



Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería en Telecomunicaciones

**INTEGRACIÓN DE LOS INVERSORES SUNNY ISLAND Y AMPERE SQUARE S
AL SISTEMA SCADA DEL LABORATORIO DE MICRO-RED DE LA
UNIVERSIDAD DE CUENCA**

Trabajo de titulación previo a la
obtención del título de Ingeniero
en Telecomunicaciones

Autores:

Santiago Ismael Araujo Ochoa
Joe Fernando Tigre Quituzaca

Director:

Darwin Fabian Astudillo Salinas
ORCID:  0000-0001-7644-0270

Cuenca, Ecuador

2023-08-28

Resumen

En la actualidad, los sistemas de energía renovable brindan la oportunidad de aprovechar recursos como la radiación solar. Para lograr una gestión eficiente de los dispositivos y sistemas que generan energía utilizando este tipo de recursos, es imprescindible que los operadores cuenten con información. Por tanto, para tener esta información es importante monitorear estos equipos, tanto en entornos industriales como en hogares. Una plataforma comúnmente utilizada para este propósito es conocida como Supervisory Control and Data Acquisition (SCADA).

En este contexto, el objetivo de este trabajo de titulación es integrar los inversores *Sunny Island* y *Ampere Square* al sistema de monitoreo SCADA existente en el laboratorio de Micro-Red de la Universidad de Cuenca. En el caso del inversor *Sunny Island*, debido a que no cuenta con un servidor Modbus integrado, se desarrolló un *Gateway* en el que se integra este servidor y un sistema de comunicación que permite leer y escribir información en el inversor usando el protocolo *SMA Data*. De esta manera, el sistema SCADA diseñado para este inversor puede leer los datos desde el *Gateway*. En cuanto al inversor *Ampere Square*, dado que ya cuenta con una interfaz de comunicación Modbus TCP, se realizó su integración directa al sistema SCADA.

Finalmente, se llevó a cabo una validación de los datos obtenidos en comparación con la información real desplegada por cada inversor. Los resultados confirmaron que la integración de cada inversor al sistema SCADA del laboratorio de Micro-Red de la Universidad de Cuenca se logró exitosamente.

Palabras clave: Inversor, SCADA, Raspberry Pi, Modbus, SMA Data



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Repositorio Institucional: <https://dspace.ucuenca.edu.ec/>

Abstract

Currently, renewable energy systems provide the opportunity to harness resources such as solar radiation. Operators need to have information to achieve efficient management of devices and systems that generate energy using these types of resources. Therefore, to have this information, monitoring these devices in both industrial and residential environments is important. A commonly used platform for this purpose is known as Supervisory Control and Data Acquisition (SCADA).

In this context, the objective of this work was to integrate the *Sunny Island* and *Ampere Square* inverters into the existing SCADA monitoring system in the Microgrid laboratory at the University of Cuenca. In the case of the *Sunny Island* inverter, as it does not have a built-in Modbus server, a *Gateway* was developed to integrate this server and a communication system that allows reading and writing information from the inverter using the *SMA Data protocol*. This way, the SCADA system designed for this inverter can read the data from the *Gateway*. As for the *Ampere Square* inverter, since it already has a Modbus TCP communication interface, it was directly integrated into the SCADA system.

Validation of the obtained data was carried out in comparison with the real information displayed by each inverter. The results confirmed that each inverter's integration into the Microgrid laboratory's SCADA system at the University of Cuenca was successfully achieved.

Keywords: *Inverter, SCADA, Raspberry Pi, Modbus, SMA Data*



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Índice

1. Generalidades	11
1.1. Introducción	11
1.2. Objetivos	11
1.2.1. Objetivo general	11
1.2.2. Objetivos específicos	11
1.3. Alcance	12
1.4. Justificación	12
1.5. Antecedentes del proyecto	13
2. Marco Teórico	14
2.1. Inversor	14
2.1.1. Ampere Square Pro	14
2.1.2. Sunny Island	16
2.2. Sistemas SCADA	17
2.3. Modbus	18
2.3.1. Modbus TCP	20
2.4. Yet Another SMA Data Implementation (YASDI)	21
2.5. Trabajos Relacionados	23
3. Integración de los inversores al sistema SCADA	26
3.1. Arquitectura	26
3.2. Sistema SCADA	27
3.3. <i>Gateway</i>	28
3.3.1. Interfaz <i>Gateway</i> -Inversor	29
3.3.2. Comunicación <i>Gateway</i> -Inversor, Yasdishell	31
3.3.3. Servidor Modbus	33
3.3.4. Servicios	35
3.4. Inversores	36

4. Resultados	37
4.1. Inversor <i>Sunny Island</i>	37
4.1.1. Comunicación <i>Gateway</i> -Inversor	37
4.1.2. Tiempos de respuesta del Gateway	38
4.1.3. Sistema SCADA	39
4.1.4. Validación de datos	41
4.2. Inversor <i>Ampere Square</i>	41
4.2.1. Sistema SCADA	41
4.2.2. Validación de datos	42
5. Conclusiones y recomendaciones	44
5.1. Conclusiones	44
5.2. Recomendaciones	44
Referencias	46
A. Anexo	48
A.1. Anexo A: Guía para la instalación, uso y verificación de funcionamiento del sistema construido para el inversor <i>Sunny Island</i>	48
A.2. Anexo B: Video tutorial para el proceso de compilación de los programas des- rollados y para solventar errores en el proceso.	48
A.3. Anexo C: Conexión entre el inversor Sunny Island y Raspberry Pi	48

Índice de figuras

2.1. Esquema Eléctrico Interno del inversor Ampere Square Pro. Fuente: [1]	15
2.2. Arquitectura de hardware típica. Fuente: [2]	17
2.3. Formato de mensaje Modbus simplificado. Fuente: [3]	18
2.4. Estructura de Modbus Serial. Fuente: [4]	19
2.5. Estructura de Modbus TCP. Fuente: [4]	21
2.6. Estructura de YASDI. Fuente: [5]	22
3.1. Arquitectura general del sistema.	26
3.2. Funcionamiento del sistema SCADA para los inversores.	28
3.3. Interfaces visuales del sistema SCADA de los inversores.	29
3.4. Funcionamiento del menú de configuración del sistema SCADA para el inversor <i>Sunny Island</i>	30
3.5. Interfaz visual del menú de configuración del sistema SCADA del inversor <i>Sunny</i> <i>Island</i>	31
3.6. Arquitectura del <i>Gateway</i>	31
3.7. Conexión física entre al inversor.	32
3.8. Diagrama de Conexión Física	33
3.9. Enlace físico entre el cable de conexión y el módulo RS485 a USB.	33
3.10. Principio de funcionamiento del programa <i>yasdishell</i>	34
3.11. Funcionamiento del programa <i>Server_V7</i>	35
4.1. Mensajes en consola de las aplicaciones.	37
4.2. Estado activo de los servicios.	38
4.3. Interfaz de supervisión para el inversor <i>Sunny Island</i>	39
4.4. Interfaz de configuración para el inversor <i>Sunny Island</i>	40
4.5. Valores del tablero de control del inversor.	41
4.6. Interfaz de supervisión para el inversor <i>Ampere Square</i>	42
4.7. Aplicación móvil del inversor <i>Ampere Square</i>	43
A.1. Conexión con el puerto ComSmaIn	48
A.2. Conexión al puerto USB de la Raspberry Pi	49
A.3. Localización del puerto ComSmaIn en el inversor	49

Índice de tablas

2.1. Mapa de Registros Modbus típico. Fuente: [3]	19
2.2. Distribución de una trama que emplea RTU. Fuente: [4]	20
2.3. Relación entre el modelo OSI y Modbus TCP. Fuente: [4]	20
2.4. Estructura de la cabecera MBAP. Fuente: [4]	21
3.1. Tabla de conexión. Fuente: [1]	30

Agradecimientos.

En este importante momento de culminación de nuestro trabajo de titulación, queremos expresar nuestro profundo agradecimiento a todas aquellas personas que han contribuido de manera significativa en el desarrollo y éxito de este proyecto.

En primer lugar, queremos agradecer al personal del laboratorio de Micro-red de la Universidad de Cuenca, quienes nos brindaron su apoyo incondicional, su amplio conocimiento y todas las facilidades necesarias para llevar a cabo nuestra investigación. Su compromiso y dedicación fueron fundamentales para la obtención de resultados valiosos y la realización de experimentos en un entorno de laboratorio propicio.

Extendemos nuestro agradecimiento especial a nuestro tutor de tesis, Ing. Fabián Astudillo, por su orientación, paciencia y constante motivación a lo largo de todo el proceso de investigación. Su experiencia y conocimientos en el campo fueron clave para enriquecer nuestro trabajo y guiarnos en la dirección correcta. Agradecemos sinceramente su tiempo, su disposición para resolver nuestras dudas y su valiosa retroalimentación, que nos ayudó a perfeccionar nuestra investigación.

También deseamos expresar nuestro agradecimiento a todos los profesores que impartieron conocimientos a lo largo de nuestra formación académica. Sus enseñanzas, experiencias compartidas y su dedicación a la educación nos han inspirado y han sido la base sólida sobre la cual hemos construido este trabajo. Agradecemos su compromiso con nuestra formación y su contribución a nuestro desarrollo como profesionales.

Dedicatoria.

Este trabajo lo dedico principalmente a Dios por brindarme la fuerza para completar los objetivos propuestos. A mis padres, que con su constante apoyo, compañía y consejos, me ayudaron a sobrepasar todos los obstáculos vividos en la vida universitaria, su apoyo incondicional y aliento constante han sido una fuente de fortaleza y motivación. A mis hermanos, quienes siempre me dieron ánimos para seguir adelante y no rendirme. A toda mi familia, que siempre me estuvo dando ánimo. Muchas gracias a todos.

Santiago

Este trabajo está dedicado en primer lugar a mi madre, cuyo apoyo incondicional desde el primer día hasta el último ha sido fundamental en mi trayectoria académica. Su dedicación y sacrificio han sido los cimientos que me han permitido avanzar incluso en los momentos más difíciles. También quiero dedicar este trabajo a mi tía y su familia, quienes generosamente me acogieron y me brindaron un hogar durante toda mi estancia en esta ciudad. Su calidez y afecto han sido una fuente constante de apoyo y aliento. No puedo dejar de mencionar a mi hermana, quien ha estado a mi lado en cada paso del camino y ha sido un ejemplo a seguir en mi carrera. Su compañía y respaldo han sido un impulso invaluable en mi búsqueda del éxito académico. A todas las personas que han dejado una huella en mi camino académico, quiero expresar mi más profundo agradecimiento. Sus contribuciones, consejos y presencia han sido de gran valor en mi crecimiento personal y profesional. A cada uno de ustedes, dedico este trabajo con profunda gratitud y cariño. Siempre llevaré en mi corazón vuestro amor y apoyo mientras continúo avanzando hacia nuevas metas y desafíos en mi vida.

Fernando

Glosario

ADU Application Data Unit. 21

API Application Programming Interface. 23

CA Corriente Alterna. 14, 16

CC Corriente Continua. 14, 16

EMS Energy Management System. 14

FV Fotovoltaico. 14

IP Internet Protocol. 20, 21

LAN Local Area Network. 17

MBAP Modbus Application Protocol. 21

OSI Open Systems Interconnection. 22

PDU Packet Data Unit. 19, 21

PLC Programmable Logic Controller. 17, 18, 25

RTDB Real-time Database. 17

RTU Remote Terminal Unit. 19

SCADA Supervisory Control and Data Acquisition. 2–6, 11–13, 17, 24–31, 33, 34, 36–44

SMA System, Mess and Anlagentechnik. 21–23

SQL Structured Query Language. 23

TCP Transmission Control Protocol. 20, 21, 28, 33–37

USB Universal Serial Bus. 27, 30, 32, 35

YASDI Yet Another SMA Data Implementation. 4, 6, 21, 22, 31

1. Generalidades

1.1. Introducción

En la actualidad, la adopción de sistemas de energía renovable ofrece la oportunidad de aprovechar recursos como la energía eólica, la radiación solar, entre otros, para complementar o reemplazar los sistemas de producción tradicionales. En particular, en el contexto de la energía solar, es crucial monitorizar las distintas variables generadas por los dispositivos que utilizan esta fuente de energía, tanto en entornos industriales como en hogares.

Para lograr una eficiente administración de procesos en una variedad de sistemas, es crucial que los operadores cuenten con información actualizada sobre su estado. Es necesario desarrollar una interfaz de comunicación que conecte a los operadores con los sistemas. Una plataforma comúnmente utilizada para este fin es conocida como Supervisory Control and Data Acquisition (SCADA). Estos sistemas permiten obtener datos de dispositivos, sensores y equipos distribuidos en una infraestructura, brindando además control remoto y una monitorización en tiempo real del estado de los equipos involucrados.

1.2. Objetivos

1.2.1. Objetivo general

El objetivo de este trabajo de titulación es integrar los inversores *Sunny Island* y *Ampere Square* al sistema de monitoreo SCADA existente en el laboratorio de Micro-Red de la Universidad de Cuenca.

1.2.2. Objetivos específicos

- Elaborar un estado de la técnica enfocado en la integración de inversores hacia sistemas SCADA.
- Implementar una plataforma física que permita recuperar información desde los inversores usando el protocolo Modbus.
- Integrar los inversores hacia el sistema SCADA del laboratorio de Micro-Red, implementado en LabVIEW, mediante los dispositivos desarrollados.

1.3. Alcance

La investigación se centrará en la integración de los inversores *Ampere Square* y *Sunny Island* en el sistema SCADA de la universidad, específicamente en el laboratorio de Micro-Red. Se considerarán únicamente los inversores *Ampere Square* y *Sunny Island* presentes en las instalaciones de la universidad.

La metodología de la investigación constará de los siguientes pasos:

- Investigar y comprender el protocolo de comunicación Modbus TCP para establecer la integración de los inversores con el sistema SCADA.
- Configurar y programar los dispositivos Raspberry Pi con el fin de extraer los datos del inversor *Sunny Island* y publicarlos en el servidor Modbus TCP.
- Realizar pruebas y validar la integración de los inversores con el sistema SCADA. Se verificará la correcta comunicación, monitoreo y control de los parámetros relevantes.

A partir de este trabajo de titulación, se espera obtener una comunicación exitosa de los inversores *Sunny Island* y *Ampere Square*, con el sistema SCADA, ya que previo al desarrollo de este trabajo, estos dispositivos no estaban enlazados a dicho sistema. Esto permitirá el monitoreo en tiempo real y la modificación de los parámetros relevantes directamente desde el sistema SCADA. Esto se logrará mediante el uso de una Raspberry Pi como intermediario. De igual forma, se establecerá una solución funcional para la integración del inversor *Ampere* al sistema SCADA.

Con este trabajo se espera obtener un mayor control y eficiencia en la gestión de los inversores en cuestión, en el laboratorio de Micro-Red de la universidad, así como la posibilidad de tomar decisiones basadas en el monitoreo en tiempo real de los parámetros relevantes.

1.4. Justificación

La Universidad de Cuenca, en su laboratorio de Micro-Red ubicado en el campus Balzay, cuenta con equipos de generación eléctrica que utilizan fuentes renovables, como paneles solares, para producir energía. Esta energía generada debe ser gestionada para su uso en el laboratorio o su conexión a la red eléctrica de la ciudad. Los dispositivos encargados de convertir la corriente continua en corriente alterna se conocen como inversores. La mayoría de los equipos presentes en el laboratorio de Micro-Red fueron adquiridos durante el primer

proceso de canje de deuda y ya han sido integrados en el sistema SCADA desarrollado en LabVIEW. En un segundo proceso de adquisición, se incorporaron nuevos equipos que aún no han sido integrados en el sistema SCADA.

1.5. Antecedentes del proyecto

En los últimos años, se ha observado un creciente interés en la implementación de sistemas de monitoreo y control en el campo de la generación de energía renovable. Estos sistemas permiten supervisar y gestionar de manera eficiente los parámetros de los inversores, mejorando así el rendimiento y la productividad de las instalaciones. En este contexto, la integración de inversores con sistemas SCADA se ha convertido en un tema relevante de investigación.

En el laboratorio de la universidad, se cuenta con inversores *Ampere Square* y *Sunny Island* que desempeñan un papel crucial en la generación y distribución de energía. Sin embargo, hasta el momento, no se ha logrado una integración completa de estos inversores con el sistema SCADA existente en la universidad. Esta limitación impide un monitoreo en tiempo real y un control eficiente de los parámetros relevantes de los inversores.

El presente proyecto de tesis busca abordar estas limitaciones y desarrollar una solución para la integración de los inversores *Ampere Square* y *Sunny Island* al sistema SCADA de la universidad. Se pretende lograr una comunicación exitosa, monitoreo en tiempo real y control eficiente de los inversores, lo que contribuirá a mejorar la gestión de la generación de energía renovable en el laboratorio de la universidad.

2. Marco Teórico

En este capítulo se abordarán todos los conceptos teóricos considerados de importancia, los cuales permitieron desarrollar el presente trabajo de titulación.

2.1. Inversor

Un importante elemento usado la transformación de la energía eléctrica es el inversor. El propósito de un inversor es el transformar una señal de voltaje que se encuentra en Corriente Continua (CC) a Corriente Alterna (CA). Para ello, se inyecta potencia hacia una carga, a una frecuencia establecida con el mayor factor de potencia posible [6]. Se pueden hallar inversores de frecuencia de 50 Hz, ya sean monofásicos o trifásicos, que pueden operar en dos modos de conexión: conectados a una batería o enlazados directamente a un generador Fotovoltaico (FV). Los inversores pueden estar conectados a dispositivos que consuman energía eléctrica o pueden inyectar esta energía a la red [7].

El inversor desempeña un papel fundamental en el sistema de generación FV. En este sistema, los módulos FV no están directamente conectados a la red eléctrica, sino que requieren de un inversor como elemento intermedio. El inversor tiene dos funciones principales: la primera consiste en cargar el módulo fotovoltaico para capturar la máxima cantidad de energía disponible, mientras que la segunda es inyectar un flujo de corriente sinusoidal en la red pública [8]. Un inversor fotovoltaico típico contiene varios componentes electrónicos, como por ejemplo capacitores de enlace de datos, transformador, tarjetas de circuito de control y relés de contactores eléctricos [9].

2.1.1. Ampere Square Pro

El *Ampere Square Pro* es un sistema de almacenamiento energético todo en uno basado en baterías de litio. Este sistema incluye la batería, el inversor/cargador y el Energy Management System (EMS). El inversor es del tipo bidireccional híbrido. Este dispositivo permite la integración con sistemas de generación de energía renovable tanto en CA como en CC. Para lograrlo, se requiere la instalación de un medidor de energía adicional o el uso de dos pinzas de un medidor de energía trifásico para medir dos puntos. Es importante destacar que el equipo puede ser instalado tanto en una configuración monofásica como en una trifásica. En la Figura 2.1 se detalla el esquema eléctrico interno del equipo [1]. A continuación se describe

cada componente:

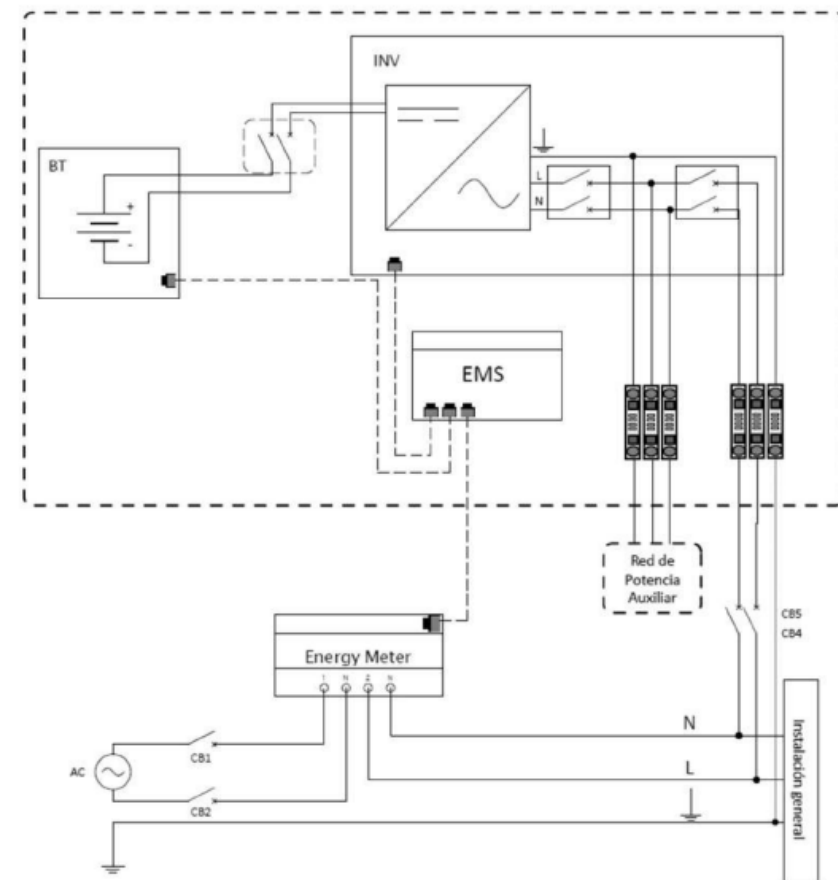


Figura 2.1: Esquema Eléctrico Interno del inversor Ampere Square Pro. Fuente: [1]

- **BT:** batería de 3 o 6 kWh, según el modelo, la cual es responsable de almacenar energía.
- **INV:** inversor bidireccional híbrido; este realiza la conversión entre las tensiones continuas (batería e instalación fotovoltaica) y alternas (red eléctrica principal y *Backup*). Además, gestiona el flujo de energía entre los diferentes componentes.
- **EMS:** Es el sistema de gestión energética. Controla la estrategia de carga y descarga del equipo según los hábitos de consumo del usuario, del precio variable de la energía y de la generación de la instalación fotovoltaica.
- **Energy Meter:** Es el sensor principal del sistema, necesario para la regulación del funcionamiento del mismo [1].

El inversor posee información de interés como por ejemplo el nivel de potencia consumida, el voltaje, la independencia energética, el nivel de carga de la batería, la producción solar, entre otros. Además, para la comunicación de la información implementa el protocolo Modbus [1].

2.1.2. Sunny Island

El dispositivo Sunny Island es un inversor bidireccional diseñado para sistemas aislados. Actúa como inversor con batería y cargador de batería. En el lado de la red aislada, el Sunny Island suministra energía a los equipos consumidores. Además, puede cargar las baterías utilizando la energía proporcionada por los equipos que inyectan en el lado de CA.

Los sistemas formados por el Sunny Island ofrecen la máxima flexibilidad porque se pueden conectar en CA y CC, y además, son fácilmente ampliables. El Sunny Island alcanza un rendimiento máximo de más del 95 %. La capacidad de sobrecarga y la gestión de potencia integrada hacen que resulte innecesario sobredimensionar el Sunny Island.

El Sunny Island puede operar con hasta tres equipos en un sistema monofásico o con hasta cuatro equipos en un sistema monofásico de tres conductores, lo que le permite crear redes aisladas con una potencia que varía desde 2 kW hasta 24 kW. En sistemas multiclúster, esta capacidad puede ampliarse hasta 100 kW. Además, el Sunny Island cuenta con una función de desconexión automática de cargas en caso de que la batería no proporcione suficiente energía eléctrica.

El Sunny Island realiza una monitorización óptima de la batería, que es el componente más delicado en los sistemas aislados. Gracias a su regulación de carga inteligente y su protección segura contra descargas profundas, este dispositivo contribuye a prolongar significativamente la vida útil de la batería en comparación con otros equipos más básicos. Además, el Sunny Island dispone de una tarjeta SD para el almacenamiento de datos y eventos.

El Sunny Island cuenta con un procedimiento de protección integrado contra el funcionamiento en isla para evitar la formación accidental de redes aisladas en la red pública. Si se activa este proceso, el equipo cambia por completo y sin interrupción al funcionamiento en red aislada [10]. El inversor posee un puerto de comunicación con el cual se puede entablar una comunicación con dispositivos externos. Para ello, es necesario emplear un cable RJ45, en el cual, un extremo es conectado al inversor, mientras que el otro se encuentra separado en tres cables para la asignación de la señal RS485, que debe ser empleada por el dispositivo externo. La velocidad de transmisión de datos es 1200 bps [10].

2.2. Sistemas SCADA

Supervisory Control and Data Acquisition (SCADA) es un sistema que no se centra explícitamente en el control, sino en un nivel de supervisión. Es un paquete de software colocado sobre el hardware que se desea monitorear. Generalmente, en la conexión de estos sistemas, se hace uso de Programmable Logic Controllers (PLCs). A continuación, se describe la arquitectura y ciertas características comunes de los sistemas SCADA [2].

- **Hardware:** En la Figura 2.2 se puede apreciar la arquitectura de Hardware, en la cual se pueden diferenciar dos capas. La primera, es la capa del cliente; esta se encarga de la interacción hombre-máquina. La segunda capa es la de datos, la cual se encarga de la mayoría de actividades de control y procesamiento de información. En dicha arquitectura, se observa que los clientes se enlazan con cada servidor a través de una conexión Ethernet, lo cual implica que ambos elementos se encuentran en la misma red Local Area Network (LAN). Adicionalmente, se observan dos tipos de servidores. El *Servidor de Datos* es aquel que se encuentra en contacto con los controladores de los dispositivos que se estén monitorizando [2].

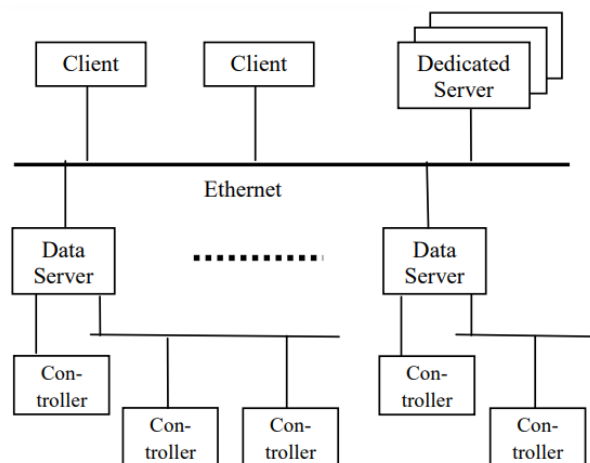


Figura 2.2: Arquitectura de hardware típica. Fuente: [2]

- **Software:** Los productos, o aplicaciones desarrolladas en un sistema SCADA, son multi-tarea, y están basados en una Real-time Database (RTDB) localizada en uno o más servidores. En estos últimos se realizan las actividades de control y configuración de parámetros [2].

Un entorno para el despliegue de sistemas SCADA es LabVIEW. Este es un lenguaje de programación gráfico, empleado como una herramienta estándar en el desarrollo de sistemas

de instrumentación virtual. Posee una amplia variedad de librerías, además que es posible la interacción con todo tipo de protocolos de comunicación.

La instrumentación virtual desarrollada en LabVIEW, facilita el control y monitoreo de variables físicas, junto con un procesamiento interno que permite la visualización de la información en un formato deseado [11].

2.3. Modbus

Modbus es un protocolo que tiene una gran apertura debido a que es libre y de fácil comprensión. Modbus es un protocolo maestro-esclavo. El protocolo permite solo un maestro, y hasta 247 esclavos. El único dispositivo capaz de iniciar la comunicación es el maestro. Los esclavos pueden ser identificados con números que se encuentren en el rango de 1 a 247. El dígito 0 está reservado para transmisiones en *broadcast*. Existen comandos específicos que son emitidos por el maestro y por el esclavo. El comando emitido por el maestro es llamado Query (Consulta), mientras que la respuesta por parte del esclavo se denomina Response (Respuesta) [3].

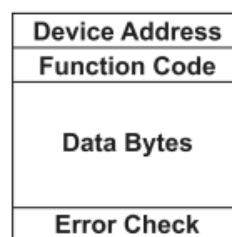


Figura 2.3: Formato de mensaje Modbus simplificado. Fuente: [3]

En la Figura 2.3 se presenta la estructura simplificada de los mensajes que se envían, tanto del tipo Query como Response. El maestro no posee dirección, por lo que ese campo es reservado para la dirección del esclavo en los dos posibles mensajes. Es decir, si el mensaje es una petición (Query), este mensaje es dirigido a un esclavo que posee la dirección de dispositivo asignada. En caso de una respuesta, esta llega al maestro desde la dirección de dispositivo asignada [3]. Al estudiar el protocolo Modbus, es importante considerar sus registros de información. En la Tabla 2.1 se presenta el Mapa de Registros Modbus clásico. La columna *Rango* hace referencia a las direcciones de los registros. Los primeros PLCs empleaban únicamente entradas y salidas discretas, donde estas eran representadas por registros de un solo bit. Las salidas y entradas discretas comienzan en las localizaciones 00001 y 10001, respectivamente. Adicionalmente, se agregaron los registros que pueden almacenar entradas analógicas. El

tamaño máximo de los valores que pueden ser almacenados en estos registros es de 16 bits.

Rango	Descripción
00001 - 10000	Salidas Discretas de Lectura/Escritura
10001 - 20000	Entradas Discretas de Lectura
30001 - 40000	Registros de Entrada de Lectura (Input) - Registros de 16 bits para entradas analógicas
40001 - 50000	Registros de Retención de Lectura/Escritura (Holding) - Registros de 16 bits

Tabla 2.1: Mapa de Registros Modbus típico. Fuente: [3]

Existen dos variaciones del protocolo Modbus que pueden ser empleadas. Estas son: Modbus Serial y Modbus TCP. Modbus Serial se emplea para entornos de comunicación en serie. Es decir, es necesario emplear cables físicos. En la Figura 2.4 se presenta la estructura del Packet Data Unit (PDU). El Campo de Dirección contiene la dirección del esclavo. La dirección del maestro no es requerida ya que el protocolo permite solo un maestro. El campo Código de Función hace referencia al tipo de solicitud que se realiza, por ejemplo la lectura o escritura de datos. El campo Datos es la información que desea ser enviada, por ejemplo, en el proceso de escritura. Por último, el CRC es el Chequeo de Redundancia cíclica que permite verificar la integridad de la trama.

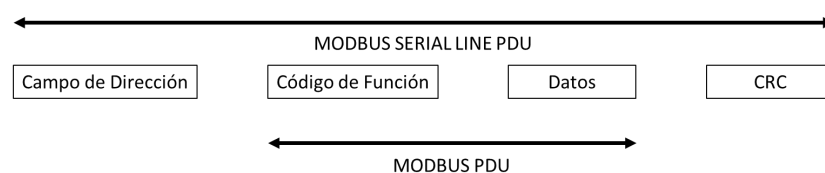


Figura 2.4: Estructura de Modbus Serial. Fuente: [4]

El modo de transmisión más empleado en Modbus Serial emplea Remote Terminal Unit (RTU). La trama que se forma al emplear este modo de transmisión se puede observar en la Tabla 2.2.

Dirección del esclavo	Código de Función	Datos	CRC
1 byte	1 byte	0 a 252 bytes	2 bytes

Tabla 2.2: Distribución de una trama que emplea RTU. Fuente: [4]

En función de los datos presentados en la Tabla 2.2, la trama más extensa que se puede formar ocupa 256 bytes. Al emplear RTU cada byte cuenta con sus ocho bits más tres bits. Estos tres son los bits de inicio, de parada y de paridad. Este último puede indicar paridad par o impar. Si la paridad no se emplea, se envía otro bit de parada [4].

2.3.1. Modbus TCP

El surgimiento de esta versión de Modbus está justificado en la necesidad de establecer conexiones a través de redes Internet Protocol (IP) [4]. Para lograrlo, emplea el protocolo de transporte Transmission Control Protocol (TCP). El modelo empleado por Modbus TCP se presenta en la Tabla 2.3.

Capa	Modelo OSI	Modbus Function
5, 6 y 7	Aplicación	Protocolo de Aplicación Modbus
4	Transporte	Protocolo de Control de Transmisión
3	Red	Protocolo de Internet
2	Enlace de Datos	IEEE 802.3
1	Física	IEEE 802.3

Tabla 2.3: Relación entre el modelo OSI y Modbus TCP. Fuente: [4]

En la implementación de Modbus TCP, se produce un cambio significativo al convertir el bus Modbus en un bus IP. Otro aspecto importante a destacar es el cambio del enfoque de maestro-esclavo a un paradigma cliente-servidor. En esta nueva configuración, el maestro se convierte en cliente y realiza solicitudes al servidor, es decir, al esclavo, y este último responde a las

solicitudes. Sin embargo, ya no existe un único maestro que se enlaza a los esclavos. Esto significa que cualquier número de clientes puede conectarse con cualquier número de servidores.

En la Figura 2.5 se presenta la modificación realizada al protocolo Modbus serial, para ser usado como Modbus TCP [4]. En esta se presenta la estructura del Application Data Unit (ADU) TCP/IP. Básicamente, el PDU, es la parte principal del ADU, y se mantiene igual a la de la variante serial de Modbus [4]. A esta se le añade una cabecera Modbus Application Protocol (MBAP), con las características observadas en la Tabla 2.4.

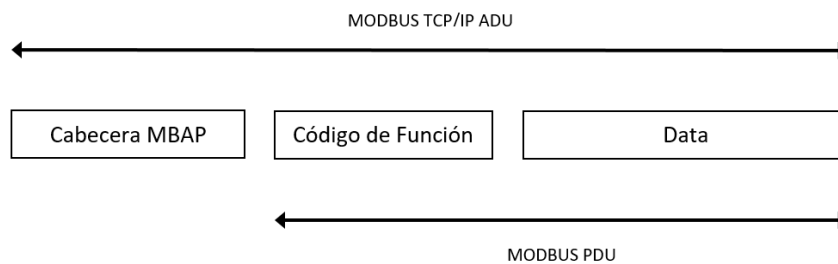


Figura 2.5: Estructura de Modbus TCP. Fuente: [4]

Unidad de Transacción	Identificador de Protocolo	Longitud	Identificador de Unidad
2 Bytes	2 Bytes	2 Bytes	1 Bytes

Tabla 2.4: Estructura de la cabecera MBAP. Fuente: [4]

En esta estructura, el *Identificador de Transacción* es proporcionado por el cliente y usado para mantener un registro de peticiones específicas. El *Identificador de Protocolo* Modbus es el cero. En la *Longitud* se detalla el tamaño del PDU y los campos restantes. El *Identificador de Unidad* provee la dirección del esclavo Modbus Serial, al cual debe ser accedido a través del *gateway*, es decir la conexión IP. Por último, el puerto TCP 502, ha sido asignado para la comunicación Modbus [4].

2.4. Yet Another SMA Data Implementation (YASDI)

YASDI es un programa diseñado para facilitar la comunicación con dispositivos System, Mess and Anlagentechnik (SMA). Actúa como un controlador sin interfaz gráfica, estableciendo la

comunicación con los dispositivos a través del protocolo SMA Data, que se transporta mediante SunnyNET o SMANet. Este software ha sido desarrollado para ser compatible con sistemas operativos como Windows o Linux y está escrito en lenguaje C. YASDI implementa la función de *maestro* en el protocolo SMA Data [5]. En la Figura 2.6 se presenta la estructura de funcionamiento del software.

Un punto importante a tomar en cuenta en esta implementación son los canales. Los canales son los lugares en los que se encuentran almacenados los valores de los parámetros de los dispositivos que emplean esta plataforma. En los canales se puede encontrar información del dispositivo, en este caso del inversor; como por ejemplo la potencia, voltaje, corriente, temperatura, nivel de batería, entre otros. Los canales, pueden ser divididos en dos tipos: *Spot Channels* y *Parameter Channels*. Los *Spot Channels*, son canales de información que solo pueden ser leídos, es decir, los valores almacenados en estos canales no pueden ser editados. Por otra parte, los *Parameter Channels*, poseen valores que pueden ser leídos y editados.

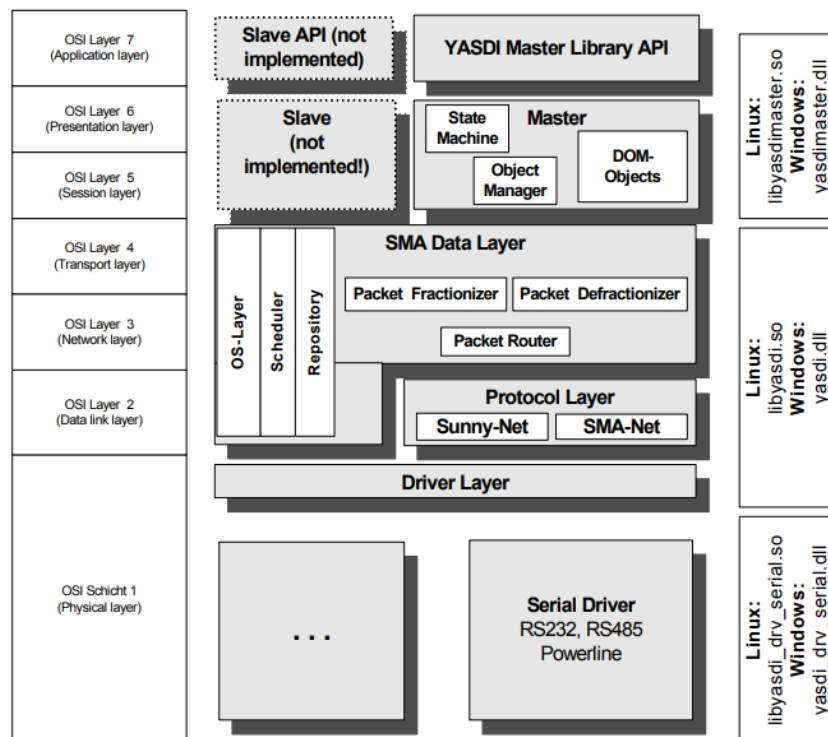


Figura 2.6: Estructura de YASDI. Fuente: [5]

La Figura 2.6 muestra como el programa se estructura en cada una de las capas del modelo Open Systems Interconnection (OSI).

- **Capa Física (Capa 1):** En esta capa, el software necesita que la interfaz de comunicación sea serial y ocupe el estándar de comunicación RS485. En Windows, se puede

establecer el enlace en los puertos COM1 a COM8, mientras que en Linux, se lo puede realizar con los puertos ttyS0 a ttyS7.[5].

- **Capa de Enlace de Datos (Capa 2):** Esta capa es responsable de procesar la trama de protocolo. En este punto, entra en juego los protocolos de transporte SunnyNet y SMA-Net. Esto son protocolos empleados en los dispositivos SMA. En el caso de SunnyNet, se emplea en dispositivos más antiguos que en los que se emplea SMANet [5].
- **Capa de Red/Transporte (Capa 3/4):** Esta capa implementa el protocolo *SMA Data*. Se tiene una Application Programming Interface (API) que simplifica el control de los diferentes dispositivos SMA. La capa incluye la encapsulación y re-ensamble de paquetes. Su mecanismo de enrutamiento asegura que los paquetes transmitidos lleguen a las interfaces correctas. La API es una interfaz asíncrona orientada a paquetes con mecanismos de *call-back*. Esta capa también implementa una interfaz que se abstrae de los comandos específicos del sistema operativo [5].
- **Capa de Sesión/Presentación (Capa 5/6):** En esta capa se implementa un maestro del tipo *SMA Data*. El maestro es capaz de detectar a los dispositivos SMA y solicitar los valores del canal. Tiene capacidad de leer y escribir parámetros. Para su funcionamiento, el maestro implementa una máquina de estados finitos [5].
- **Capa de Aplicación (Capa 7):** Describe la interfaz utilizada por el maestro. La API utiliza dispositivos y canales identificadores, a través de los cuales se puede acceder a cada uno de estos objetos [5].

2.5. Trabajos Relacionados

Para conocer el estado de la técnica, se revisaron diversos trabajos que implementan métodos de integración de datos de sistemas fotovoltaicos a sistemas de control.

El proyecto desarrollado por [12] consiste en la recopilación de datos de un sistema de generación eléctrico-solar utilizando una Raspberry Pi. La Raspberry Pi se comunica con el inversor a través del protocolo Modbus y los datos recopilados se envían a una base de datos Structured Query Language (SQL) a través de internet. Para habilitar la conectividad inalámbrica, se utiliza un módulo WiFi conectado a la Raspberry Pi. Entre los parámetros obtenidos del inversor se encuentran la potencia generada y los valores que indican fallos. Los autores establecieron que la lectura de los registros y la actualización de la base de datos se realicen cada cinco

minutos.

De igual manera, se implementó un sistema SCADA para el monitoreo de plantas fotovoltaicas, según se describe en el estudio realizado por [13]. En este estudio, se enfocaron en la implementación de mecanismos de seguridad. Para lograrlo, se emplearon dos técnicas específicas. En primer lugar, se aplicó una estrategia de control de acceso seguro, mientras que en segundo lugar se utilizaron mecanismos de redundancia. Los resultados obtenidos en este estudio fueron significativos. Se logró establecer métodos de autenticación sólidos, lo que garantiza la seguridad en el acceso a los datos. Además, se implementaron mecanismos que permitieron la gestión de los procesos en función de la prioridad del usuario que deseaba visualizar los datos, lo cual mejoró la eficiencia y el control en la gestión de las plantas fotovoltaicas.

En [14], se ha llevado a cabo un trabajo que aborda la implementación de un sistema de control de energía para una Micro-red residencial, en el cual la supervisión y adquisición de datos se realizan a través de un sistema SCADA. En este estudio se propone un sistema que tiene como objetivo monitorear el estado de funcionamiento de cada componente de la Micro-red. En la implementación de este sistema, se emplean protocolos de comunicación como WiFi y RS485 para facilitar el monitoreo. Específicamente, se utiliza un enchufe WiFi inteligente para realizar el monitoreo de los electrodomésticos y equipos de suministro de energía presentes en la Micro-red. Los datos de funcionamiento de estos dispositivos son cargados a la nube, permitiendo una gestión inteligente y la administración de energía en el hogar.

En un trabajo investigativo, se han identificado ciertas características clave de los microcontroladores utilizados en sistemas de medición de energía. En [15], se lleva a cabo una planificación de un proyecto enfocado en la exploración de innovaciones en el campo del SCADA para sistemas de adquisición de datos. En este contexto, es importante destacar que el enlace entre un sistema SCADA y el elemento del cual se obtiene la información suele ser establecido mediante el uso de un microcontrolador. En la selección del microcontrolador adecuado, se consideran aspectos como la capacidad de acoplamiento en la carcasa del equipo y el modo de alimentación, con el objetivo de garantizar una integración eficiente y confiable. En el caso específico de la toma de datos, los investigadores implementaron una red RS485 con sus periféricos correspondientes, y se incorporó una optoaislación para prevenir posibles fallas en el campo que podrían afectar al microcontrolador.

El avance tecnológico ha permitido la implementación de fuentes de energía renovable en diversas empresas. Como parte de una iniciativa para emular este entorno, se ha llevado a cabo un trabajo descrito en el artículo [16]. En dicho trabajo se ha desarrollado un inversor trifásico

diseñado para convertir la energía de voltaje continuo en una forma sinusoidal, con el fin de alimentar cargas que operan a una frecuencia de 60Hz. Además del desarrollo del inversor, se ha diseñado un sistema SCADA utilizando el software Movicon. Esta implementación permite seleccionar la fuente de energía primaria y visualizar los datos de funcionamiento del sistema. Para lograr que el sistema SCADA pueda controlar el dispositivo de forma remota, se establece una conexión punto a punto entre el PLC y el ordenador, demostrando así un rendimiento adecuado.

3. Integración de los inversores al sistema SCADA

Este capítulo presenta detalles sobre los elementos considerados durante la integración de los inversores *Sunny Island* y *Ampere Square* en el sistema Supervisory Control and Data Acquisition (SCADA) del laboratorio de Micro-Red de la Universidad de Cuenca. La estructura del capítulo se presenta de la siguiente manera: en la Sección 3.1 se expone la arquitectura utilizada para la integración de los inversores, en la Sección 3.2 se describe la metodología relacionada con el sistema SCADA, en la Sección 3.3 se detalla el proceso de desarrollo del *Gateway*, y finalmente, en la Sección 3.4 se presenta el proceso seguido por cada inversor en el sistema SCADA.

3.1. Arquitectura

En esta sección se presenta la arquitectura propuesta para la integración de los inversores. En la Figura 3.1, se muestra un esquema que ilustra la arquitectura establecida para la integración de los inversores.

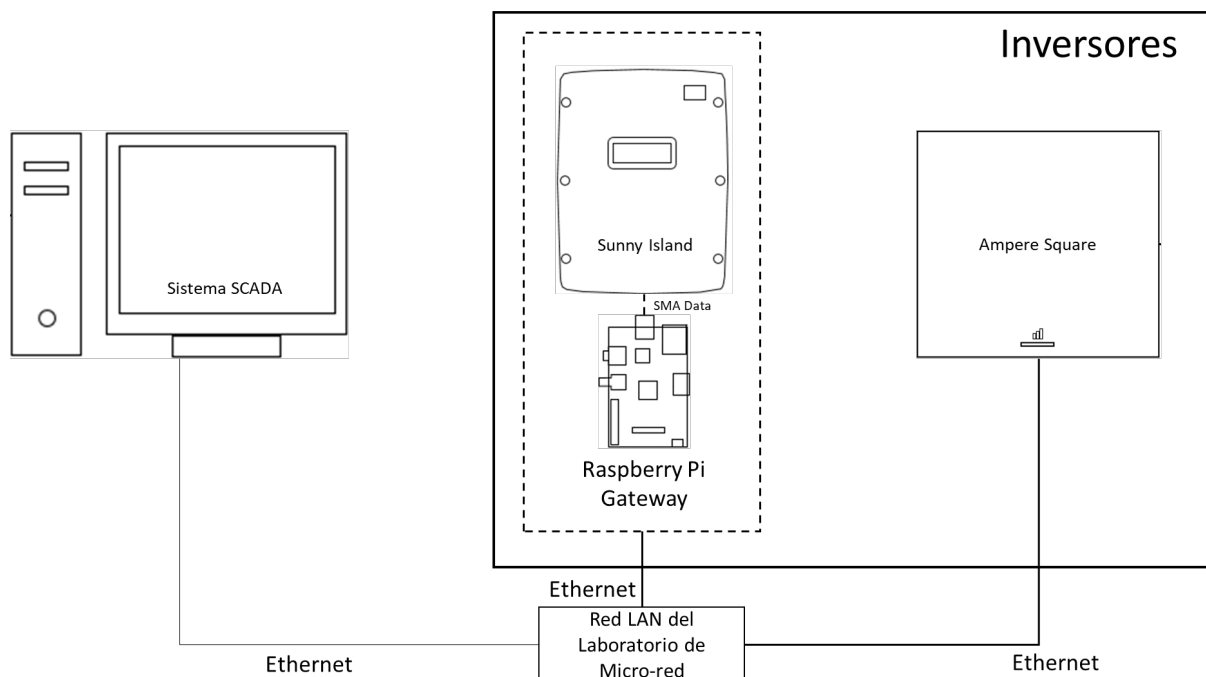


Figura 3.1: Arquitectura general del sistema.

En el caso del inversor *Sunny Island*, su interfaz de comunicación se conecta a un *Gateway* (Raspberry Pi), cuya implementación es parte de este trabajo, mediante un módulo RS485

a Universal Serial Bus (USB) empleando el protocolo *SMA Data*. El *Gateway* se encarga de realizar solicitudes al inversor para obtener los datos y luego los envía al sistema SCADA el cual emplea el protocolo Modbus TCP para leer los datos.

En el caso del inversor *Ampere Square*, su arquitectura es más sencilla, ya que se establece una conexión directa desde su interfaz Ethernet a la red del laboratorio de Micro-Red. El servidor Modbus implementado en este equipo permite adquirir los diferentes datos directamente en el sistema SCADA sin la necesidad de utilizar dispositivos intermediarios (*Gateway*), como en el caso del inversor *Sunny Island*.

3.2. Sistema SCADA

Para el desarrollo del sistema SCADA, se ha utilizado LabVIEW como herramienta de programación. Este programa se basa en el esquema de funcionamiento mostrado en la Figura 3.2.

En base a dicho funcionamiento, se ha desarrollado un diseño de la interfaz gráfica para el sistema SCADA de los inversores. Los diseños propuestos se muestran en la Figura 3.3. Es importante tener en cuenta que el inversor *Sunny Island* ofrece una mayor cantidad de datos disponibles en comparación con el inversor *Ampere Square*. Por esta razón, en la Figura 3.3(a) se pueden observar menos datos en comparación con la Figura 3.3(b).

El proceso se inicia con el establecimiento de la conexión desde el ordenador donde reside el sistema SCADA hacia el servidor Modbus, el cual tiene dos posibles estados: una conexión exitosa o una conexión fallida. En caso de no establecerse la conexión, el sistema entra en modo de espera hasta que se restablezca la conexión de manera manual mediante la interfaz. Por otro lado, si la conexión se establece correctamente, se inicia el monitoreo de los datos de funcionamiento del inversor.

Además, el sistema SCADA tiene la capacidad de almacenar un histórico de datos en un archivo dentro del ordenador, lo cual resulta útil para dar continuidad al proyecto en trabajos futuros. Asimismo, se cuenta con un archivo para registrar posibles errores durante la conexión, permitiendo la identificación y corrección oportuna y eficiente de los mismos.

Una característica destacada del inversor *Sunny Island* es su capacidad para la configuración de parámetros específicos, mismos que se han detallado en el manual para el uso del sistema. Para este propósito, se ha desarrollado una sección adicional dentro del sistema SCADA del

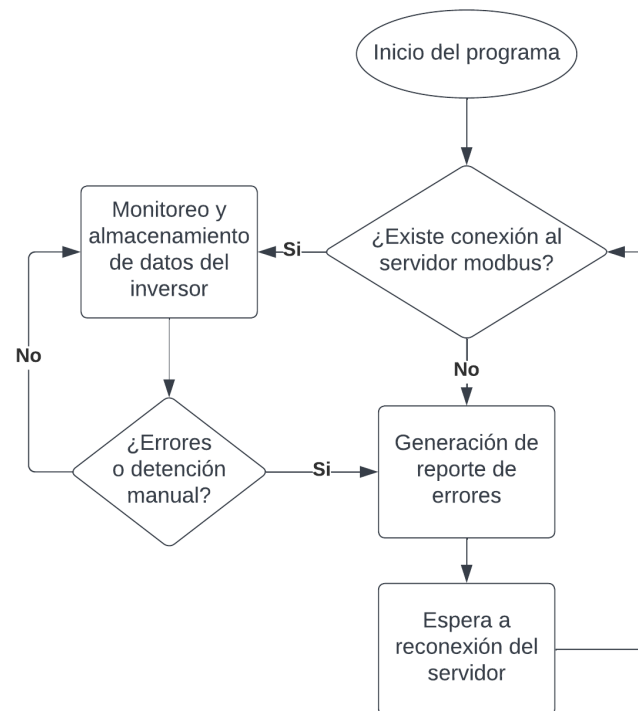


Figura 3.2: Funcionamiento del sistema SCADA para los inversores.

mencionado inversor. Esta sección se ajusta al funcionamiento descrito en la Figura 3.4.

El proceso de configuración se inicia con la verificación de la conexión adecuada con el servidor Modbus. Esta verificación es de fácil realización gracias a los bloques Transmission Control Protocol (TCP) de LabVIEW, ya que se encargan de lanzar alertas en cuando existen errores de conexión. Estos errores pueden darse debido a sucesos como una caída de la red mediante la que están conectados los equipos o fallos en el sistema de la Raspberry Pi. Una vez establecida la conexión, se visualizan los parámetros configurables del inversor y se proporciona un menú de configuración para editar los parámetros de funcionamiento del inversor *Sunny Island*. La interfaz de usuario se ha diseñado como se aprecia en la Figura 3.5, los parámetros apreciados se detallan brevemente en el Capítulo 4 y a profundidad en el manual de los anexos.

3.3. Gateway

La Sección 3.3.1 muestra la implementación física de la interfaz de comunicación entre el Gateway y el inversor. Según la Figura 3.6, en la cual presenta la arquitectura del Gateway, el envío de los datos del inversor *Sunny Island* hacia el SCADA se realiza con dos programas implementados en la Raspberry Pi. El primero, es el servidor Modbus (Sección 3.3.3), en el



(a) Ampere Square.



(b) Sunny Island.

Figura 3.3: Interfaces visuales del sistema SCADA de los inversores.

cual se publica el valor de los parámetros del inversor. Estos datos son obtenidos del segundo programa, el cual establece la comunicación con el inversor (Sección 3.3.2), mediante la interfaz *Gateway-Inversor*, con el fin de obtener el valor de los parámetros deseados. El código fuente del desarrollo realizado para implementar el Gateway se encuentra en el repositorio de Github [17].

3.3.1. Interfaz *Gateway-Inversor*

En la Figura 3.7 se ilustra el montaje final, donde la Raspberry Pi se encuentra en el gabinete observado a la derecha del inversor. Por otro lado, en la figura 3.8 se presenta un diagrama que ilustra el conexionado físico. Un cable es conectado al puerto ComSmaIn del inversor. El otro extremo del cable se conecta a un módulo RS485 a USB el cual a su vez es conectado a un puerto USB de la Raspberry Pi.

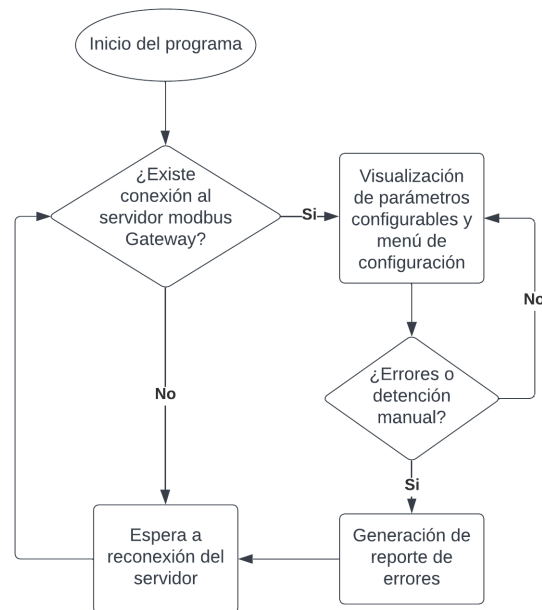


Figura 3.4: Funcionamiento del menú de configuración del sistema SCADA para el inversor *Sunny Island*.

Para realizar la conexión entre el cable de comunicación y el módulo RS485 a USB, se siguió el manual del equipo que detalla el procedimiento de conexión [1]. La Tabla 3.1 muestra la asignación utilizada para esta conexión.

Asignación de señal RS485	Hembra RJ45: Sunny Island	Código de color del conector RJ45
A (Data+)	3	Blanco a rayas verdes
GND	2	Naranja a rayas blancas
B (Data-)	6	Verde a rayas blancas

Tabla 3.1: Tabla de conexión. Fuente: [1]

En la Figura 3.9 se muestra la conexión del módulo RS485 a USB al cable de comunicación. De acuerdo con la Figura 3.9(a), el cable blanco y de rayas verdes se conecta al terminal A, mientras que el cable verde se conecta al terminal B. La Figura 3.9(b) verifica que la polarización se ajusta a lo indicado en la Tabla 3.1.

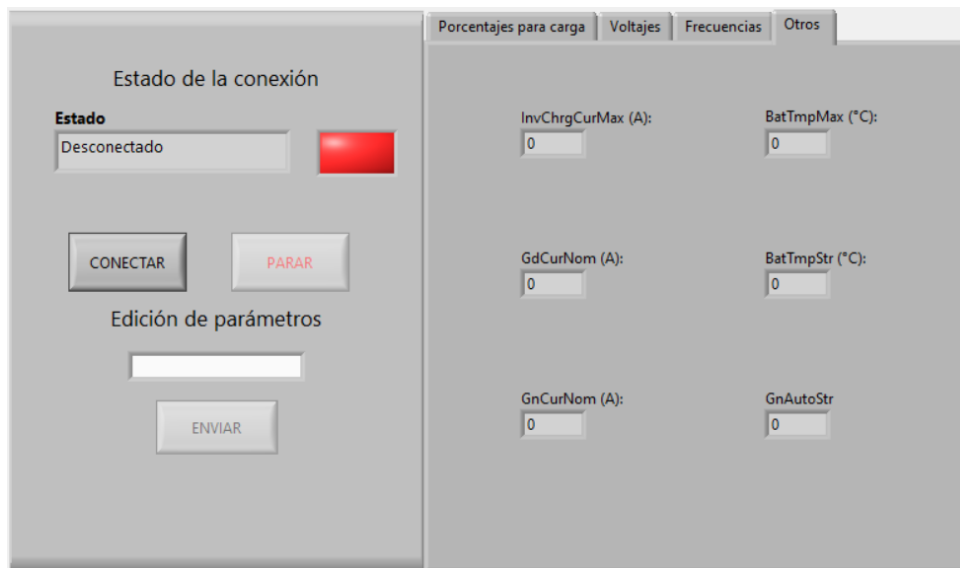


Figura 3.5: Interfaz visual del menú de configuración del sistema SCADA del inversor *Sunny Island*.

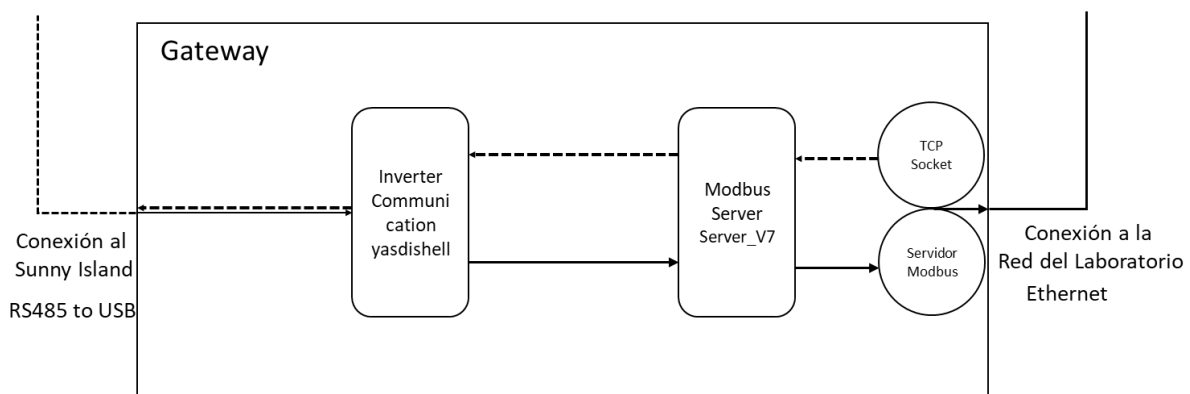


Figura 3.6: Arquitectura del *Gateway*.

3.3.2. Comunicación *Gateway*-Inversor, *Yasdishell*

La comunicación entre el *Gateway* y el inversor se realiza utilizando el protocolo *SMA Data*. Para recuperar la información se hace uso del programa *yasdishell*, el cual se encuentra disponible en los repositorios de GitHub [18, 19]. Este programa está desarrollado en C. Para obtener los datos del inversor, a través de la interfaz *Gateway*-Inversor, se modificó el programa *yasdishell*, el cual emplea un conjunto de librerías de Yet Another SMA Data Implementation (YASDI) llamadas *libyasdimaster* y *libyasdi*, que permiten el funcionamiento de este dispositivo como Maestro, y tienen el objetivo de comunicar el *Gateway*, con el inversor, de forma que sea posible la lectura y edición de información en el inversor. Este programa se ejecuta en la



Figura 3.7: Conexión física entre al inversor.

Raspberry Pi. El funcionamiento general del programa se presenta en la Figura 3.10.

La aplicación verifica al momento de iniciarse, si existe un dispositivo conectado en el puerto USB. El puerto se encuentra especificado en un archivo de configuración. Una vez detectado el puerto, se realiza la identificación del dispositivo, en este caso el inversor. Terminado este proceso, se accede a un modo de comunicación que permite el acceso a todas las variables disponibles en el dispositivo.

Una vez completados los procedimientos de inicialización, se procede a leer y escribir datos en el inversor. En este punto es importante indicar que existen dos tipos de valores que pueden ser obtenidos del inversor. Un conjunto de valores son agrupados en un grupo que se conoce como *Spot Channels*. Los valores que se agrupan en este conjunto solo pueden ser leídos, es decir, no pueden editarse. El segundo conjunto de valores disponibles son conocidos como *Parameter Channels*, los cuales pueden ser leídos y editados por parte del usuario. El manual de usuario del inversor indica cuáles son los datos que pueden ser únicamente leídos, y cuáles son los datos que pueden ser leídos y editados [10]. En la programación se establecieron canales específicos de los cuales se obtiene información del inversor. Una vez los valores de los canales son leídos, estos son almacenados momentáneamente en dos archivos denominados `SPOTCHANNELS.txt` y `PARAMCHANNELS.txt`, según corresponda. Estos dos archivos tienen el objetivo de enlazar la información recopilada por el programa `yasdishell` con el programa que implementa el servidor Modbus. Para la edición de valores en el inversor, se emplea el archivo denominado `SetInformation.txt`. En este archivo, el servidor Modbus alma-

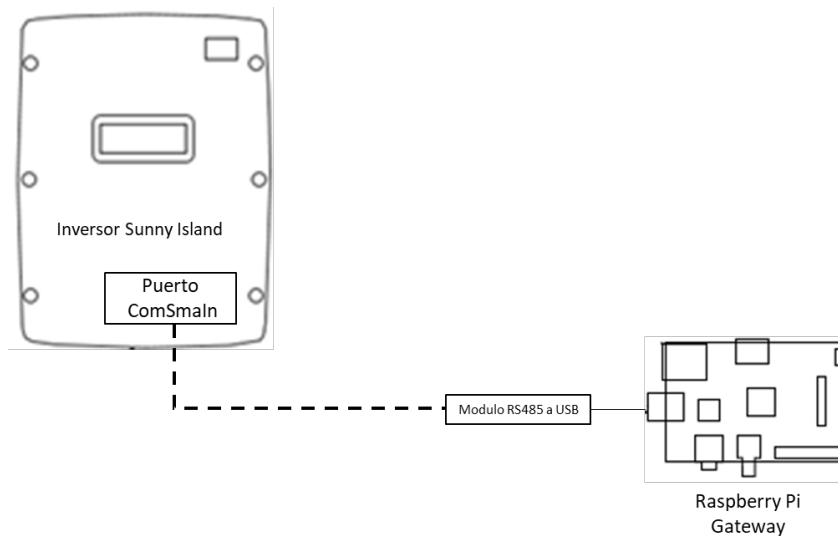
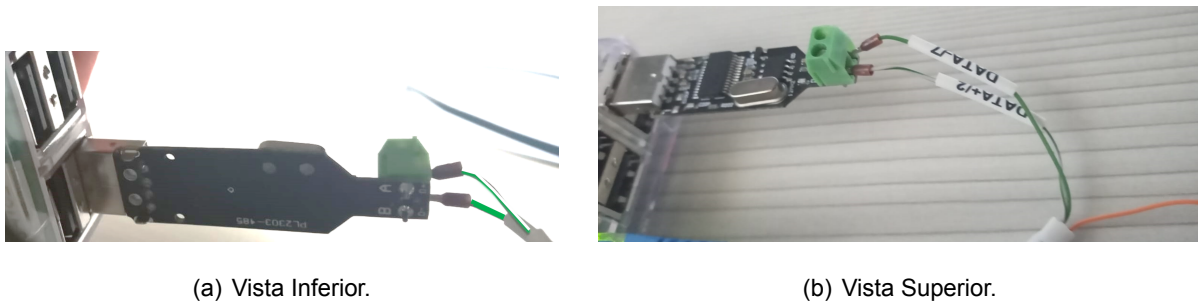


Figura 3.8: Diagrama de Conexión Física



(a) Vista Inferior.

(b) Vista Superior.

Figura 3.9: Enlace físico entre el cable de conexión y el módulo RS485 a USB.

cena la información necesaria para editar un parámetro del inversor. Para verificar si existe la intención de cambiar parámetros del inversor, el programa `yasdishell` comprueba si el archivo `SetInformation.txt` tiene datos. En caso afirmativo, el programa los lee y ejecuta la edición de datos del inversor, caso contrario sigue el proceso de lectura.

3.3.3. Servidor Modbus

Al servidor Modbus se lo denomina `Server_V7` y fue desarrollado en Python. Este servidor publica el valor de los parámetros del inversor que podrán ser leídos por el sistema SCADA. El programa recibe la información para la edición de datos del inversor a través de un `socket` TCP. En la Figura 3.11 se muestra el funcionamiento del programa desarrollado.

Los datos recolectados y almacenados a través de la interfaz RS485 del inversor *Sunny Island* y el protocolo *SMA Data* en los archivos `PARAMCHANNELS.txt` y `SPOTCHANNELS.txt`, son procesados para representarlos como números enteros, ya que originalmente los valores po-

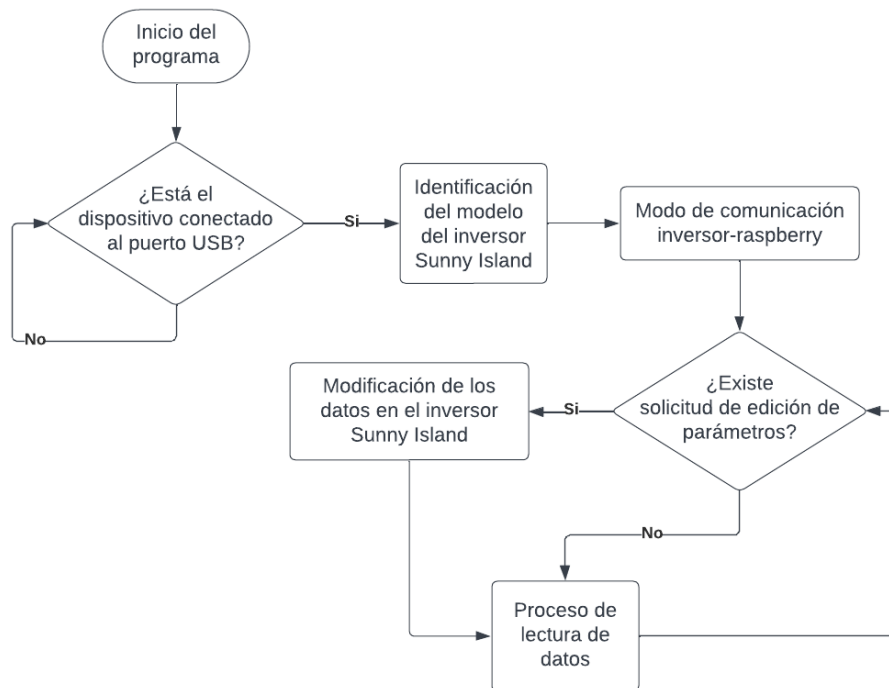
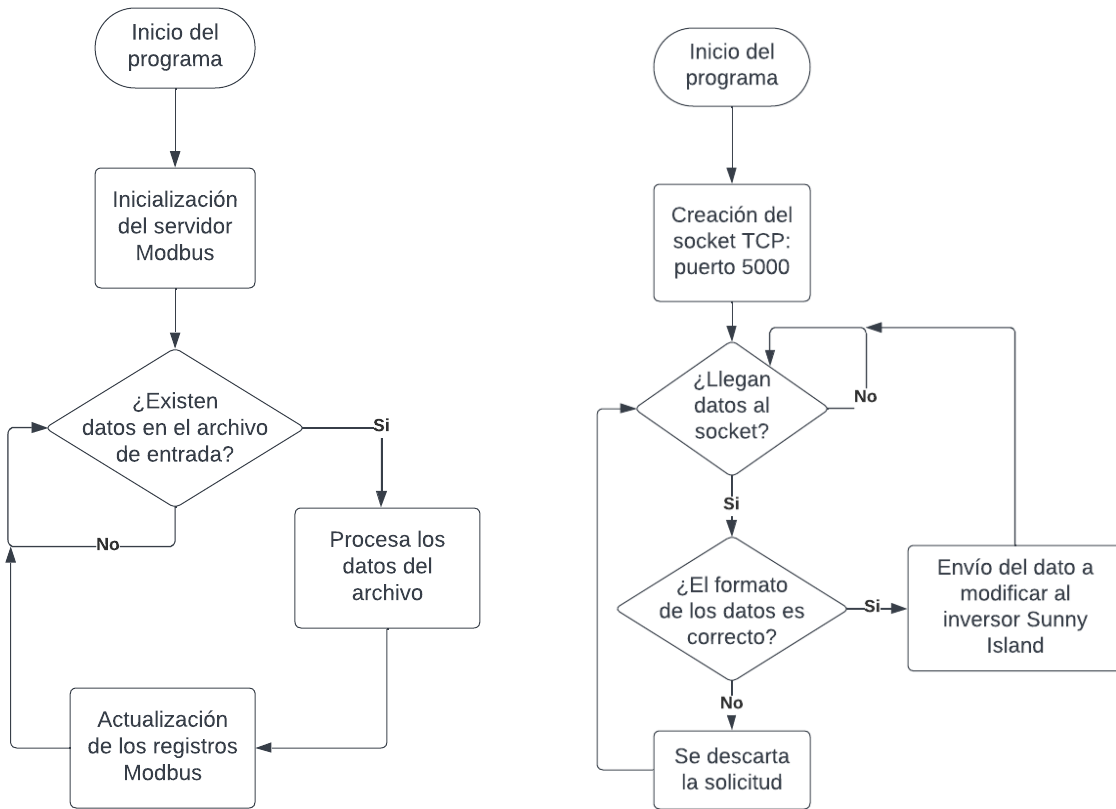


Figura 3.10: Principio de funcionamiento del programa yasdishell .

seen parte entera y parte decimal. El objetivo es facilitar su publicación en el servidor, el cual cuenta con registros de 16 bits. Los registros empleados para almacenar esta información son los Registros de Entrada o *Input Registers*, los cuales solo permiten la lectura de datos. Es importante aclarar que los registros no se almacenan físicamente en la Raspberry Pi, sino que el programa representa estos registros en memoria. Esto quiere decir, que los registros se mantienen en la memoria RAM, mientras el programa se encuentra en ejecución. El procedimiento antes expuesto puede ser visualizado en la Figura 3.11(a).

En el mismo programa, para lograr una ejecución en paralelo a la publicación de datos, se desarrolló un hilo, en el cual se implementa un *socket* TCP. Este proceso permite la modificación de los parámetros del inversor sin interrumpir el monitoreo realizado por el sistema SCADA. Los datos recibidos como solicitudes pasan por un proceso de validación para garantizar que solo los usuarios autorizados puedan realizar dichas acciones. El procedimiento de validación se realiza en el servidor Modbus. Una vez validada la información, esta es almacenada en el archivo *SetInformation.txt*, para que el programa yasdishell la procese. El procedimiento descrito previamente puede ser observado en la Figura 3.11(b)



(a) Servidor Modbus.

(b) Socket TCP.

Figura 3.11: Funcionamiento del programa Server_V7.

3.3.4. Servicios

La última configuración desarrollada corresponde a la creación de dos servicios, uno para cada programa, que tienen el objetivo de ejecutarlos automáticamente cuando la Raspberry Pi se enciende. En el servicio para el programa `yasdishell`, denominado `ComInversor`, se destaca la importancia de asegurar que los servicios de red, los puertos USB y las dependencias de la Raspberry Pi se inicien correctamente. Además, se agregó un retardo de 40 segundos para permitir que todos los recursos de la Raspberry Pi se establezcan antes de iniciar la ejecución del programa. Este tiempo fue determinado de manera empírica.

En lo que respecta a la configuración del servicio para el programa `Server_V7`, denominado `ComServidor`, es necesario asegurarse de que los servicios de red estén debidamente activados antes de su ejecución. En ambos servicios se ha establecido la funcionalidad de reinicio automático en caso de fallos del servicio.

3.4. Inversores

En el laboratorio de Micro-Red se tiene un inversor *Sunny Island* y un inversor *Ampere Square*. Para el primero fue necesario establecer un dispositivo intermediario o *Gateway* (ver Sección 3.3). Este tiene como objetivo el reenviar los datos del inversor al sistema SCADA. El *Gateway* lee o escribe en el inversor utilizando el protocolo *SMA Data* y los publica a través del servidor Modbus.

En cuanto al inversor *Ampere Square*, este cuenta con un servidor Modbus TCP integrado, el proceso de establecer la comunicación con el sistema SCADA se simplifica. Además, el sistema de monitoreo SCADA para el inversor *Ampere Square* se ejecuta en paralelo al sistema presentado para *Sunny Island*, siguiendo el esquema de funcionamiento presentado en la Figura 3.2.

Una particularidad de este equipo es que, a diferencia del inversor *Sunny Island*, no cuenta con la capacidad de editar los datos de su funcionamiento. Como resultado, las características del sistema SCADA de cada inversor varían en términos de funcionalidades, donde el inversor *Sunny Island* implementa lectura y escritura, mientras que el inversor *Ampere Square* implementa solo lectura..

Esta implementación se caracteriza por su simplicidad y consta de dos estados. Al establecer la conexión inicial con el inversor, pueden ocurrir tanto conexiones exitosas como fallidas debido a posibles errores. En caso de lograr la conexión al servidor Modbus TCP, se inicia el proceso de monitorización de los datos de funcionamiento. Sin embargo, si ocurre algún error durante la conexión, se genera un informe de error que se registra en un archivo, y se espera a restablecer la conexión. Asimismo, al igual que en el inversor *Sunny Island*, se implementa el registro de un historial de datos para su seguimiento.

4. Resultados

En esta sección se presentan los resultados conseguidos al desarrollar la integración de los inversores. En la Sección 4.1 se presentan los resultados del estado del inversor y el proceso de comunicación en la Raspberry Pi, mientras que en la Figura 4.2 se presentan los resultados relacionados al sistema Supervisory Control and Data Acquisition (SCADA) y la validación de los datos presentados.

4.1. Inversor *Sunny Island*

4.1.1. Comunicación *Gateway-Inversor*

Para verificar el funcionamiento de la interfaz de comunicación, se configuró la impresión de mensajes en consola de los programas. La Figura 4.1 muestra el resultado obtenido.

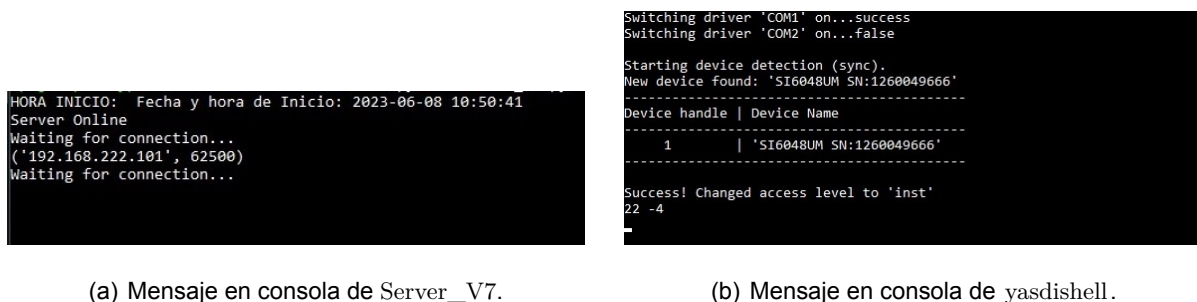


Figura 4.1: Mensajes en consola de las aplicaciones.

La Figura 4.1(b) muestra la salida en la consola al ejecutar la aplicación `yasdishell`. Se puede apreciar que esta detecta la conexión entre el *Gateway* y el inversor, mostrando en pantalla el modelo y número de serie. En la penúltima línea se confirma exitosamente el cambio al modo de comunicación que permite interactuar con todos los canales.

En la Figura 4.1(a) se observan los mensajes en la consola al ejecutar la aplicación `Server_V7`. La aplicación presenta la hora de inicio del servidor Modbus, junto con un mensaje que indica que se encuentra en línea. Esto sugiere que ahora es posible leer los parámetros del inversor. Además, se visualiza un mensaje que indica que se está esperando una conexión, lo cual hace referencia al envío de datos a través del *socket* Transmission Control Protocol (TCP).

Cuando se envía información para la edición de parámetros del inversor, el servidor imprime la dirección IP del remitente de este mensaje. Mientras tanto, la aplicación `yasdishell` recibe y procesa estos datos.

Como se mencionó previamente, se crearon dos servicios, uno para cada aplicación, con el objetivo de establecer una ejecución automática apenas el *Gateway* se encienda. En la Figura 4.2 se puede observar que cada servicio se encuentra en estado activo, el cual es el estado esperado.

En la Figura 4.2(b) se observa el estado activo del servicio del servidor Modbus, mientras que en la Figura 4.2(a) se presenta el estado activo del servicio que permite la comunicación entre el *Gateway* y el inversor.

```
pi@raspberrypi:~$ sudo systemctl status ComInversor.service
● ComInversor.service - Daemon inverter communication
   Loaded: loaded (/etc/systemd/system/ComInversor.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-06-13 13:42:29 -05; 9min ago
     Process: 337 ExecStartPre=/bin/sleep 40 (code=exited, status=0/SUCCESS)
    Main PID: 650 (yasdisshell)
      Tasks: 2 (limit: 415)
         CPU: 9.720s
   CGroup: /system.slice/ComInversor.service
           └─650 /home/rpi/Documentos/libyasdi/projects/generic-cmake/build-gcc/yasdisshell

jun 13 13:40:27 raspberrypi systemd[1]: Starting Daemon inverter communication...
jun 13 13:42:29 raspberrypi systemd[1]: Started Daemon inverter communication.
pi@raspberrypi:~$
```

(a) Estado activo del servicio ComInversor.

```
pi@raspberrypi:~$ sudo systemctl status ComServidor.service
● ComServidor.service - Modbus Server
   Loaded: loaded (/etc/systemd/system/ComServidor.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2023-06-13 13:40:27 -05; 12min ago
     Main PID: 339 (python3)
      Tasks: 4 (limit: 415)
         CPU: 15.621s
   CGroup: /system.slice/ComServidor.service
           └─339 /usr/bin/python3 /home/rpi/Documentos/Modbus/Server_V7.py

jun 13 13:40:27 raspberrypi systemd[1]: Started Modbus Server.
pi@raspberrypi:~$
```

(b) Estado activo del servicio ComServidor.

Figura 4.2: Estado activo de los servicios.

4.1.2. Tiempos de respuesta del Gateway

Se llevaron a cabo pruebas para verificar los tiempos de reacción del sistema propuesto. En primer lugar, se evaluó el tiempo necesario para que el sistema se inicie después de encender la Raspberry Pi. Este proceso tiene una duración aproximada de 3 minutos, teniendo en cuenta el tiempo requerido para encender la Raspberry Pi.

Posteriormente, se registró el tiempo requerido para que un valor editado se refleje en el campo correspondiente del sistema SCADA. El tiempo promedio registrado para esta operación es de alrededor de 27 segundos. Para comprender el motivo de este tiempo promedio, se puede utilizar el diagrama de bloques mostrado en la Figura 3.10. Cuando se envía un valor para ser editado en el inversor, pueden ocurrir dos escenarios. En el primer primer escenario es posible que la aplicación principal se encuentre en el condicional que hace referencia a la solicitud de edición de parámetros. En este caso, los datos se procesarán rápidamente, en un intervalo de 10 a 15 segundos, se modificará el valor del inversor y se continuará con el proceso de lectura

de datos para su posterior observación en el sistema SCADA.

También puede darse el segundo escenario, en el cual la aplicación está en pleno proceso de lectura de datos cuando se envía un valor para ser editado en el inversor. En esta situación, el proceso de lectura debe finalizar antes de que se pueda llevar a cabo la edición de datos en el inversor. Después de este último procedimiento, se realiza la lectura de datos para observar el cambio de valor en el sistema SCADA. Debido a que el segundo caso ejecuta un proceso de lectura adicional antes de mostrar los datos editados del inversor en el sistema SCADA, este tiene un retardo mayor que el primer caso, entre 15 a 30 segundos. Es por ello, que el valor temporal que se mencionó previamente, corresponde a un promedio. Además, el tiempo de respuesta obtenido es aceptable debido a que los valores del inversor no cambian rápidamente.

4.1.3. Sistema SCADA

Una vez que todos los sistemas previos están ejecutándose, es posible obtener los datos de funcionamiento del inversor *Sunny Island* a través del sistema SCADA. Existen dos opciones disponibles en la interfaz del sistema. La primera opción, mostrada en la Figura 4.3, permite el monitoreo de datos en tiempo real, con grupos de datos categorizados como potencia, batería, inversor y datos externos.

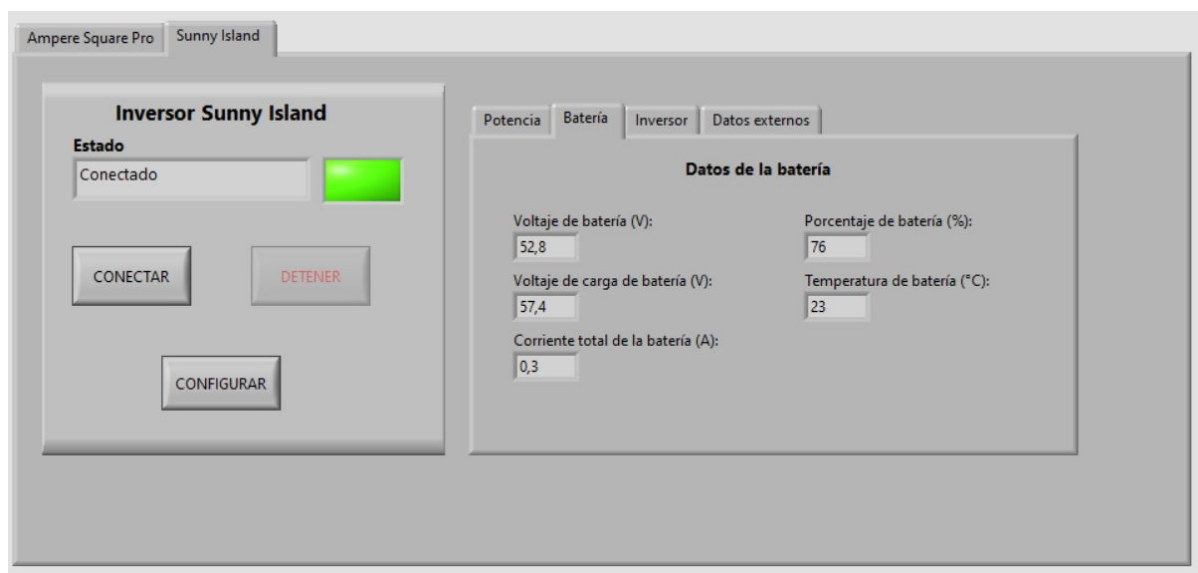


Figura 4.3: Interfaz de supervisión para el inversor *Sunny Island*.

La segunda interfaz se puede acceder a través del botón de configuración y se muestra en la

Figura 4.4. En esta pantalla se encuentran datos como la corriente de carga máxima (InvChrgCurMax), temperatura máxima de la batería (BatTmpMax), corriente de red nominal (GdCurNom), temperatura mínima para reinicio (BatTmpStr), corriente nominal del generador (GnCurNom) y el número de arranques automáticos (GnAutoStr). Además, se presentan más datos configurables del inversor, los cuales también están categorizados en porcentajes de carga, voltajes, frecuencias y otros parámetros relevantes que se especifican en el manual de usuario.

La interfaz de configuración del inversor *Sunny Island* en el sistema SCADA proporciona a los operadores del sistema la capacidad de ajustar y personalizar diversos aspectos del funcionamiento del inversor. Mediante el uso de uno de los manuales desarrollados en conjunto con el trabajo de titulación, los operadores pueden cambiar la configuración del inversor, escribiendo en el campo de texto del menú de configuración de la Figura 4.4, un texto con el formato clave;canal;valor. En el campo clave se usa una contraseña establecida para el laboratorio, el canal que corresponde al parámetro a editar y el valor nuevo a configurar. Todos estos datos se encuentran detallados en el manual de uso del sistema.

Estado de la conexión

Estado

Conectado

CONECTAR PARAR

Edición de parámetros

Porcentajes para carga Voltajes Frecuencias Otros

InvChrgCurMax (A): 48 BatTmpMax (°C): 45

GdCurNom (A): 30 BatTmpStr (°C): 40

GnCurNom (A): 30 GnAutoStr: 3

Figura 4.4: Interfaz de configuración para el inversor *Sunny Island*.

4.1.4. Validación de datos

El método utilizado para verificar la adquisición de datos en el sistema SCADA es compararlos con los datos mostrados en el tablero del inversor. En la Figura 4.5, se pueden observar los datos del porcentaje de carga de la batería (76 %) y el número de arranques automáticos (3).



(a) Porcentaje de carga de la batería.

(b) Número de arranques automáticos del inversor.

Figura 4.5: Valores del tablero de control del inversor.

Al comparar estos datos con los mostrados en las Figuras 4.3 (porcentaje de la batería) y 4.4 (GnAutoStr), podemos verificar su concordancia. Este proceso de verificación se realizó para todos los datos presentados en las diferentes ventanas de la interfaz del sistema SCADA.

La comparación de los datos entre el tablero del inversor y el sistema SCADA asegura la precisión y confiabilidad de los datos adquiridos. Al verificar la concordancia de los valores, se garantiza que el sistema SCADA está recibiendo y mostrando la información correctamente, lo que es fundamental para el monitoreo y control efectivo del inversor *Sunny Island*.

4.2. Inversor *Ampere Square*

4.2.1. Sistema SCADA

Como resultado de la implementación del sistema SCADA, se obtiene el sistema representado en la Figura 4.6. Este sistema proporciona al usuario la capacidad de monitorizar las siguientes variables:

- Voltaje del inversor.
- Producción solar en términos de potencia, generada por los paneles solares conectados

al inversor.

- Potencia consumida por la red, que representa la potencia inyectada desde el inversor a la red pública.

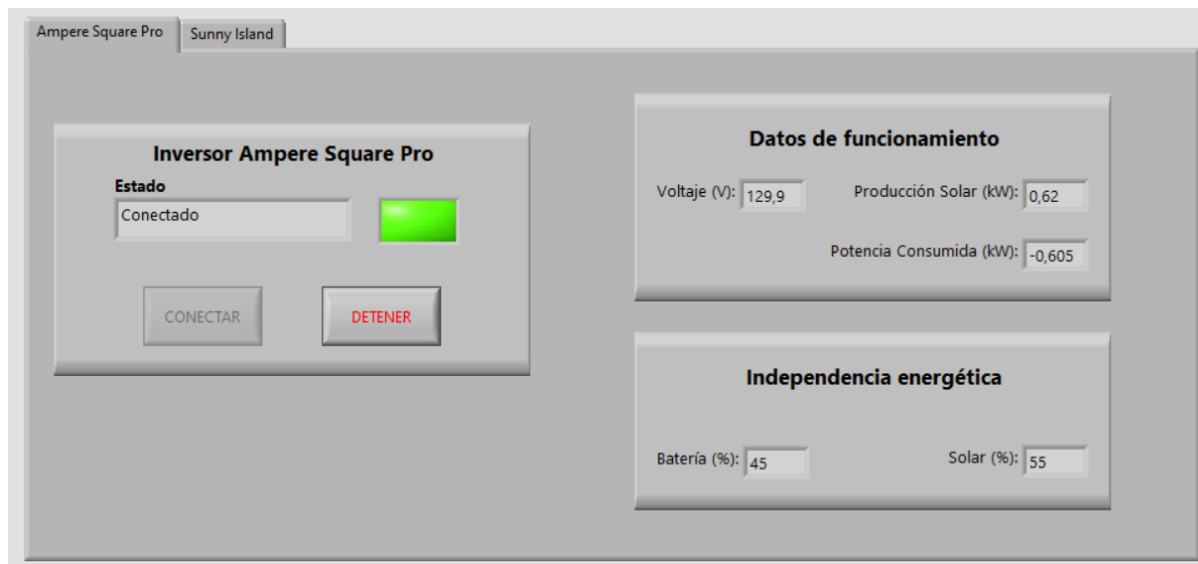


Figura 4.6: Interfaz de supervisión para el inversor *Ampere Square*.

El sistema SCADA implementado permite visualizar en tiempo real estas variables clave, brindando al usuario información precisa sobre el rendimiento y la producción de energía del inversor. Esta funcionalidad es fundamental para supervisar y controlar eficientemente el sistema fotovoltaico, lo que a su vez contribuye a una gestión eficaz de la energía y a la optimización de su uso.

4.2.2. Validación de datos

Como parte del proceso de validación de datos, se utilizó la aplicación móvil *My Ampere* propiedad de *Ampere Energy*. En la Figura 4.7 se muestra la interfaz de esta aplicación, la cual está conectada al inversor *Ampere Square* y proporciona información sobre variables clave como la inyección a la red y la producción solar. Comparando estos valores con los datos mostrados en la interfaz del sistema SCADA, se puede verificar la precisión y la veracidad de los datos obtenidos.

La aplicación móvil de *Ampere Energy* desempeña un papel crucial en la validación de los datos recopilados, ya que permite una visualización directa y en tiempo real de las variables

monitoreadas. Al comparar los valores registrados en el sistema SCADA con los mostrados en la aplicación, se asegura la exactitud y la consistencia de la información recopilada.

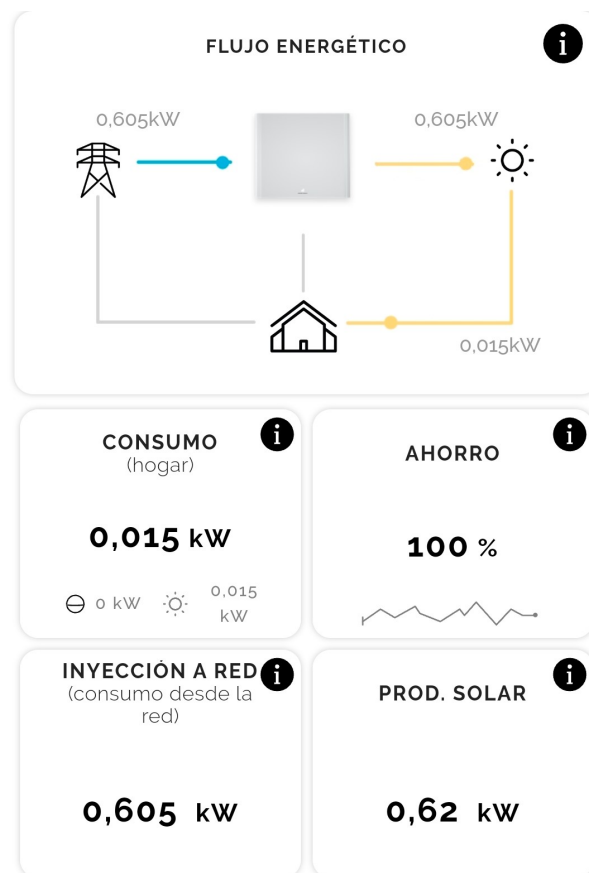


Figura 4.7: Aplicación móvil del inversor *Ampere Square*.

5. Conclusiones y recomendaciones

5.1. Conclusiones

La integración de los inversores al sistema Supervisory Control and Data Acquisition (SCADA) del laboratorio de Micro-Red se llevó a cabo de manera exitosa. El uso de LabVIEW como plataforma de desarrollo facilitó la interacción entre los dispositivos desarrollados y el sistema SCADA existente, permitiendo la visualización y el control eficiente de los inversores.

Adicionalmente, mediante el análisis del estado de la técnica sobre la integración de inversores hacia sistemas SCADA, se identificaron las mejores prácticas y enfoques utilizados en la industria. Esto proporcionó una base sólida para el diseño e implementación de la solución propuesta.

Por último, en un dispositivo Raspberry Pi se implementó un *Gateway* para el inversor *Sunny Island*. El *Gateway* recupera la información del inversor usando el protocolo *SMA Data* y la publica en el servidor Modbus implementado en dicho dispositivo.

5.2. Recomendaciones

A continuación, se presentan una serie de recomendaciones que pueden ser consideradas para mejorar y optimizar el sistema integrado, basadas en los logros obtenidos al integrar exitosamente los inversores al sistema SCADA del laboratorio de Micro-Red de la Universidad de Cuenca.

Para garantizar el rendimiento óptimo del sistema a medida que aumenta la carga de trabajo, se sugiere llevar a cabo pruebas exhaustivas de robustez y escalabilidad. Estas pruebas permitirán identificar posibles cuellos de botella, evaluar la capacidad de respuesta del sistema y anticipar limitaciones en caso de futuras expansiones o adiciones de dispositivos.

Asimismo, se recomienda establecer un plan de mantenimiento preventivo para asegurar el funcionamiento continuo y confiable del sistema integrado. Este plan debe incluir actividades regulares como la verificación de conexiones, actualizaciones de *software/firmware* y limpieza de equipos, con el objetivo de minimizar el riesgo de fallas y prolongar la vida útil del sistema.

Finalmente, es fundamental continuar con la documentación detallada de todo el proceso de integración. Mantener registros actualizados de configuraciones, cambios realizados y problemas encontrados será de gran utilidad para futuros trabajos o mejoras en el sistema. Esta

documentación asegurará la trazabilidad y facilitará la replicación del proyecto en casos similares.

Referencias

- [1] A. ENERGY, "Ampere Square Pro Manual del instalador," 2021.
- [2] A. Daneels y W. Salter, "What is SCADA?" 1999.
- [3] G. Thomas, "Introduction to the Modbus protocol," *The Extension*, vol. 9, num. 4, pp. 1–4, 2008.
- [4] —, "Introduction to Modbus serial and Modbus TCP," *The extension: A technical Supplement to Control Network*, vol. 9, num. 5, 2008.
- [5] Prüssing, "YASDI Documentation Implementation of the SMA Data Protocol."
- [6] J. Flicker, R. Kaplar, M. Marinella, y J. Granata, "PV inverter performance and reliability: What is the role of the bus capacitor?" in *2012 IEEE 38th Photovoltaic Specialists Conference (PVSC) PART 2*. IEEE, 2012, pp. 1–3.
- [7] M. A. Abella, "Sistemas fotovoltaicos," *SAPT Publicaciones Técnicas, SL*, 2005.
- [8] S. B. Kjær, *Design and control of an inverter for photovoltaic applications*. Institut for Energiteknik, Aalborg Universitet, 2005.
- [9] R. Kaplar, R. Brock, S. DasGupta, M. Marinella, A. Starbuck, A. Fresquez, S. Gonzalez, J. Granata, M. Quintana, M. Smith y otros, "PV inverter performance and reliability: What is the role of the IGBT?" in *2011 37th IEEE Photovoltaic Specialists Conference*. IEEE, 2011, pp. 001 842–001 847.
- [10] SMA, "Inversor con batería SUNNY ISLAND 4548-US/6048-US Instrucciones de funcionamiento," 2015.
- [11] A. Pineda, "Instrumentación virtual: Fundamentos de programación gráfica con LabVIEW," 2018.
- [12] K. D. Deenadayalan, A. Arunraja, S. Jayanthi, y S. Selvaraj, "IoT based remote monitoring of mass solar panels," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 1009–1014.
- [13] H. Guozhen, C. Changsong, D. Shanxu y otros, "Solutions for SCADA system communication reliability in photovoltaic power plants," in *2009 IEEE 6th International Power Electronics and Motion Control Conference*. IEEE, 2009, pp. 2482–2485.

- [14] Y. Jiang, Q. Wu, Y. Yang, y X. Chi, “Design of a Home Energy Management System Based on Cloud Service,” in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2018, pp. 1–5.
- [15] R. A. López, E. Pincirolí, y F. G. Tinetti, “Microcontroladores asociados a medición y comunicaciones en sistemas SCADA de energía,” in *XVI Workshop de Investigadores en Ciencias de la Computación*, 2014.
- [16] J. J. B. Iza, “Diseño y Construcción de un Inversor Trifásico acoplado a un Sistema de Transferencia Automático con Monitoreo SCADA,” *INNOVATION & DEVELOPMENT IN ENGINEERING AND APPLIED SCIENCES*, vol. 1, num. 2, pp. 13–13, 2019.
- [17] S. Araujo, F. Tigre, y F. Astudillo, “microred-sunnyisland-ampere.” [En línea]. Disponible: <https://github.com/fabianastudillo/microred-sunnyisland-ampere.git>
- [18] EckhardM, “Python wrapper for SMA converter YASDI C library.” [En línea]. Disponible: <https://github.com/EckhardM/pyYASDI>
- [19] Pknowledge, “Fork of the YASDI library for solar SMA inverters.” [En línea]. Disponible: <https://github.com/pknowledge/libyasdi>

A. Anexo

A.1. Anexo A: Guía para la instalación, uso y verificación de funcionamiento del sistema construido para el inversor *Sunny Island*

Restringido: Uso privado del laboratorio de Micro-Red.

A.2. Anexo B: Video tutorial para el proceso de compilación de los programas desarrollados y para solventar errores en el proceso.

Restringido: Uso privado del laboratorio de Micro-Red.

A.3. Anexo C: Conexión entre el inversor Sunny Island y Raspberry Pi

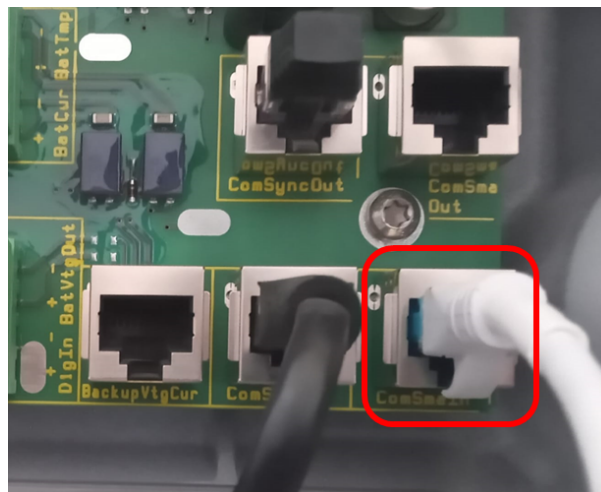


Figura A.1: Conexión con el puerto ComSmaIn

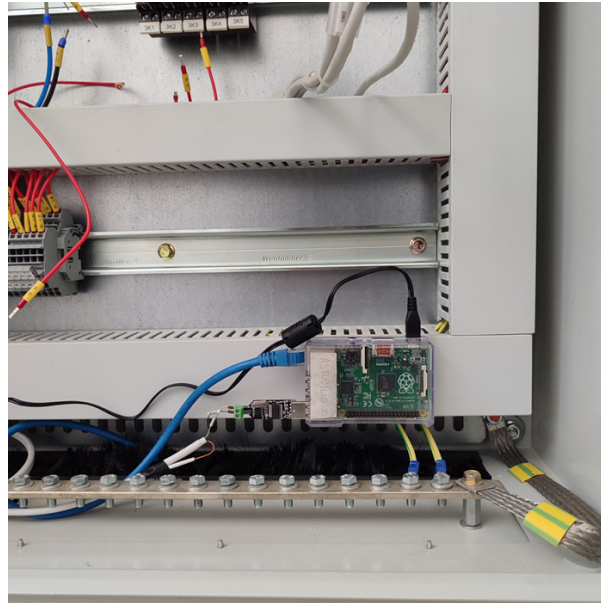


Figura A.2: Conexión al puerto USB de la Raspberry Pi



Figura A.3: Localización del puerto ComSmaIn en el inversor