

Sistema Inteligente para para la gestión automática de un generador eléctrico basado en la arquitectura del IoT

Smart System for Monitoring Electric Generators Based on Open-Source Platforms

DOI: <https://doi.org/10.17981/cesta.01.01.2020.01>

Artículo de investigación científica. Fecha de recepción: 20/10/2020 Fecha de aceptación: 23/10/2020

Diana Suárez 

Corporación Universitaria Americana. Barranquilla (Colombia)
dsuarez@coruniamericana.edu.co

Jhon Solano

Corporación Universitaria Americana. Barranquilla (Colombia)
solanoyohn@coruniamericana.edu.co

Roberto Boris Martínez

Universidad Católica San Pablo. Arequipa (Perú)
rbmartinez@ucsp.edu.pe

Manuel Angarita

Corporación Universitaria Americana. Barranquilla (Colombia)
angaritamanuel@coruniamericana.edu.co

Carlos Henríquez Miranda 

Universidad del Magdalena. Santamarta (Colombia)
chenriquezm@unimagdalena.edu.co.edu.co

Dixon Salcedo 

Universidad de la Costa. Barranquilla (Colombia)
dsalcedo2@cuc.edu.co

Para citar este artículo:

D. Suárez, J. Solano, M. Martínez, R., Angarita, C. Henríquez & D. Salcedo, "Sistema Inteligente para para la gestión automática de un generador eléctrico basado en la arquitectura del IoT", *J. Comput. Electron. Sci.: Theory Appl.*, vol. 1, no. 1, pp. 1–10, 2020. <https://doi.org/10.17981/cesta.01.01.2020.01>

Resumen— Este artículo presenta el diseño y prototipado de un sistema inteligente, el cual sirve para la gestión automática de un generador eléctrico, basado en la arquitectura del IoT, a través del protocolo de comunicación MQTT. El prototipo, permite automatizar diversas funciones de un generador eléctrico ante la interrupción del fluido eléctrico; tales como: encendido y apagado de forma automática, teniendo en cuenta las condiciones eléctricas. También permite controlar variables como el nivel de combustible, temperatura, horas de uso del equipo; además de facilitar cambios de aceite, y mantenimientos preventivos. Adicionalmente, cuenta con un gestor automático de carga de potencia, que evita que el generador inicie con una potencia máxima desde el arranque; logrando de esta forma ampliar el margen de vida útil de los circuitos electrónico de potencia. Finalmente, para controlar de manera remota las funciones mencionadas, se presenta una aplicación móvil para que el usuario final pueda monitorear en tiempo real el funcionamiento del generador, mediante la implementación del protocolo de comunicación de Message Queue Telemetry Transport (MQTT).

Palabras clave— Arduino; automatización; IoT; generador eléctrico; MQTT

Abstract— The present work presents the design and prototyping of an intelligent system, which serves for the automatic management of an electric generator, based on the IoT architecture, through the MQTT communication protocol. The prototype, allows automating several functions of an electric generator in case of electric fluid interruption; such as: turning on and off automatically, taking into account the electric conditions. It also allows to control variables such as fuel level, temperature, hours of use of the equipment; besides facilitating oil changes, and preventive maintenance. In addition, it has an automatic power load manager, which prevents the generator from starting at maximum power from the start; thus extending the life span of the electronic power circuits. Finally, to control remotely the mentioned functions, a mobile application is presented so that the end user can monitor in real time the operation of the generator, through the implementation of the Message Queue Telemetry Transport (MQTT) communication protocol.

Keywords— Arduino; automation; IoT; electric generator; MQTT



I. INTRODUCCIÓN

La energía eléctrica es un recurso indispensable, necesario en todos los sectores ya sea industrial, comercial, residencial u hospitalario, de ahí que muchas empresas dispongan de un mecanismo alternativo para la generación de energía eléctrica en caso de que el suministro principal falle por alguna razón [1]. En Colombia, más específicamente en la Región Caribe, se presentan fallas e interrupciones en el servicio de distribución eléctrica, que afecta significativamente a grandes, y pequeños comerciantes ubicados en las diferentes ciudades de la región. Debido a que el servicio es parte vital para el funcionamiento de los comercios mencionados, más específicamente a los que almacenan alimentos perecederos; para los que un corte por tiempo prolongado de energía eléctrica representa pérdidas económicas [2]; debido a la falta de refrigeración descomponen los alimentos. Por lo tanto, para minimizar las pérdidas, comerciantes han adquirido estabilizadores de voltajes, que regulen la tensión en el servicio eléctrico.

Por otro lado, se puede definir un generador eléctrico como un dispositivo o máquina capaz de transformar la energía mecánica en energía eléctrica, sus aplicaciones son muy amplias y diversas, pero la mayoría de los generadores eléctricos comerciales de baja y media potencia no disponen de un módulo de automatización como parte de sus funciones básicas, por tal motivo es necesario realizar acciones manuales para encender y/o apagar la máquina cuando las circunstancias lo requieran [3]. Así mismo, muchos de los generadores eléctricos limitan sus indicadores visuales a unos pocos en su panel de control principal como el nivel de voltaje y el indicador de encendido; en ese sentido, el usuario desconoce los valores de otras variables e indicadores importantes como la cantidad de horas de uso, nivel de reserva de gasolina, corriente y potencia de salida, temperatura y frecuencia entre otros.

En consecuencia, un sistema adecuado de automatización para un generador eléctrico, no solo debe considerar aspectos internos al generador, sino que debe también analizar variables externas a él como es el caso del detector de intermitencia eléctrica en la red de energía principal [4], es decir el suministro de energía de la red eléctrica principal, en ese sentido se determina una intermitencia al instante en que falla la energía eléctrica en un intervalo de 1 a 10 segundos; esta variable servirá para evitar el encendido innecesario del generador eléctrico por parte del sistema de automatización.

Los generadores cuentan con dos detectores: uno de voltaje y otro de frecuencia, encargados de analizar la red eléctrica principal solo en el instante en que se haya reestablecido el fluido eléctrico tras un fallo; si los niveles no son los adecuados, entonces el Sistema de Automatización (SA), no emitirá la orden de apagado y el generador seguirá activo hasta que red eléctrica principal se encuentre normalizada.

Otro factor para tener en cuenta es la forma en que los generadores eléctricos entregan su potencia, por su naturaleza estos lo hacen de una forma incremental y directamente conectado a la carga, hasta que el generador se estabilice por completo, desafortunadamente la mayoría de estos equipos no cuenta con un mecanismo electrónico o mecánico que entreguen su potencia solo en el momento en el que generador ya se encuentre estable.

Teniendo en cuenta la problemática anterior, este desarrollo tecnológico tiene como propósito brindar a los pequeños empresarios las herramientas necesarias que le permitan tener una continuidad del servicio eléctrico, cuando se presente interrupciones; sin que afecten sus actividades comerciales cotidianas. Lo anterior, se logra implementando tecnologías relacionadas con el IoT en tiempo real; de tal manera, que puedan controlar y monitorear de manera autónoma el generador eléctrico, utilizando un sistema de bajo costo, logrando disminuir las pérdidas económicas que pueden ocasionarse por no contar con el fluido eléctrico.

Finalmente, este artículo presenta el diseño y prototipado de un sistema inteligente, que sirve para la gestión automática de un generador eléctrico, basado en la arquitectura del IoT, a través del protocolo de comunicación MQTT. Adicionalmente, introduce una aplicación móvil para que el usuario final pueda monitorear, y administrar de manera remota en tiempo real, controlar variables del sistema, tales como, nivel de combustible, temperatura, horas de uso del equipo; que permite generar alertas preventivas.

II. METODOLOGÍA

El método de investigación corresponde al tipo investigación aplicada, orienta a resolver problemas en un contexto específico, en este caso el sector energético, de tal manera que se pudieron aplicar los conocimientos de una forma práctica para lograr la creación del prototipo que controlará el generador eléctrico, para lo cual se definieron las siguientes fases:

Fase 1: Se identifica la problemática y se hace un diagnóstico de la situación del sector, se consultan diferentes fuentes de información.

Fase 2: Se definieron los requerimientos del negocio, del usuario, de igual manera se diseña la arquitectura del sistema y finalmente se lleva a cabo su implementación integrando distintos dispositivos y protocolos.

Fase 3: Se llevaron a cabo las pruebas con datos reales, para ello, se escogió uno de los negocios afectado por las fallas en el fluido eléctrico, se procedió con la instalación, se validaron todas las funciones del prototipo teniendo en cuenta todos los posibles escenarios. Los resultados obtenidos permitieron realizar ajustes para mejorar el desempeño del prototipo.

A continuación, se muestra el desarrollo de cada una de las fases anteriores:

A. Diseño general del sistema

Debido a que, el trabajo se centra en la implementación de tecnologías relacionadas con el IoT integrando placas de Arduino Uno, sensores de temperatura y humedad y protocolos de comunicación MQTT, tal como se observa en la Fig. 1. el IoT ha tomado mucha fuerza en los últimos años, el concepto radica en que los objetos estén interconectados a una red que puedan hacer envío de datos, ser monitoreados y controlados [5]. De igual manera [6] afirma que el mundo actual gira entorno a la evolución de las tecnologías y sus diferentes aplicaciones; con la llegada del IoT, hoy en día se puede contar con electrodomésticos que se comunican con el usuario a través de su móvil, relojes inteligentes que envían información del estado de salud del usuario a sus médicos, en tiempo real [7].

En la Fig. 1 se asignan los valores de 1 para verificar la presencia de energía eléctrica y cero (0) la ausencia de energía eléctrica producida por el generador, como un indicador de encendido o apagado; de igual manera el uso del sensor 2 se para determinar el estado alto (presencia de energía eléctrica red principal) y el sensor 5 para estado bajo (Generador eléctrico inactivo).

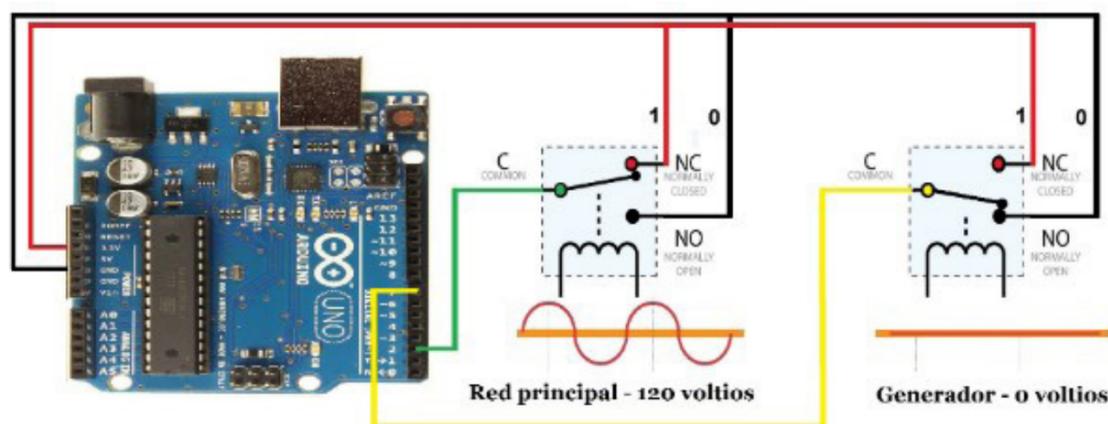


Fig. 1. Circuito esquemático sensor eléctrico de la red principal.

Fuente: Autores.

B. Implementación

Una de las características que les brinda el IoT a los usuarios es la comodidad, al paso de los años y el avance de las herramientas de hardware, se pueden desarrollar objetos más autónomos, es allí cuando los dispositivos como Arduino, Raspberry cobran importancia, porque ambas plataformas permiten crear la interactividad de todos estos sistemas, convirtiendo al Arduino en el cerebro del sistema (Fig. 2).

En el Sector energético encontramos una serie de aplicaciones donde el IoT está ayudando a impulsar su transformación, desde control de iluminación y climatización hasta medidores eléctricos inteligentes, trayendo consigo ahorros significativos en el costo de uso de la energía eléctrica [8], mejoramiento de las operaciones o actividades empresariales [9], todo eso debido a la implementación de sensores en equipos de generación, transmisión y distribución [10], que permiten monitorizar y control de manera remota y permanente el consumo de energía, dando paso a lo que hoy se conoce como Smart Energy [11] [12].

Por último se implementan protocolos de comunicación como el MQTT (Message Queuing Telemetry Transport), desarrollado por IBM en la década de los 90, funciona mediante mensajería asíncrona entre las partes (Arduino y sensores), este protocolo ofrece grandes ventajas con respecto a otros, debido a que es de código abierto, su curva de aprendizaje es empinada, su implantación es fácil y rápida; permite el uso eficiente del ancho de banda en redes inalámbricas, consumo mínimo de energía, no requiere de mucho procesador y memoria, su tiempo de respuesta óptimo, es flexible y escalable en redes que no sean de confianza [13]. Una de las ventajas que tiene este protocolo con respecto a muchos otros es que está diseñado para la transición múltiple entre los clientes, esto resulta muy costoso en protocolos como el http, esta característica en especial lo hace ideal para el IoT [14].

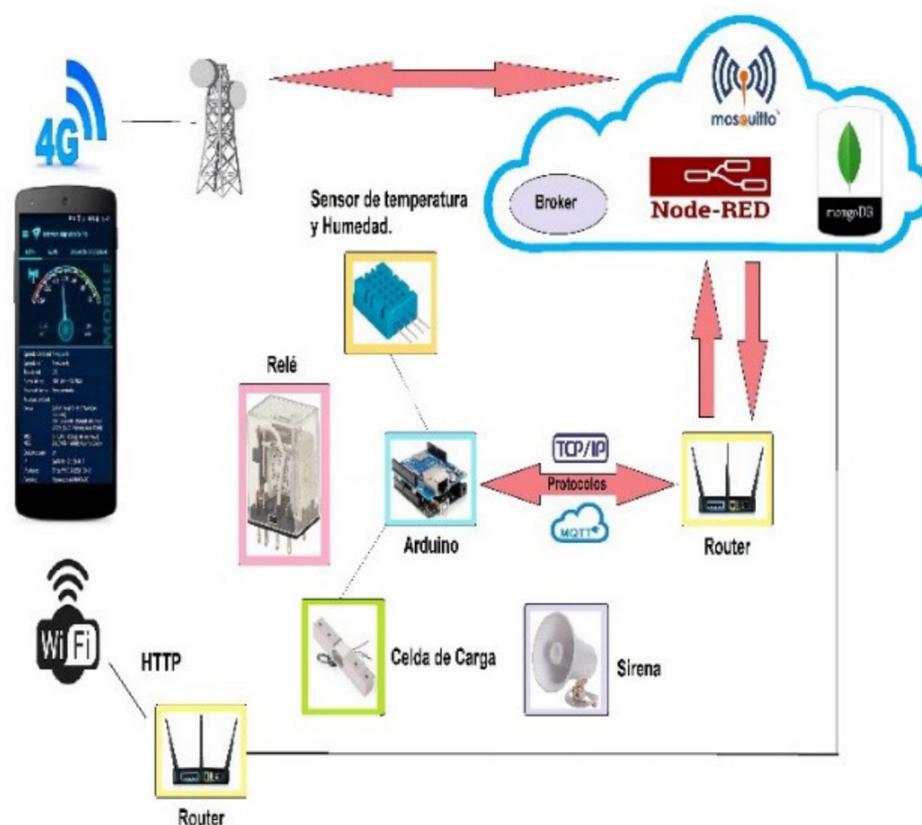


Fig. 2. Diagrama de componentes IoT.
Fuente: Autores.

C. Funcionalidades del prototipo

El sistema consta de sensores de voltaje que constantemente están midiendo el fluido eléctrico tanto del proveedor de energía como del generador; una vez se detecta que se interrumpió el fluido eléctrico, el sistema verifica dentro de un lapso de tiempo de 10 segundos que no haya sido una intermitencia, si el fallo persiste dará la orden de encendido al generador que se encuentra aislado de los circuitos de carga eléctrica, al encender el sistema se encarga de conectar secuencialmente las cargas, esto con la finalidad de no sobrecargar el generador.

Luego, una vez reestablecido el fluido eléctrico principal (del proveedor), el sistema realiza el procedimiento inverso, el cual consiste en verificar que no exista intermitencia del flujo de energía por 40 segundos, terminado este tiempo y no haber cambios en el estado de la medición procede a apagar el generador.

El sistema también cuenta con un panel de administración y control desde donde se muestra el estado de todas las mediciones en tiempo real y desde donde también puede gestionarse como trabaja el sistema (modo automático o modo manual). En la Fig. 3, se muestran las funcionalidades implementadas al generador.



Fig. 3. Funciones aplicado al generador.
Fuente: Autores.

D. Algoritmo estado energía

Estado de energía eléctrica red principal: El sistema conformado por varios sensores, cuya función es verificar el fluido eléctrico de la red primaria, para ello se utilizó un dispositivo mecánico llamado relay, cuando el voltaje alterno AC circula por la bobina del relay, se produce un efecto de electroimán, la fuerza magnética hace que la barra sea atraída hacia un extremo, realiza un cambio de estado normalmente cerrado NC al estado normalmente abierto NA, cerrando el circuito; la terminal NA está conectado a tierra, indicando un estado bajo (0) y la terminal NC está conectado a 3,3 v, indicado un estado alto (1), este estado le indica al Arduino a través del puerto 2 configurado en INPUT la presencia de energía eléctrica en la red principal; para el caso de ausencia de energía eléctrica la bobina no producirá energía magnética, por ende la barra regresara a su estado normalmente cerrado NC gracias a la ayuda de un pequeño resorte.

Estado de energía del generador eléctrico: El encendido del generador eléctrico dependerá de la función de detección de voltaje de la red principal una vez que se detecta la interrupción en el fluido eléctrico, el sistema verifica dentro de un lapso de 10 segundos que no haya sido una intermitencia, si el fallo persiste el sistema dará la orden de encendido al generador, cabe aclarar que el generador en este punto se encuentra apagado y aislado de la red de consumo (CI de carga) eléctrica.

Activar generador: Para el encendido del generador eléctrico se utilizan los puertos 3, 4 y 5 del Arduino configurado en modo salida, cada PIN se conecta a su relé correspondiente. Los pines 3 y 4 activan el modo stand-by del generador y el pin 5 es el encargado de iniciar (starte) el motor de arranque del generador.

El modo stand-by, es un paso obligatorio antes de ejecutar la acción starte, su equivalente físico es girar la llave de la posición cero (CI Cerrado) a la posición 1 (CI Abierto), la acción starte es el equivalente de girar la posición de la llave del estado 1 al estado 2 por un tiempo de 2150 milisegundo, tiempo suficiente para que el generador arranque, para luego regresar a la posición 1.

Iniciar proceso activar generador eléctrico: El encendido del generador eléctrico dependerá de la función de detección de voltaje de la red principal, una vez que se detecta la interrupción en el fluido eléctrico, el sistema verifica dentro de un lapso de 10 segundos que no haya sido una intermitencia, si el fallo persiste el sistema dará la orden de encendido al generador, cabe aclarar que el generador en este punto se encuentra apagado y aislado de la red de consumo (CI de carga) eléctrico. Para el encendido del generador eléctrico se utilizaron los puertos 3, 4 y 5 del Arduino configurado en modo salida, cada PIN se conecta a su relé correspondiente. Los pines 3 y 4 se utilizan para activar el modo stand-by del generador y el pin 5 es el encargado de iniciar (starte) del motor de arranque del generador.

Activar contactores: La función activar contactores solo se ejecutará si la función Iniciar proceso activar generador eléctrico (Fig. 4), retorna un estado alto, esto se consigue porque dentro de esta función se realiza un llamado a otra función, estado de energía del generador eléctrico.

```
int IniciarProcesoActivarGeneradorElectrico()
{
    int i = 0;
    int estado = HIGH;

    do
    {
        i++;
        ActivarGeradorElectrico();
        estado = EstadoEnergiaGeneradorElectrico();
    } while ( ( estado == LOW ) && ( i <= 2 ) );

    return estado;
}
```

Fig. 4. Algoritmo estado energía.
Fuente: Autores.

Si la función *Iniciarprocesoactivargeneradorelectrico* retorna un estado ALTO al momento de ejecutar el flujo principal, significa que el generador este encendido y produciendo energía. Por otra parte, si el estado que retorna es bajo, es porque a pesar haberse realizado varios intentos, el generador nunca se activó, en tal caso se notificara al usuario y se abortara la secuencia. La función *IniciarProcesoActivarContactores* Cumple dos objetivos específicos muy importantes:

1. Dar un tiempo de espera para que los niveles de voltaje, corriente y frecuencia se estabilicen, cabe destacar que, al momento de activar el generador, este producirá una diferencia de potencia gradual hasta alcanzar el nivel máximo, de forma similar ocurre con las demás variables. En la Fig. 5, se observa que en 2.5 segundos el generador solo ha alcanzado un diferencial de 60 voltios, por tal motivo el sistema realiza una espera de 5 segundos antes de liberar la potencia hacia la red de consumo.
2. Liberar secuencialmente los diferentes circuitos de cargas, para evitar sobrecarga de golpe en el generador, esto se logra gracias a los contactores.

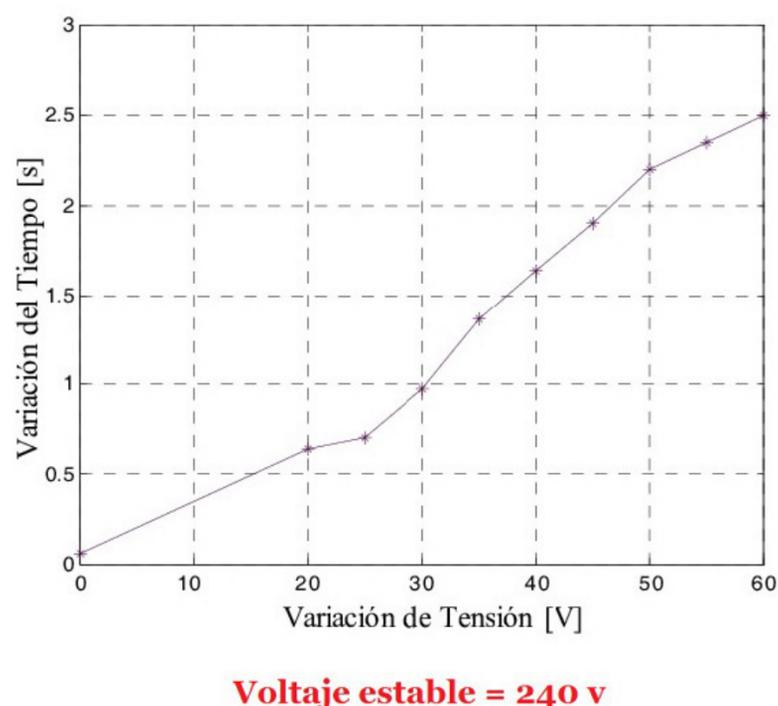


Fig. 5. Variación de voltaje vs tiempo de activación del generador.
Fuente: Autores.

En [15], se explica que “el contactor es un aparato eléctrico de mando a distancia, que puede cerrar o abrir circuitos, ya sea en vacío o en carga, convirtiéndose así en la pieza clave del automatismo en el motor eléctrico”. Su principal aplicación es la de efectuar maniobras de apertura y cierre de circuitos eléctricos relacionados con instalaciones de motores, con excepción de pequeños motores, que son accionados manualmente o por relés, el resto de los motores se accionan por contactores.

El contactor es activado a distancia, por un voltaje de 120 voltios en la bobina, pero como sabemos de antemano que el voltaje máximo que provee el Arduino en sus puertos es de 5 voltios, necesita la ayuda del relé, debido a que solo necesitan 5 voltios en la bobina para activarse.

Desactivar contactores: Esta función es la inversa de la función activar contactores, inicia desactivando el último contactor que se activó en la función anterior y termina desactivando el primer contactor que se activó, similar a una estructura LIFO. Esta función solo se ejecuta bajo las siguientes condiciones: la primera que la función *EstadoEnergíaEléctricaRedPrincipal* retorne un valor alto en el flujo principal del código, lo cual indica que la energía eléctrica de la red principal está normalizada. La segunda esperar la confirmación 40 segundos después de la función anterior, con esto se reduce la probabilidad de que no sea una intermitencia y finalmente el generador debe estar activado.

Medidor de combustible: Como su nombre lo indica, esta función es la encargada de testear el nivel de combustible almacenado en el tanque del generador eléctrico, informa en tiempo real, el porcentaje de combustible almacenado, de tal forma que:

1. Genera notificaciones de combustible bajo, en nuestro caso está configurado para los niveles 15%, 10% y 5%, esta alerta es de 2 tipo, sonora y vía web.
2. Inicia el apagado automático del generador eléctrico cuando el nivel de combustible sea inferior al 3%, de esta manera protege la tarjeta principal del generador eléctrico.
3. Evita el encendido del generador cuando el combustible sea inferior al 10%.
4. Estima el tiempo de funcionamiento en relación con combustible almacenado.

Para medir la cantidad de combustible del generador se utilizó el peso mismo del combustible a través de una celda de carga que no es más que un transductor utilizado para convertir una fuerza en una señal eléctrica mediante un dispositivo mecánico, la fuerza que se desea medir deforma el sensor y convierte el desplazamiento o deformación en señales eléctricas. La señal eléctrica de salida es de unas pocas milis voltios y debe ser ampliada mediante un amplificador antes de que pueda ser utilizada con cualquier microcontrolador que disponga de una entrada ADC. Debido a la vibración del motor se construyó un soporte que sirve de intermedio entre el tanque de combustible y la celda de carga, con esto se logró una mejor estabilidad en la medición del combustible.

- *Modo automático:* Este estado es el permite al Arduino, dar control sobre el sistema y ejecutara las funciones de encendido y apagado según las condiciones externas del fluido eléctrico.
- *Modo manual:* Este estado pone al Arduino a la espera de instrucciones por parte del usuario. Estas instrucciones son enviadas a través del panel de administración.

```
#include "HX711.h"
HX711 scale;

int MedidorGasolina()
{
  scale.power_up();
  int medicionPromedio = (int)scale.get_units(20) + 78;
  scale.power_down();
  delay(1000);
  return medicionPromedio;
}
```

Fig. 6. Algoritmo de medición de gasolina.
Fuente: Autores.

Aplicando al algoritmo de medición de gasolina (Fig. 6), se ignoró el peso del tanque de combustible, definiendo que (1):

$$\text{Peso Combustible} = \text{Peso Tanque} - \text{Peso Total (Lectura)}. \quad (1)$$

Ahora, conocemos que un litro de gasolina es equivalente a un Kilogramo y según la ficha técnica del dispositivo, indica que la capacidad máxima de combustible del generador es de 50 litros, esto quiere decir que 50 kilogramos equivalen al 100% de capacidad.

III. RESULTADOS

Una vez instalado el prototipo y conectado a la planta eléctrica se realizaron las siguientes pruebas de funcionamiento.



Fig. 7. Prueba de encendido.
Fuente: Autores.

- *Pruebas de encendido automático:* Para realizar la prueba de encendido automático del generador eléctrico, se simuló un fallo eléctrico (Fig. 7), el sistema notificó al usuario que se inició el proceso de encendido del generador eléctrico.
- *Pruebas de apagado automático:* Esta prueba se llevó a cabo bajo las siguientes condiciones iniciales: Simulación de fallo de energía eléctrica del proveedor y Generador eléctrico activado tal como se muestra en la Fig. 8. Se procede a restablecer el fluido eléctrico del proveedor, como respuesta el Arduino realizó una comprobación por 10 segundos de presencia eléctrica antes de iniciar el proceso de apagado.
- *Medición del nivel de combustible y temperatura:* Para garantizar una mejor precisión en el nivel de combustible se amortiza el resultado de la siguiente manera, se toman 20 muestra de lectura y se entrega el promedio (Fig. 8).

```

Estado proveedor: 0
Estado generador: 1
Nivel de combustible: 60 L
Temperatura del generador: 34° C
Estado proveedor: 0
Estado generador: 1
Nivel de combustible: 60 L
Temperatura del generador: 34° C
Estado proveedor: 0
Estado generador: 1
Nivel de combustible: 60 L
Temperatura del generador: 34° C

Orden recibida...
Iniciando el proceso de apagado del generador.....
Generador apagado

Estado proveedor: 0
Estado generador: 0
Nivel de combustible: 60 L
Temperatura del generador: 34° C
    
```

Fig. 8. Apagado automático.
Fuente: Autores.

En la Fig. 9 se muestra el generador eléctrico encendido previamente permitiendo ver en pantalla la temperatura y nivel de combustible.

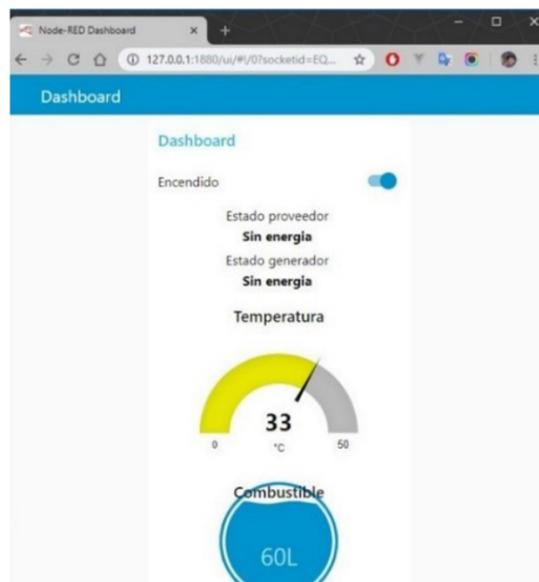


Fig. 9. Medidor de temperatura y combustible.
Fuente: Autores.

IV. CONCLUSIONES

La tecnología en Colombia ha venido creciendo con pasos agigantados, sobre todo en proyectos referentes a la implementación de IoT en los distintos sectores. En lo concerniente al campo energético aún falta por investigar y desarrollar, de lo poco que se explorado solo se enfoca en el sector industrial, no se tiene en cuenta las medianas y/o pequeñas empresas ni conjuntos residenciales.

Soluciones tecnológicas como la que se está presentando en este artículo, nos permiten tener un mayor y mejor control sobre el flujo eléctrico y de igual manera nos garantiza un mejor uso de los generadores eléctricos. Con la integración de IoT se busca darle una mejor administración al generador y a la vez comodidad al usuario; debido a que, el generador no deberá estar el lugar de instalación. Esto permite diseñar un producto fácil de manipular y gestionar.

Hoy en día, las tecnologías modernas de hardware y software permiten el desarrollo de este tipo de soluciones a muy bajo costo, esto ofrece una ventaja en el mercado porque es más accesible por parte de los usuarios que pueden interesarse por un producto que optimizara sus procesos, gestionara un mejor mantenimiento de su generador eléctrico y nada costoso de adquirir.

FINANCIAMIENTO

Artículo de investigación científica derivado del proyecto de investigación “Implementación de un sistema de automatización y administración de un generador eléctrico basados en Arduino”, financiado por “Corporación Universitaria Americana”. Año de inicio: 2019, año de finalización: 2020.

REFERENCIAS

- [1] A. Colmenar, D. Borge, E. Collado y M. Castro. *Generación Distribuida, Autoconsumo y Redes Inteligentes*. MD, Esp.: Editorial UNED, 2015.
- [2] E. Avendaño, “Tenderos reportan pérdidas de hasta \$500 mil al día por apagones”, en *El heraldo*, Mar. 2019. [Online]. <https://www.elheraldo.co/barranquilla/tenderos-reportan-perdidas-de-hasta-500-mil-al-dia-por-apagones-609276>
- [3] L. Aleaga, C. Morell y Z. García, “Generator Start-Up Automated Planning for Electric Power System Restoration,” *Ing Energ*, vol. 36, no. 2, pp. 168–179, 2019.
- [4] I. R. Campuzano-Martínez. “Diagnostic of Electric Power Generators with On-Line and Off-Line Monitoring Techniques,” *Información tecnológica*, vol. 27, no. 2, pp. 11–20, 2016. <https://doi.org/10.4067/S0718-07642016000200003>
- [5] D. Suárez, R. Morales, I. Cordero y L. Schreiner, “Diseño de una herramienta de medición de ruidos basados en tecnologías Arduino-Raspberry PI,” *Producción+ Limpia*, vol. 12, no. 1, pp. 81–87, Jun. 2017. <http://dx.doi.org/10.22507/pml.v12n1a8>
- [6] H. O. Gualli. “Análisis y control inalámbrico de iluminación del templo de la iglesia ACYM SUR mediante tecnologías Raspberry Pi y plataforma Android que garantice un manejo eficiente de luminarias,” *Trabajo de grado, Fac. Ing.*, Univ. Israel, Quito, Ecuador, 2017. Disponible en <http://repositorio.uisrael.edu.ec/handle/47000/1320>
- [7] L. Salman, S. Salman, S. Jahangirian, M. Abraham, F. German, C. Blair & P. Krenz, “Energy efficient IoT-based smart home,” presentado en el *3er IEEE World Forum on Internet of Things*, (WF-IoT), Reston, VA, USA, Dec. 12-14, 2016. <http://dx.doi.org/10.1109/WF-IoT.2016.7845449>
- [8] A. Vega. “Gestión de la energía eléctrica domiciliar con base en la gestión activa de la demanda,” *Tesis Ph.D. Fac. Ing.*, UD, Btá, D.C., CO. 2018. Disponible en <http://repository.udistrital.edu.co/>
- [9] J. García, ¿Cómo aplicar el IoT en el sector energético?, en *telcel.com*, Jun. 30. Disponible: https://www.telcel.com/empresas/tendencias/notas/aplicar-iot-en-sector-energetico?utm_campaign=Soluciones_Empresariales_C115_PYME_Necesidad_IoT&utm_source=facebook&utm_medium=social
- [10] A. Salazar, “Sistema electrónico de monitoreo y control para la distribución de energía eléctrica en los hogares,” *Tesis grado, Fac. Ing. Sist. Elect. Ind.*, UTA, Ambato, Ecuador, 2017.
- [11] S. K. Vishwakarma, P. Upadhyaya, B. Kumari y A. K. Mishra. “Smart Energy Efficient Home Automation System Using IoT,” presented at the *4th International Conference on Internet of Things: Smart Innovation and Usages*, IoT-SIU, Ghaziabad, IN, Abril 18-19, 2019. <http://dx.doi.org/10.1109/IoT-SIU.2019.8777607>
- [12] B. V. Mathiesen, H. Lund, D. Connolly, H. Wenzel, P. A. Østergaard, B. Möller & F. K. Hvelplund. “Smart Energy Systems for coherent 100% renewable energy and transport solutions,” *Applied Energy*, vol. 145, pp. 139–154, May. 2015. <http://dx.doi.org/10.1016/j.apenergy.2015.01.075>
- [13] R. A. Light “Mosquito: server and client implementation of the MQTT protocol,” *J. Open Source Soft.*, vol. 2, no. 213, pp. 265–266, May. 2017. <https://doi.org/10.21105/joss.00265>
- [14] J. Toldinas, B. Lozinskis, E. Baranauskas & A. Dobrovolskis. “MQTT Quality of Service versus Energy Consumption,” presented at the *23rd International Conference Electronics*, IEEE, Palanga, LTU, Jun. 17-19, 2019, pp. 1–4. <https://doi.org/10.1109/ELECTRONICS.2019.8765692>
- [15] J. Roldan, *Automatismos y cuadros eléctricos*. MD, Esp.: Thompson paraninfo, 2007.

Diana Suárez es Ingeniero de Sistemas, Especialista en Ingeniería de software, Magister en Administración e innovación, doctoranda en ciencia y tecnología informática. Profesor de tiempo completo de la Universidad de Libre seccional Barranquilla, Investigador asociado ante Min ciencias. Áreas de investigación aprendizaje automático, procesamiento del lenguaje natural, recuperación y extracción de información. <https://orcid.org/0000-0002-3172-7152>

Jhon Solano es Ingeniero de Sistemas de la Corporación Universitaria Americana.

Manuel Angarita es Ingeniero de Sistemas de la Corporación Universitaria Americana.

Carlos Henríquez Miranda es Ingeniero de Sistemas, Especialista en estudios pedagógicos, Magister en Ingeniería de Software y PhD en Ingeniería de Sistemas e informática. Profesor de la Universidad del Magdalena. Investigador Senior ante Minciencias. Áreas de investigación: aprendizaje automático, procesamiento del lenguaje natural y análisis de sentimientos. Editor de la revista científica Prospectiva. Líder de grupo de investigación Sistemas inteligentes y nuevas categorías categorizado A. Director de proyectos de software como consultor y profesor. Consultor e instructor en tecnología JAVA (J2EE, J2SE, JavaCard, Android). <https://orcid.org/0000-0002-0439-4954>

Dixon Salcedo es Ph.D. en Ingeniería con énfasis en Telecomunicaciones. Magister en Software Libre, con énfasis Administración de Redes de Computadores y Sistemas. Ingeniero de Sistemas. Profesor tiempo completo del programa Ingeniería de Sistemas, adscrito al Departamento de Ciencias de la Computación y Electrónica, y es miembro del grupo de Investigación de Ingeniería del Software y Redes de la Universidad de la Costa-CUC. Investigador Junior ante Minciencias. Intereses de investigación son en el campo de calidad de servicio en redes Internet, ingeniería de tráfico, redes de computadores y protocolos de nueva generación. <https://orcid.org/0000-0002-3762-8462>