

Supervisión y Gestión de Datos de Código Abierto de una Microrred Aislada en una Escuela Rural

Open Source Data Management and Supervision System of an Isolated Microgrid in a Rural School

Guillermo Catuogno ^{#1}, Guillermo Acosta ^{*}, Carlos Catuogno [#]

[#] *Laboratorio de Control Automático (LCA), INTEQUI (CONICET-UNSL),
 Villa Mercedes, Argentina.*

¹ grcatuogno@unsl.edu.ar

^{*} *Laboratorio de Control Automático (LCA), UNSL,
 Villa Mercedes, Argentina.*

Recibido: 26/02/21; Aceptado: 30/04/21

Resumen— En este trabajo se presenta la implementación de un sistema de gestión de datos recolectados de una microrred eólica-solar instalado en una escuela rural de ubicación remota. Este sistema fue realizado utilizando software libre, y consiste en la adquisición de los datos del sistema eólico-solar, el almacenamiento de la información en una base de datos relacional y la visualización de la información procesada a través del servidor y visualizada por medio de un navegador web. El sistema desarrollado fue instalado en el paraje Puerta del Sol, ubicado al norte de la provincia de San Luis y consta de un controlador lógico programable (PLC) que recolecta la información del proceso, una pantalla interface hombre-máquina (HMI) para visualización local de los datos procesados, y una computadora que contiene un servidor web y un gestor de base de datos (ubicada en las instalaciones de la Universidad). La computadora ejecuta una aplicación programada en Python que periódicamente consulta al PLC vía internet sobre la información del proceso y guarda esta información en una base de datos MariaDB (compatible MySQL). Paralelamente el servidor web responde mediante aplicaciones programadas en lenguaje PHP a las consultas que, desde un cliente externo remoto, sean solicitadas mediante una página HTML. El comportamiento del sistema de gestión propuesto se ha validado con los resultados experimentales obtenidos.

Palabras clave: Supervisión; gestión de datos; código abierto.

Abstract— This paper presents the implementation of a data management system. The data is collected from a wind-solar microgrid installed in a remote rural school. This system was made using free software, and consists of the acquisition of data from the wind-solar generator, the storage of the information in a relational database and the visualization of the information processed through the internet by means of a web browser. The developed system was installed in the Puerta del Sol area, located in the north of the province of San Luis and consists of a programmable logic controller (PLC) that collects the information of the process, a human-machine interface (HMI) screen for local visualization process data, and a computer that contains a web server and a database manager (located at the University facilities). An application programmed in Python runs in the computer and periodically queries the PLC via the internet about the process information and saves this information in a MariaDB database (MySQL compatible). At the same time, the web server responds through applications programmed in PHP language to the queries that, from a remote external client, are requested through an HTML page. The behavior of the proposed

management system has been validated with the experimental results obtained.

Keywords: Supervision; data management; open source.

I. INTRODUCCION

La demanda de electricidad ha aumentado constantemente debido al crecimiento de la población en todo el mundo. Además, los nuevos desafíos requieren de sistemas de electrificación sostenibles, basados en su relación con el medioambiente, el funcionamiento económico y la conveniencia social.

Una microrred (MR) [1-3], como se muestra en la Fig. 1, es una red de pequeña escala que puede integrar generación distribuida, fuentes de energía renovables (RES), generadores convencionales, sistemas de almacenamiento y sus consumos. Este tipo de sistemas de electrificación son respetuosos con el medioambiente, y se puede operar en modo conectado a la red o en modo aislado en el caso de comunidades rurales o dispersas donde el servicio eléctrico no llega.



Fig. 1. Escuela rural María Florentina Carreño

Esta penetración de fuentes renovables descentralizadas en la red, como las microrredes, ha producido la inclusión de Tecnologías de la Información durante la última década para brindar gestión de energía, datos y sistemas de comunicación. El avance de la informática y las comunicaciones ha hecho posible el registro de datos de procesos remotos en tiempo real, su almacenamiento en

bases de datos y la posibilidad de visualizar estos datos en forma ordenada y metódica por los usuarios. Se presenta aquí un sistema automatizado que permite la captura, almacenamiento y posterior visualización de los datos de un proceso en tiempo real. El acceso a los datos podría darse desde cualquier computadora o dispositivo móvil conectado a internet.

En este trabajo se presenta un desarrollo realizado e implementado en una escuela rural, con *software open source*, con mínimos requerimientos de *hardware*, que permite ser implementada y compartida por toda la comunidad científica. Finalmente se introduce un nuevo dispositivo denominado *OpenWee* que permite obtener un sistema completamente abierto y de bajo costo.

II. DESCRIPCIÓN DE LA MICRORRED

La escuela rural "Maestra Florentina Carreño" se encuentra en el departamento General San Martín, en uno de los puntos de mayor altura de la Provincia de San Luis, Argentina. El camino a la escuela es por la ruta provincial RP-2 donde los últimos 20 kilómetros son de un camino de ripio de difícil acceso y el cruce a través de arroyos hace imposible la circulación de vehículos urbanos. En un radio de 10 km se encuentran las localidades más cercanas que son Libertador de San Martín y la localidad de Quines perteneciente al Departamento Ayacucho.

La Fig. 2 presenta un diagrama en bloques de la estructura de la microrred, donde se observa que el sistema eólico y solar alimenta a través de sus reguladores las barras de CC al cual están conectadas las baterías. El inversor CC/CA también está conectado al bus de CC. Por otro lado, el sistema cuenta con una llave selectora que permite cambiar la alimentación de las cargas entre la microrred híbrida y el grupo diésel ya disponible en la escuela [4,5].

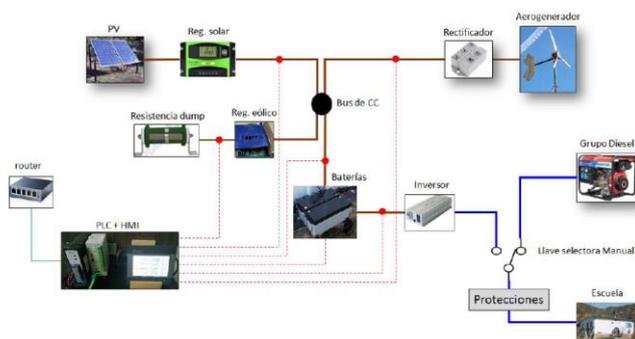


Fig. 2. Esquema de la microrred eléctrica implementada

III. SISTEMA DE SUPERVISIÓN Y GESTIÓN DE DATOS

En las siguientes subsecciones se describen el sistema de supervisión y gestión de datos *open source*, comenzando con la descripción del hardware utilizado, la captura de información, el almacenamiento de la información en la base de datos y el servidor y cliente web.

A. Descripción y uso del hardware de campo

Para la captura de los datos se adjuntó a la placa del generador eólico-solar un tablero eléctrico conteniendo un PLC (*programmable logic controller*) compuesto de un módulo CPU (*central processor unit*), un módulo de comunicación *Ethernet* que está configurado con una dirección IP fija y un módulo de entradas analógicas.

Además, en el frente del tablero se colocó una pantalla HMI (*human machine interface*) para realizar la visualización local de los datos. El PLC es el encargado de leer las entradas digitales y analógicas donde se conectan sensores de corriente, tensión, etc., que dan el valor directo de las variables de proceso involucradas. El PLC también calcula cierta información adicional a partir de los sensores con datos previamente adquiridos: corrientes inferidas, porcentajes de carga/descarga, etc. La función de la pantalla HMI es la de mostrar en tiempo real esta información en una forma comprensible para la operación y diagnóstico del sistema. Este sistema está representado en la Fig. 3.

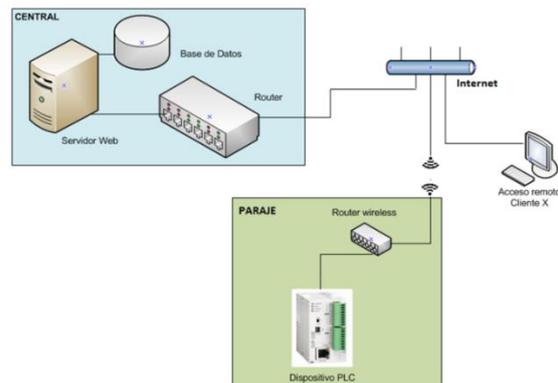


Fig. 3. Esquema del sistema de supervisión y gestión de datos

B. Captura de la información

El comienzo de la captura de datos comienza con los sensores instalados en el sistema y conectados a las entradas analógicas del PLC, se utilizó un divisor resistivo para la medición del voltaje de las baterías, un sensor de temperatura ubicado también en las baterías y cuatro sensores de corriente marca LEM LA25 tipo ventana que permiten medir valores de corriente positivos y negativos.

Luego, el PLC comercial de uso industrial marca DELTA modelo DVP12SE y el módulo analógico DELTA DVP-06AD es utilizado para adquirir la información entregada por los sensores para ser escalada convenientemente y almacenarse en registros internos del PLC.



Fig. 4. PLC DELTA y HMI DELTA utilizados en la implementación.

Esta información es accedida desde una computadora externa llamada "Servidor de Nivel 2" que tiene instalado un sistema operativo *Ubuntu (GNU/Linux)*. En esta computadora corre un programa escrito en el lenguaje de programación *Python* haciendo uso de una librería llamada *pyModbusTCP*, que implementa el protocolo de comunicación que entiende el PLC, que en este caso es el protocolo *ModbusTCP* [6]. Esta librería es de uso gratuito

con licencia GPL (*general public license*) descargable del sitio de Python (www.python.org), [7].

La lectura de los datos se realiza en forma periódica cada un (1) minuto. En la Fig. 5 puede observarse un fragmento resumido de código que ejemplifica cómo se produce la lectura de los datos del PLC en la computadora Servidor de Nivel 2.

```

from pyModbusTCP.client import ModbusClient
from pyModbusTCP import utils
import time

SERVER_HOST = "181.88.176.214"
SERVER_PORT = 15282

c = ModbusClient()

# define modbus server host, port
c.host(SERVER_HOST)
c.port(SERVER_PORT)

print("iniciado")
while True:
    # open or reconnect TCP to server
    if not c.is_open():
        if not c.open():
            print("unable to connect to "+SERVER_HOST+" "+str(SERVER_PORT))

    # if open() is ok, read register (modbus function 0x03)
    if c.is_open():
        # read 10 registers at address 0, store result in regs list
        regs = c.read_holding_registers(4096+201, 12)
        # if success display registers
        if regs:
            print("reg ad #0 to 11: "+str(regs))
            # leer floats
            for f in utils.word_list_to_long(regs, big_endian=False):
                regsF = utils.decode_ieee(f)
                print(str(regsF))

    # sleep 2s before next polling
    time.sleep(60)
    print("polling activo")
    
```

Fig. 5. Código de ejemplo de lectura de datos desde PLC.

C. Almacenamiento de la información

La aplicación escrita en lenguaje Python captura los datos e inmediatamente inserta esta información en la base de datos *MariaDB* que está instalada en la misma computadora. La información de los datos de proceso se inserta en una tabla de datos de proceso con la fecha y la hora de adquisición.

MariaDB es un sistema de gestión de bases de datos procedente de *MySQL* con licencia GPL. Está desarrollado por Michael (Monty) Widenius (fundador De *MySQL*) y la comunidad de desarrolladores de software libre [8].

D. Servidor web

En la computadora llamada "Servidor de Nivel 2" donde corre la aplicación *Python* que adquiere los datos, también se ha instalado un servidor web *Apache* que responde a las consultas web que provengan de distintos dispositivos remotos como computadoras, tablets o celulares. El servidor web genera páginas html resolviendo código php. Los clientes web invocan una página html que reside en el servidor y pueden hacerlo con parámetros adicionales, por ejemplo, con la fecha inicial y final del periodo de tiempo deseado de consulta de datos. El servidor corre código PHP que filtra, procesa y devuelve los datos requeridos. Para esto se genera internamente una consulta SQL (*structured query language*) a la base de datos *MariaDB* y luego devuelve la información en el formato JSON (*javascript object notation*) adecuado para la transmisión hacia el cliente. Estos datos enviados en formato JSON son procesados en el cliente por una librería *Javascript*, que los procesa y genera la visualización correspondiente en un gráfico del tipo x-t.

Se incluyen las referencias de archivos *javascript* que residen en el servidor y son necesarios para mostrar los datos en forma gráfica (librería de gráficos *AmCharts*) [9], y otras librerías adicionales que fueran necesarias. Se resalta aquí que todas estas librerías *javascript* son de uso libre, respondiendo a una licencia GPL o similar. En la Tabla I se muestra la función de los softwares utilizados y su tipo de licencia.

TABLA I. LISTADO DE SOFTWARE LIBRE UTILIZADOS CON SU FUNCION Y TIPO DE LICENCIA

Software utilizado	
Tipo de Licencia	Función
Python 3.4.3	
Python Software Found Licence	Lenguaje de Programación
XAMPP 5.6.12	
GNU GPL	Paquete de instalación
Apache webserver 2.4.16	
Apache License, version 2.0	Servidor web
PHP 5.5.28	
PHP License version 3.01	Leng. De progrmacion de servidor
MariaDB	
GNU GPL	Gestor de base de datos
AmChart v.3	
Free under linkware license	Lib. javascript para graficos
Pickaday	
BSD & MIT License	Lib. javascript para objeto
Moment JS	
MIT License	Lib. de javascript para fechas
PyModbusTCP 0.0.11	
MIT License	Lib. de comunicación cliente

E. Clientes web

El servidor web puede ser accedido por cualquier computadora y/o dispositivos móviles que esté conectada a la red. Solo se necesita en el cliente un navegador web actual que pueda interpretar los lenguajes *Html5*, *CCS*, y *Javascript*.

F. Confiabilidad y robustez

Un sistema de alimentación ininterrumpida (cuya sigla en inglés es *UPS*) impide el apagado del servidor ante cortes de tensión. Si el PLC y la HMI salen de servicio por un corte de alimentación en la ubicación remota, esto no resulta un problema ya que la programación *Python* en el servidor es capaz de reintentar la comunicación hasta recuperarla.

IV. VISUALIZACION Y RESULTADOS OBTENIDOS

La librería *javascript* utilizada para graficar se llama *AmCharts* v.3 y puede ser descargada de internet. La información mostrada puede tener varios formatos distintos. Uno de ellos es el formato x-t que sirve para mostrar información temporal (como es nuestro caso), pudiendo mostrar varias señales a la vez, colocar líneas límite, o bien barras verticales que sirven para medir tiempo entre eventos. Al gráfico se lo puede imprimir, grabar con formato de imagen (jpg) o con formato de datos para hoja de cálculo (xls) o con formato de documento portable (pdf); también se lo puede modificar con anotaciones, permitiéndose elegir el color de la anotación y luego grabar como imagen.

A continuación, se muestran algunas capturas de pantallas obtenidas con el presente sistema de gestión de datos (Fig. 6 y Fig. 7).

En la Fig. 6 se puede observar el comportamiento de la microrred durante un día completo, donde se puede ver el voltaje de las baterías (rojo), la corriente de los paneles PV (verde) que comienzan a generar alrededor de las 8:00am hasta las 18:00 pm, la corriente en las baterías (azul) que se están descargando hasta que comienza a generar los PV, la corriente de la carga (marrón) que se incrementa después de las 9:00am y también aumenta su consumo durante la noche. Finalmente se registra la corriente del aerogenerador (rosa) que no genera energía durante ese día.

La Fig. 7 presenta datos históricos de 10 días, donde puede verse el comportamiento de la microrred, identificando algunos días de mucho viento y por consiguiente generación del aerogenerador (martes 10 sept y jueves 16 sept.).

También se observa un día donde el sistema sale de servicio (jueves 12 sept) debido a que hubo varios días nublados y el voltaje de las baterías alcanzó su límite inferior apagando el sistema.



Fig. 6. Gráfico tipo x-t que muestra la variación en el tiempo de las corrientes y tensiones de la microrred (1 día).

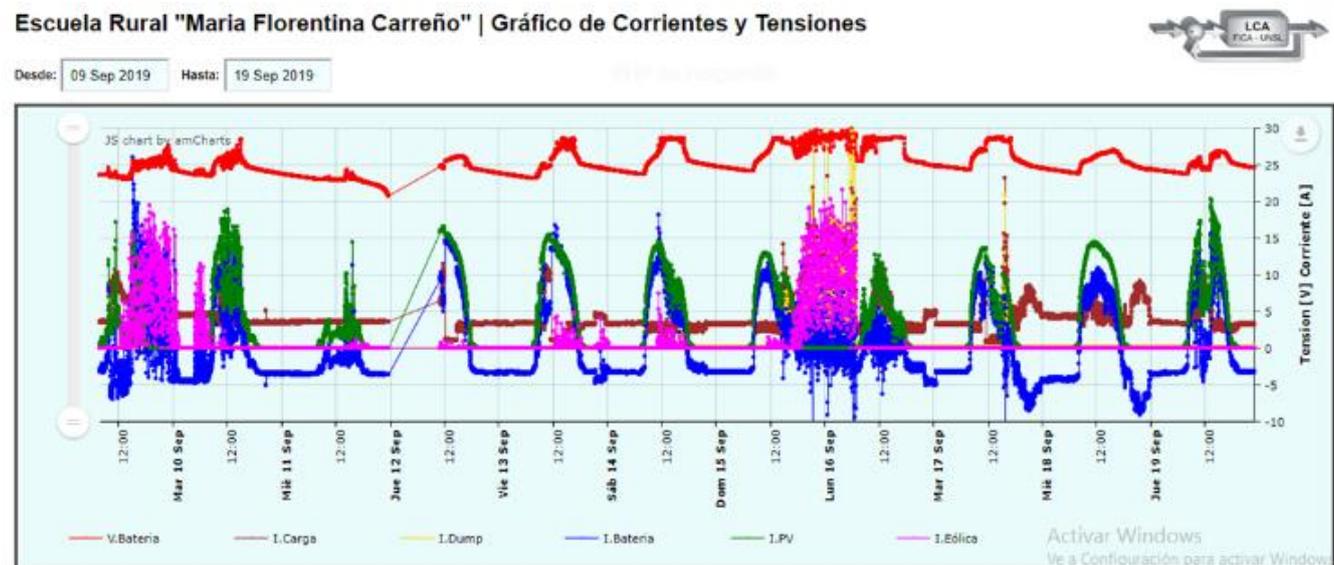


Fig. 7. Gráfico tipo x-t que muestra la variación en el tiempo de las corrientes y tensiones de la microrred (10 días).

V. TRABAJOS FUTUROS

En este trabajo se propone un sistema de gestión de datos con software abierto, que pueda ser replicado en aplicaciones de microrredes eléctricas. En la Fig. 3 se puede observar el PLC, el cuál es el dispositivo más costoso y puede ser un limitante para la implementación de este sistema. A partir de esto se está trabajando es un dispositivo de hardware abierto

que permita cumplir la función que realiza el PLC a un bajo costo y fácil de replicar.

Este dispositivo, denominado *OpenWee* [10], puede ser utilizado para aplicaciones eólicas de baja potencia con la capacidad de trabajar como regulador eólico, freno automático y también *datalogger* del sistema. El regulador eólico contempla las tres fases de carga de una batería, a

granel (*bulk*), absorción y flote, esto permite prolongar la vida útil de la misma. El freno automático se activa deteniendo el aerogenerador cuando se supera un voltaje por encima de las referencias de regulación, trabajando como protección. También es posible activarlo manualmente mediante un interruptor. Finalmente, el *datalogger*, permite monitorear diferentes variables de sensores y mediante wifi almacenarlas en una base de datos en la nube, para luego poder visualizarlas mediante celular o PC.

El dispositivo *OpenWee* está basado en un microcontrolador STM32FC106, y tiene disponible los siguientes periféricos (Tabla II).

TABLA II. CARACTERISTICAS DE LOS PERIFERICOS DEL OPENWEE

I/O	Tipo	Función
1 entrada DC Bus	Potencia	Control de DC Bus
1 salida R Dump	Potencia	Control de DC Bus
1 salida a Mosfet	Potencia	Regulador eolico
2 salidas a Rele	potencia	multipropósito
2 salidas digitales	Señal	multipropósito
5 salidas LED	Supervisión	Estado del OpenWee
1 salida Buzzer	Supervisión	Estado del OpenWee
6 entradas digitales	Señal	Multipropósito
5 entradas analógicas	Señal	Multipropósito
ESP8266 ESP-07	WIFI	datalogger

La transmisión de datos se realiza mediante WIFI a través de un módulo ESP8266 que permite comunicarse con el router de la escuela realizando la misma función que con el PLC. En la Fig. 8 se presentan imágenes con las características del *OpenWee*.

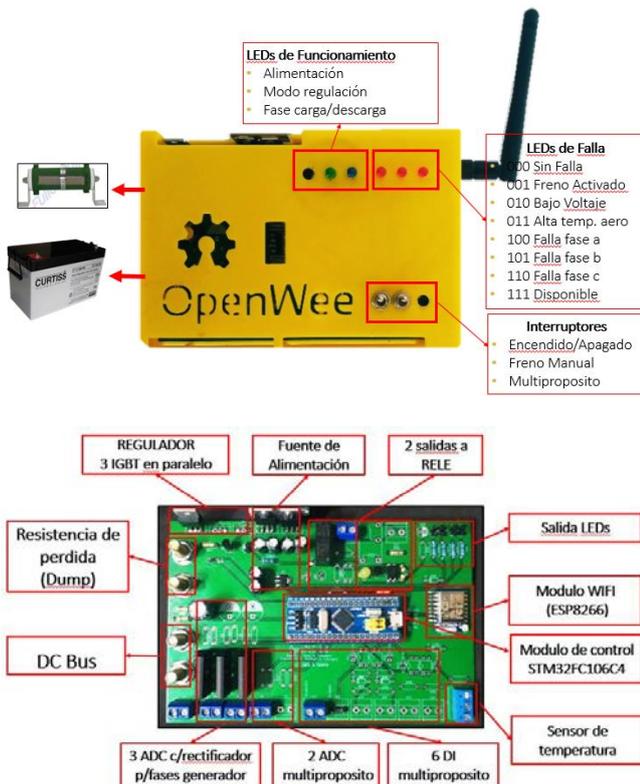


Fig. 8. Características del OpenWee.

La potencialidad del *OpenWee* (Fig. 9) reside en que al ser *open source* ofrece la posibilidad de modificarlo para que se adapte mejor a la aplicación y configuración de la microrred, incluyendo diferentes sensores y variables, también su bajo costo permite alcanzar un dispositivo asequible para las

comunidades rurales que generalmente son de bajos recursos.

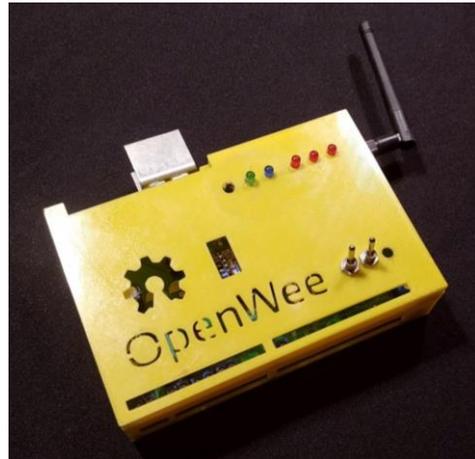


Fig. 9. OpenWee.

VI. CONCLUSIONES

A la fecha de escribir el presente trabajo el sistema lleva funcionando más de 1 año, demostrando ser robusto y confiable.

La presentación de los datos realizada con la librería AmCharts es muy conveniente porque es completamente customizable en cuanto a colores, tamaños, formas y funcionalidad. Todo el software utilizado es libre (open source), las licencias varían en los detalles, pero todas son variantes de la GPL, con excepción la licencia de la librería de gráficos AmCharts que es del tipo *linkware* (gratis si mantenemos el link a la página del desarrollador en una esquina del gráfico). El sistema completo resulta ser de muy bajo costo y con mucha potencialidad de uso.

Finalmente se propone como trabajo futuro, el reemplazo del programador lógico controlable por el dispositivo *OpenWee*, dispositivo abierto que permite realizar la misma función a un bajo costo y con la misma eficacia.

REFERENCIAS

- [1] Goel, S., Sharma, R., Performance evaluation of standalone, grid connected and hybrid renewable energy systems for rural application: a comparative review. *Renew. Sustain. Energy Rev.* 78(October), 1378 – 1389, 2017.
- [2] Khodayar, M.E. Rural electrification and expansion planning of off-grid microgrids. *Electr. J.* 2017, 30, 68–74.
- [3] A. Chauhan and R. Saini, "A review on Integrated Renewable Energy System based power generation for standalone applications: Configurations, storage options, sizing methodologies and control", *Renewable and Sustainable Energy Reviews*, 2014, vol. 38, pp. 99–129.
- [4] Guillermo R. Catuogno, Luis R. Torres, L. Proietti and Guillermo O. Garcia, "Methodology for the Selection and Sizing of an Isolated MicroGrid Based on Economic Criteria", *IEEE Latin America Transactions*, IEEE Latin America Transactions, 2019,
- [5] S. Cruz, N. Lastra, F. Patti, C. Martínez, F. Sosa, C. Catuogno, G. Frias, G. Acosta, L. Torres, L. Poietti, M. Perez Larraburu, G. Pleitavino, E. Van Dam, G. Garcia and G. Catuogno. "Metodología de Diseño e Implementación de una Microrred Aislada para Escuelas Rurales" SIMIER 2019, 3er Simposio Ibero-Americano en Microrredes Inteligentes con Integración de Energías Renovables, Itaipu, Brasil, 2, 3 y 4 de octubre de 2019.
- [6] Librería de comunicación ModbusTCP, url: <https://pypi.org/project/pyModbusTCP/>
- [7] Lenguaje de programación Python, url: www.python.org
- [8] Base de datos MariaDB, url: <https://mariadb.org/>
- [9] Librería de gráficos javascript AmCharts, url: www.amcharts.com
- [10] <https://github.com/grcatu/OpenWee>