Universidad Autónoma de Querétaro
Facultad de Informática
Maestría en Sistemas Computacionales

Arquitectura de software SIEM para el monitoreo en línea de consumo de energía eléctrica en un entorno residencial utilizando un algoritmo inteligente

Opción de titulación
Tesis

Que como parte de los requisitos para obtener el Grado de
Maestría en Sistemas Computacionales

Presenta:
Martin Santiago Francisco

Dirigido por:
M.S.I. Sandra Patricia Arreguín Rico

M.S.I Sandra Patricia Arreguín Rico
Presidente — Firma

M.C. José Arturo Gaona Cuadra
Secretario — Firma

Dra. Ma. Teresa García Ramírez
Vocal — Firma

M.I.R. José Ángel Perea García
Suplente — Firma

M.I.S.D. Carlos Alberto Olmos Trejo
Suplente — Firma

M.S.I.D. Juan Salvador Hernández Valerio
Director de la Facultad

Dra. Ma. Guadalupe Flavia Loarca Piña
Directora de Investigación y Posgrado

Centro Universitario
Querétaro, Qro.
Noviembre 2018

RESUMEN

El presente trabajo de tesis tiene como objeto proveer una arquitectura de software SIEM que permita optimizar el consumo de energía eléctrica en un entorno residencial. Se diseñó e implementó un prototipo funcional que permite mediante sensores de corriente alterna recolectar valores de corriente eléctrica, que a su vez son transmitidas mediante el protocolo de comunicación wifi a una plataforma de Internet. Se utilizó Node-RED como un motor de flujos con enfoque en el Internet de las cosas el cual nos permitió generar las interfaces web en el que se despliegan los cambios de consumo con intervalos personalizados por el usuario y a su vez hacer la conexión con la base de datos InfluxDB para poder almacenar los datos. La arquitectura es segura, interoperable, escalable y sostenible. Las tarjetas de desarrollo utilizadas fueron Particle Photon y Raspberry Pi 3 B.

(**Palabras clave**: Monitoreo, arquitectura, energía eléctrica, optimización, voltaje)

SUMMARY

The aim of this thesis study is at providing a SIEM software architecture which enables optimizing the consumption of electricity in a residential environment. A functional prototype which allows alternating current sensor to collect electrical current values, which in turn are transmitted via Wi-Fi communication protocol to an Internet platform, was designed and implemented. Node-RED was used as a flow engine with an approach on Internet of things, which allowed us to generate web interfaces in which consumption changes would display with intervals determined by the user, and in turn complete the connection to the InfluxDB database in order to store data. The architecture will be safe, interoperable, scalable and sustainable. Development boards used were Particle Photon and Raspberry Pi 3 B.

(**Key words:** Monitoring, architecture, electric energy, optimization, voltage)

“El mayor riesgo es no tomar ningún riesgo”

Mark Zuckerberg

Fundador de Facebook

Dedicado a Dios y a mis Padres:

Mi Mamá, Anacleta Francisco Blas

Mi Papá, Salvador Santiago Francisco

AGRADECIMIENTOS

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por haberme otorgado una beca por 3 semestres en la Maestría en Sistemas Computacionales realizada en la Facultad de Informática de la Universidad Autónoma de Querétaro.

Agradezco a mi directora de tesis la MSI Sandra Patricia Arreguín Rico por sus atinados comentarios, experiencia y disposición para brindarme su orientación profesional para finalización de esta tesis, también agradezco al codirector de tesis M. en C. José Arturo Gaona Cuadra por compartir el conocimiento y su experiencia para el desarrollo de esta tesis.

Finalmente agradezco a mis amigos, profesores y compañeros, quienes han estado presente durante esta etapa compartiendo sus conocimientos, apoyo, ánimos, estudio y diversión.

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. INTRODUCCIÓN

El presente proyecto surge de la necesidad de reducir el consumo de energía eléctrica en entornos residenciales en base al monitoreo en línea del consumo de electricidad, es necesario desarrollar artefactos tecnológicos que contribuyan a uso más inteligente de la electricidad en los hogares.

1.1 Contenido de los capítulos

En el capítulo “Introducción” se describe el problema de investigación, el objetivo general y los particulares del proyecto, así como el contenido que se abordara a lo largo del documento.

En el capítulo “Estado del Arte” se expone las distintas tecnologías de comunicaciones, arquitecturas, modelos y algoritmos actualmente propuestos en otras investigaciones relacionadas con el monitoreo del consumo de energía eléctrica en entornos residenciales.

En el capítulo “Metodología” se describe la serie de pasos que se siguió en este proyecto de investigación, se presentan los requisitos del proyecto, diseños del hardware y software, las herramientas y la solución propuesta.

En el capítulo “Implementación” se describe de forma detallada la propuesta de arquitectura. En esta parte se expone el prototipo implementado para demostrar la funcionalidad de la arquitectura.

En el capítulo “Resultados” se presentan las distintas pruebas que se ejecutan para poder evaluar el prototipo implementado.

En el capítulo “Conclusiones” se describe la conclusión del proyecto de investigación, se resumen los principales puntos del trabajo realizado y se presentan futuras líneas de investigación.

1.2 Antecedentes

En el reporte emitido por U.S Energy Information Administration (2016) se proyecta que el consumo de energía eléctrica de uso residencial a nivel mundial incrementará en un 69% en 2040 y a 19.5% en 2020 y analiza que la electricidad es la forma de consumo de energía de uso final que más crece en el mundo y así lo ha sido durante las últimas décadas (Conti et al., 2016).

El sistema de energía eléctrica en los últimos cien años siempre ha tenido un flujo unidireccional el cuál comienza con su generación en las plantas eléctricas hasta llegar al usuario final. Sin embargo, en los últimos años debido a problemas ambientales, técnicos, económicos se han creado nuevos conceptos y sistemas que pretenden solucionar estos problemas. De acuerdo a Wang, Xu, y Khanna (2011) el sistema de energía de última generación es conocido como “Smart Grid”, de aquí en adelante lo llamaremos redes eléctricas inteligentes.

Tradicionalmente las medidas adoptadas por los usuarios para reducir el consumo de energía se han limitado a medidas aisladas y voluntarias aplicados directo a la carga, como por ejemplo cambiar a focos ahorradores o a luces LED. Las redes eléctricas inteligentes permiten que los clientes o sistemas tomen decisiones que permitan optimizar la energía eléctrica de su entorno.

El modelo conceptual de las redes eléctricas inteligentes se compone de varios dominios, cada uno a su vez compuesto por aplicaciones y actores que interaccionan constantemente, sin embargo, en este trabajo de investigación se enfocara en la aplicación de las redes eléctricas inteligentes en el uso residencial.

Por otro lado, los desarrollos de aplicaciones sustentados en crear redes eléctricas inteligentes contribuyen a: disminuir el cambio climático, aumentar la calidad de la energía suministrada, reducir de la dependencia energética del exterior y en general fomentar el uso de energías “verdes”.

1.3 Objetivos

1.3.1 Objetivo general:

Diseñar una arquitectura SIEM que permita monitorear en línea el consumo de la energía eléctrica en un ambiente residencial utilizando un algoritmo inteligente con el fin de optimizar y reducir el consumo de electricidad.

1.3.2 Objetivos particulares:

- Desarrollar una arquitectura con los atributos de SIEM (Seguridad, interoperabilidad, escalabilidad y mantenibilidad).
- Monitorear en línea el consumo de energía eléctrica en un ambiente residencial.
- Implementar un algoritmo inteligente que permita la reducción de energía con base a la adquisición de datos.
- Desarrollar un modelo para la transmisión de la información por un canal seguro.

2. ESTADO DEL ARTE

Referente a las arquitecturas enfocadas al monitoreo de las redes eléctricas y a las tecnologías de comunicación se presenta las más recientes investigaciones en este capítulo.

2.1 Arquitectura de software

Uno de los requerimientos fundamentales para la implementación de una red inteligente es la gestión en línea de un gran volumen de información. Lo anterior implica la participación de una infraestructura de TIC escalable, confiable y segura.

Las arquitecturas de automatización de las futuras redes inteligentes se están desarrollando en una serie de programas de investigación a gran escala en todo el mundo. Una importante iniciativa de investigación financiada por la Fundación Nacional de Ciencia de los Estados Unidos es la “FREEDM” (Futura distribución y gestión de energía eléctrica renovable). Strasser et al. (2015, p. 2430) señalan que “la primera característica clave de FREEDM es una interfaz plug and play, lo que permite que los componentes puedan integrarse perfectamente con el resto de la red”.

Una de las arquitecturas basadas en inteligencia artificial son los sistemas multi-agentes (MAS). En éstos cada componente, dispositivo o actor es monitoreado y controlado por un agente autónomo. Cada agente tiene un conjunto de capacidades específicas (como el control de electrodomésticos) y objetivos locales (por ejemplo, la reducción al mínimo de los costos de energía preservando al mismo tiempo la comodidad del usuario final). Strasser et al. (2015, p.2431) expresan que “los agentes interactúan a través de mensajes para coordinar su comportamiento con el fin de lograr un equilibrio entre los objetivos locales y de todo el sistema (como el equilibrio entre oferta y demanda)”. Negeri, Baken, y Popov (2013) proponen también una arquitectura, pero basada en principios holónicos.

Por otra parte, las arquitecturas diseñadas bajo el paradigma de “sistemas definidos por software” son “un concepto que permite abstraer el control de los dispositivos de hardware a diferentes capas con componentes de software. Este modelo proporciona la capacidad de controlar una amplia gama de recursos informáticos de una manera dinámica separando la capa de control de la capa de flujo de trabajo de los datos, es decir, aislando el control de dispositivos de hardware y estableciéndolo con una capa de software” (Jararweh et al., 2016, p. 2).

De acuerdo a Jararweh et al. (2016) la idea principal detrás de los “sistemas definidos por software” es manejar el control de todos los dispositivos independientes mediante el uso de estándares y protocolos generales.

De acuerdo a Jararweh, Al-Ayyoub, Bousselham y Benkhelifa (2015) se usan diferentes formas de “sistemas definidos por software” como: redes definidos por software (SDN), almacenamiento definido por software (SDStore), seguridad definida por software (SDSec) e Internet de las cosas definidos por software (SDIoT).

Otras de las tendencias que tienen las arquitecturas son que están conectadas al internet (IoT) y a la nube (Cloud), Guinard y Trifa (2009) señalan que un elemento importante de este tipo de arquitecturas es el uso de un “Gateway inteligente”, el cual es un elemento intermediario que permite realizar la conexión de la Web con un dispositivo que no tienen un IP. Lo anterior permite hacer uso de distintos tipos de protocolos de comunicación resolviendo problemas de interoperabilidad dentro de la red eléctrica inteligente. Hoy en día muchas de las investigaciones realizadas en el campo de las redes eléctricas están enfocadas en el desarrollo de puertas de enlace inteligentes. Un ejemplo de este tipo de modelos es el que propone Cui et al. (2014) el cual está compuesta de tres partes: una puerta de enlace, el servidor de nube y un dispositivo inteligente. La función del Gateway es identificar los electrodomésticos que utilizan servicios UPnP (Universal Plug and Play), obtener información de los estados de estos dispositivos y transmitir los datos a un servidor en la nube. El servidor de la nube además de almacenar estos datos

también proporciona servicios de monitoreo a los dispositivos electrodomésticos, mientras que el dispositivo inteligente permite a los usuarios monitorear y controlar las funcionalidades de los electrodomésticos.

Las recientes investigaciones acerca del diseño de arquitecturas en el campo de las redes eléctricas inteligentes están enfocadas a cumplir requerimientos de interoperabilidad, escalabilidad e integración dinámica en diversos tipos de controladores y dispositivos. Kim et al. (2017) demuestran como cumplir estos requerimientos mediante el diseño e implementación de un prototipo que consiste en una Gateway residencial que conecta diferentes tipos de electrodomésticos y proporciona interfaces estándares que son accesibles a los usuarios a través de servicios web mediante teléfonos inteligentes. El prototipo realiza descubrimiento de nuevos dispositivos seleccionados e integra servicios basados en información semántica. Este es un trabajo en el que se integran dispositivos basados en X10, Insteon, ZigBee y UPnP.

2.2 Tecnologías de comunicación

Un sistema de comunicaciones es un componente clave de la infraestructura de la red eléctrica inteligente. Hay dos medios por la cuales se trasmite la información por cableado o inalámbricamente, cada una de ellos ofrece ventajas y limitaciones. Mahmood, Javaid, & Razzaq (2015) identifican que los protocolos de comunicación más adecuadas en un área doméstica (HAN) son ZigBee, Bluetooth, Wi-Fi, 6LoWPAN y Z-Wave, mientras que las tecnologías inalámbricas en pequeñas comunidades (NAN) más viables son WiMAX y GSM.

Recientemente Bluetooth Smart o de bajo consumo (BLE) ha estado ganando terreno en el desarrollo de aplicaciones ya que está instalado en teléfonos móviles y muchas computadoras personales, varios autores como Collotta & Pau (2017) coinciden que el protocolo Bluetooth de bajo consumo tiene un potencial muy alto para convertirse en una tecnología importante en las redes eléctricas inteligentes debido a su baja potencia, bajo costo y presencia en múltiples dispositivos. En comparación con Bluetooth clásico, BLE tiene una potencia de

enlace mejorada de 3dB, permitiéndole tener un rango de cobertura de hasta 200 a 300 metros en línea directa sin la necesidad de un amplificador de potencia adicional.

Mahmood et al. (2015) resumen los atributos técnicos de las tecnologías más comunes de la siguiente forma: Wi-Fi ofrece una alta velocidad de transmisión de datos y un alcance más amplio (100 m en el interior) en comparación con las comunicaciones seguras de Bluetooth de corto alcance (10 m típicas), pero ZigBee parece ser el mejor candidato para ambientes domésticos principalmente por su bajo costo, bajo consumo de energía, alcance interior razonable (10-75 m). La baja velocidad de datos de ZigBee (40-250 kbps) es una de las razones por las que es una tecnología de baja potencia. ZigBee soporta un gran número de nodos (más de 64.000) lo que hace que las comunicaciones sean escalables cuando nuevos nodos entran en el sistema. Z-Wave y 6LoWPAN son una opción favorable para los dispositivos de baja potencia habilitados para IP.

El ancho de banda y la gama de WiMAX lo hacen apropiado para aplicaciones de redes NAN inteligentes. Por otro lado, las redes celulares preexistentes con suficiente ancho de banda, altas velocidades de transmisión de datos (más de 300Mbps para 4G LTE), amplia cobertura, menores costos de mantenimiento y una fuerte seguridad son una alternativa a considerar.

En la Tabla 1 se compara las características más importantes de los protocolos WiFi, Zigbee, Blueetooth, Bluetooth LE y 6LoWPAN esto para mostrar a detalle que ofrece cada una de estas tecnologías inalámbricas.

Tabla 1. Tecnologías inalámbricas más usadas en entornos residenciales.

	WiFi	Zigbee	Bluetooth	Bluetooth LE	6LoWPAN
Estándar	IEEE 802.11	IEEE 802.15.4,	IEEE 802.15.1	IEEE 802.15.1	IETF RFC 4944; IEEE 802.15.4
Velocidad de datos	100 Mbps	250 kbps (2.4 GHz); 40 kbps (915 MHz)	0.7 -3 Mbps	1 Mbps	250 kbps, 2.4 GHz; 40 kbps, 915 MHz; 20 kbps, 868 MHz;
Alcance	Hasta 100 m	Hasta 100 m y el Pro hasta 1000 m	Hasta 100 m	Hasta 100 m	Hasta 200 m
Potencia	Alto	Muy bajo	Baja	Muy baja	Muy baja
Topología	Estrella	Estrella, malla, árbol	Pico redes	Estrella y punto a punto.	Estrella y malla
Adopción del mercado	Extremadamente alto	Alto	Alto	Alto	Alto

Fuente: Elaboración propia.

2.3 Inteligencia computacional

Las técnicas de inteligencia artificial están contribuyendo a las redes eléctricas inteligentes, entre las cuales se encuentran la computación evolutiva, la lógica difusa o las redes neuronales artificiales. Tal como lo comenta Santofimia R, Toro, y López (2011) sus objetivos se logran mediante procesos estocásticos fundamentados en ciclos iterativo de generación-evaluación. De acuerdo a De Vito (2006):

> "La lógica difusa como una metodología que proporciona una manera simple y elegante de obtener una conclusión a partir de información de entrada vaga, ambigua, imprecisa, con ruido o incompleta. En general la lógica difusa imita como una persona toma decisiones basado en el conocimiento anterior. Dentro de las ventajas de la lógica difusa es la posibilidad de implementar sistemas tanto en hardware como en software o en una combinación de ambos. (p. 129)".

Un ejemplo de implementación de lógica difusa para optimizar el consumo de energía en hogares es un sistema desarrollado por (Shahgoshtasbi y Jamshidi, 2014). El sistema consta de dos partes: un subsistema difuso y una tabla de búsqueda inteligente. El subsistema difuso se basa en sus reglas difusas y entradas que producen la salida adecuada para la tabla de búsqueda inteligente. La segunda parte, cuyo núcleo es un nuevo modelo de una red neural asociativa, es capaz de asignar entradas a las salidas deseadas. La tabla de búsqueda inteligente tiene tres tipos de entradas que provienen del subsistema difuso, sensores externos y salidas de retroalimentación. Este sistema propuesto por (Shahgoshtasbi & Jamshidi, 2014) es capaz de encontrar el mejor escenario de eficiencia energética en diferentes situaciones.

En la Figura 1 se resumen las técnicas utilizadas en las redes eléctricas inteligentes, si bien es cierto que existen más algoritmos y metodologías se exponen las que están más alineadas a la idea principal de este proyecto.

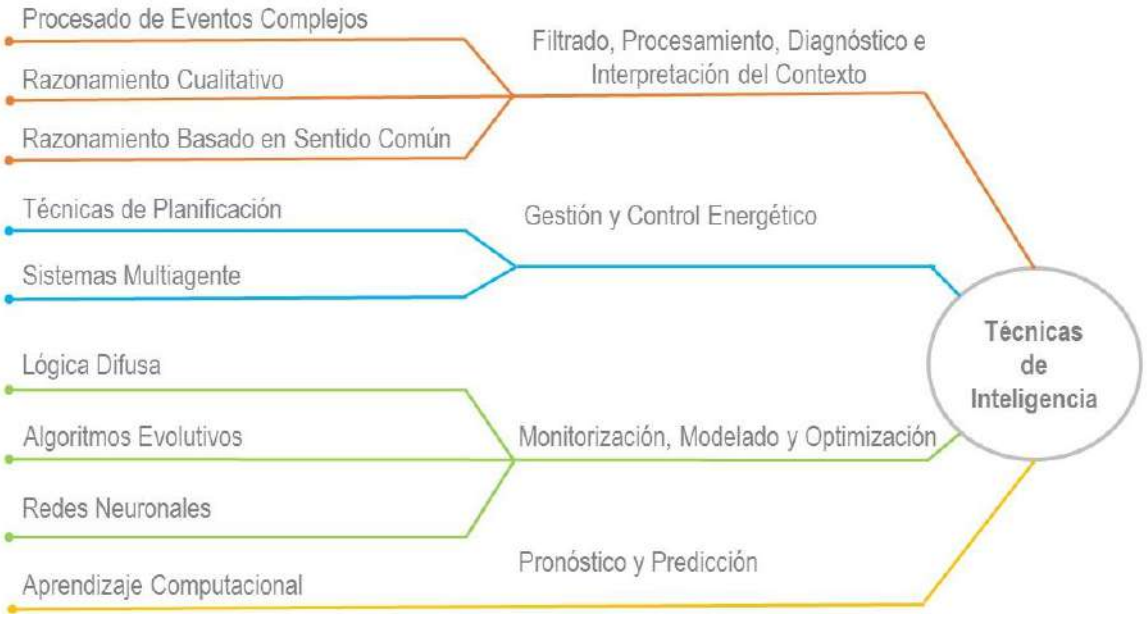


Figura 1. Técnicas de inteligencia aplicadas en las redes eléctricas inteligentes.
Fuente: Elaboración propia basado en Santofimia R, Toro, y López (2011).

En resumen, los trabajos previos cubren atributos específicos de las redes eléctricas inteligentes tales como protocolos de comunicación, inteligencia computacional y arquitecturas de software.

3. METODOLOGÍA

La metodología usada es un modelo de investigación científica basado en el diseño. Tal como los establece Hevner et al. (2004) en su primera directriz la investigación basada en el diseño debe producir un artefacto viable en alguna de sus representaciones tal como un constructo, modelo, método o instanciación.

En el desarrollo de este proyecto de investigación se apoyará en cada una de las cinco etapas propuestas por Kuechler y Vaishnavi (2011) para sistematizar y tener un marco de trabajo que propicie la generación de un artefacto tecnológico que resuelva las necesidades previamente identificadas.

1. Identificación del problema: En este primer paso se identificó un problema en donde el diseño y la creación de un artefacto podría conducir a una posible solución. En esta fase se realizó la propuesta de investigación.
2. Diseño (Sugerencia): una vez identificado el problema se sugirió una solución recurriendo a un estudio de las soluciones ya existentes y relevantes. Una vez hecho el estudio del arte se propuso un diseño.
3. Desarrollo: Durante esta fase se realizó la implementación de un prototipo que permitiera demostrar que el diseño propuesto puede cubrir las problemáticas previamente identificadas en la fase 1.
4. Evaluación: En la fase de evaluación se realizaron tanto las pruebas de funcionamiento como de robustez. Además, durante esta etapa se determina si el análisis confirma o contradice la hipótesis planteada.
5. Conclusión: En esta etapa se sintetizaron los resultados obtenidos.

En este modelo de Vaishnavi y Kuechler (2011) se considera que tanto en la fase de desarrollo y evaluación se puede generar nueva información que conduzca a la repetición del proceso, por lo que existe la alternativa de tener varias iteraciones antes de alcanzar los objetivos, a los cuales se le conoce como proceso de circunscripción. Lo anterior lo podemos resumir con lo representado en la Figura 2,

en donde se indican el flujo de cada una de las etapas que conforman la metodología, así como sus respectivas salidas.

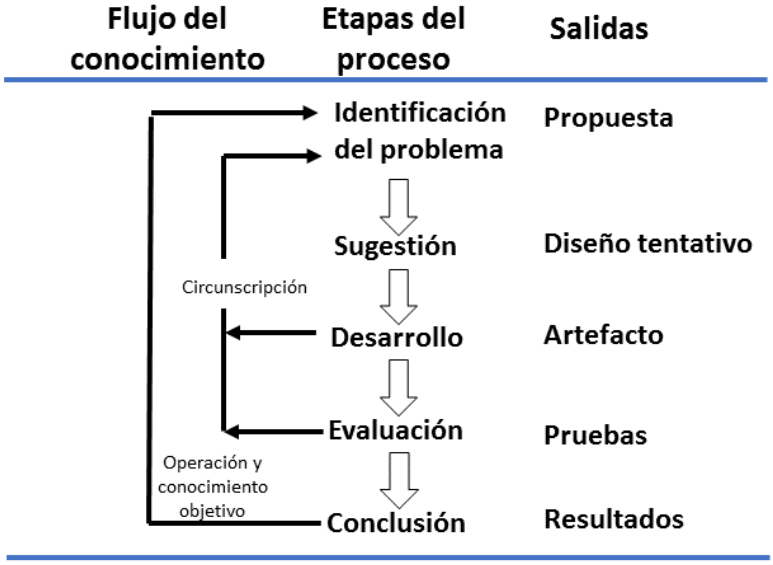


Figura 2. Metodología general de la investigación basada en diseño.
Fuente: Vaishnavi y Kuechler (2011).

3.1 Diseño

En este apartado se lleva a cabo el diseño que se va a seguir para cumplir con los objetivos planteados. Se incluyen los requisitos del sistema, el diseño del hardware, software, se describen los elementos e interacciones entre los elementos de la arquitectura para poder definir una propuesta de solución. Esta etapa se desarrolla de forma iterativa a través de varias versiones.

Antes de pasar a los detalles del diseño se exponen las bases teóricas para poder realizar el diseño de la arquitectura o artefacto. Para iniciar se definen los siguientes conceptos que son importantes para el diseño de la solución:

- **Arquitectura de software:** La arquitectura de un software es la estructura o estructuras del sistema que involucran componentes de software, sus

propiedades externamente visibles (comportamiento) y las relaciones entre ellos (Bass, 1998).

- **Puerta de enlace:** Una combinación de componentes de hardware y software que conectan una red con otra.

- **Sensores:** Un dispositivo que genera una señal de una condición física o un evento.

- **Protocolo de comunicación:** Un conjunto de reglas que proporcionan un lenguaje común para que los dispositivos se comuniquen. Se utilizan diferentes protocolos de comunicación para la comunicación dispositivo a dispositivo. Ampliamente, varían en el formato en el que se transfieren los paquetes de datos.

- **Tecnologías de comunicación:** Corresponden a todos aquellos medios físicos por los cuales es posible transferir datos o información, que permiten la comunicación entre diferentes dispositivos y/o personas.

- **AES** (Advanced Encryption Standar): Es un estándar de cifrado abierto establecido por primera vez por el Instituto Nacional de Estándares y Tecnología de los Estados Unidos (NIST, por sus siglas en inglés) en 2001. AES usa criptografía simétrica, lo que significa que la misma clave se utiliza para cifrar y descifrar los datos

- **TLS:** Se refiere a la seguridad en la capa de transporte (Transport Layer Security), es el sucesor de SSL este protocolo encapsula HTTP y SMTP para proporcionar navegación web segura a través de HTPPS. TLS usa criptografía asimétrica para autenticar a la otra parte.

- **API REST:** Significa “Representational State Transfer”, traducido al español como transferencia de estado representacional, es un patrón de diseño de software diseñado típicamente para aplicaciones web. REST pretende tratar los objetos en el lado del servidor como recursos que pueden crearse o destruirse.

3.1.1 Requisitos funcionales

Para el desarrollo de esta arquitectura de software se considera que cumplan con los atributos de seguridad, interoperabilidad, escalabilidad y mantenibilidad de acuerdo a lo señalado en la Figura 3.

Figura 3. Requisitos de la arquitectura SIEM.
Fuente: Elaboración propia (2018).

- **Seguridad externa:** Es la medida de la habilidad del sistema para resistir a intentos de uso no autorizados y negación del servicio, mientras se sirve a usuarios legítimos (Kazman et al., 2001).

- **Interoperabilidad:** Es la medida en que trabajan correctamente componentes del sistema que fueron desarrollados separadamente al ser integrados (Bass et al. 1998). Este atributo permitirá al sistema ser instalado en una variedad de escenarios, se requiere que la parte de la puerta de enlace sea compatible con distintas tecnologías de comunicación, base de datos y visualizadores en línea.
- **Escalabilidad:** Es el grado con el que se pueden ampliar el diseño arquitectónico, de datos o procedimental (Pressman, 2002).
- **Mantenibilidad:** Capacidad de modificar el sistema de manera rápida y a bajo costo (Bosch et al. 1999) Dentro de este atributo la arquitectura está diseñada para que el sistema tenga un bajo costo económico, con la finalidad de que se fácil la implementación en cualquier ambiente residencial, además que en términos de abstracción sea fácilmente interpretable por cualquier ingeniero.

- Capacidad para monitorear en línea el consumo de energía eléctrica en un ambiente residencial.
- Proveer al usuario de un sistema de notificaciones.
- Proteger el acceso a la aplicación mediante la creación de un usuario y contraseña.

3.1.2 Diseño del Sistema

En la Figura 4 se indican los elementos de la arquitectura a un alto nivel, básicamente (1) el sistema propuesto tiene lugar en un ambiente residencial donde están presentes distintos aparatos electrodomésticos que contienen varias tecnologías de comunicación, (2) a través de un circuito sensor recolectan los datos de la corriente consumida y mediante una puerta de enlace se transmiten estos datos provenientes de distintas fuentes a la nube. (3) Una vez colectados los datos estos se almacenan en una base de datos, y se grafican los datos de consumo de cada aparato electrodoméstico. (4) Finalmente el usuario puede utilizar una interface web y/o móvil para poder consultar sus consumos de energía eléctrica.

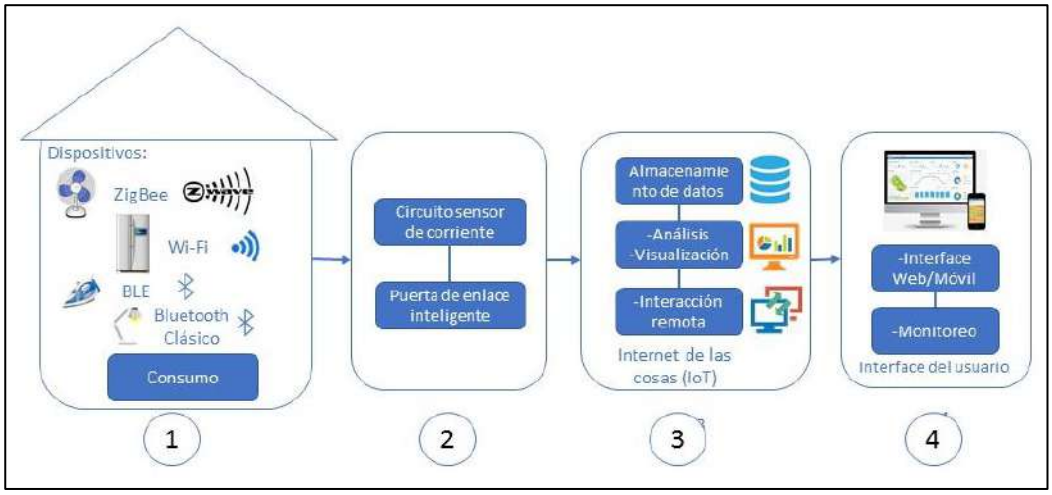


Figura 4. Arquitectura del sistema a alto nivel.
Fuente: Elaboración propia (2018).

En la Figura 5 se muestra el esquema de bloques que contempla el sistema para cumplir con cada uno de los requisitos expuestos previamente. Los cuadros en negro son los elementos de hardware del sistema propuesto y los de azul corresponden a la parte de software. Remarcar que, aunque la Raspberry Pi está catalogada como un elemento de hardware, en esta reside gran parte del software que se ejecuta. Por otro lado, las líneas negras indican una señal analógica (corriente eléctrica) y las líneas punteadas en azul se refieren a una señal discretizada (bits).

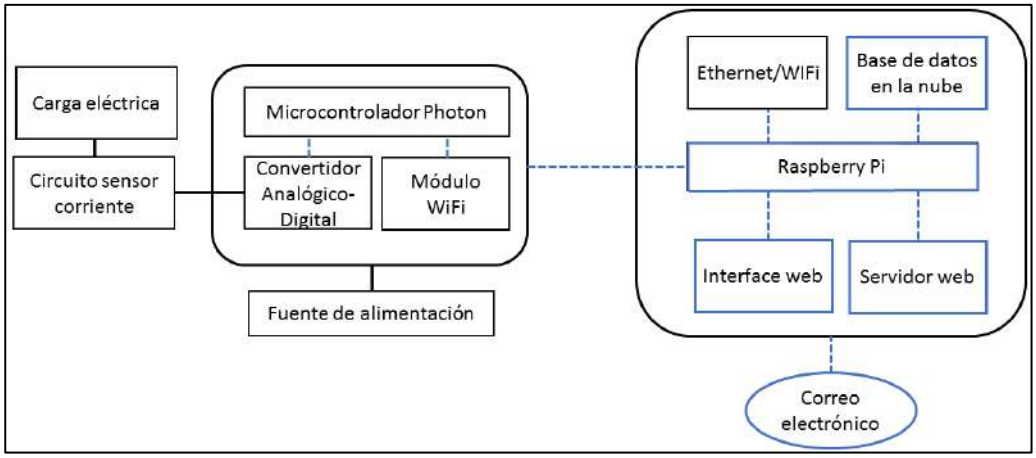


Figura 5. Diagrama de bloques propuesto.
Fuente: Elaboración propia (2018).

3.1.3 Diseño del Hardware

En esta sección se analiza el hardware seleccionado para la construcción del prototipo. Todo el sistema está basado tanto en hardware como software de código abierto esto para cumplir con el atributo de mantenibilidad del sistema propuesto. En la Figura 6 se resumen las principales plataformas de hardware enfocadas al Internet de las cosas disponibles en el mercado, después de hacer un análisis y evaluar el costo, estabilidad, facilidad de integración y enfoque al Internet de las cosas se optó por usar una tarjeta Raspberry Pi 3 y un microcontrolador llamado Photon.

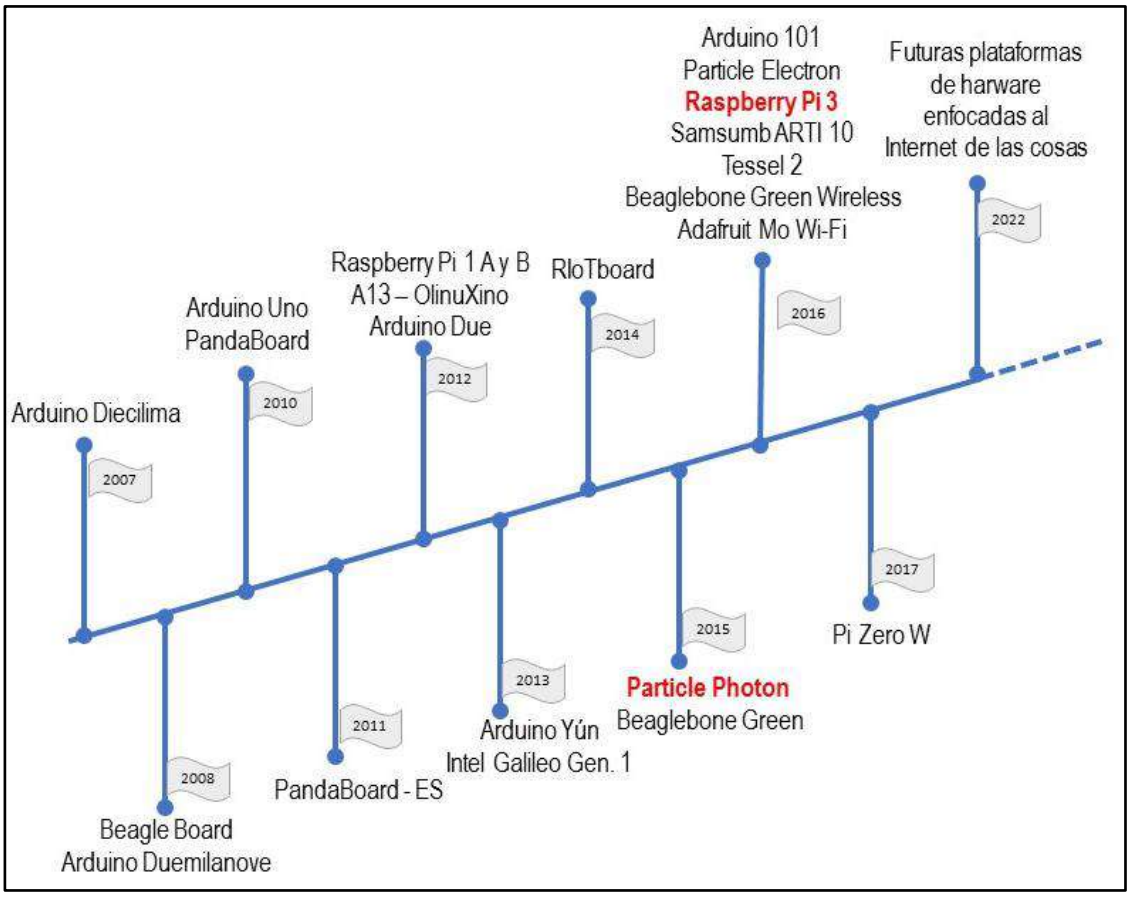


Figura 6. Cronología de la evolución de las plataformas de hardware enfocadas al Internet de las cosas.
Fuente: Hassan, Q. F., & Madani, S. A. (2017). *Internet of Things: Challenges, Advances, and Applications*. Chapman and Hall/CRC

Para el diseño electrónico se requieren tres elementos fundamentales, una Raspberry pi, un módulo de desarrollo llamada Particle Photon y el sensor de corriente que mide la corriente eléctrica que demanda el conductor.

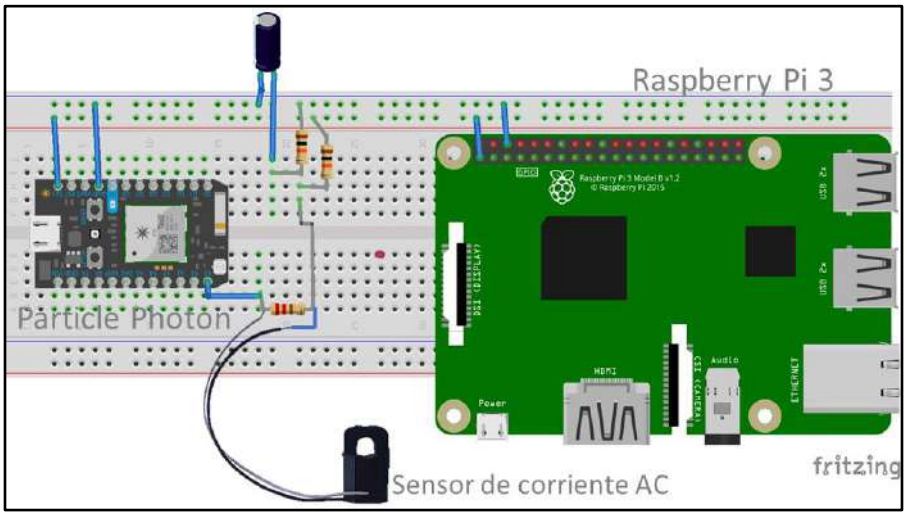


Figura 7. Diseño electrónico de la arquitectura.
Fuente: Elaboración propia (2018).

3.1.3.1 Sensor de corriente alterna:

Para obtener el valor de corriente de la red se ha optado utilizar un sensor de núcleo dividido, este tipo de sensores están basados en el principio de inductancia de tal forma que el circuito debe de pasar por lo menos una vez a través del sensor. Mediante la lectura de la cantidad de corriente generada por la bobina, se puede calcular la corriente que circula a través del conductor. El sensor seleccionado es del tipo no invasivo lo cual permitirá una versatilidad a la hora de integrar el sistema.

Figura 8. Sensor de corriente no invasivo SCT-013-030.

Este tipo de sensores son muy prácticos para acoplar a la infraestructura de la red eléctrica ya que disponen de una pinza para abrazar el cable por el cual se va obtener el valor de la corriente, entre este tipo de sensores existen modelos que pueden leer valores que van desde 0 A hasta 100 Amperios y producen una salida en su mayoría de tensión de 0-1 V. Cómo se puede apreciar en la Figura 8 cuentan con un conector Jack de 3.5 mm el cual permite adaptarse muy fácilmente ya sea con un conector tipo hembra o a una placa de pruebas. Entre los principales datos técnicos a considerar para el uso de este modelo de sensor se especifica en la Tabla 2.

Tabla 2. Datos técnicos del sensor de corriente no invasivo STC-013-03.

Especificación	Valor
Relación de giro	Np:Ns= 1:1800
Corriente de entrada	0-30 V
Salida de voltaje	0-1 V
Resistencia interna de carga	62 Ω
Linealidad	+/- 1%
Rango de temperatura operacional	-25~+70 °C

Fuente: Elaboración propia (2018).

3.1.3.2 Particle Photon

El Photon (Figura 9) es una diminuta tarjeta de desarrollado con wifi integrado que permite crear proyectos enfocado al Internet de las cosas. La principal característica de este pequeño módulo es que la combinación de un potente microcontrolador ARM Cortex M3 con un chip Broadcom Wi-Fi). Particle es el nombre del fabricante de este componente de hardware. Esta compañía desarrolla dispositivos y software que proporcionen conectividad, Wi-Fi y redes de datos móviles en un espacio optimizado.

Este módulo puede ser conectado fácilmente a la nube, basta con realizar una solicitud a la plataforma “Particle Cloud”. Además, el firmware del dispositivo es de código abierto y está escrito en lenguaje C. Finalmente resaltar el modo de bajo consumo de energía en el que puede ser configurado este módulo Photon es una

característica importante ya que está alineada al atributo de mantenibilidad de la arquitectura propuesta. Las especificaciones técnicas se muestran en la

Tabla 3.

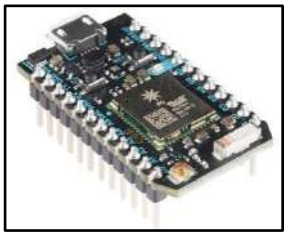


Figura 9. Particle Photon

Tabla 3. Datos técnicos del módulo Particle Photon.

Especificación	Valor
Microcontrolador	ARM Cortex M3 120Mh
Sistema operativo	FreeRTOS
Memoria	1 MB flash, 128 KB RAM
Chip Broadcom Wi-Fi	BCM43362
Wi-Fi	802.11b / g / n
Voltaje	3.3VDC
GPIO digitales	8 pines GPIO digitales (D0 a D7)
Pines analógicos	6 analógicos (A0 a A5)

Fuente: Elaboración propia (2018).

3.1.3.3 Raspberry Pi

La tarjeta de desarrollo utilizada para este proyecto es una Raspberry Pi 3 Modelo B el cual es una pequeña computadora que tiene la capacidad de procesar la misma información que cualquier computadora comercial. La Raspberry cuenta con los componentes esenciales de un ordenador como los son el procesador, memoria RAM, conexiones de entrada y salida en las que se puede conectar un teclado, monitor o ratón. Una de las ventajas que permite esta tarjeta de desarrollo es su capacidad para instalar una variedad de sistema operativos de código abierto, en este trabajo de investigación se optó por instalar el sistema operativo Raspbian

el cual es distribuido por Linux. Las especificaciones técnicas se muestran en la Tabla 4.

Figura 10. Raspberry Pi 3 Modelo B

Tabla 4. Datos técnicos de la Raspberry Pi Modelo B.

Especificación	Valor
Microprocesador	Quad Core 1.2GHz Broadcom BCM2837 64bit
Memoria	1GB RAM
Puertos USB	4 puertos USB
Conectividad	BCM43438 wireless, LAN y BLE
GPIO digitales	40 pines GPIO digitales
Voltaje de alimentación	5V, 2.5 A los recomendados

Fuente: Elaboración propia (2018).

3.1.4 Diseño del Software

En esta sección se describirá el software utilizado para el desarrollo de este proyecto. Los principales elementos de software de la arquitectura es una plataforma llamada **Particle Cloud** que permita la recolección de datos a través del microcontrolador Photon, además tiene la capacidad de procesar y transmitir datos a otras aplicaciones web. Se ha optado por utilizar una herramienta de programación visual (**Node-RED**) con un enfoque al internet de las cosas para poder gestionar el flujo del sistema. La base de datos es utilizada para almacenar los datos de corriente eléctrica y potencia de cada dispositivo y la interfaz gráfica permite al usuario ver los consumos de sus aparatos electrodomésticos. En la Figura 11 se muestran estos elementos, así como sus respectivas interacciones.

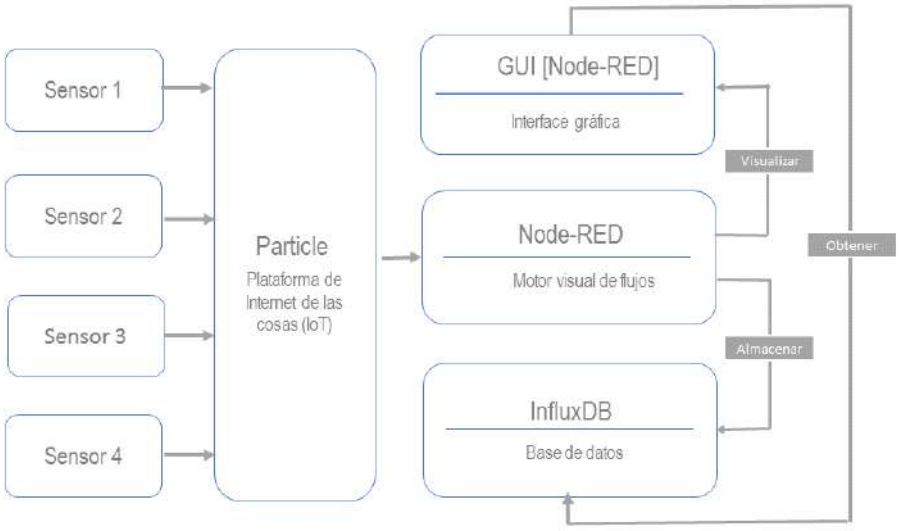


Figura 11. Interacciones y relaciones de los elementos de la arquitectura SIEM propuesta inicialmente.
Fuente: Elaboración propia basada en InfluxData (2018).

Dado que este proyecto está basado en una investigación centrada en el diseño un recurso importante dentro de la fase del diseño y evaluación son las iteraciones. Al término del primer prototipo se identificó que el sistema aún no cumplía en su totalidad con los atributos de escalabilidad por lo que se añadieron (Figura 12) ciertos elementos a la arquitectura inicialmente propuesta (Figura 11).

El principal cambio fue dar mayor apertura al sistema para que pudiera aceptar tanto más sensores como tecnologías de comunicación, además para dar mayor versatilidad para utilizar una variedad de visualizadores de acuerdo a las necesidades planteadas, ya que inicialmente se planteaba utilizar la interfaz gráfica provista por Node-RED.

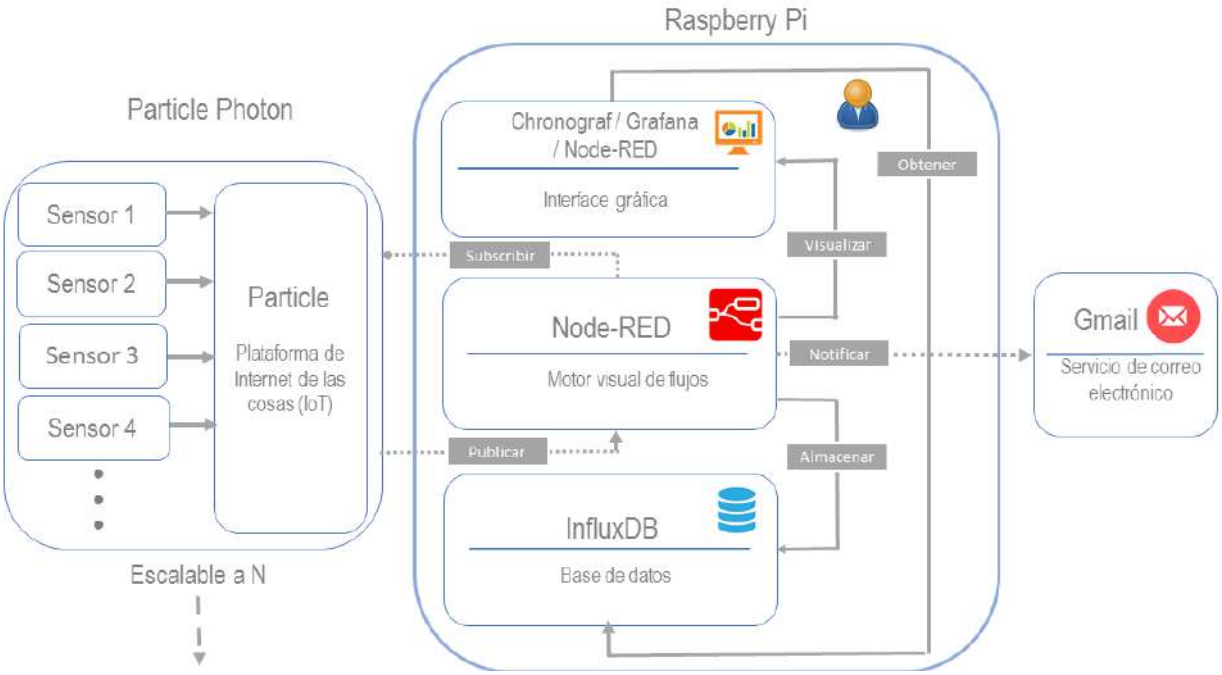


Figura 12. Arquitectura SIEM propuesta después de la primera iteración.
Fuente: Elaboración propia (2018).

3.1.4.1 Flujo del sistema

Para un mayor entendimiento del sistema se presenta el diagrama de actividades del flujo en la Figura 13, el cual se muestra el orden de las actividades ejecutadas por cada actor.

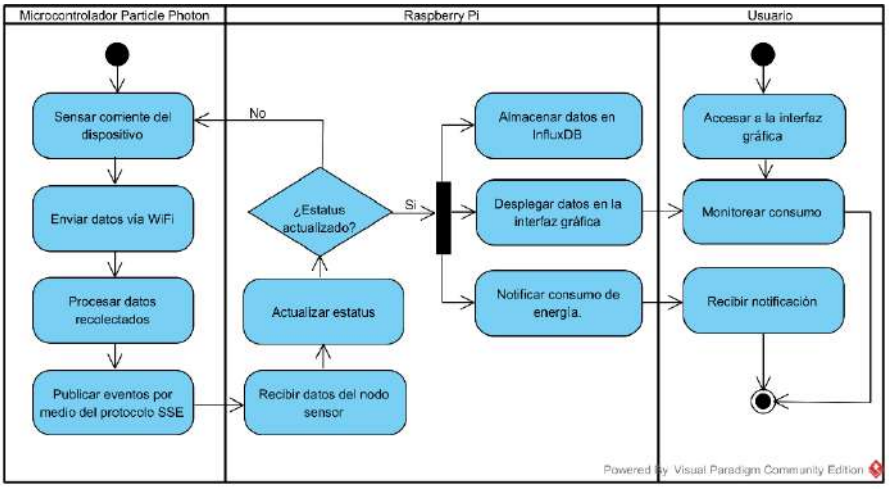


Figura 13. Diagrama de actividad del sistema propuesto.
Fuente: Elaboración propia (2018).

3.1.4.2 Obtención de datos

Uno de los objetivos de este trabajo es la consulta y monitoreo en línea de los datos obtenidos por lo que se diseñó un esquema de comunicación entre los distintos elementos de la arquitectura

Del diagrama de la Figura 14 se puede apreciar la forma que se transmiten los datos del sensor al módulo Photon mediante un cableado físico, mientras que los datos del módulo Photon a la plataforma Particle.io son transmitidos vía el protocolo wifi.

Particle Cloud es un servicio de nube gratuito que proporciona una serie de librerías para publicar eventos, funciones y variables en la nube, de manera que desde una aplicación ya sea móvil o web sea posible leer variables, llamar funciones y recibir datos mediante una API REST

Se optó por usar la plataforma **Particle.io** ya que es segura, algunas medidas que toma en cuenta es que todas las comunicaciones son encriptadas, uso de cortafuegos, detección de anomalías, balance de cargas para prevenir ataques de denegación de servicio y ataques de intermediarios

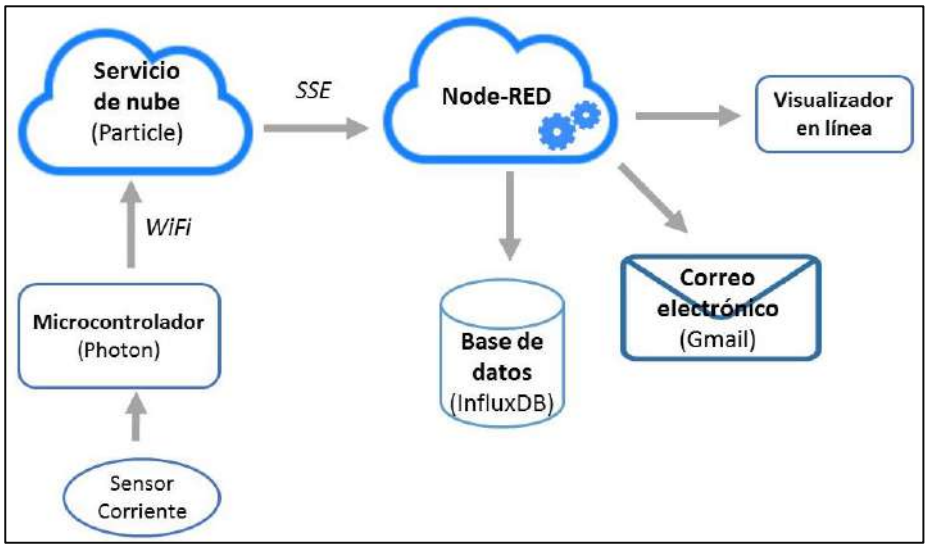


Figura 14. Diagrama del flujo de comunicación del sistema.
Fuente: Elaboración propia (2018).

Node-RED permite la conexión del dispositivo de Particle ya sea vía local o mediante la plataforma en la nube de Particle.io. El paquete de nodos de la Figura 19. Permite llamar funciones, leer variables y escuchar eventos mediante el protocolo SSE.

En el protocolo SSE (Server Sent Events) el servidor está enviando constantemente actualizaciones al cliente de una forma automatizada, una de las ventajas de utilizar este protocolo es que el flujo de los datos es unidireccional por lo que el uso del ancho de banda es menor, además que la transmisión es más rápida debido a que no se requiere esperar el regreso de algún dato. La arquitectura propuesta el servidor es la plataforma Particle.io de la cual mediante la función: Particle.publish(("Potencia", String(Potencia)) se publica el evento y la subscripción se realiza en la plataforma Node-RED.

Por otro lado, el uso de este protocolo no nos permite enviar datos en ambas direcciones esto podría ser un problema si se requiere controlar una variable con base a los datos obtenidos, pero por el enfoque de la arquitectura esta consideración no entra dentro del alcance los objetivos planteados.

Tal como se puede observar en la Figura 14 se ha planteado un protocolo de comunicación no tan complejo, cuidando el desempeño de la funcionalidad, de manera resumida el flujo del sistema está planteado para dar mayor practicidad al procesamiento de datos y a la visualización de datos del usuario final.

3.1.4.3 Seguridad del sistema

La mayor parte de soluciones actuales se enfocan en tres aspectos para cubrir la seguridad de los datos: seguridad en el hardware, seguridad en las conexiones y seguridad en la aplicación.

Figura 15. Enfoques de seguridad del sistema propuesto.
Fuente: Elaboración propia (2018).

Uno de los requisitos de la arquitectura es tener un canal seguro donde se puedan garantizar la integridad de los datos, por lo que el enfoque se centrara en la seguridad en los canales de comunicación y en la aplicación. En la Figura 16 se representa los elementos de seguridad que integra la arquitectura.

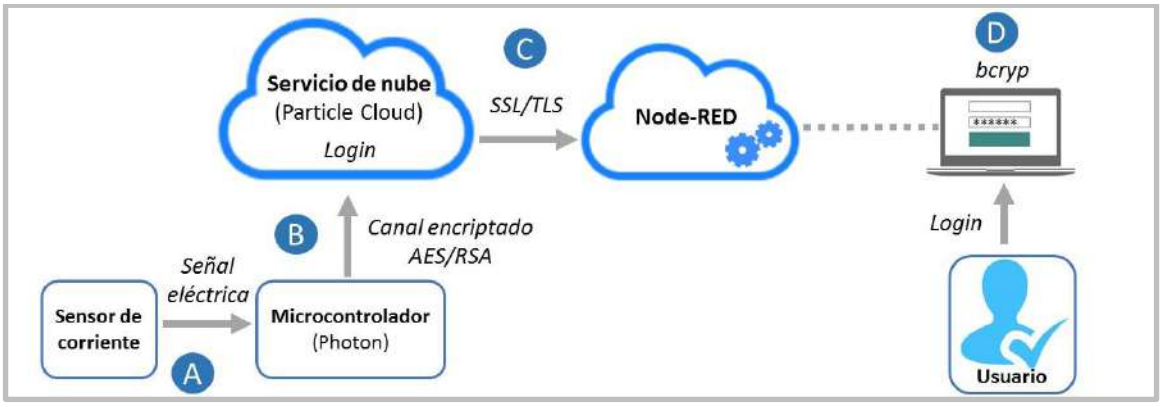


Figura 16. Elementos de seguridad de la arquitectura propuesta.
Fuente: Elaboración propia (2018).

A.-Señal eléctrica: La conexión entre el sensor de corriente y el microcontrolador se realiza mediante un cable eléctrico por lo que la única forma de obtener datos por un tercero es acceder físicamente a leer el valor. En esta sección la seguridad está garantizada a un nivel aceptable.

B.-Canal encriptado AES/RSA: Se ha optado utilizar el microcontrolador Photon ya que todas las comunicaciones hacia el servicio de nube (Particle Cloud) están encriptadas.

El cifrado comienza usando llaves RSA con un mensaje de reconocimiento (del inglés handshake). El microcontrolador Photon contiene una copia de la llave pública de la nube y su propias llaves pública y privada secreta. Particle Cloud tiene su propia llave privada secreta y contiene las claves públicas de los dispositivos con los que entabla conversación, que en el prototipo será con la tarjeta Photon. Dado que el microcontrolador Photon tiene la llave pública del servicio de nube puede asegurarse que solo Particle Cloud pudo enviar el mensaje de reconocimiento. Del mismo modo como Particle Cloud tiene la llave pública del microcontrolador, sabe que solo ese dispositivo pudo haber enviado el mensaje de reconocimiento. Una vez entablada una sesión esta expira hasta que se reinicie otra conexión o se detecte una anomalía. En Figura 17 se describe de una forma secuencial el detalle del proceso de encriptación AES/RSA, las secuencias con línea azul remarcan lo referente al cifrado AES.

C.-SSL/TLS: Como se describe en la Figura 14. Se optó por utilizar eventos enviados por servidor (SSE, por sus siglas en inglés) entre el servicio de nube llamada Particle Cloud y nuestro motor de flujos (Node-RED). Esta es una buena opción para procesar los datos enviados desde el microcontrolador Photon hasta el motor de flujos, además que permite que el servidor realice una conexión TLS/SSL encriptada y mantiene la conexión abierta.

La API de Particle Cloud es una API REST. Todas las solicitudes provienen del servidos usando la seguridad TLS. Protocolo y host: https://api.particle.io.

D.-Login: El uso de un usuario y contraseña son elementos que restringen el acceso a los sistemas, por lo cual en esta arquitectura se propone incluir estos elementos tanto para el acceso en el servicio de la nube como hacia la aplicación.

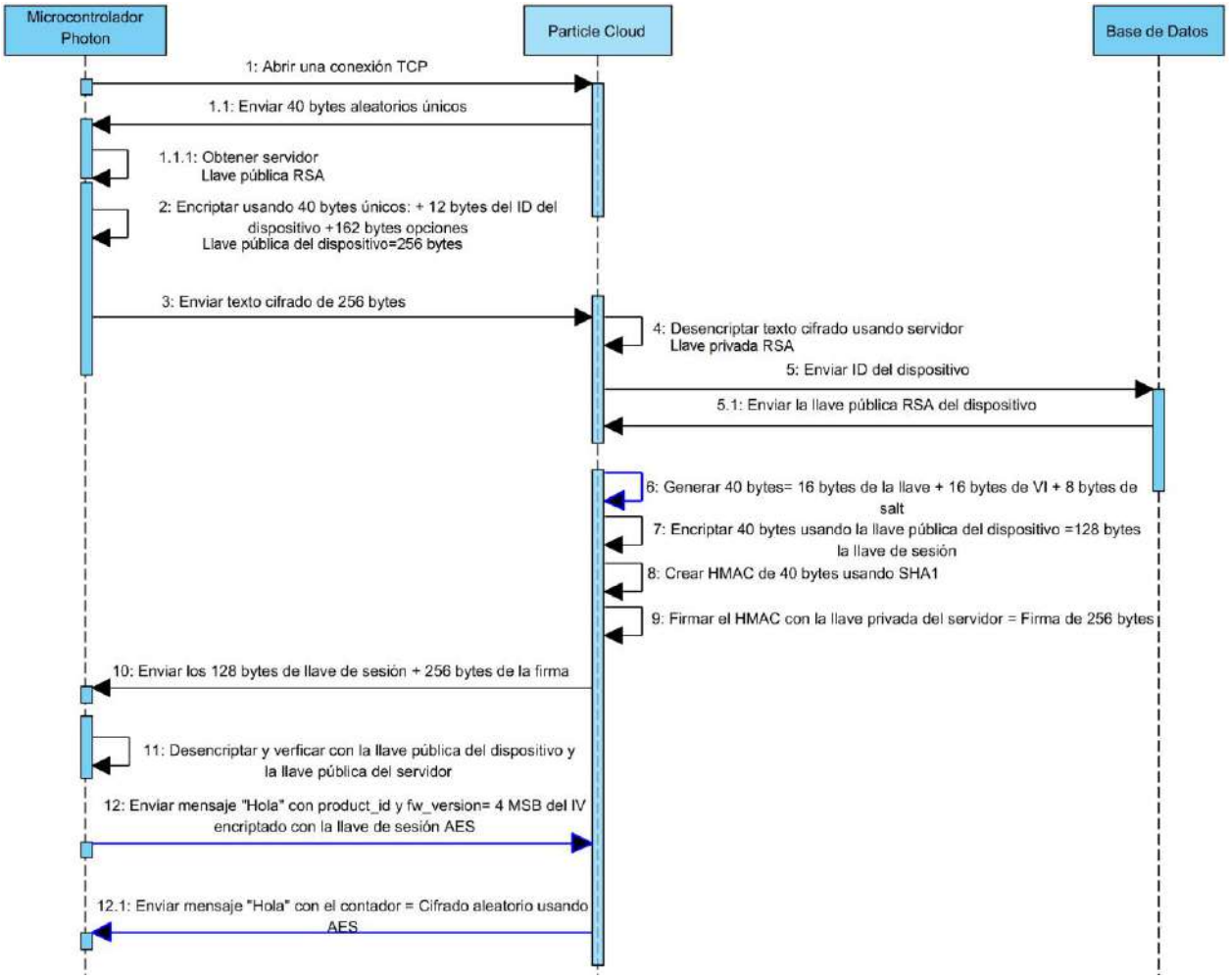


Figura 17. Diagrama de secuencias del proceso de encriptación entre Photon y Particle Cloud. Fuente: Elaboración propia (2018).

1.- Acceso al servicio de la nube: Cuando se conecta por primera vez el microcontrolador Photon a la nube esta debe asociarse a una cuenta y solo el usuario registrado tendrá permisos de controlar el dispositivo usando el token de acceso.

2. Acceso a la aplicación: Por default nuestro motor de flujos Node-RED no pide creación de un usuario y contraseña para acceder, por lo que se contempla modificar el archivo de configuración para poder agregar un usuario y una contraseña.

3.1.4.4 Manejo de datos y almacenamiento de datos

Este apartado se verá la forma de comunicación de los datos, desde la salida del sensor de corriente alterna hasta el almacenamiento en una base de datos. En concreto hablaremos de la plataforma del manejo del flujo Node-Red.

Node-RED:

Node-RED es una plataforma desarrollada por IBM, el cual permite generar prototipos rápidos debido a su capacidad parar integrar elementos de hardware, aplicaciones de software y una multivariada de APIS. Esta herramienta se maneja a través de elementos llamados nodos. En la Figura 18 podemos ver la simplicidad de la conexión entre dos nodos:

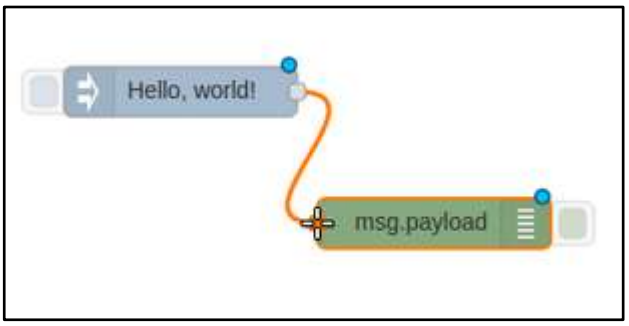


Figura 18. Conexión de dos nodos en Node-RED.
Fuente: Node-RED (2017).

Dos de los atributos de la arquitectura que se proponen son la “Interoperabilidad” y la “Escalabilidad”, Node-RED permite añadir nuevos nodos sin afectar la definición e implementación de los elementos actuales. Además de ser una plataforma de código abierto para la comunidad de desarrolladoras garantiza su mantenibilidad.

Para la implementación de la arquitectura se utilizaron varios paquetes que se encuentran fuera del paquete oficial de Node-RED, los cuales son:

- **Node-red-contrib-influxdb:** Es un paquete que permite guardar y extraer datos de una base de datos creado por la herramienta influxdb.

- **Node-red-contrib-particle:** Permite conectarse a dispositivos Particle, es usado para llamar funciones, leer variables o escuchar eventos (SSEs).

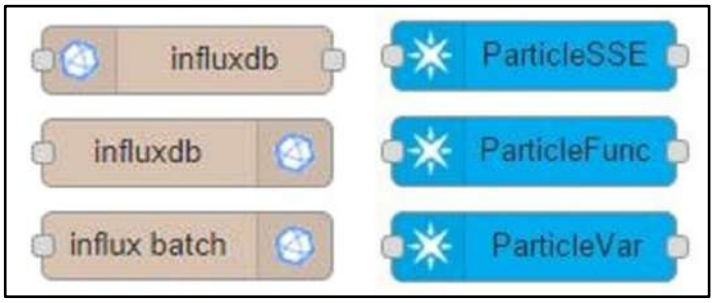


Figura 19. Nodos requeridos fuera del paquete oficial de Node-RED.
Fuente: Node-RED (2017).

InfluxDB:

Se ha designado InfluxDB como el motor de base de datos del sistema, InfluxDB es un servidor de base de datos NoSQL enfocado para gestionar series de tiempo, esta se basa en almacenar datos como puntos en el tiempo y cada punto tiene asociado una serie de atributos. Está escrito en Go y fue desarrollada por Google. Una de las grandes ventajas es una plataforma de código abierto dirigido a desarrolladores que buscan sistemas de almacenamiento para sus aplicaciones que se ejecutan en tiempo real.

3.1.4.5 Algoritmo

El objetivo del algoritmo es poder notificar al usuario de una forma inteligente y eficaz cuando el consumo excede los umbrales previamente calibrados para cada categoría de electrodomésticos y cuando tenga comportamientos anormales en base a su estado anterior. Una vez hecha la revisión de trabajo previos de investigación se propone un algoritmo basado en reglas IF-THEN y agentes reactivos basados en modelos. El alcance de este proyecto es solo el monitoreo del consumo por lo que se descartan otros enfoques como lo podría ser usar lógica difusa o algún tipo de algoritmo de aprendizaje computacional (machine learning).

En la Figura 20 se muestra de una manera visual como funciona un sistema basado en reglas. Una regla es una proposición lógica que relaciona al menos dos

o más condiciones y a partir de una unidad de procesamiento esta puede inferir una conclusión.

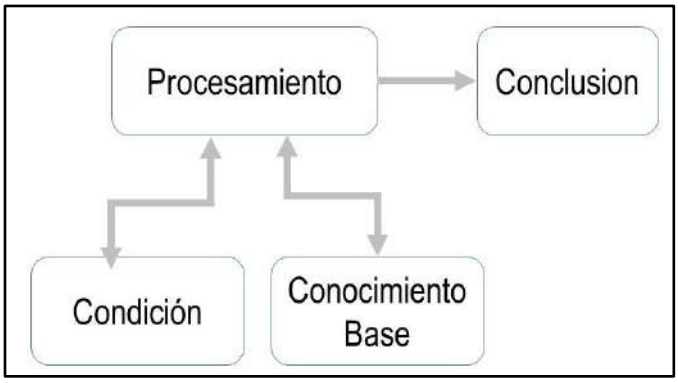


Figura 20. Esquema de un sistema basado en reglas**.**
Fuente: Elaboración propia (2018).

Este esquema funciona muy bien para tomar decisiones sin embargo existen ciertas limitaciones como una deficiente búsqueda en sistemas muy complejas y con muchas reglas y la incapacidad de aprender, como se verá más adelante estos aspectos los viene a cubrir un agente inteligente. Las reglas son relativamente fáciles de definir e interpretar, un ejemplo de pseudocódigo aplicada a nuestro sistema sería lo representado en la Figura 21.

```
#Caso de uso:
#Enviar notificación al usuario solo cuando el consumo exceda 1000 kWh
función notificar (consumo) devuelve e
      datos:
                  estado  # Descripción actual del mundo.
                  reglas  # Conjunto de reglas (IF- THEN).
   if (consumo>1000)
   {
       then
            Enviar notificación
   }
   else
   {
            Guardar registro
   }
```

Figura 21. Ejemplo de pseudocódigo de sistema basado en reglas.
Fuente: Elaboración propia (2018).

Se puede definir a un agente como un ente situado dentro de un ambiente que percibe y actúa sobre este. El agente puede estar caracterizado en distintas formas, los cuatro tipos básicos de agentes presentes en casi en todos los sistemas inteligentes son:

- Agentes reactivos simples.
- Agentes reactivos basados en modelos.
- Agentes basados en objetivos y metas.
- Agentes basados en utilidad.

El algoritmo propuesto toma conceptos, elementos e interacciones de un **agente reactivo basado en modelos.** Este tipo de agente almacena información de la parte del mundo que ha visitado (una memoria), mantiene un estado interno y un modelo del mundo según la secuencia de percepciones. En la Figura 22 se representa un agente inteligente basado en modelos.

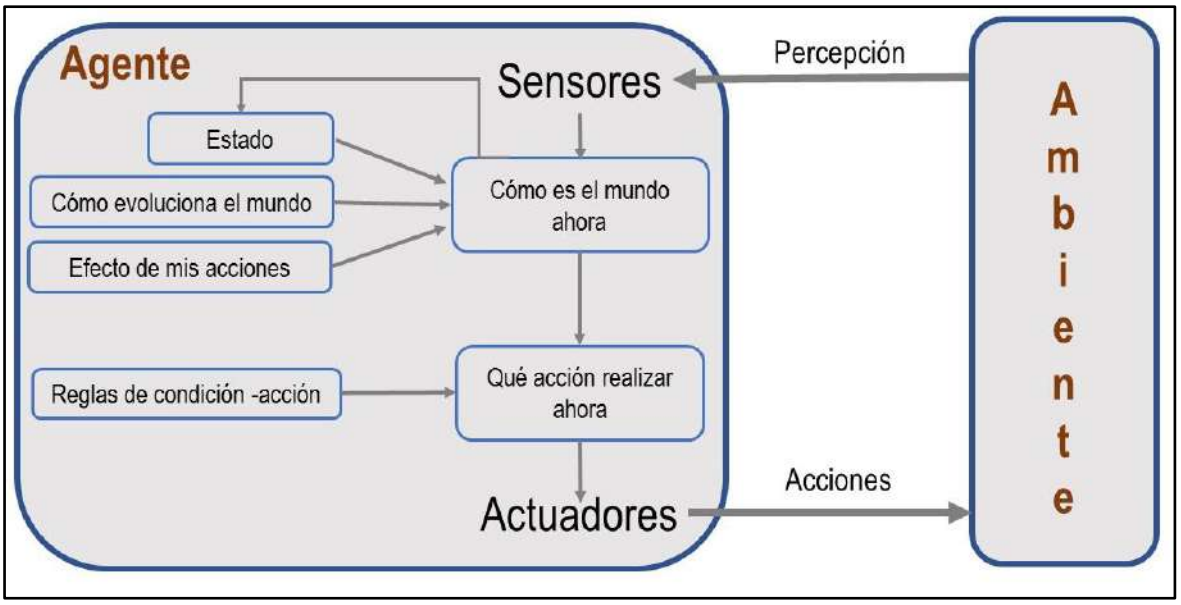


Figura 22. Esquema de un agente basado en modelos.
Fuente: Elaboración propia basada en Orozco (2018).

En términos informáticos podemos decir que un agente es una función que regresa una acción con base a una percepción, el pseudocódigo de este esquema es el mostrado en la Figura 23. De la Figura 22 se puede observar que el sistema propuesto cuenta con todos los elementos, cabe aclarar que si bien el término actuador en electrónica se le reconoce cómo un dispositivo electromecánico para controlar determinada señal en nuestro esquema se tratará de un elemento de software.

```
función AGENTE-BASADO-MODELOS (percepción) devuelve acción
    datos: estado   # Descripción actual del mundo.
      modelo    # Dado estado y acción, devuelve nuevo estado.
      reglas    # Conjunto de reglas (IF- THEN).
      acción    # Acción que realizara.
  estado = ACTUALIZAR-ESTADO (estado, acción, percepción, modelo)
  regla = reglas.buscar (estado)
  acción =regla.acción
  retornar acción.
```

Figura 23. Pseudocódigo de un agente inteligente basado en modelos.
Fuente: Elaboración propia (2018).

El enfoque de este proyecto es hacia la monitorización del consumo de electricidad, por lo que se pretende influir en el usuario para generar un cambio de hábito. Se realiza una analogía de cuando se tiene el coche lleno de gasolina el usuario gestiona que rutas, horas y frecuencia usa el auto con el fin de optimizar este combustible, parte de las notificaciones es tener un esquema similar.

Figura 24. Analogía de consumo de energía.
Fuente: Elaboración propia (2018).

Otra variable que es importante para el desarrollo del algoritmo propuesto es la tarifa. Los tipos de tarifas domésticas varían de acuerdo a la región en que se viva en México. Las diferentes tarifas domésticas que el principal proveedor de energía eléctrica en México, la Comisión Federal de Electricidad (CFE), calcula son: 1, 1A, 1B, 1C, 1D, 1E, 1F. Las letras sirven para indicar las distintas regiones en México, ya que cada una varía de acuerdo a la temperatura. En este proyecto se utilizará la tarifa 1A, la cuál es la más estándar por la CFE, la temperatura media mínima en verano es de 25 °C. En la Tabla 5 se muestra las tarifas correspondientes al año 2018 de la clasificación 1A tomando como referencia que el horario de verano comienza en abril en muchos estados del país México.

Tabla 5. Tarifa 1A del año 2018 en México.

Tipo	Básico	Intermedio	Excedente
Enero	0.793	0.956	2.802
Febrero	0.793	0.956	2.802
Marzo	0.793	0.956	2.802
Abril	0.697	0.822	2.802
Mayo	0.697	0.822	2.802
Junio	0.697	0.822	2.802
Julio	0.697	0.822	2.802
Agosto	0.697	0.822	2.802
Septiembre	0.697	0.822	2.802
Octubre	0.793	0.956	2.802
Noviembre	0.793	0.956	2.802
Diciembre	0.793	0.956	2.802

Fuente: CFE (2018).

De la Tabla 5 es necesario puntualizar que el **consumo básico** aplica para cada uno de los primeros **75** kilowatts-hora (100 kWh en horario de verano), el **consumo intermedio** para cada uno de los siguientes **75** kilowatt-hora (50 kWh en

horario de verano) y el **consumo excedente** abarca por **cada kilowatt-hora adicional** a las anteriores.

Otra condición importante que se requiere notificar al usuario es cuando el servicio alcance la categoría de alto consumo, la importancia radica en el precio de kWh se cobra más alto, en la Tabla 6 se muestra los límites fijados por la CFE correspondientes al año 2018 para las distintas tarifas del sector residencial.

Tabla 6. Tarifas y sus correspondientes límites y valores de temperatura.

Tipo	Temperatura media mensual mínima en verano	Límite para la aplicación de la tarifa DAC
Tarifa 1	<25 °C	250 kWh/mes
Tarifa 1A	25 °C	300 kWh/mes
Tarifa 1B	28 °C	400 kWh/mes
Tarifa 1C	30 °C	850 kWh/mes
Tarifa 1D	31 °C	1000 kWh/mes
Tarifa 1E	32 °C	2000 kWh/mes
Tarifa 1F	33 °C	2500 kWh/mes

Fuente: CFE (2018).

Algoritmo de configuración de los parámetros de potencia, corriente, voltaje y tarifa

Para el diseño del algoritmo se clasificaron los electrodomésticos más comunes en los hogares en tres categorías: Permanente, temporal de bajo consumo y temporal de alto consumo. Como se puede inferir las dos variables que se tomaron en cuenta para realizar esta clasificación es la frecuencia de uso y la potencia consumida. En la Tabla 7 se realizó un concentrado de los dispositivos que entrarían en cada uno de las categorías propuestas.

Tabla 7. Categorías de los electrodomésticos de una casa.

Categoría	Electrodoméstico	Potencia promedio (Watts)
Permanente	Refrigerador de 13 pies	265
	Router de WiFi	6
Temporal alto consumo	Lavadora y secadora	2000
	Tostadora	1500
	Plancha	1000
	Microondas	750
	Cafetera eléctrica	700
	Aspiradora	700
Temporal bajo consumo	Licuadora	300
	Ventilador	300
	Equipo de sonido	100
	Bombilla convencional	100
	Televisor LED	85
	Bombilla Ahorrador	20
	Laptop	20
	DVD	15
	Cargador celular	5

Fuente: Elaboración propia recopilado de PROFECO (2015).

Una vez explicados los conceptos y elementos para el desarrollo del algoritmo se presenta la secuencia de pasos, claramente las **entradas** serán: los valores de potencia, voltaje, corriente, tarifa de consumo y la **salida** será una notificación en forma de correo electrónico.

1. Seleccionar el tipo de parámetro que se desear recibir notificación, las opciones son: potencia, voltaje, corriente y kWh y tarifa de consumo.
2. Fijar el límite superior del parámetro seleccionado.
3. Obtener la primera medida existente del parámetro(s) seleccionado(s).
4. Comparar si cada uno de los parámetros del consumo obtenidos es igual o mayor al límite máximo fijado en la configuración de alerta.

5. Si el parámetro seleccionado fue kWh y tarifa de consumo, seleccionar el tipo de tarifa de la región.
6. Calibrar las tarifas del proveedor de acuerdo a la temporada
7. Obtener el consumo total del día.
8. Calcular la tarifa del consumo total.
9. Agregar el 15 del IVA a la tarifa obtenida.
10. Comparar con los límites establecidos en el paso 2, si es igual o mayor enviar una notificación al usuario.

Reglas para notificar los límites de consumo.

1. Comparar el consumo obtenido vs el límite del **consumo básico,** si es igual o mayor enviar una notificación al usuario.
2. Comparar el consumo obtenido vs el límite del **consumo intermedio,** si es igual o mayor enviar la notificación al usuario.
3. Compara el consumo obtenido vs el límite de **consumo excedente,** si es igual o mayor enviar la notificación al usuario.
4. Si rompe cualquier regla de las anteriores enviar una alerta al usuario mediante un correo electrónico.
5. Guardar registro de la notificación enviada.

Reglas para notificar los cambios del comportamiento

1. Cuando un dispositivo con la categoría de temporal de medio y alto consumo sea activado por más de una hora generar una alerta del consumo kWh y la tarifa calculado al usuario.
2. Cuando el consumo del día actual sea mayor o igual al del día anterior enviar una notificación al usuario de la diferencia de consumo.
3. Guardar registro de la notificación enviada.

En la Figura 25 se integra toda la secuencia de pasos anteriormente descritos para emitir una notificación.

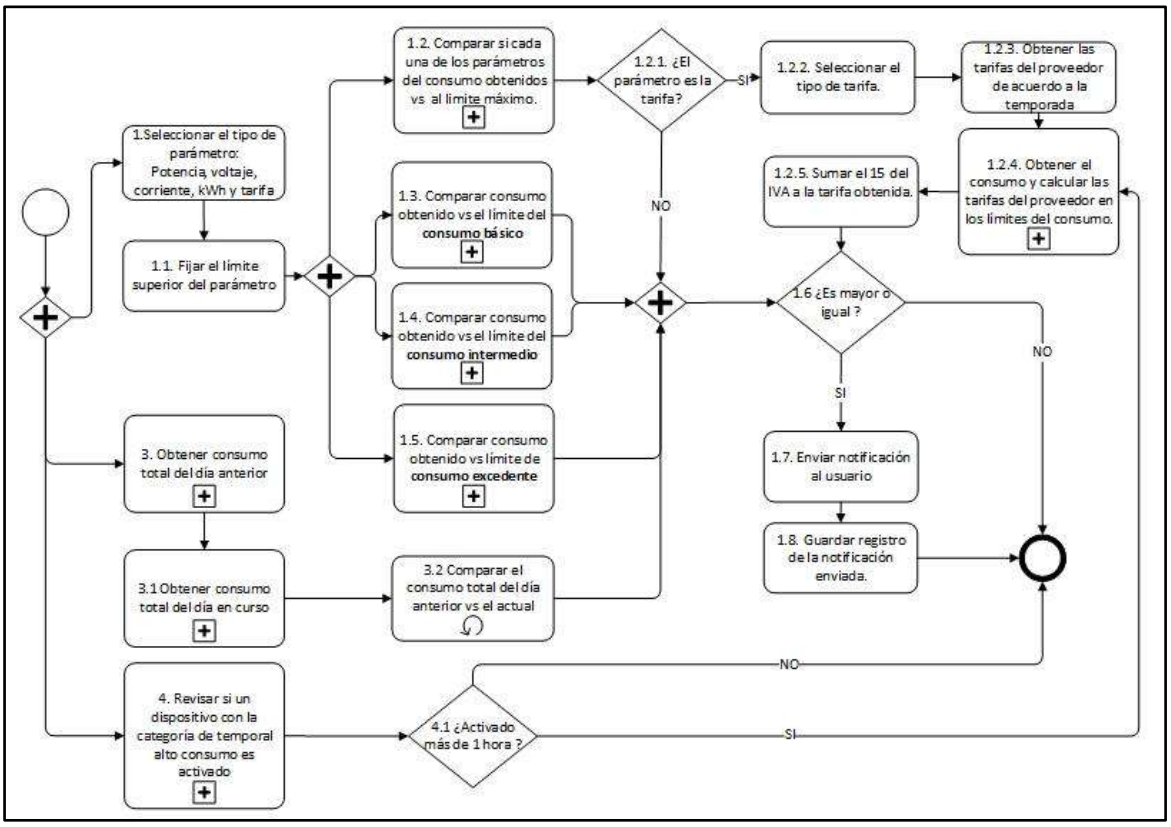


Figura 25. Diagrama de flujo del algoritmo para el sistema de notificaciones.
Fuente: Elaboración propia (2018).

3.1.4.6 Visualización y sistema de notificaciones.

Durante esta fase de diseño se ha buscado en todo momento adoptar servicios de monitorización funcionales pero que sean fáciles de usar e interpretar por parte del usuario final, por lo que para la representación de la información del consumo del aparato conectado al sistema se ha optado por utilizar gráficas de consumo en una interfaz gráfica en línea. Además, que el usuario pueda recibir una notificación vía correo electrónico cuando el consumo exceda los límites establecidos previamente en el sistema. Ambos casos de usos se resumen en la Figura 26.

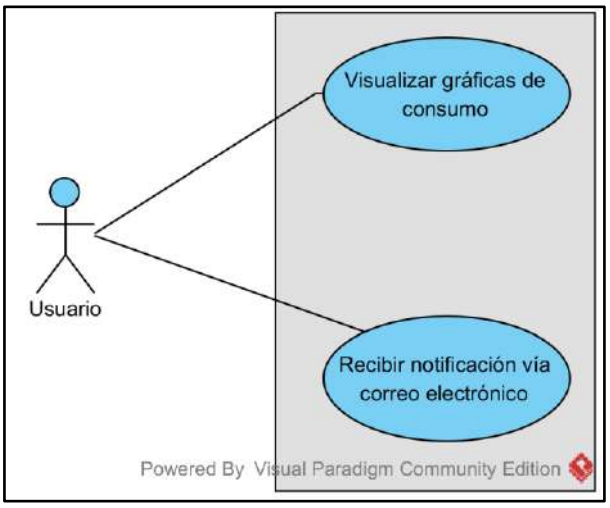


Figura 26. Diagrama de casos de uso de la plataforma web.
Fuente: Elaboración propia (2018).

La interacción con la plataforma propuesta de igual manera debe ser fácil de navegar y acceso por lo que se pretende agrupar en una misma ventana todos los elementos que conforman la monitorización previamente descritas.

En una primera iteración se identificó que la autentificación de usuarios para limitar el acceso a personas no autorizadas son elementos que se deben contemplar en el sistema propuesto sin embargo esto será considerado hasta terminar la fase de prototipo.

Grafana:

Esta herramienta es una plataforma que permite visualizar los datos colectados en las bases de datos de una forma gráfica y muy dinámica, es una herramienta con mucha capacidad para filtrar, personalizar la forma en cómo se despliegan los datos.

Grafana tiene una alta compatibilidad con InfluxDB, permite obtener datos de una gran cantidad de fuentes y se pueden configurar alertas para que envíen un correo electrónico cuando algo no se encuentre en un estado estable.

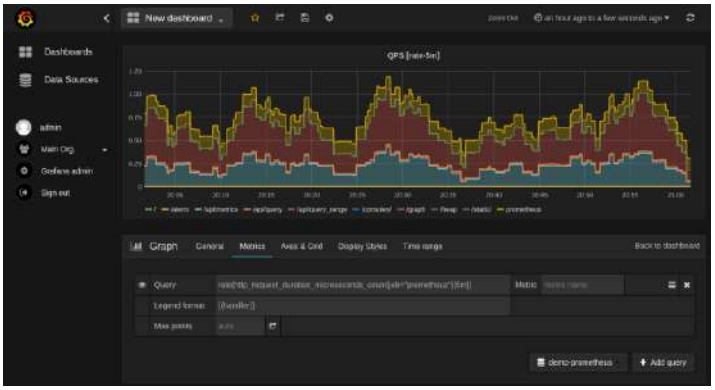


Figura 27. Dashboard de ejemplo Grafana.

Chronograf:

Esta herramienta es otra alternativa de visualizador en línea. Chronograf es al igual un proyecto de código abierto escrito en Go y React.js, en general provee elementos que permiten visualizar los datos monitoreados y crear alertar y reglas de automatización. Es fácil de usar e incluye una gran variedad de plantillas y librería para construir rápidamente tableros con monitoreo en tiempo real.

Figura 28. Dashboard de ejemplo Chronograf.

Interfaz gráfica Node-RED:

Finalmente, la interfaz gráfica incluida en la misma plataforma Node-RED es una buena opción para visualizar datos, ya que centraliza el flujo en la misma herramienta de Node-RED

Figura 29. Interfaz gráfica de ejemplo Node-RED

4. IMPLEMENTACIÓN

Con base al diseño propuesto en los apartados anteriores, se ha implementado un prototipo que se compone tanto de elementos de hardware como de software. En este apartado se expondrá como se ha construido paso a paso el prototipo de la arquitectura propuesta, incluyendo los cálculos, interfaces y programación para poder comprobar el correcto funcionamiento**.**

Cabe señalar que se optó por limitar el alcance del prototipo a solo utilizar el protocolo de comunicación wifi y se utilizó la interfaz gráfica que proporciona Node-RED en lugar de un visualizador especializado como lo es Grafana, esto para poder obtener los resultados a corto tiempo.

4.1 Nodo sensor

El principal elemento del nodo es un sensor de corriente no invasivo para este proyecto se optó por el SCT-013-030 el cual permite medir hasta una carga de 30 amperios, por lo que se tiene la capacidad para medir el consumo de la gran mayoría de dispositivos conectados en un entorno residencial.

El modo de empleo del sensor de corriente invasivo consiste en abrir la pinza amperimétrica desbloqueando su mecanismo de sujeción, se rodea **solo un cable** del dispositivo que se quiere medir, ya que si se coloca ambos cables en el centro de la pinza se anularan las mediciones y finalmente se cierra la pinza, tal como se aprecia en la Figura 30.

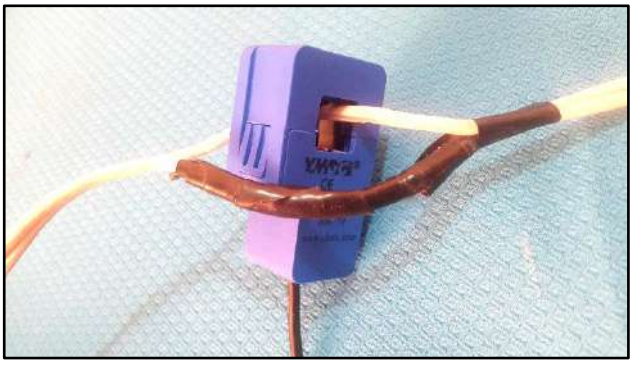


Figura 30. Modo de uso del sensor de corriente.
Fuente: Elaboración propia (2018).

Tal como se indica en la Tabla 2 de especificaciones técnicas el modelo SCT-13-030 nos da un voltaje de hasta 1 V de salida el cual es proporcional a la corriente detectada (hasta 30 A para este modelo).

Cuando la salida es de tipo corriente esta debe ser convertida a tipo voltaje por lo cual se le coloca una resistencia de carga, para este modelo no es necesario hacer los cálculos para determinar el valor de dicha resistencia dado que ya se tiene considerada la interna que es de 62 Ω, sin embargo, debido a que los modelos más populares en el mercado no contemplan esta resistencia a continuación se describe el procedimiento para obtener el cálculo:

a) Dividir la corriente pico entre el número de vueltas del sensor.

$$I_S = \frac{I_P}{Vueltas} = \frac{30\ A}{1800} = 16.667\ mA$$

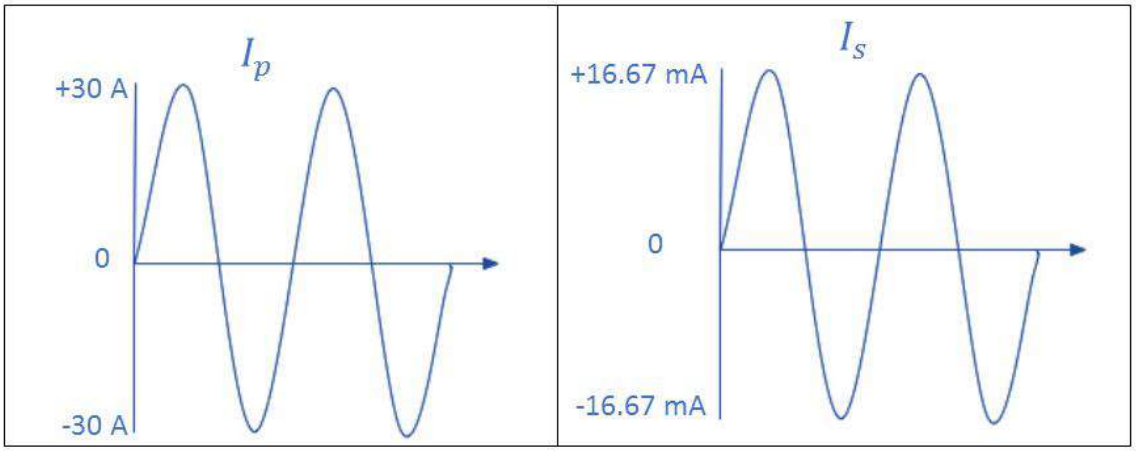


Figura 31. Señales de salida de la corriente.

b) Para comprobar el valor del voltaje de salida, multiplicamos la resistencia de cargar por la corriente de salida usando la ley de Ohm obtenemos lo siguiente:

$$V_{salida} = I_{salida} * R_{carga} = 0.0167 * 62\ \Omega \approx 1\ \text{V}$$

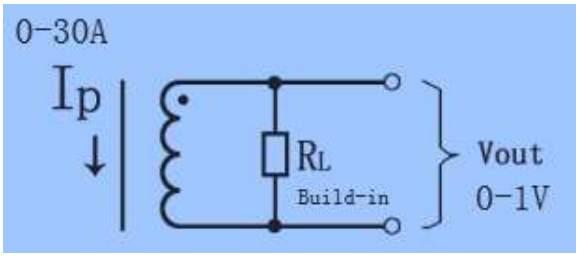


Figura 32. Esquema de la resistencia de carga.

c) Dado que el convertidor analógico digital de la tarjeta Photon no puede procesar voltajes negativos es necesario agregar un desplazamiento (offset) a la salida del sensor, para ello se hará uso de un divisor de voltaje compuesto por tres elementos electrónicos:

- 1 capacitor de 1uF.
- 2 resistencias de 10 KΩ.

En la Figura 33 se muestra la representación de divisor de voltaje en donde se utilizan dos resistencias de 10 kΩ, el voltaje de entrada es de 3.3 V el cual es el que nos proporciona el pin 1 (3.3 VDC) de la Raspberry Pi. Por lo cual efectuando los cálculos obtenemos un voltaje de salida de: 1.65 V ((3.3V x 10kΩ) / (10kΩ+10kΩ) = 1.65V).

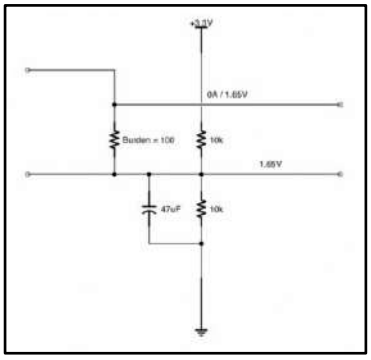


Figura 33. Divisor de voltaje.
Fuente: SparkFun (2018).

En la Figura 34 se puede ver el circuito divisor de voltaje montado en una placa de pruebas.

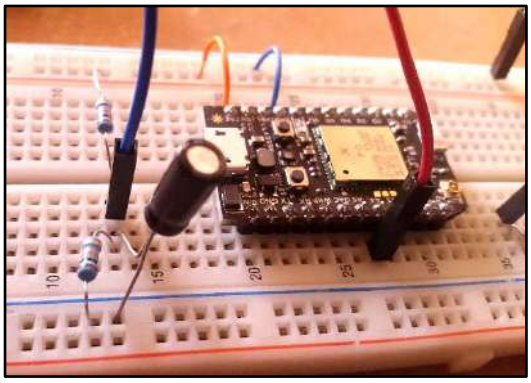


Figura 34. Circuito divisor de voltaje montado en una placa de pruebas.
Fuente: Elaboración propia (2018).

4.2 Particle Photon

Los pasos para realizar la configuración entre la tarjeta Particle Photon y la plataforma para poder realizar la recolección de datos son los siguientes:

1.- Realizar las conexiones de alimentación conforme al esquema de la Figura 35.

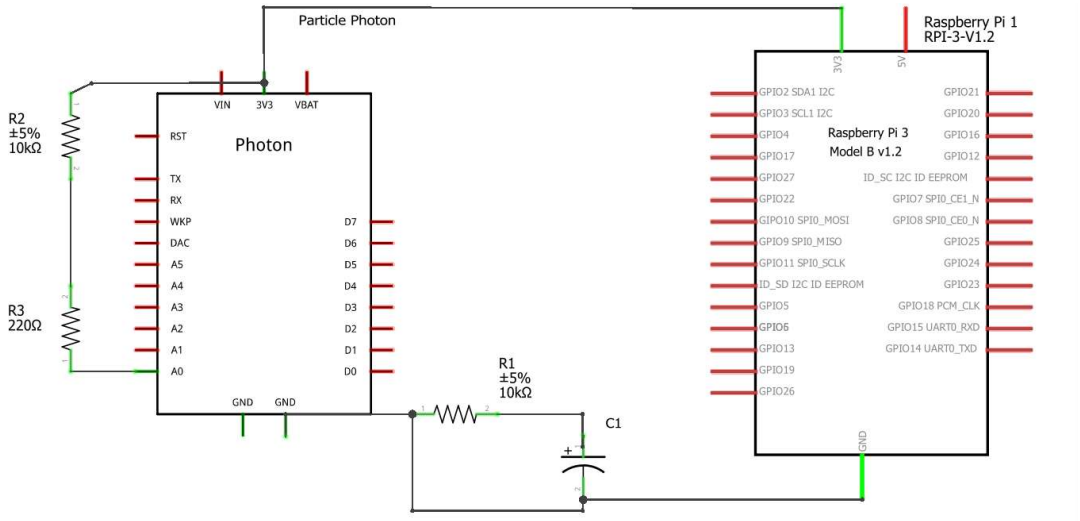


Figura 35. Esquemático de las conexiones eléctricas del módulo Photon.
Fuente: Elaboración propia (2018).

El módulo Photon posee internamente un convertidor analógico-digital de 12 bits de resolución con 8 canales (ADC0 a ADC7). Es decir, el módulo pose 8 pines de entrada que permiten leer señales analógicas de sensores o dispositivos externos tal como se puede ver en la Figura 36. El módulo Photon mapea los voltajes de entrada entre 0 a 3.3 volts a valores enteros entre 0 y 4095, lo que nos da una resolución de 0.8 mV por unidad.

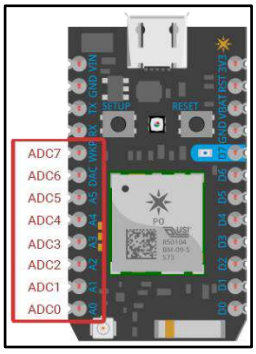


Figura 36. Pines asignados a entrada analógicas del módulo Photon.

El fabricante provee una función que permite leer de una manera muy práctica la señal analógica de un pin en específico con la siguiente función: **analogRead(pin),** sin embargo, para el proyecto se utilizó una librería provista por EmonLib: **emon1.calcIrms(número_muestras).**

2.- La forma más práctica de conectar el dispositivo Photon a la red wifi es usando la aplicación móvil “Particle”, la cual puede ser descargada desde Play Store de Google.

3.- Colocar la mini tarjeta Photon en “Modo escucha”, para ello basta con presionar el botón SETUP por tres segundos, hasta que el led empiece a parpadear en color azul.

4.- Una vez alimentado el sistema se necesita desarrollar el código que va a ser programado en el microcontrolador de la mini tarjeta Photon. El cual se utilizó el IDE web que proporciona el fabricante, para ello hacemos un previo registro antes de empezar a generar un nuevo proyecto.

Figura 37. Entorno de desarrollo integrado para el módulo Photon.

5.- El código fuente programado en el microcontrolador lee los valores recolectados para posteriormente calcular la corriente eficaz y la potencia de consumo. Finalmente se publican los valores calculados en la web. En el anexo A se indica el código utilizado en la primera iteración de la implementación.

6.- Para la segunda iteración, se encontró la librería **EmonLib** que se utiliza para obtener las lecturas de corriente eléctrica. El uso de esta librería simplifica el código además que permite mayor compatibilidad. El código fuente simplificado se presenta en el anexo B.

7.- Teniendo el código los siguientes pasos es hacer la compilación del código y posteriormente hacer la grabación en la memoria para poder ejecutar la lógica. Para ello seguimos usando la plataforma online de Particle.

Tabla 8. Secuencia de programación del microcontrolador.

	a) Guardar: el código generado
	b) Verificar: Compila el código.
	c) Flash: Programa el binario generado en la memoria del microcontrolador

Fuente: Elaboración propia (2018).

En la Figura 38 se muestran la integración completa de los elementos de hardware que componen la arquitectura propuesta.

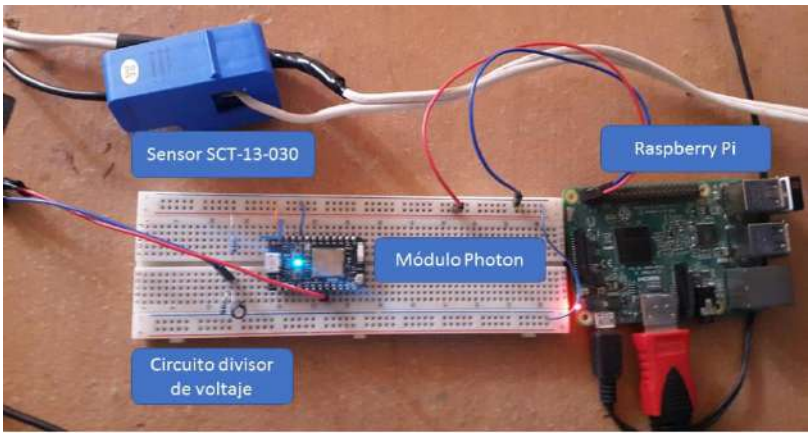


Figura 38. Integración de los elementos de hardware del sistema propuesto.
Fuente: Elaboración propia (2018).

4.3 Instalación de InfluxDB

Los datos recolectados son almacenados en la base de datos InfluxDB, para su instalación se siguieron los siguientes pasos:

1.- Descargar la última versión de InfluxDB de la página oficial: https://portal.influxdata.com/downloads

2.- Ejecutar los comandos, señalados en la Figura 39, en la consola de Raspbian:

```
pi@raspberrypi:~$ sudo apt-get update
pi@raspberrypi:~$ sudo apt-get upgrade
pi@raspberrypi:~$
wget https://dl.influxdata.com/influxdb/releases/influxdb-1.5.1_linux_armhf.tar.gz
pi@raspberrypi:~$ tar xvfz influxdb-1.5.1_linux_armhf.tar.gz
pi@raspberrypi:~$ sudo apt-get install influxdb
pi@raspberrypi:~$ sudo service influxdb start
pi@raspberrypi:~$ influx
//Comando para cambiar el formato del tiempo
> precision rfc3339
//Comando para crear una base de datos nueva
> CREATE DATABASE "p1smartmeter"
> SHOW DATABASES
name: databases
name
_internal
p1smartmeter
> USE p1smartmeter
```

Figura 39. Comandos usados en InfluxDB.
Fuente: Elaboración propia (2018).

.

4.4 Configurando Node-RED con InfluxDB

Una vez configurado Node-RED en la Raspberry Pi, se usa el menú de Palette (Figura 40) para descargar el paquete node-red-contrib-influxdb.

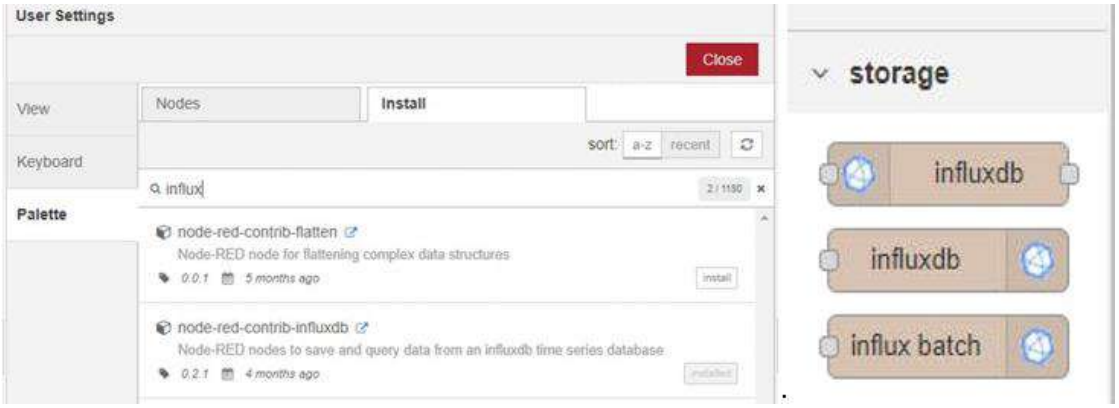


Figura 40. Paquete de nodos InfluxDB descargados en Node-RED.
Fuente: Node-RED (2018).

Adicionalmente se requiere descargar el paquete de nodos de la Figura 41 para establecer la conexión entre la plataforma Particle y Node-RED.

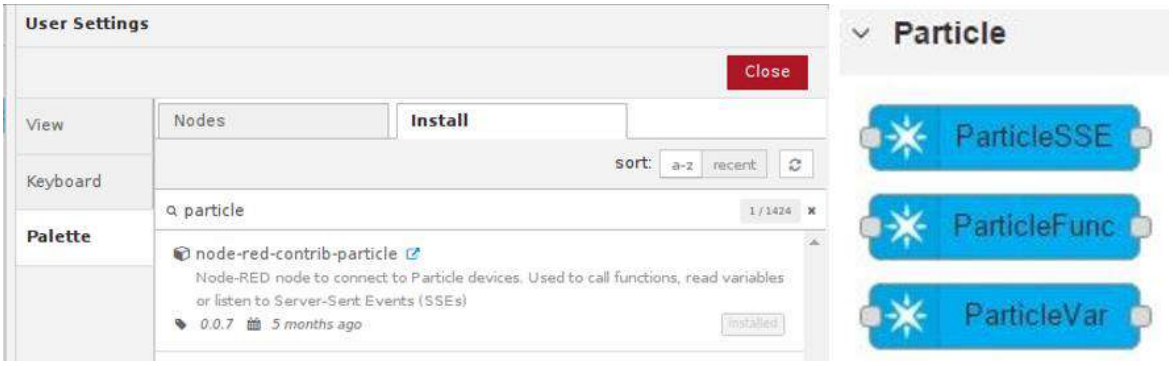


Figura 41. Paquete de nodos Particle descargados en Node-RED.
Fuente: Node-RED (2018).

Una vez agregado un nodo InfluxDB se requiere configurar la conexión con el servicio de nube Particle.io. Se configura el nodo “Particle SSE” agregando las credenciales y e token de acceso, tal como se muestra en la Figura 42.

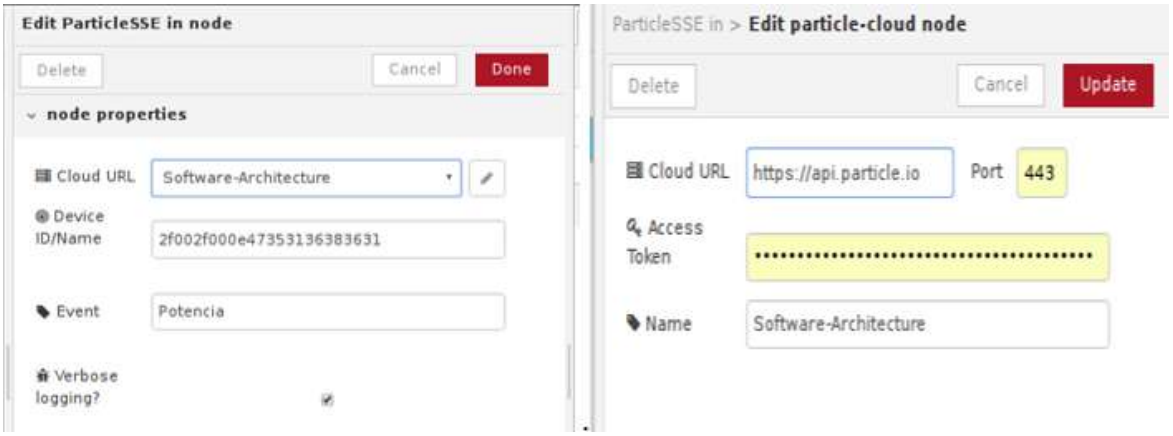


Figura 42. Configuración del nodo Particle.
Fuente: Node-RED (2018).

La configuración para establecer una correcta conexión entre la plataforma de Particle con la base de datos InfluxDB es la mostrada en la Figura 43.

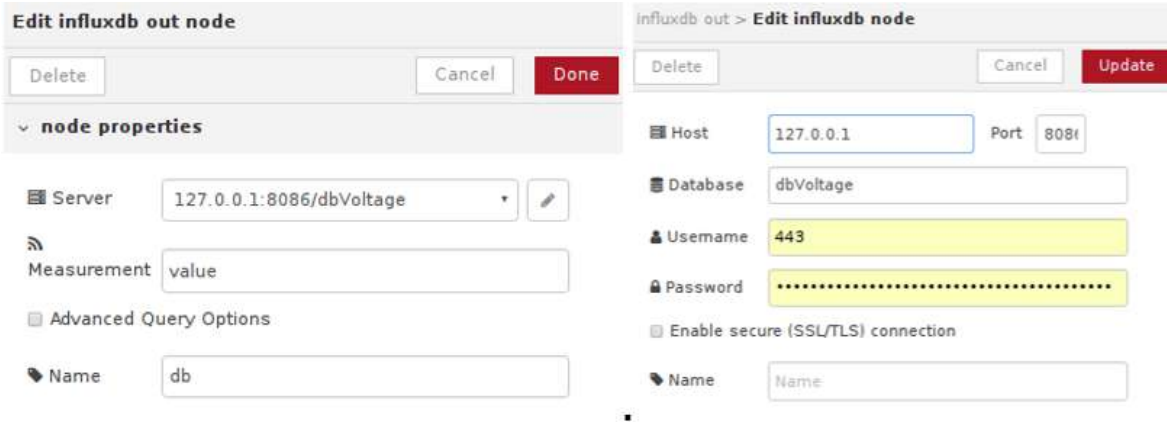


Figura 43. Configuración del nodo InfluxDB para almacenamiento.
Fuente: Node-RED (2018).

De la Figura 43 los parámetros que se tienen que considerar:

- **Host:** La dirección IP del servidor InfluxDB el cual es 127.0.0.1
- **Puerto:** Por default es el 8086.
- **Base de datos:** El nombre de la base de datos creado en InfluxDB

Una vez revisado que el sensor está enviando datos y la comunicación con Node-RED es estable se hace una consulta a la base de datos para comprobar que los registros son grabados correctamente. Para poder hacer uso del ambiente de trabajo de Node-RED ingresamos el comando de la Figura 44.

Figura 44. Consola de Raspbian.
Fuente: Elaboración propia (2018).

La programación de nodos completa está representada en la Figura 45 y la interfaz gráfica generada se muestra en la Figura 46.

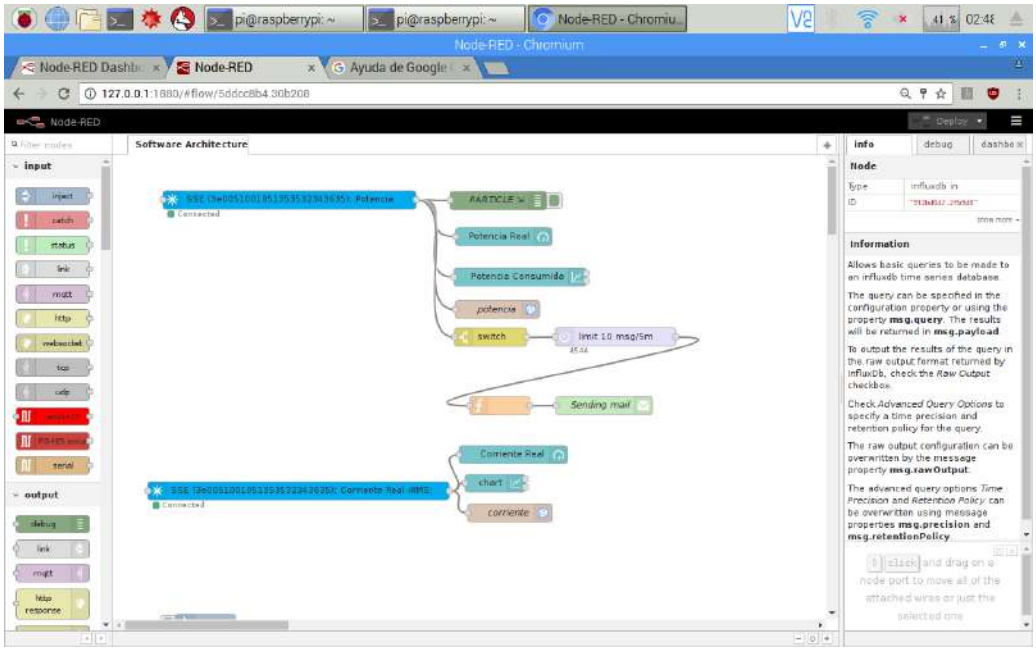


Figura 45. Flujo del sistema en Node-RED.
Fuente: Elaboración propia (2018).

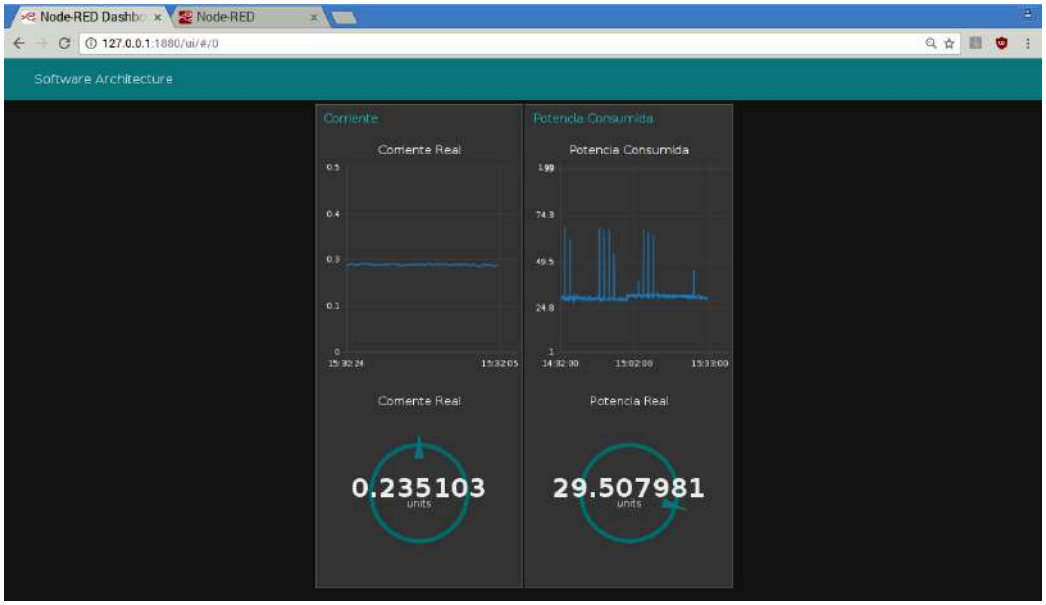


Figura 46. Interfaz gráfica del sistema propuesto.
Fuente: Elaboración propia (2018).

4.5 Configurando la seguridad en Node-RED

Para mejorar la seguridad del sistema propuesto se agregó un usuario y contraseña para proteger Node-RED. Tal como lo describe la página oficial de Node-RED por default el motor de flujos no es seguro por lo que cualquier usuario de la red quien tenga acceso a la dirección IP y si el puerto este activo, este puede acceder al editor y configurar los datos que se están recibiendo la interfaz.

1.- Para habilitar la autentificación del usuario en el Editor y la API de administración se requiere agregar en el archivo de *settings.js* lo codificado en el anexo C.

La propiedad de usuarios es una matriz de objetos de usuarios el cual permite definir distintos usuarios con sus respectivos permisos. En este prototipo se definió un único usuario llamado “admin” que tiene permiso para editar cualquier elemento del editor. La contraseña es una función resumen, también conocida como función hash, que usa el algoritmo bcryp.

2.-El siguiente paso es generar una contraseña adecuada usando el siguiente comando: node-red-admin hash-pw

3.-Aparece un aviso en el cual pedirá la nueva contraseña que se desea usar e imprimirá el hash que deberá ser agregado en el archivo de configuración.

5. PRUEBAS Y RESULTADOS

Una vez terminada la etapa de diseño e implementación del prototipo de la arquitectura propuesta se realizaron pruebas utilizando distintos escenarios para comprobar y validar el correcto funcionamiento del prototipo desarrollado. De esta forma se comprueba si se han alcanzado los objetivos planteados al inicio de este proyecto de investigación.

5.1 Pruebas de calibración

Antes de llevar a cabo las pruebas finales, se realizó una calibración para comprobar el correcto funcionamiento del prototipo. Los ajustes se hicieron por software, para la implementación del circuito medidor de corriente se utilizó la librería EmonLib, esta librería provee una función que permite hacer el cálculo de la corriente pasando como parámetros el número del pin y el factor de calibración: **emon1.current(número de pin analógico, calibración)**. Cómo se mencionó en los capítulos anteriores el sensor de corriente seleccionado es de la familia STC y el modelo utilizado para este prototipo de arquitectura fue un SCT-13-030 por lo que el factor de calibración fue 30.

Para las pruebas de calibración en un primer caso de uso se utilizó un ventilador de torre, marca DAEWOO INTERNATIONAL, Modelo: FZ10-9HB que tiene una potencia de 40 Watts de acuerdo a su etiqueta de especificación técnica.

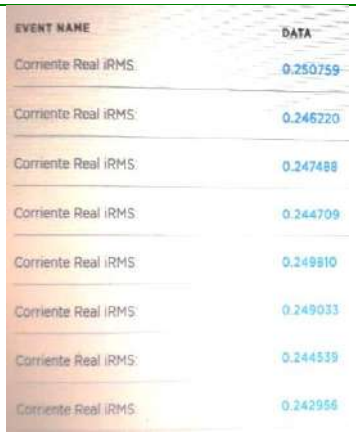
Tabla 9. Especificaciones técnicas del ventilador.

Características	Valor
Modelo:	FZ10-9HB
Voltaje de entrada	120V~
Potencia máxima	`40 W
Frecuencia de entrada	60 Hz
Consumo de energía en modo de operación	35 kWh

Fuente: Fabricante Daewoo (s.f.).

Para poder obtener medidas correctas se compararon con un multímetro digital provista con unas tenazas para tomar lecturas de corriente alterna, la primera prueba consistió en medir la corriente del ventilador a su primera velocidad.

Tabla 10. Resultados de la calibración.

Lectura reportada por el amperímetro (A)	Lectura de calibración reportada por el prototipo (A)
	

Fuente: Elaboración propia (2018).

En la Figura 47 se evidencia que el prototipo toma lecturas muy precisas, las gráficas se muestran correctamente en base a los datos recolectados y las lecturas reportadas por el sistema están calibradas.

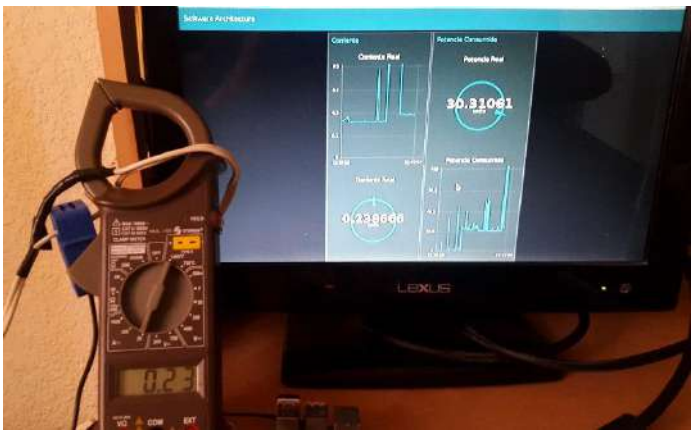


Figura 47. Verificación de lecturas utilizando un amperímetro y del prototipo.
Fuente: Elaboración propia (2018).

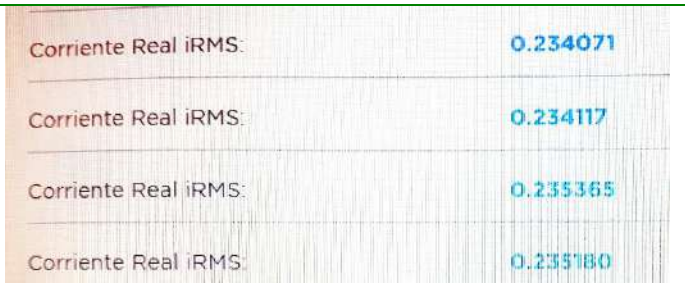
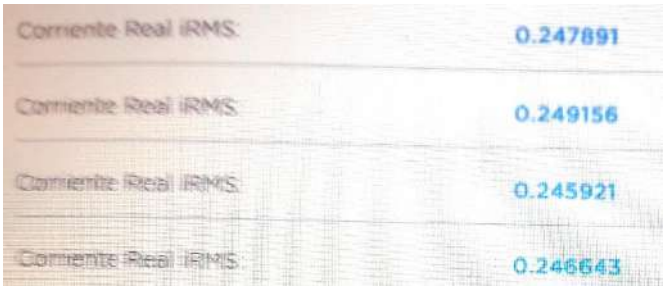
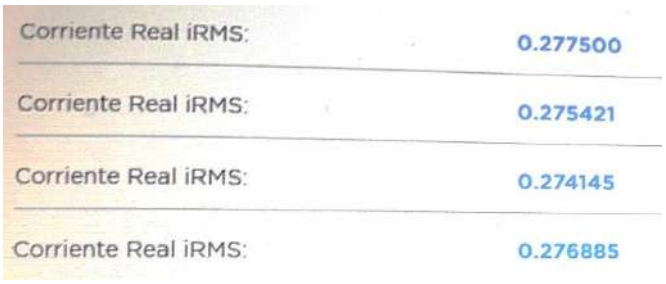
En la segunda prueba solo se apagó el ventilador y tanto el amperímetro como el sistema implementado deben de marcar una lectura de 0 A.

5.2 Pruebas del sensor de corriente

Las pruebas para determinar que el sistema implementado tiene un alto nivel de precisión se ejecutaron en distintos casos de usos. En la interpretación de los resultados es relevante indicar que la unidad de energía con la cual las compañías nos facturan es el kWh. Retomando el primer caso en el cual se usó un ventilador que de acuerdo a su etiqueta del fabricante tiene un consumo de 40 Watts, podemos inferir que si durante una hora mantenemos prendido el ventilador tendríamos un consumo de 40Wh. Con los elementos presentados de la arquitectura se pueden procesar estos datos de una forma fácil y calcular con una gran efectividad el consumo de cada dispositivo. Dado que los datos se almacenan en la base de datos InfluxDB alineados a un registro de tiempo como lo podemos ver los resultados de la Tabla 13.

Cómo se mencionó anteriormente la primera prueba se utilizó un ventilador variando la velocidad en tres modos de funcionamiento. Para poder verificar el correcto funcionamiento del sistema se realizó verificación cruzada con un amperímetro comercial. Una vez obtenido los datos teóricos se empezó con la prueba, dado que el sensor utilizado es no invasivo no se tuvo problemas para la implementación.

Tabla 11. Resultado de las lecturas de corriente de un ventilador a tres distintas velocidades.

Configuración	Medidas realizadas con el amperímetro (A)	Medida realizada con el prototipo (A)
1		
2		
3		

Fuente: Elaboración propia (2018).

Para calcular el porcentaje de error de los parámetros eléctricos de corriente obtenidos se utilizó la siguiente ecuación:

$$\%error = 100\left(\frac{X_{Amperimetro} - Y_{Prototipo}}{Y_{Prototipo}}\right)$$

Como se puede ver en las gráficas obtenidas de la interfaz gráfica de Node-RED como la corriente medida por el amperímetro, los resultados difieren muy poco contra los mostrados por el fabricante. Lo anterior puede deberse a la calibración del prototipo o incluso a que los componentes del electrodoméstico no sean exactamente los teóricos.

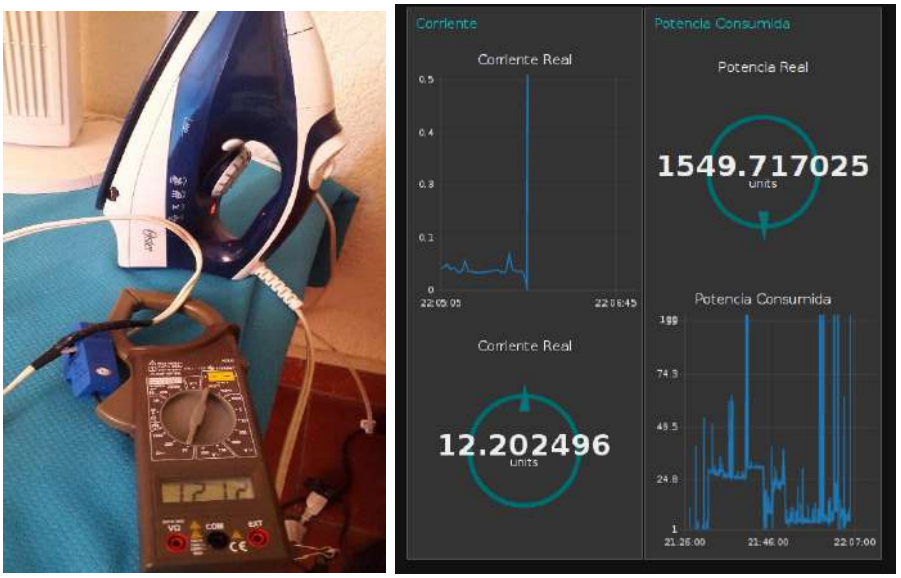


Figura 48. Prueba con una plancha a su máxima capacidad.
Fuente: Elaboración propia (2018).

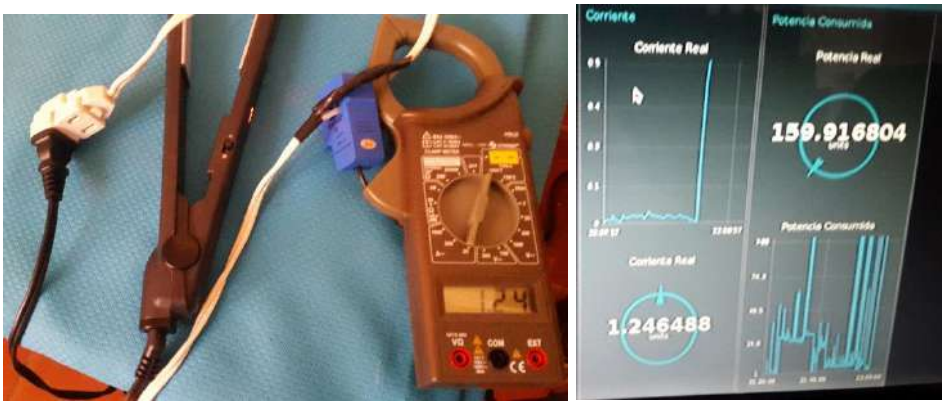


Figura 49. Prueba con un planchador de cabello en funcionamiento nominal.
Fuente: Elaboración propia (2018).

En la Tabla 12 se sintetizan los resultados obtenidos de las medidas realizadas en un entorno doméstico aplicados a distintos aparatos electrodomésticos.

Tabla 12. Resultados de las lecturas de corrientes de distintos aparatos electrodomésticos.

Electrodoméstico	Medida realizada con el amperímetro (A)	Medida realizada con el prototipo (A)
Plancha	12.12	12.20
Planchador de cabello	1.24	1.24
Televisor	0.22	0.24
Cargador de celular	0.04	0.049

Fuente: Elaboración propia (2018).

De la base de datos InfluxDB se colectaron un muestreo de 30 registros tanto del valor eficaz de la corriente alterna como de la potencia para comprobar que el sistema tiene la capacidad de almacenar los datos en memoria persistente, dichos datos pueden ser visualizados en la Tabla 13.

Tabla 13. Reporte de los datos almacenados en InfluxDB.

Marca de tiempo	Corriente alterna (A)	Marca de tiempo	Potencia (W)
2018-06-02T14:07:07.038987305Z	0.232473	2018-06-02T14:07:07.120721772Z	29.524052
2018-06-02T14:07:08.723167065Z	0.232302	2018-06-02T14:07:08.852056501Z	29.502365
2018-06-02T14:07:10.400098368Z	0.235862	2018-06-02T14:07:10.492635084Z	29.954455
2018-06-02T14:07:12.091060503Z	0.234414	2018-06-02T14:07:12.177797138Z	29.770581
2018-06-02T14:07:13.807707696Z	0.231434	2018-06-02T14:07:13.891186166Z	29.392149
2018-06-02T14:07:15.474586755Z	0.233385	2018-06-02T14:07:15.567492064Z	29.639957
2018-06-02T14:07:17.167624152Z	0.233424	2018-06-02T14:07:17.268776302Z	29.644787
2018-06-02T14:07:18.861108424Z	0.231152	2018-06-02T14:07:18.975908438Z	29.356347
2018-06-02T14:07:20.545050654Z	0.237478	2018-06-02T14:07:20.633163638Z	30.159752
2018-06-02T14:07:22.266692946Z	0.23228	2018-06-02T14:07:22.331269305Z	29.49959
2018-06-02T14:07:23.935383263Z	0.23485	2018-06-02T14:07:24.033359281Z	29.825939
2018-06-02T14:07:25.620104926Z	0.227422	2018-06-02T14:07:25.703530585Z	28.88263
2018-06-02T14:07:27.303188057Z	0.231601	2018-06-02T14:07:27.391644478Z	29.413317
2018-06-02T14:07:28.99595735Z	0.234254	2018-06-02T14:07:29.083599295Z	29.750245
2018-06-02T14:07:30.687305974Z	0.23191	2018-06-02T14:07:30.783181689Z	29.45254
2018-06-02T14:07:32.385890198Z	0.234875	2018-06-02T14:07:32.475434011Z	29.829103
2018-06-02T14:07:34.076785028Z	0.235935	2018-06-02T14:07:34.195382737Z	29.963706
2018-06-02T14:07:35.758678334Z	0.234945	2018-06-02T14:07:35.852682286Z	29.837971
2018-06-02T14:07:37.455757313Z	0.235528	2018-06-02T14:07:37.544933055Z	29.91208
2018-06-02T14:07:39.148824903Z	0.231468	2018-06-02T14:07:39.230101145Z	29.39649
2018-06-02T14:07:40.835441585Z	0.234787	2018-06-02T14:07:40.917883291Z	29.817995
2018-06-02T14:07:42.529840322Z	0.231863	2018-06-02T14:07:42.615813525Z	29.446651
2018-06-02T14:07:44.203932684Z	0.235166	2018-06-02T14:07:44.292754469Z	29.866127
2018-06-02T14:07:45.900209805Z	0.235099	2018-06-02T14:07:45.998903166Z	29.857551
2018-06-02T14:07:47.598845305Z	0.231529	2018-06-02T14:07:47.675642398Z	29.404147
2018-06-02T14:07:49.287860168Z	0.233443	2018-06-02T14:07:49.365058823Z	29.647209
2018-06-02T14:07:50.977574876Z	0.233467	2018-06-02T14:07:51.114273609Z	29.650358
2018-06-02T14:07:52.656348588Z	0.234205	2018-06-02T14:07:52.745441571Z	29.743985
2018-06-02T14:07:54.346217936Z	0.237858	2018-06-02T14:07:54.429359901Z	30.20799
2018-06-02T14:07:56.032669331Z	0.231836	2018-06-02T14:07:56.12045258Z	29.443154

Fuente: Elaboración propia (2018).

5.3 Pruebas de consumo del sistema

Uno de los atributos de la arquitectura es la mantenibilidad por lo que se evaluó el consumo que demanda el sistema. Los únicos elementos que requieren estar activos es la Raspberry Pi y el módulo Photon el cual es alimentada por la misma Raspberry Pi. De acuerdo con la prueba obtenida el consumo en operación normal de la Raspberry Pi modelo 3 es de 350 mA, podemos apreciar en la Figura 50 que el consumo es bajo.

Con esta prueba se demuestra que el consumo del sistema es muy bajo ya que basta con conectar la tarjeta Raspberry Pi para comenzar el funcionamiento.

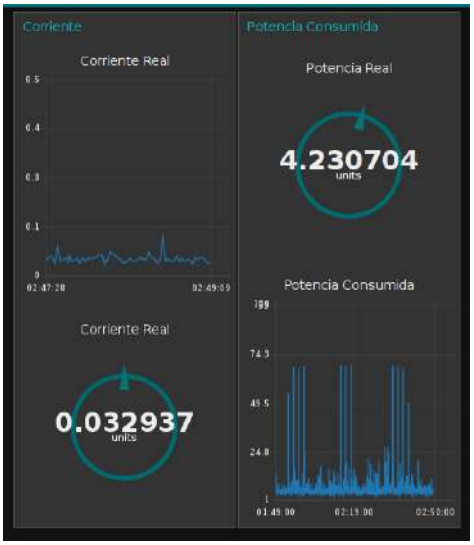


Figura 50. Pruebas de consumo del prototipo.
Fuente: Elaboración propia (2018).

5.4 Pruebas del servicio de correo electrónico.

Para evaluar que el servicio de notificación mediante correo electrónico estaba funcionando se colocó un nodo donde se estableció un límite que si el consumo en un instante de tiempo fuera mayor a 30 V enviará un mail de notificación al usuario. Adicionalmente se agregó un nodo que limitará enviar máximo 10 mensajes por cada cinco minutos, lo cual se ve representado en la Figura 51.

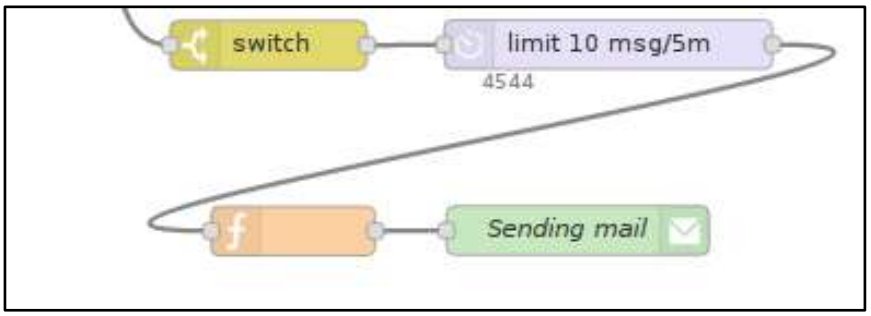


Figura 51. Configuración en Node-RED para enviar una notificación.
Fuente: Elaboración propia (2018).

En la Figura 52 se evidencia que el usuario estuvo recibiendo notificaciones provenientes de la plataforma Node-RED cuando el consumo rebasaba los límites previamente establecidos.

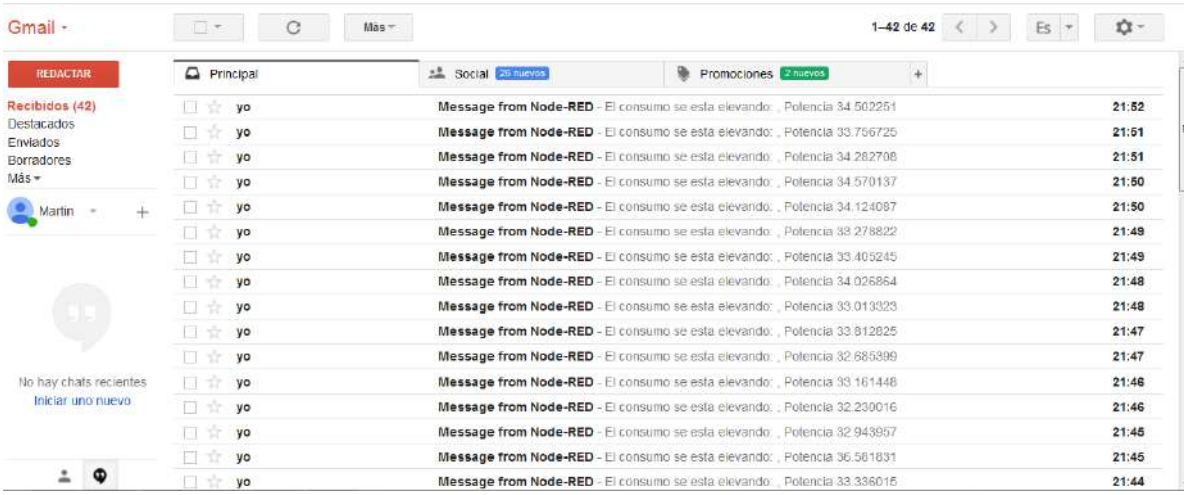


Figura 52. Notificaciones recibidas en el correo electrónico.
Fuente: Gmail (2018).

6. CONCLUSIONES

De la arquitectura propuesta se ha obtenido no solo un prototipo funcional, sino que además ha demostrado cumplir con los objetivos planteados inicialmente, los cuales se enlistan a continuación:

El sistema es sostenible puesto que se están usando tecnologías en desarrollo, lo que implica que existe una comunidad de respaldo con soporte para que el sistema se mantenga actualizado. Además de que el prototipo mostró capacidad de monitorear distintos aparatos electrodomésticos sin necesidad de una gran infraestructura comparado con otras soluciones actuales.

La arquitectura propuesta ha mostrado ser compatible y adaptable ya que los componentes tanto de hardware como de software a pesar que fueron desarrollados de manera independiente al ser integrados se logró el correcto funcionamiento. Además, la instalación en viable en cualquier ambiente residencial debido a que no requiere de muchos elementos de hardware y dispone de una gran variedad de elementos de software que se pueden utilizar.

El sistema es escalable dado que tiene la posibilidad de utilizar distintas bases de datos, una variedad de visualizadores e incluso una alta adaptabilidad a protocolos de comunicación ya que el uso de un motor de flujos como lo es Node-RED permite hacer uso de nodos. Lo anterior reutilizando el mismo código para incorporar nuevos sensores.

Así mismo, el prototipo tanto de hardware como de software permitió el monitoreo en línea del consumo de energía eléctrica de distintos aparatos electrodomésticos presentes en un ambiente residencia.

Se propuso el diseño de un algoritmo basados en reglas para notificar al usuario final de una forma inteligente y eficaz cuando el consumo excede los umbrales previamente calibrados.

Se desarrolló un modelo para la transmisión de la información en un ambiente del internet de las cosas de una forma segura.

.

6.1 Trabajo a futuro

Queda por hacer todavía iteraciones para alcanzar un sistema robusto además que como trabajo a futuro se pueden proyectar distintas líneas de investigación una de ellas sería el desarrollo de un sistema de control, ya que esta etapa podría permitir alcanzar un ahorro y eficiencia de la energía no solo de una forma visual sino también con ayuda de actuadores automatizados que tomen decisiones para eficientizar el consumo.

La arquitectura propuesta tiene la capacidad para medir solo el consumo de energía eléctrica haciendo uso del análisis de corriente alterna, pero además se tiene la versatilidad para expandir el monitoreo en línea a otros ecosistemas como podrían ser gas o agua.

Otra línea de investigación importante sería incluir más elementos a la plataforma web, como incluir gráficas de las tarifas de consumo eléctrico, visualización de las tarifas de distintos proveedores, además de poder compartir los datos con otros usuarios de entornos residenciales.

7. REFERENCIAS

CFE (2018). Portal de la Comisión Federal de Electricidad. Disponible en http://www.cfe.mx/.

Collotta, M., & Pau, G. (2017). An innovative approach for forecasting of energy requirements to improve a smart home management system based on BLE. *IEEE Transactions on Green Communications and Networking*, *1*(1), 112–120.

Conti, J., Holtberg, P., Diefenderfer, J., LaRose, A., Turnure, J. T., & Westfall, L. (2016). *International Energy Outlook 2016 With Projections to 2040*.

Cui, Y., Kim, M., Kum, S., Jung, J., Lim, T.-B., Lee, H., & Choi, O. (2014). Home Appliance Control and Monitoring System Model Based on Cloud Computing Technology. In *Mobile, Ubiquitous, and Intelligent Computing* (pp. 353–357). Springer.

De Vito, E. L. (2006). Introducción al razonamiento aproximado: lógica difusa. *Revista Americana de Medicina Respiratoria*, *6*(3).

Guinard, D., & Trifa, V. (2009). Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain* (Vol. 15).

Hassan, Q. F., & Madani, S. A. (2017). *Internet of Things: Challenges, Advances, and Applications*. Chapman and Hall/CRC

Hevner, A., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 75–105.

InfluxDB (2018). Sitio oficial de InfluxDB. Disponible en https://www.influxdata.com/

Jararweh, Y., Al-Ayyoub, M., Benkhelifa, E., Vouk, M., Rindos, A., & others. (2016). Software defined cloud: Survey, system and evaluation. *Future Generation*

Computer Systems, *58*, 56–74.

Jararweh, Y., Al-Ayyoub, M., Bousselham, A., Benkhelifa, E., & others. (2015). Software defined based smart grid architecture. In *Computer Systems and Applications (AICCSA), 2015 IEEE/ACS 12th International Conference of* (pp. 1–7).

Kim, J. E., Barth, T., Boulos, G., Yackovich, J., Beckel, C., & Mosse, D. (2017). Seamless integration of heterogeneous devices and access control in smart homes and its evaluation. *Intelligent Buildings International*, *9*(1), 23–39.

Kuechler, B., & Vaishnavi, V. (2011). Promoting relevance in IS research: An informing system for design science research. *Informing Science: The International Journal of an Emerging Transdiscipline*, *14*(1), 125–138.

Mahmood, A., Javaid, N., & Razzaq, S. (2015). A review of wireless communications for smart grid. *Renewable and Sustainable Energy Reviews*, *41*, 248–260.

Mas, A. (2005). *Agentes software y sistemas multiagente: conceptos, arquitecturas y aplicaciones*. Prentice Hall

Negeri, E., Baken, N., & Popov, M. (2013). Holonic architecture of the smart grid. *Smart Grid and Renewable Energy*, *4*(02), 202.

Node-RED (2015). Sitio oficial de Node-RED. Disponible en http://nodered.org/.

Orozco, Héctor (2016). Inteligencia Artificial: Tipos de Agentes, Disponible en: http://www.fdi.ucm.es/profesor/jpavon/doctorado/arquitecturas.pdf, Fecha de consulta: 20/07/2018.

Particle (2018). Sitio oficial de Particle. Disponible en https://www.particle.io/.

Pi, R. (3). model B (2018). *Raspberrypi. org. Saatavissa: https://www. raspberrypi. org/products/raspberry-pi-3-model-b/.*

Pressman, R. (2002). Ingenier{\'\i}a del Software-Enfoque Práctico Mc Graw Hill 5ª.

Edición Año.

PROFECO (2015). Portal de la Procuraduría Federal del Consumidor. Disponible en https://www.profeco.gob.mx/.

Santofimia, M. J., Toro, X. del, & López, J. C. (2011). Técnicas de inteligencia artificial aplicadas a la red eléctrica inteligente (Smart Grid). *Novática: Revista de La Asociación de Técnicos de Informática*, (213), 29–34.

Shahgoshtasbi, D., & Jamshidi, M. M. (2014). A new intelligent neuro--fuzzy paradigm for energy-efficient homes. *IEEE Systems Journal*, *8*(2), 664–673.

SparkFun (2018). Sitio oficial de SparkFun Electronics. Disponible en https://www.sparkfun.com/.

Strasser, T., Andrén, F., Kathan, J., Cecati, C., Buccella, C., Siano, P., … others. (2015). A review of architectures and concepts for intelligence in future electric energy systems. *IEEE Transactions on Industrial Electronics*, *62*(4), 2424–2438.

Wang, W., Xu, Y., & Khanna, M. (2011). A survey on the communication architectures in smart grid. *Computer Networks*, *55*(15), 3604–3629.

8. ANEXOS

8.1 Anexo A

```
// Este #include fue automáticamente agregado por Particle IDE.
#include <Ubidots.h>
#include "math.h"
#define TOKEN "A1E-bgeGoXBbHmGJKyf4EyRNRnwVOX7XLj"

Ubidots ubidots (TOKEN);
// Voltaje RMS
const double voltajeRMS = 127.0;      // Valor medido

// Parameters for measuring RMS current
const double desplazamiento = 1.619; // Mitad del voltaje máximo del ADC
const int numeroVueltas = 1800;     // 1:2000 Relación de vueltas del transformador
const int resistenciaBurden = 62;  // Valor de la resistencia de Burden
const int numeroMuestras = 1000;    // Número de muestras antes de calcular RMS

void setup() {

}

void loop() {

   int muestra;
   double voltaje;
   double iPrimario;
   double acc = 0;
   double iRMS;
   double potenciaCalculada;

   // Tomarun número de muestras y calcular la corriente RMS
   for ( int i = 0; i < numeroMuestras; i++ ) {
      // Leer ADC, convertir a voltaje y eliminar el desplazamiento
      muestra = analogRead(A0);
        //  Particle.publish("Valor Analog read:", String(muestra));
      voltaje = (muestra * 3.3) / 4096;
      //Particle.publish("Voltaje read:", String(voltaje));
      voltaje = voltaje - desplazamiento;
      // Calculando la corriente sensada
      iPrimario = (voltaje / resistenciaBurden) * numeroVueltas;

      // Corriente elevada y sumando lo acumulado
      acc += pow(iPrimario, 2);
   }

   // Calcular la corriente iRMS de los valores acumulados
   iRMS = sqrt(acc / numeroMuestras);
   Particle.publish("Accumulado:", String(acc));
   // Calcular la potencia y publicarla a Node-RED
   potenciaCalculada = voltajeRMS * iRMS;
```

```
    Particle.publish("Corriente iRMS:", String(iRMS));
    Particle.publish("Voltaje RMS:", String(voltajeRMS));
    Particle.publish("Potencia",String (potenciaCalculada), PRIVATE);

    delay(5000);
}
```

8.2 Anexo B

```
// Este #include fue automáticamente agregado por Particle IDE.
#include <semonlib.h
// Este #include fue automáticamente agregado por Particle IDE.
#include <semonlib.h>
// Este #include fue automáticamente agregado por Particle IDE.
#include <math.h>
// Include Emon Library
EnergyMonitor emon1;            // Crear una instancia
char resultstr[464];                //Variable de tipo string para almacenar los valores sensados.

int syc;
double val[14];
void setup()
{
  //Spark.variable("result", &resultstr, STRING);
  emon1.current(4, 30);             // Inicializar la librería: número de entrada, factor de calibración
  syc=1;
}
void loop()
{
  double Irms = emon1.calcIrms(2000);  // Calcular solo la corriente IRMs
  double IrmsCalibrada=Irms-0.03597;
  double x=Irms;
  double Potencia=IrmsCalibrada*127;
  Serial.print(x);
  Serial.println(" ");
  val[syc]=x*230;
  if(syc>11){
    sprintf(resultstr,
"{\"r1\":%f,\"r2\":%f,\"r3\":%f,\"r4\":%f,\"r5\":%f,\"r6\":%f,\"r7\":%f,\"r8\":%f,\"r9\":%f,\"r10\":%f,\"r11\":%f}",val[1],val[2],val[3],val[
4],val[5],val[6],val[7],val[8],val[9],val[10],val[11]); // Escribir los datos sensado en la cadena.
    syc=0;
  }
  syc=syc+1;
      Particle.publish("Corriente Real iRMS:", String(IrmsCalibrada));
      Particle.publish("Potencia", String(Potencia));
       delay(1000);
}
```

8.3 Anexo C

```
adminAuth: {
    type: "credentials",
    users: [{
        username: "admin",
        password:
"$2a$08$zZWtXTja0fB1pzD4sHCMyOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN.",
        permissions: "*"
    }]
}
```