IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE ENERGÍA ELÉCTRICA POR MEDIO DE LA INTEGRACIÓN DE UN EQUIPO DE MEDIDA, UNA PLATAFORMA DE DESARROLLO DE BAJO COSTO Y UNA INTERFAZ DE VISUALIZACIÓN.

SERGIO ALONSO FUENTES CASADIEGO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2019

IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE ENERGÍA ELÉCTRICA POR MEDIO DE LA INTEGRACIÓN DE UN EQUIPO DE MEDIDA, UNA PLATAFORMA DE DESARROLLO DE BAJO COSTO Y UNA INTERFAZ DE VISUALIZACIÓN.

SERGIO ALONSO FUENTES CASADIEGO

Trabajo de Grado para optar al título de:
Ingeniero Electrónico

Director:
Mg. Jaime Guillermo Barrero Pérez

Codirector:
Mg. Luis Fernando Rueda Vásquez

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
BUCARAMANGA
2019

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ANEXOS

RESUMEN

TÍTULO: IMPLEMENTACIÓN DE UN SISTEMA DE MEDICIÓN DE ENERGÍA ELÉCTRICA POR MEDIO DE LA INTEGRACIÓN DE UN EQUIPO DE MEDIDA, UNA PLATAFORMA DE DESARROLLO DE BAJO COSTO Y UNA INTERFAZ DE VISUALIZACIÓN*

AUTORES: SERGIO ALONSO FUENTES CASADIEGO**

PALABRAS CLAVE: SBC, medidor monofásico, Omega2+, Cayenne, circuito de medición, medidor inteligente, AMI.

DESCRIPCIÓN

Los cambios y transformaciones que ha sufrido el estilo de vida de los hombres, han estado caracterizados, entre otros aspectos, por el incremento del uso de la energía eléctrica, lo que ha llevado a iniciar un análisis sobre su uso, sobresaliendo la necesidad de controlar su manejo para evitar una posible crisis energética. Es por esto, que el presente trabajo presenta la implementación de un sistema de medición de energía eléctrica, que pueda ser monitorizado de forma remota por medio de la integración de un equipo de medida, una plataforma de desarrollo de bajo costo y una interfaz de visualización. Para realizar la medición de energía se utilizó un medidor monofásico de la marca YTL; para la captura y registro de datos, se aprovecharon las funciones de la Omega2+ de Onion®, y para la visualización de los datos medidos se empleó la herramienta web, Cayenne. En la aplicación del proyecto, se pudo observar que la integración de los 3 módulos (medición, captura y registro de datos, y transmisión y visualización) fue exitosa, obteniendo un dispositivo capaz de mostrar al usuario final, los parámetros propuestos (valores eficaces de tensión y corriente, potencias activa, aparente, reactiva, factor de potencia, etc.), tanto de manera instantánea, como histórica. Realizadas las comprobaciones y recolectadas las evidencias, se observó que para los datos de potencias acumuladas, el medidor monofásico sigue midiendo, a pesar de no estar conectado a la tarjeta Omega2+. Además, al estar los datos alojados en la infraestructura del PaaS Cayenne, estos se pueden acceder desde cualquier dispositivo y lugar del mundo.

---

* Trabajo de grado.

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez, Magíster en Potencia Eléctrica. Codirector: Luis Fernando Rueda Vásquez, Magíster en Electrónica.

ABSTRACT

TITLE: IMPLEMENTATION OF AN ELECTRIC POWER MEASURING SYSTEM THROUGH THE INTEGRATION OF A MEASURING DEVICE, A LOW COST DEVELOPMENT BOARD AND A GRAPHIC USER INTERFACE*

AUTHOR: SERGIO ALONSO FUENTES CASADIEGO**

KEYWORDS: SBC, Single Phase Energy Meter, Omega2+, Cayenne, Measuring Circuit, Smart Meter, AMI.

DESCRIPTION

Changes and transformations that had suffered mankind's lifestyle has been marked, among other aspects, by the raise on the use of electricity, a matter that has motivated people to start analyzing it, standing out over everything else, the necessity for controlling its handling and consumption, willing to avoid a hypothetical energetic crisis. For that reason, this project aims to materialize an electric power measuring system, capable of being monitored remotely through the integration of a measuring device, a low cost development board and a graphic user interface. For the power measurement was used a single phase energy meter (YTL®), for the data acquisition and logging, was taken advantage of the characteristics of the Omega2+ by Onion® and for the graphic interface was used a Platform as a Service tool named Cayenne. In the application of this project, was able to observe that the integration of the 3 modules (measurement, data acquisition and logging, and transmission and visualization) was successful, earning a device capable of showing to the potential user the measures proposed by its developers (power, current, power factor, etc.), in an instant way, as well as a historical way. After the tests and collection of evidence, it was noticed that if the energy meter was connected to a source and a load and has energy consumption, the kwh will continue raising up; also that as the data is stored in the PaaS (Cayenne), it can be accessed from anywhere in the globe and from any device with internet access.

* Degree project.

** Faculty of Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Jaime Guillermo Barrero Pérez, Master in Electric Power. Co-director: Luis Fernando Rueda Vásquez, Master in Electronics.

1. INTRODUCCIÓN

Desde el inicio de los tiempos, el hombre se ha caracterizado por estar en constante búsqueda de desarrollo y mejores condiciones de vida, y en esta exploración ha realizado grandes descubrimientos e innovaciones que le han permitido llegar al punto en el que se encuentra en la actualidad. Uno de los descubrimientos que cambió radicalmente la vida de los hombres fue la energía eléctrica, pero cabe resaltar que el mundo siempre ha estado rodeado de electricidad, ya que esta es parte de la naturaleza, entonces, lo que realmente hizo el hombre fue darle un uso y beneficiarse de ella. Según Roy Pura, uno de los primeros observadores de este fenómeno fue Tales de Mileto, 600 años a.c, por medio de los juegos con el ámbar, ya que, al frotarlo, este atraía pequeñas partículas.

Así pues, solo hasta el Siglo XVII, los investigadores se interesaron por el tema, y en 1660, William Gilbert se percató de que algunas sustancias como el vidrio y el azufre actuaban como el ámbar, atrayendo sustancias, fenómeno al que llamó "electricidad". Por su parte, en 1747, Benjamín Franklin, dedujo que no existían diferentes tipos de fluidos y que la electricidad era algo que se encontraba en todas las cosas. En 1897, es descubierto el electrón por J.J Thompson; por su parte Thomas Alva Edison, creó la bombilla eléctrica[1], y Nikola Tesla, considerado como el máximo genio de la electricidad, descubrió la corriente alterna, el sistema polifásico de distribución eléctrica, y el motor de corriente alterna.

La energía eléctrica empezó a ser un servicio necesario y básico para la vida de las personas. Según datos del Banco Mundial, en 1993, el 76% de la población mundial

[1] PURA, Roy. Breve historia de la electricidad. En: Panorama, 2004., p4-8.

tenía acceso a electricidad; mientras que, en el 2017, la cifra se ubicó en un 89%. Países como Alemania, China, EEUU, Francia, entre otros, cuentan con un cubrimiento del 100%; mientras que Colombia presenta una cifra de 99,6%[2]. Por el hecho que sea considerado como un servicio fundamental en la vida de las personas, y, por consiguiente, que se haya multiplicado su uso, empieza a ser importante regularlo y encontrar mecanismos más eficientes para su control, medición y conservación, ya que podría presentarse una crisis energética importante, que afectaría a millones de personas en el mundo, y cambiaría las formas y estilos de vida adoptados por los seres humanos en los últimos tiempos. Además, el uso eficiente de la energía ayuda al ahorro económico de las familias y al cuidado y conservación del medio ambiente.

Por lo anterior, el objetivo principal de este trabajo de grado es implementar un sistema de medición de energía eléctrica, que pueda ser monitorizado de forma remota por medio de la integración de un equipo de medida, una plataforma de desarrollo de bajo costo y una interfaz de visualización. Esto se logrará seleccionando una plataforma de desarrollo, teniendo en cuenta las características de hardware, software y su costo; se elegirá la interfaz de visualización que permita monitorizar el consumo de energía eléctrica de forma remota; y se realizarán pruebas de integración de la plataforma de desarrollo con el equipo de medida y con la interfaz de visualización. Para realizar la medición de los parámetros de energía eléctrica se usará un medidor monofásico y para la captura y registro de datos se utilizará, como unidad de procesamiento principal, la Omega2+.

A continuación, en la primera parte se hará la exposición de los marcos de referencia, dentro de los que se encuentran los marcos conceptual y teórico

[2] BANCO MUNDIAL. Acceso a la electricidad (% de población) [en línea]. < https://datos.bancomundial.org/indicador/EG.ELC.ACCS.ZS?end=2017&start=1990&view=chart> [citado el 15 de julio de 2019].

relacionado con el tema del trabajo de grado, así como el marco de antecedentes; posteriormente se dará a conocer la metodología utilizada, los equipos y servicios elegidos, así como el por qué fueron estos seleccionados. Por último, en el cuarto capítulo, se muestran los resultados de todo el sistema en funcionamiento, para dar respuesta a los objetivos del proyecto.

2. MARCO DE REFERENCIAS

En este capítulo se expondrá la revisión bibliográfica realizada para el desarrollo del trabajo de grado, donde se observarán los conceptos usados en el documento, y en las actividades y funciones establecidas para la implementación del sistema de medición de parámetros de energía eléctrica. Posteriormente, se presentará el marco teórico, referente a los aspectos fundamentales para el desarrollo del tema.

2.1 MARCO CONCEPTUAL

Para la realización del trabajo de grado se requiere de unos conocimientos conceptuales previos, los cuales fueron adquiridos en las aulas de clase y por medio de una revisión de documentos. A continuación, se presentan los términos y definiciones utilizados en el documento, con el fin de que el lector pueda entender con mayor facilidad lo que aquí se plasmará.

- **Interfaz gráfica de usuario**

Reconocido por su acrónimo GUI, debido a su nombre en inglés Graphic User Interface, “es un método para facilitar la interacción del usuario con el ordenador, por medio de la utilización de un conjunto de imágenes y objetos pictóricos, como íconos y ventanas, además de texto”[3]. La GUI es parte fundamental de cualquier aplicación, ya que cuando el usuario empieza a trabajar en una computadora, interactúa con la interfaz, por lo que esta es la encargada de ofrecer al usuario una interacción fluida y agradable. La interfaz es la encargada de informarle al usuario lo que es capaz de hacer el producto. La GUI puede ser definida de forma sencilla,

[3] UNIVERSIDAD DE SEVILLA. Interfaz gráfica de usuario [en línea]. < http://bibing.us.es/proyectos/abreproy/11300/fichero/PROYECTO%252FCapitulo3.pdf> [citado el 29 de junio de 2019] p17.

como la parte de la computadora y el software que los usuarios pueden ver, oír, tocar, hablar; y esta tiene dos componentes la entrada, que es la forma como los usuarios comunican sus necesidades a la computadora; y la salida, que es como la computadora transmite los resultados al usuario. Una interfaz debe ser cómoda, efectiva y eficiente, para que cumpla exitosamente con sus objetivos[4].

- **Internet de las cosas**

Este término hace referencia a "un sistema de dispositivos de computación interrelacionados, máquinas mecánicas y digitales, objetos, animales o personas que tienen identificadores únicos y la capacidad de transferir datos a través de una red"[5]. Otra definición es que el término "se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras"[6]. La IoT (siglas de su nombre en inglés Internet of Things), tiene cinco temáticas claves para estudiar sus desafíos y cuestiones tecnológicas que son: seguridad; privacidad, interoperabilidad; cuestiones legales, reglamentarias y de derechos; y las cuestiones relacionadas con las economías emergentes y de desarrollo[7]. Cuando se habla de este término, se está refiriendo a una tecnología basada en "la conexión de objetos cotidianos a internet, que intercambian, agregan y procesan información sobre el entorno físico, para dar valor agregado a los usuarios finales"[8]

[4] ALBORNOZ, Claudia; BERÓN, Mario y MONTEJANO,Germán. Interfaz Gráfica de Usuario: el Usuario como Protagonista del Diseño. En: Departamento de Informática, Universidad Nacional de San Luis. 2014., ,p570-574.
[5] ROUSE, Margaret. Internet de las cosas (IoT). En: Search Datacenter (enero 2017).
[6] ROSE, Karen; ELDRIDGE, Scott y CHAPIN, Lyman. La Internet de las Cosas- Una Breve Reseña. Internet Society, 2015. 83p, p5.
[7] ROSE, Op.cit., p7.
[8] BARRIO, Moisés. Internet de las cosas. Madrid: Editorial Reus S.A, 2018. 123p. ISBN: 978-84-290-2038-0., p19.

- **Medidor monofásico de energía eléctrica**

Es un dispositivo encargado de registrar el consumo energético del recinto en el cual se instala, como su nombre lo indica, se utiliza en sistemas de distribución de energía eléctrica de una sola fase. Existen dos tipos, los medidores monofásicos electromecánicos y los digitales, estos últimos han evolucionado hasta convertirse en *Smart Meters.*

- **Plataforma de desarrollo**

Es un sistema integrado por un microcontrolador y periféricos adicionales, que constituyen un entorno ideal para programar soluciones enfocadas a objetos. Estas pueden ser tarjetas de desarrollo primitivas o SBC.

- **SBC**

Son las siglas de “Single Board Computer”, que “son placas que contienen todos o la mayor parte de los componentes de un ordenador”[9]. Entre sus principales atractivos se encuentra su reducido tamaño y su módico precio, ya que la mayoría de ellas no sobrepasa los 100 dólares; por último, a pesar de que los SBC ofrecen poca potencia, estas tienen varias funcionalidades, de manera que lo que promete es más que suficiente para desarrollar las funciones para las cuales fue creada. Estas placas están compuestas por un algún microcontrolador de empresas como Microchip o Atmel, o por un procesador de alguna plataforma como ARM (*Advanced RISC Machines*); no tienen sistema operativo y responden a un lenguaje

[9] GARCÍA, Joaquín. ¿Qué es una placa SBC?. En: HardwareLibre.2015.

específico[10]. Entre las placas SBC se encuentran la Omega2, Raspberri Pi, Beaglebone Black, Hummingboard, ODroid-C2, Jaguar One, entre otros.

- **URE**

En el Artículo 1 de la Ley 697 de 2001 de Colombia, se define el URE como el Uso Racional y Eficiente de la Energía, el cual se ha convertido en un asunto de interés social para el aseguramiento de abastecimiento de energía a nivel nacional. "URE es el aprovechamiento óptimo de la energía en todas y cada una de las cadenas energéticas, desde la selección de la fuente, su producción, transformación, transporte, distribución, y consumo incluyendo su reutilización cuando sea posible, buscando en todas y cada una de las actividades de la cadena, el desarrollo sostenible"[11]. Por medio del URE, el gobierno pretende concientizar a la población en el cuidado de la energía, para poder seguir disfrutando de sus beneficios, bajo un uso racional y medido.

2.2 MARCO TEÓRICO

Con el auge de la tecnología y, por consiguiente, el incremento acelerado en el uso de la electricidad, los sistemas eléctricos en el país y en el mundo entero debieron pasar de ser entes pasivos, que simplemente se encargaban de comparar el consumo de los hogares para luego cobrar, a ser reguladores de consumo, debido al peligro en el que se encuentra la seguridad energética en el mundo. Este panorama ha alarmado a los prestadores de servicio, por lo que empezó a ser indispensable la optimización de los recursos, de modo que se comenzaron a aprovechar los avances tecnológicos que se tenían a la mano, por medio de las

[10] LIGHT PATH. Tarjetas Para Desarrollo De Hardware [en línea]. <http://www.lightpath.io/tarjetas-de-desarrollo/> [citado el 29 de junio de 2019].
[11] COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 697 (3, octubre, 2001). Mediante la cual se fomenta el uso racional y eficiente de la energía, se promueve la utilización de energías alternativas y se dictan otras disposiciones. Bogotá D.C. p.1.

cuales se emprendería un registro y control detallado en tiempo real sobre las variables del sistema eléctrico. La combinación de nuevas tecnologías con la comunicación para prestar un servicio más eficiente y a la vez evitar su escasez, producen los sistemas *Smart Metering.*

2.2.1 Smart Meters. Los Smart Meters son "dispositivos que tienen la capacidad de calcular el consumo de energía de una forma más detallada que los medidores convencionales, caracterizados porque permiten la comunicación de información a través de una red disponible hasta otros medidores, agregadores o centros de control"[12]. Además de este tipo de medidores, anteriormente existieron otros dos: el medidor electromecánico y el digital. El primero durante muchos años se consideró una maravilla de la ingeniería, ya que era una forma económica, durable y sencilla de contabilizar el consumo de energía, pero, aunque era muy efectivo para dar el resultado total del consumo, no proveía datos para otro tipo de factores. Por su parte, un medidor digital "muestrea las formas de onda involucradas en la medición y las procesa por medio de algoritmos lógicos y matemáticos. Estos medidores usan microcontroladores (MCU) o procesadores de señales digitales (DSP)"[13]. Igualmente, Valdiosera define los Smart Meters como "a aquellos con tecnología digital que son capaces de enlazarse a un sistema de comunicación bidireccional y vincular al usuario con la empresa suministradora del servicio, utilizando redes inalámbricas u otros medios de comunicación convencional"[14].

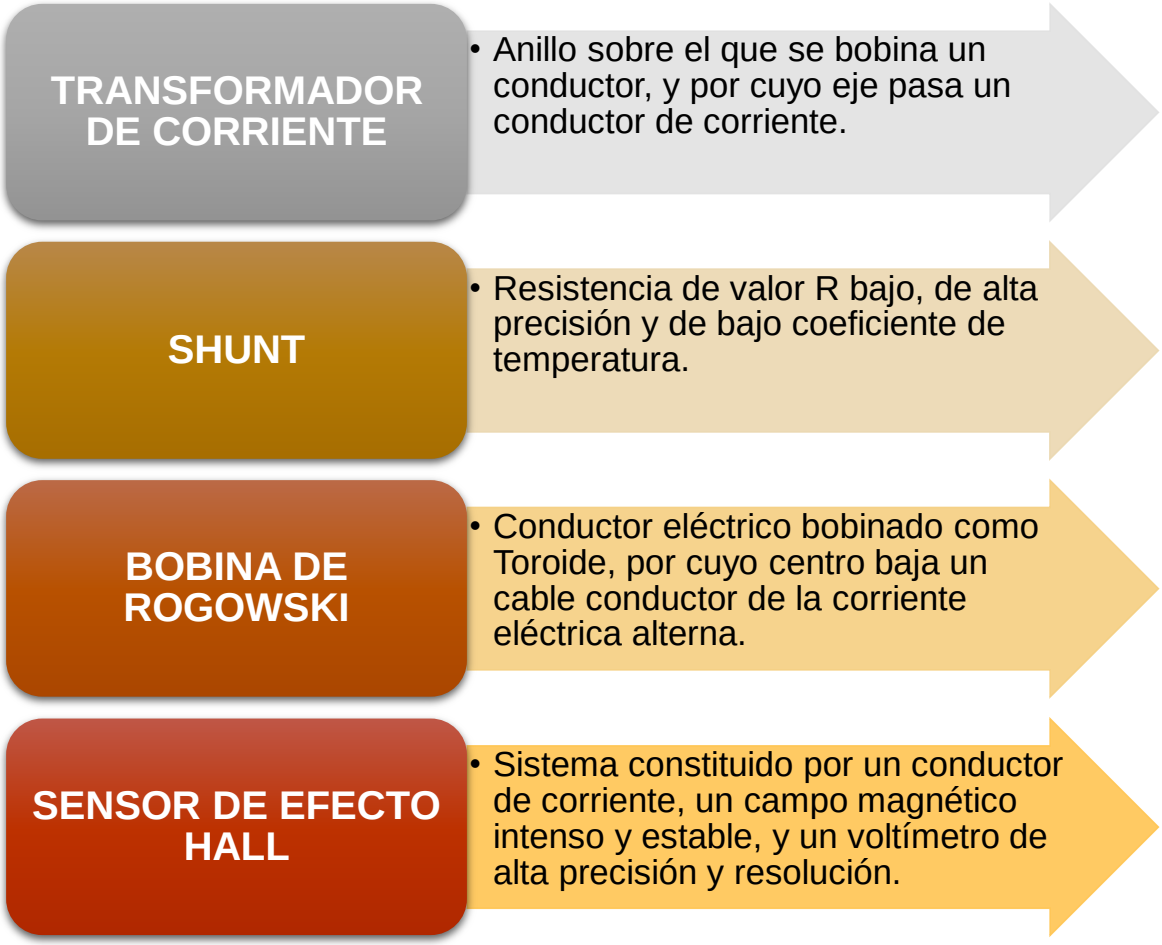
Entre las ventajas de los Smart Meters se encuentra que proporciona información requerida para la facturación; suministra información sobre la calidad de la energía

[12] SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO. Medición y gestión inteligente de consumo eléctrico. En: Boletín Tecnológico (junio de 2016)., p27.
[13] VALDIOSERA, Abraham. Diseño de medidor inteligente e implementación de sistema de comunicación bidireccional. México DF, 2013, 335p. Trabajo de grado (Maestro en Ciencias en Ingeniería Eléctrica). Instituto Politécnico Nacional. Escuela Superior de Ingeniería Eléctrica y Mecánica. Departamento de Ingeniería Eléctrica., p26.
[14] VALDIOSERA. Op.cit., p27.

eléctrica de un usuario; permite eliminar el uso de aparatos portátiles para la toma de lecturas; evita tener que realizar una conexión y reconexión manual; detección temprana de robo de energía; proporciona mejor manejo de carga en transformadores; provee información de la eficiencia, confiabilidad, pérdidas y distribución de carga; produce una reducción de costos en recopilación de datos; genera un mejor estudio de perfil de riesgo; y reduce accidentes en operadores[15]. Ahora bien, para su funcionamiento, los Smart Meters necesitan de sensores, ya que ellos envían la información del consumo de energía, y "para lograrlo, es necesario usar un sensor o transductor que convierte un tipo de señal eléctrica en una de otro tipo, que replica fielmente las variaciones de la señal original"[16]. Este tipo de medidores usa uno de los cuatro tipos de transductores mostrados en la Figura 1.

[15] Ibid., p27.
[16] TORRES, Rafael. Medidores inteligentes. En: Biblioteca del Congreso Nacional de Chile. (marzo 2019). 8p., p3.

Figura 1. Tipos de transductores.



Fuente: TORRES, Rafael. Medidores inteligentes, 2019.

2.2.2 Infraestructura de Medición Avanzada-AMI. Es un tipo de sistema que plantea una infraestructura de comunicación bidireccional, la cual posibilita el intercambio de información entre la empresa de suministro de energía eléctrica y el Smart Meter, explicado anteriormente, ubicado en los predios de los clientes. Además de esto, el AMI se encarga de recolectar información del consumo

energético y demás parámetros eléctricos[17]. El Instituto de Energía Eléctrica, define este sistema como “el sistema de colección que incluye medidores (Smart Meters) en el sitio del cliente; redes de comunicación entre el cliente y el proveedor de servicios, recepción de datos, y un sistema de gestión que facilite la información para el proveedor de servicios”[18]. Por su parte, el Ministerio de Minas y Energía de Colombia, en su Resolución 40072 del 29 de enero de 2018, que tiene como objeto principal el establecimiento e implementación de los lineamientos respecto a política energética en materia de sistemas de medición avanzada, define el AMI como “la infraestructura que permite la comunicación y que integra hardware, software, arquitectura y redes de comunicaciones que permiten la operación de los datos del sistema de distribución de energía”[19]. El sistema AMI se estructura en cinco módulos que son: unidad de medida, unidad concentradora, sistema de operación y gestión, comunicaciones y seguridad. Para objeto del trabajo de grado, cabe resaltar que la unidad concentradora “es un elemento intermedio entre la unidad de medida y el sistema de gestión y operación, que opera como puerto de enlace y almacenamiento”[20].

2.2.3 Uso de los SBC como Concentradores o Dataloggers. Los SBC se han convertido en una herramienta importante para la monitorización de diversas variables, que pueden ser captadas por infinidad de sensores, ya disponibles en el mercado; incluyendo aplicaciones ambientales, *Smart Artificial Vision Systems* y *Smart Cities*. Una de las peculiaridades de estas plataformas de desarrollo, es que

[17] CORONEL, Marco. Estudio para la implementación del sistema de infraestructura de medición avanzada (AMI) en la Empresa Eléctrica Regional Centro Sur C.A. Cuenca, Ecuador, 2011, 270p. Trabajo de Grado (Ingeniero Eléctrico). Universidad Politécnica Salesiana. Facultad de Ciencias Eléctricas., p62.

[18] EPRI. Advanced Metering Infrastructure (AMI) [en línea]. < https://www.ferc.gov/eventcalendar/Files/20070423091846-EPRI%20-%20Advanced%20Metering.pdf> [citado el 29 de junio de 2019].

[19] COLOMBIA. MINISTERIO DE MINAS Y ENERGÍA. Resolución 40072 (28, enero, 2018). Por el cual se establecen los mecanismos para implementar la Infraestructura de Medición Avanzada en el servicio público de energía eléctrica. Bogotá D.C. p4.

[20] MORALES, Tatiana. Infraestructura de medición avanzada para microrredes eléctricas. Bogotá, Colombia, 2018, 104p. Trabajo de grado (Maestría en Ciencias de la Información y Comunicación). Universidad Distrital Francisco José de Caldas. Énfasis en Teleinformática., p19.

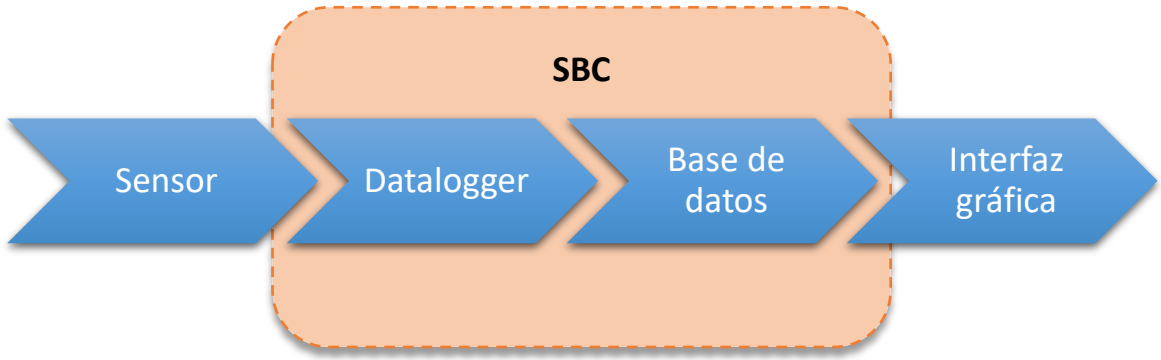
le ofrecen al usuario facilidad en su uso, y por sus características, permiten un aprendizaje rápido, convirtiéndolas en potenciales fuentes de proyectos de “hágalo usted mismo”. A pesar de sus capacidades limitadas, dichas plataformas ofrecen interconectividad con dispositivos de adquisición de datos y convertidores analógico-digital, gracias a su amplia baraja de conexiones y/o puertos de comunicación.

Ahora bien, un *datalogger* o registrador de datos “es un dispositivo electrónico que registra datos en el tiempo o en relación a la ubicación por medio de instrumentos y sensores propios o conectados externamente”[21]. La mayoría de *dataloggers* están basados en microcontroladores, lo cual permite que se utilice un SBC para implementarlos. “Un microcontrolador es un circuito integrado que en su interior contiene una unidad central de procesamiento, unidades de memoria (RAM y ROM), y puertos de entrada y salida periféricos”[22]. Los *datalogger* son pequeños, tienen y están equipados, como se mencionó anteriormente, por un microprocesador, memoria interna y sensores; algunos de estos *dataloggers* se comunican con un ordenador personal, mientras que otros cuentan con un dispositivo de interfaz local, usado en ciertas ocasiones como dispositivo independiente[23]. Su uso puede ser general o específico, este último ocasionalmente para medir variables ambientales o aplicaciones particulares. Entre sus beneficios, se encuentra el hecho que sirve para recopilar datos automáticamente las 24 horas del día, y los costos de los registradores de un solo canal de datos son reducidos, mientras que los que miden variables específicas, presentan un valor más alto.

[21] FINALTEST. ¿Qué es un Datalogger? [en línea]. <https://www.finaltest.com.mx/product-p/art-4.htm>. [citado el 29 de junio de 2019].
[22] NOVAS, Despradel. Microcontroladores: Arquitectura, programación y aplicación. En: Atlantic International University. Honolulu, Hawai. 2008. 31p., p5.
m[23] FINALTEST. Op.cit.

Por su parte, los SBC permiten implementar un *datalogger*, gracias al hardware y a la facilidad de programación. Habiendo montado el *datalogger* en un SBC, este último con una interfaz de comunicación, permitirá establecer conexión de los datos almacenados a motores de bases de datos (remotos o locales), servidores web, y, por último, interfaces gráficas para visualización en tiempo real o histórico, y demás aplicaciones que quiera darle el usuario final[24]. En la Figura 2 se puede observar un diagrama que describe la ruta de la información capturada usando un SBC.

Figura 2. Ruta de la información capturada.



Fuente: Elaboración propia.

2.3 MARCO DE ANTECEDENTES

Con la amenaza de una crisis energética y la necesidad de regular su consumo, este comenzó a ser un tema fundamental y sobre el cual se empezaron a basar manuales y trabajos de grado, en los que además de proponer formas de regulación, se crearon sistemas de medición, todos ellos, respondiendo a las

[24] GURDITA, Akshay; VOVKO, Heather y UNGRIN, Mark. A Simple and Low-Cost Monitoring System to Investigate Environmental Conditions in a Biological Research Laboratory. En: PLOSONE.11(1) (enero 2016). DOI: 10.1371.

nuevas tendencias de tecnología y desarrollo. A continuación, se presentarán algunos estudios internacionales, en Colombia y Santander, realizados previamente. Es así como en el 2016, Diego Israel Samaniego y Diego Fernando Velesaca de la Universidad Politécnica Salesiana con sede en Cuenca, Ecuador, realizaron un trabajo de grado para obtener el título de "Ingenieros Electrónicos", titulado "Diseño e Implementación de un Medidor de Energía Electrónico, para Vivienda, con Orientación a la Prevención de Consumo y Ahorro Energético". Para poder llevarlo a cabo, utilizaron un ordenador de placa reducida, Raspberry Pi, y una plataforma Arduino, el cual se encargó de procesar los datos provenientes de los elementos de medición y por un módulo GSM, enviaría al usuario lecturas de la energía consumida y su respectivo valor en dólares. Ahora, la Raspberry contó con una base de datos que permitió almacenar la información del consumo de energía; además de esto, uno de los objetivos del estudio era que se podría monitorizar desde un sitio web o mensaje de texto el consumo de energía. Luego de su implementación, los autores realizaron unas pruebas y análisis, además una evaluación de rentabilidad del proyecto, con lo que concluyeron que su implementación era viable[25].

A nivel nacional, Diego Fernando Mesa y Camilo Ricardo Rojas de la Universidad Santo Tomás de Bogotá, llevaron a cabo la creación de un sistema domótico de seguridad y consumo para hogares, donde su objetivo general fue "diseñar, desarrollar e implementar un producto tecnológico para hogares basado en la monitorización de la red de energía eléctrica integrado a una red de seguridad, desarrollado bajo la metodología SCRUM, viable para su desarrollo y comercialización en el mercado colombiano[26], esto por medio de una identificación

[25] SAMANIEGO, Diego y VELESACA, Diego. Diseño e Implementación de un Medidor de Energía Electrónico, para Vivienda, con Orientación a la Prevención de Consumo y Ahorro Energético. Cuenca, Ecuador, 2016, 122p. Trabajo de Grado (Ingeniero Eléctrico). Universidad Politécnica Salesiana. Facultad de Ciencias Eléctricas.

[26] MESA, Diego Fernando y ROJAS, Camilo Ricardo. Creación de un Sistema Domótico de Seguridad y Consumo Energético para Hogares: Homenode. Bogotá,Colombia, 2015, 160p. Trabajo de Grado (Ingeniero de Telecomunicaciones). Universidad Santo Tomas. Facultad de Ingeniería de Telecomunicaciones.

de los elementos usados para la red domótica y el estudio de las soluciones domóticas existentes; posteriormente realizaron el diseño e implementación de un prototipo de red domótica para controlar variables eléctricas, ambientales, económicas, de seguridad y confort; y por último, aplicaron un modelo de negocio para determinar la viabilidad económica del proyecto. El logro de los objetivos se llevó a cabo por medio de una metodología de investigación documental y metodología SCRUM para el desarrollo del proyecto como tal. En el proyecto, los autores elaboran un plan de negocios basado en la metodología de propuesta por el Fondo Emprender. Durante el diseño e implementación de la red domótica, los autores se encontraron con algunos inconvenientes de hardware y software, como el ruido electromagnético para el primero, por lo que aconsejaron mantener aislado los circuitos de alimentación. En cuanto al software, se concentraron en mantener la estabilidad de la conexión y reforzar la seguridad en la red, donde los estudiantes concluyeron que para evitar inconvenientes en el acceso a usuarios, HomeNode implementó una solución mediante la técnica de polling, y en el estudio financiero realizado, los estudiantes concluyeron que el proyecto es viable[27].

Por su parte, en el 2016, Gerardo Guacaneme y Didier Alexis Pardo, realizaron un proyecto donde diseñaron e implementaron un sistema de medición de consumo de energía eléctrica y agua potable remoto, para el cual usaron una plataforma de software de comunicación y almacenamiento de datos, y una plataforma de software de interfaz al usuario dentro del marco del internet de las cosas. Para su proyecto, los autores eligieron un medidor de potencia eléctrica monofásica ADE7753, basado en un sistema de registros, con interfaz, serial y salida de pulsos; para el módulo de comunicación, se seleccionó el ESP8266 destacado por sus reducidas dimensiones y su bajo costo. Para el análisis de los datos, se trabajó con cadenas de caracteres y con Node.js. El sistema creado por este proyecto, permitiría tener una interacción

[27] MESA, op.cit.

con la identidad virtual, desde cualquier por medio del uso de dispositivos inteligentes, y por último, para mantener la comunicación entre los objetivos IoT y la red, utilizaron el protocolo Websocket[28].

Edinson José Higuera y Juan Andrés Salazar, en el 2016, de la Universidad Industrial de Santander, realizaron un proyecto titulado "Desarrollo de una Interfaz de Usuario para Monitorizar Remotamente el Consumo de Energía Eléctrica por Circuito Ramal en una Instalación de Tipo Residencial Bajo la Perspectiva del Hogar Inteligente". Para su ejecución los autores usaron un sistema de monitorización dividido en tres bloques: sensor-actuador, concentrador y visualizador de datos. Utilizaron un sensor DDS-1Y, un concentrador beaglebone, y una base de datos MySQL nombrada smarthone_uis. Diseñaron una interfaz de usuario basada en un entorno web, y para la implementación de esta página web utilizaron PHP, HTML, CSS y JavaScript. Higuera y Salazar llevaron a cabo pruebas en dos fases. En la fase 1 pruebas del sistema emulando el funcionamiento de los medidores por una script, generando datos en Python y envío de información a MySQL; mientras que en la fase 2 las pruebas se realizaron para hacer uso de los medidores inteligentes reales, implementando la comunicación por medio del protocolo RS-485[29].

En 2017, para el IX Simposio Internacional sobre Calidad de la Energía Eléctrica, se presentó un artículo fruto del trabajo conjunto entre la Universidad Industrial de Santander y la empresa Electrificadora de Santander, en cabeza del profesor Gabriel O. Plata titulado: "Sistema de medición centralizada en redes de distribución

[28] GUACANEME, Gerardo y PARDO, Didier Alexis. Diseño e implementación de un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto "Internet de las Cosas", 2016, 124p. Trabajo de Grado (Ingeniero Electrónico). Universidad Distrital. Facultad de Ingeniería Electrónica.

[29] HIGUERA, Edinson José y SALAZAR, Juan Andrés. Desarrollo de una Interfaz de Usuario para Monitorizar Remotamente el Consumo de Energía Eléctrica por Circuito Ramal en una Instalación de Tipo Residencial Bajo la Perspectiva del Hogar Inteligente, 2016, 57p. Trabajo de Grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-mecánicas.

de baja tensión para la reducción de pérdidas eléctricas.", el cual tiene como objetivo la detección, reducción y seguimiento de las pérdidas de energía eléctrica atribuidas a actividades indebidas como: manipulación de medidores y de las redes eléctricas. El sistema fue diseñado e implementado considerando los esquemas de medición inteligente "Smart Metering" desarrollados a partir de los conceptos de la Infraestructura de Medición Avanzada (AMI) y permite realizar seguimiento en tiempo real sobre el consumo de energía eléctrica, vigilar la manipulación de equipos y gestionar la conexión de los clientes del servicio de energía eléctrica.[30]

Por último, también en la Universidad Industrial de Santander, Railin Santiago Toloza y Julián Andrés Piñeres elaboraron un proyecto de control remoto de encendido y apagado, y monitorización del consumo de energía eléctrica de un sistema de iluminación. Para lograrlo, usaron un sensor de detección de presencia para controlar el encendido y apagado. Desarrollaron una interfaz que permitiera realizar el encendido y apagado de una bombilla, a partir de un equipo móvil conectado a internet; y establecieron una comunicación serial entre el medidor de energía eléctrica y una tarjeta de desarrollo. El control permite cortar la energía si el usuario lo desea o si se empiezan a exceder los límites de consumo; todo esto desarrollado a partir de una tarjeta llamada NodeMCU, que no es un SBC pero cuesta tan solo 5 dólares y tienen conexión wifi. Utilizaron un medidor monofásico de condiciones similares, un relé (dispositivo que enciende o apaga la carga, dependiendo de la orden que se le dé), este relé también es particular porque no es un relé electromecánico, sino de estado sólido, lo que implica menor costo energético y mayor duración, este relé de estado sólido no lo compraron, sino que

[30] ORDÓÑEZ PLATA, Gabriel, *et al*. Sistema de medición centralizada en redes de distribución de baja tensión para la reducción de pérdidas eléctricas, IX Simposio Internacional sobre Calidad de la Energía Eléctrica, 2017.

lo hicieron "casero". Además, utilizaron una App diseñada para conectarse con tarjetas de desarrollo y a partir de ejemplos crearon el código[31]

[31] TOLOZA, Railin Santiago y PIÑERES, Julián Andrés. Control remoto de encendido y apagado y monitorización de consumo de energía eléctrica de un sistema de iluminación, 2017, 84p. Trabajo de Grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-mecánicas.

3. METODOLOGÍA

Para poder desarrollar un trabajo de grado, es necesario tener claros los mecanismos y equipos usados para el logro de los objetivos. En este capítulo, se presenta la metodología del proyecto, donde se exponen los equipos seleccionados para la medición, captura y registro de datos de energía eléctrica, así como para la transmisión de información y las formas de visualización.

3.1 MEDICIÓN DE ENERGÍA ELÉCTRICA

Para realizar la medición de energía eléctrica se hizo uso del medidor monofásico de la marca YTL el cual fue suministrado por la Universidad Industrial de Santander. Este medidor cuenta con un sistema de comunicación RS485 y con un protocolo *Modbus*, a través del cual envía datos de parámetros de valor eficaz de tensión y corriente, potencia activa, potencia reactiva, potencia activa acumulada, potencia reactiva acumulada y factor de potencia. Adicionalmente, muestra en el visualizador integrado los valores actuales de potencia activa acumulada, potencia activa, valores eficaces de tensión y corriente, frecuencia y factor de potencia. Los parámetros de este medidor monofásico se muestran en la Tabla 1.

Tabla 1. Parámetros técnicos del medidor monofásico YTL

Parámetro	Valor
Tensión	127V (Máx.)
Corriente	10A (65A Máx.)
Clase de Precisión	1,0
Estándar	EC62052-11, IEC62053-21, IEC62053-31

Parámetro	Valor
Frecuencia	50-60 Hz
Consumo de energía	≤1W ≤7VA
Baud rate	9600

Fuente: D119001-Zhejiang Yongtailong Electronic Co., Ltd disponible en: http://www.ytl-e.com/product/info/157

Figura 3. Medidor monofásico YTL

Fuente: D119001-Zhejiang Yongtailong Electronic Co., Ltd disponible en: http://www.ytl-e.com/product/info/157 .

La Figura 3 muestra el medidor monofásico YTL, el cual almacena las variables mencionadas anteriormente y pueden ser consultadas por el sistema de captura y registro de datos de energía, a través de la comunicación RS485 de la forma que se expone adelante.

Existen otros medidores monofásicos digitales en el mercado, que cumplen con la misma función de este que fue utilizado en este proyecto, puesto que cuentan con

las mismas especificaciones eléctricas, puerto de salida RS485 y protocolo de comunicación Modbus, entre esos medidores están: el LEM012SJ de CHT, el SPM91 de Pilot o el SDM120 de EASTRON (ver Figura 4); lo cual proporciona una ventaja, ya que permite que el software y hardware desarrollados en este trabajo de grado se pueda ajustar para que funcione con otros dispositivos de medición.

Figura 4. Otros medidores monofásicos del mercado con Modbus.

Fuente: Alibaba.com, 2019.

3.2 CAPTURA Y REGISTRO DE DATOS DE ENERGÍA ELÉCTRICA

El medidor monofásico utilizado ya cuenta con registro de datos, pero estos son simples módulos de memoria en los cuales se sobrescriben las magnitudes medidas cada vez que el medidor actualiza los parámetros medidos; esto anula la posibilidad de acceder a un histórico de la información. Por lo tanto, se hace necesario almacenar los datos que se quieren analizar y visualizar a futuro en la interfaz gráfica.

El módulo de captura y registro de datos está conformado, principalmente, por un SBC, ya que estos equipos proveen escalabilidad, amplia conectividad, rapidez de procesamiento y versatilidad; con el propósito de hacer de este sistema apto para

más medidores y/o más variables físicas que quieran estudiarse a futuro, en otros proyectos. Este módulo es el más importante del sistema de medición, pues en este se concentran los datos y se envían al módulo de interfaz gráfica.

Se escogió un SBC debido a que estas plataformas de desarrollo cuentan con sistema operativo, en la mayoría de los casos, basado en Linux; lo que permitirá la instalación de programas para la transmisión y visualización de datos, la instalación de motores de bases de datos, e incluso la integración con aplicaciones basadas en la nube que permiten controlar estos dispositivos y graficar los datos que estos procesen. Las opciones de SBC en el mercado actual, resultan abundantes, por lo que para la elección correcta de *Single Board Computer* para este proyecto, el primer criterio de elección fue el tamaño. Encabezan la lista de menor tamaño las tarjetas Raspberry Pi Zero y la Omega 2, entre las más populares.

Una desventaja importante de la Raspberry Pi Zero, es que no cuenta con comunicación WiFi, pero existe una versión más reciente, llamada Raspberry Pi Zero W, que sí cuenta con esta, elevando su precio un poco. Por otro lado, la Omega2 cuenta únicamente con 16MB de memoria flash, lo cual limita la capacidad de almacenamiento de los datos capturados, además de no contar con una ranura para tarjeta microSD para expandir la memoria; pero su sucesora, la Omega2+, sí cuenta con esta opción. Por esta razón, se consideraron la Raspberry Pi Zero W y la Omega2+, ya que ambas versiones aumentan su valor casi en igual proporción, rodeando los 10 USD; haciendo del precio el segundo criterio de elección.

Tabla 2. Comparativo de especificaciones técnicas entre Raspberry Pi Zero W y Omega2+.

Característica	Raspberry Pi Zero W	Omega 2+

WiFi b/g/n	Sí	Sí
Bluetooth	Sí (4.1 y Bluetooth Low Energy)	No (solo con hardware adicional)
Almacenamiento en tarjeta	No (Únicamente con memoria microSD externa)	Sí (32 MB expandibles con memoria microSD externa)
CPU	1GHz	580 MHz
Memoria	512 MB	128 MB
USB	Soporte de USB 2.0 (con hardware adicional)	Soporte de USB 2.0 (con hardware adicional)

Fuentes: Omega2+ -Onion. [En línea]. Disponible en: https://onion.io/store/omega2p/ .Buy a Raspberry Pi Zero W – Raspberry Pi. [En línea]. Disponible en: https://www.raspberrypi.org/products/raspberry-pi-zero-w/

Como puede observarse en la Tabla 2, la Raspberry Pi cuenta mayor capacidad de procesamiento al tener mayor velocidad en la CPU y mayor capacidad de memoria RAM; sin embargo, la Raspberry Pi está diseñada para conectarse a una pantalla y controlar con un teclado y *mouse,* inclusive utilizando un entorno gráfico del sistema operativo, lo que requiere mayores recursos para operar; mientras que la Omega2+ está diseñada para ser operada en un entorno gráfico, pero al que se accede a través de un navegador en otro ordenador, consumiendo así recursos de dicho ordenador y no de la Omega2.

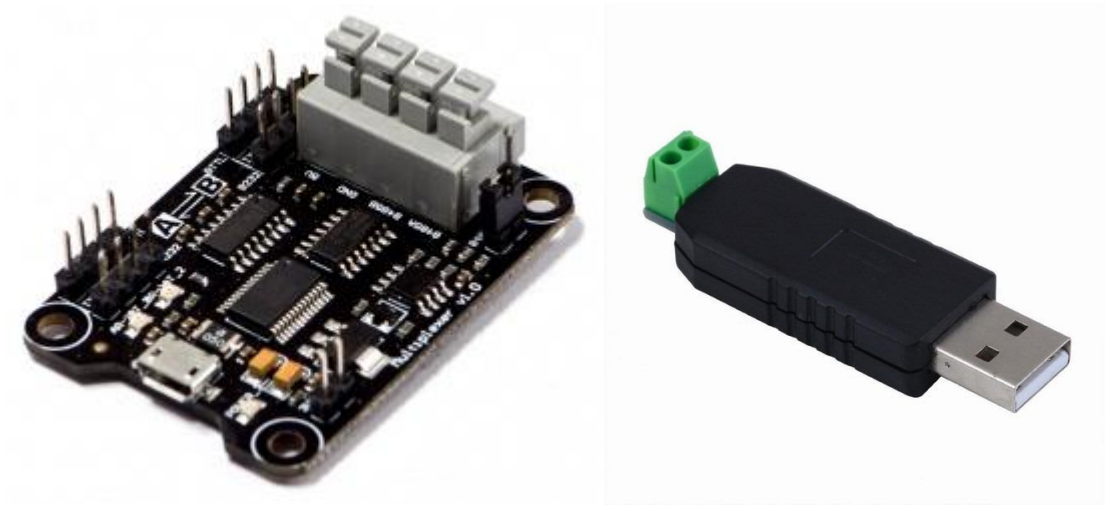
Teniendo dos de los SBC más pequeños del mercado, ambos con conexión WiFi, capacidad de memoria expandible y precios similares; además de corroborar la capacidad que ofrece el hardware de cada tarjeta para llevar a cabo el desarrollo que requiere este trabajo de grado, es entonces este primer atributo (el tamaño), el criterio que define a la Omega2+ como el SBC idóneo para el desarrollo del ptrabajo

de grado, debido a su reducida área de aproximadamente[32] 12 cm^2, contra los 19,5 cm^2 de la Raspberry Pi Zero W[33].

3.2.1 Hardware adicional. Debido a la carencia de puertos RS485 en las tarjetas de desarrollo, usualmente se opta por adherir al sistema una tarjeta que permita que esta comunicación serial se logre. En este caso se utilizó la tarjeta Multiplexer v1.0 de la empresa DFRobot, convertidor que cuenta con interfaces RS232, TTL, RS485 y USB y que permite la comunicación en dos vías entre estas interfaces. Alternativamente, para cumplir esta misma función, se usó también un convertidor de USB a RS485 de fabricación nacional que tiene un costo de aproximadamente el 15% (3 USD) del multiplexor de DFRobot (ver Figura 5). Por otro lado, por facilidad de conexión de alimentación y puertos USB, se utilizó un muelle de poder de Onion (desarrollador de la Omega2+) que facilita la comunicación con hardware adicional y la conexión de fuentes de energía convencionales, como el cargador de un celular (ver Figura 6). En la Figura 7 se representan las conexiones cableadas entre estos dispositivos con el medidor monofásico de energía eléctrica.

[32] ONION OMEGA2. Onion Omega Documentation [en línea]. < https://docs.onion.io/omega2-docs/omega2p.html> [citado el 29 de junio de 2019].
[33] PROTONEER. Raspberry Pi Zero Footprint And Dimensions [en línea]. < https://blog.protoneer.co.nz/raspberry-pi-zero-footprint-dimensions> [citado el 29 de junio de 2019].

Figura 5. Fotografía del convertidor Multiplexer v1.0 y del convertidor de USB a RS485 de fabricación nacional

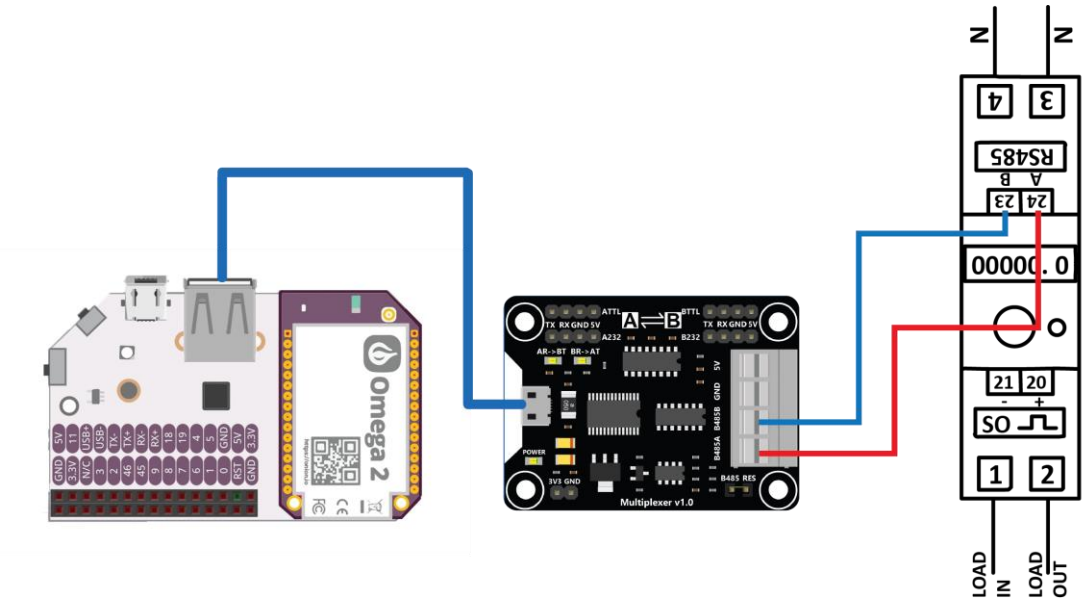


Fuentes: Multi_USB_RS232_RS485_TTL_Converter__SKU_TEL0070_-DFRoboto. [En línea]. Disponible en: https://wiki.dfrobot.com/Multi_USB_RS232_RS485_TTL_Converter__SKU_TEL0070_
Convertidor USB a RS485. | electronilab.co [En línea]. Disponible en:https://electronilab.co/tienda/convertidor-usb-rs485/

Figura 6. Fotografía del *Expansion Dock* de la Omega2+.

Fuente: Omega Expansion Dock BricoGeek | BricoGeek.com. [En línea]. Disponible en: https://tienda.bricogeek.com/onion-omega2/920-omega-expansion-dock.html

Figura 7. Esquema de conexiones cableadas



Fuente: Elaboración propia.

3.2.2 Software instalado. Tras la adquisición de la Omega2+, se instaló software adicional en el pc desde el cual se va a manejar, si el sistema operativo es Windows, deberá instalarse un servicio llamado ***Bonjour***; si es una versión de Linux, se debe instalar el servicio *Zeroconf,* y el sistema operático es OS X (MAC) no necesita que se instalen servicios adicionales. El sistema operativo de la Omega2+ ya viene previamente instalado en la tarjeta. Vale la pena recordar que una característica relevante de la Omega2+ es la posibilidad de controlarla desde un equipo remoto, ya sea a través del terminal (línea de comandos o *prompt*) o a través de una interfaz gráfica web, que emula la interfaz gráfica de un sistema operativo convencional; por esto es importante instalar los servicios mencionados anteriormente.

En la Omega2+, se actualizó el firmware y los aplicativos instalados por defecto, y se instalaron los aplicativos requeridos más adelante para el entorno de visualización de los datos de energía y esenciales para el funcionamiento del sistema como Python, PIP (*python-pip*), Nano, Serial (*pyserial*), Sudo, Git (*git-http*,

ca-bundle), OLED (*python-light*, *pyoledexp*), Cayenne (cayenne-mqtt), de la siguiente forma[34]:

`opkg update` (Actualización del gestor de paquetes del *OS* de la Omega 2+)

`opkg install python` (Instalación de *python*)

`pip install --upgrade pip` (Instalación del gestor *PIP*)

`pip install nano` (Instalación del editor de texto *Nano* vía *PIP*)

`opkg install nano` (Instalación del editor de texto *Nano* vía *OPKG*)

`pip install pyserial` (Instalación de la librería de comunicación serial)

`opkg install sudo` (Instalación de permisos de administrador *SUDO*)

`opkg install git git-http ca-bundle` (Instalación de *GIT*)

`opkg install python-light pyOledExp` (Instalación de la pantalla *OLED*)

`sudo pip install ISStreamer` (Instalación de librería para Initial State)

`cd Cayenne-MQTT-Python` (Instalación de librería para Cayenne)
`python setup.py install`

3.2.3 Programa. El programa realizado para la captura y registro de los datos medidos, parte del programa principal de medición inteligente, cuyo código se encuentra en el **Anexo A**. Este programa consiste en la inclusión de bibliotecas como *Binascii, Time, Socket* y *Serial,* y la autenticación de las credenciales de la Omega 2+ como un dispositivo creado en Cayenne, del cual se hablará con más detalle en el Capítulo 4.

```
#! /usr/bin/python

import binascii # Biblioteca para convertir entre binario y hexadecimal.
import time
import socket
import serial
import cayenne.client
```

[34] ONION OMEGA2. Op.cit.

```
from OmegaExpansion import oledExp

##=================================================================##

# Cayenne authentication info. This should be obtained from the Cayenne Dashboard.
MQTT_USERNAME  = "0c293a70-3b15-11e7-8d6e-f398b869a12a"
MQTT_PASSWORD  = "00923c17015647f315d5c73febc9431e4ee31076"
MQTT_CLIENT_ID = "548ff4a0-a4bf-11e9-94e9-493d67fd755e"

client = cayenne.client.CayenneMQTTClient()
client.on_message = on_message
client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID)
```

Posteriormente se definen las funciones de consulta de datos del medidor a través de Modbus, se separan y organizan los datos, recibidos previamente en una misma trama, de acuerdo a las variables medidas por el sensor, con las siguientes líneas de código (basado en un ejemplo publicado por Digi International)[35]:

```
INITIAL_MODBUS = 0xFFFF
INITIAL_DF1 = 0x0000

table = (
0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
```

[35] Digi International Inc. *Python CRC16 Modbus DF1*. 2018. [En línea] Disponible en: http://cms.digi.com/resources/documentation/digidocs/90001537/references/r_python_crc16_modbus.htm

```
0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040 )


def calcCRCByte( ch, crc):
    if type(ch) == type("c"):
        by = ord( ch)
    else:
        by = ch
    crc = (crc >> 8) ^ table[(crc ^ by) & 0xFF]
    return (crc & 0xFFFF)

def calcCRCString( sthx, crc):

    st = binascii.a2b_hex(sthx)
    for ch in st:
        crc = (crc >> 8) ^ table[(crc ^ ord(ch)) & 0xFF]
    return crc

def ModbusS(ser,data):
        ser.flush()
        ser.flushOutput()
        ser.write(data)
```

Seguidamente, se leen los parámetros en el medidor, según las direcciones donde se encuentran los registros del mismo y se definen las funciones para escribir los datos y actualizarlos en Cayenne:

```
def leer_consumo(ser,medidor):
    Direccion = medidor
    if Direccion<=15:
            Comando = str(hex(Direccion))+"0300000013"
            Comando=Comando.replace("x","")
    else:
            Comando = str(hex(Direccion))+"0300000013"
            Comando=Comando.replace("x","")
            Comando=Comando[1:len(Comando)]

    CRC = calcCRCString( Comando, 0xFFFF)
    # Lectura de parametros
    CRCL=chr(CRC&0xFF)
    CRCH=chr(CRC>>8)
    Comando_bin= chr(Direccion)+"\x03\x00\x00\x00\x13"+CRCL+CRCH
```

```
            ModbusS(ser,Comando_bin)

        def
        ver_lectura(Direccion,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele
        ):
            client.luxWrite(1, str(float(Activa)))
            client.luxWrite(2, str(float(Reactiva)))
            client.luxWrite(3, str(float(AC_IN)))
            client.luxWrite(4, str(float(RC_IN)))
            client.luxWrite(5, str(float(Voltaje)))
            client.luxWrite(6, str(float(Corriente)))
            client.luxWrite(7, str(float(Frecuencia)))
            client.luxWrite(8, str(float(FP)))

        def
        actualizar_lectura(Direccion,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuenci
        a,Rele):
            client.luxWrite(1, str(float(Activa)))
            client.luxWrite(2, str(float(Reactiva)))
            client.luxWrite(3, str(float(AC_IN)))
            client.luxWrite(4, str(float(RC_IN)))
            client.luxWrite(5, str(float(Voltaje)))
            client.luxWrite(6, str(float(Corriente)))
            client.luxWrite(7, str(float(Frecuencia)))
            client.luxWrite(8, str(float(FP)))
```

De manera adicional, se define una función para mostrar los datos en la *OLED* conectada al *Expansion Dock* teniendo en cuenta que la misma solo cuenta con 8 filas de 21 caracteres:

```
def ver_lectura(Direccion,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele):
    oledExp.driverInit()
        oledExp.setCursor(3,0)
        oledExp.write(" =======Welcome======")
        time.sleep(2)
        oledExp.clear()
        oledExp.setCursor(0,0)
        oledExp.write("Power [kWh]   = " + str(float(Activa)))
        oledExp.setCursor(1,0)
        oledExp.write("Power [kVarh] = " + str(float(Reactiva)))
    oledExp.setCursor(2,0)
        oledExp.write("Power [kW]    = " + str(float(AC_IN)))
    oledExp.setCursor(3,0)
        oledExp.write("Power [kVar]  = " + str(float(RC_IN)))
    oledExp.setCursor(4,0)
        oledExp.write("Voltage [V]   = " + str(float(Voltaje)))
    oledExp.setCursor(5,0)
        oledExp.write("Current [A]   = " + str(float(Corriente)))
    oledExp.setCursor(6,0)
        oledExp.write("Frequency     = " + str(float(Frecuencia)))
    oledExp.setCursor(7,0)
```

```
oledExp.write("FP          = " + str(float(FP)))
```

Por último, el programa llama todas estas funciones con el fin de leer la información por el puerto serial de la tarjeta, distribuirla y almacenarla en las diferentes variables por separado y enviarla a Cayenne, donde se almacenará en la base de datos de la plataforma para ser visualizada posteriormente, como puede observarse en las siguientes líneas de código:

```
iser=1
acc = 0

while (iser<=3):
   try:
      puerto='/dev/ttyUSB0'
      ser = serial.Serial(
      port=puerto,
      baudrate=9600,
      timeout=5,
      parity=serial.PARITY_NONE,
      stopbits=serial.STOPBITS_ONE,
      bytesize=serial.EIGHTBITS)
      if (not(ser.isOpen())):
         ser.open()
      ser.flushInput()
      ser.flushOutput()
      acc = 1
         iser = 4
   except Exception,e:
      iser = iser + 1

Crc_tipo=INITIAL_MODBUS

if acc == 1:
   ip=4
   Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele =
leer_consumo(ser,ip)
   ver_lectura(ip,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele)
   time.sleep(1)
   client.loop()
   while True:
         client.loop()
         Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele =
leer_consumo(ser,ip)
         actualizar_lectura(ip,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuen
cia,Rele)
         time.sleep(3)
   ser.close()
else:
   print("No se ha podido abrir el puerto serial, intente de nuevo.")

exit()
```

En esta parte del código, cabe resaltar que el medidor usado era parte de un lote de 6 fabricados, identificados con los números del 0 al 5, por lo cual este número se debe tener en cuenta en la línea de comando: ip=4 , siendo 4 el número del medidor usado para las pruebas de este proyecto.

El programa se continúa ejecutando y actualiza la información cada 3 segundos. El código completo de este programa se muestra en el **Anexo A**.

3.3 TRANSMISIÓN Y VISUALIZACIÓN DE LA INFORMACIÓN

Para la visualización de los datos medidos se consideró utilizar herramientas web, basadas en el modelo de PaaS (*Platform as a Service)*[36] con el fin de eliminar la dependencia del usuario final hacia elementos de infraestructura o a la instalación de software adicional, en cada dispositivo que quiera visualizar. La portabilidad de estos servicios web permite que el usuario final disponga de la información en cualquier dispositivo (computadora, *tablet,* teléfono móvil, entre otros). Adicionalmente, estas plataformas proveen de un entorno de desarrollo amigable, basado en el principio del uso intuitivo. Dos de estas plataformas, con mayor relevancia en el ámbito educativo, son **Cayenne**[37] e **Initial State**[38].

Cayenne es una plataforma que permite administrar una gran cantidad de dispostivos IoT, ya sean múltiples tarjetas de desarrollo o múltiples sensores en una misma tarjeta de desarrollo o SBC; y dispone de tres formas básicas de visualizar todo tipo de datos que el dispositivo pueda medir (aguja, numérico, o gráfico de tiempo), ya sean variables internas, como velocidad del CPU, almacenamiento

[36] ¿Qué es PaaS? Plataforma como servicio | Microsoft Azure [En línea]. <https://azure.microsoft.com/es-es/overview/what-is-paas/> [Citado el 7 de agosto de 2019].
[37] Cayenne Features – Developer. MyDevices.com [En línea]. <https://developers.mydevices.com /cayenne/features/>
[38] Initial State – Data streaming, data visualization. Initial State Technologies [En línea]. < https://www.initialstate.com/>

disponible, uso de RAM o temperatura interna; y variables externas como las adquiridas por sensores comunes en el mercado actual o sensores genéricos; inclusive permitiendo controlar actuadores que respondan a un evento disparador, configurado a alguna de las variables medidas por los sensores o simplemente generando una alerta1. Adicionalmente, Cayenne cuenta con una aplicación para teléfonos móviles, disponible en la tienda de aplicaciones de Iphone y Android, que permite acceder a las mismas funciones que ofrece cualquier navegador de un equipo de cómputo. Todas estas características, entre otras, están disponibles de manera gratuita al crear una cuenta en Cayenne[39].

Por otro lado, Initial State[40], deja un poco de lado la administración de los dispositivos y se centra en la captura y análisis estadístico y temporal de eventos, para lo cual ofrece un rango más amplio de dispositivos, pasando por las tarjetas de desarrollo, los SBC, hasta dispositivos comerciales como los *smartwatch* (pulseras que, entre otras cosas, miden variables de actividad corporal como: cantidad de pasos, ritmo cardíaco, distancia recorrida, etc.). Para el caso de Initial State, los datos son enviados a través del protocolo HTTP, que genera paquetes de mayor peso y longitud, lo que lleva a una comunicación más lenta, con mayor costo en ancho de banda y energía, respecto al protocolo MQTT, el cual encapsula la información en paquetes de menor tamaño, de menor complejidad, pero hechos a la medida de la necesidad de los dispositivos y las redes de IoT[41]. Desafortunadamente, Initial State no cuenta con una App para teléfonos móviles, y la versión gratuita de la cuenta permite únicamente, almacenar datos por un día (es decir, que se puede contar con un histórico limitado a 24 horas). Por estas dos

[39] Cayenne. How to manage a frustration of IoT devices. Network World. [En línea]. < https://www.networkworld.com/article/3093369/cayenne-how-to-manage-a-frustration-of-iot-devices.html>.
[40] Initial State, powerful IoT infrastructure data capture and analytics. Network World. [En línea]. < https://www.networkworld.com/article/3102171/initial-state-powerful-data-capture-and-analytics-for-your-iot-infrastructure.html>.
[41] MQTT vs HTTp: Which one is the best for IoT? – MQTT Buddy. Medium. [en línea]. < https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105>.

últimas razones, más que por el protocolo de encapsulamiento de datos, se seleccionó la plataforma **Cayenne**, para la visualización de los datos de este trabajo de grado.

4. RESULTADOS

La integración de los 3 módulos descritos en la metodología fue exitosa, obteniendo un dispositivo capaz de mostrar al usuario final, los siguientes parámetros: valores eficaces de tensión y corriente, potencias activa y reactiva, energías activa y reactiva y factor de potencia, tanto los valores actuales, como los históricos (gráfica respecto al tiempo). El sistema, además de medir y registrar los datos asociados al consumo de energía eléctrica, logró convertirse en un Smart Meter, ya que permite el acceso a la información medida, de manera remota; es decir, que tanto un usuario particular, que desea conocer el consumo actual o acumulado de energía eléctricade los equipos o dispositivos que tenga en funcionamiento; como una empresa prestadora del servicio público de energía eléctrica, que desea evitar costos de movilizar personal, para tomar medidas del consumo periódico o implementar la modalidad de suministro pre pagado, puede hacer uso de este sistema para satisfacer dichas necesidades.

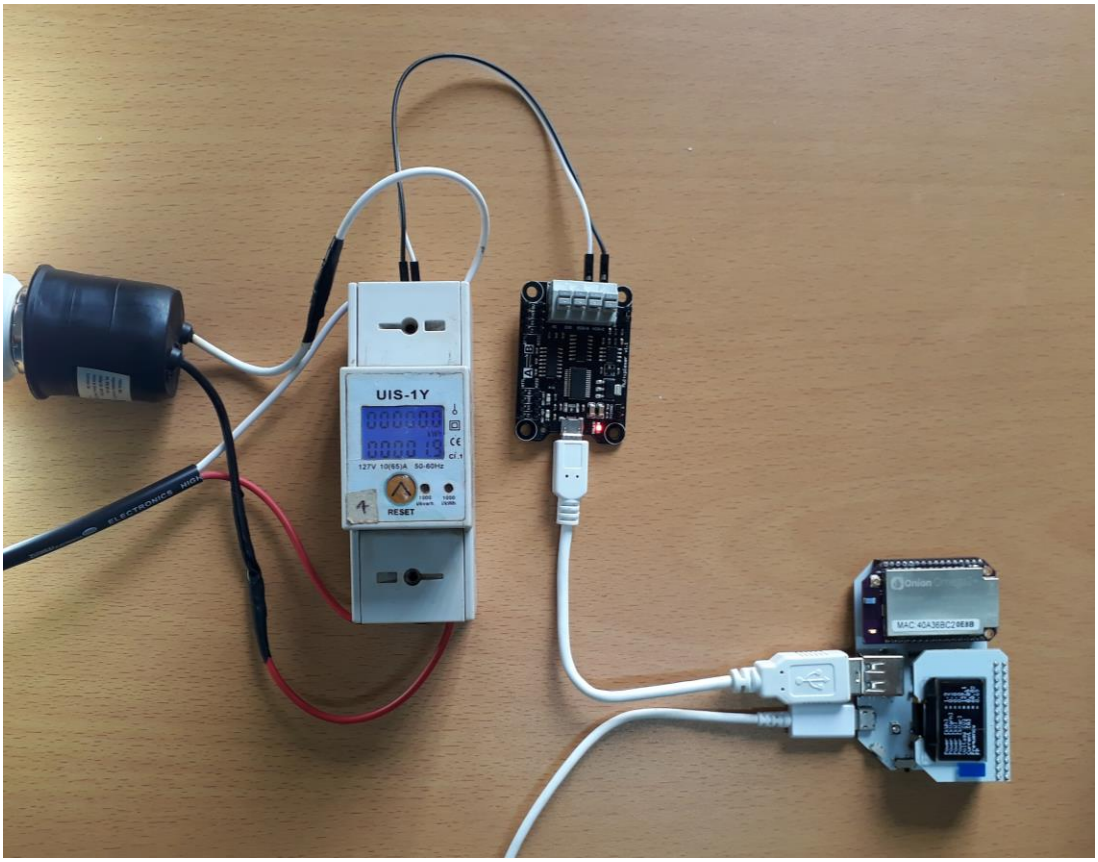
Para comprobar el correcto funcionamiento y las características descritas en el párrafo anterior, se exponen, a continuación, las evidencias recolectadas.

4.1 CONEXIÓN DE HARDWARE

Tal como se describió en la metodología, se realizó la conexión de los elementos del circuito que componen el sistema de medición, añadiendo finalmente, una carga y una alimentación de corriente alterna (proveniente del suministro doméstico) conectadas al medidor monofásico, para poder obtener medidas reales. Adicionalemente, como se observa en la Figura 8 se usó un módulo de visualización OLED para la Omega2+, como aditamento opcional para mostrar los datos en una pantalla. En la Figura 8 se observa, de izquierda a derecha, la carga (bombillo); el medidor monofásico; la tarjeta convertidora de comunicación RS485 a USB;

seguidos de la Omega2+, montada sobre el *Expansion Dock* y con el módulo de visualización instalado.

Figura 8. Fotografía del circuito de medición



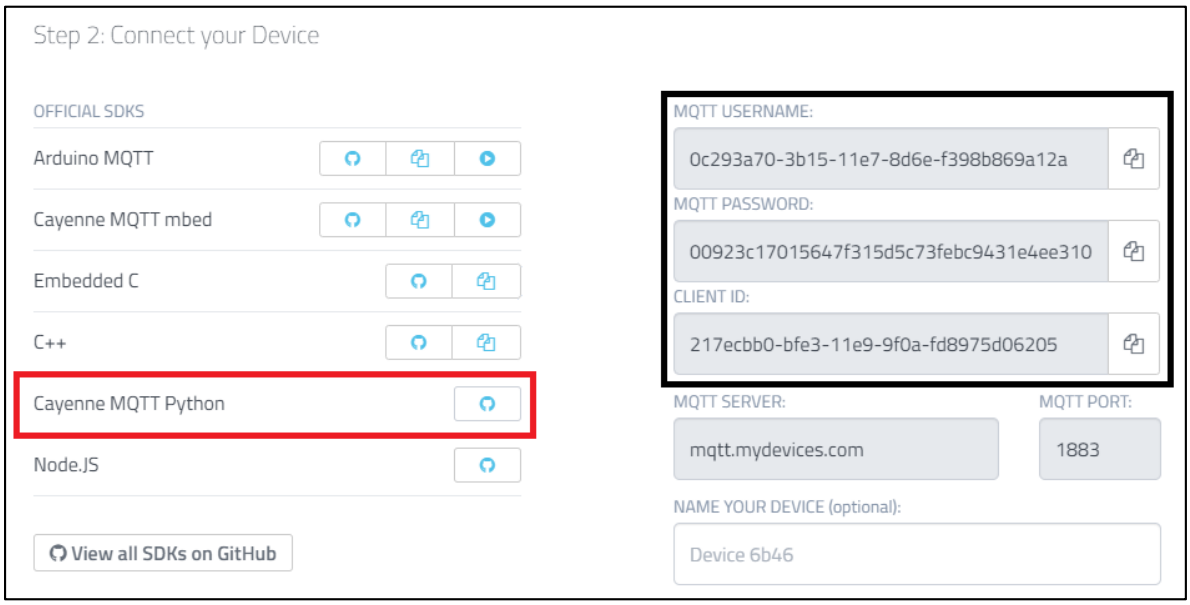
Fuente: Elaboración propia.

4.2 INTERFAZ GRÁFICA

Tras abrir una cuenta en la plataforma Cayenne, de MyDevices, se creó un nuevo dispositivo, cabe recalcar que, al momento de la realización de este proyecto, la Omega2+ no aparecía como uno de los SBC precargados en Cayenne, por lo que debe crearse a través del botón *Bring your own thing,* lo que lleva a correr uno de los programas, disponibles en diferentes lenguajes de programación (en este caso,

se utilizaron las líneas de código en Python), como se resalta en la Figura 9. En la Omega2+, debe ejecutarse el programa principal (Ver Anexo A), el cual, en primera instancia, se conecta con Cayenne a través de las credenciales MQTT de acceso, luego hace la lectura de la información adquirida por el medidor monofásico, la presenta en la pantalla OLED de la tarjeta y envía la información de cada medida (8 medidas) a Cayenne, en donde se almacenan y muestran los datos.

Figura 9. Creación del dispositivo en Cayenne

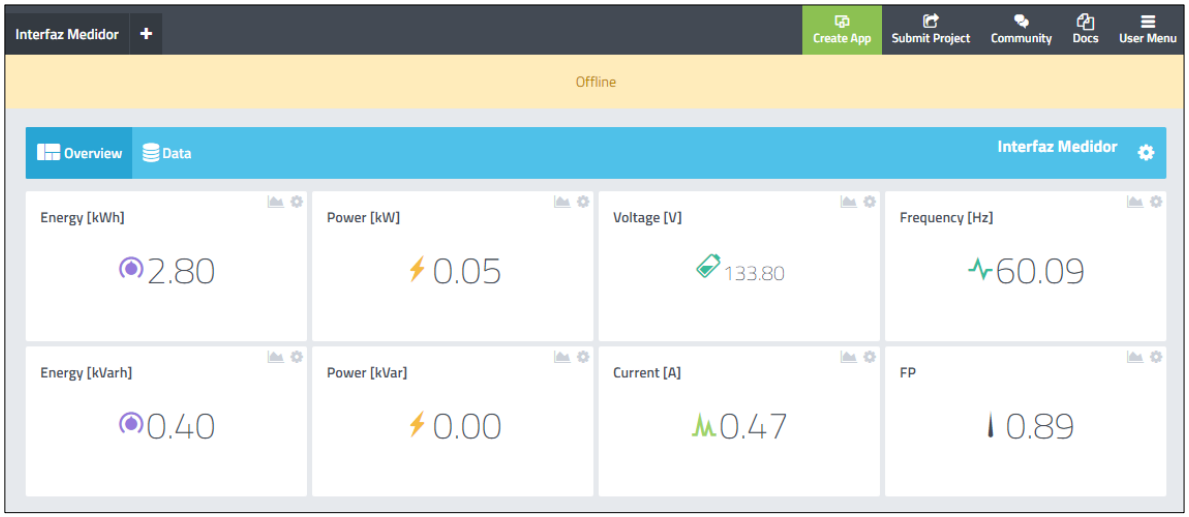


Fuente: Elaboración propia.

Seguidamente, se crean ocho *widgets*, uno para cada variable medida; estos *widgets* permiten ver los datos de tres formas en el tablero principal (ver Figura 10): aguja (emula un medidor analógico), numérico y una gráfica respecto al tiempo. La Figura 10 es un ejemplo en el que se muestra cómo cada uno de estos *widgets* en el tablero principal de Cayenne se pueden observar los valores actuales de los 8 parámetros de las variables medidas, al conectar una carga de 1,23 kW. No obstante, al acceder a cada *widget*, se muestra una línea de tiempo con múltiples

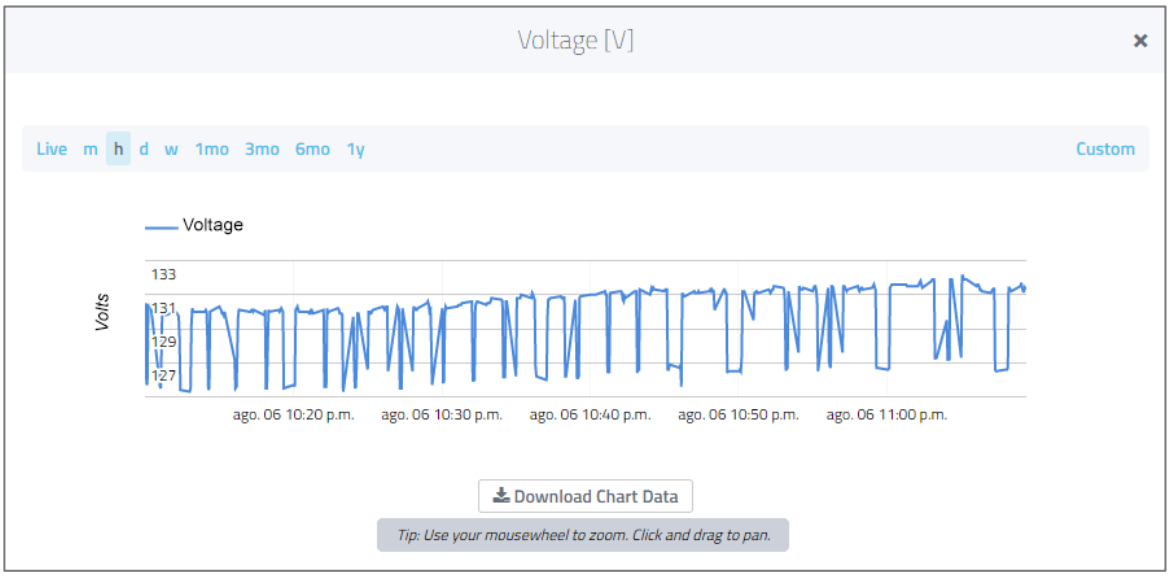
posibilidades de selección de una ventana temporal para presentar el gráfico de la variación del parámetro medido respecto al tiempo. En la Figura 11 se puede observar que en la parte superior del gráfico aparecen los vocablos *Live, m, h, d,* etc, estos hacen alusión a ventanas temporales predeterminadas para visualizar los datos, así como a la derecha se puede observar la palabra *Custom,* allí puede definirse el período de tiempo específico que se desea visualizar del histórico de los datos. Adicionalmente, la plataforma Cayenne permite descargar los datos para visualizarlos u operarlos en otros sistemas computacionales.

Figura 10. Tablero principal de Cayenne.



Fuente: Elaboración propia.

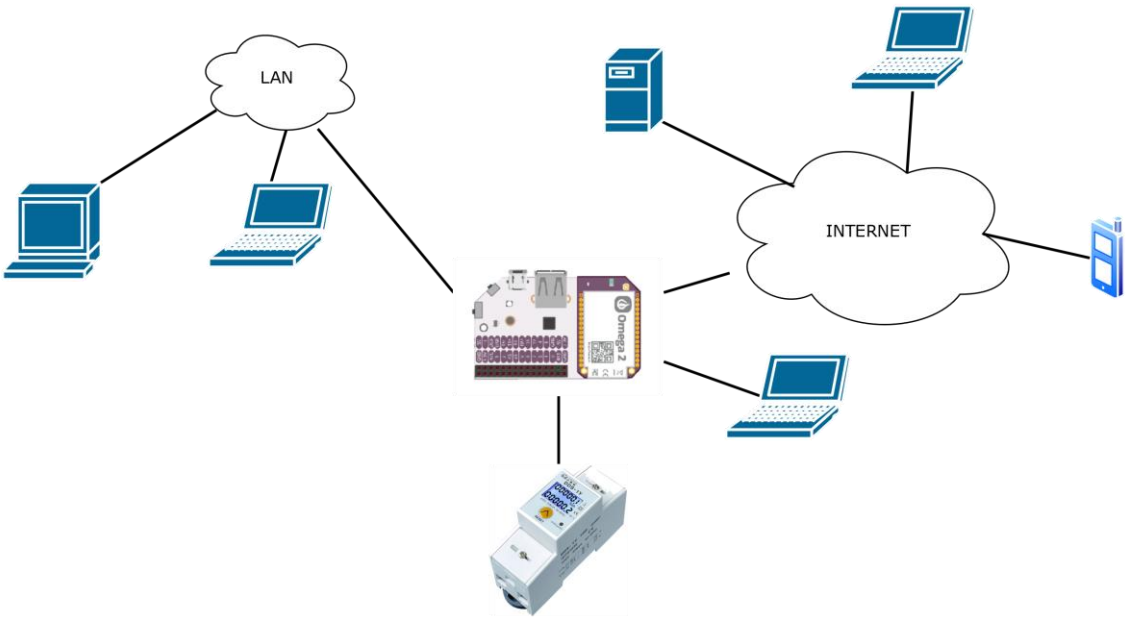
Figura 11. Ejemplo de la variación del valor eficaz de la tensión en el tiempo.



Fuente: Elaboración propia.

Al incorporar el sistema de comunicación, el medidor monofásico ha alcanzado atributos que lo convierten en un posible *Smart Meter*, gracias al concepto de Internet de las Cosas; logrando así, permitir el acceso a los parámetros del sistema eléctrico medidos por este dispositivo a servidores, repositorios, dispositivos móviles y demás terminales que cuenten con derechos de acceso a esta información. La Figura 12 muestra una potencial estructura de red del mencionado posible *Smart Meter*.

Figura 12. Potencial infraestructura de red del medidor inteligente.



Fuente: Elaboración propia.

4.3 OBSERVACIONES

Algunas observaciones relacionadas con el desarrollo de este prototipo son las siguientes:

- Para los datos de potencias acumuladas, cabe resaltar que, si el medidor monofásico se conecta a una fuente y a una carga, este va a seguir midiendo y elevando este conteo de kWh, y si el sistema no está conectado y funcionando o no cuenta con conexión a internet, pueda que existan saltos o pérdidas de continuidad en las gráficas que se visualizan de estas variables.

- Es relevante conocer que los datos medidos se alojan en la infraestructura del PaaS Cayenne, por lo que se puede acceder, desde cualquier dispositivo y lugar del mundo, a la información que está siendo enviada por el sistema de medición.

- El programa (ver Anexo A) debe almacenarse en la raíz de la memoria de la tarjeta Omega2+ o de la memoria SD de la misma para que este pueda ejecutarse de manera correcta.

- En el momento de las pruebas, también se realizó una prueba cambiando el convertidor Multiplexer v1.0 por un convertidor convencional de USB a RS485 (más económico), obteniendo exactamente los mismos resultados.

- Cada medición que se envía a la plataforma Cayenne cuenta con un número (etiqueta) asociada a la misma, de manera que puede ser individualizada la forma de mostrar cada variable.

- El costo adicional que se requiere invertir, para convertir un medidor monofásico digital en un potencial Smart Meter, el cual incluye un (1) convertidor de RS485 a USB, un (1) *Expansion Dock* para la tarjeta Omega 2+ y una (1) batería o dispositivo de alimentación es de aproximadamente 20 USD.

5. CONCLUSIONES

Al desarrollar el presente trabajo de grado, se pudieron obtener las siguientes conclusiones:

- Este trabajo de grado evidencia que un medidor de energía digital puede permitir que uno o más usuarios puedan monitorizar parámetros de un sistema eléctrico a través de equipos y servicios de muy bajo costo, convirtiendo un medidor monofásico digital, en un potencial Smart Meter.

- La implementación de una infraestructura de medición avanzada, cada día se hace más posible y viable, gracias a la gran cantidad de dispositivos, plataformas, servicios y entornos de desarrollo, que permiten que un sinnúmero de variables viaje a través de internet, optimizando procesos y mejorando la calidad de vida de las personas.

- Gracias al servicio de Cayenne como PaaS (Platform as a Service), los datos medidos se cargan a internet, lo que hace que no solo se pueda acceder a estos desde la red local, sino desde cualquier lugar y desde cualquier dispositivo.

- El medidor monofásico repotenciado en este trabajo de grado, puede aplicarse a la industria de servicios públicos, así como tiene aplicaciones en la domótica, ya que ofrece al usuario información que es útil para realizar una adecuada gestión de energía.

- La versatilidad de las plataformas de visualización ofrece una sencilla y rápida forma de monitorizar y controlar objetos a través de internet, de manera segura gracias a los avances en procesos de autenticación ofrecidos por MQTT o HTTPS.

- La Omega2+ demostró ser una plataforma versátil y robusta para el desarrollo de proyectos enfocados al IoT, con una gran ventaja: su bajo costo y tamaño.

- La integración de la plataforma de visualización, Cayenne, con la tarjeta de desarrollo, Omega2+, se logró exitosamente y sin presentar inconvenientes, a pesar de que Cayenne no tenía precargada la Omega2+ en la biblioteca de dispositivos. Esto permite ver la flexibilidad de este dispositivo, que puede utilizar cualquier tarjeta de desarrollo, siempre que esta cuente con conexión a internet y compilación de lenguajes de alto nivel, convencionales como Python, C++, entre otros.

- Para futuros trabajos de grado se recomienda agregar diferentes sensores y/o actuadores para activar o desactivar remotamente los suministros de energía u otros servicios públicos. Además, puede diseñarse una interfaz gráfica creando una infraestructura de red física y propia.

- En cuanto a la formación académica en Ingeniería, este trabajo de grado ha aportado, en lo personal, en acercar el conocimiento teórico aprendido acerca de la medición de la energía eléctrica en el ámbito local, al conocimiento práctico y comercial actual, lo cual genera una sensibilidad hacia el desarrollo y el estudio de alternativas que permitan hacer una gestión más eficiente de la misma, con ánimo de cuidar el medio ambiente y la economía de los usuarios.

BIBLIOGRAFÍA

ALBORNOZ, Claudia; BERÓN, Mario y MONTEJANO,Germán. Interfaz Gráfica de Usuario: el Usuario como Protagonista del Diseño. En: Departamento de Informática, Universidad Nacional de San Luis. 2014.

BANCO MUNDIAL. Acceso a la electricidad (% de población) [en línea]. < https://datos.bancomundial.org/indicador/EG.ELC.ACCS.ZS?end=2017&start=1990&view=chart> [citado el 15 de julio de 2019].

BARRIO, Moisés. Internet de las cosas. Madrid: Editorial Reus S.A, 2018. 123p. ISBN: 978-84-290-2038-0.

Cayenne Features – Developer. MyDevices.com [En línea]. <https://developers.mydevices.com/cayenne/features/>

Cayenne. How to manage a frustration of IoT devices. Network World. [En línea]. < https://www.networkworld.com/article/3093369/cayenne-how-to-manage-a-frustration-of-iot-devices.html>.

COLOMBIA. CONGRESO DE LA REPÚBLICA. Ley 697 (3, octubre, 2001). Mediante la cual se fomenta el uso racional y eficiente de la energía, se promueve la utilización de energías alternativas y se dictan otras disposiciones. Bogotá D.C.

COLOMBIA. MINISTERIO DE MINAS Y ENERGÍA. Resolución 40072 (28, enero, 2018). Por el cual se establecen los mecanismos para implementar la Infraestructura de Medición Avanzada en el servicio público de energía eléctrica. Bogotá D.C.

CORONEL, Marco. Estudio para la implementación del sistema de infraestructura de medición avanzada (AMI) en la Empresa Eléctrica Regional Centro Sur C.A. Cuenca, Ecuador, 2011, 270p. Trabajo de Grado (Ingeniero Eléctrico). Universidad Politécnica Salesiana. Facultad de Ciencias Eléctricas.

EPRI. Advanced Metering Infrastructure (AMI) [en línea]. < https://www.ferc.gov/eventcalendar/Files/20070423091846-EPRI%20-%20Advanced%20Metering.pdf> [citado el 29 de junio de 2019].

FINALTEST. ¿Qué es un Datalogger? [en línea]. <https://www.finaltest.com.mx/product-p/art-4.htm>. [citado el 29 de junio de 2019].

GARCÍA, Joaquín. ¿Qué es una placa SBC?. En: HardwareLibre.2015.

GUACANEME, Gerardo y PARDO, Didier Alexis. Diseño e implementación de un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto “Internet de las Cosas”, 2016, 124p. Trabajo de Grado (Ingeniero Electrónico). Universidad Distrital. Facultad de Ingeniería Electrónica.

GURDITA, Akshay; VOVKO, Heather y UNGRIN, Mark. A Simple and Low-Cost Monitoring System to Investigate Environmental Conditions in a Biological Research Laboratory. En:PLOSONE.11(1) (enero 2016). DOI: 10.1371.

HIGUERA, Edinson José y SALAZAR, Juan Andrés. Desarrollo de una Interfaz de Usuario para Monitorizar Remotamente el Consumo de Energía Eléctrica por Circuito Ramal en una Instalación de Tipo Residencial Bajo la Perspectiva del Hogar Inteligente, 2016, 57p. Trabajo de Grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-mecánicas.

Initial State – Data streaming, data visualization. Initial State Technologies [En línea]. < https://www.initialstate.com/>

Initial State, powerful IoT infrastructure data capture and analytics. Network World. [En línea]. < https://www.networkworld.com/article/3102171/initial-state-powerful-data-capture-and-analytics-for-your-iot-infrastructure.html>.

LIGHT PATH. Tarjetas Para Desarrollo De Hardware [en línea]. <http://www.lightpath.io/tarjetas-de-desarrollo/> [citado el 29 de junio de 2019].

MESA, Diego Fernando y ROJAS, Camilo Ricardo. Creación de un Sistema Domótico de Seguridad y Consumo Energético para Hogares: Homenode. Bogotá,Colombia, 2015, 160p. Trabajo de Grado (Ingeniero de Telecomunicaciones). Universidad Santo Tomas. Facultad de Ingeniería de Telecomunicaciones.

MORALES, Tatiana. Infraestructura de medición avanzada para microrredes eléctricas. Bogotá, Colombia, 2018, 104p. Trabajo de grado (Maestría en Ciencias de la Información y Comunicación). Universidad Distrital Francisco José de Caldas. Énfasis en Teleinformática.

MQTT vs HTTp: Which one is the best for IoT? – MQTT Buddy. Medium. [en línea]. < https://medium.com/mqtt-buddy/mqtt-vs-http-which-one-is-the-best-for-iot-c868169b3105>.

NOVAS, Despradel. Microcontroladores: Arquitectura, programación y aplicación. En: Atlantic International University. Honolulu, Hawai. 2008.

ONION OMEGA2. Onion Omega Documentation [en línea]. < https://docs.onion.io/omega2-docs/omega2p.html> [citado el 29 de junio de 2019].

ORDÓÑEZ PLATA, Gabriel, *et al*. Sistema de medición centralizada en redes de distribución de baja tensión para la reducción de pérdidas eléctricas, IX Simposio Internacional sobre Calidad de la Energía Eléctrica, 2017. [en línea]. <https://pdfs.semanticscholar.org/fd41/8887e4d6167fa961f4519f18116ea30c51cf.pdf/> [citado el 29 de junio de 2019].

PROTONEER. Raspberry Pi Zero Footprint And Dimensions [en línea]. < https://blog.protoneer.co.nz/raspberry-pi-zero-footprint-dimensions> [citado el 29 de junio de 2019].

PURA, Roy. Breve historia de la electricidad. En: Panorama, 2004.

¿Qué es PaaS? Plataforma como servicio | Microsoft Azure [En línea]. <https://azure.microsoft.com/es-es/overview/what-is-paas/> [Citado el 7 de agosto de 2019].

ROSE, Karen; ELDRIDGE, Scott y CHAPIN, Lyman. La Internet de las Cosas- Una Breve Reseña. Internet Society, 2015.

ROUSE, Margaret. Internet de las cosas (IoT). En: Search Datacenter (enero 2017).

SAMANIEGO, Diego y VELESACA, Diego. Diseño e Implementación de un Medidor de Energía Electrónico, para Vivienda, con Orientación a la Prevención de Consumo y Ahorro Energético. Cuenca, Ecuador, 2016, 122p. Trabajo de Grado (Ingeniero Eléctrico). Universidad Politécnica Salesiana. Facultad de Ciencias Eléctricas.

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO. Medición y gestón inteligente de consumo eléctrico. En: Boletín Tecnológico (junio de 2016).

TOLOZA, Railin Santiago y PIÑERES, Julián Andrés. Control remoto de encendido y apagado y monitorización de consumo de energía eléctrica de un sistema de iluminación, 2017, 84p. Trabajo de Grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad de Ingenierías Físico-mecánicas.

TORRES, Rafael. Medidores inteligentes. En: Biblioteca del Congreso Nacional de Chile. (marzo 2019).

UNIVERSIDAD DE SEVILLA. Interfaz gráfica de usuario [en línea]. < http://bibing.us.es/proyectos/abreproy/11300/fichero/PROYECTO%252FCapitulo3.pdf> [citado el 29 de junio de 2019].

VALDIOSERA, Abraham. Diseño de medidor inteligente e implementación de sistema de comunicación bidireccional. México DF, 2013, 335p. Trabajo de grado

(Maestro en Ciencias en Ingeniería Eléctrica). Instituto Politécnico Nacional. Escuela Superior de Ingeniería Eléctrica y Mecánica. Departamento de Ingeniería Eléctrica.

ANEXOS

Anexo A. Código de Programación en Python.

```
#! /usr/bin/python

import binascii # Libreria para convertir entre binario y hexadecimal.
import time
import socket
import serial
import cayenne.client
from OmegaExpansion import oledExp

##====================================================================##

# Cayenne authentication info. This should be obtained from the Cayenne Dashboard.
MQTT_USERNAME  = "0c293a70-3b15-11e7-8d6e-f398b869a12a"
MQTT_PASSWORD  = "00923c17015647f315d5c73febc9431e4ee31076"
MQTT_CLIENT_ID = "548ff4a0-a4bf-11e9-94e9-493d67fd755e"

# The callback for when a message is received from Cayenne.
def on_message(message):
  print("message received: " + str(message))
  if message.channel==9:
     if message.value=='1':
          time.sleep(0.5)
       rele_on(5)
          time.sleep(0.5)
     elif message.value=='0':
          time.sleep(0.5)
       rele_off(5)
          time.sleep(0.5)

client = cayenne.client.CayenneMQTTClient()
client.on_message = on_message
client.begin(MQTT_USERNAME, MQTT_PASSWORD, MQTT_CLIENT_ID)

##====================================================================##

# Funciones para el calculo del CRC, tomadas de:
# http://www.digi.com/wiki/developer/index.php/Python_CRC16_Modbus_DF1
INITIAL_MODBUS = 0xFFFF
INITIAL_DF1 = 0x0000

table = (
0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
```

```
0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040 )


def calcCRCByte( ch, crc):
    """Given a new Byte and previous CRC, Calc a new CRC-16"""
    if type(ch) == type("c"):
        by = ord( ch)
    else:
        by = ch
    crc = (crc >> 8) ^ table[(crc ^ by) & 0xFF]
    return (crc & 0xFFFF)

def calcCRCString( sthx, crc):
    """Given a bunary string and starting CRC, Calc a final CRC-16 """
    "Solo Sirve para cadenas de datos en formato Ascii-Hexa"
    st = binascii.a2b_hex(sthx)
    for ch in st:
        crc = (crc >> 8) ^ table[(crc ^ ord(ch)) & 0xFF]
    return crc

def ModbusS(ser,data):
            ser.flush()
            ser.flushOutput()
            ser.write(data)

def leer_consumo(ser,medidor):
            Direccion = medidor
            if Direccion<=15:
```

```
                Comando = str(hex(Direccion))+"0300000013"
                Comando=Comando.replace("x","")
        else:
                Comando = str(hex(Direccion))+"0300000013"
                Comando=Comando.replace("x","")
                Comando=Comando[1:len(Comando)]

        CRC = calcCRCString( Comando, 0xFFFF)
        # Lectura de parametros
        CRCL=chr(CRC&0xFF)
        CRCH=chr(CRC>>8)
        Comando_bin= chr(Direccion)+"\x03\x00\x00\x00\x13"+CRCL+CRCH
        ModbusS(ser,Comando_bin)
        # Obteniendo la respuesta
        try:
                # Espera 43 bytes en la trama de recepcion de datos
                respuestalonchera = ser.read(43)
##              for k in range(0,len(respuestalonchera)):
##                      print ("Respuesta "+str(k)+": "+str(ord(respuestalonchera[k])))
                gts=""

                for i in respuestalonchera:
                        gts=gts + str(hex(ord(i))) + " "
##              print(gts)
                off=0
                for dato in range(0,len(respuestalonchera)-1):
                        if (ord(respuestalonchera[dato]) == Direccion):
                                if (respuestalonchera[dato+1] == '\x03'):
                                        off=dato
                                        break
                respuestalonchera=respuestalonchera[off:len(respuestalonchera)]
                Direccion = ord(respuestalonchera[0])
                Comando = ord(respuestalonchera[1])
                Funcion = ord(respuestalonchera[2])
                Activa= ord(respuestalonchera[4])*65536 +
ord(respuestalonchera[6])*256+ord(respuestalonchera[8])
                Reactiva= ord(respuestalonchera[10])*65536 +
ord(respuestalonchera[12])*256+ord(respuestalonchera[14])
                Voltaje = ord(respuestalonchera[10+6])*256+ord(respuestalonchera[12+6])
                Corriente = ord(respuestalonchera[14+6])*256+ord(respuestalonchera[16+6])
                AC_IN = ord(respuestalonchera[18+6])*256+ord(respuestalonchera[20+6])
                RC_IN = ord(respuestalonchera[28])*256+ord(respuestalonchera[30])
                Frecuencia = ord(respuestalonchera[32])*256+ord(respuestalonchera[34])
                FP = ord(respuestalonchera[36])*256+ord(respuestalonchera[38])
                Rele1 = ord(respuestalonchera[39])
                Rele2 = ord(respuestalonchera[40])
                Rele = Rele1 + Rele2

        except Exception,e:
                Activa=-1
                Reactiva=-1
                AC_IN=-1
                RC_IN=-1
```

```
                Voltaje=-1
                Corriente=-1
                FP=-1
                Frecuencia=-1
                Rele = -1
                print("Error en la funcion! " + str(e))
        return
float(Activa)/10,float(Reactiva)/10,float(AC_IN)/100,float(RC_IN)/100,float(Voltaje)/10,float(Corriente
)/100,float(FP)/100,float(Frecuencia)/100,Rele

def ver_lectura(Direccion,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele):
    oledExp.driverInit()
        oledExp.setCursor(3,0)
        oledExp.write(" =======Welcome======")
        time.sleep(2)
        oledExp.clear()
        oledExp.setCursor(0,0)
        oledExp.write("Power [kWh]   = " + str(float(Activa)))
        oledExp.setCursor(1,0)
        oledExp.write("Power [kVarh] = " + str(float(Reactiva)))
    oledExp.setCursor(2,0)
        oledExp.write("Power [kW]    = " + str(float(AC_IN)))
    oledExp.setCursor(3,0)
        oledExp.write("Power [kVar]  = " + str(float(RC_IN)))
    oledExp.setCursor(4,0)
        oledExp.write("Voltage [V]   = " + str(float(Voltaje)))
    oledExp.setCursor(5,0)
        oledExp.write("Current [A]   = " + str(float(Corriente)))
    oledExp.setCursor(6,0)
        oledExp.write("Frequency     = " + str(float(Frecuencia)))
    oledExp.setCursor(7,0)
        oledExp.write("FP            = " + str(float(FP)))
##============================Cayenne================================##
        client.luxWrite(1, str(float(Activa)))
        client.luxWrite(2, str(float(Reactiva)))
        client.luxWrite(3, str(float(AC_IN)))
        client.luxWrite(4, str(float(RC_IN)))
        client.luxWrite(5, str(float(Voltaje)))
        client.luxWrite(6, str(float(Corriente)))
        client.luxWrite(7, str(float(Frecuencia)))
        client.luxWrite(8, str(float(FP)))

def
actualizar_lectura(Direccion,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele):
        oledExp.setTextColumns()
    oledExp.setCursor(0,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
    oledExp.setCursor(0,16)
    oledExp.write(str(float(Activa)))
    oledExp.setTextColumns()
    oledExp.setCursor(1,16)
    oledExp.write("     ")
```

```
        oledExp.setTextColumns()
        oledExp.setCursor(1,16)
        oledExp.write(str(float(Reactiva)))
        oledExp.setTextColumns()
        oledExp.setCursor(2,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
        oledExp.setCursor(2,16)
        oledExp.write(str(float(AC_IN)))
        oledExp.setTextColumns()
        oledExp.setCursor(3,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
        oledExp.setCursor(3,16)
        oledExp.write(str(float(RC_IN)))
        oledExp.setTextColumns()
        oledExp.setCursor(4,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
        oledExp.setCursor(4,16)
        oledExp.write(str(float(Voltaje)))
        oledExp.setTextColumns()
        oledExp.setCursor(5,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
        oledExp.setCursor(5,16)
        oledExp.write(str(float(Corriente)))
        oledExp.setTextColumns()
        oledExp.setCursor(6,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
        oledExp.setCursor(6,16)
        oledExp.write(str(float(Frecuencia)))
        oledExp.setTextColumns()
        oledExp.setCursor(7,16)
        oledExp.write("     ")
        oledExp.setTextColumns()
        oledExp.setCursor(7,16)
        oledExp.write(str(float(FP)))
##============================Cayenne================================##
        client.luxWrite(1, str(float(Activa)))
        client.luxWrite(2, str(float(Reactiva)))
        client.luxWrite(3, str(float(AC_IN)))
        client.luxWrite(4, str(float(RC_IN)))
        client.luxWrite(5, str(float(Voltaje)))
        client.luxWrite(6, str(float(Corriente)))
        client.luxWrite(7, str(float(Frecuencia)))
        client.luxWrite(8, str(float(FP)))


def Hexastring(Valor):
    if (Valor==0):
        Salida = "\x00"
```

```
    elif (Valor==1):
        Salida = "\x01"
    elif (Valor==2):
        Salida = "\x02"
    elif (Valor==3):
        Salida = "\x03"
    elif (Valor==4):
        Salida = "\x04"
    elif (Valor==5):
        Salida = "\x05"
    elif (Valor==6):
        Salida = "\x06"
    elif (Valor==7):
        Salida = "\x07"
    elif (Valor==8):
        Salida = "\x08"
    elif (Valor==9):
        Salida = "\x09"
    else:
        disp(" -- No se encuentra el valor")
    return Salida

def rele_on(Direccion):
    if Direccion<=15:
        Comando = str(hex(Direccion))+"06000C0001"
        Comando=Comando.replace("x","")
    else:
        Comando = str(hex(Direccion))+"06000C0001"
        Comando=Comando.replace("x","")
        Comando=Comando[1:len(Comando)]

    CRC = calcCRCString( Comando, Crc_tipo)
    Comando = str(Direccion) +" 06 00 0C 00 01" + str(hex(CRC>>8)) + str(hex(CRC & 0xFF))
    print("Direccion          = " + str(Direccion))
    print ("--- Encendiendo Rele")
    # Lectura de parametros
    #Comando_bin="\x01\x06\x00\x0c\x00\x00\x49\xC9"
    CRCL=chr(CRC&0xFF)
    CRCH=chr(CRC>>8)
    Comando_bin=chr(Direccion)+"\x06\x00\x0c\x00\x01"+CRCL+CRCH
    ser.write(Comando_bin)

def rele_off(Direccion):
    if Direccion<=15:
        Comando = str(hex(Direccion))+"06000C0000"
        Comando=Comando.replace("x","")
    else:
        Comando = str(hex(Direccion))+"06000C0000"
        Comando=Comando.replace("x","")
        Comando=Comando[1:len(Comando)]

    CRC = calcCRCString( Comando, Crc_tipo)
    Comando = str(Direccion) +" 06 00 0C 00 00" + str(hex(CRC>>8)) + str(hex(CRC & 0xFF))
```

```
    print("Direccion          = " + str(Direccion))
    print ("--- Apagando Rele")
    # Lectura de parametros
    #Comando_bin="\x01\x06\x00\x0c\x00\x00\x49\xC9"
    CRCL=chr(CRC&0xFF)
    CRCH=chr(CRC>>8)
    Comando_bin=chr(Direccion)+"\x06\x00\x0c\x00\x00"+CRCL+CRCH
    ser.write(Comando_bin)

iser=1
acc = 0

while (iser<=3):
    try:
        puerto='/dev/ttyUSB0'
        ser = serial.Serial(
        port=puerto,
        baudrate=9600,
        timeout=5,
        parity=serial.PARITY_NONE,
        stopbits=serial.STOPBITS_ONE,
        bytesize=serial.EIGHTBITS)
        if (not(ser.isOpen())):
            ser.open()
        ser.flushInput()
        ser.flushOutput()
        acc = 1
            iser = 4
    except Exception,e:
        iser = iser + 1

# CRC Modbus:
Crc_tipo=INITIAL_MODBUS

# Si no se puede abrir el puerto no inicia el programa
if acc == 1:
    ip=4
    Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele = leer_consumo(ser,ip)
    ver_lectura(ip,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele)
    time.sleep(1)
    client.loop()
    while True:
            client.loop()
            Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele =
leer_consumo(ser,ip)
            actualizar_lectura(ip,Activa,Reactiva,AC_IN,RC_IN,Voltaje,Corriente,FP,Frecuencia,Rele)
            time.sleep(3)
    ser.close()
else:
    print("No se ha podido abrir el puerto serial, intente de nuevo.")

# Cierra el modulo serial
exit()
```