

**Prototipo de Monitoreo de Potencia Bidireccional usando IoT  
en Unidades de Vivienda Residenciales Monofásicas**

Omar Edgardo Rojas Ríos y Linda Gabriela Moreno Rodriguez

Trabajo de grado para optar al título de Ingeniera Electrónica

Director

Jaime Guillermo Barrero Perez

Magíster en Potencia Eléctrica

Co-Director

Jeison Arley Castillo Bohorquez

Magíster en Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica Y de Telecomunicaciones

Ingeniería Electrónica

Bucaramanga

2024

**Tabla de contenido**

**Introducción.....11**

**1. Objetivos..... 13**

1.1. Objetivo General..... 13

1.2. Objetivos Específicos..... 13

**2. Marco Teórico.....14**

2.1. Paneles fotovoltaicos..... 14

2.1.1. Sistema on-grid..... 16

2.1.2. Sistema off-grid..... 17

2.1.3. Sistema híbrido..... 18

2.2. Internet de las cosas (IoT)..... 19

2.2.1. Tecnologías para el Internet de las cosas..... 20

2.2.1.1. Medidor de potencia bidireccional..... 21

2.2.2. Protocolos de comunicación para IoT..... 21

2.2.2.2. Protocolo Modbus..... 22

2.2.3. Arquitectura IoT..... 22

2.2.3.1. Modelo IoT..... 23

2.3. Firebase..... 24

2.3.1. Privacidad y seguridad en Firebase..... 26

**3. Metodología..... 27**

3.1. Selección de hardware..... 27

3.1.1. Selección del Medidor bidireccional..... 27

3.1.2. Selección del Microcontrolador..... 30

3.2. Diseño del Esquemático de la Medición de Potencia..... 34

3.3. Diseño de la Arquitectura IoT..... 35

3.4. Implementación de la Comunicación..... 36

3.4.1. Configuración de la Conexión Wi-Fi..... 37

3.4.2. Implementación de Modbus para la comunicación con el medidor DDS-669..... 38

3.4.2.1. Registros del Medidor..... 39

3.4.3.2. Estructura de la trama..... 41

3.4.2.3. Protocolo de comprobación de trama..... 42

3.4.3. Configuración de Modbus en el microcontrolador..... 42

3.4.3.1. Esquema de conexión..... 43

3.4.3.2. Programación de Comunicación Modbus.....	44
3.4.4. Comunicación con Firebase desde el Microcontrolador.....	46
3.4.5. Comunicación con Firebase en la aplicación.....	49
3.4.6. Seguridad y privacidad en Firebase.....	53
3.5. Visualización de datos.....	54
3.5.1. Diseño de la Interfaz de Usuario.....	55
3.5.2. Aplicación de windows.....	55
3.5.3. Aplicación Móvil.....	58
3.6. Diseño de la PCB.....	59
<b>4. Resultados y Análisis.....</b>	<b>62</b>
4.1. Rendimiento.....	62
4.1.1. Linealidad.....	64
4.1.2. Desviación Estándar.....	65
4.3. Eficiencia.....	67
4.3.1. Eficacia.....	68
4.3.1.1 Eficacia del dispositivo con Firebase.....	68
4.3.1.2. Eficacia de la aplicación de Windows con Firebase.....	68
4.3.2. Efectividad.....	70
4.4. Pruebas prácticas Simulando un Sistema de Paneles Fotovoltaicos On-Grid.....	72
4.5 Prueba de bidireccionalidad del medidor.....	76
4.6. Presupuesto total del diseño e implementación.....	79
<b>5. Conclusiones.....</b>	<b>80</b>
<b>6. Recomendaciones.....</b>	<b>81</b>
<b>Referencias Bibliográficas.....</b>	<b>82</b>
<b>Anexos.....</b>	<b>86</b>

**Lista de Figuras**

Figura 1. Materiales que componen un panel fotovoltaico..... 15

Figura 2. Sistema Fotovoltaico On-grid..... 17

Figura 3. Sistema Fotovoltaico Off-grid..... 18

Figura 4. Sistema Fotovoltaico híbrido..... 19

Figura 5. Modelo IoT tomado de referencia..... 24

Figura 6. Medidores Bidireccionales Seleccionados..... 29

Figura 7. Microcontrolador ESP32 DevKit V1..... 34

Figura 8. Esquemático de la medición de potencia para el calcular el ahorro de energía..... 35

Figura 9. Diseño Arquitectura IoT..... 36

Figura 10. Diagrama de flujo para la comunicación Wi-Fi..... 38

Figura 11. Esquema de conexión..... 43

Figura 12. Diagrama de flujo para la comunicación de Modbus con la ESP32..... 45

Figura 13. Diagrama de flujo para la comunicación con Firebase desde la ESP32..... 47

Figura 14. Diagrama de flujo de la comunicación con firebase hacia la aplicación..... 51

Figura 15. Creando nueva clave privada..... 52

Figura 16. Estableciendo Conexión con Firebase, URL del proyecto..... 53

Figura 17. Aplicación Windows..... 57

Figura 18. Interfaz de la Aplicación Móvil..... 59

Figura 19. Esquemático..... 60

Figura 20. Diseño de la PCB..... 61

Figura 21. Ensamble final de la PCB y el dispositivo..... 62

Figura 22. Conexión pruebas de rendimiento y bidireccionalidad..... 65

Figura 23. Linealidad de voltaje (a), corriente (b) y potencia (c)..... 66

Figura 24. Desviación estándar de voltaje (a), corriente (b) y potencia (c)..... 68

Figura 25. Tiempo de ejecución de la aplicación..... 71

Figura 26. Esquemático de conexión del dispositivo con el Lucas-Nülle..... 74

Figura 27. Sistema Lucas-Nülle, Simulador de sistema On-grid..... 75

Figura 28. Datos Programados en el Simulador de Panel solar..... 76

Figura 29. Comportamiento del dinero ahorrado a lo largo del día..... 77

Figura 30. Simulación del Comportamiento de un Panel Fotovoltaico en un día..... 78

Figura 31. Comprobación de la medición Bidireccional..... 79

**Lista de tablas**

Tabla 1. Características de los medidores..... 28

Tabla 2. Características de los microcontroladores y microprocesadores..... 30

Tabla 3. Evaluación de los microcontroladores y microprocesadores..... 32

Tabla 4. Características de Comunicación Modbus entre el dispositivo y el microcontrolador... 39

Tabla 5. Direcciones Almacenamiento de Variables del DDS-669..... 40

Tabla 6. Estructura de la Trama Enviada..... 41

Tabla 7. Estructura del Dato Enviado..... 42

Tabla 8. Estructura de Checksum de la trama..... 42

Tabla 9. Error promedio del análisis de linealidad..... 65

Tabla 10. Desviación Estándar..... 67

Tabla 11. Rapidez por intervalos de horas..... 69

Tabla 12. Detalles de los conceptos asociados al costo del dispositivo y el respectivo precio..... 79

**Lista de Anexos**

**Anexo A.** Manual de conexión y programación del medidor DDS-669 con el ESP32 ..... 87

**Anexo B.** Principios de diseño de la interfaz de usuario..... 87

## Resumen

**Título:** Prototipo de Monitoreo de Potencia Bidireccional usando IoT en Unidades de Vivienda Residenciales Monofásicas \*

**Autor:** Omar Edgardo Rojas Ríos, Linda Gabriela Moreno Rodriguez. \*\*

**Palabras clave:** Ahorro de energía, Firebase, IoT, Microcontrolador, Paneles fotovoltaicos.

### Descripción:

Los paneles fotovoltaicos se han convertido en los últimos años en una solución confiable y segura para la generación de energías limpias, las cuales toman cada vez más importancia debido a la problemática global que existe como el agotamiento de recursos naturales y el cambio climático. En Colombia se ha experimentado un crecimiento importante en la implementación de estos sistemas y mediante políticas públicas se busca seguir impulsando este tipo de proyectos, por lo que se hace importante medir el impacto que tiene al sistema interconectado nacional, así como a la economía familiar de quienes adquieren estos sistemas. Este proyecto busca ser parte de este impulso, ofreciendo a los usuarios facilidades de acceder al sistema de paneles fotovoltaicos, una visualización del ahorro de energía mediante un monitoreo bidireccional de potencia usando IoT. Para esto, desarrollamos un dispositivo IoT basado en el medidor DDS-669 TAXNELE y el microcontrolador ESP32, el cual cuenta con una conexión a internet que permite el envío de información en tiempo real. Dicho dispositivo se conecta al servicio de la nube de Firebase y cuenta con una interfaz de visualización de datos basada en Python. Este conjunto representa un mecanismo económico y eficiente de monitoreo de variables de consumo y generación de energía, para una unidad residencial con paneles solares.

---

\*Trabajo de Grado

\*\*Facultad de Ingenierías Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.  
Director: Jaime Guillermo Barrero Pérez. Magíster en Potencia Eléctrica, Co-director: Jeison Arley Castillo Bohorquez. Magíster en Ingeniería Electrónica

### Abstract

**Title:** Bidirectional Power Monitoring Prototype using IoT in Monophasic Residential Housing Units \*

**Author:** Omar Edgardo Rojas Ríos, Linda Gabriela Moreno Rodriguez. \*\*

**Key words:** Energy saving, Firebase, IoT, Microcontroller, Photovoltaic panels.

#### Description:

In recent years, photovoltaic panels have become a reliable and safe solution for generating clean energy, which is becoming increasingly important due to global issues such as the depletion of natural resources and climate change. Colombia has experienced a significant growth in the implementation of these systems, and through public policies seeks to continue promoting this type of projects, so it is important to measure the impact it has on the national interconnected system, as well as the household economy of those who acquire these systems.

Our project aims to be part of this momentum, offering users of photovoltaic systems a visualization of energy savings through bidirectional power monitoring using IoT. To do this, we have developed an IoT device based on the meter DDS-669 TAXNELE and the ESP32 microcontroller, which has an Internet connection that allows it to send information in real time.

This device connects to the Firebase cloud service and has a data visualization interface based on Python. This kit represents an economical and efficient mechanism for monitoring energy consumption and production variables for a residential unit with solar panels.

---

\*Degree Project

\*\*Faculty of Physico-Mechanical Engineering. School of Electrica, Electronics and Telecommunications Engineering. Director: Jaime Guillermo Barrero Pérez. Master of Science in Electrical Power Engineering, Co-director: Jeison Arley Castillo Bohorquez. Master of Science in Electronic Engineering

## Introducción

En la actualidad, la conciencia ambiental y la eficiencia energética se vuelven cada vez más importantes debido a la problemática global del agotamiento de los recursos naturales y el cambio climático. La fuente principal de energía es obtenida de combustibles fósiles, generando una gran emisión de gases de efecto invernadero que aumentan estas problemáticas. Como una solución a esta situación, se busca reducir la dependencia de los combustibles fósiles con otras alternativas de generación de energía, lo que conocemos hoy como energías renovables.

En este contexto, los paneles fotovoltaicos se han convertido en una solución fundamental y sostenible donde se ha logrado evidenciar un aumento de un 24% en 2022, lo que la convierte en la fuente de energía de más rápido crecimiento por 18.º año consecutivo, según un estudio de Ember, un grupo de expertos en cambio climático (Rey, 2023)

En Colombia, se ha podido evidenciar este crecimiento importante en los últimos años, impulsado por la transición energética y la búsqueda de alternativas a las fuentes fósiles. Según la Unidad de Planeación Minero Energética (UPME), el país cuenta con un gran potencial solar, especialmente en las regiones de La Guajira, Cesar, Magdalena y Santander (Person,2023).

En la región de Santander, se han empezado a impulsar este tipo de proyecto en un afán de incursionar en este tipo de energía, “de acuerdo con información de Sinergox (Insumos históricos de Agpe del Administrador del mercado de energía mayorista), Santander ocupa el segundo lugar como el departamento con más Agpe (proyectos de autogeneración de energía a pequeña escala) en el país, con una participación del 11,1 % y 3.300 usuarios.” (Alguero, 2023).

Sin embargo, surge la pregunta: *¿cómo podemos medir y maximizar el verdadero impacto de esta energía limpia en nuestra comunidad?* Aquí es donde entra en juego este proyecto, un dispositivo que combina el Internet de las Cosas (IoT) con el monitoreo de potencia bidireccional

en unidades de viviendas residenciales monofásicas. Nuestro objetivo es cuantificar el impacto de ahorro de energía logrado a través de la instalación de paneles solares y, lo más importante, brindar a los usuarios una herramienta económica para tomar decisiones informadas sobre su consumo energético.

El desarrollo de este dispositivo está basado en el medidor DDS-669 TAXNELE y en un microcontrolador ESP32, el cual cuenta con una conexión a internet que permite el envío de información en tiempo real. Dicho dispositivo se conecta a la nube del servicio Firebase y cuenta con una interfaz de visualización de datos basada en Python. Este conjunto representa un mecanismo económico y eficiente de monitoreo de variables de consumo y generación de energía, para una unidad residencial monofásica con paneles solares

Con esta idea, se busca contribuir en la transformación hacia una sociedad más sostenible y consciente en la región de Santander y más allá. Dando herramientas para cambiar la forma en que las personas interactúan con la energía renovable, fomentando así una mentalidad de consumo responsable y sentando las bases para un futuro energético más limpio y eficiente.

## 1. Objetivos

### 1.1. Objetivo General

Diseñar un prototipo de monitoreo de potencia bidireccional utilizando el Internet de las Cosas (IoT) en unidades de viviendas residenciales monofásicas.

### 1.2. Objetivos Específicos

- Seleccionar los elementos del hardware compatible para la medición bidireccional de potencia en un sistema de unidad residencial (AC), así como para medir potencia eléctrica en sistemas fotovoltaicos (DC), con comunicación alámbrica o inalámbrica que permita extraer los datos.
- Implementar un sistema basado en microcontrolador con capacidad de comunicación IoT con el fin de almacenar y extraer los datos medidos.
- Desarrollar una interfaz de usuario accesible que permita a los usuarios visualizar y analizar los datos de consumo y generación de energía de manera clara y comprensible con comunicación segura.
- Evaluar el rendimiento y la eficiencia del sistema de monitoreo, realizando pruebas y análisis para verificar la precisión de las mediciones.

## 2. Marco Teórico

### 2.1. Paneles fotovoltaicos

Los paneles fotovoltaicos o solares son dispositivos diseñados para transformar la luz solar en energía eléctrica, siendo así una fuente importante de energía renovable. Para este proyecto, se debe comprender el principio de su funcionamiento para un buen desarrollo del sistema de monitoreo. Esto permite tener una visión más amplia para los diferentes sistemas de paneles solares, que incluyen los sistemas on-grid (el enfoque de este proyecto), sistemas off-grid y sistemas híbridos.

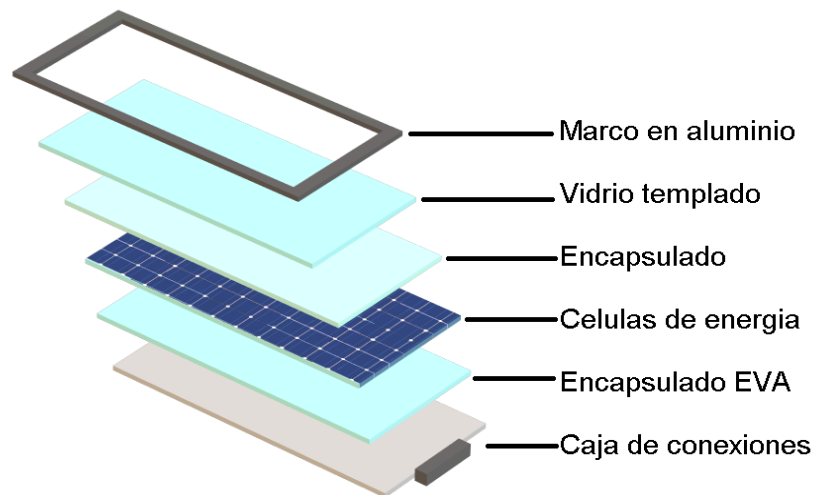
La conversión de luz solar en energía por medio de los paneles, se produce debido a que la luz se compone de fotones, los cuales son portadores de radiación electromagnética, y los materiales por los que están compuestos los paneles solares se encargan de “atrapar” estos fotones para aprovechar la energía que llevan. Estos materiales son semiconductores, con un comportamiento dual, lo que significa, que dependiendo de su entorno pueden ser conductores o aislantes. El elemento principal, más conocido y utilizado tanto en la electrónica como en los paneles fotovoltaicos, es el silicio. Una capa de silicio es dopado con otro tipos de elementos para lograr configuraciones tipo N, que permiten tener un electrón libre, y tipo P, que aceptan recibir un electrón libre; generando así el flujo de electrones, la energía eléctrica en DC.

La forma en que se componen los materiales se pueden ver en la Figura 1, estos son: El marco en aluminio, que es resistente a la corrosión y protege al sistema. Luego, el vidrio templado protege las células solares contra factores externos y garantiza baja reflectancia y alta transmitancia. Sigue el encapsulante EVA, que tiene buena transmisión de radiación, este crea un película sellante y aislante alrededor de las células de silicio. Después, la parte más importante, que son las células solares que convierten la luz solar en energía eléctrica DC. Sigue una

cubierta posterior, que protege el módulo de las radiaciones UV y proporciona el aislamiento eléctrico del sistema. Finalmente, se tiene una caja de conexión que es el punto central donde todos los componentes se interconectan. (INELDEC, 2021)

### Figura 1

*Materiales que componen un panel fotovoltaico*



*Nota.* Figura realizada con la información tomada de INELDEC (2021).

<https://ineldec.com/de-que-estan-hechos-los-paneles-solares-fotovoltaicos/>

### ***2.1.1. Sistema on-grid***

El sistema on-grid, el sistema que se analizó en este proyecto, es un tipo de instalación fotovoltaica que se conecta directamente a la red eléctrica y no tiene conexión a un almacenamiento de baterías, esto hace que sea más económica su instalación. Al tener este tipo de conexión permite la utilización tanto de la energía solar como de la energía de la red. Incluso si llega a existir un excedente de energía producido por el panel solar, este puede ser devuelto a la red.

Para inyectar la energía solar a la red eléctrica, se emplea un inversor de conexión a la red, que convierte la corriente continua (DC) producida por los paneles fotovoltaicos en corriente alterna (AC). Por esta razón, se utiliza un medidor bidireccional de potencia en este proyecto. Debido a este diseño del inversor y medidor bidireccional estos sistemas no proporcionan electricidad durante una interrupción de la red, ya que el inversor se desconecta automáticamente como medida de seguridad. Este aspecto se considera su principal desventaja.

La empresa colombiana de paneles fotovoltaicos Sun Supply (2021) y la empresa Emergente (2020) explican de forma breve tres tipos de consumo energético (ahorro, balance y excedentes) con el sistema on-grid y las regulaciones para la facturación acá en Colombia, las cuales se muestran a continuación:

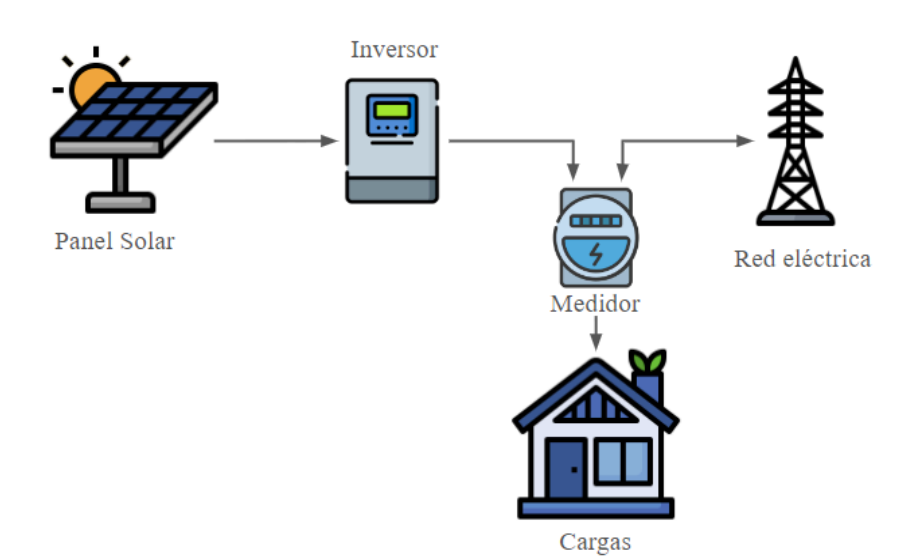
- **Ahorro y balance:** Si el consumo energético es mayor o la misma energía que genera el sistema solar, se paga la energía tomada directamente de la red a costo de comercialización (9% menos por cada kWh).
- **Excedentes:** Si el consumo energético es menor a la energía que genera el sistema solar, se puede entregar la energía extra que genera al operador de red, por lo cual, puede tomar ganancias por venta de esta energía a la red (esta será comprada a un

valor del 40% menos de la comercialización).

Estos consumos están regulados por la resolución 174 de la CREG del 2021 (CREG, 2021), la cual también establece un límite de energía para esto. Además al ser conectado a la red eléctrica, el sistema también debe contar con la certificación RETIE de alguna entidad acreditada.

**Figura 2**

*Sistema Fotovoltaico On-grid*



### 2.1.2. Sistema off-grid

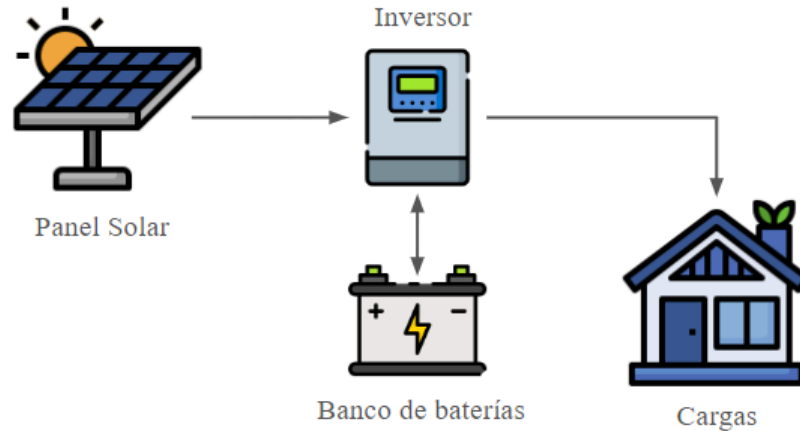
Un sistema off-grid suele conocerse como un sistema de respaldo. Su instalación puede requerir de baterías de almacenamiento o bien, de una fuente de energía de respaldo como un generador. Esta configuración es independiente de la red eléctrica, una de sus mayores ventajas ya que permite que se pueda usar en lugares donde no hay alcance de redes eléctricas.

Este sistema necesita ser capaz de generar suficiente energía para cubrir toda la demanda, también necesitan inversores pero más especializados para la conversión de la energía de las

baterías. Estos aspectos hacen que sea mucho más costoso, complejo y menos flexible. En la Figura 3 se puede evidenciar el esquema representativo de un off-grid.

**Figura 3**

*Sistema Fotovoltaico Off-grid*



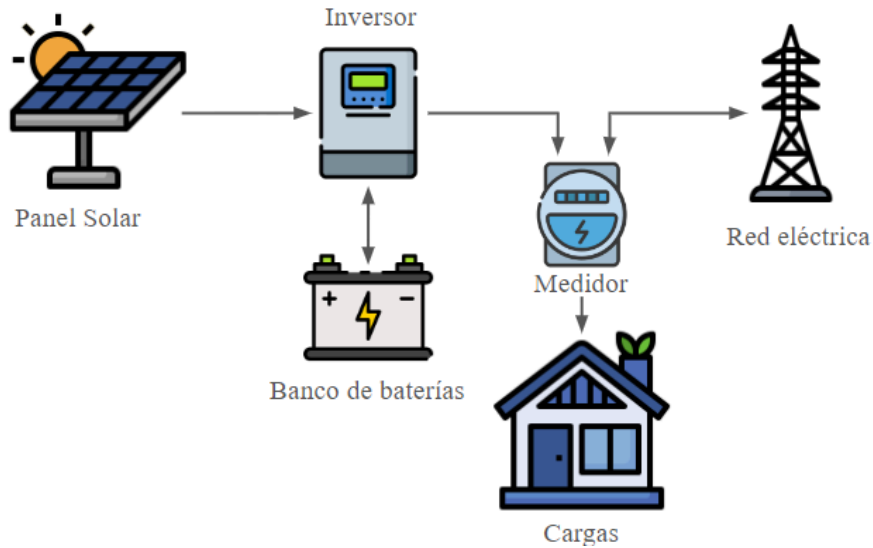
**2.1.3. Sistema híbrido**

El sistema híbrido, es una instalación que combina la conexión del sistema on-grid y la conexión del sistema off-grid, lo que permite que funcione de forma autónoma y utilice la energía solar en todo momento, sin importar si hay conexión a la red pues toma la energía almacenada en las baterías, siendo así un sistema muy flexible.

Aunque se considera el mejor sistema, la implementación de este sistema es la más costosa y compleja, por lo cual suelen ser usados en aplicaciones de alto consumo, como en infraestructuras de telecomunicaciones, de procesos industriales y ganaderos.

**Figura 4**

*Sistema Fotovoltaico híbrido*



## 2.2. Internet de las cosas (IoT)

Según la definición del IERC (2014), establece que el Internet de las Cosas (IoT) es "Una infraestructura dinámica de red global con capacidades de autoconfiguración basada en protocolos de comunicación estándar e interoperables, donde los 'objetos' físicos y virtuales tienen identidades, atributos físicos y personalidades virtuales, utilizan interfaces inteligentes y se integran de manera fluida en la red de información".

Se considera una revolución en el mundo moderno. Permite la comunicación entre dispositivos sin intervención humana, los cuales recopilan y comparten datos a través de medidores, especialmente en tiempo real. Esto hace posible analizar los datos en la nube o localmente e integrarlo con inteligencia artificial que mejore la eficiencia, ya sea de procesos industriales, salud digital, ciudades inteligentes, comercio, sistemas inteligentes para los hogares y demás procesos.

Esto ha transformado la manera en que interactuamos con nuestro entorno, permitiendo el control y monitoreo en cualquier momento de las cosas, mejorando así la calidad de vida de nosotros los usuarios y la eficiencia de los procesos.

### ***2.2.1. Tecnologías para el Internet de las cosas***

El artículo Laghari et al.(2022) ilustra en cuatro pilares las tecnológicas básicas que conforman una arquitectura IoT y se presentan a continuación:

- **Hardware del Sistema:** Se refiere a un componente físico que se pueda volver “inteligente” ya sea agregando un chip, medidor, microprocesador, etc.
- **Dispositivo de Software:** Gestiona la comunicación de datos y la nube, facilita la integración de dispositivos, la recopilación de datos del sistema y el análisis de datos en la nube, como por ejemplo, IBM Watson IoT Platform, Microsoft Azure IoT, Google Cloud IoT, etc.
- **Ruta de Comunicación:** Se necesita para la transmisión de datos entre los dispositivos, utilizando tecnologías como Wi-Fi, Bluetooth, etc.
- **Plataforma:** Interfaz para poder mostrar los datos procesados al usuario, ya sea con o sin interacción directa, como aplicaciones móviles, ejecutables de visualización, páginas web, dashboard especializadas, etc.

Entendiendo esto, a continuación se va a describir una de las tecnologías de hardware que se utilizó en este proyecto, el medidor de potencia bidireccional para comprender porqué se eligió con esta funcionalidad específicamente:

**2.2.1.1. Medidor de potencia bidireccional.** Es un dispositivo que mide la cantidad de energía eléctrica que fluye en ambas direcciones, para este caso: desde la red eléctrica hacia una residencia y desde la residencia hacia la red eléctrica. Poder diferenciar y almacenar la energía en ambas direcciones es fundamental cuando se quiere monitorear sistemas de paneles fotovoltaicos On-Grid, también evitando así una instalación doble de uno unidireccional.

Existen dos tipos de medidores, los electromecánicos que son analógicos y los electrónicos que son digitales. Los primeros ya están prácticamente en desuso, funcionan mediante un disco que registra el flujo de corriente, y la lectura de estos registros se debe hacer manualmente. Por otro lado, para este proyecto se utilizará un medidor digital, el cual opera midiendo la corriente y el voltaje, procesando las señales y entregando datos digitales que se muestran en la pantalla principal. Además, posee capacidades de comunicación que permiten la transmisión de estos datos.

### ***2.2.2. Protocolos de comunicación para IoT***

Los protocolos son fundamentales para la comunicación entre dispositivos, de forma clara, los protocolos de comunicación son estándares que establecen cómo se transmite la información entre diferentes capas o sistemas. Actúan como un tipo de contrato o acuerdo que asegura un enfoque estructurado para la transferencia de datos (Kithion, 2016).

La Capa de Enlace de Datos conecta la placa de microcontrolador a internet, incluye protocolos como Wi-Fi, Ethernet, NFC, 2G/3G/LTE. La Capa de Red necesita una IP única para unirse a internet, utiliza protocolos como IPV4, IPV6 y 6LoWPAN. La Capa de Transporte maneja la transferencia de datos con TCP y UDP. La Capa de Aplicación incluye la interacción del usuario con protocolos como HTTP, CoAP, DDS y MQTT.

**2.2.2.2. Protocolo Modbus.** El protocolo Modbus es un estándar de comunicación industrial ampliamente utilizado que facilita la transferencia de datos entre dispositivos electrónicos. Es uno de los protocolos más antiguos, diseñado originalmente por Modicon (ahora Schneider Electric) en 1979. Modbus ha evolucionado para convertirse en un protocolo de comunicación de código abierto que juega un papel vital en el ámbito de la automatización y el control. (Centro de formación técnica para la industria, n.d.)

Modbus presenta diversas variantes diseñadas para satisfacer las necesidades específicas de los sistemas de automatización industrial. Estas versiones se adaptan a diferentes medios de comunicación, como puertos serie y Ethernet.

- **Modbus TCP:** Se implementa para entornos Ethernet, su medio principal es la dirección IP. La comunicación se da por bloques de datos y suele ser usado en redes industriales.
- **Modbus RTU:** Su comunicación es serial y la transmisión de datos se da por codificación binaria, es muy usado en sistemas de control, industria y dispositivos que requieren comunicación en serie eficiente.
- **Modbus ASCII:** Su comunicación es serial y la transmisión de datos se da por paquetes en caracteres ASCII. Es el más antiguo y ya está casi en desuso.

### ***2.2.3. Arquitectura IoT***

La arquitectura del sistema IoT es fundamental para garantizar que todos los elementos en la red estén conectados entre sí, en general, se busca que la adaptabilidad, escalabilidad y flexibilidad sea primordial en el diseño IoT independientemente del entorno donde se necesite. Dos factores importantes del sistema IoT, son las tecnologías y los protocolos de comunicación a usar en el modelo de referencia escogido, para plantear la arquitectura IoT de este proyecto.

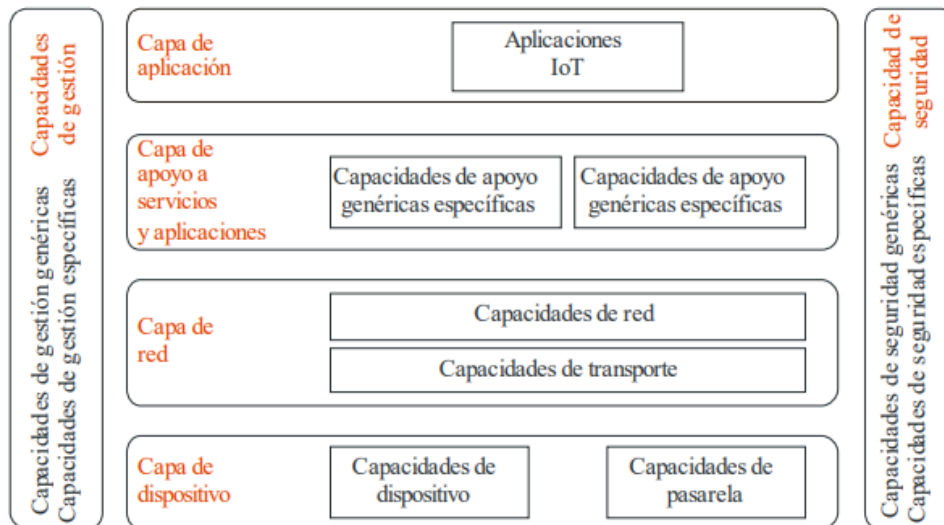
**2.2.3.1. Modelo IoT.** El modelo de referencia que tomamos para la arquitectura de este proyecto, es el propuesto por la Unión Internacional de Telecomunicaciones (UIT) (2012), el cual podemos observar en la Figura 5.

Este modelo se compone de las siguientes cuatro capas:

- **Capa de aplicación:** Contiene las aplicaciones IoT.
- **Capa de apoyo a servicios y aplicaciones:** Consiste en el procesamiento o almacenamiento de datos.
- **Capa de red:** Funciones de control de la conectividad en red y transporte de información
- **Capa de dispositivo:** Dispositivos conectados mediante tecnologías alámbricas o inalámbricas como el Wi-Fi.

**Figura 5**

*Modelo IoT tomado de referencia*



*Nota.* Tomado de UIT, 2012.

Cada una de estas capas cuenta con capacidades de gestión y seguridad. Con este enfoque tenemos una visión estructurada de las tecnologías descritas anteriormente.

- **Capacidades de gestión:** Se refiere a la gestión de fallos, de la configuración, de la contabilidad, del rendimiento, del tráfico, topologías de red y de la seguridad. Un ejemplo de esto es la actualización del software, activación y desactivación de dispositivos remotos,
- **Capacidades de seguridad:** Se refiere a la autorización, autenticación, confidencialidad de datos, protección de privacidad y más, según los requisitos específicos de cada aplicación. (UIT, 2012)

### 2.3. Firebase

Firebase es una plataforma de desarrollo de aplicaciones y páginas web creada por Google, la cual proporciona las herramientas e infraestructura necesarias para construir aplicaciones con la capacidad de mejorar y crecer de una forma más sencilla y rápida. Brinda servicios comunes que se pueden necesitar y en su mayoría son gratuitos, como una base de datos backend, autenticación segura, mensajería y otras más.

A continuación, se dará una introducción a las principales tecnologías (Moroney, 2017):

- **Firestore Authentication:** Sirve para la identificación de usuarios, facilitando el inicio de sesión mediante contraseñas, correos electrónicos, Facebook y más. Proporciona un ID y un token web único a cada usuario de forma segura e integrándose a Firebase Realtime Database para controlar el acceso a datos.

- **Firestore Realtime Database:** Es una base de datos en la nube que facilita la sincronización entre dispositivos. Guarda todos los datos en formato JSON Tree y se caracteriza por actualizar de forma rápida y en tiempo real el cambio de la información en la interfaz del usuario.
- **Cloud Storage for Firebase:** Facilita el almacenamiento de archivos grandes, como imágenes, audio, video, etc., con una API respaldada por Google Cloud Storage. Simplifica la carga/descarga de archivos y maneja fallas de comunicación. Además, administra manualmente la cantidad de espacio que utiliza según sea necesario.
- **Cloud Firestore:** A diferencia del cloud storage, almacena datos NoSQL y tiene sincronización en tiempo real. Una de sus ventajas es la compatibilidad y flexibilidad con diferentes plataformas.
- **Firebase Hosting:** Proporciona espacio de alojamiento para implementar sitios web asociados a aplicaciones. Permite archivos como HTML, CSS y JavaScript.
- **Cloud Functions:** Es una herramienta que permite ejecutar código en el backend de una aplicación Firebase, como cambios en la base de datos en tiempo real, autenticación u otros elementos.

Cabe resaltar que, aunque no se entrará en detalle de estas, tiene grandes herramientas para el análisis del rendimiento de las aplicaciones, para verificar funcionamientos de aplicaciones en diferentes dispositivos y demás.

### ***2.3.1. Privacidad y seguridad en Firebase***

Firestore cumple con el Reglamento General de Protección de Datos (GDPR) de la Unión Europea y también con la Ley de Privacidad del Consumidor de California (CCPA). Por lo cual, Google se compromete a salvaguardar la confidencialidad y seguridad de los datos, y tampoco permite la transmisión de datos personales. De esta forma, los desarrolladores tienen más facilidad para el control de los datos, pero también tienen la responsabilidad de cumplir con los reglamentos de privacidad.

Todos los servicios de Firestore completaron correctamente los procesos de evaluación de ISO 27001 y SOC 1, SOC 2 y SOC 3. Encripta datos en tránsito con HTTPS y datos de clientes aislados de manera lógica, respaldando así la seguridad y confiabilidad. Además, permite configurar reglas de seguridad que se pueden agregar según las necesidades de la aplicación, como acceso a datos específicos, regulaciones, prevención de amenazas, control de normativas y demás, lo cual permite tener una infraestructura segura para el usuario. (*Privacidad Y Seguridad En Firestore, 2023*).

### 3. Metodología

#### 3.1. Selección de hardware

Para este proyecto, se llevó a cabo la selección de hardware revisando de forma técnica la efectividad y versatilidad de los criterios necesarios para cumplir con los objetivos del plan. Se analizaron diversas opciones, tanto de medidores de potencia bidireccionales con cualidades específicas para residencias monofásicas y paneles solares a bajo precio, como también las diversas opciones de microcontroladores, teniendo en cuenta criterios esenciales para la compatibilidad, precio y opciones de desarrollo.

##### *3.1.1. Selección del Medidor bidireccional*

Para la selección del medidor de potencia, se analizaron diversos dispositivos y finalmente elegimos estas dos opciones: el medidor DDS-669 TAXNELE y el medidor DDS238-2 TOMZ, que son muy similares pero de empresas diferentes. Debido a que son medidores económicos, existe muy poca información sobre ellos, por lo cual se propuso hacer diferentes pruebas entre ellos para elegir el mejor.

Además del costo de los medidores, se tuvo en cuenta que tuvieran medición bidireccional monofásica para que permita la medición tanto de la energía consumida como de la energía generada en residencias. La conectividad a MODBUS por medio del módulo RS485 serial permite la comunicación con el microprocesador. También se consideraron los parámetros medidos para tener una mejor calidad de los datos recopilados y el análisis para el ahorro de energía, así como los criterios que se muestran en la siguiente Tabla 1:

**Tabla 1**

*Características de los medidores*

<b>Características del Medidor</b>	<b>DDS-669 TAXNELE -Shopee</b>	<b>DDS238-2 -Aliexpress</b>
Parámetros Medidos	Voltaje, Corriente, Potencia Activa, Factor de Potencia, Frecuencia, Energía Activa Total, Energía Activa Positiva, Energía Activa Negativa(inversa), Energía Activa Temporal	Voltaje, Corriente, Potencia Activa, Factor de Potencia, Frecuencia, Energía Activa Total, Energía Activa Positiva, Energía Activa Negativa(inversa), Energía Activa Temporal
Voltaje de Referencia	220/230V	220/230V
Corriente de Referencia	5(60A)	10A
Frecuencia Nominal	50-60Hz	50-60Hz
Pulso Constante	1600imp/kWh	1600imp/kWh
Consumo de Energía	≤2W, 10VA	≤2W, 10VA
Temperatura de Operación	-20 ~ 55°	-25 ~ 70°
Comunicación	RS485 (MODBUS)	RS485 (MODBUS o DL/T645-2007)
Tasa Baud de Comunicación	1200/2400/4800/9600 bps	4800 bps
Precio	97.850 COP*	110.969 COP*

*Nota.* \*Los precios son tomados de las *Shopee* y *Aliexpress*.(Noviembre de 2023)

([https://shopee.com.co/product/593962443/17394226554?d\\_id=ba8b6&utm\\_content=3NBQKY](https://shopee.com.co/product/593962443/17394226554?d_id=ba8b6&utm_content=3NBQKY)

[WHArbAF5Xvkkhv2e7YSpY7](https://es.aliexpress.com/i/32922071523.html)) (<https://es.aliexpress.com/i/32922071523.html>)

A continuación, en la Figura 6 se muestra una imagen de ambos medidores:

**Figura 6**

*Medidores Bidireccionales Seleccionados*



*Nota.* Imagen de los medidores DDS-669 TAXNELE y DDS238-2, respectivamente. Tomado de *Shopee* y *Aliexpress*, Noviembre de 2023.

([https://shopee.com.co/product/593962443/17394226554?d\\_id=a8b6&utm\\_content=3NBQKYWHArbAF5Xvkkhv2e7YSpY7](https://shopee.com.co/product/593962443/17394226554?d_id=a8b6&utm_content=3NBQKYWHArbAF5Xvkkhv2e7YSpY7)), (<https://es.aliexpress.com/i/32922071523.html#nav-specification>)

**3.1.2. Selección del Microcontrolador**

Para la selección del microcontrolador, se llevó a cabo una comparación de las mejores opciones del mercado con los criterios mostrados en la Tabla 2:

**Tabla 2**

*Características de los microcontroladores y microprocesadores*

<b>Criterio</b>	<b>Arduino Uno Rev3</b>	<b>ESP32 DevKit V1</b>	<b>Raspberry Pi Pico</b>	<b>NRF52840</b>
Conexión WiFi	No	Sí	No	Sí
Protocolo WiFi	Depende del módulo adicional	802.11 b/g/n	Depende del módulo adicional	802.11 b/g/n
Protocolo Bluetooth	No	Sí	No	Sí
Conexión Modbus	Sí, con módulos externos	Sí, con módulos externos	Sí, con módulos externos	Sí
Módulo externo para Modbus	Sí	Sí	Sí	No aplica
Núcleos de Procesamiento	Single-core	Dual-core	Single-core	Single-core
Velocidad de Reloj	16 MHz	160 Mhz hasta 240 MHz	133 MHz	64 MHz
Precio	175.000 COP*	27.000 COP*	49.990 COP*	66.900 COP*
Comodidad de desarrollo	Amplia comunidad de desarrollo	Amplia y activa comunidad de desarrollo	Comunidad en crecimiento	Comunidad en crecimiento

*Nota.* \*Los precios son tomados de *Mercado Libre*, Noviembre de 2023

*(<https://articulo.mercadolibre.com.co/>)*

Cada microcontrolador considerado fue evaluado y puntuado según una serie de criterios predefinidos los cuales tienen una puntuación en una escala de 1 a 3. Además, cada criterio fue ponderado con un máximo según su relevancia en el proyecto.

- **Módulo externo WiFi [0-2]:** Para IoT, es crucial la transmisión inalámbrica de datos. El mayor puntaje es para los que no requieren módulo externo.
- **Protocolo Bluetooth [0-1]:** Importante pero no fundamental para este proyecto. El mayor puntaje es para los que no requieren módulo externo.
- **Conexión Modbus [0-2]:** Fundamental la compatibilidad para la comunicación con los medidores de potencia. El mayor puntaje es para los que sí son compatibles.
- **Módulo externo para Modbus [0-1]:** Adaptabilidad con módulos externos que sí son compatibles. El mayor puntaje es para los que no necesitan este módulo.
- **Núcleos de Procesamiento [0-2]:** Es esencial la capacidad de procesamiento para ejecutar algoritmos y manejar datos en tiempo real. El mayor puntaje es para el mejor.
- **Precio [0-3]:** El presupuesto es fundamental en este proyecto. El mayor puntaje son las opciones más económicas.
- **Comodidad de desarrollo [0-2]:** Importante para la resolución de problemas y facilitar el desarrollo. El mayor puntaje es para opciones con amplio soporte y herramientas.

En la siguiente Tabla 3 se muestra la evaluación y selección del microcontrolador con cada criterio descrito anteriormente:

**Tabla 3**

*Evaluación de los microcontroladores y microprocesadores*

<b>Criterio</b>	<b>Arduino Uno Rev3</b>	<b>ESP32 DevKit V1</b>	<b>Raspberry Pi Pico</b>	<b>NRF52840</b>
Módulo externo WiFi	0	2	0	2
Protocolo Bluetooth	0	1	0	1
Conexión Modbus	2	2	2	2
Módulo externo para Modbus	0	0	0	1
Procesamiento	1	2	1	1
Precio	0	3	2	1
Comodidad de desarrollo	2	2	1	1
<b>TOTAL</b>	<b>4</b>	<b>10</b>	<b>6</b>	<b>7</b>

Al evaluar los microcontroladores disponibles, la mejor opción que se alinea con los criterios de selección de este proyecto y proporciona una base sólida para la implementación del medidor de potencia bidireccional escogido es la ESP32 mostrada en la Figura 7. Una de las razones es que es un dispositivo que garantiza la conectividad estable tanto en entornos alámbricos como inalámbricos, ya que tiene módulo de conexión WiFi integrado. Además, puede integrar el protocolo Modbus mediante un módulo externo Conversor RS485 a Serial, el cual es bastante económico y se encuentra fácilmente en el mercado.

Una de las grandes ventajas de la ESP32 respecto al resto de microprocesadores es que tiene dos núcleos que permiten realizar tareas en paralelo, mejorando la capacidad de respuesta. Además, la velocidad del reloj es más alta, lo que permite realizar las operaciones y ofrece una mayor capacidad de adaptación a escenarios dinámicos, lo cual es necesario para la comunicación con el medidor de potencia bidireccional.

Es importante destacar la ventaja del entorno de desarrollo, pues es bastante amplio, por lo cual cuenta con gran cantidad de recursos en línea junto con los de Arduino, facilitando la resolución de diferentes problemas. Finalmente, es el microprocesador más económico, incluso teniendo en cuenta el módulo adicional para Modbus.

### **Figura 7**

*Microcontrolador ESP32 DevKit V1*



Nota. Tomado de *Mercado Libre*, Noviembre de 2023 ([https://articulo.mercadolibre.com.co/MC-O-585538233-tarjeta-de-desarrollo-esp32-wifi-bluetooth-\\_JM#position=3&search\\_layout=stack&type=item&tracking\\_id=e8c8e5b7-9fb7-4c48-91b6-542b44efc17e](https://articulo.mercadolibre.com.co/MC-O-585538233-tarjeta-de-desarrollo-esp32-wifi-bluetooth-_JM#position=3&search_layout=stack&type=item&tracking_id=e8c8e5b7-9fb7-4c48-91b6-542b44efc17e))

### 3.2. Diseño del Esquemático de la Medición de Potencia

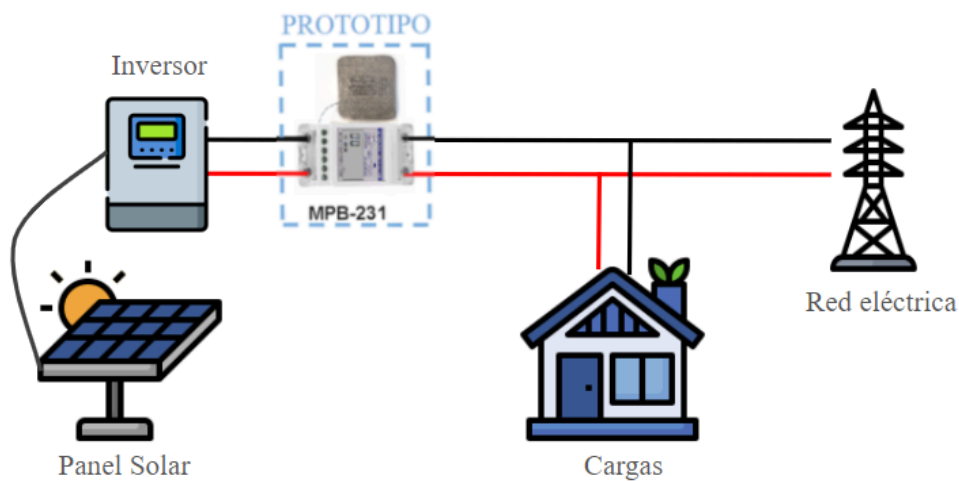
Para calcular el ahorro de energía, se establecieron dos configuraciones diferentes, dependiendo de si los paneles solares On-Grid generan un excedente de energía para inyectar a la red eléctrica.

Configuración 1: Cuando no se genera un excedente, el prototipo se conecta después del inversor del panel solar, lo que permite el seguimiento de la energía entregada a la casa.

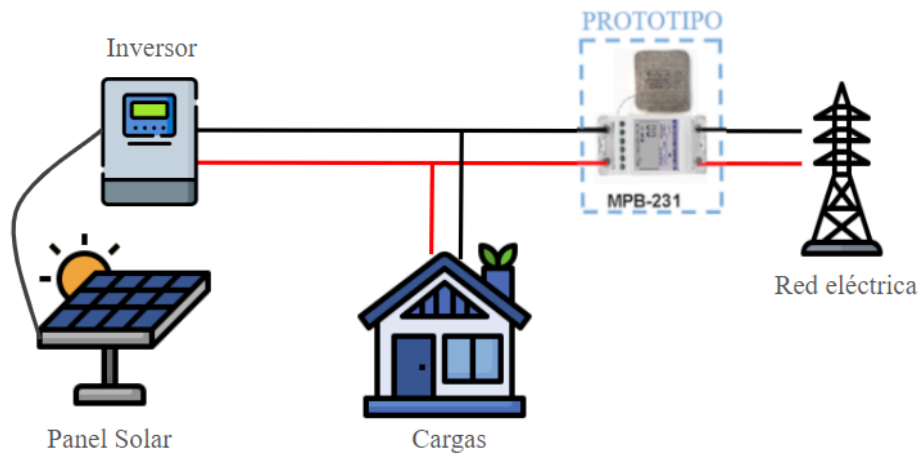
Configuración 2: Cuando sí se genera un excedente, el prototipo se conecta a la red eléctrica, permitiendo el seguimiento bidireccional del consumo tanto de la energía entregada por la red como de la energía inyectada a esta.

**Figura 8**

*Esquemático de la medición de potencia para el calcular el ahorro de energía*



*a) Configuración 1*



*b) Configuración 2*

*Nota.* Se debe tener la precaución de que el dispositivo se conecte solo a una fase.

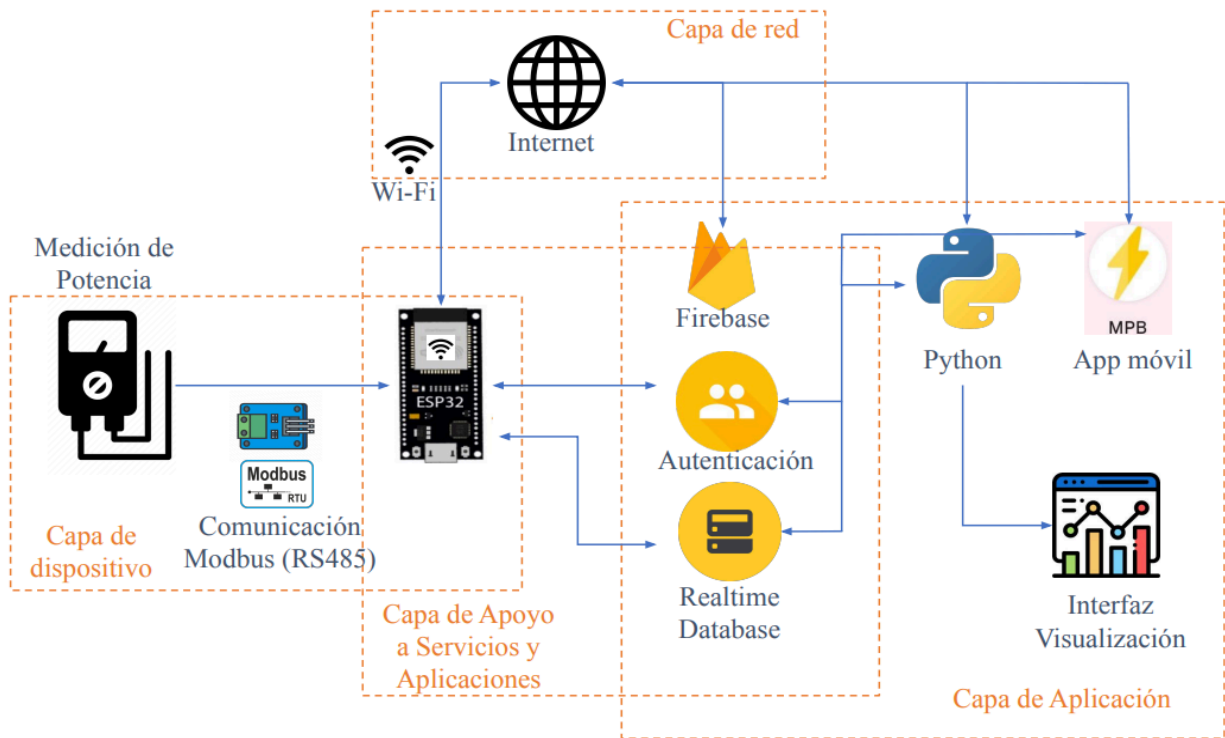
La tarifa del precio del kWh es estratificada, regulada por la CREG. Para las pruebas de este proyecto se tomó el valor 796,38 COP de marzo de 2024 de estrato 3, si se desea información actualizada puede ir directamente a la [página oficial de la ESSA](#). (*Fórmula Tarifaria Y Tarifas - Consultar Tarifas, 2024*)

**3.3. Diseño de la Arquitectura IoT**

La arquitectura IoT que se diseñó para el proyecto se puede observar en la Figura 9, este diseño se realizó con base en el modelo de la UIT mostrado en la Figura 5 de la sección 2.2.3.1 *Modelo IoT*. En este diagrama se muestran las tecnologías y los protocolos utilizados en cada capa a la que corresponden.

**Figura 9**

*Diseño Arquitectura IoT*



### 3.4. Implementación de la Comunicación

En esta sección, se muestra todo el proceso de la implementación de la comunicación de las tecnologías y protocolos utilizados en este proyecto. Esta comunicación es la que permite la transmisión de datos entre todos los componentes del sistema, permitiendo la recopilación, el procesamiento y la visualización de la información.

Este proceso de implementación abarca desde la configuración de la conexión Wi-Fi para la comunicación inalámbrica hasta la integración de protocolos como MODBUS para la

interacción con el medidor DDS-669 y la utilización de Firebase para el almacenamiento y la sincronización de datos en la nube. Se tiene un manual donde se explica a detalle de la conexión y programación en el *Anexo A*. Además, se explica la seguridad y privacidad que Firebase proporciona para la información transmitida.

Para tener toda esta información técnica se puede dirigir al repositorio colaborativo en Github accediendo a este [enlace](#), el cual cuenta con los códigos de programación realizados y además con manual de usuario diseñado por nosotros para tener una guía detallada de la implementación.

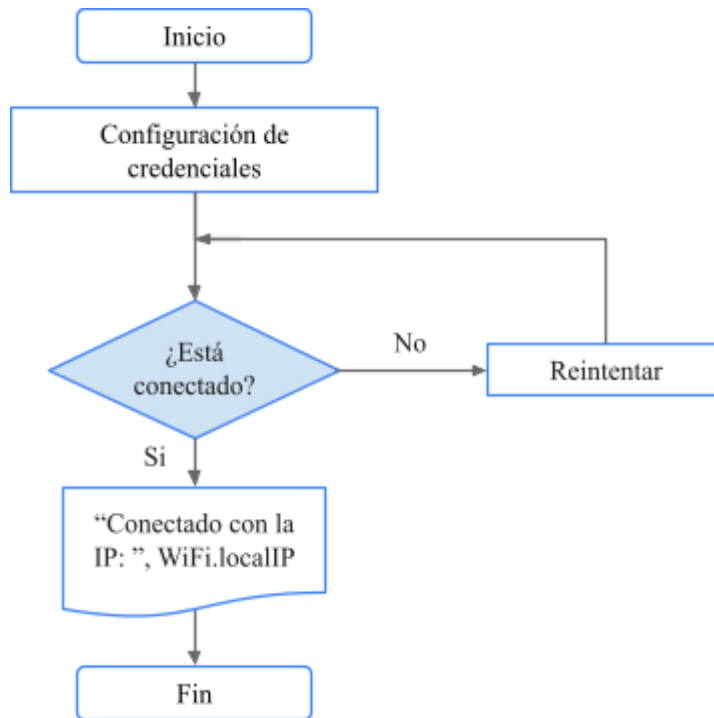
Cabe resaltar, que solo fue utilizado el medidor DDS-669 TAXNELE (Shopee) debido a que el otro medidor DDS238-2 TOMZ(Aliexpress) presentó problemas técnicos con la extracción de los datos a pesar de los esfuerzos realizados en lograr hacer que funcionará, impidiendo la comunicación de este.

### ***3.4.1. Configuración de la Conexión Wi-Fi***

La configuración de la conexión Wi-Fi con el ESP32 se realizó con la librería estándar “*wifi.h*”. Para inicializar la conexión se debe usar la función “*WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD)*”, esta función requiere como parámetros el nombre de la red (SSID) y la contraseña de la red a la cual se desea conectar.

**Figura 10**

*Diagrama de flujo para la comunicación Wi-Fi.*



**3.4.2. Implementación de Modbus para la comunicación con el medidor DDS-669**

El protocolo Modbus se implementa con la interfaz física de comunicación RS485, estableciendo así la comunicación entre los hosts (también llamado "Maestros") y los dispositivos (también llamados "Esclavos"), permitiendo una solicitud de supervisión y configuración de los dispositivos. (Protocolo RS485 Vs Modbus, 2021). Las características asociadas de la comunicación para este medidor se muestran en la siguiente Tabla 4:

**Tabla 4**

*Características de Comunicación Modbus entre el dispositivo y el microcontrolador.*

<b>Característica</b>	<b>Configuración</b>
ID dispositivo	01
Velocidad de transmisión	9600 bauds
Bits de Datos	8
Bits de Paridad	Deshabilitados (Ninguna Paridad)
Bits de Parada	2
Mode	RTU

*Nota:* Las Características de Comunicación Modbus son específicas para cada dispositivo de medición.

Teniendo en cuenta toda esta información, se procede a hacer una configuración precisa de los registros del medidor, la estructura de la trama y el protocolo de comprobación de la trama.

**3.4.2.1. Registros del Medidor.** Una vez teniendo configurada la conexión entre el host (ESP32) y el dispositivo, mediante el protocolo de comunicación Modbus RTU, se debe tener conocimiento de los registros (direcciones de las peticiones las cuales puede ver en este [enlace](#)), para así obtener los datos tomados por el dispositivo. Estos datos se almacenan en la memoria del dispositivo, por lo cual es necesario conocer la dirección en la que se encuentran alojados para así poder acceder a estos.

Para el medidor seleccionado en este proyecto, el DDS-669, contamos con la siguiente Tabla 5, la cual muestra las direcciones donde se almacenan las variables en memoria de este dispositivo:

**Tabla 5**

*Direcciones Almacenamiento de Variables del DDS-669*

*Nota:* Las Direcciones Almacenamiento de Variables son específicas para cada dispositivo de

<b>Contenido del Dato</b>	<b>Dirección de Registro</b>	<b>Formato</b>	<b>Longitud de Dato</b>	<b>Unidad</b>
Voltaje	0	int	1	V
Corriente	3	int	1	A
Potencia activa	7	long	2	W
Factor de potencia	20	int	1	0 - 1.00
Frecuencia	26	int	1	Hz
Potencia activa total	30	int	1	kWh
Potencia activa Generada	40	int	1	kWh
Potencia activa consumida	50	int	1	kWh

medición.

Teniendo la dirección de registros donde se alojan las variables se puede leer o editar (read/write). Para el caso de este medidor, la función “Read Input Registers (Función 04)”, lee los valores de los registros de entrada (analógicos) en un dispositivo esclavo.

**3.4.3.2. Estructura de la trama.** La estructura de la trama (mensaje) utilizada para la comunicación con el dispositivo permite que este responda con el valor solicitado en la trama. La trama lleva la dirección del dispositivo al cual se le realiza la petición, la función, el dato solicitado y un protocolo de verificación de la misma trama, en ese orden respectivamente, así como se ve representada en la Tabla 6 :

**Tabla 6**

*Estructura de la Trama Enviada*

<b>Dirección Dispositivo</b>	<b>Código de Función</b>	<b>Dato</b>	<b>Checksum(CR1 - CR2)</b>	
01	04	00 00 00 01	31	CA

*Nota.* El código a utilizar en el envío de la trama se encuentra en este [enlace](#).

El dato en sí, está compuesto de dos segmentos, la dirección del registro que va a leer y el número de registros a leer. Se puede hacer la petición de más de un registro en una misma trama pero para este proyecto, las peticiones se hacen individualmente a cada registro.

En la Tabla 7 se representan la estructura del segmento de datos que contiene la trama:

**Tabla 7**

*Estructura del Dato Enviado*

<b>Dirección del Registro</b>	<b>Número de registro a leer</b>
00 00	00 01

**3.4.2.3. Protocolo de comprobación de trama.** El CRC (Cyclic Redundancy Check) es un tipo de suma de comprobación utilizada para verificar la integridad de los mensajes transmitidos. El propósito principal del CRC es detectar errores de transmisión que puedan ocurrir durante la comunicación, esta comprobación se posiciona en la parte final de la trama y tiene la siguiente estructura representada en la Tabla 8 :

**Tabla 8**

*Estructura de Checksum de la trama*

CRC1	CRC2
31	CA

**3.4.3. Configuración de Modbus en el microcontrolador**

Para llevar a cabo la configuración para la comunicación de Modbus en el ESP32, es necesario el Módulo Conversor TTL a RS485 MAX485. Esta board de origen comercial funciona a base del chip MAX485, que es un transceptor de baja potencia con una velocidad de respuesta limitada, que puede alcanzar una velocidad máxima de transmisión de 2.5Mbps. Más adelante se muestra a detalle el esquema de conexión realizado para este.

Seguido a esto, la configuración en la programación para la comunicación de Modbus en el ESP32 se logra mediante una librería dedicada a este protocolo de comunicación, gracias a la amplia comunidad de desarrollo de Arduino se tienen muchas opciones para elegir librerías que cumplan estos requisitos.



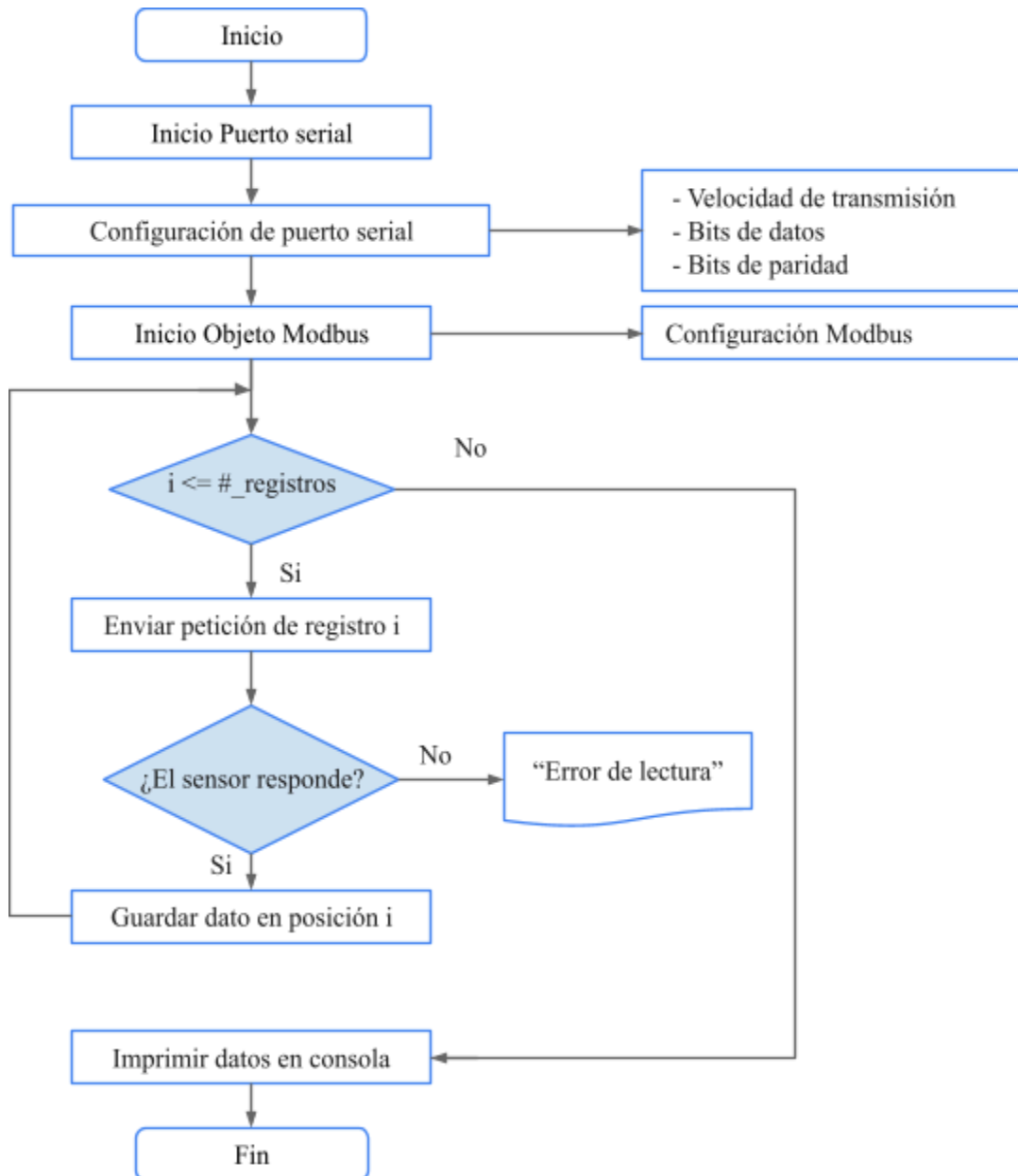
**3.4.3.2. Programación de Comunicación Modbus.** Para la programación de la comunicación Modbus del microcontrolador, se seleccionó la librería "*Mosbusmaster.h*", diseñada para la comunicación con dispositivos esclavos Modbus mediante RS232-RS485 (con protocolo RTU). Esta librería integra cuenta con pines de control que configuran al chip MAX485 para recibir o enviar la información.

Dentro del código, se debe inicializar un puerto serial "*Serial2.begin(BAUD, SERIAL\_8N1)*" y configurar la velocidad de comunicación del puerto (9600 baudio, valor por defecto para el medidor) junto con el tamaño de datos, para este caso *SERIAL\_8N1*. Por defecto, la paridad de esta librería es "none parity". Después de configurar la comunicación serial, se inicia la variable asociada a la comunicación Modbus con el medidor, en nuestro caso se dejó la variable "*node*", la cual se debe inicializar con un "*node.begin( ID, Serial\_configurado)*". Dentro de esta función se debe configurar la ID del dispositivo y el puerto serial que realiza la comunicación (en nuestro caso, Serial2).

Una vez configurados todos los canales de comunicación se realiza la petición de los datos utilizados con la función "*node.readInputRegisters( direccion\_registro, numero\_registro\_a\_leer)*". Estos datos se guardan en la variable "data" haciendo uso de la función "*node.getResponseBuffer(0)*", donde el cero es la posición de la variable buffer que guarda el valor solicitado. En la Figura 12 se muestra el diagrama de flujo diseñado para este proceso:

**Figura 12**

*Diagrama de flujo para la comunicación de Modbus con la ESP32.*



#### ***3.4.4. Comunicación con Firebase desde el Microcontrolador***

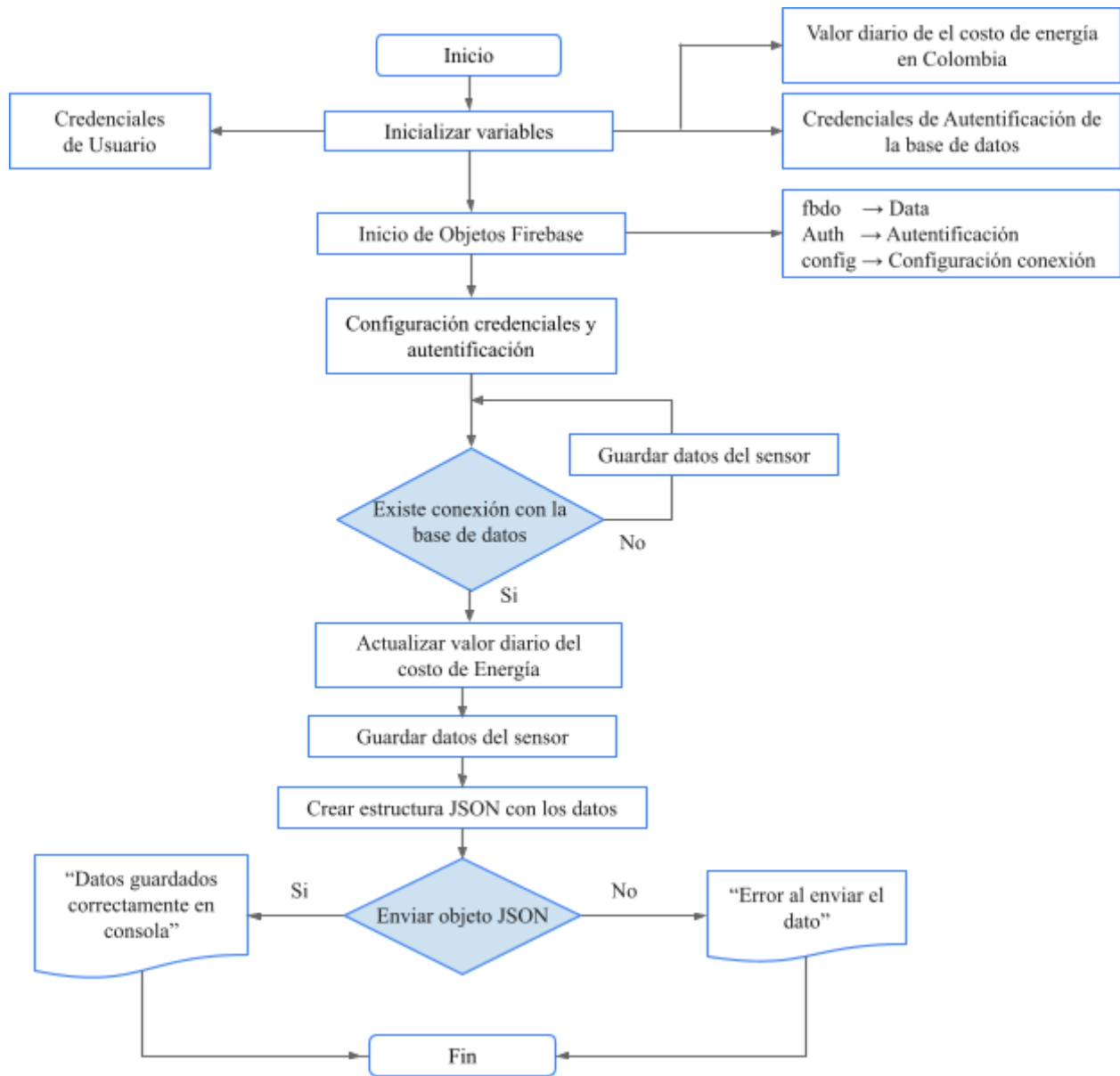
Para la comunicación con Firebase desde el ESP32, se utiliza la base de datos en la nube de Firebase Realtime Database. Este servicio permite conectar el ESP32 con la aplicaciones de visualización de los datos, aprovechando que proporciona actualizaciones en tiempo real y una administración de los datos de forma segura, sencilla y rápida.

Para explicar mejor, se describe primero, la programación general llevada a cabo con sus respectivas librerías, luego el establecimiento de la conexión y finalmente el envío de datos.

- **Programación para Firebase desde el ESP32:** Dentro del entorno de arduino IDE, software en el cual se desarrolló el código del microcontrolador, se seleccionó la librería “*FirestoreClient.h*” para realizar la comunicación con Firebase desde el ESP32. Esta biblioteca cuenta con funciones que admiten RTDB, Cloud Firestore, Firebase Cloud Storage, Google Cloud Storage, Cloud Messaging y Cloud Functions; y está enfocada para los dispositivos ESP8266, ESP32 y RP2040 Pico, en la Figura 13 podemos detallar como esta estructurado la programación del dispositivo:

**Figura 13**

*Diagrama de flujo para la comunicación con Firebase desde la ESP32.*



*Nota.* En el siguiente [enlace](#) encuentra un video donde se puede a ver a detalle cómo conectar el ESP32 con Firebase.

- **Establecimiento de conexión:** Para establecer la comunicación desde la ESP32 con la base de datos de Firebase (está localizada en Estados Unidos (us-central1) ) se deben definir variables. La primera variable involucra un correo electrónico y contraseña previamente asociado al proyecto de Firebase. La segunda variable es la API\_KEY del proyecto. La tercera y última variable es la dirección URL del proyecto. Toda esta información se puede ver en la pestaña de configuración del proyecto, a excepción de la URL de referencia que está en la pestaña de Realtime Database.

A continuación se definen los objetos que permiten la configuración de la conexión,, comenzado por el objeto “FirebaseData” como “fbdo ”, el objeto “FirebaseAuth” como “auth”, y por último, el objeto “FirebaseConfig” como “config”. Estos objetos se usan para configurar la conexión con la base de datos. La configuración básica para lograr una conexión segura, se realiza de la siguiente manera:

1. En el objeto “config”, se agregara el método API\_KEY y a esto se le asigna la variable que contiene la API\_KEY (config.api\_key = API\_KEY).
2. En el objeto “auth” se agregan los métodos “user” seguido de “email o password” dependiendo del que se esté utilizando y se les asigna la respectiva variable (auth.user.email = USER\_EMAIL).
3. Se configura la URL con (config.database\_url = DATABASE\_URL).
4. Finalmente, se inicia la conexión, para esto se debe declarar la siguiente línea de código: “*Firebase.begin(&config, &auth)*”, la cual en sus argumentos lleva la configuración realizada con la autenticación.

- **Envío de datos:** Para enviar datos a Firebase, se declara el objeto FirebaseJson como “*FirebaseJson medidorData*” con la dirección de la carpeta raíz donde estas los datos almacenados, en este caso “*medidorData*”. Para enviar o actualizar los datos en la base de datos se utiliza el método “set()”. Este método en su primer argumento recibe la dirección (en formato de texto) donde se guardará la variable “*medidor/Corriente*” y como segundo argumento, el dato a enviar, como ejemplo: “*medidorData.set("Voltaje", voltaje)*”.
- Para los datos enviados tenemos un total de 11 variables, 8 de las cuales podemos ver en la tabla 5, las variables restantes son calculadas dentro del código de la siguiente manera:
  - **kW:** Se obtiene al dividir la variable de potencia tomada del medidor.
  - **kWh:** Se obtiene al acumular con cada lectura del medidor la variable de kW dividiéndola en 3600, quiere decir que estaremos calculando la energía medidas segundo a segundo por el medidor.
  - **Dinero\_kWh:** Se obtiene al multiplicar la variable de kWh por el valor de costo el cual es actualizado tomando la variable costo de firebase, al inicio de encender el sistema y cada vez que la conexión con firebase esté establecida correctamente.

#### ***3.4.5. Comunicación con Firebase en la aplicación***

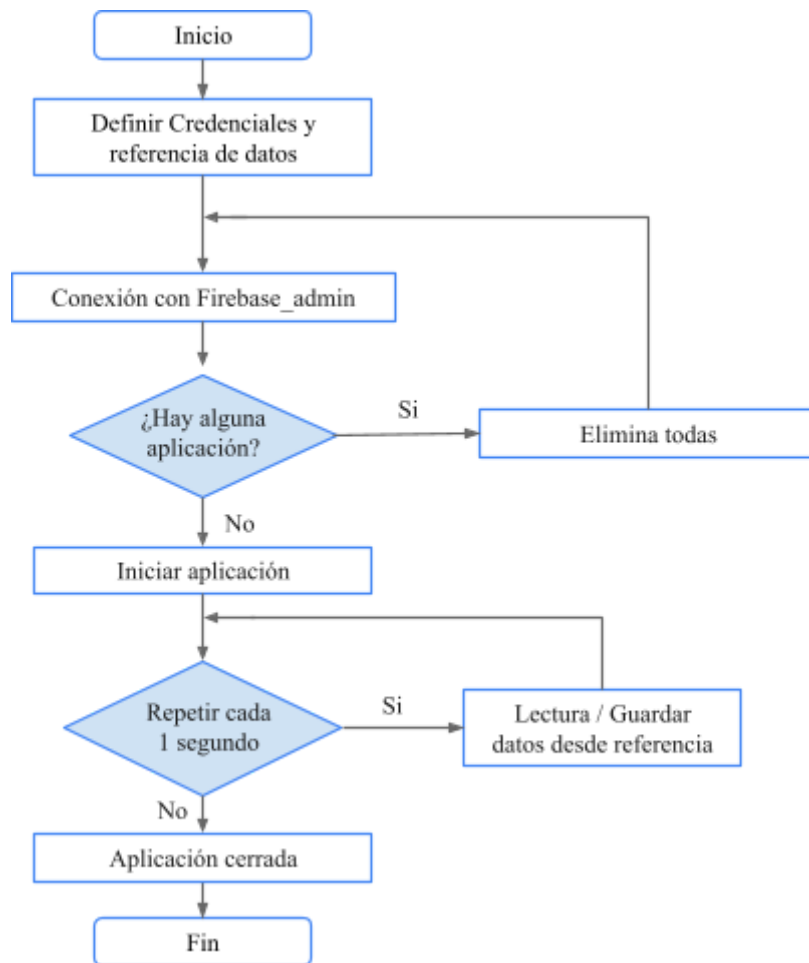
La aplicación se desarrolla en el lenguaje de programación Python, elegido, como ya se ha mencionado antes por sus extensa cantidad de librerías y comunidad que hace más amena la programación con este lenguaje.

- Programación con Firebase en la aplicación:** Se utiliza el *SDK* de Firebase Admin, esto permite ejecutar acciones como leer y escribir información en la base de datos en tiempo real como administrador. Para la programación de la comunicación con Firebase hacia la aplicación se realizó el diagrama de flujo que se observa en la Figura 14.

El SDK utiliza como término “aplicación” al enlace que se toma con un dispositivo, de esta manera puede tener múltiples dispositivos tomando datos de Firebase, siempre y cuando se tenga las credenciales necesarias y correctas.

**Figura 14**

*Diagrama de flujo de la comunicación con firebase hacia la aplicación.*



- Establecimiento de Conexión:** Para establecer la conexión con la base de datos de Firebase se debe generar una nueva clave privada en formato JSON que proporciona información de las credenciales de autenticación. Esta clave obtiene desde la configuración del proyecto de Firebase, en la pestaña “Cuentas de Servicios”, en este apartado, se encuentra la opción para generar el archivo de clave privada y un código para establecer la conexión, como se puede ver en la Figura 15:

**Figura 15**

*Creando nueva clave privada*



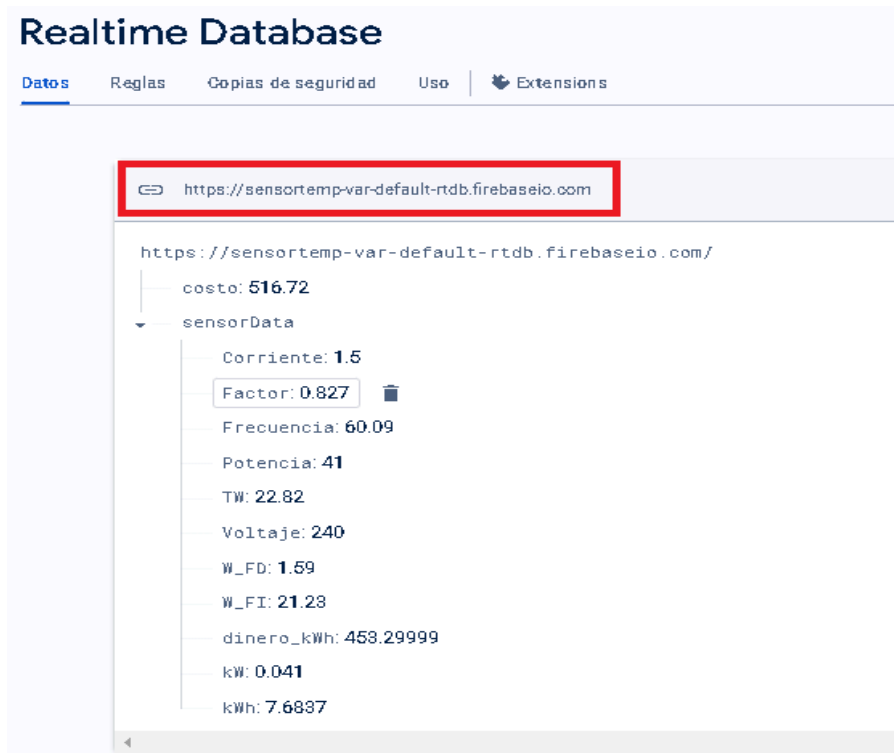
*Nota.* Para la creación del proyecto en Firebase y configurar la conexión puede ver el video de este [enlace](#).

Teniendo la clave privada, es necesario asegurar que esta clave está ubicada en la misma carpeta donde se encuentran los archivos de la aplicación Python, o en lugar de eso, ejecutar la aplicación para realizar la configuración de estos datos de autenticación.

Definidas las credenciales en una variable, se realiza la conexión con el servidor de Firebase a través de `“firebase_admin.initialize_app(cred, {'databaseURL': 'URL del proyecto'})”`, que recibe la variable de las credenciales y la URL de la base de datos en Firebase. La URL se puede adquirir dentro del proyecto de Firebase, en la opción de datos del apartado de Realtime Database. En la Figura 16 se indica con un cuadro rojo donde se puede obtener esta credencial.

**Figura 16**

*Estableciendo Conexión con Firebase, URL del proyecto*



*Nota.* Figura tomada de la página (*Monitoreo De Potencia*, n.d.)

- **Lectura de datos en firebase:** Para leer los datos, se utiliza la función

`“db.reference('/medidorData/medidor/')”`, está proporciona la dirección con la ubicación específica del dato que se está buscando en la base de datos. Esta dirección se guarda en una variable “ref” y por medio de la función `“get()”` se extraen los datos con el comando `“data = ref.get()”`.

La variable “data” almacena un diccionario con los datos que se encuentran en la referencia (dirección) de la siguiente manera: `{'Corriente': 0.22, 'Factor potencia': 0.955, 'Frecuencia': 60.02, 'Potencia': 26, 'TW': 5.2, 'Voltaje': 120.5, 'W_FD': 3.72, 'W_FI': 1.48}`. Esto permite el acceso a cada valor individualmente. Finalmente, para el almacenamiento de datos, se guardan en variables individuales utilizando el comando `“data.get('Voltaje', None)”`, lo que permite obtener el valor de la variable “Voltaje”.

#### ***3.4.6. Seguridad y privacidad en Firebase***

Siguiendo con lo descrito en el capítulo 2.3.1 *Privacidad y seguridad en Firebase*, se hará énfasis en las reglas que se establecieron en Firebase Realtime Database tanto para la comunicación con el ESP32 como para la aplicación.

La comunicación segura con Firebase se logra a través de su SDK, como ya se explicó anteriormente con el servicio de Firebase Admin. Para establecer la conexión con el proyecto en Firebase, se requiere la descarga de un archivo JSON que contiene códigos de autenticación y configuraciones esenciales. Estos códigos y configuraciones actúan como credenciales, asegurando que solo aquellos con acceso autorizado pueda modificar o acceder a la base de

datos. En este sentido, la información sensible está protegida y solo accesible para usuarios debidamente autenticados.

Además, con “*indexOn[]*” se definieron índices específicos para las diferentes variables de los datos medidos, lo cual permite ordenar mejor los datos para consultas u operaciones más específicas que requieran filtrarlos. Finalmente, con “*.validate:*” se implementó la validación para asegurar que los datos tomados sean números válidos para cada variable, evitando así que entre información incorrecta en la base de datos.

### **3.5. Visualización de datos**

Decidimos desarrollar una aplicación clara para visualizar los datos tomados, utilizando el lenguaje de programación de Python, ya que este cuenta con gran versatilidad y una comunidad de entorno de desarrollo muy amplia que favoreció el desarrollo del proyecto. Además, por medio del uso de el SDK que proporciona Firebase Admin se logra una conexión segura entre Python y Firebase Realtime Database.

Adicionalmente, teniendo en cuenta que los teléfonos móviles son parte de la vida cotidiana para la mayoría de las personas, se desarrolló una aplicación Android utilizando App Inventor, la cuál permite mostrar un resumen de las variables más importantes registradas por el medidor, facilitando al usuario acceder de forma sencilla a los datos del monitoreo y ahorro de energía desde cualquier lugar en que se encuentre.

### ***3.5.1. Diseño de la Interfaz de Usuario***

Para el desarrollo del diseño de la interfaz de usuario se siguieron los principios descritos en el *Anexo B*, donde se busca tener una interfaz clara para el usuario y así poder brindar una experiencia más cómoda para la visualización de los datos.

### ***3.5.2. Aplicación de windows***

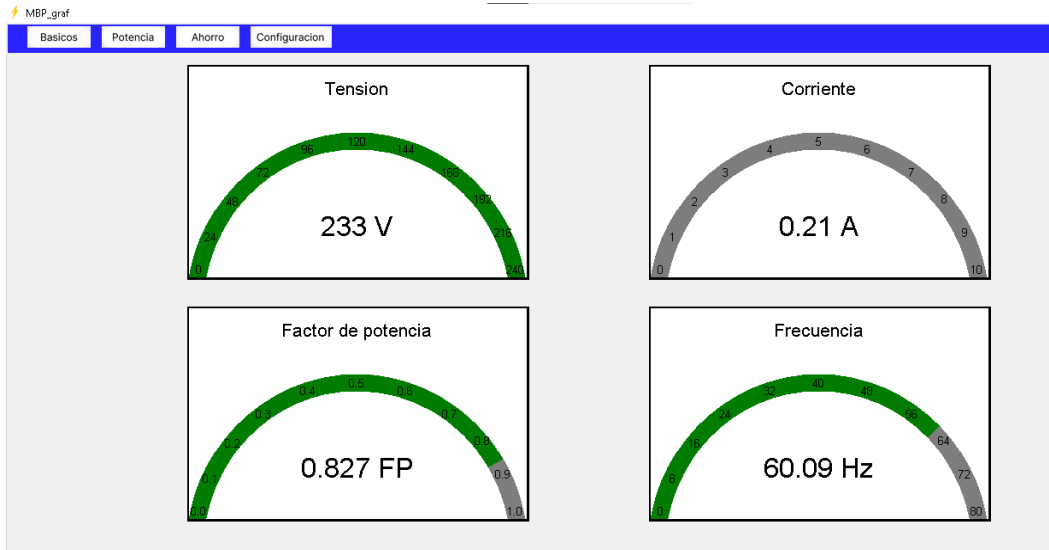
Para el desarrollo de la aplicación Windows, se hizo uso de Visual Studio Code la cual cuenta con soporte en programación en Python. Dentro de este lenguaje encontramos la librería Tkinter, especializada en el desarrollo gráfico para la interfaz de usuario.

El diseño de la aplicación está integrado por cuatro páginas: tres de ellas muestran los datos eléctricos que se miden desde el medidor y la cuarta página permite la configuración de parámetros sobre la base de datos a la cual se quiera acceder.

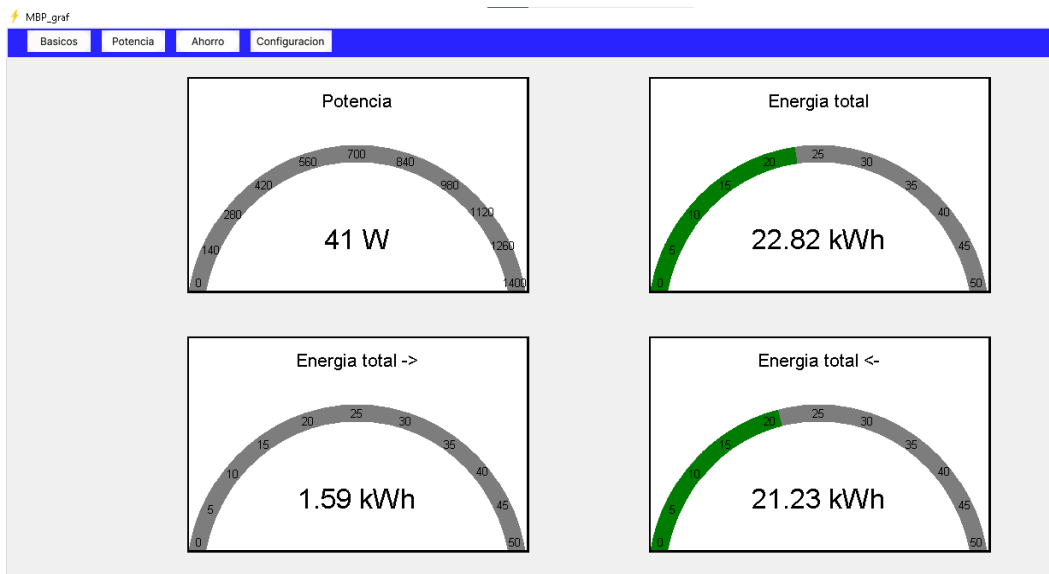
En la Figura 17 se puede ver el diseño de la aplicación que toma como base la simplicidad y comodidad a la hora de observar los datos medidos. Además se optó por el uso de barras de progreso( a) y b)) y de barras temporales (c)) para mostrar el monitoreo del ahorro obtenido mediante el uso de un sistema fotovoltaico.

**Figura 17**

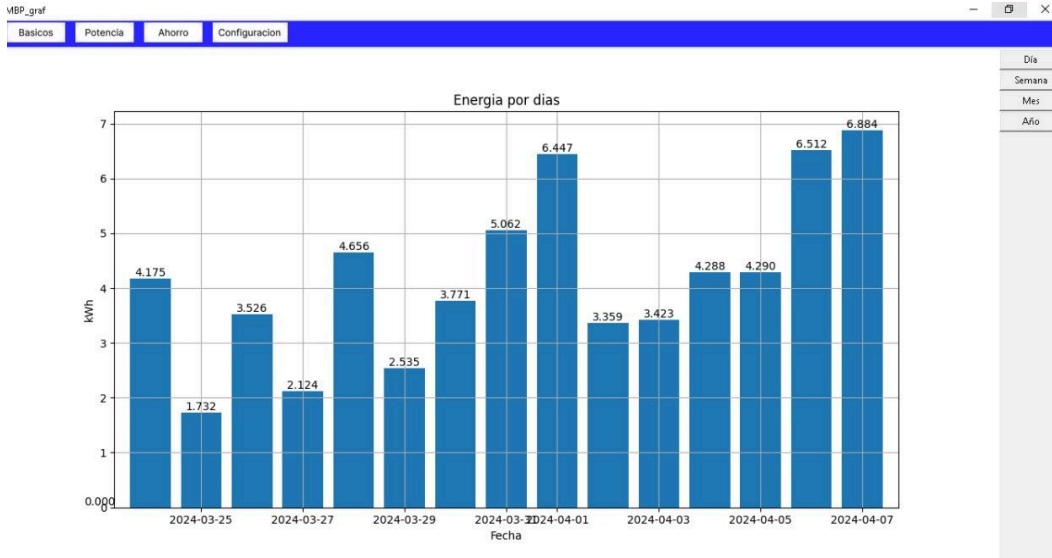
*Aplicación Windows*



*a) Visualización de tensión, corriente, FP y frecuencia*



*b) Visualización de Potencia, Energía total y energía en cada dirección*



c) Barras de temporales del Ahorro de energía

Llave generada JSON:    
 Database URL:    
 Referencia:

d) Configuración de parámetros para la base de datos

Nota. En c), la visualización en las barras temporales del Ahorro de energía se da la opción al usuario de verlo por día, semana, mes o año.

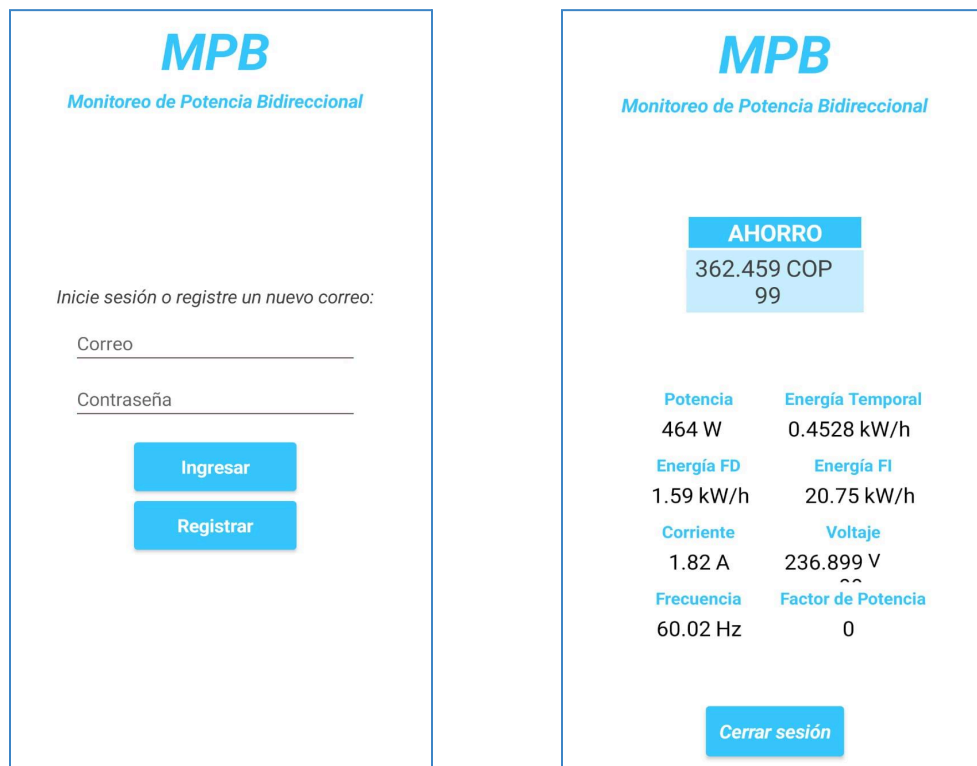
Para ver el código realizado puede dirigirse a este [enlace](#).

### 3.5.3. Aplicación Móvil

Para el desarrollo de la aplicación móvil se hizo uso de la plataforma App Inventor, es un entorno visual de programación por bloques que permite conexión directa con Firebase y mediante tokens de seguridad permite el acceso a la autenticación y la base de datos en tiempo real. Esta aplicación fue diseñada pensando en la comodidad y simplicidad de la visualización de los datos, mostrando así de forma directa y en una sola pestaña el ahorro de energía y las variables de monitoreo más importantes en tiempo real. En la Figura 18 se muestra el diseño de la interfaz de esta aplicación.

**Figura 18**

*Interfaz de la Aplicación Móvil*



*Nota.* Para descargar el aplicativo y ver cómo se desarrolló su programación puede dirigirse al siguiente [enlace](#).

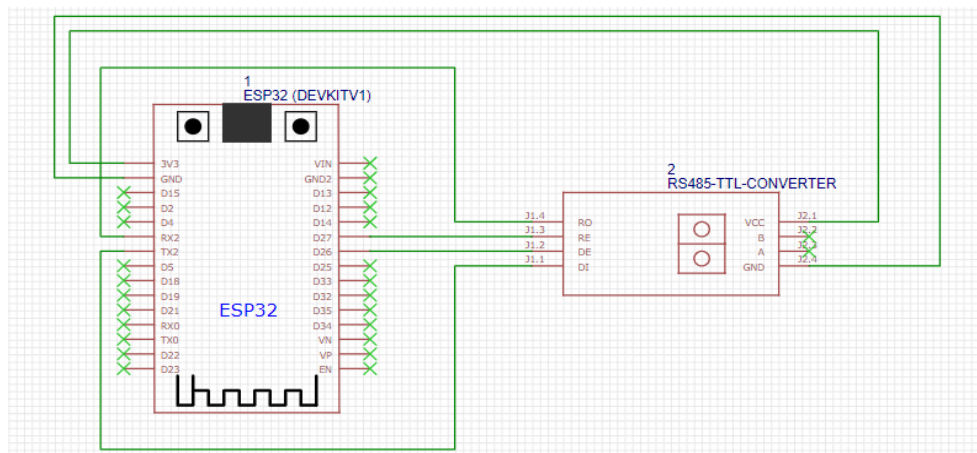
### 3.6. Diseño de la PCB

Para la implementación de este proyecto se va incorporar por medio de una PCB el medidor DDS-669 TAXNELE con el microcontrolador ESP32.

Se realizó el diseño de la PCB utilizando el software EasyEDA, este permite el diseño y simulación de circuitos y cuenta con bibliotecas integradas que se actualizan continuamente por su acceso a conexión a internet. Primero se realiza el esquemático mostrado en Figura 19 para tener una visualización correcta de todos los componentes y sus conexiones, siendo esta la base para el diseño de la PCB

**Figura 19**

*Esquemático*

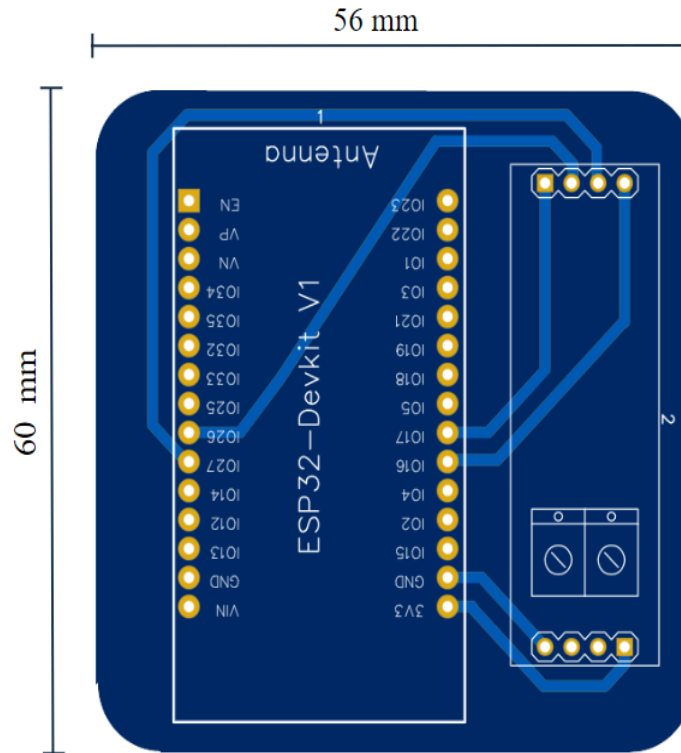


Luego, en base al esquemático mostrado en la Figura 19 se procede al diseño de la PCB mostrada en la Figura 20, el cual nos permite mejor estabilidad y eficiencia para la comunicación entre los componentes.

La PCB tiene dimensiones de 60mm x 56mm. Como es un diseño sencillo, se diseñó de una sola capa, lo que la hace más económica.

**Figura 20**

*Diseño de la PCB*



*Nota.* Representación de la PCB en 2D realizada en el software EasyEDA, (2024).

Finalmente se diseñó un compartimiento en una impresora donde se puede almacenar la PCB y se instaló a un lado del medidor, de esta forma se tiene un dispositivo compacto que protege todos los componentes, podemos ver el ensamble final en la Figura 21.

**Figura 21**

*Ensamble final de la PCB y el dispositivo*



*Nota.* Se debe tener la precaución que la caja se instale de tal forma, que no intervenga con el espacio de paso de la regleta en la que se suelen sostener los medidores.

## 4. Resultados y Análisis

Para evaluar el rendimiento y eficiencia del dispositivo, se ejecutaron diferentes pruebas tanto para medir el dispositivo, como el medidor seleccionado.

El rendimiento se evaluó por medio de un análisis de desviación estándar y linealidad de los parámetros medidos. Esto nos da una medida cuantitativa de precisión de las mediciones en comparación con los valores de referencia simulados.

Por otro lado, la eficiencia se evaluó analizando la efectividad y eficacia del dispositivo. Siendo la efectividad la capacidad que tiene el dispositivo para cumplir a cabalidad el propósito para el cual fue diseñado; y la eficacia se evalúa con la rapidez en que se toman y envían los datos y se hacen visibles en el aplicativo.

Para las pruebas del dispositivo, se utilizó el Simulador de Sistemas Fotovoltaicos Lucas Nülle, el cual podemos ver en la Figura 30 y fue puesto a disposición en la Universidad Industrial de Santander, en el Laboratorio de Uso Racional de la Energía y Energías Renovables. Este Simulador cuenta con un emulador de paneles solares hasta de 1500 [W], una salida de 500 [V] y una corriente de 10[A]. Además cuenta con un software configurable que permite cambiar configuraciones de la intensidad de la luz, cuando hay sombra, en general, las condiciones que se podrían presentar en un día.

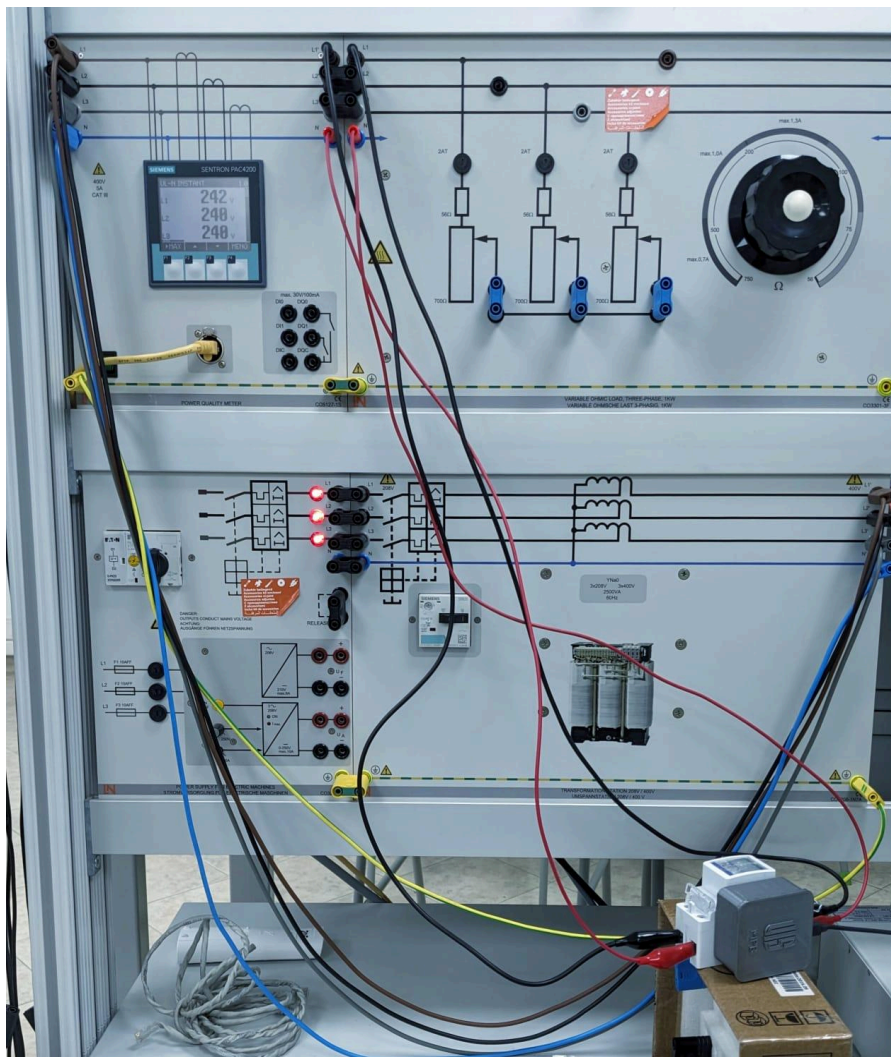
### 4.1. Rendimiento

La metodología utilizada para evaluar el rendimiento se hizo midiendo el voltaje, corriente, potencia. En estas se variaron los valores de las cargas resistivas controladas para así poder analizar el funcionamiento en la toma de medidas del dispositivo.

Estas pruebas fueron realizadas con el simulador de Sistemas Fotovoltaicos Lucas Nülle, el prototipo fue conectado a la fuente de tensión, que simula ser la red eléctrica, específicamente a una fase de tensión que entregaba un parámetro fijo de tensión de 240 [V]. Esta cuenta con un medidor con un software de SCADA que muestra las medidas de toda la implementación, las cuales fueron tomadas como referencias. La conexión se muestra en la Figura 22.

**Figura 22**

*Conexión pruebas de rendimiento y bidireccionalidad*



*Nota.* Foto tomada en el Laboratorio de Uso Racional de la Energía y Energías Renovables.

**4.1.1. Linealidad**

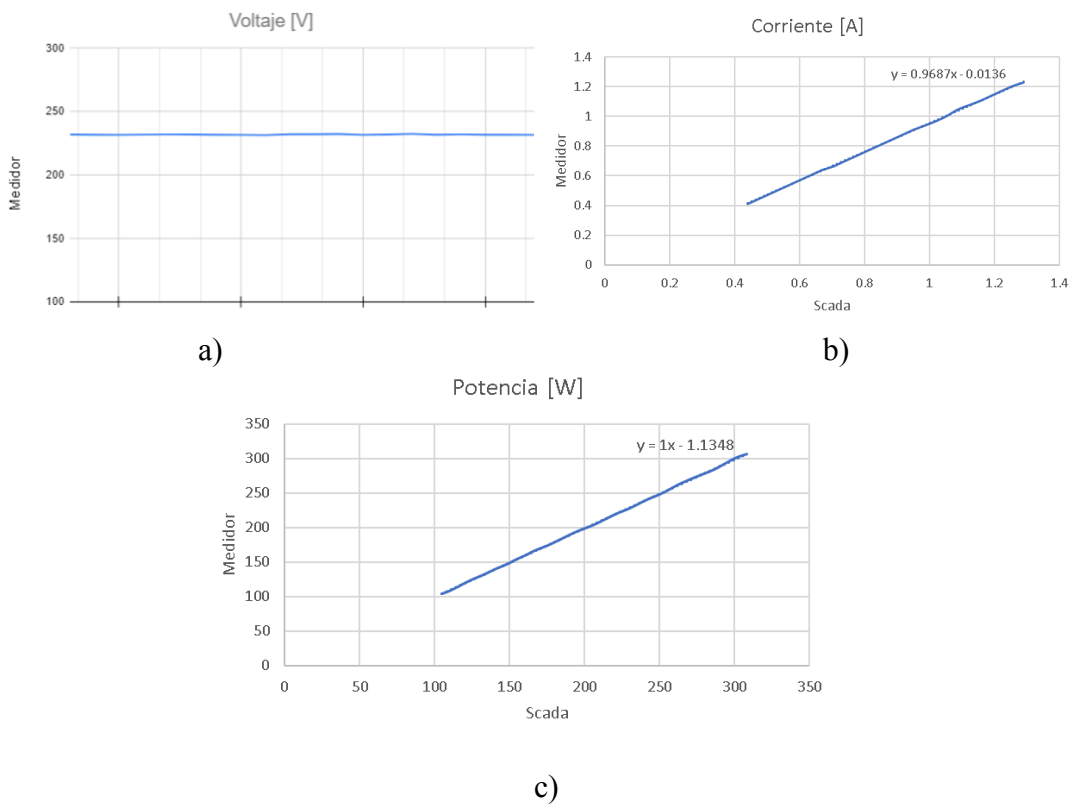
La linealidad es un indicador clave de la precisión y fiabilidad de las mediciones, ya que muestra cómo responden los datos del medidor en relación con un estándar conocido.

Para esto, se realizaron pruebas para ver la linealidad en los datos obtenidos, donde se comparan las mediciones del medidor con las del dispositivo de referencia. De esta forma se puede evaluar el comportamiento lineal del funcionamiento del dispositivo.

Se realizó la medición de la tensión, corriente y la potencia. Los comportamientos lineales correspondientes junto con su ecuación que lo representan, se pueden ver en la Figura 23

**Figura 23**

*Linealidad de voltaje (a), corriente (b) y potencia (c).*



*Nota.* Las gráficas se encuentran junto con su ecuación que lo representan.

De esta relación lineal se hallaron los errores promedios de cada una de ellas, los cuales se registran en la Tabla 9.

**Tabla 9**

*Error promedio del análisis de linealidad.*

Variable	Error promedio de las muestra
Voltaje	3,58%
Corriente	4,99%
Potencia	0,65%

*Nota.* El error promedio en todas las variables medidas es menor al 5%, por lo cual cumple con los parámetros esperados en la precisión de las mediciones del prototipo.

**4.1.2. Desviación Estándar**

La desviación estándar, es una medida estadística que muestra la dispersión entre un conjunto de datos tomados con respecto a una referencia, así que realizamos las mediciones tomando como guía (Priharti et al., 2019). En este caso, se realizaron pruebas no recurrentes con el Simulador Lucas-Nulle Photovoltaic, donde se variaron cargas resistivas y se registraron los cambios en el voltaje, corriente y potencia.

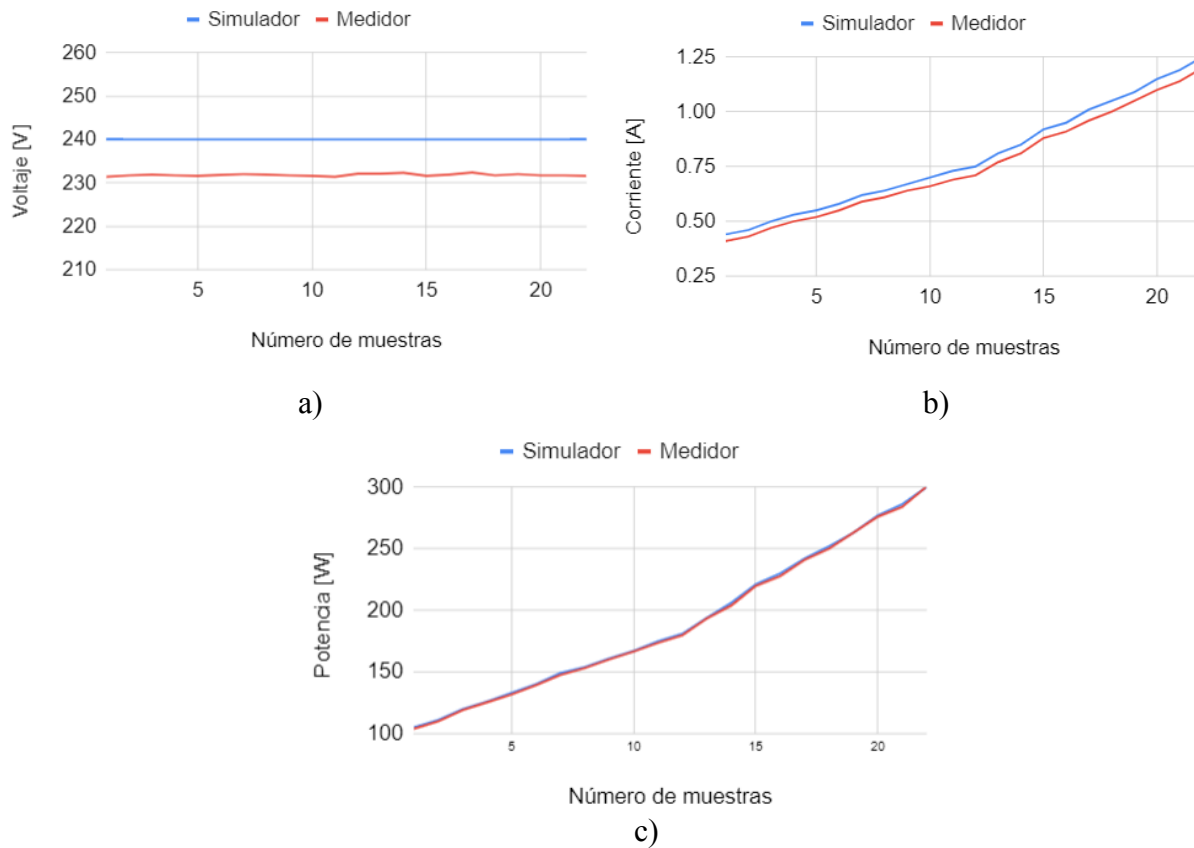
La fórmula de desviación estándar es  $\mu \pm 1\sigma\%$ , que se compone de  $\mu$  que es la media y  $\sigma$  la desviación estándar, como se muestra a continuación:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n-1}} \quad ; \quad \mu = \frac{\sum_{i=1}^n x_i}{n}$$

En los gráficos siguientes de la Figura 24, se mostrarán los resultados obtenidos experimentalmente, comparados con las referencias, esto nos permite visualizar las tendencias y errores en las mediciones.

**Figura 24**

*Desviación estándar de voltaje (a), corriente (b) y potencia (c).*



*Nota.* El voltaje (a) muestra una media de 8,19 y una baja dispersión en los datos. La corriente (b), muestra menos diferencia y más exactitud. La potencia (c) se observa que es la más precisa.

Los resultados de la desviación estándar la podemos observar en la siguiente Tabla 10:

**Tabla 10**

*Desviación Estándar:*

<b>Variable</b>	<b>Desviación Estándar</b>
Voltaje	8,19 ± 0,261%
Corriente	0,03 ± 0,007%
Potencia	1,11 ± 0,590%

*Nota.* Tabla que muestra el análisis de las gráficas de la Figura 24.

El voltaje tiene una media de 8,19, esta diferencia se puede observar fácilmente en la Figura 24 a) Voltaje. A pesar de esto se tiene una desviación estándar de 0,261%, es decir, hay consistencia en las mediciones.

Las desviaciones estándar son muy bajas, lo que nos muestra que nuestro proyecto puede proporcionar medidas consistentes y precisas, entonces si cumple con este parámetro de rendimiento.

**4.3. Eficiencia**

La eficiencia se evaluó con el análisis de la eficacia y de la efectividad del dispositivo. La eficacia está enfocada en la rapidez con la que el sistema realiza sus funciones, en este caso, la actualización de los datos y la respuesta de la aplicación. Por otro lado, la efectividad tiene un aspecto cualitativo que se centra en el análisis del cumplimiento de parámetros que puedan dar calidad al prototipo.

### ***4.3.1. Eficacia***

**4.3.1.1 Eficacia del dispositivo con Firebase.** La eficacia del dispositivo se define por la rapidez con la que toma los datos del medidor y los envía a la base de datos en Firebase. Para medir este tiempo de ejecución, se utilizó la función “millis()”, que proporciona el tiempo transcurrido desde que comienza la ejecución del programa en el microcontrolador ESP32. Se determinó un tiempo total de ejecución de 700 milisegundos  $\pm$  1 milisegundo y se pudo observar que en este intervalo de tiempo, en promedio, son necesarios 302 milisegundos más para enviar los datos medidos a Firebase. Es decir, el tiempo total es de 1,002 segundos.

**4.3.1.2. Eficacia de la aplicación de Windows con Firebase.** Para la aplicación Windows con Firebase, se realizaron pruebas para determinar la rapidez en la ejecución de procesos. Para esta prueba se utilizó el comando “data\_update”, el cual permite solicitar las variables a la plataforma de Firebase. En un periodo de 15 horas, se registraron un total de 45.419 interacciones, lo que nos permitió analizar el comportamiento con la conexión a internet, el cual se puede observar en la Figura 25.

El tiempo programado en el ESP32 está programado para la actualización de los datos en un segundo, durante este análisis se pudo observar que hay intervalos de tiempo de la ejecución del proceso supera este umbral esperado. Para determinar el tiempo promedio que tarda este proceso en ejecutarse, se debe tener en cuenta las condiciones en las que se realizaron la toma de datos, dando espacios en donde las condiciones climáticas eran lluviosas, haciendo que la conexión a internet fuera inestable lo cual provocó que lleguen a haber datos que se demoran 186,1760 segundos (el tiempo máximo registrado). Por esta situación, se dividió el análisis en dos; uno en condiciones de conexión estables, que comprende el rango desde las 24:00 a 5:00

horas, y el otro intervalo donde la conexión era inestable, comprende los rangos desde las 20:00 a 23:00 horas y desde las 5:00 a 10:00 horas. Esto se puede observar en la Tabla 11.

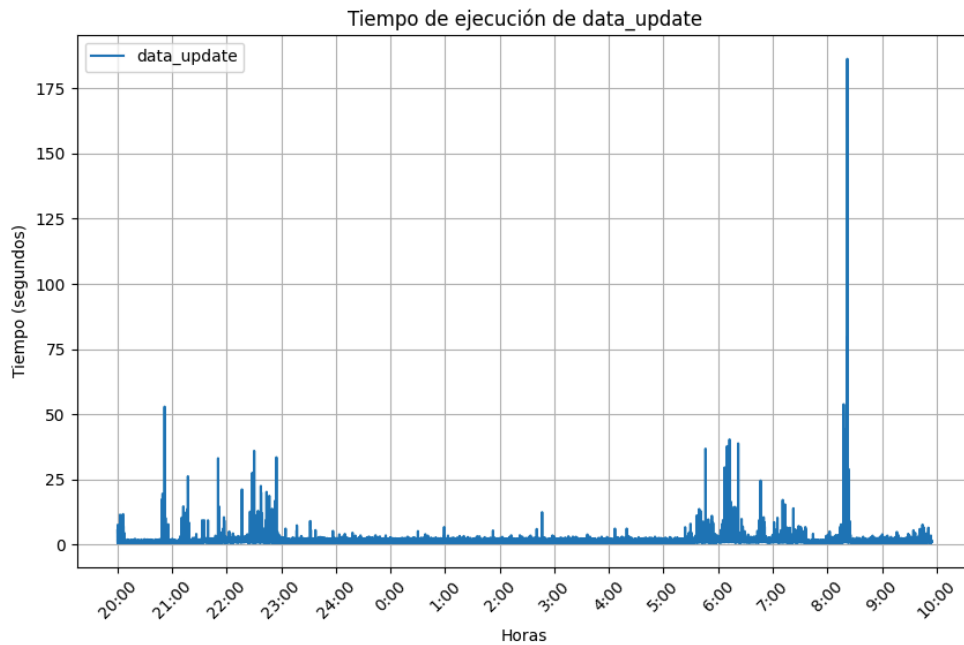
**Tabla 11**

*Rapidez por intervalos de horas*

Intervalo de horas	Promedio en segundos	Desviación Estándar
20:00 - 23:00	1,2697	1,1433
24:00 - 5:00	1,1974	0,5496
5:00 - 10:00	1,5596	1,3969
20:00 - 10:00 (Total)	1,2892	1,3752

**Figura 25**

*Tiempo de ejecución de la aplicación*



*Nota.* La gráfica muestra el tiempo de ejecución de la aplicación durante 15 horas. Estos datos fueron tomados una ubicación de una parcela a las afueras de Tame, Arauca, con una conexión de datos móvil (HSPA(H+) y 4G). Además, durante los lapsos de tiempo de entre las 20 a 23 horas, así como entre las 5 a 10 horas, se presentaron condiciones climáticas lluviosas afectando la estabilidad de la conexión.

#### ***4.3.2. Efectividad***

Con la efectividad nos referimos a la capacidad que tiene el sistema para funcionar de forma confiable revisando si cumple o no con parámetros que van desde la transmisión de datos hasta la automatización del sistema.

- **Rapidez del sistema:** Como se observa en la sección 4.2.1. La transmisión de datos a la base de datos, en total es de 1,002 segundos. Por otro lado, la rapidez de la aplicación Windows en una conexión estable es de 1,1974 segundos e incluso con momentos de conexión inestable es de 1,2982 segundos. Todo esto nos muestra que el sistema si cumple con este parámetro, dando una respuesta rápida y oportuna en la transmisión de los datos.
- **Exactitud de los datos medidos:** Como se observa en la sección 4.1.2. *Desviación Estándar*, en el análisis realizado encontramos que se logra una precisión del 94,5% en las mediciones de los datos. Por cual el dispositivo si cumple con este parámetro.
- **Automatización del proceso:** El sistema no es completamente automático, ya que para calcular el ahorro de energía, es necesaria la intervención manual para

cambiar la tasa del precio de energía mensual. Por lo cual el dispositivo no cumple a cabalidad con este parámetro.

- **Tasa de éxito de transmisión:** Para esta prueba se hizo el envío de 28.800 mensajes, de lo cuales se registraron 286 mensajes fallidos, lo que equivale a que el 99,3% de los datos si se enviaron correctamente. Esto refleja que el sistema si cumple con este parámetro.
- **Fiabilidad:** Durante todo el proceso de análisis, donde a pesar de los inconvenientes encontrados y las pruebas realizadas en diferentes condiciones, se obtuvieron resultados consistentes y coherentes. Así que podemos concluir que si cumple con este parámetro, brindando resultados confiables.

En conclusión, el sistema es eficiente pues cumple tanto con la eficacia en la transmisión de datos como en la efectividad en la calidad y fiabilidad de las mediciones, a pesar de la limitación en la automatización del proceso.

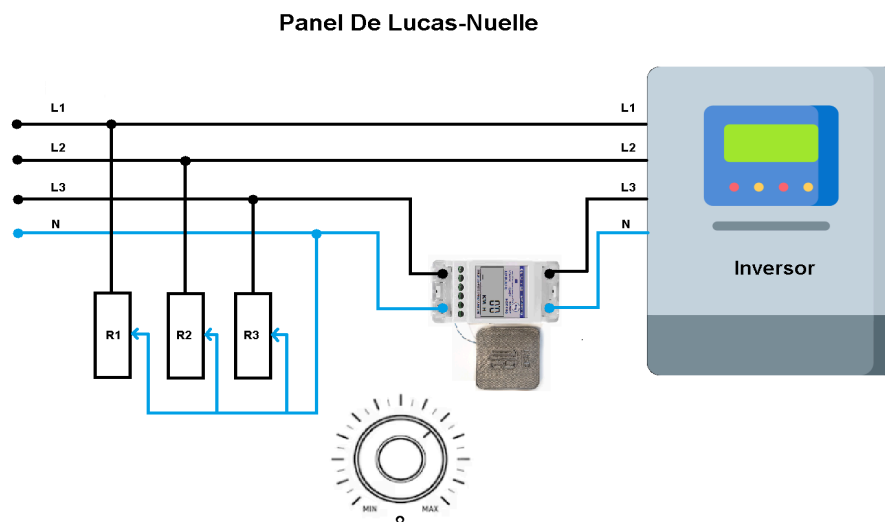
#### 4.4. Pruebas prácticas Simulando un Sistema de Paneles Fotovoltaicos On-Grid

Las pruebas de simulación del panel fotovoltaico con configuración On-grid se realizaron con el Simulador Lucas Nülle haciendo la conexión del dispositivo como se observa en la Figura 27, siguiendo el esquemático de la Figura 26.

Para este caso, se puede ver que el medidor está ubicado a la salida del inversor. En la prueba inicial, se tiene una tensión de 237 [V], la potencia de entrada entregada por el emulador del panel solar al inversor es de 1397 [W] y la potencia de salida del inversor es de 1365 [W], que es la que se entrega al medidor. Como se está midiendo solo la potencia de una fase, se tiene una potencia de 455,3 [w] con el medidor. Esta variable también se utiliza para hacer el cálculo del ahorro de energía en pesos colombianos que se pueden observar en la Figura 29.

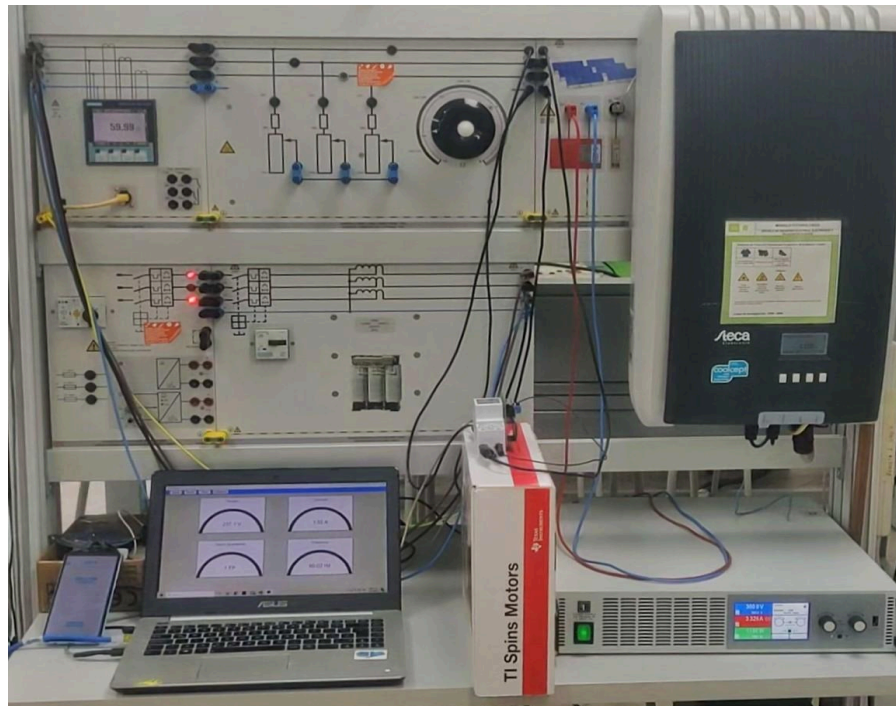
**Figura 26**

*Esquemático de conexión del dispositivo con el Lucas-Nülle*



**Figura 27**

*Sistema Lucas-Nülle, Simulador de sistema On-grid*



*Nota.* Foto tomada en el Laboratorio de Uso Racional de la Energía y Energías Renovables.

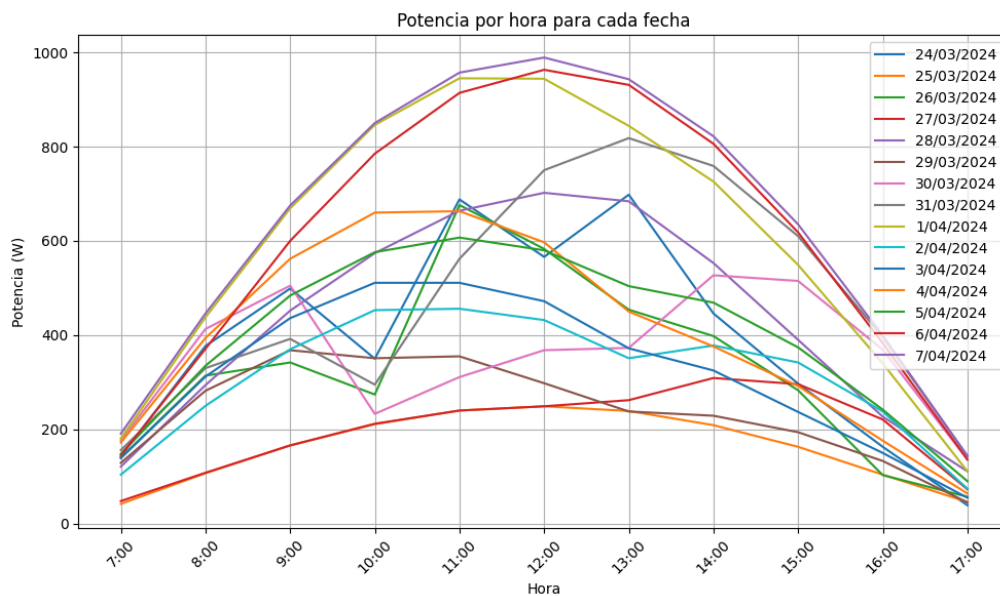
Para simular el comportamiento del panel solar y tomar datos a lo largo del tiempo, se extrajeron predicciones de radiación solar hora a hora durante el día (*Radiación Solar En Bucaramanga (Colombia) - Energía Solar, 2024*). El intervalo de los cambios entre las horas simuladas fue de cada 10 minutos, aumentando o disminuyendo de forma escalonada hasta alcanzar el siguiente valor de la predicción de radiación solar. De esta forma se buscó simular el comportamiento dinámico de un panel solar a lo largo del día, desde las 7 a.m. hasta las 4 p.m.

- **Resultados de las mediciones**

A continuación, en la Figura 28 se muestran los valores programados en el simulador de paneles solares en el cual se puede apreciar el comportamiento de las predicciones de la radiación solar en el área metropolitana de Bucaramanga.

**Figura 28**

*Datos Programados en el Simulador de Panel solar.*



*Nota.* Los datos son puntos unidos para ver un comportamiento a lo largo del día, estos datos muestran el comportamiento predicho de la realización solar para el área de Bucaramanga en el plazo de las fechas vistas en las leyendas, en el siguiente [enlace](#) se obtuvieron estas predicciones (*Radiación Solar En Bucaramanga (Colombia) - Energía Solar, 2024*).

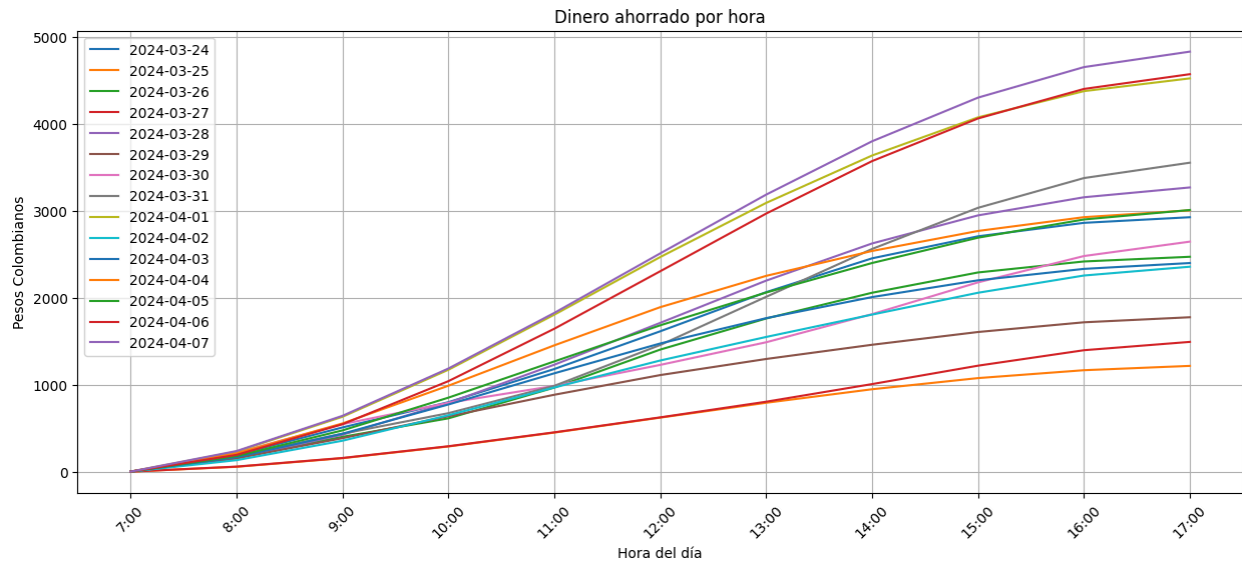
Idealmente el comportamiento de generación de energía debe percibirse como una parábola con su pico más alto entre las 10 y 13 de día, en la gráfica se puede observar que habrán

días con un comportamiento similar en la generación de energía, mientras que otros mostrarán fluctuaciones, esto debido a cambios climáticos como nubes u otros fenómenos que pueden afectar la generación de energía de los paneles solares.

En la Figura 29 se visualiza el comportamiento del dinero ahorrado al usar un sistema de paneles solares, para esto se toma la energía que entrega el inversor en una fase y el precio por kWh descrito en la Sección 3.2., en este caso de 796,38 COP. Se puede observar cómo a medida que hay mayor radiación del sol, aumenta rápidamente el ahorro que se muestra por día.

**Figura 29**

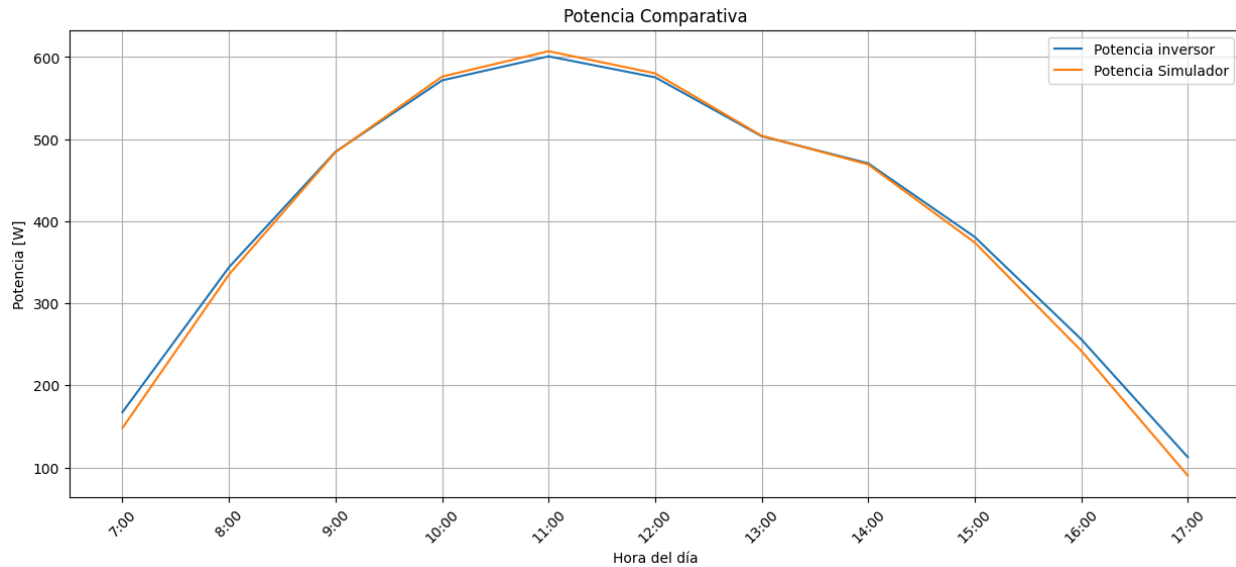
*Comportamiento del dinero ahorrado a lo largo del día.*



En la siguiente gráfica de la Figura 30 se puede ver el comportamiento de la simulación de la potencia de un panel solar en un día. En esta se hace la comparación de la potencia entregada por el emulador y la medida por el dispositivo del proyecto.

**Figura 30**

*Simulación del Comportamiento de un Panel Fotovoltáico en un día.*



**4.5 Prueba de bidireccionalidad del medidor**

Para probar la bidireccionalidad del medidor, se realizó una prueba utilizando la conexión realizada en la Figura 22. En esta prueba se hicieron mediciones tanto de la potencia directa (W\_FD) como de la potencia inversa (W\_FI), alternándolas en intervalos de tiempo que permitieran observar su comportamiento. Durante la prueba, se dejaron variables estáticas desde la fuente de tensión que simula la red eléctrica, en este caso, una tensión de 243 [V], corriente de 0,41 [A] y potencia de 100 [W].

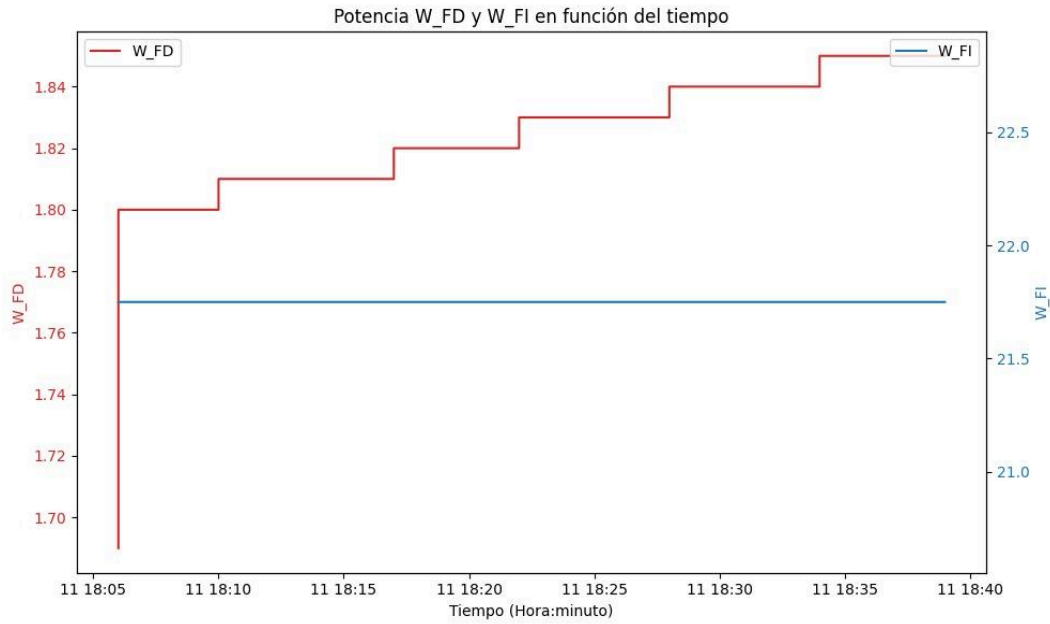
De esta forma, por media hora se registraron los datos midiendo la potencia directa. Después, se cambió la dirección de la inyección de energía en el simulador, simulando estas mismas variables desde el inductor del emulador del panel solar, midiendo así la potencia inversa por otra media hora.

El correcto funcionamiento esperado durante la prueba se puede ver en la Figura 31.

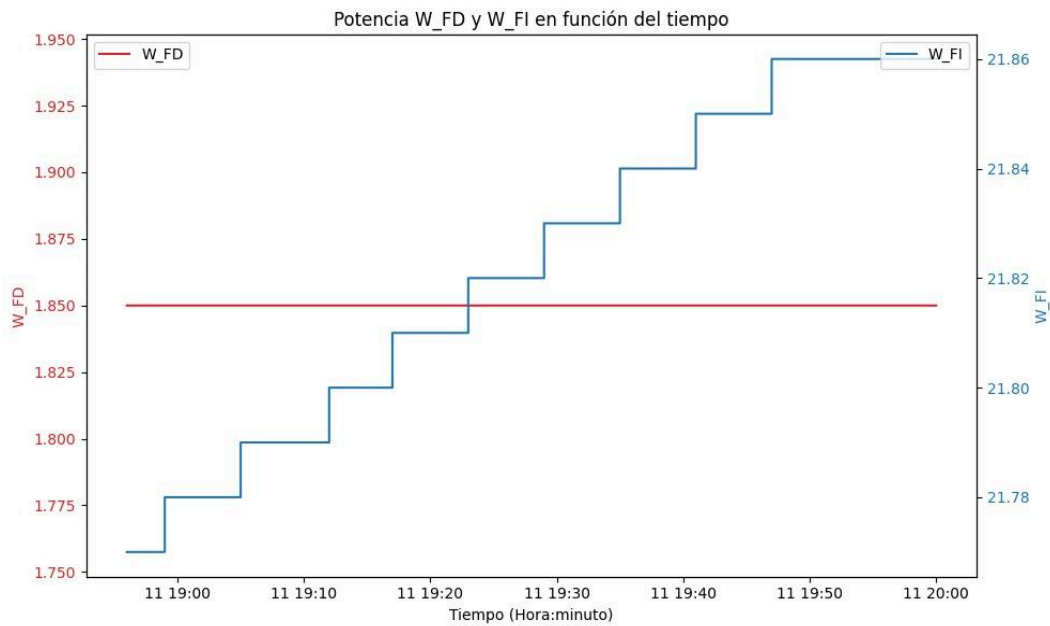
**Figura 31**

*Comprobación de la medición Bidireccional*

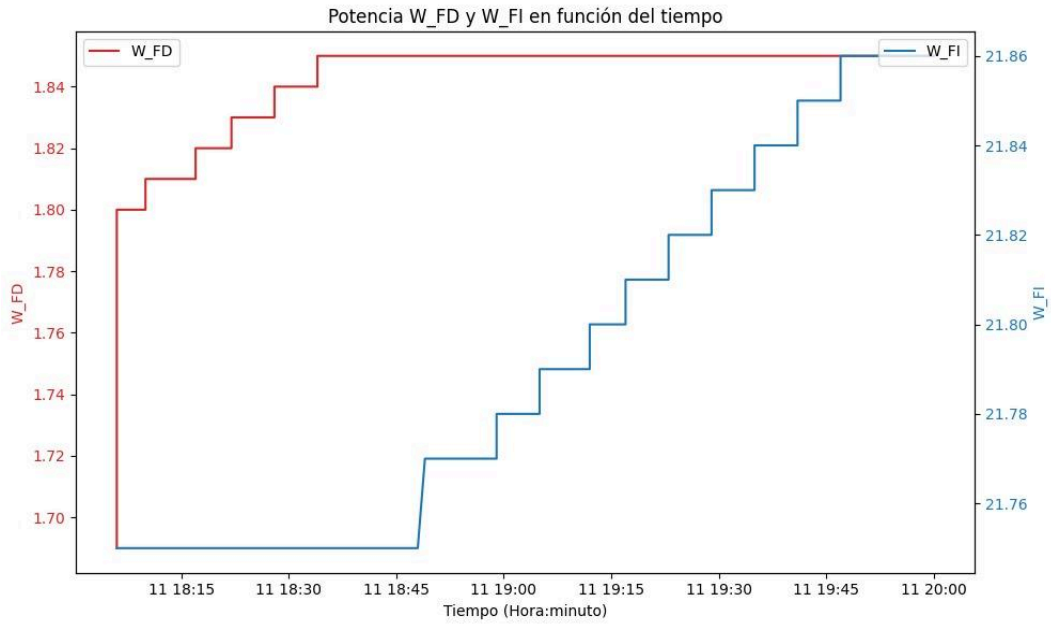
*a) Potencia Directa, b) Potencia Inversa, c) Comportamiento total en una una sola gráfica*



a)



b)



c)

*Nota.* En a) *Potencia Directa*, se puede ver que hay un aumento de Potencia FD (Forward) y no hay ninguna toma en la Potencia FI (Inverse), luego, en b) *Potencia Inversa*, se ve que hay un aumento de la Potencia FI pero la Potencia FD no realiza ninguna toma.

Finalmente en c) *Comportamiento total en una una sola gráfica*, se muestra la gráfica con toda la línea temporal, donde se ve con mejor claridad el comportamiento descrito anteriormente.

**4.6. Presupuesto total del diseño e implementación**

Se detallan los costos asociados al diseño e implementación del dispositivo, donde se incluyen los componentes que fueron necesarios para su correcto funcionamiento. A continuación, se presenta en la Tabla 13 que resume los detalles asociados a cada componente:

**Tabla 12**

*Detalles de los conceptos asociados al costo del dispositivo y el respectivo precio.*

<b>Componente</b>	<b>Precio (COP)</b>
ESP32 dev kit	27.000
Módulo Convertidor TTL a RS485	10.000
PCB	45.000
Caja Impresa 3D	34.000
Cinta doble Faz	10.000
Medidor DDS-669	97.850
<b>PRECIO TOTAL</b>	<b>223.850</b>

El costo total de la implementación del Prototipo MPB-231 fue de 223.850 COP. Esto incluye los elementos necesarios para la construcción del prototipo y los costos se basan en los precios más bajos del mercado que conseguimos, como se mencionó en un principio, buscando la economía en su desarrollo.

## 5. Conclusiones

- Se diseñó una arquitectura IoT eficiente, que permitió la comunicación entre dispositivos utilizando diferentes protocolos y tecnologías. Esto incluye la base de datos en tiempo real en la nube de Firebase, aprovechando su seguridad y privacidad integradas.  
Un entorno de desarrollo muy amplio como lo es el IDE de Arduino. Además, el protocolo Modbus, que es ampliamente usado en la industria por su flexibilidad de comunicación.
- Se utilizó un medidor de energía de bajo costo, el cual tenía muy poca información en la hoja de datos y configuración, pero con este proyecto logramos un buen funcionamiento de este. Por esto, ponemos a disposición los algoritmos desarrollados, la comunicación con Modbus y las direcciones de las variables medibles, especialmente las que están relacionadas con la energía bidireccional, facilitando así futuros proyectos en este campo.
- Aunque hay diferencias en las tomas de las medidas de tensión entre el medidor del simulador Lucas- Nülle y el medidor del prototipo, los resultados obtenidos en las mediciones son confiables y precisos.
- Se logró la visualización de los datos mediante el desarrollo de dos aplicaciones: una para sistemas operativos Windows y otra para sistemas Android. Estas aplicaciones, diseñadas con un enfoque en el usuario, ofrecen una visualización fluida y accesible de los datos, cumpliendo así con las necesidades planteadas para el proyecto.

## 6. Recomendaciones

- La interfaz de la aplicación móvil es clara, con más tiempo, se recomienda ofrecer al usuario más funciones dependiendo de las necesidades que tenga. Por ejemplo, gráficas de barras, indicadores de nivel y demás elementos visuales que puedan mostrar una interfaz más dinámica.
- Automatizar la recopilación del valor del costo del kWh cada mes, se recomienda desarrollar un programa que recopila automáticamente este dato. Esto simplifica el proceso para el usuario y garantiza la actualización constante del costo mensual en el sistema.
- Considerar buscar un servicio que tenga un servidor más cercano a la zona donde se instalará el dispositivo. Esto podría mejorar la transferencia de datos al reducir la latencia y mejorar la conectividad en general.

### Referencias Bibliográficas

- Alguero, M. O. (2023, March 7). En Colombia ya hay 38 proyectos de energía solar en funcionamiento, mientras que Santander prepara su parque solar. *Vanguardia*.  
<https://www.vanguardia.com/economia/local/2023/03/07/en-colombia-ya-hay-38-proyectos-de-energia-solar-en-funcionamiento-mientras-que-santander-prepara-su-parque-solar/>
- AliExpress. (n.d.). *Medidor de energía monofásico con DDS238-2 RS485, MODBUS-RUT de corriente de voltaje de 100a, 220V, 230V, 50HZ, 60HZ, ZN/S*. AliExpress. Retrieved November, 2023, from <https://es.aliexpress.com/i/32922071523.html>
- Centro de formación técnica para la industria. (n.d.). *Modbus: Qué es y cómo funciona | Comunicaciones Industriales*. aula21. Retrieved April 7, 2024, from <https://www.cursosaula21.com/modbus-que-es-y-como-funciona/>
- CITE Energía & Gonzales, V. (2020, 12). *Diferencias Entre El Sistema On Grid Y Sistema Off Grid Para La Generación De Energía*.
- CREG. (2021, octubre 7). *Alejandro - Resolución 174 de 2021 CREG*. Gestor normativo CREG. Retrieved May 3, 2024, from [https://gestornormativo.creg.gov.co/gestor/entorno/docs/resolucion\\_creg\\_0174\\_2021.htm](https://gestornormativo.creg.gov.co/gestor/entorno/docs/resolucion_creg_0174_2021.htm)
- electronilab. (n.d.). *Módulo Conversor TTL a RS485 MAX485*. Electronilab. Retrieved April 7, 2024, from <https://electronilab.co/tienda/modulo-conversor-ttl-rs485-max485/>
- Emergente. (2020, August 13). *Preguntas frecuentes acerca de la energía solar fotovoltaica en Colombia*. Emergente energía sostenible.  
<https://www.emergente.com.co/blog/preguntas-frecuentes-acerca-de-la-energia-solar-fotovoltaica-en-colombia/>

firebase de google. (2024, January 17). *SDK y bibliotecas cliente | Firestore | Firebase*. Firebase.

Retrieved April 7, 2024, from

<https://firebase.google.com/docs/firestore/client/libraries?hl=es-419>

*Fórmula tarifaria y tarifas - Consultar tarifas*. (2024). ESSA.

<https://www.essa.com.co/site/mi-factura/formula-tarifaria-y-tarifas/consultar-tarifas>

IERC. (2014). *Internet of Things*. European Research Cluster on the Internet of Things.

INELDEC. (2021). *¿De qué están hechos los paneles solares? - Paneles Solares Colombia*

Lamparas Solares Ingeniería Eléctrica Energía Solar Baterías Solares Cucuta Materiales.

<https://ineldec.com/de-que-estan-hechos-los-paneles-solares-fotovoltaicos/>

Kithion, B. (2016, April 6). *Different protocols in IOT – learniot*. learniot. Retrieved 2024, from

<https://learniot.wordpress.com/2016/04/06/different-protocols-in-iot/>

Laghari, A, Wu, K, & Laghari, R. (2022). A Review and State of Art of Internet of Things (IoT).

*rch Computat Methods*.

lucas-nuelle. (2024). *Lucas Nülle - Photovoltaics*. Lucas-Nuelle. Retrieved April 8, 2024, from

<https://www.lucas-nuelle.us/2768/apg/15974/Photovoltaics.htm>

Mercado Libre. (n.d.). *Tarjeta De Desarrollo Esp32 Wifi Bluetooth | Cuotas sin interés*. Mercado

Libre. Retrieved Noviembre, 2023, from

[https://articulo.mercadolibre.com.co/MCO-585538233-tarjeta-de-desarrollo-esp32-wifi-bluetooth-\\_JM#position=3&search\\_layout=stack&type=item&tracking\\_id=e8c8e5b7-9fb7-4c48-91b6-542b44efc17e](https://articulo.mercadolibre.com.co/MCO-585538233-tarjeta-de-desarrollo-esp32-wifi-bluetooth-_JM#position=3&search_layout=stack&type=item&tracking_id=e8c8e5b7-9fb7-4c48-91b6-542b44efc17e)

*Monitoreo de Potencia*. (n.d.). Realtime Database. Retrieved April 3, 2024, from

<https://console.firebase.google.com/u/0/project/sensortemp-var/database/sensortemp-var-default-rtdb/data/>

Moroney, L. (2017). *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform*. Apress.

Priharti, W., Rosmawati, A. F., & Wibawa, I. D. (2019). IoT based photovoltaic monitoring system application. *International Conference On Engineering, Technology and Innovative Researches*.

*Privacidad y seguridad en Firebase*. (2023, Febrero 28). Firebase.

<https://firebase.google.com/support/privacy?hl=es-419>

*Protocolo RS485 vs Modbus. Tutorial de la comunicación RS485*. (2021, October 12). Eltima.

Retrieved 2024, from <https://www.eltima.com/es/article/modbus-vs-rs485/>

*Radiación solar en Bucaramanga (Colombia) - Energía solar*. (2024). El Tiempo.

<https://www.tutiempo.net/radiacion-solar/bucaramanga.html>

Rey, G. H. (2023, April 17). *El 12 % de la energía mundial de 2022 provino del sol y del viento*.

ELTIEMPO.com. Retrieved May 2, 2024, from

<https://www.eltiempo.com/economia/sectores/el-12-de-la-energia-mundial-de-2022-provino-del-sol-y-del-viento-759803>

Ruiz, J. (n.d.). *Cargas y Reostatos DL 1017 | PDF*. Scribd. Retrieved April 3, 2024, from

<https://es.scribd.com/doc/227477316/Cargas-y-Reostatos-DL-1017>

Shopee. (2017, November 9). *Multifunción 230V 60A Medidor De Energía Bidireccional*

*Importación Exportación KWH Solar PV*. Shopee. Retrieved Noviembre, 2023, from

[https://shopee.com.co/product/593962443/17394226554?d\\_id=ba8b6&utm\\_content=3NBQKYWHArbAF5Xvkkhv2e7YSpY7](https://shopee.com.co/product/593962443/17394226554?d_id=ba8b6&utm_content=3NBQKYWHArbAF5Xvkkhv2e7YSpY7)

SunSupply. (2021, Noviembre 5). *Como funcionan los sistemas de energía solar on grid*.

SunSupply. Retrieved 2024, from

<https://www.sunsupplyco.com/como-funcionan-los-sistemas-de-energia-solar-on-grid/>

tutiempo.net. (n.d.). *Radiación solar en Bucaramanga*. Radiación solar en Bucaramanga.

Retrieved 03 1, 2024, from <https://www.tutiempo.net/radiacion-solar/bucaramanga.html>

UIT. (2012, Junio). *Infraestructura Mundial De La Información, Aspectos Del Protocolo Internet*

*Y Redes De La Próxima Generación* [Descripción general de Internet de los objetos].

## **Apéndices**

**Apéndice A.** Manual de conexión y programación del medidor DDS-669 con el ESP32 .

**Apéndice B.** Principios de diseño de la interfaz de usuario