consideras que el ambiente esta agradable?	Temperatura mínima: Temperatura máxima:
¿A qué temperatura dentro de una edificación consideras que el ambiente está caliente?	Temperatura mínima: Temperatura máxima:
¿A qué temperatura dentro de una edificación consideras que el ambiente está muy caliente?	Temperatura mínima: Temperatura máxima

B Tipos básicos de sistemas HVAC

CONVENCIONAL

Es el tipo más común de HVAC en el mercado. Este tipo de dispositivos tiene la capacidad de funcionar ya sea para enfriar o para calentar el ambiente. La función principal de este tipo de dispositivos es de hacer circular una masa de aire (ya sea fría o caliente) por un sistema de ductos, los cuales se distribuyen a través de la edificación, haciendo llegar a cada espacio de la instalación la masa de aire para ocasionar el intercambio térmico.

Este tipo de dispositivos realizan la función de enfriar por medio de un equipo de refrigeración, mientras que la función de calentar la realizan por quemadores ya sea en base a gas, petróleo u eléctricos. La ***figura A.1*** muestra un ejemplo de sistema convencional.

Figura A.1 Unidad HVAC convencional.

Por estándar, las señales de control en estos dispositivos son del tipo ON / OFF, cuya lógica se basa en la presencia o ausencia de un voltaje de 24VAC en las diferentes terminales para llamar a las diferentes funciones. Estas funciones están

determinadas por una terminal en específico, las cuales están designadas de la siguiente manera:

- R – Línea de referencia positiva de la fuente de 24VAC (típicamente un transformador).
- C – Línea de referencia negativa de la fuente de 24VAC (típicamente un transformador). Se diferencia de la R debido a que C comúnmente se conecta a tierra física.
- W – Señal para activar función de calefacción.
- Y - Señal para activar función de refrigeración.
- G - Señal para activar abanico manualmente.

Existen diferentes tamaños y capacidades de unidades. La correcta selección de la unidad dependerá del área que se desea controlar. Para mejorar el desempeño de este tipo de equipos, existen algunas variantes en su clasificación, basadas en la cantidad de etapas que ofrecen para realizar su función. Algunas de las clasificaciones se describen a continuación:

- *One heat/one cool* – Contienen solo una etapa de calefacción y otra de refrigeración (W y Y). Es el más común de los sistemas.
- *Two heat/two cool* – Contiene dos etapas de calefacción y dos de refrigeración (W, W2, Y y Y2).
- *Three heat/two cool* - Contiene tres etapas de calefacción y dos de refrigeración.

La ventaja de un equipo multi etapa con respecto a otro de etapa sencilla es que el equipo multi-etapa no opera a toda su capacidad en todo momento, sino que dependiendo del diferencial de temperatura ambiente contra la selección del usuario el sistema decide cuantas etapas encender o apagar, lo que ocasiona un mejor desempeño en el comportamiento del sistema y un gradiente temperatura menos agresivo en la instalación.

BOMBA DE CALOR (*HEAT PUMP*)

A diferencia del modo convencional, un equipo del tipo bomba de calor (también llamados ciclo reversible) utilizan como base de un sistema de refrigeración, pero contiene algunos accesorios adicionales que permiten una inversión en el sentido de refrigeración (de ahí el nombre de ciclo reversible), lo que permita enfriar y calentar en base al mismo principio de intercambiador de calor. La ***figura A.2*** muestra un ejemplo de sistema de bomba de calor.

.

Figura A.2 Unidad HVAC tipo bomba de calor.

De la misma manera que un equipo del tipo convencional, las señales de control en este tipo de equipos son OFF / ON a 24VAC, sin embargo, se agregan unas señales y algunas de las existentes cambian un poco su forma de operar, mientras que otras quedan de la misma manera:

- R – Línea de referencia positiva de la fuente de 24VAC (típicamente un transformador).
- C – Línea de referencia negativa de la fuente de 24VAC (típicamente un transformador). Se diferencia de la R debido a que C comúnmente se conecta a tierra física.
- O/B – Señal para activar función el modo de refrigeración o de calefacción. La ausencia de 24VAC en esta terminal indica al dispositivo que debe de

operar en modo de refrigeración, mientras que la presencia de 24VAC indica al dispositivo que debe de operar en modo de calefacción.

- Y - Señal para activar la etapa base del compresor.
- Y2 – Señal para activar la segunda etapa del compresor.
- G - Señal para activar abanico manualmente.

Algunas unidades HVAC utilizan la terminal W2 para controlar una tercera etapa del compresor, sin embargo, este tipo de dispositivos son poco comunes.

Como comentario adicional, una variante bastante común de este tipo de equipos (pero en tamaño reducido y sin señales de control externas) son los equipos *minisplit*.