

Internet de las cosas para el control y el monitoreo del consumo de energía eléctrica

Internet of things for the control and monitoring of electrical energy consumption

Marcos Emilio Chauque

marcos.chauque88@gmail.com

Ingeniería en Informática – Facultad de Ingeniería – Universidad Católica de Salta

Resumen

En los últimos años, el consumo de energía eléctrica en los hogares creció de manera exponencial, generando preocupación tanto para los usuarios finales, como para los proveedores de energía. A través de los avances tecnológicos, se busca contar con herramientas que permitan controlar el consumo de manera eficiente. Por lo tanto, este proyecto busca ofrecer una solución basada en la tecnología Internet de las cosas (IoT, por su sigla en inglés) para monitorear en tiempo real el consumo eléctrico en una vivienda, permitiendo a los usuarios reducir costos y optimizar su uso.

Todo esto es posible gracias al uso de plataformas como Raspberry Pi y Arduino, las cuales están conectadas a sensores de corriente no invasivos, permitiendo la adquisición de datos, que luego se procesan y visualizan en una interfaz web simple a través de OpenHAB (sistema de código abierto que permite la automatización de hogares).

Este enfoque permite a los usuarios reducir el consumo de energía y, en consecuencia, disminuir sus gastos y tomar decisiones más inteligentes sobre el uso de sus dispositivos, contribuyendo también a una gestión sostenible y beneficiosa para el entorno.

Palabras clave

Internet de las cosas, Raspberry, Arduino, OpenHAB, consumo energético.

Abstract

In recent years, electricity consumption in households has grown exponentially, generating concern for both end users and energy providers. Through technological advancements, the goal is to have tools that allow for efficient consumption control. Therefore, this project aims to offer a solution based on IoT (Internet of Things) technology to monitor real-time electricity consumption in a home, enabling users to reduce costs and optimize their usage.

All of this is made possible through the use of platforms such as Raspberry Pi and Arduino, which are connected to non-invasive current sensors, allowing data acquisition that is then processed and visualized in a simple web interface via OpenHAB (an open-source system that enables home automation).

This approach allows users to reduce energy consumption, consequently lowering their bills, and make smarter decisions about the use of their devices, also contributing to sustainable and environmentally beneficial management.

Keywords

Internet of Things, Raspberry, Arduino, energetic consumption

Introducción

En la actualidad el consumo energético se considera un problema, que año tras año se intensifica. La mejor manera de contrarrestar este problema es prevenirlo. Por este motivo, el presente proyecto plantea desarrollar un sistema para la supervisión en tiempo real del consumo eléctrico en entornos residenciales. Presenta un enfoque práctico de control y monitoreo del consumo energético, facilitando una gestión más eficiente de los recursos eléctricos en el hogar.

Se plantea el desarrollo de una aplicación que permita la supervisión y adquisición en tiempo real de los parámetros eléctricos, como el voltaje, la corriente y la potencia.

El sistema se basa principalmente en los conceptos de Internet de las cosas y la domótica aplicada. En la fase inicial del proyecto, se realiza un estudio detallado y un análisis exhaustivo de estos conceptos. En la siguiente fase, se define la problemática que motivó la creación del sistema, la importancia de su estudio, su objetivo, alcance y limitaciones potenciales. Luego se presenta la fase de diseño e implementación de la solución propuesta. Esto incluye el diseño de la arquitectura del sistema, la selección de sensores y dispositivos adecuados, y la programación de la interfaz de usuario para la supervisión y control. Se desarrollan los algoritmos necesarios para el procesamiento de datos y se integra la aplicación web para la visualización y análisis de la información recogida por los sensores.

Por último, se realiza una conclusión bajo los objetivos planteados inicialmente. Se evalúan los resultados del sistema a través de pruebas experimentales, basadas en los datos obtenidos durante las pruebas para medir el rendimiento y la efectividad del sistema. Esta fase es esencial para validar la funcionalidad del sistema.

1. Aspectos generales

El objetivo de esta sección es explorar cómo el Internet de las cosas (IoT, por su sigla en inglés) se convierte en una herramienta valiosa para abordar este desafío energético.

El Internet de las cosas (IoT, por su sigla en inglés) es una red de dispositivos físicos conectados a internet, capaces de recoger y compartir datos a través de sensores y *software* integrados. Estos dispositivos pueden interactuar entre sí, automatizar tareas y tomar decisiones de manera autónoma, mejorando la eficiencia en diversos sectores como el industrial, el doméstico, la salud y las ciudades inteligentes.[1]

1.1. Descripción del problema

La aplicación y utilización de la tecnología en cualquier aspecto, ya sea para brindar bienes y servicios, implica un uso de energía para que funcione. Esta energía puede provenir de diferentes fuentes, como electricidad, combustibles, etc. La producción, generación y administración de esta energía son temas de gran importancia. De hecho, se ha establecido una conexión directa entre el nivel de vida de las personas y la cantidad de energía que se consume. Esto significa que, a medida que una sociedad progresa y se desarrolla, también tiende a incrementarse, impulsada por el desarrollo de nuevas tecnologías y actividades productivas.

El incremento en el desarrollo de una nación, es decir, su crecimiento económico, social y tecnológico, está relacionado con un mayor consumo, uso y capacidad de transformación de energía de manera eficiente. En otras palabras, a medida que una sociedad avanza, su demanda de energía también aumenta debido a las actividades y tecnologías que se desarrollan y utilizan.

Este consumo energético se justifica debido a las necesidades que permiten que las personas vivan cómodamente. Esto incluye el funcionamiento de sistemas y sectores como el transporte (vehículos, trenes, aviones), el sector doméstico (hogares), el sector industrial (fábricas y manufactura), el sector comercial (negocios y tiendas), el sector institucional (escuelas, hospitales, edificios gubernamentales) y el sector agrícola (granjas y producción de alimentos), entre otros. Todos estos sectores requieren energía para operar de manera efectiva y brindar los servicios y productos que la sociedad necesita.

A nivel mundial, con respecto a la producción de energía eléctrica, la misma aumentó un 3.3 % en el último trimestre del año 2020, alcanzando los 34.564 GWh. De esta manera, se registró un aumento del 4.2 % por parte del sector residencial y del 2.8 % por parte de la demanda industrial/comercial mayor a 300 kW. Al final del cuarto trimestre del 2022, la potencia instalada en el SADI (Sistema Argentino de Interconexión) fue de 42.925 MW. [2]

1.1.2. Impacto del uso irracional de la energía

El uso irracional de la energía conlleva un uso ineficiente de los recursos naturales. Esto significa que, si la energía no se utiliza de manera responsable y eficiente, se están agotando recursos naturales sin garantizar que las necesidades de la población creciente puedan ser satisfechas en el futuro. La globalización y el avance tecnológico han llevado a un excesivo aumento en el consumo eléctrico.

Dado que gran parte del consumo energético del país proviene de los hogares, se entiende que la población en general, especialmente los hogares, serán los más afectados por una posible escasez de energía. Esto podría resultar en interrupciones de suministro o, en el caso contrario, en incrementos en los costos de generación que se reflejarían en facturas más altas para los usuarios finales.

En Argentina, en 1980 el consumo de energía eléctrica promediaba los 1318 kWh per cápita al año, mientras que en el año 2020 este valor se multiplicó casi 2.04 veces, llegando a 2700 kWh per cápita al año. [2]

Otro dato importante es el que proporciona CAMMESA (Compañía Administradora del Mercado Mayorista Eléctrico S.A.), que en el año 2020 muestra que la demanda residencial en el país fue del 47 % del total de la energía eléctrica producida. [3]

Por tales motivos, es necesario aplicar esfuerzos en el desarrollo de tecnologías que contribuyan a reducir dicho consumo y concientizar a los usuarios sobre el impacto ambiental que esto genera.

1.2. Objetivo

El objetivo general del proyecto se basa principalmente en diseñar e implementar de un prototipo funcional, sencillo y de bajo costo, con la finalidad de monitorizar y gestionar el consumo energético de los diferentes dispositivos conectados a la red eléctrica y optimizar el uso de energía en un entorno doméstico.

1.3 Alcance

Este trabajo pretende implementar un sistema de control y monitoreo del consumo eléctrico, utilizando herramientas de *software* y *hardware*, como lenguajes de programación, base de datos, Raspberry, Arduino, sensores y dispositivos de conectividad.

El sistema tendrá una interfaz web adaptable a dispositivos móviles. La función principal del sistema será reportar el consumo en tiempo real de los diferentes electrodomésticos conectados a la red residencial. Además, tendrá la capacidad de almacenar historial de los usuarios, actividades e interacciones con el sistema, generando perfiles de consumo y preferencias. También proporcionará alertas de consumos excesivos, ayudando a la toma de decisiones en el uso de energía a corto, mediano y largo plazo (diario, semanal y mensual).

Requisitos

Los requisitos del proyecto simbolizan las necesidades, requerimientos y expectativas dentro del proyecto, que serán clasificados según su función.

Requisitos funcionales

- » Monitoreo del consumo energético
- » Consulta del historial de consumo
- » Generación de notificaciones y alertas
- » Configuración de la comunicación
- » Gestión de usuarios (ABM de cuentas)
- » Configuración de cuentas de usuarios
- » Configuración de dispositivos conectados.

Requisitos no funcionales

- » Seguridad en la web
- » Disponibilidad
- » Facilidad de uso del sistema (Usabilidad)
- » Escalabilidad del sistema
- » Rendimiento eficiente
- » Bajo costo de implementación
- » Bajo consumo energético

2. Solución propuesta

2.1. Metodología de desarrollo

La metodología utilizada para el desarrollo del prototipo será la metodología *Scrum*. La metodología *Scrum*, se define como una metodología ágil utilizada para gestionar proyectos complejos y desarrollar productos de manera eficiente. Se basa en los principios de transparencia, inspección y adaptación, promoviendo la colaboración dentro de equipos multidisciplinares. La metodología divide el trabajo en ciclos cortos llamados sprints, con entregas incrementales de valor que permiten ajustes continuos y respuestas rápidas a los cambios. [4]

2.2. Módulos del sistema

2.2.1. Software

- » **Módulo web:** Comprende el desarrollo de un sistema web basada en la distribución Raspbian (Open Source), que permitirá realizar consultas del consumo de eléctrico de manera local y remota. Se plantea el diseño de dos tipos de interfaces, una con una conexión directa desde el servidor local embebido, y otra para acceso remoto a través de Internet. La interfaz será intuitiva y amigable.
- » **Reportes:** Para solucionar el problema de IP dinámica en la conexión a Internet existente en la red doméstica, se incluirá un software externo llamado "No-IP" para solucionar problemas de IP dinámica, sincronizando la IP del servidor con un dominio predefinido, asegurando acceso continuo al sistema sin depender de la IP actual del servidor.

- » **Servidor/Base de datos:** Se prevé implementar un servidor web Apache en la plataforma Raspberry Pi 3 bajo la infraestructura LAMP (Linux, Apache, MySQL, PHPMyAdmin), que manejará tanto el servidor web como la base de datos del sistema.
- » **Módulo local:** Este módulo incluirá un sistema embebido en Raspberry Pi 3 que recopilará datos a través de sensores de corriente y permitirá la automatización del corte de energía cuando el usuario lo desee.

2.2.2. Hardware

- » **Plataforma Raspberry Pi 3:** Raspberry Pi es una computadora de placa única, desarrollada en el Reino Unido, diseñada para promover la enseñanza de la informática básica y la programación. [5]

En este proyecto, la plataforma Raspberry Pi3, se configurará con el sistema operativo Raspbian (basado en Debian). Actuará como servidor web y base de datos, gestionando la conectividad con los módulos locales y periféricos.

- » **Plataforma Arduino:** Arduino es una plataforma de hardware libre basada en una sencilla placa con un microcontrolador y un entorno de desarrollo integrado (IDE, por su sigla en inglés), que permite la creación de proyectos electrónicos interactivos. [6]

Arduino actuará como interfaz física entre Raspberry Pi y los dispositivos conectados. Incluirá submódulos para procesamiento y cálculo, acondicionamiento de señal, y conectividad en red.

- » **Periféricos:** los periféricos estarán conectados a la placa Raspberry Pi de manera física. Los periféricos son: Display LCD, sensores de corriente no invasivo, actuadores e interruptores.

2.3. Diseño e implementación de hardware

El sistema está compuesto por cuatro partes: los sensores, la placa Arduino, la interfaz inalámbrica y la placa Raspberry Pi.

El componente principal es la placa Raspberry pi, en ella se conectarán la placa Arduino que servirá de interfaz con los sensores y los actuadores. También a través de este dispositivo se logrará la conexión a Internet mediante un adaptador Wi-Fi.

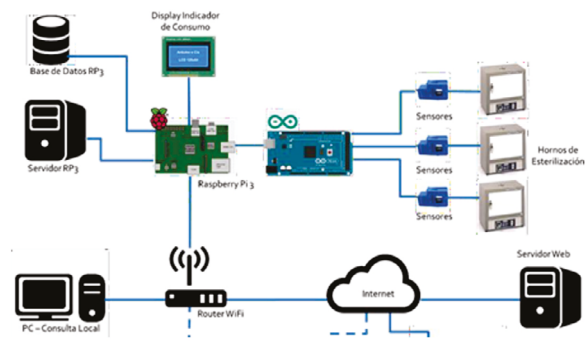


Figura 1: Diagrama de arquitectura del Sistema de Monitoreo de Consumo Eléctrico

En la Figura 1 se muestra el diagrama de los componentes principales del dispositivo y su funcionamiento.

2.3.1. Conectividad de los sensores de corriente

Para poder realizar la adquisición de los datos medidos por los sensores de corriente, estos deben estar conectados mediante un adaptador Jack 3.5 a la placa Arduino.

2.3.2. Interfaz Wi-Fi

La conexión a Wi-Fi se realiza a través de un adaptador de Red Wi-Fi tipo USB. Al conectarlo a la placa Raspberry Pi funciona como una tarjeta de red estándar.

2.4. Diseño e implementación de software

2.4.1. Programa de procesamiento de datos y cálculo

Como primer paso, se requieren los valores de voltaje y corriente. Para esto se debe implementar un circuito que atenúa la señal y elimina la componente negativa antes de ingresarla al conversor analógico-digital (ADC, por su sigla en inglés) del microcontrolador.

El código de programación está escrito en el entorno IDE de Arduino. En este se configuran el módulo Ethernet, las rutinas de adquisición de datos y el envío de información a la red LAN. Se incluyen las librerías necesarias para gestionar el módulo Ethernet, el display LCD y la tarjeta SD. Las variables globales para la manipulación de datos se inicializan al arrancar el módulo Arduino.

En la función Setup (), se declaran los parámetros del socket para la configuración del módulo Ethernet, es decir se define el puerto de comunicación y las IP del cliente y servidor, a estas se les agrega sentencias de configuración de los pines analógicos, así como también la inicialización de variables y servicios de comunicación serial. Cabe mencionar que esta función se ejecuta una sola vez después de haber inicializado el módulo Arduino.

En el bucle Loop () se fija un procedimiento de adquisición de datos de los sensores, para ello se habilitan las entradas analógicas y consecutivamente se ejecutan las sentencias para calcular los consumos electrónicos de cada circuito, dicha información es recopilada en una variable de tipo cadena, la cual es enviada hacia el display LCD y retransmitida hacia el servidor web mediante el protocolo TCP/IP. La Figura 2 muestra el diagrama de flujo del programa en Arduino.

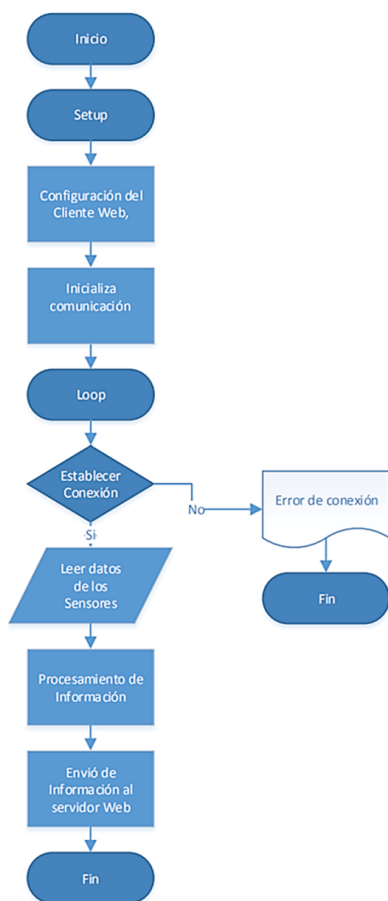


Figura 2: Diagrama de flujo de la programación de Arduino

2.4.2 Sistema operativo

Existe una gran variedad de distribuciones de Linux para la RaspberryPi. Entre ellas se encuentran las distribuciones de propósito general como las de propósito específico. Algunas, como RetroPie, la cual permite convertir la RaspberryPi en un centro de emulación de consolas antiguas, o arkOS, que ofrece herramientas para crear y administrar un servidor *cloud* privado de archivos.

Para este proyecto, se seleccionó el sistema operativo Raspbian. Fue de interés escoger una distribución de propósito general, por su flexibilidad en términos de posibles configuraciones, cualidad importante y crucial para proyectos de investigación y desarrollo.

El sistema Raspbian es una versión de Debian concebida específicamente para RaspberryPi. Debian (la combinación de los nombres Debbie e Ian) es una distribución Linux dirigida por su comunidad de usuarios, que pone mucho énfasis en el desarrollo de código abierto. Ninguna organización comercial participa en el desarrollo de Debian. Raspbian amplía Debian con herramientas y paquetes de *software* por ejemplo Java, Mathematica o Scratch, específicos para RaspberryPi[7].

2.4.3 Servidor web LAMP

LAMP es una palabra compuesta por las iniciales Linux, Apache, MySQL y PHP. Todo este paquete forma la construcción del servidor, por lo que hace posible la creación y el alojamiento de páginas web. Ver Figura 3.

El sistema operativo Raspbian trabaja en la plataforma de Linux y ayuda a ejecutar el servidor web Apache. Su funcionamiento es en conjunto, se podría decir que Apache envía un código fuente al intérprete PHP incluyendo toda la información correspondiente sobre todas las acciones que llegará a realizar el visitante web, permitiendo de tal forma poder tener acceso a la base de datos MySQL (apache.org, s.f.). [8]



Figura 3: Logo LAMP

2.4.4 Base de Datos

Una base de datos se define como “una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo principal de un sistema de bases de datos es proporcionar un ambiente que sea conveniente y eficiente para recuperar y almacenar información”. [6]

En este caso, la función de la base de datos será garantizar la integridad y seguridad de la información almacenada. Esta debe ser robusta y confiable, permitiendo la gestión eficiente de cuentas de usuarios y contraseñas. Esto va a garantizar que solo los usuarios autorizados, con los permisos adecuados, puedan acceder a la información, protegiendo así los datos sensibles frente a accesos no autorizados.

El *software* gestor de base de datos a utilizar será MySQL. MySQL es un sistema de gestión de bases de datos relacional diseñado para realizar

operaciones de manipulación de datos utilizando el lenguaje SQL. Es popular en aplicaciones de código abierto y se caracteriza por su alta velocidad y flexibilidad. [9]

En este sentido, para lograr la conexión a la base de datos, se necesitan cuatro parámetros importantes, el nombre del usuario creado en la base de datos, la contraseña respectiva del usuario, la dirección IP del servidor y el respectivo nombre de la base de datos.

Se observa en la Figura 4 el diagrama de flujo de la conexión y envío de información hacia la base de datos.

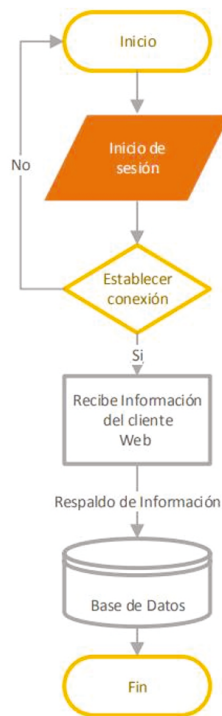


Figura 4: Diagrama de Flujo de Conexión a la Base de Datos

2.4.5 OpenHAB

OpenHAB es un *software*, desarrollado en Java y basado en Eclipse SmartHome, utilizado para la integración de diferentes sistemas y tecnologías relacionadas con el ámbito domótico. Permite utilizar reglas de automatización y dispone de diferentes interfaces para usuarios uniformes. Esto hace que OpenHAB esté diseñado para ser neutral, tanto en términos *software* como *hardware* y para integrar una gran cantidad de tecnologías domóticas diferentes en una sola. Puede ejecutarse en cualquier dispositivo que soporte una JVM (Java Virtual Machine, por su sigla en inglés) y posee un potente motor de reglas para la automatización del centro domótico instalado, además de disponer de mecanismos de persistencia para poder realizar gráficos y análisis de algunos sensores instalados. [10]



Figura 5: Logotipo OpenHAB

OpenHAB es una solución de *software* libre, posee interfaces web accesibles desde cualquier dispositivo con navegador y, además, presenta una aplicación nativa para sistema iOS y Android. Cabe destacar que es fácilmente extensible mediante la integración con nuevos dispositivos y sistemas, esto es posible gracias a diferentes API desarrolladas. Ver Figura 5

2.4.6 Configuración de OpenHAB

Tal y como se recoge en la documentación de la página oficial de openHAB, en su sitio web, la instalación se puede realizar de diferentes maneras. El primer paso será ver qué versión de Java disponible tiene el servidor. Para ello, desde el terminal, se comprobará la versión:

```
java -versión
```

En caso de no disponer de Java instalado, se deberá de instalar con los siguientes comandos:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt update
sudo apt-get install oracle-java8-set-default
```

Para la utilización del protocolo HTTPS, se agregará la clave del repositorio de openHAB Bintray, utilizando las siguientes sentencias:

```
!wget -qO- 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab' |
sudo apt-key add -
sudo apt-get install apt-transport-https
```

Luego se procede a la selección de la versión estable, puesto que será la que menos errores contenga, junto con los paquetes actualizados.

```
echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' |
sudo tee /etc/apt/sources.list.d/openhab2.list
sudo apt-get update
sudo apt-get install openhab2
sudo apt-get install openhab2-addons
```

Para finalizar, y para que se pueda ejecutar el servicio cuando inicie el servidor, se deberán ejecutar las siguientes sentencias:

```
sudo systemctl start openhab2.service
sudo systemctl status openhab2.service
sudo systemctl daemon-reload
sudo systemctl enable openhab2.service
```

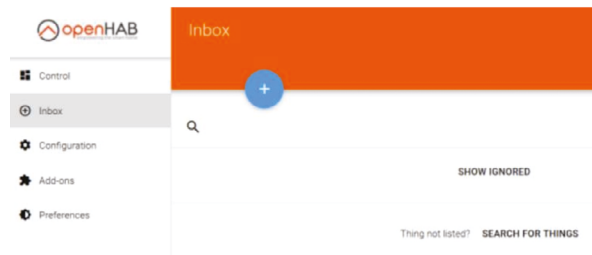


Figura 6: Interfaz Paper UI de OpenHAB

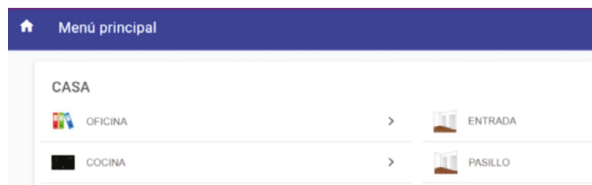


Figura 7: Interfaz Basic UI de OpenHAB

2.5. Seguridad del sistema

El uso de dispositivos IoT también presenta un gran desafío en el tema de seguridad, porque al involucrarse la privacidad del usuario hace que se vean expuestos todo tipo de datos que abren una puerta inmensa al ataque cibernético, provocando todo tipo de riesgos y amenazas para la persona que adquiere estos productos sin darse cuenta a lo que estos conllevan.

La seguridad de esta nueva tecnología se ha visto afectada en diversos aspectos con el uso de una aplicación inteligente, un dispositivo IoT, un sistema domótico, muchas veces sin que el propietario se dé cuenta de que los equipos han sufrido un ataque.

Es crucial implementar medidas de seguridad robustas para mitigar estos riesgos. Entre las soluciones propuestas se incluyen mecanismos de autenticación y autorización, que garantizan un acceso seguro a los dispositivos IoT, ayudando al usuario a priorizar la seguridad sobre la conveniencia. Estas medidas buscan proteger los sistemas domóticos y otras aplicaciones inteligentes contra posibles vulnerabilidades y ataques.

2.5.1 Seguridad en OpenHAB

OpenHAB permite la adaptación de dispositivos de diferentes protocolos, lo que la lleva a ser una de las más usadas. No presenta una interfaz compleja y su instalación es bastante sencilla. Necesita de Java para poder funcionar. En la integración de OpenHAB existe varias interfaces en las que el usuario puede trabajar para el control de dispositivos, entre esas están Basic UI, Paper UI, Classic UI y HAB Panel, cada una de ellas tienen sus características y son escogidas de acuerdo con la facilidad de entendimiento que puede considerar el usuario para la ejecución de su panel de control. Ver Figura 6 y Figura 7.

Para mejorar la seguridad en OpenHAB, se implementó un servidor VPN basado en OpenVPN, donde se pueden establecer conexiones de alta seguridad en la comunicación de red, incluso utilizando infraestructura distribuida o pública.

La solución de OpenVPN para Raspberry Pi utilizada puede resultar un tanto compleja de instalar y configurar la primera vez, pero que a la larga se ha mostrado muy robusta, con poca necesidad de mantenimiento. También existen diversos tutoriales, e incluso *scripts* que automatizan gran parte del proceso, como es el caso de PIVPN. Y, nuevamente, una comunidad de usuarios muy activa a la que se puede recurrir para solucionar dudas, aunque lo más probable es que ya hayan sido planteadas y resueltas con anterioridad.

Posteriormente, se configuró OpenHAB con MQTT. Primero, se instaló el MQTT Broker desde la pestaña MISC en la opción Addons, configurándolo manualmente.

Luego, se agregó un dispositivo genérico (*thing*) utilizando el MQTT Broker como puente para la comunicación y luego se agregó un canal de tipo *switch* donde se añadió el *topic* que tendrá el dispositivo para su estado y envío de comando, en ambos se debe considerar el ID, *local key* y la IP del dispositivo que se gestionó. Por último, se creó un ítem tipo *switch* y con eso se terminó la configuración en Open HAB.

Además de instalar el servidor de VPN (en la Raspberry Pi de servicios que también aloja la base de datos, y algún otro servicio más), son necesarios algunos pasos adicionales para tener el entorno plenamente funcional.

3. Pruebas y resultados

3.1. Prueba de calibración

Las pruebas consistieron en realizar con el sistema desarrollado mediciones de voltaje, corriente, potencia y energía consumida. La finalidad de esta etapa fue evaluar la precisión y la exactitud del sistema, comparando los valores medidos por los instrumentos de medición y los valores entregados por el sistema. Teniendo en cuenta que los instrumentos de medición utilizados en el sector tecnológico disponen de su respectivo certificado de calibración, por lo que asumimos que las mediciones entregadas por los mismos son correctas.

3.1.1. Procedimiento

Para la implementación del medidor de corriente se ha utilizado la librería Emonlib. Para el cálculo de la corriente RMS, primero, se necesita inicializar la librería con la función "emonl.current (pin, calibración)" en la que el parámetro de calibración dependerá del tipo de SCT.

Este parámetro refleja el número de espiras dividido entre su resistencia interna. Según la hoja de características (*datasheet*), el número de espiras es 2000 y no tiene resistencia interna. Sin embargo, en la implementación, se descubrió que la falta de resistencia interna generaba errores. Por tanto, esto llevó a determinar un número de calibración de forma empírica.

Las pruebas de calibración consistieron en medir un dispositivo eléctrico con su consumo conocido, dado por el fabricante. Cuando el dispositivo estaba apagado, el medidor debía indicar un consumo de cero, y cuando estaba encendido, debía registrar el consumo esperado. A partir de estas pruebas, se identificó que el SCT tenía una resistencia interna no especificada en su hoja de características, afectando las mediciones originales.

3.2. Prueba de consumo eléctrico

Un aspecto crucial en esta prueba es la velocidad a la que se generan los pulsos y el retardo causado por la red al transmitir los datos a la plataforma web. Por eso, en esta prueba se enviarán los datos cada cierto tiempo. Cuando el dispositivo envía datos, este no puede detectar pulsos. Así, en el código de este dispositivo se realizaron ajustes para estimar los pulsos perdidos durante la transmisión, basándose en el consumo medido justo antes de enviar los datos. Los resultados de esta prueba se muestran en la Figura 8.

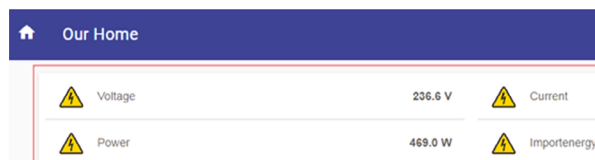


Figura 8: Panel de consumo

Además del consumo habitual de electrodomésticos, se realizaron diferentes consumos controlados. Para comprobar que el dispositivo era capaz de monitorizar tanto altos consumos como bajos consumos, se enchufaron y hacer uso de diferentes dispositivos eléctricos, como: microondas, cafetera, impresora, televisor, etc.

Siendo consciente del posible consumo que se estaba produciendo y el momento en que se estaba produciendo, se puede afirmar que los resultados obtenidos han sido de un grado muy satisfactorio. Al igual que en las pruebas anteriores, observando en todo momento el consumo antes y después de la prueba. Finalmente, se puede decir que se tiene un dispositivo capaz de monitorizar el consumo eléctrico.

3.3. Prueba del sensor de corriente

A continuación, se realizó un análisis del consumo de diferentes dispositivos electrodomésticos, comparando el consumo teórico con el medido empíricamente. Para esta prueba, se utilizó un dispositivo con tres niveles de potencia. Las especificaciones y características técnicas fueron obtenidas de la página oficial del fabricante. El dispositivo de medición, de tipo no invasivo, se puso en funcionamiento de manera sencilla, como se ilustra en la Figura 9.



Figura 9: Medidor de consumo eléctrico – Display LCD

Como se puede ver, en este caso el dispositivo está conectado a una batería externa. Para poder ver los valores medidos se utilizó tanto OpenHAB como el monitor serie de Arduino. Una vez hecho esto, los resultados obtenidos fueron los presentados a continuación.

3.3.1 Prueba de 400 W

En esta prueba se ha puesto el dispositivo a trabajar a 400 W y los resultados obtenidos por el medidor de corriente en Arduino IDE fueron los siguientes (ver Figura 10):

```

Irms: 2.07
Consumo: 464.82
Irms: 2.02
Consumo: 497.97
Irms: 2.17
Consumo: 476.02
Irms: 2.07
Consumo: 476.02
Irms: 2.07
Consumo: 480.62
Irms: 2.09
Consumo: 494.42
Irms: 2.15
Consumo: 484.28
Irms: 2.11
    
```

Figura 10: Prueba de 400 Wattios

3.3.2 Prueba de 800 W

La segunda prueba utilizó el mismo dispositivo, esta vez a una potencia de 800 W según las especificaciones del fabricante. Los resultados se presentan en la Figura 11.

Irms: 3.66
Consumo: 811.33
Irms: 3.53
Consumo: 804.96
Irms: 3.50
Consumo: 833.98
Irms: 3.63
Consumo: 857.87
Irms: 3.73
Consumo: 849.12
Irms: 3.69
Consumo: 851.01
Irms: 3.70
Consumo: 841.60
Irms: 3.66

Figura 11: Prueba de 800 Wattios

Finalmente, a la vista de los resultados en las diferentes pruebas, se ha comprobado que el dispositivo construido consigue unos resultados satisfactorios.

Este dispositivo puede utilizarse para comprobar que los electrodomésticos consumen lo que indican las especificaciones del fabricante, además para medir variaciones en el consumo en función de diferentes factores, como las condiciones de uso (por ejemplo, el consumo de una heladera varía según la temperatura seleccionada y la frecuencia de apertura de las puertas).

3.4. Prueba de conexión remota al sitio web

Otra prueba consistió en verificar la conexión remota al sistema desde un dispositivo móvil mediante el sitio web <https://myopenhab.org/>. [11]

Una vez realizada la conexión correctamente, se puede seleccionar una de las tres interfaces disponibles, en este caso, "Basic UI". Esta interfaz permite acceder de manera remota a la plataforma OpenHAB y controlar los dispositivos desde cualquier ubicación.

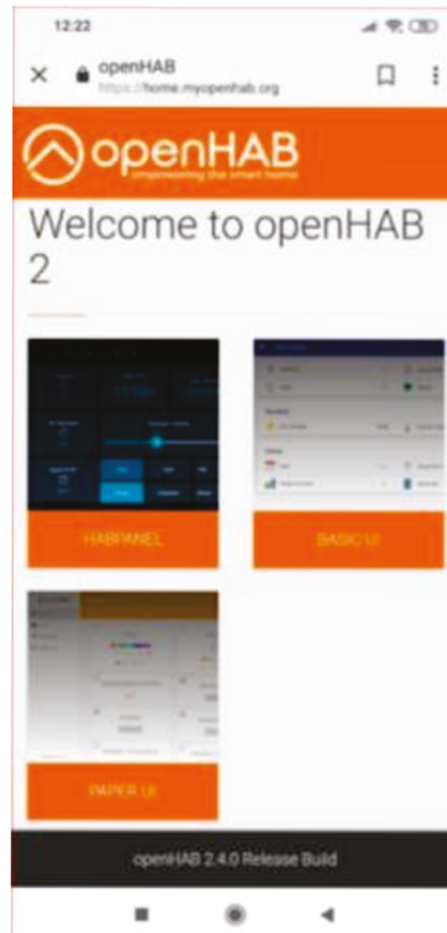


Figura 12: Interfaz MyOpenHAB Basic UI

Esta prueba de conexión remota demostró que el sistema funciona correctamente, permitiendo al usuario monitorear y gestionar el estado de los dispositivos a distancia y en tiempo real, lo cual es crucial para su operatividad. (Ver Figura 12).

4. Conclusiones y futuras líneas de investigación

4.1. Conclusiones

Desde el punto de vista electrónico, uno de los principales desafíos del proyecto fue el cálculo y acondicionamiento del sensor de corriente. Luego de algunas pruebas realizadas y recurriendo a bibliografía específica, se logró la correcta calibración y puesta a punto de los módulos del sistema.

Por otra parte, se implementaron aplicaciones innovadoras y herramientas informáticas vinculadas al Internet de las cosas (IoT, por su sigla en inglés) y la domótica para el almacenamiento y procesamiento de datos.

Debido a la crisis económica por la que atraviesa el país, la adquisición de los componentes electrónicos se facilitó a través de plataformas de e-commerce como Mercado Libre.

El sistema basado en OpenHAB se implementó exitosamente, cumpliendo con todos los requisitos funcionales establecidos. Todos los módulos de *hardware* y *software* funcionan e interactúan de manera óptima, conforme a los objetivos establecidos en el proyecto.

4.2. Valoración personal

Este proyecto forma parte de un largo trayecto de aprendizaje continuo, y un importante desafío tanto personal como profesional. Todo junto, permitió cumplir los objetivos propuestos después de varios meses de esfuerzo.

La mayor satisfacción es haber finalizado y concluido el proyecto con un sistema completamente funcional.

5.3. Líneas futuras de investigación

5.3.1 Autonomía

Uno de los principales problemas de todos los dispositivos móviles es la autonomía de las baterías, que tienden a limitar su funcionamiento. En este proyecto, los dispositivos se alimentaron mediante baterías externas. Una posible mejora puede ser, incorporar un módulo autónomo de batería que permita tanto alimentar los dispositivos como recargar las baterías, aumentando su autonomía y reduciendo la necesidad de mantenimiento frecuente.

Uno de los principales problemas de los dispositivos eléctricos móviles, es la durabilidad de las baterías. En este proyecto los dispositivos se alimentaron con una batería externa. Como se ha podido ver a lo largo del proyecto, esas baterías se deben cargar o remplazarse por otra cada cierto tiempo.

5.3.2. Tecnología para IoT

Siempre que se habla de redes inalámbricas. Tal vez las tecnologías más usadas a nivel mundial sean, el tradicional Wi-Fi, Bluetooth, 3G, 4G o 5G. En este sentido, el desarrollo del IoT ha impulsado nuevas tecnologías de redes inalámbricas más allá de las mencionadas. Estas nuevas redes emergentes, que utilizan la "banda blanca" liberada por la televisión digital terrestre, implementan soluciones de acceso IoT en áreas extensas.

Dependiendo de la aplicación, los factores como el alcance, velocidad de transferencia, seguridad, potencia y autonomía determinarán cuál es la mejor alternativa a la hora de elegir una red inalámbrica u otra. Por tanto, se sugiere estudiar el empleo de algunas de estas tecnologías para una posible mejora y evolución de los dispositivos del proyecto, especialmente en áreas donde la cobertura Wi-Fi sea insuficiente.

Referencias

- [1] D. Molloy y J. Wiley, *Raspberry Pi a fondo para desarrolladores*, Barcelona, España: Marcombo, 2019, p. 45.
- [2] Secretaría de Energía, "La Secretaría de Energía publicó el último Informe Trimestral de 2022," 2022. [En línea] Disponible en: [https://www.argentina.gob.ar/noticias/la-secretaria-de-energia-publico-el-ultimo-informe-trimestral-de-2022#:~:text=De%20esta%20manera%2C%20se%20registr%C3%B3,SA-DI%20fue%20de%2042.925MW](https://www.argentina.gob.ar/noticias/la-secretaria-de-energia-publico-el-ultimo-informe-trimestral-de-2022#:~:text=De%20esta%20manera%2C%20se%20registr%C3%B3,SA-DI%20fue%20de%2042.925MW.). [Accedido: septiembre, 2023].
- [3] CAMESA, "Información Anual 2020," 2022. [En línea]. Disponible en: <https://microfe.cammesa.com/static-content/CammesaWeb/download-manager-files/Informe%20Anual/INFORME%20ANUAL%202020%20VF.pdf>. [Accedido: septiembre, 2023].
- [4] J. Sutherland, *Scrum: El arte de hacer el doble de trabajo en la mitad de tiempo*, Nueva York, NY, USA: Crown Business, 2014, p. 5.
- [5] E. López, *Raspberry Pi. Fundamentos y aplicaciones*, Madrid, España: Ra-Ma, 2018, p. 26.
- [6] J. Núñez, *Arduino: Guía práctica para principiantes*, Barcelona, España: Marcombo, 2017, p. 12.
- [7] Apache.org, "Apache HTTP Server Project." [En línea]. Disponible en: <https://www.apache.org/>.
- [8] A. Silberschatz, H. F. Korth, y S. Sudarshan, *Fundamentos de Base de Datos, 7ª ed.*, Nueva York, NY, USA: McGraw-Hill Education, 2019, p. 4.
- [9] P. Dubois, *MySQL, 2ª ed.*, Sebastopol, CA, USA: O'Reilly Media, 2008, p. 7.
- [10] OpenHAB, "¿Qué es OpenHAB?," [En línea]. Disponible en: <https://www.openhab.org/docs/>. Accedido: [fecha de acceso].
- [11] OpenHAB, "MyOpenHAB," [En línea]. Disponible en: <https://myopenhab.org/>.