



universidad  
de león



**Escuela de Ingenierías**

**Industrial, Informática y Aeroespacial**

**GRADO EN INGENIERÍA EN ELECTRÓNICA  
INDUSTRIAL Y AUTOMÁTICA**

Trabajo de Fin de Grado

Sistema electrónico de monitorización energética

Electronic energy monitoring system

Autor: Carmen Maestro Villayandre  
Tutor: Ángela Díez Díez

**(SEPTIEMBRE, 2023)**

<b>UNIVERSIDAD DE LEÓN</b> <b>Escuela de Ingenierías Industrial, Informática y Aeroespacial</b> <b>GRADO EN INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA</b> <b>Trabajo de Fin de Grado</b>
<b>ALUMNO:</b> Carmen Maestro Villayandre
<b>TUTOR:</b> Ángela Díez Díez
<b>TÍTULO:</b> Sistema electrónico de monitorización energética
<b>TITLE:</b> Electronic Energy Monitoring System
<b>CONVOCATORIA:</b> septiembre, 2023
<b>RESUMEN:</b> <p>En el contexto actual, el uso inteligente de la energía es crucial para la preservación del medio ambiente. Mi proyecto se centra en analizar el patrón de consumo energético de una vivienda, con el propósito de implementar prácticas eficientes y detectar posibles fallas en el futuro.</p> <p>Para llevar a cabo este proyecto, se utilizó un contador de energía capaz de comunicarse con un microcontrolador. El modelo seleccionado fue el SDM230 de EASTRON, mientras que la Raspberry Pi 4 actuó como el microcontrolador. La elección del SDM230 se fundamenta en su capacidad de comunicación mediante el protocolo Modbus.</p> <p>La implementación se logró mediante el uso del lenguaje de programación Python, aprovechando las librerías "minimalmodbus" y "request". Este programa se encarga de leer los registros del contador y de enviar estos datos a una base de datos. Los parámetros para leer del contador, así como los números de los registros, se obtuvieron del datasheet proporcionado por el fabricante. Posteriormente, se adquirieron habilidades en PHP. Mediante esta tecnología y la librería "request", se logró actualizar los datos capturados por el contador. Estos datos se almacenaron en una base de datos MySQL, organizados por día y hora. La presentación de estos datos resultó interesante. Se desarrolló una página web sencilla utilizando HTML, que incluye un formulario para que el usuario elija una fecha y hora específicas. Al hacerlo, se generan gráficos que ilustran el consumo energético del día seleccionado. Esta interactividad se logró gracias al uso de JavaScript.</p> <p>Adicionalmente, el proyecto contempla la posibilidad de implementar un control de potencia mediante el uso de tiristores y resistencias. Este sistema entra en acción cuando se está utilizando una cantidad significativa de electricidad, evitando que se exceda la energía contratada y previniendo cortes inesperados en el suministro eléctrico de la vivienda.</p> <p>En resumen, mi proyecto tiene como objetivo comprender y regular el consumo energético en un entorno doméstico. Aprendí a establecer comunicación entre</p>

dispositivos, enviar datos a una página web y presentarlos de manera comprensible. Estas habilidades no solo me proporcionaron conocimientos técnicos valiosos, sino que también me sensibilizaron sobre la importancia de utilizar la energía de manera responsable en nuestra vida diaria.

**ABSTRACT:**

In today's context, intelligent energy usage is pivotal for environmental preservation. My project revolves around analyzing the energy consumption pattern of a household, aiming to implement efficient practices and identify potential faults in the future.

To carry out this project, an energy meter capable of communication with a microcontroller was employed. The chosen model was the SDM230 by EASTRON, while the Raspberry Pi 4 served as the microcontroller. The SDM230 was selected due to its communication capabilities through the Modbus protocol.

Implementation was achieved through the utilization of the Python programming language, leveraging the "minimalmodbus" and "request" libraries. This program is responsible for reading the meter's registers and transmitting this data to a database. The parameters for reading from the meter, along with register numbers, were obtained from the manufacturer's datasheet. Subsequently, proficiency in PHP was acquired. Through this technology and the "request" library, the data captured by the meter was updated. These data were stored in a MySQL database, organized by day and hour. The presentation of this data proved intriguing. A simple webpage was developed using HTML, incorporating a form for users to select a specific date and time. Upon selection, graphs illustrating the energy consumption for the chosen day are generated. This interactivity was achieved using JavaScript

Additionally, the project contemplates the potential implementation of power control via thyristors and resistors. This system comes into play when a significant amount of electricity is being used, preventing the exceeding of contracted energy, and preempting unexpected power outages in the household.

In summary, my project aims to comprehend and regulate energy consumption within a domestic setting. I acquired skills in establishing device communication, sending data to a webpage, and presenting it understandably. These abilities not only imparted valuable technical knowledge but also heightened my awareness of the importance of responsible energy utilization in our daily lives.

**Palabras clave:** Modbus, Raspberry Pi, base de datos , contador, energía

**Firma del alumno:**

**VºBº Tutor/es:**

# TABLA DE CONTENIDO

TABLA DE CONTENIDO .....	1
GLOSARIO.....	6
<b>1. INTRODUCCIÓN.....</b>	<b>7</b>
<b>2. ESTADO DEL ARTE .....</b>	<b>8</b>
<b>3. FUNDAMENTOS TEÓRICOS .....</b>	<b>10</b>
3.1. Conceptos básicos de consumo energético y su importancia. ....	10
3.2. Medición y Cuantificación del Consumo Energético.....	11
3.3. ¿Qué son las etiquetas de eficiencia energética? .....	12
3.3.1. <i>Otros factores que Influyen en el Consumo Energético</i> .....	13
3.3.2. <i>Beneficios Adicionales de la Eficiencia Energética</i> .....	13
3.4. Introducción a los sistemas de medición de energía. ....	14
3.4.1. <i>Diferentes maneras de calcular tensión y corriente</i> .....	14
3.4.2. <i>Importancia de las Herramientas de Medición:</i> .....	16
3.5. Microcontroladores y su uso en proyectos IoT.....	16
3.5.1. <i>Comparación de Plataformas para el Monitoreo de Consumo Energético:</i> <i>Raspberry Pi vs. Arduino</i> .....	16
3.5.2. <i>Diferencia Clave entre Arduino y Raspberry Pi: Acceso a peticiones HTTPS</i> ....	17
3.5.3. <i>Aplicaciones Comunes de Raspberry Pi en Proyectos IoT:</i> .....	18
3.6. Descripción del protocolo Modbus y aplicación en el monitoreo de dispositivos. .	19
3.6.1. <i>Otros protocolos de comunicación.</i> .....	19
<b>4. DISEÑO DEL SISTEMA .....</b>	<b>21</b>
4.1. Hardware y software del sistema.....	21
4.2. Arquitectura general del sistema de monitoreo.....	32
4.3. Sensor de consumo de energía. ....	34
4.3.1. <i>Registros del contador</i> .....	35

4.4.	Explicación estructura de la base de datos para el almacenamiento de datos.....	36
4.4.1.	<i>Herramientas utilizadas</i> .....	36
4.4.2.	<i>Estructura de la base de datos</i> .....	36
4.5.	Raspberry Pi: Características y capacidades. ....	38
4.5.1.	<i>Potencia de Procesamiento y Arquitectura</i> .....	38
4.5.2.	<i>Capacidad de Memoria RAM</i> .....	38
4.5.3.	<i>Conectividad Versátil</i> .....	38
4.5.4.	<i>Salida de Video de Alta Resolución</i> .....	39
4.5.5.	<i>Amplias Opciones de Sistema Operativo</i> .....	39
4.6.	Interfaz de usuario para la página web.....	40
4.7.	Implementación de control de potencia.....	40
4.7.1.	<i>Funcionamiento de los tiristores.</i> .....	41
4.7.2.	<i>Módulo de disparo de tiristor</i> .....	42
4.7.3.	<i>Esquema del circuito</i> .....	45
4.7.4.	<i>Explicación del circuito</i> .....	45
4.7.5.	<i>Resultado</i> .....	46
<b>5.</b>	<b>MARCO TECNOLÓGICO</b> .....	<b>47</b>
5.1.	Modbus: Explicación del protocolo y su uso en la comunicación de dispositivos... 47	47
5.1.1.	<i>Ventajas de protocolo Modbus</i> .....	47
5.1.2.	<i>Códigos de función</i> .....	49
5.1.3.	<i>Formato RTU</i> .....	49
5.1.4.	<i>Protocolo RS-485</i> .....	49
5.1.5.	<i>Comunicación en modo Half- duplex</i> .....	50
5.1.6.	<i>Registros en Modbus</i> .....	51
5.2.	Sensor de medición de energía: Descripción y funcionamiento.....	52
5.2.1.	<i>Ventajas de Utilizar un Contador de Energía Integrado</i> .....	52
5.2.2.	<i>Interfaz y Comunicación</i> .....	53
5.3.	Lenguajes de programación y herramientas utilizadas. ....	53
5.3.1.	<i>Librerías usadas</i> .....	53
5.3.2.	<i>Python</i> .....	54

5.3.3.	<i>HTML, CSS y PHP</i> .....	55
5.3.4.	<i>MySQL y Bases de Datos</i> .....	55
5.3.5.	<i>Librería Request</i> .....	56
5.3.6.	<i>Servidores Web y Hosting</i> .....	56
5.3.7.	<i>Herramientas utilizadas</i> .....	56
<b>6.</b>	<b>DESARROLLO</b> .....	<b>58</b>
6.1.	Configuración y programación de la Raspberry Pi. ....	58
6.1.1.	<i>Diagrama de flujo de programa Python: contadormodbus.py.</i> ....	58
6.1.2.	<i>Instalación de la Biblioteca MinimalModbus.</i> .....	59
6.1.3.	<i>Configuración del Dispositivo Modbus</i> .....	59
6.1.4.	<i>Ajuste de la Comunicación</i> .....	59
6.1.5.	<i>Lectura de Registros Modbus</i> .....	60
6.1.6.	<i>Envío de Datos a la Interfaz Web</i> .....	60
6.1.7.	<i>Control del Tiempo de Ejecución</i> .....	60
6.1.8.	<i>Comprobación del funcionamiento del código</i> .....	61
6.2.	Integración de la base de datos para el almacenamiento y recuperación de información. ....	62
6.3.	Creación de la página web para la visualización de datos de consumo. ....	64
6.3.1.	<i>Formulario de Selección de Fecha y Hora:</i> .....	64
6.3.2.	<i>Conexión a la Base de Datos:</i> .....	65
6.3.3.	<i>Consulta de Datos para la Gráfica:</i> .....	65
6.3.4.	<i>Visualización de los Datos en Tabla:</i> .....	66
6.3.5.	<i>Cierre de la Conexión a la Base de Datos:</i> .....	66
<b>7.</b>	<b>RESULTADOS</b> .....	<b>54</b>
7.1.	Ejemplos de datos capturados y visualizados en la página web.....	54
7.2.	Comparativa de datos reales con los datos obtenidos por el sistema. ....	61
7.2.1.	<i>Resultados</i> .....	62
<b>8.</b>	<b>CONCLUSIONES</b> .....	<b>61</b>
	<b>BIBLIOGRAFÍA</b> .....	<b>64</b>

---

DIRECTIVAS.....	65
FIGURAS.....	66
<b>ANEXOS.....</b>	<b>67</b>
ANEXO I-Código Python contadormodbus.py.....	67
ANEXO II-Código recopilación de datos PHP recepcion_datos.php .....	69
ANEXO III-Código web HTML PHP prueba.php .....	71
ANEXO IV CSS styles3.css .....	75
<b>ESPECIFICACIONES TÉCNICAS DEL HARDWARE .....</b>	<b>77</b>
ANEXO V-Contador Eastron .....	77
ANEXO VI – Raspberry Pi.....	78
ANEXO VII- Modulo disparador .....	79

## Índice de figuras

FIGURA 1. ETIQUETAS ENERGÉTICAS .....	12
FIGURA 2. ARQUITECTURA GENERAL DEL SISTEMA.....	32
FIGURA 3. EASTRON SDM320 .....	34
FIGURA 4. RASPBERRY PI 4 MODEL B .....	39
FIGURA 5. TIRISTOR .....	41
FIGURA 6. CURVA FUNCIONAMIENTO TIRISTOR .....	42
FIGURA 7. MÓDULO DISPARO TIRISTOR .....	44
FIGURA 8. ESQUEMA DEL CIRCUITO CONTROL DE POTENCIA .....	45
FIGURA 9. MAESTRO-ESCLAVO .....	48
FIGURA 10. DIAGRAMA FLUJO DE PYTHON .....	58
FIGURA 11. GRÁFICA 20/08 .....	55
FIGURA 12. GRÁFICA 23/08 .....	57
FIGURA 13. GRÁFICA 24/08 .....	60
FIGURA 14. MONTAJE CONTADOR .....	62
FIGURA 15. CONSUMO ORDENADOR .....	61

## Índice de tablas

TABLA 1. REGISTROS DEL CONTADOR.....	35
TABLA 2. CÓDIGOS DE FUNCIÓN.....	49
TABLA 3. CÓDIGO SOLICITADO POR EL ESCLAVO.....	61
TABLA 4. CÓDIGO RESPUESTA .....	62
TABLA 5. DATOS CAPTURADOS 20/08 .....	54
TABLA 6. DATOS CAPTURADOS 23/08 .....	56
TABLA 7. DATOS CAPTURADOS 24/08 .....	59

## Índice de ecuaciones

ECUACIÓN 1. POTENCIA ACTIVA .....	10
ECUACIÓN 2. CONSUMO ENERGÉTICO .....	11

# GLOSARIO

**Sistema HVAC:** Sistema de Calefacción, Ventilación y Aire Acondicionado utilizado para controlar la temperatura, humedad y calidad del aire en edificios y entornos cerrados.

**Gestión de Activos:** Proceso de control y mantenimiento de los activos de una organización para optimizar su rendimiento, eficiencia y durabilidad.

**Interoperabilidad:** Capacidad de diferentes sistemas, dispositivos o aplicaciones para trabajar juntos y compartir información de manera efectiva.

**BMS (Building Management System):** Sistema de Gestión de Edificios que integra y controla varios sistemas en un edificio para mejorar la eficiencia operativa y el confort.

**Voltamperio (VA):** Unidad de medida de potencia aparente en un circuito eléctrico.

**Biométricos:** Relativo a medidas físicas o características únicas de individuos, como huellas dactilares, reconocimiento facial o de voz.

**Resiliencia:** Capacidad de un sistema para resistir, adaptarse y recuperarse de perturbaciones o desafíos.

**Sistema de Supervisión y Control (SCADA):** Sistema utilizado para supervisar, controlar y administrar procesos industriales y de infraestructura crítica.

**CIP (Common Industrial Protocol):** Protocolo de comunicación ampliamente utilizado en la automatización industrial.

# 1. Introducción

En la era actual, la gestión eficiente de la energía se ha convertido en un tema muy hablado para enfrentar el crecimiento sostenible y la preservación de los recursos naturales. La búsqueda constante de formas innovadoras para monitorear, analizar y optimizar el consumo energético ha impulsado el desarrollo de nuevas soluciones tecnológicas avanzadas. Este proyecto se adentra en el mundo de la monitorización energética y la integración de dispositivos tecnológicos para la recopilación y análisis de datos en tiempo real.

El propósito fundamental de este TFG es la integración de un contador SMD230 de la reconocida marca EASTRON con una Raspberry Pi 4 Model B. Este contador, se comunica mediante Modbus, permite la captura precisa de datos energéticos esenciales para el análisis y la optimización del consumo. La Raspberry Pi, una plataforma de desarrollo versátil, se establece como el núcleo de procesamiento y control, desempeñando un papel esencial en el almacenamiento de los valores del contador.

La ejecución de un programa Python en la Raspberry Pi facilita la recopilación de datos provenientes del contador SMD230 a través del protocolo Modbus. Los datos extraídos son posteriormente almacenados en una base de datos MySQL, la utilización de librerías y tecnologías como `requests` en PHP y el método POST permite una comunicación entre la Raspberry Pi y la base de datos.

La visualización de los datos recopilados se lleva a cabo mediante una página web creada con HTML. Esta interfaz no solo proporciona una representación gráfica de los valores energéticos, sino que también facilita la interacción del usuario con los datos almacenados.

## 2. Estado del arte

- **Evolución Histórica de los Sistemas de Medición de Potencia:**

La medición de potencia ha recorrido un largo camino desde sus inicios. Desde los primeros medidores electromecánicos hasta los sistemas digitales más avanzados de hoy en día, la tecnología de medición de potencia ha experimentado una evolución notable.

**Medidores Electromecánicos:** Los primeros sistemas de medición de potencia eran medidores electromecánicos que utilizaban principios electromagnéticos para medir la corriente y el voltaje. Aunque simples, estos dispositivos carecían de precisión, por ello eran más susceptibles a desgaste y errores.

**Medidores Electrónicos:** La introducción de la electrónica permitió el desarrollo de medidores electrónicos más precisos y versátiles. Estos dispositivos digitalizaban las mediciones y permitían una mayor exactitud en la medición de parámetros.

**Sistemas de Medición Digital:** Surgieron sistemas de medición de potencia más avanzados y sofisticados. Estos sistemas podían medir una variedad de parámetros eléctricos, ofrecían pantallas digitales y podían conectarse a sistemas de control y monitoreo.

- **Impacto de la Normativa Actual:**

La normativa actual en el campo de la medición de potencia ha desempeñado un papel fundamental en la evolución de esta tecnología, impulsando cambios significativos para garantizar la precisión y la seguridad y la distribución de energía eléctrica.

**Normas de Precisión y Calibración:** La normativa actual exige que los sistemas de medición de potencia cumplan con estándares estrictos de precisión y calibración. [Directiva 2014/32/UE \(Directiva de Instrumentos de Medida\).](#)[15]

**Integración de Tecnologías Inteligentes:** La implementación de medidores inteligentes permite la recopilación y transmisión de datos en tiempo real, lo que facilita la administración de la red eléctrica y la detección de problemas. Directiva 2009/72/CE (Directiva de Mercados de Energía). [16]

**Eficiencia Energética y Gestión de Demanda:** La normativa actual también ha fomentado la medición de potencia como herramienta para promover la eficiencia energética y la gestión de la demanda. Directiva 2012/27/UE (Directiva de Eficiencia Energética). [17]

**Cumplimiento Normativo y Auditorías:** Las normativas gubernamentales y de la industria requieren que las compañías eléctricas cumplan con ciertos estándares de medición. Directiva 2019/944/UE (Directiva de Mercado Interior de la Electricidad). [18]

[1]

## 3. Fundamentos teóricos

### 3.1. CONCEPTOS BÁSICOS DE CONSUMO ENERGÉTICO Y SU IMPORTANCIA.

La energía se puede conceptualizar como la capacidad que posee un sistema físico para realizar trabajo. Por otro lado, el trabajo se manifiesta cuando se aplica una fuerza que provoca el desplazamiento de un objeto.

Para la física moderna, se establece que, para cualquier sistema y entorno, la energía es una magnitud constante. Esto implica que la energía no puede ser creada ni eliminada, sino que únicamente experimenta transformaciones de un tipo a otro. Esta comprensión se rige por la Primera Ley de la Termodinámica.

Es relevante mencionar que en este contexto también se hace necesario abordar el concepto de "potencia".

La potencia eléctrica es la proporción por unidad de tiempo, con la cual la energía eléctrica es transferida por un circuito eléctrico, o lo que es lo mismo, la cantidad de energía eléctrica entregada o absorbida por un elemento en un momento determinado.

La potencia activa es consumida por todos los dispositivos eléctricos utilizados, su registro se lleva a cabo mediante contadores electrónicos que instala la empresa suministradora para así poder medir el total de la energía consumida por el cliente en un periodo de tiempo estipulado.

Se define como potencia activa es la que se aprovecha cuando se pone en funcionamiento un dispositivo eléctrico, el cual realiza un trabajo (o produce calor). La unidad en el Sistema Internacional de Unidades es el vatio (W).

$$P = UI\cos\varphi = U * I = R^2 * I$$

ECUACIÓN 1.POTENCIA ACTIVA

El término "consumo energético" se refiere a la cantidad de energía que es utilizada por dispositivos eléctricos y sistemas durante un período de tiempo específico. Esencialmente, representa la cantidad de energía que fluye a través de estos dispositivos para mantenerlos en funcionamiento.

Es crucial destacar la importancia del análisis del consumo energético. Este análisis no solo permite cuantificar y comprender el uso de energía, sino que también desempeña un papel clave en la identificación de oportunidades para mejorar la eficiencia energética.

[2][3]

### **3.2. MEDICIÓN Y CUANTIFICACIÓN DEL CONSUMO ENERGÉTICO**

La unidad fundamental para medir la energía es el vatio-hora (Wh) o sus múltiplos, como el kilovatio-hora (kWh). Los dispositivos eléctricos generalmente indican su consumo en kilovatios-hora en sus especificaciones o etiquetas. La fórmula para calcular el consumo energético es:

$$\text{Consumo energético (kWh)} = \text{Potencia (kW)} \times \text{Tiempo (h)}$$

ECUACIÓN 2. CONSUMO ENERGÉTICO

Donde la potencia se expresa en kilovatios (kW) y el tiempo en horas (h). Esta fórmula permite estimar la cantidad total de energía consumida por un dispositivo durante un período específico.

Además de la medición del consumo total, es útil entender los patrones de consumo energético. Algunos dispositivos consumen energía constantemente mientras están encendidos, como las luces. Otros, como los electrodomésticos, pueden tener ciclos de encendido y apagado. La potencia máxima que un dispositivo consume cuando se enciende desde un estado apagado se llama "potencia de arranque". Comprender estos patrones puede ayudar a tomar decisiones informadas sobre cómo y cuándo utilizar los dispositivos para optimizar la eficiencia energética.

### 3.3. ¿QUÉ SON LAS ETIQUETAS DE EFICIENCIA ENERGÉTICA?

La eficiencia energética se refiere a la capacidad de obtener resultados óptimos utilizando la menor cantidad de recursos energéticos posible. Este enfoque permite a los usuarios llevar a cabo sus actividades de manera más responsable desde el punto de vista energético. Un ejemplo relevante es la (Directiva de Eficiencia Energética, 2012/27/UE) [17] que establece que los electrodomésticos deben estar equipados con una etiqueta de calificación energética, Reglamento (UE) 2017/1369 [20].

Esta etiqueta proporciona información esencial sobre la eficiencia energética de los electrodomésticos, tanto en términos de consumo eléctrico como de consumo de agua. La representación gráfica en forma de letras, que va desde la A hasta la G, tiene una finalidad muy concreta y no es meramente decorativa. Más bien, su objetivo es brindar a los consumidores un medio para comprender el comportamiento energético del electrodoméstico en cuestión. Además de evaluar el consumo eléctrico, esta etiqueta también contempla otros aspectos cruciales, como el consumo de agua y el nivel de ruido generado durante su funcionamiento.

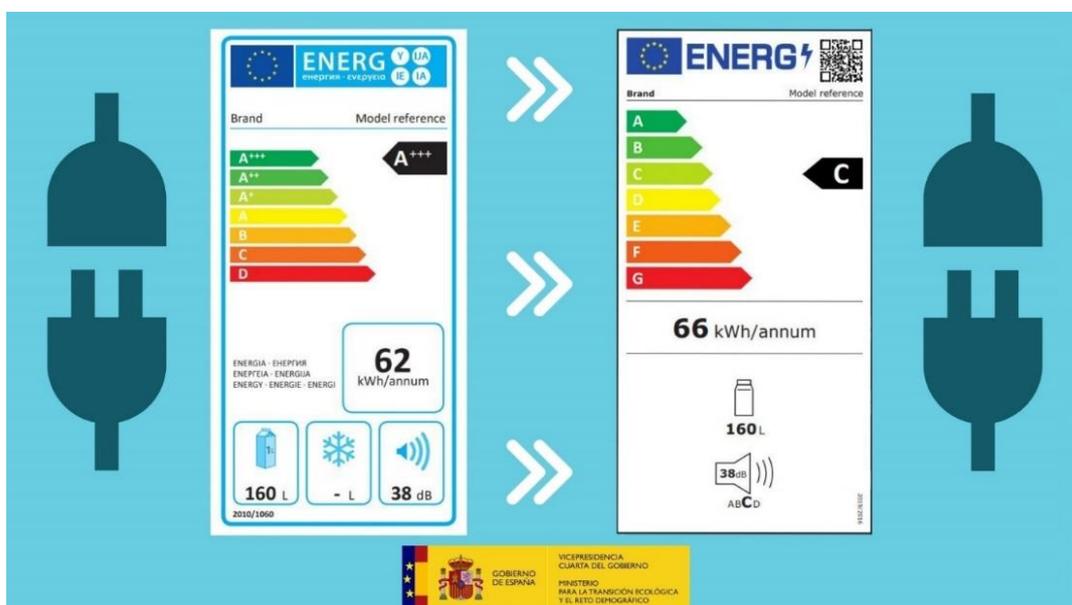


FIGURA 1. ETIQUETAS ENERGÉTICAS [21]

En última instancia, la clave reside en que, a mayor eficiencia energética, menor será el consumo. Como resultado directo, los usuarios experimentarán un ahorro significativo en sus facturas de electricidad.[4]

### **3.3.1. Otros factores que Influyen en el Consumo Energético**

- **Tiempo de Uso.**

Cuánto tiempo un dispositivo está en funcionamiento es directamente proporcional al consumo de energía. Reducir el tiempo de uso puede reducir el consumo.

- **Modo de Espera**

Muchos dispositivos consumen energía incluso cuando están en modo de espera. Desconectar los dispositivos cuando no se usan puede reducir este consumo fantasma.

- **Hábitos de Uso**

Los hábitos de los usuarios también influyen. Por ejemplo, el uso responsable de la iluminación y la calefacción/refrigeración puede tener un impacto significativo en el consumo.

- **Mantenimiento**

Los dispositivos bien mantenidos tienden a funcionar de manera más eficiente y consumir menos energía. Los dispositivos bien mantenidos tienden a funcionar de manera más eficiente y consumir menos energía.

### **3.3.2. Beneficios Adicionales de la Eficiencia Energética**

Además del ahorro económico directo y la reducción del impacto ambiental, la eficiencia energética conlleva otros beneficios importantes:

- **Resiliencia Energética**

Utilizar energía de manera eficiente puede ayudar a reducir la dependencia de los recursos energéticos y mejorar la resiliencia frente a interrupciones del suministro.

- **Contribución a la Sostenibilidad**

La eficiencia energética es un pilar clave en la construcción de sistemas energéticos sostenibles para el futuro.

- **Cambio Cultural**

Fomentar la eficiencia energética promueve una cultura de consumo responsable y cuidado del medio ambiente.

- **Innovación y Empleo**

La inversión en tecnologías y prácticas eficientes estimula la innovación y puede generar empleo en sectores relacionados.

### **3.4. INTRODUCCIÓN A LOS SISTEMAS DE MEDICIÓN DE ENERGÍA.**

Las herramientas de medición de energía son dispositivos creados para calcular y registrar la cantidad de electricidad consumida en diferentes situaciones, que abarcan desde entornos domésticos hasta instalaciones industriales. Estos instrumentos permiten una supervisión constante y un análisis detallado de los patrones de consumo de energía. Esta capacidad de seguimiento contribuye a detectar momentos de alta demanda, detectar ineficiencias y proporcionar información clave, con el objetivo de optimizar la gestión de recursos energéticos, en nuestro proyecto el instrumento elegido es un contador de energía.

#### **3.4.1. Diferentes maneras de calcular tensión y corriente**

En la medición de energía eléctrica, la determinación precisa de la tensión y la corriente es esencial para calcular el consumo de energía. Aunque se utilizará un contador como enfoque principal para medir la energía, existen diferentes métodos para medir la tensión y la corriente. Aquí exploraremos algunas de estas técnicas:

- **Divisor de Voltaje y Corriente:** Este método se basa en la Ley de Ohm y utiliza resistencias para crear caídas de tensión o corriente proporcionales en un circuito. Midiendo estas caídas, se puede calcular la tensión o corriente original. Es común en aplicaciones de bajo voltaje y corriente.

- **Transformadores de Corriente (TC)**

Los TCs son dispositivos que transforman corrientes elevadas en valores manejables para su medición. Son fundamentales en sistemas de distribución de alta potencia, como las redes eléctricas, ya que permiten medir corrientes altas de manera segura.

- **Transformadores de Potencial (TP):**

Los TPs se utilizan para reducir la tensión a niveles seguros y medibles. Son cruciales para medir tensiones de alta magnitud sin poner en riesgo a los operadores y los dispositivos de medición.

- **Efecto Hall**

Este principio aprovecha el efecto Hall, que ocurre cuando una corriente en un conductor atraviesa un campo magnético perpendicular. La tensión resultante es proporcional a la corriente y al campo magnético. Se utiliza en aplicaciones donde la medición precisa es crucial, como en dispositivos de control de motor y sistemas de tracción eléctrica.

- **Pinzas Amperimétricas**

Estas herramientas permiten medir la corriente en un conductor sin interrumpir el circuito. Son útiles para mediciones rápidas y no invasivas en entornos industriales y domésticos.

- **Registadores de Datos**

Estos dispositivos monitorizan y registran datos de tensión y corriente a lo largo del tiempo. Son esenciales para el análisis de patrones de consumo y la identificación de tendencias a largo plazo.

- **Analizadores de Red Eléctrica**

Estos equipos miden y analizan la calidad de la energía eléctrica, incluyendo parámetros como armónicos, desequilibrios y fluctuaciones. Son fundamentales en la gestión de sistemas eléctricos complejos.

### **3.4.2. Importancia de las Herramientas de Medición:**

La relevancia de las herramientas de medición de energía va más allá del simple seguimiento de los patrones de consumo. Al permitir la detección temprana de problemas, como fugas de energía o dispositivos defectuosos que consumen más energía de lo normal, estas herramientas pueden ahorrar costos y mejorar la seguridad. Además, un análisis constante y detallado de los datos recopilados a lo largo del tiempo proporciona información valiosa para la planificación y la toma de decisiones informadas sobre la optimización de recursos. Por ejemplo, si se identifican patrones de alta demanda en ciertos momentos del día, las empresas pueden implementar estrategias para redistribuir la carga y reducir los picos de demanda.

## **3.5. MICROCONTROLADORES Y SU USO EN PROYECTOS IOT.**

Los microcontroladores son herramientas destacadas por su versatilidad y rendimiento excepcional, encontrando diversas aplicaciones, entre ellas, en el vasto campo del Internet de las Cosas (IoT). Estas plataformas brindan una base flexible para el desarrollo de soluciones tecnológicas que abarcan desde la automatización doméstica hasta la captura de datos en tiempo real. Su diseño compacto y su capacidad para soportar sistemas operativos completos han catapultado su popularidad en proyectos de IoT, debido a su accesibilidad y capacidad de adaptación.[5]

### **3.5.1. Comparación de Plataformas para el Monitoreo de Consumo Energético:**

#### **Raspberry Pi vs. Arduino.**

Cuando se trata de elegir la plataforma adecuada para el monitoreo de consumo energético, es esencial considerar varias opciones disponibles. En este contexto, la elección a menudo recae en la Raspberry Pi, Arduino y otros microcontroladores conocidos. Cada una de estas

plataformas tiene sus propias ventajas y aplicaciones. Aquí comparamos estas plataformas y explicamos por qué la Raspberry Pi ha sido elegida en este caso.

#### **3.5.1.1. Características de Arduino**

- **Simplicidad y Bajo Consumo**

Arduino se destaca por su simplicidad y eficiencia en proyectos de control y automatización. Es ideal para aplicaciones que requieren bajo consumo de energía y respuestas en tiempo real.

- **Control de Sensores y Actuadores**

Arduino es especialmente adecuado para proyectos que involucran la interacción directa con sensores y actuadores, lo que es esencial en el monitoreo de consumo energético.

- **Programación Sencilla:**

La programación de Arduino se basa en un lenguaje de programación fácil de entender, similar al lenguaje C/C++, lo que hace que sea accesible para personas con diferentes niveles de experiencia en programación.

#### **3.5.2. Diferencia Clave entre Arduino y Raspberry Pi: Acceso a peticiones HTTPS**

Tanto Arduino como Raspberry Pi han ganado popularidad como plataformas versátiles y poderosas para la creación de proyectos. Sin embargo, una de las diferencias clave entre estas dos opciones radica en su capacidad para manejar peticiones HTTPS, lo cual puede tener un impacto significativo en el tipo de aplicaciones que puedes desarrollar.

Arduino, conocida por su enfoque en la programación de microcontroladores, es excepcionalmente hábil para gestionar interacciones físicas en tiempo real, tiene limitaciones en términos de conectividad web. No es nativamente capaz de manejar peticiones HTTPS de manera directa. Esto significa que, si bien es posible realizar algunas comunicaciones de HTTP (Protocolo de Transferencia de Hipertexto), el acceso a sitios web y servicios en línea que requieran conexiones seguras HTTPS puede ser más complicado y puede requerir soluciones adicionales, como proxies o servicios intermedios.

Una de las ventajas notables de Raspberry Pi es su capacidad para manejar peticiones HTTPS sin restricciones significativas. Esto le permite interactuar con servicios en línea de manera más directa y segura, lo que es esencial en aplicaciones que requieren transferencia de datos cifrada y autenticación, como aplicaciones web, sistemas de monitoreo remoto y más.

En resumen, la principal diferencia entre Arduino y Raspberry Pi en términos de manejo de peticiones HTTPS radica en la capacidad de la Raspberry Pi para acceder y comunicarse de manera más directa y segura con servicios web que requieren conexiones cifradas. Si tu proyecto implica una interacción significativa con servicios en línea y seguridad web, la Raspberry Pi podría ser la elección más adecuada. Sin embargo, si tu enfoque es principalmente la interacción.

### **3.5.3. Aplicaciones Comunes de Raspberry Pi en Proyectos IoT:**

- **Automatización Residencial**

La Raspberry Pi se puede utilizar para controlar sistemas de iluminación, termostatos, cerraduras y otros dispositivos del hogar, lo que permite la creación de hogares inteligentes.

- **Monitoreo Ambiental**

Puede utilizarse para recopilar datos ambientales como temperatura, humedad y calidad del aire, lo que es valioso en aplicaciones de monitoreo y análisis medioambiental.

- **Agricultura Inteligente**

En proyectos agrícolas, la Raspberry Pi puede ayudar a monitorear y controlar sistemas de riego, analizar datos del suelo y gestionar condiciones climáticas.

- **Industria y Manufactura**

En entornos industriales, la Raspberry Pi puede ser empleada para el monitoreo de maquinaria, la gestión de activos y el seguimiento de la cadena de suministro.

- **Seguridad y Vigilancia**

Puede utilizarse en sistemas de vigilancia y seguridad, conectando cámaras y sensores para detectar intrusiones o cambios en entornos controlados.

- **Salud y Bienestar**

En aplicaciones de salud, la Raspberry Pi puede recopilar datos biométricos y monitorear condiciones médicas para su análisis y seguimiento.

En nuestro proyecto la usaremos para el monitoreo del consumo energético de una vivienda.

### **3.6. DESCRIPCIÓN DEL PROTOCOLO MODBUS Y APLICACIÓN EN EL MONITOREO DE DISPOSITIVOS.**

Modbus es un protocolo de comunicación utilizado para transmitir información entre dispositivos electrónicos a través de redes LAN o WAN. Su simplicidad y eficiencia lo hacen especialmente adecuado para aplicaciones industriales y de automatización, incluido el monitoreo de dispositivos como medidores de energía. Modbus opera en un modelo maestro-esclavo, donde un dispositivo maestro solicita y recopila datos de uno o más dispositivos esclavo. Nuestro proyecto se usará Modbus RTU RS485 que lo explicaremos más adelante.[6]  
[8]

#### **3.6.1. Otros protocolos de comunicación.**

- **Modbus TCP/IP**

Esta variante del protocolo Modbus opera sobre redes Ethernet y utiliza el protocolo de Internet (IP) para la comunicación. Permite la transmisión de datos en tiempo real a través de redes TCP/IP, lo que lo hace ideal para sistemas que requieren comunicación rápida y confiable.

- **BACnet**

Es un protocolo ampliamente utilizado en la automatización de edificios para el control y monitoreo de sistemas HVAC, iluminación y otros equipos. BACnet permite la interoperabilidad entre dispositivos de diferentes fabricantes en sistemas de gestión de edificios (BMS).

- **Profibus**

Un protocolo serie utilizado en la automatización industrial para la comunicación entre dispositivos y sistemas en redes industriales. Tiene variantes como Profibus DP y Profibus PA para diferentes aplicaciones.

- **EtherNet/IP**

Combina la tecnología Ethernet con el protocolo CIP (Common Industrial Protocol). Se utiliza en sistemas de control industrial para la comunicación en tiempo real entre dispositivos y sistemas de automatización.

- **DNP3 (Distributed Network Protocol 3)**

Esencialmente en la industria de servicios públicos, DNP3 se utiliza para la comunicación entre dispositivos y sistemas remotos en aplicaciones como sistemas de gestión de agua y energía.

Aunque hay una variedad de protocolos disponibles, en el caso específico del monitoreo de energía, el Modbus RTU RS485 se ha destacado por sus ventajas en términos de eficiencia, confiabilidad y facilidad de implementación. El protocolo Modbus RTU opera en un formato serie, donde los datos se transmiten en bits a través de un cable RS485.

## 4. Diseño del sistema

### 4.1. HARDWARE Y SOFTWARE DEL SISTEMA.

En secciones posteriores, entraremos en más detalles sobre los componentes hardware y software seleccionados, que son los siguientes:

- **Hardware:**

**Raspberry Pi:** La Raspberry Pi es el componente principal del hardware en el proyecto. Actúa como un controlador que se encarga de la comunicación con el contador EASTRON a través del protocolo Modbus RS485.

**Contador EASTRON:** El contador EASTRON es otro componente de hardware crucial. Registra y proporciona los datos de consumo de energía de la vivienda.

**Modbus RS485:** El protocolo Modbus RS485 es la interfaz de comunicación que permite que la Raspberry Pi se comuniquen con el contador EASTRON.

- **Software:**

**Python con la librería minimalmodbus.**

**PHP:** PHP desempeña un papel importante en el aspecto de servidor web de tu proyecto. Con PHP, puedes procesar y enviar los datos recopilados por la Raspberry Pi a una base de datos y a una página web. PHP administra la parte del servidor y la interacción con la base de datos.

**HTML, CSS y JavaScript:** Estos lenguajes de programación web son fundamentales para la creación de la página web que representa los datos del consumo energético.

## 4.2. ARQUITECTURA GENERAL DEL SISTEMA DE MONITOREO.

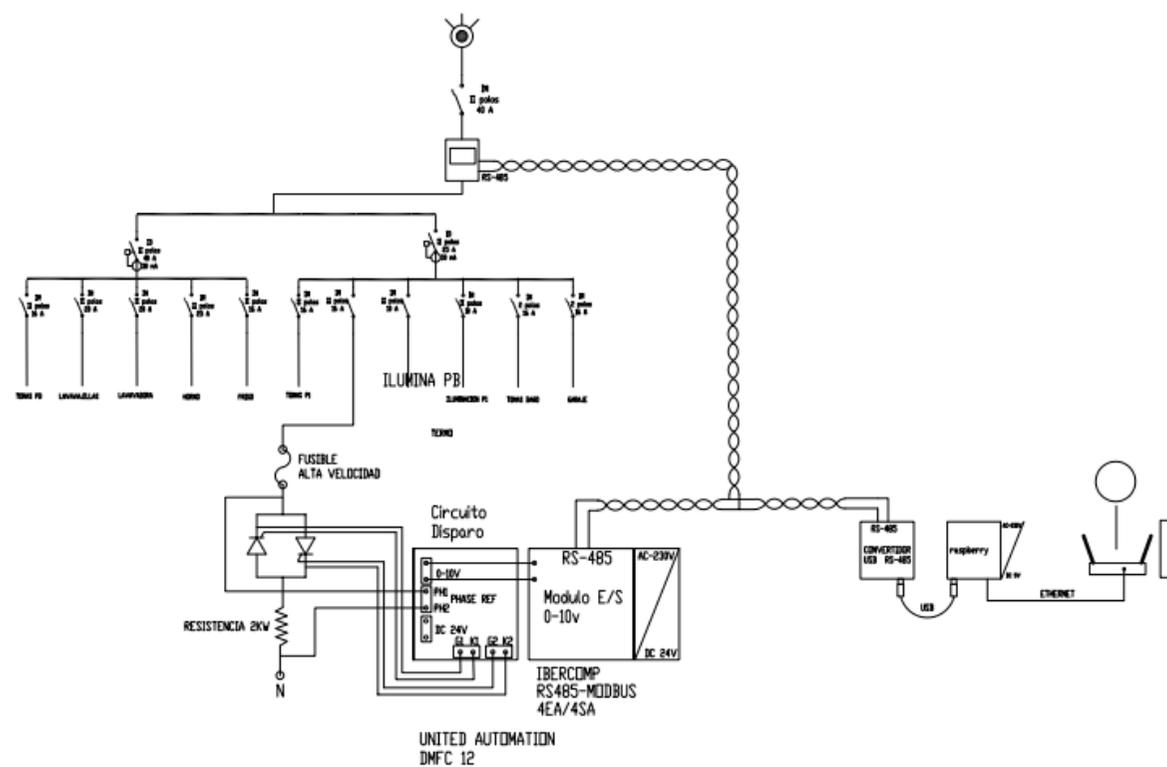


FIGURA 2.ARQUITECTURA GENERAL DEL SISTEMA

Podemos observar cómo está conectado el contador de energía al RS-485 mediante una conexión que parece de par trenzado. Esto tiene sus explicaciones que son:

- **Interferencia Electromagnética Minimizada:**

El par trenzado, con su estructura de cables entrelazados, proporciona una excelente protección contra interferencias electromagnéticas (EMI). Al conectar el contador al RS485 mediante par trenzado, estamos garantizando que las señales eléctricas sensibles, como las que representan el consumo energético, estén menos expuestas a las perturbaciones electromagnéticas generadas por otros dispositivos y cables en el entorno residencial. Esto asegura mediciones más precisas y datos confiables.

- **Reducción del Ruido Externo:**

En un entorno residencial, diversas fuentes de ruido eléctrico pueden afectar la calidad de la señal de datos. El entrelazado de los cables en el par trenzado actúa como una barrera efectiva contra el ruido externo, minimizando su impacto en las transmisiones de datos.

- **Distancias de Transmisión Ampliadas:**

El par trenzado también se destaca por su capacidad para admitir transmisiones de datos a distancias grandes. En nuestro sistema, esto es esencial, ya que el contador de consumo y otros componentes pueden estar ubicados en diferentes áreas de la vivienda.

El contador mide el consumo de electrodomésticos, como la nevera y otros aparatos, así como el uso de las luces. En cuanto a la conexión del termo, podemos observar lo siguiente.

Hemos incorporado una característica para garantizar un uso eficiente y seguro de la energía. Implementamos un "control de potencia" que entra en juego si la potencia calculada por el contador excede el límite establecido por el contrato de energía de la vivienda, en apartados posteriores explicaremos esto más detalladamente.

Para asegurar un funcionamiento seguro de este control, hemos tomado medidas adicionales. Antes de llegar al control de potencia, hemos insertado un componente de "fusible de alta velocidad". Este fusible actúa reaccionando rápidamente ante cualquier aumento repentino de energía.

El sistema funciona de la siguiente manera: el contador de consumo de la vivienda calcula la potencia utilizada. Si esta potencia supera la cantidad de energía permitida según el contrato, el control de potencia entra en acción. Sin embargo, antes de llegar al control, la corriente pasa por el fusible de alta velocidad. Este fusible está diseñado para detectar rápidamente cualquier sobrecarga y desconectar la energía antes de que cause problemas.

### 4.3. SENSOR DE CONSUMO DE ENERGÍA.

El sensor elegido para el proyecto ha sido Eastron SDM230-Modbus con comunicación en serie. Se proporciona un puerto de comunicación, RS485 Modbus.



FIGURA 3.EASTRON SDM320[22]

Uno de los mejores productos de EASTRON es el medidor inteligente SDM230Modbus, lo que distingue a este medidor es su capacidad para transmitir valores de datos en una forma codificada muy precisa. En este caso, todos los valores de datos en el SDM230Modbus se transfieren utilizando el estándar IEEE754 de 32 bits [19], lo que asegura una representación exacta de la información. Esta codificación de datos es esencial para garantizar una recopilación precisa y confiable de información en tiempo real.

Para entender la respuesta en formato decimal he implementado este código en el archivo Python que se aloja en la Raspberry pi.[9]

```

x=int(x)
N=bin(x)[2:]
N=N.zfill(16)
a=int(N[0])
b=int(N[1:9],2)
c=int("1"+N[9:],2)
return(-1)**a*c/(1<<(len(N)-9-(b-127)))

```

#### 4.3.1. Registros del contador

Más tarde se explicará la importancia de los registros en Modbus, estos registros son los parámetros que mide el contador, pues ahora para saber que registros utilizar, tenemos que ir a la documentación del fabricante, en nuestro caso EASTRON, en el datasheet encontraremos esta información.

Los registros de entrada se utilizan para indicar los valores actuales de los valores medidos y calculados. Cada parámetro se mantiene en dos registros consecutivos de 16 bits.

Dirección	Parámetro		Byte alto	Byte bajo
30001	Tensión	V	00	00
30007	Corriente	A	00	06
30013	Potencia activa	W	00	0C
30073	Energía activa importada	KWh	00	48
30085	Demanda total de energía	W	00	54
30343	Energía activa total	KWh	01	56

TABLA 1.REGISTROS DEL CONTADOR

Es decir, en nuestro proyecto los parámetros que mediremos son la tensión, la corriente, la potencia activa, la energía activa importada, la demanda total de energía y la energía activa total.

---

## **4.4. EXPLICACIÓN ESTRUCTURA DE LA BASE DE DATOS PARA EL ALMACENAMIENTO DE DATOS.**

### **4.4.1. Herramientas utilizadas**

Usamos dos herramientas fundamentales que son phpMyAdmin y WampServer. Estas dos herramientas trabajan en conjunto para automatizar el proceso de guardar la información de consumo energético en la base de datos y mostrarla en nuestra web.

#### **1. PhpMyAdmin**

Imagina phpMyAdmin como una especie de asistente para organizar y administrar la información en una base de datos. Funciona como una interfaz visual que nos permite gestionar los datos de manera más sencilla, sin tener que escribir código. Con phpMyAdmin, podemos crear, editar y eliminar datos en la base de datos de una manera amigable y accesible.

#### **2. WampServer**

Por otro lado, tenemos WampServer, que es como un "kit de herramientas" para desarrollar aplicaciones web en nuestra propia computadora. Combina tres elementos esenciales: Apache (que maneja cómo se muestra el contenido en la web), MySQL (donde almacenamos nuestros datos en la base de datos) y PHP (un lenguaje de programación para crear aplicaciones web). Lo especial de WampServer es que nos permite simular un ambiente web real en nuestra computadora, lo que es muy útil para probar y ajustar nuestra aplicación antes de publicarla en línea, esto me sirvió para hacer pruebas en el ordenador y ver su funcionamiento antes de instalarlo en la Raspberry pi.

### **4.4.2. Estructura de la base de datos**

El fundamento de nuestro sistema de almacenamiento de datos reside en una estructura que muestra las 24 horas del día y se basa en la implementación de siete columnas.

### **1. Día del Año**

Aquí reside la fecha exacta, correspondiendo a uno de los 365 días del año, por ejemplo, el 23 de agosto ocupa el día 235. Esta columna da forma a las 365 filas de nuestra base de datos, donde cada fila representa un día.

### **2. Tensión:**

Este valor, aunque suele permanecer estable, su análisis nos resulta beneficioso. Cualquier cambio significativo puede indicar problemas en el sistema. Su observación continua nos permite detectar errores.

### **3. Corriente**

### **4. Potencia Activa**

Es el dato que se presenta instantáneamente en nuestra interfaz web y se muestra en una gráfica para un análisis visual claro. Representa nuestra demanda de energía en tiempo real.

### **5. Energía activa importada**

Un factor importante para entender nuestra relación con la red eléctrica. Nos permite monitorear cuánta energía obtenemos del proveedor y evita excedernos de la energía contratada.

### **6. Demanda total de potencia:**

Este valor brinda información sobre nuestra máxima demanda de energía. Su comprensión nos ayuda a evitar excesos y mantener un uso eficiente de la electricidad.

### **7. Energía activa total**

Esta columna completa el cuadro, indicando cuánta energía hemos consumido en el día. Es un resumen esencial de nuestro consumo diario.

Cada una de estas columnas alberga 24 valores, uno para cada hora del día. Estos valores inician en 0 y se llenan automáticamente gracias al programa que opera en la Raspberry Pi. Este dispositivo captura los datos del contador eléctrico y alimenta nuestra base de datos.

## **4.5. RASPBERRY PI: CARACTERÍSTICAS Y CAPACIDADES.**

La Raspberry Pi 4 Model B, la pieza central de nuestra implementación tecnológica. Esta versión ofrece un conjunto de atributos que la hacen especialmente adecuada para proyectos de monitoreo de consumo energético y otras aplicaciones relacionadas con la automatización y el Internet de las Cosas (IoT).[5]

### **4.5.1. Potencia de Procesamiento y Arquitectura**

La Raspberry Pi 4 Model B se destaca por su potente procesador quad-core ARM Cortex-A72 de 64 bits, el cual opera a una velocidad de reloj de hasta 1.5 GHz. Esta capacidad de procesamiento brinda la potencia necesaria para ejecutar aplicaciones más complejas y llevar a cabo cálculos en tiempo real, aspectos fundamentales en el contexto del monitoreo de consumo de energía. La arquitectura de 64 bits, por su parte, contribuye a mejorar el rendimiento y la eficiencia de las operaciones, aspectos cruciales para tareas que requieren un procesamiento continuo y preciso.[5]

### **4.5.2. Capacidad de Memoria RAM**

Con opciones de memoria RAM que van desde 2 GB hasta 8 GB, la Raspberry Pi 4 Model B permite afrontar proyectos con mayores requisitos de memoria y multitarea. Esta capacidad es esencial para ejecutar aplicaciones de forma fluida, lo que es especialmente valioso en entornos donde se requiere manejar grandes volúmenes de datos, como en el monitoreo de energía.[5]

### **4.5.3. Conectividad Versátil**

La Raspberry Pi 4 Model B está diseñada para ofrecer una conectividad completa y versátil. Sus puertos USB 3.0 permiten una transferencia de datos más rápida y eficiente, lo que resulta

valioso para la captura y el procesamiento eficaz de información en tiempo real. Además, la placa ofrece soporte para redes inalámbricas Wi-Fi y Bluetooth, lo que facilita la comunicación con otros dispositivos y la transmisión de datos a través de la red. Esta conectividad versátil es esencial para la comunicación efectiva entre los componentes del sistema de monitoreo y su integración en un entorno de Internet de las Cosas. [5]

#### 4.5.4. Salida de Video de Alta Resolución

La Raspberry Pi 4 Model B también se destaca por su capacidad de salida de video de alta resolución. Con puertos HDMI duales capaces de admitir hasta dos pantallas 4K, esta placa es ideal para proyectos que requieren visualizaciones de datos en alta definición, como en aplicaciones de monitoreo de consumo energético donde los gráficos y las representaciones visuales son cruciales para la comprensión de los patrones de energía. [5]

#### 4.5.5. Amplias Opciones de Sistema Operativo

La Raspberry Pi es compatible con una variedad de sistemas operativos, incluyendo Raspbian (renombrado como Raspberry Pi OS), Ubuntu, y más. Esto permite a los desarrolladores seleccionar el sistema operativo que mejor se adapte a sus necesidades y habilidades, lo que contribuye a la flexibilidad en el desarrollo de proyectos de monitoreo de consumo energético. [5]

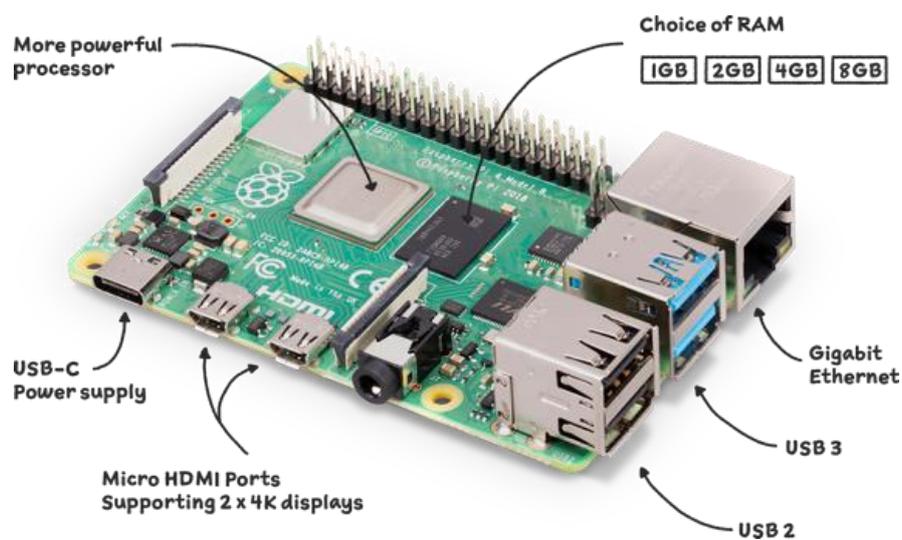


FIGURA 4. RASPBERRY PI 4 MODEL B [23]

#### **4.6. INTERFAZ DE USUARIO PARA LA PÁGINA WEB.**

En el siguiente apartado explicaré el código de la página web usando tecnologías como HTML, PHP y CSS. Ahora, quiero contar un poco más sobre cómo lo hice y cómo aseguré que mi página esté disponible para más personas, no solo en mi casa.

Para que cualquiera pueda ver mi página web en Internet, no solo en mi casa, decidí usar un servicio de hosting llamado Hostalia. Básicamente, Hostalia es como una tienda en línea donde pongo mi página para que cualquiera pueda verla en cualquier momento, sin importar la hora o el día. Esto me ayuda a compartir lo que hice con más personas en diferentes partes del mundo.

Al usar Hostalia, mi página web obtuvo una dirección web única que la identifica en Internet. Esta dirección es como la puerta de entrada a mi página, y cualquiera que tenga Internet puede usarla para entrar y ver lo que hice. Pero Hostalia hace más que solo mostrar mi página, también cuida cosas importantes como guardar los archivos, que son los programas que en apartados posteriores explicaré más detalladamente, de mi página y así protegerla de problemas de seguridad.

#### **4.7. IMPLEMENTACIÓN DE CONTROL DE POTENCIA**

En el contexto de la eficiencia energética, el control inteligente de la potencia es un elemento esencial para garantizar un uso sostenible y responsable de la energía en los hogares. Este sistema de control es una solución innovadora para el exceso de consumo y evitar costos innecesarios.

La demanda de electricidad en un hogar puede variar ampliamente a lo largo del día. En ocasiones, puede exceder la potencia que se ha contratado, lo que puede resultar en cargos adicionales o incluso cortes de suministro. Una solución para esto es introducir una implementación inteligente basada en resistencias eléctricas, que operan como un regulador de potencia.

Cuando la demanda de energía se acerca al límite contratado, el sistema de control de potencia actúa. Las resistencias eléctricas, se activan para consumir una cantidad calculada de energía adicional. Esto tiene el efecto de reducir momentáneamente la carga total y así la potencia demandada. Una vez que la demanda vuelve a estar dentro de los límites contratados, las resistencias se desactivan. Los tiristores son uno de los componentes utilizados para activar y desactivar las resistencias eléctricas en función de la demanda de potencia.

#### 4.7.1. Funcionamiento de los tiristores.

Un tiristor es un componente electrónico que permite el flujo de corriente eléctrica en una sola dirección, similar a un diodo. Sin embargo, para que conduzca en esa dirección, necesita ser activado por una pequeña corriente eléctrica, como sucede con un transistor.

En esencia, funciona como un interruptor que puede ser controlado eléctricamente. A diferencia de un transistor, el tiristor es capaz de manejar corrientes de salida mucho más grandes, lo que implica la manipulación de altas potencias.

A medida que exploramos más sobre los tiristores, identificaremos diferencias adicionales en comparación con los transistores. Los tiristores pueden ser considerados como dispositivos de dos estados (biestables), ya que tienen dos posibles estados de operación, y también son unidireccionales, ya que permiten el flujo de corriente en una sola dirección. Además, es importante destacar que los tiristores pueden manejar corrientes tan elevadas como 100 amperios o incluso más.



FIGURA 5. TIRISTOR [24]

La activación de un tiristor ocurre cuando una pequeña corriente es aplicada a su puerta G. En este estado, el tiristor se comporta como un interruptor cerrado entre su ánodo y cátodo, lo que permite que pase una corriente directa entre ambos.

Por otro lado, en ausencia de corriente en la puerta G, el tiristor se encuentra en un estado abierto, y no habrá flujo de corriente entre el ánodo y el cátodo.

En términos más sencillos, podríamos visualizar el tiristor como un interruptor compuesto por el ánodo y el cátodo, mientras que la puerta G actúa como el mecanismo para abrirlo o cerrarlo (activarlo), todo a través de una señal eléctrica.

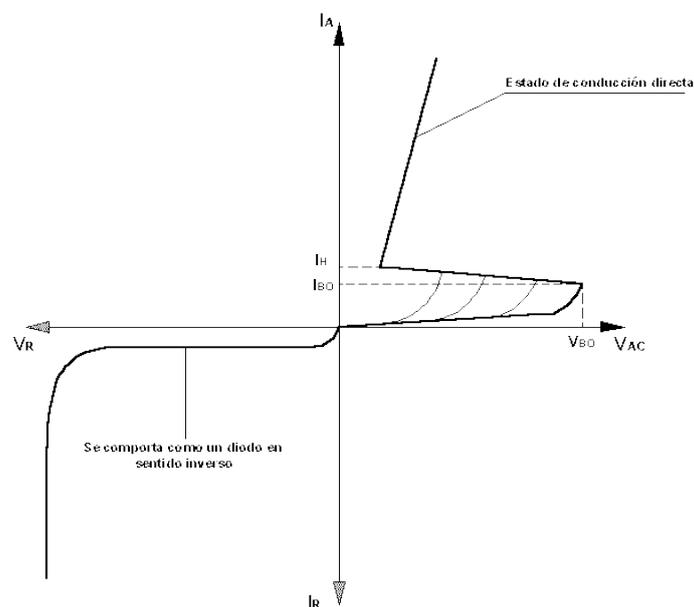


FIGURA 6. CURVA FUNCIONAMIENTO TIRISTOR [25]

El tiristor permanece activado, aunque cortemos la corriente por la puerta G y siempre debe de estar polarizado directamente. [12][14]

#### 4.7.2. Módulo de disparo de tiristor

En la arquitectura del sistema que hemos visto en apartados anteriores hemos visto como hay un un módulo de disparo para los tiristores.

Un módulo de disparo de tiristores es un componente esencial en circuitos de control de potencia que involucran tiristores o SCR (Rectificadores Controlados de Silicio). El módulo de

disparo de tiristores se encarga de generar el impulso adecuado para activar los tiristores.[13]

#### **4.7.2.1. Funcionamiento de los módulos de disparo de tiristores**

- **Detección del Punto de Disparo**

Antes de que un tiristor pueda conducir corriente, debe ser "disparado" o activado. Esto se logra aplicando un pulso de corriente de control en su compuerta. El módulo de disparo detecta el punto en el ciclo de corriente alterna en el cual el tiristor debe ser activado.

- **Generación del Pulso de Disparo**

Una vez que se identifica el momento adecuado para activar el tiristor, el módulo de disparo genera un pulso de corriente de control en la compuerta del tiristor. Este pulso es de corta duración y está sincronizado con el ciclo de corriente alterna.

- **Control de fase**

En algunos casos, es necesario controlar el ángulo de fase en el cual se activa el tiristor dentro del ciclo de corriente alterna. El módulo de disparo puede ajustar este ángulo de fase, lo que permite controlar la cantidad de energía que se entrega a la carga conectada.

- **Protección y Monitoreo**

Los módulos de disparo de tiristores también pueden incluir funciones de protección y monitoreo. Pueden detectar condiciones anormales como sobrecorriente, sobretensión o temperaturas elevadas, y tomar medidas para evitar daños a los tiristores y al sistema en general.

- **Aplicaciones**

Los módulos de disparo de tiristores se utilizan en una amplia gama de aplicaciones, como controladores de velocidad de motores, reguladores de temperatura, sistemas de iluminación controlada y en sistemas de potencia de corriente alterna en general.

#### 4.7.2.2. Módulo elegido

Módulo de disparo de tiristor, DMFC12 415VAC, 11-Pines



FIGURA 7. MODULO DISPARO TIRISTOR

El DMFC12 es un módulo de disparo monofásico o de fase a fase con la opción de control de disparo por ráfaga o ángulo de fase, o una combinación de ambos, es decir, el arranque en ángulo de fase y, a continuación, el disparo por ráfaga.[13]

### 4.7.3. Esquema del circuito

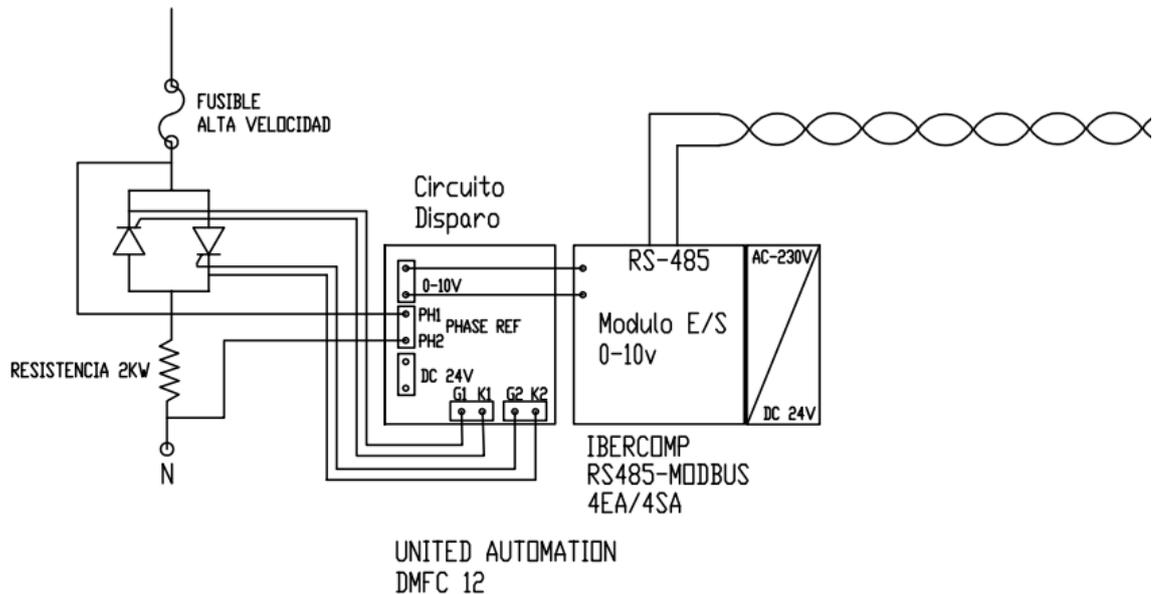


FIGURA 8.ESQUEMA DEL CIRCUITO CONTROL DE POTENCIA

### 4.7.4. Explicación del circuito

- **Comparación con la Potencia Contratada**

Una vez calculada la potencia actual por el contador, el programa de la Raspberry Pi compara este valor con la potencia contratada. Si la potencia actual se acerca o supera el límite de potencia contratada, el programa toma la decisión de activar el sistema de control de potencia para asegurar que la potencia se mantenga dentro de los límites acordados.

- **Generación de la Señal de Disparo**

Cuando el módulo E/S recibe una señal de activación, envía una señal al módulo de disparo. El módulo de disparo, al recibir la señal de activación, enciende los tiristores.

Una vez que los tiristores están encendidos, permiten que la corriente fluya a través de la resistencia y hacia la carga conectada a tierra.

Cuando el ciclo de corriente alterna se completa, los tiristores se apagan automáticamente debido a su funcionamiento unidireccional.

#### **4.7.5. Resultado**

Los beneficios de este sistema de control son grandes. En primer lugar, garantiza que los hogares no excedan su capacidad de potencia contratada, eliminando así cargos innecesarios. En segundo lugar, reduce la posibilidad de cortes de suministro debido a excesos de demanda, lo que garantiza un suministro constante de energía.

---

## 5. Marco tecnológico

### 5.1. MODBUS: EXPLICACIÓN DEL PROTOCOLO Y SU USO EN LA COMUNICACIÓN DE DISPOSITIVOS.

Modbus constituye un protocolo de comunicación ubicado en la capa 7 del Modelo OSI. Este protocolo se basa en la estructura maestro/esclavo o cliente/servidor y fue concebido por Modicon en 1979 para su línea de controladores lógicos programables.[6] [8]

#### 5.1.1. Ventajas de protocolo Modbus

##### 5.1.1.1. Modbus es de acceso público.

Permite que diferentes dispositivos y sistemas de fabricantes distintos puedan comunicarse entre sí de manera efectiva. Esto fomenta la interoperabilidad en entornos heterogéneos y promueve la adopción del protocolo en una amplia variedad de aplicaciones industriales y de automatización.

##### 5.1.1.2. Implementación sencilla

La simplicidad de Modbus no solo reduce los costos de desarrollo, sino que también minimiza los errores de implementación. Los mensajes Modbus son claros y directos, lo que facilita la programación y la depuración de sistemas de comunicación. Además, la comunidad de usuarios activa y las numerosas bibliotecas disponibles simplifican aún más la adopción de Modbus en diferentes plataformas.

##### 5.1.1.3. Flexibilidad en la Gestión de Datos

La capacidad de gestionar bloques de datos sin restricciones significativas es crucial en aplicaciones de monitoreo de consumo energético. Esto permite la adaptación a diferentes tipos de dispositivos y datos, lo que resulta fundamental para una implementación exitosa. La

flexibilidad inherente de Modbus simplifica la integración de nuevos dispositivos y la expansión de sistemas existentes.

En este protocolo, existen dos variantes que presentan diferencias en la representación numérica de los datos y en detalles del protocolo que varían ligeramente.

Modbus RTU se destaca como una representación compacta en formato binario de los datos, mientras que Modbus ASCII se caracteriza por ser una representación más legible del protocolo, aunque menos eficiente en términos de rendimiento.

Cada dispositivo presente en una red Modbus posee su propia dirección exclusiva. Si bien todos los dispositivos tienen la capacidad de enviar órdenes Modbus, es más común permitir que solo un dispositivo asuma el papel de maestro. Dentro de cada comando Modbus, se incluye la dirección del dispositivo destinatario de la orden. Aunque todas las unidades reciben la trama, únicamente el dispositivo destinatario la ejecuta, a menos que se active un modo denominado "Broadcast".[6] [8]

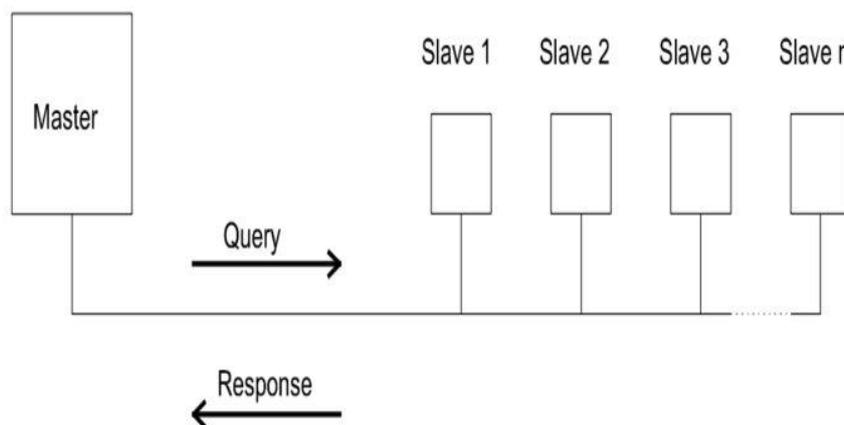


FIGURA 9. MAESTRO-ESCLAVO [26]

### 5.1.2. Códigos de función

La parte del código de función de un mensaje del protocolo MODBUS define la acción que debe realizar el esclavo.[7]

Código de función	Código de subfunción	Nombre	Descripción
3 (0x03)	–	Leer registros de mantenimiento	Leer n registros de salida o internos
4 (0x04)	–	Leer registros de entrada	Leer n registros de entrada
43 (0x2B)	14 (0x0E)	Leer identificación del dispositivo	Leer los datos de identificación del esclavo
43 (0x2B)	15 (0x0F)	Obtener fecha y hora	Leer la fecha y hora del esclavo

TABLA 2. CÓDIGOS DE FUNCIÓN

### 5.1.3. Formato RTU

Cuando los controladores se configuran para comunicar en una red Modbus utilizando modo RTU (Unidad de Terminal Remota), cada byte, 8 bits, del mensaje contiene dos caracteres hexadecimales de 4 bits. La ventaja es que permite una mejor productividad de información que el modo ASCII para la misma velocidad. Cada mensaje se transmite sin interrupción.

Cuando se transmiten mensajes en redes estándar Modbus serie, cada byte se envían de izquierda a derecha.[6] [8]

### 5.1.4. Protocolo RS-485

Para transmitir a largas distancias o con altas velocidades RS-485 es buena opción. Utilizando enlaces con RS-485 no hay limitación de dispositivos, se pueden conectar hasta 32 dispositivos en un solo bus de dato. [6] [8]

#### **5.1.4.1. Ventajas de RS-485**

- Bajo costo
- Capacidad de interconexión:
- Longitud de Enlace
- Rapidez

#### **5.1.4.2. Limitaciones y Consideraciones de RS-485**

- **Terminación de Línea**

En las redes RS-485, la terminación adecuada de la línea es esencial para evitar problemas de reflexión de señales. La falta de terminación puede resultar en señales reflejadas que interfieren con la comunicación, lo que afecta la integridad de los datos transmitidos.

- **Capacidad de Carga**

Aunque RS-485 permite conectar múltiples dispositivos en un solo bus de datos, es importante considerar la capacidad de carga total y la impedancia de la línea. Con demasiados dispositivos o una impedancia inadecuada, la calidad de la señal puede deteriorarse.

#### **5.1.5. Comunicación en modo Half- duplex**

El modo half duplex se refiere a un sistema de comunicación donde uno o varios emisores pueden interactuar entre sí o con uno o varios receptores, pero solo un emisor puede estar activo en un momento determinado. Este esquema se asemeja a una conversación en la que una persona hace una pregunta y luego escucha atentamente para obtener una respuesta, antes de que otra persona pueda hablar.

En una red RS-485, el "maestro" asume el papel de iniciar una "conversación" enviando una "consulta" dirigida a un "esclavo" específico. El "maestro" permanece atento para escuchar la respuesta del "esclavo". Sin embargo, si el "esclavo" no responde en un período de tiempo definido (determinado por el software en el "maestro"), el "maestro" abandonará la "conversación".

En resumen, el modo half duplex permite una comunicación interactiva en una dirección a la vez, donde los participantes se turnan para transmitir y recibir información. Esto se asemeja a una conversación en la vida cotidiana, donde solo una persona habla mientras que los demás escuchan, y luego se produce un intercambio de roles. [6][8]

#### **5.1.6. Registros en Modbus**

Los "registros" en Modbus son áreas de memoria dentro de los dispositivos esclavos donde se almacenan los datos. Los registros son utilizados para leer y escribir información en los dispositivos. Hay varios tipos de registros en el protocolo Modbus:

- **Registros de Entrada**

Estos registros contienen datos que provienen del dispositivo esclavo, como lecturas de sensores. Los dispositivos maestros pueden leer estos registros para obtener información actualizada del proceso.

- **Registros de Salida**

Los registros de salida se utilizan para enviar datos desde el dispositivo maestro al dispositivo esclavo. Esto podría usarse para configurar parámetros en el dispositivo esclavo o para enviar comandos.

- **Registros de Estado**

Los registros de estado, también conocidos como "coils", son utilizados para representar estados binarios (encendido/apagado) en el dispositivo esclavo. Los dispositivos maestros pueden leer y escribir en estos registros para controlar elementos.

- **Registros de Retención**

Estos registros almacenan datos en el dispositivo esclavo que pueden ser leídos y escritos por el dispositivo maestro. Pueden utilizarse para almacenar valores como ajustes de configuración o datos temporales.

---

## **5.2. SENSOR DE MEDICIÓN DE ENERGÍA: DESCRIPCIÓN Y FUNCIONAMIENTO.**

El sensor de medición de energía desempeña un papel crítico al permitir la captura precisa y en tiempo real de los datos de consumo eléctrico, lo que constituye la base para un análisis detallado y una toma de decisiones informadas en términos de eficiencia energética.

El sensor de medición de energía se basa en principios de medición eléctrica y captura de datos. Utiliza tecnología de detección de corriente y voltaje para calcular la potencia consumida por los dispositivos eléctricos conectados. Esta tecnología se basa en la Ley de Ohm y otros conceptos fundamentales de la electricidad para lograr mediciones precisas.

### **5.2.1. Ventajas de Utilizar un Contador de Energía Integrado**

En lugar de utilizar sensores de voltaje y corriente por separado para calcular la potencia consumida, otra solución eficiente es la implementación de un contador de energía integrado. Estos dispositivos combinan la medición de voltaje y corriente para calcular automáticamente la potencia, ofreciendo varias ventajas notables:

- **Simplicidad en la Implementación**

Los contadores de energía integrados eliminan la necesidad de instalar y calibrar sensores de voltaje y corriente por separado. Esto simplifica el proceso de implementación y reduce los requisitos de hardware.

- **Medición Precisa de Potencia**

Los contadores de energía están diseñados para calcular la potencia consumida, esto resulta en mediciones altamente precisas y confiables.

- **Menor Espacio Requerido**

La integración de la medición de voltaje y corriente en un solo dispositivo significa que se necesita menos espacio físico en comparación con la instalación de sensores individuales.

- **Funciones Adicionales**

Muchos contadores de energía ofrecen funciones adicionales, como la medición de factores de potencia, frecuencia y otras métricas relacionadas con el consumo eléctrico.

- **Mayor Eficiencia en la Comunicación**

Los contadores de energía integrados suelen estar equipados con interfaces de comunicación estándar, lo que facilita la conexión con sistemas de monitoreo y análisis, como la Raspberry Pi, el cual es nuestro caso mediante el protocolo Modbus.

- **Registro de Datos**

Los contadores de energía a menudo tienen capacidad para registrar y almacenar datos de consumo.

### **5.2.2. Interfaz y Comunicación**

El sensor de medición de energía se conecta a la Raspberry Pi 4 Model B mediante interfaces específicas, como puertos GPIO o buses de comunicación serie, dependiendo de las especificaciones del sensor. Además, puede incorporar interfaces de comunicación como Modbus para transmitir los datos capturados a la Raspberry Pi.

En nuestro caso usamos el protocolo Modbus, que más tarde explicaremos detalladamente como lo comunicamos con la Raspberry Pi.

## **5.3. LENGUAJES DE PROGRAMACIÓN Y HERRAMIENTAS UTILIZADAS.**

### **5.3.1. Librerías usadas**

Para facilitarnos trabajo a la hora de programar, a veces usamos piezas de código que otras personas ya han creado. A estas piezas de código se les llama "bibliotecas".

En este caso, he utilizado estas bibliotecas:

- **minimalmodbus**
- **requests**
- **time**

La biblioteca "minimalmodbus" nos permite interactuar de manera eficiente con dispositivos que utilizan el protocolo Modbus, ampliamente utilizado en la automatización industrial y el control de dispositivos electrónicos. Con sus funciones "minimalmodbus" facilita la comunicación con dispositivos Modbus a través de una interfaz sencilla y consistente. Esto nos ahorra tiempo y esfuerzo sin necesidad de crear rutinas de comunicación desde cero.

Luego está "requests", que es como un mensajero de internet. Cuando queremos obtener información de la web o enviar información a un lugar en línea, "requests" nos ayuda.

Por último, está "time", que nos permite medir el tiempo y decirle a la Raspberry que espere durante un cierto período. Puede ser útil para hacer que nuestro programa realice acciones en momentos específicos. [10]

### **5.3.2. Python**

En el proceso de implementación del sistema de monitoreo de consumo energético, se ha seleccionado Python como el lenguaje de programación principal para controlar la Raspberry Pi 4 Model B. Python es una elección sólida debido a su amplia gama de bibliotecas y módulos que facilitan la interacción con diferentes componentes del sistema. En este caso, se utiliza la librería MinimalModbus para establecer una comunicación confiable y precisa con el sensor de medición de energía, en nuestro un contador La sintaxis clara y legible de Python, junto con su enfoque en la legibilidad del código, simplifica el proceso de desarrollo y mantenimiento de aplicaciones. Además, la comunidad activa de Python proporciona acceso a recursos y soluciones, lo que agiliza la resolución de problemas y la optimización del sistema.[10]

### **5.3.3. HTML, CSS y PHP**

La creación de una interfaz web es esencial para permitir a los usuarios visualizar y analizar los datos de consumo de energía. En este sentido, la combinación de HTML, CSS y PHP desempeña un papel crucial.

HTML se encarga de estructurar y organizar la información en la página web, definiendo la posición de los elementos.

CSS aporta la presentación visual al estilo, diseño y aspecto de la interfaz, mejorando la estética y la experiencia del usuario.

PHP, por su parte, actúa como el motor de la interacción entre la interfaz web y la base de datos. Facilita la manipulación y recuperación de datos almacenados, permitiendo la presentación de los datos capturados y la interacción en tiempo real.

La combinación de estos lenguajes proporciona una interfaz efectiva para los usuarios, permitiendo analizar los datos de consumo con facilidad.[11]

### **5.3.4. MySQL y Bases de Datos**

La gestión y el almacenamiento de los datos de consumo energético son fundamentales para el funcionamiento del sistema de monitoreo. En esta implementación, se ha optado por MySQL como sistema de gestión de bases de datos debido a su eficiencia y capacidad para manejar grandes volúmenes de datos. Los datos recopilados por el contador y procesados por la Raspberry Pi se almacenan de manera estructurada en la base de datos MySQL. Esto permite una organización sistemática y un acceso eficiente a los datos para su posterior análisis. La base de datos tiene gran importancia en el análisis ya que ayudan a los usuarios a comprender su patrón de consumo energético y a tomar decisiones informadas para mejorar la eficiencia.[11]

#### **5.3.4.1. Integración con phpMyAdmin**

En esta implementación, además de utilizar MySQL como sistema de gestión de bases de datos, se ha integrado phpMyAdmin. Esta herramienta proporciona una interfaz gráfica de usuario que facilita la administración y el mantenimiento de la base de datos MySQL. phpMyAdmin permite realizar tareas como la creación y modificación de tablas, la gestión de

usuarios y permisos, y la ejecución de consultas SQL de manera intuitiva. Esta integración agrega una capa adicional de comodidad y control para administrar eficazmente la base de datos subyacente.

#### **5.3.5. Librería Request**

La librería Request, utilizada en la implementación, facilita la comunicación y el envío de datos desde la interfaz web a la base de datos. Permite la transmisión segura y eficiente de información, garantizando que los datos de consumo energético se almacenen correctamente y estén disponibles para su posterior análisis.

#### **5.3.6. Servidores Web y Hosting**

Para que los usuarios puedan acceder y utilizar la interfaz web de monitoreo desde cualquier ubicación, es necesario contar con servidores web y servicios de hosting. Estos servicios permiten alojar y compartir la interfaz, garantizando su accesibilidad en línea. La elección adecuada de servidores web y hosting es crucial para la disponibilidad y la experiencia del usuario. Se selecciona un proveedor de servicios confiable que asegura que la interfaz web esté en funcionamiento de manera constante y que los usuarios puedan interactuar con ella sin problemas.

#### **5.3.7. Herramientas utilizadas**

Para llevar a cabo los programas utilizando estos lenguajes de programación he utilizado Visual studio Code.

Visual Studio Code es una poderosa herramienta de desarrollo que se ha ganado la preferencia de muchos programadores debido a su eficiencia y versatilidad. Desarrollada por Microsoft, VS Code se ha convertido en una elección popular para programadores que trabajan en diversos lenguajes de programación y proyectos, incluidos aquellos relacionados con el monitoreo de consumo energético y el Internet de las Cosas (IoT).

Esta herramienta destaca por su interfaz intuitiva y de diseño limpio, que facilita la creación y edición de código de manera cómoda.

VS Code ofrece grandes funcionalidades que mejoran la productividad, como el resaltado de sintaxis o la corrección automática de errores.

Una característica distintiva de VS Code es su capacidad de ser extendido mediante la instalación de extensiones. Estas extensiones personalizan la experiencia de desarrollo según las necesidades individuales de cada programador. En nuestro caso, el monitoreo de consumo energético, las extensiones específicas para lenguajes como Python y tecnologías web como HTML y CSS.

En resumen, Visual Studio Code es una herramienta esencial para los programadores que buscan una experiencia de desarrollo eficiente y adaptada a sus necesidades. Su interfaz intuitiva, capacidad de extensión, compatibilidad con múltiples lenguajes y tecnologías, y su integración con control de versiones hacen de VS Code una elección sólida para proyectos como el monitoreo de consumo energético y otros relacionados con la tecnología IoT.

## 6. Desarrollo

### 6.1. CONFIGURACIÓN Y PROGRAMACIÓN DE LA RASPBERRY PI.

Se ha creado un programa utilizando Python junto con la biblioteca MinimalModbus para facilitar la comunicación con dispositivos Modbus. A continuación, se detalla el proceso de configuración y uso de esta librería:

#### 6.1.1. Diagrama de flujo de programa Python: contadormodbus.py.

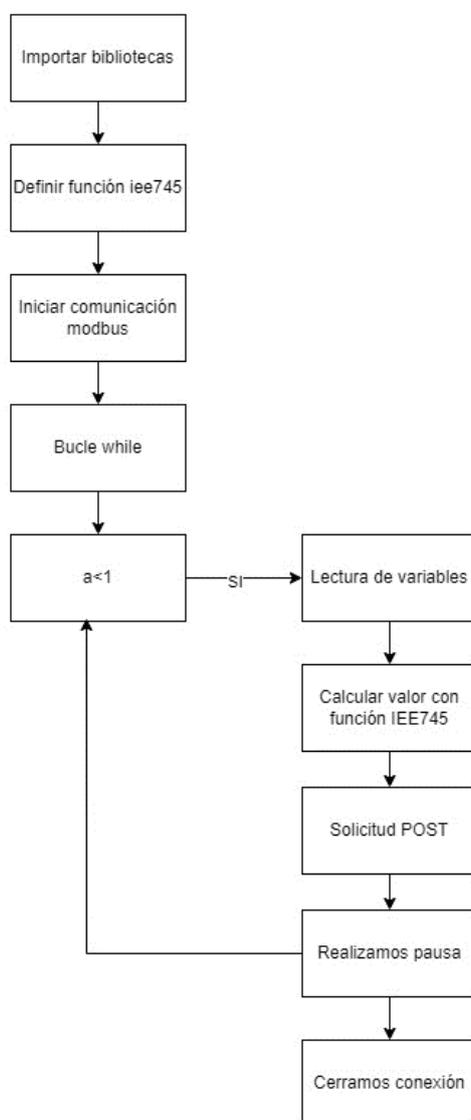


FIGURA 10. DIAGRAMA FLUJO DE PYTHON

### 6.1.2. Instalación de la Biblioteca MinimalModbus.

Para comenzar, es esencial importar la biblioteca MinimalModbus. Para hacerlo, se recomienda utilizar la terminal (CMD) y ejecutar el siguiente comando para instalar la biblioteca:

```
pip install minimalmodbus
```

### 6.1.3. Configuración del Dispositivo Modbus

Después de instalar la biblioteca, se procede a configurar la comunicación con el dispositivo Modbus. Para ello, se emplea la siguiente línea de código:

```
import minimalmodbus  
  
instrument = minimalmodbus.Instrument(port, address)
```

Aquí, instrument se refiere a la instancia del dispositivo con el que se desea comunicar. Port representa la dirección del maestro, en este caso, la Raspberry Pi, y address es la dirección del esclavo, correspondiente al contador EASTRON.

### 6.1.4. Ajuste de la Comunicación

Se deben ajustar los parámetros de comunicación para asegurar una interacción fluida y efectiva:

```
instrument.serial.baudrate = 9600  
instrument.serial.bytesize = 8  
instrument.serial.parity = minimalmodbus.serial.PARITY_NONE  
instrument.serial.stopbits = 1  
instrument.serial.timeout = 0.2  
instrument.debug = True
```

Aquí, se establece la velocidad de baudios, el tamaño de los bytes, la paridad, los bits de parada y el tiempo de espera para la comunicación. Además, el modo de depuración está activado para monitorear el proceso de comunicación.

### 6.1.5. Lectura de Registros Modbus

Para leer los registros Modbus, se utiliza el siguiente código:

```
lectura=instrument.read_register(registros[indice],functioncode=4)
```

Mediante esta línea, se leen los registros Modbus, los cuales almacenan distintos parámetros según el manual del contador. functioncode=4 establece el código de función Modbus para la operación de lectura de registros de entrada, que son registros que almacenan valores accesibles para el dispositivo maestro. registros[indice] representa el número de registro que se está leyendo.[10]

### 6.1.6. Envío de Datos a la Interfaz Web

Para enviar la información leída a una interfaz web, se utiliza la librería requests:

```
import requests
url = ""
response = requests.post(url, data=variables)
```

Aquí, url corresponde a la dirección de la interfaz web. Utilizando requests.post(), se envían los datos leídos a la interfaz para su posterior presentación y análisis.

### 6.1.7. Control del Tiempo de Ejecución

Finalmente, se agrega un control de tiempo de ejecución mediante time.sleep():

```
import time
```

```
time.sleep(10)
```

Esta línea determina la frecuencia con la que se ejecutará el programa, permitiendo la captura periódica de datos y su envío a la interfaz web.

En conjunto, este programa basado en la biblioteca MinimalModbus proporciona una solución efectiva para la comunicación con dispositivos Modbus, la lectura de registros y el envío de datos a una interfaz web. [10][11]

### 6.1.8. Comprobación del funcionamiento del código

Para saber que los datos que se están enviando son correctos, cuando ejecutamos el programa en la raspberry observamos el siguiente mensaje: [8][9]

**MinimalModbus debug mode. Will write to instrument (expecting 7 bytes back): 02 04 01 56 00 01 DO 15 (8 bytes)**

**MinimalModbus debug mode. Response from instrument: 02 04 02 40 19 0D 3A (7 bytes), roundtrip time: 0.0 ms. Time-out for reading: 200.0 ms.**

El código de protocolo MODBUS 04 lee el contenido de los registros 3X.

Vamos a fijarnos en este código 02 04 01 56 00 01 DO 15.

Nombre	HEX
Dirección del esclavo.	02
Función.	04
Dirección del registro alta (HEX)	01
Dirección del registro baja(HEX)	56
Parte alta del número de puntos.	00
Parte baja del número de puntos	01
Parte baja de la comprobación de error	DO
Parte alta de la comprobación de error	15

TABLA 3.CÓDIGO SOLICITADO POR EL ESCLAVO

Significa que el esclavo con dirección 02 y función de protocolo 04, ya que está leyendo un registro de tipo 3, le está solicitando al maestro la lectura de registro 01 56 que como ya hemos visto según el fabricante, solicitamos la potencia activa total

Ahora nos fijamos en el mensaje de salida 02 04 02 40 19 0D 3A

Nombre	
Dirección del esclavo.	02
Función.	04

<b>Cantidad de bytes.</b>	02
<b>Datos, registro alto, byte alto.</b>	40
<b>Datos, registro alto, byte bajo.</b>	19
<b>Datos, registro bajo, byte alto.</b>	0D
<b>Datos, registro bajo, byte bajo.</b>	01
<b>Error</b>	3A

TABLA 4.CÓDIGO RESPUESTA

## 6.2. INTEGRACIÓN DE LA BASE DE DATOS PARA EL ALMACENAMIENTO Y RECUPERACIÓN DE INFORMACIÓN.

### Recepcion\_datos.php

He utilizado la biblioteca MySQL para crear una base de datos en phpMyAdmin. Mediante la biblioteca Request, logré obtener los datos del contador y almacenarlos en la base de datos, habiendo desarrollado el programa en PHP.

Un ejemplo de este proceso es el siguiente:

```
$tension1 = $_POST["tension"];
```

En esta instancia, estoy solicitando información sobre la tensión. "Tensión" es uno de los parámetros que estoy midiendo en el contador, y se encuentra guardado en la lista de variables que mencioné previamente en el archivo de Python.

El método POST es uno de los métodos de solicitud empleados en el protocolo HTTP (Hypertext Transfer Protocol) para enviar datos desde el cliente hacia el servidor web.

PHP \$\_POST es una variable súper global de PHP que se utiliza para recopilar datos del formulario después de enviar un formulario HTML con el método = "post". \$\_POST también se usa ampliamente para pasar variables.

Repetiremos este procedimiento para los demás parámetros que deseamos medir, como la corriente, la potencia activa, etc.

Una vez que hayamos recibido los datos de manera adecuada, estableceremos una conexión con nuestra base de datos MySQL:

```
$conn = new mysqli($servername,$username,$password,$database);
```

```
if ($conn->connect_error) {
    die("Error de conexión: " . $conn->connect_error);
}
```

Aquí, \$servername, \$username, \$password y \$database son los valores que se utilizan para conectar con la base de datos.

Seguidamente, procederemos a realizar una consulta en la base de datos:

```
$sql1 = "SELECT * FROM datos_contador WHERE A = '$A'";
$date= date("Y-m-d");
$timestamp = strtotime($date);
$A = date("z", $timestamp) + 1;
$result = $conn->query($sql1);

if ($result->num_rows > 0) {
    $row = $result->fetch_assoc();
    $tension=explode(",",$row['C']);
```

En este caso, \$A representa el número del día en el año, y \$tension es un arreglo que contiene los valores de tensión guardados en la base de datos.

Después, actualizamos los datos con información recién capturada:

```
$tension[$n]=number_format($tension1,1);
```

Asignamos el valor \$n, que representa la hora actual, al número correspondiente en la cadena de 24 posibles valores. Por ejemplo, si son las 14:00, ocuparía la posición 14 en la lista. number\_format permite mostrar un decimal.

La actualización de datos a la base de datos ocurre mediante UPDATE:

```
$n=date("G");
$tension1=implode(",",$tension);

$sql2 = "UPDATE datos_contador SET C='$tension1' WHERE A = '$A' ";
$actualizacion = $conn->query($sql2);
```

Luego, unimos los valores del arreglo \$tension en una cadena con comas, lo que facilita la actualización en la base de datos. En este caso, "C" es el nombre de la columna donde guardamos los 24 valores de tensión cada día, y "\$A" es el número de día en los 365 días del año.

Finalmente, se verifica que la actualización se haya realizado correctamente.

Este proceso asegura que los datos del contador se capturen, almacenen y actualicen de manera efectiva en la base de datos para su posterior análisis y visualización.[11]

### 6.3. CREACIÓN DE LA PÁGINA WEB PARA LA VISUALIZACIÓN DE DATOS DE CONSUMO.

Usaremos varios lenguajes de programación para la página web, HTML, PHP, CSS para el estilo.[11]

```
<!DOCTYPE html>
<html>
<body>
<head>
AQUI VA EL RESTO DE CODIGO
</head>
</body>
</html>
```

#### 6.3.1. Formulario de Selección de Fecha y Hora:

El formulario permite al usuario seleccionar una fecha y hora.

La función printf () en el bloque PHP genera opciones para las horas del día.

Procesamiento de los Datos del Formulario:

Los datos del formulario son procesados utilizando \$\_POST para obtener la fecha y hora seleccionadas. La fecha se formatea y se calcula el número del día en el año, este mismo proceso lo haríamos para la hora.[11]

```
<form method="POST">
  <label for="fecha">Selecciona una fecha:</label>
  <input type="date" name="fecha">
  <button type="submit" name="submit">Mostrar Datos</button>
</form>
<?php
  if (isset($_POST['submit'])) {
    $fechaSeleccionada = $_POST['fecha'];
    $fechaFormateada = date("j/m/Y", strtotime($fechaSeleccionada));
    $timestamp = strtotime($fechaSeleccionada);
    $dia = date("z", $timestamp) + 1 }
?>
```

### 6.3.2. Conexión a la Base de Datos:

Se crea una conexión a la base de datos MySQL con los parámetros de conexión, ya explicada anteriormente.

### 6.3.3. Consulta de Datos para la Gráfica:

Se realiza una consulta SQL para obtener los datos relevantes para la gráfica.

Los datos se convierten en arrays para su posterior uso.

Aquí se carga la biblioteca externa Chart.js desde un CDN (Content Delivery Network). Chart.js es una librería que facilita la creación de gráficos interactivos en JavaScript.

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

La siguiente línea de código:

```
<script>
  const ctx = document.getElementById('energyChart').getContext('2d');

  const hours = ['00:00', '01:00', /Asi hasta 23/, '23:00'];
  const energyData = [<?php echo $pa1 ?>];
  const energyChart = new Chart(ctx, {
    type: 'line',
    data:
{/Aqui va el codigo que ocnfigura el estilo/}
    options: {
  }
</script>
```

#### **const ctx**

Esto obtiene el contexto 2D del lienzo con el ID "energyChart", que se utilizará para dibujar el gráfico en él.

#### **const hours**

Se define un array con las horas del día para etiquetar el eje X del gráfico.

#### **const energyData = [<?php echo \$pa1 ?>];**

Aquí se introduce el dato de consumo energético. Parece que este valor se obtiene de una variable PHP \$pa1, la cual hemos hecho la consulta anteriormente a la base de datos, que

contiene los valores de potencia activa del día que haya sido elegido por el usuario para mostrarse en la gráfica .

```
const energyChart = new Chart(ctx, { ... });
```

Aquí se crea una nueva instancia de la clase Chart proporcionada por Chart.js. Se configura el tipo de gráfico (line en este caso) y se proporcionan los datos y opciones de configuración.

#### **Data**

Se definen las etiquetas (horas) para el eje X y se establece un conjunto de datos con los valores de consumo (energyData). Se personaliza el estilo del conjunto de datos, como color de borde y relleno.

#### **Options**

Se establecen las opciones de visualización del gráfico. En este caso, se configuran los títulos para los ejes X e Y.

### **6.3.4. Visualización de los Datos en Tabla:**

Se utiliza un elemento div con imágenes y descripciones para mostrar el consumo instantáneo.

Se utiliza una tabla para mostrar los detalles del consumo eléctrico.

Los valores se obtienen de los arrays previamente procesados y se formatean con las unidades correspondientes.

### **6.3.5. Cierre de la Conexión a la Base de Datos:**

Se cierra la conexión a la base de datos después de haber obtenido y procesado los datos.

En resumen, el código HTML y PHP combina un formulario de selección con la visualización de datos en una tabla y una gráfica. Permite a los usuarios elegir una fecha y hora para mostrar información específica de consumo eléctrico obtenida de una base de datos MySQL. El código se encarga de procesar y presentar estos datos de manera comprensible y visualmente atractiva.

## 7. Resultados

### 7.1. EJEMPLOS DE DATOS CAPTURADOS Y VISUALIZADOS EN LA PÁGINA WEB.

DIA20/08

HORAS	W
0:00	485
1:00	403
2:00	288
3:00	208
4:00	209
5:00	275
6:00	204
7:00	205
8:00	271
9:00	213
10:00	2505
11:00	3416
12:00	5137
13:00	3358
14:00	3625
15:00	1427
16:00	1375
17:00	1270
18:00	3571
19:00	721
20:00	414
21:00	1527
22:00	573
23:00	671

TABLA 5.DATOS CAPTURADOS 20/08

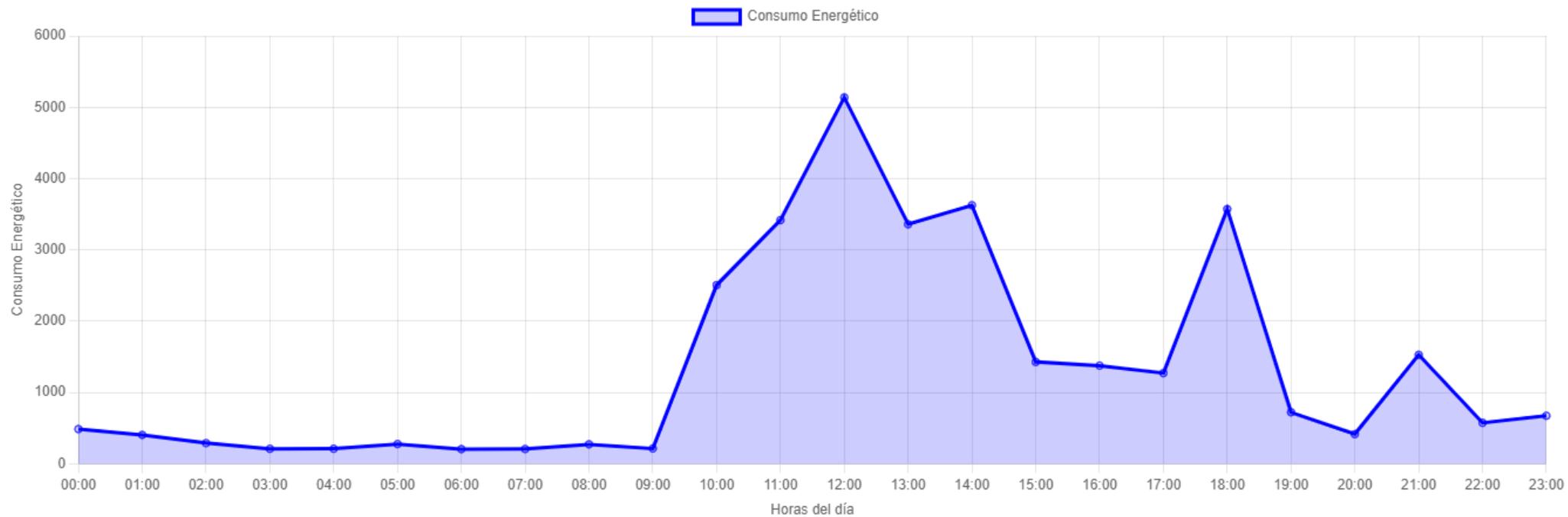


FIGURA 11.GRÁFICA 20/08

Estos son los datos capturados del día 23/08/2023

HORAS	W
0:00	692
1:00	515
2:00	280
3:00	278
4:00	339
5:00	269
6:00	200
7:00	199
8:00	2411
9:00	584
10:00	2418
11:00	3294
12:00	3204
13:00	3407
14:00	3393
15:00	1278
16:00	2368
17:00	1204
18:00	1344
19:00	1333
20:00	617
21:00	637
22:00	2848
23:00	605

TABLA 6.DATOS CAPTURADOS 23/08

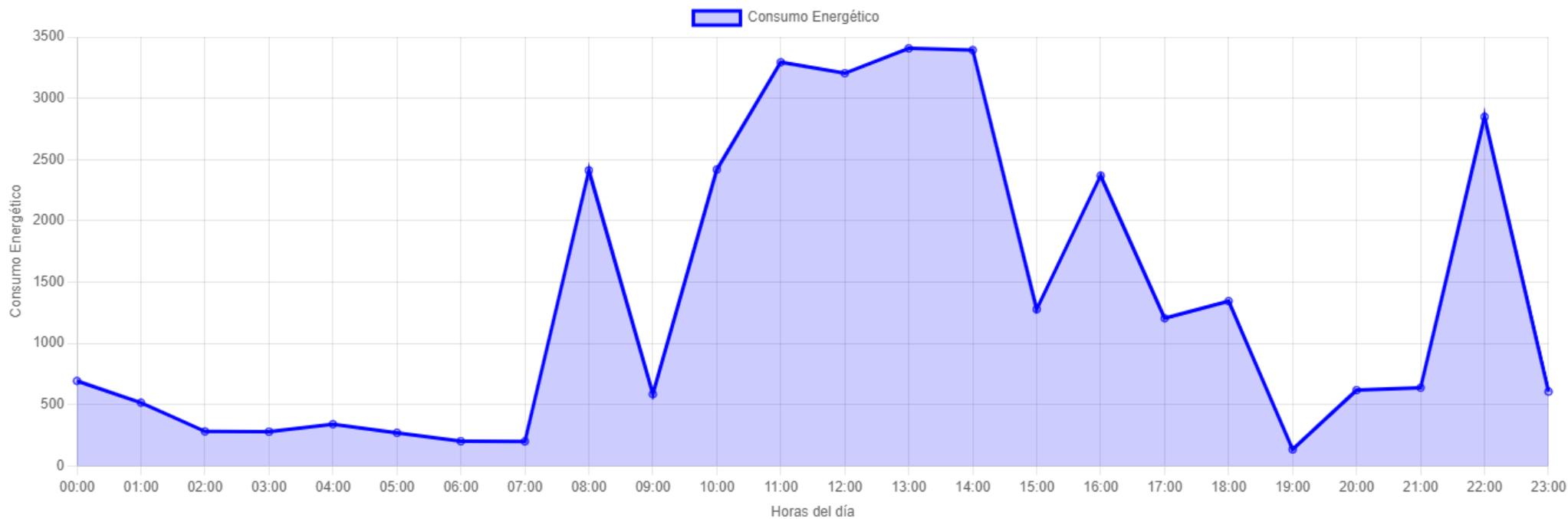


FIGURA 12.GRÁFICA 23/08

Al analizar estos valores, podemos observar patrones y tendencias en el consumo de energía a lo largo del día. Por ejemplo, el consumo tiende a ser más bajo durante las horas de la madrugada y aumenta gradualmente a medida que avanza la mañana. Es posible que durante la noche, cuando las actividades y dispositivos electrónicos suelen estar menos activos, el consumo sea menor.

Por otro lado, es interesante notar los picos de consumo en ciertos momentos del día. Estos picos podrían indicar momentos en los que se están utilizando más electrodomésticos, dispositivos electrónicos o sistemas que requieren más energía, como la calefacción o el aire acondicionado. Identificar estos momentos de alta demanda puede ser útil para tomar medidas que reduzcan el consumo en esos momentos o para evaluar la eficiencia energética de los dispositivos utilizados.

Si examinamos con atención las dos gráficas que vimos antes, podemos notar que la cantidad de consumo diario es bastante similar. Esto significa que en ambos casos la cantidad de cosas que se consumen cada día es casi igual.

Estos son los datos capturados del día 24/08/2023

HORAS	W
0:00	485
1:00	589
2:00	403
3:00	208
4:00	288
5:00	275
6:00	205
7:00	302
8:00	272
9:00	417
10:00	2609
11:00	3841
12:00	3368
13:00	1189
14:00	1302
15:00	909
16:00	1001
17:00	700
18:00	6052
19:00	3833
20:00	1081
21:00	534
22:00	1715
23:00	485

TABLA 7.DATOS CAPTURADOS 24/08

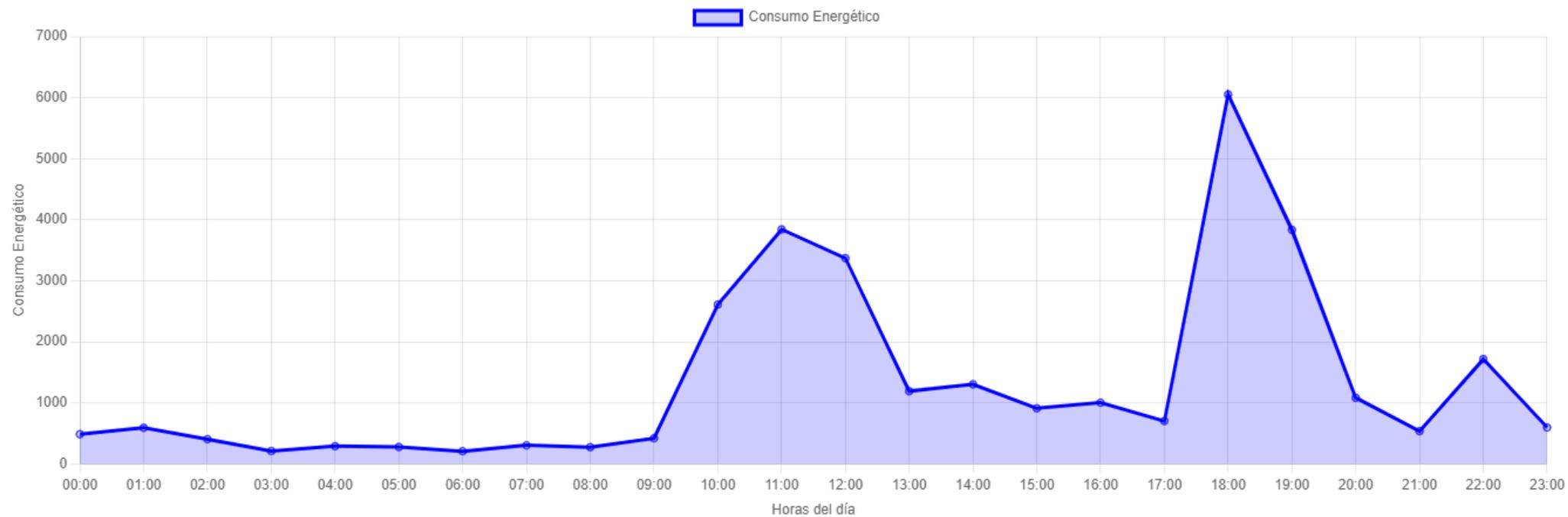


FIGURA 13.GRÁFICA 24/08

Durante la mayor parte del día, parece que se repite un patrón similar al de los dos días anteriores en lo que respecta al consumo. Sin embargo, en este día en particular, decidí hacer un pequeño experimento. A las 5 de la tarde, opté por apagar todas las luces y la mayoría de los electrodomésticos de la casa. Como resultado de esta acción, el consumo de energía disminuyó considerablemente, alcanzando alrededor de 700 vatios, que es un valor bastante bajo.

No obstante, lo interesante ocurrió en la hora siguiente, es decir, entre las 5 y las 6 de la tarde. Durante ese período, encendí nuevamente todas las luces de la casa y puse en funcionamiento varios electrodomésticos, como el horno, la vitrocerámica y la televisión. Este cambio en el patrón de uso de la energía se reflejó en un aumento notable en la cantidad de energía consumida. Los valores subieron considerablemente en ese lapso de tiempo.

En resumen, al alterar la forma en que utilizamos los electrodomésticos y las luces en el hogar, pudimos observar cómo el consumo de energía varió de manera significativa. Esto resalta la influencia que nuestras acciones cotidianas tienen en la cantidad de energía que utilizamos y cómo pequeños cambios pueden generar diferencias notables en nuestros niveles de consumo.

## **7.2. COMPARATIVA DE DATOS REALES CON LOS DATOS OBTENIDOS POR EL SISTEMA.**

Para esta sección he decidido poner el contador de esta manera para que la visualización de datos sea mucho más fácil, y se compararán con los que da el código Python.

El contador estará contando la potencia activa que consume mi ordenador cada 30 segundos.

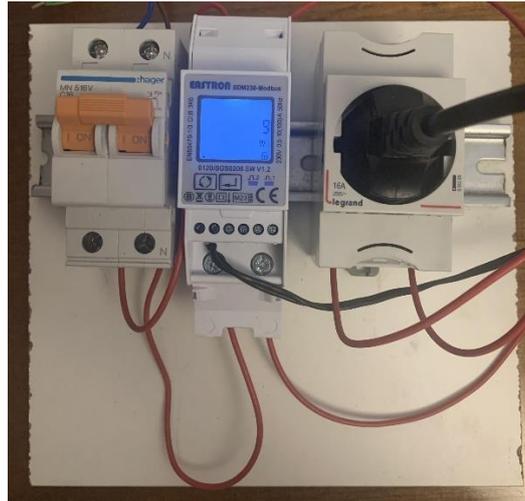


FIGURA 14.MONTAJE CONTADOR

### 7.2.1. Resultados

A las 16:15, mientras el programa de contador estaba funcionando, el consumo del ordenador era de aproximadamente 8.75 vatios. El ordenador se encontraba en un estado casi inactivo, con el software de Word abierto y una página web en ejecución.

A las 16:20, al iniciar programas como AutoCAD y Matlab, se observó un aumento notable en el consumo de energía, que ascendió a 31 vatios.

A las 16:30, se conectó el cargador al ordenador, lo que resultó en un aumento instantáneo en el consumo energético, llegando a 40.5 vatios.

A las 16:35, después de un período de tiempo y una vez que el cargador completó la carga, el consumo descendió a niveles más bajos, situándose en torno a 11 o 11.5 vatios.

Ahora si comparamos los datos de las gráficas anteriores con el consumo del día **28/08**, donde en lugar de monitorizar el consumo energético de la vivienda, se estaba registrando el consumo del ordenador, se hacen evidentes cambios significativos. Se destaca, por ejemplo, una marcada disminución en la demanda de energía en términos de vatios (W). En este caso concreto, se observa que el pico máximo de potencia registrado es de 14 W. Es importante señalar que este análisis se basa en el uso del ordenador, si bien es aplicable a cualquier otro dispositivo cuyo consumo energético deseemos evaluar.

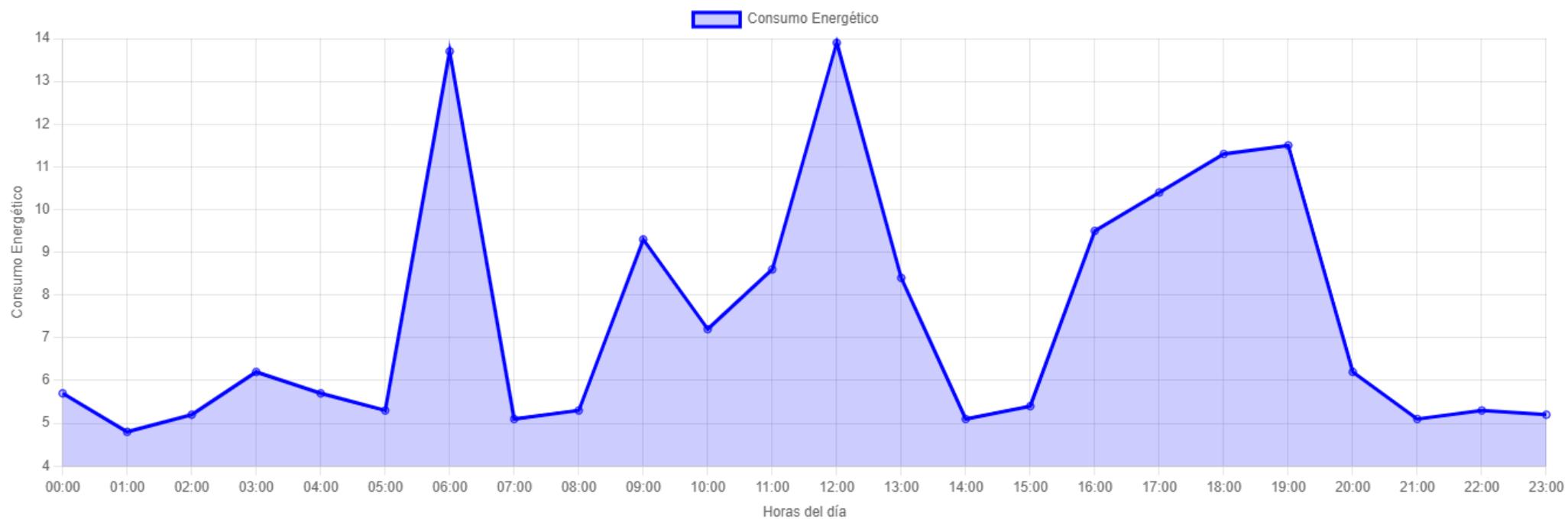


FIGURA 15.CONSUMO ORDENADOR

## 8. Conclusiones

En este proceso, me he dado cuenta de la importancia que tiene controlar y entender cómo usamos la energía en nuestras casas. A medida que profundizaba en este proyecto, se hizo evidente que el monitoreo del consumo energético no es solo una tarea técnica, sino también una responsabilidad crucial en nuestro mundo actual.

El hecho de que pudiéramos comunicar un contador con la Raspberry Pi y luego mostrar los datos en una página web personalizada me mostró cuánto poder y control tenemos sobre nuestro consumo. Al ver los números y patrones de uso, me di cuenta de cómo pequeñas acciones, como apagar las luces cuando no las necesitamos, pueden tener un impacto significativo en la cantidad de energía que utilizamos.

Además, este proyecto me ha recordado que todos podemos contribuir a la conservación de recursos y a la reducción de la huella ambiental. Al entender mejor cómo usamos la energía, podemos tomar decisiones más informadas y adoptar hábitos más sostenibles. Esto no solo beneficia a reducir los costos de energía, sino que también tiene un efecto positivo en el medio ambiente al disminuir las emisiones de carbono.

En resumen, más allá de las habilidades técnicas adquiridas, he adquirido una comprensión más profunda de la importancia de controlar nuestro consumo energético. Este conocimiento no solo ha enriquecido mi aprendizaje personal, sino que también me ha motivado a promover la conciencia y la responsabilidad en torno al uso de recursos.

# BIBLIOGRAFÍA

- [1] Mohan, N., Undeland, T. M., & Robbins, W. P. (2003). Power electronics: converters, applications, and design. John Wiley & Sons.
- [2] DE, G. (2012). Eficiencia energética. *ENERGÉTICA*, 20, 28.  
(Martínez Fernández, 2022)
- [3] Martínez Fernández. (2022). Diseño de un compensador automático de potencia reactiva mediante Arduino.
- [4] ¿Qué es la eficiencia energética y qué ventajas tiene? | Repsol. (2023, 17 julio). REPSOL. <https://www.repsol.com/es/sostenibilidad/cambio-climatico/eficiencia-energetica/que-es-la-eficiencia-energetica/index.cshtml>
- [5] Ltd, R. P. (s. f.). Buy A Raspberry Pi 4 Model B – Raspberry Pi. Raspberry Pi. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- [6] Hernández Joffre, V. H., Jiménez Anaya, E., Medina Suzunaga, G., & Montero Guerrero, P. (2009). *Control de dispositivos electrónicos mediante protocolo MODBUS RS-485 y SMS* (Doctoral dissertation).
- [7] Funciones de Modbus. (s. f.). [https://product-help.schneider-electric.com/ED/ES\\_Power/PP-HJL\\_Modbus\\_Guide/EDMS/0611IB1303/0611IB13xx/NSX\\_MB\\_Modbus\\_Protocol/NSX\\_MB\\_Modbus\\_Protocol-4.htm](https://product-help.schneider-electric.com/ED/ES_Power/PP-HJL_Modbus_Guide/EDMS/0611IB1303/0611IB13xx/NSX_MB_Modbus_Protocol/NSX_MB_Modbus_Protocol-4.htm)
- [8] Correct protocol EASTRON[ [https://www.eastroneurope.com/images/uploads/products/protocol/Correct\\_SDM230\\_Protocol.pdf](https://www.eastroneurope.com/images/uploads/products/protocol/Correct_SDM230_Protocol.pdf)
- [9] Eastron Europe - sales@eastroneurope.com. (s. f.). SDM230M Series | Eastron Europe. <https://www.eastroneurope.com/products/view/sdm230modbus>
- [10] Usage — MinimalModbus 2.1.1 Documentation. (s. f.). <https://minimalmodbus.readthedocs.io/en/stable/usage.html>
- [11] W3Schools online web tutorials. (s. f.). <https://www.w3schools.com/>

[12] Hart, D. W., & Bautista, A. B. (2001). *Electrónica de potencia* (Vol. 32). Madrid, España: Prentice Hall.

[13] Módulo de disparo de tiristor, DMFC12 415VAC, 11-Pines | RS. (s. f.). <https://es.rs-online.com/web/p/tiristores/6696039?gb=s>

[14] <https://www.areatecnologia.com>. (s. f.). Tiristor aprende fácil en continua y alterna. <https://www.areatecnologia.com/electronica/tiristor.html>

## DIRECTIVAS

[15] Parlamento Europeo y Consejo de la Unión Europea. (2014). Directiva 2014/32/UE del Parlamento Europeo y del Consejo de 26 de febrero de 2014 relativa a la armonización de las legislaciones de los Estados miembros sobre la comercialización de instrumentos de medida. Diario Oficial de la Unión Europea, L 96, 149-290.

[16] Parlamento Europeo y Consejo de la Unión Europea. (2009). Directiva 2009/72/CE del Parlamento Europeo y del Consejo de 13 de julio de 2009 sobre normas comunes para el mercado interior de la electricidad. Diario Oficial de la Unión Europea, L 211, 55-93.

[17] Parlamento Europeo y Consejo de la Unión Europea. (2012). Directiva 2012/27/UE del Parlamento Europeo y del Consejo de 25 de octubre de 2012 relativa a la eficiencia energética, por la que se modifican las Directivas 2009/125/CE y 2010/30/UE. Diario Oficial de la Unión Europea, L 315, 1-56.

[18] Parlamento Europeo y Consejo de la Unión Europea. (2019). Directiva (UE) 2019/944 del Parlamento Europeo y del Consejo de 5 de junio de 2019 sobre normas comunes para el mercado interior de la electricidad. Diario Oficial de la Unión Europea, L 158, 125-241.

[19] IEEE Standards Association. (2008). IEEE Standard for Floating-Point Arithmetic (IEEE 754). IEEE.

[20] Comisión Europea. (2017). Reglamento (UE) 2017/1369: Relativo a la etiqueta energética y por el que se deroga la Directiva 2010/30/UE. Diario Oficial de la Unión Europea, L 198/1. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R1369>

---

## FIGURAS

[21] El nuevo etiquetado energético de electrodomésticos entra en vigor el 1 de marzo. (s/f).

Gob.es. Recuperado el 31 de agosto de 2023, de

<https://www.lamoncloa.gob.es/serviciosdeprensa/notasprensa/transicion-ecologica/PublishingImages/Recursos/260221-etiquetado.jpg?RenditionID=32>

[22] Europe, E. (s/f). SDM230M series. Eastroneurope.com. Recuperado el 31 de agosto de

2023, de <https://www.eastroneurope.com/products/view/sdm230modbus>

[23] Raspberry Pi Ltd. (s/f). Raspberry pi 4. Raspberry Pi. Recuperado el 31 de agosto de

2023, de <https://assets.raspberrypi.com/static/raspberry-pi-4-labelled-f5e5dcdf6a34223235f83261fa42d1e8.png>

[24] Tiristor SCR. (2020, mayo 14). Blog Arduino, LabVIEW y Electrónica.

<https://electronicamade.com/wp-content/uploads/2020/03/SCR-tiristor.jpg>

[25] El tiristor. (s/f). Electronicafacil.net. Recuperado el 31 de agosto de 2023, de [https://st-](https://st-elf.electronicafacil.net/tutoriales/154/curvatiris.gif)

[elf.electronicafacil.net/tutoriales/154/curvatiris.gif](https://st-elf.electronicafacil.net/tutoriales/154/curvatiris.gif)

[26]Correct protocol EASTRON[

[https://www.eastroneurope.com/images/uploads/products/protocol/Correct\\_SDM230\\_Protocol.pdf](https://www.eastroneurope.com/images/uploads/products/protocol/Correct_SDM230_Protocol.pdf)

# ANEXOS

## ANEXO I-CÓDIGO PYTHON CONTADORMODBUS.PY

```

import minimalmodbus,time
import requests

def ieee745(x):
    x=int(x)
    N=bin(x)[2:]
    N=N.zfill(16)
    a=int(N[0])
    b=int(N[1:9],2)
    c=int("1"+N[9:],2)

    return(-1)**a*c/(1<<(len(N)-9-(b-127)))

def inicializa_mbus():
    global instrument
    #def print_buffer(buffer):
    #print("Buffer:", [hex(byte) for byte in buffer])

    # Configuración del puerto y dirección del contador
    port = " " #Dirección del maestro.
    address = 2 # Dirección del esclavo (contador).

    # Inicializar el objeto del instrumento MinimalModbus.

    instrument = minimalmodbus.Instrument(port, address)

    # Ajustar el modo de comunicación (RS-485)
    instrument.serial.baudrate = 9600 # Ajusta la velocidad de baudios según la
configuración del contador
    instrument.serial.bytesize = 8
    instrument.serial.parity = minimalmodbus.serial.PARITY_NONE
    instrument.serial.stopbits = 1
    instrument.serial.timeout =0.2 # Ajusta el tiempo de espera según la velocidad de
comunicación y la distancia
    instrument.debug = True

inicializa_mbus()
variables={"tension":0,"corriente":0,"potencia_activa":0,"energia_activa_importada":0,"de
m_tot_pot":0,"ene_total_act":0}

```

---

```
registros=[0,6,12,72,84,342]
a=0
while a<1:
    indice=0
    for x in variables:
        lectura=instrument.read_register(registros[indice],functioncode=4)
        time.sleep(0.1)
        valor=ieee745(lectura)
        variables[x]=valor
        indice=indice+1
    print(variables)

url = ""

# Obtener los datos del contador utilizando la función del otro archivo
# Realizar la solicitud POST con los datos del diccionario
try:
    response = requests.post(url, data=variables)
    if response.status_code == 200:
        print("Datos enviados con éxito.")
        print (response.text)
    else:
        print(f"Error al enviar datos. Código de estado: {response.status_code}")
except requests.exceptions.RequestException as e:
    print(f"Error en la solicitud: {e}")

time.sleep(10)

instrument.serial.close()
```

---

ANEXO	II-CÓDIGO	RECOPIACIÓN	DE	DATOS	PHP
<b>RECEPCION_DATOS.PHP</b>					

```

<?php
date_default_timezone_set("Europe/Madrid");
$n=date("G");
$date= date("Y-m-d");
$timestamp = strtotime($date);
$A = date ("z", $timestamp) + 1;
echo $A;

$tension1 = $_POST["tension"];
$corriente1 = $_POST["corriente"];
$potencia_activa1 = $_POST["potencia_activa"];
$energia_activa_importada1=$_POST["energia_activa_importada"];
$dem_tot_pot1=$_POST["dem_tot_pot"];
$ene_total_act1=$_POST["ene_total_act"];

$servername = "";
$username = "";
$password = "";
$dbname="" ;

$conn = new mysqli($servername,$username,$password,$dbname);

if ($conn->connect_error) {
    die ("Error de conexión: " . $conn->connect_error);
}

$sql1 = "SELECT * FROM datos_contador WHERE A = '$A'";

$result = $conn->query($sql1);

if ($result->num_rows > 0) {
// Obtener la primera fila de los resultados como un array asociativo
$row = $result->fetch_assoc();

// Acceder a los valores de la fila por sus nombres de columna
$tension=explode(",",$row['C']);
$corriente=explode(",",$row['D']);
$potencia_activa =explode(",",$row['E']);
$energia_activa_importada=explode(",",$row['F']);
$dem_tot_pot=explode(",",$row['G']);
$ene_total_act=explode(",",$row['H']);

```

---

```

$tension[$n]=number_format($tension1,1);
$corriente[$n]=number_format($corriente1,1);
$potencia_activa[$n]=number_format($potencia_activa1,1);
$energia_activa_importada[$n]=number_format($energia_activa_importada1,1);
$dem_tot_pot[$n]=number_format($dem_tot_pot1,1);
$ene_total_act[$n]=number_format($ene_total_act1,1);
// Actualizar el dato en el array
$tension1=implode(",",$tension);
$corriente1=implode(",",$corriente);
$potencia_activa1 = implode(",",$potencia_activa);
$energia_activa_importada1= implode(",",$energia_activa_importada);
$dem_tot_pot1=implode(",",$dem_tot_pot);
$ene_total_act1=implode(",",$ene_total_act);
// Convertir el array nuevamente a una cadena separada por comas
// Actualizar el dato en la base de datos
$sql2 = "UPDATE datos_contador SET C='$tension1', D='$corriente1', E =
'$potencia_activa1',
F='$energia_activa_importada1',F='$dem_tot_pot1',H='$ene_total_act1' WHERE A = '$A' ";
$actualizacion = $conn->query($sql2);

if ($actualizacion) {
    echo "Dato actualizado correctamente en la base de datos.";
} else {
    echo "Error al actualizar el dato en la base de datos: ". $conn->error;
}

// y otros valores que necesites

} else {
    echo "No se encontraron resultados.";
}
$conn->close();

```

## ANEXO III-CÓDIGO WEB HTML PHP PRUEBA.PHP

```

<!DOCTYPE html>
<html>
<body>
<head>
<link rel="stylesheet" href="styles3.css">
<h1>CONSUMO ENERGÉTICO</h1>
<form method="POST">
  <label for="fecha">Selecciona una fecha:</label>
  <input type="date" name="fecha">
  <br>
  <label for="hora">Selecciona una hora:</label>
  <select name="hora" id="hora">
    <?php
      for ($hora = 0; $hora < 24; $hora++) {
        printf('<option value="%02d:00">%02d:00</option>', $hora, $hora);
      }
    ?>
  </select>
  <button type="submit" name="submit">Mostrar Datos</button>
</form>
<br><br>
<?php
  if (isset($_POST['submit'])) {
    $fechaSeleccionada = $_POST['fecha'];
    $fechaFormateada = date("j/m/Y", strtotime($fechaSeleccionada));
    $timestamp = strtotime($fechaSeleccionada);
    $dia = date("z", $timestamp) + 1; // Sumamos 1 para que el primer día sea 1 y no 0
  }
?>

<?php
  if (isset($_POST['hora'])) {
    $horaSeleccionada = $_POST['hora'];
    $hora1 = explode(":", $horaSeleccionada);
    $numero = intval($hora1[0]);
  }
?>
<?php
date_default_timezone_set("Europe/Madrid");
$n=date("H");
$A = date('z') + 1; // Sumamos 1 ya que 'z' devuelve valores de 0 a 365

$servername = "";
$username = "";
$password = "";

```

```

$database=" ";

$conn = new mysqli($servername,$username,$password,$database);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
{
$sql = "SELECT E FROM datos_contador WHERE A='$A'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $cadena=$row['E']; // Convertir la cadena en un array
        $cadena =explode(",",$cadena);
        $C=$cadena[$n];
    }
} else {
    echo "0 results";
}
}
{
    $sql = "SELECT * FROM datos_contador WHERE A='$dia'";
    $result = $conn->query($sql);
    if ($result->num_rows > 0) {

        // output data of each row
        while($row = $result->fetch_assoc()) {
            $tension=explode(",",$row['C']);
            $corriente=explode(",",$row['D']);
            $pa=explode(",",$row['E']);
            $eai=explode(",",$row['F']);
            $dtp=explode(",",$row['G']);
            $eat=explode(",",$row['H']);
            $pa1=implode(",",$pa);

        }
    } else {
        echo "0 resultados";
    }
}
}
$conn->close();
?>
<div class="container">

```

```

    
    <div class="desc"><br>
    <?php echo $C. "Wh"; ?>
    </div>
    
    
</div>
<h2>Dia <?php echo $fechaFormateada." a las " . $horaSeleccionada?></h2>
<table>
<tr>
    <th>TENSIÓN</th>
    <td><?php echo $tension[$numero]."V"?></td>
</tr>
<tr>
    <th>CORRIENTE</th>
    <td><?php echo $corriente[$numero]."A"?></td>
</tr>
<tr>
    <th>POTENCIA ACTIVA</th>
    <td><?php echo $pa[$numero]."W"?></td>
</tr>
<tr>
    <th>ENERGIA ACTIVA IMPORTADA</th>
    <td><?php echo $eai[$numero]."kwh"?></td>
</tr>
<tr>
    <th>DEMANDA TOTAL DE POTENCIA</th>
    <td><?php echo $dtp[$numero]."W"?></td>
</tr>
<tr>
    <th>ENERGIA ACTIVA TOTAL</th>
    <td><?php echo $eat[$numero]."kwh"?></td>
</tr>
</table><br><br>

<title>Gráfico de Consumo Energético por Horas</title>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<canvas id="energyChart" width="30" height="10"></canvas>
<script>
const ctx = document.getElementById('energyChart').getContext('2d');
const hours = ['00:00', '01:00', '02:00', '03:00', '04:00', '05:00', '06:00', '07:00', '08:00', '09:00', '10:00', '11:00', '12:00', '13:00', '14:00', '15:00', '16:00', '17:00', '18:00', '19:00', '20:00', '21:00', '22:00', '23:00'];

```

```
const energyData = [<?php echo $pa1 ?>]; // Ejemplo de datos de consumo
const energyChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: hours,
    datasets: [{
      label: 'Consumo Energético',
      data: energyData,
      borderColor: 'blue',
      backgroundColor: 'rgba(0, 0, 255, 0.2)',
      fill: true
    }]
  },
  options: {
    scales: {
      x: {
        display: true,
        title: {
          display: true,
          text: 'Horas del día'
        }
      },
      y: {
        display: true,
        title: {
          display: true,
          text: 'Consumo Energético'
        }
      }
    }
  }
});
</script>
</head>
</body>
</html>
```

**ANEXO IV CSS STYLES3.CSS**

```
.container {
  display: flex;
  align-items: flex-start;
  justify-content: center;
  max-width: 300px;
  margin: auto;
}

.column {
  display: flex;
  flex-direction: column;
  align-items: left;
}

img {
  width: 200%;
  max-width: 100px;
  margin-bottom: 50px;
}

.right{
  width: 100%;
  max-width: 150px;
  margin-left: 300px;
  margin-bottom: 10px;
}

h1{text-align:center;color:Black; padding: 2px 4px;
  border-radius: 12px;
}

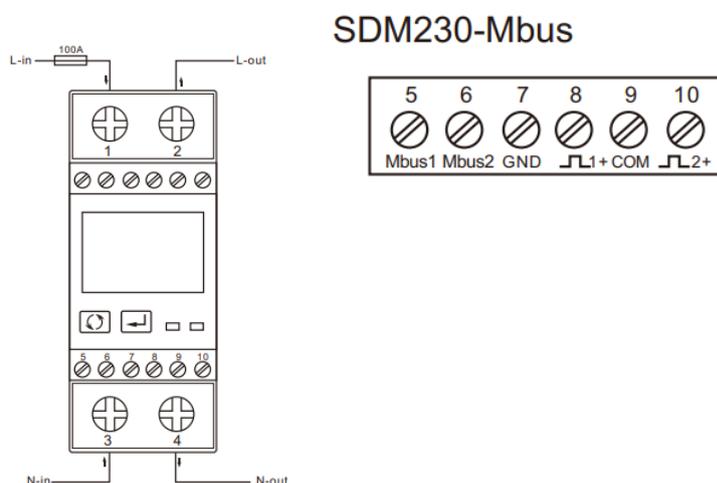
div.desc {
  padding: 5px;
  text-align: center;
```

```
}  
table, td, th {  
  border: 1px solid #ddd;  
  text-align: left;  
}  
table {  
  border-collapse: collapse;  
  width: 100%;  
}  
th, td {  
  padding: 10px;  
}  
h2{text-align:center;color:Black; padding: 2px 4px;  
  border-radius: 12px;  
}
```

# Especificaciones técnicas del hardware

## ANEXO V-CONTADOR EASTRON

Especificaciones	
Voltage AC (Un)	230V
Voltage Range	176~276V AC
Base Current (Ib/Iref)	10A
Max. Current (Imax)	100A
Mini Current (Imin)	0.5A
Starting current	0.4% of Ib/Iref
Power consumption	<2W/10VA
Frequency	50Hz(for MID version) 50Hz(default), 60Hz optional
AC voltage withstand	4KV for 1 minute
Impulse voltage withstand	6KV-1.2uS waveform
Over current withstand	30Imax for 0.01s
Pulse 1 output rate	configurable, default1000i/kWh
Pulse 2 output rate	non-configurable,1000i/kWh
Display Max	LCD with backlit
Max. Reading	999999.9kWh



## ANEXO VI – RASPBERRY PI

<b>Especificaciones de la Raspberry Pi 4 modelo B</b>	
<b>Sistema en un chip</b>	Broadcom BCM2711
<b>CPU</b>	Procesador de cuatro núcleos a 1,5 GHz con brazo Cortex-A72
<b>GPU</b>	VideoCore VI
<b>Memoria</b>	1/2/4GB LPDDR4 RAM
<b>Conectividad</b>	802.11ac Wi-Fi / Bluetooth 5.0, Gigabit Ethernet
<b>Vídeo y sonido</b>	2 x puertos micro-HDMI que admiten pantallas de 4K@60Hz a través de HDMI 2.0, puerto de pantalla MIPI DSI, puerto de cámara MIPI CSI, salida estéreo de 4 polos y puerto de vídeo compuesto.
<b>Puertos</b>	2 x USB 3.0, 2 x USB 2.0
<b>Alimentación</b>	5V/3A vía USB-C, 5V vía cabezal GPIO
<b>Expansión</b>	Cabezal GPIO de 40 pines

**ANEXO VII- MODULO DISPARADOR**

<b>Atributo</b>	<b>Valor</b>
Tipo de Montaje	Carril DIN
Tensión de Entrada AC	24V ac
Tensión de Entrada DC	24V dc
Conteo de Pines	11
Dimensiones	122 x 70 x 56mm
Temperatura Máxima de Funcionamiento	+70 °C