

UNIVERSIDAD SANTO TOMÁS

Gestión de intermitencia en la generación fotovoltaica mediante microrred con mecanismo Power to Heat y DERs del Laboratorio de Energías Renovables de la USTA

Realizado por

Santiago Beltrán Salgado

Glenda Victoria Sánchez Torres

Proyecto presentado en cumplimiento del requisito parcial
para optar por el grado de Ingeniería Electrónica



Grupo de Investigación MEM (Modelado-Electrónica-Monitoreo)
Facultad de Ingeniería Electrónica
División de Ingenierías

Diciembre de 2025

**Gestión de intermitencia en la generación fotovoltaica
mediante microrred con mecanismo Power to Heat y
DERs del Laboratorio de Energías Renovables de la USTA**

Realizado por

**Santiago Beltrán Salgado
Glenda Victoria Sánchez Torres**

Proyecto enviado en cumplimiento del requisito parcial
para optar por el grado de Ingeniería Electrónica

Dirigido por

Ing. José Luis Paternina Durán Ph.D (c).

Codirigido por

**Ing. Carlos Javier Mojica Casallas
Ing. Óscar Mauricio Gélvez Lizarazo**

Grupo de Investigación MEM (Modelado-Electrónica-Monitoreo)
Sistemas de Energía y Automatización
Facultad de Ingeniería Electrónica
División de Ingenierías

Diciembre de 2025

Autoridades de la Universidad

RECTOR GENERAL

R.P. FRAY ÁLVARO JOSÉ ARANGO RESTREPO, O.P

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL

R.P. FRAY HERNÁN YESID RIVERA ROBERTO, O.P

VICERRECTOR ACADÉMICO GENERAL

R.P. FRAY MAURICIO ANTONIO CORTÉS GALLEGO, O.P

SECRETARIO GENERAL

Dra. LUCERO GALVIS CANO

DECANO DIVISIÓN DE INGENIERÍAS

R.P FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P

SECRETARIA DE DIVISIÓN

E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

ING. LEONARD FABIO YÉPES ARBELÁEZ, PhD.

Nota de aceptación

Firma del tutor

Firma del jurado

Firma del jurado

BOGOTÁ D.C. _____ DE 2025

Advertencia

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

Dedicatoria

Dedico este proyecto a mis padres, por inculcarme los valores que me definen, por guiarme con su ejemplo y por hacer posible este logro. A mi padre, Mario, quien, aunque hoy no esté presente, me inspiró a seguir esta carrera y dejó en mí enseñanzas que marcaron mi vida; a mi madre, Mirtian, y a mi hermana, Melisa, por su apoyo incondicional, paciencia y por acompañarme en cada desafío de mi vida; y a mi novio, Christian, por su paciencia, su compañía constante y por enseñarme, con su ejemplo, a perseverar y superar cada obstáculo en este camino.

Victoria Sánchez

Dedico este proyecto a mis padres, a mi hermano y a mi abuela, por estar siempre conmigo y apoyarme durante todo este proceso de formación. Gracias por sus consejos, por las charlas y por motivarme a seguir adelante. También agradezco a mis profesores por su acompañamiento, al director José Luis Paternina, por su guía y orientación a lo largo de este camino académico.

Santiago Beltrán

Agradecimientos

Agradezco a la vida por darme la fuerza y la perseverancia para superar los obstáculos, así como la oportunidad de avanzar a lo largo de este camino. Pero, sobre todo, agradezco a las personas que me rodean: a mi madre y a mi hermana, por escucharme, motivarme y apoyarme incondicionalmente; a mi padre, por ser una guía fundamental en mi formación, sin cuya ayuda no habría podido estudiar ni culminar esta carrera; a mi novio, por su paciencia y acompañamiento constante; a mis compañeros de semestre de Ingeniería Electrónica, quienes me enseñaron a trabajar en equipo; y a los profesores de la facultad, por transmitirme conocimientos fundamentales y valores importantes. De manera especial, agradezco a mi director, José Luis Paternina, por guiarme, y a mi amigo y compañero Santiago Beltrán, con quien logramos sacar este proyecto adelante apoyándonos mutuamente.

Victoria Sánchez

Agradezco a mis padres por su apoyo incondicional, por sus palabras y por no dejarme rendir cuando las cosas se ponían difíciles. A mi hermano, gracias por su motivación constante y por recordarme siempre por qué vale la pena seguir. A mi abuela, una de las razones más importantes de quien soy hoy como persona, gracias por todo su amor y ejemplo. También agradezco a los docentes que me brindaron las bases para seguir creciendo como ingeniero, a mis compañeros que me acompañaron en este proceso y a Victoria Sánchez, quien ha estado conmigo desde el inicio y comparte conmigo la historia de este proyecto de grado

Santiago Beltrán

Índice general

Resumen	1
Abstract	2
List of Figures	2
List of Tables	5
Glosario	7
1. Introducción	9
1.1. Planteamiento del problema	9
1.2. Objetivos	10
1.2.1. Objetivo general	10
1.2.2. Objetivos específicos	10
1.3. Justificación	11
1.4. Impacto social	12
2. Estado del arte	13
2.1. Flexibilidad en las redes de distribución híbridas	13
2.2. Innovación Energética: El Impacto de la Tecnología PtH en la Eficiencia y Sostenibilidad	14
2.3. Gestión energética y flexibilidad en la demanda	14
2.4. Integración y Operación de Microrredes: Modelos, elementos y Desafíos	15
2.5. Tabla resumen estado del arte	16
3. Marco Teórico	20
3.1. Flexibilidad en las FER	20
3.1.1. Métodos Cuantificables	20
3.1.1.1. Método Implícito	20
3.1.1.2. Método Explícito	20
3.1.2. Servicios Manuales	21
3.1.2.1. Reserva de Restauración de Frecuencia (mFRR)	21
3.1.3. Métodos de modelado	21
3.1.3.1. Modelo Matemático para la Maximización del Valor de Flexibilidad mediante mFRR	21
3.1.3.2. Cargas Controladas Termostáticamente (TCL)	22

3.2.	Indicadores de flexibilidad	24
3.3.	Almacenamiento de energía (inercia térmica):	26
3.4.	Modelo de un sistema virtual de almacenamiento de energía	26
4.	Diseño Metodológico	28
4.1.	Revisión bibliográfica de las técnicas de modelado matemático	29
4.2.	Modelo matemático del horno	29
4.3.	Implementación de una microrred mediante la integración del horno.	30
4.4.	Validación del desempeño de la microrred propuesta	30
5.	Desarrollo Conceptual	32
5.1.	Fase 1: Búsqueda bibliográfica en bases de datos científicas sobre el modelado de tecnologías PtH	32
5.1.1.	Clasificación de las técnicas de modelado aplicadas a tecnologías Power-to-Heat (PtH)	32
5.1.1.1.	Los Modelos Físicos o Fenomenológicos	32
5.1.1.2.	Modelos Estadísticos o Empíricos.	33
5.1.1.3.	Modelos Basados en Machine Learning	34
5.1.2.	Caracterización del Horno	34
5.2.	Fase 2: Modelo matemático del horno	35
5.2.1.	Desarrollo del plan experimental	35
5.2.1.1.	Circuito de acondicionamiento	36
5.2.1.2.	Ecuación linealización del sensor PT100	37
5.2.1.3.	Caracterización de Dispositivos y Equipo de Medición	41
5.2.1.4.	Comunicación Modbus RTU	46
5.2.1.5.	Funcionamiento del sistema de medición	50
5.2.2.	Pruebas Experimentales	50
5.2.2.1.	Estudio de Datos y Ajuste según los Modelos Matemáticos	51
5.2.2.2.	Ecuación del Modelo Matemático	52
5.2.2.3.	Validación del Modelo con los Datos Experimentales	52
5.3.	Fase: Implementación de una Microrred Mediante la Integración del Horno	54
5.3.1.	Cargas Disponibles en el Laboratorio de Energías	54
5.3.2.	Conexiones Cargas	54
5.3.3.	Integración Microrred	56
5.3.3.1.	Recursos Energéticos Distribuidos (DER)	56
5.3.3.2.	Configuración Eléctrica de la Microrred	57
5.3.3.3.	Comunicación Growatt MIC 1500TL-X	58
5.3.3.4.	Integración del inversor fotovoltaico y monitoreo completo de la microrred	59
5.3.3.5.	Conexiones física y Eléctrica	60
5.3.3.6.	Tablero Eléctrico	61
5.3.3.7.	Cableado y Protecciones	62
5.3.3.8.	Cableado DC	63
5.3.4.	Montaje Experimental	63
5.3.4.1.	Operación de la Microrred	71
5.3.5.	Comportamiento Variación de Demanda y Generación	71

5.3.6. Aprovechamiento Energía Fotovoltaica	72
5.4. Validación del Desempeño de la Microrred	72
5.4.1. Indicadores de Flexibilidad	72
5.4.2. Emulación Almacenamiento Virtual de Energía	73
6. Resultados y Discusión	77
6.1. Caso 1	78
6.2. Caso 2	81
6.3. Caso 3	83
6.4. Indicadores de Flexibilidad	86
6.4.1. Energía Exportada y Suministrada por la Red	88
7. Conclusiones y Trabajos Futuros	92
7.1. Conclusiones	92
7.2. Trabajos Futuros	93
8. Anexos	94
8.1. Caracterización Térmica del Horno	94
8.1.1. Nivel 1: 104,40 °C	95
8.1.2. Nivel 2: 117,59 °C	95
8.1.3. Nivel 3: 142,09 °C	95
8.1.4. Nivel 4: 163,37 °C	96
8.1.5. Nivel 5: 183,09°C	96
8.1.6. Nivel 6: 205,26°C	96
8.2. Código fuente del sistema de medición en Raspberry Pi Pico	97
8.3. Código Validación Modelo del Horno	107
8.4. Código Monitoreo Microrred	109
8.5. Código Inercia Termica	130
8.6. Resultados Microrred	137
8.6.1. Resultados Caso 1.	138
8.6.2. Resultados Caso 2.	139
8.6.3. Resultados Caso 3.	140
8.7. Datos experimentales de la potencia de red	141
8.7.1. Caso 1	141
8.7.2. Caso 2	142
8.7.3. Caso 3	142

Resumen

En el presente documento se implementa una microrred experimental orientada a la gestión de la energía solar fotovoltaica mediante mecanismos Power-to-Heat (PtH) y el uso de fuentes distribuidas de energía (DERs) disponibles en el Laboratorio de Energías Renovables de la Universidad Santo Tomás. El propósito de este trabajo es diseñar un sistema que permita aprovechar la generación fotovoltaica variable a través del acoplamiento con cargas térmicas controladas, contribuyendo a la flexibilidad y eficiencia del sistema energético.

En el desarrollo de la investigación se identifican las principales técnicas de modelado matemático empleadas para la caracterización de tecnologías PtH, con base en una revisión bibliográfica especializada. Posteriormente, se selecciona y valida el modelo matemático del horno eléctrico disponible en el laboratorio, a partir de pruebas experimentales que permiten representar su comportamiento térmico y su capacidad de almacenamiento de energía.

Finalmente, se implementa una microrred que integra el horno como carga térmica flexible para la gestión de la generación fotovoltaica, junto con un conjunto de paneles solares, un inversor, cargas eléctricas complementarias y un analizador de redes. La comunicación entre dispositivos se realiza mediante el protocolo Modbus RS-485, gestionada por un sistema embebido basado en Raspberry Pi Pico 2.

Los resultados obtenidos demuestran que la aplicación del mecanismo Power-to-Heat permite un mejor aprovechamiento de la energía fotovoltaica disponible, reduciendo la dependencia de la red eléctrica y mejorando los índices de autoconsumo y autosuficiencia. Asimismo, se evidencia que la correcta programación térmica del horno contribuye a mitigar la intermitencia de la generación solar, haciendo un mejor uso del sistema.

Palabras clave: Power-to-Heat, microrred, gestión energética, intermitencia fotovoltaica, flexibilidad térmica, DERs.

Abstract

This document presents the implementation of an experimental microgrid aimed at managing solar photovoltaic energy through Power-to-Heat (PtH) mechanisms and the use of distributed energy resources (DERs) available in the Renewable Energy Laboratory at Universidad Santo Tomás. The purpose of this work is to design a system that enables the utilization of variable photovoltaic generation through coupling with controlled thermal loads, contributing to the flexibility and efficiency of the energy system.

In the development of this research, the main mathematical modeling techniques used for the characterization of PtH technologies are identified based on a specialized literature review. Subsequently, the mathematical model of the electric furnace available in the laboratory is selected and validated through experimental tests, allowing the representation of its thermal behavior and energy storage capacity.

Finally, a microgrid is implemented integrating the furnace as a flexible thermal load for the management of photovoltaic generation, together with a set of solar panels, an inverter, complementary electrical loads, and a network analyzer. Communication between devices is carried out through the Modbus RS-485 protocol, managed by an embedded system based on the Raspberry Pi Pico 2.

The results demonstrate that the application of the Power-to-Heat mechanism allows for better utilization of the available photovoltaic energy, reducing dependence on the electrical grid and improving self-consumption and self-sufficiency indices. Likewise, it is shown that the proper thermal programming of the furnace contributes to mitigating solar generation intermittency, leading to a more efficient use of the system.

Keywords: Power-to-Heat, microgrid, energy management, photovoltaic intermittency, thermal flexibility, DERs.

Lista de Figuras

1.	Diagrama de la metodología planteada para el desarrollo del proyecto. Tomado de: Autores.	28
2.	Circuito de acondicionamiento de la PT100. Tomado de: Autores.	37
3.	Comportamiento térmico del horno durante el proceso de calentamiento y enfriamiento. Tomado de: Autores.	38
4.	Comparación entre los valores teóricos simulados y los valores experimentales. Tomado de: Autores.	39
5.	Pines Raspberry Pi Pico 2. Tomado de: [34].	42
6.	Implementación del protocolo Modbus RTU entre el microcontrolador y el analizador de redes. Tomado de: [35].	47
7.	Conexión del transformador de corriente MBS KBU con el analizador de redes CIRCUTOR CVM-C10. Tomado de: [36].	48
8.	Circuito de conexión entre la Raspberry Pi Pico, el sensor DHT11, el amplificador LM324 y los módulo RS485 para lectura del CVM-C10 e Inversor. Tomado de: Autores.	49
9.	Relación entre la potencia promedio y la variación de temperatura (ΔT) del horno. Tomado de: Autores.	51
10.	Comparación entre los modelos de los datos experimentales del horno. Tomado de: Autores.	53
11.	Diagrama de conexión de las cargas utilizadas en las pruebas individuales. Tomado de: Autores.	55
12.	Diagrama de conexión para la prueba con varias cargas. Tomado de: Autores.	56
13.	Diagrama Microrred. Tomado de: Autores.	57
14.	Conexión cable de señales MIC 1500TL-X. Tomado de: [40].	58
15.	Esquemático de potencia de la microrred. Tomado de: Autores.	60
16.	Distribución del tablero de potencia. Tomado de: [41].	61
17.	Tablero de combinación de los módulos fotovoltaicos. Fuente: Autores.	64
18.	Inversor fotovoltaico alimentado mediante transformador elevador de 110 V a 220 V. Fuente: Autores.	65
19.	Integración del tablero fotovoltaico, transformador e inversor sincronizado con la red (indicador verde). Fuente: Autores.	66
20.	Estado del inversor sin sincronización por falta de tensión o irradiancia (indicador rojo). Fuente: Autores.	67
21.	Tablero general de protección y distribución de la microrred. Fuente: Autores.	67
22.	Transformador reductor de 220 V a 110 V para alimentación de cargas. Fuente: Autores.	68

23.	Horno eléctrico utilizado como carga variable tipo PTH. Fuente: Autores.	69
24.	Luminarias resistivas de 9 W, 20 W y 100 W empleadas como cargas estáticas. Fuente: Autores.	69
25.	Clavija utilizada para la conexión de las cargas al tablero. Fuente: Autores.	70
26.	Sistema de comunicación RS-485 y adquisición de datos con Raspberry Pi Pico 2. Fuente: Autores.	70
27.	Gráfica de demanda y generación. Tomado de: Autores.	71
28.	Curvas experimentales y polinómicas de calentamiento del horno. Tomado de: Autores.	74
29.	Curvas experimentales y polinómicas de enfriamiento del horno. Tomado de: Autores.	75
30.	Gráfica de la temperatura del horno para el caso 1. Tomado de: Autores.	79
31.	Gráfica de la potencia del horno y la potencia de los paneles para el caso 1. Tomado de: Autores.	80
32.	Evolución de la temperatura del horno para el caso 2. Tomado de: Autores.	82
33.	Potencia promedio del horno y de los paneles solares para el caso 2. Tomado de: Autores.	83
34.	Evolución de la temperatura del horno para el caso 3. Tomado de: Autores.	84
35.	Potencia promedio del horno y de los paneles solares para el caso 3. Tomado de: Autores.	85
36.	Comportamiento horario de la energía aportada y exportada en el Caso 1 (Horno + luminaria de 9 W). Tomado de: Autores.	89
37.	Comportamiento horario de la energía aportada y exportada en el Caso 2 (Horno + luminaria de 20 W). Tomado de: Autores.	90
38.	Comportamiento horario de la energía aportada y exportada en el Caso 3 (Horno + luminaria de 100 W). Tomado de: Autores.	91
39.	Comparación del balance energético total entre los tres casos analizados. Tomado de: Autores.	91

Lista de Tablas

1.	Tabla resumen estado del arte	17
2.	Parámetros del horno TO1705SG-LA (Black+Decker). Tomado de: Autores [32].	35
3.	Características técnicas de la Raspberry Pi Pico 2. Tomado de: [33]	41
4.	Parámetros de alimentación del Circutor CVM-C10. Tomado de: [35]	43
5.	Parámetros de medición de voltaje del Analizador de redes CVM-C10. Tomado de: [35]	43
6.	Parámetros de medición de corriente del Analizador de redes CVM-C10. Tomado de: [35]	43
7.	Datos técnicos del transformador de corriente MBS KBU. Tomado de: [36]	44
8.	Especificaciones del módulo RS-485. Tomado de: [37]	44
9.	Especificaciones del sensor de humedad y temperatura. Tomado de: [38]	44
10.	Datos eléctricos a Condiciones de Prueba Estándar (STC). Tomado de: [39]	45
11.	Datos eléctricos a NMOT Tomado de: [39]	45
12.	Parámetros de entrada (DC) Tomado de: [40]	45
13.	Datos de salida (AC) del inversor MIC 1500TL-X Tomado de: [40]	46
14.	Especificaciones del transformador elevador de tensión. Tomado de: Autores	46
15.	Clasificación de las cargas existentes en el Laboratorio de Energías Tomado de: Autores.	54
16.	Tabla de tiempos para el caso 1. Tomado de: Autores.	79
17.	Tiempos y temperaturas objetivo para el caso 2. Tomado de: Autores.	81
18.	Tiempos y temperaturas objetivo para el caso 3. Tomado de: Autores.	84
19.	Relación entre la energía, potencia y variación de temperatura del horno en el nivel 1 (104,4 °C).	95
20.	Relación entre la energía, potencia y variación de temperatura del horno en el nivel 2 (117,59 °C).	95
21.	Relación entre la energía, potencia y variación de temperatura del horno en el nivel 3 (142,09 °C).	95
22.	Relación entre la energía, potencia y variación de temperatura del horno en el nivel 4 (163,37 °C).	96
23.	Relación entre la energía, potencia y variación de temperatura del horno en el nivel 5 (183,09 °C).	96
24.	Relación entre la energía, potencia y variación de temperatura del horno en el nivel 6 (205,26 °C).	96
25.	Resultados experimentales Caso 1.	138

26.	Resultados experimentales Caso 2.	139
27.	Resultados experimentales Caso 3.	140
28.	Potencias promedio para el caso 1. Tomado de: Autores.	141
29.	Potencias promedio para el caso 2. Tomado de: Autores.	142
30.	Potencias promedio para el caso 3. Tomado de: Autores.	142

Glosario

Power-to-X (PtX): Conjunto de tecnologías que convierten la energía eléctrica, principalmente renovable, en otros portadores energéticos o productos útiles, como calor, hidrógeno o combustibles sintéticos, facilitando el almacenamiento y la flexibilidad energética.

Recursos Energéticos Distribuidos (DER): Sistemas de generación o almacenamiento de energía instalados cerca de los usuarios finales, que funcionan de manera descentralizada y complementan al sistema eléctrico convencional.

Intermitencia: Variación o discontinuidad en la generación de energía renovable, especialmente solar o eólica, debido a cambios en las condiciones ambientales, lo que afecta la disponibilidad constante de electricidad.

Fuentes de Energía Renovable (FER): Recursos naturales inagotables, como el sol, el viento o el agua, utilizados para generar energía de forma sostenible y con bajas emisiones contaminantes.

Cargas Térmicas Controladas (TCL): Dispositivos que almacenan o consumen calor (como calentadores o bombas de calor) cuya operación puede ajustarse para gestionar la demanda energética.

Flujo de potencia óptimo (OPF): Problema de optimización que busca la mejor distribución de potencias en un sistema eléctrico cumpliendo restricciones técnicas y minimizando costos o pérdidas.

Power-to-Heat: Tecnología que convierte energía eléctrica, generalmente proveniente de fuentes renovables, en energía térmica útil. Esta conversión se realiza mediante dispositivos como resistencias eléctricas o bombas de calor, y permite utilizar excedentes de electricidad para calefacción, agua caliente o procesos industriales, contribuyendo al equilibrio y flexibilidad del sistema energético.

Power-to-Gas: Tecnología que convierte excedentes eléctricos (por ejemplo, de fuentes renovables) en gas combustible, como hidrógeno o metano sintético.

Sector coupling: Integración de los sectores eléctrico, térmico y del transporte para optimizar el uso de energía y aumentar la flexibilidad del sistema energético.

Flexibilidad: Capacidad del sistema energético para adaptarse a variaciones de generación o demanda manteniendo estabilidad y eficiencia.

Balancing Responsible Parties (BRP): Entidades encargadas de equilibrar en tiempo real la energía generada y consumida dentro de su cartera, evitando desbalances en la red eléctrica.

Flujo de calor: Transferencia de energía térmica entre cuerpos o sistemas debido a una diferencia de temperatura.

Eficiencia energética: Relación entre la energía útil obtenida y la energía total consumida; mide cuán bien se aprovecha la energía.

Temperatura de setpoint (T_{set}): Valor de referencia de temperatura que se desea mantener en un sistema de control térmico.

UART (Universal Asynchronous Receiver-Transmitter): Protocolo de comunicación serial asíncrono usado en microcontroladores para transmitir datos entre dispositivos.

ADC (Analog-to-Digital Converter): Conversor que transforma una señal analógica (voltaje) en una señal digital interpretable por un microcontrolador.

TTL (Transistor-Transistor Logic): Nivel de voltaje estándar para señales digitales, comúnmente 5 V o 3.3 V, usado en circuitos lógicos y comunicación serial.

Déficit energético: Situación en la que la demanda de energía supera la capacidad de generación o suministro.

Microrred: Sistema eléctrico local con generación, almacenamiento y cargas, que puede operar conectado o aislado de la red principal.

Demanda de energía: Cantidad de energía requerida por los consumidores en un periodo determinado.

Emulación de demanda de energía: Técnica que reproduce de forma controlada el comportamiento de consumo energético de un sistema real.

Radiación solar: Energía emitida por el sol que llega a la superficie terrestre, expresada normalmente en W/m^2 , usada para generación solar.

Capítulo 1

Introducción

1.1. Planteamiento del problema

Desde principios del siglo XX, los combustibles fósiles han sido fundamentales para la generación de energía, el transporte y la industria, impulsando el crecimiento económico y el desarrollo global. Sin embargo, es evidente que el uso y dependencia a dichos combustibles ha representado un aumento en las emisiones de dióxido de carbono (CO₂) en la atmósfera [1]. Debido a esto, muchos de los países con las economías más fuertes del mundo, representan también los que más emisiones de CO₂ generan. Por ejemplo, China y Estados Unidos representaron el 30.12 % y el 15 % respectivamente de las emisiones totales en 2018, proviniendo en su mayoría, de la producción de energía basada en combustibles fósiles [2].

Debido a lo anterior, se han presentado aumentos en la temperatura del planeta, provocando lo que se ha denominado cambio climático. Sin embargo, para la mitigación de esta problemática, se ha establecido una hoja de ruta mediante la cual sea posible una descarbonización del sector energético (altamente dependiente de combustibles fósiles), mediante la electrificación basada en las Fuentes de Energía Renovable (FER) [3].

Sin embargo, existen diferentes sectores de consumo, cuya electrificación no es viable en el corto plazo [4] como el transporte. De esta manera, desde 2018 se han hablado de estrategias que mencionan el hidrógeno para mitigar las emisiones de CO₂ y contrarrestar el efecto invernadero. Existen tres tipos de hidrógeno: gris, azul y verde. El hidrógeno verde se produce con energías renovables, una opción más sostenible. Sin embargo, la Agencia Internacional de Energías Renovables (IRENA) ha afirmado que su implementación enfrenta varios desafíos, principalmente el costo elevado de las tecnologías necesarias para llevar a cabo la electrólisis, que es el proceso utilizado para separar el hidrógeno del oxígeno en el agua [5].

Además de lo anterior, la masificación de las FER (crucial para la descarbonización) representan retos técnicos debido a su intermitencia e incertidumbre. Debido a esto, suelen emplearse mecanismos de almacenamiento basados en baterías, los cuales tienen limitaciones debido a su baja densidad energética. Es por esto que, en la actualidad se están evaluando diferentes alternativas para el almacenamiento de energía como las tecnologías Power-to-X (PtX), a través de las cuales, es posible realizar una gestión energética de la producción proveniente de FER mediante su transformación en otros sectores energéticos como el calor (PtH), gas (Ptg) o el mismo hidrógeno (PtH₂) [6].

Es gracias a lo anterior que, el concepto de microrredes ha permitido la gestión de la generación y el consumo de electricidad de manera local, para reducir la dependencia de combustibles fósiles. Sin embargo, es necesario realizar estudios que permitan la integración de tecnologías PtH al concepto de microrred [7].

Con base a lo anterior se presenta la siguiente pregunta problema:

¿Cómo implementar una microrred que gestione la intermitencia de la energía fotovoltaica mediante mecanismos PtH, utilizando los recursos disponibles en el laboratorio de energías de la universidad Santo Tomás, y que permita la reducción de las emisiones de CO₂?

1.2. Objetivos

1.2.1. Objetivo general

Diseñar una microrred que permita la gestión de energía solar fotovoltaica mediante mecanismos PtH y los DER disponibles en el Laboratorio De Energías Renovables de la Universidad Santo Tomás.

1.2.2. Objetivos específicos

- Identificar las principales técnicas de modelado matemático para la caracterización de tecnologías PtH con base en una revisión bibliográfica.
- Seleccionar el modelo matemático del horno disponible en el laboratorio de energías, con base en pruebas experimentales, para su uso como tecnología PtH.
- Implementar una microrred mediante la integración del horno para la gestión de la generación fotovoltaica.

- Validar el desempeño de la microrred propuesta para la gestión de generación fotovoltaica empleando indicadores de flexibilidad.

1.3. Justificación

En un contexto de crecientes desafíos globales como el cambio climático, la transición energética ha sido imprescindible para asegurar un nuevo enfoque en el suministro energético, aprovechamiento de excedente de energía renovable e innovación de tecnologías que fortalezcan la competitividad. Esto no solo responde a necesidades inmediatas, sino que también establece una ruta para el desarrollo energético de Colombia en las próximas décadas, alineándose con los Objetivos de Desarrollo Sostenible y las metas de reducción de emisiones de carbono. En este contexto, Colombia ha actualizado su plan energético nacional, comprometiéndose a la Contribución Nacional Determinada (NDC), con el objetivo de reducir las emisiones de CO₂ en un 51 % para el año 2030. Presentando distintas estrategias para la producción y el consumo de la energía. A nivel global existen nuevas e innovadoras tecnologías disponibles para la generación y una mejor gestión de la demanda, complementado los procesos de sustitución de combustibles fósiles, acelerando el proceso de descarbonización y promoviendo una transición energética justa [8].

Para la descarbonización en Colombia, se propone reemplazar combustibles fósiles con FER, transformando la matriz energética nacional. Por lo que en este proyecto se implementan microrredes que utilizan energía fotovoltaica (PV), lo que contribuye a reducir las emisiones de CO₂. Dado que estas fuentes son intermitentes, es esencial flexibilizarlas mediante microrredes y tecnologías PtH. Esto permite almacenar energía en forma de calor, utilizando el horno disponible en el laboratorio de energías, lo que mejora la gestión y el aprovechamiento de la energía renovable [7].

Teniendo en cuenta lo anterior, al implementar un horno eléctrico (el cual funciona como tecnología PtH) proporciona flexibilidad a las FER, adaptándose a las variaciones en la producción de energía renovable. Al almacenar calor durante períodos de alta producción, permite utilizar esa energía en momentos de menor disponibilidad. Esto optimiza el uso de recursos renovables y reduce la dependencia de fuentes tradicionales, contribuyendo a un sistema energético más resiliente y sostenible [9]. Asimismo, este proyecto de grado contribuye de manera significativa al Grupo de Investigación MEM, al fortalecer las líneas de investigación como lo son Energía Eléctrica y Automatización, brindando un punto de partida para futuros proyectos.

1.4. Impacto social

En el documento “Agenda 2030 para el Desarrollo Sostenible” realizada por la Cepal (Comisión Económica para América Latina y el Caribe), se evidencia la relevancia de esta organización de abordar una amplia gama de desafíos a nivel global. El principal desafío que se ha identificado desde hace varias décadas es el cambio climático [10]. El cual se ha intensificado debido a la degradación ambiental. El cambio climático se presenta principalmente por las emisiones de CO₂, causando el gas de efecto invernadero (GEI), estos gases son perjudiciales tanto para la supervivencia humana, especies de animales, vegetales, alteraciones en los fenómenos meteorológicos y escasez de alimentos [11].

Considerando lo anterior, si se empieza a priorizar la transición energética con las FER, se puede “Garantizar el acceso a una energía asequible, segura, sostenible y moderna” a las comunidades que dependen de combustibles fósiles, mermando así las desigualdades sociales. Entre menos emisiones de GEI se generen, se tiene un ambiente más limpio y saludable. Adicionalmente, este proyecto puede generar un impacto positivo en la industria, “Garantizando modalidades de consumo y producción sostenibles” lo que influye directamente en el consumo de recursos ambientales. Al implementar tecnologías sostenibles y optimizar el uso de recursos, se puede reducir la huella ecológica y promover prácticas más responsables. Esto no solo beneficiará al medioambiente, sino que también incentivará a otras empresas a patrocinar enfoques similares, creando un efecto multiplicador en la industria hacia la sostenibilidad [12].

De igual manera, las instituciones de Educación Superior, como la Universidad Santo Tomás (USTA) en Bogotá, buscan fortalecer su relación con el contexto social, contribuyendo al desarrollo sostenible. En el proyecto, se utilizan los recursos del Laboratorio De Energías Renovables, lo que no solo potencia la investigación y la formación técnica, sino que también promueve el bienestar de la comunidad. Al abordar desafíos como el cambio climático y las energías renovables, se genera oportunidades para quienes más lo necesitan, contribuyendo a una transformación social positiva.

Capítulo 2

Estado del arte

Dado el desafío que representa el cambio climático a nivel global, es crucial implementar sistemas que utilicen fuentes de energía renovable FER como base para la transición hacia un futuro más sostenible. No obstante, debido a problemas como la intermitencia e inestabilidad inherentes a estas fuentes, es necesario realizar una revisión exhaustiva de la literatura científica [13]. Esta revisión debe abordar temas clave como las tecnologías de acople entre vectores energéticos, la densidad energética, la flexibilidad de las FER y la gestión energética.

Las tecnologías PtX son un conjunto de soluciones innovadoras que permiten convertir electricidad en otras formas de energía o combustibles útiles [14]. En el contexto de la transición energética, la integración de las FER plantea un desafío significativo debido a su variabilidad. Este término se refiere a la naturaleza intermitente de la generación de energía renovable; por ejemplo, la energía solar se produce solo durante el día y depende de las condiciones climáticas. Esta variabilidad puede causar desequilibrios entre la oferta y la demanda en las redes eléctricas. La implementación de tecnologías PtX puede contribuir a mejorar este equilibrio energético, ayudando a gestionar la masificación de renovables de forma más efectiva.

2.1. Flexibilidad en las redes de distribución híbridas

La flexibilidad en el contexto de los sistemas de energía se refiere a la capacidad del sistema para adaptarse de manera eficiente a los cambios en las FER. Flexibilizar un sistema de energía consiste en adaptarlo para enfrentar la incertidumbre y variabilidad [15]. En las redes de corriente continua (DC), la flexibilidad del flujo de potencia y la ausencia de potencia reactiva mejoran la eficiencia energética y la estabilidad de voltaje en comparación con las redes de corriente alterna (AC), al reducir las pérdidas y complejidad en la gestión de la energía [16].

2.2. Innovación Energética: El Impacto de la Tecnología PtH en la Eficiencia y Sostenibilidad

Como mecanismos de innovación en el sector energético, diferentes autores han estudiado la posibilidad de que los consumidores estén en la capacidad de ofertar servicios de flexibilidad con ayuda de tecnologías de acople y, con especial enfoque en las PtH.

En [17], los autores investigan cómo los edificios pueden transformarse en productores de energía a través de servicios de flexibilidad, enfocándose en la tecnología Power-to-Heat (PtH). Esta tecnología permite convertir el exceso de energía renovable en calor y aprovechar la inercia térmica para gestionar el consumo energético. Se destacan indicadores clave de flexibilidad, como la mejora de la eficiencia energética, la capacidad de ajustar el consumo en función de la demanda de la red y el equilibrio entre costos de suministro y demanda.

Adicionalmente en [18], los autores han investigado el uso de Power-to-Heat (PtH) como parte de la electrificación industrial, siendo clave en los esfuerzos de descarbonización. En sectores como el cemento, demostraron que es posible electrificar completamente la demanda energética con PtH. Sin embargo, la mayor parte de las emisiones de CO₂ provienen de los procesos químicos, lo que limita la reducción total de emisiones a un 31 %. Para abordar el 68 % restante, Europa ha avanzado en tecnologías como la captura y almacenamiento de carbono (CCS) y el uso de materias primas alternativas, con el objetivo de alcanzar los objetivos de neutralidad climática del Green Deal.

2.3. Gestión energética y flexibilidad en la demanda

La gestión energética y la flexibilidad en la demanda, buscan optimizar el consumo eléctrico para mejorar su eficiencia y sostenibilidad, debido a la masificación de fuentes renovables. La flexibilidad de la demanda se presenta como una estrategia clave para alinear el consumo eléctrico con los momentos de alta producción de energía renovable, [19] lo que ha sido objeto de estudio por varios autores, como se evidencia a continuación:

En [19], se analiza un sistema energético distribuido en Colombia que integra generación renovable y baterías de iones de litio, aplicando flexibilidad en la demanda. Los autores demuestran que el desplazamiento de cargas no críticas, como electrodomésticos inteligentes, puede aumentar la participación de energías renovables en un 6.5 % y generar ahorros anuales del 18 % en las facturas eléctricas en comparación con un sistema sin gestión de carga. Esta investigación

resalta el papel crucial de la gestión de la demanda en la transición hacia sistemas energéticos más sostenibles.

Por otro lado, el estudio [20] examina la flexibilidad de las Cargas Térmicas Controladas (TCL) en un congelador de un supermercado danés, donde la temperatura se mantiene alrededor de $-18\text{ }^{\circ}\text{C}$, salvo durante los ciclos de descongelación. La demanda aumenta durante el horario de apertura debido a la constante apertura de vitrinas, mientras que en el cierre, el consumo disminuye gracias al aislamiento. La inercia térmica de las TCL permite ajustar el consumo sin afectar su función principal, convirtiéndolas en recursos valiosos para la Regulación de Frecuencia de Reserva de Mediano Plazo (mFRR). Esta flexibilidad no solo contribuye al equilibrio de la red eléctrica, sino que también subraya el papel fundamental de las TCL en la integración de energías renovables y la sostenibilidad de los sistemas eléctricos.

2.4. Integración y Operación de Microrredes: Modelos, elementos y Desafíos

De acuerdo con el artículo [21], una microrred es un sistema compuesto por cargas y recursos energéticos distribuidos (DER), que opera como una unidad controlable. Estas microrredes pueden desconectarse y reconectarse de forma segura a la red principal, manteniendo los parámetros de voltaje y frecuencia dentro de los límites permitidos. Además de mejorar la resiliencia del sistema y garantizar un suministro continuo, son esenciales para electrificar zonas remotas sin acceso a la red. Basándose en el informe "*Program on Technology Innovation: Microgrid Implementations, Literature Review*", se subraya los avances tecnológicos en microrredes y las barreras para su integración efectiva, como la variabilidad de los recursos energéticos, la interconexión con la red, el marco normativo y el análisis económico, factores claves para evaluar su viabilidad y costos.

Asimismo a planificación y diseño de microrredes requieren estudios eléctricos detallados que consideren la generación, protección y los esquemas operativos. Por ejemplo, en [22] se emplean modelos de optimización, tanto lineales como no lineales, para maximizar la eficiencia y resiliencia en microrredes que integran DER, como paneles solares y turbinas eólicas.

Por otra parte en [23], se resalta la creciente importancia de las microrredes de corriente alterna (AC) debido a la utilización de las FER, como la energía solar fotovoltaica y las celdas de combustible, junto con la integración de sistemas de almacenamiento de energía. Además permiten la integración de diversas fuentes de energía renovable, como turbinas eólicas (WTG) y paneles

solares, lo que contribuye a la sostenibilidad y la reducción de emisiones de carbono. Para asegurar su operación, se emplean técnicas de control centralizado, descentralizado, distribuido, multinivel y jerárquico, adaptando el sistema a las condiciones operativas y necesidades específicas. Estos beneficios hacen de las microredes AC una opción atractiva para avanzar hacia sistemas energéticos más sostenibles, flexibles y resilientes.

2.5. Tabla resumen estado del arte

La siguiente tabla presenta una síntesis de las investigaciones y estudios más relevantes relacionados con la gestión energética y la flexibilidad en la demanda. Se abordan diferentes enfoques para la integración de microredes y las tecnologías de almacenamiento, proporcionando un panorama de los avances más significativos en el campo y los desafíos pendientes en la operación de estos sistemas.

TABLA 1: Tabla resumen estado del arte

Referencia	Año	Aporte
[22]	2017	El artículo de IEEE Power and Energy Magazine describe microredes que pueden operar conectadas a la red o de forma independiente en emergencias. Usan recursos distribuidos como energía solar y baterías, y cumplen con el estándar IEEE 1547-2003. Se destacan varios tipos de microredes y un análisis tecnoeconómico que optimiza su diseño, como en el centro médico BNMC en Buffalo.
[21]	2019	Las microrredes operan con cargas y recursos distribuidos (DER) y deben mantener la estabilidad al desconectarse y reconectarse a la red. Este artículo destaca estudios clave y simulaciones que muestran la estabilidad de frecuencia y voltaje en microrredes aisladas, identificando reglas para su operación.
[18]	2020	La electrificación del sector industrial europeo, mediante Power-to-Heat (PtH), es clave para reducir emisiones de CO ₂ . Se estima que el 78 % de la demanda energética puede electrificarse con tecnologías actuales, y hasta un 99 % con tecnologías en desarrollo, impulsando la transición hacia bajas emisiones.
[24]	2020	Este artículo presenta un modelo matemático convexo (SOCP) para optimizar la operación de microredes DC con energías renovables y baterías. El modelo garantiza el óptimo global y reduce costos operativos. Además, utiliza redes neuronales para predecir fuentes renovables.
[14]	2021	Alemania desarrolló Power-to-Gas para almacenar energía eólica y solar como gas sintético, resolviendo el problema del almacenamiento estacional y creando el concepto de "sector coupling". Esto evolucionó hacia Power-to-X, que convierte electricidad en combustibles y productos climáticamente neutros. Aunque útil, su eficiencia es menor que el uso directo de electricidad o hidrógeno.

[13]	2022	El cambio climático afecta la energía renovable, como la eólica y solar, aumentando la vulnerabilidad a fenómenos extremos. Este artículo propone un sistema energético-climático regional para evaluar su potencial y sugiere extender las evaluaciones a largo plazo, desarrollar un modelo refinado de impacto y adoptar modelos climáticos en la planificación de sistemas eléctricos.
[19]	2022	El estudio modela un sistema energético con solar, eólica y baterías de litio. Al aplicar flexibilidad en la demanda, se incrementa en un 6,5 % la participación de energías renovables y se reduce en un 18 % el costo eléctrico en comparación con un sistema sin gestión de demanda.
[15]	2023	Este documento evalúa la capacidad de los sistemas de energía renovable para gestionar fluctuaciones mediante un índice de ajuste entre la flexibilidad disponible y la requerida. Los indicadores se validan con datos operativos reales de una región en China, confirmando su efectividad.
[17]	2023	El estudio analiza la transición de los edificios de consumidores a productores de energía mediante servicios de flexibilidad y la tecnología Power-to-Heat. Destaca la inercia térmica, el control predictivo y la optimización de sistemas como elementos clave, e identifica indicadores de rendimiento en eficiencia y costos, subrayando la necesidad de integrar tecnologías avanzadas para mejorar la flexibilidad energética.
[16]	2024	El artículo revisa el flujo de potencia óptimo (OPF) en redes híbridas AC/DC (HDN), que mejoran la eficiencia y facilitan la integración de energías renovables. Se abordan las no linealidades en la función objetivo y se proponen métodos para resolver el OPF, destacando la necesidad de considerar futuras investigaciones sobre fallas y calidad de energía en estas redes.
[20]	2024	El artículo compara la flexibilidad de la demanda en Dinamarca entre el servicio mFRR y el cambio de carga en un congelador de supermercado. Aunque el cambio de carga es casi tan rentable como mFRR, afecta más la temperatura. Se recomienda que los operadores ofrezcan incentivos para que las TCL proporcionen mFRR, manteniendo la estabilidad de la red.

[23]	2024	Este artículo presenta una microrred AC aislada con integración de fuentes renovables y cargas AC. Se emplea control vectorial para gestionar eficientemente las cargas y un sistema de almacenamiento para mantener la estabilidad ante fluctuaciones de carga y generación. La validación mediante experimentos en hardware demuestra su capacidad para operar de forma eficiente y estable.
------	------	--

Capítulo 3

Marco Teórico

3.1. Flexibilidad en las FER

Según [25], la flexibilidad en las FER es un factor crucial para hacer frente al incremento de su participación en los sistemas eléctricos a gran escala. La flexibilidad se entiende como la capacidad del sistema para adaptarse y satisfacer la demanda de manera eficiente. Es fundamental para asegurar que las FER se integren adecuadamente al sistema eléctrico, permitiendo su operación de manera económica y sostenible, a pesar de la variabilidad inherente a estas fuentes de energía.

3.1.1. Métodos Cuantificables

3.1.1.1. Método Implícito

En este se usa la incertidumbre de manera general y resuelve problemas de programación de generación de energía utilizando métodos como la probabilidad, técnicas estocásticas y enfoques robustos. Adicionalmente se hace uso de parámetros inciertos. Este enfoque busca optimizar los recursos disponibles para enfrentar una cantidad de incertidumbre ya conocida, pero no toma en cuenta la flexibilidad del sistema ante cambios inesperados.

3.1.1.2. Método Explícito

Estos consideran que los sistemas eléctricos del futuro tendrán una capacidad de flexibilidad limitada. Estos enfoques señalan que, si se intenta cubrir escenarios extremos de incertidumbre

con generadores caros, los costos serán muy altos. En casos aún más extremos, puede que ni siquiera sea posible encontrar soluciones que funcionen.

3.1.2. Servicios Manuales

3.1.2.1. Reserva de Restauración de Frecuencia (mFRR)

Los servicios de mFRR son gestionados por entidades como el Operador del Sistema de Transmisión (TSO), cuya misión es garantizar un transporte de energía eléctrica estable y equilibrado, manteniendo la igualdad entre la generación y la demanda de electricidad. El mFRR se activa cuando otros servicios auxiliares más rápidos no logran resolver una caída o un aumento en la frecuencia del sistema eléctrico, como podría suceder en caso de una desconexión imprevista de una planta de energía o un incremento súbito en la demanda.

En situaciones donde la demanda supera la oferta, los generadores deben incrementar su velocidad de rotación para suplir la diferencia, lo que provoca un aumento en la frecuencia de la red. Por el contrario, si la oferta excede la demanda, los generadores reducen su velocidad de rotación, lo que genera una disminución en la frecuencia del sistema eléctrico. (*Synergy Among Flexible Demands: Forming a Coalition to Earn More from Reserve Market*)

3.1.3. Métodos de modelado

3.1.3.1. Modelo Matemático para la Maximización del Valor de Flexibilidad mediante mFRR

De acuerdo con [20], los participantes del mercado, como los *Balancing Responsible Parties (BRP)*, tienen como objetivo maximizar los ingresos generados a través de la provisión de flexibilidad. Esta flexibilidad se gestiona mediante la oferta de capacidad de reserva, la activación de regulación, y el cumplimiento de compromisos previamente establecidos. El valor económico total para el BRP se calcula considerando cuatro componentes clave:

- **Pago por reserva (Reservation Payment):** Se genera al ofrecer la capacidad de reserva aceptada al mercado mFRR. Este pago está relacionado con el precio de reserva.
- **Pago por activación (Activation Payment):** Corresponde a los ingresos obtenidos al activar la capacidad de regulación ascendente a un precio de balance.
- **Costo por rebote (Rebound Cost):** Representa el costo asociado al consumo adicional necesario para restaurar los niveles de demanda después de una activación de regulación.

- **Costo por penalización (Penalty Cost):** Se incurre en este costo si el BRP no logra entregar la capacidad prometida, lo que resulta en una penalización determinada por el precio de penalización y el déficit entre la capacidad comprometida y la entregada.

$$\underbrace{\sum_{h=1}^{24} \lambda_h^r p_h^{r,\uparrow}}_{\text{Reservation payment}} + \underbrace{\sum_{h=1}^{24} \lambda_h^b p_h^{b,\uparrow}}_{\text{Activation payment}} - \underbrace{\sum_{h=1}^{24} \lambda_h^b p_h^{b,\downarrow}}_{\text{Rebound cost}} - \underbrace{\sum_{h=1}^{24} \lambda^p s_h}_{\text{Penalty cost}} \quad (3.1)$$

Donde:

- $p_h^{(r,\uparrow)}$: Capacidad de reserva ascendente ofrecida y aceptada en la hora h .
- λ_h^r : Precio por reserva aceptada en la hora h .
- $p_h^{(b,\uparrow)}$: Capacidad de regulación ascendente activada en la hora h .
- λ_h^b : Precio de balance en la hora h .
- $p_h^{(b,\downarrow)}$: Rebote de demanda tras la activación de la regulación ascendente en la hora h .
- λ^p : Precio de penalización.
- $s_h = \max\{0, p_h^{(r,\uparrow)} - p_h^{(b,\uparrow)}\}$: Déficit entre la capacidad comprometida y la entregada en la hora h .

3.1.3.2. Cargas Controladas Termostáticamente (TCL)

Como se menciona en [26], las Cargas Controladas Termostáticamente, conocidas como TCL (por sus siglas en inglés, Thermostatically Controlled Loads), son dispositivos eléctricos capaces de regular su consumo de energía en función de un controlador de temperatura, manteniendo la temperatura dentro de un rango preestablecido. Estas cargas son ampliamente utilizadas en sistemas que almacenan energía térmica, como sistemas de calefacción, aire acondicionado y otros equipos similares.

Además de su función principal, las TCL desempeñan un papel fundamental en las redes eléctricas modernas al proporcionar flexibilidad en la demanda. Esto se logra al ajustar su consumo energético en respuesta a señales de la red, lo que contribuye a equilibrar la oferta y la demanda de energía en tiempo real. Asimismo, participan activamente en servicios auxiliares, como la regulación de frecuencia, ayudando a mantener la estabilidad del sistema eléctrico.

Modelo matemático de la TCL: Los sistemas están formados por diversos materiales con propiedades específicas que influyen en la disipación y retención del calor. La selección de estos materiales es fundamental para optimizar el almacenamiento térmico y garantizar una transferencia de calor eficiente dentro del sistema. Factores como la conductividad térmica, la capacidad de calor específico y la estabilidad térmica determinan el rendimiento del almacenamiento energético, afectando la rapidez con la que el sistema puede absorber, almacenar y liberar energía.

A continuación se mencionan los parámetros térmicos que se deben de tener en cuenta:

- **Capacidad de Calor Específico (C_{air}, C_w):** representa la cantidad de energía necesaria para cambiar la temperatura del aire en la habitación y la de las paredes del horno.
- **Resistencia Térmica ($R_{\text{room-wall}}, R_{\text{wall-ext}}$):** Determina la velocidad a la que el calor se transfiere entre la habitación y las paredes, así como entre las paredes y el exterior.
- **Dinámica de la Temperatura:** El comportamiento térmico del sistema se modela mediante dos ecuaciones diferenciales:

Ecuación de la temperatura de la habitación:

$$\frac{dT_{\text{room}}}{dt} = \frac{1}{C_{\text{air}}} \left(m_{\text{mode}} P_{\text{heater}} - \frac{T_{\text{room}} - T_{\text{wall}}}{R_{\text{room-wall}}} \right) \quad (3.2)$$

Donde:

- T_{room} : Temperatura de la habitación.
- P_{heater} : Potencia del calentador.
- m_{mode} : Representa el modo de operación del calentador (encendido/apagado o modulado).
- $R_{\text{room-wall}}$: Resistencia térmica entre la habitación y la pared.

Ecuación de la temperatura de la pared:

$$\frac{dT_{\text{wall}}}{dt} = \frac{1}{C_w} \left(\frac{T_{\text{wall}} - T_{\text{out}}}{R_{\text{wall-ext}}} \right) \quad (3.3)$$

Donde:

- T_{wall} : Temperatura de la pared.
- T_{out} : Temperatura exterior.
- $R_{\text{wall-ext}}$: Resistencia térmica de la pared hacia el ambiente externo.

3.2. Indicadores de flexibilidad

Indicadores de flexibilidad en sistemas eléctricos

La flexibilidad en sistemas eléctricos es un aspecto fundamental para garantizar la estabilidad y eficiencia operativa ante variaciones en la generación y la demanda. En este contexto, diversos indicadores permiten cuantificar la capacidad de respuesta del sistema. Algunos de los más relevantes incluyen:

- Normalized Flexibility Index (NFI): Evalúa la contribución de cada generador a la flexibilidad del sistema. Un valor alto de este índice indica una mejor capacidad de adaptación a las fluctuaciones de la generación y la demanda. Se define a partir de la flexibilidad de cada unidad generadora con la siguiente ecuación:

$$NFI_A = \sum_{i \in A} \frac{P_{\max}(i)}{\sum P_{\max}(i)} \times \text{flex}(i) \quad (3.4)$$

Donde $P_{\max}(i)$ es la capacidad máxima de la unidad y representa la flexibilidad de la unidad.

- Loss of Wind Expectation (LOWE): Representa la probabilidad de vertimiento de generación eólica debido a restricciones operativas. Este indicador ayuda a cuantificar la pérdida de energía renovable disponible debido a limitaciones en la red o la demanda. Se calcula como:

$$LOWE = 1 - [1 - P(VMLL)] \times \left[F \left(\sum Ramp - R_{dn} \right) \right] \times \left[F \left(\sum Ramp - R_{up} \right) \right] \quad (3.5)$$

Donde $P(VMLL)$ representa la probabilidad de que la demanda neta esté por debajo del nivel mínimo de carga, R_{up} y R_{dn} son las capacidades de rampa.

- Insufficient Ramping Resource Probability (IRRP): Mide la probabilidad de no contar con suficiente capacidad de rampa para satisfacer cambios en la demanda. Este indicador

permite evaluar si el sistema eléctrico es lo suficientemente rápido para adaptarse a fluctuaciones en la generación y el consumo. Se calcula considerando la capacidad de los recursos flexibles en relación con los cambios en la demanda neta. La ecuación es:

$$IRRP_{t,i} = AFD_i(NLR_{t,i} - 1) \quad (3.6)$$

Donde:

- $IRRP_{t,i}$ es la probabilidad de que el sistema no pueda seguir los cambios de la demanda neta en el tiempo t para un horizonte i .
- AFD_i es la distribución acumulativa de la disponibilidad de flexibilidad para el horizonte de tiempo i .
- $NLR_{t,i}$ es la rampa de la demanda neta en el intervalo de tiempo t para el horizonte de tiempo i , y se calcula como:

$$NLR_{t,i} = NL_t - NL_{t-i} \quad (3.7)$$

Donde NL_t representa la demanda neta en el tiempo t . Esto permite estimar la probabilidad de que el sistema tenga suficientes rampas disponibles para responder a los cambios en la demanda.

- **Lack of Ramp Probability (LORP):** Evalúa la capacidad del despacho eléctrico para responder ante errores en la predicción de la demanda neta. Este indicador mide la flexibilidad operativa del sistema y su capacidad de ajuste ante incertidumbre en la demanda y generación. Se calcula mediante:

Para la capacidad de subida:

$$LORP_{dn,\tau}[t] = P \left(\sum (P_{ig}[t] - \min\{\tau R_i, P_i[t] - P_{imin}\}) > P_{sl}[t + \tau] \right) \quad (3.8)$$

La aplicación de estos indicadores en la operación de sistemas eléctricos permite optimizar la gestión de la generación y el consumo, mejorando la estabilidad y reduciendo pérdidas de energía. Estrategias como el almacenamiento, el control predictivo y la programación de cargas pueden ayudar a mejorar estos indicadores y garantizar un uso eficiente de la energía disponible [27].

3.3. Almacenamiento de energía (inercia térmica):

En una microrred con generación fotovoltaica y un horno eléctrico como única carga, el almacenamiento de energía se obtiene en forma de temperatura, en lugar de baterías eléctricas convencionales. Este enfoque permite gestionar la variabilidad de la producción solar y garantizar un suministro térmico estable, desacoplando la generación y el consumo de energía [9].

La gestión del almacenamiento térmico en la microrred se basa en un control dinámico de carga y descarga térmica, la fase de carga sucede cuando la producción solar excede la demanda del horno, el excedente de energía se convierte en calor y se almacena en un medio térmico. La fase de descarga se tiene en cuenta al momento donde la radiación solar disminuye, la energía térmica acumulada se libera de manera controlada para mantener el funcionamiento del horno. Este enfoque optimiza el uso de la energía renovable, garantizando estabilidad en el suministro térmico y minimizando pérdidas energéticas [9].

Inercia Virtual.

La inercia virtual es una técnica utilizada en redes eléctricas con alta penetración de las FER para compensar la reducción de la inercia natural proporcionada por los generadores síncronos. Su objetivo es minimizar el desbalance entre generación y demanda, regulando la frecuencia del sistema, evitando inestabilidad [6].

3.4. Modelo de un sistema virtual de almacenamiento de energía

En una microrred con una única carga TLC, el almacenamiento de energía se realiza en forma de temperatura, en lugar de baterías eléctricas convencionales. Este enfoque permite gestionar la variabilidad de la producción solar y garantizar un suministro térmico estable, desacoplando la generación y el consumo de energía [9].

La gestión del almacenamiento térmico en la microrred se basa en un control dinámico de carga y descarga térmica, la fase de carga sucede cuando la producción solar excede la demanda de la carga, el excedente de energía se convierte en calor y se almacena en un medio térmico. La fase de descarga se tiene en cuenta al momento donde la radiación solar disminuye, la energía térmica acumulada se libera de manera controlada para mantener el funcionamiento de la carga.

Este enfoque optimiza el uso de la energía renovable, garantizando estabilidad en el suministro térmico y minimizando pérdidas energéticas [9].

Para analizar la eficiencia del sistema, se pueden desarrollar modelos de simulación que estudie la evolución de la temperatura en el almacenamiento térmico bajo distintos perfiles de generación solar y demanda térmica. Estos modelos permiten seleccionar el material de almacenamiento más adecuado según su capacidad calorífica y estabilidad térmica, optimizar las estrategias de control para mejorar el rendimiento del sistema y maximizar la eficiencia energética [9].

Capítulo 4

Diseño Metodológico

En el siguiente apartado se muestra la metodología del proyecto, la cual se divide en cuatro etapas, respectivamente estas etapas contienen las distintas actividades a completar para cumplir con los objetivos específicos. En la Figura 1 se muestra el cuadro de metodología en cascada del proyecto.

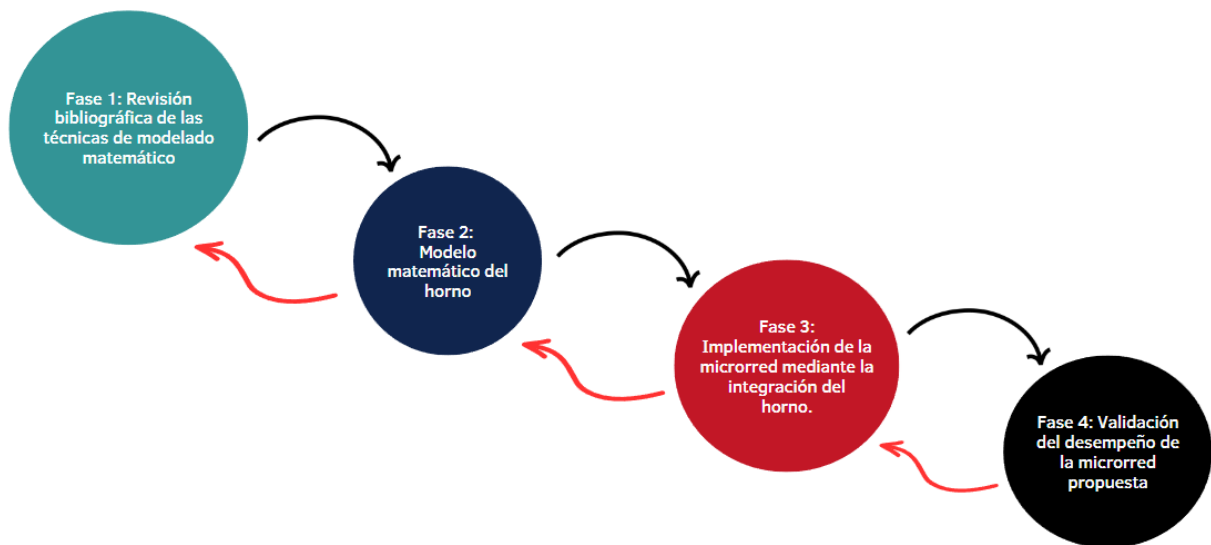


FIGURA 1: Diagrama de la metodología planteada para el desarrollo del proyecto.
Tomado de: Autores.

4.1. Revisión bibliográfica de las técnicas de modelado matemático

En la primera fase del proyecto, se selecciona la tecnología PtH adecuada para desarrollar un modelo matemático que permita la caracterización del sistema, en este caso, el horno disponible en el laboratorio de energías. Para fundamentar esta elección, se realiza una revisión bibliográfica de las principales tecnologías PtH, evaluando sus características, aplicaciones y comportamientos bajo diferentes condiciones operativas. Esta revisión sirve como base para la identificación del modelo matemático adecuado que represente con precisión el comportamiento del horno, con el objetivo de integrarlo de manera efectiva en la microrred.

- Tarea 1: Búsqueda bibliográfica en bases de datos científicas sobre modelado de tecnologías PtH.
- Tarea 2: Identificación y selección de los artículos y documentos más relevantes.
- Tarea 3: Clasificación de las técnicas de modelado según su enfoque y sus aplicaciones a tecnologías PtH.
- Tarea 4: Caracterización del Horno.

4.2. Modelo matemático del horno

Una vez obtenida la caracterización del horno, se procede a desarrollar su modelo matemático. Para ello, se ejecutan pruebas experimentales que permiten la obtención de datos reales sobre su comportamiento térmico y energético en diversas condiciones operativas. Estos datos son utilizados para validar y ajustar modelos preexistentes. A través de esta etapa, se garantiza que el modelo matemático refleje de manera confiable el funcionamiento del horno, permitiendo su integración a la microrred propuesta y contribuyendo a la gestión de la generación fotovoltaica mediante mecanismos PtH.

- Tarea 5: Desarrollo de un plan experimental para obtener los datos necesarios del horno.
- Tarea 6: Pruebas experimentales, registrando variables claves como temperatura, consumo energético, flexibilidad energética, etc.
- Tarea 7: Estudio de los datos obtenidos y ajuste según los modelos matemáticos preexistentes.
- Tarea 8: Modelo matemático que se ajuste al comportamiento real del horno.
- Tarea 9: Validación del modelo mediante simulaciones con los datos experimentales.

4.3. Implementación de una microrred mediante la integración del horno.

Basándose en el diseño de la microrred se integra el modelo matemático del horno para realizar pruebas experimentales bajo distintos escenarios de carga. Estas pruebas permiten evaluar el comportamiento del sistema en condiciones reales, analizando la interacción entre la generación fotovoltaica, las diferentes cargas conectadas y la tecnología PtH representada por el horno. A través de esta integración, se busca optimizar el balance energético y la flexibilidad operativa de la microrred, mejorando su capacidad para gestionar la demanda y la generación de manera eficiente, asegurando la estabilidad y el aprovechamiento de la energía solar disponible.

- Tarea 10: Identificación y clasificación de los diferentes tipos de carga disponibles en el laboratorio de energías.
- Tarea 11: Esquemático que permita la conexión de las diferentes cargas con los sistemas de generación fotovoltaica y horno.
- Tarea 12: Integración en la microrred, asegurando una adecuada comunicación y control entre los componentes.
- Tarea 13: Validación mediante pruebas de gestión de energía, observando el comportamiento de la microrred ante variaciones de demanda y generación.
- Tarea 14: Mejora en el aprovechamiento de la energía fotovoltaica utilizando el horno.

4.4. Validación del desempeño de la microrred propuesta

En la última fase, se valida la microrred propuesta mediante el uso de la carga, la cual es el horno, teniendo en cuenta su comportamiento a través de indicadores de flexibilidad, los cuales permiten evaluar la capacidad de adaptación de las microrredes frente a las fluctuaciones en la demanda energética. Para realizar esta validación, se desarrollarán las siguientes actividades:

- Tarea 15: Definición de los indicadores de flexibilidad aplicables a la microrred.
- Tarea 16: Obtención de resultados de la microrred
- Tarea 17: Emulación del almacenamiento virtual de la energía mediante la inercia térmica del horno.

- Tarea 18: Escritura del documento final.

Capítulo 5

Desarrollo Conceptual

5.1. Fase 1: Búsqueda bibliográfica en bases de datos científicas sobre el modelado de tecnologías PtH

Para el primer apartado, se realiza una búsqueda de documentos en las bases de datos disponibles en la página del CRAI, dentro del apartado de recursos electrónicos. Las bases de datos consultadas fueron ScienceDirect e IEEE Xplore, por ser las que proporcionaron mayor cantidad de documentos relevantes para el estudio.

5.1.1. Clasificación de las técnicas de modelado aplicadas a tecnologías Power-to-Heat (PtH)

Las técnicas de modelado aplicadas a las tecnologías Power-to-Heat (PtH) se agrupan en tres enfoques principales: modelos físicos o fenomenológicos, basados en los principios de la termodinámica y la transferencia de calor; modelos estadísticos o empíricos, sustentados en el ajuste de datos experimentales; y modelos basados en *machine learning*, que emplean algoritmos de inteligencia artificial para predecir el comportamiento de los sistemas bajo distintas condiciones operativas. A continuación, se describen cada uno de estos enfoques con mayor detalle.

5.1.1.1. Los Modelos Físicos o Fenomenológicos

Se basan en la formulación matemática de los principios fundamentales de la física, como la termodinámica, la transferencia de calor y la dinámica de sistemas térmicos. Estos modelos describen el comportamiento del sistema a partir de ecuaciones diferenciales que representan

la evolución de variables clave, como la temperatura, el flujo de calor y la eficiencia energética. En el caso de las tecnologías Power-to-Heat (PtH), los modelos físicos permiten analizar el rendimiento de dispositivos como hornos eléctricos, calentadores de agua y sistemas de almacenamiento térmico. Por ejemplo, en un horno eléctrico, se pueden modelar las pérdidas de calor por conducción, convección y radiación, junto con la capacidad térmica del material del horno y la eficiencia de la conversión de energía eléctrica en calor [28].

Un enfoque común es el modelo de balance de energía, donde la ecuación general es:

$$mc_p \frac{dT}{dt} = P_{\text{entrada}} - Q_{\text{pérdidas}} \quad (5.1)$$

donde:

- m es la masa del sistema [kg].
- c_p es el calor específico del material [J/kg·K].
- T es la temperatura del sistema [K].
- P_{entrada} es la potencia suministrada al sistema [W].
- $Q_{\text{pérdidas}}$ representa las pérdidas de calor por conducción, convección y radiación [W].

5.1.1.2. Modelos Estadísticos o Empíricos.

Los modelos estadísticos o empíricos se basan en el análisis de datos experimentales para establecer relaciones entre las variables del sistema sin necesidad de conocer en detalle los principios físicos subyacentes. Estos modelos utilizan técnicas como regresión lineal, modelos autorregresivos (ARX), modelos de caja negra y métodos de identificación de sistemas para representar el comportamiento térmico de las tecnologías PtH [29].

En la caracterización de un horno eléctrico [30], un modelo empírico puede construirse a partir de mediciones de temperatura en función del tiempo y de la potencia suministrada. Un modelo de regresión lineal simple podría expresarse como:

$$T(t) = \beta_0 + \beta_1 P + \beta_2 T_{\text{ambiente}} + \varepsilon \quad (5.2)$$

donde:

- $T(t)$ es la temperatura del sistema en un instante t [°C].
- P es la potencia eléctrica aplicada al sistema [W].
- T_{ambiente} es la temperatura del ambiente [°C].
- $\beta_0, \beta_1, \beta_2$ son coeficientes de ajuste obtenidos por regresión.
- ε es un término de error asociado a la incertidumbre del modelo.

5.1.1.3. Modelos Basados en Machine Learning

Los modelos de machine learning y deep learning se han utilizado ampliamente para analizar y predecir comportamientos complejos en diferentes sistemas. Entre los más destacados se encuentran las redes neuronales profundas (DNN), convolucionales (CNN), recurrentes (RNN), los autoencoders (DAE) y los enfoques de aprendizaje por refuerzo (DRL-IDS), además de métodos clásicos como los árboles de decisión y el aprendizaje federado. Estos modelos, evaluados en distintos conjuntos de datos, han mostrado altos niveles de precisión y capacidad para reconocer patrones no lineales, siendo los modelos híbridos y de deep learning los que presentan mejores resultados [31].

Teniendo en cuenta lo investigado anteriormente, se selecciona para el modelado matemático el enfoque estadístico o empírico, ya que este permite establecer una relación directa entre la potencia promedio del sistema y el diferencial térmico existente entre la temperatura ambiente y la temperatura de consigna (setpoint). Este enfoque resulta particularmente adecuado, pues se basa en datos experimentales y facilita la formulación de un modelo capaz de describir el comportamiento energético del sistema sin requerir una caracterización física detallada de sus parámetros.

5.1.2. Caracterización del Horno

Los datos presentados en la Tabla 2 corresponden a la caracterización del horno TO1705SG-LA de la marca Black+Decker. Los parámetros eléctricos generales, como el voltaje de entrada, la potencia máxima y el tipo de control de temperatura, se obtuvieron del manual técnico del fabricante [32]. Por otro lado, la corriente y la temperatura se determinaron experimentalmente a través de mediciones realizadas en condiciones controladas, con el fin de verificar el comportamiento real del equipo y establecer los niveles de funcionamiento del termostato mecánico. Estos valores experimentales sirvieron como base para el proceso de linealización y posterior

modelado térmico del sistema.

TABLA 2: Parámetros del horno TO1705SG-LA (Black+Decker). **Tomado de:** Autores [32].

Parámetro	Valor
Voltaje de entrada [V AC]	120
Potencia máxima [kW]	1.15
Corriente máxima [A]	9.2
Control de temperatura	termostato mecánico
Niveles de temperatura aproximados	Nivel 1: 104.4°C
	Nivel 2: 117.6°C
	Nivel 3: 142.1°C
	Nivel 4: 163.4°C
	Nivel 5: 183.1°C
	Nivel 6: 205.3°C

5.2. Fase 2: Modelo matemático del horno

El modelado matemático del horno tiene como objetivo representar de manera analítica su comportamiento térmico a partir de las variables medidas experimentalmente. Este proceso permite comprender cómo responde el sistema ante distintas condiciones operativas y establecer relaciones cuantitativas entre la potencia eléctrica aplicada y la evolución de la temperatura interna. El propósito fundamental del modelado es disponer de una herramienta que permita predecir el consumo energético del horno en función del diferencial térmico entre la temperatura de setpoint (T_{set}) y la temperatura ambiente (T_a). De este modo, se facilita el análisis del desempeño térmico y la identificación de estrategias para mejorar la eficiencia y la gestión de energía del sistema.

El modelo desarrollado se concibe como una base para la planificación energética en aplicaciones de tecnologías Power-to-Heat (PtH), ya que posibilita estimar la potencia requerida para mantener condiciones térmicas específicas y evaluar la interacción del horno con fuentes renovables y sistemas de almacenamiento virtual.

5.2.1. Desarrollo del plan experimental

El plan experimental se elabora con el propósito de definir las condiciones, instrumentos y procedimientos necesarios para la obtención de los datos requeridos en el modelado matemático

del horno. En esta etapa se establecen los niveles de temperatura (T_{set}) que permiten analizar el comportamiento térmico del sistema en distintos regímenes de operación, así como las variables a registrar: temperatura interna, potencia activa, voltaje, corriente y temperatura ambiente.

Con el fin de garantizar la fiabilidad de las mediciones, se realiza la linealización del sensor de temperatura PT100, de modo que la señal eléctrica generada mantenga una relación proporcional con la temperatura real medida. Posteriormente, se diseña y simula el circuito de acondicionamiento correspondiente, verificando su correcto funcionamiento dentro de los márgenes de precisión requeridos.

Previo al desarrollo de las pruebas experimentales, se lleva a cabo la caracterización de los equipos y dispositivos empleados en el sistema de adquisición de datos, incluyendo los equipos de medición y los elementos auxiliares como los transformadores de corriente. Esta etapa tiene como propósito asegurar la precisión, estabilidad y trazabilidad de los registros obtenidos durante las pruebas experimentales.

Finalmente, se implementa la comunicación Modbus entre el analizador CIRCUTOR y el microcontrolador, lo que permite la adquisición automatizada y sincronizada de datos durante los ensayos experimentales. Este plan experimental garantiza la obtención de información suficiente y confiable para el posterior análisis estadístico y la formulación del modelo matemático.

5.2.1.1. Circuito de acondicionamiento

Dado que el sensor PT100 presenta una respuesta no lineal y una variación de resistencia relativamente pequeña, se emplea un circuito de acondicionamiento que amplifica y adapta su salida para facilitar la lectura. En la Figura 2 se muestra el circuito implementado, el cual utiliza un amplificador operacional LM324 configurado con retroalimentación negativa. El diseño utiliza un puente de Wheatstone, donde el terminal E+ del sensor se conecta al conjunto de resistencias R1 y R2 (ambas de 5.1 k Ω). El terminal S+ se une a la resistencia R4 de 1 k Ω , vinculada a la entrada no inversora del amplificador. La entrada inversora recibe la señal proveniente del punto medio entre las resistencias R3 (20 Ω) y R5 (1 k Ω), lo que completa el balance del puente.

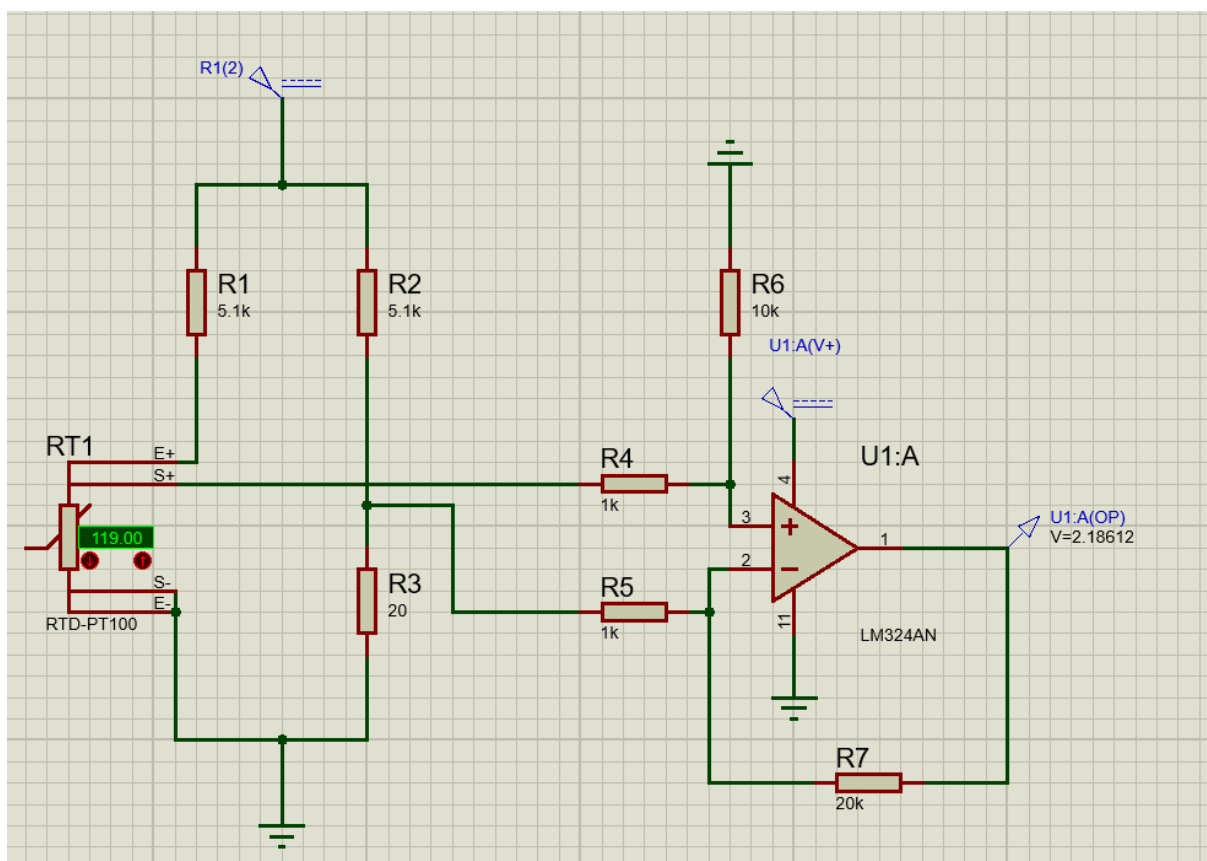


FIGURA 2: Circuito de acondicionamiento de la PT100. Tomado de: Autores.

El puente de Wheatstone se alimenta con una tensión de 4.8 V, lo que garantiza una excitación adecuada del sensor y un nivel de señal compatible con las etapas posteriores de acondicionamiento. El amplificador operacional LM324 recibe una alimentación positiva de 14.6 V y utiliza el potencial de tierra como referencia. La retroalimentación negativa se implementa mediante una resistencia R7 de 20 k Ω , que permite establecer la ganancia del circuito y asegurar una respuesta lineal dentro del rango de operación. Como resultado, la etapa de salida entrega una señal comprendida aproximadamente entre 0 V y 5 V, proporcional a la temperatura medida por la PT100. En las condiciones de simulación, para una temperatura de 119 °C, el voltaje de salida corresponde a 2.186 V.

5.2.1.2. Ecuación linealización del sensor PT100

El primer paso en el proceso de modelado consiste en establecer una relación lineal entre la señal de salida del circuito de acondicionamiento, descrito en la Sección 5.2.1.1, y la temperatura real medida en el horno. Dado que el sensor PT100 entrega una respuesta en voltaje, es

necesario ajustar los datos experimentales mediante un proceso de linealización que permita representar su comportamiento mediante una ecuación aproximada.

Para ello, se registraron simultáneamente los valores de voltaje proporcionados por el circuito de acondicionamiento y las temperaturas correspondientes, medidas con un sensor de referencia de cocina TP300. A partir de estos datos, se obtuvo una ecuación lineal que relaciona el voltaje de salida (V_{out}) con la temperatura (T), la cual servirá como base para la estimación de la temperatura a partir de las lecturas de voltaje adquiridas durante los ensayos experimentales.

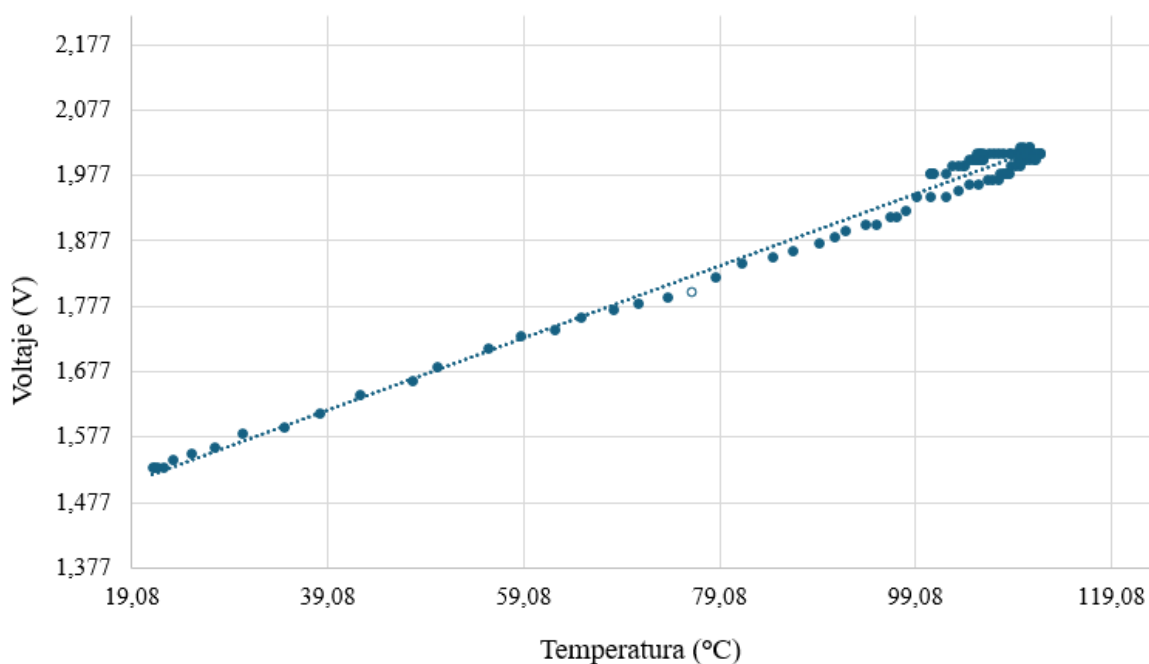


FIGURA 3: Comportamiento térmico del horno durante el proceso de calentamiento y enfriamiento. Tomado de: Autores.

Tal como se muestra en la Figura 3, las mediciones presentan un comportamiento prácticamente lineal en el rango de operación analizado. En la gráfica, el eje Y corresponde al voltaje de salida (V_{out}) obtenido del circuito de acondicionamiento, mientras que el eje X representa la temperatura medida por el sensor de referencia. A partir de estos datos experimentales se realiza un ajuste lineal, del cual se obtiene la ecuación que relaciona el voltaje de salida con la temperatura. Posteriormente, esta ecuación se despeja para expresar la temperatura en función del voltaje, como se muestra a continuación:

$$V_{out} = 0,0055 T + 1,4042 \quad \Rightarrow \quad T = \frac{V_{out} - 1,4042}{0,0055} \quad (5.3)$$

Esta relación permite estimar la temperatura del horno a partir de la señal eléctrica medida, sirviendo como base para el modelado matemático del sistema térmico.

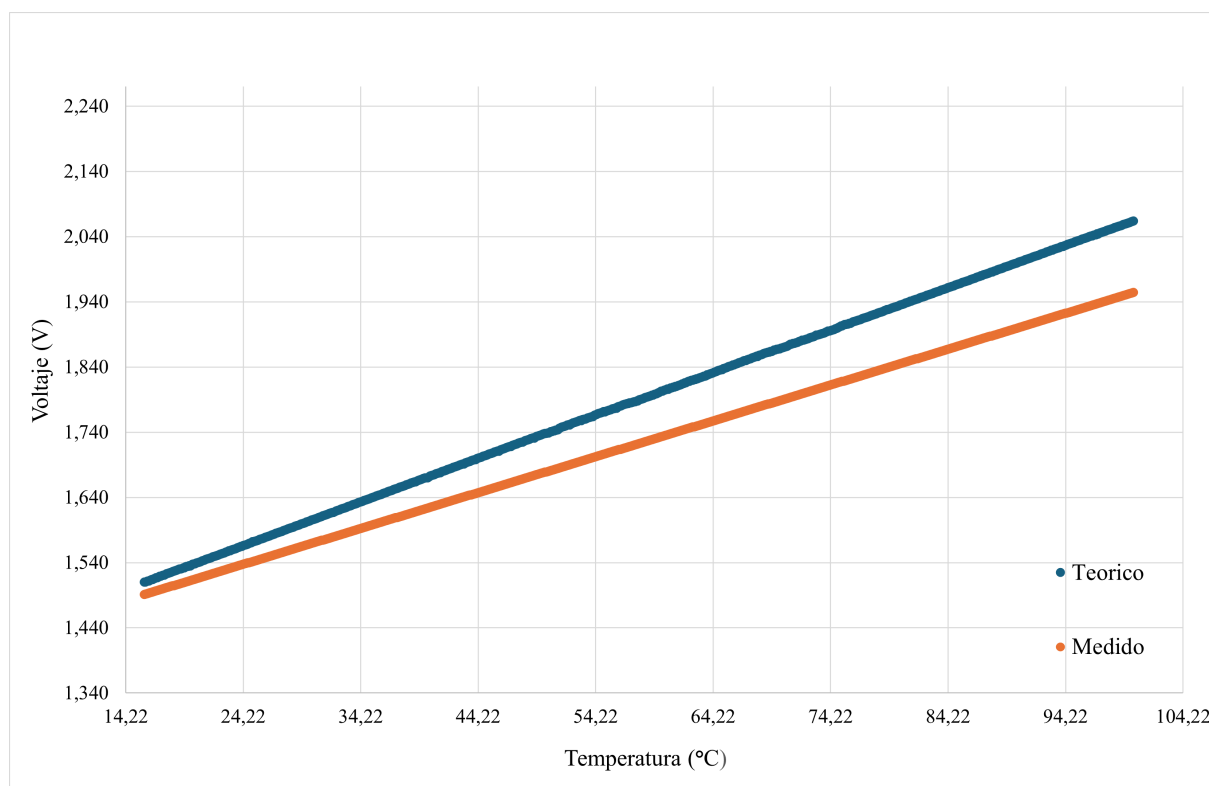


FIGURA 4: Comparación entre los valores teóricos simulados y los valores experimentales. **Tomado de:** Autores.

Para corroborar el funcionamiento del montaje experimental con respecto a los valores obtenidos en la simulación, en la Figura 4 se presenta la comparación entre los datos teóricos obtenidos mediante la simulación del circuito de acondicionamiento mostrado en la Figura 2 y los valores experimentales registrados. En ambos casos se representa la relación entre la temperatura (T) y el voltaje de salida (V_{out}) del circuito. Esta comparación permite evaluar el grado de concordancia entre ambos conjuntos de datos y, a partir de ello, determinar el porcentaje de error utilizando (5.6).

En la gráfica se observa que las curvas correspondientes a los datos simulados y experimentales presentan un comportamiento lineal semejante, con pequeñas desviaciones atribuibles a las tolerancias de los componentes electrónicos, variaciones en la resistencia del sensor PT100 y pérdidas térmicas durante el proceso de medición.

A partir del análisis de los datos obtenidos, se determinaron las ecuaciones lineales que describen la relación entre la temperatura y el voltaje de salida. Para el modelo teórico, la relación se expresa como:

$$V_{\text{out}} = 0,0066 T + 1,4080 \quad \Rightarrow \quad T = \frac{V_{\text{out}} - 1,4080}{0,0066} \quad (5.4)$$

Mientras que, para los datos experimentales, la ecuación correspondiente es:

$$V_{\text{out}} = 0,0055 T + 1,4042 \quad \Rightarrow \quad T = \frac{V_{\text{out}} - 1,4042}{0,0055} \quad (5.5)$$

Estas expresiones permiten estimar la temperatura a partir del voltaje de salida y sirven como base para el cálculo del error relativo entre el modelo teórico y los valores experimentales. Por lo cual para conocer el porcentaje de error se usa la siguiente ecuación:

$$\text{Error (\%)} = \frac{T_{\text{medido}} - T_{\text{teórico}}}{T_{\text{medido}}} \times 100 \quad (5.6)$$

Sustituyendo los valores de $V_{\text{out}} = 2,18 \text{ V}$ en las ecuaciones obtenidas para los modelos medido y teórico, se tiene:

$$T_{\text{teórico}} = \frac{2,18 - 1,4080}{0,0066} = 116,96 \text{ }^{\circ}\text{C}$$

$$T_{\text{medido}} = \frac{2,18 - 1,4042}{0,0055} = 141,05 \text{ }^{\circ}\text{C}$$

Sustituyendo estos resultados en la ecuación del error:

$$\text{Error (\%)} = \frac{T_{\text{medido}} - T_{\text{teórico}}}{T_{\text{medido}}} \quad \Rightarrow \quad \frac{141,05 - 116,96}{141,05} \times 100 = 17,07 \%$$

El error porcentual obtenido, del 17,07 %, entre las temperaturas estimadas teóricamente y las medidas experimentalmente se considera aceptable, dado que las diferencias observadas se

deben principalmente a las características no ideales del sistema real. En la simulación, la PT100 se representa como un sensor lineal ideal, sin errores ni pérdidas, mientras que en el montaje físico influyen factores como el autocalentamiento del sensor, la resistencia de los conductores, el ruido eléctrico y las tolerancias de los componentes del puente de Wheatstone.

Además, el circuito presenta una alta sensibilidad frente a variaciones en la tensión de alimentación: una variación de tan solo 0,1 V en la excitación del puente puede modificar significativamente el voltaje diferencial medido. Esto se debe a que la ganancia del amplificador operacional determinada por una resistencia de retroalimentación de 20 k Ω amplifica notablemente pequeñas desviaciones en la señal de entrada, afectando así la precisión de la medición.

En consecuencia, el error del 17.07% entre los datos teóricos y experimentales se considera aceptable, dado el comportamiento no ideal del sensor y las limitaciones prácticas del circuito de acondicionamiento.

5.2.1.3. Caracterización de Dispositivos y Equipo de Medición

- a) **Raspberry Pi Pico 2:** La Raspberry Pi Pico 2 es una placa pequeña, económica y muy flexible. Trae 4 MB de memoria, puerto Micro-USB para alimentación y programación, y 40 pines, de los cuales 26 son GPIO. En la Tabla 3 se muestran los parámetros más relevantes.

TABLA 3: Características técnicas de la Raspberry Pi Pico 2. Tomado de: [33]

Aspecto	Descripción
Modelo	Raspberry Pi Pico 2
Microcontrolador	RP2350A
Memoria Flash	4 MB
Alimentación	Micro-USB B port
GPIOs	26 pines multifunción, 3 pueden ser ADC
Velocidad del núcleo	Hasta 150 MHz

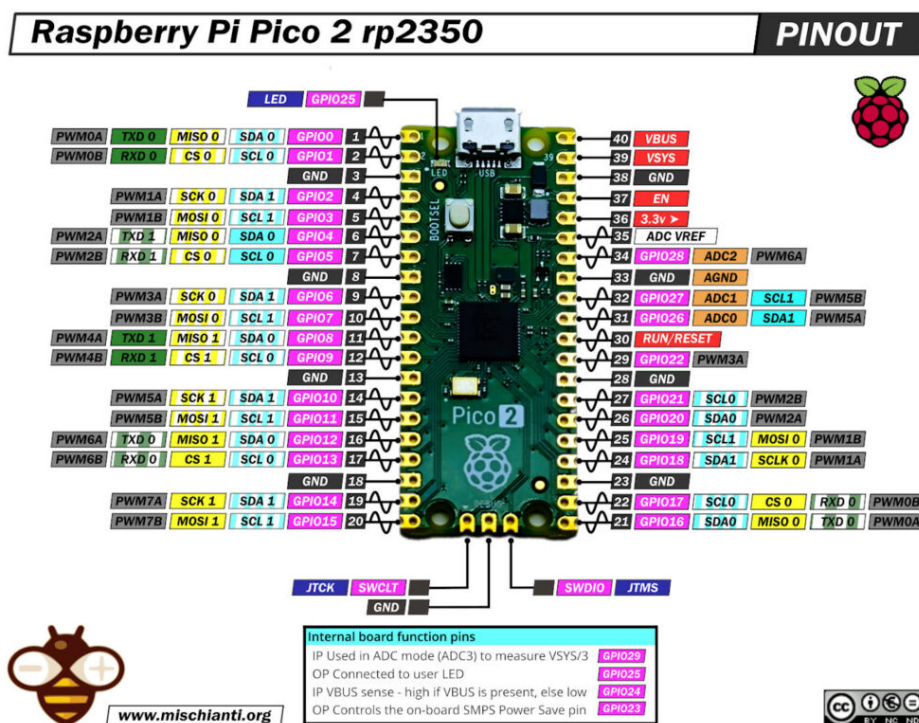


FIGURA 5: Pines Raspberry Pi Pico 2. Tomado de: [34].

Como se evidencia en la Figura 5, la Raspberry Pi Pico 2 dispone de una amplia disposición de pines multifunción que permiten adaptar la placa a diversas aplicaciones de control y adquisición de datos. Entre ellos se destacan cuatro pares de pines UART (TXD y RXD), que facilitan la comunicación serial independiente con distintos dispositivos externos, como módulos RS-485, sensores digitales o sistemas de monitoreo.

Además, incorpora tres canales de conversión analógica (ADC0, ADC1 y ADC2) ubicados en los pines GPIO26, GPIO27 y GPIO28, los cuales permiten la lectura de variables analógicas como temperatura, corriente o voltaje. Esta capacidad de configurar los pines para funciones específicas —ya sea como UART, SPI, I2C, ADC o PWM— brinda una gran flexibilidad en el diseño del sistema embebido, haciendo que la Raspberry Pi Pico 2 sea ideal para proyectos donde se requiere integrar múltiples interfaces de comunicación y sensores de naturaleza diversa.

- b) **Analizador de redes CIRCUTOR CVM-C10:** El Analizador de redes CVM-C10 es un equipo que permite medir, calcular y visualizar parámetros en redes monofásicas, bifásicas o trifásicas. Cuenta con un protocolo de comunicaciones RS-485 para obtener mediciones de voltaje, corriente y potencia instantánea. En la Tabla 4 se muestran los parámetros de alimentación.

TABLA 4: Parámetros de alimentación del Circutor CVM-C10. Tomado de: [35]

Parámetro	Alimentación	Valor
Tensión nominal	AC	95 - 240 V $\sim \pm 10\%$
Frecuencia	AC	50 - 60 Hz
Consumo	AC	4 - 6 VA
Categoría de la instalación	AC	CAT III 300 V
Tensión nominal	DC	105 - 272 V c.c. $\pm 10\%$
Consumo	DC	2 - 6 W
Categoría de la instalación	DC	CAT III 300 V

Como primer parámetro de medida se tiene el voltaje, que corresponde a la tensión fase-neutro. En la Tabla 5 se muestran los rangos de medición para el voltaje del Analizador de redes CVM-C10.

TABLA 5: Parámetros de medición de voltaje del Analizador de redes CVM-C10. Tomado de: [35]

Parámetro	Valor
Tensión nominal	300 V F-N, 520 V F-F
Margen de medida de tensión	5-120 % U_n
Margen de frecuencia	45-65 Hz
Tensión mínima de medida (V_{start})	15 V
Categoría de instalación	CAT III 300 V

Como segundo parámetro de medida se tiene la corriente, necesaria para calcular la potencia. En la Tabla 6 se muestran los rangos de medición de corriente.

TABLA 6: Parámetros de medición de corriente del Analizador de redes CVM-C10. Tomado de: [35]

Parámetro	Valor
Corriente nominal (I_n)	-/5 A, -/1 A
Margen de medida de corriente de fase	2-120 % I_n
Corriente máxima de impulso	100 A
Corriente mínima de medida	10 mA
Consumo máx. en entrada de corriente	0,9 VA
Categoría de instalación	CAT III 300 V

- c) **Transformador de corriente:** Debido a que el horno consume 9 A, es necesario usar un transformador de corriente, ya que el analizador mide máximo 6 A. En la Tabla 7 se observan los parámetros técnicos.

TABLA 7: Datos técnicos del transformador de corriente MBS KBU. Tomado de: [36]

Parámetro	Valor
Corriente térmica nominal continua (I_{cth})	$1,0 \times I_{pr}$
Corriente térmica nominal a corto plazo (I_{th})	$60 \times I_{pr}, 1 \text{ s}$
Tensión máxima de operación (U_m)	0,72 kV
Tensión de prueba de aislamiento (U_{eff})	3 kV, 50 Hz, 1 min
Frecuencia nominal	50 Hz
Clase de aislamiento	E

- d) **Módulo RS-485:** El módulo usa el C.I. MAX485, un transeptor que convierte señales TTL a RS-485, permitiendo comunicación a largas distancias y conexión de varios dispositivos en el mismo bus. En la Tabla 8 se muestran sus especificaciones.

TABLA 8: Especificaciones del módulo RS-485. Tomado de: [37]

Parámetro	Valor
Voltaje de alimentación	5 V
Consumo de corriente máximo	$500 \mu\text{A}$ (máx)
C.I de control	MAX485

- e) **Sensor DHT11:** El DHT11 es un sensor digital que mide temperatura Y humedad del ambiente. Utiliza un termistor NTC y un sensor resistivo para obtener lecturas. En la Tabla 9 se muestran los parámetros y características tecnicas mas relevantes.

TABLA 9: Especificaciones del sensor de humedad y temperatura. Tomado de: [38]

Parámetro	Valor
Rango de temperatura	0°C a 50°C
Precisión de temperatura	$\pm 2^{\circ}\text{C}$
Resolución	1 nivel en la lectura de temperatura
Consumo de corriente (medición)	Aproximadamente 2.5 mA
Consumo de corriente (modo de espera)	Alrededor de $100\text{--}150 \mu\text{A}$
Voltaje de operación	3 – 5.5 V DC
Tiempo de respuesta 6s – 30s (temperatura)	

- f) **Paneles solares:** En el caso de los paneles solares se tiene la referencia HELIOS PLUS 425–450 W / 144 Cells / 9 Bus Bars Monocrystalline Module [39]. En la Tabla 10 se muestran los parámetros y especificaciones mas relevantes.

TABLA 10: Datos eléctricos a Condiciones de Prueba Estándar (STC). Tomado de: [39]

Parámetro	Valor
Potencia de salida (Pmax)	425 [W]
Tolerancia de potencia	0~+5 [W]
Eficiencia del módulo	19.4 [%]
Voltaje a potencia máxima (Vmp)	40.40 [V]
Corriente a potencia máxima (Imp)	10.52 [A]
Voltaje de circuito abierto (Voc)	49.00 [V]
Corriente de cortocircuito (Isc)	11.09 [A]

En [39] se pueden observar los parámetros eléctricos bajo las condiciones de Irradiancia: 1000 W/m^2 , con una temperatura de celda: 25°C y un AM: 1.5, obteniéndose una potencia de salida de 425 W. En la tabla 10 se presentan los valores más relevantes para el dimensionamiento del sistema.

También se cuentan con los datos a una Temperatura Nominal de Operación de la Célula (NMOT), mostrados en la Tabla 11. En estas condiciones se observa una reducción de la potencia de salida, considerando una irradiancia de 800 W/m^2 , una temperatura de celda de 20°C , un AM de 1.5 y una velocidad del viento de 1 m/s.

TABLA 11: Datos eléctricos a NMOT Tomado de: [39]

Parámetro	Valor
Potencia de salida (Pmax)	316.16 W
Voltaje a potencia máxima (Vmp)	38.00 V
Corriente a potencia máxima (Imp)	8.32 A
Voltaje de circuito abierto (Voc)	46.00 V
Corriente de cortocircuito (Isc)	8.93 A

- g) **Inversor:** El inversor seleccionado para el sistema fotovoltaico es el modelo Growatt MIC 1500TL-X, el cual presenta compatibilidad con módulos solares monofásicos. En la Tabla 12 se muestran los parámetros de entrada en DC extraídos de [40].

TABLA 12: Parámetros de entrada (DC) Tomado de: [40]

Parámetro	Valor
Potencia fotovoltaica máxima recomendada (para módulo STC)	2100 W
Tensión máxima de entrada (DC)	500 V
Tensión de arranque	50 V
Tensión nominal	250 V
Rango de tensión en el punto de máxima potencia (MPP)	50 V - 500 V

En la Tabla 12 se describe una tensión de arranque la cual es 50V, este valor se mide en un tablero de paneles solares ubicado en el laboratorio de energías, antes de realizar cualquier prueba se debe asegurar que el voltaje mínimo entregado por los paneles sea mayor a 49V.

TABLA 13: Datos de salida (AC) del inversor MIC 1500TL-X Tomado de: [40]

Parámetro	Valor
Potencia nominal de salida (AC)	1500 W
Potencia aparente máxima (AC)	1500 VA
Tensión nominal de salida (AC)	240 V (por defecto), opción: 208 V monofásico
Frecuencia de red (AC)	50/60 Hz (45–55 Hz / 55–65 Hz)
Corriente máxima de salida	7,1 A
Factor de potencia ajustable	0,8 adelantado a 0,8 atrasado
Distorsión armónica total (THDi)	<3 %
Tipo de conexión a red (AC)	Monofásico

h) **Transformadores 220V:** En la Tabla 14 se muestran los parámetros de los transformadores usados para elevar la tensión de 120V a 220V.

TABLA 14: Especificaciones del transformador elevador de tensión. Tomado de: Autores .

Parámetro	Valor
Tipo de transformador	Elevador (Boost Transformer)
Entrada	Monofásico 110 V
Salida	Monofásico 220 V
Frecuencia	60 Hz
Potencia nominal	1000 W

5.2.1.4. Comunicación Modbus RTU

Se desarrolla la caracterización del comportamiento térmico y eléctrico del horno mediante un sistema de adquisición de datos basado en comunicación Modbus RTU y sensores de temperatura. Este sistema posibilita la obtención de parámetros eléctricos como tensión, corriente y potencia activa, los cuales son esenciales para el análisis energético del horno.

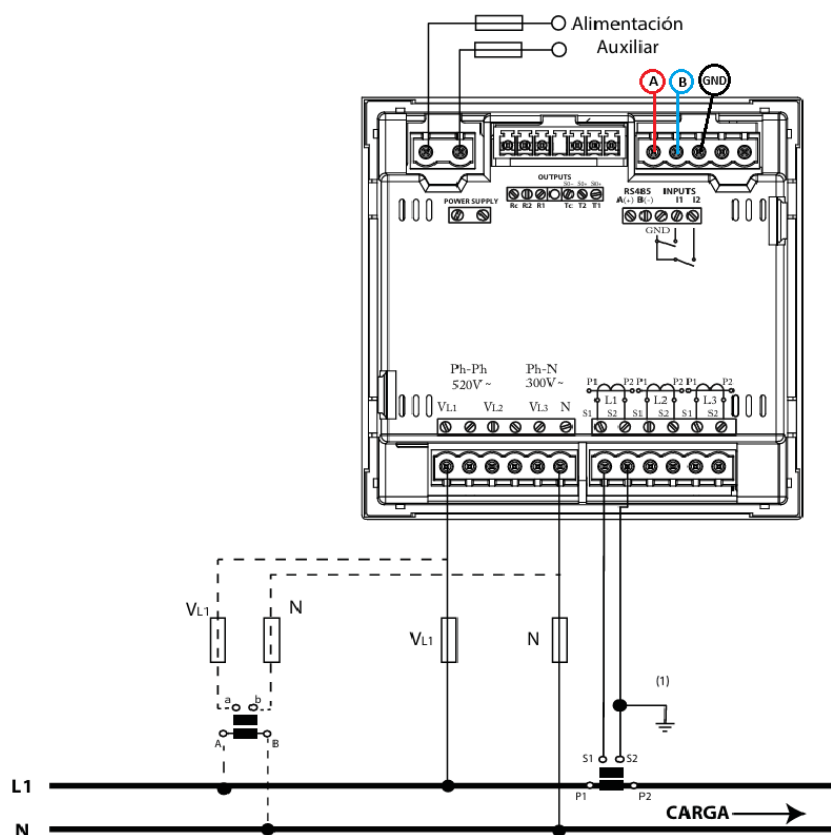


FIGURA 6: Implementación del protocolo Modbus RTU entre el microcontrolador y el analizador de redes. Tomado de: [35].

En la Figura 6 se evidencia como el analizador de redes CIRCUTOR CVM-C10 se configura para operar en una red monofásica de fase a neutro de dos hilos. En esta disposición, tanto el horno como el analizador se alimentan de manera independiente mediante sus respectivas líneas monofásicas, aunque ambos pertenecen al mismo sistema eléctrico. La línea de fase ($L1$) que suministra energía al horno atraviesa el transformador de corriente, cuya salida se conecta al terminal correspondiente del analizador, permitiendo así la medición indirecta de la corriente consumida. El neutro (N) se encuentra referenciado al mismo punto de neutro del sistema, garantizando precisión en la medición sin alterar el funcionamiento del horno.

La comunicación entre el analizador de redes y el sistema de control se realiza mediante el protocolo Modbus RTU, utilizando los terminales diferenciales A y B del analizador, conectados respectivamente a las entradas A y B del módulo RS-485.



FIGURA 7: Conexión del transformador de corriente MBS KBU con el analizador de redes CIRCUTOR CVM-C10. Tomado de: [36].

Para la medición de corriente se utiliza un transformador de corriente tipo MBS KBU, tal como se muestra en la Figura 7. Este componente es fundamental para el sistema, pues, según la Sección 5.2.1.3, el horno tiene una potencia nominal máxima de aproximadamente 1,1 kW, equivalente a una corriente de 9,2 A. Dado que el analizador CVM-C10 admite una corriente máxima de entrada directa de 5 A, se emplea un transformador con relación de transformación 200/5.

Con base en lo anterior se establece el circuito de adquisición de datos, núcleo del sistema de medición diseñado para el horno. Este circuito permite registrar las variables térmicas y eléctricas que determinan su funcionamiento. En el montaje se integran varios componentes que operan de forma coordinada: la Raspberry Pi Pico 2 como unidad central de procesamiento, el sensor de temperatura DHT11 para el monitoreo ambiental, el amplificador operacional LM324 encargado del acondicionamiento de la señal proveniente del sensor PT100 y el módulo de comunicación RS-485, que permite la conexión con el analizador de redes CIRCUTOR CVM-C10.

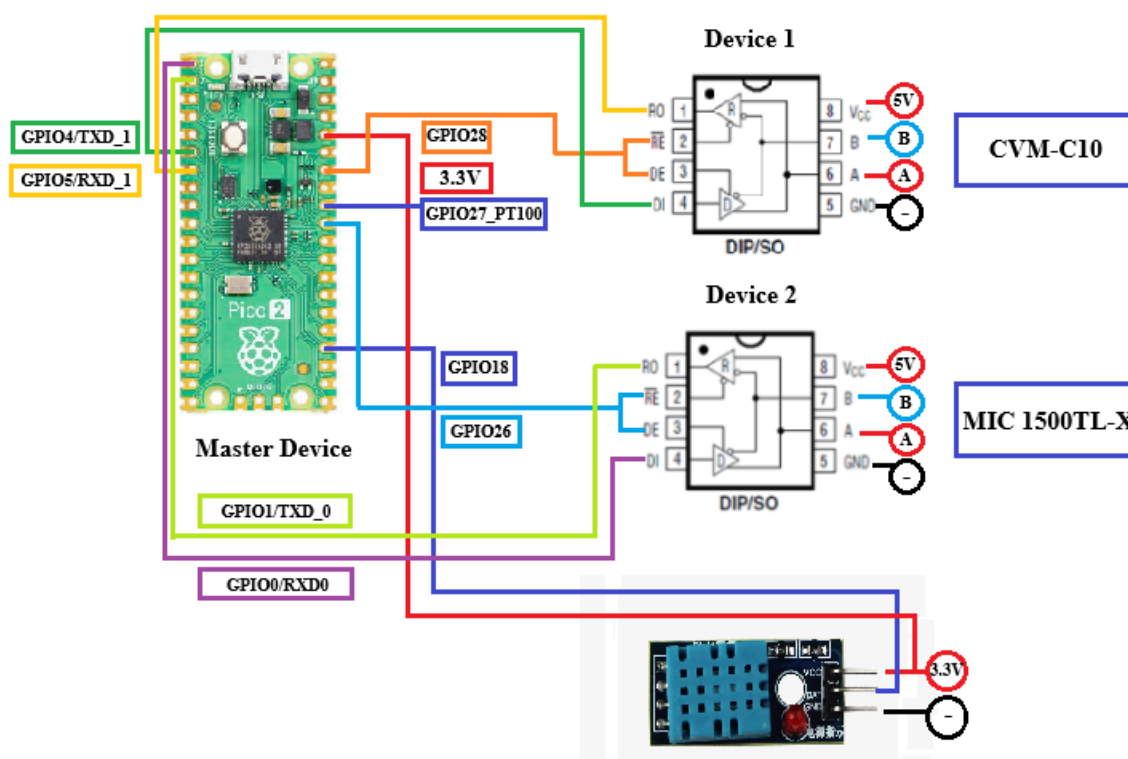


FIGURA 8: Circuito de conexión entre la Raspberry Pi Pico, el sensor DHT11, el amplificador LM324 y los módulos RS485 para lectura del CVM-C10 e Inversor.
Tomado de: Autores.

Como se muestra en la Figura 8, la Raspberry Pi Pico 2 se comunica con dos dispositivos a través de módulos MAX485, que permiten la conversión entre los niveles lógicos TTL y las señales diferenciales RS-485. El primer dispositivo es el analizador de redes CIRCUTOR CVM-C10, que utiliza el protocolo Modbus RTU. La conexión se realiza con el pin RO conectado a GP5 (UART1_TX), el pin DI a GP4 (UART0_RX) y los pines DE y RE controlados desde GP28. Las líneas A y B se enlazan al puerto RS-485 del analizador.

El segundo dispositivo corresponde al inversor MIC 1500TL-X. En este caso, el pin RO se conecta a GP1 (TXD0), el pin DI a GP0 (RXD0) y los pines DE y RE se controlan mediante GP26. En ambas conexiones se comparte la referencia de tierra (GND).

La integridad de las tramas Modbus se verifica mediante el algoritmo CRC-16, que calcula un residuo de verificación a partir de operaciones de desplazamiento y suma binaria sobre todos los bytes del mensaje. Este valor, de 16 bits, se adjunta al final de la trama antes de su envío. Al recibirla, el dispositivo repite el cálculo y compara ambos resultados; si coinciden, se confirma que la información se ha transmitido sin alteraciones.

5.2.1.5. Funcionamiento del sistema de medición

El funcionamiento general del sistema como se evidencia en el Anexo 8.2 inicia con la calibración del ADC de la Raspberry Pi Pico para estabilizar las lecturas y reducir errores. Posteriormente, se adquieren de forma continua tres tipos de datos: la temperatura del horno (pin GP27), la temperatura ambiente y humedad (sensor DHT11 en GP18), y las variables eléctricas del analizador CIRCUTOR CVM-C10 mediante el protocolo Modbus RTU a través del módulo MAX485.

Las mediciones se procesan mediante un filtro estadístico que elimina valores atípicos y aplica promedios móviles para reducir el ruido. Los datos se registran cada 2 segundos en un archivo `datos_sensores.csv`, el cual se sobrescribe en cada ejecución y puede analizarse en plataformas como Excel o MATLAB.

El sistema, programado en MicroPython, integra todos los sensores y rutinas de adquisición, filtrado y almacenamiento. Finalmente, la información obtenida se emplea para el modelado térmico del horno, estableciendo la relación entre la potencia aplicada y la variación de temperatura (ΔT).

5.2.2. Pruebas Experimentales

El objetivo principal del sistema, como se explicó anteriormente, consiste en registrar los valores de temperatura del horno y del ambiente, junto con la potencia eléctrica suministrada al horno en cada nivel de operación. Para ello, se establecieron seis niveles de temperatura de referencia (setpoints), definidos a partir del valor promedio alcanzado una vez que el sistema térmico llegaba a su régimen estable; es decir, cuando las variaciones de temperatura durante el estado transitorio se reducían de manera significativa y los valores permanecían prácticamente constantes.

A continuación, se presentan los niveles de operación establecidos, junto con los valores promedio de temperatura obtenidos para cada uno.

Nivel 1: 104.4°C

Nivel 2: 117.6°C

Nivel 3: 142.1°C

Nivel 4: 163.4°C

Nivel 5: 183.1°C

Nivel 6: 205.3°C

En esta etapa se analizan los datos experimentales con el propósito de establecer la relación entre el comportamiento térmico observado en el horno y el modelo físico de balance energético descrito en (5.1) de la Sección 5.1.1.1. Dicho modelo, ampliamente empleado en sistemas Power-to-Heat, describe la variación de la temperatura como resultado del equilibrio entre la potencia suministrada y las pérdidas térmicas del sistema.

5.2.2.1. Estudio de Datos y Ajuste según los Modelos Matemáticos

A partir de los registros experimentales de temperatura y potencia, se determina el incremento térmico ΔT mediante la siguiente relación:

$$\Delta T = T_{\text{setpoint}} - \bar{T}_{\text{ambiente}} \quad (5.7)$$

Durante el análisis se observa que la potencia promedio del sistema presenta una dependencia directa con el diferencial térmico ΔT , lo que refleja la energía requerida para compensar las pérdidas por conducción, convección y radiación. En este sentido, la relación empírica obtenida entre la potencia promedio y ΔT puede interpretarse como una manifestación práctica del equilibrio energético planteado en (5.1). No obstante, dado que se trata de un modelo ajustado a datos experimentales, la dependencia se describe mediante un ajuste polinómico de segundo orden, el cual aproxima de forma empírica el comportamiento físico del sistema bajo las condiciones reales de operación del horno.

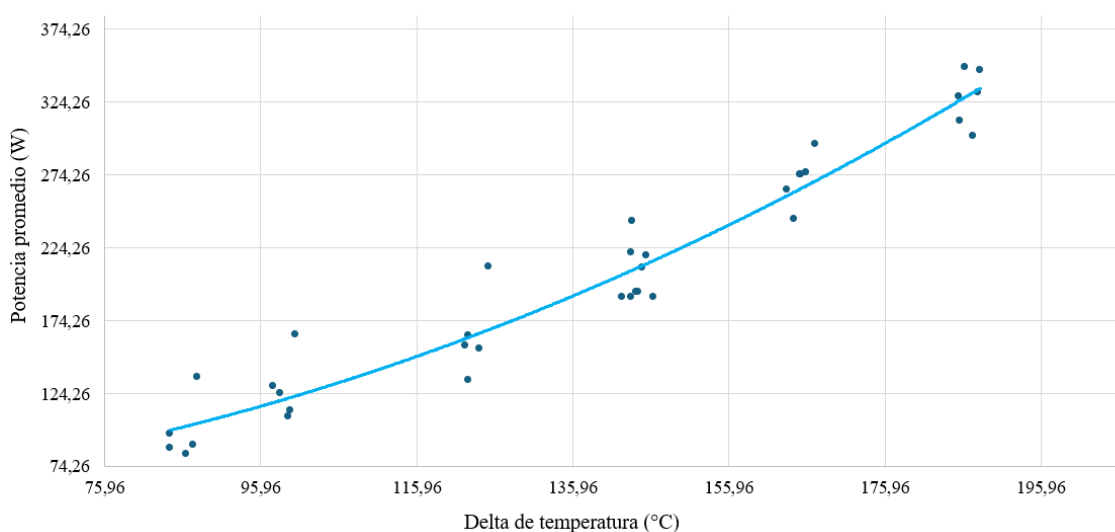


FIGURA 9: Relación entre la potencia promedio y la variación de temperatura (ΔT) del horno. **Tomado de:** Autores.

La Figura 9 evidencia que la potencia promedio suministrada por el horno presenta una tendencia no lineal creciente en función del incremento de temperatura ΔT . Este comportamiento es consistente con los principios de transferencia de calor, ya que, a medida que aumenta la diferencia térmica entre el interior del horno y el ambiente, el sistema requiere una mayor potencia eléctrica para compensar las pérdidas energéticas y mantener la temperatura estable.

5.2.2.2. Ecuación del Modelo Matemático

A partir de los datos experimentales representados en la Figura 9, se realiza un ajuste polinómico de segundo orden que describe la relación existente entre la potencia promedio y el incremento térmico. El modelo resultante se expresa mediante (5.8):

$$\bar{P} = 0,0091 (\Delta T)^2 - 0,2181 (\Delta T) + 52,993 \quad (5.8)$$

La Ecuación (5.8) permite estimar la potencia eléctrica requerida para alcanzar un determinado incremento térmico, constituyendo una herramienta de apoyo en el análisis energético y en la predicción del comportamiento operativo del sistema térmico.

5.2.2.3. Validación del Modelo con los Datos Experimentales

Para validar el modelo térmico del horno se empleó un script en Python (ver Anexo 8.3), el cual ajusta diferentes funciones a los datos experimentales de potencia promedio y temperatura interna. El propósito de esta simulación fue determinar la ecuación que mejor representa el comportamiento energético del sistema y cuantificar la precisión de cada ajuste mediante las métricas de desempeño RMSE y R^2 .

Se evaluaron tres tipos de modelos: lineal, polinómico de segundo orden y exponencial. Cada uno fue calibrado con los datos obtenidos experimentalmente y comparado con los valores medidos. Los resultados de los ajustes se presentan a continuación:

a) Modelo lineal:

$$P(T) = 2,2525 T - 104,4866 \quad (5.9)$$

$$\text{RMSE} = 21,03 \quad \text{y} \quad R^2 = 0,9292$$

Representa una tendencia proporcional entre la potencia y la temperatura, con un ajuste aceptable pero limitado en rangos no lineales.

b) Modelo polinómico de segundo orden:

$$P(T) = 9,088 \times 10^{-3} T^2 - 0,2181 T + 52,9929 \quad (5.10)$$

$$\text{RMSE} = 18,87 \quad \text{y} \quad R^2 = 0,9430$$

Presenta el mejor desempeño, indicando que la potencia crece de forma ligeramente curvada respecto a la temperatura, reflejando las pérdidas térmicas progresivas.

c) Modelo exponencial:

$$P(T) = 40,6343 e^{0,011272T} \quad (5.11)$$

$$\text{RMSE} = 19,25 \quad \text{y} \quad R^2 = 0,9406$$

Describe un comportamiento de crecimiento acumulativo de la potencia, típico de sistemas con disipación térmica dependiente de la temperatura.

Los tres modelos muestran un coeficiente de determinación superior a 0.92, lo cual confirma una alta correlación entre la potencia promedio y la temperatura del horno. No obstante, el modelo polinómico de segundo orden presenta el menor error cuadrático medio, por lo que se adopta como el más representativo para describir la dinámica térmica del sistema.

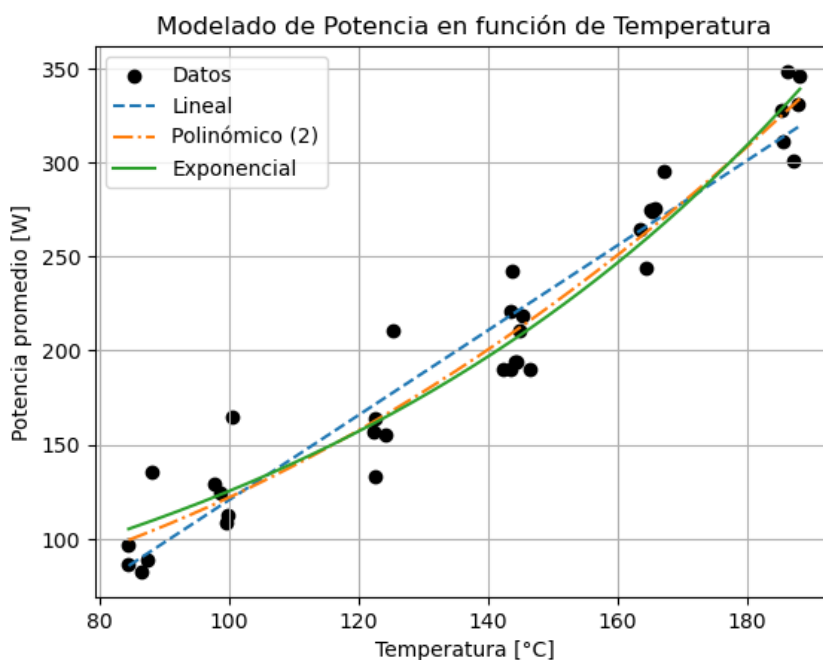


FIGURA 10: Comparación entre los modelos de los datos experimentales del horno. Tomado de: Autores.

La Figura 10 muestra la comparación entre las curvas simuladas y los datos experimentales, evidenciando que tanto el modelo polinómico como el exponencial generan adecuadamente la tendencia real observada durante las pruebas.

5.3. Fase: Implementación de una Microrred Mediante la Integración del Horno

5.3.1. Cargas Disponibles en el Laboratorio de Energías

Para el desarrollo de la microrred, se tiene como primer apartado la clasificación de las cargas disponibles en el laboratorio de energías, dado que en el caso de este proyecto se utiliza el horno disponible y una roseta la cual se usa para alimentar una luminaria.

TABLA 15: Clasificación de las cargas existentes en el Laboratorio de Energías
Tomado de: Autores.

Carga	Descripción	Clasificación
Luminaria 1	Bombillo de 100 W	Carga resistiva
Luminaria 2	Bombillo de 20 W	Carga resistiva
Luminaria 3	Bombillo de 9 W	Carga resistiva
Pistola de calor	Dispositivo de calentamiento eléctrico	Carga resistiva
CPU	Torres fuera de uso	Carga electrónica
TO1705SG-LA(Black+Decker)	Tecnología PtH	TLC
Baterías	Batería plomo-ácido 12V	Almacenador de energía

En la Tabla 15 se presentan las cargas identificadas en el laboratorio. Para la microrred se decidió integrar el horno y las luminarias, teniendo en cuenta la potencia máxima disponible del transformador, no se tiene en cuenta las baterías debido a que se utiliza el horno como un sistema que emula almacenar la energía. Esta selección permite asegurar un funcionamiento adecuado y estable de la microrred.

5.3.2. Conexiones Cargas

El sistema cuenta con dos tipos de cargas utilizadas en las pruebas experimentales:

- **Horno eléctrico:** Se emplea el modelo TO1705SG-LA (Black+Decker), cuyas características se presentan en la Tabla 2. Este equipo opera a una tensión nominal de 120 V AC, con una potencia máxima de 1.15 kW y una corriente de 9.2 A. Su control de temperatura

es mecánico mediante un termostato que permite ajustar seis niveles de calentamiento, alcanzando temperaturas de hasta 205.3 °C. El horno constituye el principal mecanismo Power to Heat (PtH) del sistema, ya que convierte directamente la energía eléctrica proveniente de la microrred en energía térmica útil.

- **Luminarias:** Se dispone de tres luminarias con potencias nominales de 9 W, 20 W y 100 W, las cuales se utilizan para evaluar el comportamiento del sistema bajo diferentes niveles de carga. Estas luminarias permiten analizar la estabilidad del voltaje y la capacidad del inversor para mantener una entrega de energía constante ante variaciones en la demanda.

En la Figura 11 se ilustra el diagrama de conexión correspondiente a las pruebas con una sola carga, donde se utiliza el horno eléctrico junto con una de las luminarias, dependiendo del caso analizado. Esta configuración permite observar el comportamiento del sistema frente a distintas combinaciones de potencia.

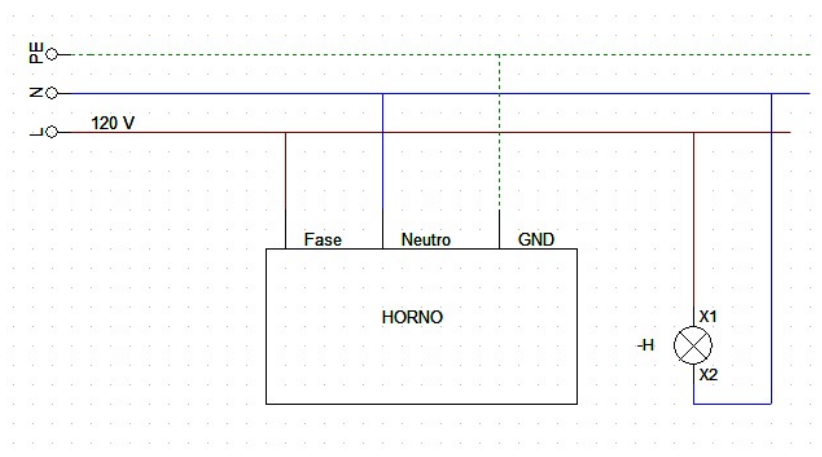


FIGURA 11: Diagrama de conexión de las cargas utilizadas en las pruebas individuales. **Tomado de:** Autores.

Por otro lado, la Figura 12 muestra el diagrama de conexión correspondiente a la prueba con múltiples cargas. En este caso, se implementa una secuencia de conexión de cargas con intervalos de 15 min entre cada cambio, con el fin de evaluar la respuesta dinámica del sistema ante variaciones progresivas en la demanda eléctrica.

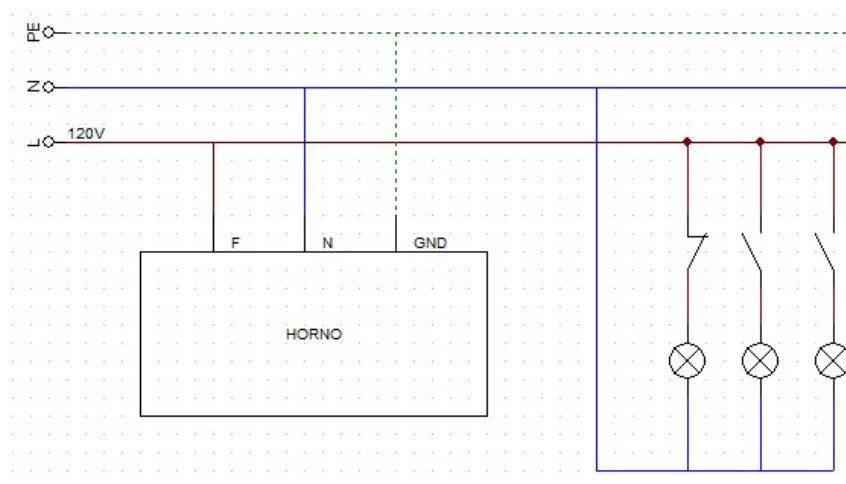


FIGURA 12: Diagrama de conexión para la prueba con varias cargas. Tomado de: Autores.

5.3.3. Integración Microrred

En el contexto del diseño de la microrred propuesta, es fundamental considerar los Recursos Energéticos Distribuidos (DER) disponibles en el Laboratorio de Energías Renovables de la Universidad Santo Tomás, los cuales constituyen la base física del sistema. Estos recursos permiten realizar la conversión, gestión y aprovechamiento de la energía solar fotovoltaica mediante mecanismos Power to Heat (PtH), integrando tanto los elementos de generación como los dispositivos de consumo controlado. A continuación, se describen los principales DER que conforman la infraestructura experimental utilizada en este proyecto.

5.3.3.1. Recursos Energéticos Distribuidos (DER)

Los Recursos Energéticos Distribuidos (DER) implementados en el Laboratorio de Energías Renovables comprenden los siguientes elementos principales:

- **Paneles solares fotovoltaicos:** módulos HELIOS PLUS de 425–450 W, tecnología monocristalina, que actúan como fuente principal de generación renovable.
- **Inversor solar monofásico:** modelo Growatt MIC 1500TL-X, encargado de convertir la energía continua (DC) proveniente de los paneles en energía alterna (AC) para las cargas.
- **Cargas eléctricas controladas:** entre ellas un horno eléctrico (cuyas especificaciones se detallan en la Sección 5.1.2) y tres luminarias de distintas potencias (9 W, 20 W y 100 W), empleadas en los ensayos de respuesta térmica y consumo energético.

En la Sección 5.2.1.3 se presentan las características técnicas de los módulos fotovoltaicos HELIOS PLUS, que constituyen la fuente primaria de generación en la microrred. Cada panel proporciona una potencia máxima de 425 W bajo condiciones estándar (STC), con una eficiencia del 19.4%, una tensión en el punto de máxima potencia (V_{mp}) de 40.4 V y una corriente correspondiente (I_{mp}) de 10.52 A.

El sistema utiliza dos módulos conectados en paralelo, de modo que la tensión se mantiene constante mientras la corriente total se incrementa a aproximadamente 21 A, alcanzando una potencia combinada cercana a 850 W. Esta configuración permite satisfacer los niveles de tensión y corriente requeridos por el inversor solar Growatt MIC 1500TL-X.

El inversor admite una potencia fotovoltaica máxima recomendada de 2100 W y opera con un rango de tensión en el punto de máxima potencia entre 50 V y 500 V DC, con una tensión nominal de 250 V DC. En su salida entrega una tensión alterna monofásica de 240 V AC, adecuada para la alimentación del sistema experimental del laboratorio.

5.3.3.2. Configuración Eléctrica de la Microrred

El sistema se conecta a la red de suministro del laboratorio a través de un primer transformador elevador de 110 V a 220 V, el cual alimenta el inversor solar. Este dispositivo convierte la energía fotovoltaica en corriente alterna, con una tensión nominal de 240 V, compatible con las especificaciones de salida del equipo.

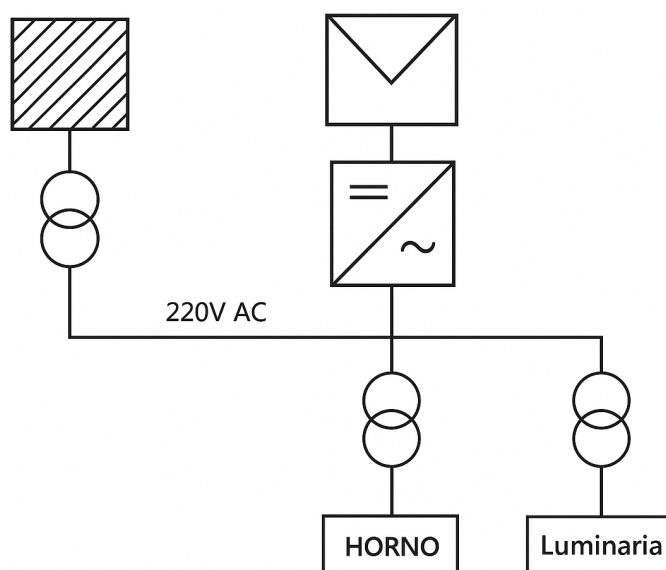


FIGURA 13: Diagrama Microrred. Tomado de: Autores.

Posteriormente, la energía proveniente del inversor se dirige hacia un tablero de distribución, desde donde se conecta un segundo transformador (utilizado de manera inversa) para reducir la tensión de 220 V a 110 V. Esta etapa permite alimentar las cargas de baja tensión disponibles en el laboratorio, tales como el horno eléctrico y las luminarias. La Figura 13 muestra la estructura general de conexión de estos elementos.

5.3.3.3. Comunicación Growatt MIC 1500TL-X

Como se explica en la Sección 5.2.1.4, el esquema completo de comunicación entre el microcontrolador Raspberry Pi Pico 2 y el inversor se realiza mediante un módulo RS-485, del mismo modo que con el analizador de redes. Para establecer correctamente esta conexión, es necesario conocer la disposición de pines y la forma en que el inversor transmite y recibe las señales.

NO.	Definition		NO.	Definition	
1	N/A	This Pin is no signal	5	CT-P	Signal for export Limitation (Optional)
2	N/A		6	CT-N	
3	RS485A1	Signal for communication	7	RS485A2	Signal for Smart
4	RS485B1		8	RS485B2	

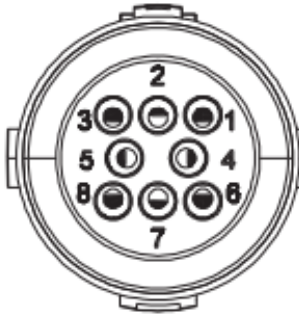


FIGURA 14: Conexión cable de señales MIC 1500TL-X. Tomado de: [40].

En la Figura 14 se muestra la distribución de pines de señales del inversor MIC 1500TL-X, el cual cuenta con un total de ocho pines visibles. Para la comunicación con la Raspberry Pi Pico 2 solo se emplean los pines 3 y 4, que establecen el enlace RS-485 mediante el módulo MAX485, tal como se explicó en la Sección 5.2.1.4. A través de esta conexión, el sistema permite la lectura de variables eléctricas del inversor, como la potencia generada por los paneles y el voltaje de salida.

Los demás pines no se utilizan en esta configuración. Sin embargo, los pines 7 y 8 también pueden emplearse para comunicación RS-485 con un medidor inteligente; el pin 7 corresponde a *A2* y el pin 8 a *B2*. Finalmente, los pines 5 y 6, identificados como *CT-P* y *CT-N*, permiten la conexión de un transformador de corriente (CT) externo que el inversor utiliza para limitar el flujo de energía hacia la red eléctrica, en aplicaciones de autoconsumo o cuando se busca evitar el vertido de excedentes.

5.3.3.4. Integración del inversor fotovoltaico y monitoreo completo de la microrred

El código ampliado, presentado en el Anexo 8.4, extiende las funcionalidades del sistema de medición para incorporar la lectura y supervisión del inversor fotovoltaico Growatt MIC 1500TL-X, junto con las variables de temperatura y humedad previamente integradas. La Raspberry Pi Pico 2 se configuró como maestro en una red Modbus RTU sobre UART, utilizando el módulo MAX485. El inversor Growatt opera como esclavo con ID = 5, mientras que el analizador de redes CIRCUTOR CVM-C10 se mantiene en ID = 10. Ambos dispositivos comparten la misma línea de comunicación diferencial, con una velocidad configurada en 9600 baudios, paridad N, y un intervalo de consulta de 2 segundos entre lecturas.

El código, desarrollado en MicroPython, se estructura en bloques funcionales definidos por tareas concurrentes:

- **Inicialización y configuración:** Se establecen los parámetros de comunicación UART y las direcciones de los registros Modbus de cada dispositivo. Asimismo, se configuran los pines de entrada analógica para la medición de la temperatura del horno mediante el ADC interno (pin GP27) y la adquisición de temperatura y humedad ambiente a través del sensor DHT11 (pin GP18).
- **Lectura del inversor Growatt:** Se implementa una rutina de consulta periódica que lee los registros Modbus asociados a la tensión de entrada, corriente del generador fotovoltaico, potencia generada y estado operativo del inversor. Los datos se convierten a magnitudes físicas mediante factores de escala definidos por el fabricante.
- **Adquisición de variables locales:** De forma simultánea a la lectura Modbus, el sistema mide la temperatura ambiente y la humedad relativa por medio del sensor DHT11, y la temperatura del horno a través del ADC. Los valores son filtrados mediante un promedio móvil con el fin de reducir el ruido y estabilizar las mediciones.
- **Procesamiento y almacenamiento:** Todas las variables obtenidas se registran de manera estructurada en un archivo `datos_microrred.csv`, el cual se sobrescribe en cada ejecución. Este archivo consolida la información de tensión, corriente, potencia, temperatura y humedad, facilitando su posterior análisis en MATLAB o Excel.
- **Supervisión térmica del horno:** Se mantiene la función de modelado térmico descrita en la Sección 8.2, que estima la variación de temperatura en función del tiempo y la potencia aplicada, emulando el comportamiento de almacenamiento energético virtual basado en la inercia térmica.

En conjunto, este desarrollo permite centralizar el monitoreo de la microrred en un único sistema embebido, integrando la información proveniente del inversor, del analizador de redes y de los sensores locales. La estructura modular del código facilita la expansión futura hacia esquemas de control más avanzados o la incorporación de nuevos dispositivos Modbus.

5.3.3.5. Conexiones física y Eléctrica

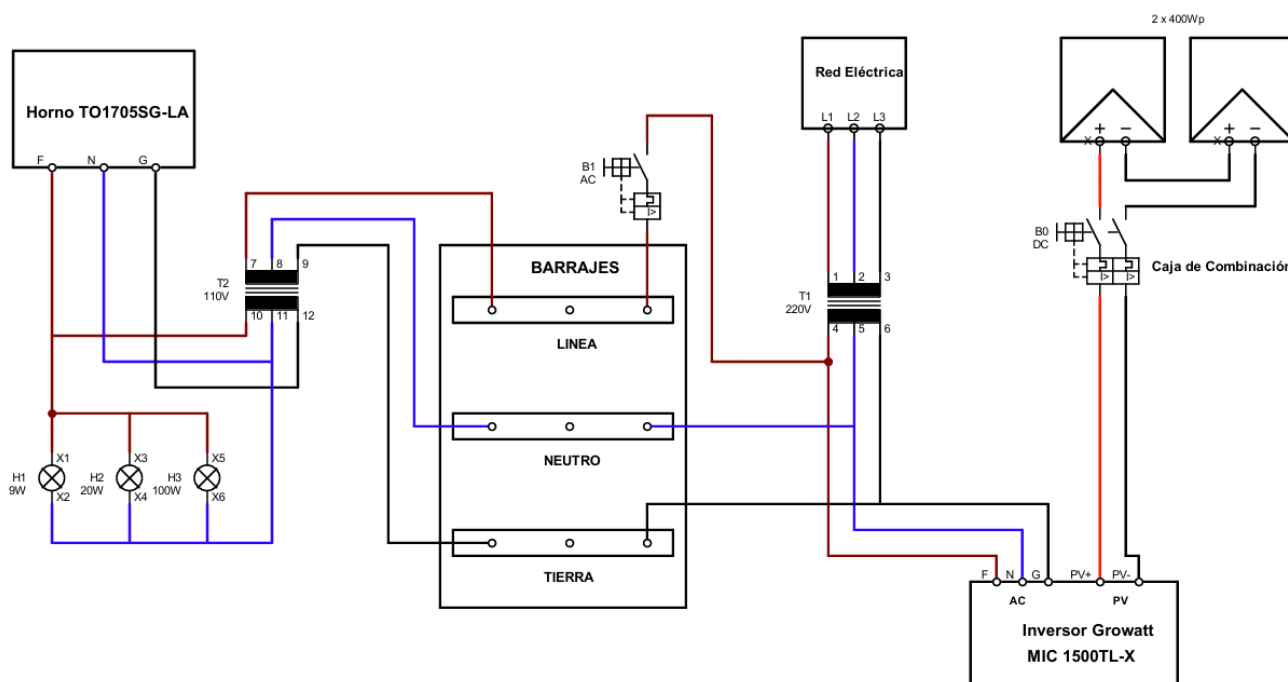


FIGURA 15: Esquemático de potencia de la microrred. **Tomado de:** Autores.

En la Figura 15 se muestra el esquema general de potencia de la microrred implementada. De derecha a izquierda, se observan los dos módulos fotovoltaicos de 400 Wp cada uno, cuya conexión en paralelo conforma el conjunto identificado como PV1. Estos paneles convergen en una caja de combinación, donde se agrupan las salidas positivas y negativas antes de ingresar al inversor solar Growatt MIC 1500TL-X.

En la parte posterior del inversor se encuentran las terminales rotuladas como $PV+$ y $PV-$, a las cuales se conectan los polos positivo y negativo provenientes de la caja de combinación. Para su funcionamiento, el inversor debe estar acoplado a la red eléctrica, ya que requiere una referencia de tensión alterna. Según sus especificaciones técnicas, opera con una tensión nominal de 240 V AC, aunque admite un rango de operación que permite su uso con 220 V AC. Por esta razón, se emplea un transformador elevador de 110 V a 220 V AC, cuya salida se conecta a las terminales de entrada del inversor, identificadas como F (fase), N (neutro) y G (tierra).

Posteriormente, la salida del inversor se conecta al tablero eléctrico principal del laboratorio, el cual cuenta con barras de distribución para línea, neutro y tierra, así como con interruptores termomagnéticos para la protección de cada circuito. Desde este tablero se deriva la conexión hacia un segundo transformador reductor de 220 V a 110 V AC, destinado a la alimentación de las cargas del sistema.

Entre las cargas conectadas se encuentra un horno eléctrico, cuyas líneas F , N y G se conectan directamente a la salida del transformador, y tres luminarias de diferentes potencias (9 W, 20 W y 100 W). Aunque en el esquema las luminarias aparecen dispuestas en paralelo, durante las pruebas se realizaron mediciones individuales para cada una de ellas con el fin de evaluar su comportamiento eléctrico por separado.

5.3.3.6. Tablero Eléctrico

Para el tablero de conexiones se utiliza el tablero eléctrico diseñado en [41], el cual cumple con los requerimientos del Reglamento Técnico de Instalaciones Eléctricas (RETIE).

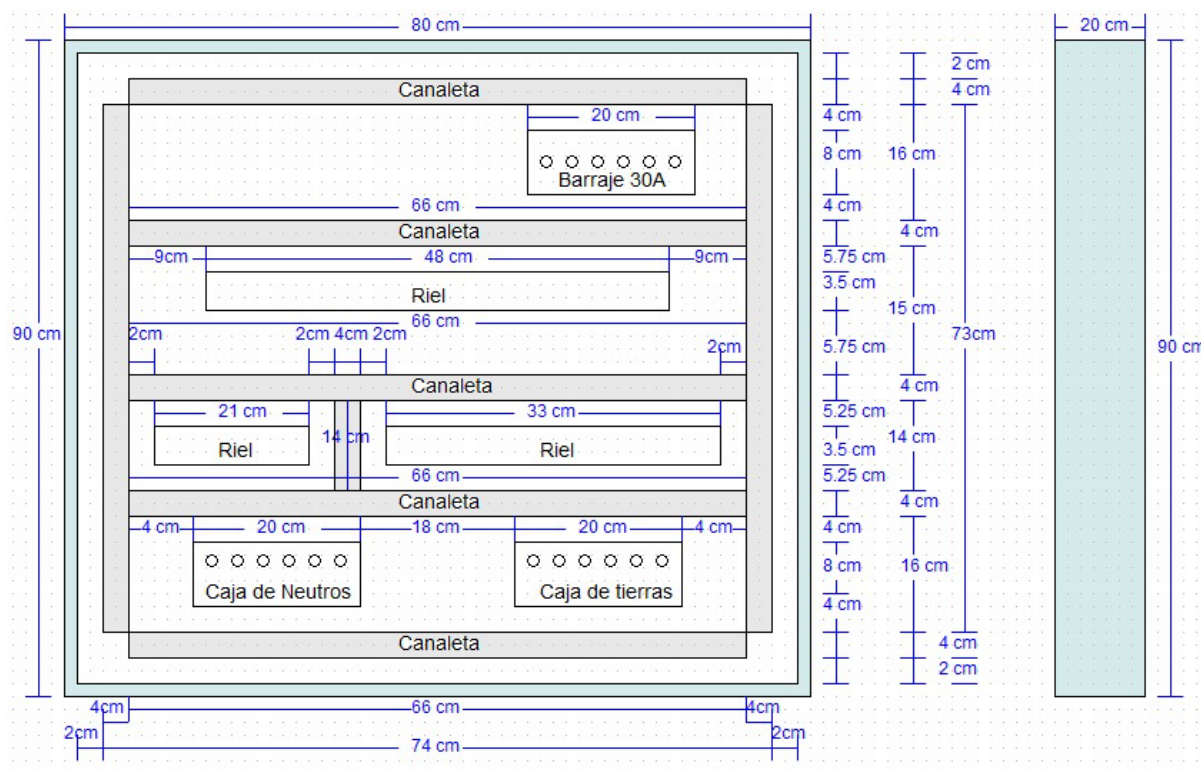


FIGURA 16: Distribución del tablero de potencia. Tomado de: [41].

En la Figura 21, se muestra como esta diseñado el tablero, teniendo 3 barrajes principales los cuales son la tierra, neutro y fase, el voltaje que se debe tener en la red son 220V según el manual del inversor MIC1500TL-X.

5.3.3.7. Cableado y Protecciones

Al momento de elegir el cable y las protecciones se calcula el consumo de los elementos en la microrred, utilizando los valores nominales para el dimensionamiento de las protecciones y los cables.

Teniendo los valores nominales de operación se tiene un factor de sobre dimensionamiento del 25 %. El cual se muestra en las ecuaciones que van desde (5.12) a (5.17) .

$$I_{AC_{Horno}} = \frac{1000[W]}{120[V]} = 8,3[A] * 1,25 = 10,375[A] \quad (5.12)$$

$$I_{AC_{TrafoPrim}} = \frac{1000[W]}{120[V]} = 8,3333[A] * 1,25 = 10,375[A] \quad (5.13)$$

$$I_{AC_{TrafoSec}} = \frac{1000[W]}{240[V]} = 4,166[A] * 1,25 = 5,2075[A] \quad (5.14)$$

$$I_{AC_{Lamp1}} = \frac{100[W]}{240[V]} = 0,83[A] * 1,25 = 1,041[A] \quad (5.15)$$

$$I_{AC_{Lamp2}} = \frac{20[W]}{120[V]} = 0,16[A] * 1,25 = 0,2[A] \quad (5.16)$$

$$I_{AC_{Lamp3}} = \frac{9[W]}{120[V]} = 0,075[A] * 1,25 = 0,0093[A] \quad (5.17)$$

Para saber cual es la corriente entregada por la red eléctrica se debe sumar los elementos de consumo que en este caso se tiene el horno y las diferentes luminarias, debido a esto en (5.18) se evidencia el valor de una corriente de 11.625A.

$$I_{AC_{Red}} = 10,375[A] + 1,041[A] + 0,2[A] + 0,0093[A] = 11,625[A] \quad (5.18)$$

Como se muestra en (5.18) se tienen las diferentes luminarias, en las de menor potencia se evidencia un menor consumo que es casi despreciable, pero se tiene en cuenta al momento de elegir el calibre.

Teniendo en cuenta lo anterior se selecciona un cable American Wire Gauge (AWG) calibre 14, el cual para este caso es el que se utiliza para la entrada AC del inversor donde se conecta la Red eléctrica.

5.3.3.8. Cableado DC

Para el cableado en DC, solamente se usa en la conexión de los paneles solares y el inversor, en (5.19) se muestra el valor de la corriente máxima entregada por los paneles.

$$I_{DC_{panel}} = \frac{1500[W]}{500[V]} = 4,2[A] * 1,25 = 5,25[A] \quad (5.19)$$

Teniendo en cuenta el valor de 5.25A se puede utilizar el mismo cable AWG calibre 14 para la conexión en DC de los paneles a la entrada positiva y negativa de PV del inversor.

5.3.4. Montaje Experimental

En las Figuras 15, 21, 8, 7 y 6 se presentan los componentes físicos que conforman la microrred. En estas se observan las conexiones de potencia entre los paneles fotovoltaicos, el inversor, los transformadores y las cargas, el tablero con los barrajes de línea, neutro y tierra, la comunicación con la Raspberry Pi, el transformador de corriente del analizador de redes y la conexión de potencia del analizador con el horno.

Como se evidencia en la Figura 15, de derecha a izquierda se muestra la conexión general de potencia de la microrred. En la parte derecha se encuentran los módulos fotovoltaicos conectados al tablero de combinación, seguidos del inversor fotovoltaico, el tablero de protección y distribución, el transformador reductor, las cargas (PTH y resistivas), y finalmente el sistema de adquisición y comunicación.

Generación fotovoltaica y tablero de combinación:

El sistema fotovoltaico está conformado por dos conjuntos de módulos solares. En el lado izquierdo del tablero de combinación se encuentran dos paneles de 400 Wp conectados en serie, mientras que en el lado derecho se ubican dos paneles de 250 Wp también en serie. Para el desarrollo experimental se emplearon los módulos de 400 Wp, cuya interconexión se observa en la Figura 17. En esta etapa se agrupan y protegen las salidas de corriente continua mediante la caja de combinación, que centraliza la potencia proveniente de los arreglos fotovoltaicos.

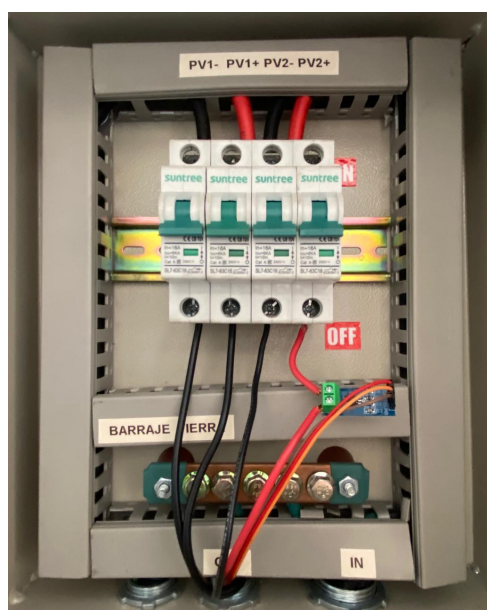


FIGURA 17: Tablero de combinación de los módulos fotovoltaicos. **Fuente:** Autores.

Etapa de conversión (Inversor fotovoltaico):

Dado que las cargas del sistema operan a 120 V, mientras que el inversor requiere una alimentación de 220 V, se utilizó un transformador elevador que convierte la tensión de 110 V a 220 V AC a 60 Hz. La conexión del inversor y su alimentación mediante el transformador pueden apreciarse en la Figura 18. En el momento de la fotografía, el inversor aún no se encontraba sincronizado con la red, proceso que tarda aproximadamente cinco minutos antes de iniciar la inyección de energía.

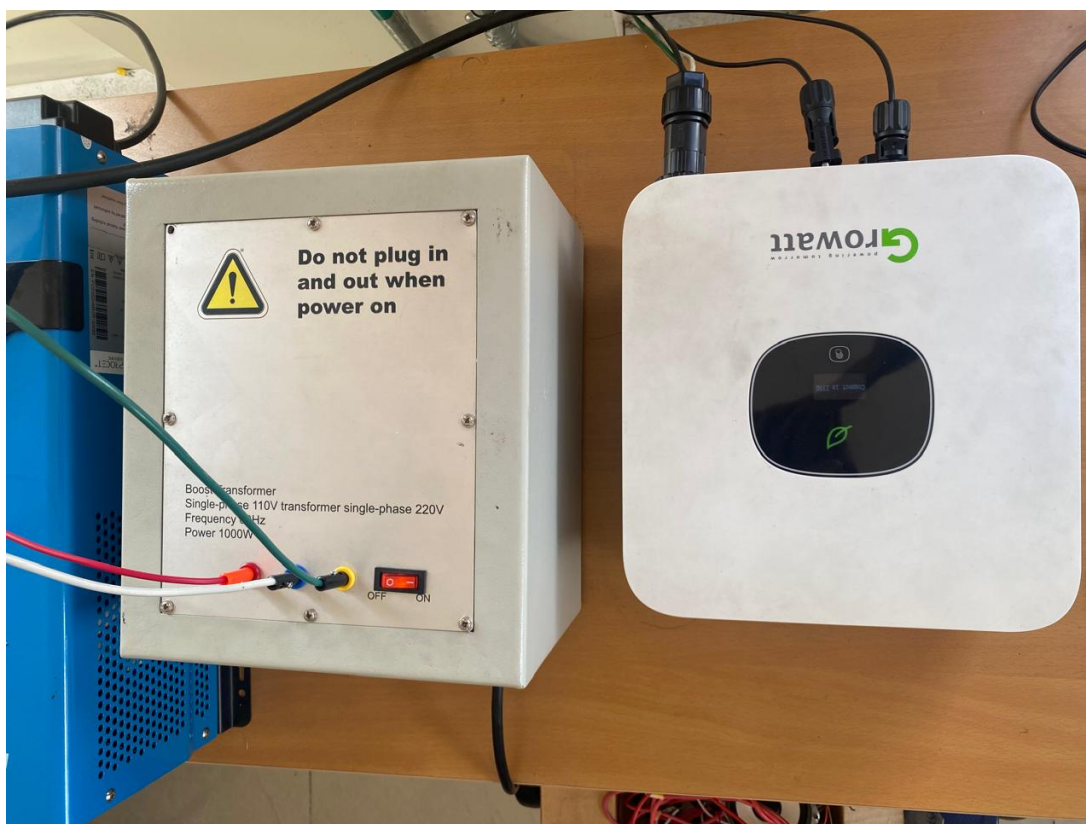


FIGURA 18: Inversor fotovoltaico alimentado mediante transformador elevador de 110 V a 220 V. **Fuente:** Autores.

Una vez completada la sincronización, el sistema opera de manera integrada, como se observa en la Figura 19, donde se visualizan el tablero de paneles, el transformador y el inversor. En este estado, la hoja de señalización del inversor se ilumina en color verde, indicando su conexión estable con la red.



FIGURA 19: Integración del tablero fotovoltaico, transformador e inversor sincronizado con la red (indicador verde). Fuente: Autores.

Por el contrario, cuando el inversor no dispone de tensión o irradiación suficiente, el indicador cambia a color rojo, como se muestra en la Figura 20, evidenciando la falta de sincronización



FIGURA 20: Estado del inversor sin sincronización por falta de tensión o irradiancia (indicador rojo). **Fuente:** Autores.

Tablero de protección y distribución:

El tablero general de protección y distribución, que se evidencia en la Figura 21, integra los interruptores, barrajes y borneras de conexión que permiten la organización del sistema eléctrico. Desde este punto se distribuye la energía hacia las diferentes cargas experimentales y equipos de monitoreo, garantizando una operación segura y ordenada de la microrred.

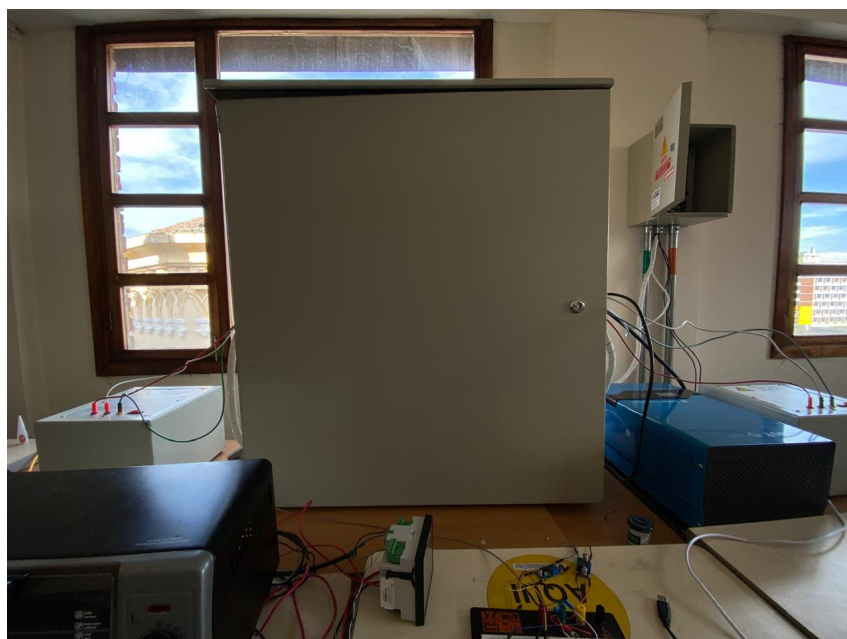


FIGURA 21: Tablero general de protección y distribución de la microrred. **Fuente:** Autores.

Etapa de cargas:

Las cargas se alimentan a través de un transformador reductor que convierte la tensión de 220 V a 110 V, como se ilustra en la Figura 22. Este componente permite operar diferentes dispositivos eléctricos a un nivel de tensión seguro para las prueba experimentales.

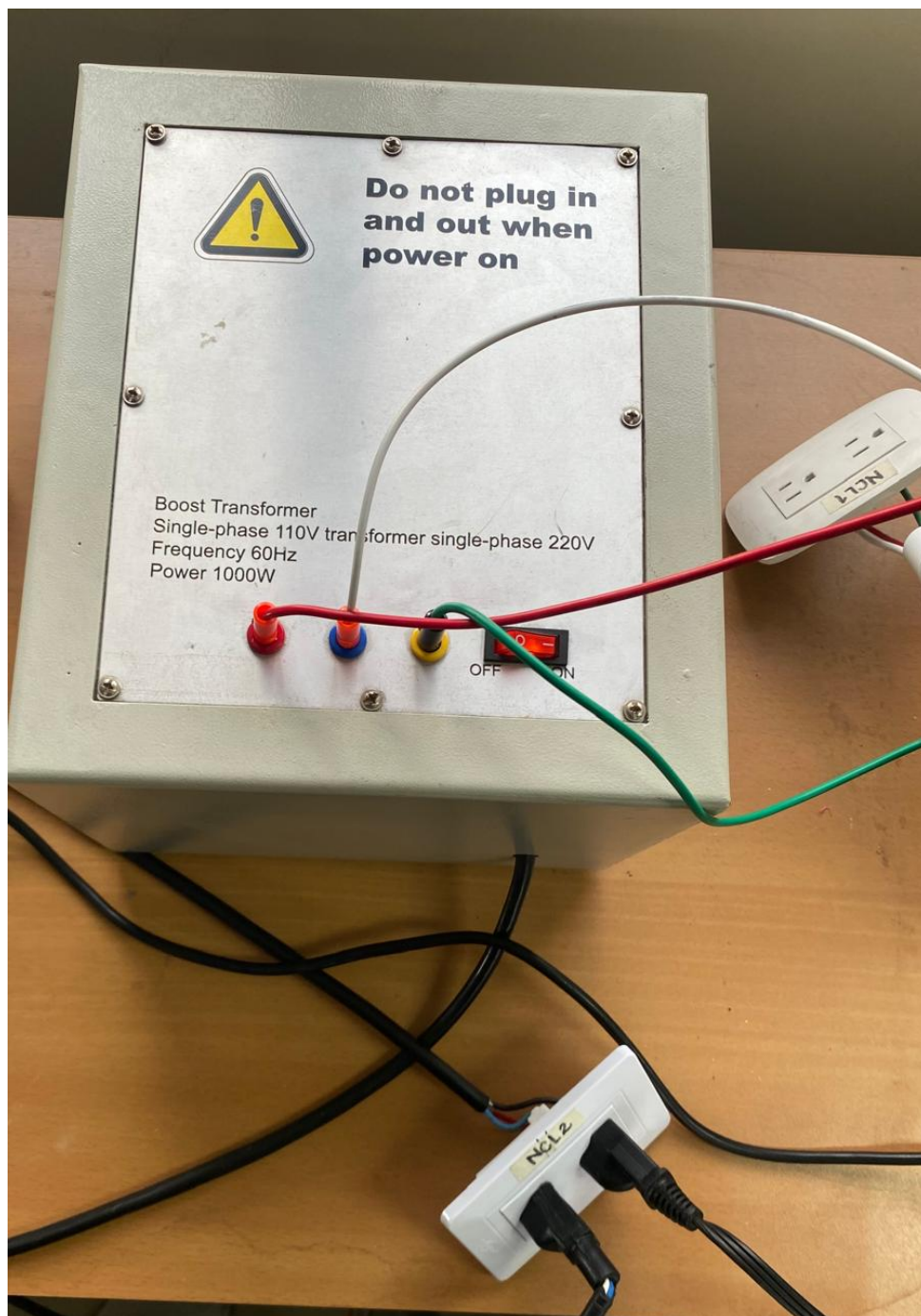


FIGURA 22: Transformador reductor de 220 V a 110 V para alimentación de cargas. Fuente: Autores.

Entre las cargas utilizadas se encuentra un horno eléctrico (Figura 23), que actúa como una carga variable tipo PTH (*Power-to-Heat*). Este dispositivo convierte la energía eléctrica en calor, permitiendo emular la respuesta térmica de un sistema con almacenamiento energético.



FIGURA 23: Horno eléctrico utilizado como carga variable tipo PTH. Fuente: Autores.

Asimismo, se emplearon para las pruebas luminarias resistivas de 9 W, 20 W y 100 W, mostradas en la Figura 24. Estas cargas estáticas se conectan mediante rosetas con polaridad positiva y negativa, permitiendo realizar pruebas de diferentes niveles de consumo.



(a) Carga de 9 W

(b) Carga de 20 W

(c) Carga de 100 W

FIGURA 24: Luminarias resistivas de 9 W, 20 W y 100 W empleadas como cargas estáticas. Fuente: Autores.

En la Figura 25 se observa una de las clavijas utilizadas para conectar las cargas al transformador que entrega 110V, facilitando el montaje y garantizando seguridad eléctrica durante los ensayos.

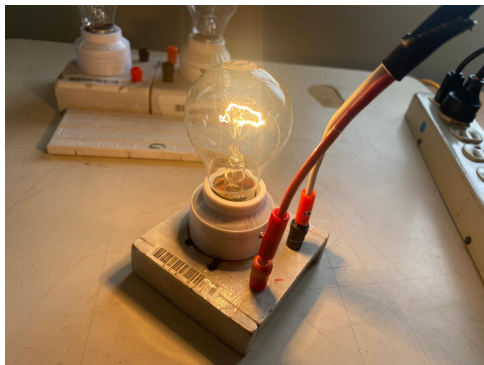


FIGURA 25: Clavija utilizada para la conexión de las cargas al tablero. **Fuente:** Autores.

Sistema de comunicación y monitoreo:

En la Figura 26 se aprecian los componentes conectados que conforman la interfaz de comunicación descrita en la Sección 5.2.1.4. En esta se observan los módulos RS-485, encargados de establecer la comunicación entre el inversor, el analizador de redes y el sistema embebido basado en Raspberry Pi Pico 2. Asimismo, se distingue el circuito de acondicionamiento de señal del sensor de temperatura PT100, utilizado para registrar la temperatura del horno durante las pruebas experimentales.

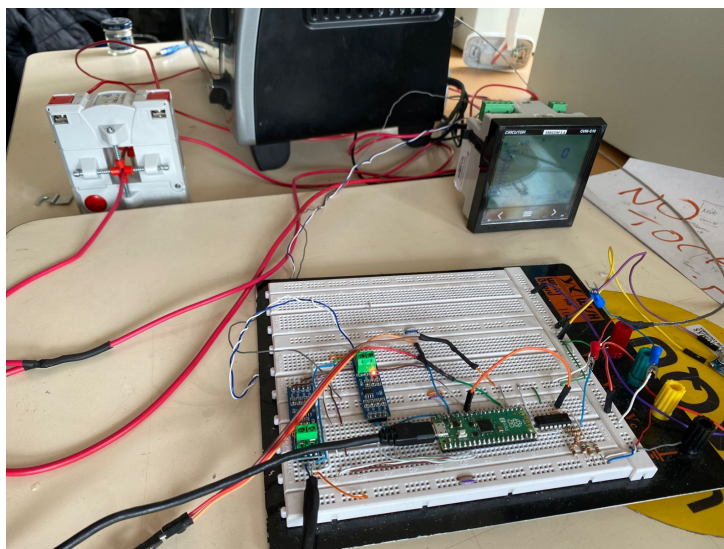


FIGURA 26: Sistema de comunicación RS-485 y adquisición de datos con Raspberry Pi Pico 2. **Fuente:** Autores.

5.3.4.1. Operación de la Microrred

Durante la operación, los paneles solares generan energía eléctrica que es convertida por el inversor en corriente alterna. La energía es luego distribuida mediante los transformadores hacia las cargas conectadas, simulando diferentes escenarios de consumo. El horno eléctrico actúa como una carga variable de alta potencia (mecanismo PtH), mientras que las luminarias representan cargas ligeras y constantes.

De esta forma, el sistema permite realizar pruebas controladas de gestión energética, eficiencia de conversión, y comportamiento dinámico del inversor ante distintos perfiles de carga, contribuyendo al estudio y desarrollo de microrredes solares híbridas dentro del laboratorio.

5.3.5. Comportamiento Variación de Demanda y Generación

En este apartado, la prueba consistió en mantener el horno en funcionamiento de forma continua, sin apagarse en ningún momento durante el experimento. Cada 15 minutos se realizaba un cambio en la carga de las luminarias, siguiendo el orden de encendido: primero la luminaria de 100 W, luego la de 20 W, posteriormente la de 9 W, y así sucesivamente. Este ciclo de cambios se repitió de manera secuencial hasta completar todas las mediciones.

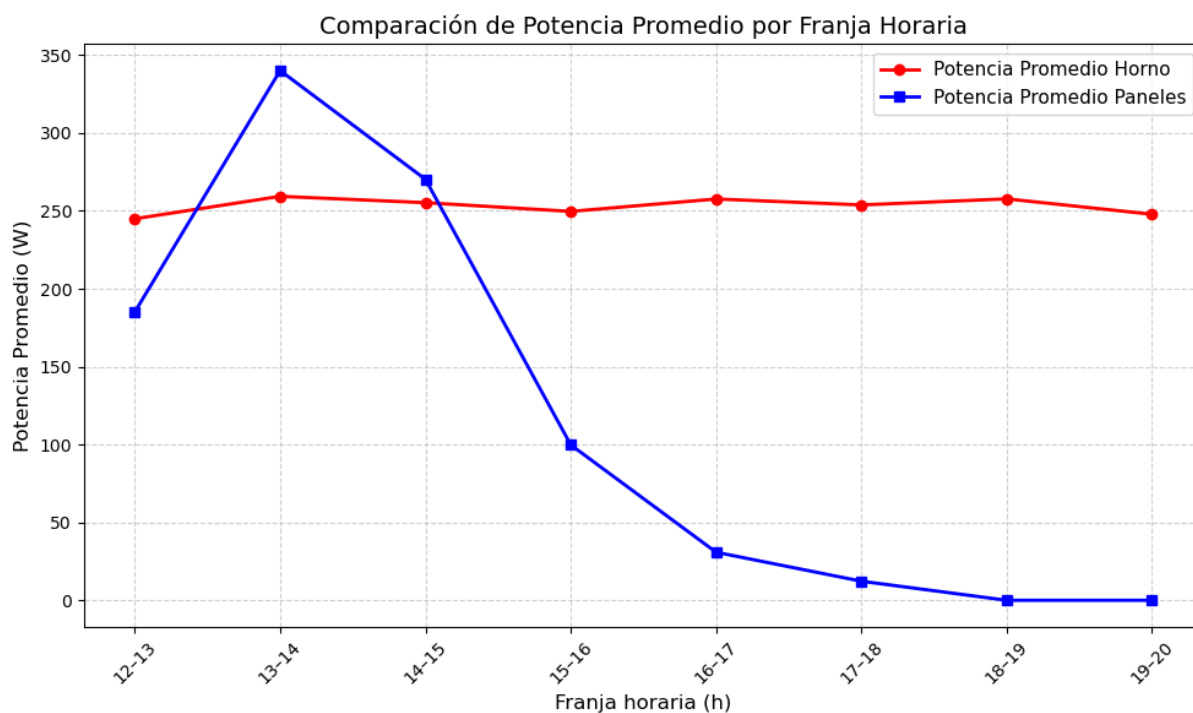


FIGURA 27: Gráfica de demanda y generación. **Tomado de:** Autores.

En la Figura 27 se presenta la comparación de la potencia promedio por franja horaria entre la generación fotovoltaica y la demanda del horno. Estos valores corresponden a promedios calculados dentro de intervalos de una hora, lo que permite observar el comportamiento general de ambas variables a lo largo del día. La potencia promedio de los paneles solares refleja el patrón típico de la irradiación solar, alcanzando su máximo valor alrededor de las 13:00 h, disminuyendo gradualmente en las horas siguientes hasta aproximarse a cero hacia las 18:00 h, cuando la radiación solar es mínima o inexistente. En cambio, la potencia promedio del horno se mantiene prácticamente constante durante todo el periodo de análisis. Esto indica un consumo estable que no depende directamente de la disponibilidad de energía fotovoltaica, lo cual evidencia que, sin una estrategia o gestión energética, el horno mantiene su demanda incluso cuando la generación solar es baja o nula. La gráfica permite identificar periodos de excedente energético, principalmente entre las 12:00 p.m y 14:00 p.m, cuando la generación supera la demanda del horno, y periodos de déficit energético a partir de las 16:00 p.m, cuando la potencia generada por los paneles no es suficiente para cubrir el consumo. Teniendo en cuenta lo anterior, el horno se comporta como una carga flexible, ya que puede ajustar su operación en función de la disponibilidad de la PV. Aprovechando su inercia térmica, puede emular el almacenamiento de energía en forma de calor durante las horas de mayor generación solar y reducir su consumo en los periodos de menor generación de la PV.

5.3.6. Aprovechamiento Energía Fotovoltaica

Para tener un mejor aprovechamiento de la PV, es fundamental conocer en qué horas del día se presentan los picos de irradiación solar, ya que en esos momentos los paneles fotovoltaicos pueden generar una mayor potencia. Al identificar y analizar estos periodos, es posible programar o ajustar el funcionamiento del horno para que su consumo coincida con las horas de mayor generación, reduciendo así la dependencia de la red aprovechando el uso de la energía disponible. Además, mediante estrategias de gestión energética como el uso de almacenamiento térmico o la inercia térmica del propio horno se puede almacenar calor durante los momentos de alta generación solar y aprovecharlo posteriormente.

5.4. Validación del Desempeño de la Microrred

5.4.1. Indicadores de Flexibilidad

Los indicadores de flexibilidad permiten medir, cuantificar y comparar la capacidad que tiene una microrred o un sistema eléctrico para adaptarse de forma dinámica a los cambios en la

generación y la demanda de energía.

Según [42], la capacidad de la microrred para aprovechar la generación fotovoltaica y disminuir la dependencia de la red puede evaluarse mediante varios indicadores, de los cuales se seleccionan dos para el estudio de flexibilidad.

$$SC = \frac{E_{PVload}}{E_{gen}} \quad (5.20)$$

Donde:

- E_{PVload} : Energía generada por el sistema fotovoltaico (PV) y consumida localmente.
- E_{gen} : Energía total generada por el sistema PV.

El autoconsumo (SC) (5.20) representa la proporción de la energía generada por los sistemas fotovoltaicos que es utilizada directamente dentro de la microrred. Este indicador permite evaluar qué tan eficiente es el aprovechamiento de la energía solar local, evitando inyecciones innecesarias a la red eléctrica. Un valor alto de auto-consumo refleja una mejor gestión de la generación distribuida y un mayor uso interno de los recursos energéticos disponibles.

$$SS = \frac{E_{PVload}}{E_{load}} \quad (5.21)$$

Donde:

- E_{load} : Demanda total de energía en el período analizado.

La autosuficiencia (SS) (5.21) indica el porcentaje de la demanda total de la microrred que puede ser abastecida con la energía generada localmente. Este indicador refleja el grado de independencia del sistema frente a la red eléctrica, mostrando la capacidad de la microrred para operar de forma autónoma. Un valor elevado de autosuficiencia implica una menor dependencia energética y una mejor integración de los recursos renovables dentro del sistema.

5.4.2. Emulación Almacenamiento Virtual de Energía

Para representar el comportamiento del horno como un sistema con capacidad de almacenamiento energético, se realiza una emulación del almacenamiento virtual de energía basada en su inercia térmica. Para ello, se emplearon los datos experimentales obtenidos en las pruebas

de calentamiento y enfriamiento, registrando la temperatura cada 5 minutos durante una hora de operación.

El procesamiento y análisis se efectuaron mediante un script en Python, el cual se encargó de ajustar modelos polinómicos de cuarto grado a los datos experimentales, utilizando la librería NumPy. El modelo empleado se expresa como:

$$T(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4$$

donde $T(t)$ corresponde a la temperatura del horno en función del tiempo t . A partir del ajuste se obtuvieron los coeficientes que permiten describir la evolución térmica del horno tanto en el calentamiento como en el enfriamiento, lo cual refleja la capacidad del sistema para retener y liberar energía después de cesar el suministro eléctrico.

Los valores de temperatura registrados para cada nivel de consigna fueron 95°C, 120°C, 150°C, 180°C, 200°C y 230°C, con temperaturas máximas reales de 104.4°C, 117.6°C, 142.1°C, 163.4°C, 183.1°C y 205.3°C, respectivamente. Las regresiones obtenidas mostraron coeficientes de determinación (R^2) superiores a 0.99 para el enfriamiento y entre 0.70 y 0.79 para el calentamiento.

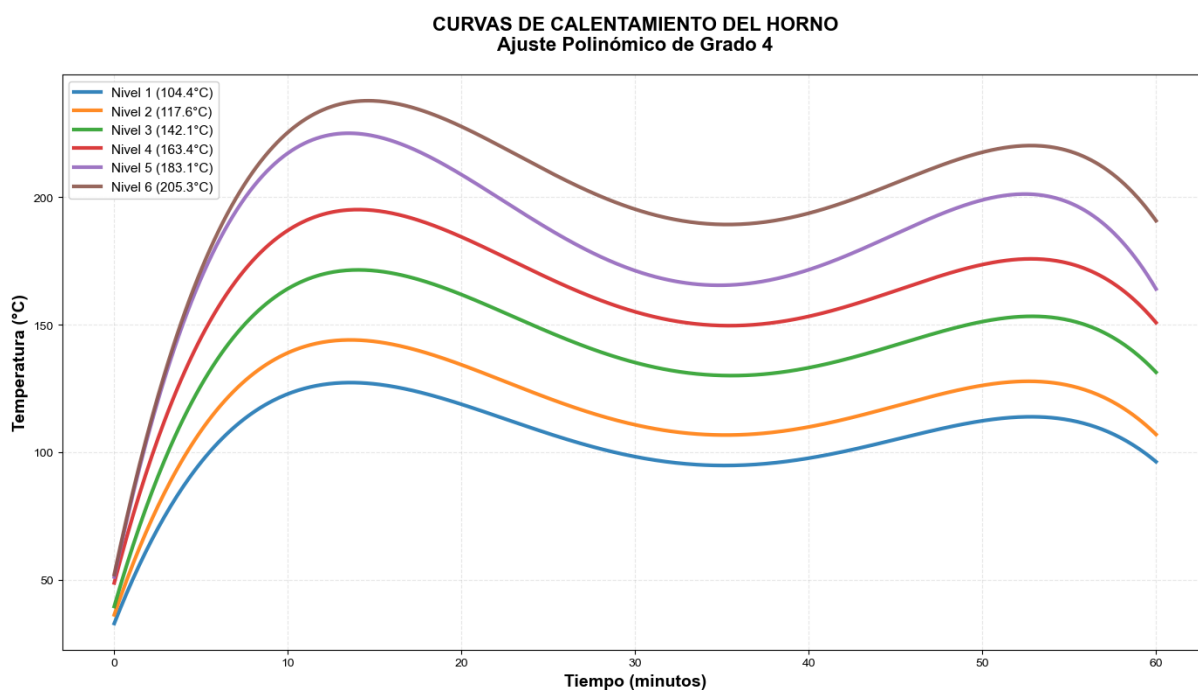


FIGURA 28: Curvas experimentales y polinómicas de calentamiento del horno.
Tomado de: Autores.

A partir de los datos experimentales de calentamiento, se obtuvieron las siguientes ecuaciones polinómicas ajustadas para cada nivel de temperatura real:

$$T_{104,4}(t) = 3,28 \times 10^1 + 2,89 \times 10^{-1}t - 2,91 \times 10^{-4}t^2 + 1,08 \times 10^{-7}t^3 - 1,32 \times 10^{-11}t^4$$

$$T_{117,6}(t) = 3,62 \times 10^1 + 3,30 \times 10^{-1}t - 3,32 \times 10^{-4}t^2 + 1,23 \times 10^{-7}t^3 - 1,51 \times 10^{-11}t^4$$

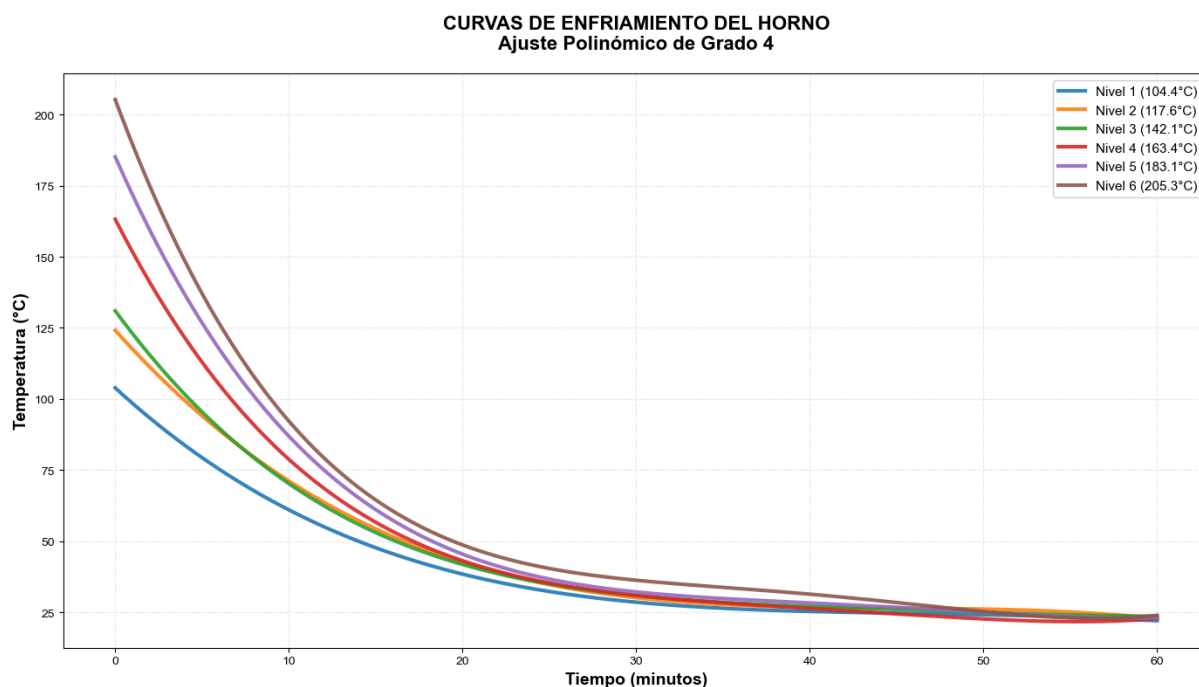
$$T_{142,1}(t) = 3,95 \times 10^1 + 3,93 \times 10^{-1}t - 3,87 \times 10^{-4}t^2 + 1,41 \times 10^{-7}t^3 - 1,72 \times 10^{-11}t^4$$

$$T_{163,4}(t) = 4,87 \times 10^1 + 4,36 \times 10^{-1}t - 4,30 \times 10^{-4}t^2 + 1,57 \times 10^{-7}t^3 - 1,92 \times 10^{-11}t^4$$

$$T_{183,1}(t) = 5,11 \times 10^1 + 5,36 \times 10^{-1}t - 5,44 \times 10^{-4}t^2 + 2,03 \times 10^{-7}t^3 - 2,52 \times 10^{-11}t^4$$

$$T_{205,3}(t) = 5,20 \times 10^1 + 5,37 \times 10^{-1}t - 5,18 \times 10^{-4}t^2 + 1,87 \times 10^{-7}t^3 - 2,28 \times 10^{-11}t^4$$

Estas ecuaciones describen la evolución de la temperatura del horno durante la fase de calentamiento para cada nivel de temperatura real. En la Figura 28 se observa cómo, al aumentar el nivel de consigna, la pendiente inicial de la curva es mayor, indicando una tasa de calentamiento más rápida hasta alcanzar el régimen estacionario. El comportamiento no lineal evidencia la influencia de la inercia térmica y de las pérdidas por convección y radiación a medida que la temperatura se incrementa.



De igual manera, las ecuaciones polinómicas ajustadas para el proceso de enfriamiento fueron:

$$T_{104,4}(t) = 1,04 \times 10^2 - 9,32 \times 10^{-2}t + 4,07 \times 10^{-5}t^2 - 7,63 \times 10^{-9}t^3 + 4,91 \times 10^{-13}t^4$$

$$T_{117,6}(t) = 1,24 \times 10^2 - 1,13 \times 10^{-1}t + 4,62 \times 10^{-5}t^2 - 7,28 \times 10^{-9}t^3 + 2,87 \times 10^{-13}t^4$$

$$T_{142,1}(t) = 1,31 \times 10^2 - 1,38 \times 10^{-1}t + 7,16 \times 10^{-5}t^2 - 1,69 \times 10^{-8}t^3 + 1,50 \times 10^{-12}t^4$$

$$T_{163,4}(t) = 1,63 \times 10^2 - 1,99 \times 10^{-1}t + 1,15 \times 10^{-4}t^2 - 3,04 \times 10^{-8}t^3 + 3,02 \times 10^{-12}t^4$$

$$T_{183,1}(t) = 1,85 \times 10^2 - 2,31 \times 10^{-1}t + 1,31 \times 10^{-4}t^2 - 3,34 \times 10^{-8}t^3 + 3,16 \times 10^{-12}t^4$$

$$T_{205,3}(t) = 2,05 \times 10^2 - 2,74 \times 10^{-1}t + 1,69 \times 10^{-4}t^2 - 4,68 \times 10^{-8}t^3 + 4,77 \times 10^{-12}t^4$$

Las curvas de la Figura 29 evidencian la disminución gradual de la temperatura una vez interrumpido el suministro eléctrico, lo que representa la disipación de la energía acumulada en las paredes y en la masa térmica del horno. La tendencia suavemente decreciente y de carácter no lineal confirma el efecto de la inercia térmica, en la cual el sistema libera calor de forma progresiva, emulando un comportamiento análogo al de un dispositivo de almacenamiento térmico pasivo.

Estas gráficas reflejan el efecto de la inercia térmica del horno, en la cual la energía absorbida durante el calentamiento se libera progresivamente en el tiempo, demostrando su capacidad de almacenamiento energético. El código utilizado para el procesamiento de datos y la generación de los modelos polinómicos se encuentra en el Anexo 8.5.

Capítulo 6

Resultados y Discusión

Inicialmente se tomaron mediciones del funcionamiento del horno en la microrred durante tres días diferentes, con el fin de definir y analizar tres casos experimentales orientados a evaluar la gestión de la intermitencia fotovoltaica. En todos los casos, el horno opera desde las horas de la mañana hasta la noche, mientras que la generación fotovoltaica varía según la irradiación solar a lo largo del día. A partir del análisis previo de la potencia generada por los paneles solares, se identificaron los periodos de mayor irradiación, lo que permitió ajustar los perfiles térmicos del horno para sincronizar su funcionamiento con las horas de mayor disponibilidad de energía fotovoltaica. De esta manera, se busca reducir la dependencia de la red eléctrica y aprovechar la energía solar disponible mediante estrategias de gestión como el almacenamiento térmico o el uso de la inercia térmica del propio horno, que permite conservar calor durante los momentos de alta generación para emplearlo posteriormente.

Durante las pruebas se registraron variables como la temperatura del horno, la temperatura ambiente, el voltaje, la potencia del horno y la potencia generada por los paneles solares, con un intervalo de adquisición de 2 segundos y verificación cada 5 segundos de las temperaturas objetivo. Este procedimiento experimental permite caracterizar el comportamiento energético de la microrred y evaluar su capacidad de respuesta ante la variabilidad de la generación solar. El balance de potencia del sistema se expresa mediante la ecuación:

$$P_{\text{red}} + P_{\text{paneles}} = P_{\text{horno}} + P_{\text{luminarias}} \quad (6.1)$$

donde P_{red} corresponde a la potencia suministrada por la red eléctrica, P_{paneles} a la potencia generada por el sistema fotovoltaico, y P_{horno} y $P_{\text{luminarias}}$ representan las cargas conectadas en

la microrred. Esta relación permite analizar los flujos de energía y determinar el grado en que los paneles fotovoltaicos contribuyen a suplir la demanda de las cargas térmicas y eléctricas.

Adicionalmente, se evaluaron los indicadores de flexibilidad, la ecuación (5.20) y (5.21) las cuales permiten cuantificar el aprovechamiento energético dentro de la microrred. El indicador de autoconsumo mide la proporción de la energía solar generada que es utilizada directamente en el sistema, mientras que el indicador de autosuficiencia expresa el porcentaje de la demanda total que es cubierta por la generación fotovoltaica. Estos parámetros son fundamentales para valorar la gestión de la intermitencia, ya que reflejan la eficiencia con la cual la microrred integra la energía renovable y reduce la dependencia de la red convencional.

Con base en lo anterior, se establecieron tres casos experimentales para analizar el comportamiento del sistema bajo diferentes condiciones de operación, variando las horas objetivo de calentamiento del horno y el perfil térmico programado.

El análisis comparativo de estos tres escenarios permite evaluar la capacidad de la microrred para gestionar la intermitencia en la generación fotovoltaica mediante el mecanismo Power to Heat, aprovechando la flexibilidad térmica del horno como elemento de compensación energética y de optimización del uso de la energía solar disponible.

6.1. Caso 1

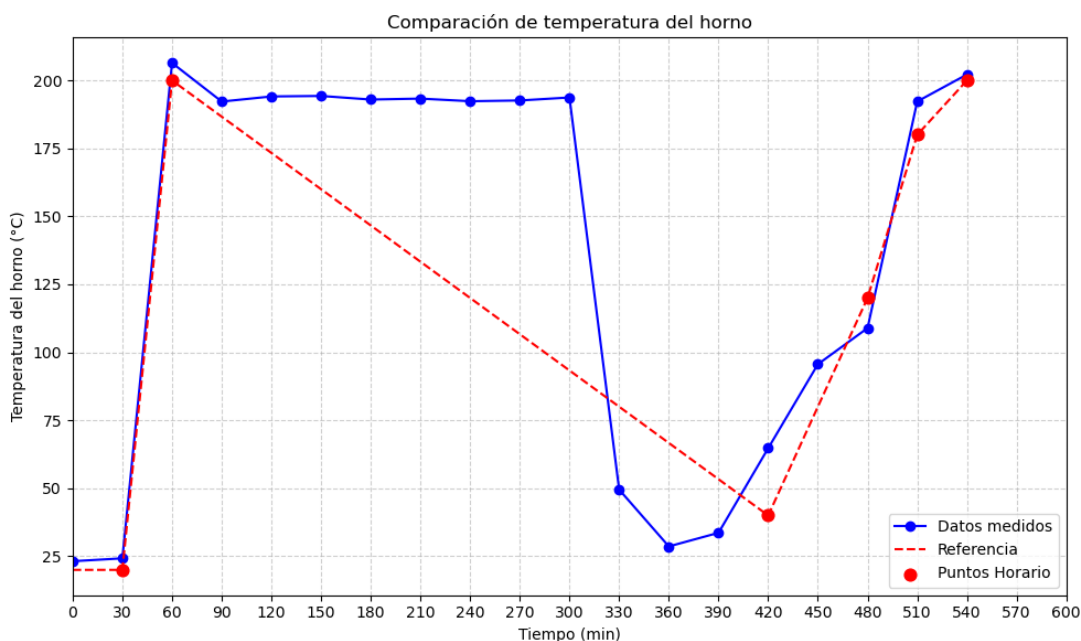
El caso se define como la rutina que se desarrolla a lo largo del día, determinando los momentos en los que el horno se enciende y las horas en las que debe mantener una determinada temperatura durante la noche. En la Tabla 16 se muestra esta programación, donde se establecen los tiempos en horas, el tiempo mínimo estimado para alcanzar cada temperatura y la temperatura objetivo que debe obtenerse en cada intervalo. Como se observa en esta tabla, el análisis busca evaluar el comportamiento térmico del horno durante todo el día, aprovechando la disponibilidad de energía fotovoltaica en las horas diurnas y su inercia térmica durante los periodos nocturnos.

TABLA 16: Tabla de tiempos para el caso 1. **Tomado de:** Autores.

Tiempo (H)	Tiempo (min)	Temperatura_Horno_°C
11:30	30	20
12:00	60	200
18:00	420	40
19:00	480	120
19:30	510	100
20:00	540	120
20:30	570	130
21:00	600	200

En la Tabla 16 se presenta entonces el caso 1, donde se establece que a las 11:30 a.m. el horno parte desde 20 °C y debe alcanzar los 200 °C hacia las 12:00 p.m., momento en el que la disponibilidad fotovoltaica es mayor. Posteriormente, a las 6:00 p.m., cuando ya no se dispone de generación solar, la temperatura objetivo desciende a 40 °C. A partir de ese punto, el horno continúa variando su temperatura, alcanzando los 120 °C en etapas posteriores. Cada intervalo de 30 minutos se analiza aplicando las ecuaciones de enfriamiento, con el fin de determinar el tiempo aproximado requerido para que el horno alcance cada nueva consigna térmica.

En la Figura 30 se representa la variación de la temperatura del horno en función del tiempo para este caso.

FIGURA 30: Gráfica de la temperatura del horno para el caso 1. **Tomado de:** Autores.

En la Figura 30, se observan dos señales: la de color azul corresponde a los datos medidos de la temperatura del horno cada 30 minutos, mientras que la de color rojo representa la señal de referencia o las distintas temperaturas objetivo. La temperatura medida no coincide exactamente con la referencia, pero se mantiene dentro de un rango cercano. A las 11:30 a.m. se mantiene entre 190 °C y 200 °C, y alrededor de las 16:00 p.m. se inicia el aprovechamiento de la inercia térmica del horno, momento en el que ya no se cuenta con suficiente aporte fotovoltaico. A partir de las 16:30 p.m., esta inercia adquiere un papel relevante al permitir conservar calor sin requerir potencia adicional constante, aunque para alcanzar nuevamente las temperaturas objetivo durante la noche es necesario encender el horno por breves intervalos.

En cuanto al análisis de potencia, la Figura 31 muestra la comparación entre la potencia del horno y la potencia generada por los paneles solares a lo largo del día.

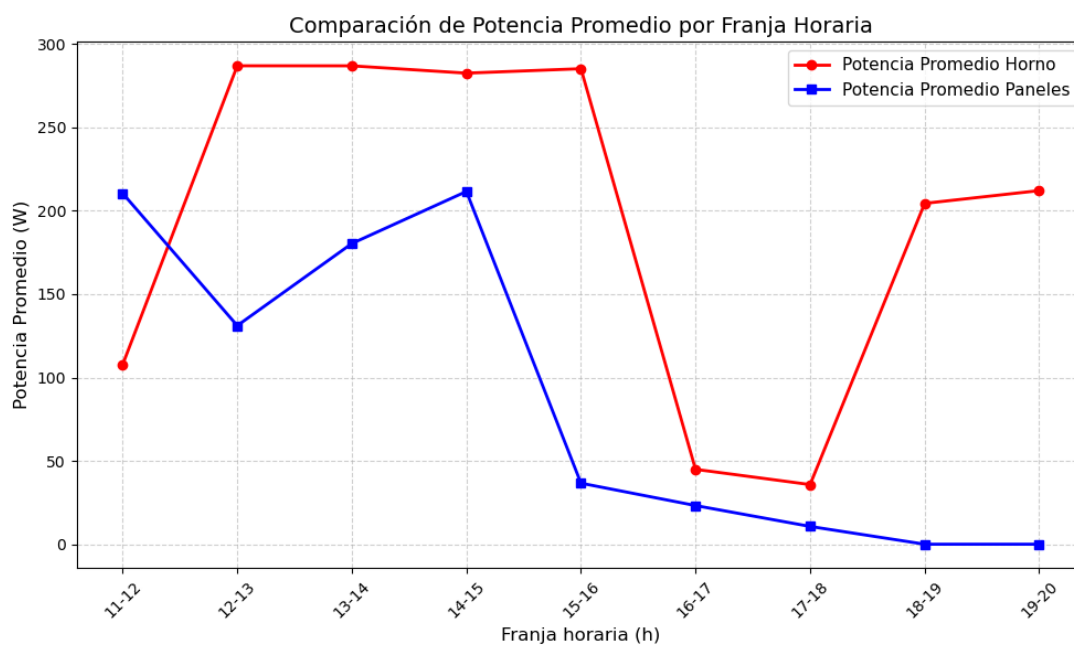


FIGURA 31: Gráfica de la potencia del horno y la potencia de los paneles para el caso 1. **Tomado de:** Autores.

En la Figura 31, se aprecian dos señales: la de color azul representa la potencia promedio por hora entregada por los paneles solares y la de color rojo la potencia promedio consumida por el horno. En este caso, la potencia del horno fue superior a la suministrada por la PV durante casi toda la jornada. En las horas nocturnas, entre las 18:00 p.m. y las 19:00 p.m., la energía fotovoltaica deja de estar disponible, por lo que el horno opera únicamente con la red eléctrica. Sin embargo, gracias a la inercia térmica, el consumo disminuye al aprovechar el calor acumulado durante el día, contribuyendo así a una gestión más eficiente de la energía.

6.2. Caso 2

Para el segundo caso, el procedimiento es similar al del caso 1 en cuanto a la ejecución, ya que se definen diferentes temperaturas objetivo a lo largo del día. En la Tabla 17 se presentan los valores establecidos de tiempo y temperatura para este caso.

TABLA 17: Tiempos y temperaturas objetivo para el caso 2. **Tomado de:** Autores.

Hora	Tiempo (min)	Temperatura del horno (°C)
10:00	60	25
11:00	120	120
13:00	180	150
15:30	510	180
17:30	570	200
19:30	690	90
21:00	720	40

En la Tabla 17 se observa que durante el día se establecen varias temperaturas objetivo, con la particularidad de que estas varían conforme avanza la jornada. Por ejemplo, de 10:00 a.m. a 11:00 a.m. se mantiene una temperatura de 25 °C, mientras que a las 13:00 p.m. se incrementa a 150 °C. Durante el día, cuando los paneles fotovoltaicos están activos, el horno sigue consumiendo energía apoyándose en la generación solar. Sin embargo, en horas de la noche, al no contar con aporte fotovoltaico, se aprovecha la inercia térmica del horno para mantener la temperatura y reducir el consumo de la red eléctrica.

En la Figura 32 se presenta la evolución de la temperatura del horno a lo largo del tiempo para el caso 2.

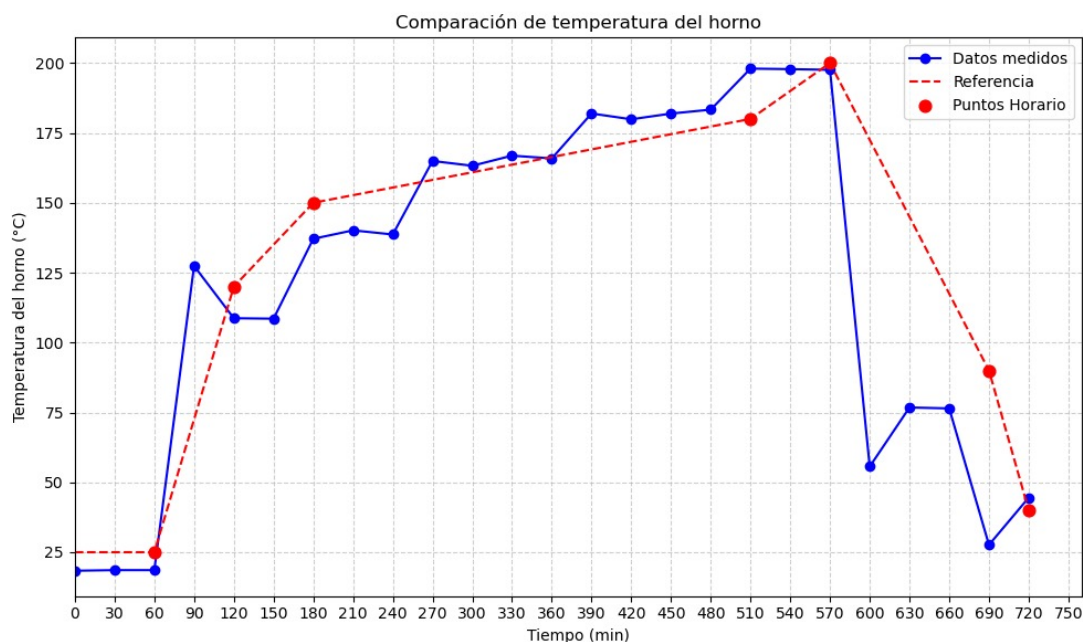


FIGURA 32: Evolución de la temperatura del horno para el caso 2. **Tomado de:** Autores.

En la Figura 32 se observan dos señales: la de color azul corresponde a las temperaturas medidas, mientras que la de color rojo representa las temperaturas objetivo. Se evidencia que el comportamiento medido se aproxima a los valores de referencia, mostrando variaciones acordes con la disponibilidad de potencia fotovoltaica. Por ejemplo, a las 11:00 a.m. se alcanza una temperatura de 108.7 °C, que se encuentra dentro del rango de 100–120 °C, y a las 13:00 p.m. el valor medido es de 138.66 °C, cumpliendo con el objetivo establecido. A partir de las 18:30 p.m., el sistema se apoya en la inercia térmica para mantener la temperatura sin requerir un consumo elevado de potencia de la red.

En cuanto al análisis de potencia, en la Figura 33 se muestra la comparación entre la potencia promedio del horno y la potencia generada por los paneles solares.

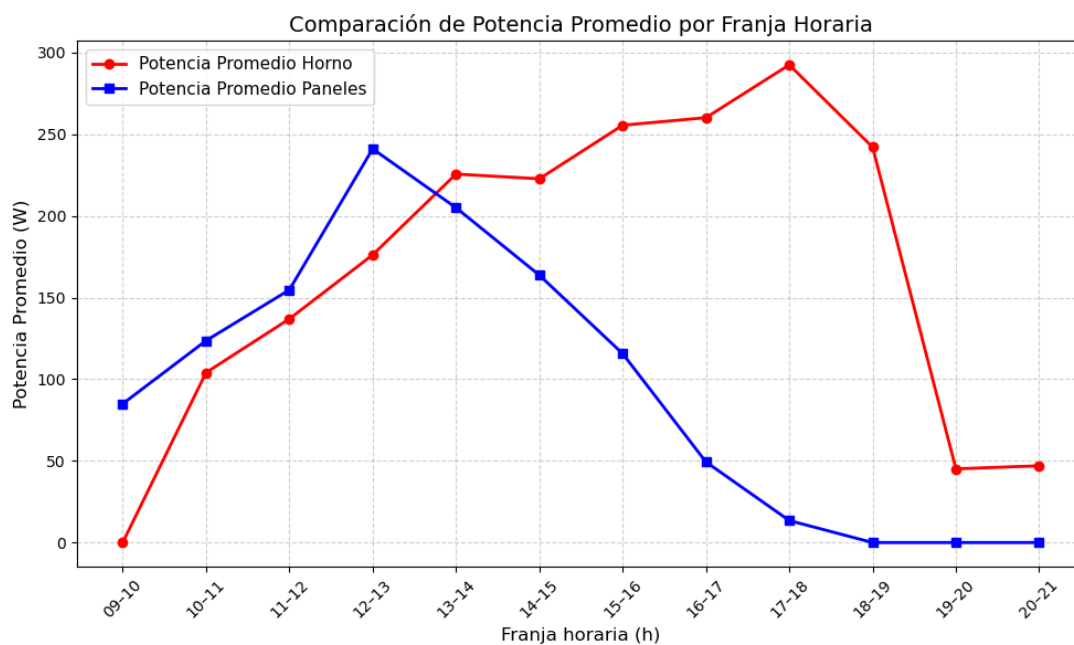


FIGURA 33: Potencia promedio del horno y de los paneles solares para el caso 2.
Tomado de: Autores.

En la Figura 33 se pueden identificar dos señales: la de color azul representa la potencia promedio generada por los paneles solares, mientras que la de color rojo corresponde a la potencia consumida por el horno. Durante las horas comprendidas entre las 9:00 a.m. y las 13:00 p.m. se observa una mayor generación fotovoltaica, coincidiendo con el periodo de mayor irradiación solar. Posteriormente, hacia las 18:00 p.m., la potencia del horno disminuye debido al aprovechamiento de la inercia térmica para mantener la temperatura sin necesidad de aporte energético externo.

6.3. Caso 3

Para el tercer caso, el comportamiento es similar al del caso 1, con la diferencia de que presenta una menor duración, ya que finaliza a las 19:00 p.m. En la Tabla 18 se presentan las temperaturas objetivo definidas para este caso.

TABLA 18: Tiempos y temperaturas objetivo para el caso 3. Tomado de: Autores.

Hora	Tiempo (min)	Temperatura del horno (°C)
10:00	60	25
13:00	240	200
17:00	480	200
17:30	510	200
18:00	540	150
18:30	570	90
19:00	600	50

En la Tabla 18 se observan las diferentes temperaturas objetivo establecidas durante la jornada, manteniendo intervalos de aproximadamente 30 minutos entre cada punto de referencia y un tiempo total de operación hasta las 19:00 p.m. En la Figura 34 se muestra la variación de la temperatura del horno en función del tiempo para el caso 3.

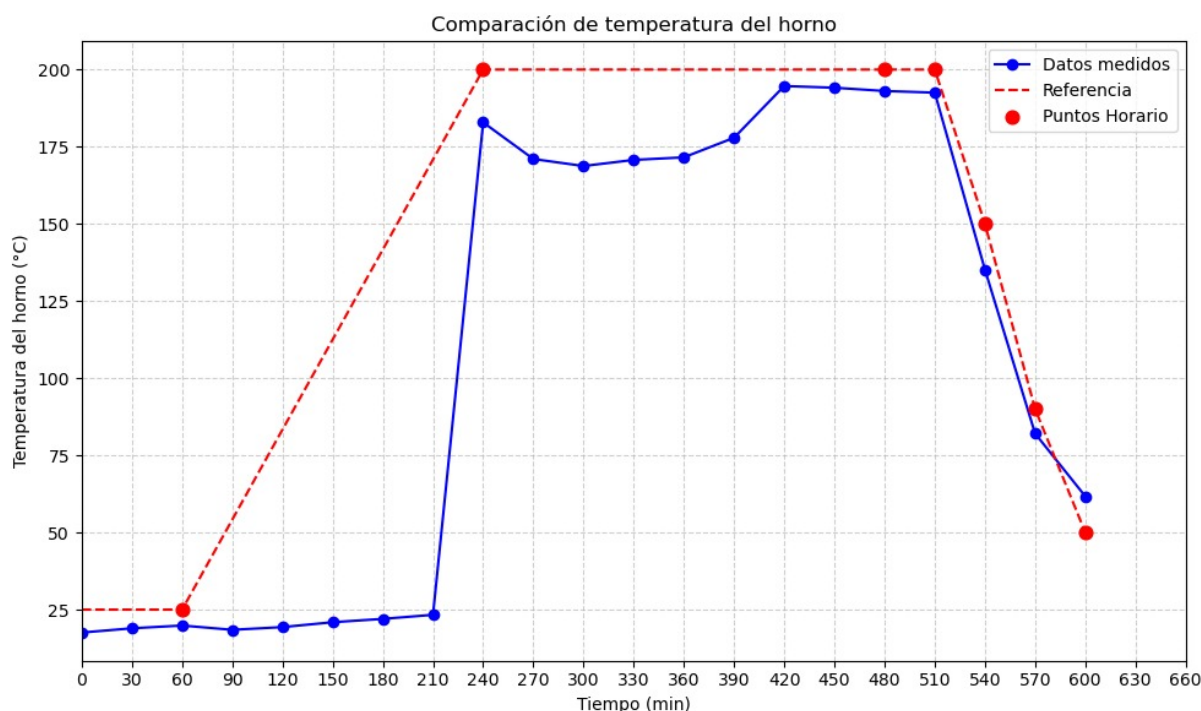


FIGURA 34: Evolución de la temperatura del horno para el caso 3. Tomado de: Autores.

En la Figura 34 se presentan dos señales: la de color azul corresponde a las temperaturas medidas y la de color rojo a las temperaturas objetivo. Se evidencia que el comportamiento medido sigue adecuadamente las referencias establecidas, manteniendo una buena correspondencia entre ambas. En este caso, el horno alcanza una temperatura de 182.88 °C hacia las 13:00 p.m., la

cual se conserva estable hasta las 17:30 p.m. A partir de ese momento, se aprovecha la inercia térmica del sistema para mantener la temperatura sin requerir una alta demanda de potencia, finalizando el ciclo a las 19:00 p.m.

En cuanto al comportamiento de la potencia, la Figura 35 presenta la comparación entre la potencia promedio generada por los paneles fotovoltaicos y la potencia promedio consumida por el horno.

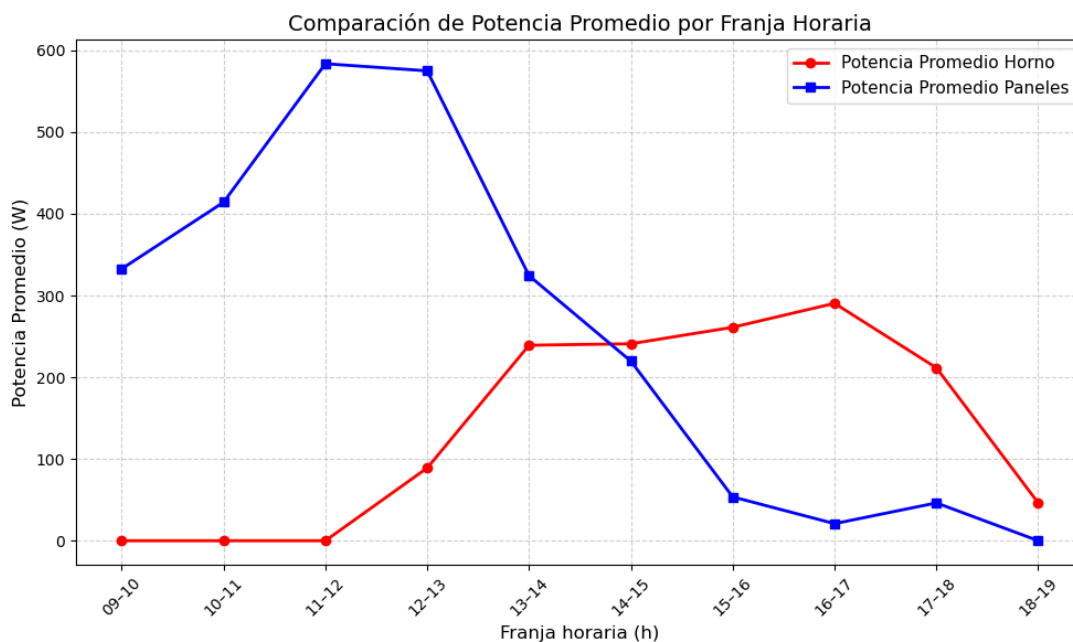


FIGURA 35: Potencia promedio del horno y de los paneles solares para el caso 3.
Tomado de: Autores.

En la Figura 35 se distinguen dos señales: la de color azul representa la potencia promedio generada por los paneles solares, mientras que la de color rojo muestra la potencia promedio consumida por el horno. Se observa que la potencia fotovoltaica es mayor en los momentos en los que el horno no está en funcionamiento. Sin embargo, hacia las 12:00 p.m. se alcanza un pico de generación solar que coincide con el inicio del consumo del horno, permitiendo que parte de la energía requerida sea suministrada por los paneles. Entre las 13:00 p.m. y las 14:00 p.m. la potencia consumida por el horno se aproxima a la potencia generada por la PV, mostrando una operación coordinada. Finalmente, a partir de las 17:00 p.m., la potencia del horno disminuye progresivamente debido al aprovechamiento de la inercia térmica para mantener las temperaturas objetivo sin requerir aporte adicional de la red eléctrica.

En los tres casos representados en las Figuras 31, 33 y 35, se observa un menor consumo de energía durante las horas en las que no hay disponibilidad de la PV. Gracias al aprovechamiento

de la inercia térmica, el sistema puede adaptarse a cambios o perturbaciones de la red; sin embargo, debido a que el horno presenta una baja inercia térmica, la temperatura desciende con rapidez. Por este motivo, durante la noche el horno se enciende de manera momentánea para recuperar la temperatura objetivo. Cuando la temperatura supera el valor deseado en las horas nocturnas, el horno se apaga nuevamente, permitiendo que la inercia térmica actúe hasta alcanzar la temperatura requerida. De esta forma, si se empleara un horno con mayor inercia térmica, el consumo eléctrico durante la noche sería considerablemente menor. Además, al aprovechar la PV durante el día y gestionar el uso de la energía mediante la inercia térmica, el sistema puede responder de manera flexible a la demanda, reduciendo el consumo en los periodos de alta carga de la red y desplazando el uso de energía hacia las horas con mayor disponibilidad fotovoltaica.

6.4. Indicadores de Flexibilidad

Usando los indicadores de flexibilidad, se verifica la capacidad del sistema para adaptarse a la variabilidad de la generación y la demanda de energía. A partir de (5.20) y (5.21), se determina el grado de aprovechamiento de la generación fotovoltaica y el nivel de dependencia de la red eléctrica.

Teniendo en cuenta lo anterior, es necesario considerar las variables relacionadas con la energía aportada. Cuando el horno no está en funcionamiento, los excedentes generados por el sistema fotovoltaico se envían a la red eléctrica. En cambio, cuando el horno está consumiendo energía, dichos excedentes son aprovechados directamente por este, sin ser exportados a la red. Por otro lado, cuando la generación fotovoltaica no es suficiente para cubrir la demanda, la red eléctrica suministra la energía faltante. Finalmente, la energía exportada corresponde a aquella que la red recibe en los momentos en que el horno no realiza consumo.

Para los 3 casos se realizan las siguientes operaciones.

$$SC = \frac{710,297Wh}{803,799Wh} * 100 = SC = 88,36 \% \quad (6.2)$$

$$SS = \frac{710,297Wh}{1827,726Wh} * 100 = SS = 38,86 \% \quad (6.3)$$

$$SC = \frac{964,133Wh}{1151,333Wh} * 100 = SC = 83,74 \% \quad (6.4)$$

$$SS = \frac{964,133Wh}{2247,295Wh} * 100 = SS = 42,90 \quad (6.5)$$

$$SC = \frac{668,666Wh}{2569,846Wh} * 100 = SC = 26,01 \% \quad (6.6)$$

$$SS = \frac{668,666Wh}{2378,456Wh} * 100 = SS = 28,11 \% \quad (6.7)$$

En (6.2) y (6.3) se presentan los valores porcentuales de los indicadores de autoconsumo (SC) y autosuficiencia (SS). Para este caso, se obtuvo un valor de $SC = 88,34 \%$ y $SS = 38,86 \%$. El valor de SC indica que el $88,34 \%$ de la energía generada por el sistema fotovoltaico fue utilizada directamente dentro de la microrred, mientras que el porcentaje restante provino de la red eléctrica. Por su parte, el valor de SS muestra que solo el $38,86 \%$ de la demanda total fue cubierta con generación fotovoltaica, lo que evidencia una mayor dependencia de la red eléctrica para suplir la energía requerida en el caso 1.

En (6.4) y (6.5) se presentan los valores porcentuales de los indicadores de autoconsumo (SC) y autosuficiencia (SS) para el segundo caso. Se obtuvo un valor de $SC = 83,74 \%$ y $SS = 42,90 \%$. El valor de SC indica que el $83,74 \%$ de la energía generada por el sistema fotovoltaico fue utilizada directamente dentro de la microrred, lo que evidencia un alto aprovechamiento local de la energía y una menor inyección a la red eléctrica. Por su parte, el valor de SS muestra que el $42,90 \%$ de la demanda total fue cubierta con energía fotovoltaica, representando una mejora respecto al caso 1, donde el porcentaje fue de $38,86 \%$. Esto implica una ligera reducción en la dependencia de la red eléctrica, aunque aún se requiere apoyo de esta para suplir el restante $57,10 \%$ del consumo.

En las ecuaciones (6.6) y (6.7) se presentan los valores porcentuales de los indicadores de autoconsumo (SC) y autosuficiencia (SS) para el tercer caso. Se obtuvo un valor de $SC = 26 \%$ y $SS = 28,11 \%$. El bajo valor de SC indica que una gran parte de la energía generada por el sistema fotovoltaico no fue utilizada dentro de la microrred, sino que fue inyectada a la red eléctrica. De igual manera, el valor de SS muestra que solo el $28,11 \%$ de la demanda total fue cubierta con energía fotovoltaica, lo que evidencia una alta dependencia de la red eléctrica, cercana al 70% . Esto sugiere que los excedentes de generación no fueron aprovechados de manera eficiente para cubrir la demanda local.

6.4.1. Energía Exportada y Suministrada por la Red

La energía aportada por la red (E_{aportada}) representa la cantidad total de energía que el sistema toma del suministro eléctrico convencional para cubrir la demanda del horno y demás cargas conectadas. Su valor se determina mediante la integración de la potencia activa de la red en el intervalo de medición, según (6.8), formulación empleada en diversos estudios sobre evaluación de flujos energéticos y balance de consumo en sistemas eléctricos conectados a red [42], [43]:

$$E_{\text{aportada}} = \sum_t P_{\text{red}}(t) \cdot \Delta t \quad (6.8)$$

Por su parte, la energía exportada hacia la red ($E_{\text{exportada}}$) corresponde a la fracción de energía que retorna al sistema eléctrico desde la instalación cuando la potencia generada supera el consumo instantáneo. Su cálculo se realiza mediante (6.9), la cual sigue la misma formulación empleada por [42] para el análisis del intercambio energético bidireccional:

$$E_{\text{exportada}} = \sum_t P_{\text{exportada}}(t) \cdot \Delta t \quad (6.9)$$

Asimismo, la energía realmente aprovechada por la carga (E_{usada}) se obtiene como la diferencia entre la energía total aportada por la red y la exportada, según (6.10), metodología utilizada para cuantificar el aprovechamiento energético instantáneo de la instalación [43]:

$$E_{\text{usada}} = E_{\text{aportada}} - E_{\text{exportada}} \quad (6.10)$$

La energía aportada por la red (E_{aportada}) corresponde a la fracción de la demanda que no pudo ser cubierta por la generación local y, por tanto, debió ser suministrada por la red eléctrica. De acuerdo con las definiciones de consumo total (C) y energía generada y consumida localmente (GC) [44], esta magnitud se expresa como:

$$E_{\text{aportada}} = C - GC \quad (6.11)$$

Por otro lado, la energía exportada hacia la red ($E_{\text{exportada}}$) representa la porción de la generación fotovoltaica que no fue utilizada localmente y que, por tanto, se inyecta a la red cuando la generación supera el consumo instantáneo [44]. Esta energía formula como:

$$E_{\text{exportada}} = G - GC \quad (6.12)$$

Estas expresiones permiten cuantificar de manera directa la energía intercambiada con la red en los casos 1, 2 y 3. Para cada escenario, la energía utilizada desde la PV, así como la energía aportada y exportada, se obtiene a partir de las tablas presentadas en la sección de anexos de Datos Experimentales, específicamente en las Tablas 28, 29 y 30, correspondientes a cada caso de estudio.

En todos los casos analizados, el comportamiento energético depende directamente de la relación entre la potencia generada por el sistema fotovoltaico y la potencia demandada por las cargas. Cuando la generación es insuficiente, la red debe complementar la demanda; por el contrario, cuando la PV produce más de lo requerido, se presentan exportaciones hacia la red.

En el Caso 1 (horno + luminaria de 9 W), la red aportó $E_{\text{aportada}} = 1117,42$ Wh y se exportó $E_{\text{exportada}} = 93,68$ Wh. La generación fotovoltaica inicia entre las 11:00 y las 15:00 h, alcanzando un máximo de 211,59 W en 14–15 h. Sin embargo, este aporte no es suficiente para cubrir toda la demanda del sistema, por lo que la red continúa suministrando energía en la mayoría de los intervalos. La única devolución a la red registrada entre 11–12 h, con un valor de 93,68 W, corresponde a un excedente temporal de la PV cuando la demanda era baja. Esto concuerda con la Tabla 28.

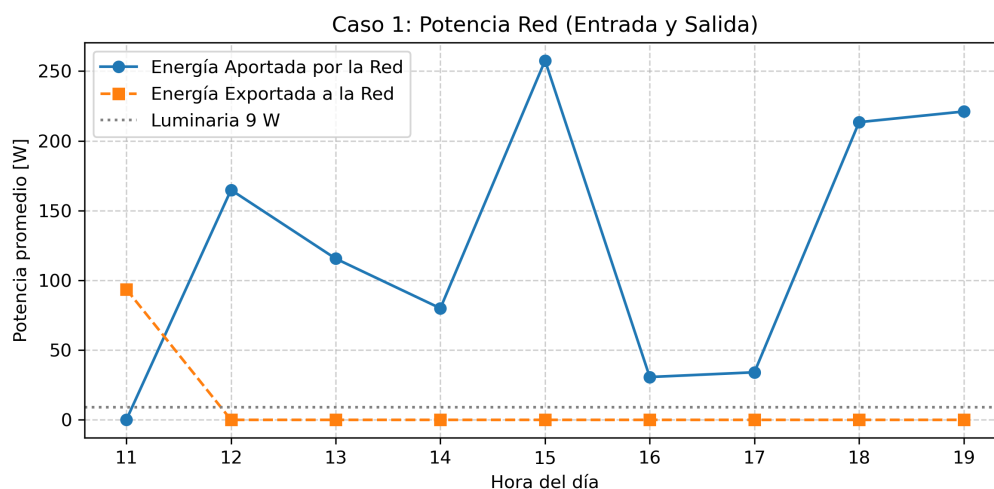


FIGURA 36: Comportamiento horario de la energía aportada y exportada en el Caso 1 (Horno + luminaria de 9 W). **Tomado de:** Autores.

Como muestra la Figura 36, la red incrementa su aporte hacia el final del día debido a la disminución natural de la irradiancia, mientras que la exportación únicamente aparece en el intervalo

donde hubo excedente fotovoltaico. En total, se utilizaron 710,30 Wh desde la PV, la red aportó 1117,42 Wh y se exportaron 93,68 Wh.

En el Caso 2 (horno + luminaria de 20 W), la red aportó 1205,80 Wh y la exportación alcanzó 109,84 Wh. En este caso se observa una mejor coincidencia entre la generación PV y la demanda. La exportación se concentra únicamente cuando la potencia generada supera la requerida por la carga, como ocurre en 9–10 h y 12–13 h, tal como muestra la Tabla 29. El resto del tiempo, la red complementa la demanda cuando la PV es insuficiente, especialmente en la tarde.

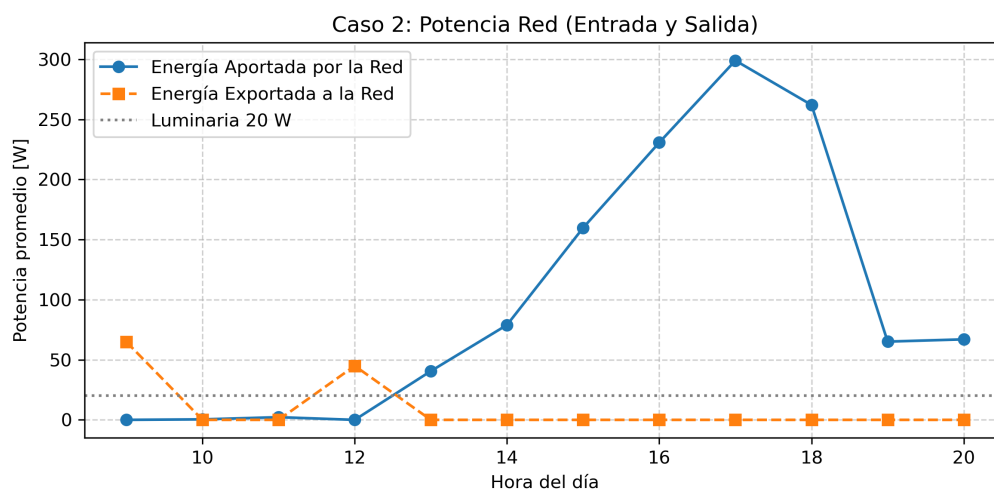


FIGURA 37: Comportamiento horario de la energía aportada y exportada en el Caso 2 (Horno + luminaria de 20 W). **Tomado de:** Autores.

En el Caso 3 (horno + luminaria de 100 W), la red aportó 1224,68 Wh mientras que la exportación fue significativamente mayor, alcanzando 1416,07 Wh. Este comportamiento se debe a que durante la mañana la generación fotovoltaica fue considerablemente más alta que en los otros casos, y como la carga era fija, se generaron excedentes continuos entre 9–13 h. Según la Tabla 30, en ese periodo toda la potencia PV se destina a exportación porque no hay coincidencia con la demanda. A partir de las 13:00 h, la red empieza a aportar energía nuevamente debido a la disminución progresiva de la irradiancia y a que la potencia de la carga (100 W) supera lo que la PV puede entregar en la tarde.

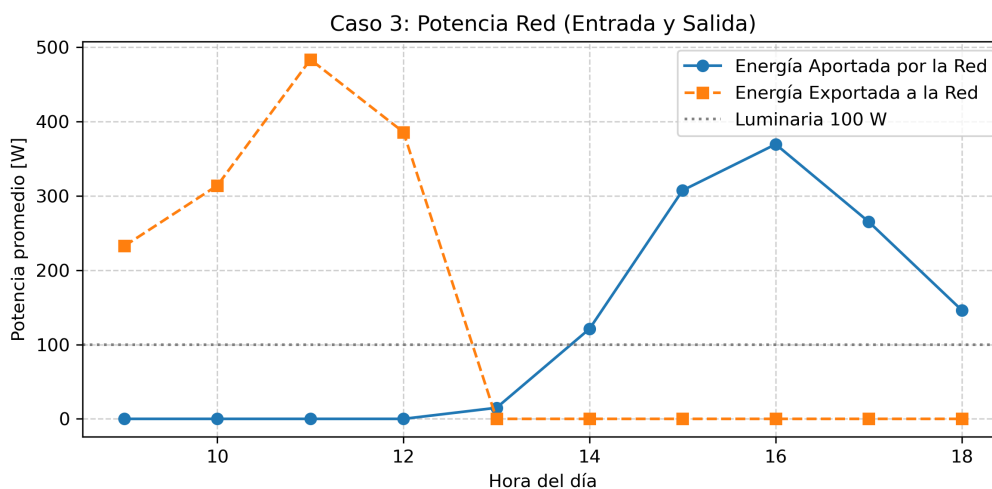


FIGURA 38: Comportamiento horario de la energía aportada y exportada en el Caso 3 (Horno + luminaria de 100 W). **Tomado de:** Autores.

La Figura 39 permite comparar el balance energético total entre los tres casos. En los Casos 1 y 2, la energía aportada por la red es mayor que la exportada, indicando que la demanda superó la generación disponible. En contraste, en el Caso 3 la energía exportada es mucho mayor porque ese día hubo una mayor producción fotovoltaica durante la mañana, mientras que la demanda no coincidió temporalmente con esa generación. Por esta razón se observa un alto nivel de exportación en las primeras horas del día y, posteriormente, a partir de las 14:00 h la red comienza nuevamente a aportar debido a la disminución de irradiancia.

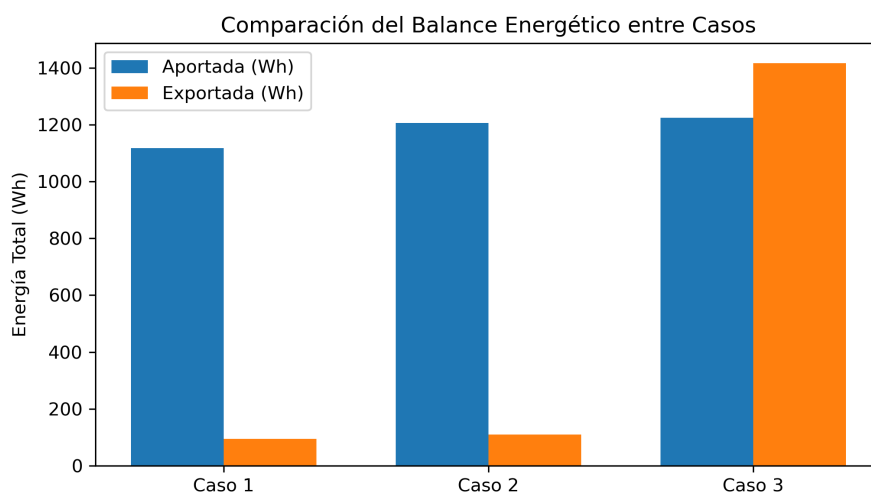


FIGURA 39: Comparación del balance energético total entre los tres casos analizados. **Tomado de:** Autores.

Capítulo 7

Conclusiones y Trabajos Futuros

7.1. Conclusiones

A partir de la revisión bibliográfica realizada, se identificaron diferentes enfoques para el modelado matemático de tecnologías Power-to-Heat (PtH), entre ellos los modelos empíricos, los modelos basados en aprendizaje automático (Machine Learning) y los modelos físicos o fenomenológicos. No obstante, para el desarrollo del presente trabajo se optó por emplear el modelado empírico, dado el nivel de familiaridad existente con esta metodología y la relación con los datos experimentales en las pruebas.

La selección del modelo matemático se realizó mediante un programa en Python, con el objetivo de identificar la función que mejor se ajusta al comportamiento observado experimentalmente. Como resultado, se eligió un modelo polinómico, presentado en (5.10), el cual permite estimar la potencia consumida en función de la diferencia de temperatura dentro de un determinado rango de tiempo.

Durante la implementación de la microrred con la integración del horno no se presentaron inconvenientes, gracias al adecuado dimensionamiento realizado. El horno operó correctamente dentro del sistema; sin embargo, al gestionar la generación fotovoltaica se evidenció una limitación relacionada con su baja inercia térmica. En los casos analizados (1, 2 y 3), fue necesario encender el horno durante las horas sin disponibilidad de energía solar para alcanzar la temperatura objetivo. Se considera que el uso de un horno con mayor inercia térmica permite una mejor gestión de la energía fotovoltaica.

A partir del análisis de los indicadores de flexibilidad, se validó el desempeño de la microrred propuesta frente a diferentes escenarios de operación. En el caso 1, se obtuvo un autoconsumo

del 88,36 % y una autosuficiencia del 38,86 %, reflejando un buen aprovechamiento de la generación fotovoltaica, aunque con una dependencia moderada de la red eléctrica. En el caso 2, ambos indicadores mejoraron $SC = 83,74\%$ y $SS = 42,90\%$, evidenciando una mayor utilización local de la energía solar y una ligera reducción en la demanda proveniente de la red. Finalmente, en el caso 3, los valores disminuyeron significativamente $SC = 26,11\%$ y $SS = 28,11\%$, lo que indica un bajo aprovechamiento de la generación fotovoltaica y una fuerte dependencia de la red externa. En conjunto, estos resultados permiten analizar como la microrred responde adecuadamente a la variabilidad de generación y demanda, alcanzando su mejor desempeño bajo condiciones que favorecen el uso de la PV.

7.2. Trabajos Futuros

A partir de los resultados obtenidos, se identifican varias líneas de trabajo que pueden fortalecer el desarrollo y la gestión de la microrred fotovoltaica. Estas proyecciones buscan mejorar la eficiencia energética, la autonomía operativa y la integración de fuentes renovables en entornos distribuidos.

- **Integración de microrredes:** Implementar la interconexión de múltiples microrredes locales que operen de forma cooperativa, permitiendo el intercambio de energía entre ellas y un uso más eficiente de los recursos fotovoltaicos disponibles.
- **Gestión basada en optimización o control:** Desarrollar un sistema de gestión energética autónomo que, mediante algoritmos de optimización o control distribuido, regule el flujo de potencia y aproveche de forma dinámica la generación solar disponible.
- **Ampliación funcional del sistema:** Incorporar almacenamiento energético, pronóstico de irradiancia y estrategias de respuesta a la demanda, con el fin de evaluar escenarios de operación más complejos y resilientes.

Capítulo 8

Anexos

En esta sección se presentan las tablas de caracterización térmica del horno utilizadas durante las pruebas experimentales. Se documentan los datos de energía consumida, potencia empleada y variaciones de temperatura para cada uno de los seis niveles de operación del horno, con temperaturas que oscilan entre 104°C y 205°C. Esta información permite establecer el comportamiento energético del equipo bajo diferentes condiciones de trabajo.

8.1. Caracterización Térmica del Horno

En esta sección se presenta la caracterización térmica del horno a través de seis niveles de temperatura (desde 104°C hasta 205°C). Para cada nivel se documentan los datos experimentales de delta de temperatura, energía consumida y potencia utilizada durante el proceso de calentamiento. Esta información permite cuantificar el comportamiento energético del equipo y establecer la relación entre el consumo eléctrico y las variaciones de temperatura alcanzadas, datos fundamentales para la optimización de procesos y el cálculo de costos operativos.

8.1.1. Nivel 1: 104.40 °C

Delta de Temperatura (°C)	Energía (Wh)	Potencia (W)
87,91	243720	135,40
97,63	232560	129,20
122,58	295080	163,93
145,37	394080	218,93
165,88	496080	275,60
188,21	622480	345,82

TABLA 19: Relación entre la energía, potencia y variación de temperatura del horno en el nivel 1 (104.4 °C).

8.1.2. Nivel 2: 117.59 °C

Delta de Temperatura (°C)	Energía (Wh)	Potencia (W)
100,4	296960	164,98
84,4	174320	96,84
122,17	282520	156,96
143,54	398120	221,18
165,09	494320	274,62
187,89	595760	330,98

TABLA 20: Relación entre la energía, potencia y variación de temperatura del horno en el nivel 2 (117.59 °C).

8.1.3. Nivel 3: 142.09 °C

Delta de Temperatura (°C)	Energía (Wh)	Potencia (W)
125,21	379840	211,02
84,4	156240	86,80
98,56	224080	124,49
144,94	378640	210,36
165,27	493640	274,24
185,65	560360	311,31

TABLA 21: Relación entre la energía, potencia y variación de temperatura del horno en el nivel 3 (142,09 °C).

8.1.4. Nivel 4: 163,37 °C

Delta de Temperatura (°C)	Energía (Wh)	Potencia (W)
143,64	436440	242,47
143,54	342200	190,11
185,43	590440	328,02
146,37	342200	190,11
163,44	475640	264,24
144,09	349600	194,22

TABLA 22: Relación entre la energía, potencia y variación de temperatura del horno en el nivel 4 (163,37 °C).

8.1.5. Nivel 5: 183,09°C

Delta de Temperatura (°C)	Energía (Wh)	Potencia (W)
167,09	531440	295,24
86,4	148520	82,51
99,59	194720	108,18
124,09	279040	155,02
144,37	349600	194,22
187,26	541400	300,78

TABLA 23: Relación entre la energía, potencia y variación de temperatura del horno en el nivel 5 (183,09 °C).

8.1.6. Nivel 6: 205,26°C

Delta de Temperatura (°C)	Energía (Wh)	Potencia (W)
186,26	627160	348,42
87,4	160040	88,91
99,83	203160	112,87
122,57	239680	133,16
142,38	342200	190,11
164,39	439520	244,18

TABLA 24: Relación entre la energía, potencia y variación de temperatura del horno en el nivel 6 (205,26 °C).

8.2. Código fuente del sistema de medición en Raspberry Pi Pico

En esta sección se presenta el código fuente completo desarrollado en MicroPython para la Raspberry Pi Pico, encargado de la adquisición y procesamiento de datos del sistema de medición. El programa integra la lectura de sensores analógicos (temperatura del horno), digitales (DHT11 para temperatura ambiente) y comunicación Modbus RTU para obtener parámetros eléctricos del medidor CVM-C10. Incluye algoritmos de filtrado digital, detección de drift, promedio móvil y almacenamiento de datos en formato CSV, garantizando la estabilidad y confiabilidad de las mediciones durante todo el proceso experimental.

```
1 from machine import ADC, Pin, UART
2 import dht
3 import utime
4 import time
5 import math
6 import os
7
8 # ===== CONFIGURACIÓN ADC Y SENSORES =====
9 adc_temperatura = ADC(Pin(27)) # Sensor de temperatura
10 sensor_dht = dht.DHT11(Pin(18)) # Sensor DHT11
11
12 # Constantes
13 AREF = 3.237
14
15 # Variables para estabilización de temperatura
16 buffer_temperatura = []
17 BUFFER_SIZE = 10
18 VOLTAJE_MIN_ESPERADO = 0.5
19 VOLTAJE_MAX_ESPERADO = 3.0
20
21 # Control de drift y desviación
22 MAX_DESVIACION_PERMITIDA = 0.015
23 CONTADOR_DRIFT = 0
24 MAX_DRIFT_COUNT = 5
25 VALOR_REFERENCIA = None
26 LECTURAS_PARA_REFERENCIA = 50
27
28 inicio_ms = utime.ticks_ms()
29
30 # ===== CONFIGURACIÓN MODBUS =====
31 TX_PIN = 0
32 RX_PIN = 1
33 MAX485_EN = 26
```

```
34 UART_ID = 0
35 BAUDRATE = 9600
36 SLAVE_ID = 10
37
38 # Inicializa UART
39 uart = UART(UART_ID, baudrate=BAUDRATE, tx=Pin(TX_PIN), rx=Pin(RX_PIN))
40 # Inicializa MAX485
41 ctrl_pin = Pin(MAX485_EN, Pin.OUT)
42 ctrl_pin.value(0)
43
44 # ===== FUNCIONES MODBUS =====
45 def calculate_crc(frame):
46     """Calcula el CRC para verificación Modbus"""
47     crc = 0xFFFF
48     for pos in frame:
49         crc ^= pos
50         for _ in range(8):
51             if crc & 0x0001:
52                 crc >>= 1
53                 crc ^= 0xA001
54             else:
55                 crc >>= 1
56     return crc
57
58 def construct_modbus_request(slave, func, start, count):
59     """Construye una petición Modbus"""
60     frame = [
61         slave,
62         func,
63         (start >> 8) & 0xFF,
64         start & 0xFF,
65         (count >> 8) & 0xFF,
66         count & 0xFF
67     ]
68     crc = calculate_crc(frame)
69     frame.append(crc & 0xFF)
70     frame.append((crc >> 8) & 0xFF)
71     return bytes(frame)
72
73 def verify_crc(frame):
74     """Verifica el CRC de la respuesta Modbus"""
75     received_crc = (frame[-1] << 8) | frame[-2]
76     calculated_crc = calculate_crc(frame[:-2])
77     return received_crc == calculated_crc
78
79 def send_modbus_request(frame):
```

```
80     """Envía petición Modbus"""
81     ctrl_pin.value(1)
82     time.sleep_ms(10)
83     uart.write(frame)
84     time.sleep_ms(10)
85     ctrl_pin.value(0)
86
87 def read_modbus_response(expected_length):
88     """Lee respuesta Modbus con timeout"""
89     start = time.ticks_ms()
90     response = bytearray()
91     while time.ticks_diff(time.ticks_ms(), start) < 1000:
92         if uart.any():
93             response.extend(uart.read())
94             if len(response) >= expected_length:
95                 break
96             time.sleep_ms(5)
97     return response if len(response) >= expected_length else None
98
99 def parse_registers(frame):
100     """Extrae los registros de la respuesta Modbus"""
101     byte_count = frame[2]
102     data = frame[3:3+byte_count]
103     registers = []
104     for i in range(0, len(data), 2):
105         val = (data[i] << 8) | data[i+1]
106         registers.append(val)
107     return registers
108
109 def read_voltage():
110     """Lee voltaje del medidor CVM-C10"""
111     try:
112         request = construct_modbus_request(SLAVE_ID, 0x04, 0x0000, 2)
113         send_modbus_request(request)
114         resp = read_modbus_response(9)
115         if resp and verify_crc(resp):
116             regs = parse_registers(resp)
117             voltage = regs[1] / 10.0
118             return voltage
119         else:
120             return -99.0
121     except:
122         return -99.0
123
124 def read_current():
125     """Lee corriente del medidor CVM-C10"""
```

```
126     try:
127         request = construct_modbus_request(SLAVE_ID, 0x04, 0x0002, 2)
128         send_modbus_request(request)
129         resp = read_modbus_response(9)
130         if resp and verify_crc(resp):
131             regs = parse_registers(resp)
132             current = regs[1] / 1000.0
133             return current
134         else:
135             return -99.0
136     except:
137         return -99.0
138
139 def read_power():
140     """Lee potencia del medidor CVM-C10"""
141     try:
142         request = construct_modbus_request(SLAVE_ID, 0x04, 0x0004, 2)
143         send_modbus_request(request)
144         resp = read_modbus_response(9)
145         if resp and verify_crc(resp):
146             regs = parse_registers(resp)
147             power = regs[1]
148             return power
149         else:
150             return -99.0
151     except:
152         return -99.0
153
154 # ===== FUNCIONES DE LECTURA DE TEMPERATURA =====
155 def read_voltage_temperatura_mejorada(n=800):
156     """
157     Lee el voltaje del ADC con filtrado de lecturas inválidas
158     """
159     lecturas = []
160     lecturas_validas = 0
161     intentos = 0
162     max_intentos = n * 2
163
164     while lecturas_validas < n and intentos < max_intentos:
165         raw = adc_temperatura.read_ul6()
166         voltage = (raw / 65535) * AREF
167
168         # Filtrar lecturas fuera del rango esperado
169         if VOLTAJE_MIN_ESPERADO <= voltage <= VOLTAJE_MAX_ESPERADO:
170             lecturas.append(voltage)
171             lecturas_validas += 1
```

```
172
173     intentos += 1
174     utime.sleep_us(200)
175
176     if len(lecturas) < 100:
177         print(f"Advertencia: Solo {len(lecturas)} lecturas válidas")
178         return None
179
180     # Aplicar filtro IQR: eliminar el 20% superior e inferior
181     lecturas.sort()
182     inicio = len(lecturas) // 5
183     fin = len(lecturas) - inicio
184     lecturas_filtradas = lecturas[inicio:fin]
185
186     # Calcular promedio
187     promedio = sum(lecturas_filtradas) / len(lecturas_filtradas)
188
189     # Calcular desviación estándar
190     varianza = sum((x - promedio) ** 2 for x in lecturas_filtradas) / len(lecturas_filtradas)
191     desviacion = math.sqrt(varianza)
192
193     # Si la desviación es alta, aplicar filtro adicional
194     if desviacion > 0.01:
195         lecturas_estables = [x for x in lecturas_filtradas if abs(x - promedio) <= 2 *
196                             desviacion]
197         if len(lecturas_estables) > 50:
198             promedio = sum(lecturas_estables) / len(lecturas_estables)
199
200     return promedio
201
202 def promedio_movil_controlado(nuevo_valor, buffer, tamaño_buffer):
203     """
204     Calcula promedio móvil con detección de drift
205     """
206     global CONTADOR_DRIFT, VALOR_REFERENCIA
207
208     if len(buffer) == 0:
209         buffer.append(nuevo_valor)
210         return nuevo_valor
211
212     promedio_actual = sum(buffer) / len(buffer)
213     diferencia = abs(nuevo_valor - promedio_actual)
214
215     # Detectar drift (desviación excesiva)
216     if diferencia > MAX_DESVIACION_PERMITIDA * 3:
217         CONTADOR_DRIFT += 1
```

```
217     print(f"Drift detectado: diff={diferencia:.6f}V (count: {CONTADOR_DRIFT})")
218
219     # Si hay drift persistente, resetear buffer parcialmente
220     if CONTADOR_DRIFT >= MAX_DRIFT_COUNT:
221         print("Reseteando buffer por drift persistente")
222         buffer_temp = buffer[-3:] if len(buffer) >= 3 else buffer
223         buffer.clear()
224         buffer.extend(buffer_temp)
225         CONTADOR_DRIFT = 0
226     else:
227         if CONTADOR_DRIFT > 0:
228             CONTADOR_DRIFT = max(0, CONTADOR_DRIFT - 1)
229
230     buffer.append(nuevo_valor)
231     if len(buffer) > tamaño_buffer:
232         buffer.pop(0)
233
234     return sum(buffer) / len(buffer)
235
236 def control_desviacion_acumulativa(buffer, lectura_count):
237     """
238     Controla que la desviación no se acumule indefinidamente
239     """
240     global VALOR_REFERENCIA
241
242     if len(buffer) < 5:
243         return False
244
245     promedio = sum(buffer) / len(buffer)
246     varianza = sum((x - promedio) ** 2 for x in buffer) / len(buffer)
247     desviacion = math.sqrt(varianza)
248
249     # Actualizar valor de referencia periódicamente
250     if lectura_count % LECTURAS_PARA_REFERENCIA == 0:
251         VALOR_REFERENCIA = promedio
252         print(f"Actualizando referencia: {VALOR_REFERENCIA:.4f}V")
253
254     # Verificar drift acumulativo
255     if VALOR_REFERENCIA is not None:
256         drift_total = abs(promedio - VALOR_REFERENCIA)
257         if drift_total > 0.05:
258             print(f"Drift acumulativo detectado: {drift_total:.6f}V")
259             return True
260
261     # Verificar desviación excesiva
262     if desviacion > MAX_DESVIACION_PERMITIDA:
```

```
263     print(f"Desviación alta: {desviacion:.6f}V > {MAX_DESVIACION_PERMITIDA:.6f}V")
264     return True
265
266     return False
267
268 def resetear_sistema_medicion():
269     """
270     Resetea el sistema de medición cuando hay problemas persistentes
271     """
272     global buffer_temperatura, CONTADOR_DRIFT, VALOR_REFERENCIA
273
274     print("RESETEANDO SISTEMA DE MEDICIÓN")
275     buffer_temperatura.clear()
276     CONTADOR_DRIFT = 0
277     VALOR_REFERENCIA = None
278
279     # Re-calibrar ADC
280     print("Re-calibrando ADC...")
281     for i in range(20):
282         adc_temperatura.read_u16()
283         utime.sleep_ms(10)
284     print("Re-calibración completada")
285
286 def calibrar_adc():
287     """
288     Calibración inicial del ADC para estabilizar lecturas
289     """
290     print("Calibrando ADC... Esto puede tomar unos segundos.")
291
292     for i in range(50):
293         adc_temperatura.read_u16()
294         utime.sleep_ms(10)
295         if i % 10 == 0:
296             print(f"Calibrando... {(i+1)*2}%")
297
298     print("Calibración completada.")
299
300 def calcular_temperatura(voltage, m, b, offset):
301     """
302     Calcula temperatura usando la ecuación de linealización
303      $T = ((V - b) / m) + \text{offset}$ 
304     """
305     return ((voltage - b) / m) + offset
306
307 # ===== TIMESTAMP =====
308 contador_lecturas = 0
```

```

309
310 def obtener_timestamp_real():
311     """Genera timestamp basado en tiempo real transcurrido"""
312     ms = utime.ticks_diff(utime.ticks_ms(), inicio_ms)
313     s = ms // 1000
314     h = s // 3600
315     m = (s % 3600) // 60
316     sec = s % 60
317     return "{:02d}:{:02d}:{:02d}".format(h, m, sec)
318
319 # ===== FUNCIONES CSV =====
320 def inicializar_csv():
321     """Crea el archivo CSV con encabezados"""
322     with open("datos_sensores.csv", "w") as f:
323         f.write("\ufeff")
324         f.write("Timestamp\tVoltaje_Temp_V\tTemperatura_Horno_C\tTemperatura_Ambiente_C
325             \tDesviacion_V\tModbus_Voltaje_V\tModbus_Corriente_A\tModbus_Potencia_W\n")
326
327 def guardar_en_csv_linea(ts, vt, tc, ta, desv, mb_v, mb_c, mb_p):
328     """Guarda una línea de datos en el CSV"""
329     try:
330         with open("datos_sensores.csv", "a") as f:
331             # Convertir números a string con coma decimal
332             vt_str = f"{vt:.4f}".replace('.', ',')
333             tc_str = f"{tc:.2f}".replace('.', ',')
334             ta_str = f"{ta:.2f}".replace('.', ',')
335             desv_str = f"{desv:.6f}".replace('.', ',')
336             mb_v_str = f"{mb_v:.2f}".replace('.', ',')
337             mb_c_str = f"{mb_c:.3f}".replace('.', ',')
338             mb_p_str = f"{mb_p:.1f}".replace('.', ',')
339
340             linea =
341             f"{ts}\t{vt_str}\t{tc_str}\t{ta_str}\t{desv_str}\t{mb_v_str}\t{mb_c_str}\t{mb_p_str}\n"
342             f.write(linea)
343     except Exception as e:
344         print("Error al guardar CSV:", e)
345
346 # ===== PROGRAMA PRINCIPAL =====
347 print("=== SISTEMA DE MONITOREO COMPLETO ===")
348
349 # Coeficientes de la ecuación de linealización
350 m = 0.0055
351 b = 1.4042
352 offset = -10.2
353

```

```
354 # Calibración inicial del ADC
355 calibrar_adc()
356
357 # Reiniciar contador de tiempo después de calibración
358 inicio_ms = utime.ticks_ms()
359
360 # Inicializar archivo CSV
361 inicializar_csv()
362 print(f"\nCoeficientes: m={m}, b={b}, offset={offset}")
363 print("Presiona Ctrl+C para detener y guardar.")
364 print("-" * 80)
365
366 # Variables para estadísticas
367 lecturas_count = 0
368 ultima_temperatura = None
369
370 try:
371     while True:
372         datos_validos = True
373
374         # Leer voltaje de temperatura
375         vt_raw = read_voltage_temperatura_mejorada(800)
376
377         if vt_raw is None:
378             print("Error: No se pudieron obtener lecturas estables")
379             datos_validos = False
380         else:
381             # Aplicar promedio móvil
382             vt = promedio_movil_controlado(vt_raw, buffer_temperatura, BUFFER_SIZE)
383
384             # Verificar drift acumulativo
385             if control_desviacion_acumulativa(buffer_temperatura, lecturas_count + 1):
386                 resetear_sistema_medicion()
387                 vt_raw = read_voltage_temperatura_mejorada(500)
388                 if vt_raw is not None:
389                     vt = promedio_movil_controlado(vt_raw, buffer_temperatura, BUFFER_SIZE)
390                 else:
391                     print("Error crítico en lectura ADC")
392                     datos_validos = False
393
394             if datos_validos:
395                 # Detectar cambios bruscos
396                 if ultima_temperatura is not None:
397                     diff_temp = abs(vt - ultima_temperatura)
398                     if diff_temp > 0.1:
399                         print(f"Advertencia: Cambio brusco detectado: {diff_temp:.4f}V")
```

```
400
401     # Calcular temperatura del horno
402     t_horno = calcular_temperatura(vt, m, b, offset)
403
404     # Leer temperatura ambiente
405     try:
406         sensor_dht.measure()
407         t_amb_raw = sensor_dht.temperature()
408         t_amb = t_amb_raw - 2.0
409     except Exception as e:
410         t_amb = -99.0
411         datos_validos = False
412
413     # Lecturas Modbus
414     mb_voltaje = read_voltage()
415     if mb_voltaje == -99.0:
416         datos_validos = False
417     time.sleep_ms(40)
418
419     mb_corriente = read_current()
420     if mb_corriente == -99.0:
421         datos_validos = False
422     time.sleep_ms(40)
423
424     mb_potencia = read_power()
425     if mb_potencia == -99.0:
426         datos_validos = False
427     time.sleep_ms(40)
428
429     # Obtener timestamp
430     ts = obtener_timestamp_real()
431
432     # Guardar y mostrar datos si son válidos
433     if datos_validos:
434         # Calcular desviación
435         desviacion = 0
436         if len(buffer_temperatura) > 1:
437             prom = sum(buffer_temperatura) / len(buffer_temperatura)
438             desviacion = math.sqrt(sum((x - prom) ** 2 for x in buffer_temperatura) /
439 len(buffer_temperatura))
440
441             print(f"{ts} | {vt:6.4f}V | {t_horno:6.2f}°C | {t_amb:6.1f}°C |  $\sigma$ 
442 :{desviacion:.6f}")
443
444             guardar_en_csv_linea(ts, vt, t_horno, t_amb, desviacion, mb_voltaje,
445 mb_corriente, mb_potencia)
```

```

443     ultima_temperatura = vt
444     lecturas_count += 1
445
446     # Mostrar estadísticas cuando hay drift significativo
447     if lecturas_count % 50 == 0:
448         drift_ref = 0
449         if VALOR_REFERENCIA is not None and len(buffer_temperatura) > 0:
450             drift_ref = abs(sum(buffer_temperatura) / len(buffer_temperatura) -
451 VALOR_REFERENCIA)
452             if drift_ref > 0.01:
453                 print(f"--- Drift detectado: {drift_ref:.6f}V después de
{lecturas_count} lecturas ---")
454             else:
455                 print(f"{ts} | ERROR EN DATOS - No se guardará esta lectura")
456
457     # Tiempo de muestreo
458     utime.sleep(1)
459
460 except KeyboardInterrupt:
461     print("\nDeteniendo monitoreo...")
462     print(f"Total de lecturas: {lecturas_count}")
463     print("Datos guardados correctamente.")

```

8.3. Código Validación Modelo del Horno

En esta sección se presenta el código Python utilizado para la validación y comparación de los modelos matemáticos del comportamiento térmico del horno. El programa implementa tres enfoques de modelado: lineal, polinómico de segundo grado y exponencial, utilizando los datos experimentales de temperatura y potencia promedio obtenidos durante las pruebas. Mediante las bibliotecas NumPy, Matplotlib y Scikit-learn, se calculan las métricas de desempeño (RMSE y R^2) de cada modelo y se genera una visualización comparativa que permite identificar el modelo que mejor ajusta el comportamiento real del sistema.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from sklearn.metrics import mean_squared_error, r2_score
4 from scipy.optimize import curve_fit
5
6 # -----
7 # DATOS (Temperatura °C, Potencia Promedio W)

```

```

8 # -----
9 data = np.array([
10 [87.91,135.4],[97.63,129.2],[122.58,163.9333333],[145.37,218.9333333],
11 [165.88,275.6],[188.21,345.8222222],[100.4,164.9777778],[84.4,96.84444444],
12 [122.17,156.9555556],[143.54,221.1777778],[165.09,274.6222222],[187.89,330.9777778],
13 [125.21,211.0222222],[84.4,86.8],[98.56,124.4888889],[144.94,210.3555556],
14 [165.27,274.2444444],[185.65,311.3111111],[143.64,242.4666667],[143.54,190.1111111],
15 [185.43,328.0222222],[146.37,190.1111111],[163.44,264.2444444],[144.09,194.2222222],
16 [167.09,295.2444444],[86.4,82.51111111],[99.59,108.1777778],[124.09,155.0222222],
17 [144.37,194.2222222],[187.26,300.7777778],[186.26,348.4222222],[87.4,88.91111111],
18 [99.83,112.8666667],[122.57,133.1555556],[142.38,190.1111111],[164.39,244.1777778]
19 ])
20
21 T = data[:,0] # Temperatura
22 P = data[:,1] # Potencia promedio
23
24 # -----
25 # MODELOS
26 # -----
27
28 # Modelo Lineal
29 coef_lin = np.polyfit(T, P, 1)
30 P_pred_lin = np.polyval(coef_lin, T)
31
32 # Modelo Polinómico (cuadrático)
33 coef_poly = np.polyfit(T, P, 2)
34 P_pred_poly = np.polyval(coef_poly, T)
35
36 # Modelo Exponencial:  $P(T) = a * \exp(b*T)$ 
37 def model_exp(T, a, b):
38     return a * np.exp(b * T)
39
40 popt, _ = curve_fit(model_exp, T, P, p0=[40, 0.01])
41 P_pred_exp = model_exp(T, *popt)
42
43 # -----
44 # MÉTRICAS
45 # -----
46 def metrics(P, P_pred):
47     rmse = np.sqrt(mean_squared_error(P, P_pred))
48     r2 = r2_score(P, P_pred)
49     return rmse, r2
50
51 rmse_lin, r2_lin = metrics(P, P_pred_lin)
52 rmse_poly, r2_poly = metrics(P, P_pred_poly)
53 rmse_exp, r2_exp = metrics(P, P_pred_exp)

```

```

54
55 print("Modelo Lineal: P(T) = {:.4f}*T + {:.4f}".format(coef_lin[0], coef_lin[1]))
56 print("  RMSE = {:.4f}, R  = {:.4f}".format(rmse_lin, r2_lin))
57
58 print("\nModelo Polinómico: P(T) = {:.4e}*T  + {:.4f}*T + {:.4f}".format(coef_poly[0],
    coef_poly[1], coef_poly[2]))
59 print("  RMSE = {:.4f}, R  = {:.4f}".format(rmse_poly, r2_poly))
60
61 print("\nModelo Exponencial: P(T) = {:.4f} * exp({:.6f} * T)".format(popt[0], popt[1]))
62 print("  RMSE = {:.4f}, R  = {:.4f}".format(rmse_exp, r2_exp))
63
64 # -----
65 # GRAFICAR
66 # -----
67 T_line = np.linspace(min(T), max(T), 200)
68 plt.scatter(T, P, label="Datos", color="black")
69
70 plt.plot(T_line, np.polyval(coef_lin, T_line), label="Lineal", linestyle="--")
71 plt.plot(T_line, np.polyval(coef_poly, T_line), label="Polinómico (2)", linestyle="-.")
72 plt.plot(T_line, model_exp(T_line, *popt), label="Exponencial", linestyle="-")
73
74 plt.xlabel("Temperatura [°C]")
75 plt.ylabel("Potencia promedio [W]")
76 plt.title("Modelado de Potencia en función de Temperatura")
77 plt.legend()
78 plt.grid(True)
79 plt.show()

```

8.4. Código Monitoreo Microrred

En esta sección se presenta el código del sistema de monitoreo y control automático del horno integrado con la microrred fotovoltaica. El programa implementa un sistema inteligente de predicción térmica utilizando las ecuaciones de inercia calculadas del horno para determinar los tiempos óptimos de operación. Integra comunicación Modbus RTU dual para lectura del medidor CVM-C10 y del inversor fotovoltaico, sincronización mediante RTC, sistema de avisos programados por horarios de cocción, y algoritmos de suavizado para las predicciones. El sistema calcula continuamente el tiempo necesario para alcanzar las temperaturas objetivo y genera alertas de urgencia para maximizar el aprovechamiento de la energía solar disponible.

```

1 from machine import ADC, Pin, UART, RTC
2 import dht

```

```
3 import utime
4 import time
5 import math
6 import os
7
8 # ===== CONFIGURACIÓN RTC (HORA REAL) =====
9 rtc = RTC()
10 # DESCOMENTA Y AJUSTA CON LA FECHA Y HORA ACTUAL:
11 # rtc.datetime((2024, 10, 30, 2, 18, 8, 0, 0)) # Miércoles 30 oct 2024, 18:08:00
12
13 # ===== CONFIGURACIÓN ADC Y SENSORES =====
14 adc_temperatura = ADC(Pin(27))
15 sensor_dht = dht.DHT11(Pin(18))
16
17 # Constantes
18 AREF = 3.237
19 buffer_temperatura = []
20 BUFFER_SIZE = 10
21 VOLTAJE_MIN_ESPERADO = 0.5
22 VOLTAJE_MAX_ESPERADO = 3.0
23 MAX_DESVIACION_PERMITIDA = 0.015
24 CONTADOR_DRIFT = 0
25 MAX_DRIFT_COUNT = 5
26 VALOR_REFERENCIA = None
27 LECTURAS_PARA_REFERENCIA = 50
28
29 # ===== CONFIGURACIÓN MODBUS CVM-C10 =====
30 TX_PIN_CVM = 4
31 RX_PIN_CVM = 5
32 MAX485_EN_CVM = 28
33 UART_ID_CVM = 1
34 BAUDRATE_CVM = 9600
35 SLAVE_ID_CVM = 10
36
37 uart_cvm = UART(UART_ID_CVM, baudrate=BAUDRATE_CVM, tx=Pin(TX_PIN_CVM), rx=Pin(RX_PIN_CVM),
38                bits=8, parity=None, stop=1, timeout=500, rxbuf=256)
39 ctrl_pin_cvm = Pin(MAX485_EN_CVM, Pin.OUT)
40 ctrl_pin_cvm.value(0)
41
42 # ===== CONFIGURACIÓN MODBUS INVERSOR =====
43 TX_PIN_INV = 0
44 RX_PIN_INV = 1
45 MAX485_EN_INV = 26
46 UART_ID_INV = 0
47 BAUDRATE_INV = 9600
48 SLAVE_ID_INV = 5
```

```
49
50 uart_inv = UART(UART_ID_INV, baudrate=BAUDRATE_INV, tx=Pin(TX_PIN_INV), rx=Pin(RX_PIN_INV),
51                 bits=8, parity=None, stop=1, timeout=500, rxbuf=256)
52 ctrl_pin_inv = Pin(MAX485_EN_INV, Pin.OUT)
53 ctrl_pin_inv.value(0)
54
55 # ===== ECUACIONES TÉRMICAS (MUESTRAS CADA 5 MIN) =====
56 # ENFRIAMIENTO: desde temperatura máxima hasta ambiente
57 ECUACIONES_ENFRIAMIENTO = {
58     95: [1.038826e+02, -9.324831e-02, 4.068553e-05, -7.627196e-09, 4.905664e-13],
59     120: [1.241473e+02, -1.134699e-01, 4.617908e-05, -7.276968e-09, 2.868682e-13],
60     150: [1.309973e+02, -1.384918e-01, 7.161386e-05, -1.692388e-08, 1.504307e-12],
61     180: [1.632157e+02, -1.993894e-01, 1.148856e-04, -3.039488e-08, 3.015946e-12],
62     200: [1.851743e+02, -2.310612e-01, 1.310743e-04, -3.343257e-08, 3.160322e-12],
63     230: [2.052943e+02, -2.739231e-01, 1.688003e-04, -4.679886e-08, 4.766218e-12],
64 }
65
66 # CALENTAMIENTO: desde temperatura ambiente hasta máxima
67 ECUACIONES_CALENTAMIENTO = {
68     95: [3.279822e+01, 2.888772e-01, -2.910547e-04, 1.076018e-07, -1.324524e-11],
69     120: [3.620169e+01, 3.298065e-01, -3.324235e-04, 1.229343e-07, -1.514597e-11],
70     150: [3.949232e+01, 3.928951e-01, -3.870246e-04, 1.411520e-07, -1.721997e-11],
71     180: [4.870639e+01, 4.363204e-01, -4.303843e-04, 1.572791e-07, -1.922377e-11],
72     200: [5.105295e+01, 5.361830e-01, -5.444281e-04, 2.028732e-07, -2.516496e-11],
73     230: [5.200409e+01, 5.372131e-01, -5.177102e-04, 1.874376e-07, -2.280719e-11],
74 }
75
76 # Mapeo de niveles del horno a temperaturas objetivo
77 NIVEL_A_TEMP = {
78     1: 104,
79     2: 117,
80     3: 142,
81     4: 163,
82     5: 183,
83     6: 20
84 }
85
86 # ===== HORARIOS DE COCCIÓN =====
87 HORARIOS_COCCION = [
88     {"hora": "11:30", "temp_objetivo": 20, "descripcion": "Inicio coccion principal"},
89     {"hora": "12:00", "temp_objetivo": 200, "descripcion": "Inicio coccion principal"},
90     {"hora": "18:00", "temp_objetivo": 40, "descripcion": "Enfriamiento (sin paneles)"},
91     {"hora": "19:00", "temp_objetivo": 200, "descripcion": "Primera recarga"},
92     {"hora": "19:30", "temp_objetivo": 50, "descripcion": "Enfriamiento ciclo 1"},
93     {"hora": "20:00", "temp_objetivo": 200, "descripcion": "Segunda recarga"},
94     {"hora": "20:30", "temp_objetivo": 95, "descripcion": "Enfriamiento ciclo 2"},
```

```
95     {"hora": "21:00", "temp_objetivo": 120, "descripcion": "Mantenimiento final"}
96 ]
97
98 # ===== INTERVALOS =====
99 INTERVALO_MEDICION = 2 # Cada 2 segundos - NO CAMBIAR
100 INTERVALO_RECALCULO = 300 # MODIFICADO: Recalcular cada 60 segundos (antes era 300)
101 RANGO_TEMPERATURA = 10 # Rango 10 °C
102 DELTA_TIEMPO_SEGURIDAD = 180 # 3 minutos de margen para compensar retrasos
103 MAX_LINEAS_CSV = 21600 # 12 horas exactas
104
105 # ===== VARIABLES DE SUAVIZADO (NUEVO) =====
106 tiempo_necesario_anterior = None
107 contador_cambios_bruscos = 0
108 MAX_CAMBIO_PERMITIDO = 300 # 5 minutos máximo de cambio por ciclo
109 FACTOR_SUAVIZADO = 0.3 # 30% nuevo valor, 70% valor anterior
110
111 nivel_actual_horno = 0
112 ultimo_recalculo_ms = 0
113 evento_activo_idx = None
114 evento_completado = False
115
116 # ===== FUNCIONES DE HORA REAL =====
117 def obtener_hora_actual():
118     dt = rtc.datetime()
119     return "{:02d}:{:02d}:{:02d}".format(dt[4], dt[5], dt[6])
120
121 def obtener_fecha_hora_completa():
122     dt = rtc.datetime()
123     return "{:04d}-{:02d}-{:02d} {:02d}:{:02d}:{:02d}".format(dt[0], dt[1], dt[2], dt[4],
124 dt[5], dt[6])
125
126 def configurar_hora_inicial():
127     print("\n" + "="*80)
128     print("CONFIGURACION DE HORA INICIAL")
129     print("="*80)
130     print("Hora configurada: {}".format(obtener_fecha_hora_completa()))
131     print("="*80 + "\n")
132
133 def hora_a_minutos(hora_str):
134     """Convierte 'HH:MM' a minutos totales"""
135     partes = hora_str.split(":")
136     return int(partes[0]) * 60 + int(partes[1])
137
138 # ===== FUNCIONES DE PREDICCIÓN TÉRMICA =====
139 def calcular_temperatura_enfriamiento(nivel_temp, tiempo_segundos):
140     """Predice temperatura durante ENFRIAMIENTO usando ecuaciones polinómicas"""
```

```
140     if nivel_temp not in ECUACIONES_ENFRIAMIENTO:
141         return None
142     coef = ECUACIONES_ENFRIAMIENTO[nivel_temp]
143     t = tiempo_segundos
144     return coef[0] + coef[1]*t + coef[2]*(t**2) + coef[3]*(t**3) + coef[4]*(t**4)
145
146 def calcular_temperatura_calentamiento(nivel_temp, tiempo_segundos):
147     """Predice temperatura durante CALENTAMIENTO usando ecuaciones polinómicas"""
148     if nivel_temp not in ECUACIONES_CALENTAMIENTO:
149         return None
150     coef = ECUACIONES_CALENTAMIENTO[nivel_temp]
151     t = tiempo_segundos
152     return coef[0] + coef[1]*t + coef[2]*(t**2) + coef[3]*(t**3) + coef[4]*(t**4)
153
154 def encontrar_nivel_mas_cercano(temp_objetivo):
155     """Encuentra el nivel estándar más cercano a la temperatura objetivo"""
156     niveles = sorted(NIVEL_A_TEMP.values())
157     mejor = niveles[0]
158     menor_dif = abs(temp_objetivo - niveles[0])
159     for nivel in niveles[1:]:
160         dif = abs(temp_objetivo - nivel)
161         if dif < menor_dif:
162             menor_dif = dif
163             mejor = nivel
164     return mejor
165
166 def detectar_nivel_actual(temp_actual):
167     """Detecta en qué nivel está operando el horno según su temperatura"""
168     nivel_temp = encontrar_nivel_mas_cercano(temp_actual)
169     # Convertir temperatura a número de nivel (1-6)
170     for num, temp in NIVEL_A_TEMP.items():
171         if temp == nivel_temp:
172             return num
173     return 1
174
175 def temperatura_en_rango(temp_actual, temp_objetivo, rango=RANGO_TEMPERATURA):
176     """Verifica si la temperatura está en rango aceptable ( rango °C)"""
177     return (temp_objetivo - rango) <= temp_actual <= (temp_objetivo + rango)
178
179 def calcular_tiempo_calentamiento(temp_actual, temp_objetivo):
180     """
181     Calcula tiempo necesario para CALENTAR de temp_actual a temp_objetivo
182     Usa ecuaciones de calentamiento por bisección
183     CORREGIDO: Considera la temperatura actual del horno
184     """
185     if temp_actual >= temp_objetivo:
```

```
186     return 0
187
188 nivel_objetivo = encontrar_nivel_mas_cercano(temp_objetivo)
189
190 # PASO 1: Encontrar en qué tiempo t1 está la temperatura actual
191 t1_min = 0
192 t1_max = 3600
193 t1_actual = 0
194
195 for _ in range(50):
196     t1_medio = (t1_min + t1_max) / 2
197     temp_pred = calcular_temperatura_calentamiento(nivel_objetivo, t1_medio)
198
199     if temp_pred is None:
200         break
201
202     if abs(temp_pred - temp_actual) < 2.0:
203         t1_actual = t1_medio
204         break
205
206     if temp_pred < temp_actual:
207         t1_min = t1_medio
208     else:
209         t1_max = t1_medio
210 else:
211     t1_actual = t1_medio
212
213 # PASO 2: Encontrar en qué tiempo t2 se alcanza la temperatura objetivo
214 t2_min = t1_actual # Empezar desde donde estamos
215 t2_max = 3600
216 t2_objetivo = t1_actual
217
218 for _ in range(50):
219     t2_medio = (t2_min + t2_max) / 2
220     temp_pred = calcular_temperatura_calentamiento(nivel_objetivo, t2_medio)
221
222     if temp_pred is None:
223         return 1500 # Default: 25 minutos
224
225     if abs(temp_pred - temp_objetivo) < 2.0:
226         t2_objetivo = t2_medio
227         break
228
229     if temp_pred < temp_objetivo:
230         t2_min = t2_medio
231     else:
```

```
232         t2_max = t2_medio
233     else:
234         t2_objetivo = t2_medio
235
236     # PASO 3: El tiempo necesario es la diferencia
237     tiempo_necesario = int(t2_objetivo - t1_actual)
238
239     # Si el resultado es negativo o muy pequeño, usar mínimo
240     if tiempo_necesario < 60:
241         return 60 # Mínimo 1 minuto
242
243     return tiempo_necesario + DELTA_TIEMPO_SEGURIDAD
244
245 def calcular_tiempo_enfriamiento(temp_actual, temp_objetivo):
246     """
247     Calcula tiempo necesario para ENFRIAR de temp_actual a temp_objetivo
248     Usa ecuaciones de enfriamiento por bisección
249     CORREGIDO: Considera la temperatura actual del horno
250     """
251     if temp_actual <= temp_objetivo:
252         return 0
253
254     nivel_actual_temp = encontrar_nivel_mas_cercano(temp_actual)
255
256     # PASO 1: Encontrar en qué tiempo t1 está la temperatura actual
257     # (en enfriamiento, t=0 es la temperatura máxima)
258     t1_min = 0
259     t1_max = 7200
260     t1_actual = 0
261
262     for _ in range(50):
263         t1_medio = (t1_min + t1_max) / 2
264         temp_pred = calcular_temperatura_enfriamiento(nivel_actual_temp, t1_medio)
265
266         if temp_pred is None:
267             break
268
269         if abs(temp_pred - temp_actual) < 2.0:
270             t1_actual = t1_medio
271             break
272
273         if temp_pred > temp_actual: # En enfriamiento: mayor temp = menor tiempo
274             t1_min = t1_medio
275     else:
276         t1_max = t1_medio
277 else:
```

```
278     t1_actual = t1_medio
279
280     # PASO 2: Encontrar en qué tiempo t2 se alcanza la temperatura objetivo
281     t2_min = t1_actual # Empezar desde donde estamos
282     t2_max = 7200
283     t2_objetivo = t1_actual
284
285     for _ in range(50):
286         t2_medio = (t2_min + t2_max) / 2
287         temp_pred = calcular_temperatura_enfriamiento(nivel_actual_temp, t2_medio)
288
289         if temp_pred is None:
290             return 1800 # Default: 30 minutos
291
292         if abs(temp_pred - temp_objetivo) < 2.0:
293             t2_objetivo = t2_medio
294             break
295
296         if temp_pred > temp_objetivo: # En enfriamiento: mayor temp = menor tiempo
297             t2_min = t2_medio
298         else:
299             t2_max = t2_medio
300     else:
301         t2_objetivo = t2_medio
302
303     # PASO 3: El tiempo necesario es la diferencia
304     tiempo_necesario = int(t2_objetivo - t1_actual)
305
306     # Si el resultado es negativo o muy pequeño, usar mínimo
307     if tiempo_necesario < 60:
308         return 60 # Mínimo 1 minuto
309
310     return tiempo_necesario + DELTA_TIEMPO_SEGURIDAD
311
312 def formatear_tiempo(segundos):
313     """Convierte segundos a formato HH:MM:SS"""
314     h = segundos // 3600
315     m = (segundos % 3600) // 60
316     s = segundos % 60
317     return "{:02d}:{:02d}:{:02d}".format(h, m, s)
318
319 def suavizar_tiempo_necesario(tiempo_nuevo, temp_actual, temp_objetivo):
320     """
321     NUEVA FUNCIÓN: Suaviza cambios bruscos en el tiempo necesario calculado
322     Evita saltos de 57min a 5min aplicando un filtro de suavizado
323     """
```

```
324 global tiempo_necesario_anterior, contador_cambios Bruscos
325
326 # Primera vez
327 if tiempo_necesario_anterior is None:
328     tiempo_necesario_anterior = tiempo_nuevo
329     return tiempo_nuevo
330
331 # Calcular cambio
332 cambio = abs(tiempo_nuevo - tiempo_necesario_anterior)
333
334 # Si el cambio es muy brusco
335 if cambio > MAX_CAMBIO_PERMITIDO:
336     contador_cambios Bruscos += 1
337
338 # Si hay múltiples cambios bruscos consecutivos, aceptar el nuevo valor
339 # (significa que hubo un cambio real de fase térmica)
340 if contador_cambios Bruscos >= 3:
341     tiempo_necesario_anterior = tiempo_nuevo
342     contador_cambios Bruscos = 0
343     return tiempo_nuevo
344
345 # Suavizar el cambio usando promedio ponderado
346 tiempo_suavizado = int(tiempo_necesario_anterior * (1 - FACTOR_SUAVIZADO) +
347                        tiempo_nuevo * FACTOR_SUAVIZADO)
348 tiempo_necesario_anterior = tiempo_suavizado
349 return tiempo_suavizado
350 else:
351     # Cambio normal, resetear contador
352     contador_cambios Bruscos = 0
353     tiempo_necesario_anterior = tiempo_nuevo
354     return tiempo_nuevo
355
356 def calcular_avisos_mejorado(hora_actual_str, temp_actual):
357     global evento_activo_idx, evento_completado
358
359     hora_actual_min = hora_a_minutos(hora_actual_str)
360
361     # Si no hay evento activo, buscar el evento apropiado
362     if evento_activo_idx is None:
363         # Primero: buscar eventos que ya pasaron pero están cerca (últimos 30 min)
364         for idx, horario in enumerate(HORARIOS_COCCION):
365             hora_evento_min = hora_a_minutos(horario["hora"])
366             dif_min = hora_actual_min - hora_evento_min
367             # Si el evento pasó hace menos de 30 min, considerarlo activo
368             if 0 <= dif_min <= 30:
369                 evento_activo_idx = idx
```

```
370         evento_completado = False
371         break
372
373     # Si no encontramos evento reciente, buscar el próximo futuro
374     if evento_activo_idx is None:
375         for idx, horario in enumerate(HORARIOS_COCCION):
376             hora_evento_min = hora_a_minutos(horario["hora"])
377             if hora_evento_min > hora_actual_min:
378                 evento_activo_idx = idx
379                 evento_completado = False
380                 break
381
382     if evento_activo_idx is None:
383         return None # No hay más eventos hoy
384
385     # Obtener evento activo
386     evento = HORARIOS_COCCION[evento_activo_idx]
387     hora_evento_min = hora_a_minutos(evento["hora"])
388     temp_obj = evento["temp_objetivo"]
389     desc = evento["descripcion"]
390
391     # Calcular tiempo restante hasta el evento
392     tiempo_restante_min = hora_evento_min - hora_actual_min
393     tiempo_restante_seg = tiempo_restante_min * 60
394
395     # Si ya pasó la hora del evento
396     if tiempo_restante_min <= 0:
397         # Verificar si alcanzamos la temperatura
398         if temperatura_en_rango(temp_actual, temp_obj, RANGO_TEMPERATURA):
399             evento_completado = True
400             # Avanzar al siguiente evento
401             if evento_activo_idx < len(HORARIOS_COCCION) - 1:
402                 evento_activo_idx += 1
403                 evento_completado = False
404                 return calcular_avisos_mejorado(hora_actual_str, temp_actual)
405             else:
406                 return None # Terminamos todos los eventos
407         else:
408             # Aún no alcanzamos la temperatura - MANTENER EVENTO ACTIVO
409             # No avanzar, seguir mostrando el mismo evento
410             tiempo_restante_seg = 0 # Ya pasó el tiempo, pero seguimos intentando
411
412     # Determinar acción necesaria
413     if temperatura_en_rango(temp_actual, temp_obj, RANGO_TEMPERATURA):
414         accion = "MANTENER"
415         nivel_sug = nivel_actual_horno if nivel_actual_horno > 0 else 1
```

```
416     tiempo_nec = 0
417     urgencia = "OK"
418     detalle = "Temperatura en rango correcto"
419 elif temp_obj > temp_actual:
420     # NECESITA CALENTAR - CON SUAVIZADO
421     tiempo_nec_raw = calcular_tiempo_calentamiento(temp_actual, temp_obj)
422     tiempo_nec = suavizar_tiempo_necesario(tiempo_nec_raw, temp_actual, temp_obj)
423
424     nivel_temp = encontrar_nivel_mas_cercano(temp_obj)
425     nivel_sug = [k for k, v in NIVEL_A_TEMP.items() if v == nivel_temp][0]
426
427     if nivel_actual_horno == 0:
428         accion = "ENCENDER nivel {}".format(nivel_sug)
429     elif nivel_sug > nivel_actual_horno:
430         accion = "SUBIR a nivel {}".format(nivel_sug)
431     elif nivel_sug < nivel_actual_horno:
432         accion = "BAJAR a nivel {}".format(nivel_sug)
433     else:
434         accion = "MANTENER nivel {}".format(nivel_sug)
435
436     # Determinar urgencia
437     if tiempo_nec > tiempo_restante_seg:
438         urgencia = "URGENTE"
439         detalle = "{}! Faltan {}".format(accion, formatear_tiempo(tiempo_nec -
440 tiempo_restante_seg))
441     elif tiempo_nec > tiempo_restante_seg * 0.8:
442         urgencia = "PRONTO"
443         detalle = "{} pronto (margen: {})".format(accion,
444 formatear_tiempo(tiempo_restante_seg - tiempo_nec))
445     else:
446         urgencia = "OK"
447         detalle = "{} (margen: {})".format(accion, formatear_tiempo(tiempo_restante_seg -
448 tiempo_nec))
449 else:
450     # NECESITA ENFRIAR - CON SUAVIZADO
451     tiempo_nec_raw = calcular_tiempo_enfriamiento(temp_actual, temp_obj)
452     tiempo_nec = suavizar_tiempo_necesario(tiempo_nec_raw, temp_actual, temp_obj)
453
454     accion = "APAGAR"
455     nivel_sug = 0
456
457     if tiempo_nec > tiempo_restante_seg:
458         urgencia = "URGENTE"
459         detalle = "{}! Faltan {}".format(accion, formatear_tiempo(tiempo_nec -
460 tiempo_restante_seg))
461     elif tiempo_nec > tiempo_restante_seg * 0.8:
```

```
458         urgencia = "PRONTO"
459         detalle = "{} pronto (margen: {})".format(accion,
formatear_tiempo(tiempo_restante_seg - tiempo_nec))
460     else:
461         urgencia = "OK"
462         detalle = "{} (margen: {})".format(accion, formatear_tiempo(tiempo_restante_seg -
tiempo_nec))
463
464     rango_str = "{}-{}C".format(temp_obj - RANGO_TEMPERATURA, temp_obj + RANGO_TEMPERATURA)
465
466     return {
467         'hora': evento["hora"],
468         'desc': desc,
469         'temp_obj': temp_obj,
470         'tiempo_nec': tiempo_nec,
471         'tiempo_rest': tiempo_restante_seg,
472         'nivel': nivel_sug,
473         'urgencia': urgencia,
474         'accion': accion,
475         'detalle': detalle,
476         'rango': rango_str,
477         'dif_min': tiempo_restante_min
478     }
479
480 def mostrar_aviso(aviso, hora_actual_str, temp_actual):
481     """Muestra el aviso del evento activo"""
482     if aviso is None:
483         print("\n" + "="*80)
484         print("TODOS LOS EVENTOS COMPLETADOS")
485         print("="*80)
486         return
487
488     print("\n" + "="*80)
489     print("AVISO ACTIVO - Hora: {} | Temp: {:.1f}C | Nivel: {}".format(
490         hora_actual_str, temp_actual, nivel_actual_horno if nivel_actual_horno>0 else
"Apagado"))
491     print("="*80)
492     print("\n{} [{}] {}".format(aviso['urgencia'], aviso['hora'], aviso['desc']))
493     print("  Objetivo: {}C (Rango: {})".format(aviso['temp_obj'], aviso['rango']))
494     print("  Tiempo hasta evento: {} ({} min)".format(
495         formatear_tiempo(aviso['tiempo_rest']), aviso['dif_min']))
496     print("  Tiempo necesario: {}".format(formatear_tiempo(aviso['tiempo_nec'])))
497     print("  >>> {}".format(aviso['accion']))
498     print("  {}".format(aviso['detalle']))
499     print("="*80)
500
```

```
501 # ===== FUNCIONES MODBUS =====
502 def calculate_crc(frame):
503     crc = 0xFFFF
504     for pos in frame:
505         crc ^= pos
506         for _ in range(8):
507             if crc & 0x0001:
508                 crc >>= 1
509                 crc ^= 0xA001
510             else:
511                 crc >>= 1
512     return crc
513
514 def construct_modbus_request(slave, func, start, count):
515     frame = [slave, func, (start>>8)&0xFF, start&0xFF, (count>>8)&0xFF, count&0xFF]
516     crc = calculate_crc(frame)
517     frame.append(crc & 0xFF)
518     frame.append((crc >> 8) & 0xFF)
519     return bytes(frame)
520
521 def verify_crc(frame):
522     if len(frame) < 5:
523         return False
524     received = (frame[-1] << 8) | frame[-2]
525     calculated = calculate_crc(frame[:-2])
526     return received == calculated
527
528 def clear_uart_buffer_cvm():
529     while uart_cvm.any():
530         uart_cvm.read()
531     time.sleep_ms(5)
532
533 def send_modbus_request_cvm(frame):
534     clear_uart_buffer_cvm()
535     ctrl_pin_cvm.value(1)
536     time.sleep_ms(2)
537     uart_cvm.write(frame)
538     uart_cvm.flush()
539     time.sleep_ms(2)
540     ctrl_pin_cvm.value(0)
541
542 def read_modbus_response_cvm(timeout=1000):
543     start = time.ticks_ms()
544     response = bytearray()
545     while time.ticks_diff(time.ticks_ms(), start) < timeout:
546         if uart_cvm.any():
```

```
547         break
548     time.sleep_ms(5)
549     if not uart_cvm.any():
550         return None
551     last_time = time.ticks_ms()
552     while time.ticks_diff(time.ticks_ms(), last_time) < 100:
553         if uart_cvm.any():
554             response.extend(uart_cvm.read())
555             last_time = time.ticks_ms()
556         time.sleep_ms(5)
557     return response if len(response) >= 5 else None
558
559 def parse_registers_cvm(frame):
560     if len(frame) < 5 or frame[0] != SLAVE_ID_CVM or frame[1] & 0x80:
561         return None
562     byte_count = frame[2]
563     if len(frame) < 3 + byte_count + 2:
564         return None
565     data = frame[3:3+byte_count]
566     regs = []
567     for i in range(0, len(data), 2):
568         regs.append((data[i] << 8) | data[i+1])
569     return regs
570
571 def read_voltage():
572     try:
573         req = construct_modbus_request(SLAVE_ID_CVM, 0x04, 0x0000, 2)
574         send_modbus_request_cvm(req)
575         resp = read_modbus_response_cvm()
576         if resp and verify_crc(resp):
577             regs = parse_registers_cvm(resp)
578             if regs and len(regs) >= 2:
579                 return regs[1] / 10.0
580         return -99.0
581     except:
582         return -99.0
583
584 def read_current():
585     try:
586         req = construct_modbus_request(SLAVE_ID_CVM, 0x04, 0x0002, 2)
587         send_modbus_request_cvm(req)
588         resp = read_modbus_response_cvm()
589         if resp and verify_crc(resp):
590             regs = parse_registers_cvm(resp)
591             if regs and len(regs) >= 2:
592                 return regs[1] / 1000.0
```

```
593     return -99.0
594 except:
595     return -99.0
596
597 def read_power():
598     try:
599         req = construct_modbus_request(SLAVE_ID_CVM, 0x04, 0x0004, 2)
600         send_modbus_request_cvm(req)
601         resp = read_modbus_response_cvm()
602         if resp and verify_crc(resp):
603             regs = parse_registers_cvm(resp)
604             if regs and len(regs) >= 2:
605                 return regs[1]
606         return -99.0
607     except:
608         return -99.0
609
610 def clear_uart_buffer_inv():
611     while uart_inv.any():
612         uart_inv.read()
613     time.sleep_ms(5)
614
615 def send_modbus_request_inv(frame):
616     clear_uart_buffer_inv()
617     ctrl_pin_inv.value(1)
618     time.sleep_ms(2)
619     uart_inv.write(frame)
620     uart_inv.flush()
621     time.sleep_ms(2)
622     ctrl_pin_inv.value(0)
623
624 def read_modbus_response_inv(timeout=500):
625     start = time.ticks_ms()
626     resp = bytearray()
627     while time.ticks_diff(time.ticks_ms(), start) < timeout:
628         if uart_inv.any():
629             break
630         time.sleep_ms(5)
631     if not uart_inv.any():
632         return None
633     last_time = time.ticks_ms()
634     while time.ticks_diff(time.ticks_ms(), last_time) < 100:
635         if uart_inv.any():
636             resp.extend(uart_inv.read())
637             last_time = time.ticks_ms()
638         time.sleep_ms(5)
```

```
639     return resp if len(resp) >= 5 else None
640
641 def parse_registers_inv(frame):
642     if len(frame) < 5 or frame[0] != SLAVE_ID_INV or frame[1] & 0x80:
643         return None
644     byte_count = frame[2]
645     if len(frame) < 3 + byte_count + 2:
646         return None
647     data = frame[3:3+byte_count]
648     regs = []
649     for i in range(0, len(data), 2):
650         regs.append((data[i] << 8) | data[i+1])
651     return regs
652
653 def read_inversor_multiple(start_reg, count=2, retries=2):
654     for attempt in range(retries):
655         frame = construct_modbus_request(SLAVE_ID_INV, 0x04, start_reg, count)
656         send_modbus_request_inv(frame)
657         resp = read_modbus_response_inv(timeout=500)
658         if resp and verify_crc(resp):
659             regs = parse_registers_inv(resp)
660             if regs:
661                 return regs
662             if attempt < retries - 1:
663                 time.sleep_ms(50)
664     return None
665
666 def read_inversor_data():
667     regs = read_inversor_multiple(2, 2)
668     if regs and len(regs) >= 2:
669         return regs[0] / 10.0, regs[1] / 10.0
670     return -99.0, -99.0
671
672 # ===== FUNCIONES DE TEMPERATURA =====
673 def read_voltage_temperatura_mejorada(n=400):
674     lecturas = []
675     lecturas_validas = 0
676     intentos = 0
677     max_intentos = n * 2
678     while lecturas_validas < n and intentos < max_intentos:
679         raw = adc_temperatura.read_ul6()
680         voltage = (raw / 65535) * AREF
681         if VOLTAJE_MIN_ESPERADO <= voltage <= VOLTAJE_MAX_ESPERADO:
682             lecturas.append(voltage)
683             lecturas_validas += 1
684         intentos += 1
```

```
685     utime.sleep_us(200)
686     if len(lecturas) < 100:
687         return None
688     lecturas.sort()
689     inicio = len(lecturas) // 5
690     fin = len(lecturas) - inicio
691     lecturas_filt = lecturas[inicio:fin]
692     promedio = sum(lecturas_filt) / len(lecturas_filt)
693     varianza = sum((x - promedio)**2 for x in lecturas_filt) / len(lecturas_filt)
694     desv = math.sqrt(varianza)
695     if desv > 0.01:
696         estables = [x for x in lecturas_filt if abs(x - promedio) <= 2*desv]
697         if len(estables) > 50:
698             promedio = sum(estables) / len(estables)
699     return promedio
700
701 def promedio_movil_controlado(nuevo_valor, buffer, tamaño):
702     global CONTADOR_DRIFT
703     if len(buffer) == 0:
704         buffer.append(nuevo_valor)
705         return nuevo_valor
706     prom_actual = sum(buffer) / len(buffer)
707     dif = abs(nuevo_valor - prom_actual)
708     if dif > MAX_DESVIACION_PERMITIDA * 3:
709         CONTADOR_DRIFT += 1
710         if CONTADOR_DRIFT >= MAX_DRIFT_COUNT:
711             buffer_temp = buffer[-3:] if len(buffer) >= 3 else buffer
712             buffer.clear()
713             buffer.extend(buffer_temp)
714             CONTADOR_DRIFT = 0
715     else:
716         if CONTADOR_DRIFT > 0:
717             CONTADOR_DRIFT = max(0, CONTADOR_DRIFT - 1)
718     buffer.append(nuevo_valor)
719     if len(buffer) > tamaño:
720         buffer.pop(0)
721     return sum(buffer) / len(buffer)
722
723 def control_desviacion_acumulativa(buffer, lectura_count):
724     global VALOR_REFERENCIA
725     if len(buffer) < 5:
726         return False
727     promedio = sum(buffer) / len(buffer)
728     varianza = sum((x - promedio)**2 for x in buffer) / len(buffer)
729     desv = math.sqrt(varianza)
730     if lectura_count % LECTURAS_PARA_REFERENCIA == 0:
```

```
731     VALOR_REFERENCIA = promedio
732     if VALOR_REFERENCIA is not None:
733         drift = abs(promedio - VALOR_REFERENCIA)
734         if drift > 0.05:
735             return True
736     if desv > MAX_DESVIACION_PERMITIDA:
737         return True
738     return False
739
740 def resetear_sistema_medicion():
741     global buffer_temperatura, CONTADOR_DRIFT, VALOR_REFERENCIA
742     buffer_temperatura.clear()
743     CONTADOR_DRIFT = 0
744     VALOR_REFERENCIA = None
745     for i in range(20):
746         adc_temperatura.read_u16()
747         utime.sleep_ms(10)
748
749 def calibrar_adc():
750     print("Calibrando ADC...")
751     for i in range(50):
752         adc_temperatura.read_u16()
753         utime.sleep_ms(10)
754     print("Calibracion completada.")
755
756 def calcular_temperatura(voltage, m, b, offset):
757     return ((voltage - b) / m) + offset
758
759 def inicializar_csv():
760     with open("datos_sensores.csv", "w") as f:
761         f.write("\u00eff")
762
763     f.write("Timestamp\tVoltaje_Temp_V\tTemperatura_Horno_C\tTemperatura_Ambiente_C\tDesviacion_V\tModb")
764
765 def contar_lineas_csv():
766     try:
767         with open("datos_sensores.csv", "r") as f:
768             return sum(1 for _ in f) - 1
769     except:
770         return 0
771
772 def rotar_csv_si_necesario():
773     try:
774         lineas = contar_lineas_csv()
775         if lineas >= MAX_LINEAS_CSV:
```

```

776         os.rename("datos_sensores.csv", "datos_sensores_backup.csv")
777         print("\n*** CSV ROTADO: {} lineas guardadas en backup ***".format(lineas))
778     except:
779         pass
780     inicializar_csv()
781     return True
782     return False
783 except:
784     return False
785
786 def guardar_en_csv_linea(ts, vt, tc, ta, desv, mb_v, mb_c, mb_p, inv_v, inv_p, nivel):
787     try:
788         with open("datos_sensores.csv", "a") as f:
789             linea =
790             "{}\t{:.4f}\t{:.2f}\t{:.2f}\t{:.6f}\t{:.2f}\t{:.3f}\t{:.1f}\t{:.1f}\t{:.1f}\t{}\n".format(
791                 ts, vt, tc, ta, desv, mb_v, mb_c, mb_p, inv_v, inv_p, nivel)
792             linea = linea.replace('.', ',')
793             f.write(linea)
794     except:
795         pass
796
797 # ===== PROGRAMA PRINCIPAL =====
798 print("=== CONTROL DE HORNO - SISTEMA MEJORADO CON SUAVIZADO ===")
799 configurar_hora_inicial()
800
801 print("\n** CONFIGURACION **")
802 print("Mediciones: cada {} seg | Recalculo avisos: cada {} min".format(
803     INTERVALO_MEDICION, INTERVALO_RECALCULO//60))
804 print("Rango de temperatura: +/- {}C".format(RANGO_TEMPERATURA))
805 print("Delta de seguridad: {} seg ({} min)".format(DELTA_TIEMPO_SEGURIDAD,
806     DELTA_TIEMPO_SEGURIDAD//60))
807 print("Suavizado: Factor={:.1%} | Max cambio permitido={{}s ({}min)".format(
808     FACTOR_SUAVIZADO, MAX_CAMBIO_PERMITIDO, MAX_CAMBIO_PERMITIDO//60))
809 print("CVM: UART{} ID:{} | INV: UART{} ID:{}".format(UART_ID_CVM, SLAVE_ID_CVM, UART_ID_INV,
810     SLAVE_ID_INV))
811 print("\n** HORARIOS **")
812 for h in HORARIOS_COCCION:
813     print(" {} -> {}C (rango: {}-{}C) - {}".format(
814         h['hora'], h['temp_objetivo'],
815         h['temp_objetivo'] - RANGO_TEMPERATURA,
816         h['temp_objetivo'] + RANGO_TEMPERATURA,
817         h['descripcion']))
818 print("-" * 80)
819
820 m = 0.0055
821 b = 1.4042

```

```
819 offset = -23.2
820
821 calibrar_adc()
822 inicializar_csv()
823 print("\nPresiona Ctrl+C para detener.")
824 print("-" * 80)
825
826 lecturas_count = 0
827 proximo_ciclo = utime.ticks_ms()
828 ultimo_recalculo_ms = utime.ticks_ms()
829 aviso_actual = None
830
831 try:
832     while True:
833         ahora = utime.ticks_ms()
834         if utime.ticks_diff(ahora, proximo_ciclo) >= 0:
835             datos_validos = True
836             vt_raw = read_voltage_temperatura_mejorada(400)
837             if vt_raw is None:
838                 datos_validos = False
839             else:
840                 vt = promedio_movil_controlado(vt_raw, buffer_temperatura, BUFFER_SIZE)
841                 if control_desviacion_acumulativa(buffer_temperatura, lecturas_count + 1):
842                     resetear_sistema_medicion()
843                     vt_raw = read_voltage_temperatura_mejorada(300)
844                     if vt_raw is not None:
845                         vt = promedio_movil_controlado(vt_raw, buffer_temperatura,
846                                                         BUFFER_SIZE)
847                     else:
848                         datos_validos = False
849                     if datos_validos:
850                         t_horno = calcular_temperatura(vt, m, b, offset)
851
852                 try:
853                     sensor_dht.measure()
854                     t_amb = sensor_dht.temperature() - 2.0
855                 except:
856                     t_amb = -99.0
857
858             mb_voltaje = read_voltage()
859             time.sleep_ms(30)
860             mb_corriente = read_current()
861             time.sleep_ms(30)
862             mb_potencia = read_power()
863             time.sleep_ms(30)
864             inv_potencia, inv_voltaje = read_inversor_data()
```



```
906 print("Datos en: datos_sensores.csv")
907 print("="*80)
```

8.5. Código Inercia Termica

En esta sección se presenta el código Python desarrollado para el análisis de la inercia térmica del horno mediante el modelado de las curvas de calentamiento y enfriamiento. El programa procesa los datos experimentales de temperatura en función del tiempo para los seis niveles de operación del horno, ajustando polinomios de cuarto grado que describen el comportamiento térmico en ambas fases

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import os
4
5 print("="*80)
6 print("GENERADOR DE ECUACIONES TÉRMICAS DEL HORNO")
7 print("="*80)
8
9 # ===== DATOS DE ENFRIAMIENTO =====
10 # Tiempo en segundos (cada 5 min = 300s)
11 tiempo_enfriamiento = np.array([0, 300, 600, 900, 1200, 1500, 1800,
12     2100, 2400, 2700, 3000, 3300, 3600])
13
14 # Temperaturas de ENFRIAMIENTO para cada nivel (desde temperatura máxima)
15 # Los niveles están organizados por temperatura de configuración
16 datos_enfriamiento = {
17     95: [102.38, 82.83, 60.12, 46.34, 37.18, 33.09, 29.07, 27.2, 24.98,
18         24.56, 23.88, 22.9, 22.37],
19     120: [121.66, 99.57, 70.1, 52.15, 40.89, 34.49, 31.71, 29.01, 27.17,
20         25.88, 25.02, 24.9, 23.34],
21     150: [130.14, 97.9, 68.93, 51.94, 41.37, 36.1, 31.6, 28.4, 26.6,
22         25.06, 25.36, 24.66, 23.3],
23     180: [162.69, 114.9, 76.87, 56.31, 43.65, 35.36, 32.19, 27.72,
24         25.86, 23.67, 23.04, 22.46, 22.37],
25     200: [184.22, 129.89, 84.67, 59.83, 45.89, 37.3, 33.42, 30.24,
26         26.95, 25.44, 25.37, 24.96, 22.29],
27     230: [205.53, 138.18, 88.89, 65.09, 49.97, 42.06, 36.78, 31.99,
28         29.65, 28.63, 25.93, 24.45, 22.93]
29 }
```

```

24 # ===== DATOS DE CALENTAMIENTO =====
25 # Tiempo en segundos (cada 5 min = 300s)
26 tiempo_calentamiento = np.array([0, 300, 600, 900, 1200, 1500, 1800,
    2100, 2400, 2700, 3000, 3300, 3600])
27
28 # Temperaturas de CALENTAMIENTO para cada nivel (desde temperatura
    ambiente ~20°C)
29 datos_calentamiento = {
30     95: [16.19, 129.15, 122.98, 111.92, 106.39, 103.79, 104.02, 105.21,
    105.02, 105.66, 104.32, 102.74, 104.18],
31     120: [17.68, 145.19, 139.59, 126.63, 120.05, 117.01, 117.82, 118.73,
    118.52, 117.19, 117.5, 116.16, 115.37],
32     150: [17.22, 171.79, 162.27, 149.88, 147.23, 144.8, 141.88, 142.57,
    143.54, 141.96, 140.04, 141.04, 140.8],
33     180: [24.49, 194.58, 184.38, 173.73, 166.14, 165.38, 163.63, 165.01,
    162.46, 162.51, 162.55, 162.46, 160.89],
34     200: [16.3, 241.69, 208.57, 193.54, 186.76, 184.08, 184.1, 183.98,
    184.13, 185.05, 184.18, 181.83, 177.79],
35     230: [22.58, 232.79, 220.77, 210.08, 208.4, 207.17, 207.37, 204.51,
    205.02, 204.3, 205.2, 204.73, 202.42]
36 }
37
38 # Mapeo de niveles a temperaturas máximas reales
39 niveles_temp = {
40     95: {'nivel': 1, 'temp_max': 104.4},
41     120: {'nivel': 2, 'temp_max': 117.6},
42     150: {'nivel': 3, 'temp_max': 142.1},
43     180: {'nivel': 4, 'temp_max': 163.4},
44     200: {'nivel': 5, 'temp_max': 183.1},
45     230: {'nivel': 6, 'temp_max': 205.3}
46 }
47
48 print("\n" + "="*80)
49 print("ANÁLISIS DE DATOS")
50 print("="*80)
51 print(f"Tiempo total: {tiempo_enfriamiento[-1]/60:.0f} minutos
    ({tiempo_enfriamiento[-1]} segundos)")
52 print(f"Intervalo de muestreo:
    {tiempo_enfriamiento[1]-tiempo_enfriamiento[0]} segundos (5 minutos)")
53 print(f"Número de muestras: {len(tiempo_enfriamiento)}")
54 print("\nNiveles de temperatura:")
55 for config, info in niveles_temp.items():
56     print(f" Nivel {info['nivel']}: {config}°C      Temp máxima real:
    {info['temp_max']}°C")
57
58 # ===== GENERACIÓN DE ECUACIONES DE ENFRIAMIENTO =====

```

```

59 print("\n" + "="*80)
60 print("ECUACIONES DE ENFRIAMIENTO (desde temperatura máxima hasta
    ambiente)")
61 print("="*80)
62 print("Formato: T(t) = a + a *t + a *t + a *t + a *t ")
63 print("Donde t está en SEGUNDOS y T en °CELSIUS")
64 print("-"*80)
65
66 ecuaciones_enfriamiento = {}
67
68 for config, temperaturas in datos_enfriamiento.items():
69     # Ajustar polinomio de grado 4
70     coef = np.polyfit(tiempo_enfriamiento, temperaturas, 4)
71     # Invertir coeficientes para formato [ a , a , a , a , a ]
72     coef = coef[::-1]
73     ecuaciones_enfriamiento[config] = coef
74
75     # Calcular R para validar ajuste
76     temp_predichas = np.polyval(coef[::-1], tiempo_enfriamiento)
77     ss_res = np.sum((temperaturas - temp_predichas)**2)
78     ss_tot = np.sum((temperaturas - np.mean(temperaturas))**2)
79     r_cuadrado = 1 - (ss_res / ss_tot)
80
81     nivel_num = niveles_temp[config]['nivel']
82     temp_max = niveles_temp[config]['temp_max']
83
84     print(f"\nNIVEL {nivel_num} (Config: {config}°C, Temp máxima real:
    {temp_max}°C)")
85     print(f" Coeficientes:")
86     print(f"     a (constante) = {coef[0]:.6e}")
87     print(f"     a (lineal) = {coef[1]:.6e}")
88     print(f"     a (cuadrático) = {coef[2]:.6e}")
89     print(f"     a (cúbico) = {coef[3]:.6e}")
90     print(f"     a (cuártico) = {coef[4]:.6e}")
91     print(f" R = {r_cuadrado:.6f} {' Excelente ajuste' if
    r_cuadrado > 0.95 else ' Ajuste mejorable'}")
92
93     # Verificar temperatura a 2500 segundos
94     t_test = 2500
95     temp_2500 = coef[0] + coef[1]*t_test + coef[2]*(t_test**2) +
    coef[3]*(t_test**3) + coef[4]*(t_test**4)
96     print(f" T(2500s) = {temp_2500:.2f}°C")
97
98 # ===== GENERACIÓN DE ECUACIONES DE CALENTAMIENTO =====
99 print("\n" + "="*80)

```

```

100 print("ECUACIONES DE CALENTAMIENTO (desde temperatura ambiente hasta
      máxima)")
101 print("="*80)
102 print("Formato: T(t) = a + a *t + a *t + a *t + a *t ")
103 print("Donde t está en SEGUNDOS y T en °CELSIUS")
104 print("-"*80)
105
106 ecuaciones_calentamiento = {}
107
108 for config, temperaturas in datos_calentamiento.items():
109     # Ajustar polinomio de grado 4
110     coef = np.polyfit(tiempo_calentamiento, temperaturas, 4)
111     # Invertir coeficientes para formato [ a , a , a , a , a ]
112     coef = coef[::-1]
113     ecuaciones_calentamiento[config] = coef
114
115     # Calcular R para validar ajuste
116     temp_predichas = np.polyval(coef[::-1], tiempo_calentamiento)
117     ss_res = np.sum((temperaturas - temp_predichas)**2)
118     ss_tot = np.sum((temperaturas - np.mean(temperaturas))**2)
119     r_cuadrado = 1 - (ss_res / ss_tot)
120
121     nivel_num = niveles_temp[config]['nivel']
122     temp_max = niveles_temp[config]['temp_max']
123
124     print(f"\nNIVEL {nivel_num} (Config: {config}°C, Temp máxima real:
125           {temp_max}°C)")
126     print(f"  Temp inicial: {temperaturas[0]:.2f}°C      Temp máxima
127           alcanzada: {max(temperaturas):.2f}°C")
128     print(f"  Coeficientes:")
129     print(f"    a (constante) = {coef[0]:.6e}")
130     print(f"    a (lineal) = {coef[1]:.6e}")
131     print(f"    a (cuadrático) = {coef[2]:.6e}")
132     print(f"    a (cúbico) = {coef[3]:.6e}")
133     print(f"    a (cuártico) = {coef[4]:.6e}")
134     print(f"  R = {r_cuadrado:.6f} {' Excelente ajuste' if
135           r_cuadrado > 0.95 else ' Ajuste mejorable'}")
136
137 # ===== GENERACIÓN DE GRÁFICAS =====
138 print("\n" + "="*80)
139 print("GENERANDO GRÁFICAS...")
140 print("="*80)
141
142 # Configurar matplotlib para español
143 plt.rcParams['font.sans-serif'] = ['Arial', 'DejaVu Sans']
144 plt.rcParams['axes.unicode_minus'] = False

```

```

142
143 # Crear tiempo continuo para las curvas suaves
144 tiempo_continuo = np.linspace(0, 3600, 500)
145
146 # Colores para cada nivel
147 colores = {
148     95: '#1f77b4', # Azul
149     120: '#ff7f0e', # Naranja
150     150: '#2ca02c', # Verde
151     180: '#d62728', # Rojo
152     200: '#9467bd', # Púrpura
153     230: '#8c564b' # Marrón
154 }
155
156 # ===== GRÁFICA 1: CALENTAMIENTO =====
157 fig, ax = plt.subplots(figsize=(14, 8))
158
159 for config in sorted(datos_calentamiento.keys()):
160     temperaturas = datos_calentamiento[config]
161     coef = ecuaciones_calentamiento[config]
162     nivel_num = niveles_temp[config]['nivel']
163     temp_max = niveles_temp[config]['temp_max']
164
165     # Curva ajustada - SOLO CURVA, SIN PUNTOS
166     temp_continua = (coef[0] + coef[1]*tiempo_continuo +
167                     coef[2]*(tiempo_continuo**2) +
168                     coef[3]*(tiempo_continuo**3) +
169                     coef[4]*(tiempo_continuo**4))
170     ax.plot(tiempo_continuo/60, temp_continua,
171             color=colores[config], linewidth=3,
172             label=f'Nivel {nivel_num} ({temp_max}°C)', alpha=0.9)
173
174 ax.set_xlabel('Tiempo (minutos)', fontsize=14, fontweight='bold')
175 ax.set_ylabel('Temperatura (°C)', fontsize=14, fontweight='bold')
176 ax.set_title('CURVAS DE CALENTAMIENTO DEL HORNO\nAjuste Polinómico de
177             Grado 4',
178             fontsize=16, fontweight='bold', pad=20)
179 ax.grid(True, alpha=0.3, linestyle='--', linewidth=0.8)
180 ax.legend(loc='best', fontsize=11, framealpha=0.95)
181 plt.tight_layout()
182 plt.savefig('grafica_calentamiento.png', dpi=300, bbox_inches='tight')
183 print(" Gráfica de CALENTAMIENTO guardada: grafica_calentamiento.png")
184 print(f" Ubicación: {os.path.abspath('grafica_calentamiento.png')}")
185 plt.show()
186
187 # ===== GRÁFICA 2: ENFRIAMIENTO =====

```

```

187 fig, ax = plt.subplots(figsize=(14, 8))
188
189 for config in sorted(datos_enfriamiento.keys()):
190     temperaturas = datos_enfriamiento[config]
191     coef = ecuaciones_enfriamiento[config]
192     nivel_num = niveles_temp[config]['nivel']
193     temp_max = niveles_temp[config]['temp_max']
194
195     # Curva ajustada - SOLO CURVA, SIN PUNTOS
196     temp_continua = (coef[0] + coef[1]*tiempo_continuo +
197                    coef[2]*(tiempo_continuo**2) +
198                    coef[3]*(tiempo_continuo**3) +
199                    coef[4]*(tiempo_continuo**4))
200     ax.plot(tiempo_continuo/60, temp_continua,
201            color=colores[config], linewidth=3,
202            label=f'Nivel {nivel_num} ({temp_max}°C)', alpha=0.9)
203
204 ax.set_xlabel('Tiempo (minutos)', fontsize=14, fontweight='bold')
205 ax.set_ylabel('Temperatura (°C)', fontsize=14, fontweight='bold')
206 ax.set_title('CURVAS DE ENFRIAMIENTO DEL HORNO\nAjuste Polinómico de
207             Grado 4',
208             fontsize=16, fontweight='bold', pad=20)
209 ax.grid(True, alpha=0.3, linestyle='--', linewidth=0.8)
210 ax.legend(loc='best', fontsize=11, framealpha=0.95)
211 plt.tight_layout()
212 plt.savefig('grafica_enfriamiento.png', dpi=300, bbox_inches='tight')
213 print("    Gráfica de ENFRIAMIENTO guardada: grafica_enfriamiento.png")
214 print(f"    Ubicación: {os.path.abspath('grafica_enfriamiento.png')}")
215 plt.show()
216
217 # ===== CÓDIGO PARA MICROPYTHON =====
218 print("\n" + "="*80)
219 print("CÓDIGO PARA COPIAR EN MICROPYTHON")
220 print("="*80)
221
222 print("\n# ECUACIONES DE ENFRIAMIENTO (por configuración de
223     temperatura)")
224 print("ECUACIONES_ENFRIAMIENTO = {")
225 for config, coef in ecuaciones_enfriamiento.items():
226     print(f"    {config}: [{coef[0]:.6e}, {coef[1]:.6e}, {coef[2]:.6e},
227           {coef[3]:.6e}, {coef[4]:.6e}],")
228 print("}")
229
230 print("\n# ECUACIONES DE CALENTAMIENTO (por configuración de
231     temperatura)")
232 print("ECUACIONES_CALENTAMIENTO = {")

```

```

229 for config, coef in ecuaciones_calentamiento.items():
230     print(f"    {config}: [{coef[0]:.6e}, {coef[1]:.6e}, {coef[2]:.6e},
231           {coef[3]:.6e}, {coef[4]:.6e}],")
232
233 # ===== FUNCIÓN DE PREDICCIÓN =====
234 print("\n# FUNCIONES DE PREDICCIÓN")
235 print("""
236 def calcular_temperatura_enfriamiento(config_temp, tiempo_segundos):
237     '''
238     Predice temperatura durante ENFRIAMIENTO
239     config_temp: temperatura configurada (95, 120, 150, 180, 200, 230)
240     tiempo_segundos: tiempo transcurrido en segundos
241     '''
242     coef = ECUACIONES_ENFRIAMIENTO[config_temp]
243     t = tiempo_segundos
244     return coef[0] + coef[1]*t + coef[2]*(t**2) + coef[3]*(t**3) +
245           coef[4]*(t**4)
246
247 def calcular_temperatura_calentamiento(config_temp, tiempo_segundos):
248     '''
249     Predice temperatura durante CALENTAMIENTO
250     config_temp: temperatura configurada (95, 120, 150, 180, 200, 230)
251     tiempo_segundos: tiempo transcurrido en segundos
252     '''
253     coef = ECUACIONES_CALENTAMIENTO[config_temp]
254     t = tiempo_segundos
255     return coef[0] + coef[1]*t + coef[2]*(t**2) + coef[3]*(t**3) +
256           coef[4]*(t**4)
257
258 # Ejemplo de uso:
259 # temp = calcular_temperatura_calentamiento(120, 1500) # Config 120°C a
260 #           25 minutos
261 # print(f"Temperatura estimada: {temp:.2f}°C")
262 # print("""
263 # ===== VALIDACIÓN CON DATOS REALES =====
264 print("\n" + "="*80)
265 print("VALIDACIÓN A 2500 SEGUNDOS (41.67 minutos)")
266 print("="*80)
267
268 print("\nENFRIAMIENTO:")
269 for config in sorted(ecuaciones_enfriamiento.keys()):
270     coef = ecuaciones_enfriamiento[config]
271     nivel_num = niveles_temp[config]['nivel']
272     temp_max = niveles_temp[config]['temp_max']

```

```

271     t = 2500
272     temp = coef[0] + coef[1]*t + coef[2]*(t**2) + coef[3]*(t**3) +
coef[4]*(t**4)
273     print(f" Nivel {nivel_num} ({{temp_max}}°C)      T(2500s) =
{{temp:6.2f}}°C")
274
275 print("\nCALENTAMIENTO:")
276 for config in sorted(ecuaciones_calentamiento.keys()):
277     coef = ecuaciones_calentamiento[config]
278     nivel_num = niveles_temp[config]['nivel']
279     temp_max = niveles_temp[config]['temp_max']
280     t = 2500
281     temp = coef[0] + coef[1]*t + coef[2]*(t**2) + coef[3]*(t**3) +
coef[4]*(t**4)
282     print(f" Nivel {nivel_num} ({{temp_max}}°C)      T(2500s) =
{{temp:6.2f}}°C")
283
284 print("\n" + "="*80)
285 print("ANÁLISIS COMPLETADO")
286 print("="*80)
287 print("Archivos generados en el directorio actual:")
288 print(" - grafica_calentamiento.png")
289 print(" - grafica_enfriamiento.png")
290 print("\nLas leyendas muestran: Nivel X (Temperatura máxima real)")
291 print("Copia las ecuaciones generadas al código MicroPython")
292 print("="*80)

```

8.6. Resultados Microrred

En esta sección se presentan los resultados de temperatura, donde el tiempo se expresa en minutos con intervalos de 30 minutos. Se registra la temperatura alcanzada en cada intervalo y se compara con la temperatura objetivo establecida para el sistema.

8.6.1. Resultados Caso 1.

TABLA 25: Resultados experimentales Caso 1.

Tiempo (min)	Temperatura_Horno_°C
0	23.18
30	24.24
60	206.44
90	192.23
120	194.11
150	194.29
180	192.97
210	193.32
240	192.36
270	192.63
300	193.73
330	49.51
360	28.62
390	33.63
420	64.74
450	95.62
480	108.92
510	192.26
540	202.18

8.6.2. Resultados Caso 2.

TABLA 26: Resultados experimentales Caso 2.

Tiempo (min)	Temperatura_Horno_°C
0	18.37
30	18.59
60	127.74
90	108.7
120	108.7
150	108.54
180	137.12
210	140.17
240	138.66
270	164.96
300	163.24
330	166.86
360	165.86
390	181.94
420	179.87
450	181.91
480	183.32
510	198
540	197.81
570	197.56
600	55.64
630	76.77
660	76.44
690	27.62
720	44.53

8.6.3. Resultados Caso 3.

TABLA 27: Resultados experimentales Caso 3.

Tiempo (min)	Temperatura_Horno_°C
0	17.57
30	18.95
60	19.85
90	18.46
120	19.34
150	20.92
180	22
210	23.32
240	182.88
270	171.01
300	168.74
330	170.72
360	171.55
390	177.9
420	194.66
450	164.11
480	193.05
510	192.52
540	135.18
570	82.05
600	61.72

8.7. Datos experimentales de la potencia de red

En este anexo se presentan los valores de potencia promedio obtenidos para cada uno de los tres casos evaluados en la microrred. Se incluyen los datos de potencia suministrada por el sistema fotovoltaico (PV usada), la potencia que debió ser tomada de la red eléctrica y la potencia exportada cuando se generaron excedentes. Esta información permite observar el comportamiento del sistema en diferentes condiciones de operación y comparar la contribución real de cada fuente de potencia.

8.7.1. Caso 1

TABLA 28: Potencias promedio para el caso 1. Tomado de: Autores.

Intervalo (h)	PV Usada (W)	Red Aporta (W)	Exporta (W)
11–12	116.77	0	93.68
12–13	131.11	164.86	0
13–14	180.36	115.59	0
14–15	211.59	79.99	0
15–16	36.61	257.62	0
16–17	23.19	30.72	0
17–18	10.67	34.11	0
18–19	0	213.45	0
19–20	0	221.10	0
Total	710.30	1117.43	93.68

8.7.2. Caso 2

TABLA 29: Potencias promedio para el caso 2. Tomado de: Autores.

Intervalo (h)	PV Usada (W)	Red Aporta (W)	Exporta (W)
9-10	0	0.00	64.93
10-11	104	0.42	0
11-12	136.90	2.23	0
12-13	176.10	0.00	44.91
13-14	204.97	40.59	0
14-15	163.70	78.97	0
15-16	115.66	159.77	0
16-17	49.34	230.74	0
17-18	13.47	298.97	0
18-19	0	262.03	0
19-20	0	65.12	0
20-21	0	66.98	0
Total	964.13	1205.80	109.84

8.7.3. Caso 3

TABLA 30: Potencias promedio para el caso 3. Tomado de: Autores.

Intervalo (h)	PV Usada (W)	Red Aporta (W)	Exporta (W)
9-10	0	0.00	232.61
10-11	0	0.00	314.08
11-12	0	0.00	483.55
12-13	89.01	0.00	385.83
13-14	239.14	14.89	0
14-15	219.70	121.24	0
15-16	53.63	307.51	0
16-17	20.91	369.53	0
17-18	46.26	265.25	0
18-19	0	146.26	0
Total	668.67	1224.68	1416.07

Bibliografía

- [1] Tanushree Gupta et al. «A Robust Approach for Analysis and Visualization of CO2 and Greenhouse Gas Emission and Its Effect». En: *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*. Mar. de 2023, págs. 238-243. URL: <https://ieeexplore.ieee.org/document/10112389/?arnumber=10112389> (visitado 20-08-2024).
- [2] Mohammad Mafizur Rahman, Nahid Sultana y Eswaran Velayutham. «Renewable energy, energy intensity and carbon reduction: Experience of large emerging economies». en. En: *Renewable Energy* 184 (ene. de 2022), págs. 252-265. ISSN: 09601481. DOI: 10.1016/j.renene.2021.11.068. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0960148121016517> (visitado 22-08-2024).
- [3] A. K. Onalapo et al. «Assessment of Sustainable Development: a Nexus between Reliability, Greenhouse Gas Emissions and Renewable Energy Technologies». En: *2022 30th Southern African Universities Power Engineering Conference (SAUPEC)*. Ene. de 2022, págs. 1-5. DOI: 10.1109/SAUPEC55179.2022.9730748. URL: <https://ieeexplore.ieee.org/document/9730748/?arnumber=9730748> (visitado 22-08-2024).
- [4] *Hoja de Ruta del Hidrógeno*. es. URL: <https://www.miteco.gob.es/es/ministerio/planes-estrategias/hidrogeno.html> (visitado 23-08-2024).
- [5] IRENA. «Green hydrogen strategy: A guide to design». en. En: *International Renewable Energy Agency, Abu Dhabi (2024)*, pág. 127. URL: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2024/Jul/IRENA_Green_hydrogen_strategy_design_2024.pdf.
- [6] Gabriela Corredor Liberato y Juan Sebastián Osorio Sánchez. «IMPACTO DE LAS TECNOLOGÍAS POWER-TO-X EN LA OPERACIÓN DE SISTEMAS ELÉCTRICOS CON ALTA PENETRACIÓN DE FUENTES DE ENERGÍA RENOVABLE». es. En: (2023), pág. 151. URL: <https://repository.usta.edu.co/handle/11634/55083>.

-
- [7] Tingting Xu et al. «The implementation limitation of variable renewable energies and its impacts on the public power grid». en. En: *Energy* 239 (ene. de 2022), pág. 121992. ISSN: 03605442. DOI: 10.1016/j.energy.2021.121992. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0360544221022404> (visitado 22-08-2024).
- [8] República de Colombia Ministerio de Minas y Energía. «Actualización plan energético nacional (PEN) 2022 - 2052». es. En: (jun. de 2023), pág. 159. URL: https://www1.upme.gov.co/DemandayEficiencia/Documents/PEN_2020_2050/Actualizacion_PEN_2022-2052_VF.pdf.
- [9] Andreas Bloess, Wolf-Peter Schill y Alexander Zerrahn. «Power-to-heat for renewable energy integration: A review of technologies, modeling approaches, and flexibility potentials». en. En: *Applied Energy* 212 (feb. de 2018), págs. 1611-1626. ISSN: 03062619. DOI: 10.1016/j.apenergy.2017.12.073. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0306261917317889> (visitado 29-08-2024).
- [10] Naciones Unidas. «La Agenda 2030 y los objetivos de desarrollo sostenible una oportunidad para América Latina y Caribe». es. En: (2018). OCLC: 1337334530. URL: <https://repositorio.cepal.org/server/api/core/bitstreams/cb30a4de-7d87-4e79-8e7a-ad5279038718/content>.
- [11] Asif Raihan. «The influences of renewable energy, globalization, technological innovations, and forests on emission reduction in Colombia». en. En: *Innovation and Green Development* 2.4 (dic. de 2023), pág. 100071. ISSN: 29497531. DOI: 10.1016/j.igd.2023.100071. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2949753123000395> (visitado 30-08-2024).
- [12] Fengsheng Chien, Ka Yin Chau y Muhammad Sadiq. «The effect of energy transition technologies on greenhouse gas emissions: New evidence from ASEAN countries». en. En: *Sustainable Energy Technologies and Assessments* 58 (ago. de 2023), pág. 103354. ISSN: 22131388. DOI: 10.1016/j.seta.2023.103354. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2213138823003478> (visitado 30-08-2024).
- [13] Yi Ge et al. «A Review of the Impact of Long-term Climate Change on Renewable Energy». En: *2022 IEEE 6th Conference on Energy Internet and Energy System Integration (EI2)*. Nov. de 2022, págs. 3228-3233. DOI: 10.1109/EI256261.2022.10117070. URL: <https://ieeexplore.ieee.org/document/10117070/?arnumber=10117070> (visitado 06-09-2024).
- [14] Michael Sterner y Michael Specht. «Power-to-Gas and Power-to-X—The History and Results of Developing a New Storage Concept». En: *Energies* 14 (oct. de 2021), pág. 6594. DOI: 10.3390/en14206594. URL: <https://www.researchgate.net/publication/>

- 355226549_Power-to-Gas_and_Power-to-X-The_History_and_Results_of_Developing_a_New_Storage_Concept.
- [15] Sha Luo et al. «Power System Flexibility Assessment Method for Matching Supply and Demand with Flexibility». En: *2023 8th International Conference on Power and Renewable Energy (ICPRE)*. ISSN: 2768-0525. Sep. de 2023, págs. 107-112. DOI: 10.1109/ICPRE59655.2023.10353889. URL: <https://ieeexplore.ieee.org/document/10353889/?arnumber=10353889> (visitado 06-09-2024).
- [16] Mufid Altorok et al. «Review on optimal power flow in a hybrid AC/DC distribution network». en. En: *Electric Power Systems Research* 236 (nov. de 2024), pág. 110956. ISSN: 03787796. DOI: 10.1016/j.epsr.2024.110956. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0378779624008423> (visitado 06-09-2024).
- [17] Zhengguang Liu et al. «Power to heat: Opportunity of flexibility services provided by building energy systems». en. En: *Advances in Applied Energy* 11 (sep. de 2023), pág. 100149. ISSN: 26667924. DOI: 10.1016/j.adapen.2023.100149. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2666792423000288> (visitado 06-09-2024).
- [18] Silvia Madeddu et al. «The CO₂ reduction potential for the European industry via direct electrification of heat supply (power-to-heat)». en. En: *Environmental Research Letters* 15.12 (dic. de 2020), pág. 124004. ISSN: 1748-9326. DOI: 10.1088/1748-9326/abbd02. URL: <https://iopscience.iop.org/article/10.1088/1748-9326/abbd02> (visitado 06-09-2024).
- [19] Daniel A. Perez-Moscote y Mikhail G. Tyagunov. «Modeling of a Distributed Energy System with Renewable Generation, Demand-side Flexibility, and Behind-the-meter Batteries». En: *2022 4th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE)*. Mar. de 2022, págs. 1-5. DOI: 10.1109/REEPE53907.2022.9731359. URL: <https://ieeexplore.ieee.org/document/9731359/?arnumber=9731359> (visitado 16-09-2024).
- [20] Peter A.V. Gade et al. «Load shifting versus manual frequency reserve: Which one is more appealing to thermostatically controlled loads in Denmark?» en. En: *Electric Power Systems Research* 232 (jul. de 2024), pág. 110364. ISSN: 03787796. DOI: 10.1016/j.epsr.2024.110364. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0378779624002529> (visitado 06-09-2024).
- [21] Mobolaji Bello et al. «Power system studies considerations for Microgrid design». en. En: *2019 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*. Manizales, Colombia: IEEE, mayo de 2019, págs. 1-5. ISBN: 978-1-72811-626-6. DOI: 10.1109/PEPQA.2019.8851559. URL: <https://ieeexplore.ieee.org/document/8851559/> (visitado 29-08-2024).

-
- [22] Arindam Maitra et al. «Microgrid Controllers : Expanding Their Role and Evaluating Their Performance». En: *IEEE Power and Energy Magazine* 15.4 (jul. de 2017). Conference Name: IEEE Power and Energy Magazine, págs. 41-49. ISSN: 1558-4216. DOI: 10.1109/MPE.2017.2690519. URL: <https://ieeexplore.ieee.org/document/7947303> (visitado 17-09-2024).
- [23] Komeil Nosrati et al. «Intelligent frequency control of AC microgrids with communication delay: An online tuning method subject to stabilizing parameters». en. En: *Energy and AI* 18 (dic. de 2024), pág. 100421. ISSN: 26665468. DOI: 10.1016/j.egyai.2024.100421. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2666546824000879> (visitado 15-10-2024).
- [24] Walter Gil González et al. «Economic Dispatch of Renewable Generators and BESS in DC Microgrids Using Second-Order Cone Optimization». En: *Energies* 13 (abr. de 2020), págs. 1-15. DOI: 10.3390/en13071703.
- [25] Mohamed Awadalla y François Bouffard. «Flexibility Characterization of Sustainable Power Systems in Demand Space: A Data-Driven Inverse Optimization Approach». en. En: *IEEE Transactions on Power Systems* 39.5 (sep. de 2024). arXiv:2212.05963 [math], págs. 6196-6209. ISSN: 0885-8950, 1558-0679. DOI: 10.1109/TPWRS.2024.3364502. URL: <http://arxiv.org/abs/2212.05963> (visitado 13-02-2025).
- [26] María Victoria Gasca, Federico Ibáñez y David Pozo. «Flexibility quantification of thermostatically controlled loads for demand response applications». en. En: *Electric Power Systems Research* 202 (ene. de 2022), pág. 107592. ISSN: 03787796. DOI: 10.1016/j.epsr.2021.107592. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0378779621005733> (visitado 13-02-2025).
- [27] Ministerio de Energía de Chile y ACERA. *Estudio de Flexibilidad del Sistema Eléctrico*. Inf. téc. Informe técnico oficial. Gobierno de Chile, Ministerio de Energía, 2019. URL: https://energia.gob.cl/sites/default/files/2019_acera_estudio_flexibilidad.pdf.
- [28] Mohammad Fakhroleslam y Seyed Mojtaba Sadrameli. «Thermal/catalytic cracking of hydrocarbons for the production of olefins; a state-of-the-art review III: Process modeling and simulation». en. En: *Fuel* 252 (sep. de 2019), págs. 553-566. ISSN: 00162361. DOI: 10.1016/j.fuel.2019.04.127. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0016236119306921> (visitado 09-10-2025).
- [29] Samuel Chukwujindu Nwokolo y Julie C. Ogbulezie. «A qualitative review of empirical models for estimating diffuse solar radiation from experimental data in Africa». en. En: *Renewable and Sustainable Energy Reviews* 92 (sep. de 2018), págs. 353-393. ISSN: 13640321.

-
- DOI: 10.1016/j.rser.2018.04.118. URL: <https://linkinghub.elsevier.com/retrieve/pii/S1364032118303356> (visitado 09-10-2025).
- [30] ENEA – Italian National Agency for New Technologies, Energy and Sustainable Economic Development. *RDS–PTR 2021: Report 133*. Inf. téc. Research report published within the Italian National Electric System Research Programme. ENEA, 2021. URL: <https://www.ricercasistemmaelettrico.enea.it/archivio-documenti/category/484-report-2021-progetto-1-2.html>.
- [31] P.L.S. Jayalaxmi et al. «PIGNUS: A Deep Learning model for IDS in industrial internet-of-things». en. En: *Computers & Security* 132 (sep. de 2023), pág. 103315. ISSN: 01674048. DOI: 10.1016/j.cose.2023.103315. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0167404823002250> (visitado 18-10-2025).
- [32] *IMPORTANT SAFEGUARDS*. Accessed: Nov. 02, 2025. Black+Decker Hogar. 2025. URL: https://www.blackanddeckerhogar.com/wp-content/uploads/2025/08/TO1705SGLA_UC.pdf.
- [33] Raspberry Pi Ltd. *Raspberry Pi Pico 2 Datasheet: An RP2350-based Microcontroller Board*. Build version 2b6018e-clean, released 15 October 2024. Licensed under CC BY-ND 4.0. Raspberry Pi Ltd. Cambridge, United Kingdom, oct. de 2024. URL: <https://datasheets.raspberrypi.com/pico/pico-2-datasheet.pdf>.
- [34] Renzo Mischianti. *Raspberry Pi Pico 2: high-resolution pinout, datasheet, and specs*. <https://mischianti.org/raspberry-pi-pico-2-high-resolution-pinout-datasheet-and-specs/>. Accessed: 2025-11-04. Jun. de 2025.
- [35] *Analizador de redes CVM-C10*. <https://docs.circutor.com/docs/M001B01-01.pdf>. Circutor. 2024.
- [36] MBS AG. *MBS AG MBS-80244*. Accessed: Oct. 27, 2025. 2025. URL: <https://www.tme.com/mx/es/details/mbs-80244/transformadores-de-corriente/mbs-ag/80244/>.
- [37] AG Electrónica S.A.P.I. de C.V. *OKY3473: Módulo Convertidor TTL a RS485*. Hoja de datos técnica del módulo OKY3473 (MAX485). Fecha de revisión: 04/10/2019. AG Electrónica S.A.P.I. de C.V. República del Salvador N° 20, Segundo Piso, Ciudad de México, 2019. URL: <http://www.agelectronica.com>.
- [38] OSEPP Electronics Ltd. *DHT11 Humidity & Temperature Sensor Datasheet*. Translated version of the manufacturer's datasheet. OSEPP Electronics. 2021. URL: <http://www.droboticonline.com/>.

-
- [39] Eco Green Energy. *HELIOS PLUS 425-450 W / 144 Cells / 9 Bus Bars Monocrystalline Module EGE-425-450W-144M (M6)*. Datasheet del módulo fotovoltaico HELIOS PLUS 425–450 W. 2024. URL: <https://cdn.autosolar.co/pdf/HELIOS-PLUS-425-450M.pdf>.
- [40] Growatt New Energy Technology Co., Ltd. *Growatt MIC Series Inverter User Manual*. Technical manual for Growatt MIC series inverters. Growatt New Energy Technology Co., Ltd. n.d. URL: <https://cdn.autosolar.co/pdf/manual-inversor-growatt-mic-1500tl-x-mppt.pdf>.
- [41] Diego Camargo Castillo y María Gabriela Chocontá Rubio. «GESTIÓN DEL CONSUMO ENERGÉTICO PARA MOTORES DE INDUCCIÓN MEDIANTE UNA MICRORRED ELÉCTRICA». es. En: ().
- [42] Alessandro Ciocia et al. «Self-Consumption and Self-Sufficiency in Photovoltaic Systems: Effect of Grid Limitation and Storage Installation». en. En: *Energies* 14.6 (mar. de 2021), pág. 1591. ISSN: 1996-1073. DOI: 10.3390/en14061591. URL: <https://www.mdpi.com/1996-1073/14/6/1591> (visitado 12-11-2025).
- [43] Inés Suárez-Ramón y ... Álvarez. «A general sizing methodology of grid-connected PV systems to meet the zero-energy goal in buildings». En: *Energy* 308 (2024). ISSN: 0360-5442. DOI: 10.1016/j.energy.2024.23545. URL: <https://www.sciencedirect.com/science/article/pii/S0360544224023545>.
- [44] Fernando Arribas Uguet. *Autoconsumo y autosuficiencia fotovoltaicos*. Documento final. Fecha de revisión: 24 de febrero de 2023. Autor independiente. Feb. de 2023. URL: <https://www.solidarenergia.es/wp-content/uploads/2023/02/Autoconsumo-y-autosuficiencia-fotovoltaicas-0.2.pdf>.