



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

TRABAJO DE INTEGRACIÓN CURRICULAR

previo a la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

“DISEÑO DE UN PROTOTIPO DE SISTEMA DE MONITOREO Y
PREDICCIÓN DEL CONSUMO ELÉCTRICO EN ZONAS RESIDENCIALES
USANDO REDES NEURONALES ARTIFICIALES”.

AUTOR

VILLAO PAREDES KEVIN ADRIAN

TUTOR

ING. CARLOS ALBERTO SALDAÑA ENDERICA

LA LIBERTAD – ECUADOR

2023

AGRADECIMIENTO

A mi querida madre, mi eterna creyente y fuente inagotable de apoyo, gracias por estar siempre a mi lado, creyendo en mí incluso cuando mis objetivos parecían inalcanzables. Tu amor y confianza han sido mi mayor inspiración.

A mi esposa, mi compañera de vida y mi mayor apoyo, gracias por compartir conmigo este emocionante viaje. Tu presencia ha llenado cada día de felicidad y motivación. Juntos construiremos un futuro maravilloso.

A mi amado padre, cuyo ejemplo de dedicación y determinación me ha impulsado a enfrentar los desafíos laborales con valentía. Tu presencia y carácter son un constante recordatorio de perseverancia.

A mi hermano, mi amigo, gracias por ser mi apoyo en mis momentos difíciles. Lograr este objetivo es solo el comienzo de muchas metas que perseguiré con determinación.

A todos ustedes, mi familia y mi esposa, les agradezco profundamente por su amor, apoyo y aliento incondicional. Vuestra confianza en mí ha sido mi mayor motivación y estoy emocionado por compartir más éxitos juntos en el futuro.

DEDICATORIA

Deseo dedicar este proyecto a mi amada familia, quienes han sido mi mayor apoyo desde el inicio de mis estudios universitarios. Su aliento y cariño incondicional han sido mi fuerza impulsora para seguir adelante y alcanzar este logro.

A mis amigos y compañeros, les dedico este trabajo con gratitud y reconocimiento por estar siempre a mi lado. La confianza y solidaridad que compartimos han sido fundamentales para enfrentar los desafíos académicos juntos.

Asimismo, quiero dedicar este proyecto a aquellos compañeros que, lamentablemente, han decidido abandonar esta carrera. Mi deseo es transmitirles mi convicción de que, con dedicación y perseverancia, habrían alcanzado sus metas. A veces, solo se necesita un poco más de esfuerzo para lograr el éxito.

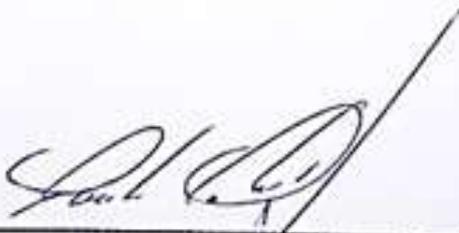
Aprovecho este momento para agradecer de manera especial a mi esposa, quien ha sido mi compañera incondicional en este camino académico y personal. Su apoyo inquebrantable y su amor han sido mi motivación para enfrentar los desafíos con valentía y determinación. Te amo y agradezco por ser mi inspiración constante.

Por último, expreso mi sincero reconocimiento a mis respetados profesores, quienes han dedicado su tiempo y conocimiento para guiarnos en nuestro desarrollo profesional. Su compromiso con la enseñanza y su exigencia constante han sido fundamentales para moldear mi formación académica y prepararme para los retos futuros. Agradezco sus enseñanzas valiosas y el impacto positivo que han dejado en mi vida.

APROBACIÓN DEL TUTOR

En mi calidad de tutor del trabajo de titulación denominado: **“Diseño de un prototipo de sistema de monitoreo y predicción del consumo eléctrico en zonas residenciales usando redes neuronales artificiales”**, elaborado por el estudiante Villao Paredes Kevin Adrián, de la carrera de Electrónica y Automatización de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes y autorizo al estudiante que inicie los trámites legales correspondientes.

La Libertad, 4 de agosto 2023.

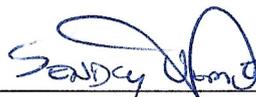


Ing. Carlos Saldaña Enderica

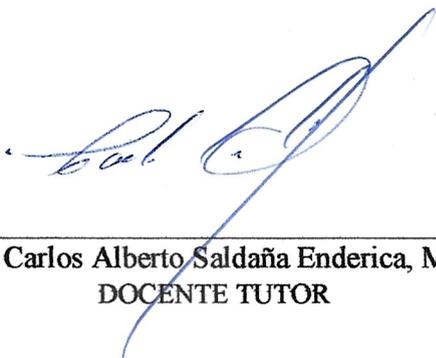
**TRIBUNAL DE SUSTENTACIÓN DEL TRABAJO DE TITULACIÓN DE
INTEGRACIÓN CURRICULAR**



**Ing. Ronald Rovira Jurado, Ph.D.
DIRECTOR DE LA CARRERA DE
ELECTRÓNICA Y AUTOMATIZACIÓN**



**Ing. Sency Agustín Vera González, Mgt.
DOCENTE ESPECIALISTA**



**Ing. Carlos Alberto Saldaña Enderica, Mgt.
DOCENTE TUTOR**



**Ing. Luis Chuquimarca Jiménez, Mgt.
DOCENTE GUIA UIC**



**Ing. Corina Gonzabay, Mgt.
SECRETARÍA DE LA CARRERA DE
ELECTRÓNICA Y AUTOMATIZACIÓN.**

RESUMEN

El consumo eléctrico ha adquirido un rol esencial en la administración eficiente de recursos y la búsqueda de eficiencia óptima. Este estudio fusiona redes neuronales artificiales (RNA) e Internet de las cosas (IoT) en un enfoque exploratorio que culmina en un prototipo de predicción de consumo eléctrico en hogares. La sinergia de RNA e IoT permite la recopilación y análisis de datos, proporcionando información sobre la potencia eléctrica de dispositivos en viviendas.

El propósito central de este proyecto es crear un prototipo de monitoreo de consumo energético para zonas residenciales con alto consumo eléctrico. Empleando técnicas avanzadas, como redes neuronales artificiales para predecir el consumo de los próximos meses, los usuarios podrán tomar decisiones informadas para optimizar su consumo y reducir costos, disminuyendo el impacto ambiental.

ABSTRACT

Electric consumption has taken on an essential role in efficient resource management and the pursuit of optimal efficiency. This study merges artificial neural networks (ANNs) and the Internet of Things (IoT) in an exploratory approach that culminates in a precise prototype for predicting household electrical consumption. The synergy of ANNs and IoT enables efficient data collection and analysis, providing information about the electrical power of three devices in homes.

The central purpose of this project is to create an energy consumption monitoring prototype for residential areas with high electricity usage. By employing advanced techniques such as artificial neural networks to predict consumption for upcoming months, users will be able to make informed decisions to optimize their usage, reduce costs, and mitigate environmental impact.

INDICE GENERAL

AGRADECIMIENTO	I
DEDICATORIA.....	II
APROBACIÓN DEL TUTOR.....	III
RESUMEN	IV
INDICE GENERAL.....	VI
INDICE DE FIGURAS	IX
INDICE DE TABLA	XII
INDICE DE ANEXOS	XIII
INTRODUCCIÓN.....	1
CAPITULO I.....	2
1.1 ANTECEDENTES.....	2
1.2 DESCRIPCIÓN DEL PROYECTO	3
1.3 OBJETIVOS DEL PROYECTO	4
1.3.1 OBJETIVO GENERAL	4
1.3.2 OBJETIVOS ESPECÍFICOS	4
1.4 JUSTIFICACIÓN.....	5
1.5 METODOLOGÍA.....	6

1.5.1 INVESTIGACIÓN EXPLORATORIA.	6
1.5.2 INVESTIGACIÓN DESCRIPTIVA.	6
CAPÍTULO II.....	7
2.1 MARCO CONTEXTUAL.....	7
2.1.1 CONSUMO ENERGÉTICO DE ZONAS RESIDENCIALES.....	7
2.2 MARCO CONCEPTUAL	8
2.2.1 INTERNET DE LAS COSAS.....	8
2.2.2 PROCESAMIENTOS DE SEÑALES	10
2.2.3 REDES NEURONALES ARTIFICIALES	15
2.2.4 APRENDIZAJE DE LA RED NEURONAL ARTIFICIAL.....	18
2.2.5 DESEMPEÑO DE RNA	21
2.2.6 EFICIENCIA ENERGÉTICA.....	23
2.2.7 NORMA ISO 50001	23
2.2.8 NORMA ISO 50006.....	23
2.2.9 PLIEGO TARIFARIO DEL SERVICIO ELÉCTRICO.	23
CAPITULO III	28
3.1 COMPONENTES DE LA PROPUESTA.....	28
3.1.1 COMPONENTES FÍSICOS.....	28
3.1.2 COMPONENTES LÓGICOS.	29

3.2 DISEÑO DE LA PROPUESTA	30
3.2.1 ELECCIÓN DE ARTEFACTOS ELÉCTRICOS	31
3.2.2 IMPLEMENTACIÓN DEL SISTEMA DE ADQUISICIÓN DE DATOS DEL CONSUMO ELÉCTRICO.	32
3.2.3 CONFIGURACIÓN DE ELEMENTOS LÓGICOS.....	33
3.2.4 TOPOLOGÍA DE LA RED NEURONAL ARTIFICIAL.	35
3.2.5 PROCESAMIENTO DE SEÑALES DE POTENCIA ELÉCTRICA.....	37
3.2.6 CONFIGURACIÓN NEURONAL NET FITTING.....	38
3.4. DISEÑO DE LA APLICACIÓN WATT-ADVISION.	48
3.4.1 CONSUMO ELÉCTRICO DE ZONA RESIDENCIAL.	51
CAPITULO IV	53
4.1 PRUEBAS DE ADQUISICIÓN Y ALMACENAMIENTO DE DATOS EN LA NUBE.	53
4.2. EVALUACIÓN DE DESEMPEÑO DE LAS REDES NEURONALES ARTIFICIALES.	55
4.3 EJECUCIÓN DE LA APP WATTADVISION	59
CONCLUSIONES Y RECOMENDACIONES.	69
BIBLIOGRAFÍA	71
ANEXOS	75

INDICE DE FIGURAS

FIGURA 1. ETAPA DE ADQUISICIÓN DE DATOS DEL PROTOTIPO.....	3
FIGURA 2. CONEXIONES FÍSICAS Y LÓGICAS.....	32
FIGURA 3. FLUJOGRAMA DE CONFIGURACIÓN DE ELEMENTOS LÓGICOS.	33
FIGURA 4. FLUJOGRAMA DE PROGRAMACIÓN DEL ESP32.....	34
FIGURA 5. ARQUITECTURA DE LA RNA DE N1	39
FIGURA 6. HISTOGRAMA DE LOS ARTEFACTOS.....	40
FIGURA 7. MEJOR RENDIMIENTO DE LA RNA DE N1.	40
FIGURA 8. RESULTADOS DE REGRESIÓN DE LA RNA DE N1.	41
FIGURA 9. ARQUITECTURA DE LA RNA DE A.A.	43
FIGURA 10. MEJOR RENDIMIENTO DE LA RNA DE A.A.	43
FIGURA 11. RESULTADOS DE REGRESIÓN DE LA RNA DE A.A.	44
FIGURA 12. ARQUITECTURA RNA DE I.1.	46
FIGURA 15. INTERFAZ PRINCIPAL.	48
FIGURA 16. INTERFAZ POTENCIA ELÉCTRICA TIEMPO REAL.....	49
FIGURA 17. INTERFAZ PREDICCIÓN DE CONSUMO Y COSTO DE ARTEFACTOS ELÉCTRICOS.	50

FIGURA 18. INTERFAZ DE REGISTROS Y ALARMAS DE CONSUMO ELÉCTRICO.	51
FIGURA 19. POTENCIA REAL Y POTENCIA DE LA RNA DE N1.	56
FIGURA 20. POTENCIA REAL Y LA POTENCIA DE LA RNA DE A.A.	57
FIGURA 21. POTENCIA REAL Y POTENCIA RNA DE I.1.	58
FIGURA 22. GRÁFICA DE POTENCIA EN TIEMPO REAL DE N1.	60
FIGURA 23. GRÁFICA DE POTENCIA EN TIEMPO REAL DE A.A.	60
.....	61
FIGURA 24. GRÁFICA DE POTENCIA EN TIEMPO REAL I.1.	61
FIGURA 25. INTERFAZ DE PREDICCIÓN DEL CONSUMO ELÉCTRICO	62
FIGURA 26. INTERFAZ DE REGISTROS Y ALARMAS DEL CONSUMO ELÉCTRICO	63
FIGURA 27. INTERFAZ DE REGISTROS Y ALARMAS DEL CONSUMO ELÉCTRICO	64
FIGURA 29. CREACIÓN DE EVENTOS IFTTT	88
FIGURA 30. WEBHOOK EN ARDUINO IOT CLOUD	88
FIGURA 31. HOJAS DE DATOS DE POTENCIA	89
FIGURA 32. GRÁFICA DE PROCESAMIENTO DE DATOS DE POTENCIA DE N1.	109

FIGURA 33. GRÁFICA DE PROCESAMIENTO DE DATOS DE POTENCIA DE A.A.	110
FIGURA 34. GRÁFICA DE PROCESAMIENTO DE DATOS DE POTENCIA DE I.1.....	110
FIGURA 35. DATOS DE ENTRADA DE LA RNA DE N1.	110
FIGURA 36. DATOS DE INGRESO DE LA RNA DE N1	111
FIGURA 37. CONFIGURACIÓN DE PORCENTAJES PARA PRUEBAS Y VALIDACIONES PARA N1.....	111
FIGURA 38. ENTRENAMIENTO DE LA RNA DE N1.....	112
FIGURA 39. DATOS DE ENTRADA DE LA RNA DE A.A.	118
FIGURA 40. CONFIGURACIÓN DE PORCENTAJES PARA PRUEBAS Y VALIDACIONES PARA A.A.	118
FIGURA 41. DATOS DE ENTRADA DE LA RNA DE I.1	124
FIGURA 42. DATOS DE INGRESO DE LA RNA DE I.1.....	124
FIGURA 43. CONFIGURACIÓN DE PORCENTAJES PARA PRUEBAS Y VALIDACIONES PARA I.1.	125
FIGURA 44. FACTURACIÓN DE LA RESIDENCIA DEL MES DE ABRIL A MAYO.....	132
FIGURA 45. IMPLEMENTACIÓN DE EQUIPOS.	133
FIGURA 46. MEDICIÓN DE AMPERAJE DE ILUMINACIÓN.....	133

FIGURA 47. VISUALIZACIÓN DE DATOS DE ENERGÍA DE LOS ARTEFACTOS ELÉCTRICOS.....	134
FIGURA 48. PREDICCIÓN DEL CONSUMO Y COSTO DE ENERGÍA DEL MES DE ENERO.	134
FIGURA 49. PREDICCIÓN DEL CONSUMO Y COSTO DE ENERGÍA DEL MES DE FEBRERO.....	135
FIGURA 50. PREDICCIÓN DEL CONSUMO Y COSTO DE ENERGÍA DEL MES DE MARZO.	135
FIGURA 51. PREDICCIÓN DEL CONSUMO Y COSTO DE ENERGÍA DEL MES DE ABRIL.....	136
FIGURA 52. PREDICCIÓN DEL CONSUMO Y COSTO DE ENERGÍA DEL MES DE MAYO.....	136
 INDICE DE TABLA	
TABLA 1. CLASIFICACIÓN DEL NIVEL DE VOLTAJE.	24
TABLA 2. CLASIFICACIÓN DE CATEGORÍAS TARIFARIAS Y LOS NIVELES DE VOLTAJE.	24
TABLA 3. DATOS DE ARTEFACTOS SELECCIONADOS	31
TABLA 4. RESULTADOS DEL ENTRENAMIENTO N1.	42
TABLA 5. RESULTADOS DEL ENTRENAMIENTO A.A.....	45
TABLA 6. RESULTADOS DEL ENTRENAMIENTO	48

TABLA 7: CONSUMO ELÉCTRICO Y COSTOS ENERO-MAYO.....	52
TABLA 8. DESEMPEÑO DE RNA.....	59
TABLA 9. COMPARACIÓN DEL CONSUMO ELÉCTRICO REAL-COSTO Y EL CONSUMO ELÉCTRICO-COSTO PREDICHO.....	68
TABLA 7. ESPECIFICACIONES PZEM004T.	75
TABLA 8. ESPECIFICACIONES ESP32 NODE MCU.	76
TABLA 10. RUBRO DE CONSUMO DE ENERGÍA.	131

INDICE DE ANEXOS

ANEXO 1: ESPECIFICACIONES TÉCNICAS DE EQUIPOS FÍSICOS.	75
ANEXO 2: CONFIGURACIÓN RED LOCAL.....	77
ANEXO 3: PROGRAMACION DE ESP32	77
ANEXO 5: PROGRAMACIÓN Y GRÁFICAS DE PROCESAMIENTO DE DATOS DE POTENCIA.	89
ANEXO 7: GRÁFICAS DE PROCESAMIENTO DE DATOS DE POTENCIA.....	109
.....	109
ANEXO 8 CREACION DE RED NEURONAL ARTIFICIAL PARA LA NEVERA.	110

ANEXO 9: CONFIGURACIÓN DE LA RED NEURONAL ARTIFICIAL PARA A.A.	117
ANEXO 10: CREACION DE LA RED NEURONAL ARTIFICIAL PARA I.1.....	123
ANEXO 11: DISEÑO DE GRÁFICAS DE PREDICCIÓN DE POTENCIA ELÉCTRICA	130
ANEXO 12: PLIEGO TARIFARIO.....	131
ANEXO 13: FACTURACIÓN DEL SERVICIO ELÉCTRICO.	132
ANEXO 14: IMPLEMENTACIÓN DE EQUIPO DE ADQUISICIÓN DE DATOS.	133
ANEXO 15: PREDICCIÓN DE CONSUMO Y COSTO DE ARTEFACTOS ELECTRICOS.....	134

INTRODUCCIÓN

En zonas residenciales, el uso de equipos tecnológicos modernos tiene el objetivo mejorar la comodidad del hogar y esta comodidad afecta a su consumo energético. En Ecuador no es común que personas realicen un correcto uso de la eficiencia energética, en el artículo “Eficiencia energética en el sector residencial de la Ciudad de Cuenca, Ecuador”, en las encuestas realizadas en el ámbito de aspectos y sostenibilidad, menciona que, “el 32% considera muy importante que una vivienda ahorre energía, agua y que en su proceso de construcción cause mínimo impacto al ambiente, sobre todo el nivel socio económico alto; sin embargo, el 66% no estarían dispuestos a invertir en viviendas con estas características, sobre el uso de energías alternativas, si bien el 32% considera importante el uso de estos sistemas en la vivienda, el 100% de los encuestados no utiliza ninguno [1].

En Ecuador existen tarifas que impone la Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables (ARCERNNR), clasificando en categorías residencial y General [2]. Estas tarifas varían según el grupo y la categoría del usuario. Sin embargo, muchos habitantes del país no están familiarizados con el verdadero valor de la energía eléctrica y carecen de conocimientos sobre cómo aplicar prácticas de eficiencia energética. Por lo tanto, cuando enfrentan un consumo energético elevado, los usuarios tienden a mantenerlo en un nivel estable sin conocer cuál es el equipo que más contribuye a este consumo, lo que puede afectar negativamente su economía y la gestión adecuada de la energía.

Con el objetivo de abordar esta problemática, el presente trabajo de investigación se centra en el diseño de un prototipo de sistema de monitoreo y predicción del consumo eléctrico en viviendas, haciendo uso de redes neuronales artificiales. El prototipo de esta investigación es capaz de adquirir datos del consumo energético mediante un controlador, para procesar y almacenar los datos en la nube monitoreando el consumo energético del equipo desde cualquier lugar. Los datos del consumo energético son parte del estudio para modelar el comportamiento, implementado redes neuronales artificiales para predecir el consumo energético de los equipos.

CAPITULO I

1.1 ANTECEDENTES.

El consumo energético en todo el mundo ha aumentado significativamente, ya que, a lo largo del tiempo, nosotros los seres humanos tenemos ese afán de ser ambiciosos y creativos para hacer nuevos dispositivos que facilitan nuestra vida. En algunas ocasiones exalimitamos nuestro consumo energético provocando nuevas fuentes de abastecimiento de energía.

La norma ISO 50001 nos indica que, cuando las organizaciones incorporan la gestión de la energía en sus prácticas empresariales, establecen un proceso para mejorar constantemente su rendimiento energético. Esta mejora no solo implica reducir los costos de energía, sino también incrementar su competitividad en el mercado. Además, al implementar medidas de gestión energética, las organizaciones contribuyen a alcanzar los objetivos de mitigación del cambio climático al disminuir las emisiones de gases de efecto invernadero asociadas a su consumo energético [3].

En Ecuador se planteó el plan Nacional de eficiencia energética en 2016 y afirma que El Balance Energético Nacional 2015 (BEN) (año base 2014) el consumo final de energía alcanzó 101 Mbep, lo que representó un incremento de 11% respecto al año anterior [4]. Este plan se encargará de que el Ecuador consuma menos cantidad de energía, analizando puntos estratégicos para crear proyectos de eficiencia energética.

En el artículo de “Modelamiento de Eficiencia Energética en el Consumo de Energía Eléctrica en Santa Elena a lo Largo del Tiempo, Zonas y Grupos”, se hizo un estudio en la provincia de Santa Elena, donde se recopiló datos de medidores de energía en 4 sectores diferentes y en Excel hizo un modelo matemático para una proyección hasta el año 2015 en ese sector, dando como conclusión un incremento de su potencia máxima del 25,74 kW al 32,15 kW [3].

En la actualidad, se han desarrollado diversos prototipos de sistemas de monitoreo eléctrico en tiempo real, como un sistema de monitoreo eléctrico inalámbrico en tiempo real. Este sistema proporciona a los usuarios la capacidad de acceder a información

detallada sobre su consumo eléctrico histórico a través de una interfaz gráfica generada mediante el uso del software NI LabView y un hardware controlado por la tarjeta programable Arduino [5].

Las redes neuronales artificiales tienen un amplio ámbito de aplicación, incluyendo la predicción del consumo de energía eléctrica, así como en otros sectores, como la industria y la agricultura. Por ejemplo, en el ámbito industrial, se han utilizado para identificar estados de madurez de frutas [6]. Asimismo, en el campo de la predicción del consumo de energía eléctrica, se han realizado investigaciones enfocadas en los factores que influyen en el consumo en el sector público, destacando la elección de las redes neuronales artificiales debido a su alta precisión [8].

1.2 DESCRIPCIÓN DEL PROYECTO

El proyecto consiste en el diseño de un prototipo de controlador para la eficiencia energética dentro de una residencia, el cual permite monitorear y predecir el consumo energético dentro de ella.

Los componentes físicos en la adquisición de datos son: dispositivos de protección eléctrica, equipos de medición, PZEM004T, equipo de comunicación, ESP32 y conductores eléctricos para su interconexión.

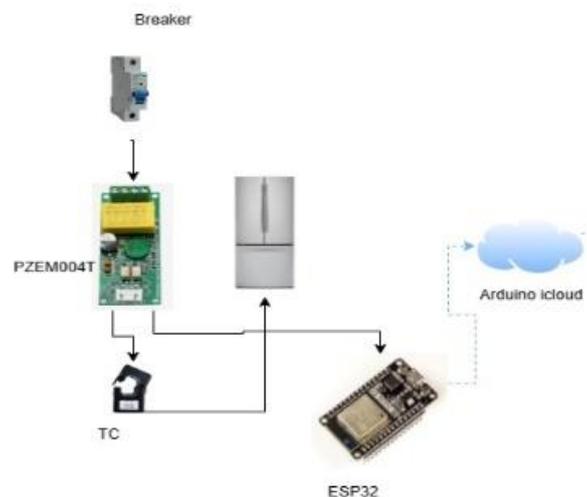


Figura 1. Etapa de adquisición de datos del prototipo

En la Figura 1 el ESP32 se enlaza con la red de la zona residencial para enviar los datos del consumo a la nube. Los componentes lógicos de este segmento son: El entorno arduino cloud, el enlace IFTTT, la base de datos en la hoja de cálculo en el drive de Google y el entorno de programación Matlab, para el diseño de la interfaz gráfica de monitoreo y predicción.

Estos valores son procesados en Matlab con el propósito de obtener una señal limpia que no afecte de manera incorrecta los datos de entrada de la red neuronal artificial. Con la ayuda de esta herramienta, se analiza el comportamiento del sistema y se procede a desarrollar una interfaz gráfica que muestra los resultados de la predicción del consumo eléctrico

1.3 OBJETIVOS DEL PROYECTO

1.3.1 Objetivo general

Implementar un prototipo de sistema de monitoreo y predicción del consumo eléctrico en zonas residenciales utilizando redes neuronales artificiales para obtener ahorro energético.

1.3.2 Objetivos específicos

Implementar un sistema de adquisición de datos con ESP32 para procesar y almacenar los datos del consumo energético en la nube.

Implementar redes neuronales para modelar el comportamiento del consumo energético en zonas residenciales.

Analizar los resultados del entrenamiento de la red neuronal y comparar con los datos del consumo eléctrico.

Desarrollar una interfaz gráfica para la visualización de las predicciones del consumo energético.

1.4 JUSTIFICACIÓN

En zonas residenciales, es donde hay más consumo de energía, con 7.959.12 GWh, ocupando un 34.77% del porcentaje total de las zonas de consumo energético en comparación con la industrial o comercial, en total la facturación del año 2021 fue de 1.963,62 millones de dólares [8].

En la tabla de “Evolución histórica del número de consumidores de las empresas distribuidoras” del ARCERNNR, muestra un claro crecimiento de consumidores de las empresas de distribución tanto en zonas residenciales como en la zona comercial e industrial [8], el consumo eléctrico va en aumento año a año, y los recursos no renovables que utilizan para la captación, transmisión y distribución de la energía eléctrica afecta al medio ambiente.

El costo del kWh varía de acuerdo con las diferentes distribuidoras de energía eléctrica, las cuales están divididas por sectores, incluso, existe un subsidio de consumo eléctrico. Adicional del costo del kWh esta relacionados con un horario de consumo, clasificación de nivel voltaje y de grupo de consumo, siendo esto primordial para el desarrollo de este proyecto, en diferentes zonas y en diferentes horarios producen diferentes costos del kWh según lo indica [2], incentivando a desarrollar esta tesis. Si se establece un monitoreo en el consumo energético en una zona residencial el cliente tiene la capacidad de saber cuánto está consumiendo en cualquier instante para así tener presente cuando el consumo afecta económicamente, con el objetivo de establecer un ahorro energético.

El desarrollo de este prototipo y su análisis consiste en ser la base para futuros proyectos en la Universidad Estatal Península de Santa Elena en el área de eficiencia energética, industria 4.0 y redes neuronales artificiales que son áreas de gran importancia actualmente y en la universidad son temas continuos de investigación. Las redes neuronales artificiales tienen gran impacto en aplicaciones de predicción y con el estudio del consumo eléctrico en la zona residencial es capaz de predecir el gasto en el consumo eléctrico, para que el usuario tenga la posibilidad de tener un ahorro energético.

1.5 METODOLOGÍA.

1.5.1 Investigación Exploratoria.

El presente trabajo de investigación es de tipo exploratoria, se ejecuta en zonas residenciales, primero se obtiene los datos de consumo energético con la planilla de CNEL y después con el prototipo se realiza la adquisición de los datos del consumo energético en tiempo real.

1.5.2 Investigación Descriptiva.

Con los datos obtenidos del consumo energético de CNEL y con el resultado final de la investigación, se realiza un análisis exhaustivo, comparando los datos de la planilla y los datos que se espera con la predicción del consumo usando redes neuronales artificiales.

CAPÍTULO II

2.1 MARCO CONTEXTUAL.

2.1.1 Consumo energético de zonas residenciales

La Agencia de regulación y control de energía y recursos naturales no renovables (ARCERNNR) en Ecuador proporciona información detallada sobre el consumo de energía en diferentes sectores, como zonas residenciales, industriales y comerciales. Esto se encuentra registrado en las estadísticas anuales y multianuales del sector eléctrico ecuatoriano. Según estos registros, el pliego tarifario establece categorías específicas, incluyendo la categoría residencial que cubre el servicio público de energía destinado exclusivamente al uso doméstico. Esta categoría abarca a aquellos que residen en unidades familiares independientes, sin importar la carga eléctrica utilizada. Además, se considera a los consumidores de bajos recursos económicos que tienen actividades comerciales o artesanales de pequeña escala [8].

El presente prototipo ha sido diseñado considerando las particularidades de una zona residencial en la provincia de Santa Elena, Ecuador. Está especialmente adaptado para una casa con cinco habitaciones, un baño y una sala, la cual es habitada por cuatro personas, incluyendo dos adultos y dos estudiantes universitarios. En esta área residencial, la asignación de categorías tarifarias para los consumidores es responsabilidad de la distribuidora de energía, la cual evalúa las características de la carga y el uso de energía declarado por los usuarios [2].

El prototipo comprende herramientas como el controlador ESP32 y el medidor de energía PZEM 004T, que permiten el monitoreo no invasivo de datos de consumo energético. La arquitectura en la nube se basa en Arduino Cloud e IFTTT para almacenar datos. A través de MATLAB, se realiza el análisis de datos históricos, la creación del modelo de redes neuronales artificiales y la implementación de la interfaz gráfica para supervisar y predecir el consumo eléctrico. Con los datos obtenidos, el sistema propuesto identifica patrones y tendencias específicas de cada artefacto eléctrico, utilizando estos patrones, el modelo de predicción basado en redes neuronales artificiales predice de manera precisa el consumo energético futuro de un artefacto en particular.

Esta capacidad de predicción proporciona a los usuarios una visión clara de cómo sus dispositivos eléctricos consumen energía en un período determinado, lo que les permite planificar y ajustar su consumo en consecuencia. Al calcular el valor en dólares del consumo de energía, el sistema de predicción ayuda en la planificación financiera de los usuarios, también al conocer el valor en dólares del consumo de un artefacto eléctrico específico, los usuarios pueden tener una comprensión más precisa de cuánto están gastando en electricidad. Esto les permite realizar presupuestos más efectivos y tomar decisiones informadas sobre cómo administrar su consumo energético para reducir costos.

2.2 MARCO CONCEPTUAL

2.2.1 Internet de las cosas

El internet de las cosas (IoT) por sus siglas en inglés es la conectividad de dispositivos físicos a través de Internet, posibilitando la comunicación y el intercambio de datos entre ellos. Los avances en electrónica han propiciado el desarrollo de sensores y dispositivos más avanzados y precisos para capturar información de equipos eléctricos. Estos sensores tienen la capacidad de supervisar diversos parámetros, como potencia, corriente, tensión y otros indicadores relevantes del desempeño eléctrico. La reducción de tamaño de estos sensores ha facilitado su integración en los propios equipos, lo que permite obtener datos en tiempo real sin afectar su funcionamiento normal [9]. El IoT está compuesto por múltiples capas, cada una con un propósito específico:

Capa de Percepción

En la capa de percepción se encuentran los sensores que son dispositivos para captar un estímulo de su entorno y traducir esa información que recibe. Esa información recibida es normalmente convertida a un impulso eléctrico que posteriormente es procesado por una serie de circuitos que generan una acción predeterminada en un aparato, sistema o máquina [11].

Capa de Comunicación

Esta capa se encarga de transmitir los datos de los sensores a través de dispositivos afines, como el microcontrolador. Un microcontrolador es un dispositivo electrónico capaz de llevar a

cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario y se introducen en él a través de un programador [9].

Capa de Conectividad

La capa de conectividad es la infraestructura necesaria para que los dispositivos IoT se comuniquen entre sí. Se utilizan protocolos de comunicación y servicios de conectividad, entre otros [9].

Protocolo de comunicación UART

El protocolo de comunicación UART (Universal Asynchronous Receiver-Transmitter) es un componente fundamental en los sistemas de comunicación serie. Su función principal es convertir los datos serie a paralelos durante la recepción (entrada) y convertir los datos paralelos a serie durante la transmisión (salida). Este protocolo es utilizado en diversas aplicaciones, como comunicaciones con PC o PDA, transmisiones de audio y video, juegos mediante infrarrojos o Bluetooth en teléfonos móviles, y en matrices de diodos para paneles de atención al cliente, vallas publicitarias, entre otros [11].

Protocolo de comunicación MQTT

El protocolo de comunicación MQTT (Message Queuing Telemetry Transport) es un protocolo de publicación-suscripción que facilita la conexión de una amplia variedad de dispositivos. Este protocolo ofrece la capacidad de verificar que los mensajes lleguen a su destino y proporciona seguridad mediante el uso de certificados electrónicos. Los objetivos del protocolo MQTT son minimizar el ancho de banda, permitir la comunicación bidireccional entre dispositivos, minimizar los recursos y el consumo de los dispositivos, y garantizar la fiabilidad y cierto grado de seguridad. En MQTT, los mensajes se envían de manera asíncrona, lo que significa que no es necesario esperar una respuesta después de enviar un mensaje [12].

Protocolo de comunicación HTTP

El protocolo de comunicación HTTP (Hypertext Transfer Protocol) se basa en operaciones

de solicitud y respuesta. Un cliente establece una conexión con un servidor y envía un mensaje de solicitud con los datos requeridos. El servidor responde con un mensaje que contiene el estado de la operación y su resultado potencial. Todas las operaciones de HTTP pueden adjuntar un objeto o recurso identificado por una URL (Uniform Resource Locator) [13].

Capa de Aplicación

La capa de aplicación se encarga del desarrollo de aplicaciones y servicios que utilizan los datos recopilados por los dispositivos IoT, incluyendo sistemas de monitoreo. En este proceso, es posible utilizar varias plataformas o una sola para el desarrollo de estas aplicaciones.

Los simuladores numéricos son ejemplos de plataformas ampliamente utilizadas en el ámbito de la programación y el cálculo numérico. Estos simuladores gozan de popularidad entre ingenieros y científicos debido a su capacidad para analizar datos, desarrollar algoritmos y construir modelos [14].

Capa de Procesamiento y Análisis

En la capa de análisis y procesamiento, se realiza un procesamiento de señales con el propósito de asegurar la precisión de los datos recopilados por los sensores. Este procedimiento está destinado a garantizar la corrección y la exactitud de los datos recopilados, permitiendo obtener información significativa [9].

2.2.2 Procesamientos de señales

El procesamiento de señales es un campo esencial en ingeniería que se dedica a la manipulación y análisis de señales con el fin de extraer información relevante o mejorar su calidad. Existen tres métodos de filtro muy utilizados: el filtro Savitzky-Golay, el filtro de mediana y el filtro de media móvil. Estos métodos son aplicados para filtrar y suavizar las señales, con el objetivo de reducir el ruido y resaltar características importantes.

Filtro Savitzky-Golay

El filtro se fundamenta en la aplicación de una técnica de regresión lineal mediante mínimos cuadrados, suponiendo una función $F(x)$ con ruido, la idea clave para realizar el suavizado a esta función se obtiene tomando un promedio de los datos vecinos. Se ajusta un polinomio de grado M a través de un número fijo de puntos, para obtener el valor de suavizado g_i en x_i . Se denota el número de puntos a la derecha de x_i , por n_R entonces $g_i = p_i(x_i)$. El polinomio $p_i(x)$, de grado M se denota en la siguiente ecuación [15].

$$p_i(x) := \sum_{k=0}^M b_k \left(\frac{x - x_i}{\Delta x} \right)^k \quad (1)$$

Donde:

- x_i representa una abscisa específica en los datos.
- b_k son los coeficientes que ajustan el polinomio para minimizar los errores.
- M es el grado del polinomio de ajuste.
- Δx es la diferencia entre abscisas adyacentes.

La determinación de los coeficientes b_k sigue la estructura que se describe a continuación:

$$\sum_{j=i-n_L}^{i+n_R} (p_i(x_j) - f_j)^2 = \min \quad (2)$$

- $p_i(x_j)$ es el valor del polinomio en el punto x_j .
- f_j son los datos medidos.
- n_L y n_R son el número de puntos a la izquierda y derecha de x_i considerados.

Se define la matriz A.

$$A = \begin{pmatrix} -n_L^M & \cdots & -n_L & 1 \\ \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \\ \vdots & \cdots & \vdots & \vdots \\ n_R^M & \cdots & n_R & 1 \end{pmatrix} \in R^{(n_L+n_R+1)*(M+1)}$$

Y los dos vectores

$$B = \begin{pmatrix} b_M \\ \vdots \\ b_1 \\ b_0 \end{pmatrix} \in R^{M+1}$$

$$f = \begin{pmatrix} f_{i-n_L} \\ \vdots \\ f_i + n_R \end{pmatrix} \in R^{n_L+n_R+1}$$

La matriz A no depende de la abscisa x_i ni de Δx . Usando estas definiciones se reformula el problema de mínimos cuadrados ecuación (2).

$$\|Ab - f\|_2 = \min \tag{3}$$

Ahora se resuelve la ecuación (3) por medio de descomposición de A conociendo $g_i = p_i(x_i) = b_0$ la solución de b se expresa como la solución normal de la ecuación

$$A^T A b = A^T f \quad (4)$$

Obteniendo:

$$g_i = e_{M+1}^T (A^T A)^{-1} A^T f \quad (5)$$

En esta expresión, se calcula el valor suavizado g_i en el punto x_i usando el vector de coeficientes b y la matriz de diseño A^T , Representado g_i como una combinación lineal de f_i se define el vector.

$$c = A(A^T A)^{-1} e_{M+1} \quad (6)$$

- c contiene los coeficientes del filtro que ponderan los datos en el proceso de suavizado.
- A es la matriz de diseño.
- e_{M+1} es un vector de uno de tamaño $M+1$.

La descomposición de mínimos cuadrados ecuación (4) se resuelve de manera estable por descomposición QR. Denotamos una matriz ortogonal $(nL + nR + l)$ por $(M + 1)$, y por R denotamos una matriz triangular superior, sí sustituimos y descomponiendo la ecuación (4) en (7) tenemos.

$$c = \frac{1}{M^T + 1, M + 1} Qe_{M+1}. \quad (7)$$

Una vez evaluados los parámetros de filtro c para valores específicos de n_R , n_L y M , no es necesario volver a calcularlos. Por lo tanto, cada valor filtrado g_i se puede obtener de manera eficiente mediante un simple producto escalar de longitud $n_R + n_L + 1$, como se muestra en la ecuación (8).

Para simplificar el proceso matemático, se recurre a simuladores numéricos que agilizan estas operaciones. Entre ellos, se encuentra la función "sgolayfilt", la cual es utilizada para aplicar un filtro Savitzky-Golay en Matlab. Al emplear esta función, se requiere definir dos parámetros esenciales: la longitud de la ventana del filtro [11].

Filtro de Mediana

El filtro de la mediana es una técnica de procesamiento de señales que se utiliza ampliamente para eliminar ruido impulsivo de una señal. Dada una señal unidimensional, el valor filtrado en cada punto de la señal (excepto en los bordes) se obtiene de la siguiente manera:

$$y_i = \text{mediana}(\{x_{i-k}, x_{i-k+1}, \dots, x_i, \dots, x_{i+k}\})$$

Donde

- y_i es el valor filtrado en el punto i de la señal.
- *Median* es la función mediana que calcula el valor mediano de los elementos en la ventana deslizante
- k es la mitad del tamaño de la ventana.

La ventana deslizante se mueve a lo largo de la señal, y en cada posición, se selecciona el valor mediano de los elementos en la ventana para reemplazar el valor original en ese punto. En Matlab,

se utilizar la función medfilt1 para aplicar el filtro de la mediana a una señal unidimensional [17].

Filtro de media móvil

En un filtro de media móvil de largo M, la salida actual consiste en el promedio de las últimas M muestras de la entrada [16].

La función de transferencia de este filtro es:

$$y[n] = \frac{1}{m} \sum_{k=0}^{M-1} x[n - k] \quad (8)$$

- $y[n]$: Es la salida del filtro en el instante de tiempo n.
- $x[n]$: Es la entrada del filtro en el instante de tiempo n.
- M: Es la longitud del filtro, es decir, el número de muestras que se promedian.

En el simulador matemático, se utiliza la función “movmean” para calcular la media móvil de una señal. Esta función calcula la media móvil de una serie de datos utilizando una ventana de tamaño k. En cada posición de la ventana, se promedian los valores de k puntos, y el resultado se almacena en la correspondiente posición en la serie de datos de salida.

2.2.3 Redes Neuronales Artificiales

El libro "Redes Neuronales: Conceptos Básicos y Aplicaciones" de los profesores Carlos Alberto Ruiz y Marta Susana Basualdo presenta distintas perspectivas para definir las redes neuronales, desde enfoques breves y genéricos hasta explicaciones más detalladas. Una de las definiciones propuestas es la siguiente: las redes neuronales son "una nueva forma de computación, inspirada en modelos biológicos". Esta definición destaca que las redes neuronales son

consideradas como una forma innovadora de computación que se basa en la inspiración de los modelos biológicos presentes en el sistema nervioso [11].

Elementos básicos de una red neuronal artificial.

Neurona artificial

La neurona, es la unidad básica de procesamiento en una red neuronal. En una neurona artificial realiza cálculos y operaciones matemáticas en su entrada para generar una salida. Consiste en tres componentes principales: entradas, pesos y una función de activación.

Función de entrada:

La neurona tiene la capacidad de procesar múltiples valores de entrada como si fueran un único valor, conocido como entrada global. Para lograr esto, se utiliza la función de entrada, la cual combina estas entradas individuales en la entrada global gin_i . El cálculo de la función de entrada se realiza a partir del vector de entrada, y su ecuación se describe de la siguiente manera:

$$input_i = (in_{i1} \times w_{i1}) * (in_{i2} \times w_{i2}) * \dots (in_{in} \times w_{in})$$

Donde \times representa al operador apropiado (máximo, sumatoria, productoria), al número de entradas a la neurona in_i y w_i al peso. Algunas funciones de entrada más comunes y utilizadas son:

Sumatoria: es la suma de todos los valores de entrada a la neurona, multiplicados por sus correspondientes pesos.

$$\sum_j (n_{ij}, w_{ij}), \text{ con } j = 1, 2, \dots, n \text{ ec.}$$

Función de activación:

La función de activación es una función que opera sobre la entrada global de una neurona en la RNA. Se introduce para introducir no linealidades en el modelo, lo que permite capturar

relaciones complejas entre las entradas y las salidas de la red como ejemplos se tiene las siguientes funciones de activación.

Función lineal

$$F(x) = \begin{cases} -1 & x \leq -\frac{1}{a} \\ a * x & -\frac{1}{a} < x < \frac{1}{a} \\ 1 & x \geq \frac{1}{a} \end{cases}$$

$$x = g(gin_i - \theta_i) \text{ y } a > 0$$

Donde x es la entrada global, g es una función no lineal y θ es el umbral.

Función sigmoidea:

$$f(x) = \frac{1}{1+e^{-gx}} ; x = gin_i - \theta_i$$

Los valores de salida proporcionan corresponden en el rango de 0 a 1. Al modificar el valor de gin_i se ve afectada la pendiente de la función de activación.

Función tangente hiperbólica

$$f(x) = \frac{e^{gx} - e^{-gx}}{e^{gx} + e^{-gx}} ; x = gin_i - \theta_i$$

La ecuación corresponde a la función tangente hiperbólica comprendida dentro de un rango que va de -1 a 1, si se modifica el valor de g se ve afectada la pendiente de la función de activación.

Función de salida

El resultado obtenido a partir de esta función representa la salida de la neurona i_{out} . En consecuencia, la función de salida determina qué valor se transmite a las neuronas conectadas. Si la función de activación está por debajo de un umbral específico, ninguna salida se transmite a la

neurona subsiguiente. Existen diferentes tipos de funciones de activación que pueden utilizarse en una red neuronal:

Función nula: En este caso, la salida de la neurona es idéntica a la entrada, sin realizar ninguna transformación.

Función binaria: La salida toma el valor de 1 si la acción supera el umbral establecido, y 0 en caso contrario. Esta función proporciona una salida discreta y binaria en función de si se alcanza o no el umbral establecido.

Arquitectura de una red neuronal artificial

La arquitectura de una red neuronal se refiere a la forma en que las neuronas están organizadas y dispuestas dentro de la red. Esta organización incluye la formación de capas o agrupaciones de neuronas que pueden estar más cerca o más lejos de las entradas y salidas de la red. Al diseñar una red neuronal, es importante considerar varios parámetros clave, como el número de capas, el número de neuronas en cada capa, el grado de conectividad entre ellas y el tipo de conexiones utilizadas.

2.2.4 Aprendizaje de la red neuronal artificial

En el aprendizaje supervisado se pueden identificar tres enfoques principales, que dan origen a los siguientes tipos de aprendizaje supervisado:

- Aprendizaje por corrección de error.
- Aprendizaje por refuerzo.
- Aprendizaje estocástico.

El estudio de las diferentes formas de aprendizaje en las redes neuronales desempeña un papel fundamental en el campo del aprendizaje automático. Estas herramientas computacionales

demuestran su capacidad para adquirir conocimientos y mejorar su rendimiento a través de la experiencia. Por lo tanto, comprender cómo las redes neuronales aprenden y se adaptan a los datos de entrada es esencial para su aplicación efectiva en una amplia gama de dominios. En esta sección, se explora una de las principales modalidades de aprendizaje en las redes neuronales como el aprendizaje supervisado.

Entrenamiento de la red neuronal artificial.

El aprendizaje supervisado se caracteriza por ser un proceso en el cual el entrenamiento de la red neuronal se realiza bajo la supervisión y control de un agente externo, como un supervisor o maestro. Este agente determina la respuesta que la red debería generar ante una determinada entrada. El supervisor monitoriza la salida de la red y, en caso de que no coincida con la respuesta deseada, se procede a ajustar los pesos de las conexiones con el objetivo de acercar la salida obtenida a la deseada.

Redes de propagación hacia atrás (backpropagation)

El nombre "backpropagation" (BP) es un algoritmo de entrenamiento de RNA, se utiliza la regla de la cadena para minimizar el error cuadrático medio, su nombre se debe a la forma en que el error se propaga hacia atrás a través de la red neuronal, específicamente desde la capa de salida hacia las capas ocultas. Este proceso permite ajustar los pesos de las conexiones entre las neuronas en las capas ocultas durante el entrenamiento, en base al error cometido. Un método que refina la técnica de backpropagation es el algoritmo Levenberg-Marquardt, que combina aspectos del método de Gauss-Newton y el método del gradiente descendente [18] .

El algoritmo Levenberg-Marquardt busca minimizar la función de error $f(w)$, la cual es una suma de los cuadrados de los errores individuales e_i en el conjunto de datos de entrenamiento:

$$f(w) = \sum_{i=0}^m e_i^2(w) \quad (9)$$

Donde m es la longitud de los datos de entrenamiento. En este enfoque, se estima una matriz Jacobiana aproximada (H) y un vector gradiente, y se incorpora un término regulador μ en la matriz H .

$$H \approx J^T \cdot J + \mu \cdot I$$

La actualización de los pesos (W) de las neuronas se calcula mediante la siguiente expresión:

$$W_{nuevo} = W_{actual} - [J^T \cdot J + \mu \cdot I]^{-1} \cdot J^t \cdot E$$

Donde E es la matriz de errores en la red, I es la matriz identidad y μ es un valor escalar. La matriz Jacobiana se obtiene a partir de la derivada de los errores con respecto a los pesos, y cada elemento de la matriz Jacobiana $J_{ij}(w)$ se calcula considerando las derivadas de los errores individuales con respecto a los pesos [19] :

$$J_{ij}(w) = \frac{\partial e_i}{\partial w_i} \quad (10)$$

Validación

Una vez finalizado el proceso de entrenamiento, los pesos de las conexiones en la red neuronal se mantienen fijos. El siguiente paso consiste en evaluar si la red neuronal puede resolver nuevos problemas de naturaleza general para los cuales ha sido entrenada. Para lograr esto, es necesario contar con un conjunto adicional de datos llamado conjunto de validación o conjunto de pruebas. Cada ejemplo en este conjunto de evaluación incluye los valores de las variables de entrada, pero sin proporcionar a la red neuronal la solución correspondiente.

2.2.5 Desempeño de RNA

Para evaluar la calidad del rendimiento de una RNA, se emplean métricas específicas como el Error Cuadrático Medio (MSE) y la Raíz del Error Cuadrático Medio (RMSE) y el coeficiente de determinación (R^2). Estas métricas proporcionan una medida objetiva para determinar la eficacia de la RNA en su funcionamiento [20].

Error cuadrático medio (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (11)$$

Donde:

- n es el número total de observaciones.
- y_i es el valor real en la observación.
- \hat{y}_i es el valor predicho por la RNA.

Entonces el MSE representa la magnitud de las diferencias entre las predicciones y los valores reales.

Raíz del error cuadrático medio (RMSE)

La Raíz del Error Cuadrático Medio (RMSE) es la raíz cuadrada del MSE y proporciona una medida de la dispersión de los errores entre los valores reales y los valores predichos.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (12)$$

Coefficiente de determinación (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (13)$$

- y_i son los valores reales.
- \hat{y}_i son los valores predichos de la RNA.
- \bar{y} es el valor medio de todos los valores reales.

El coeficiente compara la varianza de las predicciones de la red neuronal con la varianza total en los valores reales, e indica la precisión con la que las predicciones se ajustan a los datos reales[20].

Un alto desempeño en la RNA es esencial para lograr predicciones precisas del consumo eléctrico. Estas predicciones, respaldadas por datos eficientes, no solo resultan útiles en sí mismas, sino que también permiten realizar análisis que culminan en la obtención de una eficiencia energética óptima. Estos análisis, basados en las respuestas de la RNA, desempeñan un papel fundamental en la evaluación y mejora de la eficiencia energética.

2.2.6 Eficiencia energética.

La eficiencia energética (EE) se define como el cociente entre la energía requerida para desarrollar una actividad específica, y la cantidad de energía primaria usada para el proceso. Se considera una parte esencial del futuro de la energía sustentable, ya que permite la disminución del consumo de energía, los gases de efecto invernadero y las emisiones, y a la vez genera oportunidades de inversión, facilitando la creación adicional de nuevos puestos de trabajo [21]. Es importante destacar que garantizar una alta calidad y un uso adecuado de las eficiencias energéticas requiere la observancia de ciertas normativas y estándares.

2.2.7 Norma ISO 50001

La norma ISO 50001 pretende reducir el consumo de energía con beneficios de reducción de costos, minimizar las emisiones de gases de efectos invernaderos y mejorar la seguridad en el suministro mediante un enfoque estructurado. [22].

2.2.8 Norma ISO 50006

La norma ISO 50006 proporciona directrices para medir el desempeño energético utilizando líneas de base energéticas en sistemas de gestión de la energía. Establece un enfoque estructurado y consistente para establecer líneas de base energéticas y evaluar la eficacia de las acciones y medidas implementadas para mejorar la eficiencia energética [23].

2.2.9 Pliego tarifario del servicio eléctrico.

El servicio eléctrico en Ecuador está respaldado por un pliego tarifario que establece las pautas y regulaciones que las empresas de distribución eléctrica deben seguir. Este pliego tarifario abarca diversas indicaciones relacionadas con la tarificación y regulación del consumo de electricidad en el país [24].

Tabla 1. Clasificación del nivel de voltaje.

Nivel de voltaje - NV	Grupo	Voltaje de Suministro en el punto de entrega
Bajo		Menor o igual a 0,6 KV
Medio		Mayor a 0,6 y menor a 40 KV
Alto	Grupo 1-Av1	Mayor a 40 y menor igual a 138 Kv
	Grupo 2-Av2	Mayor a 138 KV

Elaborador por: ARCERNNR indicando la clasificación del nivel de voltaje de suministro en el punto de entrega del consumidor [2].

El servicio público de energía eléctrica (SPEE) considera dos categorías de tarifas, que depende de las características del consumidor, en la Tabla 1.1 se visualiza la clasificación a nivel de voltaje [2].

Tabla 2. Clasificación de categorías tarifarias y los niveles de voltaje.

Categoría	Nivel de Voltaje-NV	Grupo de Consumo.	Registro de Demanda
Residencial	Bajo Voltaje NV<600 V	Residencial	Sin demanda

Elaborador por: ARCERNNR es la clasificación por el grupo de consumo, su voltaje y registro de demanda [2].

En la tabla 2 se visualiza la clasificación por tarifas y niveles de voltaje, en la categoría residencial está considerada con un nivel de voltaje bajo y como registro sin demanda. Indicando que tiene un cargo de comercialización por mes y un cargo de incremento de consumo energético [2].

Facturación:

La facturación SPEE se calcula sumando varios elementos que incluyen energía consumida, potencia demandada, pérdidas en transformadores, costos de comercialización y una penalización por un factor de potencia bajo. La Corporación Nacional de Electricidad (CNEL) aplica una tarifa mensual basada en la demanda máxima durante el horario de 18:00 a 22:00, que representa el pico de consumo energético (P). Para los consumidores regulados, la fórmula utilizada para calcular la tarifa es la siguiente [2].

$$FSPEE = E + P + PIT + C + P_{bfp} \quad (14)$$

Donde:

$FSPEE$ =Factura por servicio público de energía eléctrica-USD

E = Facturación de energía (USD).

P =Facturación de demanda (USD).

PIT =Pérdidas en transformadores (USD).

C =Comercialización (USD).

P_{bfp} =Penalización por bajo factor de potencia (USD).

Estos componentes técnicos, incluye costos de comercialización (C) y una penalización por un factor de potencia bajo P_{bfp} . Los costos de comercialización cubren los gastos asociados con la administración y operación del sistema eléctrico, así como los servicios adicionales

proporcionados a los clientes. Por otro lado, la penalización por un bajo factor de potencia se aplica cuando el cliente no utiliza eficientemente la energía eléctrica, lo que puede tener un impacto en la estabilidad y eficiencia del sistema.

2.3 MARCO TEÓRICO.

Los siguientes trabajos investigativos son guías donde se detalla el proceso de estudio que apoya a la presente propuesta de titulación.

En el artículo titulado "Electricity price forecasting using artificial neural networks", escrito por Deepak Singhal y K.S. Swarup, se analizan diversas variables utilizadas como entradas en el entrenamiento de una red neuronal. Estas variables incluyen el día de la semana, la demanda actual de electricidad y su cambio en comparación con el período anterior. Además, se incorporan datos históricos de precios de los días anteriores, semanas anteriores, así como de tres y cuatro semanas atrás. Estos datos se utilizan para proporcionar a la red neuronal la información necesaria para predecir los precios futuros de la electricidad. El enfoque planteado en el artículo se basa en aprovechar la capacidad de las redes neuronales artificiales para identificar patrones complejos y no lineales en los datos de entrada. Al considerar múltiples variables y su relación temporal, se espera lograr pronósticos más precisos y confiables sobre los precios de la electricidad [25].

La tesis de la Mg. María José Ramírez Prado, titulada "Diseño y prototipo de un sistema de monitoreo de consumo eléctrico residencial utilizando una plataforma virtual de Internet de las cosas", investiga a fondo la recopilación de datos de consumo eléctrico en hogares residenciales y la implementación del Internet de las cosas (IoT) a través de la plataforma IFTTT para su monitoreo. El objetivo principal de esta investigación es crear un sistema que proporcione información detallada sobre el consumo de electricidad en residencias. Para lograrlo, se utilizan diversas técnicas y dispositivos de medición para recopilar datos precisos sobre el consumo eléctrico. Estos datos son esenciales para comprender los patrones de consumo y tomar decisiones informadas con respecto a la eficiencia energética [26].

La tesis titulada "Diseño de un sistema de monitoreo inalámbrico de parámetros eléctricos

en cargas residenciales", se centra en el desarrollo de un sistema innovador para el monitoreo de parámetros eléctricos en hogares residenciales de forma inalámbrica y diseñar un sistema que permita la monitorización y seguimiento de diversos parámetros eléctricos, tales como consumo de energía, voltaje y corriente, en cargas residenciales. Para lograrlo, se emplea una tecnología inalámbrica que elimina la necesidad de conexiones físicas, lo que proporciona flexibilidad y facilidad de uso [27].

En el proyecto "Desarrollo de un sistema de monitoreo en tiempo real para el consumo de energía eléctrica" realizado por el Lic. Ramón de Jesús López Suárez, se lleva a cabo el diseño y la implementación de un circuito electrónico para el procesamiento de señales y el control de un sensor instalado. Este sistema permite la medición, registro, visualización y análisis del comportamiento del consumo eléctrico, así como la determinación del consumo total de energía. Además, se desarrolla una interfaz intuitiva que facilita el control y la supervisión de los datos obtenidos [28].

CAPITULO III

3.1 COMPONENTES DE LA PROPUESTA

Para la adquisición de datos del consumo eléctrico de la zona residencial se utiliza componentes físicos (Sistemas de protección eléctrica, Modulo Pzem 004T y Esp32) y componentes lógicos (Arduino IoT cloud, ifttt, neuronal net fitting, thingspeak).

3.1.1 Componentes físicos.

PZEM 004T

El pzem de monitoreo es un módulo que permite medir el consumo de cargas, adaptándose a diversas capacidades, se conecta con un transformador de corriente y a través de una comunicación TTL, el módulo proporciona información detallada incluyendo el amperaje, voltaje, potencia, energía y factor de potencia, el porcentaje de error de sus mediciones se visualizan en Anexo 1.

Sensor de corriente PZCT-02

El sensor de corriente PZCT-02 no invasivo cuenta con una bobina primaria de 100 A y una bobina secundaria de 5 A, es un complemento que necesita el módulo de potencia pzem 004T para capturar el amperaje del circuito eléctrico.

Esp32 node mcu

El esp32 node mcu tiene un módulo wifi integrado, recibe los datos de potencia del pzem 004T mediante la comunicación TTL, su programación es escrita en lenguaje C, sus características se encuentran en Anexo 1.

3.1.2 Componentes Lógicos.

Arduino IoT cloud

En esta plataforma, se encuentran disponibles diversas herramientas que permiten programar múltiples tarjetas como el Esp32. Además, se dispone de numerosas librerías y una interfaz intuitiva para agregar "Dashboards" o "Things".

Ifttt

Esta página web proporciona una plataforma para programar diversas acciones que conectan el Arduino iCloud con otras aplicaciones. En este caso particular, se ha creado un programa que permite enlazar los datos obtenidos de arduino IoT cloud y visualizarlos en una hoja de datos en Google Drive.

Neuronal net fitting

La aplicación NEURONAL NET FITTING, disponible en la versión 2020B de Matlab, permite crear redes neuronales artificiales. Con esta aplicación, se ingresan los datos de entrada y salida y se siguen los pasos detallados para entrenar la red neuronal. Una vez completado el proceso, la aplicación proporciona una función que representa la red neuronal artificial.

App designer

En Matlab 2020B, existe un apartado denominado "App Designer" que permite crear interfaces. Esta función posibilita mostrar los resultados de la predicción del consumo eléctrico de manera que el usuario pueda apreciarlos y utilizarlos de forma intuitiva.

Thingspeak

Los datos de potencia obtenidos por el Esp32 se envían a la plataforma de la nube ThingSpeak para establecer una conexión con Matlab. De esta manera, los datos se pueden visualizar en tiempo real y monitorear directamente desde Matlab.

3.2 DISEÑO DE LA PROPUESTA

El diseño de la propuesta de este prototipo se inicia seleccionando los dispositivos específicos de los cuales se desea medir la potencia eléctrica, priorizando aquellos que tienen un mayor impacto en el consumo energético. A continuación, se implementa un sistema de adquisición de datos del consumo eléctrico para cada dispositivo.

Se configura el hardware y software para la adquisición de datos utilizando la plataforma Arduino IoT Cloud. Se desarrolla un enlace mediante IFTTT entre la nube de Arduino y Google Drive, permitiendo que los datos de consumo eléctrico se envíen y almacenen automáticamente en un archivo con formato XLS en una carpeta específica de Google Drive. Todo este proceso tiene como objetivo crear un sistema de predicción por medio de modelos entrenados de redes neuronales artificiales (RNA). Estos modelos utilizarán los datos de consumo eléctrico almacenados en Google Drive para entrenarse y, una vez entrenados, se utilizan para realizar predicciones sobre el consumo eléctrico futuro.

Se realiza un análisis de identificación de datos específicos de cada artefacto que pueden influir en el entrenamiento. Posteriormente, se lleva a cabo el procesamiento de señales de potencia eléctrica, con el fin de obtener datos correctos para el entrenamiento de la RNA. Este procesamiento permite extraer información relevante y eliminar posibles interferencias o ruido en las mediciones de potencia eléctrica, lo que contribuye a mejorar la precisión de los resultados de la red neuronal.

Una vez completado el análisis de identificación de datos y procesamiento de señales, se procede a la configuración de la app de Matlab "Neuronal Net Fitting". Esta etapa implica establecer los parámetros de la red neuronal, como el número de capas ocultas, el número de neuronas por capa, la función de activación y los algoritmos de optimización. La configuración de "Neuronal Net Fitting" se realiza de forma individual para cada artefacto, con el objetivo de adaptar la red neuronal a las características y patrones específicos de consumo de energía de cada dispositivo.

Se diseña una aplicación de escritorio que permite visualizar los resultados obtenidos durante el entrenamiento y análisis de la red neuronal. Esta aplicación proporciona una interfaz amigable para visualizar los patrones de consumo de energía, generar gráficos y realizar análisis más detallados de los datos recopilados.

3.2.1 Elección de artefactos eléctricos

Con el objetivo de recopilar datos sobre el consumo energético en la zona residencial bajo estudio, se realiza una evaluación para identificar aquellos dispositivos que tienen un mayor impacto en el consumo.

Tabla 3. Datos de artefactos seleccionados

Artefacto	AREA (m^2)	CAT	VOLTAGE (V)	CORRIENTE (A)	POTENCI A (W)	CAPACIDAD (litros)	CONSUMO DE ENERGIA. kWh/mes
Nevera	30	A	120	2	265	500	37,5
Aire A.	35	B	120	10	822	-	288
Iluminaria	60	C	120	1.6	200	-	144

En la tabla 3 se detalla, dos artefactos eléctricos específicos, así como de un circuito de iluminación, 10 focos ahorradores de 20 W. La selección de estos equipos está relacionada con el mayor porcentaje de consumo eléctrico dentro de la vivienda.

3.2.2 Implementación del sistema de adquisición de datos del consumo eléctrico.

Una vez seleccionados los artefactos eléctricos junto con sus respectivas características, se procede a diseño eléctrico y electrónico para la adquisición y transferencia de datos a la nube.

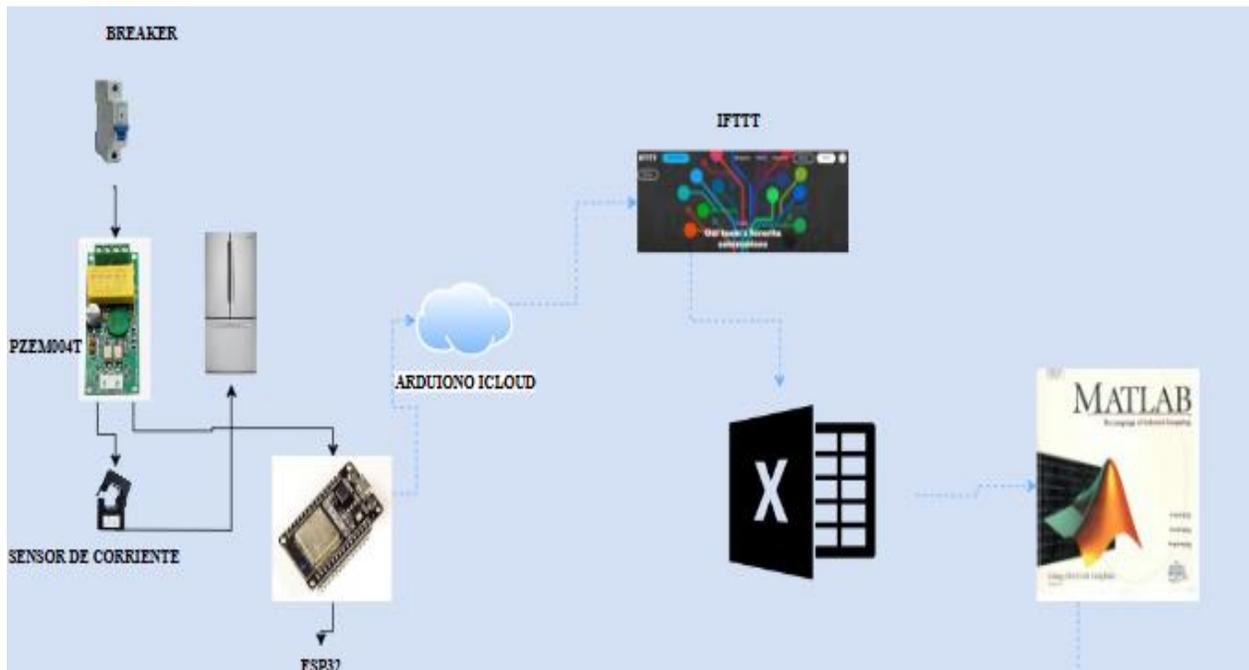


Figura 2. Conexiones físicas y lógicas.

La figura 2 muestra las conexiones físicas y lógicas, la conexión se establece utilizando el ESP32 y su comunicación TTL, el cual se conecta al Pzem 004t. A su vez, el Pzem 004t se enlaza con la carga. De esta manera, se logra una configuración completa que permite medir y monitorear el consumo energético de los artefactos eléctricos seleccionados.

El ESP32 se conecta mediante el protocolo MQTT hacia la plataforma Arduino IoT Cloud, lo que posibilita la captura de los datos de consumo eléctrico. Además, se establecen conexiones con las plataformas IFTTT y ThingSpeak. Estas conexiones habilitan el envío de los datos hacia Google Drive, donde se almacenan y están disponibles para su posterior análisis. Luego, estos datos pueden procesarse en MATLAB, aprovechando los resultados obtenidos de la red neuronal implementada.

3.2.3 Configuración de elementos lógicos.

El siguiente flujograma representa el proceso de obtención de datos del consumo eléctrico hacia Google Drive realizado por el prototipo.

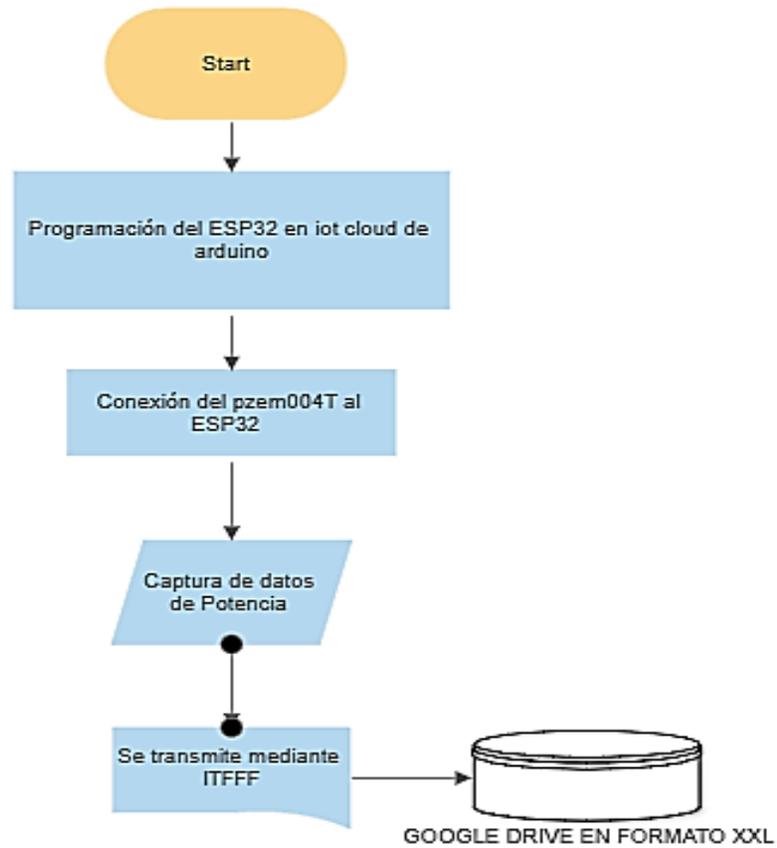


Figura 3. Flujograma de configuración de elementos lógicos.

Este procedimiento se repite individualmente para cada artefacto eléctrico que presenta un mayor consumo en la zona residencial en estudio. A través de este flujo de trabajo, el prototipo recopila eficientemente la información requerida, lo que posibilita un análisis del consumo eléctrico de cada artefacto en esa área específica.

Programación del Esp32.

Se crean los dispositivos en Arduino IoT Cloud en la pestaña 'THINGS'. Para este caso, se utiliza el modelo NodeMCU-32S.

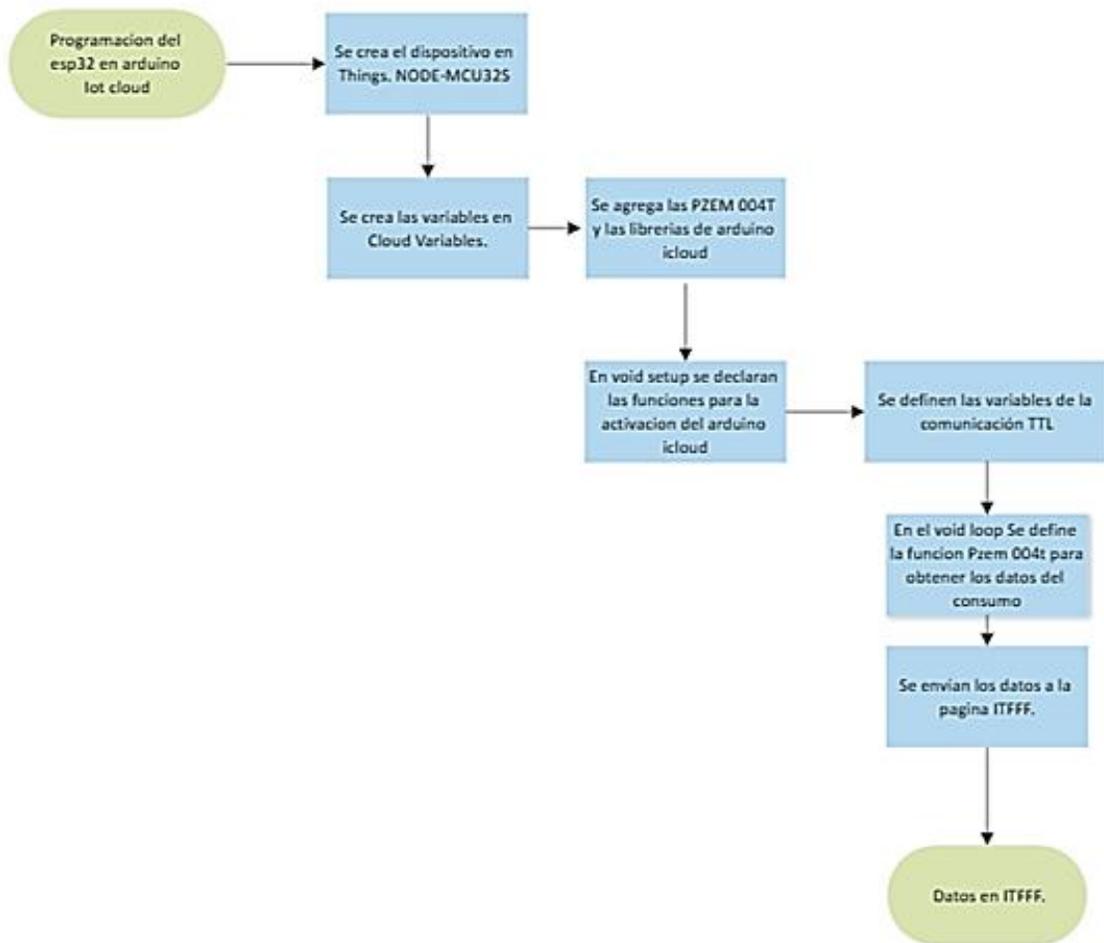


Figura 4. Flujograma de programación del ESP32.

A continuación, se agrega la red local a utilizar en la zona residencial y se añaden las variables necesarias en el apartado de 'Cloud Variables'. Esta plataforma proporciona automáticamente un sketch para facilitar la programación en la nube. Luego, se procede a programar en el sketch, añadiendo las librerías necesarias que son ofrecidas en la misma plataforma. Las variables se definen con el mismo nombre que se les ha dado en las 'Variables de nube'. Para obtener los valores, se utiliza las funciones que ofrece la librería de PZEM. A partir de ahí, es posible separar los valores en variables independientes y a su vez enviarlos a thinkspeak. Para más detalles de la

programación consulte Anexo 3(código 1).

Programación en IFTTT

En la página IFTTT, se crea el evento y se enlaza con una cuenta de Google. Además, la misma página proporciona un enlace con la programación creada. Este enlace se agrega en el apartado de webhook de la pestaña 'THINGS' de Arduino IoT Cloud. Como resultado, se generan automáticamente hojas de cálculo con los datos del consumo eléctrico en Google Drive para información más detallada ir al Anexo 4.

3.2.4 Topología de la red neuronal artificial.

Identificación de datos.

En la identificación de datos para la entrada de la red RNA, se analiza los datos que influyen en cada artefacto seleccionado. Los datos en común que influyen en cada artefacto son:

- Fecha: El factor temporal afecta la forma en que se utiliza la nevera y, por ende, su consumo de energía en el presente.
- Días festivos (entrada lógica): Durante días festivos o celebraciones especiales, es común que se almacenen diferentes tipos de alimentos y bebidas, lo cual puede tener un impacto en el consumo energético de la nevera en el presente.
- Día de la semana: Los hábitos y rutinas diarias de las personas varían según el día de la semana, lo que influye en la frecuencia y los tipos de alimentos que se almacenan, afectando así el consumo de energía de la nevera en el presente.
- Horas del día: Dependiendo de los patrones de uso y la actividad en la cocina, ciertos momentos del día pueden implicar un mayor uso de la nevera en el presente. Por ejemplo, durante las horas de preparación de alimentos o las comidas principales.
- La función de salida de la red neuronal artificial de los artefactos está influenciada por las

variables de entrada mencionadas anteriormente. En particular, la potencia real es el resultado obtenido a partir de esta RNA.

Datos de Aire acondicionado marca LG.

En el caso del aire acondicionado, las entradas adicionales para la red neuronal artificial incluyen variables que influyen en su funcionamiento y consumo de energía. Estas variables son:

- Estación (verano o invierno): Se utiliza una variable lógica, con un valor de 0 o 1, para indicar si es invierno o verano. Esto ayuda a adaptar el funcionamiento del aire acondicionado según la estación del año.

La salida de la red neuronal es la potencia real medida en cada momento y está directamente relacionada con las variables de entrada. La red neuronal utiliza las variables de entrada, como la fecha, los días festivos, el día de la semana, la hora del día, la estación. Esta potencia real refleja la demanda de energía actual del sistema de aire acondicionado, considerando los factores internos que influyen en el consumo energético.

Datos del circuito de Iluminación.

Las entradas adicionales de la RNA para estimar la potencia del circuito de iluminación incluyen los siguientes datos:

- Día y noche (entrada lógica): Un valor binario que indica si es de día o de noche. Esto se puede determinar utilizando información sobre el amanecer y el anochecer del lugar.

La potencia real del circuito de iluminación es la salida deseada de la RNA, ya que representa la información relevante para ajustar el consumo de energía en las iluminarias de acuerdo con las condiciones y demandas específicas del entorno. A partir de este punto, se hará referencia al artefacto de refrigeración como N1, al aire acondicionado como A.A. y al sistema de iluminación como I.1.

3.2.5 Procesamiento de señales de potencia eléctrica.

En la configuración de una red neuronal artificial, se lleva a cabo un proceso de preprocesamiento utilizando Matlab para los datos de entrada y salida. Durante el procesamiento, se realiza una serie de pasos para preparar las fechas y horas. En primer lugar, se identifica si una fecha corresponde a un día festivo o no, asignando un valor de 1 si es festivo y 0 en caso contrario. Además, se asigna un valor numérico a cada día de la semana para facilitar su procesamiento en la RNA. Por ejemplo, se utiliza el valor 3 para representar el miércoles y el valor 1 para representar el lunes.

Adicionalmente, se establece un valor binario para indicar si es de día (0) o de noche (1). Este aspecto resulta relevante para tener en cuenta las variaciones en el consumo de energía en función del periodo del día. Para lograr esto, se utilizan las funciones correspondientes que se detallan en el Anexo 5.

Se aplican diversas técnicas para mejorar la calidad de los datos de potencia de los artefactos seleccionados. Entre estas técnicas se encuentran el filtro de Savitzky-Golay, el filtro de mediana y el filtro de media móvil. Estos filtros tienen como objetivo eliminar el ruido y suavizar los datos, lo que contribuye a obtener una representación más precisa de la potencia eléctrica.

En Matlab, se utiliza filtros mediante las funciones proporcionadas por la propia aplicación. Por ejemplo, la función "sgolayfilt" aplica el filtro de Savitzky-Golay a la señal de entrada, el número 2 representa el orden del polinomio utilizado para ajustar el filtro. Un valor más alto puede adaptarse a señales más complejas, pero también puede introducir más ruido, el número 27 es el tamaño de la ventana del filtro, que debe ser un número impar. Esta ventana determina la cantidad de muestras que se utilizarán para calcular el valor filtrado en cada punto de la señal de entrada.

En la señal de salida que se ha filtrado se utiliza el filtro de mediana, que aplicamos utilizando la función "medfilt1" aplica el filtro de mediana a la señal, el número 100 representa el tamaño de la ventana del filtro de mediana. De nuevo, este valor debe ser un número impar, el

filtro de mediana toma las muestras dentro de esta ventana, ordena los valores y toma la mediana como el valor filtrado para cada punto de la señal de entrada.

Y con este resultado se utiliza el filtro de media móvil cuadrático la función toma cada valor de la señal de potencia y lo eleva al cuadrado, almacenando los resultados en la señal. Esta operación es conocida como el filtro de media móvil cuadrático, después de obtener la señal, se calcula la raíz cuadrada de cada valor para obtener la magnitud de la señal cuadrática, con ventana de tamaño 50 a la señal cuadrática, lo que significa que se promedian los valores de 50 muestras consecutivas para suavizar la señal y obtener la señal filtrada deseada, que representa la raíz cuadrada del valor cuadrático medio.

En el Anexo 7 se encuentra las gráficas que muestran los resultados obtenidos después de aplicar estos filtros. Estas gráficas proporcionan una visualización cómo los datos de potencia se ven afectados por la aplicación de cada filtro en cada artefacto.

3.2.6 Configuración Neuronal net Fitting.

Configuración de la RNA para N1

En la configuración de la RNA, se importan los datos previamente procesados de N1 como datos de entrada. Estos datos abarcan información como la fecha, la hora, días festivos días laborales, junto con otros atributos relevantes que han sido preparados durante el procesamiento. Los datos de salida corresponden a la potencia real medida para N1, los cuales también han sido previamente procesados y estructurados de manera adecuada para su utilización en el entrenamiento y la evaluación de la red neuronal. Los detalles y la presentación completa de estos datos de entrada y salida se encuentran en el Anexo 8.

Arquitectura de la RNA

Las capas ocultas desempeñan un papel fundamental al procesar y analizar los datos de entrada, identificando características relevantes y realizando cálculos complejos para generar resultados.

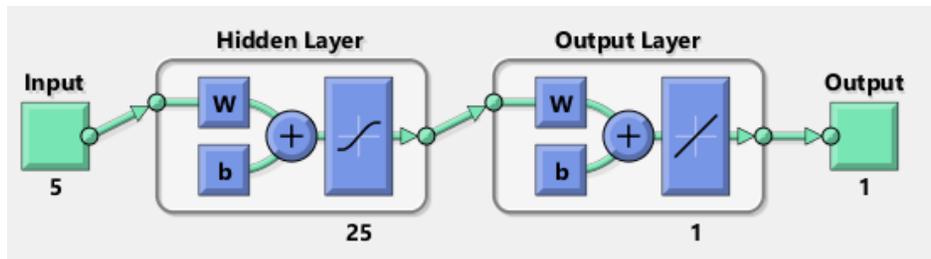


Figura 5. Arquitectura de la RNA de N1

En la figura 5 se visualiza la arquitectura de la RNA de N1, esta arquitectura, está conformada por 5 entradas, 25 capas ocultas y una capa de salida.

Entrenamiento de la red neuronal artificial

En el entrenamiento de la red neuronal, se emplea el algoritmo Levenberg-Marquardt para ajustar los pesos y sesgos de la red. Este algoritmo es eficaz en la reducción del error y su capacidad de converger rápidamente hacia los resultados deseados. Durante el entrenamiento, se utiliza una función objetivo cuyo propósito es minimizar la discrepancia entre las salidas pronosticadas por la red y los valores reales de potencia de la N1. Mediante iteraciones sucesivas, la red neuronal se perfecciona para mejorar su habilidad en realizar predicciones precisas. Los resultados del entrenamiento se observan en las siguientes gráficas:

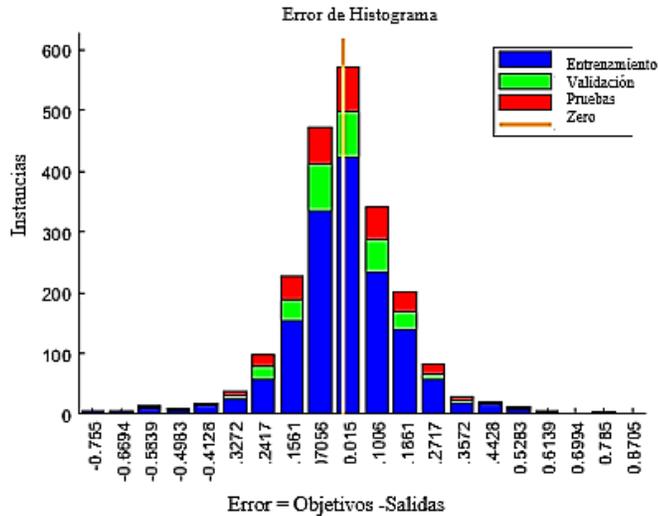


Figura 6. Histograma de los artefactos.

En la figura 6 el eje x representa el valor del error, calculado como la diferencia entre el objetivo y la salida de la red neuronal. Mientras que en el eje y se encuentran las instancias o muestras del conjunto de datos. En el histograma, se utiliza el color rojo para representar los datos de prueba, el color azul para los datos de entrenamiento y el color verde para las validaciones, las mismas ponderaciones de este histograma se tomaron en cuenta para I.1 y A.A.

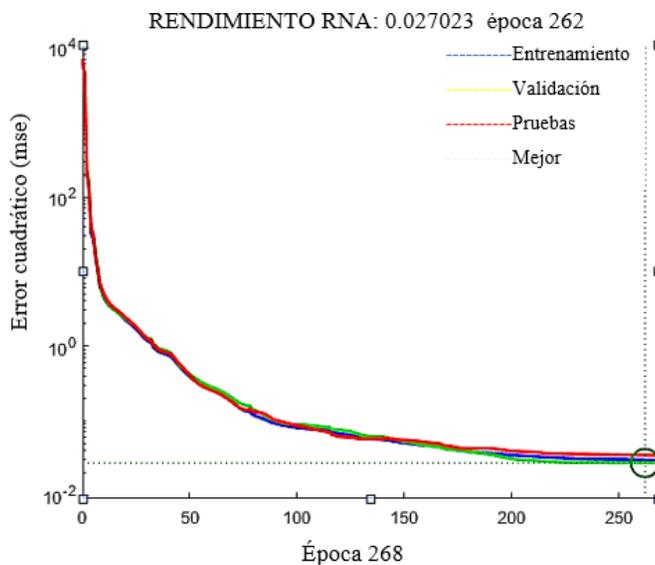


Figura 7. Mejor rendimiento de la RNA de N1.

Como resultado del entrenamiento, en la figura 7 se obtiene el rendimiento de la RNA de 0.027023 época de 262. Este valor indica la eficacia de la red neuronal en cada iteración durante el entrenamiento.

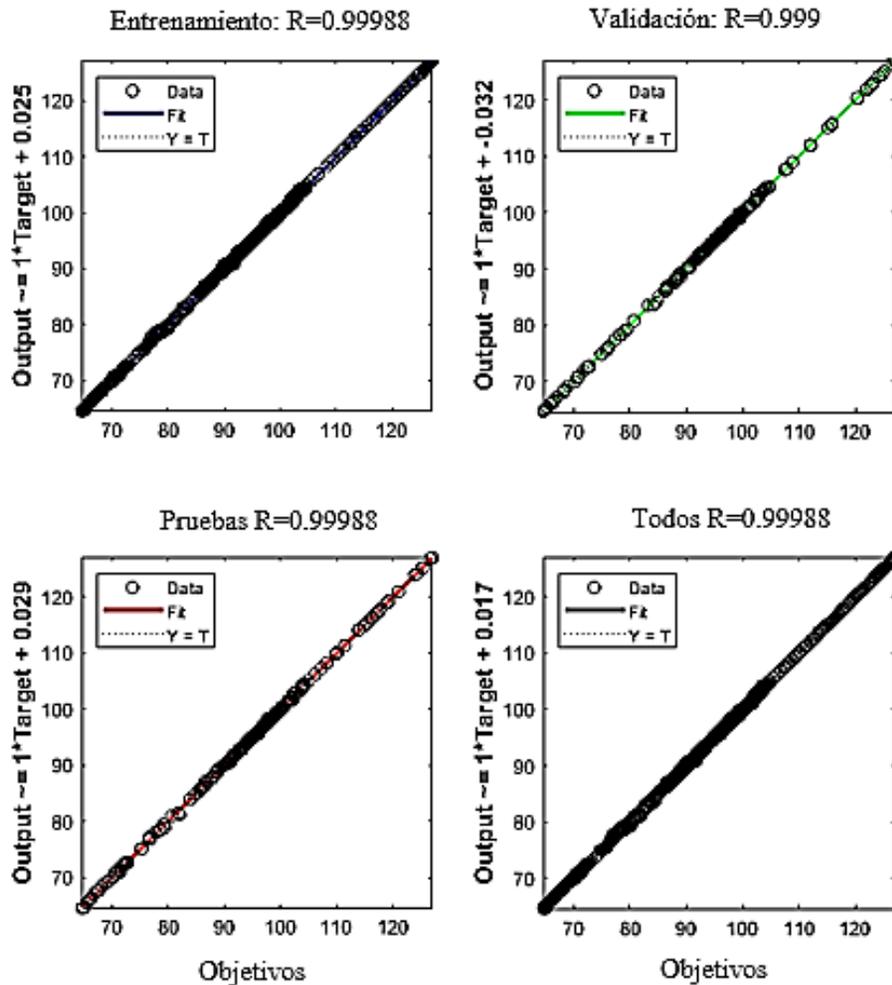


Figura 8. Resultados de regresión de la RNA de N1.

En la figura 8 se observa la regresión presentada por los valores R correspondientes a las fases de entrenamiento, validación y pruebas. El coeficiente de determinación R refleja lo bien que se ajusta la regresión a una relación lineal entre los datos. Esto indica que las predicciones realizadas por el modelo se están ajustando cada vez mejor a los valores reales de los datos, en la

siguiente tabla se muestra los resultados de la rna.

Tabla 4. Resultados del entrenamiento de la RNA de N1.

	MUESTRAS	MSE	R
ENTRENAMIENTO	1510	2.97071×10^{-2}	9.99988×10^{-1}
VALIDACIONES	323	2.70227×10^{-2}	9.9999×10^{-1}
PRUEBAS	323	3.41712×10^{-2}	9.9988×10^{-1}

La tabla 4 muestra los valores de MSE y R tanto para el entrenamiento, validación y pruebas, con un valor muy bajo de MSE que indica que el error entre la salida de la RNA y la salida de la potencia real es mínimo. Asimismo, el valor de regresión R se acerca a 1, lo que indica que la correlación entre los datos de salidas sí existe.

Configuración de la RNA para A.A

Para la configuración de entrenamiento de la RNA de A.A, en primer lugar, se importan los datos de entrada a la red neuronal, como la fecha, los días festivos, el día de la semana, las horas de uso, la estación del año. Los detalles específicos de estos datos se encuentran en el Anexo 9.

Arquitectura de la red neuronal artificial

La arquitectura de la RNA del A.A se configura, teniendo en cuenta nuevos parámetros de entrada para mejorar su desempeño.

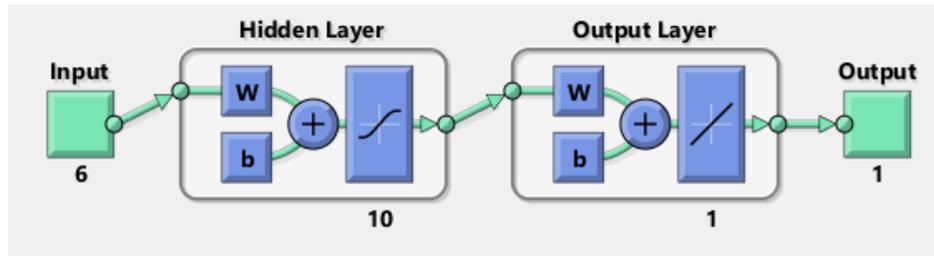
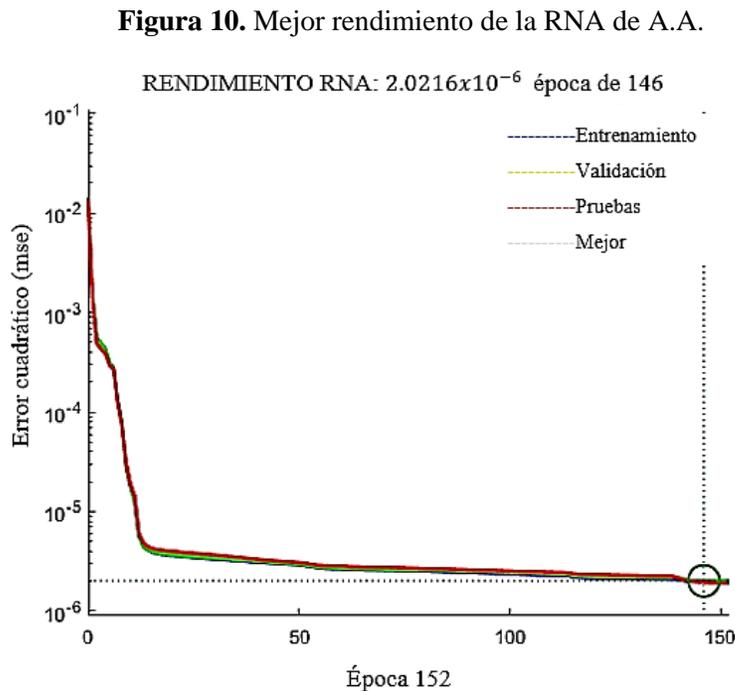


Figura 9. Arquitectura de la RNA de A.A.

La figura 9 muestra la arquitectura de RNA para A.A y se compone de 6 entradas, 10 capas ocultas y una salida que representa la potencia eléctrica.

Entrenamiento de la red neuronal artificial.

El primer entrenamiento de la red neuronal para A.A se visualiza en las siguientes gráficas:



En la figura 10, se observa el rendimiento óptimo de la red neuronal en la época 152, donde se alcanzó de 2.0216×10^{-6} . Este valor es indicativo de la precisión de las predicciones realizadas por la red neuronal en este punto del entrenamiento.

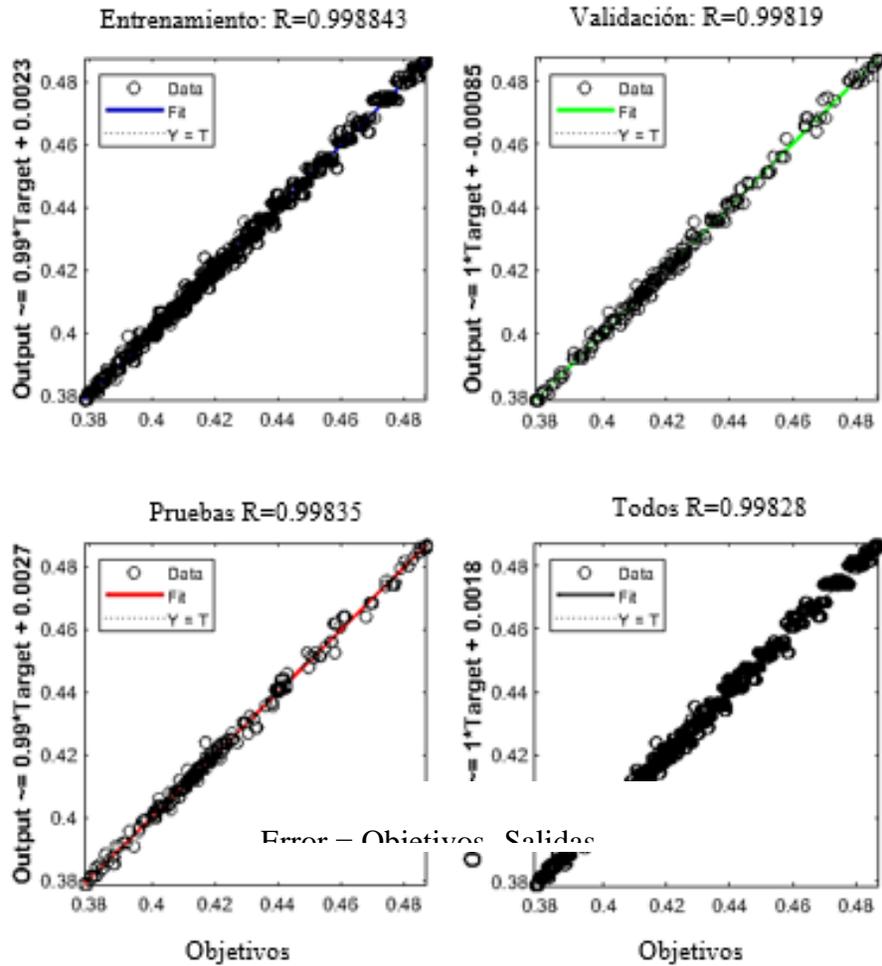


Figura 11. Resultados de regresión de la RNA de A.A.

La figura 11 presenta los resultados del entrenamiento, donde se incluyen los valores de regresión R para el conjunto de entrenamiento, así como las pruebas, entrenamiento y validación, la tabla de resultados del entrenamiento de la rna se muestra a continuación.

Tabla 5. Resultados del entrenamiento A.A

	MUESTRAS	MSE	R
ENTRENAMIENTO	1510	1.952751×10^{-6}	9.98425×10^{-1}
VALIDACIONES	323	1.221261×10^{-6}	9.998197×10^{-1}
PRUEBAS	323	1.134531×10^{-6}	9.98835×10^{-1}

El MSE del entrenamiento, prueba y validaciones de la RNA es de un valor muy cercano a 0 es decir el error de la potencia eléctrica y la real es mínimo y el R de regresión se acerca a 1 por lo sí existe una correlación de datos.

Configuración de la RNA para I.1

Se realiza la importación de los datos de entrada de la red neuronal, que incluyen variables como la fecha, los días festivos, la potencia nominal y la distinción entre día y noche, entre otros. En total, se utilizan 6 variables de entrada, y la salida corresponde a la potencia que ha sido previamente filtrada Anexo 10.

Arquitectura de la RNA

En la figura 12 se visualiza la arquitectura de la RNA para I.1, consta de 6 entradas y 25 capas ocultas.

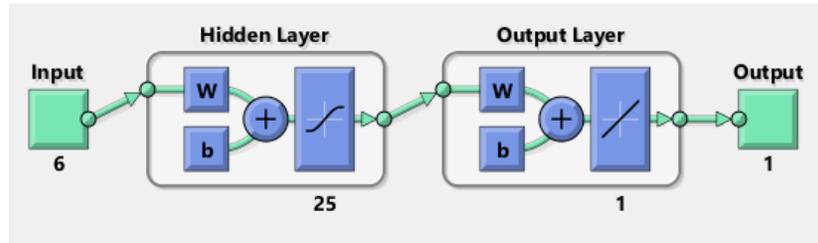


Figura 12. Arquitectura RNA de I.1.

Entrenamiento de la RNA

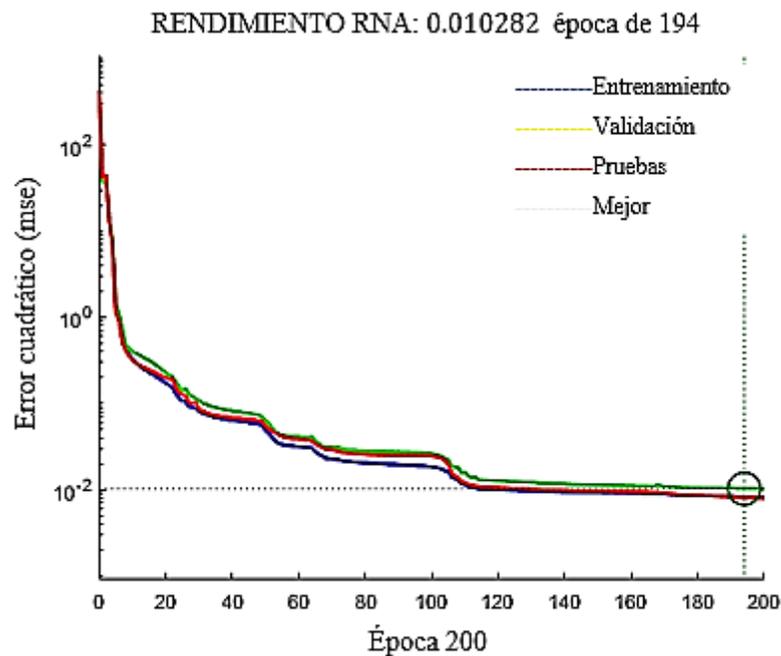


Figura 13. Mejor rendimiento de la RNA de I.1

En la figura 13, se visualiza el rendimiento óptimo de la red neuronal entrenada, donde se alcanzó un valor de 0.010282 de rendimiento época de 194, se aprecia una paridad entre la tendencia de los datos del entrenamiento, los datos de validación y pruebas.

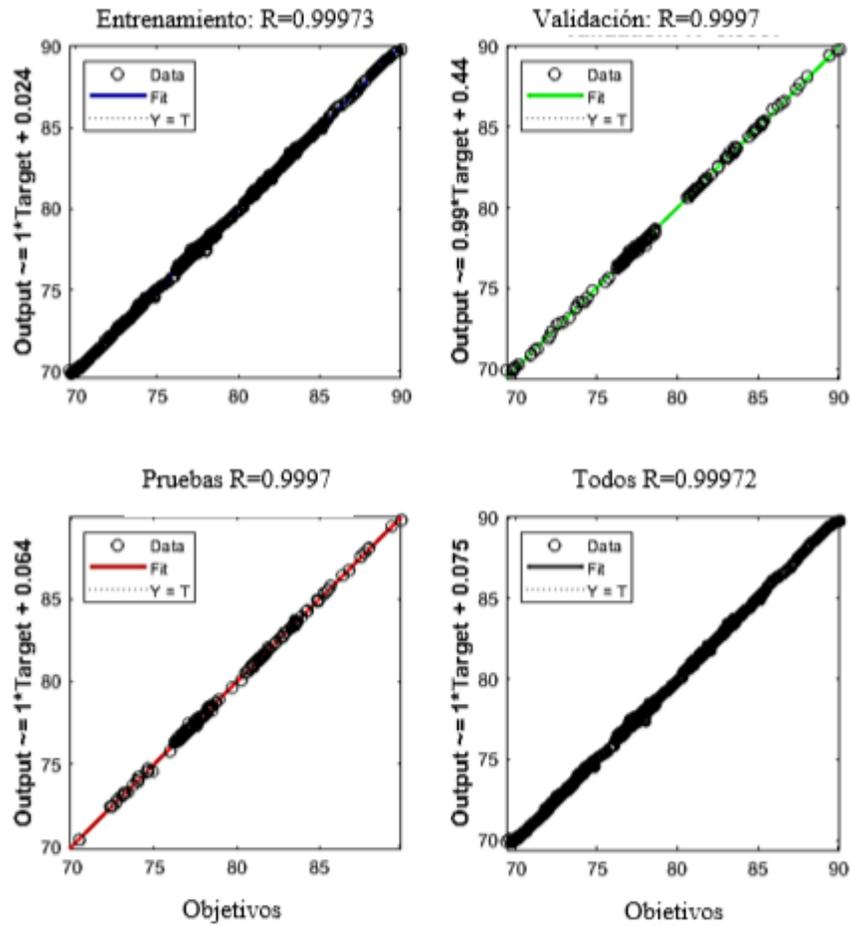


Figura 14. Resultados de Regresión de la RNA de I.1

La figura 14 muestra el progreso del entrenamiento a través de los valores de regresión lineal (R). Al analizar la figura, se observa que algunos puntos en el conjunto de pruebas se encuentran ligeramente alejados de los valores del conjunto de entrenamiento, por lo que se procede a volver a entrenar a la red neuronal modificando sus capas ocultas a 25 ver Anexo 10.

Tabla 6. Resultados del entrenamiento

	MUESTRAS	MSE	R
ENTRENAMIENTO	1724	8.16993×10^{-3}	9.99728×10^{-1}
VALIDACIONES	216	1.02820×10^{-2}	9.99700×10^{-1}
PRUEBAS	216	7.96802×10^{-3}	9.99700×10^{-1}

En la tabla 6 se presentan los resultados correspondientes al entrenamiento, las pruebas y las validaciones. Los valores muestran que los valores de R se acercan a 1, lo cual indica un buen ajuste del modelo a los datos y un mse cercano a 0 que me indican un error mínimo en las salidas de la rna.

3.4. Diseño de la Aplicación Watt-ADvision.

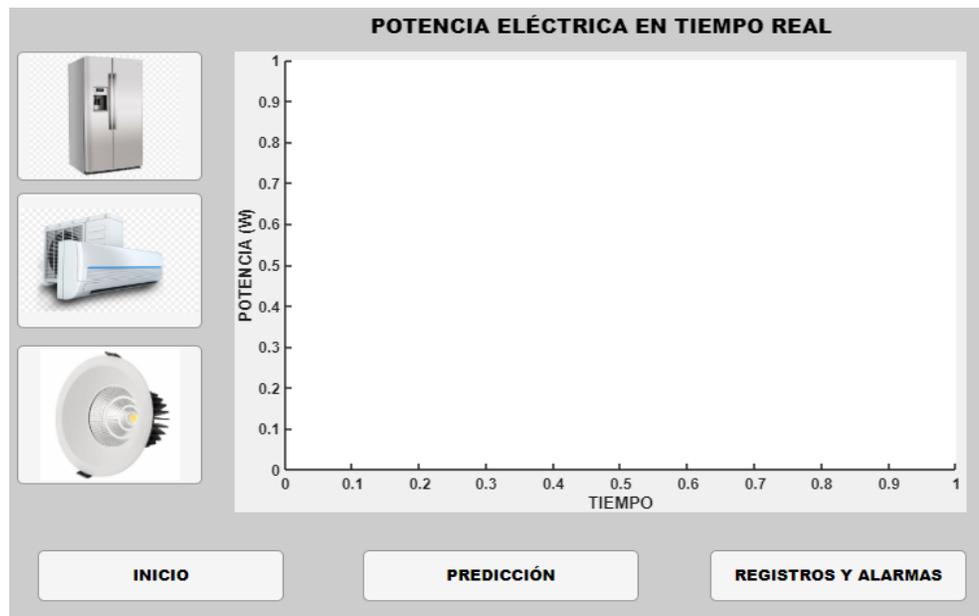
Matlab cuenta con un apartado de “Designer App” donde se realiza la aplicación para el monitoreo y predicción del consumo eléctrico.



Figura 15. Interfaz principal.

En la figura 15 se presenta la interfaz principal de la aplicación, la cual cuenta con tres botones. El primero permite acceder a la aplicación de visualización en tiempo real de la potencia de cada artefacto. A continuación, se encuentra la aplicación de predicción de potencia eléctrica, y finalmente, el último botón brinda acceso a los registros y alarmas relacionados con el exceso **de potencia de cada artefacto**.

Figura 16. Interfaz potencia eléctrica tiempo real.



La Figura 16 presenta la interfaz gráfica que muestra la potencia eléctrica de cada artefacto. Al presionar el botón correspondiente a N1, se muestra la tendencia del consumo eléctrico en tiempo real de dicho artefacto. Lo mismo se aplica al botón de A.A e I1, donde se puede observar la tendencia del consumo eléctrico en tiempo real de cada uno de ellos.

PREDICCIÓN DE CONSUMO Y COSTO MENSUAL DE ARTEFACTOS ELÉCTRICOS

FECHA

<p>POTENCIA <input type="text" value="0"/></p> <p>CAPACIDAD <input type="text" value="0"/></p>		<p>CONSUMO <input type="text"/></p> <p>COSTO \$ <input type="text"/></p>
<p>POTENCIA <input type="text" value="0"/></p>		<p>COSTO \$ <input type="text"/></p> <p>CONSUMO <input type="text"/></p>
<p>POTENCIA <input type="text" value="0"/></p>		<p>CONSUMO <input type="text"/></p> <p>COSTO \$ <input type="text"/></p>
<p>CONSUMO TOTAL DE LA VIVIENDA</p> <p>CONSUMO TOTAL <input type="text"/> CALCULAR</p> <p>COSTO TOTAL \$ <input type="text"/> CLEAR</p>		

Figura 17. Interfaz predicción de consumo y costo de artefactos eléctricos.

En la figura 17 se muestra el diseño de la interfaz gráfica destinada a la predicción del consumo y costo de los artefactos eléctricos. En la sección de "FECHA", se introduce la fecha para la cual deseamos predecir la potencia de los artefactos. También se ingresan la potencia nominal, la capacidad y el área correspondientes a cada artefacto en sus respectivas secciones. Luego, se presionan los botones presentes en la figura para mostrar el consumo mensual de cada artefacto seleccionado. Al presionar el botón "Calcular", se mostrará el consumo predictivo del mes para todos los artefactos, así como el costo asociado a dicho consumo



Figura 18. Interfaz de registros y alarmas de consumo eléctrico.

En la figura 18 se muestra la interfaz de registros y alarmas de consumo eléctrico. Al presionar el botón "Mostrar registros", se visualizan todos los registros de potencia eléctrica, incluyendo la fecha, la hora y la potencia eléctrica correspondiente a cada registro. Por otro lado, al presionar el botón "Mostrar alarmas", se muestran únicamente las potencias que exceden los rangos establecidos para la potencia eléctrica

3.4.1 Consumo eléctrico de zona residencial.

Se identifica la zona residencial a realizar el proyecto como categoría residencial y de voltaje de nivel bajo Tabla 2, En el mes de Enero la facturación véase ec.(14) para el cálculo de la facturación) es de \$60.18 y los datos del pliego tarifario es de 386 kWh véase el pliego tarifario Anexo 12.

Cálculo del valor del consumo energético

En el ámbito residencial, el consumo energético se mide mediante un medidor monofásico de consumo eléctrico, comúnmente ubicado en un costado de la casa, permitiendo un acceso sencillo para los encargados de lectura. Este medidor desglosa y registra el consumo total de energía, sumando las potencias utilizadas por todos los equipos eléctricos durante cada hora. Durante el período de abril hasta mayo, el consumo de energía eléctrica registrado fue de 536 kWh. Para determinar el costo de facturación de energía, se establece un valor de \$70.46 en el Anexo 13 para el rubro de energía. Utilizando la ecuación (1), se procede a calcular el costo total de energía consumida.

Tabla 7: Consumo eléctrico y costos Enero-mayo

Fecha de consumo	Consumo eléctrico KwH	\$Costo consumo eléctrico	\$Costo de Facturación.
ENERO	386	\$37.82	\$60.18
FEBRERO	400	\$39.29	\$56.42
MARZO	417	\$41.076	\$67.83
ABRIL	536	\$53.48	\$70.46
MAYO	443	\$43.806	\$60,30

En la tabla 7 se presentan los consumos eléctricos mensuales desde enero hasta mayo, junto con los costos asociados para cada período. Se puede observar cómo el consumo varía de un mes a otro, influyendo directamente en el costo de facturación.

CAPITULO IV

PRUEBAS Y RESULTADOS

En este capítulo, se presentan en detalle los hallazgos derivados de la recopilación y el análisis de los datos. Se examinan los resultados en relación con los objetivos específicos de la investigación y se proporcionan explicaciones sobre la adquisición y almacenamiento de los datos en la nube, las pruebas desde el funcionamiento del monitoreo de la aplicación y el análisis de resultados de la red neuronal artificial para obtener la predicción del consumo eléctrico en la vivienda.

4.1 Pruebas de adquisición y almacenamiento de datos en la nube.

En primer lugar, se llevaron a cabo pruebas utilizando el ESP32 y un medidor de potencia para verificar la precisión de los datos obtenidos. Mediante el primer código implementado, se logró obtener datos en tiempo real de un dispositivo y visualizarlos en la página web de Arduino IoT Cloud. Los detalles de esta implementación se encuentran en el Anexo 14.

A continuación, se presenta el procedimiento que se realiza para la obtención de datos de potencia:

1. Establecer un tablero eléctrico específico para un dispositivo con el fin de enviar los datos de consumo eléctrico a través de la plataforma IFTTT.
2. Observar la variabilidad de los datos
3. Ya en la nube utilizar la variable proporcionada por la plataforma de Arduino IoT Cloud, para verificar los valores reales con los medidos, se muestra un valor de 1.27 A para la corriente de I.1.
4. Comparar este valor con el resultado medido con un amperímetro físico y obtener un valor de 1.3 A, que es muy cercano al dato de la plataforma de Arduino.

Datos tomados fecha 2 de Julio 11:09 AM

Tabla 8. Datos de artefactos en arduino IoT , google drive y real.

Artefacto	Corriente Real (A)	Corriente Medida (A)	Corriente Google Drive (A)	Error Corriente (%)	Voltaje Real (V)	Voltaje Medido (V)	Error Voltaje (%)
N1	0.75	0.7588	0.7588	0.88%	118.800	119.6	0.59%
A.A	0.19	0.193000	0.193000	0.0001%	122	122.6	0.49%
I.1	1.4	1.376000	1.376000	1.29%	122	122.4	0.25%

La tabla 8 compara las mediciones de corriente y voltaje a una hora determinada, destacando tanto los valores medidos en el Google Drive como los errores de medición en términos porcentuales. Los resultados muestran diferencias entre las mediciones reales y las registradas en Google Drive, y resaltan un error mínimo en las comparaciones.

Datos tomados fecha 8 de Julio 10:40 AM

Tabla 9. Datos de artefactos en arduino IoT , google Drive y Real.

Punto	Corriente Real (A)	Corriente Medida (A)	Corriente Google Drive (A)	Error Corriente (%)	Voltaje Real (V)	Voltaje Medido (V)	Error Voltaje (%)
N1	0.76	0.75	0.75	1.32%	119.9	119.6	0.25%

A.A	0.19	0.190999	0.190999	0.00%	122	122.3	0.24%
I.1	1.47	1.488000	1.488000	1.22%	122	122.3	0.25%

La tabla 9 y 8 muestra validez de los datos al observar mediciones tomadas en diferentes fechas y horas. Esta consistencia en las comparaciones a lo largo de distintos momentos subraya la autenticidad de las mediciones capturadas, ya que se aprecia una coherencia en los resultados en cada intervalo horario analizado.

5. Se observa que no hay un cambio significativo de potencia dentro de segundos y se decide aumentar el intervalo de obtención de datos a cada 20 minutos.
6. Realizar un tablero adicional para capturar los datos de potencia de los demás artefactos.
7. Concluir que los datos obtenidos son confiables, lo que indica que la interconexión entre IFTTT y Arduino IoT Cloud y la medición del consumo eléctrico son confiables.

Es importante resaltar que el procedimiento muestra cómo se estableció la conexión entre la plataforma de IFTTT y Arduino IoT Cloud, cómo se configuró el tablero eléctrico y cómo se verificó la precisión de los datos obtenidos, asegurando la confiabilidad de la medición del consumo eléctrico de la N1. También se realizó una modificación en el intervalo de obtención de datos para facilitar el almacenamiento y análisis de estos.

4.2. Evaluación de desempeño de las redes neuronales artificiales.

En esta sección se detalla los resultados obtenidos en el entrenamiento de la red neuronal artificial, se muestran las gráficas de la potencia real obtenida por la adquisición de datos y la potencia entrenada por la red neuronal de cada artefacto con estos datos se verifica los resultados del entrenamiento de la red neuronal, considerando la potencia real de cada artefacto como la potencia predicha por la red neuronal para cada uno de ellos.

Resultados de potencia real y RNA de N1.

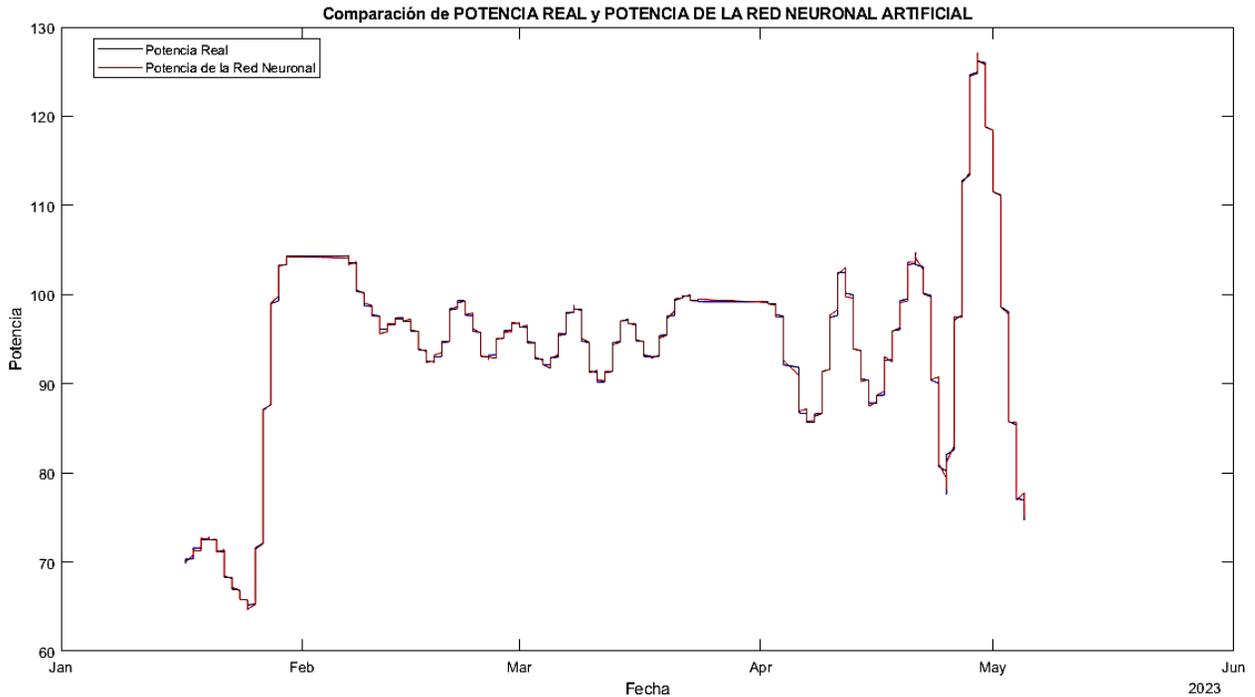


Figura 19. Potencia real y potencia de la RNA de N1.

En la figura 19, se puede apreciar que la red neuronal entrenada muestra una similitud notable con la tendencia de la potencia real. Esto indica que la red neuronal ha aprendido correctamente los patrones y características del conjunto de datos y es capaz de realizar predicciones que se asemejan de manera cercana a los valores reales.

Esta similitud entre la red neuronal entrenada y la tendencia de la potencia real demuestra la capacidad de la red neuronal para capturar y modelar el comportamiento del sistema. Al lograr una concordancia cercana entre las predicciones y los datos reales.

Resultados de potencia real vs resultados de la red neuronal de A.A.

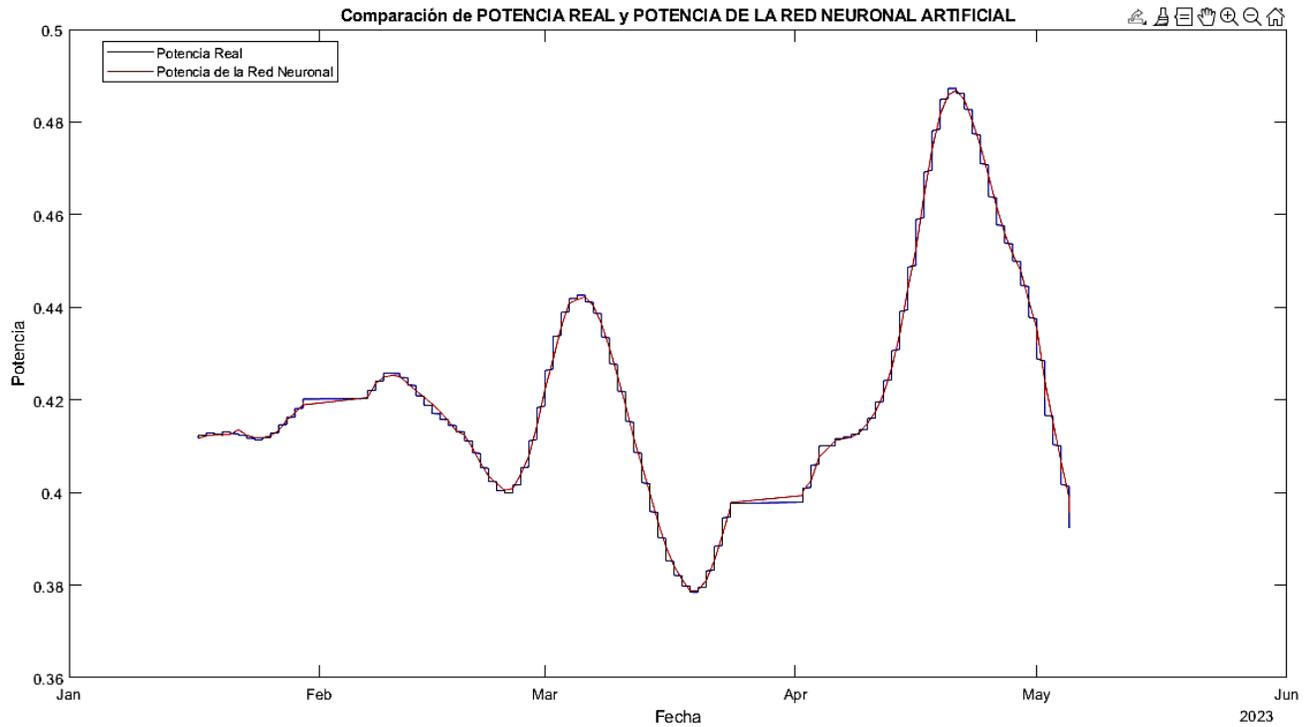


Figura 20. Potencia real y la potencia de la RNA de A.A.

En la figura 20, al comparar la potencia real con la potencia predicha por la red neuronal, podemos inferir que la red ha aprendido de manera efectiva los patrones subyacentes en los datos y es capaz de generar resultados que se asemejan a los valores reales. Esta concordancia entre las predicciones y los datos reales es de gran utilidad, ya que nos permite tomar decisiones informadas y anticipar el comportamiento futuro de la potencia

Resultados de potencia real y resultados de la RNA de I.1

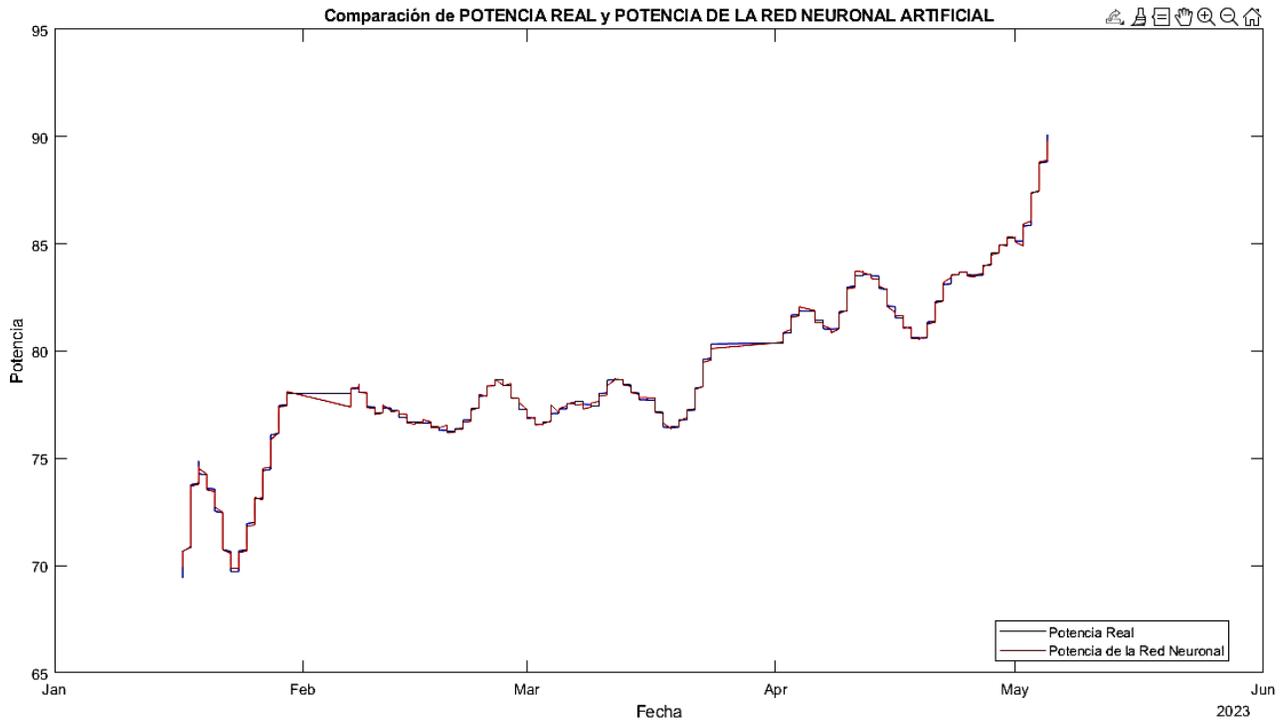


Figura 21. Potencia real y potencia RNA de I.1.

En la figura 21 que compara la potencia real con la potencia predicha por la red neuronal entrenada para el sistema eléctrico de I.1, se puede apreciar una ligera discrepancia entre los resultados de la red neuronal y la potencia real. Se observa que, en la gráfica, las curvas de potencia real y potencia predicha tienen una tendencia similar, lo cual indica que la red neuronal está capturando ciertos patrones de consumo de I.1. Sin embargo, también se puede notar que hay pequeñas variaciones o desviaciones entre las dos curvas.

Como muestra de desempeño de la RNA para cada artefacto, se detalla en la siguiente tabla el rendimiento de cada uno de los modelos, incluyendo el número de neuronas, el algoritmo utilizado, la función de activación empleada, el valor del RMSE, el valor de R y la precisión correspondiente.

Tabla 10. Desempeño de RNA

Modelo RNA	Capas Ocultas	Algoritmo	Función de activación	Tasa de aprendizaje	RMSE	R^2
N1	25	Levenberg.Marquardt Backpropagation	tansig	0.01	0.1731	0.9998
A.A	10	Levenberg.Marquardt Backpropagation	tansig	0.01	0.0014	0.9996
I1	15	Levenberg.Marquardt Backpropagation	tansig	0.01	0.0914	0.9994

En la tabla 8 se muestra el RMSE de cada red neuronal entrenada para cada artefacto, con un valor de 0.1731 en la red neuronal correspondiente a la N1. Este valor indica el error entre la potencia real de la N1 y la potencia predicha por la red neuronal artificial. Es notable que existe una similitud considerable entre las predicciones y los resultados reales, lo que indica que el error es mínimo. Esta alta precisión de la red neuronal en la predicción de la potencia demuestra su capacidad para realizar estimaciones confiables.

4.3 Ejecución de la App WattADvision

En la aplicación WattADvision se pueden apreciar 3 interfaces. El primer interfaz muestra el monitoreo del consumo eléctrico en tiempo real. Para capturar estos datos, se ha implementado una conexión en thingspeak, lo que permite obtener los datos en tiempo real de los diferentes artefactos eléctricos.

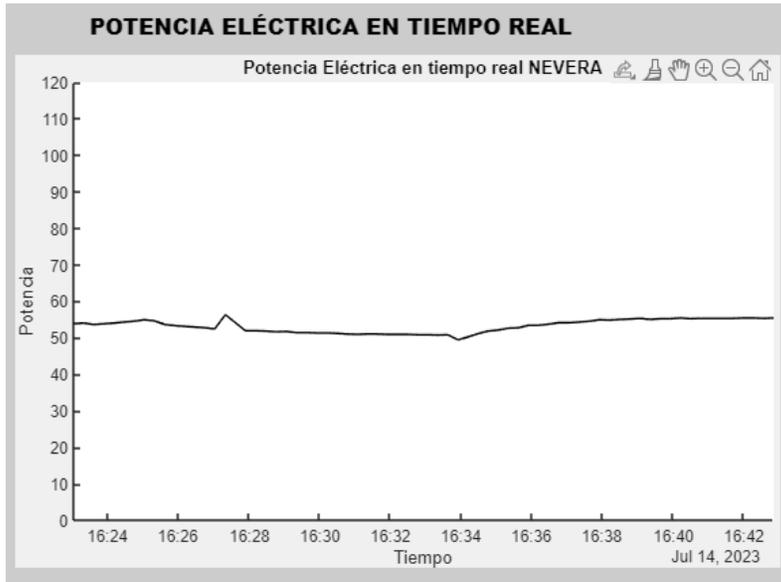


Figura 22. Gráfica de potencia en tiempo real de N1.

En la figura 22 se presenta la potencia eléctrica en tiempo real, la cual se ha obtenido a partir de los datos recopilados en thingspeak. Al analizar estos datos, es posible observar el comportamiento de la potencia consumida por la N1 a lo largo del tiempo.

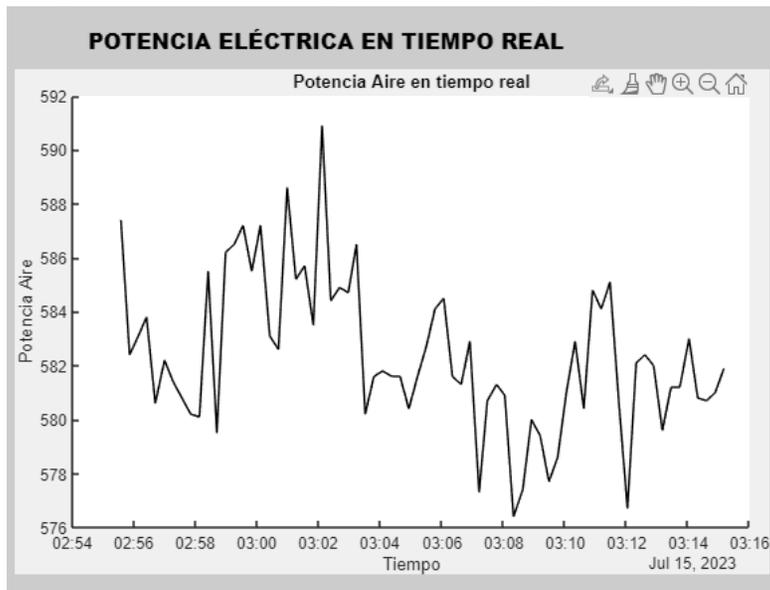


Figura 23. Gráfica de potencia en tiempo real de A.A.

En la figura 23 presentada, se observa el comportamiento de la potencia consumida por A.A a lo largo del tiempo. Este gráfico nos proporciona una representación visual de cómo varía el consumo de energía A.A en diferentes momentos.

Al analizar la figura, se identifican diferentes patrones y tendencias en el consumo de potencia. Por ejemplo, podríamos notar picos de consumo durante ciertos períodos del día, lo que indica que el A.A está trabajando más intensamente en esos momentos. También se identifican los períodos de menor consumo, posiblemente durante las horas de menor actividad o en momentos en los que la temperatura exterior es más favorable.

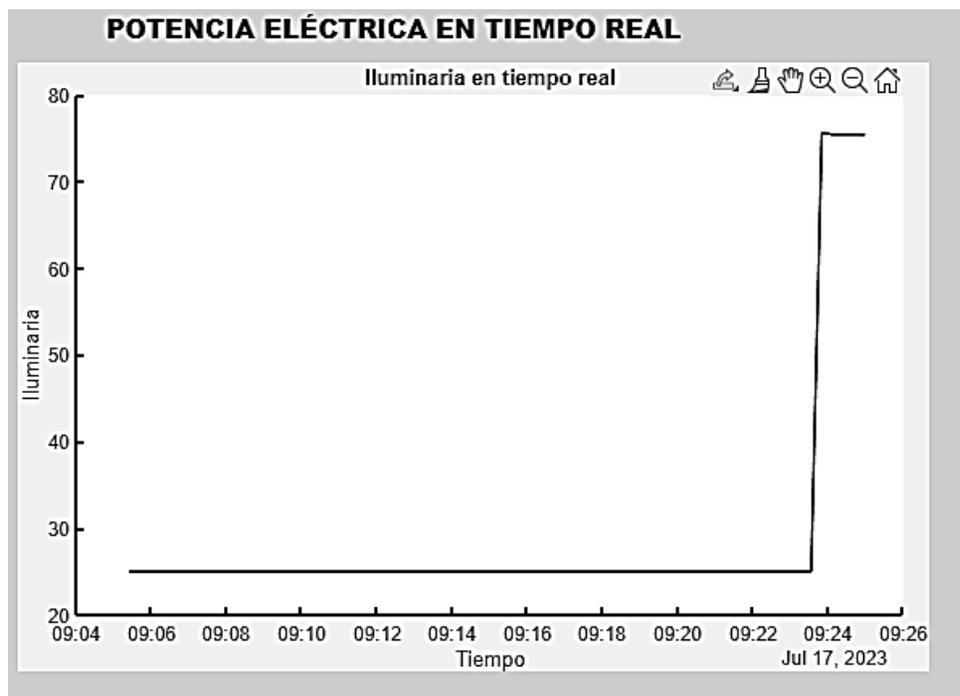


Figura 24. Gráfica de potencia en tiempo real I.1.

En la figura 24 se aprecia el comportamiento de la gráfica en tiempo real, donde se observa una potencia que en algunos periodos se mantiene estable y en otros aumenta abruptamente debido al sistema eléctrico instalado. Esta visualización en tiempo real permite tener una visión actualizada del consumo de potencia del sistema y detectar cambios o fluctuaciones significativas en la demanda eléctrica. Es posible identificar momentos en los que la potencia se mantiene

constante, lo que podría indicar un funcionamiento estable del sistema. Esta sección de la interfaz cumple con las Normas ISO 50001 y ISO 50006 al recopilar, analizar y monitorear en tiempo real las gráficas y tendencias del consumo eléctrico de cada uno de los artefactos seleccionados.



Figura 25. Interfaz de predicción del consumo eléctrico

En la siguiente interfaz de Predicción de consumo eléctrico figura 25, se ingresan los datos de los electrodomésticos solicitados por la aplicación, como la potencia nominal, la capacidad en litros y el área correspondiente a cada artefacto eléctrico. A través de una red neuronal artificial entrenada específicamente para cada electrodoméstico, se realiza una predicción del consumo eléctrico estimado para todo el mes.

Utilizando los valores ingresados como se visualiza en la figura 25, la red neuronal calcula la cantidad de energía que se espera que cada electrodoméstico consuma durante el mes. Además, con los datos de consumo estimado, también se puede predecir el costo correspondiente al mes ingresado. Esta información resulta útil para tener una idea aproximada de cuánto se gastará en términos monetarios en concepto de consumo eléctrico. Esta sección de la interfaz cumple con la

norma ISO 50006 al presentar una oportunidad de ahorro energético a través de los resultados de la predicción del consumo eléctrico

Figura 26. Interfaz de registros y alarmas del consumo eléctrico

FECHA	HORA	DESCRIPCION	POTENCIA ELECTRICA
23/07/2023	02:57	Nevera	284.8000
23/07/2023	03:00	Nevera	285.0000
23/07/2023	04:00	Nevera	118.8000
23/07/2023	05:13	Nevera	53.7000
23/07/2023	06:12	Nevera	52.1000
23/07/2023	07:10	Nevera	51.2000
23/07/2023	08:37	Nevera	88.1000
23/07/2023	09:15	Nevera	82.2000
23/07/2023	10:35	Nevera	84.9000
23/07/2023	11:01	Nevera	55.6000

En la interfaz figura 26, podemos visualizar el valor máximo de potencia eléctrica consumida por N1 dentro de cada hora. Esta información nos permite identificar los picos máximos de consumo de energía de N1 durante el transcurso de una hora determinada. Al observar el valor máximo de potencia eléctrica consumida por la N1 en cada hora, podemos identificar los momentos en los que el consumo es más alto. Esto puede ser útil para comprender los patrones de uso de N1 y cómo afectan el consumo energético.

REGISTROS		ALARMAS		ELIMINAR ALARMA	
FECHA	HORA	DESCRIPCION	POTENCIA ELECTRICA		
23/07/2023	02:57	Nevera	284.8000		
23/07/2023	03:00	Nevera	285.0000		
23/07/2023	22:28	Nevera	272.2000		
INICIO		MONITOREO		PREDICCIÓN	

Figura 27. Interfaz de Registros y Alarmas del consumo eléctrico

En la figura 27 se muestra la fecha y hora en la que el dispositivo registró un consumo eléctrico notablemente alto. Podemos observar una tendencia consistente de consumo en esa hora específica. Analizar esta tendencia de consumo nos brinda la oportunidad de tomar medidas para gestionar mejor el uso de la energía. En el caso de N1 se ha considerado un umbral de hasta 250 watts, ya que en los datos históricos no existen los registros que alcanzan esa potencia. Por lo tanto, se deduce que hay una acción específica que provoca que la potencia de N1 supere este umbral seleccionado en ocasiones puntuales.

Análisis y comparación de predicción del consumo eléctrico y consumo eléctrico de cada artefacto.

Tabla 11. Consumo real y consumo-costo predicho de N1

MES	N1-CONSUMO REAL(kWh)	N1-CONSUMO PREDICHO (kWh)	N1-COSTO PREDICHO	ERROR
ENERO	48.62	46.7	\$4.29	3.95%
FEBRERO	61.30	62.35	\$5.79	1.71%
MARZO	69.78	70.67	\$6.56	1.27%
ABRIL	66.00	68.74	\$6.38	4.15%
MAYO	68.03	71.20	\$6.61	4.65%

Esta tabla proporciona una visión esclarecedora del comportamiento del consumo eléctrico del artefacto N1 en los primeros cinco meses del año. Cada entrada nos muestra el consumo real registrado en kilovatios-hora (kWh) para el mes correspondiente, junto con las predicciones generadas por nuestra red neuronal y los costos estimados asociados.

Tabla 12. Consumo real y consumo-costo predicho de A.A.

MES	N1-CONSUMO REAL(kWh)	N1-CONSUMO PREDICHO(kWh)	N1-COSTO PREDICHO	ERROR
ENERO	52.32	54.24	\$5.04	3.71%
FEBRERO	49.27	51.55	\$4.79	4.62%
MARZO	56.20	58.39	\$5.42	3.89%
ABRIL	57.38	59.21	\$5.50	6.47%
MAYO	60.36	64.26	\$5.97	4.79%

La tabla proporciona un análisis detallado de las predicciones de consumo eléctrico para el artefacto A.A durante los primeros cinco meses del año. Se incluye el consumo real, el consumo y el costo predicho para cada mes. Además, se ha calculado el error porcentual en las predicciones, mostrando la diferencia entre el consumo real y el predicho. En general, las predicciones están en línea con los valores reales, con errores que oscilan entre el 3.19% y el 6.47%..

Tabla 13. Consumo real y consumo-costo predicho de I.1.

MES	N1-CONSUMO REAL(kWh)	N1-CONSUMO PREDICHO(kWh)	N1-COSTO PREDICHO	ERROR
ENERO	281.68	289.95	27.88	2.94%
FEBRERO	275.72	278.32	26.70	0.94%
MARZO	290.4	305.34	29.44	5.14%
ABRIL	313.00	317.74	30.72	1.51%
MAYO	308,32	312.09	30.14	1.22%

La tabla 13 presenta tanto el consumo real como el pronosticado, junto con sus costos asociados, de I.1, además de proporcionar un índice de error porcentual para cada predicción. Este índice destaca la discrepancia entre los valores reales y pronosticados del consumo. De manera global, las estimaciones exhiben una coherencia satisfactoria en comparación con los valores reales, con discrepancias que varían en un intervalo que abarca desde el 0.94% hasta el 5.14%

Análisis y comparación de predicción del consumo eléctrico y el consumo eléctrico de la vivienda.

Tabla 14. Comparación del consumo eléctrico real-costo y el consumo eléctrico-costo predicho.

MES	Consumo Real (kWh)	Costo (\$)	Consumo Predicha(kWh)	Costo Predicho (\$)	ERROR Consumo
ENERO	386	\$37.82	390.36	\$38.30	1.14%
FEBRERO	400	\$39.29	392.22	\$38.47	1.95%
MARZO	417	\$41.07	434.4	\$42.90	4.07%
ABRIL	536	\$54.41	525.69	\$50.08	1.9%
MAYO	443	\$ 43.80	447.55	\$44.28	0.92%

La tabla 14 muestra los resultados de la predicción de consumo realizada por la red neuronal artificial para diferentes meses. Se comparan los valores del consumo real, el costo real en dólares, el consumo y costo predicho en cada mes. Los meses de enero, febrero, marzo y mayo tienen mucha similitud en cuanto al consumo y costos, excepto en abril, donde se registró un consumo real de 536 kWh con un costo de \$54.41. La red neuronal predijo un consumo de 525.69 kWh con un costo estimado de \$50.08 y su error de 1.9%.

Obteniendo el resultado de la predicción del consumo total mensual y el consumo de cada artefacto, se puede lograr un ahorro energético significativo. Al tener una estimación precisa del consumo eléctrico, es posible identificar patrones de consumo y tomar decisiones informadas para optimizar el uso de la energía.

CONCLUSIONES Y RECOMENDACIONES.

Conclusiones

Se ha logrado desarrollar un prototipo que emplea redes neuronales artificiales y la tecnología IoT para predecir el consumo eléctrico en una vivienda. Este prototipo proporcionó información periódica sobre la potencia eléctrica de tres artefactos seleccionados de la vivienda. Además, se aplicaron técnicas de procesamiento de señales en MATLAB, como los filtros de Savitzky-Golay, mediana y media móvil, para mejorar la calidad y el análisis de los datos recopilados. Estas técnicas permitieron suavizar la señal, eliminar ruido y fluctuaciones no deseadas, lo que resultó en una representación del comportamiento del consumo energético de los artefactos estudiados, la combinación exitosa de hardware (ESP32 y medidores de potencia) y software (MATLAB y técnicas de procesamiento de señales) ha demostrado su eficiencia en el monitoreo, brindando una visión detallada de los patrones de consumo de los artefactos.

Este análisis de la predicción del consumo eléctrico en esta vivienda se ha basado en el comportamiento diario de sus habitantes, lo que ha permitido diseñar un prototipo exclusivamente adaptado para esta residencia. El estudio arrojó resultados positivos. Este prototipo representa un avance significativo en el campo de la eficiencia energética. A través de la implementación de Watt-advisión, los usuarios adquieren la capacidad de visualizar las predicciones de consumo eléctrico para cada electrodoméstico, junto con una comprensión clara de cómo se distribuye el consumo en su hogar. Pueden identificar los dispositivos que más contribuyen al gasto energético, lo cual resulta especialmente beneficioso para las personas de clase media. Al tomar decisiones informadas sobre el uso de los aparatos y ajustar los hábitos de consumo, es posible lograr reducciones significativas en la factura eléctrica.

El desarrollo de esta interfaz, basada en un sistema SCADA, pone de relieve que esta tecnología ha trascendido su uso tradicional en entornos industriales y ha encontrado aplicaciones cada vez más significativas en el ámbito residencial. El avance en la integración de dispositivos inteligentes y sistemas de automatización ha permitido que los hogares modernos aprovechen las ventajas del SCADA para la supervisión y el control de diversos aspectos, como la gestión

energética y la optimización del consumo.

El análisis de la predicción del consumo eléctrico en esta residencia se ha desarrollado considerando el comportamiento diario de sus residentes y se ha fundamentado en el cumplimiento de las Normas ISO 50001 y 50006. Los resultados obtenidos mediante este estudio indica que la metodología adoptada ha demostrado ser eficaz en la predicción del consumo eléctrico. Esta investigación refuerza la viabilidad y utilidad del enfoque propuesto, y brindan una base para futuras investigaciones y aplicaciones en el campo de la eficiencia energética y la gestión del consumo eléctrico

Recomendaciones

Al evaluar el impacto y los beneficios reales del uso de la aplicación "WattADvisión" a través de estudios a largo plazo. Estos estudios podrían incluir la recopilación de datos de consumo durante los meses en los que la red neuronal no ha sido entrenada y compararlos con los meses posteriores a la implementación. Además, se deben considerar los costos asociados al consumo eléctrico antes y después de la aplicación.

Es importante tener en cuenta que el prototipo desarrollado en esta investigación se limita a la conexión de tres artefactos específicos. Sin embargo, es necesario reconocer que, en un domicilio real, hay muchos más artefactos que pueden influir en el consumo eléctrico, por esta razón se recomienda la implementación previa de este prototipo como parte integral del sistema eléctrico, con el objetivo de lograr una mejor implementación del sistema de monitoreo.

BIBLIOGRAFÍA

- [1] M. Baquero and F. Quesada, “Eficiencia energética en el sector residencial de la Ciudad de Cuenca, Ecuador,” *MASKANA*, vol. 7, no. 2, pp. 147–165, Dec. 2016, doi: 10.18537/mskn.07.02.11.
- [2] Agencia de Regulación y Control de Energía y de Recurso Renovables y No Renovables, “Pliego Tarifario Servicio Público de Energía Eléctrica_ Año 2022,” *Dirección de regulación económica y tarifas del sector eléctrico* , pp. 1–39, 2022.
- [3] ISO/TC 301, “ISO 50001:2018(en).” <https://www.iso.org/obp/ui/#iso:std:iso:50001:ed-2:v1:en> (accessed Jul. 13, 2022).
- [4] R. Patricia *et al.*, “Plan Nacional de Eficiencia Energética,” pp. 1–112, 2016.
- [5] R. A. Tejada, L. Carlos Méndez González, and F. Lom Monarrez, “Sistema de monitoreo de consumo eléctrico inalámbrico y analizador de la calidad de la energía.”
- [6] G. A. Figueredo-Ávila and J. A. Ballesteros-Ricaurte, “Identificación del estado de madurez de las frutas con redes neuronales artificiales, una revisión,” *Ciencia y Agricultura*, vol. 13, no. 1, pp. 117–132, Jan. 2016, doi: 10.19053/01228420.4811.
- [7] A. Santiesteban Velázquez, J. Osvaldo, N. González, D. R. Peña, and D. J. García, “PRONOSTICO DE CONSUMO DE ENERGÍA ELÉCTRICA USANDO REDES NEURONALES ARTIFICIALES,” no. 16, 2014, [Online]. Available: <http://www.eumed.net/rev/tlatemoani/index.htm>
- [8] Agencia de Regulación y Control de Energía y and Recursos Naturales No Renovables, “ESTADISTICA ANUAL Y MULTIANUAL DEL SECTOR ELECTRICO ECUATORIANO,” 2021. <https://www.controlrecursosyenergia.gob.ec/wp->

- content/uploads/downloads/2022/04/Estadistica2021.pdf (accessed Oct. 13, 2022).
- [9] M. B. Andrés, “INTERNET DE LAS COSAS”.
- [10] “DIFERENCIA ENTRE UN MICROPROCESADOR Y UN MICROCONTROLADOR.” <https://1library.co/article/diferencia-entre-un-microprocesador-y-un-microcontrolador.yewr7oey> (accessed Jan. 26, 2023).
- [11] TAAE’2002, “V Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica,”
- [12] L. del V. Hernández, “Guía de introducción a MQTT con ESP8266 y Raspberry Pi.”
- [13] R. Fielding *et al.*, “Hypertext Transfer Protocol -- HTTP/1.1”.
- [14] “MATLAB - El lenguaje del cálculo técnico - MATLAB & Simulink.” <https://la.mathworks.com/products/matlab.html> (accessed Dec. 01, 2022).
- [15] W. Gander and U von Mat, *Solving Problems in Scientific Computing Using MAPLE, and MATLAB*. 1993.
- [16] S. W. , Smith, *The Scientist & Engineer’s Guide to Digital Signal Processing*. California Technical Pub., 1997.
- [17] C. Alberto Ruiz Marta Susana Basualdo Autor and D. Jorge Matich, “Cátedra: Informática Aplicada a la Ingeniería de Procesos-Orientación I Redes Neuronales: Conceptos Básicos y Aplicaciones”.
- [18] B. Martín-del-Brío and A. Sanz, *Redes neuronales y sistemas borrosos / B. Martín del Brío, A. Sanz Molina ; pról. de Lotfi A. Zadeh*. 2006.
- [19] M. Hudson, M. Hagan, and H. Demuth, “Deep Learning Toolbox Getting Started Guide - nnet_gs”.
- [20] B. Pecar and G. Davis, “Time series based predictive analytics modelling: using

MS excel.” Amazon Kindle: Seattle, WA, USA, 2018.

- [21] International Energy Agency., *World Energy Outlook 2012*. OECD/IEA, 2012.
- [22] J. Ibarra, “Artículo técnico ISO 50001, Gestión de Energía ISO 50001, Gestión de Energía”.
- [23] ISO, “ISO 50006:2017(E) - Energy management systems - Measuring energy performance using energy baselines (EnMS),” 2017. <https://www.en-standard.eu/bs-iso-50015-2014-energy-management-systems-measurement-and-verification-of-energy-performance-of-organizations-general-principles-and-guidance/#:~:text=This%20International%20Standard%20establishes%20general%20principles%20and%20guidelines,can%20be%20applied%20to%20all%20types%20of%20energy.> (accessed Jul. 12, 2023).
- [24] ISO, “ISO 50015:2014(E) - ‘Energy management systems - Measurement and verification of energy performance of organizations - General principles and guidance,’” 2014.
- [25] D. Singhal and K. S. Swarup, “Electricity price forecasting using artificial neural networks,” *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 3, pp. 550–555, Mar. 2011, doi: 10.1016/j.ijepes.2010.12.009.
- [26] M. J. RAMÍREZ PRADO, “DISEÑO Y PROTOTIPO DE UN SISTEMA DE MONITOREO DE CONSUMO ELÉCTRICO RESIDENCIAL UTILIZANDO UNA PLATAFORMA VIRTUAL DE INTERNET DE LAS COSAS.,” 2020.
- [27] Y. Telecomunicaciones, “UNIVERSIDAD DE GUAYAQUIL FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS CARRERA DE INGENIERÍA EN NETWORKING”.
- [28] R. López Suárez, “SISTEMA PARA EL MONITOREO DEL CONSUMO DE ENERGÍA ELÉCTRICA EN TIEMPO REAL.,” Oct. 26, 2020. https://www.researchgate.net/publication/359221006_SISTEMA_PARA_EL_MONITOREO_DEL_CONSUMO_DE_ENERGIA_ELECTRICA_EN_TIEMPO_

REAL (accessed Feb. 12, 2023).

ANEXOS

ANEXO 1: ESPECIFICACIONES TÉCNICAS DE EQUIPOS FÍSICOS.

Tabla 7. Especificaciones Pzem004t.

Medida	Características
Voltaje	Rango 80-260V. Precisión 0.5%.
Corriente	Rango 0-100A. Precisión 0.5%.
Potencia Activa	Rango 0-23Kw. Precisión 0.5%.
Factor de Potencia	Rango 80-260V. Precisión 0.5%.
Energía	Rango 0-99999kWh. Precisión 0.5%.
Frecuencia	Rango 45Hz-65Hz. Precisión 0.5%.

Tabla 8. Especificaciones Esp32 node mcu.

Característica	Especificación
Microprocesador	Xtensa dual-core 32-bit LX6 hasta 240 MHz
Memoria ROM	448 KB
Memoria SRAM	520 KB
Memoria SRAM en RTC	16 KB
Redes inalámbricas	802.11b/g/n
Velocidad de transferencia	802.11n superior a 150 Mbps
Agregación de frames	A-MPDU y A-MSDU
Rango de frecuencia	2412 ~ 2484 MHz
Controladores UART	UART0, UART1 y UART2
Periféricos SPI	4 periféricos SPI
Controlador SDIO	Controlador esclavo de tarjeta SDIO
Protocolo I2C	Comunicación semidúplex y síncrona

ANEXO 2: CONFIGURACIÓN RED LOCAL

Network

Wi-Fi Name: ROSSAN...
Password:
Secret Key:



Change

ANEXO 3: PROGRAMACION DE ESP32

CÓDIGO 1

```
// PZEM004Tv30 - Version: Latest

#include <PZEM004Tv30.h>

#define PZEM_RX_PIN 16

#define PZEM_TX_PIN 17

#define CONSOLE_SERIAL Serial

#define PZEM_SERIAL Serial2

float pzem_energy.

float pzem_potencia.
```

```
#include "thingProperties.h"

bool conmuta= true.

PZEM004Tv30 pzem (PZEM_SERIAL, 16, 17);

void setup () {

    // Initialize serial and wait for port to open:

    Serial.begin(9600);

    delay (1500);

    // Defined in thingProperties.h

    initProperties ();

    // Connect to Arduino IoT Cloud

    ArduinoCloud.begin(ArduinoIoTPreferredConnection).
```

the higher number the more granular information you'll get.

The default is 0 (only errors).

Maximum is 4

```

*/

setDebugMessageLevel (2);

ArduinoCloud.printDebugInfo().

CONSOLE_SERIAL.begin(115200).

int contador=0.

}

void loop() {

    pzem.readAddress();

    pzem_energy= pzem.energy();

    pzem_potencia=pzem.power();

    CONSOLE_SERIAL.print("Energy: ");
    CONSOLE_SERIAL.print(pzem_energy, 3);    CONSOLE_SERIAL.println("kWh");

    CONSOLE_SERIAL.print("POWER: ");
    CONSOLE_SERIAL.print(pzem_energy, 3);    CONSOLE_SERIAL.println("kWh");

    ArduinoCloud.update();

    delay(5000);

```

```
}

/*

  Since Programador is READ_WRITE variable, onProgramadorChange() is
  executed every time a new value is received from IoT Cloud.

*/

void onProgramadorChange() {

  // Add your code here to act upon Programador change

}
```

```
// Code generated by Arduino IoT Cloud, DO NOT EDIT.

#include <ArduinoIoTCloud.h>

#include <Arduino_ConnectionHandler.h>

const char DEVICE_LOGIN_NAME[] = "92ef6d64-8901-4b3e-b279-
de7543701e5c";
```

```
const char SSID[]          = SECRET_SSID; // Network SSID (name)

const char PASS[]         = SECRET_OPTIONAL_PASS; // Network
password (use for WPA, or use as key for WEP)

const char DEVICE_KEY[] = SECRET_DEVICE_KEY; // Secret device
password

float pzem_energy;

float pzem_potencia;

void initProperties(){

    ArduinoCloud.setBoardId(DEVICE_LOGIN_NAME);

    ArduinoCloud.setSecretDeviceKey(DEVICE_KEY);

    ArduinoCloud.addProperty(pzem_energy, READ, 3600 * SECONDS,
NULL);

    ArduinoCloud.addProperty(pzem_potencia, READ, 600 * SECONDS,
NULL);

}
```

```
WiFiConnectionHandler ArduinoIoTPreferredConnection(SSID, PASS);
```

CÓDIGO

```
#include <ThingSpeak.h>

#include <PZEM004Tv30.h>

#if !defined(PZEM_RX_PIN) && !defined(PZEM_TX_PIN)

#define PZEM_RX_PIN 16

#define PZEM_TX_PIN 17

#endif

#define CONSOLE_SERIAL Serial

#define PZEM_SERIAL Serial2

#include <SoftwareSerial.h>

SoftwareSerial pzemSWSerial(PZEM_RX_PIN, PZEM_TX_PIN);

#include "thingProperties.h"

/*
```

Sketch generated by the Arduino IoT Cloud Thing "Tesis_1"

<https://create.arduino.cc/cloud/things/2de5c240-e1e7-4143-a36f-819ac49d2725>

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the Thing

```
float pzem_current;
```

```
float pzem_current2;
```

```
float pzem_poten;
```

```
float pzem_poten2;
```

```
float pzem_voltaje;
```

```
float pzem_voltaje2;
```

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions

which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.


```

    {

        // Initialize the PZEMs with Hardware Serial2 on RX/TX pins 16 and 17

        pzems[i] = PZEM004Tv30(PZEM_SERIAL, PZEM_RX_PIN,
PZEM_TX_PIN, 0x10 + i);

    }

    setDebugMessageLevel(2);

    ArduinoCloud.printDebugInfo();

    CONSOLE_SERIAL.begin(115200);

    int contador = 0;
}

void loop() {

    for(int i = 0; i < NUM_PZEMS; i++){

        pzems[i].getAddress();

        if (i==0){

            pzem_poten=pzems[i].power();

```

```

    pzem_current=pzems[i].current();

    pzem_voltaje=pzems[i].voltage();

    if(isnan(pzem_poten)){

        CONSOLE_SERIAL.println("Error reading potencia1");

    }

    }else if (i==1)

    {

        pzem_poten2=pzems[i].power();

        pzem_current2=pzems[i].current();

        pzem_voltaje2=pzems[i].voltage();

        if(isnan(pzem_poten2)){

            CONSOLE_SERIAL.println("Error reading potencia2");

        }

    }

    }

    for(int i = 0; i < NUM_PZEMS; i++){

    if (i==0){

```

```
ThingSpeak.setField(1, pzem_poten);

}else if(i==1)

{

    ThingSpeak.setField(2, pzem_poten2);

}

}

ArduinoCloud.update();

int httpCode = ThingSpeak.writeFields(channelID, apiKey);

if (httpCode == 200) {

    Serial.println("Datos enviados correctamente a ThingSpeak.");

} else {

    Serial.println("Error al enviar datos a ThingSpeak. Código HTTP: " +
String(httpCode));

}
```

```
delay(5000);  
  
}
```

ANEXO 4 CONFIGURACIÓN IFTTT

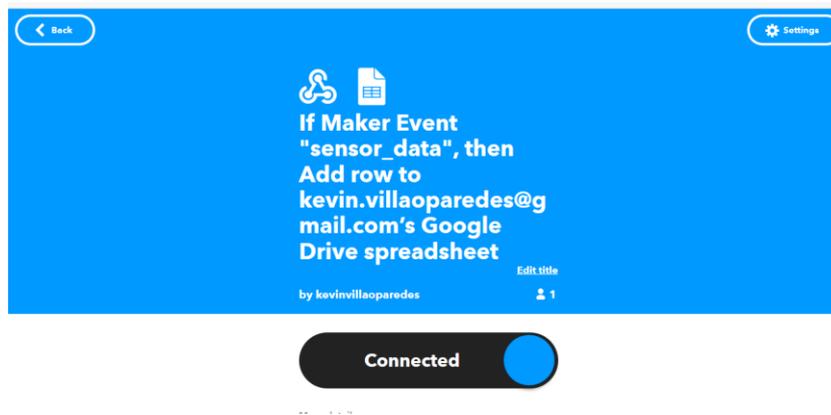


Figura 29. Creación de eventos IFTTT.

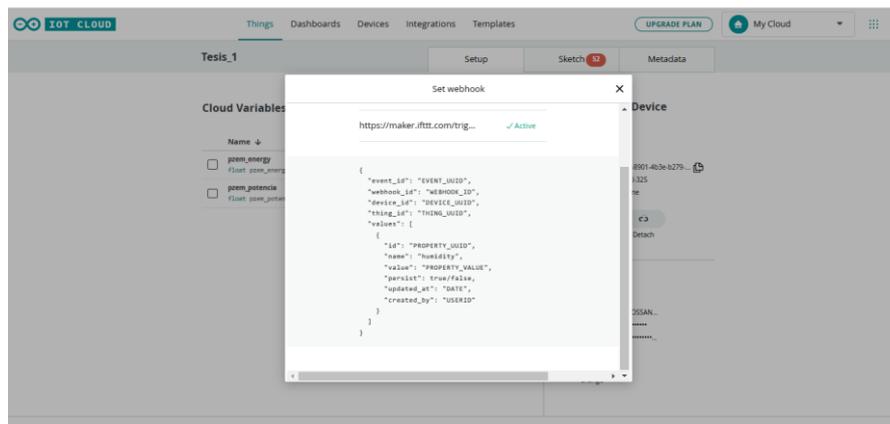


Figura 30. Webhook en arduino IoT cloud.

	A	B	C	D	E	F	G	H	I	J	K
1	January 11, 2023 at 01:53AM	sensor_data	pzem_voltage	12.130.000.306	2023-01-11T06:53:32.067Z						
2	January 11, 2023 at 01:53AM	sensor_data	pzem_energy	24.899.999.797	2023-01-11T06:53:32.067Z						
3	January 11, 2023 at 01:53AM	sensor_data	pzem_lp	8.899.999.856.9	2023-01-11T06:53:37.195Z						
4	January 11, 2023 at 01:53AM	sensor_data	pzem_frecuency	59.900.001.525	2023-01-11T06:53:32.067Z						
5	January 11, 2023 at 01:53AM	sensor_data	pzem_energy	24.899.999.797	2023-01-11T06:53:37.195Z						
6	January 11, 2023 at 01:53AM	sensor_data	pzem_current	4.269.999.861.7	2023-01-11T06:53:32.067Z						
7	January 11, 2023 at 01:53AM	sensor_data	pzem_frecuency	60.0	2023-01-11T06:53:37.195Z						
8	January 11, 2023 at 01:53AM	sensor_data	pzem_lp	8.899.999.856.9	2023-01-11T06:53:32.067Z						
9	January 11, 2023 at 01:53AM	sensor_data	pzem_current	4.269.999.861.7	2023-01-11T06:53:37.195Z						
10	January 11, 2023 at 01:53AM	sensor_data	pzem_voltage	1.210.999.904.7	2023-01-11T06:53:37.195Z						
11	January 11, 2023 at 01:53AM	sensor_data	pzem_current	4.269.999.861.7	2023-01-11T06:53:42.329Z						
12	January 11, 2023 at 01:53AM	sensor_data	pzem_energy	24.899.999.797	2023-01-11T06:53:42.329Z						
13	January 11, 2023 at 01:53AM	sensor_data	pzem_lp	8.899.999.856.9	2023-01-11T06:53:42.329Z						
14	January 11, 2023 at 01:53AM	sensor_data	pzem_voltage	12.130.000.306	2023-01-11T06:53:42.329Z						
15	January 11, 2023 at 01:53AM	sensor_data	pzem_frecuency	60.0	2023-01-11T06:53:42.329Z						
16	January 11, 2023 at 01:53AM	sensor_data	pzem_energy	24.899.999.797	2023-01-11T06:53:47.463Z						
17	January 11, 2023 at 01:53AM	sensor_data	pzem_current	4.259.999.990.4	2023-01-11T06:53:47.463Z						
18	January 11, 2023 at 01:53AM	sensor_data	pzem_lp	8.899.999.856.9	2023-01-11T06:53:47.463Z						
19	January 11, 2023 at 01:53AM	sensor_data	pzem_voltage	1.210.999.904.7	2023-01-11T06:53:47.463Z						
20	January 11, 2023 at 01:53AM	sensor_data	pzem_frecuency	59.900.001.525	2023-01-11T06:53:47.463Z						
21	January 11, 2023 at 01:53AM	sensor_data	pzem_lp	8.899.999.856.9	2023-01-11T06:53:52.596Z						
22	January 11, 2023 at 01:53AM	sensor_data	pzem_current	4.259.999.990.4	2023-01-11T06:53:52.596Z						

Figura 31. Hojas de datos de Potencia

ANEXO 5: PROGRAMACIÓN Y GRÁFICAS DE PROCESAMIENTO DE DATOS DE POTENCIA.

PROGRAMACIÓN PRINCIPAL.

```

%%Procesamiento de señal

Nan2Horas=ismissing(HORA);

Nan=ismissing(FECHA);

u=1;

for i=1:1:numel(FECHA)

    if Nan(i,1)~=1;

        c=FECHA(i,1);

        Fechas(u,1)=c;
    end
end

```

```

else
    Fechas (u,1)=c;
    u=u+1;
end
end
z=":00:00"
u=1;
for i=1:1:numel (HORA)
    if Nan2Horas (i,1)~=1
        Horasr (u,1)=HORA (i,1);
        Po (u,1)=MxdePOTENCIAL (i,1);
        u=u+1;
    else
        u=u;
    end
end
u=1;
for i=1:1:numel (HORA)
    if Nan2Horas (i,1)~=1
        Hor (u,1)=HORA (i,1);

        u=u+1;
    else
        u=u;
    end
end
end
%
```

```

% u=1;

% for i=1:1:numel(Fechas)

%
%     FECHATH(i,1)=Fechas(i,1)+Horasr(i,1);
%
% end

u=1;
for i=1:1:numel(Fechas)

    FECHATH(i,1)=Fechas(i,1);
Pox(i,1)=Po(i,1);
    Hrr(i,1)=Hor(i,1);
end

[dia, mes, ano] = separarFecha(FECHATH);
[diasel, festivos] = identificarDiasFestivosEcuador(FECHATH);
%%

%%FILTRADO DE SEÑALES

%%Datos de ingreso a red neuronal.

%identificacion de dias festivos y dias de semana

```

```

%%Potencia nominal y capacidad en litros de nevera
for i=1:1:numel(FECHATH)
PN(i,1)=256;
Cap(i,1)=500;
end

%%Potencia nominal
for i=1:1:numel(FECHATH)
PN(i,1)=256;
Cap(i,1)=500;
end

%%NEVERA
in=table(FECHATH,rms);
input2R=table(anio,mes,dia, festivos, diase, Horr, PN, Cap)
writetable(input2,'dDD1.xlsx','sheet','WriteTable','Range','A1');
fechp='1-Jul-2023';
[dial, mes1, ano1] = separarFecha(fechp);
[diasel1, festivos1] = identificarDiasFestivosEcuador(fechp);

x1=[ano1 mes1 dial festivos1 diasel 12 256 500]

%%POTENCIA Aire acondicionado.
u=1;
for i=1:1:numel(HORA)

```

```

if Nan2Horas(i,1)~=1

    PoA(u,1)=MxdePotenciaAire(i,1);

    % PoIL(u,1)=MxPotenciaIL(i,1);

    u=u+1;

else

    u=u;

end

end

for i=1:1:numel(Fechas)

    PoxA(i,1)=PoA(i,1);

    % PoxIL(i,1)=PoIL(i,1);

end

%%Procesamiento de datos de aireA

%Normalizar datos.

alpha=0.8;s=0;

PoxAr=PoxA/1000;

% sA = zeros(size(PoxAr));

% for i = 1:size(PoxAr, 2)

%     s = alpha * PoxAr(:, i) + (1 - alpha) * s;

%     sA(:, i) = s;

% end

;

sA = sgolayfilt(PoxAr, 2, 155);

%Mediana

```

```

pmdA=medfilt1(sA,150)

%Mediana Movil

sqA=pmdA.^2

rmsA=sqrt(movmean(sqA,150))

%%GRÁFICAS

figure('Name', 'Datos de potencia Nevera')

subplot(2,2,1)

plot(FECHATH, PoxAr, 'color', 'blue')

title('Gráfica de Potencia Aire Acondicionado')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,2)

plot(FECHATH, sA, 'color', '#FF0000')

title('Gráfica de Filtro Savitzky-Golay:')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,3)

plot(FECHATH, pmdA, 'color', [0.2, 0.7, 0.3])

title('Gráfica de Filtro Mediana')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,4)

plot(FECHATH, rmsA, 'color', 'magenta')

```

```

title('Gráfica de Filtro de media móvil')

xlabel('Tiempo')
ylabel('Potencia (kW)')

% Parámetros del filtro de Kalman
A = 1; % Matriz de transición de estado
C = 1; % Matriz de observación
Q = 0.1; % Covarianza del ruido del proceso
R = 1; % Covarianza del ruido de medición

% Variables de estado inicial y estimación inicial
x = 0; % Estado inicial
P = 1; % Covarianza inicial

% Vector para almacenar las estimaciones
estimaciones = zeros(size(PoxAr)); % Y es tu vector de potencia

% Aplicar el filtro de Kalman
for i = 1:numel(PoxAr)
    % Predicción del estado
    x_pred = A * x;
    P_pred = A * P * A' + Q;

    % Actualización de la estimación
    K = P_pred * C' / (C * P_pred * C' + R);
    x = x_pred + K * (PoxAr(i) - C * x_pred);
    P = (eye(size(P)) - K * C) * P_pred;
end

```

```

    % Almacenar la estimación
    estimaciones(i) = x;
end

% Graficar los resultados en dos gráficas separadas
subplot(2, 1, 1);
plot(FECHATH, PoxAr, 'b');
title('Datos originales');
xlabel('Tiempo');
ylabel('Potencia (kW)');

subplot(2, 1, 2);
plot(FECHATH, estimaciones, 'r');
title('Estimaciones del filtro de Kalman');
xlabel('Tiempo');
ylabel('Potencia (kW)');

%%
%%Potencia nominal
for i=1:1:numel(FECHATH)
PNA(i,1)=822/1000;
Area(i,1)=32;
pers(i,1)=1;
end
%Uso de aire acondicionado
verinv= estacion(FECHATH)

```

```

usoAire = usoH(PoxAr);

%%Entradas para el entrenamiento

input2A=table(ano,mes,dia, festivos,diasel,usoAire,verinv,pers,PNA,Ar
ea)

writetable(input2A,'datosAire2.xlsx','sheet','WriteTable','Range','A
1');

aire=table(FECHATH,rmsA)

fechpA='7-Jul-2023';

fechapa = datetime(fechpA); % Convertir a objeto datetime

[dialA, mes1A, ano1A] = separarFecha(fechpA);

[diaselA, festivos1A] = identificarDiasFestivosEcuador(fechpA);

verinv1= estacion(fechapa);

x1A=[ano1A mes1A dialA festivos1A diaselA 0 verinv1 1 0.822 32];

[y1A] = redneuronalar(x1A)

%%

%Carga de datos de iluminarias

u=1;

for i=1:1:numel(HORA)

    if Nan2Horas(i,1)~=1

        PoIL(u,1)=MxdePOTENCIAIL(i,1);

        u=u+1;

    else

        u=u;

```

```

    end

end

for i=1:1:numel(Fechas)

    PoxIL(i,1)=PoIL(i,1);

end

%%Procesamiento de datos Iluminarias

%Normalizar datos.

alpha=0.8;s=0;

% sA = zeros(size(PoxAr));
% for i = 1:size(PoxAr, 2)
%     s = alpha * PoxAr(:, i) + (1 - alpha) * s;
%     sA(:, i) = s;
% end

;

sIL = sgolayfilt(PoxIL, 2, 55);

%Mediana Movil

sqIL=sIL.^2

rmsIL=sqrt(movmean(sqIL,100))

%Mediana

pmdIL=medfilt1(rmsIL,50)

%%GRAFICAS

figure('Name', 'Datos de potencia Iluminarias')

```

```

subplot(2,2,1)

plot(FECHATH, PoxIL, 'color', 'blue')

title('Gráfica de Potencia Iluminarias')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,2)

plot(FECHATH, sIL, 'color', '#FF0000')

title('Gráfica de Filtro Savitzky-Golay:')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,3)

plot(FECHATH, rmsIL, 'color', 'magenta')

title('Gráfica de Filtro de media móvil')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,4)

plot(FECHATH, pmdIL, 'color', [0.2, 0.7, 0.3])

title('Gráfica de Filtro Mediana')

xlabel('Tiempo')

ylabel('Potencia (kW)')

%%

%%Potencia nominal Iluminarias

```

```

for i=1:1:numel(FECHATH)

PNIL(i,1)=110;

end

hora_amanecer = 6;

hora_anocheecer = 18;

dn = clasificarDiaNoche(Horr, hora_amanecer, hora_anocheecer)

%%Entradas para el entrenamiento

input2A1=table(ano,mes,dia, festivos, diasel, Horr, dn, PNIL)

writetable(input2A1, 'datosilum.xlsx', 'sheet', 'WriteTable', 'Range', 'A
1');

fechpA='1-May-2023';

fechapa = datetime(fechpA); % Convertir a objeto datetime

[dia1L, mes1L, ano1L] = separarFecha(fechpA);

[diaselL, festivos1L] = identificarDiasFestivosEcuador(fechpA);

xIL=[ano1L mes1L dia1L festivos1L diaselL 12 0 110];

[yIL] = redneuonalIL(xIL)

nev=table(FECHATH, rms)

writetable(nev, 'never.xlsx', 'sheet', 'WriteTable', 'Range', 'A1');

aire=table(FECHATH, rmsA)

writetable(aire, 'airer.xlsx', 'sheet', 'WriteTable', 'Range', 'A1');

```

```

ilu=table(FECHATH,pmdIL);
writetable(ilu,'ilum.xlsx','sheet','WriteTable','Range','A1');

%%VALIDACION NEVERA

vx1=table(aniol,mes1,dial,festivos1,diasel,Horrl,PN1,Cap1)
x1=[aniol mes1 dial festivos1 diasel Horrl PN1 Cap1]

[yN]=redneuronalR(x1)

% Etiqueta y título para la primera gráfica
subplot(1,2,1)
plot(FECHATH, rms1, 'color', [0.2, 0.7, 0.3])
xlabel('Fecha')
ylabel('Potencia')
title('POTENCIA REAL')

% Etiqueta y título para la segunda gráfica
subplot(1,2,2)
plot(FECHATH, yN, 'color', [0.2, 0.7, 0.3])
xlabel('Fecha')
ylabel('Potencia')
title('POTENCIA DE LA RED NEURONAL ARTIFICIAL')

%%Validacion AIRE ACONDICIONADO
x1A=[ano mes dia festivos diasel usoAire verinv pers PNA Area];

```

```

[yA] = redneuronalAR(x1A)

subplot(1,2,1)

plot(FECHATH,rmsA , 'color', [0, 0, 0])

xlabel('Fecha')

ylabel('Potencia (Kw)')

title('POTENCIA REAL')

% Etiqueta y título para la segunda gráfica

subplot(1,2,2)

plot(FECHATH, yA, 'color', [1, 0, 0])

xlabel('Fecha')

ylabel('Potencia (Kw)')

title('POTENCIA DE LA RED NEURONAL ARTIFICIAL')

%%VALIDACION ILUMINARIA

xIL=[ano mes dia festivos diasel Horr dn PNIL];

[yIL] = redneuronalIL(xIL)

subplot(1,2,1)

plot(FECHATH,pmdIL , 'color', [0, 0, 0])

xlabel('Fecha')

ylabel('Potencia (W)')

title('POTENCIA REAL')

% Etiqueta y título para la segunda gráfica

subplot(1,2,2)

```

```
plot(FECHATH,yIL , 'color', [1, 0, 0])  
  
xlabel('Fecha')  
  
ylabel('Potencia (W)')  
  
title('POTENCIA DE LA RED NEURONAL ARTIFICIAL')
```

FUNCION BINARIA DE CLASIFICACION DIA-NOCHE.

```
function resultados = clasificarDiaNoche(Horas, hora_amanecer,  
hora_anohecer)  
  
    % Inicializar vector de resultados  
  
    resultados = zeros(size(Horas));  
  
    % Calcular resultado para cada elemento en Horas  
  
    for i = 1:length(Horas)  
  
        if Horas(i) >= hora_amanecer && Horas(i) <  
hora_anohecer  
  
            resultados(i) = 1; % Es de día  
  
        else  
  
            resultados(i) = 0; % Es de noche  
  
        end  
  
    end
```

```
end

end
```

FUNCIÓN IDENTIFICAR DÍAS FESTIVOS Y DÍA DE SEMANA.

```
function [diaSemana, esFestivo] =
identificarDiasFestivosEcuador (fecha)

    % Definir las fechas de los días festivos en Ecuador

    festivos = {

        '01-Jan',    'Año Nuevo';

        '24-Feb',    'Carnaval';

        '25-Feb',    'Carnaval';

        '02-Apr',    'Viernes Santo';

        '01-May',    'Día del Trabajo';

        '24-May',    'Batalla de Pichincha';

        '10-Aug',    'Primer Grito de Independencia';

        '09-Oct',    'Independencia de Guayaquil';

        '02-Nov',    'Día de los Difuntos';

        '25-Dec',    'Navidad'

    };
```

```

    % Convertir la fecha a formato de fecha de MATLAB

    fecha_matlab = datetime(fecha, 'Format', 'dd-MMM-yyyy');

    % Obtener el número correspondiente al día de la semana
    (1: Domingo, 2: Lunes, ..., 7: Sábado)

    diaSemana = weekday(fecha_matlab);

    % Convertir la fecha a formato 'dd-mmm' para comparación

    fecha_comparacion = datestr(fecha_matlab, 'dd-mmm');

    % Verificar si la fecha es un día festivo en Ecuador

    esFestivo = ismember(fecha_comparacion, festivos(:, 1),
'rows');

    end

```

FUNCIÓN PARA SEPARAR DÍA MES Y AÑO

```

function [dia, mes, anio] = separarFecha(Fechas)

Fecha2 = cellstr(Fechas);

```

```

dia = zeros (length (Fecha2),1);

mes = zeros (length (Fecha2),1);

anio = zeros (length (Fecha2),1);

% Extraer año, mes y día de cada fecha

for i = 1:numel (Fecha2)

    fecha = datetime (Fecha2{i}, 'InputFormat', 'dd-MMM-yyyy');

    anio (i) = year (fecha);

    mes (i) = month (fecha);

    dia (i) = day (fecha);

end

end

```

ANEXO 6: FILTRADO DE SEÑAL DE POTENCIA

```

% Savitzky-Golay

s = sgolayfilt (Pox, 2, 27);

```

```

%Mediana

pmd=medfilt1(s,100)

%media movil

series=datenum(FECHATH)

rms=sqrt(movmean(sq,50))

%%

%%GRAFICAS

figure('Name', 'Datos de potencia Nevera')

subplot(2,2,1)

plot(FECHATH, Pox, 'color', 'blue')

title('Gráfica de Potencia Nevera')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,2)

plot(FECHATH, s, 'color', '#FF0000')

title('Gráfica de Filtro Savitzky-Golay:')

xlabel('Tiempo')

ylabel('Potencia (kW)')

```

```
subplot(2,2,3)

plot(FECHATH, pmd, 'color', [0.2, 0.7, 0.3])

title('Gráfica de Filtro Mediana')

xlabel('Tiempo')

ylabel('Potencia (kW)')

subplot(2,2,4)

plot(FECHATH, rms, 'color', 'magenta')

title('Gráfica de Filtro de media móvil')

xlabel('Tiempo')

ylabel('Potencia (kW)')
```

ANEXO 7: GRÁFICAS DE PROCESAMIENTO DE DATOS DE POTENCIA

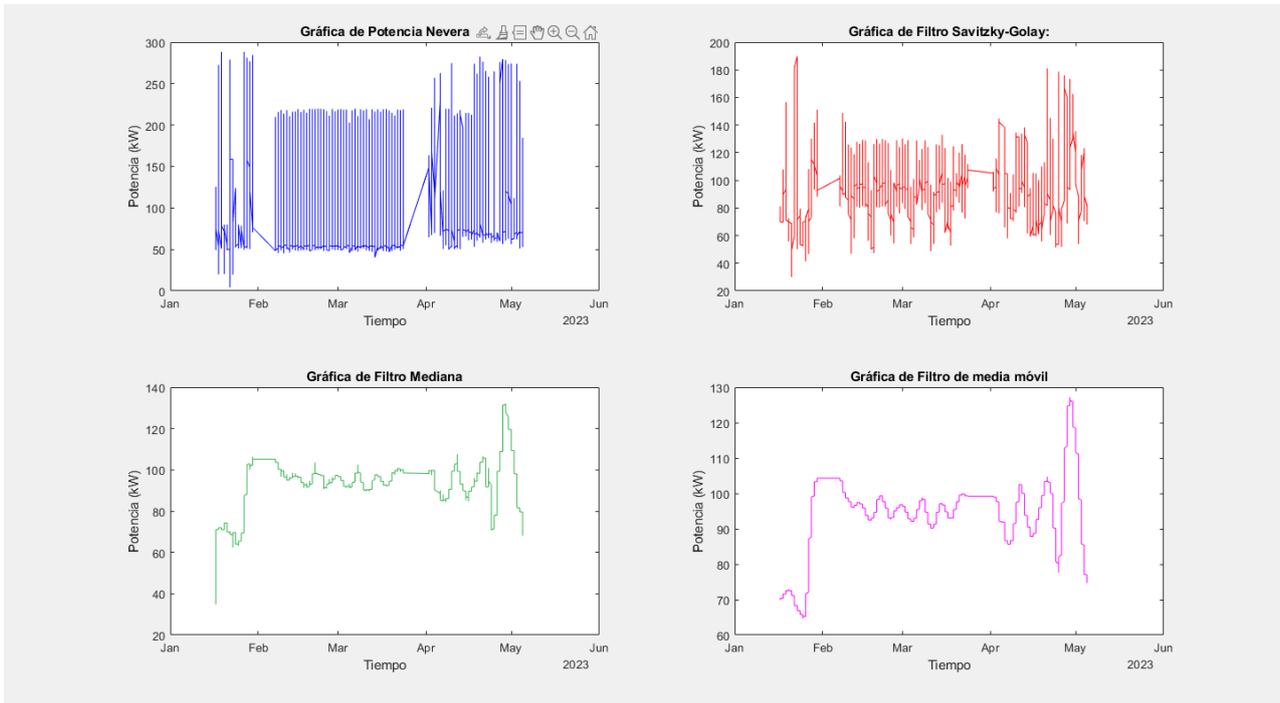


Figura 32. Gráfica de procesamiento de datos de potencia de N1.

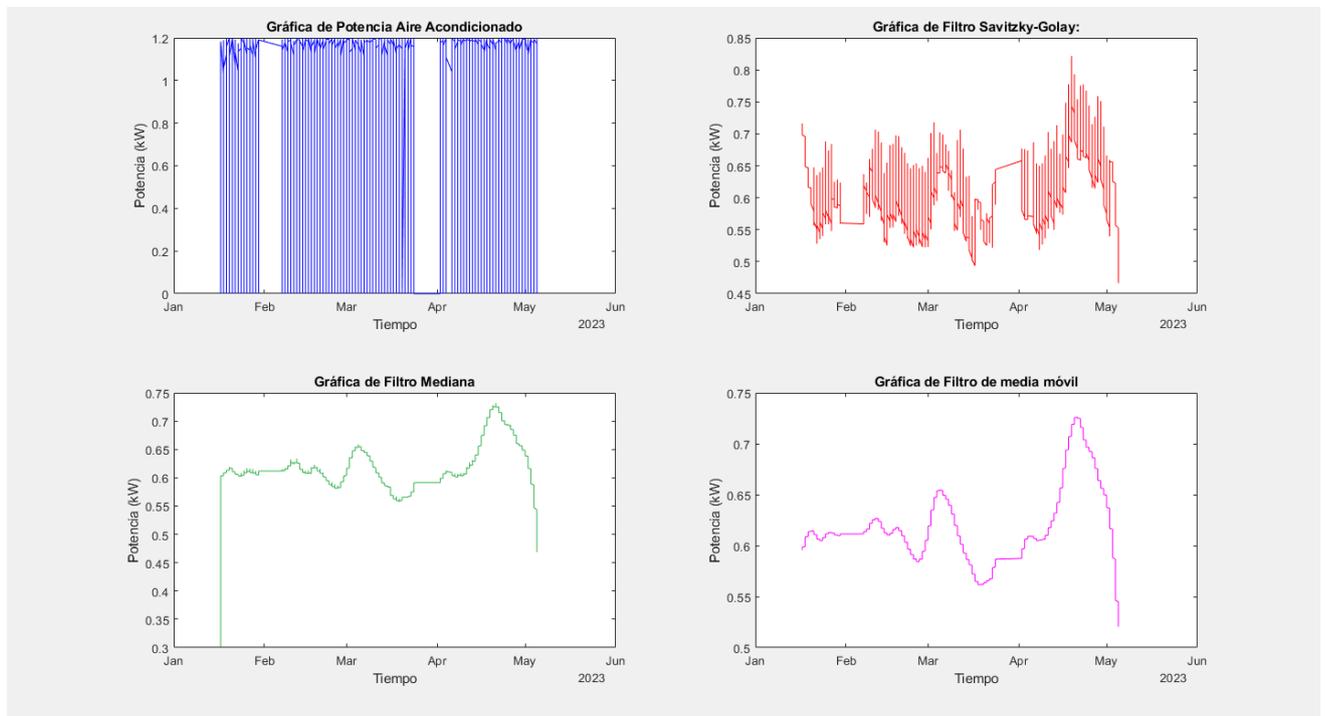


Figura 33. Gráfica de procesamiento de datos de potencia de A.A.

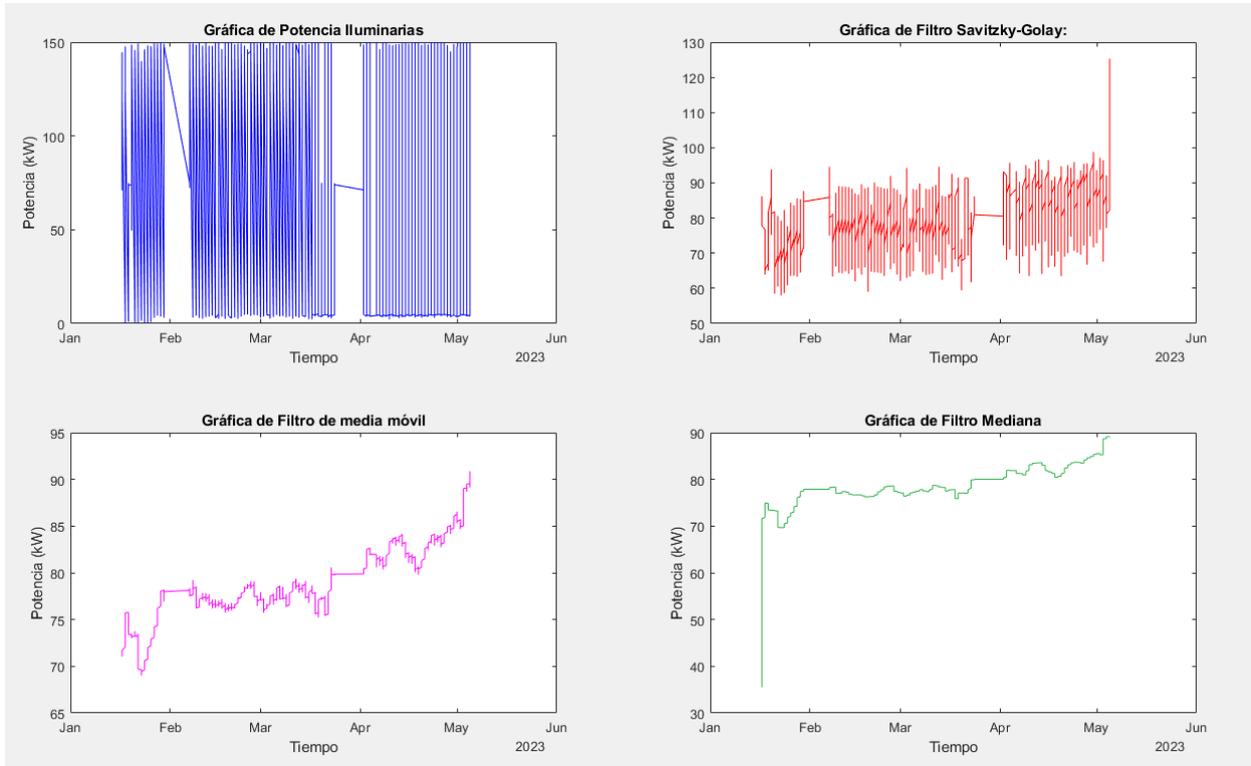


Figura 34. Gráfica de procesamiento de datos de potencia de I.1.

ANEXO 8 CREACION DE RED NEURONAL ARTIFICIAL PARA LA NEVERA.

Import Wizard

Preview of C:\Users\Kevin\OneDrive\Escritorio\Octavo Semestre\Tesis\Tesis\APP\NeveraEntradas.xlsx

Worksheets: WriteTable

	1	2	3	4	5
1	1	17	3	0	16
2	1	17	3	0	17
3	1	17	3	0	18
4	1	17	3	0	19
5	1	17	3	0	20
6	1	17	3	0	21
7	1	17	3	0	22
8	1	17	3	0	23
9	1	18	4	0	0
10	1	18	4	0	1
11	1	18	4	0	2
12	1	18	4	0	3
13	1	18	4	0	4
14	1	18	4	0	5

Buttons: Help, < Back, Next >, Finish, Generate MATLAB code, Cancel

Figura 35. Datos de entrada de la RNA de N1.

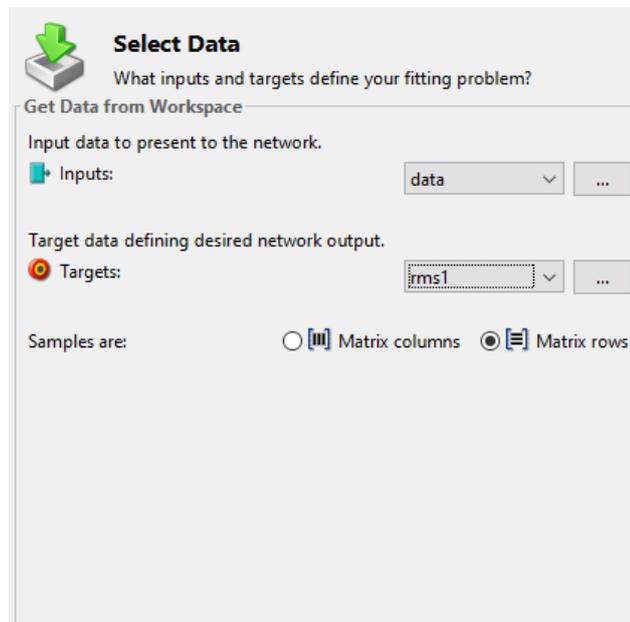


Figura 36. Datos de ingreso de la RNA de N1

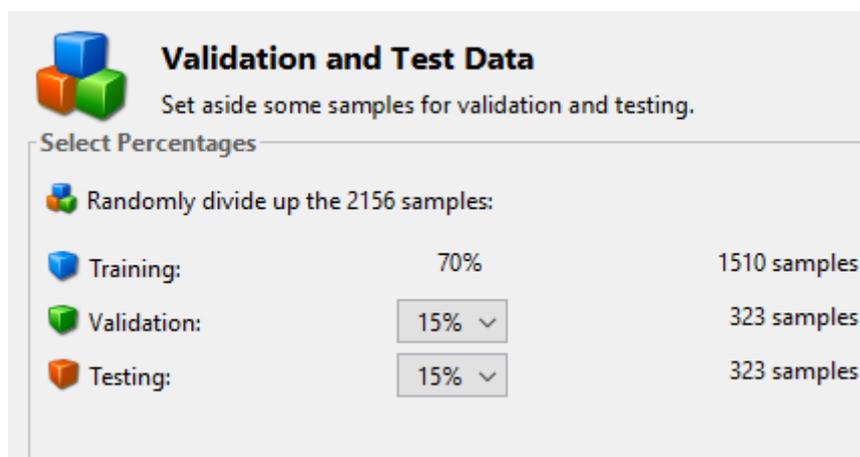


Figura 37. Configuración de porcentajes para pruebas y validaciones para N1.

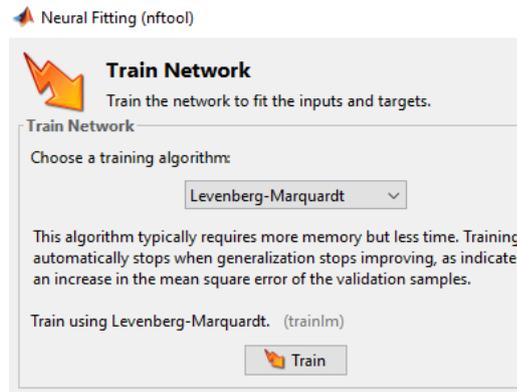


Figura 38. Entrenamiento de la RNA de N1.

RED NEURONAL ARTIFICIAL

```
function [y1] = redneuronalR(x1)

%MYNEURALNETWORKFUNCTION neural network simulation function.

%

% Auto-generated by MATLAB, 28-Jun-2023 03:02:09.

%

% [y1] = myNeuralNetworkFunction(x1) takes these arguments:

%   x = Qx8 matrix, input #1

% and returns:

%   y = Qx1 matrix, output #1

% where Q is the number of samples.

%#ok<*RPMT0>
```

```

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1

x1_step1.keep = [2 3 4 5 6];

x1_step2.xoffset = [1;1;1;0;0];

x1_step2.gain =
[0.5;0.0689655172413793;0.3333333333333333;2;0.0869565217391304];

x1_step2.ymin = -1;

% Layer 1

b1 = [-0.46128947478583370856;2.9665470386388945734;-
2.6990627063120684603;10.771060663926519396;-
0.20988623848205001376;-0.54166699117276073672;-
2.7515511281719371439;-
1.9735649399134000959;0.0035866100555371187991;1.5162598294041012004
;1.6101362365611473493;-
0.41291457069226833454;0.22517279946754181763;1.6851804312529703456;
0.67402237807717213514;-2.7106592431539646704;-
0.27455487819537804928;-
3.8391045920418802595;0.82861225796128334942;8.8974127192667822328];

IW1_1 = [1.6082148071036119763 -0.056068710688801912112 -
0.21352498739491074242 1.7161773323257834001
0.0077436451942531157183;-0.11484026918094117353
0.36909175588944848068 -1.4476582951715004643 3.0773773230789003286
-0.21051398195456083595;-4.1643415016329923262
0.93403082452425512106 2.4506949501824131232 -4.8023171582729906604
0.38646184115493775746;-0.34353510467629844172 -
6.9926483635876053313 -4.3293544236736538622 -0.86077204973793619569
-0.56977210075394701949;-1.63329225340219919 -2.7817369450623585792
1.9805557761171039832 -0.34592816749890381578
0.20819378966177345958;2.1135538614305118266 -

```

```
0.029756578095205123841 -0.43603878084909103441 1.849008712962874057
-0.016076119958298652063;0.82939586573815038406
5.3507146537073362325 -2.064510735891773141 2.4991852910843865843 -
0.15284717472538414396;-5.2460943610291828065 -4.7525433017558889048
-9.0484650409957652073 2.608107951168822769 1.0116625331479054406;-
3.2512612147270880669 -0.46063954847555615979 -1.2921859483448383532
-0.69514579795535902385 3.3299477644585264713;-1.651724825008676234
-3.1300633808810540692 -0.60292437699633694148 2.4672233368466733872
-0.16380261846316149232;-0.70663004183613642439 -
5.4384279261731220956 2.5693487425115666412 -4.0047395403905126088
0.2303805224568417076;-1.0140459700741655258 0.36442291892119055285
-0.8364125444270972265 -2.4384101512636311426 -
4.5763279071830851308;0.40857730582168638733 1.8033986463351123763 -
1.5496028488308621895 0.55315929279399134177 -
0.17937675535268141069;5.7112540770970516846 -2.3969061313232535859
3.0445017112123520242 -0.096342793245699220872
0.39527719225873314546;2.3918085236562487417 1.8373064755716121965
2.0377276228648621981 1.3331837429435289266
0.36136555988748164658;2.2202098945538883967 1.7093550154813579312
2.0241935074683539142 -2.026799428707825701
0.36283382845970396646;5.4426650675146897029 9.7302093373507716478
0.73517350037365991611 -3.7545351160305595606
0.41365899525270782844;0.58929695647135649228 11.308458078007561198
-0.037944578292384435747 4.6632027769788688687
0.35099661878396148929;1.5482412599207480142 0.18173191899319704468
0.36150440772469427664 -0.52949702234281625923
4.2300234843294495235;15.736412979317668359 -12.290549990297312988 -
0.11427497171240687257 -7.2350004115960109274 -
0.29957455890082029804];
```

```
% Layer 2
```

```
b2 = -2.1503555367505842177;
```

```
LW2_1 = [-4.4912114016562743046 0.23106913335817055866
0.0898945017779798905 0.37822230644986376547 -0.76162899033211406419
2.8909322328820330483 -2.4927087649921428891 -
```

```

0.010543791224352904584 0.004074813858210194796 -
0.25955133484546111244 -2.2606823065659042982 -
0.003155883025737425164 -1.1102030954637631677 -
0.11888086703431059721 1.538533244300543279 -1.6109990661624158381 -
0.22481800911497915041 0.57723061931700814231
0.0037990733132031752339 0.40847933320202067797];

% Output 1

y1_step1.ymin = -1;

y1_step1.gain = 0.0320108778388514;

y1_step1.xoffset = 64.6355901074244;

% ===== SIMULATION =====

% Dimensions

Q = size(x1,1); % samples

% Input 1

x1 = x1';

xp1 = removeconstantrows_apply(x1,x1_step1);

xp1 = mapminmax_apply(xp1,x1_step2);

% Layer 1

a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

```

```

% Layer 2

a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1

y1 = mapminmax_reverse(a2,y1_step1);

y1 = y1';

end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function

function y = mapminmax_apply(x,settings)

y = bsxfun(@minus,x,settings.xoffset);

y = bsxfun(@times,y,settings.gain);

y = bsxfun(@plus,y,settings.ymin);

                                end

% Remove Constants Input Processing Function

function y = removeconstantrows_apply(x,settings)

y = x(settings.keep,:);

end

% Sigmoid Symmetric Transfer Function

```

```

function a = tansig_apply(n,~)

a = 2 ./ (1 + exp(-2*n)) - 1;

end

% Map Minimum and Maximum Output Reverse-Processing Function

function x = mapminmax_reverse(y,settings)

x = bsxfun(@minus,y,settings.ymin);

x = bsxfun(@rdivide,x,settings.gain);

x = bsxfun(@plus,x,settings.xoffset);

end

```

ANEXO 9: CONFIGURACIÓN DE LA RED NEURONAL ARTIFICIAL PARA A.A.

The screenshot shows the 'Import Wizard' dialog box in MATLAB. The title bar reads 'Import Wizard'. Below the title bar, the file path is shown: 'C:\Users\Kevin\OneDrive\Escritorio\Octavo Semestre\Tesis\Tesis\APP\Aireentradas.xlsx'. The dialog is divided into two main sections. On the left, under 'Worksheets', the 'WriteTable' option is selected. On the right, a preview of the spreadsheet data is shown. The data is organized into columns: 'data', 'textdata', 'colheaders', and six numbered columns (1-6). The rows are numbered 1 through 14. The data values are as follows:

	1	2	3	4	5	6
1	1	17	3	0	1	0
2	1	17	3	0	1	0
3	1	17	3	0	1	0
4	1	17	3	0	1	0
5	1	17	3	0	0	0
6	1	17	3	0	0	0
7	1	17	3	0	0	0
8	1	17	3	0	0	0
9	1	18	4	0	0	0
10	1	18	4	0	0	0
11	1	18	4	0	0	0
12	1	18	4	0	0	0
13	1	18	4	0	0	0
14	1	18	4	0	0	0

At the bottom of the dialog, there are several buttons: 'Help', '< Back', 'Next >', 'Finish', a checkbox for 'Generate MATLAB code', and 'Cancel'. The 'Next >' button is currently selected and highlighted with a blue border.

Figura 39. Datos de entrada de la RNA de A.A.

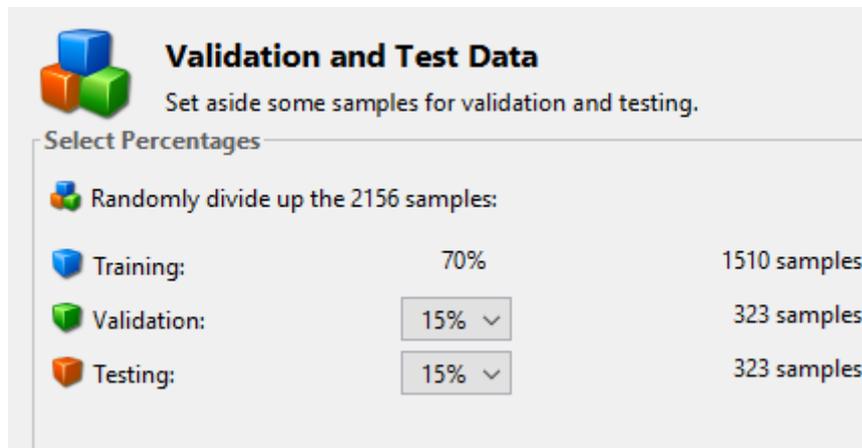


Figura 40. Configuración de porcentajes para pruebas y validaciones para A.A.

RED NEURONAL ARTIFICIAL

```
function [y1] = redneuronAR(x1)

%MYNEURALNETWORKFUNCTION neural network simulation function.

%

% Auto-generated by MATLAB, 28-Jun-2023 15:51:26.

%

% [y1] = myNeuralNetworkFunction(x1) takes these arguments:

%   x = Qx9 matrix, input #1

% and returns:

%   y = Qx1 matrix, output #1
```

```

% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1

x1_step1.keep = [2 3 4 5 6];

x1_step2.xoffset = [1;1;0;1;0];

x1_step2.gain =
[0.5;0.0689655172413793;2;0.3333333333333333;0.0869565217391304];

x1_step2.ymin = -1;

% Layer 1

b1 = [-2.6286089165606161266;5.3761324574102120621;-
3.6882513100206524292;-1.8141467132939435025;-
2.3620857189680961596;-
1.1439925515563174052;0.6967135501420914645;1.20708135388183746;0.50
403155000398514041;-
0.15473919713172781809;1.0602068532352590946;0.73340626450809509596;
-0.95875934245215743701;1.2575702678305218374;-
1.3907864078846117195;1.2637390605534097876;1.1358231435279466659;2.
080022276692760741;-2.6866032990773374145;-2.2666530723135553949];

IW1_1 = [1.4516201449148500302 -0.53040845306462314213
0.51187280067873042189 0.84885545114857907656 -
1.5723684740880543576;-4.1837892500948283825 0.4563704495623666646 -
0.47548143136223053329 -0.71427695254364020894 -
0.10007720390005410416;0.57430239241276348494 -3.6294746586701327473

```

-0.30222029726030746888 0.31520744881458406805 -
0.065374945937600487111;4.0816426929620170938 -2.212618550886160218
-1.6087749720850790247 0.63594980374218423869
0.088982353032175900975;4.0421955650857315234 -0.3520174116446841972
3.4923068569205750755 -1.2099479111400148756 -
0.2196558571409130689;0.91619512219238097295 3.1884302113786966437 -
1.0848190192650999375 0.010729090688608155027
0.10113443428801693891;-2.075209764043845162 -0.42016430713284541598
2.2872302725015072333 1.3790927884439161399 0.18910491376155716403;-
0.074157075234098118743 3.1742368019954567693 -1.90692876259885713 -
0.21975926841499554554 0.065824280618951375166;-
3.0397323656054209273 -3.8321710404457411947 -1.491133460716593806 -
0.048613024363022651342 -0.13188455739918569787;-
1.362365049478106771 0.68509890585781829664 1.5166412187988993487
1.1378459692814368864 -1.0313305180004277428;-5.8564885669690562864
2.3988115306833215357 0.58810070786483958916 0.099149281318826304465
0.10830154800741238763;1.2744610959384523596 0.27529059205223938012
1.0104860197641560848 -1.6133344091813572874 1.887046023254124183;-
0.74903601502878647267 -0.77971722085037709071 1.1705004318478278158
-1.7166055925076511013 -0.22772217545133219274;1.6119799443723050736
-0.21730857338508946519 0.12353916742264327289 -
1.1891383210118124936 -0.18266825254513377463;-3.2131418422279591773
-1.078150244049873141 -0.86342416239846209169
0.040322057896692374557 -
0.026934598973963046198;1.510923676410362182 -0.4606436456429369164
0.20101928887771985943 -1.2146539471568211788 -
0.19187261356231816167;0.38555188007693474983 -1.948046474234896408
0.43699731644395306551 -0.058980633518206489196 -
0.070209374239879246415;0.29572738530057740558 -
2.1340454079477639837 0.7724022651647205473 -0.67197653959627312581
1.0950723930478700741;-0.54347742677075672191 -
0.41936884321509665385 2.2392976828597186412 -0.44578303824572773628
-0.55365709543370056345;-1.03095936678156197 -0.15267039463918211095
2.9253449882983706409 0.40859266210928069096
0.56074300325438442005];

```

% Layer 2

b2 = -1.3851140345150896316;

LW2_1 = [-0.0059927370050254433198 2.8439087286560815748
1.8892395614370767198 0.071509898182547118606 2.3745231553736587671
0.40597989146271790117 0.31421994349433529647 3.2502380335784293308
-0.7443466015761540655 -0.0038852076913098100748 -
0.55131605113744019953 0.0025789287637675802935 -
0.029600953260976753389 1.1095853551392889091 0.89481034578939500079
-0.77953183541132509138 0.61382014762933523322 -
0.0038497651409178574623 0.051130755949707819286
0.15659989812985289359];

% Output 1

y1_step1.ymin = -1;

y1_step1.gain = 13.2087187191995;

y1_step1.xoffset = 0.564646344173078;

% ===== SIMULATION =====

% Dimensions

Q = size(x1,1); % samples

% Input 1

x1 = x1';

xp1 = removeconstantrows_apply(x1,x1_step1);

```

```

xp1 = mapminmax_apply(xp1,x1_step2);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);

y1 = y1';

end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);

end

```

```

% Remove Constants Input Processing Function

function y = removeconstantrows_apply(x, settings)

y = x(settings.keep, :);

end

% Sigmoid Symmetric Transfer Function

function a = tansig_apply(n, ~)

a = 2 ./ (1 + exp(-2*n)) - 1;

end

% Map Minimum and Maximum Output Reverse-Processing Function

function x = mapminmax_reverse(y, settings)

x = bsxfun(@minus, y, settings.ymin);

x = bsxfun(@rdivide, x, settings.gain);

x = bsxfun(@plus, x, settings.xoffset);

end

```

ANEXO 10: CREACION DE LA RED NEURONAL ARTIFICIAL PARA I.1.

	1	2	3	4	5	6
1	1	17	3	0	16	1
2	1	17	3	0	17	1
3	1	17	3	0	18	0
4	1	17	3	0	19	0
5	1	17	3	0	20	0
6	1	17	3	0	21	0
7	1	17	3	0	22	0
8	1	17	3	0	23	0
9	1	18	4	0	0	0
10	1	18	4	0	1	0
11	1	18	4	0	2	0
12	1	18	4	0	3	0
13	1	18	4	0	4	0
14	1	18	4	0	5	0

Figura 41. Datos de entrada de la RNA de I.1

 **Select Data**
 What inputs and targets define your fitting problem?

Get Data from Workspace

Input data to present to the network.

Inputs:

Target data defining desired network output.

Targets:

Samples are: Matrix columns Matrix rows

Figura 42. Datos de ingreso de la RNA de I.1.

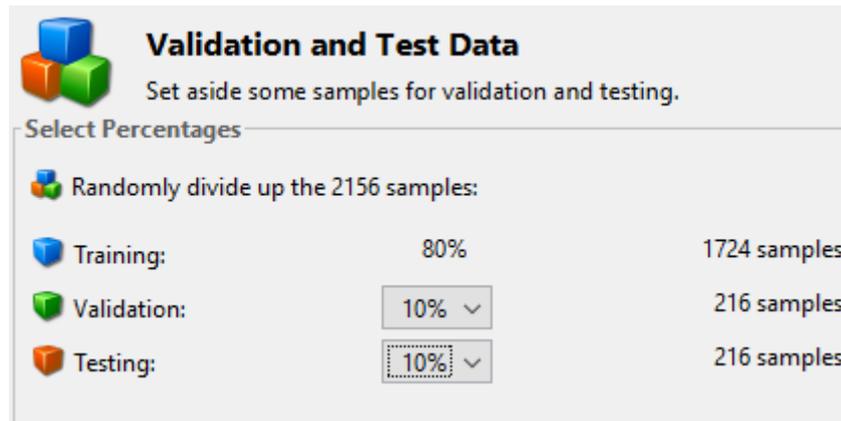


Figura 43. Configuración de porcentajes para pruebas y validaciones para I.1.

RED NEURONAL ARTIFICIAL

```
function [y1] = redneuralIL(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Autogenerated by MATLAB, 07-Jul-2023 00:14:15.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = Qx8 matrix, input #1
% and returns:
%   y = Qx1 matrix, output #1
% where Q is the number of samples.
%#ok<*RPMT0>
% ===== NEURAL NETWORK CONSTANTS =====
% Input 1
```

```

x1_step1.keep = [2 3 4 5 6 7];

x1_step2.xoffset = [1;1;0;1;0;0];

x1_step2.gain =
[0.5;0.0689655172413793;2;0.333333333333333;0.0869565217391304
;2];

x1_step2.ymin = -1;

% Layer 1

b1 = [-2.3343351292180081202;1.3122926865770363491;-
6.7270054055924797254;3.318184443375407966;-
0.72509818904218170665;0.49025084301998511638;-
0.68299343668348400094;-0.50270661459421073225;-
0.229604184031735159;1.2508869870031360527;0.38482717283851197
72;0.10064834132520594778;-
1.9030821571778233636;0.067752291101740996782;0.28516709552140
007133;-0.3690755627840133779;-1.3959582570524162737;-
1.1305632820101203517;-1.3779760080749123841;-
0.36759705317070523467;0.30587233549993930737;3.29043831026385
72467;2.589582389238612059;-
1.565091294871031824;4.3278366498456666278];

IW1_1 = [-1.980566554361606002 0.85833489324496714534 -
0.5112925381903303812 2.7557495555115241181
1.4066221211519471801 -0.23399624449807382431;-
3.7183577580094895332 -0.81772712887296739037
0.14648069397557197946 4.6350157194830670093 -
4.4415770239799776675 -
2.6339508538197353715;0.030417859895444918084 -
2.1174828107704786184 -3.8364779750292279381 -
0.18402823535552190193 -0.00090054640727507239816 -
3.1277006941317870847;-2.0787478307369822161 -
0.86478496710154395544 0.79908443982498222891
0.38000541030488438343 0.0050327511534424811818 -
2.5236229571962320861;2.3799053712990709819
0.43480603926308292184 1.9820506180959844844 -
0.3120897719387387137 -0.010925233617064745995 -
0.40679831331376226089;-3.1343488145342561779 -
1.4379880781135634393 0.3894008866792200152
0.61850364897276577381 0.0098598663770100899301
0.78126207525010282762;-0.70890208249013064723 -
3.068910049496138992 1.3813546738208462017 -
0.31195332695198896289 0.0058284575080114874729
2.4708646801285589767;-3.2652660816209642114 -
1.2374309479391705224 -2.2784346662941827866
0.42641540124090326902 0.0093399578410245029381 -
2.2032454572952322458;-0.37043939150402654947
0.69846928287528353696 2.917573723650014994

```

0.029955824309158857188 0.010318423332916590307 -
2.0933249210192967915;2.9851916747687412546
3.4555862125806515728 -2.7686754183361226289
1.5464116496388711219 -2.3542566880833275533 -
0.54772724654752102946;0.58778662921168822741
4.4505832617325404854 -0.13577702219410750506 -
2.3860499228147502571 -0.26780623601074643458
0.85124687013473432362;2.9733964530729917186 -
0.077592655518150102645 0.12989922156401428888 -
2.6334596699004761966 -2.9432749980164722992 -
3.3969643674715297443;-2.7271945240888033624
2.8912425541036776977 -0.70196004694887825792 -
2.974255111707256205 -0.070723642403297204417 -
3.492363814128392363;0.90100111723826759036
1.2186147013483215051 -3.0189307433194882968
2.7997987727902624577 0.15905629036474547289 -
2.518348815883601155;-0.23951634944060559707 -
1.1738167264221779362 0.87984944430207379096 -
2.7872574296806273964 -0.17670972261412548621 -
0.044905284772720009678;0.75595199146257285872
1.2796914560688785745 2.3102289984737778283
2.853153248874407133 0.19050077245321017649 -
3.2709736646836069518;3.0916031671461614572
1.4991942568323681773 -4.7777874814471470444 -
0.66618193096562228206 -0.00067874035697276205621
2.6746411861150987122;6.6240935193995857944 -
3.5440382269439965057 3.1818893364713489724 -
0.57500870466505038525 -0.066418316049739273277
4.5188854296575717839;-1.0253614223396159133
1.854643644734773833 -0.25388283782907161701
0.16257463403544383795 0.013662281853899388775
0.88655399855076866267;-1.5672997695623882564
1.1368855144270388902 -2.9071566656376743865
1.6173506713899197251 0.37279043633071651254 -
0.19567420503393717635;-0.027368528586290741256 -
1.1844514721669932023 0.92441011571791353862 -
2.8663994965039609752 -0.17623194879248677003 -
0.065436034813077123551;2.6161092821657807583 -
2.9547008039797764134 2.0044502068987419818
3.0642366013207316122 0.07238628227363176626
3.5995560462725300077;5.0084928846072021003 -
0.33065006039323224796 0.63873134932535213792
2.9090758907122769017 0.22716718575419928006 -
1.8168429473980916011;0.067241537339500459525
2.1638084129563157632 -1.2494507354230488527
0.21813370279817581587 -0.0051789997659034752928 -
0.016459367345243298281;3.3027970221427320752 -
0.32724152361677255607 2.1230737055301474037 -
2.6763212814366754699 2.0859416882089836243
0.66351804791730051303];

```

% Layer 2

b2 = 5.7639788103639340022;

LW2_1 = [0.032890085252558702578 0.008259718636817830334
2.5270220179912477043 -2.4592552729782233278
1.2010447069843821488 0.79332037009919276205
1.1104555750844986584 1.3737104700608626651
3.1765958117997890042 0.011538514252858358966 -
0.023296856511066939971 0.011636174072372919286 -
1.0286109208762301837 -0.80619293893032328846 -
2.7847653768219808867 -0.79895025645954020899
0.8918380401855368067 -0.23343439820997211043 -
1.1534016138856233447 0.074831771249103229415
1.9834282379329015722 -0.98870337218382287059
0.040239795966105489089 2.0848645769882838685 -
0.026350915181281565208];

% Output 1

y1_step1.ymin = -1;

y1_step1.gain = 0.037297913262969;

y1_step1.xoffset = 35.5223242711721;

% ===== SIMULATION =====

% Dimensions

Q = size(x1,1); % samples

% Input 1

x1 = x1';

xp1 = removeconstantrows_apply(x1,x1_step1);

xp1 = mapminmax_apply(xp1,x1_step2);

```

```

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
y1 = y1';
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Remove Constants Input Processing Function
function y = removeconstantrows_apply(x,settings)
y = x(settings.keep,:);
end

% Sigmoid Symmetric Transfer Function

```

```
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings)
x = bsxfun(@minus, y, settings.ymin);
x = bsxfun(@rdivide, x, settings.gain);
x = bsxfun(@plus, x, settings.xoffset);
end
```

ANEXO 11: DISEÑO DE GRÁFICAS DE PREDICCIÓN DE POTENCIA ELÉCTRICA

ANEXO 12: PLIEGO TARIFARIO

Tabla 10. Rubro de consumo de energía.

	DICIEMBRE -MAYO	
RANGO DE CONSUMO	Energía kWh/usd	USD
0-50	0,091	4,55
51-100	0,093	4,65
101-150	0,095	4,75
151-200	0,097	4,85
201-250	0,099	4,95
251-300	0,101	5,05
301-350	0,103	5,15
351-400	0,105	5,25
401-450	0,105	5,25
451-500	0,105	5,25
501-700	0,105	21

701-1000	0,145	7,54
	Total E.	78,24

ANEXO 13: FACTURACIÓN DEL SERVICIO ELÉCTRICO.

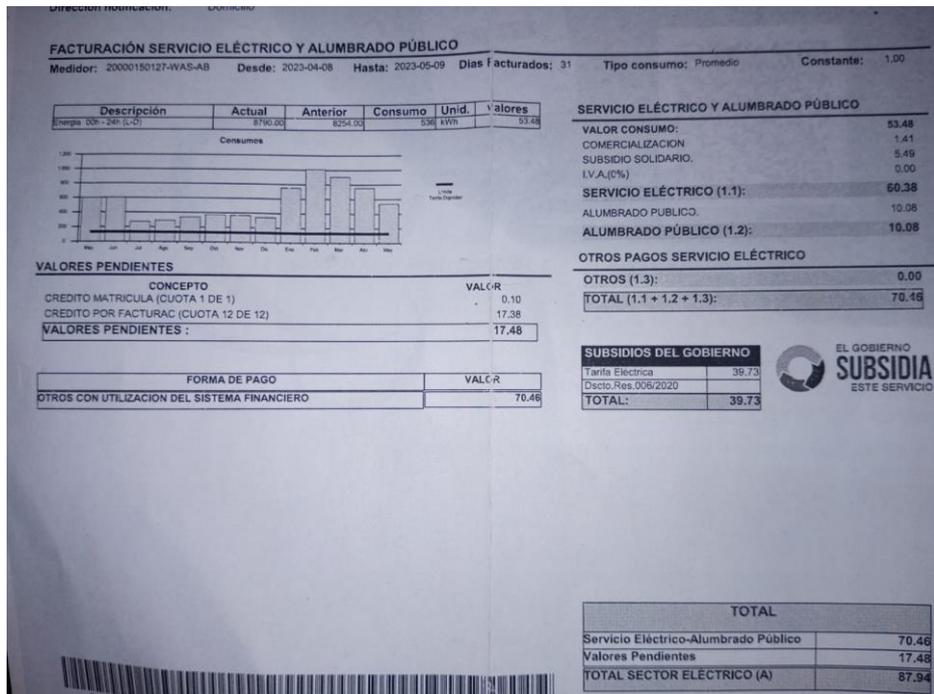


Figura 44. Facturación de la residencia del mes de abril a mayo.

ANEXO 14: IMPLEMENTACIÓN DE EQUIPO DE ADQUISICIÓN DE DATOS.

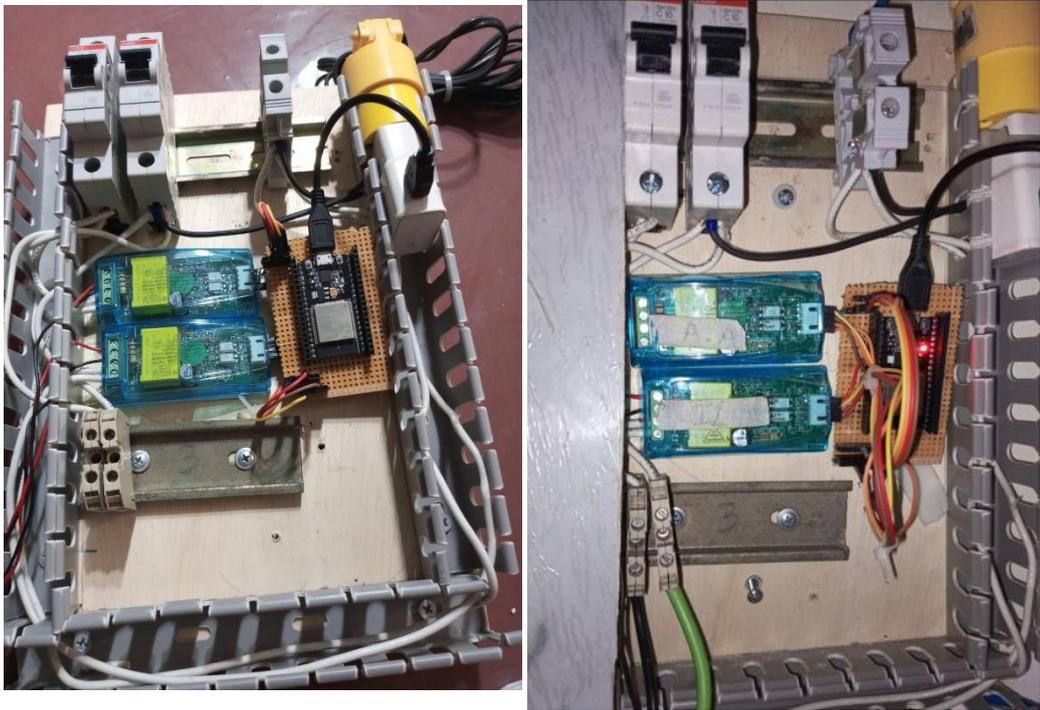


Figura 45. Implementación de equipos.



Figura 46. Medición de amperaje de iluminación.

<input type="checkbox"/>	pzem_current float pzem_current;	0.197	21 Jul 2023 00:39:05	⋮
<input type="checkbox"/>	pzem_current2 float pzem_current2;	1.327	21 Jul 2023 00:39:05	⋮
<input type="checkbox"/>	pzem_poten float pzem_poten;	3.7	21 Jul 2023 00:39:05	⋮
<input type="checkbox"/>	pzem_poten2 float pzem_poten2;	117.8	21 Jul 2023 00:39:05	⋮
<input type="checkbox"/>	pzem_voltaje float pzem_voltaje;	122.7	21 Jul 2023 00:39:05	⋮

Figura 47. Visualización de datos de energía de los artefactos eléctricos.

ANEXO 15: PREDICCIÓN DE CONSUMO Y COSTO DE ARTEFACTOS ELECTRICOS.

PREDICCIÓN DE CONSUMO Y COSTO MENSUAL DE ARTEFACTOS ELÉCTRICOS

FECHA ▾

<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p>POTENCIA 265</p> <p>CAPACIDAD 500</p>  <p>CONSUMO 46.17</p> <p>COSTO \$ 4.29</p> </div>	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p>POTENCIA 822</p> <p>AREA 35</p>  <p>CONSUMO 289.95</p> <p>COSTO \$ 27.88</p> </div>
<div style="border: 1px solid #ccc; padding: 5px;"> <p>POTENCIA 200</p>  <p>CONSUMO 54.24</p> <p>COSTO \$ 5.04</p> </div>	<div style="border: 1px solid #ccc; padding: 5px;"> <p style="text-align: center;">CONSUMO TOTAL DE LA VIVIENDA</p> <p>CONSUMO TOTAL <input type="text" value="390.36"/> <input type="button" value="CALCULAR"/></p> <p>COSTO TOTAL \$ <input type="text" value="38.2788"/> <input type="button" value="CLEAR"/></p> </div>

INICIO

MONITOREO

REGISTROS Y ALARMAS

Figura 48. Predicción del consumo y costo de energía del mes de enero.



Figura 49. Predicción del consumo y costo de energía del mes de febrero.



Figura 50. Predicción del consumo y costo de energía del mes de marzo.



Figura 51. Predicción del consumo y costo de energía del mes de abril.



Figura 52. Predicción del consumo y costo de energía del mes de mayo.