

# IoT-based energy monitoring system, to deploy an AMI in a Micro Smart Grid environment.

Gonzalo Di Paola<sup>1</sup>, Tomás Suarez<sup>1</sup> Gustavo Mercado<sup>2</sup>, Ana Lattuca<sup>1</sup>.

<sup>2</sup> Cátedra Proyecto Final

Departamento de Electrónica / Facultad Regional Mendoza / UTN  
gonzaloopez2206@gmail.com

<sup>1</sup>gridTICs – Grupo en Tecnologías de la Información y las Comunicaciones  
Departamento de Electrónica / Facultad Regional Mendoza / UTN  
gmercado@frm.utn.edu.ar

## Resumen

Desde un contexto global, la red eléctrica inteligente (o REI; smart grid en inglés) se puede definir como la integración dinámica de los desarrollos en ingeniería eléctrica, almacenamiento energético y los avances de las tecnologías de la información y comunicación (o TIC), dentro del negocio de la energía eléctrica (generación, transmisión, distribución, almacenamiento y comercialización, incluyendo las energías alternativas)

El concepto de “micro smart grid” se diferencia de la “smart grid” en el concepto del alcance de la administración de la red, mientras una smart grid supone una sistema de distribución de energía con control administrativo público, una micro smart grid, es un sistema cerrado e interno a una entidad privada, que pretende administrar la energía de acuerdo a criterios y normas propias de la institución.

Por otro lado un AMI (Advanced Metering Infrastructure) es un sistema de comunicación bidireccional para recopilar información detallada de medición en todo el sector de servicios de una micro smart grid. La AMI suele estar automatizada y permite realizar consultas en tiempo real y a petición con los puntos finales de medición.

En este proyecto se pretende diseñar e implementar un sistema de registro de energía basado en IoT( Internet of Thing), con el que se pueda desplegar una AMI.

**Palabras Claves:** Smart Grid, AMI, Internet de las Cosas.

## Contexto

El presente trabajo está inserto en el proyecto de investigación acreditado por la UTN código ENUTIME0005424TC “*Smart Micro Grid de Campus Universitario - Desarrollo, implementación y prueba.*” y al Proyecto Final de Grado, denominado “*Micro Smart Grid*”

El proyecto es llevado adelante por el grupo gridTICs y la Cátedra de Proyecto Final,. ambos del Departamento de Electrónica de la Facultad Regional Mendoza de la Universidad Tecnológica Nacional.

## 1. INTRODUCCIÓN

Desde un contexto global, la red eléctrica inteligente (o REI; smart grid en inglés) [1] se puede definir como la integración dinámica de los desarrollos en ingeniería eléctrica, almacenamiento energético y los avances de las tecnologías de la información y comunicación (o TIC), dentro del negocio de la energía eléctrica (generación, transmisión, distribución, almacenamiento y comercialización, incluyendo las energías alternativas); permitiendo que las áreas de coordinación de protecciones, control, instrumentación, medida, calidad y administración de energía, etc., sean concatenadas en un solo sistema de gestión con el objetivo primordial de realizar un uso eficiente y racional de la energía. El término red inteligente se asocia a menudo con el concepto de medidores inteligentes capaces de ofrecer una facturación detallada por franjas horarias, lo que permitiría a los consumidores no solo el elegir las mejores tarifas de entre las diferentes empresas eléctricas, sino también discernir entre las horas de consumo, lo que a su vez permitiría un mejor uso de la red. Este sistema también permitiría mapear con más

precisión el consumo y anticipar mejor las necesidades futuras a nivel más local. La irrupción de las energías renovables en el panorama energético ha cambiado notablemente los flujos de energía en la red eléctrica: ahora los usuarios no sólo consumen, sino que también producen electricidad a través de la misma red. Por tanto, el flujo de energía es ahora bidireccional. Una red inteligente envía electricidad desde los proveedores a los consumidores y viceversa, usando una tecnología digital para controlar las necesidades del consumidor. El concepto de “micro smart grid” [2] se diferencia de la “smart grid” en el concepto del alcance de la administración de la red, mientras una smart grid supone un sistema de distribución de energía con control administrativo público, una micro smart grid, es un sistema cerrado e interno a una entidad privada, que pretende administrar la energía de acuerdo a criterios y normas propias de la institución.

Por otro lado un AMI (Advanced Metering Infrastructure) [3] es un sistema de comunicación bidireccional para recopilar información detallada de medición en todo el sector de servicios de una micro smart grid. La AMI suele estar automatizada y permite realizar consultas en tiempo real y a petición con los puntos finales de medición.

En este proyecto se pretende diseñar e implementar un sistema de registro de energía basado en IoT( Internet of Thing), con el que se pueda desplegar una AMI.

### **Antecedentes**

Este proyecto es una continuación del proyecto FONARSEC 2013. Denominación: "Red inteligente ciudad Gral San Martín Mendoza", Proyecto FITS Energía 2013 - UREE. financiado por el Fondo sectorial de energía. Participantes: CAPP (EDEST, EMESA y UTN FRM), con vigencia desde el 1/1/2015 [4][5][6].

## **3. OBJETIVOS Y AVANCES DEL PROYECTO**

### **Objetivo Principal**

Se busca, a largo plazo proveer, una herramienta asequible que permita la

escalabilidad de las micro smart grid.

### **Objetivos específicos**

Con este prototipo se busca validar los conceptos:

- La información entregada por el sistema tiene potencial de acoplarse con un sistema de toma de decisiones que permite ahorrar energía en el edificio.
- El servicio tiene como costo una fracción de la energía eléctrica ahorrada.
- Integrar la herramienta AMI a un sistema general de Micro Smart Grid

## **Avances del Proyecto**

### **Alcance**

Se obtendrán datos de dos circuitos eléctricos y se mostrará esa información en una plataforma web interactiva.

### **Requisitos de Alto Nivel**

El siguiente listado presenta los requisitos que el resultado del proyecto (producto o servicio) debe cumplir como condición necesaria.

1. Otorgar datos en tiempo real de al menos un dispositivo o red eléctrica
2. Almacenar datos de consumo históricos de cada edificio.
3. Implementar una interfaz amigable para el usuario donde se visualicen los datos.

### **Desarrollo**

### **Resumen**

El fin del proyecto es mostrar de manera clara en un panel tipo dashboard un resumen de los datos obtenidos por un medidor de energía eléctrica en una fecha determinada.

En la Figura 1 se muestra un esquema con los módulos que conforman al proyecto

## **MÓDULO HARDWARE**

### **Medidor de Energía Eléctrica**

Se adquirió un medidor de la marca SolverBox [7]. Este medidor posee la capacidad de medir: Potencia activa, Corriente RMS, THD, Nivel de armónicos hasta el 7mo y energía acumulada. Estos datos fueron considerados como suficientes para una correcta medición de la **calidad de energía**.

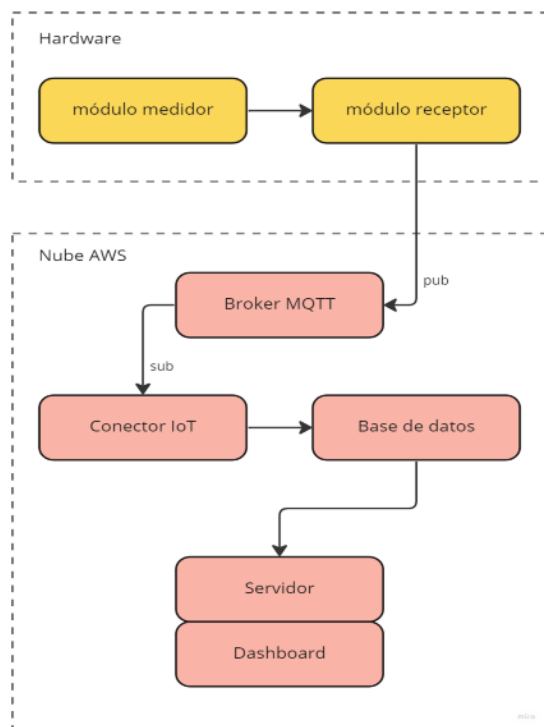


Figura 1: Esquema de AMI

Este dispositivo obtiene datos de corriente/tensión eléctrica a través de pinzas amperométricas no invasivas, que tienen una capacidad de medición de hasta 100 A

Las mediciones obtenidas son enviadas al módulo procesador mediante un puerto de comunicaciones serial (UART), con un formato de una trama tipo String.

### **Módulo de Adquisición y Transmisión de Datos: NODEMCU 1.0 - ESP8266**

NodeMCU 1.0 [8] es una placa de desarrollo basada en el chip ESP8266, con un microcontrolador de bajo costo, bajo consumo de energía y con capacidades Wi-Fi.

NodeMCU 1.0 incluye el chip ESP8266 y una serie de componentes adicionales, memoria flash, EEPROM, puertos USB y GPIO.. Este módulo se programó en C++ utilizando el IDE PlatformIO [9], integrado en el editor de texto Visual Studio code.

La función del nodo es obtener la trama de datos, que es provista por el medidor de energía, procesarla y enviarla a través de la red Wi-Fi hacia un broker MQTT. El procesamiento consiste en identificar y organizar los datos de manera que sea apropiados para enviarlos al broker y adicionalmente agrega la fecha en que se

adquirieron los datos.

## **MÓDULO SOFTWARE**

### **Criterios, limitaciones y desafíos:**

A la hora de elegir un protocolo de comunicación, se contempló la posibilidad de usar CoAP (Constrained Application Protocol) [10]. Es un protocolo de aplicación diseñado específicamente para aplicaciones IoT y dispositivos restringidos. Es muy similar a HTTP y, por lo tanto, es fácil de implementar en dispositivos con limitaciones de recursos. CoAP es eficiente en términos de ancho de banda y es adecuado para aplicaciones en las que la frecuencia de actualización es alta.

Finalmente se eligió usar MQTT [11] debido a la escalabilidad y la facilidad de desarrollo, pues al ser muy utilizado, tiene mucha documentación en internet. MQTT es un protocolo altamente escalable, debido a su arquitectura basada en publicador-suscriptor y su capacidad para manejar una gran cantidad de dispositivos conectados de manera eficiente. El protocolo MQTT permite a los clientes publicar mensajes en un tópico y a los servidores responder a esos mensajes, lo que lo hace adecuado para aplicaciones con grandes cantidades de datos y dispositivos.

También se optó por construir un broker MQTT propio en vez de contratar un broker privado, ya que hacerlo de esta manera, nos brindó libertad y control sobre el broker.

### **Módulo Broker MQTT**

En un sistema de comunicación basado en MQTT, el broker actúa como un intermediario confiable y eficiente entre los dispositivos publicadores y suscriptores.

Los publicadores envían mensajes a un tópico específico, mientras que los suscriptores se registran para recibir mensajes de ese tópico. Para este proyecto se optó por desarrollar un broker MQTT propio. El mismo fue programado en Express.js utilizando la librería “mosca”, que permite correr un broker en un puerto específico.

Express.js está escrito en JavaScript, y utiliza la programación asíncrona y “event-driven” de JavaScript para crear aplicaciones web rápidas y escalables.

## Módulo Conector IoT: MQTT to MongoDB

Este módulo es responsable de recibir los mensajes que llegan a un tópico específico en el broker MQTT. El módulo toma los datos, los formatea según un formato especificado, y finalmente los guarda en la base de datos MongoDB.

Está programado en node.js utilizando la librería “mqtt”, la cual tiene amplia documentación y ejemplos. Corre sobre el mismo servidor que el broker MQTT en la nube de AWS.

## Módulo Base de Datos

MongoDB [12] es una base de datos NoSQL (sin esquema) de tipo documental. Esto significa que en lugar de almacenar datos en tablas con filas y columnas como lo hacen las bases de datos relacionales, MongoDB almacena datos en documentos en un formato similar a JSON, donde cada documento es un registro independiente con su propio conjunto de campos y valores.

En el caso del proyecto se utilizó la siguiente estructura de datos para ser almacenada:

```
const measureSchema = mongoose.Schema({
  measure: String,
  date: Date
});
```

Esto permite guardar la trama recibida por el suscriptor y la fecha en la cual fue adquirida sin mayores complicaciones a la hora de tener un manejo de datos eficiente.

A continuación se muestra un ejemplo de los datos almacenados en la base de datos:

```
{
  "_id": ObjectId("63e40058fb8c410bfd8799a8"),
  "measure": "r,250.00,27.88,41.00,32.73,2.79,2.37,1.95,1.53,1.12,0.70,0.28,0.13,12...",
  "date": "2023-02-08T17:04:39.000+00:00",
  "__v": 0
}
```

Para el manejo de datos en mongodb se utilizó la librería mongoose.

Mongoose es una biblioteca de modelado de objetos de MongoDB para Node.js. Es un enrutador que actúa como una capa intermediaria entre la aplicación y la base de datos MongoDB, permitiendo interactuar con la base de datos a través de objetos en lugar de

tener que escribir manualmente código de consulta a la base de datos.

## Panel DashBoard

En el contexto de sistemas de monitorización de consumo eléctrico, el dashboard es una herramienta que permite visualizar los datos adquiridos por los medidores en tiempo real y de forma organizada. El objetivo principal es proporcionar una representación clara y concisa del consumo eléctrico, con el fin de permitir a los usuarios expertos tomar decisiones informadas basadas en la información proporcionada.

El dashboard recopila, procesa y muestra los datos adquiridos, permitiendo a los usuarios expertos analizar y comparar los patrones de consumo eléctrico a lo largo del tiempo, identificar tendencias y patrones anómalos, y ajustar la gestión de la energía en consecuencia.

## BackEnd

El Backend es la parte de una aplicación web que se encarga de las tareas y operaciones que ocurren en el servidor. Este es el encargado de procesar y almacenar datos, y de ofrecer servicios a la parte del cliente (frontend) de la aplicación.

Express [13] es una popular librería de Node.js que se utiliza para crear aplicaciones web y servicios API en el lado del servidor.

Además es común que en el backend se cree una API para comunicar el frontend con la base de datos.

La API desarrollada permite comunicar el Frontend con la base de datos. El Frontend realiza una petición GET el cuál envía a una ruta específica 4 parámetros importantes: Las fechas de inicio y fin, el mes y el piso del cual se quieren obtener los datos.

Con estos datos podemos realizar la consulta a la base de datos para obtener los datos relacionados con las fechas relacionadas. El formato de la fecha recibido es convertido al formato 2023-02-08T17:00:00.000+00:00 según la norma ISO 8601 [14].

Todos estos datos son agregados a un objeto JSON y es enviado al Frontend.

## FrontEnd

Del lado del Frontend tenemos la página web desarrollada la cual muestra datos procesados por el servidor. Estos son mostrados en distintas gráficas de forma tal que pueda sacar una rápida conclusión sobre lo que está ocurriendo en el edificio bajo medición.

Los datos presentados son extraídos según las fechas que se indiquen en la página web. Estos incluyen la energía acumulada, el factor de distorsión armónica total (THD), la potencia reactiva, el coseno phi, los armónicos del 1 al 7 con sus respectivos valores en un gráfico de barras.

La página también presenta información sobre la potencia activa, la corriente rms y la corriente pico en tiempo real.

Además de presentar información visual sobre la energía eléctrica, la página web desarrollada en React, también se comunica con un backend desarrollado en Node.js. La comunicación entre la página web y el backend se realiza mediante la utilización de la biblioteca Axios.

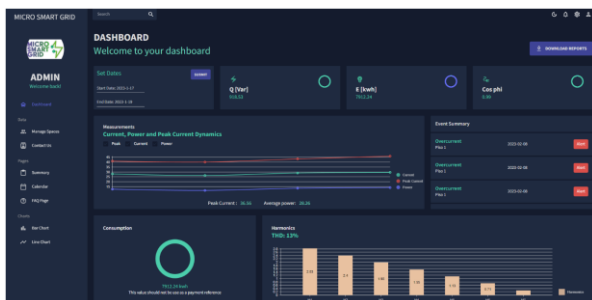


Figura 2: Front End

Axios es un cliente HTTP que permite hacer solicitudes desde el lado del cliente. Esto permite que la página web obtenga los datos necesarios del backend de manera dinámica, actualizando en tiempo real la información presentada en la página. La combinación de React para el desarrollo de la interfaz y Node.js para el backend, junto con Axios para la comunicación, brinda una solución completa y eficiente para la visualización y monitoreo de datos.

#### 4. FORMACIÓN DE RECURSOS HUMANOS

Uno de los principales objetivos del proyecto es la capacitación de los recursos humanos.

La meta como investigadores es fortalecer la capacidad para realizar investigación

científica, generar conocimientos y facilitar la transferencia de tecnología que permita el desarrollo humano.

Para lograr estos objetivos se dispuso del siguiente personal:

Un Investigadores formado  
Un Investigador de apoyo  
Dos alumnos en Proyecto Final de Carrera

#### 5. BIBLIOGRAFÍA

- [1] Shengwei Mei, et al, "Paving the Way to Smart Micro Energy Grid: Concepts, Design Principles, and Engineering Practices" IEEE © 2017 CSEE
- [2] Padmanaban, Sanjeevikumar, et al. Microgrids. CRC Press, 2020.
- [3] A. K. Chakraborty and N. Sharma, "Advanced metering infrastructure: Technology and challenges," 2016 IEEE/PES Transmission and Distribution Conference and Exposition (T&D), Dallas, TX, 2016.
- [4] G. Mercado et al "SG-SM - Smart Grid San Martín. Red de Distribución y Generación de Energía Inteligente en Ciudad Gral San Martín Mendoza" WICC 2015, Salta 16-17, Abril 2015
- [5] G. Mercado, J. Da Peña, G. Vivone, M. Ledda, R. Caceres, C. Taffernaberry, S. Pérez, L. Álvarez, A. Arena, G. Henderson, M. Funes, J. Catalá, "Diseño, Implementación y Análisis de Red de Distribución Inteligente y Generación Distribuida de Energía", Anales de CIIS 2016, May 2016, Universidad de Córdoba, Córdoba.
- [6] A. P. Arena, L. Álvarez, J. Da Peña, R. Stasi, G. López, A. Burlot, L. Barnabo, M. Ledda, R. Caceres, C. Taffernaberry and G. Mercado, "Diseño, Implementación y Análisis de una Red Eléctrica Inteligente (Smart Grid)", Anales del ENIDI 2017, Mendoza 22 al 24 de Noviembre 2017
- [7] Solvers Box Medidores de Energía Eléctrica, <https://solversbox.com/> visitado 10-Noviembre-2022
- [8] Node MCU 1.0 [docs.platformio.org/en/stable/boards/espressif8266/nodemcu-v2.html](https://docs.platformio.org/en/stable/boards/espressif8266/nodemcu-v2.html) visitado 10-Noviembre-2022
- [9] PlatformIO - Professional collaborative platform for embedded development. [platformio.org/](https://platformio.org/) visitado 20-Nov-2022
- [10] Joshi, Manveer, and Bikram Pal Kaur. "Coap protocol for constrained networks." International journal of wireless and microwave technologies 5.6 (2015): 1-10.
- [11] Soni, Dipa, and Ashwin Makwana. "A survey on mqtt: a protocol of internet of things (iot)." International conference on telecom, power analysis and computing (ICTPACT-2017). Vol. 20. 2017.
- [12] MongoDB Atlas, [www.mongodb.com/](https://www.mongodb.com/) visitado 12-Oct-2022
- [13] Express Infraestructura web para Node.js <https://expressjs.com/es/> visitado 6-Oct-2022
- [14] Houston, Gary. "ISO 8601: 1988 Date/Time Representations." (1993).