

Sistema de Gestión de demanda eléctrica basada en la Web

Web-based Electricity Demand Management System

MORENO-DZUL, Julio†*, ALVAREZ-IBARRA, Maricela, SILVA-DZIB, Ismael y ARCEO-DIAZ, Rocío

Universidad Tecnológica de Cancún, Carretera Cancún-Aeropuerto, Km. 11.5, Mz. 5, Lt 1, SM 299, 77565 Cancún, Q.R., México

ID 1^{er} Autor: *Julio, Moreno-Dzul*

ID 1^{er} Coautor: *Maricela, Alvarez-Ibarra*

ID 2^{do} Coautor: *Ismael, Silva-Dzib*

ID 3^{er} Coautor: *Rocío, Arceo-Diaz*

DOI: 10.35429/JTO.2019.13.4.14.22

Recibido 8 de Marzo, 2020, Aceptado, 30 de Junio, 2020

Resumen

Se plantea una solución integral, innovadora y tecnológica, acerca de la predicción y gestión inteligente de la energía eléctrica, como la implementación de Redes Neuronales Artificiales (RNA) para el aprendizaje de patrones de comportamiento de la energía no lineal, la implementación de software de control inteligente que pueda ser la base de gestión de la energía a través de una interfaz de comunicación con las RNA. Igualmente, este proyecto busca a través de la implementación de técnicas computacionales vanguardistas construir un ecosistema multi-agente dotado de inteligencia artificial y conocimientos de la tarifa horaria para servicio general en media tensión, con demanda de 100 kW o más (tarifa HM) y módulos de gestión de cargas eléctricas críticas, pero que además a través del tiempo pueda ir aprendiendo nuevos conocimientos del ecosistema eléctrico y lo pueda aplicar en una gestión mejorada de la demanda. En esta fase de pruebas se trabajó en el diseño y construcción de dispositivos de hardware de medición, control e instrumentación electrónica de bajo costo, comandados por un sistema computacional distribuido con arquitectura Web 2.0 de control y monitoreo. La aplicación Web propuesta puede analizar el valor de diferentes variables eléctricas y medioambientales, puede calendarizar arranques y paros de equipos por medio de la comunicación del sistema con un controlador electrónico.

Demanda eléctrica, Redes neuronales artificiales

Abstract

The purpose of this investigation is to propose a comprehensive, innovative and technological solution about the intelligent prediction and management of electrical energy, such as the implementation of Artificial Neural Networks (ANN) used for detecting and learning patterns of behavior of nonlinear energy, including the deployment of intelligent software used to control energy management via a communication interface with the RNA. Furthermore, through the implementation of innovative computational techniques, this project aspires to develop a multi-agent endowed with artificial intelligence and knowledge of the rate for service general medium voltage with demand or 10 kW or more and modules of managents of critical electrical loads. Additionally, over time to learn new skills related to the electrical ecosystem that can be applied in an improved demand management. In this initial test phase, we have been working on the design and construction of hardware measurement devices, as well as low-cost controls and electronic instrumentation regulated by a computer system equipped with Web 2.0 architecture. The Web application proposal can analyze the value of different electrical and environmental variables, as well as schedule equipment starts and stops through communication with an electronic controller device.

Electricity demand, Artificial neural networks

Citación: MORENO-DZUL, Julio, ALVAREZ-IBARRA, Maricela, SILVA-DZIB, Ismael y ARCEO-DIAZ, Rocío. Sistema de Gestión de demanda eléctrica basada en la Web. Revista de Operaciones Tecnológicas. 2020. 4-13: 14-22

* Correspondencia del Autor (Correo electrónico: jmoreno@utcancun.edu.mx)

† Investigador contribuyendo como primer autor.

Introducción

La Secretaría de energía en México plantea la importancia de la gestión de la demanda eléctrica con base en los múltiples beneficios que traen tanto a los clientes como a los proveedores de servicios, tales beneficios como:

- Conocimiento de la estructura tarifaria del personal operativo.
- Involucramiento del personal para conocer todas las etapas del proceso.
- Crecimiento de la cultura del ahorro en la organización.
- Disminución del cargo de la demanda facturable en kW.
- Disminución del cargo por consumo kWh.
- Empresas más competitivas.
- Aumento de la vida útil de los equipos.

Por otro lado una gestión eficiente de la demanda no solo trae beneficios para los usuarios de la energía sino también a las empresas suministradoras (Lascurain, 2012), estos beneficios pueden ser:

- Reducir el requerimiento de demanda de energía eléctrica en el horario punta, generando la disminución de operación de equipos de generación de energía eléctrica basadas en combustibles no renovables en el Sistema Eléctrico Nacional.
- Disminución de pérdidas en los conductores en las redes de transmisión, y distribución.
- Aumento de la vida útil de los equipos.
- Diferir las inversiones en infraestructura.

A pesar de que existen diversos mecanismos y que los métodos y la tecnología para el control de la demanda eléctrica está en constante avance, aún sigue teniendo un costo elevado, además de que las soluciones generales o internacionales pueden presentar problemas como el desconocimiento de la tarifa HM o la diferenciación entre los conceptos y tiempos que se utilizan para el cobro de energía.

El proyecto planteado busca implementar técnicas computacionales vanguardistas para construir un ecosistema multi-agente dotado de inteligencia artificial y conocimientos de la tarifa HM.

Como se mencionó con anterioridad los sistemas de gestión de energía de última generación tienen un costo elevado de adquisición para la gran mayoría de las PyMES en México e incluso para algunas grandes empresas, y aunque la inversión realizada para adquirir estos equipos o sistemas está completamente justificada por el ahorro económico y el cuidado ambiental, existe una enorme cantidad de empresas que no consideran su adquisición debido a cuestiones financieras.

Para demostrar la disminución en costos que se piensa alcanzar se realizará la comparación de un dispositivo fundamental dentro de un sistema de gestión de demanda, como es una tarjeta de adquisición de datos.

Por ejemplo el MDS NETio de la marca General Electric empresa líder mundial en el ramo energético es un dispositivo cuyo fabricante describe como una solución inalámbrica que provee comunicación a distancia, permitiendo a los usuarios adquirir datos de diferentes sensores como por ejemplo de presión y flujo, pero que además puede fungir como elemento de control; todo esto gracias a sus puertos de entrada y salida tanto analógicos como digitales. (GE General Electric, 2016). Este dispositivo cuenta con ocho puertos, dos entradas y dos salidas digitales así como dos entradas y dos salidas analógicas.

El costo de este producto directamente adquirido con los fabricantes es de 1,323.00 dólares americanos (USD) como se puede observar en el sitio Web oficial del fabricante. Una solución similar pero basada en el SBC Beaglebone Black con la capacidad de manejar hasta 65 pines de entrada/salida digitales y 7 pines de entrada analógicos, con la solución inalámbrica es posible fabricarla con 140 USD, con la flexibilidad de poder ejecutar software creado en cualquier lenguaje de programación soportado por el Sistema Operativo GNU/Linux. (BeagleBone.org, 2016).

Al comparar los números de soluciones similares, se puede hablar de una relación de costos de uno a diez. Es importante mencionar que este proyecto no busca en una primera etapa competir de manera directa con tecnologías consolidadas en el mercado como lo son los productos de marcas como GE o Siemens.

El objetivo es proveer a la industria mexicana una alternativa de gestión inteligente de demanda más económica que la existente en el mercado, basada en componentes libres y que brinden a los desarrolladores tener disponible una plataforma base para la implementación de nuevos elementos computacionales que permitan una gestión más inteligente que conlleve a mejores resultados. La nueva Ley de la Industria Eléctrica en México permite la inclusión de entes particulares para la generación y distribución de energía eléctrica, lo que propiciará la aparición de entornos desagregados de manera parcial o total por lo cual la predicción de la demanda eléctrica se volverá fundamental, no solo para las empresas, sino para los consumidores finales, es por ello que el desarrollo de modelos computacionales de inteligencia artificial que permitan realizar una predicción de demanda a corto plazo le dan un valor agregado importante y fundamental al sistema de gestión propuesto en este proyecto.

Marco teórico

Control de demanda de energía eléctrica.

La energía eléctrica es un bien indispensable y de interés estratégico en la economía de México, es por ello que es importante, no solo tener la capacidad de producir la energía necesaria sino que es importante producir lo justo para evitar desperdicios que solo impactarían negativamente en el costo de la energía; entonces la predicción de la demanda eléctrica se vuelve fundamental para las empresas generadoras y distribuidoras, pero también para los consumidores quienes pueden ver afectados sus costos de producción de manera significativa si no tienen un cuidado adecuado de la demanda facturable, según los parámetros acordados en su contrato de servicio eléctrico. (Barrero, 2004)

La demanda eléctrica con frecuencia presenta comportamientos no lineales, por lo cual suena lógico encontrar soluciones más óptimas para el problema de predicción de la demanda eléctrica a través de modelos no lineales de entre los cuales sobresalen las técnicas basadas en Redes Neuronales Artificiales, las cuales debido a su capacidad de generalización y sobre todo, la facilidad de descubrir relaciones no lineales en entornos complejos parecieran ideales para abordar el problema de predicción de demanda. (Amjady, 2007)

En los modelos desarrollados de RNA e híbridos utilizados en gestión de demanda eléctrica se tienen los siguientes: (Villagarcía, 2008), (Alejandro, 2002)

- Modelos supervisados feedforward: Backpropagation, Perceptrón Multicapa, Red Neuronal Probabilística, Máquinas de Soporte Vectorial, Maquinas de Soporte Vectorial de Mínimos Cuadrados.
- Modelos supervisados feedback: Modelo Elman.
- Modelos no supervisados: Mapas Auto Organizados.
- Modelos Híbridos: Mapas Auto Organizados con fuzzy-rough, Modelo de Combinación Fuzzy Adaptativo, Función de Base Radial.

La principal diferencia de las técnicas no lineales frente a las lineales radica en que aquéllas no emplean tiempo en disponer de un modelo matemático claro de las dependencias de los parámetros para explicar el problema de la predicción de la demanda, en cambio, los modelos lineales requieren de la definición precisa de las ecuaciones que lo definen.

Lo que la totalidad de los modelos RNA requerirán es la adecuada parametrización de las variables de entrada al modelo, así como la definición topológica de la red (número de capas, neuronas en las mismas, funciones de entrenamiento, etc.); no obstante, esto es un trabajo sencillo y asumible (un script) si se compara con la parametrización de las funciones de los modelos lineales. No obstante, si el sistema de predicción va acumulando etapas en la predicción, con sucesivos modelos, estos modelos adicionales, supondrán un coste computacional adicional al modelo global, tanto para la fase de aprendizaje global, como para la fase de operación (predicción). (J. Wang, 2012)

Microcontroladores

Los microcontroladores están presentes en muchos de los productos electrónicos que se emplean en la vida cotidiana. Existe una gran variedad de modelos existentes en el mercado y una gran cantidad de aplicaciones posibles de funcionamiento y en las arquitecturas de muchos microcontroladores.

El microcontrolador Beaglebone es una placa de computadora de hardware libre para el desarrollo de software de OpenSource. Esta placa funciona con OMAP3530 System-on-a-chip. Fue desarrollada por un pequeño grupo de ingenieros como una placa que podría ser usada alrededor del mundo para enseñar las capacidades del software y hardware libre. La placa mide aproximadamente 75mm por 75 mm y cuenta con todas las funcionalidades de una computadora básica (S. F. Barrett, 2013), es uno de los microcontroladores más potentes y eficaces que existen en el mercado, gracias a su capacidad de manejar sistemas operativos embebidos.

El Beaglebone Black es un modelo del microcontrolador Beaglebone, esta placa al igual que los demás modelos cuenta con el OMAP3530 que incluye un CPU ARM Cortex-A8, en el cual pueden ser instalados los sistemas operativos Linux (cualquier distribución), Minix, FreeBSD, OpenBSD, RISC OS / Symbian, un DSP TMS320C64x+ para la decodificación acelerada de audio y video y una GPU PowerVR SGX530 para el renderizado en 2D y 3D. La salida de video está provista de dos conexiones separadas S-Video y HDMI. Igualmente cuenta con una entrada para tarjetas SD/MMC con soporte SDIO, un puerto USB On-The-Go, un conector serial RS-232, una conexión JTAG y dos jacks de 3.5 mm para entrada y salida de audio. El almacenamiento y la memoria interna son provistos por un chip PoP que incluye 256mb de memoria flash NAND y 256mb de RAM. Usa arriba de 2W de alimentación y puede ser alimentada por el conector USB o por una fuente externa de 5V. (McLaughlin, 2016)

Inteligencia Artificial

Una de las ramas más destacadas del campo científico de la Inteligencia Artificial es la que corresponde a las Redes Neuronales Artificiales (RNAs). Las RNAs tratan de emular el comportamiento del cerebro humano, caracterizado por el aprendizaje a través de la experiencia y la extracción de conocimiento genérico a partir de un conjunto de datos. Esto puede ser mediante la construcción física de sistemas cuya arquitectura se aproxima a la estructura de la red neuronal biológica. (R. Flórez, 2008)

Metodología

El proyecto se centra en el desarrollo de la aplicación Web 2.0 para el control y medición de la energía eléctrica y la interfaz de comunicación entre las cargas eléctricas y el sistema, que comprende el microcontrolador y las conexiones TCP. La aplicación será capaz de realizar lecturas y mediciones apoyada por el microcontrolador Beaglebone black y presentarlas mediante gráficas. Igualmente este software debe de ser capaz de aprender las tendencias del lugar donde se implemente para que en un futuro pueda decidir de manera autónoma. La aplicación tiene que almacenar todos los datos recibidos en la base de datos, para que el usuario pueda acceder a ellos cuando lo desee. De la misma manera la aplicación debe de tener la capacidad de gestionar las cargas eléctricas mediante un control de lazo cerrado y salida de potencia tipo on/off en tiempo real, visualizar parámetros eléctricos, ambientales y de proceso, generación de alertas y capacidad de administrar los propios usuarios.

Esta aplicación debe ser agradable, fácil de usar y eficaz.

Se utilizó para el desarrollo del Sistema en Tiempo Real el método llamado Hard Real-Time Hierarchical Object Oriented Design, el cual está enfocado a la construcción de los sistemas críticos, ya que de no producir una respuesta dentro de un intervalo de tiempo prefijado pueden ocasionarse graves daños en los procesos. (Wolfgang Nebel, 2003)

En HTR-HOOD se utilizó un proceso de desarrollo iterativo.

El diseño se realizó progresivamente a través de la especificación de más responsabilidades, mismas que definen las propiedades del sistema con las que opera el diseñador y que este no tiene la libertad de cambiar.

En el proceso de refinamiento del diseño, se transformaron las obligaciones (aspectos del diseño que no están sujetas a responsabilidades) y responsabilidades, a menudo se encuentra sujeto a restricciones impuestas por el entorno de ejecución, es decir el conjunto de componentes de hardware y software sobre el que se construye el sistema.

Se imponen restricciones de recursos como la velocidad del procesador, el ancho de banda de las comunicaciones y restricciones de los mecanismos como la prioridad de las interrupciones, la planificación de tareas y el bloqueo de los datos, con esto se debe proporcionar reconocimiento explícito de los tipos de actividades/objetos que se pueden encontrar y catalogar como críticos en la construcción del Sistema Inteligente de Gestión, como son:

1. Integración de los parámetros apropiados de planificación de los procesos.
2. Definición explícita de los requisitos temporales de cada objeto y aplicación.
3. Definición de la importancia relativa de cada objeto para el funcionamiento correcto de la aplicación.
4. Facilidades para realizar el análisis de la planificación del sistema.

El proceso de desarrollo iterativo constó de las siguientes etapas: definición de requisitos, diseño de la arquitectura física, diseño detallado, codificación y cálculo de tiempos de ejecución, pruebas y medida de tiempos.

El diseño de la arquitectura lógica será la fase destinada fundamentalmente a la satisfacción de los requisitos funcionales. El resultado de esta etapa será un conjunto de objetos terminales.

En cuanto a la arquitectura física, se tiene como objetivo:

1. Relacionar la arquitectura lógica con los recursos de ejecución.
2. Asignar atributos temporales a los objetos y asegurar el cumplimiento de los requisitos no funcionales.
3. Análisis de plazos de respuesta.
4. Fiabilidad y seguridad.

En el diseño de la arquitectura física se consideran principalmente cuatro actividades:

1. Asociación de los objetos resultantes de la arquitectura lógica a los recursos de hardware (procesadores) disponibles.

2. Diseño de la red, debido a que se está considerando un sistema distribuido, hay que planificar la comunicación entre los diferentes nodos (ordenadores de placa única, microcontroladores, equipo de red), de forma que el tiempo de retardo en las comunicaciones esté acotado y sea aceptable para asegurar de no causar problemas en la operación.
3. Planificación de los procesadores. Determinar la planificación, estática o dinámica que asegure que todas las tareas del sistema cumplan con los plazos establecidos.
4. Análisis de la fiabilidad: determinar si es conveniente emplear técnicas de fallos, para aumentar la fiabilidad del sistema.

El resultado del modelo físico es un sistema de objetos terminales con atributos temporales.

Aunque la arquitectura física es un refinamiento de la arquitectura lógica, el desarrollo de ambas será iterativo y concurrente.

Las pruebas del sistema de medición tanto unitario como distribuido se realizaron comparando los resultados medidos y calculados con un equipo registrador trifásico de parámetros eléctricos de voltaje, corriente, demanda, consumo, factor de potencia, frecuencia.

Las simulaciones de los modelos de red neuronal se realizaron con Matlab en diferentes escenarios, desde consumidores de tarifa HM convencionales en la red, como en entornos desagregados como pequeñas ciudades, plantas industriales o grandes hoteles.

En cuanto a la estructura lógica, o la aplicación se desarrollo bajo la metodología de desarrollo de software Extreme-Programming, que permite desarrollar la aplicación al mismo tiempo del análisis y el diseño. Igualmente permite detectar errores de cualquier fase del desarrollo y corregirlos sin la necesidad de volver a la fase. Gracias a esto se pudo ahorrar tiempo en el desarrollo y se centro en las pruebas.

Diseño del sistema

En la Figura 1 se muestra el caso de uso general, que representa los módulos de la aplicación y los accesos permitidos de cada usuario.

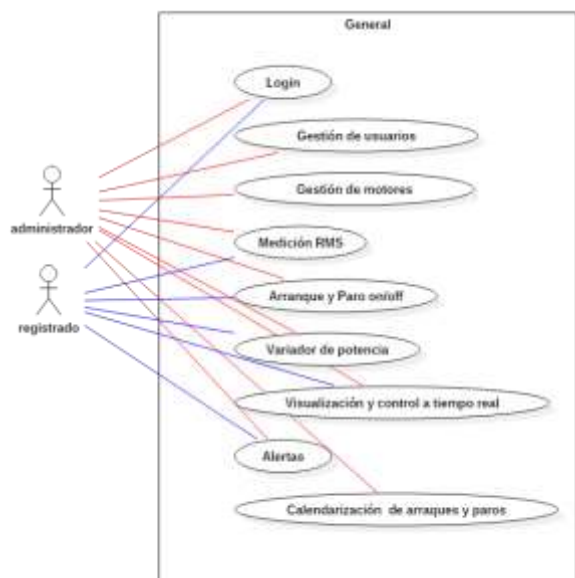


Figura 1 Caso de uso general

En la Figura 2 se muestra el diagrama de secuencia del proceso más importante de comunicación entre la aplicación Web y el microcontrolador.

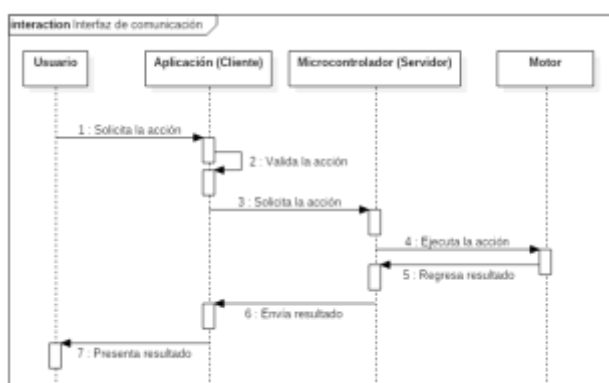


Figura 2 Interfaz de comunicación

En la Figura 3 se presenta el diagrama de secuencia del proceso de generación de alertas en el sistema.

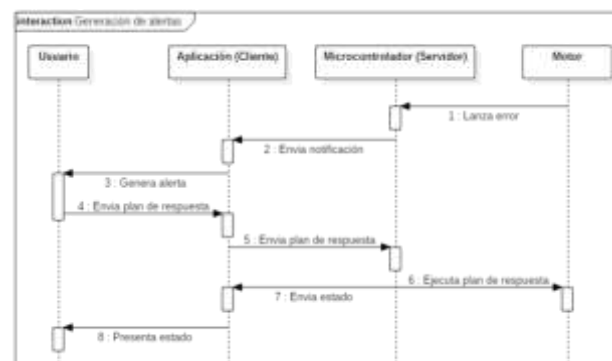


Figura 3 Generación de alertas

Codificación de la aplicación

La aplicación Web se desarrolló en su totalidad con en el entorno integrado de desarrollo (IDE) Eclipse Luna. Las interfaces gráficas fueron desarrolladas con el lenguaje de etiquetado XML y HTML. Se utilizó Javascript para efectos visuales y CSS3 para el diseño y personalización. Igualmente se utilizó para la implementación de componentes enriquecidos ZK Framework. Los procesos del sistema junto con la funcionalidad y las conexiones fueron desarrollados en el lenguaje de programación Java. Para la base de datos se utilizó el gestor MySQL y fue construida bajo un esquema de mapeo ORM (Object-Relational Mapping) con JPA y Hibernate. Esto permite mapear la base de datos directamente desde la aplicación, convirtiendo las entidades en tablas. Para la seguridad de la aplicación se utilizó Spring Security y Spring Framework. Igualmente el proyecto fue gestionado a nivel de paquetería por Maven.

Toda la aplicación fue desarrollada bajo el patrón de diseño MVC (Modelo-Vista-Controlado), esto permite un desarrollo más organizado y rápido, además de que facilita el mantenimiento y soporte de la aplicación. (Díaz González, 2012)

Construcción de la interfaz de comunicación.

La construcción de la interfaz de comunicación consistió en el desarrollo de una plataforma cliente-servidor que sea capaz de comunicar en tiempo real las cargas eléctricas con la aplicación Web. Esta interfaz se desarrolló con el microcontrolador Beaglebone Black con el sistema operativo embebido Debian Jessie. Igualmente en el microcontrolador se instaló la clase TCP Server, la cual proveía y fungía como puente de comunicación entre las cargas eléctricas y la aplicación.

Pruebas e implementación

Las pruebas estuvieron compuestas en dos partes. La primera parte fueron las pruebas de caja blanca, que se realizaron a la par de la codificación de la aplicación. Estas consistieron en ir detectando errores a la hora de compilar o ejecutar la aplicación e ir corrigiéndolos.

La segunda parte consistió en las pruebas de caja negra, en las cuales se hicieron pruebas de estrés a la aplicación para saber el número de conexiones entrantes y salientes que permite. Igualmente se hicieron pruebas de funcionamiento y conectividad para detectar posibles bugs, ya sean de seguridad o de codificación, que puedan poner en peligro la integridad de la aplicación y la interfaz de comunicación. Estas pruebas se hicieron con la ayuda y soporte de herramientas como, Junit y Selenium, las cuales nos facilitaron la detección de posibles fallos gracias a su gran eficacia.

La aplicación fue implementada en un servidor Tomcat 8. El proyecto fue exportado en un archivo war, y desde ahí se genera la ejecución de la aplicación para poder ser accedida.

Resultados obtenidos

1. Se obtuvo un análisis completo de los requerimientos, variables de entrada y salida y procesos de la aplicación con base al desarrollo de software.
2. Se obtuvo una aplicación Web 2.0 para el monitoreo y gestión de los parámetros eléctricos y medioambientales con compatibilidad de plataformas.

En la Figura 4 se muestra la pantalla de acceso a la aplicación.



Figura 4 Acceso a la aplicación

En la Figura 5 se muestra la pantalla principal de la aplicación en la cual se pueden apreciar las mediciones en tiempo real del motor seleccionado, igualmente de lado derecho se puede apreciar el módulo de arranque y paro y el variador de potencia. De lado izquierdo, en la barra lateral, se puede apreciar el botón de paro de emergencia.



Figura 5 Dashboard

En la Figura 6 se muestran las mediciones graficadas de corriente y voltajes del motor seleccionado.

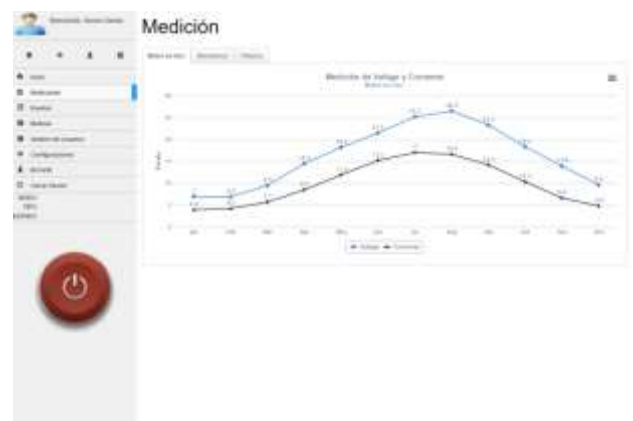


Figura 6 Medición de corriente y voltaje

En la Figura 7 se muestra las mediciones graficadas de los motores monofásicos.

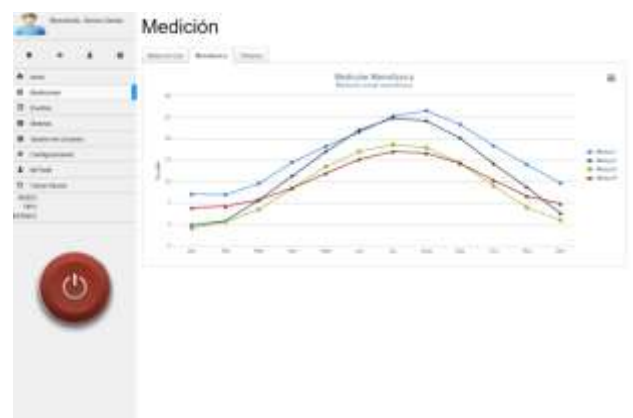


Figura 7 Medición monofásica de motores

En la Figura 8 se muestra las mediciones graficadas de los motores trifásicos.

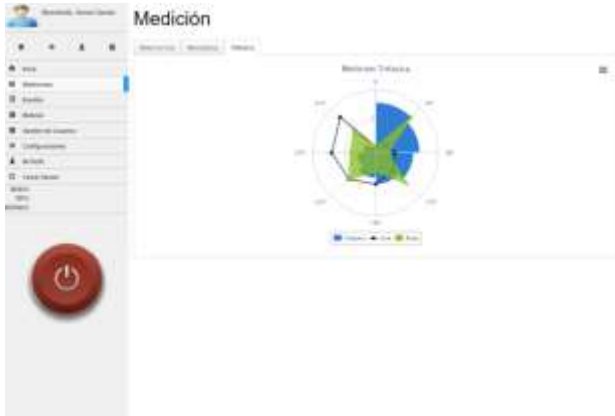


Figura 8 Medición trifásica de motores

3. Se obtuvo una base de datos compatible con la aplicación Web para que pueda ser usada por la misma.
4. Se obtuvieron mediciones de parámetros eléctricos en tiempo real corriente RMS y Voltaje RMS para sistemas monofásicos y trifásicos de hasta 600 voltios (CAT IV) de acuerdo al estándar IEC 61010.

En las Figuras 9 y 10 se muestran los resultados de las mediciones RMS realizadas.

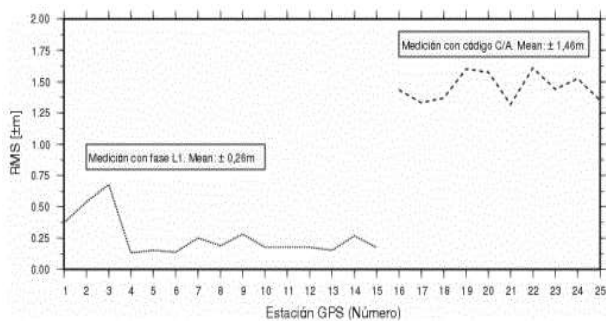


Figura 9 Medición 1

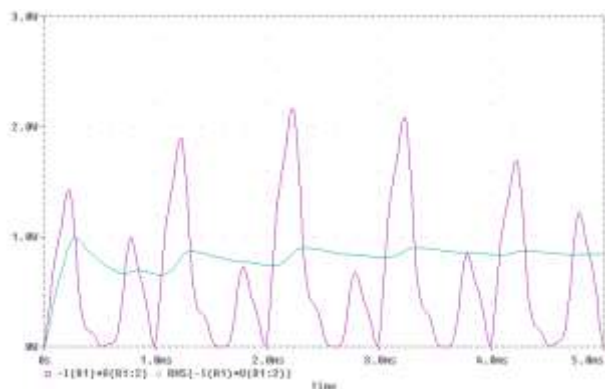


Figura 10 Medición 2

5. Se obtuvo un cálculo de potencias (activa, reactiva, aparente) y factor de potencia.
6. Se obtuvo una calendarización de arranques y paros de actuadores industriales según programa operativo (diario, mensual, anual) mediante la aplicación Web.

En la Figura 11 se muestra la calendarización de arranques y eventos de motores.

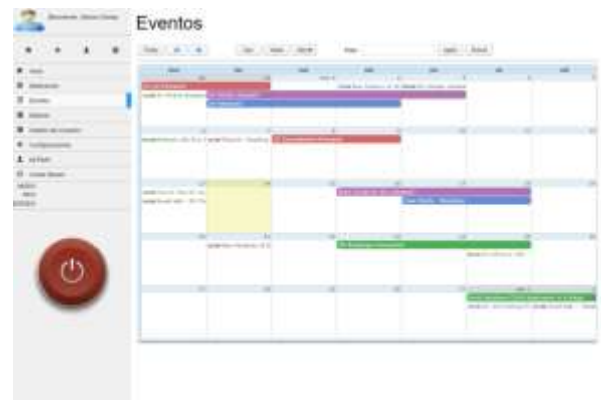


Figura 11 Calendarización de arranques y eventos

7. Se obtuvo una generación de alertas en tiempo real según mediciones o funcionamientos anómalos de los equipos en la red.
8. Se obtuvo una gestión autónoma de arranques y paros según las predicciones realizadas.
9. Se obtuvo la interfaz de comunicación entre los motores y la aplicación con el Beaglebone Black.

En la Figura 12 se muestran las pruebas de la construcción de la interfaz de comunicación en una protoboard.

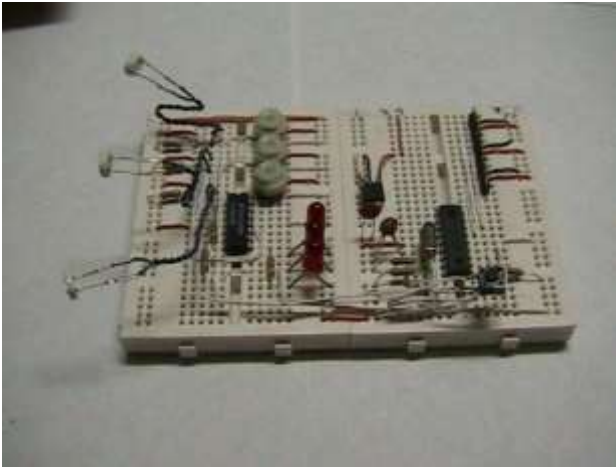


Figura 12 Prueba en protoboard

En la Figura 13 se muestran las pruebas de la interfaz de comunicación con el Beaglebone Black.

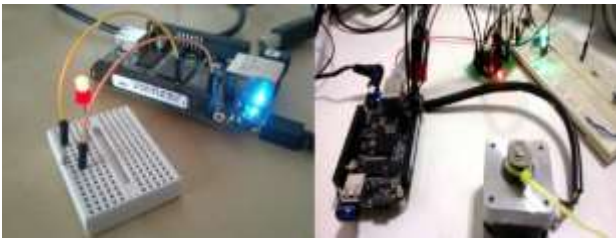


Figura 13 Pruebas con Beaglebone Black

Conclusiones

Con el desarrollo de la aplicación Web se puede gestionar y monitorear de manera inteligente el uso de energía, también por medio del aprendizaje de patrones se puede calcular la demanda de energía eléctrica.

Se logró la construcción de la interfaz de comunicación con un Beaglebone Black funcionando como servidor en un modelo TCP cliente/servidor. También se logró el análisis, diseño y desarrollo de la aplicación Web, con los requerimientos estipulados. Igualmente se lograron pruebas satisfactorias reales con motores trifásicos y monofásicos.

El prototipo está en fase de pruebas y adecuaciones para su implementación, sin embargo falta más trabajo de investigación y análisis de los patrones del comportamiento no lineales de la energía eléctrica, diseño y adecuación del modelo de predicciones, así como pruebas con Beaglebone Black.

Posteriormente se debe realizar la adecuación con cargas eléctricas variadas para que se pueda liberar el prototipo, además de incluir nuevas funciones en la aplicación Web para que sea comercializable.

Referencias

Alejandro, J. C. (2002). *Análisis de sistemas de energía eléctrica*. Servicio de Publicaciones. Universidad de Ovideo.

Amjady, N. (2007). *Short-Term Bus Load Forecasting of Power Systems by a New Hybrid Method*. IEEE Transactions on Power Systems.

Barrero, F. (2004). *Sistemas de energía eléctrica*. Madrid: International Thomson Editores Spain Paraninfo, S. A.

BeagleBone.org. (09 de 2016). *beagleboard*. Recuperado el 1 de 10 de 2016, de <https://beagleboard.org/black>

Díaz González, F. R. (2012). Patrón Modelo-Vista-Controlador. *Revista Telemática*, 11 (1). GE General Electric. (2016). *gegridsolutions*. Recuperado el 27 de 04 de 2016, de <http://www.gegridsolutions.com/Communications/catalog/netio.htm>

Lascurain, G. I. (07 de 05 de 2012). <http://www.bib.uia.mx/tesis/pdf/015599/015599.pdf>. Recuperado el 25 de 04 de 2016, de <http://www.bib.uia.mx/tesis/pdf/015599/015599.pdf>: h

McLaughlin, B. (2016). *The BeagleBone Black Primer*. Que Publishing.

R. Flórez, J. M. (2008). *Las redes neuronales artificiales. Fundamentos teóricos y aplicaciones prácticas*. Netbiblio, S. L.

Wolfgang Nebel, F. O.-R. (26 de 05 de 2003). *Object-Oriented Specification and Design of Embedded Hard Real-Time Systems*. Recuperado el 14 de 05 de 2016, de http://www.offis.de/uploads/tx_useroffis/icda-49-fromproceedings.pdf:

Villagarcía, C. (2008). *Atlas de la demanda eléctrica Española*. Red Eléctrica de España.