



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
MECÁNICA – MECATRÓNICA

DISEÑO DE UN SISTEMA DE INSTRUMENTACIÓN Y CONTROL PARA
CONSERVAR LAS CONDICIONES DE CONFORT DENTRO DE UNA CASA

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
LUIS ROBERTO MAYA WILSON

TUTOR PRINCIPAL
DRA. MARÍA DEL PILAR CORONA LIRA
FACULTAD DE INGENIERÍA

CIUDAD UNIVERSITARIA, CD. MX., OCTUBRE DE 2023



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Alberto Caballero Ruiz
Secretario: Dr. Edmundo Gabriel Rocha Cózatl
1 er. Vocal: Dra. María Del Pilar Corona Lira
2 do. Vocal: Dr. Víctor Manuel Lomas Barrie
3 er. Vocal: M. en I. Anahí Velázquez Silva

Lugar o lugares donde se realizó la tesis: Ciudad Universitaria, UNAM

TUTOR DE TESIS:

Dra. María Del Pilar Corona Lira

FIRMA

Agradecimientos

Tengo a tantos a quien agradecer, que comenzaré con el Consejo Nacional de Ciencia y Tecnología (Conacyt), por el apoyo económico para mi trayectoria académica, que de esta manera pude enfocarme por completo a mi trabajo de investigación y a la dedicación en mis asignaturas de Maestría, así como de todos mis profesores que me guiaron y acompañaron durante mi estancia en la UNAM.

A la Escuela Militar de Ingenieros, que me abrieron las puertas para poder realizar mi Retribución Social de Conacyt y echarle *fibra* con ustedes, en verdad gracias camaradas por *pasar la pauta* y permitirme además aprender de su estilo de vida.

A Industrializadora de Proteínas y Grasas, gracias a ustedes me interesé tanto en los instrumentos y controles automáticos, que me motivé a hacer la Maestría, gracias por la oportunidad de haber trabajado con ustedes, el aprendizaje adquirido me sirvió muchísimo.

Finalmente, a mi Madre, María Wilson, gracias por tu apoyo incondicional, siempre has sido un pilar importante en mi vida y mi guía, a mi abuela Lola que es mi segunda Madre, a mi hermana Daniela, que juntos nos hemos apoyado y así seguirá siendo, también a mi perro Kepler, porque ese animalito fiel me acompañó en mis noches enteras de estudio y madrugadas difíciles. Soy tan afortunado, que debo agradecerte a ti Dios, porque nunca me has abandonado y siempre has puesto en mi camino la luz para tomar mis decisiones y la fortaleza para nunca rendirme.

Abstract

At this current work, it is possible to visualize the design and building of an instrumentation and control system to improve the comfort conditions in a family home, where the testing sites were promptly three rooms belonging to the house bedrooms. The system handles two actuators, one of them is a portable evaporative cooling air conditioner, and the other one an air extractor; all these regulate the room's indoor temperature and the air quality by the presence of polluting agents such as dust particles and cigarette smoke. The implementation of a set of sensors serves as a pair of feedback control system controllers and be used into an IoT network (Internet Of Things); in order to interact with the system, a graphic interface has been designed and can be handled by any web navigator, where the user can designate the desired parameters of temperature and air quality into the room. On the other hand, the IoT network architecture was built based on functional layers, in which the semantic analysis was used through a thermal comfort prediction algorithm based on the ISO 7730 standard technique identified the outcomes using the instrumentation system.

Finally, in the results of this work, it is possible to visualize the controller's functionality, the estimation of thermal comfort, and the comparison before and after implementing the system in the testing rooms, pretending to demonstrate the correct operation of the project.

Resumen

En el trabajo presente, se muestra el diseño y construcción de un sistema de instrumentación y control para mejorar las condiciones de confort de un hogar familiar, donde los sitios de pruebas fueron específicamente tres habitaciones correspondientes a tres dormitorios. El sistema gobierna dos actuadores, un aire acondicionado portátil por enfriamiento evaporativo y un extractor de aire, y tiene el objetivo de regular la temperatura interna de la habitación y la calidad del aire por presencia de agentes contaminantes como partículas de polvo o humo de cigarro. La implementación de un conjunto de sensores sirve para retroalimentarse con el diseño de dos controladores por lazo cerrado, con el objetivo de ser utilizados en el diseño de una red de *IoT* (*internet de las cosas*), y poder interactuar con el sistema por medio de una interfaz gráfica para el usuario y que puede ser vista en cualquier navegador *web*, donde el usuario puede designar los parámetros deseados de temperatura y calidad de aire dentro de la habitación. Por otro lado, la arquitectura de la red de *IoT* se construyó basándose en capas de funcionamiento, de donde se emplea una semántica de análisis, donde por medio de un algoritmo de predicción de confort térmico basado en una técnica de la norma *ISO 7730*, se identificaron los resultados estimados de satisfacción, con ayuda de los sensores implementados.

Finalmente, en este trabajo se observa el funcionamiento de los controladores, la estimación del confort térmico, así como la comparativa antes y después de implementar el sistema en las habitaciones de pruebas, pretendiendo demostrar el correcto funcionamiento del proyecto.

Contenido

Índice de figuras	4
Introducción	9
Objetivos	11
Identificación del problema y propuesta de solución	12
Capítulo 1: Marco teórico.....	16
1.1 La higiene y el confort en los espacios confinados	17
1.2 Identificación de las variables involucradas en el confort térmico y calidad del aire de un hogar	19
1.2.1 Temperatura y humedad relativa en interiores	19
1.2.2 Concentración de gas monóxido de carbono CO y partículas <i>PM2.5</i> en interiores	21
1.3 Control de las variables de confort	23
1.4 Instrumentación de las variables de confort.....	26
1.5 El <i>Internet de las Cosas (IoT)</i> en las variables de confort	31
Capítulo 2: Metodología para la construcción del sistema de instrumentación y control.....	43
2.1 Identificación de las características del hogar de pruebas experimentales	43
2.2 Metodología para el control automático del sistema	44
2.2.1 Selección y uso de actuadores	49
2.2.2 Etapas del control automático.....	54
2.3 Metodología para la instrumentación del sistema	62
2.3.1 Selección y uso de sensores	63
2.3.2 Selección y uso del microcontrolador	79
2.4 Metodología para la predicción del confort térmico de acuerdo con la norma <i>ISO 7730</i>	80
2.5 Metodología para la construcción de la red de comunicación en <i>IoT</i>	87

Capítulo 3:	Construcción del sistema de instrumentación y control.....	94
3.1	Construcción de la red eléctrica de mando y potencia.....	94
3.2	Construcción de algoritmos de control	95
3.2.1	Algoritmo de control de temperatura	95
3.2.2	Algoritmo de control de la calidad del aire	97
3.3	Construcción de algoritmo de adquisición de datos	99
3.3.1	Obtención experimental de curvas de sensibilidad del sensor <i>MQ7</i>	99
3.3.2	Algoritmos para la adquisición de datos	103
3.4	Construcción de algoritmo para la predicción del confort térmico basado en la norma <i>ISO 7730</i>	105
3.5	Construcción de red de comunicación en <i>IoT</i>	107
3.5.1	Pruebas de comunicación con el <i>ESP32</i>	107
3.5.2	Pruebas de comunicación con el <i>Raspberry Pi 4</i>	109
3.5.3	Pruebas de comunicación con el servidor de <i>Amazon AWS</i>	110
3.5.4	Pruebas de comunicación con el servidor de base de datos	111
3.5.5	Pruebas de comunicación con la interfaz gráfica.....	113
Capítulo 4:	Ejecución del sistema de control e instrumentación.....	118
4.1	Experimentos en las habitaciones del hogar de pruebas	120
4.1.1	Habitación 1	121
4.1.2	Habitación 2.....	124
4.1.3	Habitación 3.....	126
4.2	Conclusiones de los datos obtenidos	129
4.3	Trabajos a futuro.....	131
Referencias.....		133
Anexos.....		139
Anexo A		139
Anexo B		140
Anexo C		142

Anexo D	151
Anexo E.....	153
Anexo F.....	158

Índice de figuras

Figura 1. Propuesta de diseño simplificado para el control del confort e higiene.....	13
Figura 2. Diagrama simplificado de propuesta de domótica.	14
Figura 3. Ejemplo de un perfil ideal de usuario con un rango aceptable de valores, para hábitos deseados de concentración de gas CO ₂ , temperatura y humedad relativa en el ambiente [2].	20
Figura 4. Tamaño relativo de las partículas en micrómetros [8].	23
Figura 5. Clasificación resumida de algunos controladores clásicos.	25
Figura 6. Correspondencia entre entradas y salidas de un controlador.	26
Figura 7. Proceso de muestreo, cuantificación y codificación binaria de una señal [12].	29
Figura 8. Arquitectura de IoT por capas. (a) Forma en tres capas. (b) Forma basada en Middleware. (c) Forma en SOA. (d) Forma en 5 capas [16].	33
Figura 9. Elementos constitutivos del IoT.....	34
Figura 10. Capas de seguridad en el flujo de la información en MQTT.	38
Figura 11. Ventajas de usar el protocolo MQTT.....	39
Figura 12. Red de comunicación simplificada a través de un bróker MQTT.	40
Figura 13. Regla de correspondencia entre los parámetros de entrada y de salida del sistema.....	47
Figura 14. Propuesta del sistema de control y monitoreo a detalle.	48
Figura 15. Carta psicométrica del enfriamiento evaporativo, [21].	50
Figura 16. Actuador comercial de acondicionamiento de aire por enfriamiento evaporativo marca Artic Air Ultra 18008.	50

Figura 17. Diagrama de funcionamiento del actuador Artic Air Ultra 18008 en conjunto con el controlador diseñado.	51
Figura 18. Especificaciones técnicas de extractor Igotto Swiss EX4-16W.....	52
Figura 19. Extractor de aire Igotto Swiss EX4-16W y conexión con tubo flexible de 3" x 2m.	52
Figura 20. Diagrama de funcionamiento del actuador Igotto Swiss EX4-16W en conjunto con el controlador diseñado.	53
Figura 21. Correlación de temperatura y humedad relativa del cuarto.	55
Figura 22. Correlación de partes por millón de monóxido de carbono y partículas PM 2.5.....	58
Figura 23. Estructura del sistema de control y medición para la habitación.	62
Figura 24. Registros de memoria del sensor PMSA003I.	64
Figura 25. Algoritmo planteado en pseudocódigo para obtener las lecturas del sensor PMSA003I en unidades de microgramos por metro cúbico, en base a sus registros de memoria 6 y 7.	65
Figura 26. Diagrama eléctrico de sensor MQ7 alimentado a tensión de 5 V.	66
Figura 27. Curvas de sensibilidad del sensor MQ7 [24].	67
Figura 28. Relación de proporción entre los factores de sensibilidad. El triángulo verde representa el factor dado por el fabricante y el triángulo azul representa el factor a calibrar.	69
Figura 29. Algoritmo de calibración del sensor MQ7.	72
Figura 30. Relación no proporcional entre los factores de sensibilidad. El triángulo azul representa el factor calibrado de la sustancia aire y el triángulo morado representa el factor del monóxido de carbono CO.	73
Figura 31. Divisor de voltaje para LDR.	76
Figura 32. Conjuntos de espacio muestral para personas satisfechas e insatisfechas.	81
Figura 33. Formato de gráfico representativo de PMV (voto medio estimado de satisfechos) y PPD (porcentaje estimado de insatisfechos por el sistema de control térmico), [4].	84

Figura 34. Herramientas utilizadas para desarrollo en IoT.	87
Figura 35. Diseño de la red de comunicación IoT basada en jerarquía militar.	88
Figura 36. Bosquejo de referencia para construcción de la interfaz gráfica para el dispositivo supervisor.	93
Figura 37. Objetos de la red de IoT.	95
Figura 38. Algoritmo de control de temperatura en pseudocódigo para usarse como subrutina en Python.	96
Figura 39. Algoritmo de control de calidad del aire en pseudocódigo para usarse como subrutina en Python.	98
Figura 40. Evidencia de obtención inicial de precalentamiento y calibración de sensor MQ7.	100
Figura 41. Obtención de resultados experimentales del algoritmo de calibración.	100
Figura 42. Lectura de voltaje V _{RL} obtenida con el multímetro.	101
Figura 43. Curvas de sensibilidad obtenidas para detección de monóxido de carbono en relación con los datos experimentales de la calibración del sensor MQ7.	103
Figura 44. Algoritmo de adquisición de datos con el ESP32 escrito en C.	104
Figura 45. Algoritmo para estimación de PMV y PPD en pseudocódigo para usarse en Python.	105
Figura 46. Display mostrando conexión exitosa a la red de área local y bróker del servidor.	107
Figura 47. Pruebas de funcionamiento de comunicación con el ESP32.	108
Figura 48. Pruebas de funcionamiento de comunicación con el Raspberry Pi 4.	109
Figura 49. Pruebas de funcionamiento de comunicación con el servidor de Amazon AWS.	110
Figura 50. Relación 1 a 1 entre tablas.	112
Figura 51. Prueba de conexión remota a los registros de la base de datos.	112

Figura 52. Flujos en Node-RED.....	113
Figura 53. Prueba de conexión de la interfaz gráfica con el sistema, mostrando la primera mitad de la página del panel de control.....	114
Figura 54. Prueba de conexión de la interfaz gráfica con el sistema, mostrando la segunda mitad de la página del panel de control.....	116
Figura 55. Panel de control para dispositivo móvil. (a)Panel de temperatura. (b) Panel de calidad del aire. (c) Panel de control manual.....	117
Figura 56. Plano del hogar de pruebas.	121
Figura 57. Pruebas efectuadas antes y después de la implementación del sistema de control en la habitación 1. (a) Situación de confort térmico antes. (b) Situación de confort térmico después. (c)Situación de confort por calidad del aire antes. (d)Situación de confort por calidad del aire después.	122
Figura 58. Comportamiento de los controladores de temperatura y calidad del aire en la habitación 1. (a) Control de temperatura. (b) Control de partículas PM2.5. (c) Control de partes por millón de CO.	124
Figura 59. Pruebas efectuadas antes y después de la implementación del sistema de control en la habitación 2. (a) Situación de confort térmico antes. (b) Situación de confort térmico después. (c)Situación de confort por calidad del aire antes. (d)Situación de confort por calidad del aire después.	125
Figura 60. Comportamiento de los controladores de temperatura y calidad del aire en la habitación 2. (a) Control de temperatura. (b) Control de partículas PM2.5. (c) Control de partes por millón de CO.	126
Figura 61. Pruebas efectuadas antes y después de la implementación del sistema de control en la habitación 3. (a) Situación de confort térmico antes. (b) Situación de confort térmico después.	

(c) Situación de confort por calidad del aire antes. (d) Situación de confort por calidad del aire después.127

Figura 62. Comportamiento de los controladores de temperatura y calidad del aire en la habitación 3. (a) Control de temperatura. (b) Control de partículas PM2.5. (c) Control de partes por millón de CO.128

Figura 63. Ejecución del algoritmo de predicción de confort térmico basado en la técnica de la norma ISO 7730. (a) Predicción para habitación 1. (b) Predicción para habitación 2. (c) Predicción para habitación 3.129

Figura 64. Influencia de la luminiscencia en la temperatura dentro de cada una de las habitaciones. (a) Habitación 1. (b) Habitación 2. (c) Habitación 3.131

Introducción

El *hogar*, es una palabra de gran valor emocional para las personas, ya que desde miles de años atrás, el ser humano decidió asentarse en un sitio específico para resguardarse de los peligros y amenazas del exterior, para luego formar los primeros grupos humanos más elementales de una sociedad, la *familia*. De este modo el hogar sigue siendo hasta la actualidad, un espacio físico protegido donde se alojan las personas, y que durante el transcurso de los años, los peligros siempre han estado presentes, y sin embargo, se ha entendido que ciertos de ellos inclusive logran penetrar en él, tales como amenazas virales, proliferación bacteriana y micótica provocadas por la mala calidad del aire, teniendo como hipótesis que controlando aquellas variables relacionadas con la proliferación, tales como temperatura, emisiones de gases nocivos que afecten al sistema respiratorio, así como partículas contaminantes suspendidas, se puede evitar y controlar de manera indirecta también, la salud, la higiene y por ende el confort en los espacios cerrados utilizando actuadores comerciales y de bajo costo. El tiempo que se pasa viviendo dentro del hogar, implica hacerse en condiciones ideales de comodidad y limpieza, lo cual está intrínsecamente relacionado con el nivel de calidad de vida de una persona, es decir el *confort e higiene*, los cuales son aspectos medibles y controlables. Es por esto que la motivación del presente trabajo es atender los sectores de viviendas mexicanas más representativas de acuerdo a sus necesidades; desde el punto de vista de la ingeniería con el control e instrumentación de procesos, así como con la aplicación del *Internet De las Cosas*, conocido por sus siglas *IoT*. Basado en estas tecnologías, el prototipo funcional desarrollado en este trabajo puede ser escalable y usarse de manera remota gracias al diseño de una red de dispositivos funcionando con servicios en la nube (Cloud Services), creando

un servidor web propio, una base de datos relacional y la creación de una interfaz gráfica amigable con el usuario que lo oriente a entender la situación de confort e higiene dentro de su hogar. Además, en la construcción del prototipo se empleó un hardware y un sistema operativo de código abierto conocidos como de tipo Open Source, ofreciendo una ventaja en costos y accesibilidad. Los diseños mostrados son propuestas pensadas para controlar y automatizar la higiene y confort de un hogar mexicano, a través del acondicionamiento térmico del aire y su purificación, con ayuda de sensores y actuadores de bajo costo así como bajo consumo energético. Finalmente, se presentan resultados de funcionamiento y se enfatiza la ventaja de implementar y usar sistemas basados en el *internet de las cosas* así como aprovechar su potencial.

Objetivos

Desarrollar un sistema de medición y control basado en sensores y el uso de *Internet de las Cosas (IoT)*, capaz de controlar un actuador de aire acondicionado y un extractor para purificar el aire a través de un controlador del tipo *todo o nada*.

Objetivos específicos:

- i. Construir un sistema instrumentación y control para mejorar las condiciones de confort, en un entorno doméstico empleando una interfaz gráfica para el usuario y la tecnología del *Internet De Las Cosas (IoT)*, de manera que sea de bajo costo empleando solo *Software y Hardware* basado en *Open Source*.
- ii. Implementar el sistema de instrumentación y control dentro de un hogar de pruebas y verificar el funcionamiento correcto del mismo.
- iii. Relacionar los parámetros ambientales medidos por el sistema de instrumentación con el confort, para que se pueda definir si también la higiene es adecuada, alertando al usuario y manipulando a los elementos finales de control a través del *Internet De Las Cosas (IoT)*.
- iv. Implementar un algoritmo de predicción de confort térmico, basado en la técnica de la norma *ISO 7730*, para que pueda estimar indirectamente si un grupo de usuarios se sentirá satisfecho con el sistema.

Identificación del problema y propuesta de solución

En materia de salud pública, y hasta la fecha de la elaboración de este documento, la población ha atravesado una etapa reciente de postpandemia que ha implicado hacer conciencia de las consecuencias de tener una mala higiene en los espacios cerrados y de quienes requieren de una correcta ventilación y purificación del aire, por medio de un control térmico. Controlar estos aspectos se puede lograr con actuadores de bajo costo y bajo consumo energético, destinados para uso de espacios relativamente pequeños como en habitaciones de un hogar promedio, con ayuda de tecnologías que siguen en mejora continua, tal es el caso del *IoT*.

El proyecto consiste en la medición y el control de las variables de higiene y confort inherentes de un espacio doméstico, para una habitación estándar de referencia o a cualquier otro espacio confinado. La Figura 1 muestra el sistema de control propuesto de manera simplificada, del cual se observa un bloque central denominado *planta*, refiriéndose a la habitación misma y detallando esto a mayor profundidad en el Capítulo 2, así mismo se observa que la *planta* conecta a un bloque denominado *parámetros deseados de confort*, que contiene todos los valores deseados de confort por el usuario, donde en dicha figura también se observa otro bloque que contiene los *parámetros medidos de confort* para la regulación de variables, el cual consta principalmente de tres consignas de control, las cuales son temperatura, partículas suspendidas tipo *PM2.5* y concentración de gas monóxido de carbono (CO) en la habitación de pruebas, es decir la *planta*, ya que estas variables están relacionadas con el confort térmico y calidad del aire en interiores domésticos.

Posteriormente debe existir una retroalimentación de señal hacia otro bloque llamado *control del confort*, el cual se encarga de regular dos actuadores del hogar o elementos finales de control para así manipular los estados de encendido y apagado, el de un acondicionador de aire como el de un extractor para purificar el ambiente de partículas suspendidas.

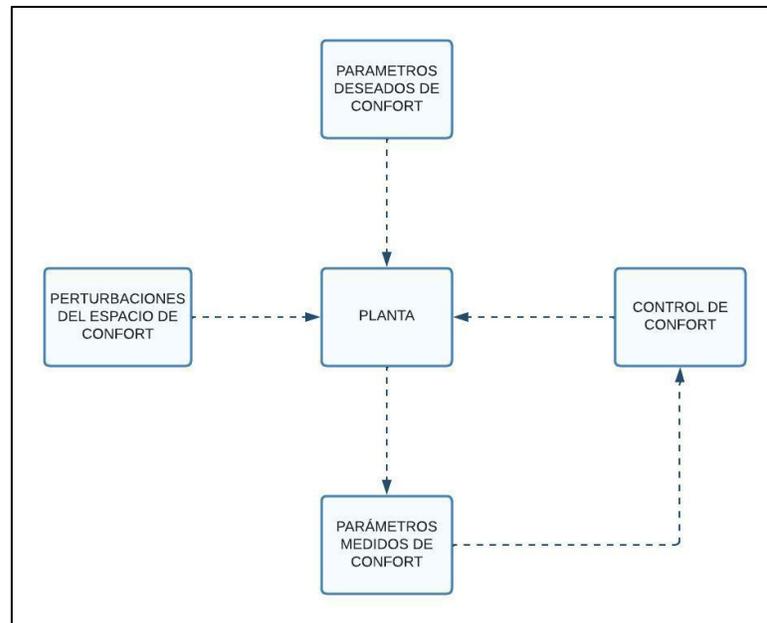


Figura 1. Propuesta de diseño simplificado para el control del confort e higiene.

Los actuadores empleados no poseen sensores de monitoreo así que se tienen que diseñar los controladores adecuados así como seleccionar los sensores para su cometido, los cuales se encuentran en una gran variedad de marcas comerciales, capaces de detectar perturbaciones externas a través del bloque *perturbaciones del espacio de confort*, quienes deben ser atendidos por el bloque de control, esperando una estabilización de la señal de medición respecto a una consigna deseada, en un determinado tiempo.

El modo de comunicación entre los sensores y actuadores se puede lograr a través de una Red de Internet, en la cual los Clientes se podrán comunicar para recibir datos desde una Base de Datos Relacional a distancia o en su defecto, en tiempo real y de manera inalámbrica por medio de un servidor *web* en la *nube*. *Amazon AWS* es un proveedor de servicios que debido a su gran versatilidad de usos, los desarrolladores pueden crear sus propias soluciones en domótica a través del *internet de las cosas*. La Figura 2 muestra cómo los actuadores del hogar son gobernados por un microcontrolador, que a su vez es empleado por un dispositivo servidor local en la red y que interactúa con su servidor en la *nube*, para que después otros clientes de Internet puedan acceder a esta información por medio de una interfaz gráfica para el usuario.

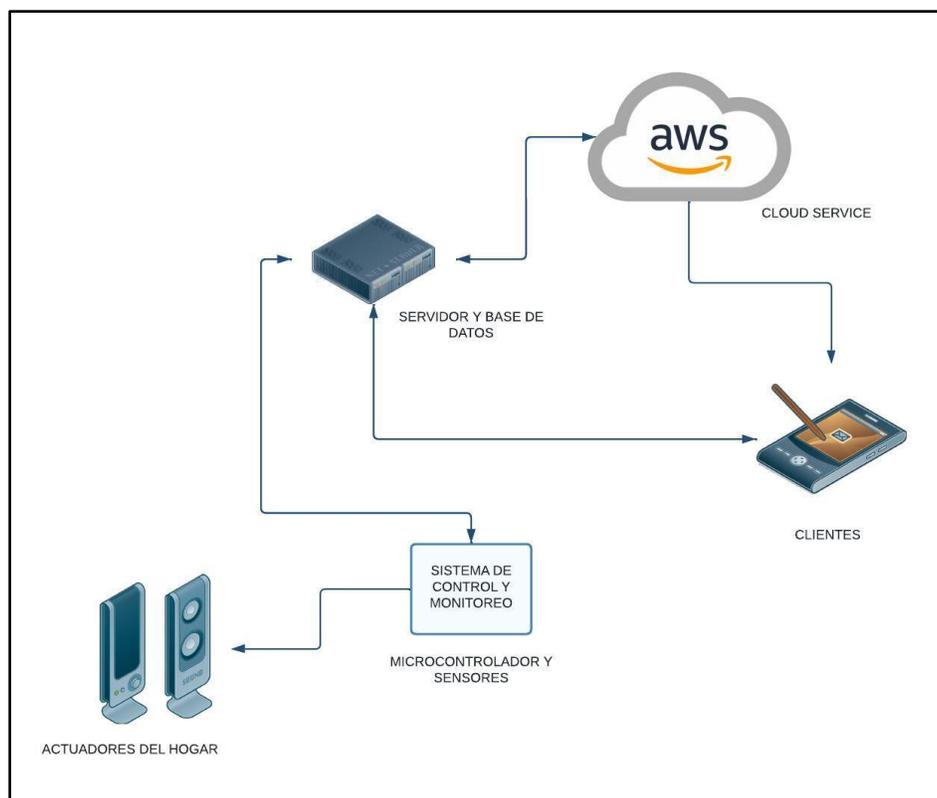


Figura 2. Diagrama simplificado de propuesta de domótica.

Finalmente, se emplea el uso de tecnologías basadas en *Open Source Hardware*, es decir todos aquellos controladores y dispositivos que hayan sido fabricados para usarse en código abierto, lo cual reduce el costo de tener que adquirir licencias de uso de programas comerciales como de ciertas marcas para la instrumentación virtual, sin mencionar el uso de sistemas operativos también del tipo *Open Source*, el cual ofrece grandes ventajas, esto se detalla en el Capítulo 3.

Capítulo 1: Marco teórico

Existe una relación intrínseca entre el confort como comodidad y la higiene en los espacios de un hogar, dando lugar al concepto de *variables de confort*, así como su relación con la limpieza y salud de las personas en sus viviendas. A saber, las condiciones actuales en materia de salud pública internacional han sido críticas desde el año 2020 y con el transcurso de los meses, han sido numerosos los esfuerzos por mantener a raya la propagación de enfermedades virales y que son transmisibles con una relativa sencillez. Diversas áreas del conocimiento se han sumado a aportar ideas y soluciones para abordar las problemáticas de salud, evidenciando la gran necesidad e importancia de involucrar a investigadores y desarrolladores a sumarse a estos esfuerzos colectivos y multidisciplinarios, así que la *Organización Mundial De La Salud (OMS)*, ha estado actualizando sus recomendaciones en materia de desarrollo tecnológico en etapas de postpandemia, dirigidos a los centros de salud y población confinada en sus hogares, para tener ciertos requerimientos tecnológicos, pero esclareciendo la característica principal de que deben ser tecnologías que puedan estar al mejor alcance posible. Es por esto por lo que el documento titulado *Compendio de la OMS de tecnologías sanitarias innovadoras para entornos de bajos recursos* [1], divulga un listado de tecnologías propuestas para sectores socioeconómicos específicos de la población y por lo que como desarrollo de este proyecto, se pretende sumarse a este conjunto, pues de acuerdo a su publicación actualizada a partir del año 2021, es primordialmente una invitación abierta a investigadores, profesionales, desarrolladores y público en general, para sumarse e interactuar en la aportación de ideas y generar una difusión abierta del conocimiento en tecnologías sanitarias adecuadas, pero sobre todo en dirección hacia un acceso universal a ellas.

El uso de metodologías de control e instrumentación en conjunto con el uso de comunicación por Internet es atractivo en todo proceso moderno hoy en día, al menos a la fecha de publicación del trabajo presente, donde el interés de uso es aplicarlo a las condiciones de los entornos del hogar en México, a través de la medición y control de las *variables de confort*, desde un punto de vista prioritario a aquellas personas confinadas por mucho tiempo en sus hogares debido a que puedan estar en calidad de pacientes o de trabajo remoto. Conocer las metodologías de desarrollo a bajo costo para construir sistemas de instrumentación y control que permitan mantener las condiciones de salud, higiene y confort, garantiza un espacio interior adecuado y más digno en la calidad de vida de las personas.

1.1 La higiene y el confort en los espacios confinados

Definidas las circunstancias actuales en materia de salud pública a nivel mundial, la sociedad ha tenido que adoptar nuevos hábitos de higiene y consigo una mayor concientización referente a los mecanismos de transmisión no solo del *COVID-19*, sino también de cualquier otro tipo de enfermedades, ya sean o no virales. Esto ha implicado que sea altamente necesario mantener los espacios del hogar en condiciones correctas de higiene y por consiguiente, asegurar el confort.

Esta alta difusión de recomendaciones sigue amplificando el mensaje en asegurar que en los espacios compartidos exista una correcta ventilación y que permita el flujo continuo de aire, como un salón de clases, oficinas, gimnasios, restaurantes, etc. Siendo el ambiente mismo en el que se encuentre una persona interactuando, sea cual sea la circunstancia. Las *variables confort*, que son aquellas variables relacionadas con el confort térmico y la calidad del aire [2], tales como la

temperatura, humedad relativa, partículas contaminantes suspendidas en el aire y concentración de gases nocivos, las cuales tienen como características principales que son medibles, controlables y pueden retroalimentarse con algún modelo de actuación para regularse a sí mismas bajo ciertas circunstancias. Esto conlleva a la siguiente pregunta, *¿Por qué controlar estas variables?*, siendo las posibles respuestas, por higiene, bienestar y para generar un espacio de confort para las personas.

Por otro lado, la *OMS* indica algunos aspectos relacionados con la propagación de *COVID-19* y el mal acondicionamiento o purificación del aire, y que con fines de higiene propone sistemas naturales o mecánicos.

La ventilación consiste en introducir aire limpio en un espacio mientras se elimina el aire sucio, con el fin de mantener o mejorar la calidad ambiental. La ventilación se puede lograr de forma natural (por ejemplo, abriendo una ventana), o instalando un sistema mecánico. La propagación del *COVID-19* ocurre con mayor frecuencia cuando una persona infectada está en contacto cercano o directo con otra persona. El riesgo de propagación del virus es mayor en espacios abarrotados y mal ventilados donde las personas pasan largos períodos de tiempo juntas en las proximidades. Mejorar la ventilación interior puede reducir el riesgo de que el virus se propague en interiores [3].

Por lo anterior, es entonces necesario involucrar la importancia y relación entre la temperatura y humedad en el ambiente, así como la calidad del aire interior, considerándolos como los parámetros de higiene y confort para los usuarios dentro de un espacio.

1.2 Identificación de las variables involucradas en el confort térmico y calidad del aire de un hogar

Uno de los aspectos más importantes es identificar cuáles son las denominadas *variables de confort*, que están relacionadas con la higiene y bienestar adentro de las habitaciones y espacios confinados del hogar. Como se mencionó en la sección 1.1, la calidad del aire al interior de una habitación es fundamental para evitar la proliferación de virus, bacterias y hongos, mientras que la temperatura y humedad al interior, están relacionadas con la sensación térmica y comodidad de las personas, sin mencionar que también existe una relación intrínseca entre dichas variables, con el crecimiento de ciertos tipos de microorganismos micóticos que se desarrollan en las paredes y techos de casas mal ventiladas o con exceso de humedad.

1.2.1 Temperatura y humedad relativa en interiores

Algunos autores, han estudiado e identificado la humedad relativa en el ambiente y la temperatura, como variables relacionadas con el confort térmico, siendo este último considerado subjetivo a los usuarios, e involucran otras variables como la tasa metabólica, aislamiento de la ropa, temperatura del aire, temperatura radiante media de cada persona y la velocidad del aire [2], por lo que es posible simplificar estos censos de personas votantes, a través de modelos matemáticos que puedan predecir el confort térmico de las personas, tal es el caso de un modelo llamado *voto medio estimado*, encontrado en el estándar *ASHRAE 55-2004* o en normas basadas en el estándar internacional *ISO 7730-2005* [4], por medio de una escala de siete puntos de sensación térmica del usuario. De acuerdo con [2], este tipo de modelos han sido criticados debido a la posible inexactitud de las predicciones, y que en la realidad los usuarios podrían diferir y sería

poco prudente modelar la opinión humana bajo un modelo único para todos. Por tanto, lo que sugiere el autor es utilizar algo denominado *controlador de calidad del aire y confort térmico basado en perfiles*, el cual consiste en hallar patrones de hábitos de los usuarios (en cuanto a los límites deseados que estos eligen), y así el controlador aprende a interpretar la subjetividad del confort térmico. La Figura 3 muestra como el perfil de un usuario está delimitado por un rango aceptable de consignas definidas por él, siendo la temperatura, humedad relativa y otro eje más relacionado con las partes por millón de gas dióxido de carbono (CO₂). El prisma central representa el rango ideal de valores deseados por el usuario, mientras que el prisma exterior es un rango aceptable máximo [2].

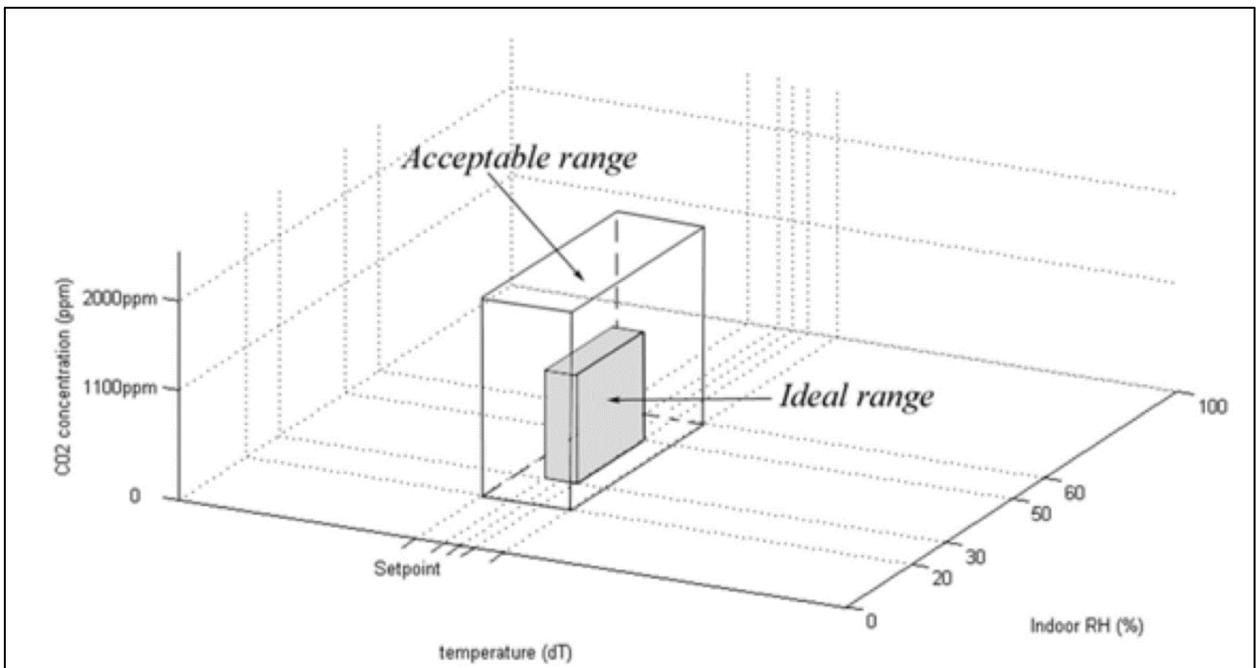


Figura 3. Ejemplo de un perfil ideal de usuario con un rango aceptable de valores, para hábitos deseados de concentración de gas CO₂, temperatura y humedad relativa en el ambiente [2].

Esto quiere decir que las variables de confort sugeridas en la Figura 3 deben mantenerse dentro de un umbral de aceptación deseado, siendo pertenecientes a una de las habitaciones del hogar.

1.2.2 Concentración de gas monóxido de carbono CO y partículas *PM2.5* en interiores

En cuanto a la calidad del aire en el interior de los hogares, tal como se menciona en [2, p. 110], ésta depende de los niveles de contaminación del exterior que representen un riesgo para la salud humana, es decir, la calidad de aire del interior del hogar depende de la calidad del exterior, tomando en consideración que el a partir del 80% o más de la población expuesta al medio ambiente, no presenten algún tipo de insatisfacción, esto según las definiciones de la *Sociedad Americana de Ingenieros en Calefacción, Refrigeración y Aire Acondicionado (ASHRAE)* citadas por el autor, pero respetando que solo son las autoridades competentes, quienes determinen lo anterior mencionado. Ciertos contaminantes tales como el dióxido de carbono (CO₂), el cual es medible en *partes por millón*, es un agente de interés en [2, p. 111] que puede definir la calidad del aire, pero es también una fuente natural emitida por los seres humanos en el proceso de exhalación, y al no mencionar explícitamente algún tipo de actuador, los altos niveles podrían regularse por medio de plantas de interior, que pudieran mejorar la calidad del aire para este tipo de gas y de manera natural. Como se mencionó anteriormente, las autoridades competentes de cada país determinan los criterios de contaminación, y dada la naturaleza de estos mismos, dependen también de cada ciudad conurbada o zona rural, así que, para el trabajo presente, se les da prioridad a las recomendaciones estipuladas por las *Normas Oficiales Mexicanas (NOM)*, en materia de salud ambiental. Algunas de ellas son utilizadas por la *Comisión Federal para la Protección Contra Riesgos Sanitarios (COFEPRIS)* [5], el cual menciona que la calidad del aire está determinada por los niveles de contaminación de agentes como partículas tipo *PM 1.0* y *PM2.5*, así como gases de ozono, plomo, dióxidos de azufre o nitrógeno, y monóxido de carbono (CO), siendo este último de interés particular para el trabajo presente, ya que es un compuesto mayoritario en fuentes como el humo de tabaco.

Otros autores como [6], estudiaron el comportamiento de otros agentes contaminantes en la atmósfera relacionados con las partículas *PM2.5* explícitamente en la zona urbana de la Ciudad de México, en donde de las muestras examinadas, concluyeron la presencia en abundancia de *microplásticos* suspendidos en el aire, por lo que este tipo de variable es de interés particular para el trabajo presente, ya que dichas partículas pueden penetrar y perturbar la higiene en cualquier espacio cerrado como las habitaciones del hogar, así como aquellos lugares que alberguen una mayor cantidad de personas como en las estaciones de servicio de transporte *metro* o bien, cualquier otro lugar de estancia para personas por largos periodos de tiempo. Por otro lado, los *microplásticos* no son solo agentes de qué preocuparse, ya que la calidad del aire dentro de instalaciones es objeto de análisis para algunas investigaciones que relacionan la presencia de virus y bacterias en el ambiente con las partículas suspendidas como en [7]. El trabajo del autor menciona un instrumento de medición directo denominado *virómetro*, que consiste en la detección y medición de algunas clases de virus suspendidos, basado en el fenómeno óptico fluorescente en fluidos como el que propone, así que es posible poner a discusión, el usar métodos cualitativos para determinar la presencia viral o bacteriana en partículas suspendidas en el aire, quienes podrían ser del tipo *PM2.5*, y transmisibles en el aire, como lo describe [3], recomendando una buena ventilación.

El conteo de partículas suspendidas en aire, independientemente de lo que contengan, pueden medirse, siempre que tengan como mínimo un diámetro de 2.5 micrómetros, y de allí la justificación del nombre de *partículas PM2.5*. Como se puede observar en la Figura 4, las partículas *PM2.5* son del tamaño similar a bacterias y partículas de humo [8], aunque el tamaño de un virus es aún más pequeño, como se puede observar. Por lo tanto, es de interés particular poder medir la

cantidad de ellas para poder controlarlas, las cuales son detectables por sensores basados en el principio de funcionamiento de dispersión láser.



Figura 4. Tamaño relativo de las partículas en micrómetros [8].

Por lo tanto, la concentración de gases como el monóxido de carbono (CO) y las partículas tipo $\text{PM}_{2.5}$, son variables de interés particular debido a que están involucradas en la medición de la calidad del aire interior de una habitación.

1.3 Control de las variables de confort

Si las variables de confort son la temperatura, humedad relativa, partículas suspendidas en el aire y concentración de gases, entonces es necesario comprender cómo gobernarlas. Para ello, la teoría del control es un campo de estudio de interés particular para la ingeniería en general, pues su uso es de gran relevancia para la manipulación de gran cantidad de procesos que ocurren desde la industria hasta el mismo hogar. Sea cual sea la necesidad de manipular, gestionar y regular cualquier tipo de variables, el estudio del control debe ser involucrado para el análisis y desarrollo de controladores encargados de sus cometidos. En la Figura 5 se puede observar la clasificación de los controladores, de una manera resumida al control clásico, en la que se puede notar cómo existen dos tipos de retroalimentación, de *lazo abierto* y *lazo cerrado*. El lazo abierto no retroalimenta la salida hacia la entrada del controlador, ya que tiene como referencia la unidad del tiempo, un ejemplo podría ser un controlador *temporizador*, mientras que uno de lazo cerrado requiere de un conocimiento de la respuesta de salida respecto a su entrada, para hacer una comparación con las variables deseadas por el usuario. Por otro lado, existen controladores digitales y analógicos, por ejemplo, se tiene el caso de la máquina de Watt, la cual posee un tipo de *control analógico* ya que es de tipo mecánico y no hay un proceso computacional que se haga cargo de las operaciones y tareas para cumplir su cometido de regulación automática de fluido de trabajo. En contraparte, existe también el *control digital*, quien se encarga de involucrar procesos computacionales como los que hace un *microcontrolador* o cualquier otro dispositivo que pueda efectuar operaciones digitales. Por consiguiente, como se observa la Figura 5, existen dos tipos de controladores de interés particular en el control clásico, los $P+I+D$ también denominados *proporcionales, integrales y derivativos* y el de *eventos discretos*, en los que para el caso del *lazo cerrado* se denominan *sistemas de control de lazo cerrado discreto* [9, p. 30], esto debido a que el tipo de señales para este controlador son de tipo discretas, siendo su objetivo activar o desactivar dispositivos, de

acuerdo con la retroalimentación de una variable física medible, siendo este tipo de controlador de interés particular para el trabajo presente.

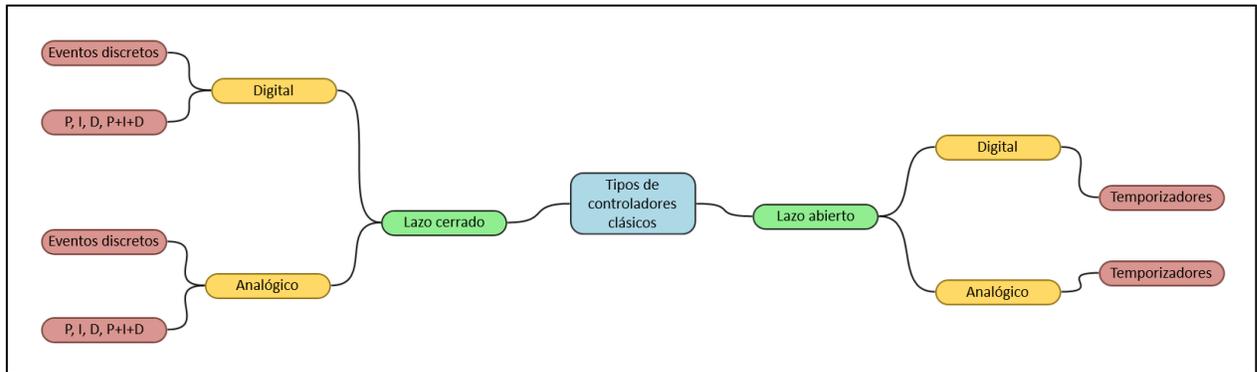


Figura 5. Clasificación resumida de algunos controladores clásicos.

Si existen variables relacionadas con la higiene y el confort de un hogar, entonces es importante identificar cuáles son y además si son controlables, esto en función de su naturaleza física y de qué manera son medibles. Es por lo que existen técnicas y metodologías estudiadas en la Teoría del Control Clásico, el cual se encarga de diseñar sistemas de control capaces de responder a mediciones deseadas de cualquier variable, para obtener las salidas deseadas. Por ejemplo, si la sensación térmica en el interior de una habitación es relativamente alta, entonces el sistema de control debe actuar gobernando a uno o más dispositivos relacionados con la manipulación de la temperatura para refrescar el interior, pero teniendo como referencia la temperatura deseada asignada por el usuario. Algunas de las definiciones para el concepto de *sistema de control*, especifican que son la constitución de procesos encargados de la manipulación y regulación de variables físicas, en las cuales sus comportamientos son denominados *respuestas de salida*, porque son consecuencia de una respuesta deseada por el usuario, conocida como *estimulo de entrada*

porque el *sistema de control* se comportará de acuerdo con las entradas recibidas [10]. En términos generales, la Figura 6 muestra cómo es esta regla de correspondencia entre entradas y salidas.

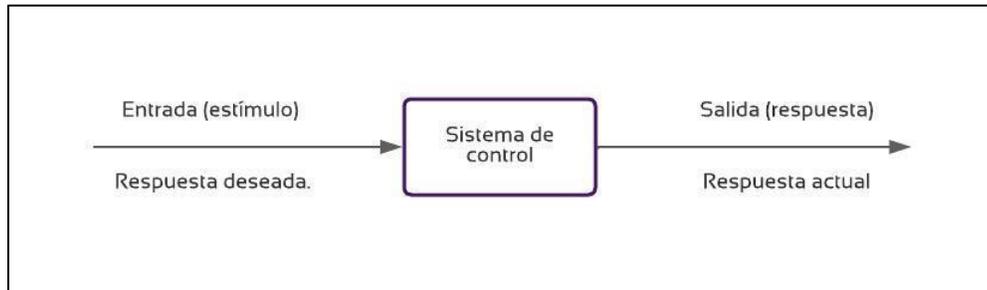


Figura 6. Correspondencia entre entradas y salidas de un controlador.

Finalmente, la higiene y el confort fueron dos aspectos de interés para implementar controladores de *lazo cerrado* y de *eventos discretos*, ya que el control clásico ha sido uno de los fundamentos más importantes para el desarrollo de sistemas de regulación y control de variables como lo son la temperatura y calidad del aire, pues dada la naturaleza de las variables, los costos y la versatilidad de funcionamiento con los dispositivos, se tiene la confianza de que se ejecutarán correctamente, siempre que exista un correcto diseño.

1.4 Instrumentación de las variables de confort

Los orígenes de la instrumentación comenzaron al mismo tiempo de cuando surgieron los primeros sistemas de control automático, y se volvió un campo de estudio independiente al control de procesos, pero el correcto funcionamiento de un controlador depende primordialmente de un correcto sistema de instrumentación. Para el trabajo presente, el control de las variables de confort depende de lograr interpretarlas adecuadamente, esto se lleva a cabo a través de sus valores de

medida. Es entonces que la participación de la *instrumentación electrónica* debe estar presente en todo sistema de control automático moderno, ya que esta área de estudio se encarga de desarrollar y estudiar sistemas electrónicos para acondicionar señales provenientes de *sensores* y *transductores*, así como procesar la información obtenida para generar lecturas de medición, en función de un sistema de unidad de medida, con el objetivo de ser encargados de la vigilancia y seguimiento de procesos [11]. Así que uno de los conceptos más básicos que se requiere conocer en la *instrumentación electrónica*, es el de identificar a los *sistemas de medida* empleados, en la que una definición de acuerdo con [11], indica que es una propiedad o cualidad para un objeto de estudio, que lo pueda describir, tal es el caso de la temperatura de cuarto, en el que independientemente de la aceptación internacional o que en un principio de manera regional, como las unidades de medida inglesas, la temperatura es ampliamente conocida por su uso en escala relativa dada en unidades de *grados centígrados*. Otro concepto fundamental es conocer la diferencia entre *sensores* y *transductores*, pues de acuerdo con [11], un *sensor* es un dispositivo capaz de reaccionar a la energía inducida por el ambiente, y en consecuencia ofrecer una señal interpretable en función de la variable medida. Por ejemplo, sensores biológicos como los del ser humano, pueden reaccionar a la temperatura ambiente dentro de una habitación e interpretar una *sensación térmica*, la respuesta de una persona podría ser la sudoración y la insatisfacción térmica, es decir, señales de respuesta interpretables y que pueden servir o no, para conocer el valor de la temperatura. Por otro lado, un *transductor*, es un dispositivo capaz de convertir esas respuestas del sensor a otras, por medio de un proceso de transformación de la energía. Esto último es de gran utilidad ya que aquellos transductores que convierten señales interpretables de sensores a energía eléctrica pueden garantizar una relación intrínseca entre la lectura de voltaje o corriente eléctrica, con las unidades de medida de la variable física sensada. Por otro lado, la *instrumentación electrónica* requiere de algún método para *acondicionar la señal transducida*, esto significa que la

señal eléctrica obtenida, deba ser interpretable por medio de técnicas de amplificación, filtrado de ruidos que dañen la calidad de la señal y su posterior discretización para ser admitida por sistemas digitales. Algunos sensores ya cuentan dentro de su fabricación con la parte del acondicionamiento de la señal de lectura, por otro lado, otros tipos de sensores requieren de la interpretación de la señal de salida a través de medios como los *convertidores analógicos a digitales*, así como calibrar las lecturas, siendo esto un caso muy común para los sensores analógicos. En la *instrumentación electrónica*, existen otros dos conceptos de interés, que son las lecturas *digitales* y *analógicas*, quienes están íntimamente relacionadas con el tipo de salidas del sensor, tal es el caso de algunos sensores de temperatura como el modelo comercial *DHT11*, que ofrece lecturas digitales, esto quiere decir que la naturaleza de la temperatura es continua e infinitesimal entre valores, pero en el proceso de medición se discretiza en unidades de 1°C, lo que significa que una variación de 20 a 20.5°C es imperceptible por el instrumento, pero de 20 a 21 °C, es posible su interpretación.

Aunque gran cantidad de sensores comerciales ya poseen en sus encapsulados los procesos de amplificación, filtrado y discretización de la señal medida, es prudente mencionar el proceso de tratamiento de la señal, con el fin de comprender el modo en que funciona la *resolución* de los sensores e instrumentos empleados, es decir el paso de lectura mínimo de variación. De acuerdo con [12], la señal de un sensor obtenida es inicialmente analógica, por lo que el fabricante debe definir en sus hojas de datos técnicos o *datasheets*, el *intervalo de muestreo* empleado, que es el tiempo que debe transcurrir antes de tomar una siguiente lectura. La amplitud y periodo de la señal se representan como una función $x(t)$ para señales analógicas y $x[n]$ cuando la señal ya esté muestreada, donde t es el parámetro de tiempo continuo y n es el número de muestra.

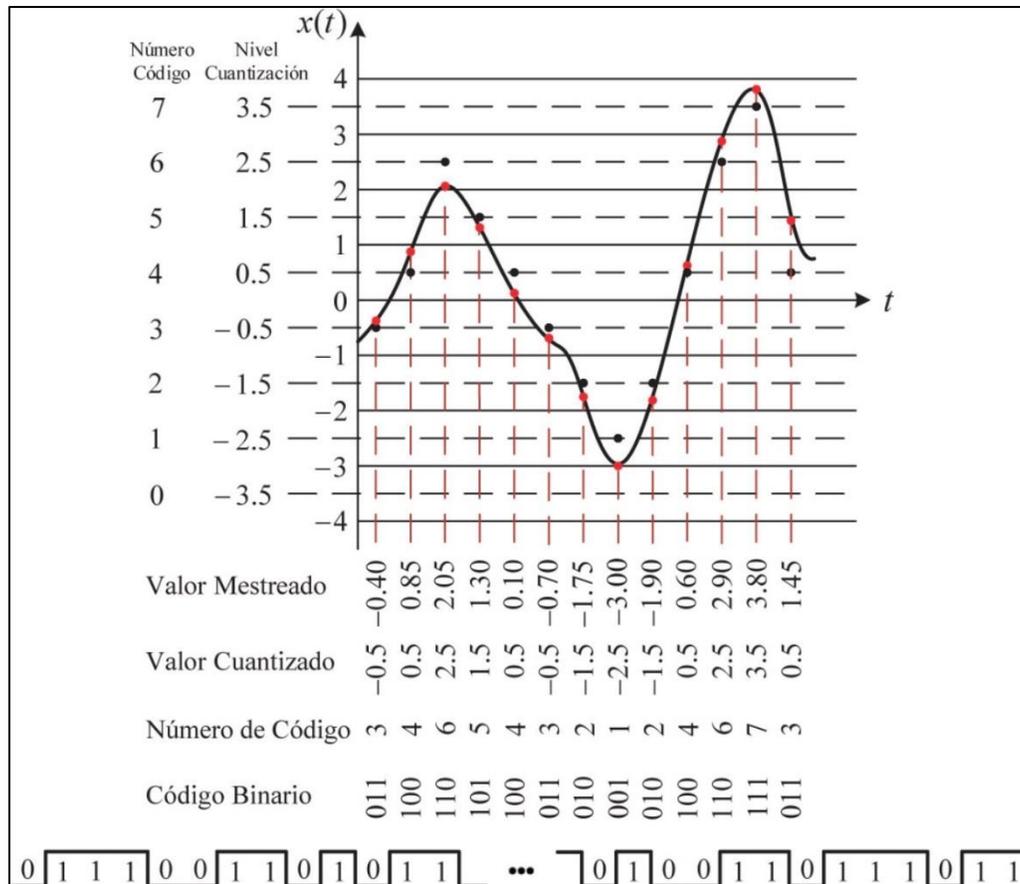


Figura 7. Proceso de muestreo, cuantificación y codificación binaria de una señal [12].

El siguiente paso en el acondicionamiento de la señal es el *proceso de cuantificación* [12], que consiste en discretizar los valores analógicos de la amplitud de la señal, en un número finito de valores, donde procedimiento es representar el paso de conteo, por medio una división en del eje de las ordenadas en intervalos iguales, e identificar el valor de cada división con un punto, tal y como se observa en la Figura 7 [12], los puntos en color rojo son los valores muestreados y los puntos en color negro son los valores cuantizados. Como último paso, se debe digitalizar la señal, esto se logra *codificando en sistema binario*, los valores cuantizados, tal y como ocurre con el sensor *DHT11* por ejemplo. Como se observa en la Figura 7, a cada valor binario le debe corresponder un valor decimal, el cual a su vez representa el valor cuantizado, así que de este modo

se logra convertir una señal analógica a digital, esto último está relacionado con la *resolución del instrumento digital de medición*, es decir la discretización de las lecturas, así que conocer estos aspectos es fundamental antes de utilizar cualquier sensor o instrumento comercial.

Otro concepto clave en la *instrumentación* es seleccionar el tipo de sensores o instrumentos adecuados, con base a *criterios del diseñador*, para el presente trabajo se usó el criterio de *bajo costo*, ya que en la actualidad se cuenta con sensores accesibles que ofrecen las mismas mediciones que los sistemas de instrumentación comerciales de mayor costo.

Por otro lado, la *exactitud* de un instrumento se refiere al modo en el que las lecturas que ofrece se acercan al valor real de la variable física, mientras que la *precisión* se refiere al modo en el que el instrumento ofrece los mismos valores de lecturas o datos en una tolerancia adecuada, siendo una medida de interés particular de la variable física [13], por lo tanto se podría afirmar que la calidad de un instrumento depende en gran medida de su precisión y exactitud.

Algunos trabajos mostrados en conferencias relacionados con la instrumentación y monitoreo como en [14], surgieron de la necesidad de monitorear variables relacionadas con la calidad del aire de los hogares durante la etapa de pandemia de *COVID-19*, en las que existe una relación entre la temperatura interior, la humedad relativa, y las partes por millón de monóxido y dióxido de carbono con la susceptibilidad a enfermarse por dicho agente viral, así que crearon un sistema de monitoreo de calidad del aire, que informara al usuario en tiempo real sobre las condiciones de su hogar, alertándolo por medio de su dispositivo móvil. Su sistema instrumental fue exitoso al lograr disminuir el error de las lecturas, por medio de un ajuste en la calibración de sus sensores, en los que usaron el modelo comercial *DHT11* para la temperatura y un modelo comercial *MQ7* para el monitoreo de partes por millón de concentración de gas monóxido y dióxido de carbono, por lo

que estos tipos de modelos son atractivos por sus bajos costos y la posibilidad de calibrar el sensor de gas con técnicas para reducir el error de medición de las lecturas, siendo un caso de interés también para el presente trabajo .

1.5 El *Internet de las Cosas (IoT)* en las variables de confort

El uso del Internet en la sociedad ha ido en aumento en los últimos años, convirtiéndose en una herramienta fundamental para la vida cotidiana. El trabajo presente se orienta a respaldar y estar a favor de su uso para mejorar la calidad de vida de las personas, utilizando una red de dispositivos que puedan interactuar entre sí, para que se permita controlar y monitorear las *variables de confort*.

Como se mencionó en las secciones 1.3 y 1.4, la relación entre el control y la instrumentación es fundamental y dependiente, sin embargo podrían requerirse de más tareas que tan solo medir y controlar procesos, pues es necesario adquirir y almacenar datos para su estudio, monitorear y controlar dispositivos a distancia, es decir sin la necesidad de que un usuario tenga que estar presente en donde se efectúen los procesos, deslindarse del uso de cableados que puedan complicar la interacción entre los dispositivos, así como también requerir de tareas informativas para el usuario, cuando las condiciones en el confort e higiene dentro del hogar puedan ser carentes. Esto conlleva al concepto clave de *Internet de las Cosas*, que por sus siglas en inglés es conocido como *IoT (Internet of Things)*, [15] lo define como una necesidad para la sociedad para el presente y el futuro, donde las personas en conjunto con las cosas puedan formar sistemas de información robustos y eficientes a través de redes y nodos inalámbricos. Para lograr lo anterior, se requiere del uso de sensores y actuadores trabajando en conjunto con una buena instrumentación.

El estudio del *IoT* es bastante extenso y lleno de gran cantidad de información, por lo que primero sería prudente y necesario, comenzar por esclarecer el uso que se le va a dar, siendo para el trabajo presente, el uso en soluciones en el hogar a través del control de variables de higiene y confort, que como se mencionó en la sección 1.2, son la temperatura y calidad del aire. Algunos trabajos de autores estudian a mayor profundidad las tecnologías necesarias y tipos de comunicación en *IoT*, tal como se explica en [16], donde los autores concluyen efectivamente que esta tecnología apunta hacia mejorar el nivel de calidad de vida de una persona, con el uso del Internet en sus dispositivos cotidianos.

Lo básico para comprender el *IoT*, es analizando su arquitectura de funcionamiento, tal como lo describe [16], indicando que se compone de *layers* o *capas*, que son niveles en la arquitectura de diseño del *IoT*, y que cada una se encarga de esclarecer, organizar y dividir todas aquellas funcionalidades y componentes necesarios para su uso. Los autores indican, a través de la Figura 5, que existen cuatro maneras de expresar la arquitectura *IoT*, las cuales son en forma de *tres capas* Fig. 8(a), en forma basada por el *middleware* Fig. 8(b), donde el *middleware* es el *software* intermediario encargado de relacionar los dispositivos conectados a Internet y aplicaciones de *IoT*. También en forma de *SOA* que por sus siglas en inglés significa *Arquitectura Orientada a Servicios*, Fig. 8(c) y en forma de 5 capas Fig. 8(d), donde esta última es de interés particular debido a que esclarece mejor las partes de la percepción a través de los sensores, la red de comunicación y las aplicaciones de *IoT*.

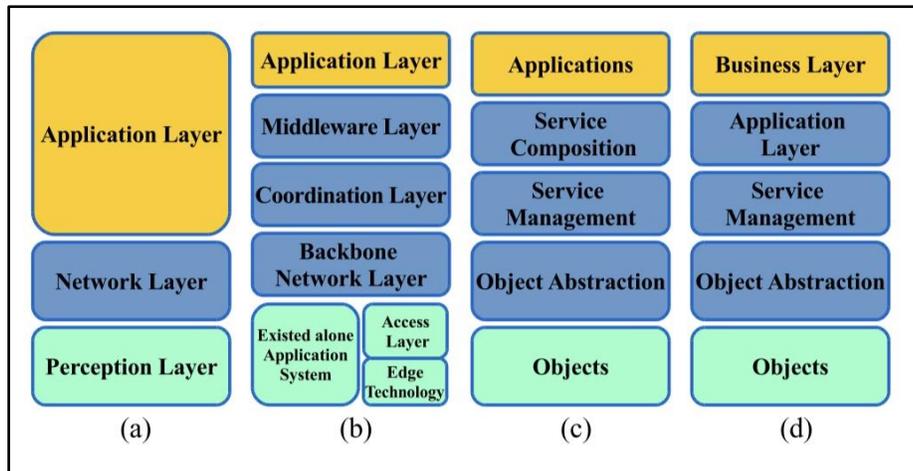


Figura 8. Arquitectura de IoT por capas. (a) Forma en tres capas. (b) Forma basada en Middleware. (c) Forma en SOA. (d) Forma en 5 capas [16].

En la Figura 8(d) se puede observar la primera capa llamada *objects*, que es la *capa de objetos*, los cuales son todos aquellos dispositivos encargados de la percepción e interacción con el ambiente, es decir que representa a todos los sensores y actuadores que monitorean y regulan la temperatura, humedad o calidad del aire, para el caso particular del presente trabajo. La Figura 8(d) muestra también cómo hay una denominada *object abstraction*, que es la *capa de abstracción*, encargada de transferir los datos producidos por la *capa de objetos* hacia la *capa de gestión de servicios*, por medio de canales seguros, tal es el caso de tecnologías como *WiFi*, *5G* o *Bluetooth*, por tan solo mencionar algunas. La *capa de abstracción* también puede encargarse de tareas de administración de datos y la *computación en la nube* conocida como *cloud computing*. La siguiente capa es *service management layer*, que como se mencionó, es la *capa de gestión de servicios*, encargada del *middleware*, que es el software intermediario, quien recibe datos de la capa de abstracción para gestionar los datos, así como emparejar un determinado servicio con el solicitante, por medio de direcciones y nombres. Esta capa ayuda al programador a trabajar con múltiples objetos conectados. La siguiente capa en la arquitectura es *application layer* o *capa de aplicación*,

encargada de proveer los servicios solicitados por los *clientes*, a quienes se les identifica de este modo a todos aquellos dispositivos de la red tales como computadoras y celulares, quienes soliciten información de datos como la temperatura y calidad del aire de la habitación del hogar, con el fin de ser interpretable por los usuarios. Por último, la capa *Buisness layer* o *capa de negocios*, es la encargada de representar los datos obtenidos del monitoreo y supervisión de variables, por medio de gráficos o diagramas de flujo provenientes de la capa de aplicación. El nombre de *negocios* proviene de que esta capa puede construir también modelos de negocios, para el caso de trabajar con datos relacionados con aplicaciones empresariales, pero para el caso de soluciones en el hogar, representa el comportamiento de los datos de variables de higiene y confort y obtener un análisis de comportamiento.

Los elementos constitutivos del *IoT* son la *identificación*, *sensado*, *comunicación*, *computación*, *servicios* y *semántica*, esto se puede observar en la Figura 9.

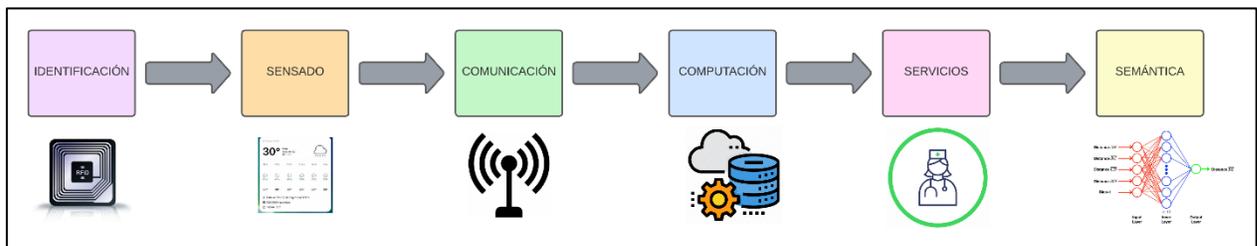


Figura 9. Elementos constitutivos del *IoT*.

El elemento de *identificación* se refiere al nombramiento de los dispositivos participantes en la red *IoT*, donde la identificación se efectúa a través de los *protocolos de internet*, que por sus siglas en inglés se conoce como *IP* y es un código numérico asignado por las redes de área local *WiFi* a los dispositivos que estén conectados. El elemento de *sensado* en *IoT*, se refiere a la producción de

datos de los objetos dentro de la red, como son sensores y ciertos tipos de actuadores conectables a Internet, donde los datos son alojados en alguna base de datos existente, donde la acumulación de estos datos es primordial para tomar decisiones de actuación en función de su análisis. El siguiente elemento es el de *comunicación*, esto se refiere a los *protocolos de comunicación de Internet*, esto es el uso de tecnologías de transmisión de datos a través de canales de comunicación, por ejemplo, en *IoT* es común usar protocolos *WiFi*, *Bluetooth*, *IEEE 802.15.4* o *Zig-Bee*, por mencionar solo algunos. El protocolo *WiFi* es de interés particular para el trabajo presente ya que este permite a los dispositivos conectados a comunicarse entre sí, intercambiando información mutuamente. El protocolo *IEEE 802.15.4* también es de suma importancia porque este expresa las topologías de red formadas por sensores con capacidades de monitoreo [15], operando en anchos de banda *WiFi* de 2.4 GHz, en el que el propósito principal de este estándar es dar seguridad a transmisión de datos de baja cantidad de información, pero alta velocidad de transferencia en la red. El elemento constitutivo de *computación*, se refiere a los recursos computacionales de *hardware* y *software* empleados en *IoT* para el procesamiento de los datos generados por sensores, por ejemplo microcontroladores o tarjetas de adquisición de datos. El elemento de *servicio* en *IoT* se refiere a todas las aplicaciones encargadas de la actuación con base a los datos generados. Para el uso del trabajo presente, esta capa representa a los dispositivos actuadores que regulan el confort e higiene en el hogar, como el aire acondicionado y el extractor de aire para mejorar las condiciones ambientales. Finalmente, el elemento de *semántica en IoT* se refiere a casos relacionados con la habilidad de extraer datos de manera inteligente, así como filtrar los datos y analizarlos con algoritmos, donde en el trabajo presente se emplea la técnica de predicción del confort térmico mostrada en la norma *ISO 7730* [4], por lo que la semántica y filtrado de datos se logra por medio del uso de una base de datos.

Algunos trabajos funcionales de autores relacionados con el uso de *IoT*, son mostrados en [17], donde ellos desarrollaron un dispositivo de monitoreo de calidad de aire en tiempo real, con ayuda del *IoT*, quienes concluyeron que se pueden usar sensores de bajo costo que no se pueden conectar a Internet por sí solos, pero que con ayuda de ciertos microcontroladores como el modelo comercial *ESP32*, se puede lograr un nodo de emisión de datos muy accesible. Los autores monitorearon la calidad del aire con un sensor de gas *MQ135* y un sensor de temperatura *DHT11*, como el usado en [14], sin embargo, solo son sistemas de instrumentación, por lo que es pertinente saber el modo de regular las variables, usando actuadores de igual bajo costo, siendo esto último también de interés particular para el trabajo presente.

Como se detalla en la Figura 8(d), la capa de aplicación es la encargada de proveer los servicios de comunicación requeridos y presentar los datos obtenidos por la capa de objetos a la red de Internet. Es de gran importancia agregar que los datos deben tener una forma estructurada para su transmisión, ya que ellos pasan por un proceso de *codificación*, *decodificación*, *compresión* y *homologación* [15]. Estos datos pueden representar parámetros de medición, órdenes de los usuarios y cualquier otro tipo de mensaje representado por caracteres de texto o numéricos, es por esto que el mensaje debe cifrarse por medio de ciertos formatos, tal es el caso del *UTF-8*, que por sus siglas en inglés significa *formato de transformación a unicódigo de 8 bits*, esto quiere decir que el mensaje pasa por un proceso de *encriptación*, esto significa que el mensaje no puede ser leído explícitamente por dispositivos y usuarios terceros, ya que es ilegible por motivos de seguridad, por lo que si un dispositivo no participante en la red *IoT* tuviera acceso al flujo de datos, este solo podría obtener mensajes compuestos por símbolos, letras o números generados aparentemente de manera aleatoria.

Existen diversas herramientas para trabajar en la capa de aplicación de *IoT*, una de ellas es el uso del *protocolo de mensajería MQTT*, que significa *transporte de telemetría de cola de mensajes*, el cual es un servicio para transmitir datos por medio del protocolo *TCP* por *IP*, que por sus siglas en inglés significa *protocolo de control de transmisión por protocolos de internet*. La función del protocolo *TCP* sirve para que los dispositivos participantes en la red *IoT* puedan comunicarse entre sí. Por otro lado, el flujo de la información cifrada lleva otra capa de seguridad propia de la comunicación por *MQTT*, esta es la capa *TLS*, que por sus siglas en inglés significa *capa de seguridad de transporte*, esto es una capa especializada en proteger la privacidad de los datos transferidos, así como establecer una conexión segura entre los dispositivos por métodos de autenticación y así evitar que terceros malintencionados puedan interceptar las vías de comunicación. La Figura 10 hace una alusión a lo mencionado anteriormente, donde se puede suponer la interacción entre dos dispositivos, el *dispositivo emisor A* y el *dispositivo receptor B*, donde *A* pretende enviar un mensaje hacia *B*, donde primeramente el protocolo *TCP* controla el flujo de este mensaje por medio de un conducto asegurado.

Otro concepto importante en el uso de *MQTT* es el concepto de *bróker*, esta palabra tiene un significado que podría ser *intermediario*, el cual se encarga de direccionar los flujos de datos como en la Figura 10, sin embargo, no está explícitamente representado, ya que el dispositivo *A* y *B* no pueden comunicarse directamente, sino que el mensaje del emisor es enviado al *bróker*, el cual se encarga de las capas de seguridad, así que del *bróker* salen los mensajes para ser recibidos por el receptor. Todo *bróker* debe tener explícitamente un *puerto de recepción y emisión de datos*, el cual está numerado por cifras de cuatro dígitos para identificarlos. En la Figura 10 el dispositivo *A* emite su mensaje hacia el puerto número 1883 del *bróker*, mientras que el dispositivo *B* recibe datos de salida del mismo puerto. El objetivo de los puertos existe por motivos de seguridad y algunos de

ellos están cerrados, a menos que se programe su apertura, sin embargo en *MQTT* es el puerto 1883 quien está abierto por defecto, por lo que es importante recalcar que queda a decisión del programador usar o no los puertos necesarios a sus criterios de seguridad, que para motivos del trabajo presente y dada la naturaleza del tipo de datos a compartir, el uso de otros puertos que no sean el 1883 para el *bróker MQTT* es factible.

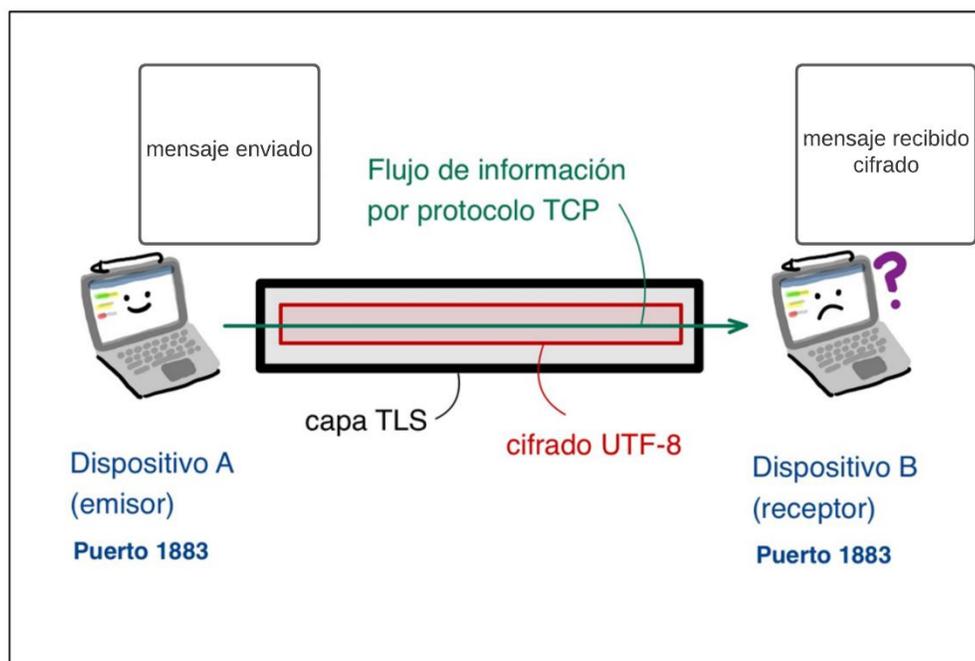


Figura 10. Capas de seguridad en el flujo de la información en MQTT.

Los fundamentos de uso para el protocolo de capa de aplicación por *MQTT* se basan en el concepto de *publicación* y *suscripción*, conocidos en inglés como *publish* y *subscribe*, que, en términos generales, la publicación al *bróker* permite enviar mensajes a otros dispositivos, y para que dichos dispositivos puedan recibir el mensaje, deben estar conectados al *bróker*, este modo es

el funcionamiento en *MQTT*. A continuación, en la Figura 11 se presentan las ventajas de usar este protocolo de comunicación.

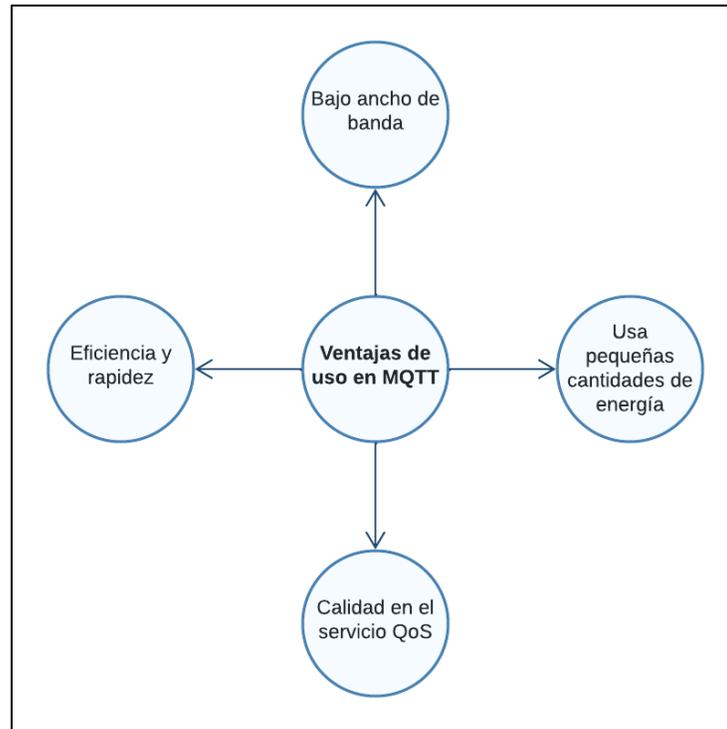


Figura 11. Ventajas de usar el protocolo *MQTT*.

Analizando la Figura 11, se puede observar que una de las ventajas es referente a la calidad en el servicio tipo *QoS*, que significa *calidad en el servicio*, esto es un conjunto de técnicas y herramientas que utilizan los ingenieros en redes de comunicación, para garantizar calidad y eficiencia en la transmisión de los datos, por lo que la construcción de *MQTT* ofrece esto.

Aclarado lo anterior, se puede representar la forma de comunicación entre clientes y servidor, esto a través de *MQTT*, la Figura 12 representa al servidor como aquella instancia donde se encuentra alojado el *bróker* interactuado con otros clientes, los cuales pueden ser celulares,

ordenadores, sensores o actuadores, donde cualquier objeto conectable a Internet puede interactuar con otros clientes.

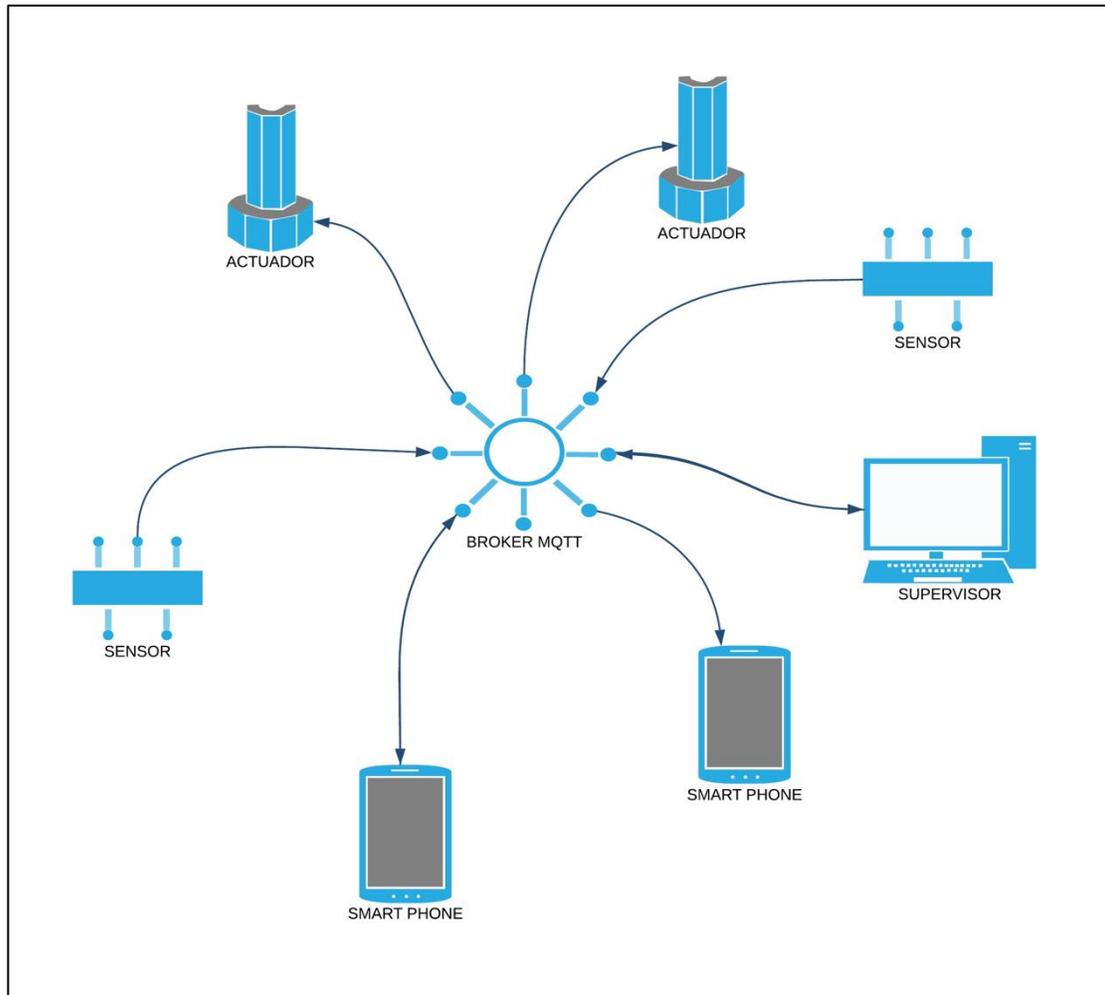


Figura 12. Red de comunicación simplificada a través de un bróker MQTT.

Es primordial diseñar una correcta *topología de red*, esto es el modo de interacción entre los dispositivos, a quienes se les puede denominar también como *nodos*, ya que los flujos pueden ser entrantes, salientes o ambos, en referencia a cada dispositivo participante, como se observa en la Figura 10, así como definir cuál será la función y el motivo de su participación en la red de *IoT*. Por otro lado, por motivos de seguridad, el *bróker* puede encontrarse por defecto inhabilitado a la

publicación de mensajes de los clientes, así que puede ser necesario que el programador tenga que habilitarlo, por otro lado y dada la flexibilidad de *MQTT*, cualquier cliente puede emitir mensajes, órdenes, lecturas o cualquier otro tipo de información relevante, casi al mismo tiempo, por lo que es recomendable establecer funcionalidades y jerarquías para una comunicación más entendible y estructurada.

Una *base de datos relacional*, de acuerdo con [18], se define como un modelo de datos para representar su descripción, semántica y restricciones de consistencia a través de las relaciones entre conjuntos de datos. Cada conjunto de datos está representado por medio de una tabla, donde cada tabla tiene un nombre específico que la define y se les denominan *campos*. Cada columna de la tabla o *campo* está compuesta por un conjunto de filas denominados *registros*, los cuales contienen las instancias de información relevantes, tal es el caso de datos de temperatura, humedad o variables de calidad del aire, lo cual es de interés particular para el trabajo presente. Por otro lado, una *base de datos relacional* es definida por toda aquella capaz de efectuar las operaciones fundamentales conocidas como *CRUD*, que por sus siglas en inglés significan *crear, leer, actualizar y borrar* registros. Para el uso particular de *IoT*, las operaciones de actualización y borrado de registros no son de interés particular, ya que modificar y anular datos no tiene algún objetivo de uso en cuanto a que son los sensores y actuadores quienes ofrecen sus lecturas de medición y estados de actuación, así como hacer énfasis en que es muy importante usar la operación de borrado de una base de datos siempre que se haga con responsabilidad y cuando sea necesario, pues los registros se eliminan por completo y sin ningún método de recuperación, a menos que se programe para casos accidentales.

Por otro lado, *MySQL*, así como otros lenguajes trabajan con una estructura básica de consultas de los datos, esto es de interés en las capas de gestión de *servicios, de aplicación y de negocios*,

reiterando que esta última capa para *IoT* en domótica, se refiere al uso y análisis de los datos. Las consultas se nombran como *select*, *from* y *where*, que sirven para seleccionar tablas de interés particular, desde una cláusula y donde existan restricciones para filtrar los datos. Esto se requiere para operaciones básicas de unión o intersección de datos, así como cualquier otra operación basada en *álgebra de conjuntos*, porque en un sentido matemático, las tablas son conjuntos de elementos, ya que la relación entre conjuntos es fundamental y es motivo del nombre de *bases de datos relacionales*. Por otro lado, la programación en *MySQL* usa un término denominado *query*, que por su traducción significa *petición*, estas peticiones están basadas en las operaciones *CRUD*, y se podría definir como el tipo de comunicación para solicitar manipular los registros de la base de datos relacional.

Finalmente, es importante esclarecer la participación del tipo de servidor que alojará el gestor de base de datos, esto es a través de un servidor local o en la nube. Un servidor local es un ordenador o equipo de cómputo del que se puede acceder directamente y que se encuentra físicamente presente en la red de área local, para brindar servicios a los clientes conectados en la misma red, mientras que un servidor en la nube es aquel ordenador remoto, en una red pública, es decir Internet, para brindar de igual manera, servicios a cualquier dispositivo conectado a Internet. Dicho servicio puede ser el acceso a la base de datos, por lo que para el trabajo presente, los datos se alojarán en un servidor local, debido a que en caso de desconexión o la imposibilidad de conectarse a Internet, los sensores podrán seguir insertando datos de estado, caso contrario con un servidor en la nube, donde es necesaria la conexión permanente de los dispositivos, así como también aislar la base de datos del dominio público por motivos de seguridad.

Capítulo 2: Metodología para la construcción del sistema de instrumentación y control

Para utilidad de este trabajo presente, la metodología para el diseño y obtención de resultados, así como las pruebas del correcto funcionamiento de éste, consiste en integrar un sistema de instrumentación y control dentro del hogar, manipulando las variables de confort por medio de un actuador acondicionador de aire portátil y un extractor de aire para eliminar las impurezas en el ambiente, dentro de un diseño red de *IoT* con sensores. En las secciones siguientes de este capítulo se describe esto anteriormente mencionado, consistiendo primero en identificar al hogar de pruebas a implementarle el sistema, luego establecer las herramientas de *hardware* y *software* para el diseño de la red de *IoT*, establecer un método de análisis con los datos obtenidos por medio de una algoritmo de predicción de estado de confort térmico de los usuarios, una planeación de diseño para los algoritmos de control, otra para la instrumentación y finalmente una más para la red de comunicación en *IoT* por medio de sus capas de arquitectura, lo anterior descrito tiene el objetivo de ser implementado en el Capítulo 3.

2.1 Identificación de las características del hogar de pruebas experimentales

Es preciso comprender las características del hogar de pruebas, con la finalidad de conocer su geometría y tipo de material de construcción, así como el número de miembros confinados para

una correcta selección de los sistemas de acondicionamiento y purificación de aire para que así los datos experimentales puedan ser representativos en estos tipos de espacios. El *Instituto Nacional de Estadística y Geografía (INEGI)* tiene censos [19] [20], cuyo objetivo es identificar las características promedio de los hogares en México, tal como se muestra en la Tabla 1, donde esta información fue de gran ayuda para poder seleccionar los actuadores más adecuados. La superficie efectiva de cada una de las tres habitaciones donde se harán las pruebas (tres dormitorios), se encuentran dentro del promedio mostrado en la tabla, por lo que se supondrá que los datos experimentales a obtener son propios de un hogar mexicano representativo.

Tabla 1. Condiciones del hogar mexicano que puede cubrir el diseño del sistema de control y monitoreo, [19] [20].

Condición más representativa del hogar mexicano	Material o rango de alcance
Material de construcción en techos	Losa de concreto o bovedilla
Material de construcción en suelo	Piso con recubierta cerámica o solo concreto
Material de construcción en paredes	Pared de concreto o ladrillo
Superficie de vivienda	55 a 100 m ²
Habitantes por vivienda	Hasta 4 miembros

2.2 Metodología para el control automático del sistema

La metodología de control del sistema, consiste en manipular las variables de confort por medio de un método de control por eventos discretos en lazo cerrado. El vector de salida de este controlador se muestra en la expresión 1.

$$\mathbf{y}_N = \begin{bmatrix} T_N \\ PM25_N \\ PPMCO_N \end{bmatrix} \quad (1)$$

En donde \mathbf{y}_N es el vector de salida del sistema de control y que contiene todos los parámetros de medición, N es una variable de identificación para el *número de habitación de la casa* (serán 3 habitaciones de pruebas), T_N es la temperatura operativa de la habitación en [°C], $PM25_N$ es la medición de partículas contaminantes suspendidas en el aire en [$\frac{\mu g}{m^3}$] y $PPMCO_N$ son las partes por millón contaminantes de gas monóxido de carbono CO en [ppm], estos últimos dos parámetros debido a fuentes como humo de cigarro o una mala ventilación dentro de las habitaciones.

Por otro lado, se deben manejar otras variables auxiliares que son H_N , que es la humedad relativa de la habitación en [%], y L_N es la luminosidad del interior en [lux], pero no son parámetros de control para el diseño, en el caso de la luminosidad se requiere como variable informativa para el usuario e identificar en los experimentos de funcionamiento si podría ser capaz de inducir ruido térmico que afecte el desempeño del sistema de control, esto debido a la transferencia de calor por radiación que emiten algunos sistemas de iluminación incandescentes. Para el caso de la humedad relativa, de igual manera es influyente en el desempeño en los actuadores de aire acondicionado. El modo en el que se ejecuta el sistema de control se logra a través de un monitoreo y transducción de los sensores en una retroalimentación de lazo cerrado hacia el controlador.

En cuanto a los elementos finales de control, se requiere del *vector de estados* \mathbf{w}_N que es especificado por la expresión 2.

$$\mathbf{w}_N = \begin{bmatrix} A_{N1} \\ A_{N2} \end{bmatrix} \quad (2)$$

Donde A_{N1} y A_{N2} son los actuadores de control de Aire Acondicionado y Extractor de Aire respectivamente, en la habitación de pruebas de número N . Esto quiere decir que sus valores solo son de tipo lógico y solo pueden adoptar valores de 1 o 0, es decir la activación o desactivación respectivamente.

Con ayuda de las expresiones 1 y 2 se formula una siguiente expresión que es de utilidad para el transcurso de los próximos capítulos, y solo tiene por objetivo una identificación con mejor claridad para la red de comunicación por internet. La expresión 3 lo representa.

$$\mathbf{C}_N = \begin{bmatrix} \mathbf{y}_N \\ \mathbf{w}_N \\ H_N \\ L_N \end{bmatrix} = \begin{bmatrix} T_N \\ PM25_N \\ PPMCO_N \\ A_{N1} \\ A_{N2} \\ H_N \\ L_N \end{bmatrix} \quad (3)$$

Donde \mathbf{C}_N será considerado como el *vector de parámetros de información y retroalimentación*, donde sus componentes son los vectores \mathbf{y}_N y \mathbf{w}_N , así como los parámetros informativos H_N y L_N .

Por otro lado, el controlador requiere de un conjunto de entradas las cuales son los parámetros deseados y asignados por el usuario, esto se representa por medio de la expresión 4.

$$\mathbf{x}_N = \begin{bmatrix} T_{SN} \\ PM2.5_{SN} \\ PPMCO_{SN} \end{bmatrix} \quad (4)$$

En donde \mathbf{x}_N es el *vector de parámetros deseados* especificados por el usuario, T_{SN} es la temperatura deseada en la habitación, $PM2.5_{SN}$ es el límite deseado de permisibilidad de partículas en el aire y $PPMCO_{SN}$ es el límite deseado de permisibilidad de concentración de *CO*. Los parámetros de \mathbf{x}_N hacen alusión a los valores deseados de la habitación N .

Por lo tanto, los valores de \mathbf{x}_N y \mathbf{y}_N deberían tratar de *equilibrarse* para afirmar que el controlador funciona correctamente, por medio de las lecturas obtenidas en los experimentos de funcionamiento. La Figura 13 muestra esta regla de correspondencia.

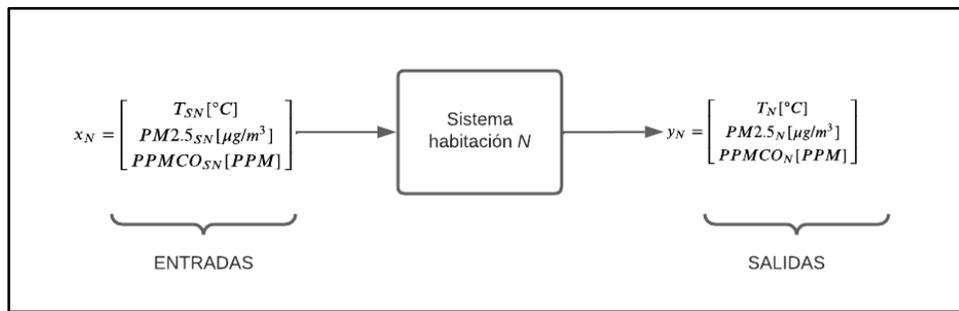


Figura 13. Regla de correspondencia entre los parámetros de entrada y de salida del sistema.

Para controlar los valores del vector \mathbf{y}_N en función de las entradas de \mathbf{x}_N pertenecientes a las expresiones 1 y 4 respectivamente, se emplea el diagrama mostrado en la Figura 14, donde existe una retroalimentación en lazo cerrado con ayuda de cada uno de los sensores empleados y correspondientes a cada uno de los parámetros de \mathbf{y}_N , dichos sensores se establecen en la sección 2.3.1.

2.2.1 Selección y uso de actuadores

Enfriar la temperatura de una habitación se puede lograr con sistemas que empleen el principio de funcionamiento por *enfriamiento evaporativo* [21], el cual es de muy bajo costo y no requieren hacer modificaciones de construcción y albañilería en el hogar. Con este método, el aire caliente y seco se acondiciona con agua líquida y fría, donde una fracción del agua es evaporada para absorber calor del flujo de aire, esto provocará que la temperatura del flujo del aire T_N disminuya y su humedad H_N aumente, esto ocurre en la trayectoria desde la etapa 1 hacia la etapa 2, como se ilustra en la Figura 15. En esta carta psicrométrica existe una etapa límite 2', que especifica que el aire saliente del enfriador será saturado, es decir, que es la temperatura más baja que puede alcanzarse con este proceso, y que en los resultados experimentales del presente trabajo, será de interés particular, pues al ajustar la temperatura de consigna T_{SN} , ésta deberá considerar el valor límite permisible por el actuador, ya que si la temperatura deseada es muy baja, el actuador solo podrá ofrecer su temperatura de saturación como mínimo, y en el controlador mostrado en la Tabla 4, dicha temperatura ocurrirá en un tiempo de asentamiento, donde gráficamente se observará que será la temperatura de convergencia, correspondiente a la temperatura de saturación. En la carta psicrométrica pueden visualizarse dos variables T_{bh} y h , que se refieren a la temperatura de bulbo húmedo y entalpía de la habitación respectivamente, y dada la naturaleza adiabática del sistema, se consideran de valor constante dentro de la habitación.

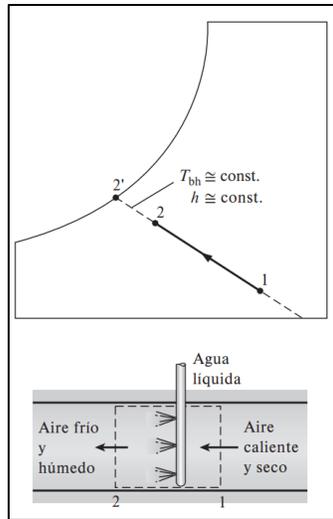


Figura 15. Carta psicométrica del enfriamiento evaporativo, [21].

El modelo comercial adquirido del actuador es un *Artic Air Ultra 18008*, cuyo funcionamiento es por enfriamiento evaporativo, tal actuador se muestra en la Figura 16.



Figura 16. Actuador comercial de acondicionamiento de aire por enfriamiento evaporativo marca Artic Air Ultra 18008.

El esquemático de funcionamiento se ilustra en la Figura 17.

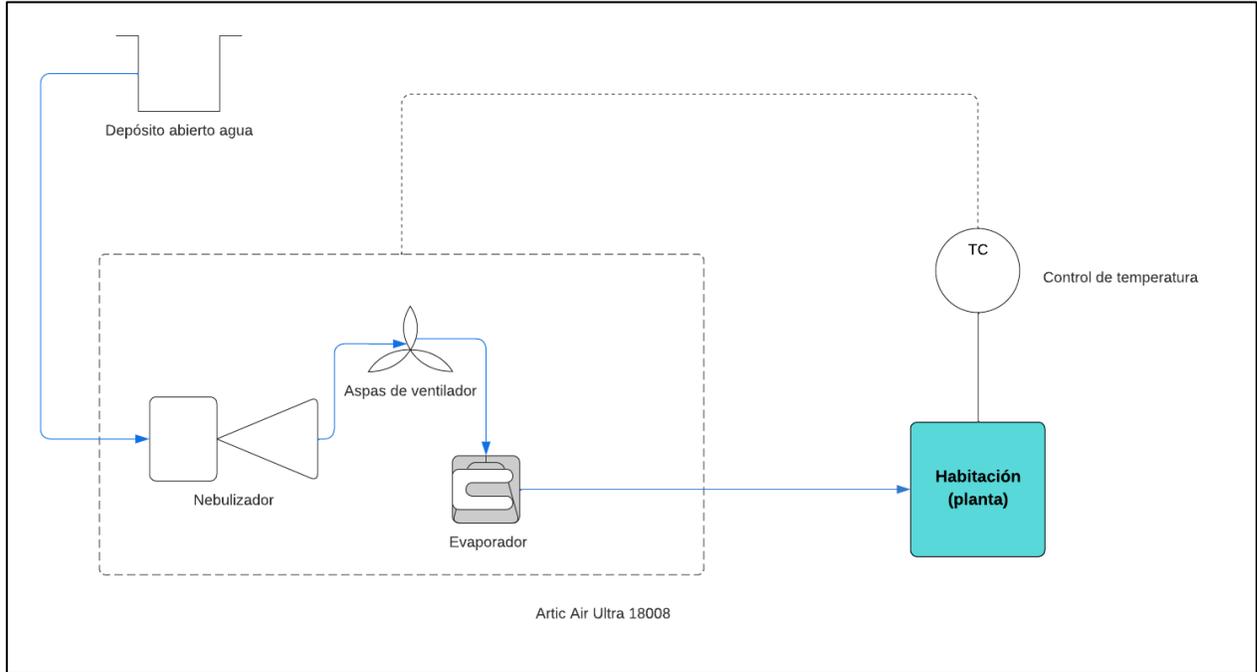


Figura 17. Diagrama de funcionamiento del actuador Artic Air Ultra 18008 en conjunto con el controlador diseñado.

En cuanto a la purificación del aire, puede lograrse extrayendo las partículas contaminantes y concentración de gases nocivos alojados en el interior de la habitación hacia el exterior con actuadores de extracción. El principio de funcionamiento se basa en la geometría de las aspas de ventiladores que al girar, el flujo de aire es succionado de manera axial y en dirección del eje de giro, por lo que la fuerza de empuje moviliza el aire hacia el interior de una tubería sobrepuesta, empotrada en la pared o en el techo del hogar, por lo que su uso es muy útil en cocinas, baños, habitaciones, sótanos, o cualquier otro sitio de espacio cerrado que requiera purificar y aspirar las impurezas del aire, tal es el caso de las partículas $PM\ 2.5$ y las partes por millón de gas monóxido de carbono CO . El actuador comercial utilizado para el presente trabajo es un extractor de aire axial, marca y modelo *Igoto Swiss EX4-16W*, ver Figura 18.

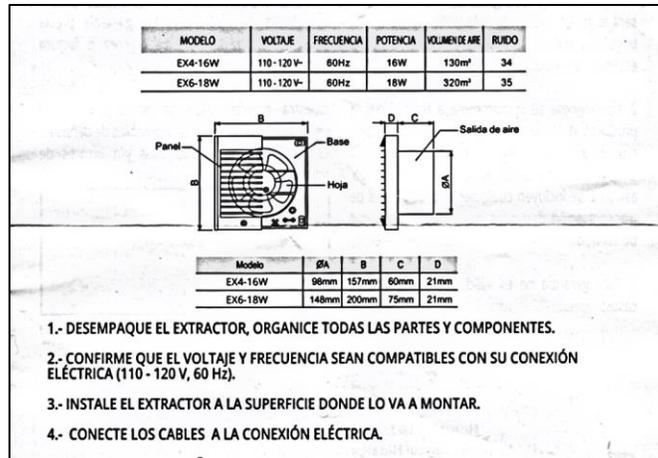


Figura 18. Especificaciones técnicas de extractor Igoto Swiss EX4-16W.

El fabricante de este extractor, indica que es recomendable empotrarlo en la pared, sin embargo, puede usarse de manera sobrepuesta para evitar modificaciones en el hogar, por lo que para motivos del trabajo presente, en su lugar se utilizó un conducto flexible de aluminio de 4''x 2 m de longitud, tal como se muestra la Figura 19.



Figura 19. Extractor de aire Igoto Swiss EX4-16W y conexión con tubo flexible de 3'' x 2m.

Por otro lado, el funcionamiento del sistema de extracción se puede representar por medio de la Figura 20.

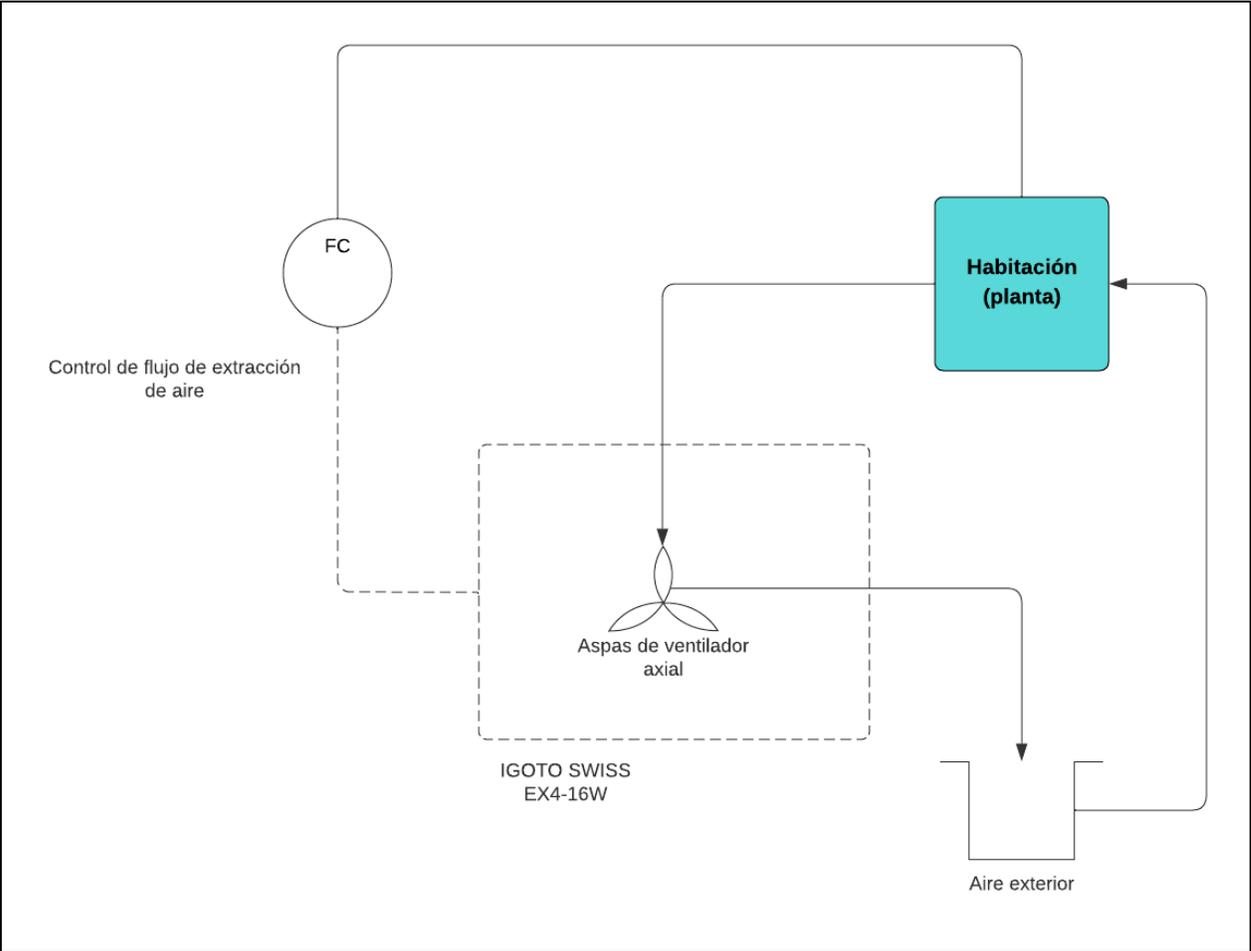


Figura 20. Diagrama de funcionamiento del actuador Igoto Swiss EX4-16W en conjunto con el controlador diseñado.

La Tabla 2 representa un resumen con la información necesaria para el uso del aire acondicionado y purificador en conjunto con una tarjeta ESP32 como microcontrolador.

Tabla 2. Resumen de características de actuadores a utilizar para el sistema de control [22],

ACTUADOR	TIPO	PRINCIPIO DE FUNCIONAMIENTO	VARIABLE BOOLEANA DEL ACTUADOR	SUMINISTRO DE TENSIÓN	RELÉ
Artic Air Ultra 18008	Eléctrico	Enfriamiento evaporativo	A_{N1}	110-120 V AC	5V DC/ 125 V AC
Igoto Swiss EX4-16W	Eléctrico	Extracción axial	A_{N2}	5-12V DC	5V DC/ 125 V AC

2.2.2 Etapas del control automático

El control automático se puede lograr empleando el uso de *brechas*, las cuales son regiones en un gráfico donde se alojan los puntos de medición correspondientes a los valores de T_N , $PM2.5_N$ y $PPMCO_N$, con ayuda de los actuadores seleccionados anteriormente.

Primero el usuario deberá definir una consigna deseada escogiendo los valores de x_N . Para el valor de confort de temperatura deseada T_{SN} , a éste le será asignado un par de límites permisibles de temperaturas T_A y T_B , inferior y superior respectivamente, en una distancia de dos unidades (ancho de la brecha, ver Figura 21). La variable auxiliar de humedad relativa debe considerarse también, y sus límites inferior y superior son H_A y H_B respectivamente, dicho valor no puede controlarse, pero puede considerarse un valor de confort por humedad de $H_{SN} \approx 50\%$. Entonces, todos aquellos puntos de medición que se encuentren fuera de los límites antes mencionados de confort, se encontrarán en la denominada *región de desconfort*, tal como se muestra en la Figura 21.

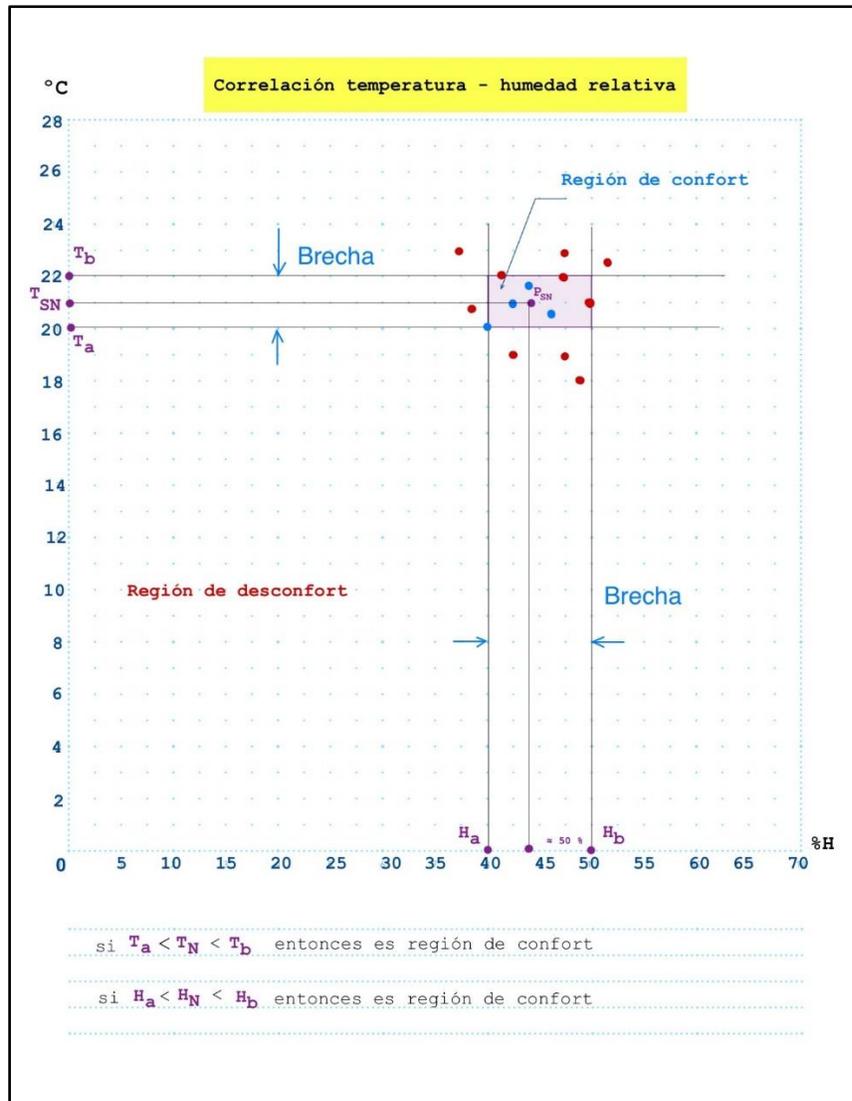


Figura 21. Correlación de temperatura y humedad relativa del cuarto.

Para lograr que todos los puntos se encuentren dentro de la región de confort, se propone la metodología mostrada en la Tabla 3, que servirá para la construcción del algoritmo de control de temperatura dentro de la habitación.

Tabla 3. Etapas de control de temperatura T_N de habitación.

Etapa	Acotación	Estado del actuador A_{N1}	Descripción
-------	-----------	------------------------------	-------------

0	$T_a < T_N < T_b$ O bien $T_a > T_N < T_b$ O bien $T_a < T_N > T_b$	$A_{N1} = 0$	Esta es la etapa inicial de la <i>planta</i> . El controlador está preparado para que la condición inicial de la habitación se encuentre dentro de la brecha, por debajo de la brecha o por encima de ella. Sea cual sea el caso, el actuador inicialmente está siempre apagado.
1, opción a	$T_a < T_N > T_b$	$A_{N1} = 1$	Si la condición inicial es que la temperatura de habitación se encuentra por encima de la brecha, encender y salta a la etapa 2. Pero si la condición viene de la etapa 4, entonces sigue a la etapa 1, opción b).
1, opción b	$T_a < T_N < T_b$	$A_{N1} = 1$	Si la condición inicial es que la temperatura de habitación se encuentra dentro de la brecha, encender. Pero si la condición viene de la etapa 1, opción a), sigue a la etapa 1, opción c).
1, opción c	$T_a > T_N < T_b$	$A_{N1} = 0$	Si la condición inicial es que la temperatura de habitación se encuentra por debajo de la brecha, mantener apagado por ahorro energético.
2	$T_a > T_N < T_b$	$A_{N1} = 0$	Se asegura que la temperatura de habitación se encuentre por debajo de la brecha. De acuerdo con la resolución del sensor, esta pequeña diferencia podría ser imperceptible en el orden de las décimas de grado centígrado, mantener el apagado.

3	$T_a < T_N < T_b$	$A_{N1} = 0$	Eventualmente la temperatura de la habitación tendrá que aumentar naturalmente debido al fenómeno de transferencia de calor por radiación y conducción, principalmente, ocurre en la puesta máxima de la posición del sol. Se mantiene apagado por eficiencia energética.
4	$T_a < T_N > T_b$	$A_{N1} = 1$	Eventualmente la temperatura de la habitación ya habrá recorrido toda la brecha por calentamiento, a esta etapa el actuador deberá encenderse, repitiendo el proceso pero comenzando en la etapa 1 opción a) y recorriendo también las opciones b) y c).

De manera similar, para gobernar las variables de calidad del aire $PPMCO_N$ y $PM2.5_N$, la región de confort por calidad de aire se muestra en la Figura 22, donde se observa un ancho de brechas de 2 unidades para la permisibilidad de error. A diferencia de la manipulación de la temperatura, para este control de calidad del aire se debe adicionar una *operación lógica de conjunción* de tipo *negada*, donde p, q son variables *booleanas* y la operación de conjunción es explícitamente el valor del actuador del extractor de aire A_{N2} , tal como se muestra en la ecuación 5 y en la Tabla 4.

$$A_{N2} = \overline{p \wedge q} \tag{5}$$

Esto es requerido porque el controlador pretende manipular las variables $PM2.5_N$ y $PPMCO_N$ a la vez. La propuesta de control se esclarece en la Tabla 5.

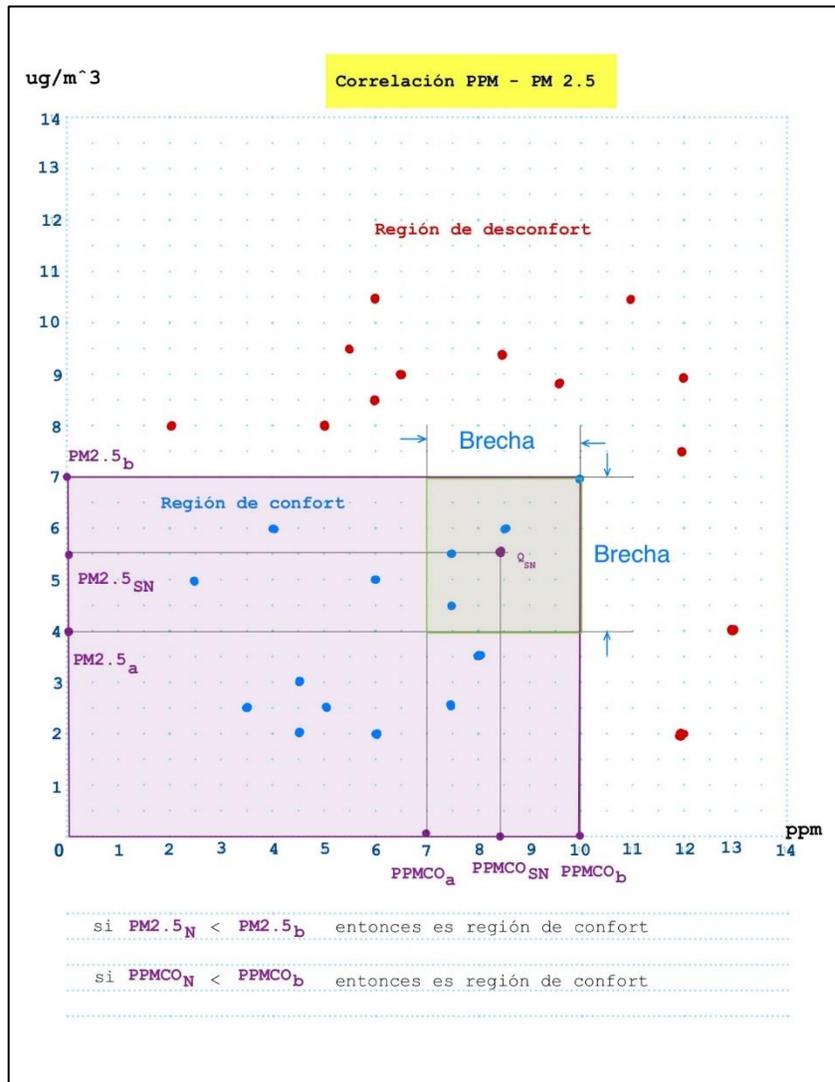


Figura 22. Correlación de partes por millón de monóxido de carbono y partículas PM 2.5.

Tabla 4. Tabla de verdad para el funcionamiento del actuador del purificador de aire.

Partículas PM2.5 [ug/m^3]	Partes por millón de monóxido de carbono [ppm]	¿Encender el actuador A_{N2} ?
$PM2.5_N < PM2.5_b$		$\overline{p \wedge q}$

p	$PPMCO_N < PPMCO_b$	
	q	
NO, 0	NO, 0	SÍ, 1
NO, 0	SÍ, 1	SÍ, 1
SÍ, 1	NO, 0	SÍ, 1
SÍ, 1	SÍ, 1	NO, 0

Tabla 5. Etapas de control de partículas $PM_{2.5_N}$ y de monóxido de carbono $PPMCO_N$ de la habitación.

Etapa	Acotación	Estado del actuador A_{N1}	Descripción
0	$PM_{2.5_a} < PM_{2.5_N} < PM_{2.5_b}$ O bien $PM_{2.5_a} > PM_{2.5_N} < PM_{2.5_b}$ O bien $PM_{2.5_a} < PM_{2.5_N} > PM_{2.5_b}$ O bien $PPMCO_a < PPMCO_N < PPMCO_b$ O bien $PPMCO_a > PPMCO_N < PPMCO_b$ O bien $PPMCO_a < PPMCO_N > PPMCO_b$	$A_{N2} = 0$	Esta es la etapa inicial de la <i>planta</i> . El controlador está preparado para que la condición inicial de la habitación se encuentre dentro de la brecha, por debajo de la brecha o por encima de ella. Sea cual sea el caso, el actuador inicialmente está siempre apagado.
1, opción a	$PM_{2.5_a} < PM_{2.5_N} > PM_{2.5_b}$ Y además $PPMCO_a < PPMCO_N > PPMCO_b$	$A_{N2} = 1$	Si la condición inicial es que las partículas $PM_{2.5}$ y monóxido de carbono de habitación se encuentren por encima de las brechas, encender y salta a la etapa 2. Pero si

			la condición viene de la etapa 4, entonces sigue a la etapa 1, opción b).
1, opción b	$PM2.5_a < PM2.5_N < PM2.5_b$ <p style="text-align: center;">Y además</p> $PPMCO_a < PPMCO_N < PPMCO_b$	$A_{N2} = 1$	Si la condición inicial es que las partículas PM 2.5 y monóxido de carbono de habitación se encuentren por encima de las brechas de habitación, encender. Pero si la condición viene de la etapa 1, opción a), sigue a la etapa 1, opción c).
1, opción c	$PM2.5_a > PM2.5_N < PM2.5_b$ <p style="text-align: center;">Y además</p> $PPMCO_a > PPMCO_N < PPMCO_b$	$A_{N2} = 0$	Si la condición inicial es que las partículas PM 2.5 y monóxido de carbono de habitación se encuentren por encima de las brechas, mantener apagado por ahorro energético.
2	$PM2.5_a > PM2.5_N < PM2.5_b$ <p style="text-align: center;">Y además</p> $PPMCO_a > PPMCO_N < PPMCO_b$	$A_{N2} = 0$	Se asegura que las partículas 2.5 y monóxido de carbono de habitación se encuentren por debajo de las brechas. De acuerdo con la resolución del sensor, esta pequeña

			diferencia podría ser imperceptible en el orden de las décimas de grado centígrado, mantener el apagado.
3	$PM2.5_a < PM2.5_N < PM2.5_b$ <p>Y además</p> $PPMCO_a < PPMCO_N < PPMCO_b$	$A_{N2} = 0$	Eventualmente las partículas PM2.5 y monóxido de carbono de la habitación tendrán que aumentar, aunque no de manera inmediata, debido a perturbaciones como fuentes de emisión como el humo de cigarro. Se mantiene apagado por eficiencia energética.
4	$PM2.5_a < PM2.5_N > PM2.5_b$ <p>Y además</p> $PPMCO_a < PPMCO_N > PPMCO_b$	$A_{N2} = 1$	Eventualmente las partículas PM2.5 y monóxido de carbono de la habitación ya habrán recorrido todas las brechas por perturbaciones, a esta etapa el actuador deberá encenderse, repitiendo el proceso pero comenzando en la etapa 1 opción a) y

			recorriendo también las opciones b) y c).
--	--	--	---

2.3 Metodología para la instrumentación del sistema

Las etapas del sistema de instrumentación se muestran en la estructura de la Figura 23.

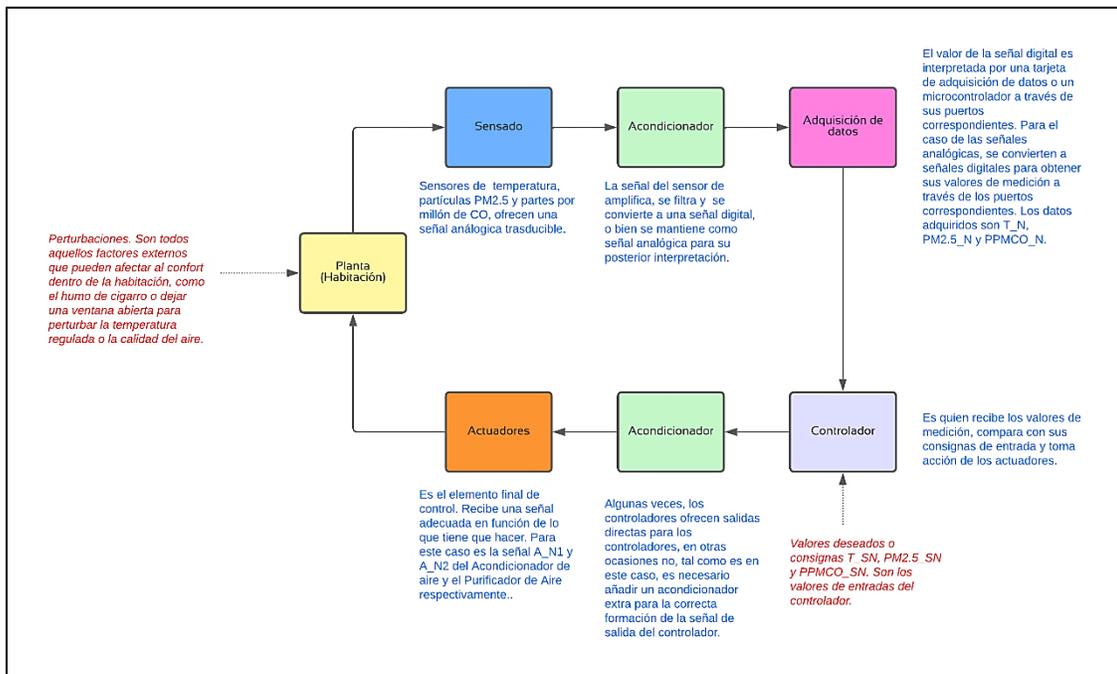


Figura 23. Estructura del sistema de control y medición para la habitación.

Dado que las entradas del controlador son las consignas deseadas por el usuario T_{SN} , $PM2.5_{SN}$ y $PPMCO_{SN}$, estas se comparan con las lecturas T_N , $PM2.5_N$ y $PPMCO_N$ ofrecidas por los sensores, esto significa que es de gran importancia tener todas las medidas correctas, puesto que

no ser de este modo, el controlador no podría lograr su cometido. Dichas entradas se representan por medio de la expresión 4.

2.3.1 Selección y uso de sensores

Para poder medir la temperatura de la habitación en tiempo real, es de interés el uso del termistor *NTC* modelo comercial *DHT11*. Al tipo de temperatura medida se le denominará *temperatura operativa*, ya que es la combinación de la temperatura producida por los fenómenos de transferencia de calor por conducción y radiación, siendo las fuentes el material de construcción del inmueble y las personas confinadas radiando calor respectivamente [23] [4], ver tabla 6.

Tabla 6. Condiciones de temperatura operativa, [23].

AMBIENTE DE TEMPERATURA OPERATIVA
Actividad M_N
Vestimenta I_{clo}
Presión de vapor de agua P_v
Velocidad del aire $V_1 < 0,2\text{m/s}$
Temperatura seca T_o
Temperatura radiante media $T_{rm} = T_o$

Velocidad del aire $v=0,2\text{ m/s}$
Humedad relativa ϕ

En la Tabla 6, la temperatura operativa T_o se refiere a la variable T_N (como identificación para el trabajo presente), y esta es igual a la temperatura radiante media T_{rm} , tal como se establece en dicha tabla. Por otro lado, dentro de este mismo encapsulado del sensor *DHT11*, se tiene un

segundo sensor de medición de humedad relativa, la cual es una variable de entrada para poder predecir el confort térmico, técnica explicada en la próxima sección 2.4.

Como se hace mención en la sección 1.2.2, las partículas contaminantes *PM2.5* y partes por millón de CO suspendidos en el aire pueden medirse con los sensores de modelo comercial *PMSA003I* y *MQ7* respectivamente. El sensor *PMSA003I* requiere de un algoritmo de acceso a su memoria interna, para solicitar la detección de las partículas *PM2.5*, el planteamiento de dicho algoritmo se muestra en el diagrama de flujo en la Figura 25, y sirve para acceder a los registros correspondientes de memoria del sensor, tal como se representa en la Figura 24.

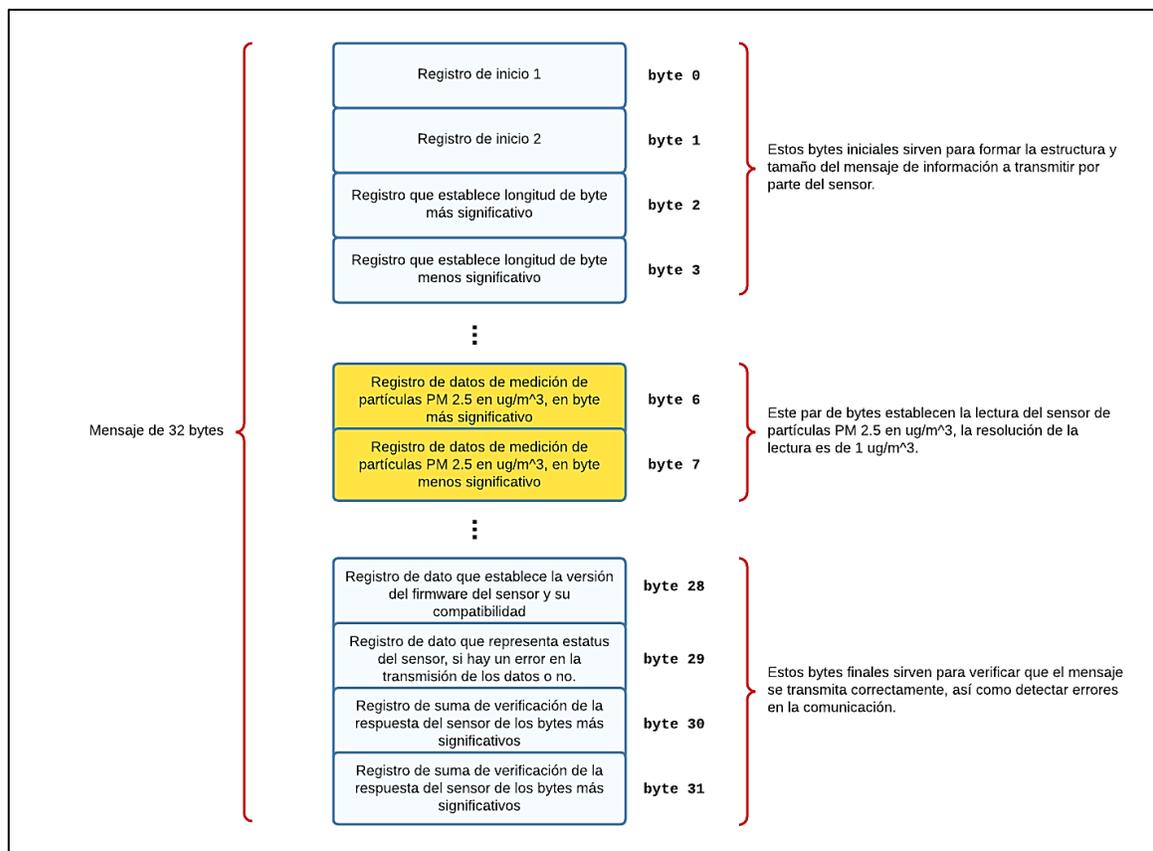


Figura 24. Registros de memoria del sensor PMSA003I.

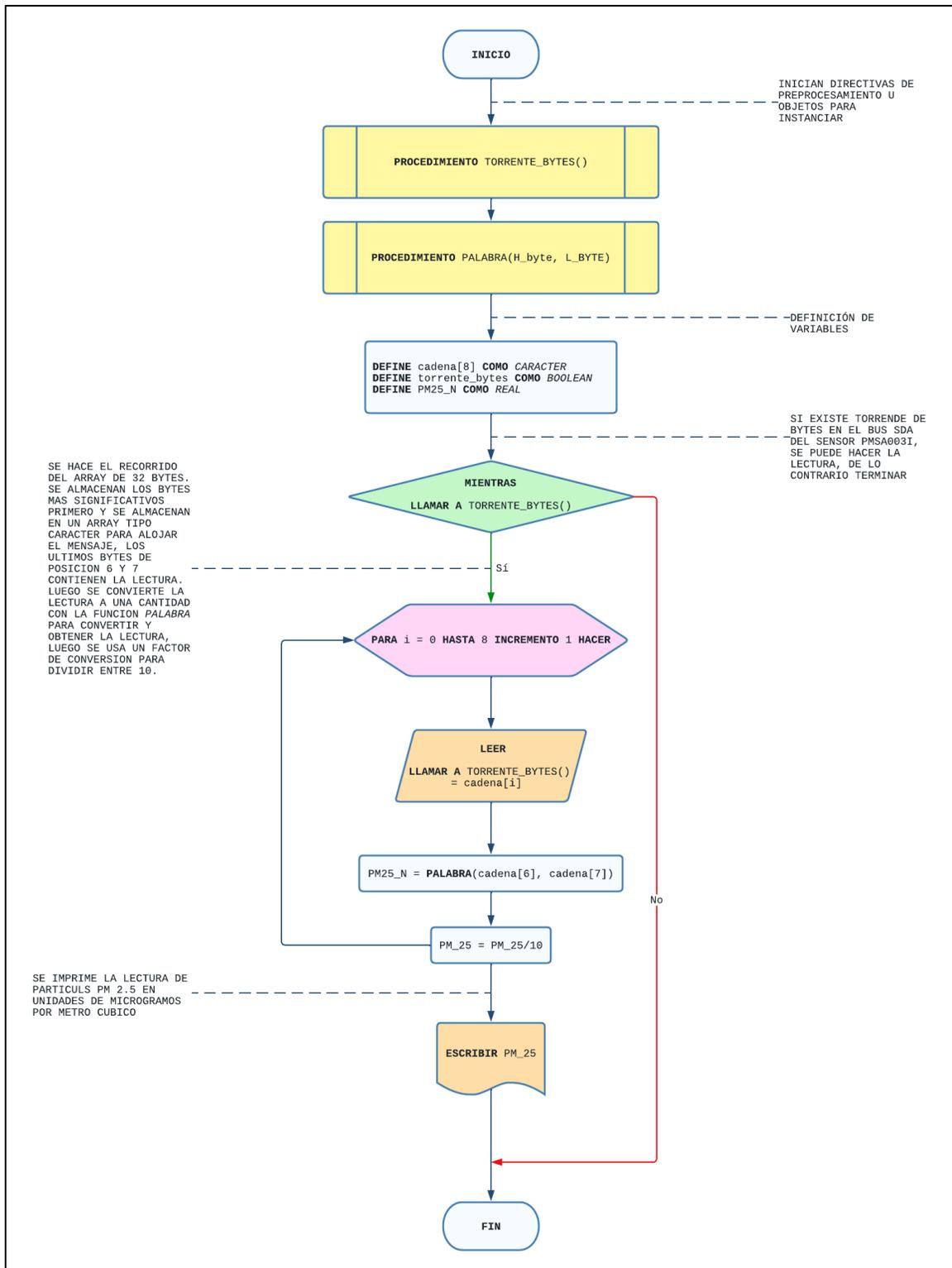


Figura 25. Algoritmo planteado en pseudocódigo para obtener las lecturas del sensor PMSA003I en unidades de microgramos por metro cúbico, en base a sus registros de memoria 6 y 7.

La siguiente lectura sirve para medir el contaminante de CO y se empleará el uso del sensor *MQ7*, pero al ser el único sensor analógico en y_N , se requiere modelar una expresión para obtener las lecturas de las partes por millón de CO, usando el valor medido de voltajes de las resistencias R_S (resistencia del sensor *MQ7*) y R_L (resistencia de carga del sensor *MQ7*), en un esquemático como el que se muestra en la Figura 26, el cual representa el circuito del encapsulado de este sensor.

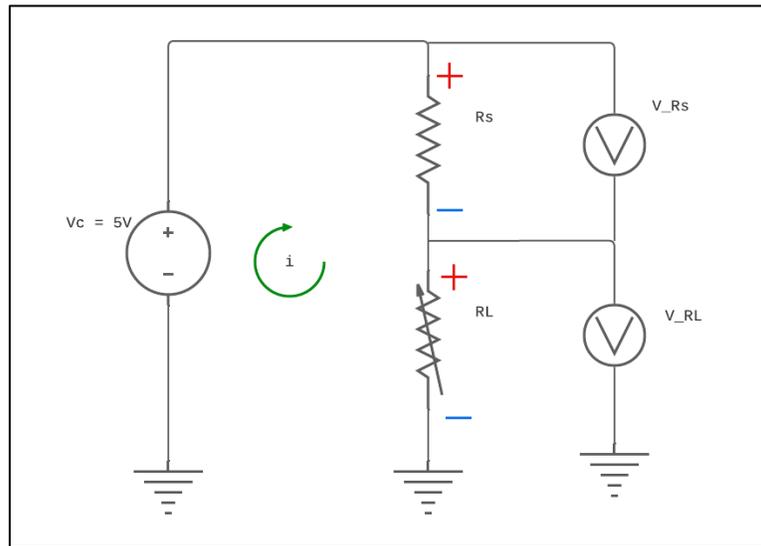


Figura 26. Diagrama eléctrico de sensor *MQ7* alimentado a tensión de 5 V.

Al analizar el circuito por malla, se llegó a la siguiente expresión.

$$R_S = R_L \cdot \frac{V_C - V_{RL}}{V_{RL}} \quad (6)$$

La ecuación 6 sirve para obtener los valores experimentales de la resistencia del sensor *MQ7* y usarse para llegar a un modelo matemático que relacione el valor de R_S con el valor de $PPMCO_N$.

La hoja de datos del sensor *MQ7* indica que debe efectuarse un proceso denominado *tiempo de*

precalentamiento inicial, esto significa que no debe usarse el sensor inmediatamente para tomar dichas lecturas, por lo que debe dejarse energizado a $V_c = 5 V$ (que es su suministro nominal de tensión), y esperar al menos 48 horas antes de poder utilizarlo [24].

Después, con ayuda de las curvas de sensibilidad que ofrece la hoja de datos del fabricante, tal como se observa en el gráfico de la Figura 27, se deben identificar dos curvas de interés particular, la curva de sensibilidad del CO y la curva de sensibilidad del aire, donde el comportamiento es de tipo logarítmico tanto en el eje de las ordenadas como de las abscisas. En el eje de las abscisas se establece el valor de las partes por millón *ppm* de concentración de la sustancia que sea de interés particular, en un intervalo de definición visible de entre 50 a 4000 décadas. Mientras que en el eje de las ordenadas se establece un factor de sensibilidad denotado por el cociente $\frac{R_s}{R_o}$, donde R_o es el valor de la resistencia del sensor R_s pero a una concentración exacta de 100 *ppm* de CO.

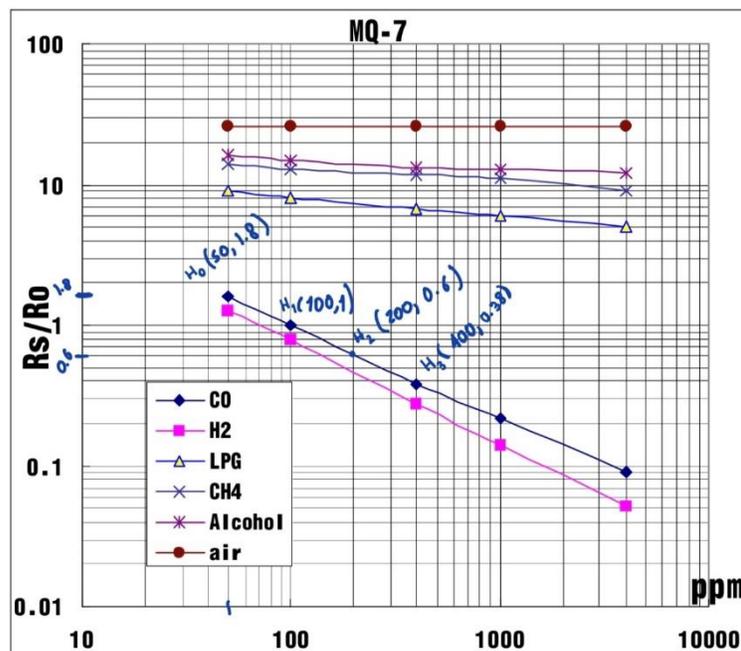


Figura 27. Curvas de sensibilidad del sensor MQ7 [24].

El método consiste en obtener de manera experimental el valor de la resistencia R_o , bajo las mejores condiciones de temperatura, humedad y aire puro que se puedan lograr en el ambiente. Para ello, es necesario conocer el valor del cociente $\frac{R_s}{R_o}$ en la curva de sensibilidad del aire de referencia, así que como se puede observar en la Figura 27, este valor es de 28, tal y como se muestra en la expresión 7.

$$\frac{R_{s \text{ aire}}}{R_o} \approx 28 \quad (7)$$

Donde $R_{s \text{ aire}}$ es la variabilidad de la resistencia del sensor en presencia de aire puro. La ecuación 7 es una relación de proporción y se puede representar por medio del triángulo en color verde mostrado la Figura 28. Como se observa, la base del triángulo es R_o y la altura es $R_{s \text{ aire}}$, y dado que esta proporción es constante como se muestra en la curva del aire de la Figura 27, se puede dibujar un segundo triángulo en color azul, pero inscrito dentro del verde, representando la misma proporción, pero con valores distintos en la base y altura del triángulo. Estos catetos representan la resistencia a 100 pm de CO medida por el sensor, nombrada como $R_{o \text{ medido}}$, mientras que el comportamiento de la resistencia del sensor a cualquier concentración de sustancia de aire, pero medido por el dispositivo, es representada como $R_{s \text{ aire medido}}$.

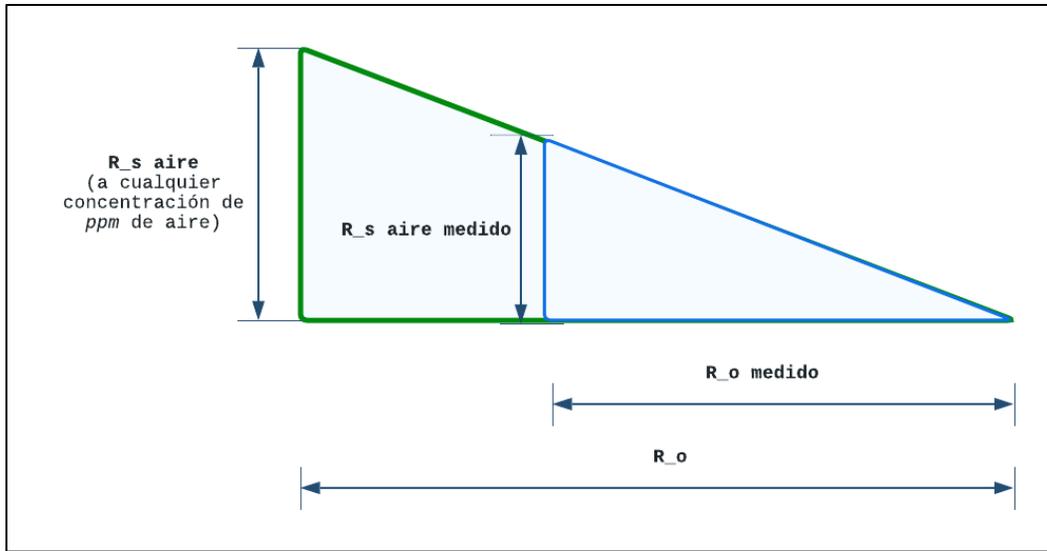


Figura 28. Relación de proporción entre los factores de sensibilidad. El triángulo verde representa el factor dado por el fabricante y el triángulo azul representa el factor a calibrar.

El valor del cateto $R_{o\ medido}$ es lo que se requiere conocer, para ello se emplea la ayuda de ecuación 8, que representa la igualdad de proporciones entre ambos triángulos.

$$\frac{R_{s\ aire}}{R_o} = \frac{R_{s\ aire\ medido}}{R_{o\ medido}} \quad (8)$$

De la expresión 8 se despeja $R_{o\ medido}$ y se obtiene la expresión 9.

$$R_{o\ medido} = \frac{R_{s\ aire\ medido}}{\frac{R_{s\ aire}}{R_o}} \quad (9)$$

Se sustituye la ecuación 7 en 9 y se obtiene la expresión 10.

$$R_{o\ medido} = \frac{R_{s\ aire\ medido}}{28} \quad (10)$$

La expresión 10 representa explícitamente el valor buscado que es $R_{o\ medido}$.

Como se observa en la ecuación 6, R_s representa la variabilidad de la resistencia del sensor, en presencia de la sustancia gaseosa a medir que sea de interés particular (aire o CO), mientras que la resistencia de carga R_L es designada por el fabricante [24], con un valor de $R_L = 1\ k\Omega$, el cual es un potenciómetro colocado en el encapsulado comercial del sensor *MQ7*, y se deberá ajustar al mínimo para que la resistencia de carga pueda considerarse como $R_L \approx 1\ k\Omega$. De acuerdo con el diagrama de la Figura 26, el circuito debe energizarse a $V_c = 5\ V$, para luego analizar su comportamiento primeramente en presencia de aire, por lo que se usa la ecuación 6 para obtener la resistencia del sensor $R_{s\ aire}$, así que dicha ecuación queda modificada sustituyendo estos valores mencionados y se obtiene la expresión 11.

$$R_{s\ aire} = R_L \cdot \frac{V_c - V_{RL}}{V_{RL}} = 1 \cdot \left(\frac{5 - V_{RL}}{V_{RL}} \right) = \frac{5 - V_{RL}}{V_{RL}} \quad (11)$$

Donde V_{RL} está dada en V y $R_{s\ aire}$ en $k\Omega$.

Luego se necesitar escoger una entrada analógica del microcontrolador *ESP32*, designando el pin 34 para recibir los datos de señal analógica del sensor *MQ7*, por medio de una función en las librerías del lenguaje *C* y cuando se haga la lectura se almacenará en una variable que ha sido nombrada como A_{RL} , en referencia a la entrada analógica que medirá el valor de la resistencia R_L ,

tal como se muestra en la expresión 12, con ayuda de una función en *C* conocida como *analogRead()*, cuyo argumento es el pin 34 del *ESP32*.

$$A_{RL} = \text{analogRead}(34) \quad (12)$$

El valor del voltaje V_{RL} de la ecuación 11, se obtiene por medio de la lectura analógica A_{RL} , así que de acuerdo con la hoja de datos del *ESP32* [25], sus canales *ADC* poseen una resolución de 12 bits, es decir, un conteo discreto de $2^{12} - 1 = 4095$, y un voltaje de operación de 3.3 V, obteniendo dicha lectura de voltaje por medio de la expresión 13.

$$V_{RL} = \frac{A_{RL}}{4095} \cdot 3.3 \quad (13)$$

Con un multímetro, se debe comprobar también el valor de este mismo voltaje V_{RL} en las terminales de la resistencia R_L , esto se hace colocando la punta en el pin 34 en referencia a tierra y luego corroborar que las lecturas del multímetro y las que ofrezca el *ESP32* sean similares, esto servirá para demostrar que el sensor *MQ7* está ofreciendo los valores correctamente a través del *ESP32*. Las lecturas del *ESP32* se obtienen programando en código *C* las ecuaciones 12, 13, 11 y 10 en un algoritmo propuesto, el cual consiste en obtener el valor de $R_{o\ medido}$, obteniendo 100 muestras del valor de A_{RL} por medio de la ecuación 12 y usar el promedio para ser sustituido en la expresión 13, para finalmente ser sustituido en la expresión 11. Este algoritmo se detalla en la Figura 29 y el código en *C* en el Anexo B.

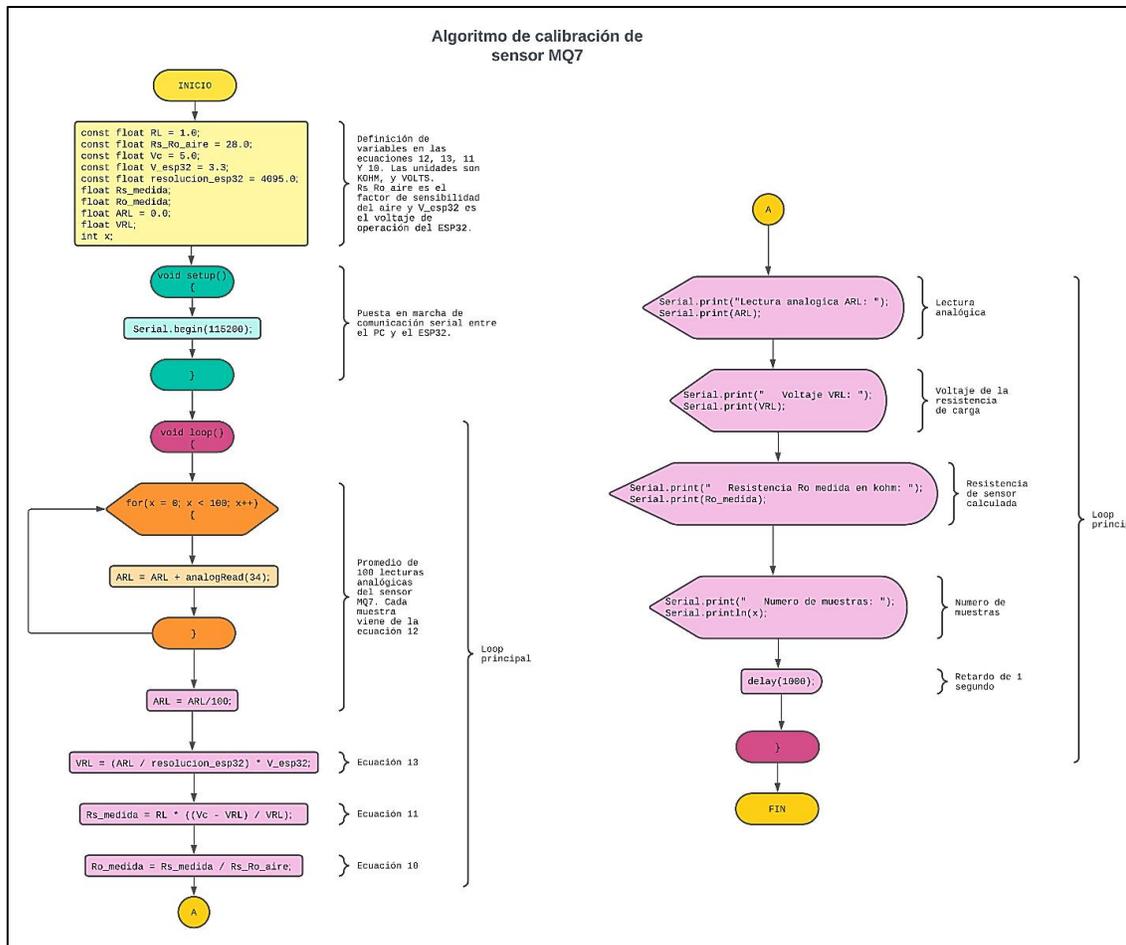


Figura 29. Algoritmo de calibración del sensor MQ7.

El programa arrojará un conjunto de columnas de datos, cada una correspondiente a los valores de A_{RL} , V_{RL} y $R_{o\ medido}$, donde dichas lecturas podrán variar continuamente debido a que el tiempo de precalentamiento indicado por el fabricante, es como mínimo de 72 hrs. así que es posible que se pueda requerir de más tiempo hasta llegar a un valor estable [24], pero llegará un momento en el que se llegará a un valor constante de $R_{o\ medido}$ fijo o con variaciones imperceptibles, y cuando esto ocurra se tomará nota de este valor constante.

El valor obtenido de $R_{o\ medido}$ se representa por medio de la base de un triángulo en color azul como el mostrado en la Figura 30, y representa el factor de sensibilidad del aire ya calibrado de manera experimental gracias al procedimiento de la Figura 29. Luego, un segundo triángulo inscrito en él, que representará al monóxido de carbono CO que medirá el sensor, se puede visualizar en el triángulo en color morado de la misma Figura 30.

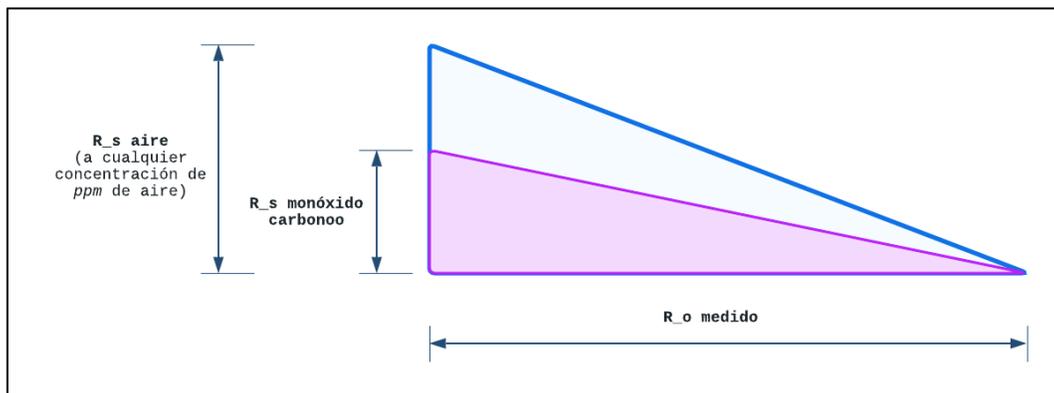


Figura 30. Relación no proporcional entre los factores de sensibilidad. El triángulo azul representa el factor calibrado de la sustancia aire y el triángulo morado representa el factor del monóxido de carbono CO.

Este triángulo en color morado puede entenderse como la cantidad de monóxido de carbono CO presente que se encuentra disuelto en el aire, es decir, el triángulo azul. Se puede notar que ambos triángulos no son proporcionales, pero tienen en común el mismo tamaño de sus bases, es decir el valor de la resistencia $R_{o\ medido}$. Luego, del comportamiento logarítmico de la curva de sensibilidad del gas CO mostrada en la Figura 27, se denotan cuatro puntos de interés particular nombrados como $H_0(50,1.8)$, $H_1(100,1)$, $H_2(200,0.6)$ y $H_3(400,0.38)$, donde los primeros dos puntos se tomarán como referencia para su *interpolación logarítmica* ya que son los más cercanos entre sí y no se requiere perder tanta precisión tratando de medir más de 100 ppm de gas CO. Este tipo de interpolación requiere el uso de la ecuación 14.

$$F(x) = F_i \cdot \left(\frac{x}{x_i}\right)^{\frac{\log\left(\frac{F_j}{F_i}\right)}{\log\left(\frac{x_j}{x_i}\right)}} \quad (14)$$

Donde $F(x)$ es la función logarítmica que representa los valores en el eje de las ordenadas y x es la variable independiente en el eje de las abscisas, en relación con la Figura 27. Las variables x_i , x_j , F_i y F_j representan los parámetros del par de puntos de interés a interpolar $H_i(x_i, F_i)$ y $H_j(x_j, F_j)$, mientras que los subíndices i y j representan el número de punto.

Como los puntos de interés particular son $H_0(50,1.8)$ y $H_1(100,1)$, éstos son representados por medio de las ecuaciones 15 y 16 respectivamente.

$$H_i(x_i, F_i) = H_0(x_0, F_0) = H_0(50,1.8) \quad (15)$$

$$H_j(x_j, F_j) = H_1(x_1, F_1) = H_3(100,1) \quad (16)$$

Se sustituyen las expresiones 15 y 16 en 14 y se obtiene la expresión 17.

$$F(x) = 1.8 \cdot \left(\frac{x}{50}\right)^{\frac{\log\left(\frac{1}{1.8}\right)}{\log\left(\frac{100}{50}\right)}} = \frac{49.6585}{x^{0.847997}} \quad (17)$$

Debe notarse que en el eje de las ordenadas de la Figura 27, se tiene que $F(x) = \frac{R_s}{R_o \text{ medido}}$, mientras que en el eje de las abscisas se tiene que $x = PPMCO_N$, por lo que la ecuación 17 se reescribe como la expresión 18.

$$\frac{R_{s\ CO}}{R_{o\ medido}} = \frac{49.6585}{PPMCO_N^{0.847997}} \quad (18)$$

Como se muestra en la ecuación 18, se ha utilizado una nueva variable denominada $R_{s\ CO}$, esto es para poder diferenciar que la resistencia del sensor *MQ7* será referencia para medir su variabilidad frente al monóxido de carbono CO, ya que dicha ecuación solo podrá ser usada para esta sustancia en particular. Posteriormente se despeja $PPMCO_N$ y se obtiene la expresión 19, siendo así el modelo final para la obtención de las lecturas de partes por millón de CO en función de la resistencia del sensor $R_{s\ CO}$ adquirida con la expresión 18.

$$PPMCO_N = \frac{99.9999}{\left(\frac{R_{s\ CO}}{R_{o\ medido}}\right)^{1.17925}} \quad (19)$$

Por otro lado, para obtener el valor de $R_{s\ CO}$ de la expresión 19, se emplea el uso de la ecuación 6, la cual mide la resistencia del sensor *MQ7* en presencia de cualquiera de los gases detectables por él. Esto implica definir que $R_s = R_{s\ CO}$ para la detección de CO, además de que se reitera que el fabricante establece una resistencia de carga $R_L = 1k\Omega$ y el voltaje de suministro es de $V_c = 5\ V$, por lo tanto, la ecuación 6 se reescribe como la expresión 20. Estas últimas expresiones 19 y 20 servirán para obtener la curva de comportamiento del sensor, donde este procedimiento se detalla en la sección 3.3.1, que explica la ejecución de todo el procedimiento anteriormente descrito para este sensor.

$$R_{s\ CO} = 1 \cdot \frac{5 - V_{RL}}{V_{RL}} = \frac{5 - V_{RL}}{V_{RL}} \quad (20)$$

Finalmente, para la luminosidad de una habitación como parámetro informativo que puede indicar la presencia de ruido térmico inducido en la habitación, se utilizará un fotorresistor *LDR*, que por sus siglas significa *resistencia dependiente por la luz*. Al igual que el sensor *MQ7*, este es también de tipo analógico, por lo que de manera similar es necesario implementar un circuito divisor de voltaje y una configuración matemática para la correcta lectura de esta variable. Ver la figura 31.

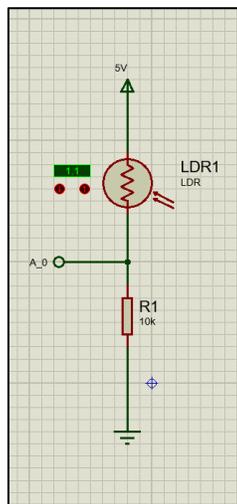


Figura 31. Divisor de voltaje para LDR.

Para obtener el valor del voltaje en la terminal *A0* del diagrama, se utiliza la expresión 21.

$$V_{A0} = \frac{R1}{R_{LDR} + R1} \cdot V_c \quad (21)$$

Donde $R1 = 10 \text{ k}\Omega$, $V_c = 5 \text{ volts}$ es el suministro de la fuente de todos los sensores utilizados anteriormente, V_{A0} es el voltaje divisor por parte del *LDR* y R_{LDR} es el valor de la resistencia variable, [26]. Despejando R_{LDR} de la ecuación 21 se obtiene la expresión 22.

$$R_{LDR} = \frac{R \cdot V_c}{V_{A0}} - R1 \quad (22)$$

Así mismo, la resolución del *ESP32* es de 12 bits, por lo que el voltaje del divisor V_{A0} se obtiene con la lectura obtenida en el pin A0 de entrada analógico del microcontrolador, y debe ser convertido a unidades de voltaje. Dicha conversión se logra a través de la ecuación 23.

$$V_{A0} = A0 \cdot \frac{V_c}{4095} \quad (23)$$

Donde $A0$ es la lectura en unidades de 0 a 4095 discretizada por el convertidor analógico a digital del microcontrolador, equivalente al voltaje de 0 a 5 volts variado por la resistencia del sensor R_{LDR} . Sustituyendo 23 en 22 se obtiene la función del valor de la resistencia del *LDR* en términos del voltaje medido en la expresión 24.

$$R_{LDR} = \frac{4095 \cdot R}{A0} - R \quad (24)$$

Para obtener el valor de la iluminancia, se usa la ecuación 24, y usar una referencia de lectura de otro dispositivo de medición llamado *luxómetro*, pero para el trabajo presente y dada la naturaleza de ser solo una variable informativa al usuario, así como para entender la posible presencia de inducción térmica por radiación del sistema de iluminación, se utiliza una función ya modelada y propuesta por [27, p. 127]. La ecuación 25 lo representa.

$$L_N = 800.69 \cdot R_{LDR}^{-1.283} \quad (25)$$

Donde la iluminancia L_N medida en unidades de *lux*, depende directamente de la resistencia variable por la luz R_{LDR} .

Por último, la Tabla 7 muestra un resumen de todos los sensores seleccionados anteriormente.

Tabla 7. Resumen de características de sensores a utilizar para el sistema de instrumentación, [28], [29], [30], [24].

SENSOR	TIPO	PRINCIPIO DE FUNCIONAMIENTO	UNIDAD DE MEDIDA	VARIABLE DE LA MEDIDA	ALCANCE	RESOLUCIÓN	PRECISIÓN	EXACTITUD
DHT11	Digital	Termistor <i>NTC</i> para la temperatura, donde disminuye la resistencia al aumentar la temperatura y Capacitivo con variabilidad en la capacitancia por la permitividad eléctrica entre placas.	Temperatura en [°C], Humedad relativa en [%]	Temperatura T_N , Humedad H_N	Temperatura [0,50] °C, Humedad relativa [20,90] %	Temperatura 1 °C y Humedad 1%	Temperatura ±1 °C, Humedad relativa ±1 %	Temperatura ±2 °C, Humedad relativa ±5 %
LDR	Analógico	Efecto <i>fotorresistivo</i> , donde la resistencia disminuye en aumento de la luminosidad del ambiente	[Ω], pero se puede relacionar con la unidad de luminosidad [<i>lux</i>]	L_N	Hasta 2 <i>M</i> Ω en completa oscuridad	Depende de resolución ADC del microcontrolador	-----	-----
PMSA003I	Digital	Detección por dispersión de luz láser.	$\left[\frac{\mu g}{m^3}\right]$	$PM_{2.5N}$	[0,1000] $\frac{\mu g}{m^3}$	1 $\frac{\mu g}{m^3}$	±10 $\frac{\mu g}{m^3}$	-----
MQ7	Analógico	Detección electroquímica, en aumento de las partes por millón de concentración de <i>CO</i> , disminuye la resistencia del sensor.	[Ω], pero se puede relacionar con la unidad de [<i>ppm</i>]	$PPMCO_N$	[2,20] <i>k</i> Ω	Depende de resolución ADC del microcontrolador	-----	-----

2.3.2 Selección y uso del microcontrolador

Para lograr el cometido de la instrumentación del sistema, es preciso establecer previamente todos los pines necesarios de alguna tarjeta adquisidora de datos y así transmitir las lecturas a la red de *IoT*. El modelo comercial *ESP32* es escogido debido a su conectividad inalámbrica a una red de Internet y bajo consumo energético de 3 V. La tabla 8 muestra la planeación designada de cada uno de los pines con su respectivo sensor de lectura.

Tabla 8. Relación de pines del ESP32 con los dispositivos a conectar.

NOMBRE DEL PIN	DISPOSITIVO	USO	FUNCIÓN
3V3	PMSA003I, TXS0108E	Fuente de tensión DC de 3.3 V	Suministra 3.3 V DC al sensor PMSA003I, así como para el convertidor lógico de nivel TXS0108E
GND	PMSA003I, TXS0108E, DHT11, LDR, MQ7, LCD 16x2, LM2596, LED AZUL, LED VERDE	Tierra	Referencia de conexión
IO4	DHT11	Entrada digital	Entrada de datos al microcontrolador por SPI
IO33	LDR	Entrada analógica	Entrada para la transmisión de señales analógicas para convertirlas por ADC
IO13	PMSA003I	I2C	Comunicación de datos digitales por medio de protocolo I2C, por SDA
IO32	PMSA003I	I2C	Comunicación de datos de reloj por medio de protocolo I2C, por SCL
IO34	MQ7	Entrada analógica	Entrada para la transmisión de señales analógicas para convertirlas por ADC
IO18	Relé de Artic Air Ultra 18008	Salida digital	Salida para encendido o apagado de aire acondicionado
IO19	Relé de Igoto Swiss EX4-16W	Salida digital	Salida para encendido o apagado del extractor de aire

IO21	Pantalla LCD 16x2	I2C	Comunicación de datos digitales por medio de protocolo I2C, por SDA
IO22	Pantalla LCD 16x2	I2C	Comunicación de datos de reloj por medio de protocolo I2C, por SCL

La Tabla 8 muestra también tres dispositivos comerciales auxiliares, el encapsulado *LM2596*, el *TXS0108E* y una pantalla *LCD* de 18 columnas por 2 filas de espacio, los cuales servirán como una fuente estable auxiliar de 5 V, un conversor lógico de nivel y visualizador de las lecturas de medición respectivamente.

2.4 Metodología para la predicción del confort térmico de acuerdo con la norma *ISO 7730*

Una vez establecidos los lugares de pruebas y las herramientas para efectuar el diseño y desarrollo del sistema, tal como se esclarece en las secciones anteriores, la semántica de la red de *IoT* se puede lograr por medio de técnicas basadas en la norma *ISO 7730: Determinación de los índices PMV y PPD y especificación de las condiciones para confort térmico*, donde se puede implementar una metodología para predecir si el sistema de control de confort térmico puede ser aceptado o no por un porcentaje de personas alojadas en un espacio cerrado.

El objetivo de la norma es poder medir los grados de satisfacción e insatisfacción térmica en un espacio interior, por medio de la predicción de emisión de votos de un grupo de personas y que a través de una directriz, se almacenen en dos variables llamadas *PMV* y *PPD*, que por sus siglas en inglés significan *Voto Medio Estimado* y *Porcentaje Estimado de Insatisfechos* respectivamente.

El valor de PMV debe encontrarse dentro de una escala de valores como la que se muestra en la Tabla 9, donde la predicción del usuario indicaría en que grado de satisfacción térmica se pudiera encontrar.

Tabla 9. Escala de sensación térmica de 7 niveles, [4].

+3	Muy caluroso
+2	Caluroso
+1	Ligeramente caluroso
0	Neutro
-1	Ligeramente fresco
-2	Fresco
-3	Frío

Para el valor de PPD , éste revela el porcentaje de personas insatisfechas sesgadas en alguno de los límites mostrados en la Tabla 9, por lo que un 0% de personas insatisfechas se encontraría en el valor de 0 de la tabla, es decir una sensación térmica neutra. La Figura 32 muestra un espacio muestral S conteniendo al grupo de insatisfechos y su complemento como los satisfechos, esperando que los resultados experimentales puedan predecir el menor porcentaje posible de PPD con esta metodología.

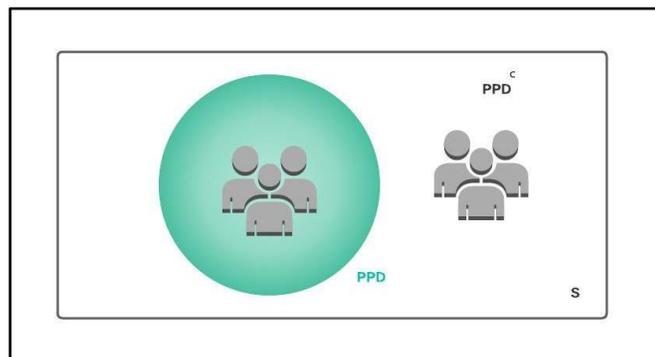


Figura 32. Conjuntos de espacio muestral para personas satisfechas e insatisfechas.

El procedimiento de predicción emplea el uso de las ecuaciones 26 a 30, pero no pueden usarse directamente, hasta emplear un algoritmo que tome ciertas decisiones basadas en el tipo de ropa, humedad relativa, velocidad del viento en el interior y la temperatura operativa dentro de la habitación, esto logrado por medio de los sensores que fueron seleccionados.

$$\begin{aligned}
 PMV = & [0.303 \cdot e^{-0.036 \cdot M} + 0.028] \cdot \{(M - W) - 3.05 \times 10^{-3} \\
 & \cdot [5733 - 6.99 \cdot (M - W) - p_a] - 0.42 \cdot [(M - W) - 58.15] \\
 & - 1.7 \times 10^{-5} \cdot M \cdot (5867 - p_a) - 0.0014 \cdot M \cdot (34 - t_a) \\
 & - 3.96 \times 10^{-8} \cdot f_{cl} \cdot [(t_{cl} + 273)^4 - (\bar{t}_r + 273)^4] - f_{cl} \cdot h_c \\
 & \cdot (t_{cl} - t_a)\}
 \end{aligned} \tag{26}$$

$$\begin{aligned}
 t_{cl} = & 35.7 - 0.028 \cdot (M - W) - I_{cl} \cdot \{3.96 \times 10^{-8} \cdot f_{cl} \\
 & \cdot [(t_{cl} + 273)^4 - (\bar{t}_r + 273)^4] + f_{cl} \cdot h_c \cdot (t_{cl} - t_a)\}
 \end{aligned} \tag{27}$$

$$h_c = \begin{cases} 2.38 \cdot |t_{cl} - t_a|^{0.25}, & 2.38 \cdot |t_{cl} - t_a|^{0.25} > 12.1 \cdot \sqrt{v_{ar}} \\ 12.1 \cdot \sqrt{v_{ar}}, & 2.38 \cdot |t_{cl} - t_a|^{0.25} < 12.1 \cdot \sqrt{v_{ar}} \end{cases} \tag{28}$$

$$f_{cl} = \begin{cases} 1 + 1.290 \cdot I_{cl}, & I_{cl} \leq 0.078 \\ 1.05 + 0.645 \cdot I_{cl}, & I_{cl} > 0.078 \end{cases} \tag{29}$$

$$PPD = 100 - 95 * e^{-0.03353 \cdot PMV^4 - 0.02179 \cdot PMV^2} \tag{30}$$

El significado de las variables y sus unidades de medida se establecen en la Tabla 10.

Tabla 10. Variables necesarias para medir el voto de confort de los usuarios, [4].

Variable involucrada en las ecuaciones 5 a 9	Significado	Unidad física de medida	Valor
M	Tasa metabólica del usuario	$\left[\frac{W}{m^2}\right], met$	[46 – 232]
W	Potencia mecánica efectiva del ser humano efectuando actividades físicas de manera cíclica	$\left[\frac{W}{m^2}\right]$	-----
I_{cl}	Medición de aislamiento de la ropa	$\left[m^2 \cdot \frac{^{\circ}K}{W}\right], clo$	[0 – 0.310]
f_{cl}	Factor de superficie de la ropa	Adimensional	-----
t_a ó T_N	Temperatura del aire	$[^{\circ}C]$	[10 – 30]
\bar{t}_r ó T_{rm}	Temperatura radiante media	$[^{\circ}C]$	[10 – 40]
v_{ar}	Velocidad relativa del aire	$\left[\frac{m}{s}\right]$	[0. –1]
p_a ó P_{H_2O}	Presión parcial del vapor de agua	$[Pa]$	[0 – 2700]
h_c	Coefficiente de transmisión del calor por convección	$\left[\frac{W}{m^2 \cdot ^{\circ}K}\right]$	-----
t_{cl}	Temperatura de la superficie de la ropa	$[^{\circ}C]$	-----

Las variables de *PPD* y *PMV* se relacionan por medio de la gráfica mostrada en la Figura 33, para establecer los resultados experimentales finales del análisis de predicción.

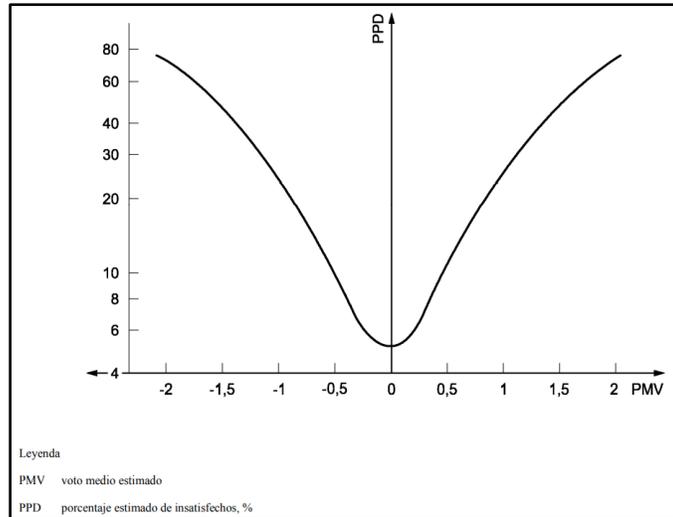


Figura 33. Formato de gráfico representativo de PMV (voto medio estimado de satisfechos) y PPD (porcentaje estimado de insatisfechos por el sistema de control térmico), [4].

Para este método solamente se solicita llenar un conjunto de datos, que para el trabajo presente es identificado como el *vector de parámetros auxiliares A* mostrado en la siguiente expresión, siendo las variables las mostradas en la Tabla 10.

$$\mathbf{A} = \begin{bmatrix} I_{cl} \\ M \\ W \\ v_{ar} \end{bmatrix} \quad (31)$$

Cada uno de los parámetros de \mathbf{A} puede determinarse por medio de valores ya establecidos por medio de tablas y son las entradas de la función del algoritmo de predicción, por lo que es preciso comprender como determinar los valores adecuados. Para obtener el valor de I_{cl} de la expresión 31, se solicita que sea en las unidades mencionadas y pueden determinarse por medio de la Tabla 11.

Tabla 11. Aislamiento térmico para prendas, [4].

Prenda	I_{clu}	
	clo	$m^2 \cdot K/W$
Ropa interior		
Bragas	0,03	0,005
Calzoncillos de perneras largas	0,10	0,016
Camisetas sin mangas	0,04	0,006
Camisetas de manga corta	0,09	0,014
Camiseta de manga larga	0,12	0,019
Bragas y sujetador	0,03	0,005
Camisas/blusas		
Mangas cortas	0,15	0,023
Ligeras, mangas largas	0,20	0,031
Normales, mangas largas	0,25	0,039
De franela, mangas largas	0,30	0,047
Blusa ligera, mangas largas	0,15	0,023
Pantalones		
Cortos	0,06	0,009
Ligeros	0,20	0,031
Normales	0,25	0,039
De franela	0,28	0,043
Vestidos/Faldas		
Faldas ligeras (verano)	0,15	0,023
Faldas gruesas (invierno)	0,25	0,039
Vestidos ligeros, mangas cortas	0,20	0,031
Vestido de invierno, mangas largas	0,40	0,062
Monos	0,55	0,085
Jerseys		
Chalecos sin mangas	0,12	0,019
Jersey fino	0,20	0,031
Jersey	0,28	0,043
Jersey grueso	0,35	0,054
Chaquetas		
Ligeras, de verano	0,25	0,039
Chaquetas	0,35	0,054
Batas	0,30	0,047
Muy aislantes, de fieltro		
Mono	0,90	0,140
Pantalones	0,35	0,054
Chaqueta	0,40	0,062
Chaleco	0,20	0,031
Ropa de abrigo		
Chaquetón	0,60	0,093
Cazadora	0,55	0,085
Parka	0,70	0,109
Pantalones de fieltro	0,55	0,085
Varios		
Calcetines	0,02	0,003
Calcetines gruesos, tobilleros	0,05	0,008
Calcetines gruesos, largos	0,10	0,016
Medias de nilón	0,03	0,005
Zapatos (suela fina)	0,02	0,003
Zapatos (suela gruesa)	0,04	0,006
Botas	0,10	0,016
Guantes	0,05	0,008

Para obtener el valor de la tasa metabólica M de la expresión 31, puede utilizarse los valores de la Tabla 12.

Tabla 12. Valores para tasas metabólicas, [4].

Actividad	Tasa metabólica	
	W/m ²	met
Reposo, tendido	46	0,8
Reposo, sentado	58	1,0
Actividad sedentaria (oficina, domicilio, escuela, laboratorio)	70	1,2
Actividad ligera, de pie (de compras, laboratorio, industria ligera)	93	1,6
Actividad media, de pie (dependiente de comercio, tareas domésticas, trabajo con máquinas)	116	2,0
Caminar en llano		
2 km/h	110	1,9
3 km/h	140	2,4
4 km/h	165	2,8
5 km/h	200	3,4

Para el caso de la potencia mecánica por unidad de superficie corporal, su valor debe tomarse como $W = 0 \text{ W/m}^2$, a menos que el sitio de pruebas pueda ser usado para hacer actividades cíclicas, como las que pudieran ocurrir dentro de un gimnasio o un centro laboral [4].

En cuanto al caso de la velocidad relativa del aire, ésta puede tomar un valor mínimo de $v_{ar} = 0.10 \frac{m}{s}$, ya que es un valor de referencia en [4] y representa el flujo de aire circulante dentro de la habitación (esto debido a el ventilador del aire acondicionado), y donde dicha velocidad podría verse influenciada por corrientes naturales de aire, pero la variabilidad podría despreciarse y considerar una velocidad constante, por lo que en la expresión 31, solo sería necesario identificar los aspectos relacionados con el aislamiento de la ropa y el tipo de actividad física del usuario de referencia.

2.5 Metodología para la construcción de la red de comunicación en IoT

Por último, para probar el sistema de instrumentación y control en la red de IoT, es necesario establecer una metodología de construcción de la red de comunicación y enlace entre los dispositivos conectados, esto con ayuda de las herramientas de *Hardware* y *Software* mostradas en la Figura 34. Estas herramientas se dividen en cuatro bloques, las empleadas para la adquisición de datos de los sensores y actuadores, las empleadas para el servidor local de base de datos, las del servidor en la *nube* y aquellas para el acceso y desarrollo de del gestor gráfico de la red de IoT.

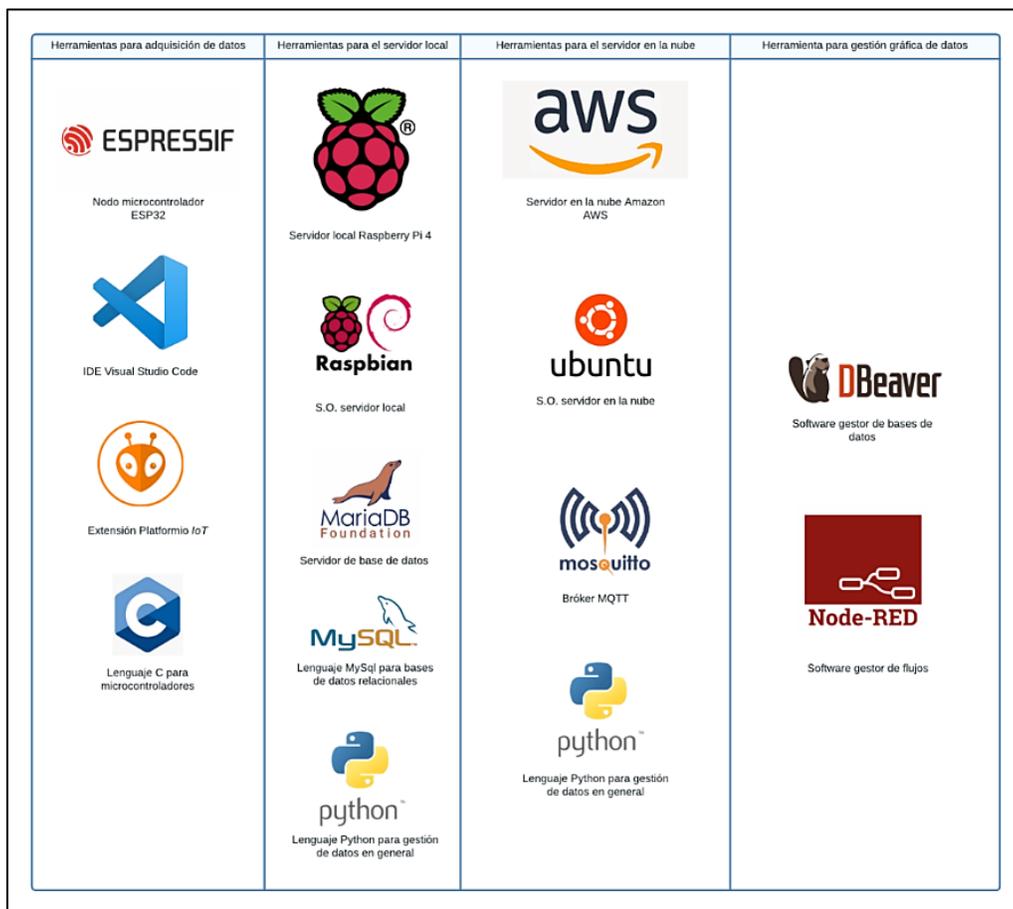


Figura 34. Herramientas utilizadas para desarrollo en IoT.

Los procedimientos de comunicación en la red de *IoT* se efectuarán por niveles de jerarquía, igualando al de un *sistema de comunicación militar*, empleando analogías con la red que permitan un mejor entendimiento de la estructura y control de los flujos de información. En la Figura 35, en conjunto con la Tabla 13, se explica el diseño de la red empleando cuatro niveles de jerarquía principales, la de los *Cabos*, *Sargento*, *Teniente* y el *Alto mando*.

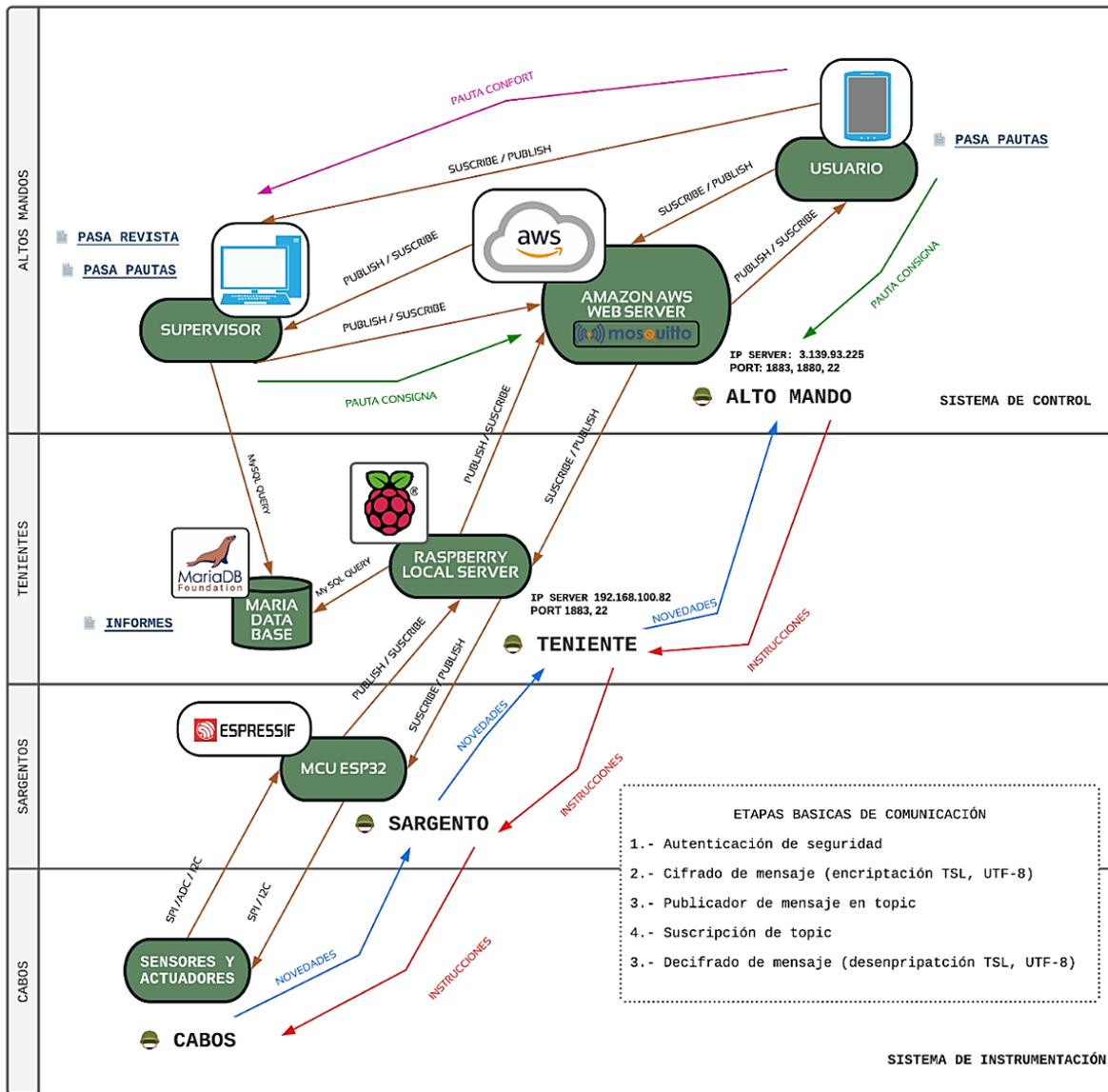


Figura 35. Diseño de la red de comunicación IoT basada en jerarquía militar.

Tabla 13. Procedimientos de comunicación por topics y jerarquías alusivas al sistema militar.

No.	ENLACE DE COMUNICACIÓN	TOPICS DE ENLACE	PROCEDIMIENTO MILITAR	INTERPRETACIÓN DEL PROCEDIMIENTO
1	Cabos → Sargento	No aplica, comunicación por <i>I2C</i> o <i>SPI</i>	Los Cabos dan parte de las <i>novedades</i> a su Sargento	El <i>ESP32</i> recibe datos de lectura de los sensores y estado de actuadores con ayuda del algoritmo de adquisición de datos y luego da estructura de los parámetros del vector C_N .
2	Sargento → Teniente	sargento/novedades/teniente	El Sargento se identifica. Cifra las <i>novedades</i> y da parte de ellas a su Teniente.	El <i>ESP32</i> autenticado en el <i>bróker</i> puede publicar el mensaje de las lecturas al <i>topic</i> de este paso. El <i>Raspberry</i> autenticado puede suscribirse a este <i>topic</i> y recibir la información de las lecturas y estados de los actuadores.
3	Teniente → Alto Mando	teniente/novedades/alto_mando	El Teniente se identifica. Recibe parte de las <i>novedades</i> del Sargento y descifra el mensaje. El contenido del mensaje es archivado. Se vuelve a cifrar el mensaje y el Teniente da parte de él al Alto Mando.	El <i>Raspberry</i> suscrito al <i>topic</i> del paso 2 inserta los registros recibidos por <i>queries</i> en la base de datos de <i>MariaDB</i> . Luego publica el mensaje de lecturas al <i>topic</i> de este paso 3. El servidor de <i>Amazon AWS</i> autenticado en el <i>bróker</i> se suscribe a este mismo <i>topic</i> para recibir el mensaje.
4	Alto Mando → Teniente	(a) alto_mando/instrucciones/teniente (b) PMV	El Alto Mando se identifica. Recibe parte de las <i>novedades</i> del Teniente y descifra el mensaje. Entre los integrantes del Alto	El servidor suscrito al <i>topic</i> del paso 3 recibe el vector de consignas deseadas x_N como entrada para las funciones de control y

		(c) PPD	<p>Mando, se toma una decisión en base a las <i>novedades y pautas de consigna</i> recibidas para generar un mensaje de <i>instrucciones</i>. Se archiva el mensaje y luego se cifra para dar parte de las <i>instrucciones</i> al Teniente. El Alto Mando puede recibir también las <i>pautas de confort</i> para luego <i>pasar revista</i> de toda la información obtenida.</p>	<p>retornar las salidas como un mensaje de <i>instrucciones</i>, es decir el vector w_N. También puede recibir el vector A como entrada de la función del método numérico para retornar los valores de estimación <i>PMV</i> y <i>PPD</i>. El mensaje de <i>instrucciones</i> se publica en el <i>topic</i> (a) mostrado en este paso y los valores de <i>PMV</i> y <i>PPD</i> en los <i>topics</i> (b) y (c) respectivamente. Se grafican todos los datos y se analiza la información para generar un <i>informe</i>, esto es la semántica de la red de <i>IoT</i>.</p>
5	Teniente → Sargento	Teniente/instrucciones/sargento	<p>Teniente se identifica. Recibe <i>instrucciones</i> del Alto Mando y descifra el mensaje. Cifra de nuevo el mensaje y da parte de él al Sargento.</p>	<p>El <i>Raspberry</i> autenticado en el <i>bróker</i> se suscribe al <i>topic</i> (a) del paso 4 para recibir el mensaje de <i>instrucciones</i>. Luego publica el mensaje al <i>topic</i> de este paso, en el que el <i>ESP32</i> está suscrito para recibirlo. Las <i>instrucciones</i> pueden ser generadas por el servidor de <i>Amazon AWS</i> o de manera manual por el dispositivo <i>usuario</i> o <i>supervisor</i>.</p>
6	Sargento → Cabos	No aplica, comunicación por <i>I2C</i> o <i>SPI</i>	<p>Sargento se identifica. Recibe <i>instrucciones</i> del Teniente, descifra el mensaje y lo convierte en</p>	<p>El <i>ESP32</i> suscrito al <i>topic</i> del paso 5 recibe el mensaje de los estados lógicos del vector w_N para activar o</p>

			una orden para los Cabos. Los Cabos ejecutan la orden y se repite el proceso.	desactivar las salidas digitales de los pines 18 y 19 del ESP32. El proceso se repite al paso 1.
--	--	--	---	---

Esto facilita la trazabilidad de los mensajes en los procedimientos de *publicación* y *suscripción* de los *topics*, facilitando la programación de código en *Python* y *C*, usando palabras clave y nombramiento de los mensajes alusivos al léxico militar que se suele emplear. Además, haber encomendado tareas específicas a cada dispositivo de la red de *IoT*, permitirá dividir los procedimientos y hacer más liviano el código a que si un solo dispositivo tuviera que hacerse cargo de varias operaciones a la vez.

Los vectores \mathbf{x}_N , \mathbf{A} , \mathbf{C}_N y \mathbf{w}_N se transcriben en la red de *IoT* como *arreglos*, representados por las expresiones 32, 33, 34 y 35 respectivamente, y sirven para mejorar la legibilidad en los códigos en *C* y *Python* así como su representación en mensajes de variables tipo *String*. El nombre de cada uno de los arreglos mostrados hace alusión al tipo de mensajes empleados en el léxico militar, donde una *pauta* es un mensaje de recomendación de un elemento a otro, una *novedad* es toda aquella información emitida y una *instrucción* es una orden dada desde los altos mandos. Por otro lado, la palabra *revista* en la Figura 35 hace alusión al procedimiento de revisión y análisis de los datos experimentales, en el procedimiento de la sección 2.4.

$$pauta_consigna = [T_{SN}, PM2.5_{SN}, PPMCO_{SN}] \quad (32)$$

$$pauta_confort = [I_{cl}, M, W, v_{ar}] \quad (33)$$

$$novedades = [T_N, H_N, L_N, PM2.5_N, PPMCO_N, A_{N1}, A_{N2}] \quad (34)$$

$$instrucciones = [A_{N1}, A_{N2}] \quad (35)$$

Finalmente, la terminación del sistema de instrumentación y control se complementa con la implementación de una interfaz gráfica para el usuario. Servirá para la interacción con la red de *IoT*, de tal manera que se pretende imitar un conjunto de instrumentos virtuales para el dispositivo *Supervisor* de la Figura 35. Por tal motivo, se pretende diseñar la interfaz gráfica como un panel de control, para que el usuario pueda ingresar los valores deseados del mensaje *pauta_consigna* y *pauta_confort*, así como la visualización grafica de comportamiento de las variables T_N , H_N , $PM2.5_N$ y $PPMCO_N$ en función del tiempo t para poder visualizar el tiempo de asentamiento de los controladores, así como los datos adicionales referentes a L_N , H_N , PMV y PPD ya que constituyen la semántica de la arquitectura mostrada en la Figura 9, cuando se obtengan los datos experimentales y resultados. La Figura 36 muestra el bosquejo de diseño de la interfaz gráfica, compuesta por 4 bloques principales. Como se observa, el primer bloque contiene los controles manuales de los actuadores, así como los parámetros auxiliares de luminosidad, humedad relativa y los valores de confort térmico. El segundo bloque contiene las gráficas relacionadas al confort térmico y los valores actuales dentro de la habitación con relación a la consigna deseada por el usuario, esto ocurre de igual manera para el resto de los bloques para el confort de calidad el aire.

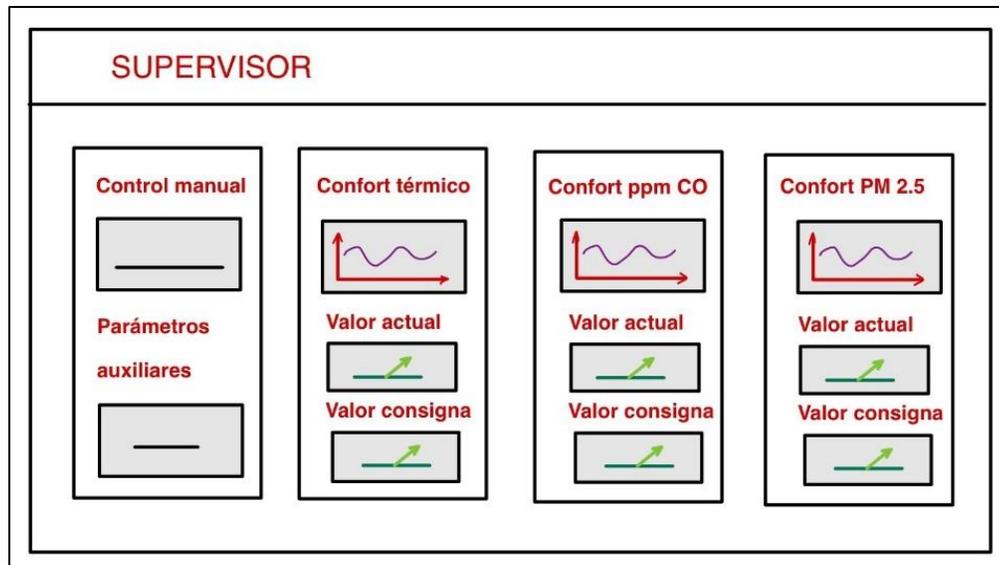


Figura 36. Bosquejo de referencia para construcción de la interfaz gráfica para el dispositivo supervisor.

La construcción de la interfaz se logra por medio de la herramienta *Node-RED*, tal como es establecido en la Figura 34 y puede requerir de la escritura de algunas funciones en *JavaScript* dentro de su uso. Por otro lado, la interfaz gráfica se podrá visualizar directamente en un navegador *web* como *Edge* o *Chrome* si se tiene el enlace y los permisos de seguridad designados.

Capítulo 3: Construcción del sistema de instrumentación y control

En este capítulo se hace la ejecución de las metodologías mostradas en el Capítulo 2, para llegar a la construcción final del sistema de control e instrumentación en la red de *IoT*, así como la ejecución de pruebas de funcionamiento preliminares antes de realizar los experimentos en el Capítulo 4. Por otro lado, se muestran todos los algoritmos diseñados y construidos para cumplir con el cometido antes descrito. Al cumplir con cada uno de los procedimientos del Capítulo 2, se llegó al diseño final del sistema de instrumentación y control de todo el sistema, ya implementado en la red de *IoT* mostrada en la Figura 35.

3.1 Construcción de la red eléctrica de mando y potencia

De acuerdo con las secciones 2.2.1 y 2.3.1 y con ayuda de la planeación de pines de entradas y salidas del *ESP32* en la Tabla 8, se efectuó la integración de todos los sensores y actuadores, esto se muestra en el esquema de conexión del Anexo A. Finalmente, a este conjunto de interconexión se le denominará con el nombre de *objetos de la red de IoT*, tal como se muestra en la Figura 37, y representó la constitución de la capa de objetos en la arquitectura de *IoT*.

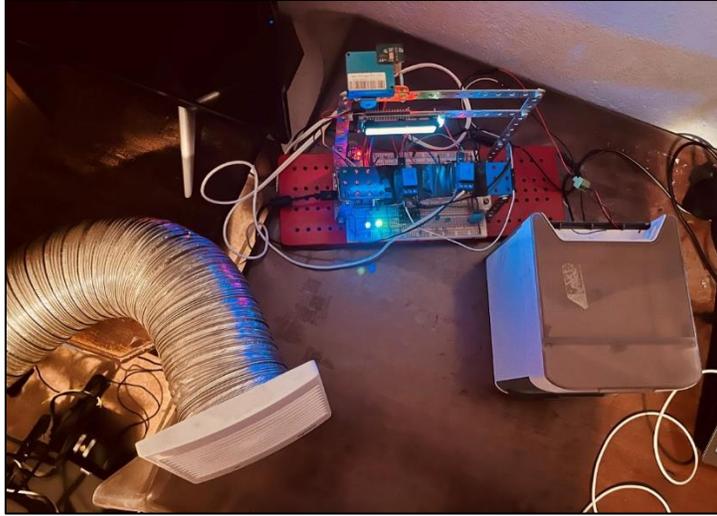


Figura 37. Objetos de la red de IoT.

3.2 Construcción de algoritmos de control

Una vez que la capa de objetos ya fue construida, tal y como se estableció en la sección anterior, se sigue con la construcción de los algoritmos de control para los actuadores especificados en la Tabla 2. Para ello, se ejecutó el procedimiento de la sección 2.2 y con ayuda de las tablas 3 y 5. Estos algoritmos fueron diseñados y escritos en el lenguaje *Python* para ser ejecutados en el servidor en la nube de *Amazon AWS* localizado en el nivel de jerarquía mostrado en la Figura 35, y los códigos se pueden ver implementados en el Anexo F.

3.2.1 Algoritmo de control de temperatura

Este algoritmo sirve para obtener como resultado el estado lógico de la variable A_{N1} de la expresión 2, recibiendo como entrada el valor deseado de temperatura T_{SN} y la retroalimentación

de la lectura de sensor T_N , por eventos discretos en cada una de las etapas de la Tabla 3. Esto se detalla en el diagrama de flujo en pseudocódigo mostrado en la Figura 38.

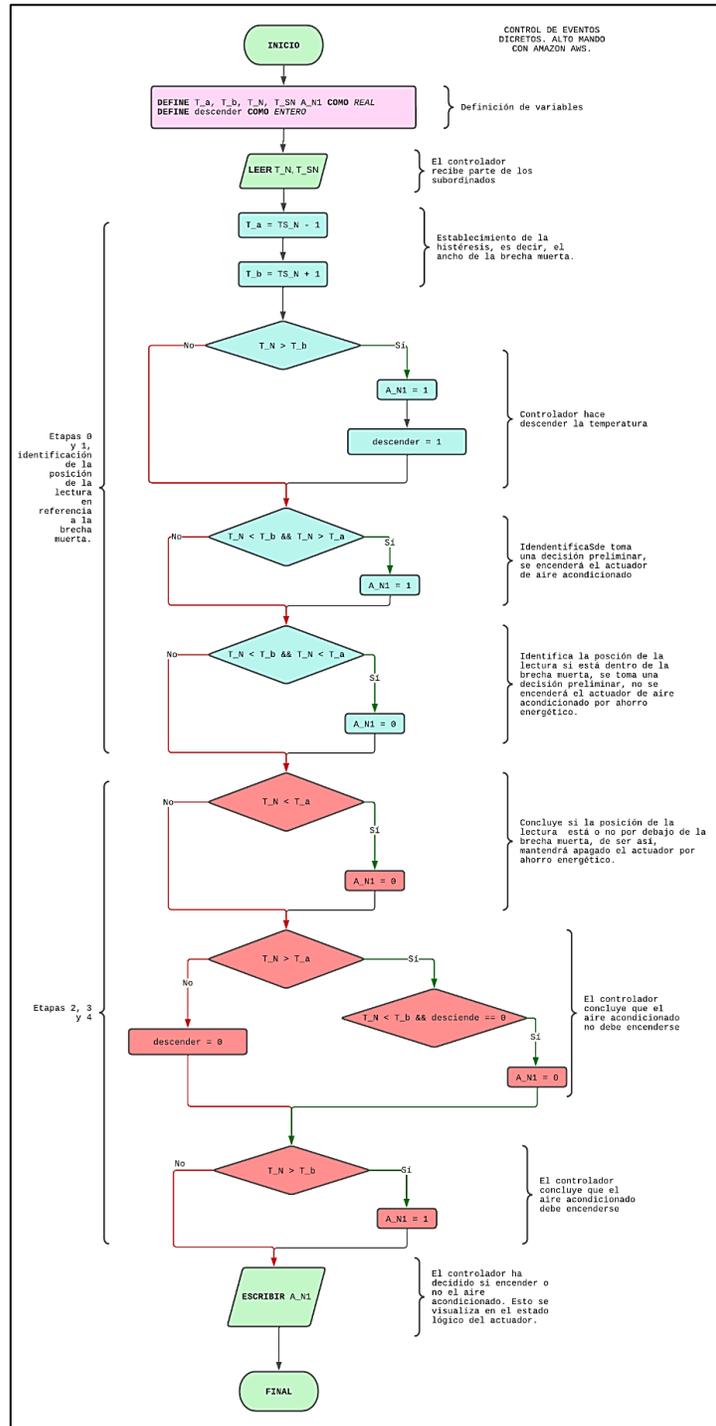


Figura 38. Algoritmo de control de temperatura en pseudocódigo para usarse como subrutina en Python.

En términos generales, el algoritmo de la Figura 38, detecta si la variación de la lectura de temperatura ocurre de manera ascendente o descendente, por lo que los bloques en color azul determinan el estado lógico del actuador A_{N1} si la temperatura va en dirección descendente debido a la activación del aire acondicionado y los bloques color rojo actúan cuando la temperatura sea ascendente debido a la desactivación del actuador, permitiendo que la naturaleza del incremento térmico ocurra de acuerdo al clima en el ambiente, afectando la temperatura operativa dentro de la habitación.

3.2.2 Algoritmo de control de la calidad del aire

Este algoritmo sirve para obtener como resultado el estado lógico de la variable A_{N2} de la expresión 2, recibiendo como entradas los límites permisibles y deseados por el usuario $PM2.5_{SN}$ y $PPMCO_{SN}$ en conjunto con la retroalimentación de las lecturas de sensor $PM2.5_N$ y $PPMCO_N$, por eventos discretos en cada una de las etapas de la Tabla 5. Esto se detalla en el diagrama de flujo en pseudocódigo mostrado en la Figura 39.

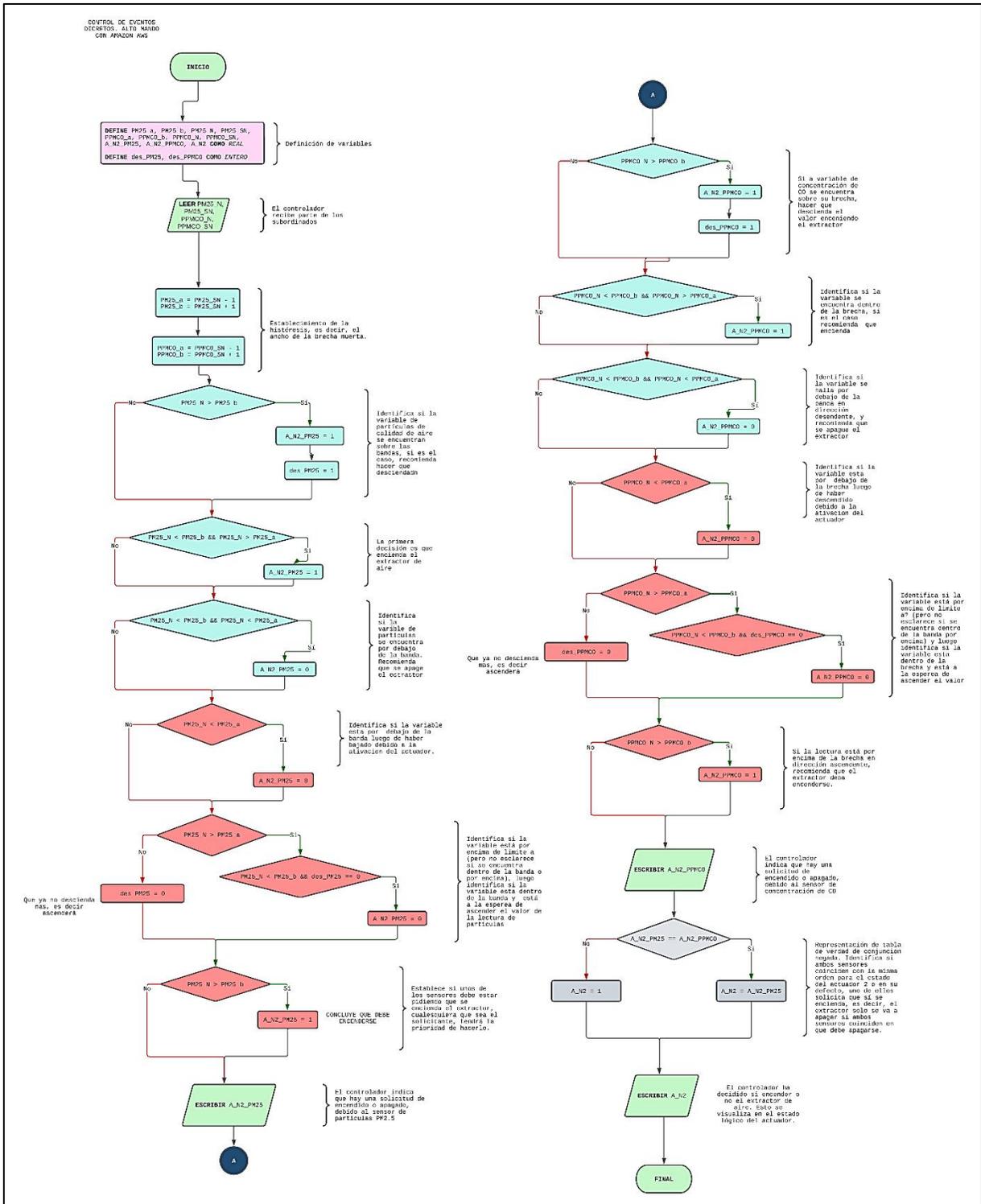


Figura 39. Algoritmo de control de calidad del aire en pseudocódigo para usarse como subrutina en Python.

3.3 Construcción de algoritmo de adquisición de datos

Para la construcción del algoritmo de adquisición de datos para el sistema de instrumentación, se empleó la ejecución de la metodología mostrada en la sección 2.3, para definir la estructura de la Figura 23.

Antes de la construcción del algoritmo, tal y como se especificó en la metodología de la sección 2.3.1, se tuvo que obtener primero el modelo experimental que ofrece las lecturas de medición del sensor *MQ7*, que indica las partes por millón de *CO* en función de la resistencia del sensor *MQ7*, obteniendo un conjunto de curvas de sensibilidad, de las cuales se seleccionó la más estable. Una vez seleccionado el modelo de la curva correspondiente, se implementó finalmente en el algoritmo de adquisición de datos.

El algoritmo de adquisición de datos fue escrito en el lenguaje *C* para ser ejecutado en el microcontrolador *ESP32* localizado en el nivel de jerarquía mostrado en la Figura 35, y el código se puede ver implementado en el Anexo C.

3.3.1 Obtención experimental de curvas de sensibilidad del sensor *MQ7*.

El procedimiento mostrado en la sección 2.3.1 se implementó en el mes de marzo de 2023. Como se observa en la Figura 40, el sensor fue energizado directamente por la fuente de poder estable *LM2596*, para que al término del periodo de tiempo de precalentamiento, se pudiera utilizarlo con el *ESP32* para iniciar con la calibración del sensor.

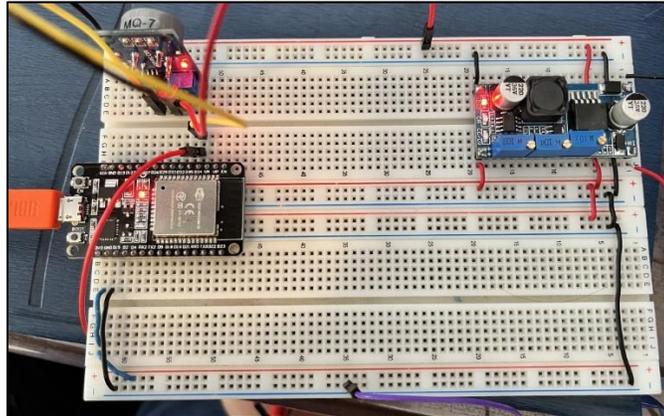


Figura 40. Evidencia de obtención inicial de precalentamiento y calibración de sensor MQ7.

Finalmente, el programa del algoritmo de la Figura 29 e implementado en el Anexo B para el ESP32, arrojó los valores de A_{RL} , V_{RL} , R_o medido por cada segundo, donde dichos resultados se pueden observar en la Figura 41.

Lectura analogica ARL: 480.23	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 480.77	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 479.16	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 479.16	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 474.17	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 476.44	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 474.22	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 477.35	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 483.12	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 486.51	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 494.64	Voltaje VRL: 0.40	Resistencia Ro medida en kohm: 0.41	Numero de muestras: 100
Lectura analogica ARL: 487.45	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 491.57	Voltaje VRL: 0.40	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 490.74	Voltaje VRL: 0.40	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 492.00	Voltaje VRL: 0.40	Resistencia Ro medida en kohm: 0.41	Numero de muestras: 100
Lectura analogica ARL: 484.21	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 483.93	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 481.45	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 477.26	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 481.91	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.42	Numero de muestras: 100
Lectura analogica ARL: 475.07	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 479.98	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 476.09	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 478.52	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 473.87	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 479.78	Voltaje VRL: 0.39	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 476.76	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 472.92	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 473.44	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100
Lectura analogica ARL: 475.42	Voltaje VRL: 0.38	Resistencia Ro medida en kohm: 0.43	Numero de muestras: 100

Figura 41. Obtención de resultados experimentales del algoritmo de calibración.

En la segunda columna de datos de la Figura 41, se hallan los resultados del voltaje V_{RL} , esta variable se midió también directamente con un multímetro entre tierra y el pin analógico del sensor *MQ7*. Esto quiere decir que los resultados de voltaje mostrados en la Figura 42, fueron similares a los obtenidos por el multímetro, donde dichas lecturas se muestran en la Figura 41, donde se observó que la variabilidad del voltaje fue cercana a $V_{RL} = 0.529 V$.

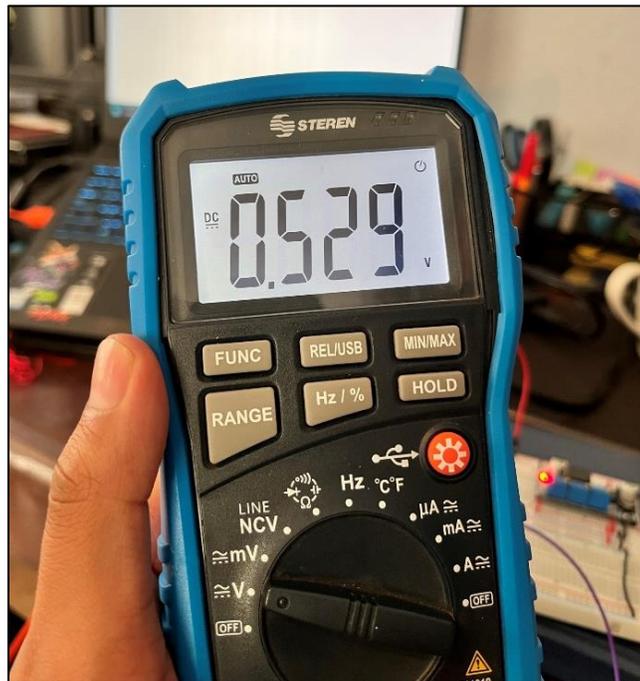


Figura 42. Lectura de voltaje V_{RL} obtenida con el multímetro.

La Tabla 14 muestra un extracto de algunas de las lecturas de voltaje V_{RL} obtenidas por el *ESP32* y por el multímetro.

Tabla 14. Extracto de lecturas de V_{RL} obtenidas por el *ESP32* y el multímetro.

FECHA DE CALIBRACIÓN	HORA	CONDICIONES DE TEMPERATURA Y HUMEDAD	V_{RL} <i>ESP32</i>	V_{RL} multímetro

		RELATIVA EN EL AMBIENTE		
2 de marzo de 2023	16:14 hrs.	28 °C, 19%	0.39 V	0.529 V
2 de marzo de 2023	16:20 hrs.	28 °C, 20%	0.40 V	0.563 V
2 de marzo de 2023	16:30 hrs.	27 °C, 20%	0.40 V	0.533 V
2 de marzo de 2023	16:40 hrs.	26 °C; 21%	0.38 V	0.518 V
2 de marzo de 2023	17:00 hrs.	26 °C, 20%	0.40 V	0.524 V

Como se puede notar en la Tabla 14, a partir de las 14:00 horas aproximadamente, del día 2 de marzo de 2023, después de más de 48 horas de precalentamiento, se identificó un instante en particular cuando los valores de V_{RL} empezaron a estabilizarse, por lo tanto, las lecturas de la resistencia $R_{o\ medido}$ terminaron siendo cercanas a un valor de $R_{o\ medido} \approx 0.43\ k\Omega$, tal y como se observa en la Figura 41, así que por tal motivo, la selección de la resistencia es esta misma, y representada por la expresión 36.

$$R_{o\ medido} \approx 0.43\ k\Omega \quad (36)$$

Se sustituye la ecuación 36 en 19 y se obtiene la expresión 37.

$$PPMCO_N = \frac{99.9999}{\left(\frac{R_{s\ CO}}{0.43}\right)^{1.17925}} = \frac{36.963}{R_{s\ CO}^{1.17925}} \quad (37)$$

La ecuación 37 es el modelo final encontrado. El gráfico de la Figura 43 muestra el historial de comportamiento de varios valores de $R_{o\ medido}$ obtenidos, donde la curva de sensibilidad en color azul representa a la ecuación 37, cuando finalmente el valor de la resistencia se estabilizó.

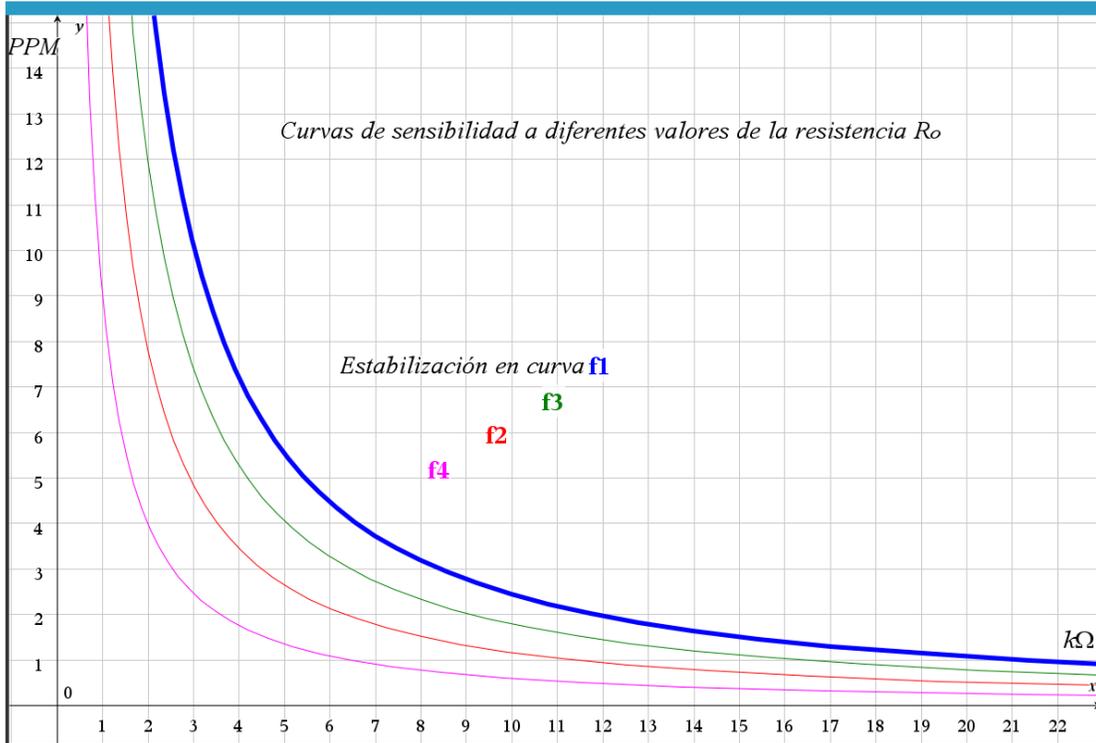


Figura 43. Curvas de sensibilidad obtenidas para detección de monóxido de carbono en relación con los datos experimentales de la calibración del sensor MQ7.

Puede notarse cómo al aumento de las partes por millón de CO , disminuye la resistencia eléctrica del sensor $R_{S\ CO}$, el cual se obtiene por medio de la expresión 20.

3.3.2 Algoritmos para la adquisición de datos

Dado que en la sección anterior se logró identificar la curva de sensibilidad que mide las partes por millón de CO , siendo un caso especial del sensor analógico $MQ7$ y además, el resto de los sensores a emplear son digitales, se logró proponer directamente el algoritmo de adquisición de datos programado en el $ESP32$. Por medio del diagrama de flujo mostrado en la Figura 44, se puede

expresar dicho algoritmo, el cual fue escrito en C y mostrado en el Anexo C. El objetivo del algoritmo es leer las mediciones de los sensores y poder retroalimentarse en la red de *IoT* mostrada en la Figura 35.

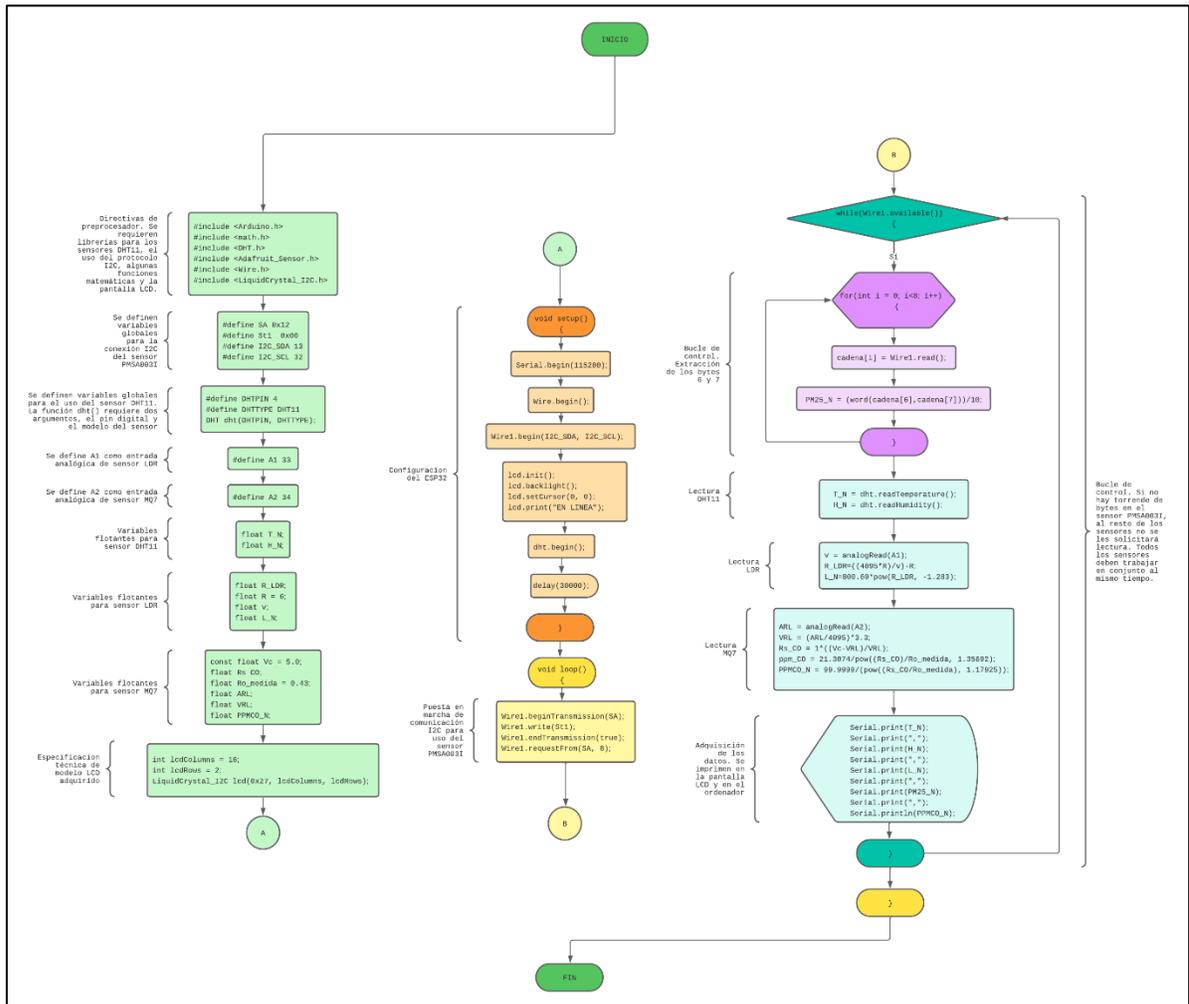


Figura 44. Algoritmo de adquisición de datos con el ESP32 escrito en C.

Este algoritmo representa la construcción finalizada de la capa de abstracción en la arquitectura de *IoT*.

Las entradas de la función de este algoritmo son los valores de T_N y H_N provenientes de las lecturas del sensor *DHT11*, así que el dispositivo *supervisor* se programó para ser comunicado con el servidor en la *nube* y recibir estos datos directamente desde la red de *IoT*, tal y como fue establecido en la planeación de la jerarquía de red de la Figura 35. El algoritmo requiere el ingreso manual de las variables mostradas en la expresión 33, que son I_{cl} con ayuda de la Tabla 11, la tasa metabólica del usuario M en *met* con ayuda de la Tabla 12, la potencia por unidad de superficie corporal disipada por esfuerzos físicos de manera cíclica efectuados por el usuario W en W/m^2 , la cual tendrá un valor de *cero* (porque no se considerarán esfuerzos físicos como ejercicios deportivos dentro de las habitaciones de pruebas), y finalmente la velocidad relativa del aire v_{ar} dentro de la habitación en $\frac{m}{s}$, (siendo ella debido a la convección de aire inducido por el aire acondicionado en conjunto con la convección de aire natural) y su valor medido es de $0.10 \frac{m}{s}$, de acuerdo al modelo del actuador de aire acondicionado. Esto quiere decir que la expresión 33 debe ser ingresada manualmente por el usuario, dando la estructura mostrada en la expresión 38.

$$pauta_comfort = [I_{cl}, M, 0, 0.10] \quad (38)$$

El usuario podrá visualizar en la interfaz gráfica la solicitud del ingreso de esta información, donde en conjunto con el resto de los valores de entrada del algoritmo de la Figura 45, los cuales son los datos provenientes de las lecturas T_N y H_N del sensor *DHT11*, y a través del servidor en la *nube* y según la planeación de diseño de la Figura 35, el algoritmo podrá funcionar arrojando el par ordenado PMV y PPD . Debe notarse que la temperatura radiante media $T_{rm}=T_N$, de acuerdo con lo especificado en la Tabla 6.

3.5 Construcción de red de comunicación en *IoT*

Finalmente, se ejecutó el último procedimiento mostrado en la sección 2.5, implementando las herramientas mostradas en la Figura 34, para la red de *IoT*, siendo específicamente el servidor en la *nube*, el servicio de comunicación *MQTT* y el servidor de base de datos. El seguimiento de construcción se logró por medio de la jerarquía de red mostrada en la Figura 35 y cada una de las etapas de comunicación establecidas en la Tabla 13.

3.5.1 Pruebas de comunicación con el *ESP32*

Una vez que los sensores y actuadores en conjunto con el *ESP32* se lograron conectar a la red de comunicación como se muestra en la Figura 46, la pantalla del *display* devuelve un aviso con el mensaje *En línea*, indicando que el *ESP32* está listo para enviar las lecturas dentro de alguna de las habitaciones de pruebas, así como estar preparado para recibir las órdenes de actuación para los actuadores A_{N1} y A_{N2} .

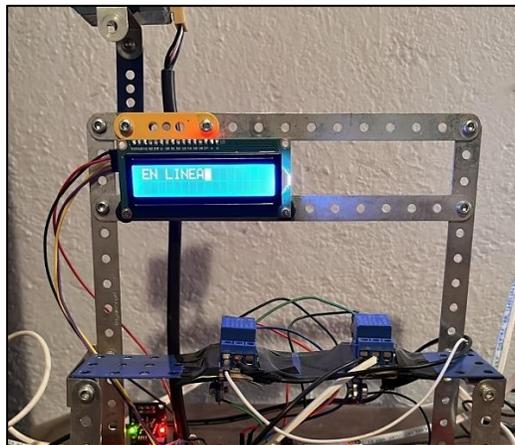


Figura 46. Display mostrando conexión exitosa a la red de área local y bróker del servidor.

El proceso de conexión se programó para permitir un retraso de 30 s al inicio previo de la conexión, para permitir a los sensores terminar los preparativos de emisión de datos, así como la correcta sincronización entre ellos. Una vez terminado el proceso, el *display* muestra las lecturas de medición de los sensores. Para observar el comportamiento, se efectuaron pruebas de comunicación en la red y luego se estableció una comunicación serial con la extensión de *platformio* de *visual studio code* para mostrarse en la *Shell de comandos* de un ordenador *Windows*, tal y como se observa en la Figura 47.

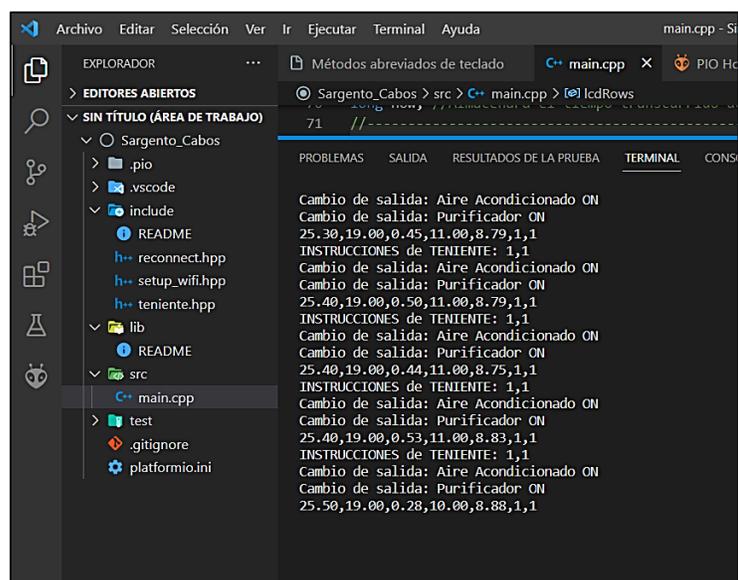


Figura 47. Pruebas de funcionamiento de comunicación con el ESP32.

Como se observa, las lecturas del *ESP32* se estructuraron tal y como se planeó en el arreglo del mensaje *novedades*, visualizando un conjunto de líneas de datos, siendo las cifras correspondientes a las lecturas de los sensores y estados de los actuadores, esta comunicación se representa por medio de los puntos 1 y 2 de la Tabla 13. Se reitera que esta información mostrada en la Figura 47 no está dirigida al usuario sino a los dispositivos de red en *IoT*. Por otro lado, se visualizan otros

mensajes con el encabezado de mensaje *Instrucciones del teniente*, esto representa la conexión exitosa con el servidor y el *Raspberry*, retornando el mensaje de *instrucciones* donde el *ESP32* puede activar o desactivar el aire acondicionado y extractor de aire según se le indique a dicho dispositivo, esta comunicación son los puntos 5 y 6 de la Tabla 13. El código se puede visualizar en el Anexo C.

3.5.2 Pruebas de comunicación con el *Raspberry Pi 4*

Se siguió con la verificación de funcionamiento del *Raspberry*, siendo el procedimiento de conexión inicial similar al *ESP32*, primeramente, estableciendo la conexión a Internet del área local y luego al *bróker* de *Mosquitto*. Posteriormente se lograron visualizar las lecturas enviadas del *ESP32* en la *Shell de comandos* tal y como se muestra en la Figura 48.

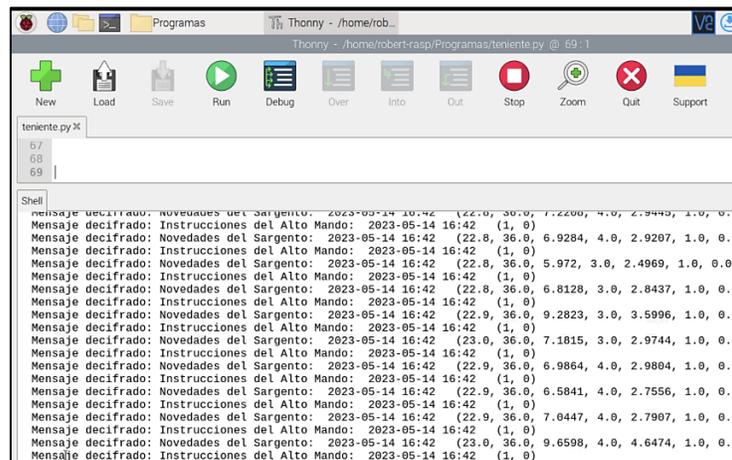


Figura 48. Pruebas de funcionamiento de comunicación con el *Raspberry Pi 4*

Como se observa en la figura, la *Shell de comandos* del *Raspberry* muestra los mensajes de *novedades*, con un encabezado mostrado como *Novedades del sargento*, haciendo alusión a la

comunicación con el *ESP32*, del punto 2 de la Tabla 13. De igual manera se visualizan las *instrucciones*, en un encabezado mostrado como *Instrucciones del alto mando*, haciendo alusión a la conexión con el servidor de *Amazon AWS*, esto quiere decir que el *Raspberry* es el responsable de comunicar las entradas a los controladores, y cada vez que lo haga, recibirá una instrucción del servidor que posteriormente comunicará al *ESP32*, esta comunicación se refiere los puntos 3 y 4 de la Tabla 13. El código fue escrito en *Python* y se muestra en el Anexo D.

3.5.3 Pruebas de comunicación con el servidor de *Amazon AWS*

Posteriormente, las pruebas de conexión con el servidor son las más importantes porque de él dependen los servicios de comunicación en la red de *IoT*. El *Raspberry* le transfiere el mensaje de *novedades* al servidor, donde una vez recibido el mensaje, se reestructura el mensaje y se convierten en entradas para los algoritmos de las Figuras 38 y 39, para luego devolver el mensaje de *instrucciones* al *Raspberry*, esto de acuerdo con los puntos 3 y 4 de la Tabla 13. En la Figura 49 se muestra la estructura de la información recibida al servidor de *Amazon AWS* y sus instrucciones devueltas a toda la red de *IoT*. El código fue escrito en *Python* y se muestra en el Anexo E.

```

root@ip-172-31-8-182:/home/ubuntu
root@ip-172-31-8-182:/home/ubuntu# cd /home/ubuntu/programas
root@ip-172-31-8-182:/home/ubuntu/programas# ls
Alto_Mando.py
root@ip-172-31-8-182:/home/ubuntu/programas# python3 Alto_Mando.py
ALTO MANDO EN LINEA
Mensaje decifrado del Teniente: 2023-04-26 02:16 (27.0, 15.0, 0.6623, 11.0, 8.1569, 0.0, 0.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:16 (26.9, 15.0, 0.635, 11.0, 7.8671, 0.0, 0.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:16 (27.1, 15.0, 0.5901, 12.0, 8.1439, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:16 (26.9, 15.0, 0.635, 13.0, 8.1308, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:17 (27.0, 15.0, 0.5548, 11.0, 8.2484, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:17 (26.9, 15.0, 0.608, 11.0, 8.17, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:17 (27.0, 15.0, 0.6441, 12.0, 8.1308, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:17 (26.9, 15.0, 0.5461, 12.0, 8.1439, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:17 (27.0, 15.0, 0.6532, 12.0, 8.1962, 1.0, 1.0)
Instrucciones: (1, 1)
Mensaje decifrado del Teniente: 2023-04-26 02:17 (26.9, 15.0, 0.5287, 13.0, 8.0125, 1.0, 1.0)

```

Figura 49. Pruebas de funcionamiento de comunicación con el servidor de *Amazon AWS*.

Las pruebas de comunicación del servidor hacia el resto de los dispositivos en la red de *IoT* fueron exitosas.

3.5.4 Pruebas de comunicación con el servidor de base de datos

La implementación de la base de datos se hizo de acuerdo con la jerarquía mostrada en la Figura 35 usando el servicio de *MariaDB*, tal como fue establecido en la Figura 34. La base de datos consta de tres campos que fueron nombrados *novedades*, *pauta_consigna* y *confort* en alusión a las estructuras de datos mostradas en las expresiones 34, 32 y 33 respectivamente. En ese orden, y al tratarse de una base de datos relacional, como se muestra en la Figura 50, cada una de las tres tablas se relacionaron por medio de una operación de unión y los registros obtenidos fueron todas las lecturas de los sensores, estados de los actuadores y los datos arrojados por el algoritmo de predicción de confort térmico. El modo de acceder a esta base de datos es por medio de una conexión remota desde el programa *Excel* o *Dbeaver* al servidor local de *MariaDB* alojado en el *Raspberry*, tal como se observa en la Figura 51. Al efectuar las pruebas de comunicación se estableció una conexión exitosa, de tal manera que el dispositivo *supervisor* podrá obtener esta información y entender el comportamiento y funcionamiento de todo el sistema de instrumentación y control.

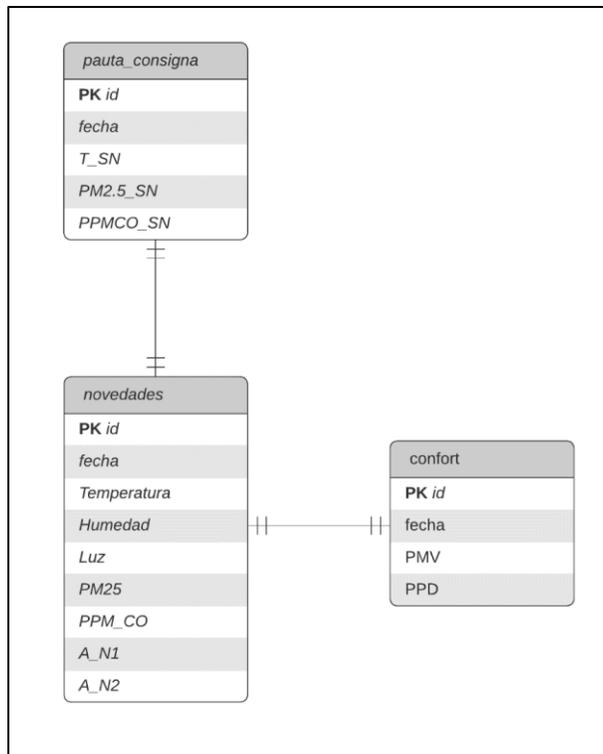


Figura 50. Relación 1 a 1 entre tablas.

SELECT confort.*, novedades.*, pauta_consigna.*
FROM lecturas.confort confort, lecturas.novedades novedades, lecturas.pauta_consigna pauta_consigna
WHERE
confort.id = novedades.id
AND confort.id = pauta_consigna.id

id	fecha	PMV	PPD	id	fecha	Temperatura	Humedad	Luz	PM25	PPM_CO
1	2023-05-16 13:55:53.000	1,572	54,802	1	2023-05-16 13:55:53.000	22,6	33	2,679	4	3
2	2023-05-16 13:55:57.000	1,572	54,802	2	2023-05-16 13:55:56.000	22,6	33	2,625	4	3
3	2023-05-16 13:56:02.000	1,572	54,802	3	2023-05-16 13:56:01.000	22,6	33	2,734	4	0
4	2023-05-16 13:56:03.000	1,574	54,945	4	2023-05-16 13:56:03.000	22,5	33	2,411	3	2
5	2023-05-16 13:56:06.000	1,58	55,233	5	2023-05-16 13:56:06.000	22,3	33	2,064	3	2
6	2023-05-16 13:56:09.000	1,577	55,089	6	2023-05-16 13:56:09.000	22,4	33	2,359	4	2
7	2023-05-16 13:56:12.000	1,577	55,089	7	2023-05-16 13:56:12.000	22,4	33	2,451	4	2
8	2023-05-16 13:56:15.000	1,588	55,71	8	2023-05-16 13:56:15.000	22,3	34	2,625	4	2
9	2023-05-16 13:56:18.000	1,594	55,994	9	2023-05-16 13:56:18.000	22,1	34	2,571	3	2
10	2023-05-16 13:56:21.000	1,591	55,852	10	2023-05-16 13:56:21.000	22,2	34	2,491	3	2
11	2023-05-16 13:56:25.000	1,594	55,994	11	2023-05-16 13:56:25.000	22,1	34	2,191	3	2
12	2023-05-16 13:56:28.000	1,601	56,423	12	2023-05-16 13:56:28.000	21,8	34	2,544	3	2
13	2023-05-16 13:56:31.000	1,596	56,197	13	2023-05-16 13:56:31.000	22	34	2,424	3	2
14	2023-05-16 13:56:34.000	1,604	56,567	14	2023-05-16 13:56:34.000	21,7	34	2,504	3	2
15	2023-05-16 13:56:37.000	1,61	56,886	15	2023-05-16 13:56:37.000	21,8	35	2,346	3	2
16	2023-05-16 13:56:40.000	1,613	57,027	16	2023-05-16 13:56:40.000	21,7	35	2,491	3	2
17	2023-05-16 13:56:43.000	1,613	57,027	17	2023-05-16 13:56:43.000	21,7	35	3,673	8	2
18	2023-05-16 13:56:46.000	1,646	58,861	18	2023-05-16 13:56:46.000	21,7	39	2,517	14	2
19	2023-05-16 13:56:49.000	1,638	58,403	19	2023-05-16 13:56:49.000	21,7	38	2,491	26	2
20	2023-05-16 13:56:51.000	1,632	58,08	20	2023-05-16 13:56:52.000	21,6	37	2,477	36	2
21	2023-05-16 13:56:56.000	1,632	58,08	21	2023-05-16 13:56:56.000	21,6	37	2,346	47	2
22	2023-05-16 13:56:59.000	1,637	58,353	22	2023-05-16 13:56:59.000	21,4	37	1,939	52	1
23	2023-05-16 13:57:02.000	1,624	57,624	23	2023-05-16 13:57:02.000	21,6	36	1,957	65	1
24	2023-05-16 13:57:05.000	1,634	58,216	24	2023-05-16 13:57:05.000	21,5	37	1,854	81	1
25	2023-05-16 13:57:08.000	1,634	58,216	25	2023-05-16 13:57:08.000	21,5	37	1,686	87	1

Figura 51. Prueba de conexión remota a los registros de la base de datos.

Al compilar el programa compuesto por *nodos* mostrados en la estructura de la Figura 52, la interfaz gráfica se pudo visualizar tal y como se muestra en la Figura 53.

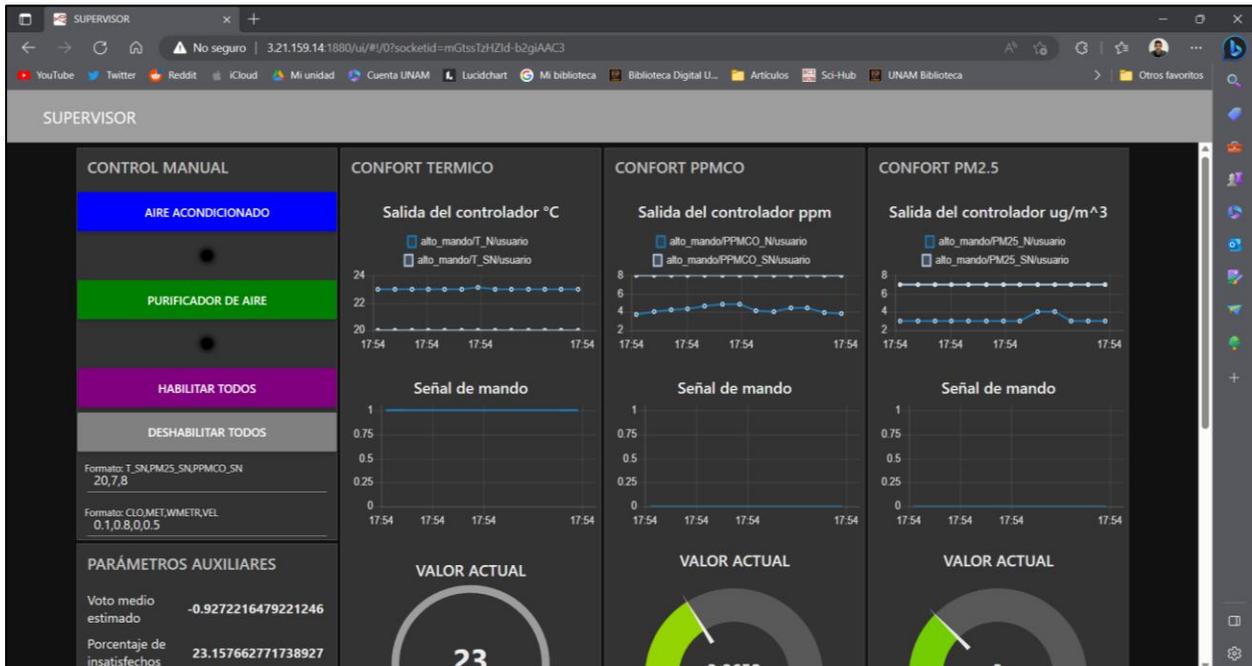


Figura 53. Prueba de conexión de la interfaz gráfica con el sistema, mostrando la primera mitad de la página del panel de control.

Así mismo, en la Figura 53 se muestran las pruebas efectuadas de comunicación con la red de *IoT*, para lo que se requiere solo del *enlace* para cualquier navegador de Internet con quien se desee compartir y autenticar la información de manera remota. Se observan tres gráficas con curvas en color azul y gris. Las del primer color representan las mediciones dentro de la habitación, las cuales deben estar próximas a las curvas de consigna deseada en color gris, en los límites permisibles establecidos en la sección 2.2.2. Esto quiere decir que el funcionamiento de los algoritmos de control, adquisición de datos y el de análisis de predicción por confort térmico, se visualizan en esta misma interfaz gráfica y revela el funcionamiento de los controladores y su desempeño por

medio del tiempo de establecimiento de los controladores. Cabe mencionar que las curvas en color gris son en esencia los valores del vector x_N , es decir las entradas del controlador, y su valor deseado se puede ingresar directamente en la sección de *control manual*, donde se muestra una cadena de texto ingresada con el valor $pauta_consigna = [20,7,8]$, de acuerdo con la estructura de la expresión 32, siendo que estos valores fueron escritos al azar sólo para corroborar que las curvas en color gris respondieran al ajuste por dichos valores deseados, lo cual se muestra que en efecto, cada recta se ajustó a ellos, tal y como se muestra en la Figura 53.

Por otro lado, el usuario puede ingresar también los valores del mensaje *pauta_comfort*, mostrado en la expresión 33, esto sirve para el algoritmo de la Figura 45, representado por el código en el Anexo F, en el que el dispositivo *supervisor* puede ejecutar dicho programa para recibir este arreglo como entrada de la función, y devolver los valores mostrados en la sección de *parámetros auxiliares* de la Figura 53. En las pruebas efectuadas, el dispositivo recibió el mensaje $pauta_comfort = [0.1,0.8,0,0.5]$ siendo esos valores ingresados al azar solo para verificar la correcta comunicación y que lograra arrojar los valores calculados de *PMV* y *PPD*. En cuanto a los valores de luminosidad L_N y humedad relativa H_N de los sensores, éstos se muestran por medio de los gráficos circulares de la Figura 54, mientras que también se pueden observar las comparativas entre los valores de consigna deseados por medio de los gráficos indicadores en color amarillo y las lecturas medidas dentro de la habitación por medio de los de color verde, esto sirve para visualizar de mejor manera que ambos medidores deben estar equilibrados, en referencia a las consignas deseadas.

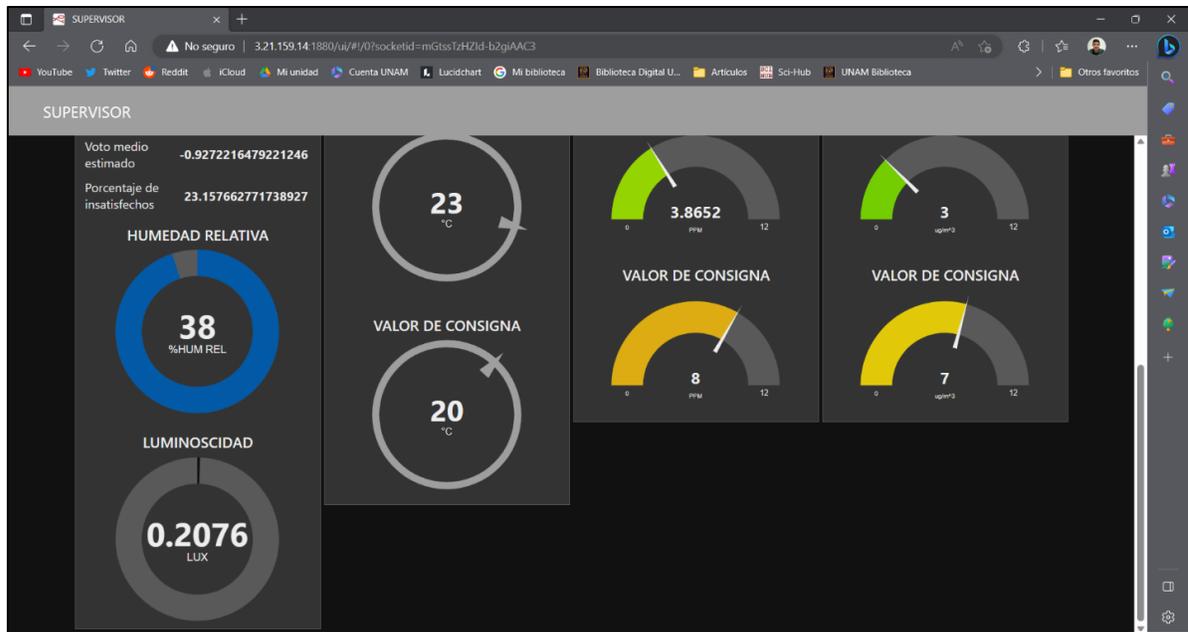


Figura 54. Prueba de conexión de la interfaz gráfica con el sistema, mostrando la segunda mitad de la página del panel de control.

En la Figura 53, en la sección del *control manual*, se muestran cuatro botones, el de color azul sirve para encender o apagar manualmente el aire acondicionado, en caso de no querer usar el controlador automático; mientras que el botón verde sirve para encender y apagar manualmente el extractor de aire. Los botones morado y gris sirven para encender al mismo tiempo ambos actuadores o desactivarlos en su totalidad, de igual manera en las pruebas realizadas. El control manual también funcionó correctamente y la orden fue recibida de manera inmediata.

Lo descrito anteriormente, fue pensado para uso del dispositivo *supervisor*, tal y como se esclarece en la Figura 35, pero para la participación del dispositivo *usuario*, el cual como elemento auxiliar se trata de un dispositivo móvil, tiene como función hacer lo mismo anteriormente mencionado, pero con una interfaz gráfica adecuada a él, con ayuda de la aplicación para programar bloques en la plataforma *Android*, [31]. La implementación del panel se muestra en la Figura 55.

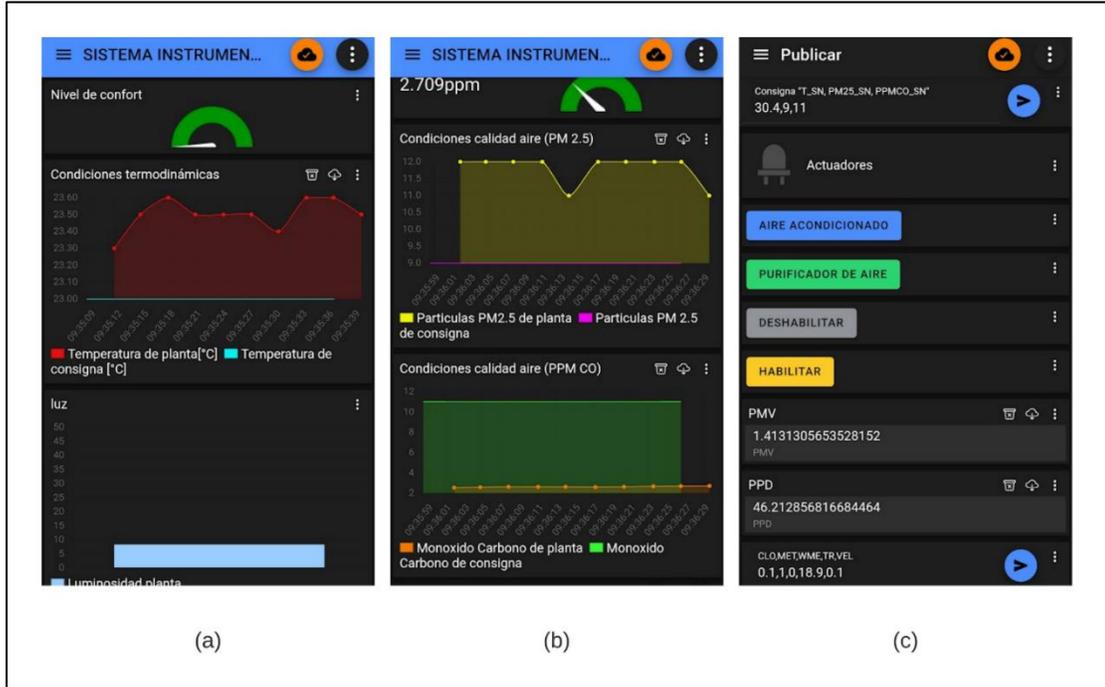


Figura 55. Panel de control para dispositivo móvil. (a) Panel de temperatura. (b) Panel de calidad del aire. (c) Panel de control manual.

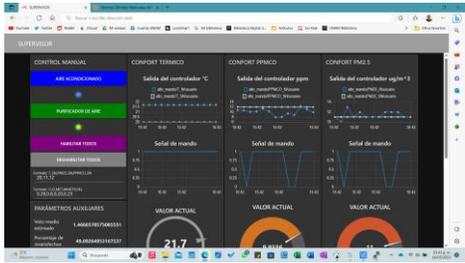
Finalmente, se efectuaron las pruebas de conexión con esta última interfaz gráfica, por lo que se finalizó formalmente con la construcción de todo el sistema de control e instrumentación. Cabe mencionar que el dispositivo *supervisor* y *usuario* son los únicos dispositivos pensados para participar en la red de *IoT*.

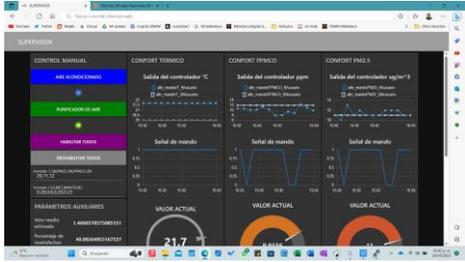
Capítulo 4: Ejecución del sistema de control e instrumentación

En este capítulo, se esclarece la ejecución de uso del sistema de instrumentación y confort para conservar las condiciones de confort dentro de un hogar, de donde se hicieron las pruebas. El objetivo es utilizar la interfaz gráfica ya previamente diseñada y verificar que los controladores y la instrumentación del sistema respondan correctamente a las perturbaciones térmicas y de calidad del aire dentro de cada una de las 3 habitaciones designadas a las pruebas de ejecución del sistema, con el objetivo de demostrar el buen funcionamiento. Por medio de la Tabla 15, se muestra un instructivo de uso recomendado, para usar correctamente el sistema dentro de un hogar.

Tabla 15. Instrucciones recomendadas para el uso correcto del sistema de instrumentación y control dentro de una habitación del hogar.

Paso	Descripción
1	Designar una habitación deseada dentro del hogar, para controlar condiciones de temperatura de confort y calidad de aire. 
2	Colocar la estructura que contiene los sensores, el aire acondicionado portátil y extractor de aire en un sitio adecuado dentro de la habitación. La tubería del extractor de aire, por motivos experimentales, se puede colocar sobrepuesta y con salida a la intemperie por medio de las ventanas de las habitaciones para expulsar el aire. Los sensores pueden colocarse descansando encima de una 

	<p>mesa, asegurando que tengan un buen campo de detección a través del flujo de aire dentro de la habitación.</p>	
3	<p>Conectar al tomacorriente doméstico los dos cables que energizan el aire acondicionado y extractor de aire, así como la fuente de poder que alimenta al conjunto de sensores. Automáticamente, la pantalla LCD mostrará un mensaje como <i>En línea</i> que indicará que se ha conectado a Internet correctamente.</p>	
4	<p>Encender el <i>Raspberry Pi</i>, y colocarlo de manera independiente en cualquier lugar de la casa, tratando que tenga buena ventilación ya que debe estar encendido todo el tiempo. El programa en <i>Python</i> que contiene se ejecutará automáticamente.</p>	
5	<p>Acceder al enlace http://3.21.159.14:1880/ui/ por medio de un navegador <i>web</i>. Si el acceso está autenticado, se mostrará en pantalla la interfaz gráfica mostrada. Se debe ingresar el valor deseado de temperatura y límites permisibles de calidad del aire en el formato de cadena de texto mostrado en la expresión 32. Se reitera que las unidades deben darse en [°C], [$\frac{\mu g}{m^3}$] y [ppm] respectivamente al formato, por ejemplo, ingresar la cadena de texto <i>21,12,10</i> indica al controlador los valores deseados. De acuerdo con las recomendaciones de la norma oficial mexicana para calidad del aire [5], se recomienda ingresar $12 \frac{\mu g}{m^3}$ y 11</p>	

	ppm como límites de una buena calidad del aire.	
6	De manera opcional, y solo por fines experimentales para el trabajo presente, se puede ingresar en la interfaz gráfica la cadena de texto en el formato de la expresión 33, es decir solo se necesita conocer el aislamiento de ropa del usuario y su tasa metabólica, con ayuda de las Tablas 11 y 12, en las unidades correspondientes en [clo] y [met]	

4.1 Experimentos en las habitaciones del hogar de pruebas

Usando las instrucciones de uso mostradas en la Tabla 15, se efectuaron los experimentos en cada una de las tres habitaciones del hogar donde se hicieron las pruebas. Las habitaciones se muestran en el plano de la Figura 56.

Se efectuaron dos experimentos de funcionamiento en cada una de las tres habitaciones, el primero es implementando el sistema de instrumentación, pero sin el sistema de control y el segundo experimento ya usando el sistema de control, de esta manera se puede ver la comparativa antes y después de la solución controlando las variables de confort y así demostrar el correcto funcionamiento y desempeño del sistema.

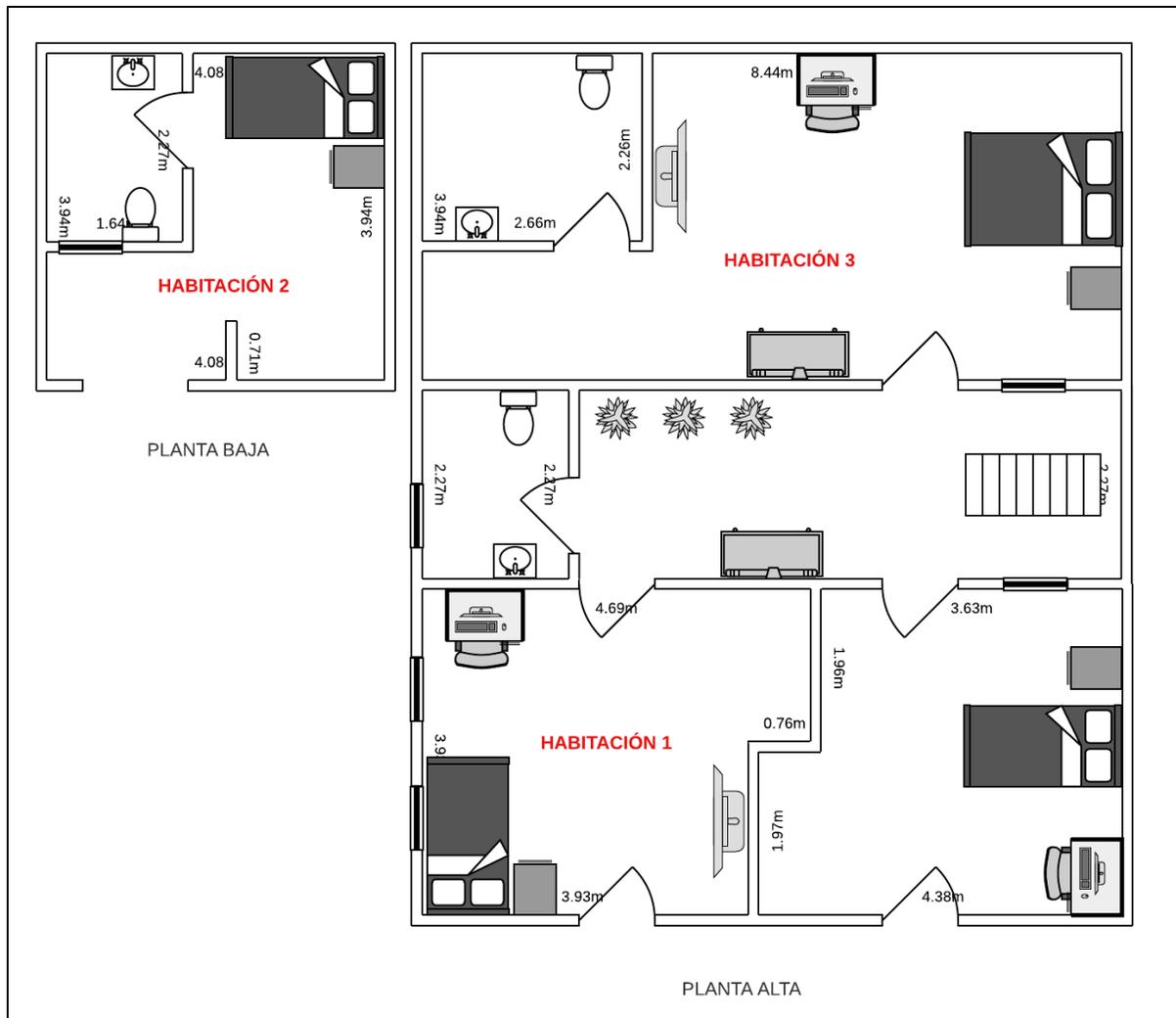


Figura 56. Plano del hogar de pruebas.

4.1.1 Habitación 1

Las pruebas hechas antes de la implementación del sistema de control para manipular el confort térmico y por calidad del aire arrojaron los resultados mostrados en las Figuras 57(a) y 57(c) respectivamente, en los periodos de tiempo mostrados para la habitación 1. Se puede observar que los puntos que indican las lecturas medidas por el sistema de instrumentación se hallaron fuera de las regiones de confort, esto para el caso de la temperatura. Para el caso de la calidad del aire, los

puntos se hallaron por debajo del límite permisible de 10 ppm de CO, mientras que para las partículas contaminantes tipo PM2.5, los puntos de lectura se hallaron por debajo y por encima del límite permisible. Esto significó que las condiciones iniciales de confort en esta habitación sugieren una incomodidad térmica y ligeramente por partículas contaminantes, aunque las concentraciones de CO fueron óptimas.

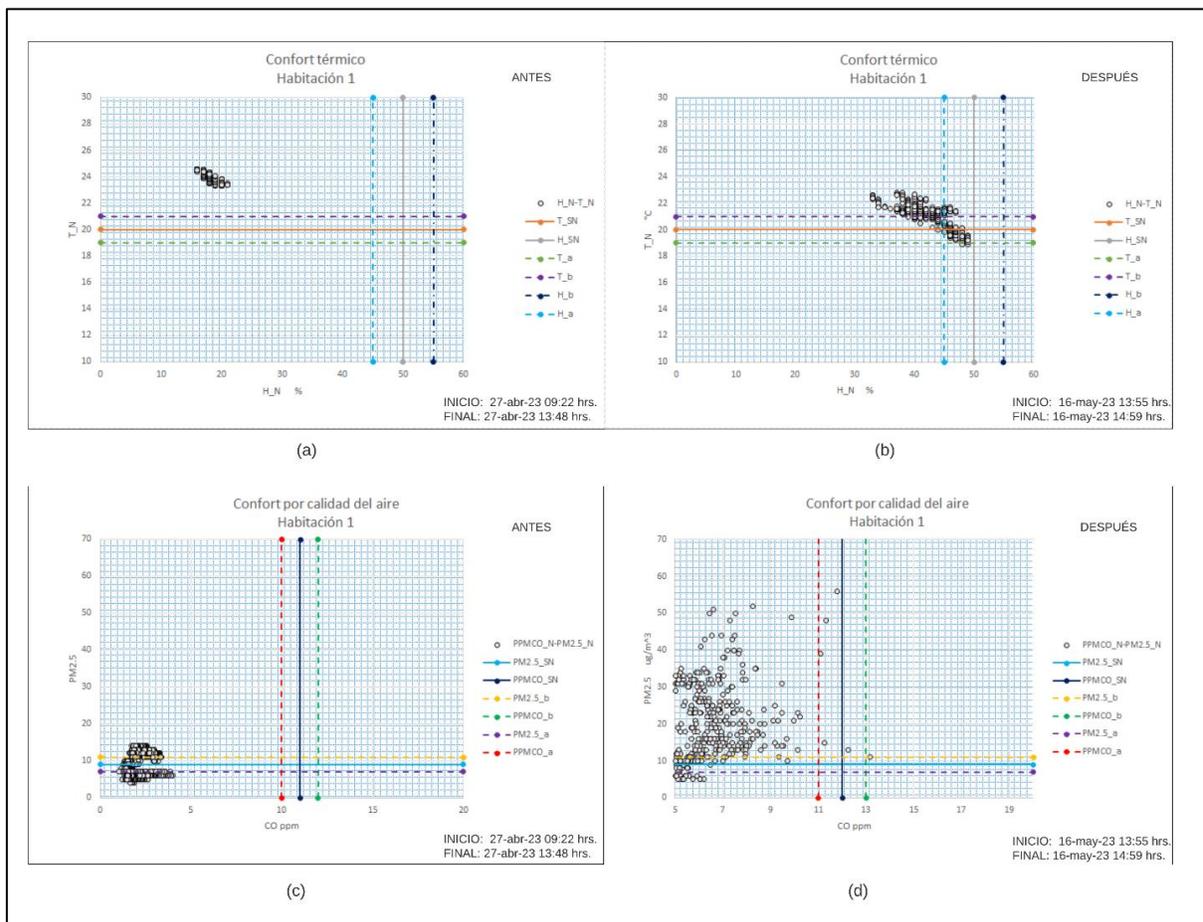


Figura 57. Pruebas efectuadas antes y después de la implementación del sistema de control en la habitación 1. (a) Situación de confort térmico antes. (b) Situación de confort térmico después. (c) Situación de confort por calidad del aire antes. (d) Situación de confort por calidad del aire después.

Posteriormente, se implementó el sistema de control para las variables de confort, donde los puntos de lectura para el confort térmico tuvieron un desplazamiento con tendencia a aglomerarse

a la región de confort térmico, como se observa en la Figura 57(b). Para el caso de la calidad del aire, se probó el comportamiento de su controlador en presencia de humo de cigarro, lo que provocó un aumento caótico de las partículas tipo PM2.5 y ppm de CO, pero el controlador logró no permitir un mayor aumento, provocando un aglomeramiento cercano a la región de confort por calidad del aire, tal como se muestra en la Figura 57(d). En cuanto al comportamiento de los controladores, la Figura 58(a) muestra como la consigna deseada fue de 21 °C y luego cambiada a 23 °C como se observa en el escalonamiento de la curva anaranjada, con el fin de visualizar que el sistema obedeció a las consignas deseadas ingresadas desde la interfaz gráfica, en donde el comportamiento térmico dentro de la habitación se ajustó dentro de la tolerancia permisible de temperatura, es decir la curva azul de lecturas medidas, logró acercarse a la consigna deseada. En cuanto a la calidad del aire, el humo de cigarro se indujo tres veces, esto se observa en el gráfico de la Figura 58(b), donde la tendencia de la curva en color azul fue disminuir para acercarse a la consigna deseada mostrada en la curva color naranja. Para el caso de las ppm de CO, la Figura 58(c) muestra que la concentración de este contaminante propio del humo de cigarro se comportó más caóticamente, aunque por debajo de la consigna deseada, en la mayoría de sus puntos.

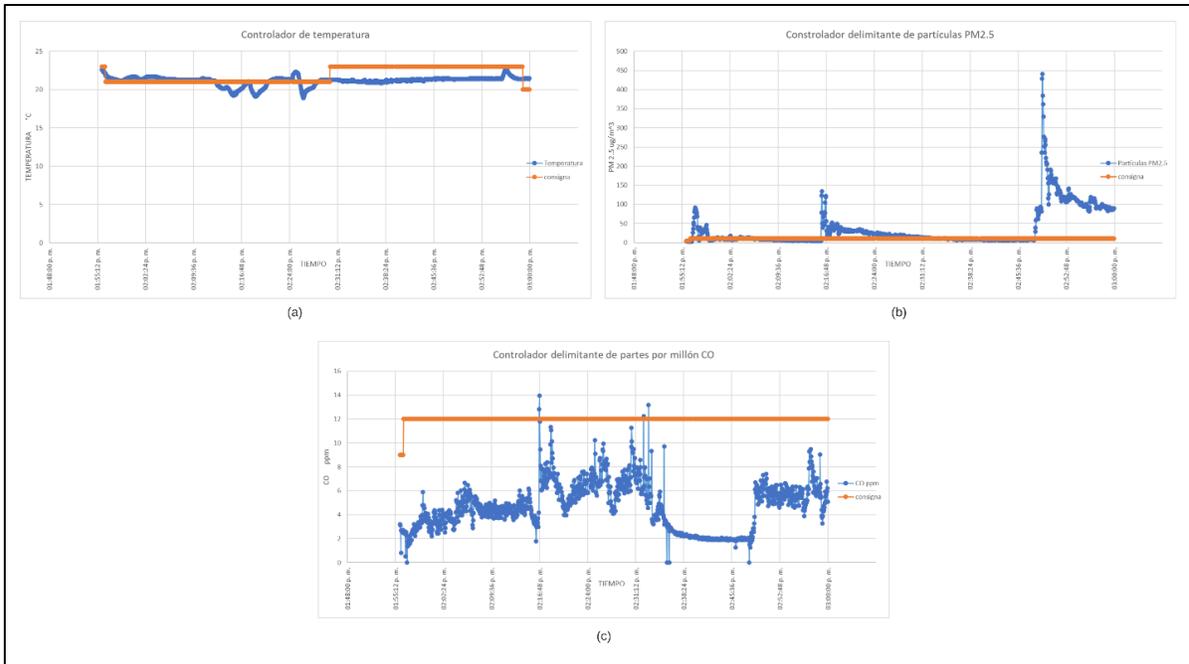


Figura 58. Comportamiento de los controladores de temperatura y calidad del aire en la habitación 1. (a) Control de temperatura. (b) Control de partículas PM2.5. (c) Control de partes por millón de CO.

4.1.2 Habitación 2

Las pruebas efectuadas en la habitación 2, tuvieron un comportamiento similar al de la habitación 1, aunque en esta habitación, la mayoría de los puntos se aglomeraron dentro de la región de confort por temperatura, tal como se muestra en el antes y después de la implementación del sistema de control, esta comparativa se muestra en las figuras 59(a) y 59(b). Para el caso de la calidad del aire, en un principio y de manera natural, se encontró en condiciones mejores que en la habitación 1, tal como se muestra en la Figura 59(c), donde todos los puntos están contenidos dentro de la región de confort. Posteriormente, se indujo humo de cigarro en esta habitación, para observar el comportamiento del controlador, y como se observa en la Figura 59(d), los niveles de

contaminación aumentaron caóticamente, pero al final se observa una aglomeración de los puntos en los límites de la región de confort, evidenciando el actuar del controlador de calidad del aire.

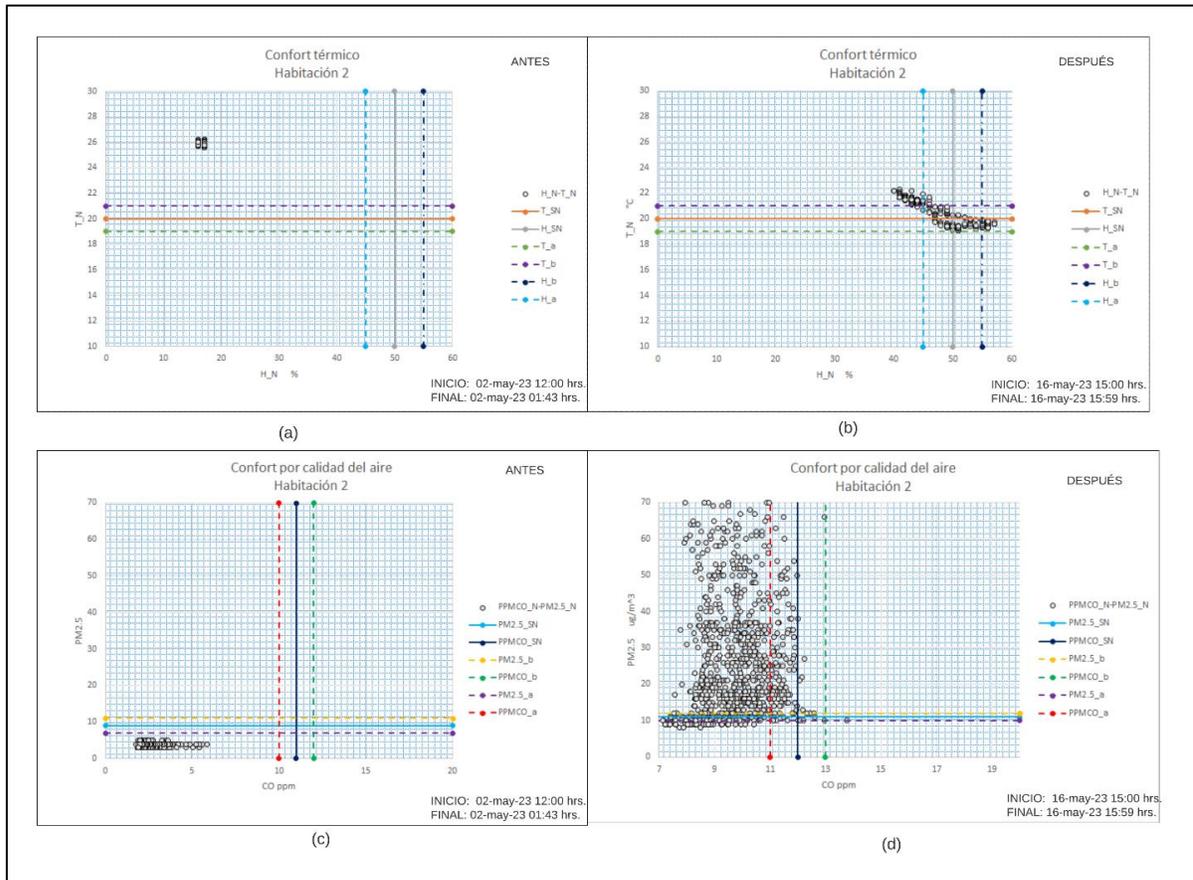


Figura 59. Pruebas efectuadas antes y después de la implementación del sistema de control en la habitación 2. (a) Situación de confort térmico antes. (b) Situación de confort térmico después. (c) Situación de confort por calidad del aire antes. (d) Situación de confort por calidad del aire después.

EL comportamiento del controlador de temperatura se muestra en la Figura 60(a), donde se observa que la curva de temperatura medida en color azul, tiende a parecerse a la curva de consigna de temperatura en color azul. En la Figura 60(b) se observa el comportamiento del controlador de partículas PM2.5, donde el humo de cigarro solo se indujo una vez, y su tendencia a disminuir debido al controlador, logra un comportamiento como el mostrado en la curva en color para

delimitarse con la curva de consigna deseada en color naranja. Para el caso de las ppm de CO, los aumentos de monóxido ya no fueron tan caóticos, pues a diferencia de la habitación 1, el humo de cigarro solo se indujo una vez, es decir había menos contaminación por controlar.

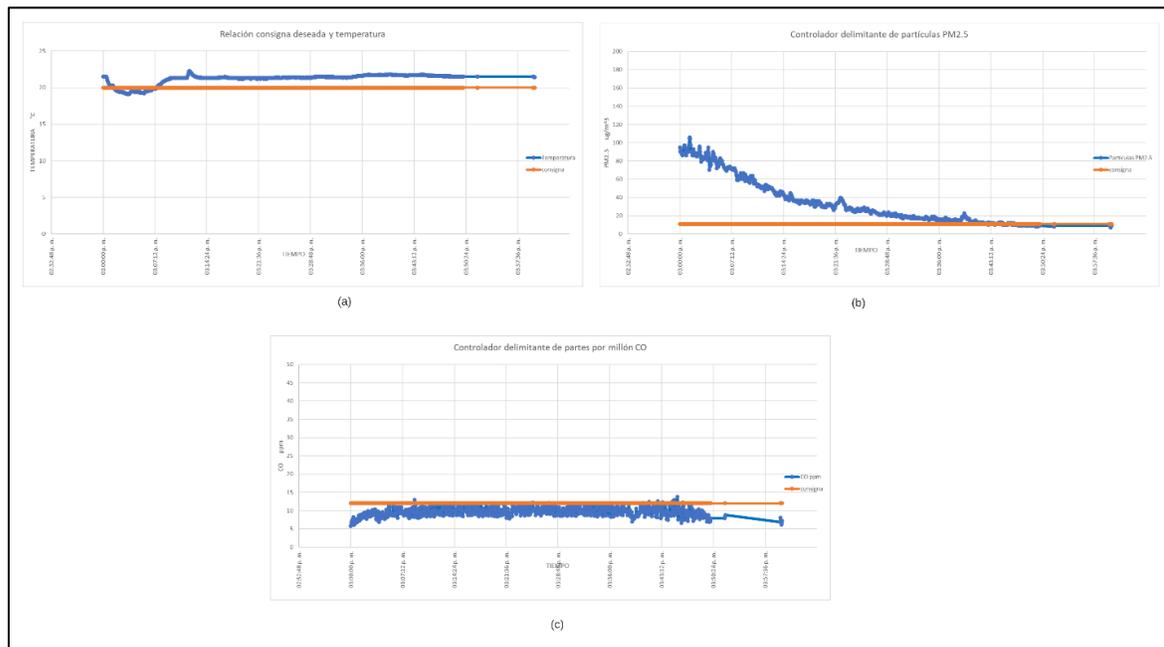


Figura 60. Comportamiento de los controladores de temperatura y calidad del aire en la habitación 2. (a) Control de temperatura. (b) Control de partículas PM2.5. (c) Control de partes por millón de CO.

4.1.3 Habitación 3

Las pruebas efectuadas en la habitación 3 tuvieron un comportamiento muy diferente al de las habitaciones 1 y 2, de hecho, como se observa en el plano de la Figura 56, este sitio es más grande con una superficie de terreno de $27.302 m^2$, por lo que los actuadores de aire acondicionado y extractor de aire tuvieron una cierta dificultad para controlar las variables de confort. En las Figuras 61(a) y 61(b) se muestra la comparativa del antes y después de la implementación del sistema de control de confort térmico, donde se observa como los puntos se aglomeraron cercanamente a la

región de confort térmico, pero sin llegar a contenerse dentro de ella. Esto indicó que de los 26°C que habían dentro de esta habitación, solo se logró disminuir la temperatura a los 22°C aproximadamente, pero nunca a las 20 °C deseados e indicados al controlador por medio de la interfaz gráfica. En cuanto a la calidad del aire, en esta habitación no se indujo contaminación por humo de cigarro, sino se observó el polvo natural propio dentro de ella, por lo que en las Figuras 61(c) y 61(d) se muestra la comparativa del antes y después de la implementación del sistema de control de calidad del aire, donde solo se observó una aglomeración de puntos, disminuyendo ligeramente las ppm de CO, pero las partículas PM2.5 naturales dentro de la habitación, no disminuyeron tanto, revelando óptimas condiciones de calidad del aire en esta habitación.

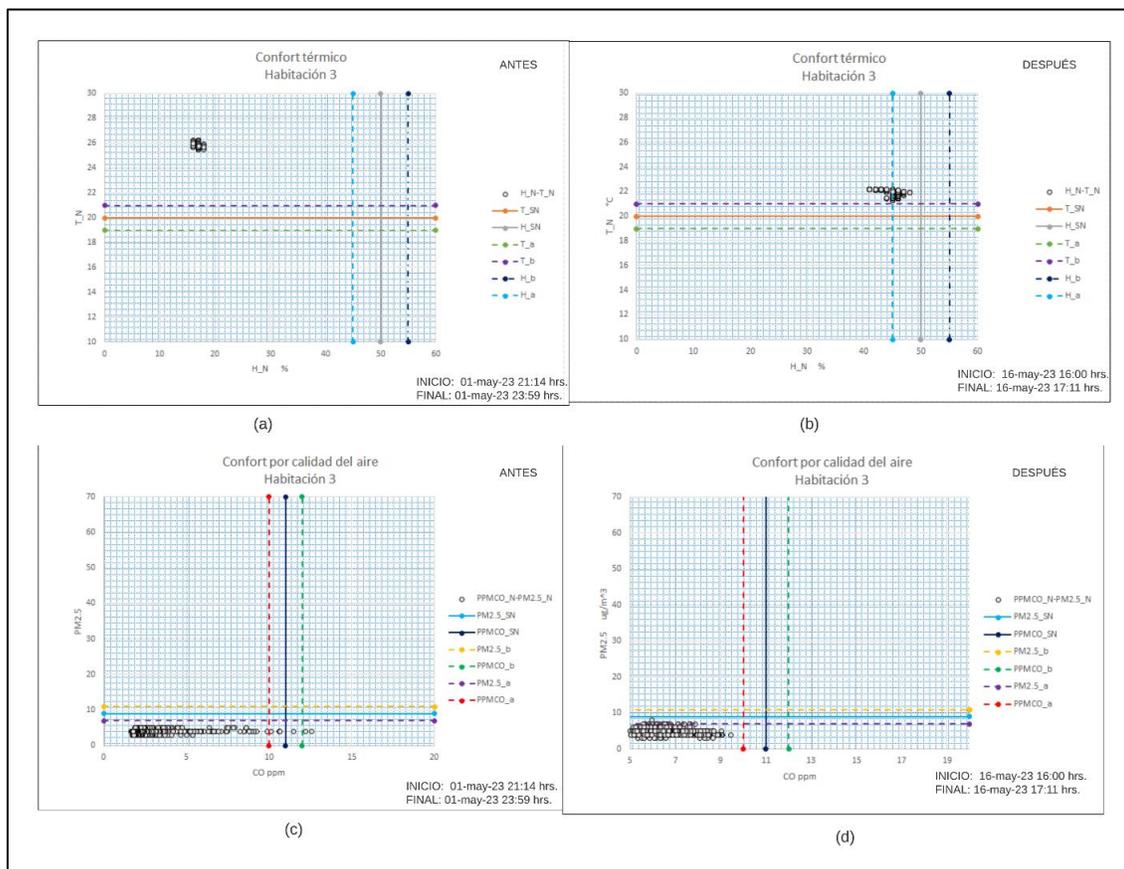


Figura 61. Pruebas efectuadas antes y después de la implementación del sistema de control en la habitación 3. (a) Situación de confort térmico antes. (b) Situación de confort térmico después. (c) Situación de confort por calidad del aire antes. (d) Situación de confort por calidad del aire después.

El comportamiento del controlador de temperatura se muestra en la Figura 62(a), donde la curva azul tiende a acercarse a la curva deseada en color naranja, pero con un ancho constante en todo el recorrido del tiempo, a diferencia de las habitaciones 1 y 2, no hubo un cruce o un acercamiento más significativo. El comportamiento del controlador de partículas PM2.5 se muestra en la Figura 62(b), y se observa como la curva azul se mantuvo por debajo de la curva naranja en todo el recorrido de tiempo, revelando que las partículas se encontraron por debajo del límite deseado. Esto mismo ocurrió con el controlador de ppm de CO, mostrado en la Figura 62(c), pero de igual manera, las lecturas se comportaron por debajo del límite permitido, reiterando que en esta habitación no se indujo contaminación alguna en el aire.

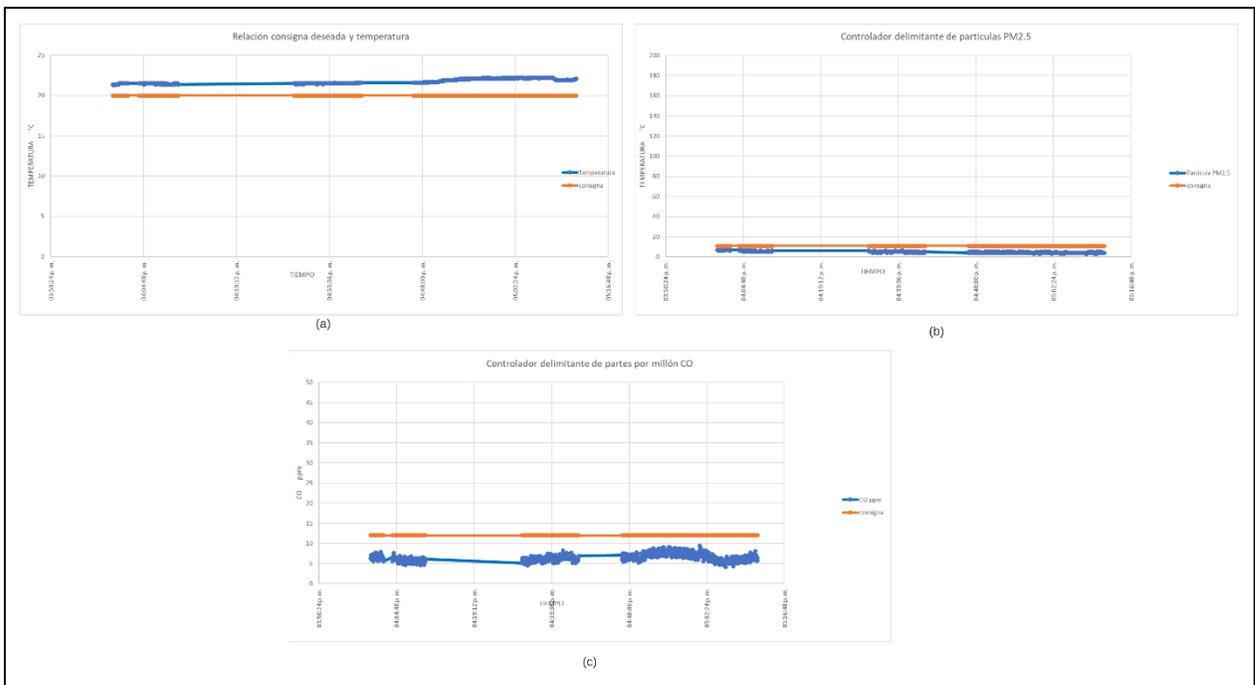


Figura 62. Comportamiento de los controladores de temperatura y calidad del aire en la habitación 3. (a) Control de temperatura. (b) Control de partículas PM2.5. (c) Control de partes por millón de CO.

4.2 Conclusiones de los datos obtenidos

Con ayuda de los experimentos efectuados en las habitaciones, y usando el algoritmo de predicción de confort térmico, así como el uso de su escala mostrada en la Tabla 9, se obtuvieron los gráficos mostrados en la Figura 63, uno por habitación.

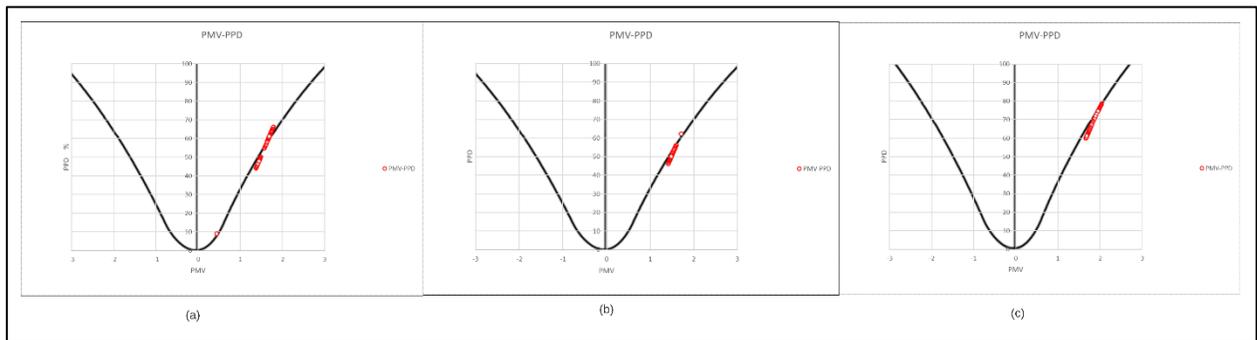


Figura 63. Ejecución del algoritmo de predicción de confort térmico basado en la técnica de la norma ISO 7730. (a) Predicción para habitación 1. (b) Predicción para habitación 2. (c) Predicción para habitación 3.

En la Figura 63(a) correspondiente a la habitación 1, se puede observar una transición de puntos recorriendo la campana hacia el valor de $PPD \approx 42\%$ (eje de las ordenadas), en una escala de confort térmico cercana al valor adimensional de $PMV \approx 1.4$ (eje de las abscisas), lo que revela que el 42% de las personas (de haberlas dentro de la habitación), sentirá una insatisfacción térmica ligeramente calurosa, mientras que el resto sentirá la satisfacción térmica.

En la Figura 63(b) correspondiente a la habitación 2, se puede observar que la transición de puntos a lo largo de la campana se detuvo en el valor cercano a $PPD \approx 46\%$ y en $PMV \approx 1.5$, donde la predicción para esta habitación es que de haber personas contenidas en dicha habitación, el 46% sentirá insatisfacción térmica por sensación térmica ligeramente calurosa.

En cuanto a la Figura 63(c) correspondiente a la habitación 3, y siendo de las tres, la habitación más grande, el desempeño del sistema de acondicionamiento de aire fue menor, ya que la transición de los puntos se detuvo en $PPD \approx 60\%$ con un valor aproximado en la escala de $PMV \approx 1.8$, que dentro de la escala significa insatisfacción térmica un poco calurosa para el 60% de las personas dentro de ella.

Los porcentajes de población en la predicción térmica para las habitaciones 1 y 2 fueron menores al 50%, por lo que se puede afirmar que el desempeño de sistema de control de confort térmico tuvo un desempeño permisible, aunque lo ideal era que la transición de puntos pudieran recorrer más la campana para acercarse a su vértice, lo que indica que el sistema se puede mejorar implementando otro actuador de aire acondicionado más potente o con otro tipo de principio de funcionamiento al de enfriamiento evaporativo. En cuanto a la habitación 3, esta superó más de la mitad de las personas insatisfechas por confort térmico, lo que se puede atribuir a que el actuador del sistema de control no tiene un buen desempeño en habitaciones más grandes.

Por otro lado, se observó que el ruido térmico inducido por la iluminación influyó un poco, aunque no de manera tan significativa, tal como se observa en la Figura 64, cada gráfico correspondiente a cada habitación, donde los puntos se aglomeran más a la derecha que a la izquierda, debido a que los luxes son más altos y por ende aumenta más la temperatura, esto puede revelar que el desempeño del aire acondicionado puede verse afectado por el calor radiante de los focos incandescentes.

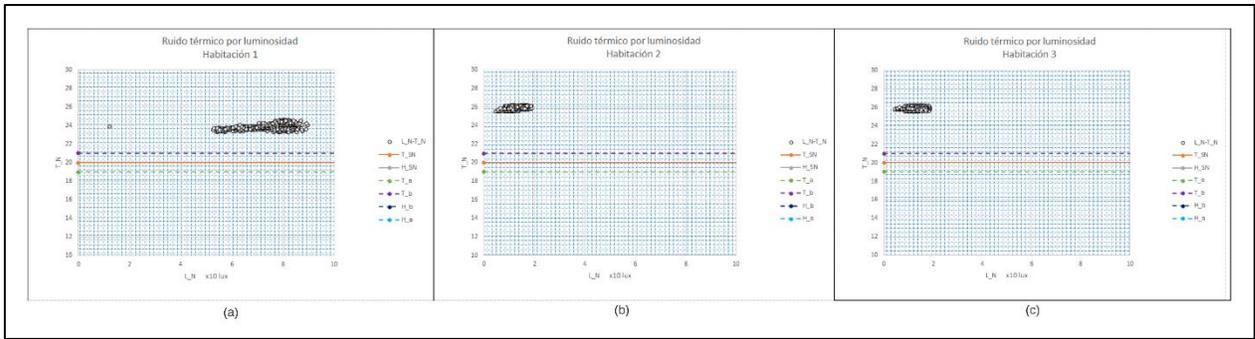


Figura 64. Influencia de la luminiscencia en la temperatura dentro de cada una de las habitaciones. (a) Habitación 1. (b) Habitación 2. (c) Habitación 3.

Finalmente, en cuanto a la calidad del aire dentro de las habitaciones, se observó un buen desempeño del extractor de aire, pues al momento de detectar los contaminantes el controlador se adecuó a los límites permisibles y designados por el usuario, por lo que las partículas y gases contaminantes de CO se lograron disminuir extrayendo el aire de la habitación, aunque de igual manera, el funcionamiento de este actuador en habitaciones más grandes podría ser diferente, ya que el contaminante podría ser más difícil de detectar en altas concentraciones debido a la disolución en espacios más grandes, por lo que en habitaciones pequeñas existe mayor concentración, tal como ocurrió en las habitaciones 1 y 2.

4.3 Trabajos a futuro

La arquitectura de la red de *IoT* provee múltiples maneras de analizar los datos obtenidos, esto es la etapa de su semántica. En el trabajo presente se empleó una técnica de predicción de datos basada en la norma *ISO 7730*, pero algunos autores como [2] discuten que esta norma puede tener como

alternativa, otro tipo de predicción basada en técnicas de identificación de patrones de preferencias de confort del usuario, por lo que el trabajo presente podría implementar otro tipo de análisis basado en inteligencia artificial para poder entender de mejor manera el historial de los datos arrojados. De hecho, a la implementación de estas técnicas con el *IoT*, se le conoce como *AIoT*, debido a que su semántica es analizada con algún tipo de algoritmo inteligente. De este modo, otro tipo de controladores llamados *controladores inteligentes* podrían aprender de los patrones de uso de una persona en cuanto al confort térmico dentro de su hogar o inclusive, de calidad del aire.

Referencias

- [1] World Health Organization, «WHO compendium of innovative health technologies for low-resource settings,» 2021.
- [2] F. I. Vázquez y W. Kastner, «Thermal Comfort Support Application for Smart Home Control,» de *Ambient Intelligence - Software and Applications*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2012, pp. 109-118.
- [3] World Health Organization, «Coronavirus disease (COVID-19): Ventilation and air conditioning [Q&A],» 29 julio 2020. [En línea]. Available: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019/question-and-answers-hub/q-a-detail/coronavirus-disease-covid-19-ventilation-and-air-conditioning>.
- [4] Normalización Española, «Ergonomía del ambiente térmico. Determinación analítica e interpretación del bienestar térmico mediante el cálculo de los índices PMV y PPD y los criterios de bienestar térmico local (UNE-EN ISO 7730),» 2006.
- [5] Comisión Federal para la Protección contra Riesgos Sanitarios, «Normas Oficiales Mexicanas (NOM) de Calidad del Aire Ambiente,» gob.mx, [En línea]. Available: <https://www.gob.mx/cofepris/acciones-y-programas/4-normas-oficiales-mexicanas-nom-de-calidad-del-aire-ambiente>. [Último acceso: 8 Abril 2023].
- [6] V. C. Shruti, G. Kutralam-Muniasamy, F. Pérez-Guevara, P. D. Roy y I. E. Martínez, «Occurrence and characteristics of atmospheric microplastics in Mexico City,» *The Science of the total environment*, vol. 847, n° 157601, p. 157601, 2022.

- [7] T. Hirschfeld, M. J. Block y W. Mueller, «Virometer: An optical instrument for visual observation, measurement and classification of free viruses,» *The Journal of Histochemistry and Cytochemistry*, vol. 25, nº 7, pp. 719-723, 1977.
- [8] C. Ang, «Cómo se compara el coronavirus con las partículas más pequeñas del mundo,» Foro Económico Mundial, 16 Octubre 2020. [En línea]. Available: <https://es.weforum.org/agenda/2020/10/asi-es-como-el-coronavirus-se-compara-con-las-particulas-mas-pequenas-del-mundo/>. [Último acceso: 9 Abril 2023].
- [9] S. Soria Tello, *Sistemas automáticos industriales de eventos discretos*, 1 ed., Alfaomega, 2013.
- [10] N. S. Nise, *Control Systems Engineering*, 6 ed., John Wiley & Sons, 2010.
- [11] R. Pallás Areny, *Sensores y acondicionadores de señal*, Alfaomega, 2007.
- [12] J. P. Tello Portillo, *Introducción a las señales y sistemas*, Barranquilla: Universidad del Norte, 2017.
- [13] A. Creus, «Instrumentación industrial,» de *Instrumentación industrial*, 8 ed., Alfaomega, 2010.
- [14] M. Komarudin, H. D. Septama, T. Yulianti, A. Yudamson, M. Pratama y T. P. Zuhelmi, «Air quality monitoring device for smart health solution during covid-19 pandemic,» de *2021 International Conference on Converging Technology in Electrical and Information Engineering (ICCTEIE)*, 2021.
- [15] B. O. Ganazhapa, *Arduino: Internet de las Cosas*, Alfaomega, 2021.

- [16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari y M. Ayyash, «Internet of things: A survey on enabling technologies, protocols, and applications,» *IEEE Communications Surveys & Tutorials*, vol. 17, nº 4, pp. 2347-2376, 2015.
- [17] B. K. Moharana, P. Anand, S. Kumar y P. Kodali, «Development of an IoT-based real-time air quality monitoring device,» de *2020 International Conference on Communication and Signal Processing (ICCSPP)*, 2020.
- [18] A. Silberschatz, H. F. Korth y S. Sudarshan, *Database System Concepts*, 6 ed., New York, NY: McGraw-Hill Professional, 2010.
- [19] INEGI, «Encuesta nacional de vivienda 2020,» 2020.
- [20] INEGI, «Promedio de ocupantes en viviendas particulares habitadas por entidad federativa, serie de años censales de 1995 a 2020,» 2020.
- [21] Y. A. Cengel, *Termodinámica*, 9 ed., McGraw Hill, 2019.
- [22] ManualsLib, «ARCTIC AIR ULTRA GUÍA DEL PRODUCTO,» ManualsLib, [En línea]. Available: <https://www.manualslib.es/manual/567638/Arctic-Air-Ultra.html>. [Último acceso: 4 Noviembre 2022].
- [23] ATECYR, *Fundamentos de climatización para instaladores e ingenieros recién titulados*, Madrid: ATECYR, 2010.
- [24] UNIT ELECTRONICS, «MQ-7 Detector de Monóxido de Carbono,» 2023. [En línea]. Available: <https://uelectronics.com/wp-content/uploads/2018/01/MQ-7.pdf>. [Último acceso: 4 Noviembre 2022].

- [25] Espressif, «ESP32-WROOM-32 Datasheet. [PDF],» [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf. [Último acceso: 4 Noviembre 2022].
- [26] F. R. Cortés y J. C. Monjaraz, Arduino: aplicaciones en robótica, mecatrónica e ingenierías, Alfaomega, 2015.
- [27] L. G. Corona Ramírez, G. S. Abarca Jiménez y J. Mares Carreño, Sensores y actuadores, México: Patria, 2015.
- [28] «DHT11 Humidity & Temperature Sensor,» [En línea]. Available: <https://www.mouser.mx/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>. [Último acceso: 4 Septiembre 2022].
- [29] A. Olsson, «GL5537/Light Dependent Resistor (LDR),» KTH Royal Institute of Technology, 2013. [En línea]. Available: <https://www.kth.se/social/files/54ef17dbf27654753f437c56/GL5537.pdf>. [Último acceso: 4 Noviembre 2022].
- [30] Plantower, «PMSA003 Laser PM2.5 Sensor-Plantower Technology,» Plantower.com, [En línea]. Available: https://www.plantower.com/en/products_33/77.html. [Último acceso: 4 Septiembre 2022].
- [31] R. Kundu, «IoT MQTT Panel,» versión 1.1.2, Plataforma Android, 2019. [En línea]. Available: <https://play.google.com/store/apps/details?id=snr.lab.iotmqttpanel.prod>.
- [32] Diario oficial de la federación, «DECRETO por el que se reforma el artículo 311 y se adiciona el capítulo XII Bis de la Ley Federal del Trabajo, en materia de Teletrabajo.,» 11

enero 2021. [En línea]. Available:
https://dof.gob.mx/nota_detalle.php?codigo=5609683&fecha=11/01/2021.

- [33] D. A. Baller, Interviewee, *Episode #36 - Safe care at home [podcast]*. [Entrevista]. 30 abril 2021.
- [34] Secretaría del trabajo y previsión social, «Guía de acción para los centros de trabajo ante el COVID-19,» 2020.
- [35] L. Socconini y R. Carlo, *Lean Six Sigma : sistema de gestión para liderar empresas*, 1 ed., Alfaomega, 2020.
- [36] K. Ogata, *Ingeniería de control moderna*, 5 ed., Prentice Hall, 2010.
- [37] E. Espinosa, «¿Cómo ayuda el internet de las cosas en temas de industria y sostenibilidad?,» *Expansión*, 9 Noviembre 2022. [En línea]. Available: <https://expansion.mx/opinion/2022/11/09/internet-de-las-cosas-temas-industria-sostenibilidad>. [Último acceso: 10 Abril 2023].
- [38] J. G. B. Saldaña, C. del Carmen Gutiérrez Torres y J. A. J. Bernal, *Termodinámica para ingenieros*, Grupo Editorial Patria, 2015.
- [39] Adafruit Industries, «Adafruit PMSA003I Air Quality Breakout,» [Adafruit.com](https://www.adafruit.com/product/4632), [En línea]. Available: <https://www.adafruit.com/product/4632>. [Último acceso: 4 Septiembre 2022].
- [40] Texas Instruments, «TXS0108E 8-Bit Bidirectional Voltage-Level Translator with Auto-Direction Sensing and ± 15 -kV ESD Protection,» Texas Instruments, 2016. [En línea]. Available: <https://www.ti.com/lit/ds/symlink/txs0108e.pdf>.

- [41] Node-RED, «Running on Amazon web services,» Node-RED, [En línea]. Available: <https://nodered.org/docs/getting-started/aws>. [Último acceso: 15 Febrero 2023].
- [42] Raspberry Pi Ltd, «Raspberry pi OS,» Raspberry Pi, [En línea]. Available: <https://www.raspberrypi.com/software/>. [Último acceso: 1 Diciembre 2022].

Anexo B

Programa en C para calibración de sensor MQ7

```
/*-----*/
/*-----CALIBRACIÓN MQ7-----*/
/*-----*/

const float RL = 1.0; //Valor de la resistencia de carga en kohm (la que está soldada a la placa)
const float Rs_Ro_aire = 28.0; //Relación Rs/Ro del aire, aproximadamente 28, de acuerdo al grafico del
Datasheet
const float Vc = 5.0; //Suministro del sensor (Fuente externa)
const float V_esp32 = 3.3; //Voltaje de resolución de ADC del MCU para la conversión de lectura analogica
a voltaje
const float resolucion_esp32 = 4095.0; //Resolución de lectura analógica del ESP32 para la conversión a
voltaje (1023 para arduino, 4095 para ESP32)
float Rs_medida; //Variable de la resistencia de sensor a cualquier sustancia, para este caso de
calibración y detectará gas aire en kohm y es la medida por nosotros
float Ro_medida; //Variable de la resistencia de sensor a 100 ppm de concentración de CO en el aire, en
kohm y es la medida por nosotros
float ARL = 0.0; //Variable de la lectura analógica. Como el ESP32 es de 12 bits y por tanto, un
omtervalo de [0-4095]-->[0-3.3 Volts]
float VRL; //Variable del voltage de la resistencia de carga RL. La conversion a este voltaje se obtiene
de lectura_analogica
int x; //Numero de muestras

void setup()
{
  Serial.begin(115200);
}

void loop()
{
  /*-----Inicia bucle de samples para 100 muestras y luego se promedia el resultado-----*/
  for(x = 0; x < 100; x++)
  {
    ARL = ARL + analogRead(34);
  }
  ARL = ARL/100;
  /*-----Luego se convierte la lectura analogica promediada a voltaje-----*/
  VRL = (ARL / resolucion_esp32) * V_esp32;
  /*-----Se calcula la resistencia del sensor Rs que medirá las ppm de aire-----
  */
  /*-----Recordar que sus unidades son el kohms-----
  */
  Rs_medida = RL * ((Vc - VRL) / VRL);
```

```
/*-----Se calcula la Ro medida ----- */
Ro_medida = Rs_medida / Rs_Ro_aire;
/*-----Se imprime la información de la calibración-----
--*/
Serial.print("Lectura analogica ARL: ");
Serial.print(ARL);
Serial.print(" Voltaje VRL: ");
Serial.print(VRL);
Serial.print(" Resistencia Ro medida en kohm: ");
Serial.print(Ro_medida);
Serial.print(" Numero de muestras: ");
Serial.println(x);
delay(1000);
}
```

Anexo C

Código en C para ESP32 (función de conexión al *broker* MQTT).

```
#include <Arduino.h>
#include <PubSubClient.h> //Librería para poder utilizar funciones y clases de la comunicación MQTT para
publicar y recibir mensajes, suscribirse a topics, etc.
#include<WiFi.h> //Librería Open Source para poder utilizar funciones y clases relacionadas con la
configuración de red WiFi del h
//-----Definición de autenticación de seguridad MQTT
-----
//const char* user_mqtt = Roberto;
//const char* pass_mqtt = 1234;
//-----Definición de objetos para el ESP32 WiFi -----
-----
WiFiClient espClient; //Define objeto a la variable 'espClient'
PubSubClient client(espClient); //Define objeto a la variable 'client', heredado de 'espclient'

void reconnect() //Función que intenta reconectar al broker en caso de desconexión al MQTT
{
  while (!client.connected()) //mientras el cliente este conectado ejecutar el bucle
  {
    //Serial.print("Intentando conexión MQTT...");
    if (client.connect("ESP32 Client")) //if (client.connect("ESP32 Client" user_mqtt, pass_mqtt))
    {
      //Serial.println("SARGENTO EN LINEA");
      client.subscribe("teniente/sargento"); //Intenta suscribirse al topic de suscripción. Topic
referente al ESP32 como cliente para recibir mensaje de su servidor, el Raspberry
    }
    else
    {
      //Serial.print("Falló, rc=");
      //Serial.print(client.state());
      //Serial.println(" Inténtelo de nuevo en 5s");
      delay (5000);
    }
  }
}
```

Código en C para ESP32 (función de conexión a red de área local).

```
#include <Arduino.h>
//*****
***** ***** Región directivas de preprocesador *****
```

```

//*****
*****
//-----Definición de librerías a utilizar-----
#include<WiFi.h> //Librería Open Source para poder utilizar funciones y clases relacionadas con la
configuración de red WiFi del hardware ESP32
//-----Definición de variables del SSID de la Red
Local -----
const char* ssid = "Totalplay-20A5_2.4Gnormal";
const char* password = "kEplEr2014360507";
//-----El objetivo de esta funcion es colocar los valores de la red local-----
-----
void setup_wifi() //Función auxiliar de setup de WiFi. Indicará si la conexión es o no ha sido exitosa
{
    delay(10);
    //Serial.println("");
    //Serial.println("Iniciando conexión WiFi a la red local: ");
    //Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) //Mientras la conexión WiFi se mantenga hacer el bucle.
    {
        delay(500);
        //Serial.print(".");
    }
}

```

Código en C para ESP32 (función de conexión a la red de IoT como suscriptor).

```

#include <Arduino.h>

#define A3 18 //Define GPIO 18 Para Relé de Aire Acondicionado
#define A4 19 //Define GPIO 19 Para Relé de Purificador de Aire
int A_N1 = 0; //Actuador de Aire Acondicionado. Inicialmente está apagado.
int A_N2 = 0; //Actuador de Purificador de Aire. Inicialmente está apagado.
int anterior;
//-----Se reciben instrucciones del teniente-----
//-----Función auxiliar teniente() (SUSCRIBER, ESP32 recibe datos de su servidor)-----

void teniente(char* topic, byte* message, unsigned int length) //Funcion auxiliar de ESP32 como cliente en
la red MQTT. Requiere el nombre del topic, el mensaje cifrado como suscriptor y su longitud
{
    String instrucciones = "";
    //Serial.print("INSTRUCCIONES de TENIENTE: ");

    for (int i = 0; i < length; i++)
    {
        instrucciones += (char) message[i];
    }
}

```

```

sscanf(instrucciones.c_str(), "%d,%d", &A_N1, &A_N2);
//Serial.println(instrucciones);
anterior = A_N2;
//-----
//-----Region de actuadores o elementos finales de control-----
//-----Output GPIO 12 del encendido y apagado de actuador-----
    if (String(topic = "teniente/sargento")) //Si recibe mensaje sobre topic 'teniente/sargento',
verifica si es "on" u "off". En referencia al ESP32 como cliente suscriptor para recibir mensajes de sus
servidor, el Raspberry
    {
        //Serial.print("Cambio de salida: ");
        if (A_N1 == 1)
        {
            //Serial.println("Aire Acondicionado ON");
            digitalWrite(A3, HIGH);
        }
        else if(A_N1 == 0)
        {
            //Serial.println("Aire Acondicionado OFF");
            digitalWrite(A3, LOW);
        }
        //Serial.print("Cambio de salida: ");
        if (A_N2 == 1)
        {
            //Serial.println("Purificador ON");
            digitalWrite(A4, HIGH);
        }
        else if(A_N2 == 0)
        {
            if (anterior == 1)
            {
                delay(7000); //retardo al apagado
                //Serial.println("Purificador OFF");
                digitalWrite(A4, LOW);
            }
            else
            {
                //Serial.println("Purificador OFF");
                digitalWrite(A4, LOW);
            }
        }
    }
}
//-----fin de esa región-----
}

```

Código en C para ESP32 (función principal del programa).

```
/**Funciones del sargento y cabos (programa principal)
//1.-La función de los Cabos es solo poder dar parte de las Novedades a su Sargento y acatar ordenes de
Instrucciones. Solo pueden comunicarse con su Sargento. (Protocolo I2C o UART al ESP32)
//2.- Sargento recibe Novedades de los Cabos, que son los sensores y actuadores. Sargento empaqueta el
mensaje de Novedades y los guarda en una array tipo String
//3.- Sargento se autentica con el Teniente y le envía las Novedades (MQTT connect y MQTT publish)
//4.- Si el sargento no logró comunicarse con el Teniente (Desonexión con Broker o a la IP), retorna al
punto 1.-
//5.- Sargento recibe mensaje de Instrucciones (MQTT SUSCRIBER), decifra el mensaje y el vector obtenido
es un array de dos componentes para los Cabos de Actuadores
//6.- Sargento manda la orden a los Cabos de actuadores para encenderse o apagarse
//7.- Se retorna al punto 1.-
//*****
//***** Región directivas de preprocesador *****
//*****
//-----Definición de librerías a utilizar-----
#include <setup_wifi.hpp> //Libreria creada por Roberto
#include <reconnect.hpp> //Libreria creada Roberto
#include <teniente.hpp> //Libreria creada por Roberto
#include <Arduino.h> //Libreria para usar metodos de C
#include <math.h> //Librería para poder utilizar operadores matematicos como 'pow()' o 'cos()'
#include <DHT.h> //Librería para poder utilizar funciones especiales del sensor DHT 11 como 'dht.read()'
#include <Adafruit_Sensor.h> //Libreria para poder usar metodos de sensores de las marcas Adafruit
#include <PubSubClient.h> //Librería para poder utilizar funciones y clases de la comunicación MQTT para
publicar y recibir mensajes, suscribirse a topics, etc.
#include <WiFi.h> //Librería Open Source para poder utilizar funciones y clases relacionadas con la
configuración de red WiFi del hardware ESP32
#include <Wire.h> //Libreria para usar Arduino o ESP32 en protocolo de comunicación I2C
#include <LiquidCrystal_I2C.h>
//----- Definición de variables globales -----
#define SA 0x12 //Direccion del registro del sensor esclavo principal.
#define St1 0x00 //Registro de memoria 0x06 para medición de ug/m^3 de partículas PM2.5 estándar
#define I2C_SDA 13 //Define GPIO 13 como SDA (gris)
#define I2C_SCL 32 // Define le pin 32 como SCL (azul)
#define DHTPIN 4 //Define GPIO 4 como entrada de datos de sensor DHT 11
#define DHTTYPE DHT11 //Define modelo de DHT11 como DHTTYPE de su tipo de modelo
DHT dht(DHTPIN, DHTTYPE);
#define A1 33 //Define GPIO 33 para sensor LDR
#define A2 34 //Define GPIO 34 para MQ7 CO monóxido de carbono
//----- Definición de objetos para el ESP32 -----
//-----y de variables para ESP32-----
long lastMsg = 0; //Control de tiempo transcurrido desde el último mensaje publicado al broker
//char msg[50];
//int value = 0;
```

```

//-----Definición de variables para los sensor PM 2.5-----
-----
char cadena[8]; //Se define un array de 8 variables tipo char, ya que cada 1 byte equivale a 1 variable
char
float PM25_N; //Se define variable de lectura de las partículas PM 2.5 en ug/m^3
//char C_4[8]; //Se crea array para lectura PM25_N. Novedad de Cabo 4.
//-----Definición de variables para sensor DHT11-----
-----
float T_N; //Se define variable de temperatura en °C
float H_N; //Se define variable de humedad relativa en %
//-----Definición de variables para sensor LDR-----
-----
float R_LDR; //Resistencia variable de LDR
float R = 6; //Resistencia en serie de 6 kOhm
float v; //Voltaje entregado por el divisor
float L_N; // Lectura de luxes
//char C_3[8]; //Se crea array para lectura PM25_N. Novedad de Cabo 3.
//-----Definición de variables para sensor MQ7-----
-----
const float Vc = 5.0; //Suministro para el sensor (Fuente externa)
float Rs_CO; //Variable de la resistencia de sensor a cualquier sustancia, para este caso de calibración y
detectará gas CO en kohm y es la medida por el sensor
float Ro_medida = 0.43; //Variable de la resistencia de sensor a 100 ppm de concentración de CO en el
aire, en kohm y es la medida por nosotros
float ARL; //Variable de la lectura analógica. Como el ESP32 es de 12 bits y por tanto, un omtervalo de
[0-4095]-->[0-3.3 Volts]
float VRL; //Variable del voltage de la resistencia de carga RL. La conversion a este voltage se obtiene
de lectura_analogica
float PPMCO_N; //Partes por millón de monóxido de carbono
const char* mqtt_server = "3.21.159.14"; //Dirección del IP de la Nube
long now; //Almacenará el tiempo transcurrido desde la conexión en MQTT
//-----Definicion LCD-----
-----
int lcdColumns = 16;
int lcdRows = 2;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);
//-----Definición de mensajes -----
-----

//*****
//***** Inicia función de configuración del MCU *****
//*****
void setup()
{
  //Serial.begin(115200); //Puesta en marcha de monitor serial a 115200 baudios

```

```

    setup_wifi();
    client.setServer(mqtt_server, 1883); //Se define al cliente (esp32) para conectarse al Raspberry en su
puerto 1883 habilitado por defecto
    client.setCallback(teniente); //Invoca al argumento la función auxiliar 'teniente', relacionada con la
SUSCRIPCION de mensajería MQTT
    pinMode(A3, OUTPUT); //A3 como salida de RELAY 1
    pinMode(A4, OUTPUT); //A4 como salida de RELAY 2
    Wire.begin(); //Puesta en marcha de comunicación I2C, direccion del esclavo en pines SDA y SCL por
defecto son GPIO 21 y GPIO 22
    Wire1.begin(I2C_SDA, I2C_SCL); //Puesta en marcha de comunicación I2C, direccion del esclavo en pines
SDA y SCL para sensor PM25
    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("EN LINEA");
    dht.begin(); //Puesta en marcha de sensor DHT11
    delay(30000); // Espera 30 segundos para que el sensor se estabilice
}

//*****
***** Inicia función principal del programa*****
//*****
void loop()
{
    //-----Inicia conexión MQTT como PUBLISHER (ESP32 envía datos a su servidor)-----
    //-----Sargento da parte de las novedades a su Teniente-----

    if (!client.connected()) //Si esp32 pierde conexion con Broker, entonces procede a reintentar conexión. Si
sargento pierde comunicación con su Teniente, solicita reconexión
    {
        reconnect(); //Se invoca funcion de reconexión. Sargento solicita comunicación con su Teniente
    }

    client.loop(); //Inicia instanciación de método 'loop' de bucle infinito de la clase 'Client' para publicar
en el Broker. Sargento ya puede pasar las novedades a su Teniente.
    now = millis(); //Se crea objeto 'now' que almacena el milisegundos transcurrido desde que inició el loop.
El Sargento lleva conteo del tiempo activo.
    if (now - lastMsg > 100) //Definición de tiempo de muestreo con el que van a inicializar los datos, cada
100 ms. Sargento pasa novedad cada 100 ms
    {
        lastMsg = now; //Se almacena el instante de tiempo en el que se publico mensaje al Broker. Sargento
registra "a que hora pasó una novedad" a su teniente
    }
    //-----Inicia SENSADO-----
    -----
    //-----CABOS pasan sus novedades a su SARGENTO-----

```

```

//-----A través de protocolo UART, I2C, BUSES DE GPIO-----
-----

Wire1.beginTransmission(SA); //Inicia enlace de transmisión a la dirección 'SA' (contiene dirección de
memoria del sensor esclavo)
Wire1.write(St1); //ESP32 Envía o Escribe el valor de 'ST1' al sensor para darle la orden de comenzar
medión
Wire1.endTransmission(true); //ESP32 termina la transmisión del primer dato enviado por el sensor,
pero NO LIBERARÁ EL BUS I2C, simplemente reinicia para recibir
//otro dato anterior que mandó el sensor esclavo y que estuvo en cola de
espera, el argumento debe ser FALSE.
//Si se le coloca TRUE, entonces ESP32 liberará el BUS I2C Y MANDA UN AVISO
DE PARADA TOTAL.
Wire1.requestFrom(SA, 8); //Solicita datos del sensor esclavo en bytes. El argumento requiere la
dirección 'SA' del esclavo y la cantidad de bytes a enviar, este caso 8 bytes
//-----Inicia lectura de datos-----
-----

while(Wire1.available()) //Inicia Bucle. Mientras haya disponibles bytes de almacenamiento de las
lecturas emitidas despues del 'requestfrom()', se hace el bucle.
//'Wire.available()' devuelve el numero de bytes DISPONIBLES para su lectura.
//Al no haber una comparación booleana en el argumento de 'While', pregunta si
está vació o no, es decir si hay o no bytes disponibles para lectura.
//Mientras sea verdad que 'wire.available' no devuelve bytes vacíos, correr el
bucle, de lo contrario se sale de él. Es decir, mientras haya torrente de bytes
//Lo mismo con el puerto serial. Ambos deben estar activos
{

//.....SENSOR PM 2.5 .....
for(int i = 0; i<8; i++) //Bucle de control para arrays de 8 caracteres. Véase la línea 10.
//AL OCTAVO MUESTREO DA LA MEDICIÓN, ES DECIR AL TERMINAR DE RECORRER DE 0 A
7 DEL ARRAY
{
cadena[i] = Wire1.read(); // Una vez solicitado la trasnmisión de bytes con
'wire.requestfrom', se puede usar función 'Wire.read()'
//Lee solamente 1 byte de la cola en bus I2C que se le
solicitó al sensor.
//el primer dato byte que halla llegado, se almacena en la i-
ésima posición del array 'cadena[i]' que es de 8 elementos char, vease linea 10

PM25_N = (word(cadena[6],cadena[7]))/10; //Una palabra almacena un número sin signo de 16
bitsb(2 bytes), de 0 a 65535. Igual que un int. pero sin el signo.
//Se convierte a palabra, el Byte. Para esta
coneverción, se solicita las ultimas dos posiciones del Byte alojado en la cadena
//esto porque esos ultimos 2 bytes son los
significativos

```

```

//Al principio imprimirá 7 valores de cero, porque al
principio toda la cadena vale puros caracteres '0'
//Mas significativo decenas, menos significativo
unidades
}
//.....SENSOR DHT11.....
//delay(1000); //Retardo de 1 segundo para dar tiempo al sensor DHT 11 de alcanzar el tiempo de
respuesta de la trasducción
T_N = dht.readTemperature(); //Toma lectura de temperatura
H_N = dht.readHumidity(); //Toma lectura de humedad relativa
//.....SENSOR LDR.....
v = analogRead(A1);
R_LDR=((4095*R)/v)-R;
L_N=800.69*pow(R_LDR, -1.283);
//.....SENSOR MQ7.....
ARL = analogRead(A2); //Lestrura de MQ7
VRL = (ARL/4095)*3.3; //Conversión a voltaje, en la resolución de 12 bits y 3.3 V maximos que
puede leer
Rs_CO = 1*((Vc-VRL)/VRL); //Valor de la resistencia del sensor RL en kohm
//ppm_CO = 21.3074/pow((Rs_CO)/Ro_medida, 1.35692); //media década
PPMCO_N = 99.9999/(pow((Rs_CO)/Ro_medida), 1.17925)); //1 decada
//-----Inicia publicación de lecturas al broker-----
//-----SARGENTO PASA NOVEDADES A SU TENIENTE-----
-----

float novedades[7] = {T_N, H_N, L_N, PM25_N, PPMCO_N, A_N1, A_N2};
String payload = "";

for(int i=0; i<7; i++)
{
  payload += String(novedades[i], 4);
  if(i<7 - 1)
  {
    payload += ",";
  }
}
client.publish("sargento/teniente", payload.c_str()); //Se mandan las Novedades al Teniente

//-----Inicia impresión de lecturas en el LCD-----
-----

lcd.setCursor(0,0);
lcd.print("Temp, Hum");
lcd.setCursor(0,1);
lcd.print(T_N); lcd.print(","); lcd.print(H_N);

```

```
    delay(1500);  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("PM25, PPM");  
    lcd.setCursor(0,1);  
    lcd.print(PM25_N); lcd.print(","); lcd.print(PPMCO_N);  
    delay(1500);  
    lcd.clear();  
  }  
}  
}
```

Anexo D

Código en Python para Raspberry como servidor local

```
#Funciones del Teniente:

#1.- Autenticarse con el Sargento para recibir las Novedades por parte de él.

#2.- Decifrar el mensaje (desencriptación TSL).

#3.- Descomponer el mensaje de Novedades (se obtienen componentes del vector de datos)

#4.- Alojara los datos en el servidor de Base de Datos y proteger la base de datos con autenticación.

#5.- Se vuelve a cifrar el mensaje por TSL y transmite las Novedades al Alto Mando, preparando el
mensaje.

#6.- Recibe las Instrucciones del Alto Mando y prepara el mensaje para enviarlo al Sargento.

#7.- Se repite el proceso desde el punto 1.

#Inicia Raspberry como SUSCRIPTOR

import paho.mqtt.client as mqtt #Libreria MQTT CLIENTE

import time #Librería para usar delays

import mysql.connector as mysql

from datetime import datetime #Libreria para usar tiempo y fecha

#region de definicion de funciones que se activara cuando el cliente SE CONECETE AL BROKER y cuando
RECIBA UN MENSAJE

def conectado(teniente, userdata, flags, rc): #Devuelve estado del teniente si se logro conectar al
broker, si lo hace debe suscribirse para recibir las novedades del SARGENTO y las instrucciones del Alto
Mando (Payloads)

    if rc==0:

        print("TENIENTE EN LINEA")

        teniente.subscribe([("sargento/teniente",0),("alto_mando/teniente",0)]) #se suscribe a los
topics

    else:

        print("TENIENTE NO SE PUDO CONECTAR")

def receptor(teniente, userdata, mensaje): #Imprime las novedades si el Teniente recibe las novedades
del Sargento o instrucciones del Alto Mando

    payload_str = mensaje.payload.decode("utf-8") #payload o el topic, añadir .decode para pasar a
string

    if mensaje.topic == "sargento/teniente" :

        novedades_str = payload_str #Transfiere el mensaje a la variable de novedades en string
```

```

novedades_lst = [float(i) for i in payload_str.split(",")]
novedades = tuple(novedades_lst)
tiempo = datetime.today().strftime('%Y-%m-%d %H:%M')
cursor.execute(query_novedades, novedades)
db.commit()

print("Mensaje decifrado: ", tiempo, " ", novedades)
teniente.publish("teniente/alto_mando", novedades_str)

elif mensaje.topic == "alto_mando/teniente" :
    instrucciones_str = payload_str
    instrucciones_lst = [int(j) for j in payload_str.split(",")]
    instrucciones = tuple(instrucciones_lst)
    print("Instrucciones: ", instrucciones)

time.sleep(1)

#*****Región de instanciación de objetos DATA
BASE*****

db = mysql.connect(host="192.168.100.82", user="robertoDB", passwd="1234", database="lecturas") #Se
ingresa conexión a la Base de Datos

cursor = db.cursor() #Se crea objeto Cursor que apunta a los registros, se encarga de ejecutar el
query.

query_novedades = "INSERT INTO novedades(Temperatura, Humedad, Luz, PM25, PPM_CO, A_N1, A_N2) VALUES
(%s, %s, %s, %s, %s, %s, %s)" #petición para insertar registros

#*****Región de instanciación de objetos
SUBSCRIBER*****

#Se define objeto cliente como TENIENTE
teniente = mqtt.Client() #opcional ID nombre

#Se conecta el cliente de IP de raspberry a puerto 1883 del MQTT
teniente.connect("3.139.93.225", 1883) #AWS server

#Inicia crear funciones para que se CONECTE y se SUSCRIBA
teniente.on_connect = conectado #¿Que función ejecutar cuando me conecte al broker?
teniente.on_message = receptor#¿Que función ejecutar cuando reciba un mensaje?

#INICIA BUCLE INFINITO DE LA SUSCRIPCIÓN
teniente.loop_forever()
print("Fin de programa")

```

Anexo E

Código en Python para servidor en la nube de Amazon AWS

```
#Funciones del Alto Mando

#1.- Autenticarse con el Teniente para recibir las Novedades por parte de él.

#2.- Decifrar el mensaje (desencriptación TSL).

#3.- Descomponer el mensaje de Novedades. Separa las variables por variables de sensado y variables de
actuación.

#4.- Analiza las variables de sensado y las coloca en la entrada del controlador TODO/NADA. Ejecuta el
controlador

#5.- Obtiene vector de respuesta para los valores de A_N1 y A_N2 de Aire Acondicionado y Purificador de
Aire respectivamente almacenandolo en variable "Instrucciones"

#6.- Se encripta por TSL mensaje de Instrucciones y lo envía hacia el Teniente.

#7.- Genera retroalimentación con otros clientes suscritos al topic, unicamente emitiendo información
de sensado.

#Programa para usar las funciones MQTT

#Servidor AWS como SUSCRIPTOR

import paho.mqtt.client as mqtt #Libreria MQTT CLIENTE

import time #Librería para usar delays

from datetime import datetime #Libreria para usar tiempo y fecha

T_N, H_N, L_N, PM25_N, PPMCO_N, T_SN, PM25_SN, PPMCO_SN= 0, 0, 0, 0, 0, 23, 5, 9 #Condiciones iniciales
para el controlador

#region de definicion de funciones que se activara cuando el cliente SE CONCETE AL BROKER y cuando
RECIBA UN MENSAJE

def conectado(Alto_Mando, userdata, flags, rc): #Devuelve estado del teniente si se logro conectar al
broker, si lo hace debe suscribirse para recibir las novedades del SARGENTO y las instrucciones del Alto
Mando (Payloads)

    if rc==0:

        print("ALTO MANDO EN LINEA")

        Alto_Mando.subscribe([("teniente/alto_mando",0),("usuario/pauta_consigna/alto_mando",0)]) #se
suscribe al topic para recibir las Pautas de consigna

    else:

        print("ALTO MANDO NO SE PUDO CONECTAR")
```

```

def receptor(Alto_Mando, userdata, mensaje): #Imprime las novedades si el Teniente recibe las novedades
del Sargento

    global T_N, H_N, L_N, PM25_N, PPMCO_N, T_SN, PM25_SN, PPMCO_SN

    payload_str = mensaje.payload.decode("utf-8") #payload o el topic, añadir .decode para pasar a
string

    if mensaje.topic == "teniente/alto_mando":

        novedades_lst = [float(i) for i in payload_str.split(",")]

        novedades = tuple(novedades_lst)

        tiempo = datetime.today().strftime('%Y-%m-%d %H:%M')

        T_N = novedades[0]

        H_N = novedades[1]

        L_N = novedades[2]

        PM25_N = novedades[3]

        PPMCO_N = novedades[4]

        print("Mensaje decifrado del Teniente: ", tiempo, " ", novedades)

    elif mensaje.topic == "usuario/pauta_consigna/alto_mando":

        pauta_consigna_lst = [float(j) for j in payload_str.split(",")]

        pauta_consigna = tuple(pauta_consigna_lst)

        T_SN = pauta_consigna[0]

        PM25_SN = pauta_consigna[1]

        PPMCO_SN = pauta_consigna[2]

        print("Consigna es: ", T_SN, PM25_SN, PPMCO_SN)

    instrucciones = control_A(T_N, T_SN), control_B(PM25_N, PM25_SN, PPMCO_N, PPMCO_SN) #las
instrucciones generadas son arrojadas por la funcion de control en forma de tupla

    instrucciones_str = ",".join(str(k) for k in instrucciones)

    print("Instrucciones: ", instrucciones)

    pauta_consigna = T_SN, PM25_SN, PPMCO_SN

    pauta_consigna_str = ",".join(str(n) for n in pauta_consigna)

    #Region de publicadores

    Alto_Mando.publish("alto_mando/T_N/usuario", T_N) #Publicador para cliente celular

    Alto_Mando.publish("alto_mando/H_N/usuario", H_N) #Publicador para cliente celular

    Alto_Mando.publish("alto_mando/L_N/usuario", L_N) #Publicador para cliente celular

    Alto_Mando.publish("alto_mando/PM25_N/usuario", PM25_N) #Publicador para cliente celular

    Alto_Mando.publish("alto_mando/PPMCO_N/usuario", PPMCO_N) #Publicador para cliente celular

    Alto_Mando.publish("alto_mando/T_SN/usuario", T_SN) #Publicador para cliente celular

```

```

Alto_Mando.publish("alto_mando/PM25_SN/usuario", PM25_SN) #Publicador para cliente celular
Alto_Mando.publish("alto_mando/PPMCO_SN/usuario", PPMCO_SN) #Publicador para cliente celular
Alto_Mando.publish("alto_mando/teniente", instrucciones_str) #Publicacion para Raspberry y
raspberry manda instrucciones a ESP32

Alto_Mando.publish("usuario/pauta_consigna/alto_mando/teniente", pauta_consigna_str) #Publicacion
para raspberrry y raspberrry guarda consigna en Base de datos

#time.sleep(1)

```

```

def control_A(T_N, T_SN): #Control de temperatura, revisar diagrama de flujo

```

```

    descender, A_N1 = None, 0

```

```

    T_a = T_SN - 1

```

```

    T_b = T_SN + 1

```

```

    if T_N > T_b :

```

```

        A_N1 = 1

```

```

        descender = 1

```

```

    if T_N < T_b and T_N > T_a :

```

```

        A_N1 = 1

```

```

    if T_N < T_b and T_N < T_a :

```

```

        A_N1 = 0

```

```

    if T_N < T_a:

```

```

        A_N1 = 0

```

```

    if T_N > T_a:

```

```

        if T_N < T_b and descender == 0 :

```

```

            A_N1 = 0

```

```

    else:

```

```

        descender = 0

```

```

    if T_N > T_b :

```

```

        A_N1 = 1

```

```

    return A_N1

```

```

def control_B(PM25_N, PM25_SN, PPMCO_N, PPMCO_SN): #Control de calidad de aire, revisar diagrama de
flujo

```

```

    des_PM25, des_PPMCO, A_N2, A_N2_PM25, A_N2_PPMCO = None, None, 0, 0, 0

```

```

    PM25_a = PM25_SN - 1

```

```

    PM25_b = PM25_SN + 1

```

```

PPMCO_a = PPMCO_SN - 1
PPMCO_b = PPMCO_SN + 1
#Control PM 2.5
if PM25_N > PM25_b :
    A_N2_PM25 = 1
    des_PM25 = 1
if PM25_N < PM25_b and PM25_N > PM25_a :
    A_N2_PM25 = 1
if PM25_N < PM25_b and PM25_N < PM25_a :
    A_N2_PM25 = 0
if PM25_N < PM25_a:
    A_N2_PM25 = 0
if PM25_N > PM25_a:
    if PM25_N < PM25_b and des_PM25 == 0 :
        A_N2_PM25 = 0
else:
    des_PM25 = 0
if PM25_N > PM25_b :
    A_N2_PM25 = 1
#Control PPM CO
if PPMCO_N > PPMCO_b :
    A_N2_PPMCO = 1
    des_PPMCO = 1
if PPMCO_N < PPMCO_b and PPMCO_N > PPMCO_a :
    A_N2_PPMCO = 1
if PPMCO_N < PPMCO_b and PPMCO_N < PPMCO_a :
    A_N2_PPMCO = 0
if PPMCO_N < PPMCO_a:
    A_N2_PPMCO = 0
if PPMCO_N > PPMCO_a:
    if PPMCO_N < PPMCO_b and des_PPMCO == 0 :
        A_N2_PPMCO = 0
else:
    des_PPMCO = 0
if PPMCO_N > PPMCO_b :

```

```

    A_N2_PPMCO = 1
#Compara las solicitudes de ambos sensores PM 2.5 Y PPM CO
if A_N2_PM25 == A_N2_PPMCO:
    A_N2 = A_N2_PM25
else:
    A_N2 = 1
return A_N2

```

```

*****Región de instanciación de objetos SUBSCRIBER *****
#Se define objeto cliente como ALTO MANDO
Alto_Mando = mqtt.Client() #opcional ID nombre
#Se conecta el cliente de IP de raspberry a puerto 1883 del MQTT
Alto_Mando.connect("3.21.159.14", 1883) #AWS server
#Se autentica el dispositivo en la red
#Alto_Mando.username_pw_set(username="Roberto", password="1234")
#Inicia crear funciones para que se CONECTE y se SUSCRIBA
Alto_Mando.on_connect = conectado #¿Que funcion ejecutar cuando me conecte al broker?
Alto_Mando.on_message = receptor #¿Que funcion ejecutar cuando reciba un mensaje?
#INICIA BUCLE INFINITO DE LA SUSCRIPCION
Alto_Mando.loop_forever()
print("Fin de programa")

```

Anexo F

Código en Python para el algoritmo de predicción de confort térmico basado en la técnica de norma ISO 7730.

```
#Calculo de las condiciones de Confort
#Se utiliza metodo numero con un maximo de hasta 150 iteraciones para obtener los valores de PMV Y PPD
#Esta información se alojara en la base de datos para su posterior análisis de datos
import paho.mqtt.client as mqtt #Libreria MQTT CLIENTE
import time #Librería para usar delays
from datetime import datetime #Libreria para usar tiempo y fecha
import math
import mysql.connector as mysql

#region de definicion de funciones que se activara cuando el cliente SE CONECETE AL BROKER y cuando
RECIBA UN MENSAJE
CLO, MET, WME, TR, VEL = 0.3, 0.8, 0, 20, 0.10
def conectado(PC, userdata, flags, rc): #Devuelve estado del teniente si se logro conectar al broker,
si lo hace debe suscribirse para recibir las novedades del SARGENTO y las instrucciones del ALto Mando
(Payloads)
    if rc==0:
        print("SUPERVISOR EN LINEA")
        PC.subscribe(["teniente/alto_mando",0],("usuario/pauta_confort/alto_mando",0)) #se suscribe
al topic para recibir las Novedades y variables de medicion del confort
    else:
        print("NO SE PUDO CONECTAR")

def receptor(PC, userdata, mensaje): #Imprime las novedades si el Teniente recibe las novedades del
Sargento
    global T_N, H_N, CLO, MET, WME, TR, VEL
    payload_str = mensaje.payload.decode("utf-8") #payload o el topic, añadir .decode para pasar a
string
    if mensaje.topic == "teniente/alto_mando": #En este topic se publican las novedades del teniente,
obtenidas de su sargento
        termicas_lst = [float(i) for i in payload_str.split(",")]
        termicas = tuple(termicas_lst)
        tiempo = datetime.today().strftime('%Y-%m-%d %H:%M')
        T_N = termicas[0]
```

```

    H_N = termicas[1]

    print("Condiciones termodinámicas de la planta: ", tiempo, " ", T_N, H_N)

elif mensaje.topic == "usuario/pauta_confort/alto_mando" :

    persona_lst = [float(j) for j in payload_str.split(",")]

    persona = tuple(persona_lst) #Las variables relacionados con confort de la persona, se
almacnan en variable "persona", quien es el usuario

    CLO = persona[0]

    MET = persona[1]

    WME = persona[2]

    TR = persona[3]

    VEL = persona[4]

    print("[CLO, MET, WME, TR, VEL] = : ", persona)

    time.sleep(2)

    PMV_PPD = calculo(T_N, H_N, CLO, MET, WME, TR, VEL) #las instrucciones generadas son arrojadas por
la funcion de control en forma de tupla

    PMV_PPD_str = ",".join(str(k) for k in PMV_PPD)

    print("[PMV_PPD]= ", PMV_PPD)

    PMV = PMV_PPD[0]

    PPD = PMV_PPD[1]

    #Region de publicadores

    PC.publish("PMV", PMV) #Publicador para cliente celular

    PC.publish("PPD", PPD)

    cursor.execute(query_confort, PMV_PPD)

    db.commit()

    #time.sleep(1)

def calculo(T_N, H_N, CLO, MET, WME, TR, VEL): #Control de temperatura, revisar diagrama de flujo

    PA = H_N*10*math.exp(16.6536-4030.183/(T_N+235))

    ICL = 0.155*CLO

    M = MET*58.15

    W = WME*58.15

    MW = M-W

    if ICL <= 0.078 :

        FCL = 1+1.29*ICL

    else :

```

```

    FCL = 1.05+0.645*ICL
HCF = 12.1*math.sqrt(VEL)
TAA = T_N+273
TRA = TR+273
#Inicia calculo de la temperatura de superficie de la ropa por iteracion
TCLA = TAA+(35.5-T_N)/(3.5*ICL+0.1)
P1 = ICL*FCL
P2 = P1*3.96
P3 = P1*100
P4 = P1*TAA
P5 = 308.7-0.028*MW+P2*(TRA/100)*4
XN = TCLA/100
XF = XN
N = 0
EPS = 0.00015
continuar = True
while continuar :
    XF = (XF+XN)/2
    HCN = 2.38*(math.pow(abs(100*XF-TAA),0.25))
    if HCF>HCN :
        HC = HCF
    else :
        HC = HCN
    XN = (P5 + P4 * HC - P2 * math.pow(XF, 4)) / (100 + P3 * HC)
    N = N+1
    if N>150 :
        continuar = False
    if abs(XN-XF)>EPS :
        continuar = True
    else :
        continuar = False
if N>150 :
    PMV = 2
    PPD =100
else:

```

```

TCL = 100*XN-273
HL1 = 3.05*0.001*(5733-6.99*MW-PA)
if MW>58.15 :
    HL2 = 0.42*(MW-58.15)
else :
    HL2 = math.factorial(0)
HL3 = 1.7*0.00001*M*(5867-PA)
HL4 = 0.0014*M*(34-T_N)
HL5 = 3.96 * FCL * (math.pow(XN, 4) - math.pow((TRA/100), 4))
#Inicia calculo de los indices PMV y PPD
TS = 0.303*math.exp(-0.036*M)+0.028
PMV = TS*(MW-HL1-HL2-HL3-HL4-HL5)
PPD = 100 - 95 * math.exp(-0.03353*PMV**4-0.2179 *PMV**2)
return (PMV, PPD)

```

```

*****Región de instanciación de objetos DATA BASE*****
db = mysql.connect(host="192.168.100.82", user="robertoDB", passwd="1234", database="lecturas") #Se
ingresa conexión a la Base de Datos
cursor = db.cursor() #Se crea objeto Cursor que apunta a los registros, se encarga de ejecutar el
query.
query_comfort = "INSERT INTO confort(PMV, PPD) VALUES (%s, %s)"
*****Región de instanciación de objetos SUBSCRIBER*****
#Se define objeto cliente como ALTO MANDO
PC = mqtt.Client() #opcional ID nombre
#Se conecta el cliente de IP de raspberry a puerto 1883 del MQTT
PC.connect("3.21.159.14", 1883) #AWS server
#Se autentica el dispositivo en la red
#PC.username_pw_set(username="Roberto", password="1234")
#Inicia crear funciones para que se CONECTE y se SUSCRIBA
PC.on_connect = conectado #¿Que funcion ejecutar cuando me conecte al broker?
PC.on_message = receptor #¿Que funcion ejecutar cuando reciba un mensaje?
#INICIA BUCLE INFINITO DE LA SUSCRIPCION
PC.loop_forever()
print("Fin de programa")

```