

Universidad Tecnológica Nacional

Proyecto Final

Medición distribuida de energía.

Autores:

- Catinot, Francisco.
- Ludi, Marino.
- Weber, Adrián.

Director:

- *Burgos, Sergio.*

*Proyecto final presentado para cumplimentar los requisitos académicos
para acceder al título de Ingeniero ELECTRÓNICO*

en la

Facultad Regional Paraná

Julio 2018

Declaración de autoría:

Nosotros declaramos que el Proyecto Final “Medición distribuida de energía” y el trabajo realizado son propios. Declaramos:

- Este trabajo fue realizado en su totalidad, o principalmente, para acceder al título de grado de Ingeniero Electrónico, en la Universidad Tecnológica Nacional, Regional Paraná.
- Se establece claramente que el desarrollo realizado y el informe que lo acompaña no han sido previamente utilizados para acceder a otro título de grado o pre-grado.
- Siempre que se ha utilizado trabajo de otros autores, el mismo ha sido correctamente citado. El resto del trabajo es de autoría propia.
- Se ha indicado y agradecido correctamente a todos aquellos que han colaborado con el presente trabajo.
- Cuando el trabajo forma parte de un trabajo de mayores dimensiones donde han participado otras personas, se ha indicado claramente el alcance del trabajo realizado.

Firmas:

-
-
-

Fecha:

Agradecimientos:

En los años que ha durado este proceso de aprendizaje y preparación profesional, tenemos que agradecer a cada una de las personas que han estado a nuestro lado, mostrando apoyo, confianza y ayuda.

En especial, tenemos que agradecer el constante apoyo de nuestras familias.

A los docentes de la UTN Facultad Regional Paraná y compañeros que han compartido nuestro camino.

A la sociedad en general por brindarnos la posibilidad de estudiar en una universidad pública.

GRACIAS.

Catinot, Francisco.

Ludi, Marino.

Weber, Adrián.

Universidad Tecnológica Nacional

Abstract

Facultad Regional Paraná

Ingeniero en Electrónica

Medición distribuida de energía.

Catinot, Francisco.

Ludi, Marino.

Weber, Adrián.

Abstract:

The project consisted in developing a system that allows the analysis of individual energy consumption data of several devices and generates, based on their interpretation, statistics that help users to reduce consumption, either through the modification of habits or by the replacement of equipment for others of better performance.

The system is made up of specific measuring equipment to collect the information and a data concentrator that will be communicated with the measurement equipment, process the information and send it to a web server that will interface with the user.

The development of the project consisted of several stages. The first and most important was the energy measurement; for this an CIRRUS LOGIC integrated circuit was used. This equipment is able to measure voltage, current, power, cosine fi and harmonic components, both in its average and instantaneous values.

Another important stage of the project was the communication between the point meters and the data concentrator. Each point meter has an ESP8266 module, which was programmed with open and free language.

The data concentrator was made with a Raspberry Pi. This is a minicomputer of high versatility and low cost. On the other hand, the communication between the meters and the data concentrator was made through a protocol called MQTT, known for its excellence in IoT.

The user interface was another relevant stage. The project contemplated the programming of a web through an open source HTTP web server called APACHE, so that it was possible to access from any terminal or PC, resulting in clear and precise information.

Finally, for the aesthetic aspect of the project, it was decided to use high quality modern carcasses manufactured in our country.

The final product was contrasted in the meters laboratory of the Provincial Energy Company of the city of Santa Fe, which is enabled by INTI.

Keywords:

ESP8266, IoT (Internet of Things), MQTT (Message Queue Telemetry Transport), Raspberry Pi, Telemetric.

Resumen:

El proyecto consistió en desarrollar un sistema que permita analizar los datos de consumo energético individual de varios equipos y genere, en base a su interpretación, estadísticas que ayuden a los usuarios a reducir el consumo energético, ya sea por medio de la modificación de hábitos o por el cambio de equipos por otros de mejor rendimiento.

El sistema está compuesto por equipos de medición puntual que serán los encargados de recaudar la información, y un concentrador de datos que se comunicará con los equipos de medición y procesa la información, para luego representarla a través de una web que oficiará de interfaz con el usuario.

La realización del proyecto consto de varias etapas, la primera y la más importante es la de medición de energía y los parámetros asociados al consumo, para lo que se utilizó un circuito integrado de CIRRUS LOGIC, el mismo es capaz de medir tensión, corriente, potencia, coseno fi y componentes armónicos, tanto en sus valores promedios como instantáneos.

Otra etapa importante del proyecto fue la comunicación entre los medidores distribuidos y el concentrador de datos. Cada medidor puntual cuenta con un módulo ESP8266, que fue programado con un lenguaje de programación abierto y gratuito.

El concentrador de datos se realizó con una Raspberry Pi. Ésta es una minicomputadora de mucha versatilidad y bajo costo.

Por otra parte, la comunicación entre los medidores y el concentrador de datos se realizó a través de un protocolo llamado MQTT, conocido por sus antecedentes en IoT.

El interfaz de usuario, fue otra etapa relevante en el proyecto, donde se optó por desarrollar una web, a través de un servidor web HTTP de código abierto (Apache), de manera de que sea posible acceder desde cualquier terminal o PC, teniendo información clara y precisa.

Por último, en lo que refiere a la estética del proyecto, se utilizaron carcazas para los equipos con terminaciones modernas, de alta calidad y fabricada en nuestro país.

El producto final fue contrastado en el laboratorio de medidores de la empresa provincial de la energía de Santa Fe, el cual se encuentra habilitado por INTI.

Palabras Clave:

ESP8266, IoT (Internet of Things), MQTT (Message Queue Telemetry Transport), Raspberry Pi, Telemetría.

Reconocimientos:

A nuestro Guía en este proyecto, Sergio Burgos, por sus consejos, su desinteresada ayuda y su predisposición constante.

A Emanuel Gonzales y el personal del laboratorio de medidores de EPE, por abrirnos las puertas de su laboratorio y ponerlo a nuestra disposición, los consejos y asesoramientos brindados y el constante apoyo.

Índice:

Capítulo 1: Introducción	1
1.1 Motivación y Contexto.....	2
1.2: Fortalezas y Debilidades.....	4
Capítulo 2: Desarrollo	5
2.1 Diagramas en bloques.....	5
2.2 Medición de Energía con CI CS5463.....	6
2.2.1 Teoría de funcionamiento.....	7
2.2.2 Medición de Tensión y Corriente.....	9
2.2.3 Medición de Potencia.....	10
2.2.4 Velocidad de cálculo.	12
2.2.5 Comandos.....	13
2.2.6 Fundamentos de elección de este circuito integrado.	14
2.3 Transformador de corriente.....	15
2.4 NodeMCU.....	17
2.4.1 Principales especificaciones técnicas.	19
2.4.2 Programa empleado en NodeMCU.....	20
2.5 Arduino IDE.	21
2.6 Comunicación entre medidor de energía y el NodeMCU.....	22
2.7 MQTT (Message Queue Telemetry Transport).....	24
2.8 Funcionamiento, conexión, MQTT y medición.....	27
2.9 Calibración de los medidores de energía.....	29
2.10 Protocolo UDP.....	30
2.11 Datalogger o Concentrador de datos (Raspberry Pi).....	32
2.11.1 Puertos de comunicación de una Raspberry Pi.	33
2.11.2 Funcionamiento básico.	34
2.11.3 Inicio Raspberry Pi.	35
2.12 Mosquitto.	37
2.13 Instalación de Mosquitto en Raspberry Pi.....	38
2.14 Base de Datos.	42

2.15	Servidor Web.....	44
2.15.1	Instalación de Apache en la Raspberry Pi.....	45
2.15.2	Protocolo HTTP.....	46
2.15.3	Lenguaje HTML.....	48
2.15.4	Lenguaje PHP.....	50
2.15.5	Lenguaje JavaScript.....	52
2.15.6	Nuestra Web.....	53
2.16	Código QR.....	56
Capítulo 3:	Verificación de la calidad del instrumento.....	57
3.1:	Fotos del Laboratorio de Medidores de la Empresa Provincial de Energía de Santa Fe (EPE).....	58
Capítulo 4:	Hardware.....	61
4.1	Esquemático.....	61
4.2	Imágenes 3D.....	62
4.3	Hardware terminado.....	64
Capítulo 5:	Análisis de Costos.....	66
5.1	Análisis de costos por unidad.....	67
5.2	Análisis de costos por cantidad.....	68
5.3	Retorno sobre la inversión (ROI).....	69
5.5	Plan de marketing e inversión.....	70
5.7	Validación del plan de negocios.....	71
Capítulo 6:	Conclusión y líneas futuras.....	72
Literatura Citada.....		73

Lista de Ilustraciones:

Ilustración 1-Diagrama conceptual	2
Ilustración 2-Fortalezas y Debilidades.....	4
Ilustración 3-Diagrama en bloques.....	5
Ilustración 4-Diagrama de bloques funcionales de CS5463.....	6
Ilustración 5-Diagrama en bloques del Medidor de Energía.....	7
Ilustración 6- CR 8320-1600.....	15
Ilustración 7- Tensión vs Corriente TI	16
Ilustración 8-ESP8266.....	17
Ilustración 9- NodeMCU.	18
Ilustración 10-Estructura general del protocolo SPI.	22
Ilustración 11-Modos en SPI.....	23
Ilustración 12-Principio de funcionamiento protocolo MQTT.....	24
Ilustración 13-Paquete de Control MQTT.....	25
Ilustración 14-Estructura del paquete de control MQTT.....	25
Ilustración 15-QoS del protocolo MQTT.....	26
Ilustración 16-Segmento UDP	30
Ilustración 17-Raspberry Pi 3.	32
Ilustración 18-Inicializacion Raspberry Pi.....	35
Ilustración 19-Nivel de arranque Raspberry Pi.....	36
Ilustración 20-Diagrama en bloques inicio Raspberry Pi	36
Ilustración 21-Instalación Broker MQTT, paso 1.....	38
Ilustración 22-Instalación Broker MQTT, paso 2.....	39
Ilustración 23--Instalación Broker MQTT, paso 3.....	39
Ilustración 24- Instalación cliente MQTT, paso 1.....	40
Ilustración 25-Instalación cliente MQTT, paso 2.....	41
Ilustración 26-Pagina de inicio del administrador de base de datos de MariaDB.....	42
Ilustración 27-Captura de pantalla conteniendo la estructura de una de las tablas de registro.....	43
Ilustración 28-Captura de pantalla de los datos registrados en una de las tablas.....	43
Ilustración 29-Protocolo HTTP.....	47

Ilustración 30- En nuestro proyecto ver <html> <body> <head>	51
Ilustración 31-Nuestra web.....	53
Ilustración 32-Nuestra web, selección.....	54
Ilustración 33-Nuestra web, zona gráficas.....	55
Ilustración 34-Código QR, para acceder a la web UTN-FRP	56
Ilustración 35-Foto 1, Laboratorio de Medidores de EPE.....	58
Ilustración 36-Foto 2, Laboratorio de Medidores de EPE.....	59
Ilustración 37-Foto 3, Laboratorio de Medidores de EPE.....	59
Ilustración 38-Foto 4, Laboratorio de Medidores de EPE.....	60
Ilustración 39-Equipos aprobados por INTI.	60
Ilustración 40-3D superior derecha.	62
Ilustración 41-3D superior izquierda.....	62
Ilustración 42-3D superior.....	63
Ilustración 43-3D inferior.....	63
Ilustración 44- medidor puntual en funcionamiento.	64
Ilustración 45- Interior medidor.	64
Ilustración 46-Interior medidor.	65
Ilustración 47-Medidor lateral.	65
Ilustración 48-Superior Concentrador de datos.	66

Lista de Tablas:

Tabla 1-Especificación técnica de nodeMCU.	19
Tabla 2-Verificacion de la calidad del instrumento.....	58
Tabla 3-Costo por unidad Datalogger/Medidor puntual.	67
Tabla 4-Costo por cantidad Datalogger/Medidor puntual.....	68

Dedicado a:

Es difícil realizar una dedicatoria cuando en el camino recorrido hubo muchas personas e instituciones que ayudaron, directa o indirectamente, a lograr generar el contexto necesario para que nos sea posible conseguir este logro.

Es cierto que hemos realizado muchos esfuerzos y sacrificios para llegar donde hoy estamos, pero también lo es que no podríamos haberlo hecho solos, sin el apoyo de nuestros padres, parejas, hermanos, hermanas y nuestros grupos familiares en general.

Tampoco hubiese sido posible transitar este camino sin la buena predisposición y la vocación que tantos de nuestros docentes mostraron y con la que lograron enamorarnos de esta hermosa profesión, ni sin el lugar que muchas veces nos brindaron en nuestros trabajos, cubriéndonos para que podamos terminar algún trabajo práctico o simplemente mostrándonos las realidades del mercado a las que debemos adaptarnos para ser buenos profesionales.

Imprescindible fueron también los mates, los apuntes prestados y cada una de las charlas que compartimos con nuestros compañeros de curso a través de todos estos años. Algunos de ellos ya se han convertido en colegas, otros están en vías de hacerlo y también hay quienes decidieron cambiar el rumbo de su camino, pero sin dudas todos ellos contribuyeron a marcar el rumbo del nuestro.

Gracias a todos.

Capítulo 1: Introducción

El desarrollo tecnológico y la introducción de nuevos dispositivos en nuestras vidas cotidianas han generado una creciente dependencia por parte de toda la sociedad de suministros de energía eléctrica. Sin embargo, la generación de la misma, al menos en lo que respecta a formas tradicionales, no es para nada económico ni amigable con el medio ambiente. En nuestro país esta situación no representaba un problema hasta hace algunos años dado que no existía conciencia masiva con respecto al cuidado del medio ambiente y además el costo de la energía estaba fuertemente subsidiado con fondos nacionales.

Este escenario ha cambiado y hoy en día han surgido con mucha fuerza las llamadas fuentes de energía alternativa, brindando la posibilidad de producir energía reduciendo en forma considerable la contaminación asociada. Poco a poco comienza a ser cada vez más frecuente encontrarse, por ejemplo, con paneles solares colocados en los techos de las viviendas. Al mismo tiempo, esta iniciativa de origen ecológica, se vio favorecida por las recientes quitas de subsidios y el incremento significativo del costo de energía. Ahora reducir los consumos ya no representa un interés sólo de quienes desean colaborar en el cuidado del ambiente sino también de quienes buscan reducir los costos mensuales.

1.1 Motivación y Contexto.

¿Cómo se puede reducir el consumo si no se conoce qué lo genera? En la información se encuentra el poder, y es este el contexto, en que la idea de un equipamiento que pueda adaptarse a una vivienda y documente los consumos individuales de energía de los equipos conectados en ella toma fuerza.

El objetivo del proyecto fue desarrollar un sistema que permita analizar los datos de consumo energético individual de varios equipos y genere en base a su análisis, estadísticas que ayuden a los usuarios a reducir el consumo, ya sea por medio de la modificación de hábitos o por el cambio de equipos por otros de mejor rendimiento.

El sistema de la ilustración 1 compuesto por equipos de medición puntual que serán los encargados de recaudar la información y un concentrador de datos que se comunicará con ellos. Este último procesará la información, generando las interfaces adecuadas para su representación gráfica.

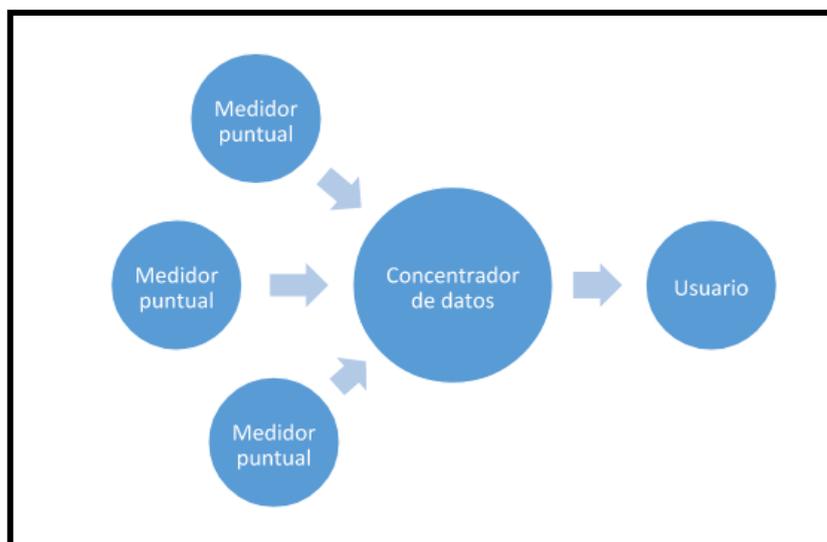


Ilustración 1-Diagrama conceptual

Los equipos de medición son compactos, con un diseño estético, desarrollados de forma tal que se conecten entre el toma corriente de la instalación y el enchufe del dispositivo a medir, como si fuese una especie de “adaptador”, de modo que conectarlos en una instalación preexistente no represente ningún tipo de dificultad y que cualquier persona pueda hacerlo.

Por su parte el concentrador tiene un aspecto similar al de un router y es único para todos los medidores. Es responsable de recibir los datos de todos los medidores, los procesará y generará la interfaz que oficiará de nexo con el usuario, a la que podrá accederse desde cualquier PC, Tablet o Smartphone. No brindará información al usuario (exceptuando indicadores de encendido y correcta conexión entre dispositivos), sino por medio de dispositivos externos a los que se lo vincule. Su instalación es también muy sencilla y rápida.

Las mediciones a realizarse serán: tensión, corriente, potencia activa, potencia reactiva, potencia aparente, factor de potencia y contenido armónico y la clase de los medidores deberá ser tal, que se encuadre en las exigidas por las normativas vigentes para los medidores de energía suministrados por las empresas distribuidoras de energía. A fin de que los resultados obtenidos mediante el uso del sistema de medición puntual, sean confiables y puedan usarse para estimar el costo económico asociado al consumo registrado.

El proyecto en cuestión, tiene como posibles interesados a toda la población, dado que todos son consumidores de energía, motivados por el ahorro o por la posibilidad de implementar fuentes renovables, están interesados en analizar sus consumos a fin de poder reducirlos.

El presente proyecto busca ser una herramienta útil para quienes buscan analizar al detalle el consumo de su instalación, y por lo tanto cubre las necesidades tanto de quienes intentan reducir los consumos con fines estrictamente económicos, como a quienes lo hacen con el fin de instalar fuentes de energía renovables.

Este proyecto tiene la particularidad que a nivel nacional existe solamente un producto sustituto, de la empresa *Wabee*, pero es medidor global de energía, es decir, un dispositivo que registra el consumo total de la instalación, sin discriminar por equipos. A diferencia de este, el presente desarrollo contará con un medidor por equipo, por lo que además de poder brindar la información asociada al consumo total (que surgirá de sumar los consumos individuales), permitirá analizar particularmente cada uno de ellos. *Wabee* además no se encuentra aún disponible en el mercado, está en fase de producción. [1]

A nivel internacional se encontraron varias empresas en el rubro. Se enumeran a continuación las más relevantes y sus principales características:

Efergy: ofrece un producto que mide el consumo de energía global de la instalación. Precio obtenido de la página oficial de la empresa €55 - €85.

Curb: producto de similares características que el de *Efergy* y *Wabee*, con la diferencia de que no solo permite saber el consumo total, sino que lo discrimina por cada una de las llaves termo magnéticas que sectorizan la instalación. En el mejor de los casos, si la instalación eléctrica de la casa esta sectorizada de esta manera, mostrará el consumo por habitación. Precio obtenido de la página oficial U\$S400.

Cliensol: desarrolló el “energomonitor”, es un producto con características similares a las buscadas en el presente proyecto, pero con una variedad de sensores a asociar al concentrador de datos mucho más amplia. Permiten censar consumos individuales de cada equipo, comunicarse con el medidor de la empresa proveedora de energía para conocer el consumo registrado por este, censar temperatura, agua y gas. Costo obtenido de la página oficial de la empresa de un pack compuesto por concentrador más 3 sensores de energía puntuales €300.

Plugwise: posee tanto medidores generales como un producto similar al que se desea desarrollar. Ofrecen un pack de concentrador más 9 medidores puntuales por €400 en su página oficial. [2]

1.2: Fortalezas y Debilidades.

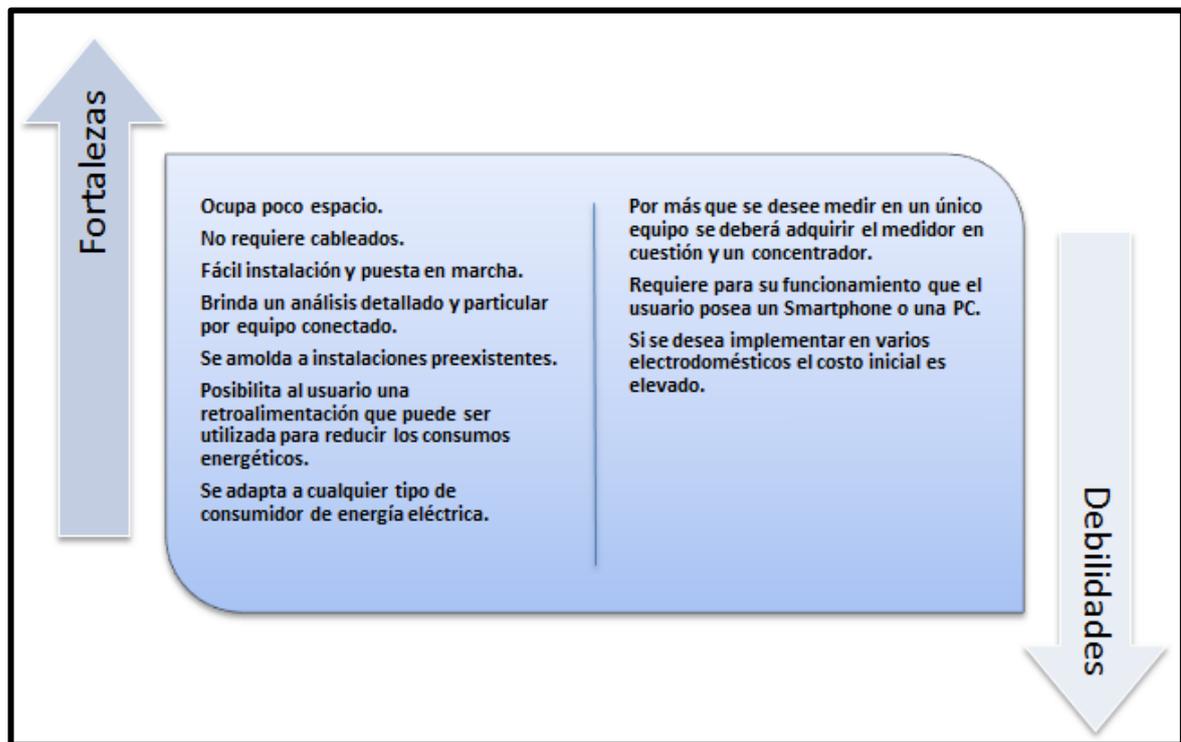


Ilustración 2-Fortalezas y Debilidades.

Capítulo 2: Desarrollo

2.1 Diagramas en bloques.

El sistema está compuesto por equipos de medición puntual de energía que serán los encargados de recaudar la información y un concentrador de datos que se comunicara con los equipos de medición y procesara la información para luego volcarla a un servidor web.

A lo largo de este capítulo vamos a ir describiendo el funcionamiento de cada una de las partes y el porqué de las distintas decisiones que se tomaron a la hora de diseñar y desarrollar el producto.

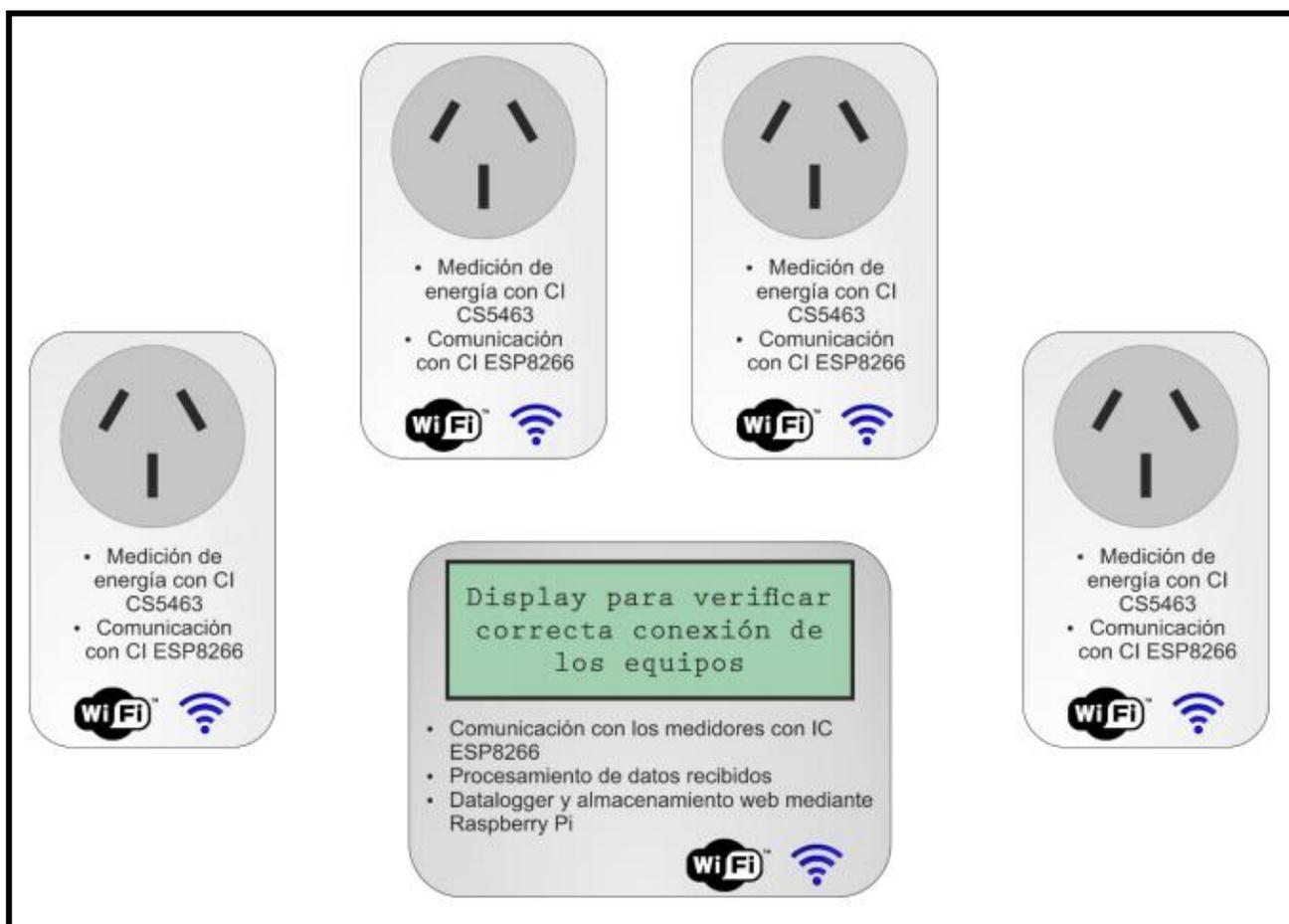


Ilustración 3-Diagrama en bloques.

Si bien en la ilustración 3 pueden apreciarse 4 medidores puntuales de energía, todos ellos funcionan de la misma manera y son idénticos en lo que a funcionamiento respecta. Se optó por representar esta cantidad con la única intención de mostrar en forma gráfica la posibilidad de conectar múltiples puntos de medición y adquisición de datos a un único concentrador. Cabe aclarar, además, que es posible utilizar más o menos de cuatro dispositivos y que esta determinación quedará en manos del cliente.

2.2 Medición de Energía con CI CS5463.

El corazón de nuestros medidores de energía es el circuito integrado CS5463, de la fábrica Cirrus Logic, cuyo esquema puede observarse en la ilustración 4.

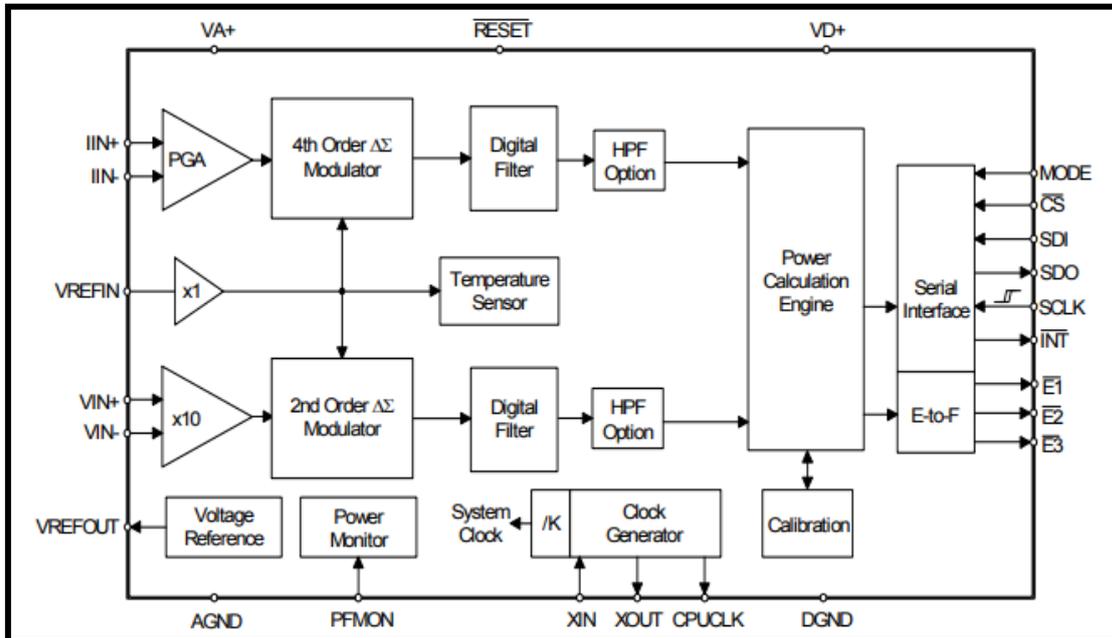


Ilustración 4-Diagrama de bloques funcionales de CS5463.

El CS5463, es un medidor de energía integrado, que combina dos convertidores analógicos-digital e interfaz de comunicación serial, en un solo chip. Está diseñado para medir con precisión corriente y tensión instantánea, calcula valores de tensión y corriente RMS, potencia activa, potencia reactiva y de calcular frecuencias y armónicos.

Las entradas de los conversores AD están optimizadas para interactuar con resistencias de derivación (shunt), transformadores de corriente o divisores resistivos.

Como se comentaba antes, posee una interfaz serie bidireccional para la comunicación con un procesador y una función programable de salida de energía a pulso que para el presente proyecto se utilizó con un diodo led a la salida, como interfaz de comunicación con el equipo de calibración y contraste.

2.2.1 Teoría de funcionamiento.

El CS5463 es un convertidor analógico-digital (ADC) de doble canal seguido por un motor de procesamiento, que realiza cálculos de potencia y energía. El flujo de datos para la medición de voltaje, corriente, así como los algoritmos de cálculo de potencia, se pueden observar en la hoja de datos que se adjuntara al final del informe. Utiliza dos canales debido a que realiza mediciones de corriente por un lado y de tensión por otro.

En la entrada de voltaje, se colocó un divisor resistivo con el fin de reducir la amplitud de la tensión de entrada a valores que puedan ser menores a los 250[mVp] que es el máximo admitido por el CI. Esta tensión ingresa en la entrada del canal $V_{IN\pm}$ y ya dentro del integrado se le aplica una ganancia de 10x para luego ser muestreada por un modulador delta-sigma de segundo orden.

Simultáneamente, la entrada de corriente es adaptada por un transformador de corriente que debe reducir la amplitud de la misma y convertirla en una señal de tensión menor a los 250[mVp] para que pueda ser aplicada a la entrada del canal $I_{IN\pm}$. Ya dentro de la pastilla está sujeta a dos posibles ganancias seleccionables del amplificador programable de ganancia (PGA) 10x o 50x, utilizándose la de 10x. La señal amplificada es muestreada por un modulador delta-sigma de cuarto orden para digitalización.

Ambos convertidores muestrean a una tasa de $MCLK / 8$ donde MCLK hace referencia a la frecuencia de clock principal que en este caso es un cristal de cuarzo de 4,096[MHz].

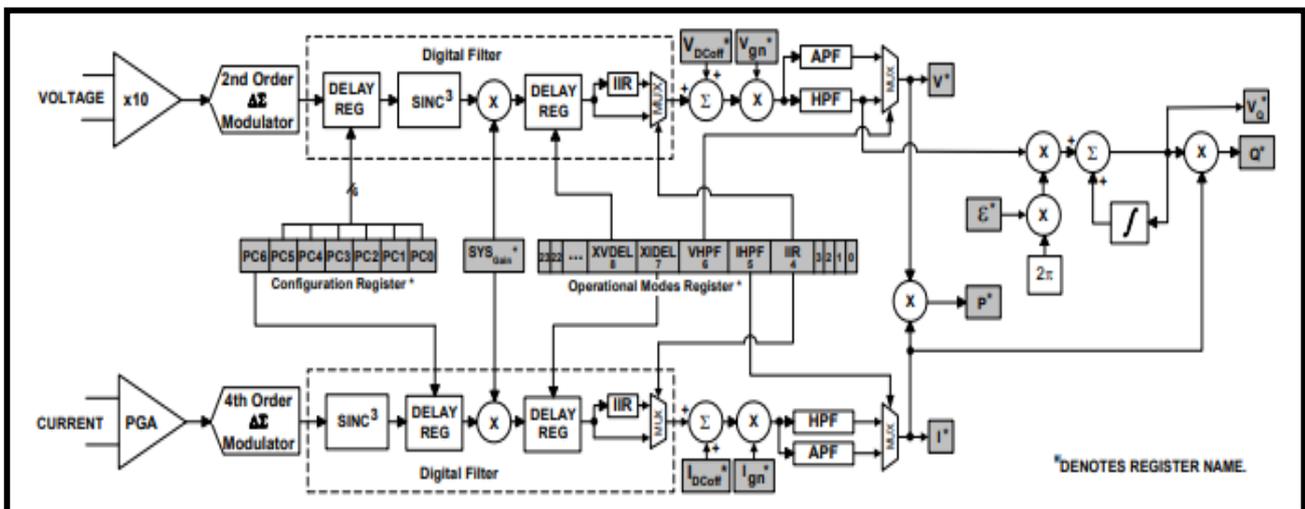


Ilustración 5-Diagrama en bloques del Medidor de Energía.

Los filtros digitales en ambos canales son sinc^3 , seguidos por filtros IIR de cuarto orden.

En ambos canales (tensión y corriente), existe la posibilidad de habilitar un filtro de paso alto digital (HPF) que quite cualquier componente de DC de la señal. Al eliminar el componente DC de tensión, cualquier contenido de DC también será eliminado de la potencia activa. En el presente proyecto se tomó la decisión de activarlo solamente en el canal de corriente, dado que el transformador de intensidad que adecua dicha señal no

permite el paso de componentes de continua y, por ende, cualquier presencia de ellas significaría ruido.

Cuando el HPF opcional en cualquiera de los canales está deshabilitado, el filtro pasa todo (APF) está implementado. El APF tiene una respuesta de amplitud que es plana dentro del ancho de banda del canal y se utiliza para hacer coincidir la fase de ambas señales cuando solo un HPF está activado.

2.2.2 Medición de Tensión y Corriente.

Los cálculos de Root Mean Square (RMS) se realizan con los valores de voltaje y corriente instantáneo, V_n e I_n respectivamente. Donde N es la cantidad de ciclos que se utilizan para generar un resultado y en el caso del presente proyecto se le asignó un valor de 4000 (valor que recomienda la hoja de datos).

$$I_{RMS} = \sqrt{\frac{\sum_{n=0}^{N-1} I_n^2}{N}}$$

Los resultados del cálculo de I_{RMS} y V_{RMS} pueden accederse por medio de la lectura de los registros homónimos y están disponibles cada N ciclos (4000).

La corriente máxima (I_{peak}) y la tensión máxima (V_{peak}) son accesibles por medio de lectura de registros y surgen de la mayor magnitud detectada durante el último ciclo computacional en la corriente y el voltaje instantáneos respectivamente.

2.2.3: Medición de Potencia.

2.2.3 Medición de Potencia.

La potencia instantánea se obtiene al multiplicar las muestras instantáneas de voltaje y corriente. El producto se promedia a través de N conversiones para calcular la potencia activa.

La potencia aparente (S) es la combinación de la potencia activa y potencia reactiva, sin referencia al ángulo de fase generado por alguna impedancia, y se calcula usando la siguiente fórmula:

$$S = V_{RMS} \times I_{RMS}$$

El Factor de potencia (PF) es la potencia activa dividida por la potencia aparente (S), y su signo está determinado por la potencia activa.

$$PF = \frac{P_{Active}}{S}$$

Para la potencia reactiva, el CS5463 ofrece 2 resultados frutos de distintas metodologías de cálculo, Q_{Trig} o Q_{Avg} .

La potencia reactiva trigonométrica (Q_{Trig}) surge de aplicar identidades trigonométricas y se obtiene aplicando la fórmula.

$$Q_{Trig} = \sqrt{S^2 - P_{Active}^2}$$

Por otra parte, la potencia reactiva promedio (Q_{Avg}) se genera promediando el voltaje multiplicado por la corriente con un desfase de 90° entre ellos. El desfase se consigue aplicando un filtro digital IIR en el canal de voltaje, para obtener un voltaje de cuadratura. Este filtro proporcionará un cambio de fase de -90° en todas las frecuencias, y utiliza el valor de un registro llamado "épsilon" (ϵ) para lograr ganancia de unidad a la frecuencia de línea

El producto de la tensión instantánea en cuadratura (VQ) y la corriente (I), da como resultado la potencia de cuadratura (Q). El resultado de Q_{Trig} surge de promediar N resultados, utilizando la fórmula:

$$Q_{Avg} = \frac{\sum_{n=1}^N Q_n}{N}$$

La componente fundamental de la potencia activa (P_F) y reactiva (Q_F) se calculan realizando una transformada discreta de Fourier (DFT) a la frecuencia relevante en la corriente y voltajes instantáneos. Épsilon se usa para establecer la frecuencia del generador

de onda senoidal (componente imaginario) y cosenoidal (componente real) que posee el CI internamente. El componente armónico de la potencia activa (P_H) se calcula restando la potencia activa fundamental (P_F) a la potencia activa (P_{Active}).

Todas las mediciones descritas anteriormente son accesibles por medio de la lectura de registros y se actualizan cada N (4000) muestreos.

2.2.4 Velocidad de cálculo.

El circuito integrado realiza mediciones instantáneas de voltaje (V_n) y corriente (I_n) y calcula la potencia instantánea (P_n) a una tasa de salida (OWR).

$$\text{OWR} = \frac{(\text{MCLK}/K)}{1024}$$

Donde MCLK es 4,096 [MHz] y K un divisor de frecuencia de reloj configurable, que para este caso se programó en 1. Por lo que OWR vale 4000[Hz], lo que significa que se realizan 4000 mediciones de tensión y corriente instantánea y cálculos de potencia instantánea por segundo.

El voltaje RMS (V_{RMS}), la corriente RMS (I_{RMS}), la potencia Activa (P_{Active}), potencias reactivas (Q_{Trig} y Q_{Avg}), potencia aparente (S), factor de potencia (FP), los componentes fundamentales de la potencia activa y reactiva (P_F y Q_F) y la componente armónica de la potencia activa (P_H) se calculan utilizando N muestras instantáneas de V_n , I_n y P_n , como ya se mencionó, N por decisión de diseño tiene un valor de 4000, y por lo tanto los valores antes mencionados estarán disponibles cada OWR/N, es decir, cada 1[s].

$$\text{Computation Cycle} = \frac{\text{OWR}}{N}$$

Todas las medidas están disponibles como un porcentaje del fondo de escala y en complemento a dos con notación Q fraccionaria. El valor normalizado es entre -1 y +1.

Para registros sin signo, el valor normalizado es 0 o 1.

Dado que como ya se mencionó, el valor de los registros representa un porcentaje del valor de fondo de escala, para obtener los valores de las mediciones, se deben multiplicar por factores que surgen de la relación de transformación del transformador de intensidad y del divisor resistivo.

2.2.5 Comandos.

Existen 4 tipos de comandos: Instrucciones, sincronización, escritura de registros y lectura de registros.

Tanto las operaciones de lectura como de escritura se realizan enviando un valor numérico que indican el tipo de operación a realizar.

Luego, en el caso de operaciones de lectura, deben enviarse 3 bytes de datos. Por su parte las operaciones de lectura comienzan con un comando para leer datos que debe ser encadenado con otros comandos (es necesario escribir para recibir datos) se puede enviar un nuevo comando que se puede ejecutar durante la lectura original o bien comandos de sincronización. Para más información sobre los comandos, lea la hoja de datos que se agregara al final.

2.2.6 Fundamentos de elección de este circuito integrado.

Si bien en el mercado existen varios integrados con similares funciones, estudiando documentación de los medidores de energía utilizados por varias empresas distribuidoras de energía en la región, el circuito más utilizado es el CI CS5463. Comparando con otros, éste tiene muy buenas prestaciones, tanto en lo que respecta a la comunicación como en la posibilidad de calibración.

A la hora de realizar la compra, no tuvimos ningún problema, el mismo se encuentra en el mercado a nivel nacional y su precio es similar a sus competidores. [3]

2.3 Transformador de corriente.

Para medir corrientes de manera segura, la práctica más común es hacerlo con un Transformador de Corriente (CT) de propósito general o transductor el cual reduce la magnitud de la corriente en el circuito a un nivel más manejable. Así como en otras tecnologías de medición, hay una gran variedad de tipos de sensores específicos para diferentes aplicaciones.



Ilustración 6- CR 8320-1600.

Los transformadores de corriente usan la inductancia para medir la corriente, de esta forma el circuito a medir debe pasar al menos una vez por el interior del núcleo, aunque varias vueltas pueden ser usadas para multiplicar la corriente de salida. Cuando se instala los CT en un circuito existente, los de núcleo dividido son vistos con mayores ventajas ya que pueden ser abiertos y puestos alrededor de los cables existentes sin tener que interrumpir el circuito. Por otra parte, los CT de núcleo sólido requieren que el circuito sea re-cableado para que pueda pasar a través del núcleo. Debido a que los costos de la instalación pueden exceder el precio de los sensores, los sensores de núcleo dividido, aunque son más costosos, pueden reducir el costo total de la implementación. Los transductores de núcleo sólido pueden ser preferidos, en situaciones en donde mediciones extremadamente precisas son requeridas, ya que por el mismo precio de uno de núcleo dividido puede obtener más precisión.

En nuestro caso utilizamos uno de núcleo sólido, dado que, en la topología utilizada, el cableado del CT se realiza en la etapa de fabricación del dispositivo. De las opciones disponibles en el mercado, se optó por el modelo CR MAGNETICS 8320-1600, por su reducido tamaño y bajo costo. Además, ya se había utilizado en otro proyecto, por lo que se conocía su correcto funcionamiento y se disponía de un relevamiento de su curva de respuesta. [7]

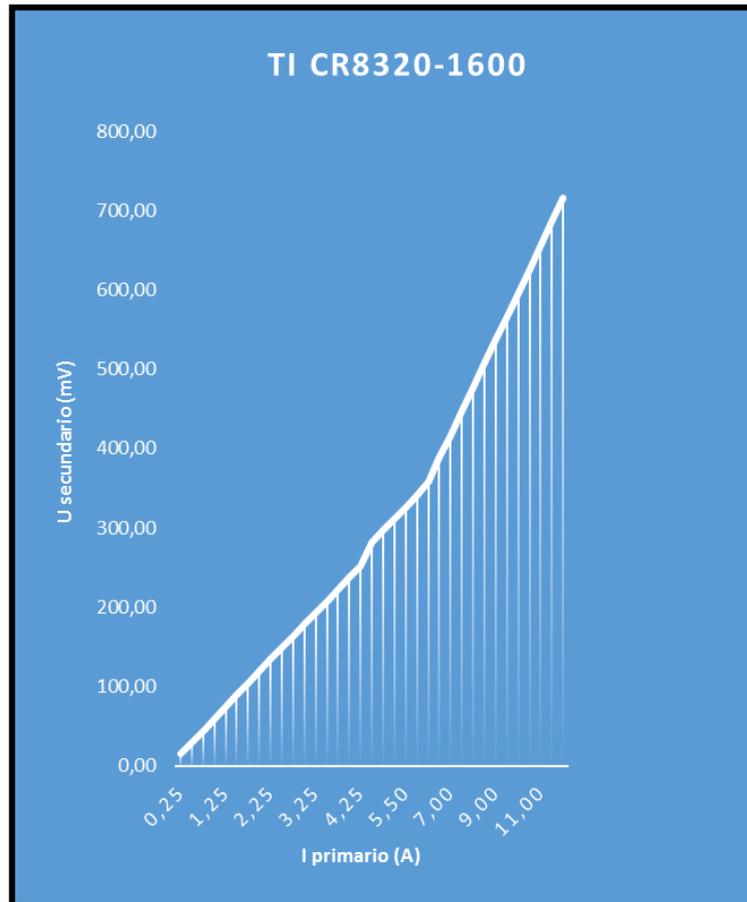


Ilustración 7- Tensión vs Corriente TI

Al final del informe se anexará la hoja de datos de este componente.

2.4 NodeMCU.

Para la comunicación se utilizó una placa NodeMCU, que integra un regulador y un adaptador UART-USB y un ESP8266.

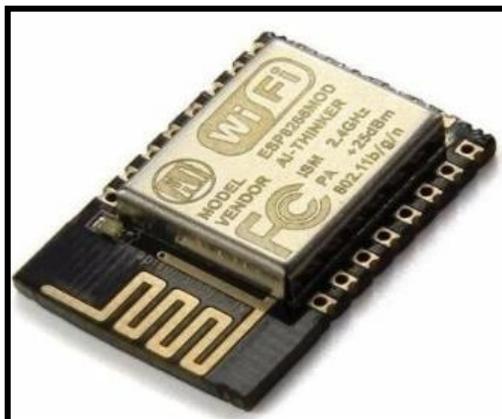


Ilustración 8-ESP8266.

El NodeMCU (véase ilustración 9) es un kit de desarrollo basado en el chip ESP8266, que cuenta con diez pines GPIO, los cuales pueden ser PWM, IIC, Serial y ADC. Es utilizado para conectar elementos inteligentes con la filosofía de "Internet de las Cosas" (IoT) y mediante el empleo de un firmware gratuito.

Las características más relevantes de este kit de desarrollo son que es, programable, de bajo costo e incorpora conectividad Wi-Fi. Con él se permite el control de hardware IO mediante una API avanzada, consiguiendo una reducción del trabajo de configuración y manipulación de hardware mediante lenguaje Arduino. Además, posee una API orientada a eventos para aplicaciones de red, facilitando la ejecución de código en una MCU de pequeñas dimensiones y mejorando la velocidad de desarrollo de aplicaciones IoT.

Existen empresas que están desarrollando este tipo de placas basadas en el chip ESP8266.

Las aplicaciones a ejecutar son cargadas en la memoria flash y se inician automáticamente cuando se le conecta una fuente de alimentación. Además, la utilización de una memoria cache integrada de alta velocidad, proporciona un aumento de rendimiento del sistema.

Una característica importante presente en todas las versiones de este módulo es la posibilidad de conectarle sensores externos u otros dispositivos mediante los GPIOs. Al tratarse de un dispositivo de pequeñas dimensiones, se consigue una minimización de tamaño de las PCBs diseñadas. [4]

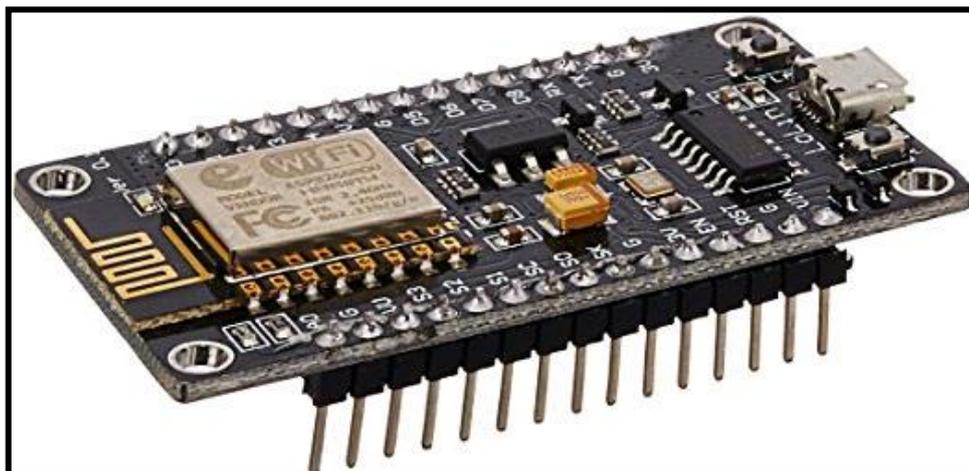


Ilustración 9- NodeMCU.

2.4.1 Principales especificaciones técnicas.

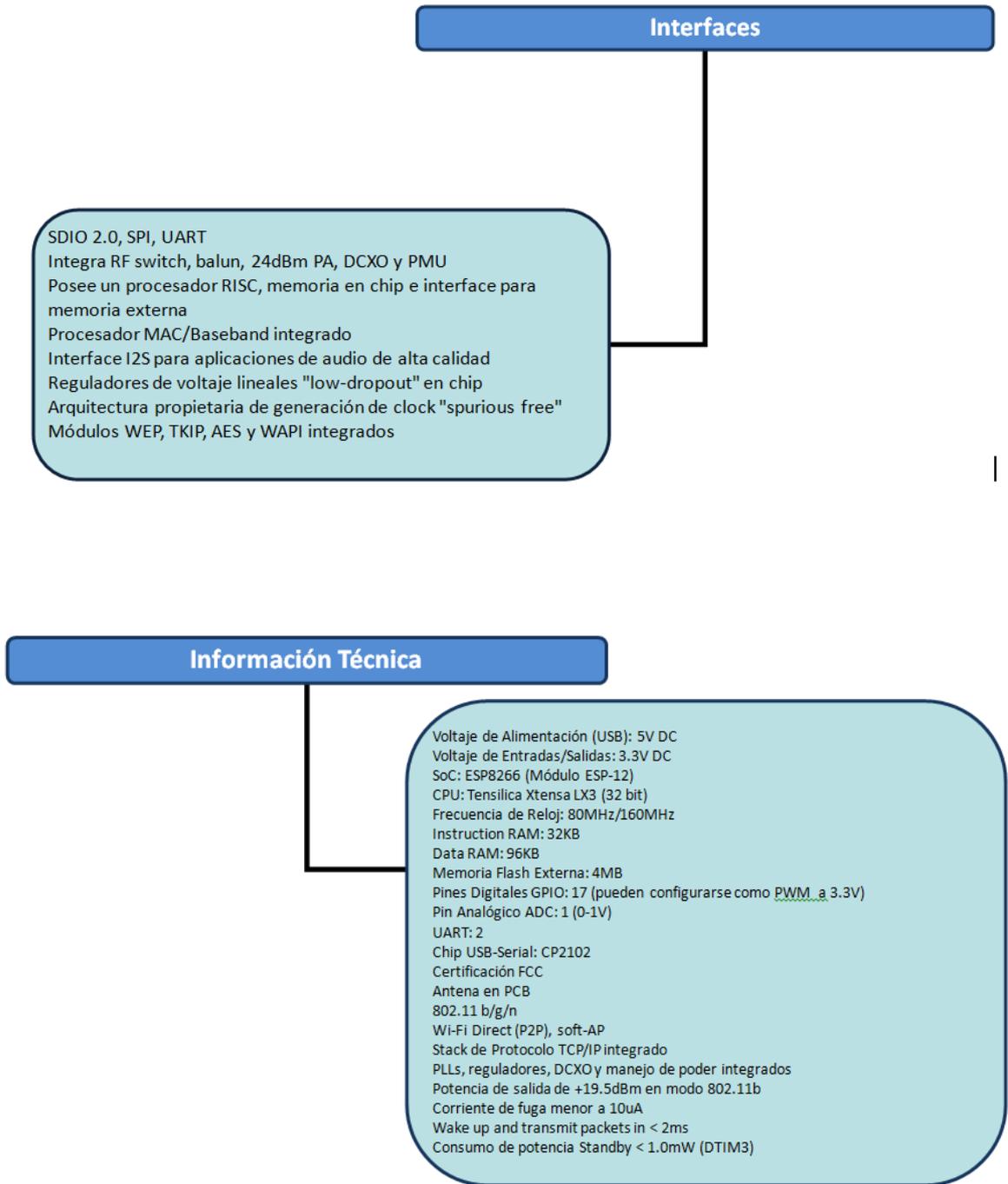
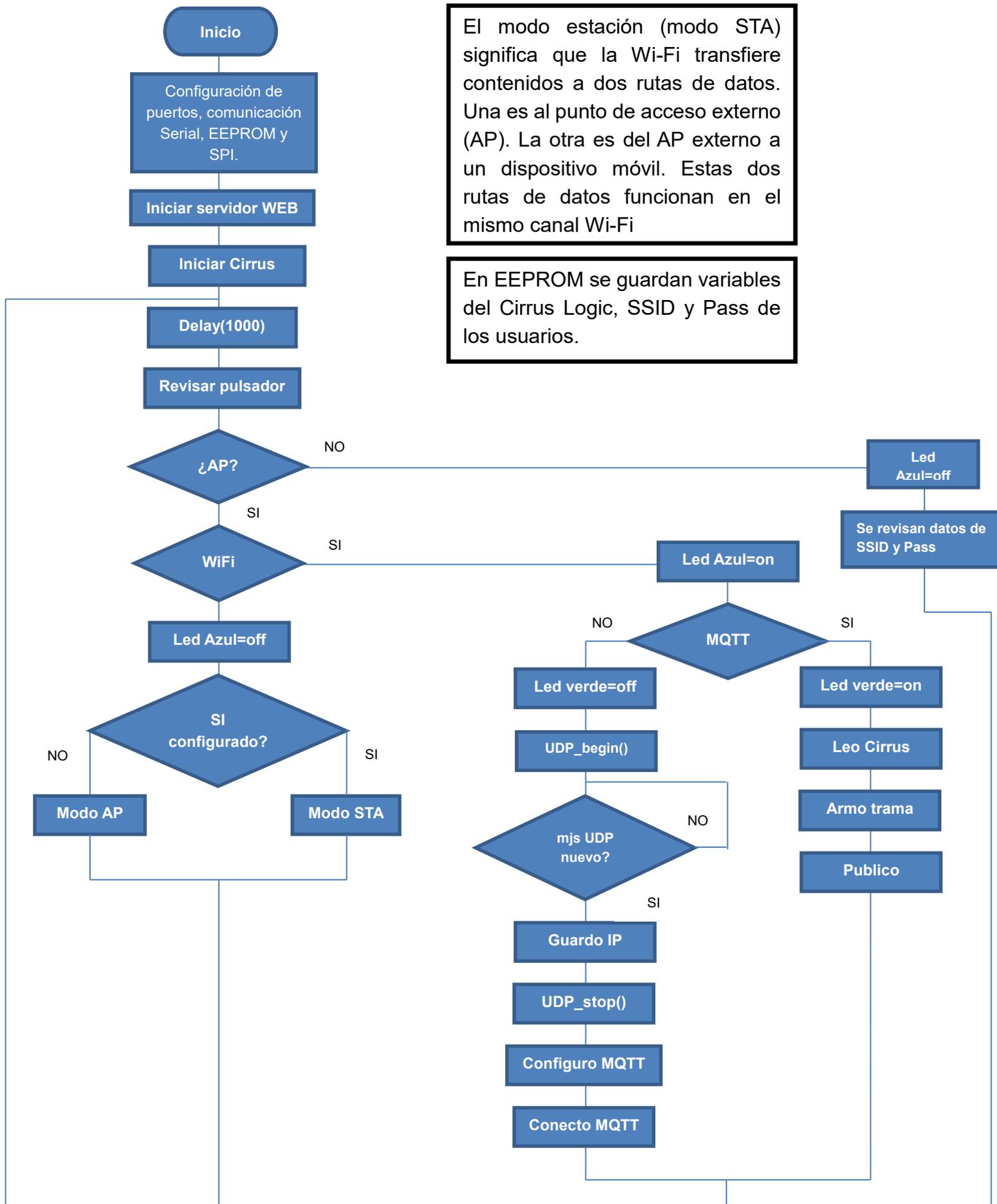


Tabla 1-Especificación técnica de nodeMCU.

2.4.2 Programa empleado en NodeMCU.



El modo estación (modo STA) significa que la Wi-Fi transfiere contenidos a dos rutas de datos. Una es al punto de acceso externo (AP). La otra es del AP externo a un dispositivo móvil. Estas dos rutas de datos funcionan en el mismo canal Wi-Fi

En EEPROM se guardan variables del Cirrus Logic, SSID y Pass de los usuarios.

2.5 Arduino IDE.

Para la programación de los nodeMCU, se utilizó la plataforma Arduino IDE es una herramienta de código abierto y difundida, proporcionada por la empresa desarrolladora de la placa Arduino. Esta herramienta es utilizada para crear código y poder cargarlo directamente en la placa que deseemos. Este entorno de desarrollo integrado (IDE) está escrito en lenguaje Java y otros softwares de código abierto.

Estas herramientas son utilizadas para programar cualquier placa de Arduino o incluso para otras placas de desarrollo compatibles con este entorno.

Para saber cómo trabajar con esta herramienta, en la web www.arduino.org, se explica el proceso paso a paso de descarga y configuración del entorno para una placa determinada. El primer paso consiste en realizar la descarga del entorno de desarrollo integrado, para ello, en la web oficial de Arduino tendremos la opción de descarga o el entorno en línea. Para obtener la herramienta deberemos de actuar de diferente forma en función del sistema operativo utilizado. Para usuarios del sistema Windows hay disponibles dos opciones, un instalador (opción recomendada) y un paquete ZIP. [5]

2.6 Comunicación entre medidor de energía y el NodeMCU.

La comunicación entre el CI CS5463 y el NodeMCU, se realizó a través de un protocolo llamado SPI (Serial Peripheral Interface).

SPI se ha convertido en uno de los protocolos más populares para trabajar con comunicación serial debido a su velocidad de transmisión, simplicidad, y a que muchos dispositivos electrónicos pueden trabajar con él.

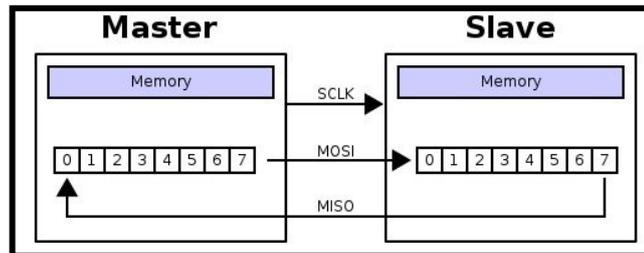


Ilustración 10-Estructura general del protocolo SPI.

El SPI es un protocolo síncrono que trabaja en modo full duplex para recibir y transmitir información, permitiendo que dos dispositivos pueden comunicarse entre sí al mismo tiempo utilizando canales diferentes o líneas diferentes en el mismo cable. Al ser un protocolo síncrono el sistema cuenta con una línea adicional a la de datos que se encarga de llevar la señal de reloj. Dentro de este protocolo se define un maestro que es el que controla las transacciones de lectura y escritura que transmite información a sus esclavos. Los esclavos serán aquellos dispositivos que se encarguen de recibir y acatar las órdenes del maestro. El maestro también puede recibir información de sus esclavos. Para que este proceso se haga realidad es necesario la existencia de dos registros de desplazamiento, uno para el maestro y uno para el esclavo respectivamente (véase ilustración 10). Los registros de desplazamiento se encargan de almacenar los bits de manera paralela para realizar una conversión paralela a serial para la transmisión de información.

Existen cuatro líneas lógicas encargadas de realizar todo el proceso:

MOSI (Master Out Slave In): Línea utilizada para llevar los bits que provienen del maestro hacia el esclavo.

MISO (Master In Slave Out): Línea utilizada para llevar los bits que provienen del esclavo hacia el maestro.

CLK (Clock): Línea proveniente del maestro encargada de enviar la señal de reloj para sincronizar los dispositivos.

SS (Slave Select): Línea encargada de seleccionar y a su vez habilitar un esclavo.

Se presenta una imagen donde se tienen todas estas líneas con sus respectivos registros de desplazamiento y su dirección de flujo (véase ilustración 11):

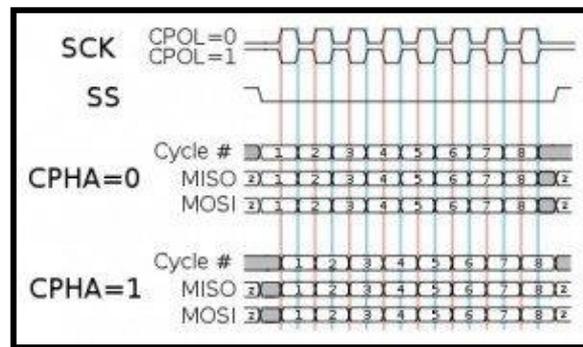


Ilustración 11-Modos en SPI.

Existen cuatro modos de configuración para el protocolo SPI. El primero de ellos es la polaridad y el segundo es la fase. Al tener dos parámetros donde cada uno puede tomar dos estados se tendrá entonces cuatro modos distintos de poder llevar a cabo el proceso de transmisión y envío de información.

Modo 0: CPOL = 0 y CPHA = 0. Modo en el cual el estado del reloj permanece en estado lógico bajo y la información se envía en cada transición de bajo a alto, es decir alto activo.

Modo 1: CPOL = 0 y CPHA = 1. Modo en el cual el estado del reloj permanece en estado lógico bajo y la información se envía en cada transición de alto a bajo, es decir bajo activo.

Modo 2: CPOL = 1 y CPHA = 0. Modo en el cual el estado del reloj permanece en estado lógico alto y la información se envía en cada transición de bajo a alto, es decir alto activo.

Modo 3: CPOL = 1 y CPHA = 1. Modo en el cual el estado del reloj permanece en estado lógico alto y la información se envía en cada transición de alto a bajo, es decir bajo activo.

Cada esclavo puede tener una configuración de CPOL y CPHA distinta a la de otros esclavos, inclusive una frecuencia de trabajo distinta y entonces para esto, el maestro deberá adaptarse a la configuración de cada esclavo.

En este protocolo se define únicamente un maestro y varios esclavos (en el presente proyecto en particular, uno y uno).

El principal campo de aplicación de este protocolo son las comunicaciones entre dispositivos a corta distancia.

Las ventajas que ofrece este protocolo, es la velocidad de transmisión ya que es configurable a través de software y dependerá también de los dispositivos utilizados en el sistema. Otros parámetros configurables a través de software son la frecuencia del reloj, la configuración de fase (CPHA) y polaridad (CPOL). Si solo existe un esclavo, puede colocarse la línea SS fija si el esclavo lo permite.

2.7 MQTT (Message Queue Telemetry Transport).

Este protocolo, fue el que se utilizó para la comunicación entre los medidores individuales y nuestro datalogger.

El motivo de la elección de MQTT es que es un protocolo abierto, ligero y además responde a las necesidades del proyecto de manera óptima.

Está especialmente adaptado para utilizar un ancho de banda mínimo, es ideal para utilizar redes inalámbricas, consume muy poca energía, es muy rápido y posibilita un tiempo de respuesta superior al resto de protocolos web actuales, permite una gran fiabilidad si es necesario y tal vez lo más importante requiere pocos recursos de procesadores y memorias.

Por todo lo anterior es el protocolo más utilizado en IoT. [6]

MQTT es un servicio de publicación/suscripción TCP/IP sencillo, su funcionamiento se basa en el principio cliente/servidor.

El servidor, llamado broker, recopila los datos que objetos comunicantes (en este caso los medidores puntuales de energía) le transmiten. Determinados datos recopilados por el broker se enviarán a determinados objetos comunicantes que previamente así se lo hayan solicitado (véase ilustración 12).

El principio de intercambio se parece mucho al de Twitter. Los objetos comunicantes envían los mensajes a un canal llamado topic y los suscriptores pueden leer esos mensajes. Los topics (o canales de información) pueden estar distribuidos jerárquicamente de forma que se puedan seleccionar exactamente las informaciones que se desean.

Los mensajes enviados por los objetos comunicantes pueden ser de todo tipo, pero no pueden superar los 256 Mb.

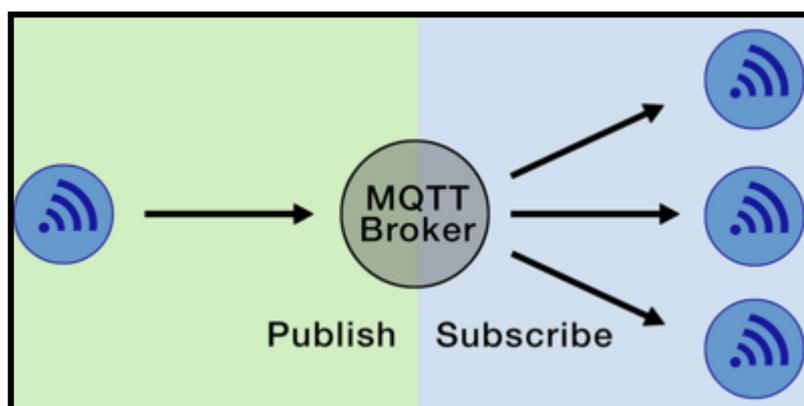


Ilustración 12-Principio de funcionamiento protocolo MQTT.

Ejemplo:

El topic «Medidor de energía 1/Tensión/TensiónRMS» comunicará la tensión RMS dentro de las posibles mediciones de tensión de un objeto conectado. Si a un objeto comunicante se suscribe al topic «Medidor de energía 1/Tensión/#», recibirá todos los datos tensión (por ejemplo, TensiónRMS, Tensión de pico....). Si se suscribe al topic «Medidor de energía 1/#», recibirá todas las mediciones que arroje el Medidor de energía 1.

Los mensajes de comunicación del protocolo MQTT poseen una pequeña porción de código, y cada uno de ellos está compuesto por un encabezado fijo, 2 bytes; un encabezado variable, utilizado solo en algunos de los mensajes, y una carga útil o payload limitada a 256 MB de información. Además, en todos los mensajes se informará del nivel de calidad de servicios (QoS).

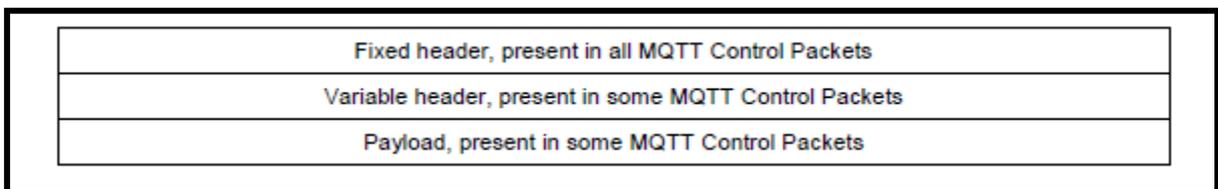


Ilustración 13-Paquete de Control MQTT.

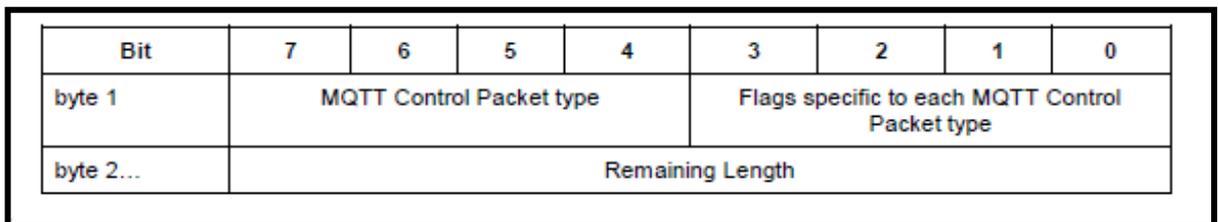


Ilustración 14-Estructura del paquete de control MQTT.

Como se nombró antes, algo muy importante de este protocolo es que soporta QoS (Quality of Service), calidad de servicio. En efecto, el objeto comunicante tiene la posibilidad de definir la calidad de su mensaje (véase ilustración 15).

Hay tres niveles posibles:

Un mensaje de QoS *nivel 0*, se entregará como mucho una vez. Eso significa que el mensaje se envía sin garantías de recepción (el broker no informa al remitente de que ha recibido el mensaje)

Un mensaje de QoS *nivel 1*, se entregará al menos una vez. El cliente lo transmitirá varias veces si es necesario, hasta que el broker le confirme que lo ha enviado a la red.

Un mensaje de QoS *nivel 2*, será obligatoriamente guardado por el emisor, que lo transmitirá siempre que el receptor no confirme su envío a la red. La principal diferencia radica en que el emisor utiliza una fase de reconocimiento más sofisticada con el broker para evitar la duplicación de los mensajes (más lento, pero más seguro).

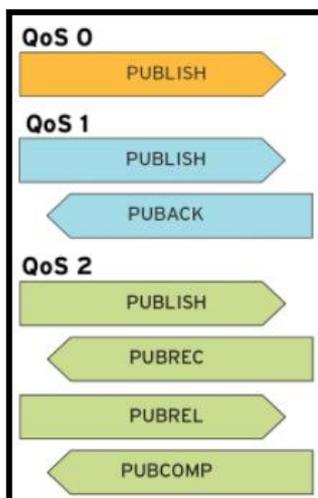


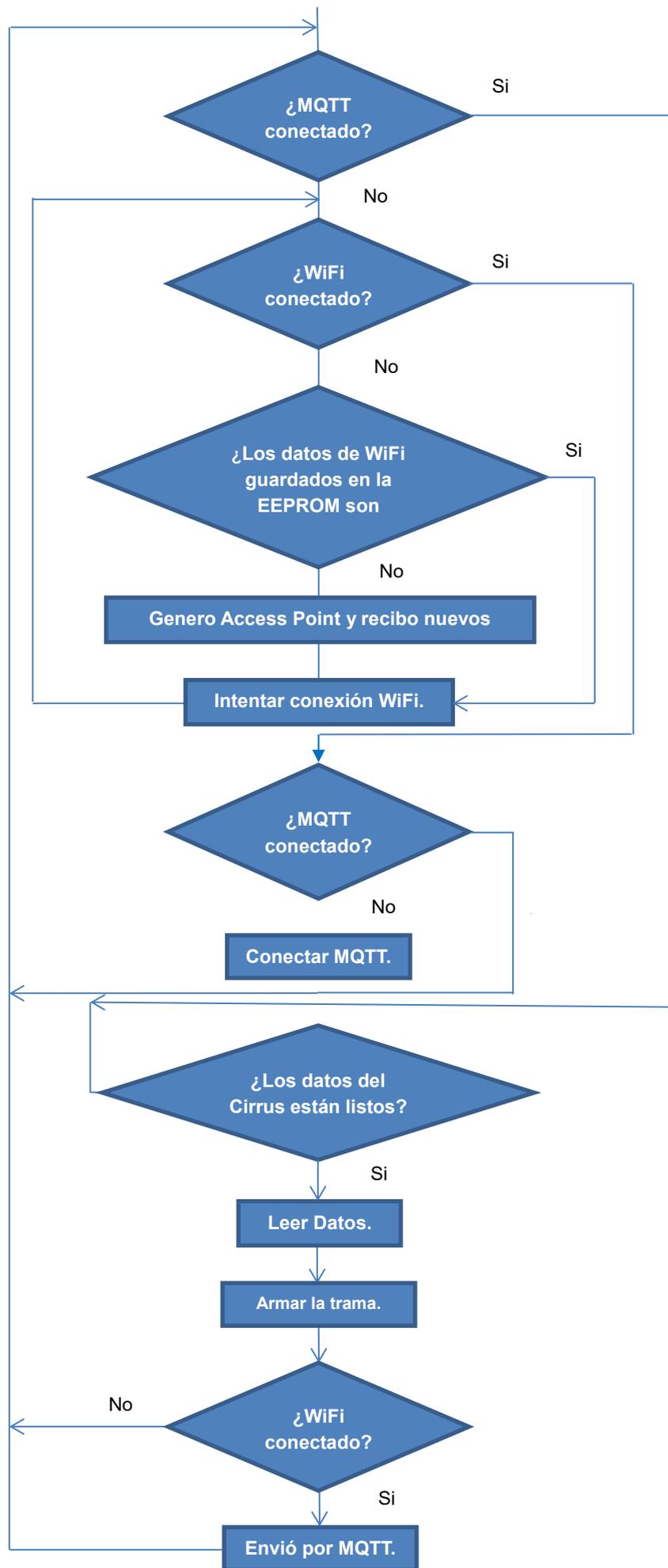
Ilustración 15-QoS del protocolo MQTT.

2.8 Funcionamiento, conexión, MQTT y medición.

Agregamos los siguientes comentarios antes de ver el próximo diagrama de flujo.

Si al intentar la conexión WiFi, ésta no puede establecerse tras 60 intentos distanciados 0,5 segundos, entonces se establece una bandera que indica que los datos almacenados en EEPROM son incorrectos. Si la conexión es satisfactoria se establece una bandera indicando que los datos en EEPROM (SSID y Pass de los usuarios) son correctos.

1. La comprobación de que esté lista la medición de datos del Cirrus Logic se realiza verificando el bit DRDY (Data Ready) del registro STATUS del CS5463.
2. Los datos leídos por el Cirrus Logic son: voltaje RMS, voltaje Pico, Corriente RMS, corriente pico, Contenido armónica de la potencia activa, potencia activa fundamental (50hz), potencia reactiva fundamental, potencia activa, potencia reactiva trigonométrica, potencia aparente y factor de potencia.
3. Los datos leídos del Cirrus Logic están en notación Q fraccionaria. Si son negativos, entonces deban hacerse como valores complemento a 2.
4. La trama a enviar vía MQTT está compuesta por todos los datos leídos del Cirrus Logic separados por "|". Todos en dos números después de la coma salvo FP que tiene tres.



2.9 Calibración de los medidores de energía.

Como ya hemos mencionado, el CI CS5463 cuenta con la posibilidad de realizar calibraciones por software. Las posibilidades que ofrece son: offset de tensión de continua, offset de corriente continua, offset de tensión alterna, offset de corriente alterna, ganancia de tensión, ganancia de corriente, calibración de constante del medidor, offset de potencia y desfase entre tensión y corriente.

Para el presente diseño no fue necesario implementarlas a todas, sino que únicamente se utilizaron las calibraciones de offset de DC en tensión, offset de AC en tensión y corriente y constante del medidor. La calibración de offset de DC en corriente no fue necesaria dado que la medición de la misma se realiza por medio de un transformador de corriente.

Por otro lado, las calibraciones de ganancia no se implementaron por ser un sistema de protección por software de las entradas diferenciales de tensión y corriente; y para el presente diseño se contemplaron por hardware en el momento de diseñar el circuito impreso.

En el caso de la calibración de offset de potencia, su función es que el medidor no contemple algún consumo fijo propio del sistema de medición o de la alimentación de algún dispositivo complementario. Pero dado el conexionado utilizado no fue necesario realizar estas contemplaciones.

Para poder realizar las calibraciones de offset antes mencionadas se dotó al medidor de una serie de jumper que permiten cortocircuitar a GND las entradas diferenciales. Hecho esto, se deben enviar vía serie los comandos de calibración correspondientes y esperar a que el Circuito integrado indique la correcta culminación del proceso por medio del bit DRDY del registro STATUS. Una vez terminada la calibración el CI guarda los resultados en registros específicos, pero que pertenecen a una memoria volátil, por lo que si se desea que esa calibración pueda ser útil para futuras puestas en marcha del medidor deben ser leídas por el NodeMCU y guardadas en la memoria ROM del mismo.

Cada vez que se encienda el módulo medidor (conjunto CS5463 + NodeMCU), en el proceso de inicialización del integrado Cirrus Logic deberán escribirse los registros de offset con los valores obtenidos de la calibración.

2.10 Protocolo UDP.

UDP (User Data Protocol, protocolo de datos de usuario). El Protocolo UDP nos ofreció un mecanismo para enviar datagramas IP encapsulados sin tener que establecer una conexión de alto nivel.

Muchas aplicaciones cliente-servidor que requieren de una solicitud usan el UDP, esto evita tener que establecer una conexión y luego liberarla. Un segmento UDP consiste en una cabecera de 8 bytes seguida de los datos, los dos puertos sirven para identificar los puntos terminales de las máquinas origen y destino (véase ilustración16). La suma de comprobación UDP incluye la misma pseudocabecera de formato, la cabecera UDP, y los datos, rellenos con una cantidad par de bytes de ser necesario.

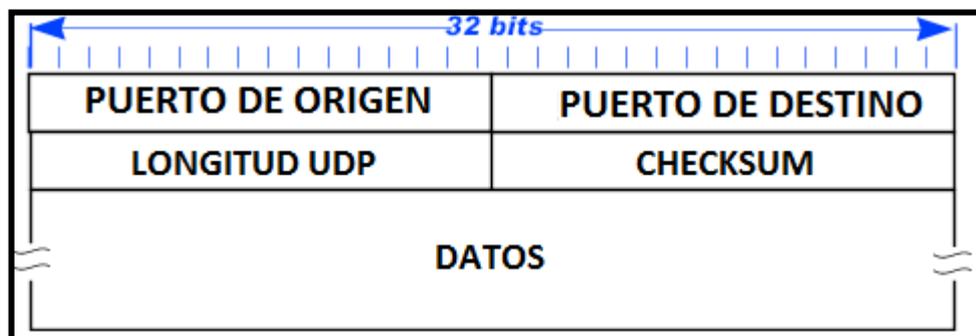


Ilustración 16-Segmento UDP

UDP no admite numeración de los datagramas esto sumado a que tampoco utiliza señales de confirmación de entrega, hace que no haya la garantía de que un paquete llegue a su destino. Esto también origina que los datagramas pueden llegar duplicados y/o desordenados a su destino. Por estos motivos el control de envío de datagramas, si existe, debe ser implementado por las aplicaciones que usan UDP como medio de transporte de datos, al igual que el reensamble de los mensajes entrantes.

Es por esto que es un protocolo del tipo best-effort (máximo esfuerzo), porque hace lo que puede para transmitir los datagramas hacia la aplicación, pero no puede garantizar que la aplicación los reciba.

Tampoco utiliza mecanismos de detección de errores. Cuando se detecta un error en un datagrama, en lugar de entregarlo a la aplicación destino, se descarta. [19]

El protocolo UDP se utilizó para realizar la acción de autodescubrimiento. Que consistirá en el envío de un mensaje de broadcast por parte del concentrador de datos a todos los dispositivos que se encuentren en la red. Esto ocurrirá de forma automática a intervalos de 10 minutos, existiendo también la posibilidad de forzarlo por medio de un botón colocado en la web para este fin.

Cuando los dispositivos que se encuentren en la red reciban este mensaje, se desempaquetará el mismo con el fin de obtener la ip y puerto de destino (en este caso la ip y puerto del datalogger), si este proceso se realiza con éxito los medidores puntuales estarán vinculados al broker, para así iniciar la conexión al servidor MQTT y comenzar a enviar tramas.

2.11 Datalogger o Concentrador de datos (Raspberry Pi).

Para el datalogger se utilizó una Raspberry Pi 3 (véase ilustración 17), según sus creadores, el objetivo era crear un dispositivo de bajo costo que permitiera acercar a todos los públicos un ordenador con el que mejorar las habilidades en informática, programación y electrónica.

Más allá de su objetivo inicial de acercar la informática a todos los públicos, especialmente en la docencia, lo cierto es que su bajo precio, bajo consumo energético, pequeño tamaño, e sus capacidades en electrónica (GPIOs, UART, etc) supuso que fuera un gran éxito entre la comunidad especializada.

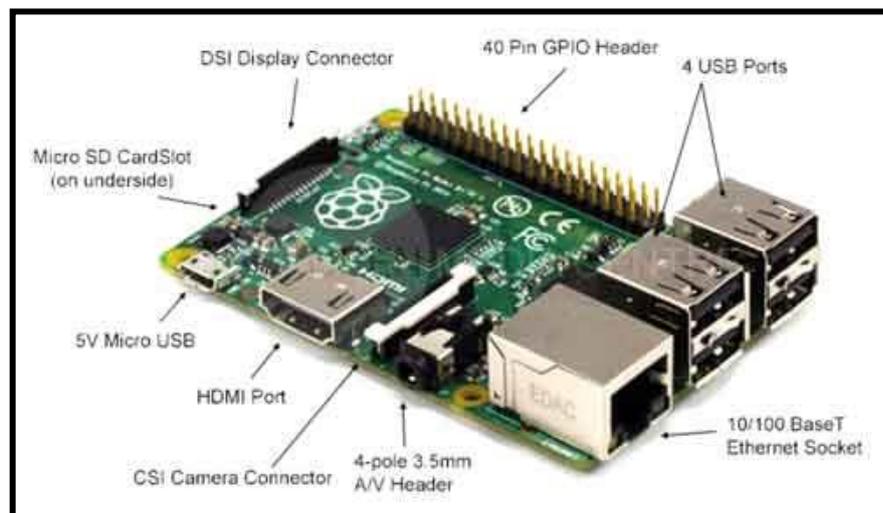


Ilustración 17-Raspberry Pi 3.

Uno de los puntos más altos de la Raspberry, es que existen una gran cantidad de desarrollos, documentación y tutoriales, que colaboran a la introducción y popularización del mismo.

2.11.1 Puertos de comunicación de una Raspberry Pi.

Las principales ventajas y uno de los motivos que determino su utilización, es la cantidad de puertos de comunicación distintitos con los que cuenta. Los principales son:

HDMI

Jack de 3.5mm para salida de audio.

Puertos USB.

Puertos para cámara.

I2C.

SPI.

Serial TTL.

Pines entra/salida.

Wifi integrado.

Bluetooth (4.1 baja energía BLE.)

Ethernet.

4× ARM Cortex-A53, 1.2GHz.

1 Gb de memoria RAM.

40 pines de propósito general.

2.11.2 Funcionamiento básico.

Para poder utilizar la Raspberry se debe de instalar algunos de los sistemas operativos disponibles en una tarjeta micro-SD. Los sistemas principales son:

- Raspbian, una distribución de Debian optimizada para su utilización en este tipo de arquitecturas (además, es el sistema operativo oficial de la fundación Raspberry).
- NOOBS, este es un instalador de diferentes sistemas operativos, entre los cuales se incluye Raspbian en su versión completa, Pidora que es una distribución de Fedora para la Raspberry, OSMC o LibreELEC que son distribuciones utilizadas para crear nuestro propio centro multimedia para la televisión, Lakka que es un emulador de consolas tipo retro, etc.

Una que hayamos elegido el sistema operativo a utilizar, (en nuestro caso utilizamos Raspbian) y se tenga grabado en la tarjeta de memoria, lo primero que deberemos de hacer, es introducir la tarjeta micro-SD en la ranura. Conectar la Raspberry a una pantalla, ya sea mediante el conector HDMI o utilizando la salida de video RCA y, en caso de que se desee, añadiremos un teclado y mouse mediante USB. Se puede conectar a Internet mediante el puerto Ethernet integrado o con una conexión wifi. Se conecta la fuente de alimentación al puerto micro-USB, destinado a tal fin, y se espera a que se inicie el sistema para poder comenzar a utilizarlo.

2.11.3 Inicio Raspberry Pi.

Es de extrema importancia lo que suceda con la raspberry cuando inicia. Por ejemplo.

Para saber esto, debido a que tenemos instalado en nuestra raspberry el sistema operativo Raspbian, como nombramos antes. Existe un comando para visualizar que se está ejecutando en este momento (véase ilustración 18).

```
sudo service --status-all
```

Donde obtenemos la siguiente información:

```
pi@raspberrypi:~ $ sudo service --status-all
[ - ] alsa-utils
[ + ] apache-htcacheclean
[ + ] apache2
[ + ] avahi-daemon
[ ? ] blinkled
[ ? ] blinkled.save
[ ? ] blinkled.save.1
[ + ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ + ] dbus
[ - ] detector-init
[ - ] detector-init.save.1
[ - ] dhcpcd
[ + ] dnsmasq
[ + ] dphys-swapfile
[ + ] fake-hwclock
[ - ] hostapd
[ - ] hwclock.sh
[ - ] isc-dhcp-server
[ - ] keyboard-setup.sh
[ + ] kmod
[ + ] lightdm
[ ? ] ml.py
[ + ] mosquitto
[ + ] mysql
[ + ] network-manager
```

Ilustración 18-Inicializacion Raspberry Pi.

Los signos “+” indican los servicios que se ejecutan, el “-“ los servicios que no ejecutan, el “?” los servicios que no permiten que el estado como un posible comando, la mayoría son comandos que no se necesitan para funcionar todo el tiempo.

Ahora para ver el nivel de ejecución, identifica los servicios que pueden/deben ejecutarse.

Una secuencia de arranque progresa de un nivel de arranque, se puede visualizar con el comando (véase ilustración 19).:

```
grep -nrl Default.Start /etc/init.d
```

```

pi@raspberrypi:~ $ grep -nrI Default-Start /etc/init.d
/etc/init.d/rsyslog:7:# Default-Start:      2 3 4 5
/etc/init.d/blinkled:9:# Default-Start:    2 3 4 5
/etc/init.d/mysql:9:# Default-Start:      2 3 4 5
/etc/init.d/kmod:8:# Default-Start:        S
/etc/init.d/apache2:6:# Default-Start:     2 3 4 5
/etc/init.d/detector-init.save.1:8:# Default-Start:    2 3 4 5
/etc/init.d/dbus:6:# Default-Start:        2 3 4 5
/etc/init.d/blinkled.save.1:9:# Default-Start:    2 3 4 5
/etc/init.d/isc-dhcp-server:9:# Default-Start:    2 3 4 5
/etc/init.d/blinkled.save:9:# Default-Start:    2 3 4 5
/etc/init.d/sudo:8:# Default-Start:        2 3 4 5
/etc/init.d/detector-init:8:# Default-Start:    2 3 4 5
/etc/init.d/raspi-config:6:# Default-Start: S 2 3 4 5
/etc/init.d/console-setup.sh:7:# Default-Start:    2 3 4 5
/etc/init.d/network-manager:8:# Default-Start:    2 3 4 5
/etc/init.d/bluetooth:6:# Default-Start:    2 3 4 5
/etc/init.d/ssh:7:# Default-Start:        2 3 4 5
/etc/init.d/plymouth:9:# Default-Start:    2 3 4 5
/etc/init.d/dphys-swapfile:20:# Default-Start:    2 3 4 5
/etc/init.d/dhcpd:9:# Default-Start:       2 3 4 5
/etc/init.d/triggershappy:6:# Default-Start:    2 3 4 5

```

Ilustración 19-Nivel de arranque Raspberry Pi

Esto muestra todos los archivos que identifican los diferentes servicios y el nivel de ejecución en cada servicio que se puede iniciar. Si un servicio inicia en un nivel de ejecución N, cuando el sistema avanza en N+1 no es necesario ser iniciado de nuevo.

El comando anterior nos da una idea exacta de las diferentes etapas en las que se inician los servicios. [17]

En este caso los servicios que se utilizan en el inicio son:

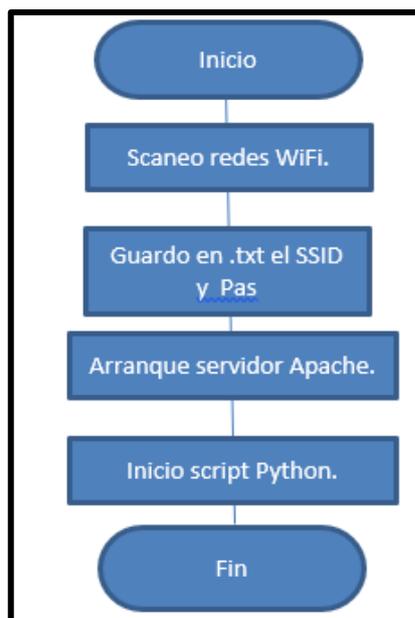


Ilustración 20-Diagrama en bloques inicio

2.12 Mosquitto.

Como se mencionó antes, una de las características más importantes del protocolo MQTT, es que los clientes o nodos no depende unos de otros ya que no tienen conocimiento de quién está al otro lado. Puede incluso que no haya nadie en el otro extremo.

Esto permite algo muy importante en proyectos de este tipo, la escalabilidad.

Al contrario de lo que ocurre con el protocolo HTTP, no hay que hacer una petición para recibir información desde un cliente. Cada cliente MQTT abre una conexión permanente TCP con el Broker.

El Broker tiene la capacidad de hacer que los mensajes sean persistentes, guardando el mensaje hasta que se conecte el cliente al que va dirigido. Recordemos que el Broker es el único que sabe quién está suscrito a un topic.

Para instalar MQTT en nuestra raspberry, utilizaremos un servidor muy conocido en IoT, mosquitto.

Mosquitto es un broker de código abierto que implementa la versión 3.1.1 del protocolo MQTT. Esto lo convierte en un potencial servidor de este protocolo para implementaciones de "Internet de las Cosas" en sistema con recursos limitados. Está destinado para su utilización en cualquier situación en la que exista la necesidad de mensajera ligera, tanto en máquinas muy potentes como en sistemas embebidos y de baja potencia.

Normalmente, la implementación actual de Mosquitto tiene un ejecutable que ocupa 120 kB y consume 3MB de RAM si se conectan 1000 clientes simultáneamente aproximadamente. Cabe destacar que esta implementación del protocolo MQTT es capaz de establecer conexiones con otros servidores de MQTT. Esto permite la utilización de Mosquitto para la creación de redes de servidores MQTT, consiguiendo una conectividad total entre diferentes puntos de este tipo de redes.

El proyecto Mosquitto es miembro de la Fundación Eclipse y consta de tres partes fundamentales y bien diferenciadas: [10]

- El servidor Mosquitto.
- El cliente `\mosquitto pub`", utilizado para realizar publicaciones en el servidor y el cliente `\mosquitto sub`", empleado para poder suscribirse a un tema, por lo tanto, son los métodos de comunicación de los clientes con el broker del protocolo.
- Una biblioteca cliente de MQTT escrita en C, con un contenedor en C++.

2.13 Instalación de Mosquitto en Raspberry Pi.

A la hora de realizar este proyecto, nunca se había trabajado con una Raspberry y tampoco se tenía mucho conocimiento de Linux, por eso se explicará a continuación la instalación de mosquitto en el concentrador de datos, con la intención de que pueda ser de utilidad para quien este en una situación similar. Ésta cuenta de dos pasos: [9]

Paso 1: Instalar Broker MQTT.

En principio la Raspberry debe estar conectada a internet, luego se abre el terminal del mismo (véase ilustración 21).

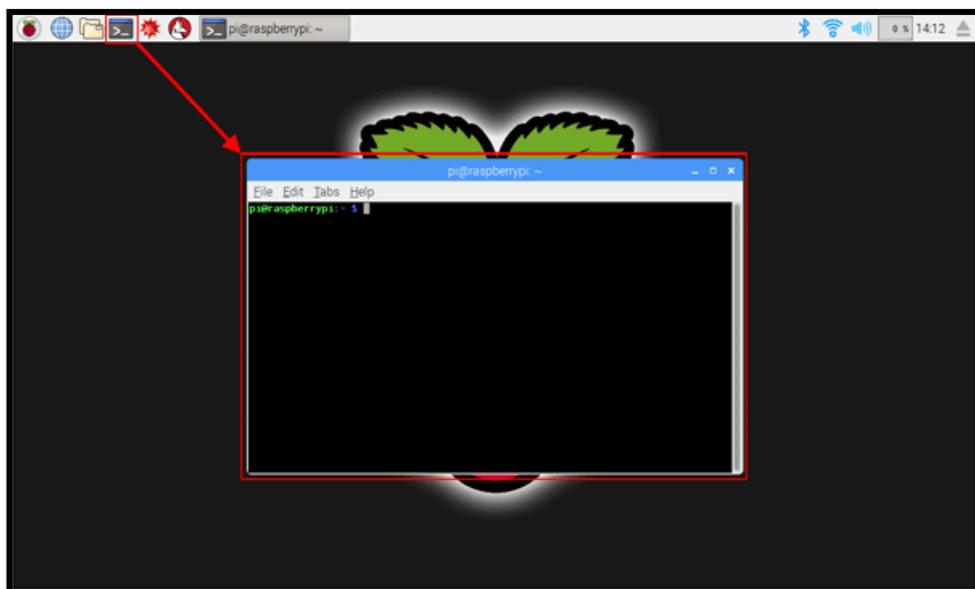
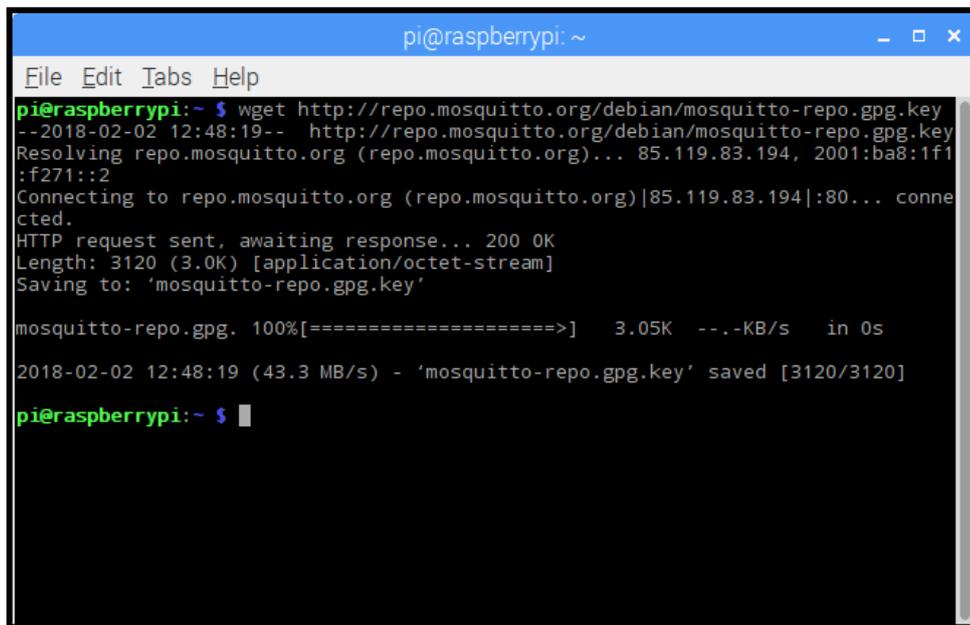


Ilustración 21-Instalación Broker MQTT, paso 1.

Ahora en la terminal de comandos, se ingresa la siguiente línea.

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
```

Este comando se usa para descargar la clave de firma, éste descarga el fichero en el que nos encontramos (véase ilustración 22).



```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
--2018-02-02 12:48:19-- http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
Resolving repo.mosquitto.org (repo.mosquitto.org)... 85.119.83.194, 2001:ba8:1f1:f271::2
Connecting to repo.mosquitto.org (repo.mosquitto.org)|85.119.83.194|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3120 (3.0K) [application/octet-stream]
Saving to: 'mosquitto-repo.gpg.key'

mosquitto-repo.gpg. 100%[=====] 3.05K --.-KB/s in 0s
2018-02-02 12:48:19 (43.3 MB/s) - 'mosquitto-repo.gpg.key' saved [3120/3120]

pi@raspberrypi:~ $

```

Ilustración 22-Instalación Broker MQTT, paso 2.

Luego se coloca la clave para una lista, para autenticar un paquete que luego se descargará (véase ilustración 22).

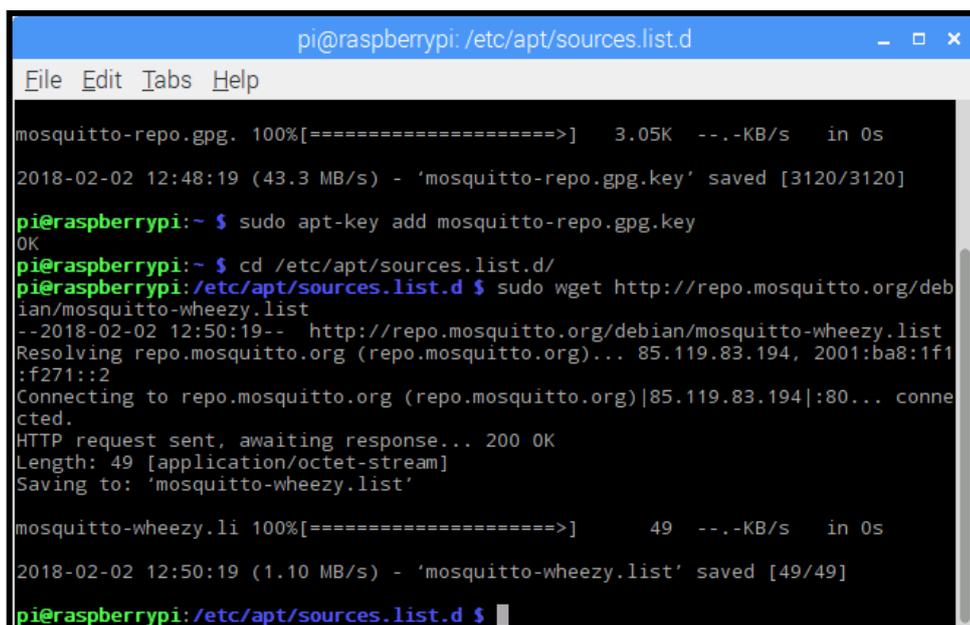
```
sudo apt-key add mosquitto-repo.gpg.key
```

Ahora con el comando `cd`, se cambia a la siguiente carpeta.

```
cd /etc/apt/sources.list.d/
```

Luego se descarga la lista de repositorios de Mosquitto con `wget`.

```
sudo wget http://repo.mosquitto.org/debian/mosquitto-jessie.list
```



```

pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
mosquitto-repo.gpg. 100%[=====] 3.05K --.-KB/s in 0s
2018-02-02 12:48:19 (43.3 MB/s) - 'mosquitto-repo.gpg.key' saved [3120/3120]

pi@raspberrypi:~ $ sudo apt-key add mosquitto-repo.gpg.key
OK
pi@raspberrypi:~ $ cd /etc/apt/sources.list.d/
pi@raspberrypi:/etc/apt/sources.list.d $ sudo wget http://repo.mosquitto.org/debian/mosquitto-wheezy.list
--2018-02-02 12:50:19-- http://repo.mosquitto.org/debian/mosquitto-wheezy.list
Resolving repo.mosquitto.org (repo.mosquitto.org)... 85.119.83.194, 2001:ba8:1f1:f271::2
Connecting to repo.mosquitto.org (repo.mosquitto.org)|85.119.83.194|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 49 [application/octet-stream]
Saving to: 'mosquitto-wheezy.list'

mosquitto-wheezy.li 100%[=====] 49 --.-KB/s in 0s
2018-02-02 12:50:19 (1.10 MB/s) - 'mosquitto-wheezy.list' saved [49/49]

pi@raspberrypi:/etc/apt/sources.list.d $

```

Ilustración 23--Instalación Broker MQTT, paso 3.

Ahora se actualizan los paquetes disponibles con el comando:

```
apt-get update
```

Por último se instala el bróker Mosquitto, con siguiente comando:

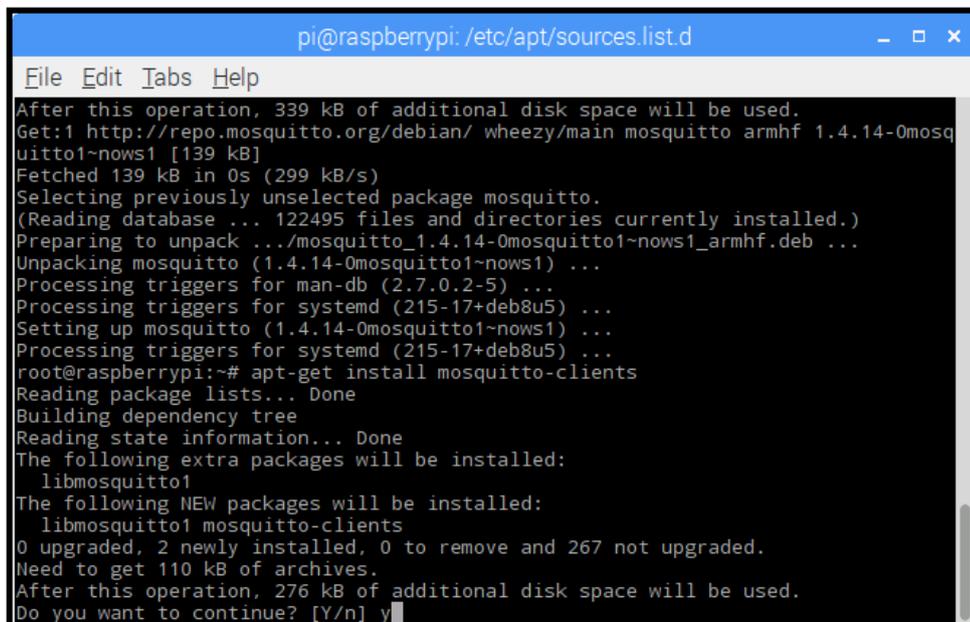
```
apt-get install mosquitto
```

Paso 2: Instalar cliente MQTT

Este paso es muy sencillo, se realiza con el siguiente comando:

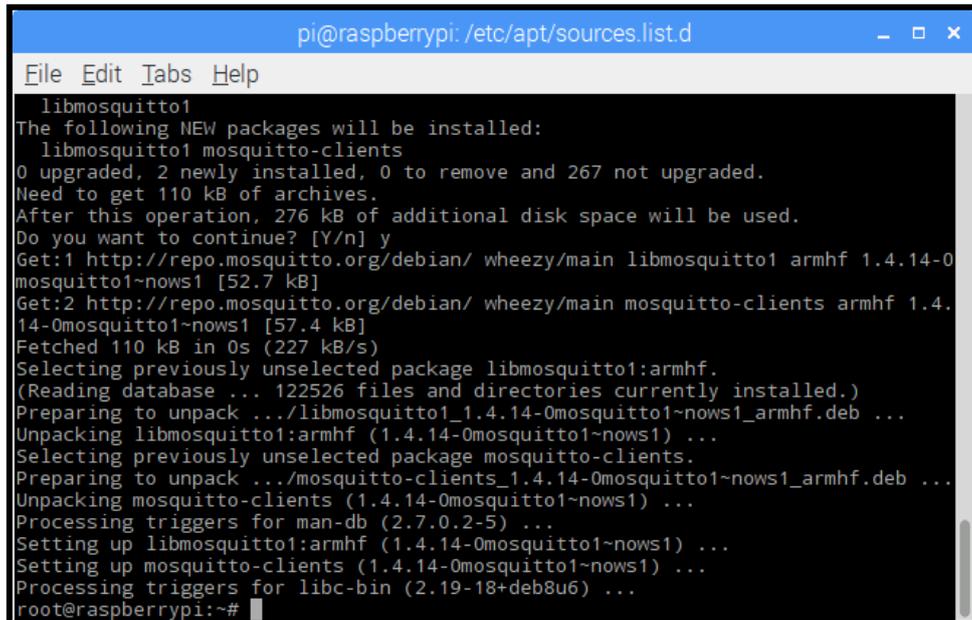
```
sudo apt-get install mosquitto-clients
```

Tener en cuenta que, en determinado momento de la instalación, dirá ¿Quiere continuar? Le colocamos “y” (véase ilustración 24 y 25).



```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
After this operation, 339 kB of additional disk space will be used.
Get:1 http://repo.mosquitto.org/debian/ wheezy/main mosquitto armhf 1.4.14-0mosq
uito1~nows1 [139 kB]
Fetched 139 kB in 0s (299 kB/s)
Selecting previously unselected package mosquitto.
(Reading database ... 122495 files and directories currently installed.)
Preparing to unpack ../mosquitto_1.4.14-0mosquitto1~nows1_armhf.deb ...
Unpacking mosquitto (1.4.14-0mosquitto1~nows1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Processing triggers for systemd (215-17+deb8u5) ...
Setting up mosquitto (1.4.14-0mosquitto1~nows1) ...
Processing triggers for systemd (215-17+deb8u5) ...
root@raspberrypi:~# apt-get install mosquitto-clients
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmosquitto1
The following NEW packages will be installed:
  libmosquitto1 mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 267 not upgraded.
Need to get 110 kB of archives.
After this operation, 276 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Ilustración 24- Instalación cliente MQTT, paso 1.



```
pi@raspberrypi: /etc/apt/sources.list.d
File Edit Tabs Help
libmosquitto1
The following NEW packages will be installed:
libmosquitto1 mosquitto-clients
0 upgraded, 2 newly installed, 0 to remove and 267 not upgraded.
Need to get 110 kB of archives.
After this operation, 276 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://repo.mosquitto.org/debian/ wheezy/main libmosquitto1 armhf 1.4.14-0
mosquitto1~nows1 [52.7 kB]
Get:2 http://repo.mosquitto.org/debian/ wheezy/main mosquitto-clients armhf 1.4.
14-0mosquitto1~nows1 [57.4 kB]
Fetched 110 kB in 0s (227 kB/s)
Selecting previously unselected package libmosquitto1:armhf.
(Reading database ... 122526 files and directories currently installed.)
Preparing to unpack ../libmosquitto1_1.4.14-0mosquitto1~nows1_armhf.deb ...
Unpacking libmosquitto1:armhf (1.4.14-0mosquitto1~nows1) ...
Selecting previously unselected package mosquitto-clients.
Preparing to unpack ../mosquitto-clients_1.4.14-0mosquitto1~nows1_armhf.deb ...
Unpacking mosquitto-clients (1.4.14-0mosquitto1~nows1) ...
Processing triggers for man-db (2.7.0.2-5) ...
Setting up libmosquitto1:armhf (1.4.14-0mosquitto1~nows1) ...
Setting up mosquitto-clients (1.4.14-0mosquitto1~nows1) ...
Processing triggers for libc-bin (2.19-18+deb8u6) ...
root@raspberrypi:~#
```

Ilustración 25-Instalación cliente MQTT, paso 2.

2.14 Base de Datos.

Los medidores puntuales de energía se encontrarán enviando continuamente datos conteniendo la información del estado en el que se encuentran y las mediciones que se desean visualizar, esta información debe ser almacenada y para ello se utilizó una base de datos. De entre todas las alternativas de libre distribución que se podían utilizar, se eligió MariaDB que es el futuro reemplazo de MySQL. Los motivos de haber elegido este sistema concreto están relacionados con la interfaz web, debido a que ésta tendrá que realizar muchas consultas rápidas hacia la base de datos, para mantener la Información a tiempo real.

Además, se eligió por sobre MySQL, por tener un mayor rendimiento, nuevas funcionalidades e incorporar nuevas tablas a nivel sistema que ayudan en las tareas de optimización de base de datos gracias al almacenamiento de estadísticas del servicio. [7]



The image shows the phpMyAdmin login interface. At the top, there is a logo for phpMyAdmin and the text "Bienvenido a phpMyAdmin". Below this, there is a language selection dropdown menu labeled "Idioma - Language" with "Español - Spanish" selected. A "Iniciar sesión" button is present. The login form includes a "Usuario:" field with "phpmyadmin" entered and a "Contraseña:" field with masked characters. A "Continuar" button is located at the bottom right of the form.

Ilustración 26-Página de inicio del administrador de base de datos de MariaDB.

La base de datos deberá estar formada por:

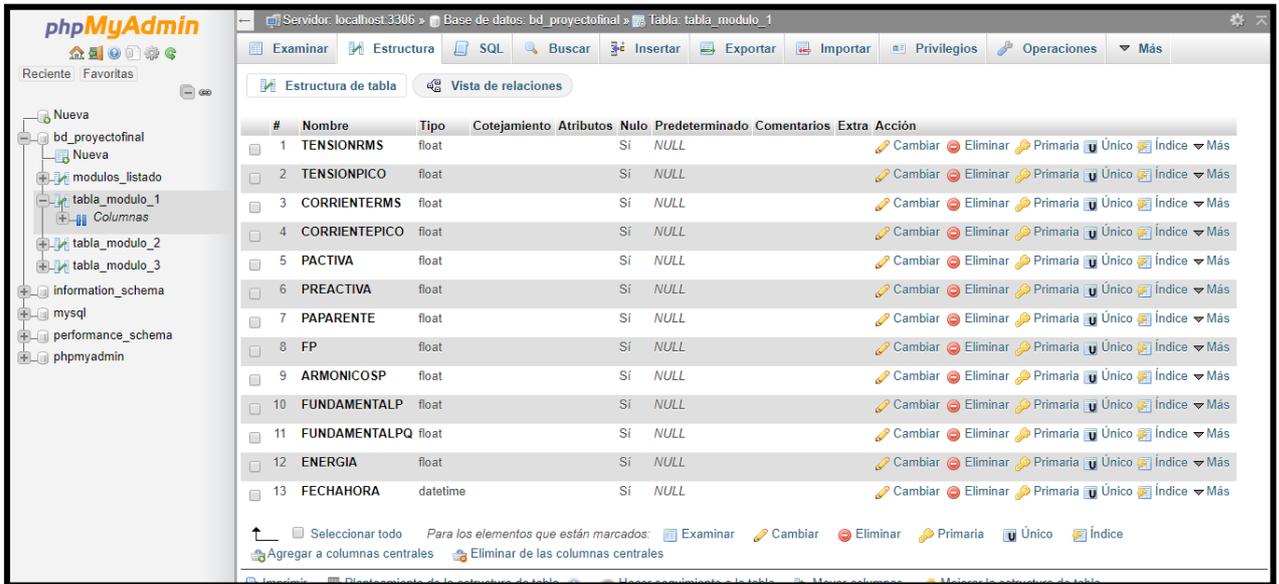


Ilustración 27-Captura de pantalla conteniendo la estructura de una de las tablas de registro.

TENSIONRMS	TENSIONPICO	CORRIENTERMS	CORRIENTEPICO	PACTIVA	PREACTIVA	PAPARENTE	FP	ARMONICOSP	FUNDAMENTALP	FUNDAMENTALPQ	ENERGIA	FECHAHORA
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	257 2018-04-01 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	290 2018-04-02 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	152 2018-04-03 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	94 2018-04-04 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	321 2018-04-05 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	71 2018-04-06 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	266 2018-04-07 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	44 2018-04-08 12:39:11
102	70	65	12.9	12.3	12.3	12.3	12.3	12.3	12.3	12.3	12.3	233 2018-04-09 12:39:11
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:07:55
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:07:57
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:07:59
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:01
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:03
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:05
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:07
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:09
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:11
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:13
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:15
228.2	12.08	2.49	1.2	0.95	10.26	25.33	10.02	15.15	12.23	77.52	100	2018-04-14 00:08:17

Ilustración 28-Captura de pantalla de los datos registrados en una de las tablas.

2.15 Servidor Web.

Apache es el servidor Web con mayor presencia en el mercado mundial. Aunque su configuración es relativamente sencilla, fortalecer sus condiciones de seguridad implica entender y aplicar un conjunto de reglas generales conocidas, aceptadas y disponibles.

La configuración de Apache se realiza mediante la edición del archivo de texto `httpd.conf`, el cual tiene todas las instrucciones que debe seguir Apache para su funcionamiento.

Apache se caracteriza por ser estable, multiplataforma, modular y altamente configurable, lo cual significa que se puede adaptar para satisfacer diferentes necesidades. Éste registra los diferentes eventos que ocurren cuando está en servicio a través de archivos log, facilitando la obtención de estadísticas que son usadas para la toma de decisiones por parte del administrador. Además, dispone de componentes de seguridad, los cuales pueden ser aprovechados para fortalecer las condiciones de acceso a recursos Web disponibles para ser recuperados a través de solicitudes HTTP realizadas por un navegador, siempre y cuando sean configurados apropiadamente. Apache se caracteriza también por ser de código abierto y gratuito. [8]

2.15.1 Instalación de Apache en la Raspberry Pi.

Para instalar el servidor Apache en el datalogger, se ejecutaron los siguientes comandos en la terminal de la Raspberry:

```
sudo apt-get install apache2 -y
```

Luego debemos guardar el contenido a publicar en el siguiente directorio:

```
/var/www/html/
```

La página principal se llamara index.html. Ahora para poder visualizar el contenido se debe colocar en un navegador la dirección IP asignada por la Raspberry.

Una vez que tenemos instalado Apache debemos instalar PHP (Hypertext Preprocessor), se hizo con el siguiente comando:

```
sudo apt-get install php5 libapache2-mod-php5 -y
```

```
sudo leafpad index.php
```

Para finalizar se reinicia el servidor web con el comando:

```
sudo service apache2 restart
```

2.15.2 Protocolo HTTP.

El protocolo HTTP (Hypertext Transfer Protocol), protocolo de transferencia de hipertexto, fue el que permitió realizar una petición de datos y recursos, como pueden ser documentos HTML. Representa la base de cualquier intercambio de datos en la Web.

Es un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada, por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así una página web completa, resulta de la unión de distintos sub-documentos recibidos, como pueden ser, un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc.

El funcionamiento de manera sencilla sería el siguiente, cada vez que un cliente realiza una petición a un servidor, se ejecutan los siguientes pasos:

Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en la barra de direcciones de algún navegador web.

El cliente Web decodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.

Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente.

Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor.

El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información. Luego se cierra la conexión TCP.

Este proceso se repite en cada acceso al servidor HTTP (véase ilustración 29). Por ejemplo, si se recoge un documento HTML en cuyo interior están insertadas cuatro imágenes, el proceso anterior se repite cinco veces, una para el documento HTML y cuatro para las imágenes.

Vamos a describir, de manera concisa el por qué es necesario enviar el GET, POST y HEAD.

GET: Se utiliza para recoger cualquier tipo de información del servidor. Se utiliza siempre que se pulsa sobre un enlace o se teclea directamente a una URL. Como resultado, el servidor HTTP envía el documento correspondiente a la URL seleccionada.

HEAD: Solicita información sobre un objeto (fichero): tamaño, tipo, fecha de modificación, etc.

Es utilizado por los gestores de cachés de páginas o los servidores proxy, para conocer cuándo es necesario actualizar la copia que se mantiene de un fichero.

POST: Sirve para enviar información al servidor, por ejemplo, los datos contenidos en un formulario. La operación que se realiza con la información proporcionada depende de la URL utilizada. Se utiliza, sobre todo, en los formularios.

Un cliente Web selecciona automáticamente los comandos HTTP necesarios para recoger la información requerida por el usuario. Así, ante la activación de un enlace, siempre se ejecuta una operación GET para recoger el documento correspondiente. El envío del contenido de un formulario utiliza GET o POST, en función del atributo de <FORM METHOD="...">. [11]

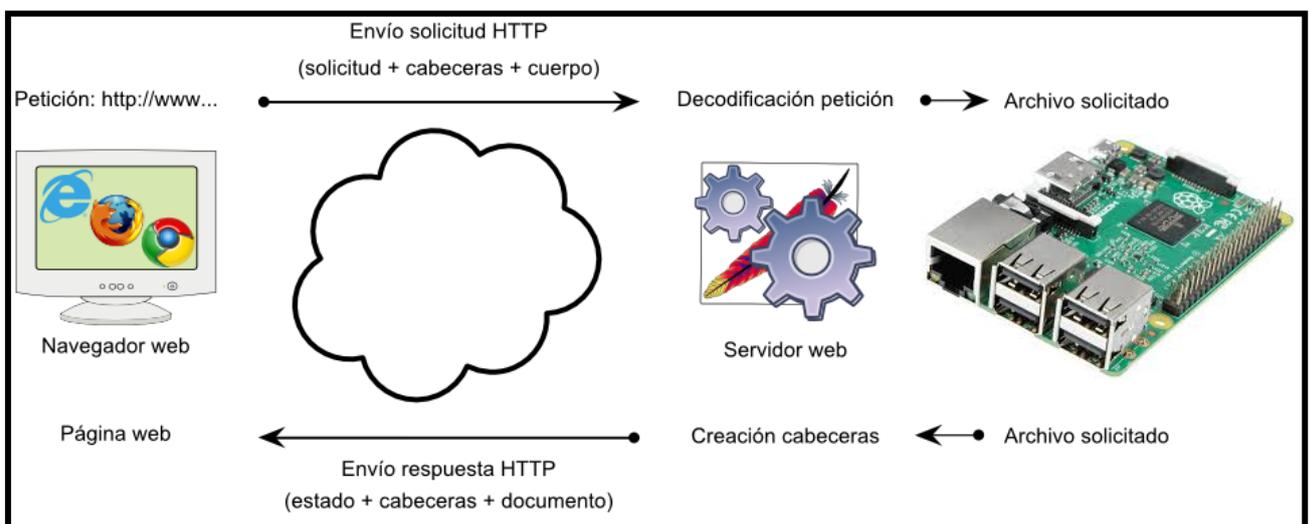


Ilustración 29-Protocolo HTTP.

2.15.3 Lenguaje HTML.

HTML (HyperText Markup Language), Lenguaje de Marcas de Hipertexto, es un lenguaje de programación que se debió utilizar para indicar como va ordenado el contenido de la página web, esto no incluye el diseño gráfico de la web.

Para lograrlo se utilizan marcas de hipertexto, las cuales son etiquetas que en ingles de denominan "tags".

Algo importante es que las etiquetas no solo sirven para ordenar el contenido de la web, si no también sirven para que buscadores como Google, Yahoo, etc. puedan encontrar la información.

Las etiquetas van dentro de los caracteres <>, por ejemplo <Proyecto Final UTN FRP>.

Además de esto deben tener un inicio y un fin, que se deben colocar de la siguiente manera:

<Inicio de la etiqueta>

</ Fin de la etiqueta>

También tiene la posibilidad de algunas excepciones, como:

<head>, <title>, <body>, <header>, <article>, <section>, <p>, <div>, , .

Siempre se debe empezar con la etiqueta HTML. [12]

Un ejemplo podría ser el siguiente:

```
<html>
<head>
  <title>Proyecto final</title>
</head>
<body>
  <h1>Medidores Puntales de Energía <u>h1</u>.</h1>
  <h2>UTN Facultad regional Paraná <u>h2</u>.</h2>
  <p>Catinot-Marino-Weber <b>p</b> de <i>html</i></p>
</body>
</html>
```

A continuación, se hace una breve descripción de las etiquetas más importantes:

<head> Provee información acerca del documento, incluyendo su título y enlaces a scripts y hojas de estilos.

<body> Representa el contenido de un documento HTML. Sólo puede haber un elemento <body> en un documento.

`<header>` Representa un grupo de ayudas introductorias o de navegación. Puede contener algunos elementos de encabezado, pero también otros elementos como un logo, una sección que aglutine secciones de encabezados, un formulario de búsqueda o cosas parecidas.

`<article>` Representa una composición auto-contenida en un documento, página, una aplicación o en el sitio, que se destina a distribuir de forma independiente o reutilizable. Podría ser un mensaje en un foro, un artículo de una revista o un periódico, una entrada de blog, un comentario de un usuario, un widget interactivo o gadget, o cualquier otro elemento independiente del contenido.

`<section>` Representa una sección genérica de un documento. Sirve para determinar qué contenido corresponde a qué parte de un esquema. Pensemos en un esquema como en el índice de un libro. Es, por lo tanto, una etiqueta semántica. Su funcionalidad principal es estructurar semánticamente un documento a la hora de ser representado por parte de un usuario.

`<p>` Es el apropiado para distribuir el texto en párrafos.

`<div>` Sirve para crear secciones o agrupar contenidos.

`` Es un contenedor en línea. Sirve para aplicar estilo al texto o agrupar elementos en línea.

`` Representa una imagen en el documento.

2.15.4 Lenguaje PHP.

Un lenguaje de programación que se utilizó para hacer la web, fue PHP (Hypertext Preprocessor), es un lenguaje abierto muy utilizado para el desarrollo web y tiene la particularidad que puede ser incrustado en HTML.

El formato es el siguiente:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Proyecto final</title>
  </head>
  <body>

    <?php
      echo ";Proyecto final UTN FRP!";
    ?>

  </body>
</html>
```

Las paginas PHP contienen HTML con un código incrustado que realiza alguna acción. El código PHP esta siempre encerrado entre las etiquetas especiales <?php "codigo"?>, que permiten entrar y salir del modo PHP.

Lo que distingue a PHP de algo del lado cliente como javascript es que el código es ejecutado en el servidor (Raspberry Pi), generando HTML y enviándolo al cliente. El cliente

Básicamente lo que hicimos fue programación basada en scripts. [13]

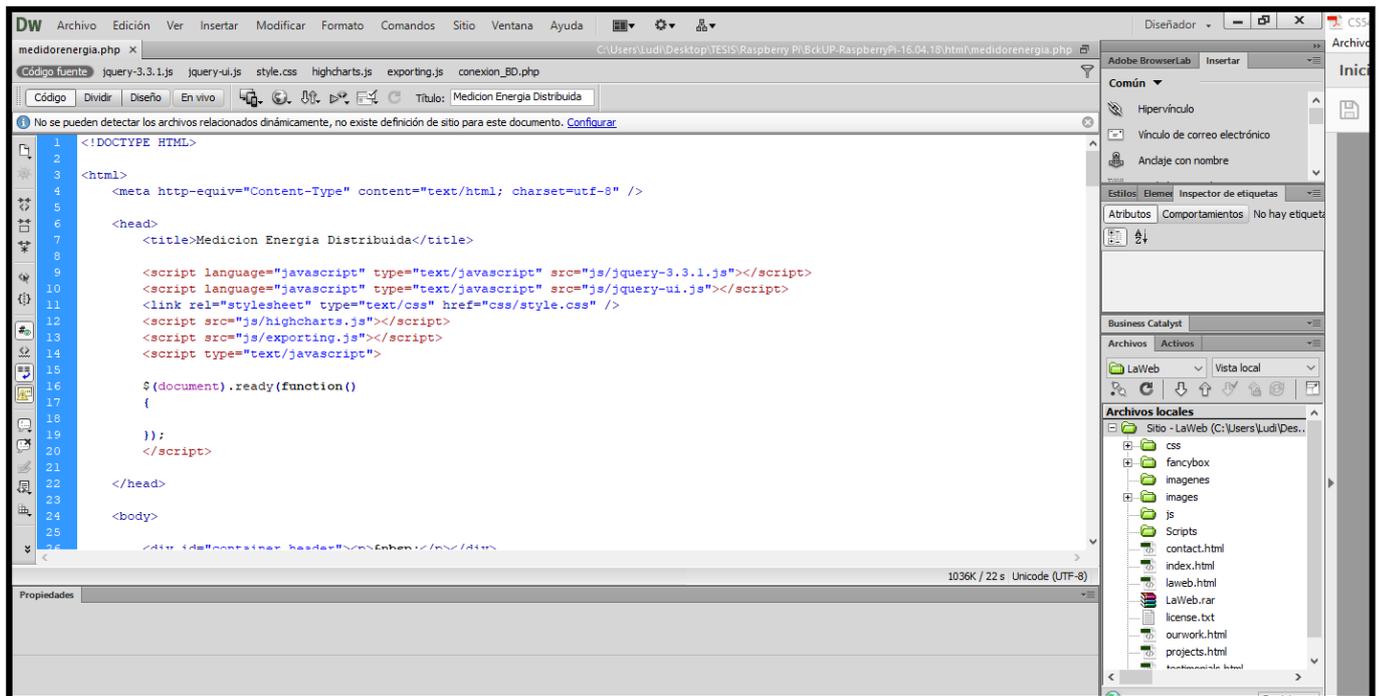


Ilustración 30- En nuestro proyecto ver <html> <body> <head>

2.15.5 Lenguaje JavaScript.

Para el diseño de la página web, también se usó JavaScript.

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en el ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web.

A modo de ejemplo como para tomar dimensión de lo que se puede lograr con este lenguaje, supongamos que un estudiante escribe una dirección web en su navegador, por ejemplo `http://www.frp.utn.edu.ar`, el servidor recibe la petición y como respuesta a esa petición envía al ordenador del usuario código HTML junto a código JavaScript. El código HTML se encarga de que en la pantalla se muestre algo, por ejemplo, una imagen, un menú, etc. El código JavaScript se puede encargar de crear efectos dinámicos en respuesta a acciones del usuario, por ejemplo, que se despliegue un menú tipo acordeón cuando el usuario pasa el ratón por encima de un elemento del menú.

La ventaja de JavaScript es que al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc.

La forma más sencilla que encontramos de usar JavaScript dentro de una página web es embebiendo directamente el código JavaScript dentro del código HTML.

```
<html>
<head>

  <title>Embeler JavaScript - Proyecto final</title>
</head>
<body>

  <?scrip type="text/javascript">
    document.write('Proyecto final UTN FRP!');
  </scrip>

</body>
</html>
```

Entonces se podría decir que los documentos HTML permiten incrustar fragmentos de código JavaScript, dentro del propio archivo HTML. Dentro de un documento HTML puede haber ninguno, uno o varios scripts de JavaScript. El uso de JavaScript es muy habitual en la programación web. [14]

2.15.6 Nuestra Web.

Abajo una imagen de la web realizada, en ella podemos observar y configurar los datos que para el usuario sean más relevantes.

Medición Distribuida de ENERGÍA

Inicio Quienes Somos Testimonios Proyectos Contacto

Listado de Medidores

- Medidor 1 : " Televisor "
- Medidor 2 : " Microondas "
- Medidor 3 : " Heladera "

Medidores activos

Últimos registros del medidor "Televisor"

Tension RMS	Tension Pico	Corriente RMS	Corriente Pico
246.69 [V]	510.67 [V]	0.03 [A]	-0.09 [A]
Potencia Activa	Potencia Reactiva	Potencia Aparente	Factor de Potencia
5.33 [W]	6.55 [VAr]	8.44 [VA]	0.63

Última lectura: 2018-06-26 09:25:12

CONFIGURACIÓN

Cambiar Nombre:

Establecer Fecha:

Mostrar Gráfica:

Consumo Energético

Potencia [Watt]

Consumo

Mediciones instantáneas más relevantes

Configuración

Gráficos de históricos

Innovaciones | IOT | Energía | MQTT

Diseño exclusivo para exposición de Proyecto Final, diseñado por Catinot Francisco, Ludi Marino, Weber Adrian

Ilustración 31-Nuestra web.

Se podrá observar a un lateral la cantidad de medidores puntuales que están reportando datos (Medidores activos). En la parte de configuración el usuario podrá configurar el nombre del medidor y establecer la fecha y hora en el caso que sea necesario.

Al hacer click sobre algunos de los medidores, el usuario podrá visualizar valores instantáneos de Tensión y Corriente RMS, Potencia Activa, Aparente, Reactiva, Factor de Potencia, entre otros.

También se puede visualizar en forma gráfica los históricos de consumo, Corriente y Tensión RMS. Teniendo la posibilidad de hacer zoom en el punto de grafica que se desee.

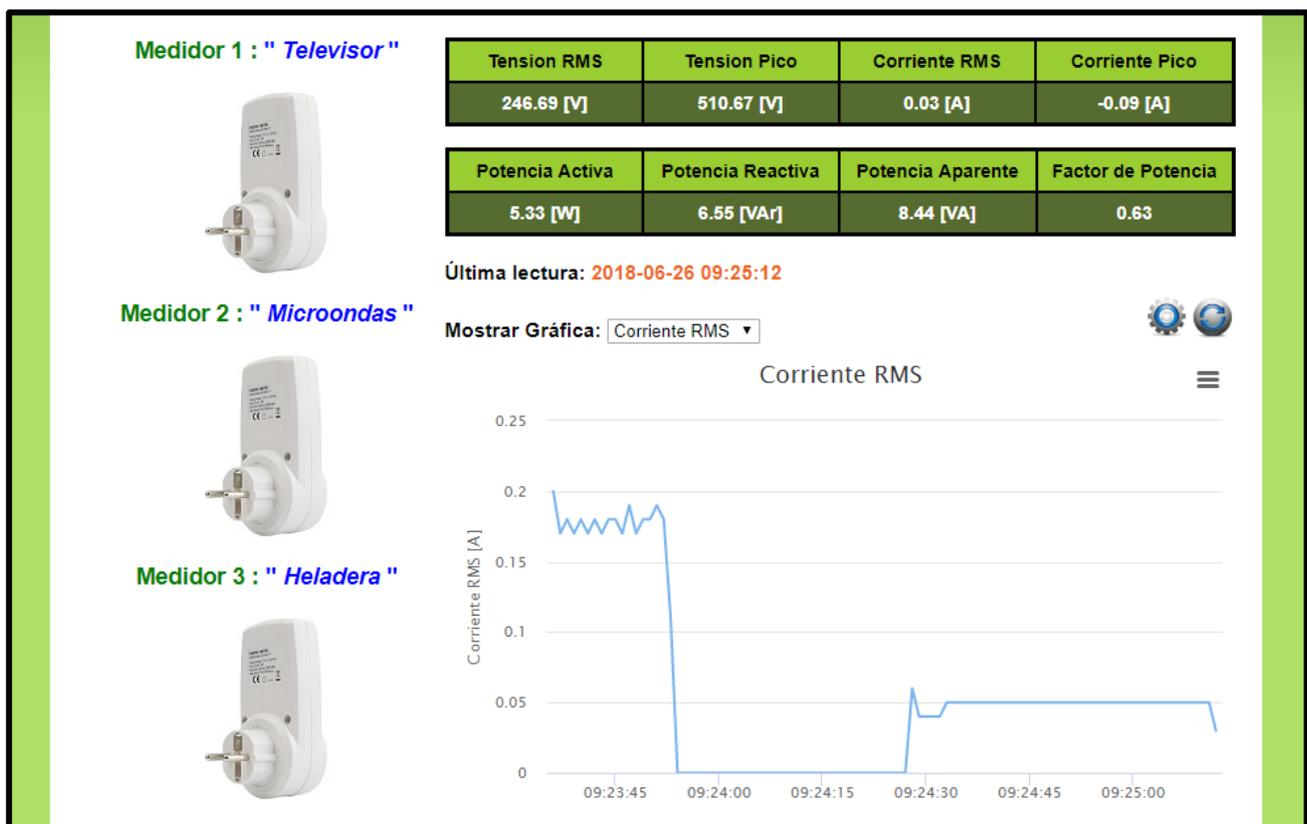


Ilustración 32-Nuestra web, selección.

Para acceder al menú de configuración se debe hacer click en la zona del engranaje.

También existe la posibilidad de hacer un reload con el icono que tienen las flechas.

En la zona de gráficos, donde se puede visualizar el histórico de las mediciones, se tiene la opción de seleccionar la gráfica que se desea visualizar, con la posibilidad de descargarlas en diferentes formatos haciendo click en el icono de las tres líneas horizontales.

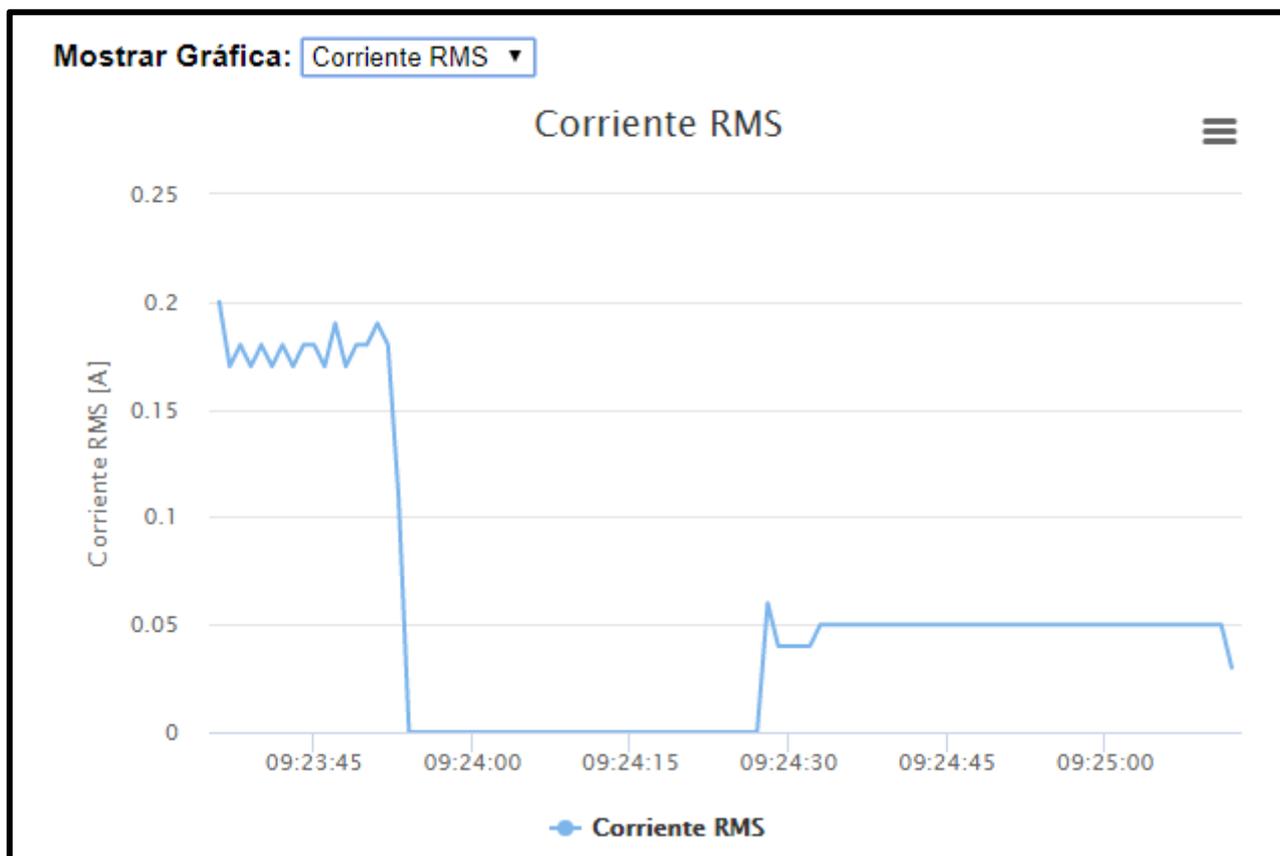


Ilustración 33-Nuestra web, zona gráficas.

2.16 Código QR.

Cada uno de los medidores puntuales de energía, tendrán un Código QR. Leyendo este código con un Smartphone el dispositivo ingresará a una página web donde se deberán ingresar el nombre y contraseña de la red WiFi en la que se desea conectar el datalogger. Si los datos ingresados son correctos, la conexión será automática.

Los códigos QR (Quick Response), respuesta rápida; son códigos de barras bidimensionales, que fueron desarrollados por la compañía japonesa Denso Wave, en 1994.

Respecto a los datos que puede manejar, sé que pueden contener hasta 4.200 caracteres alfanuméricos, es decir, letras, números y caracteres.

La matriz de puntos en la que se guardan los datos no es legible para el ojo humano. Se debe leer con un Smartphone o con un dispositivo que disponga de la aplicación correspondiente (un lector de códigos QR). La lectura del código se lleva a cabo en cuestión de segundos. Además, gracias a la corrección de errores, la lectura también funciona si falta alguna pieza en el código.

Los primeros códigos QR fueron utilizados por el fabricante de automóviles japonés Toyota. Sirven para el marcado y registro de piezas y todo tipo de componentes.

Muy a menudo los códigos QR aparecen en revistas y periódicos. Al escanearlos, el lector accede de una manera fácil y directa a determinados sitios web sin que sea necesario, por ejemplo, teclear direcciones URL largas y complejas.

Los códigos QR utilizados en este proyecto se generaron por medio de la web gratuita <http://www.codigos-qr.com>, donde se puede regular el grado de redundancia (baja, media, alta y muy alta). Abajo podemos visualizar el código QR para ingresar a la web de la UTN FRP. [15]



Ilustración 34-Código QR, para acceder a la web UTN-FRP

Capítulo 3: Verificación de la calidad del instrumento.

Tuvimos la gran oportunidad de tener un compañero trabajando en la EPE (Empresa Provincial de la Energía de Santa Fe). Esta empresa distribuidora de energía, posee bancos de carga y patrones fijos y portátiles, que utiliza para realizar las verificaciones periódicas de los medidores de energía comerciales, donde pudieron realizarse pruebas en la etapa de diseño de los medidores desarrollados y calibrarlos a fin de determinar la clase de estos instrumentos.

De entre los patrones con los que se contaba, el utilizado fue una “Fuente patrón para medidores de energía” marca “MTE”, modelo “PTS3.3C” clase 0.1, que como todos los equipos del laboratorio cuentan con la verificación del INTI (Instituto Nacional de Tecnología Industrial) que certifica su clase y correcto funcionamiento.

El proceso de contraste consiste en que el dispositivo contrastador inyecte valores de corriente y tensión al dispositivo a medir y al instrumento patrón al mismo tiempo, para luego comparar los resultados de ambos. La comunicación entre el patrón y el medidor se establece mediante destellos de un led perteneciente al medidor que son captados por un sensor óptico con el que cuenta el patrón.

Los destellos del led antes mencionado responden a una constante dada en imp/kWh (Impulsos por kilo watt hora) que en el caso de los medidores desarrollados es de 2815 [imp/kWh]

Los ensayos realizados fueron los especificados en el “Anexo B: Reglamento de verificación periódica para medidores de energía” que forma parte del “Secretaría de Comercio Interior, Metrología Legal- Resolución 90/2012. <Reglamento técnico y metrológico para los medidores de energía eléctrica activa en corriente alterna>”. [18]

En lo concreto, los ensayos realizados fueron:

- Ensayo de variación de corriente.
- Ensayo de integración.
- Ensayo de arranque.
- Ensayo de marcha en vacío.

Los resultados de los mismos se adjuntan como anexo en el presente documento y determinan que el medidor desarrollado es clase 2, según lo especificado en la Tabla 13 – “Límites de errores porcentuales” de la norma IRAM 2420. [20]

Valor de la corriente		Factor de potencia	Límites de errores porcentuales para los medidores de clase	
			1	2
Para medidores directos	Para medidores alimentados por transformadores			
$0,05I_b \leq I < 0,1I_b$	$0,02I_n \leq I < 0,05I_n$	1	$\pm 1,5$	$\pm 2,5$
$0,1I_b \leq I < I_{max}$	$0,05I_n \leq I < I_{max}$	1	$\pm 1,0$	$\pm 2,0$
$0,1I_b \leq I < 0,2I_b$	$0,05I_n \leq I < 0,1I_n$	0,5 inductivo	$\pm 1,5$	$\pm 2,5$
		0,8 capacitivo	$\pm 1,5$	-
$0,2I_b \leq I < I_{max}$	$0,1I_n \leq I < I_{max}$	0,5 inductivo	$\pm 1,0$	$\pm 2,0$
		0,8 capacitivo	$\pm 1,0$	-
Por requerimiento especial del usuario:		0,5 inductivo	$\pm 3,5$	-
$0,2I_b \leq I < I_b$	$0,1I_n \leq I < I_n$	0,8 capacitivo	$\pm 2,5$	-

Tabla 2-Verificación de la calidad del instrumento.

3.1: Fotos del Laboratorio de Medidores de la Empresa Provincial de Energía de Santa Fe (EPE).



Ilustración 35-Foto 1, Laboratorio de Medidores de EPE.



Ilustración 36-Foto 2, Laboratorio de Medidores de EPE.



Ilustración 37-Foto 3, Laboratorio de Medidores de EPE.



Ilustración 38-Foto 4, Laboratorio de Medidores de EPE.

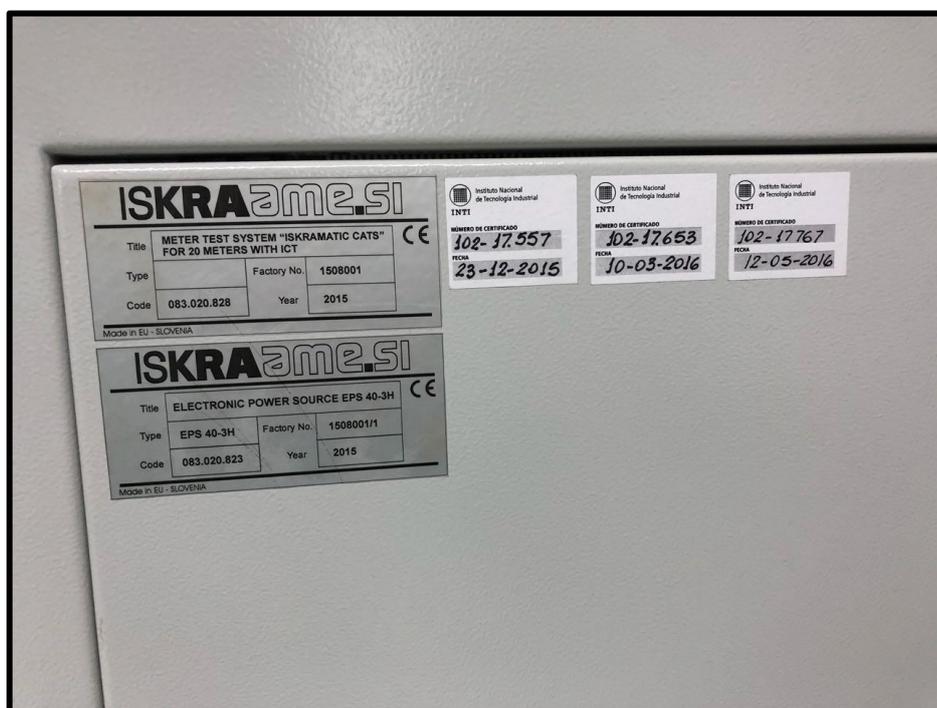
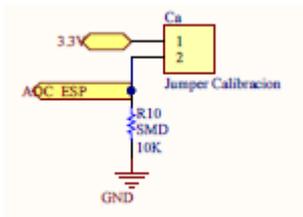
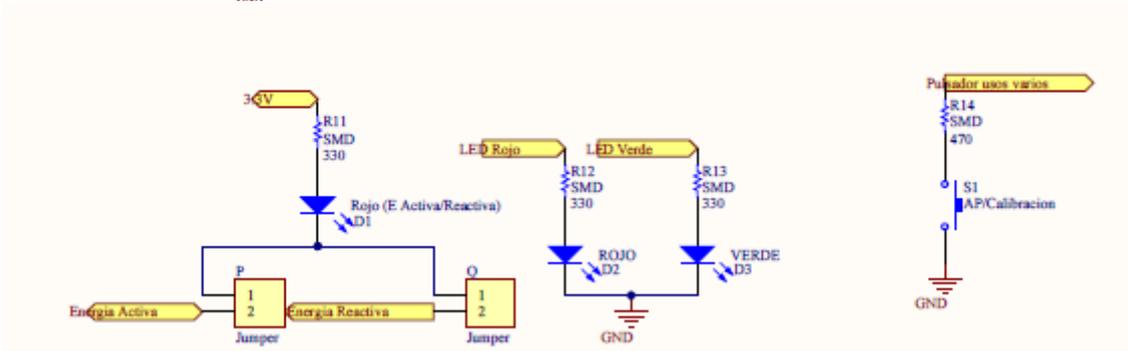
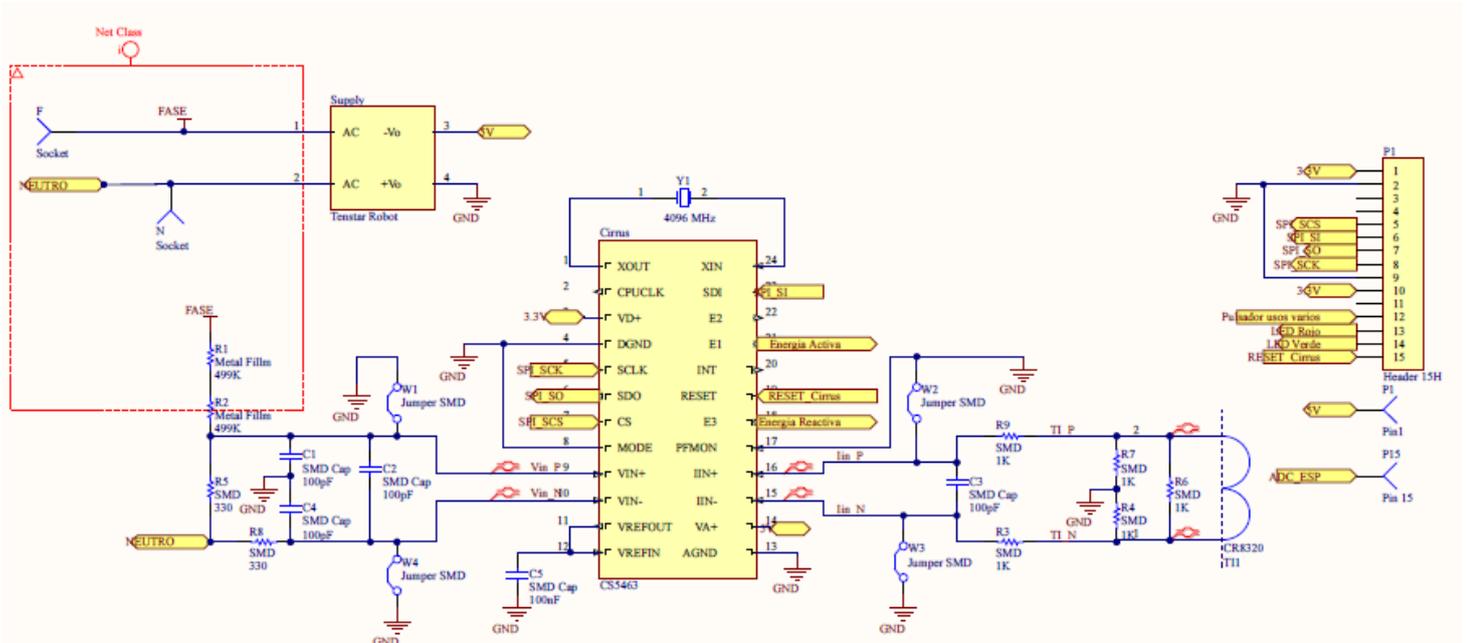


Ilustración 39-Equipos aprobados por INTI.

Capítulo 4: Hardware.

4.1 Esquemático.



4.2 Imágenes 3D.

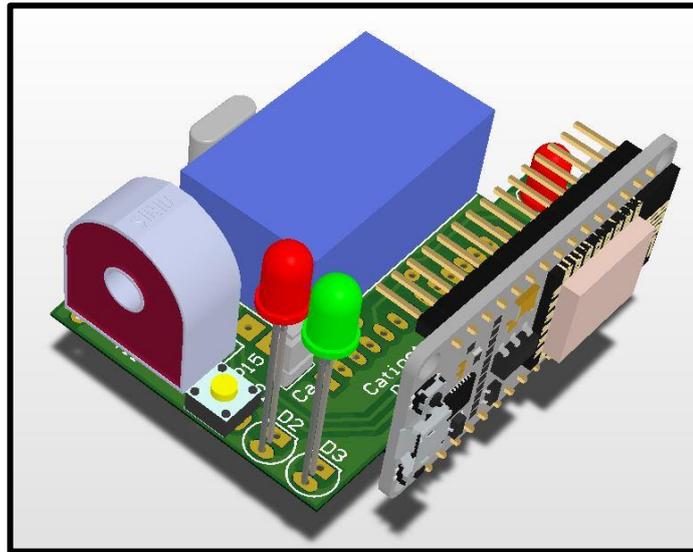


Ilustración 40-3D superior derecha.

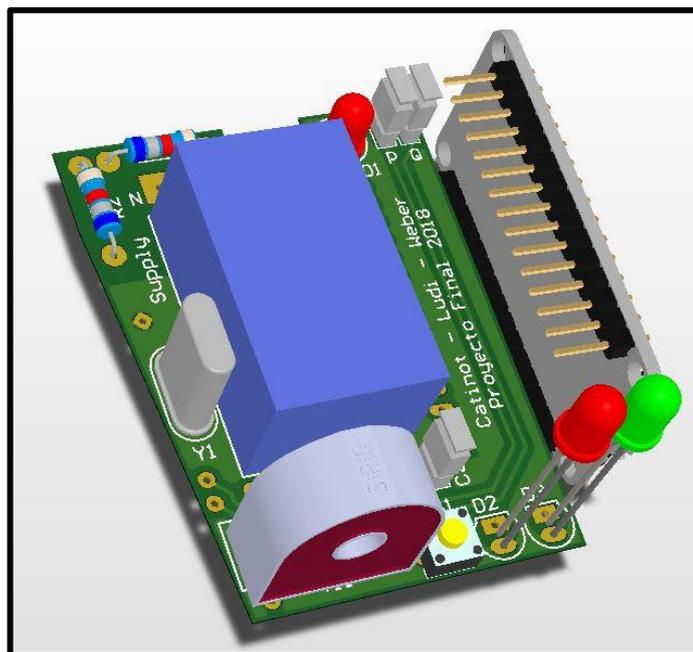


Ilustración 41-3D superior izquierda

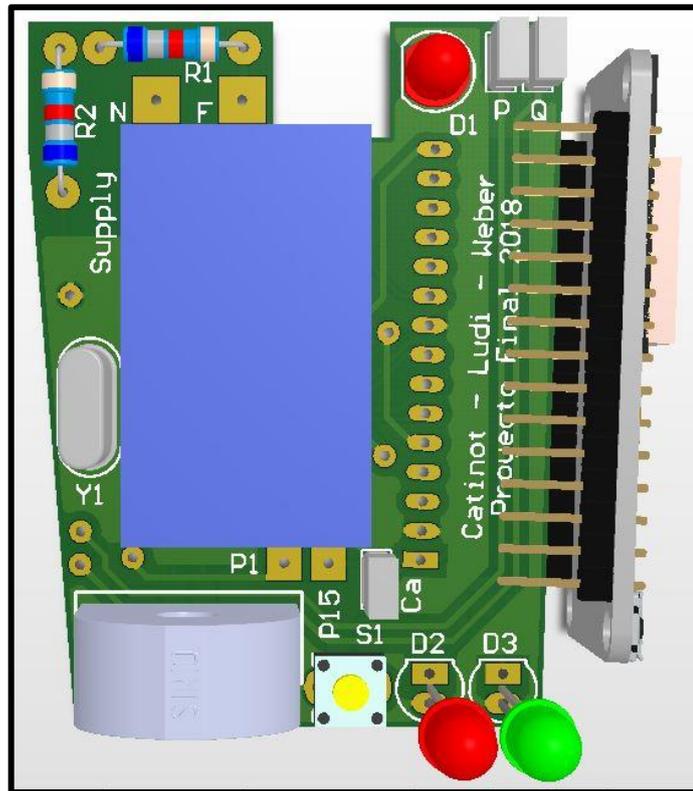


Ilustración 42-3D superior.

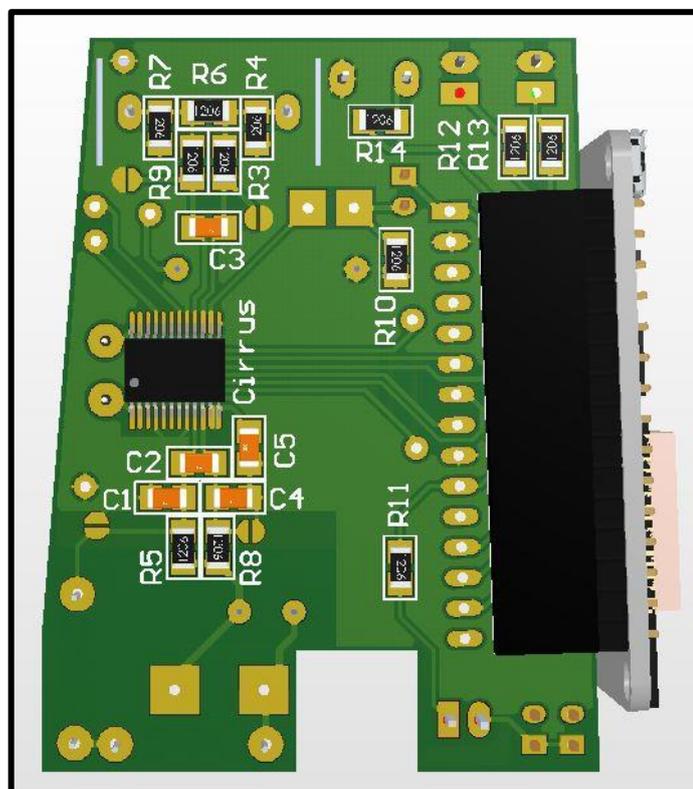


Ilustración 43-3D inferior.

4.3 Hardware terminado.



Medidas 87x57x44 mm

Ilustración 44- medidor puntual en funcionamiento.

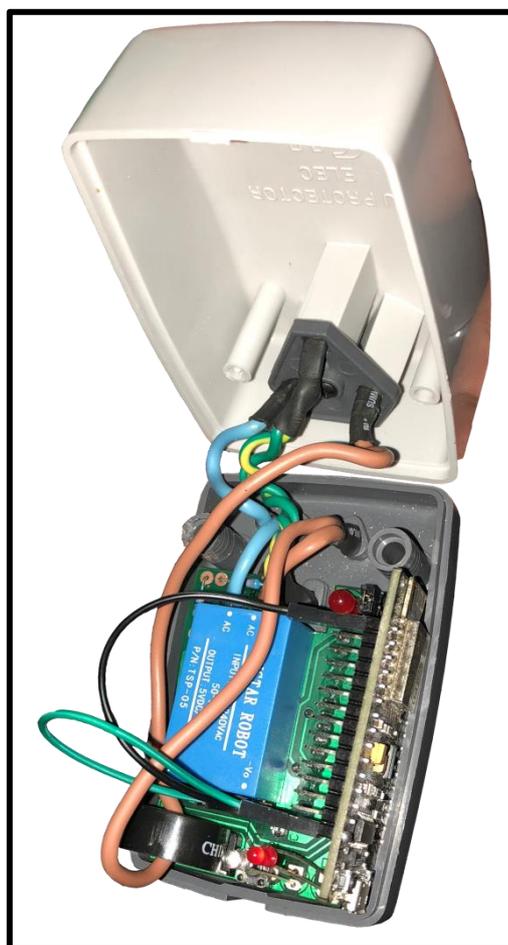


Ilustración 45- Interior medidor.

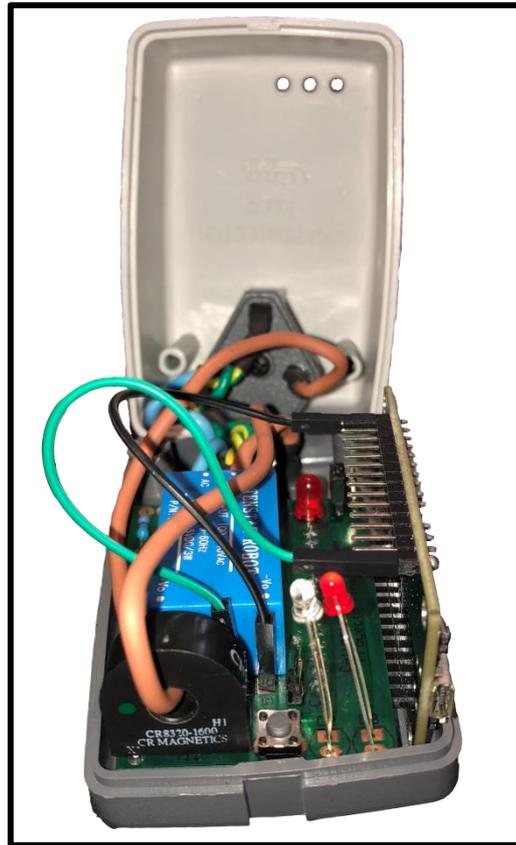


Ilustración 46-Interior medidor.

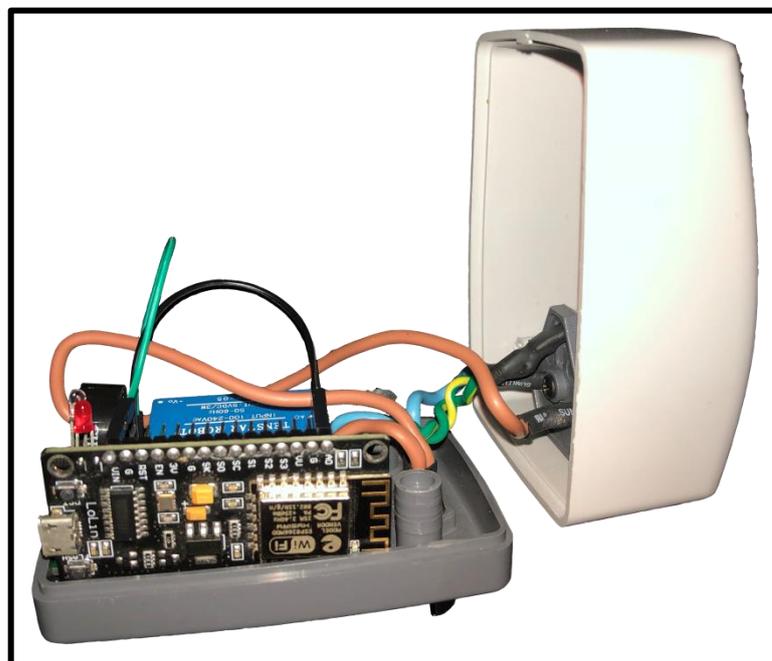


Ilustración 47-Medidor lateral.



Medidas 110x106x35 mm

Ilustración 48-Superior Concentrador de datos.

5.1 Análisis de costos por unidad.

Del total de componentes se sumó un 50%, debido a las horas hombre aportadas al proyecto.

Datalogger		
Artículos	Cantidad	Costo [\\$]
Raspberry Pi	1	1.200
Cable alimentación	1	250
Carcaza	1	230
Componentes electrónica	1	10
Total Componentes		1.690
Medidor puntual		
Artículos	Cantidad	Costo [\\$]
Node MCU	1	235
Medidor de energía	1	80
Circuito impreso	1	100
Componentes electrónica	1	250
Carcaza	1	165
Total Componentes		830

Tabla 3-Costo por unidad Datalogger/Medidor puntual.

5.2 Análisis de costos por cantidad.

Se calculó para construir 100 dataloggers y 300 medidores puntuales, se pueden conseguir los componentes a un 15% menos que al comprarlos de forma individual.

Quedando los costos de la siguiente manera:

Datalogger		
Artículos	Cantidad	Costo [€]
Raspberry Pi	1	1020
Cable alimentación	1	212,5
Carcaza	1	195,5
Componentes electrónica	1	8,5
Total Componentes		1.436,5
Medidor puntual		
Artículos	Cantidad	Costo [€]
Node MCU	1	199,75
Medidor de energía	1	68
Circuito impreso	1	85
Componentes electrónica	1	212,5
Carcaza	1	140,25
Total Componentes		705,5

Tabla 4-Costo por cantidad Datalogger/Medidor puntual.

5.3 Retorno sobre la inversión (ROI).

Haciendo una proyección a 12 meses y suponiendo que se concretan 8 ventas al mes, se calcula.

$$\text{ROI} = [(\text{Ingresos} - \text{Inversión}) / \text{Inversión}] \times 100 =$$

$$\text{ROI} = [((1436,5 + 705,5 \times 3) \times 1,5 \times 8 \times 12) - (1436,5 \times 100 + 707,5 \times 300)] / (1436,5 \times 100 + 707,5 \times 300) \times 100$$

$$\text{ROI} = 44\%$$

El ROI (Return On Investment), hace referencia a la métrica que expresa la relación entre lo invertido en un negocio y el beneficio obtenido proveniente de dicha inversión. Suele utilizarse a menudo para cuantificar si la inversión vale la pena desde el punto de vista financiero. En nuestro caso, nos dio un 44%, lo que es número muy beneficioso para nuestro proyecto de inversión.

5.5 Plan de marketing e inversión.

El producto se podría comercializar tanto en el mercado interno como en el externo. En ambos casos la metodología será mediante e-commerce. La idea es construir una página web donde podrán encontrarse descripciones del producto, especificaciones técnicas, fotos, videos explicativos, principales clientes y además un apartado referido a “contacto” donde el potencial cliente encontrará teléfonos y correos electrónicos para solicitar cotizaciones. De concretarse la venta podrá optar por realizar transferencias bancarias o bien abonar mediante tarjetas de crédito o servicios como MercadoPago.

En lo que refiere a publicidad, se podría realizar en redes sociales como Facebook e Instagram, programas televisivos del rubro tecnológico, notas en los principales diarios y artículos publicitarios en revistas orientadas al rubro eléctrico o al de las PYME.

Si bien en principio este desarrollo es netamente por cuestiones académicas, hemos tenido varias propuestas para poder llegar a comercializarse. Todas serán evaluadas por el grupo de trabajo.

5.7 Validación del plan de negocios.

En la investigación de mercado que realizamos pudimos observar que las empresas competidoras del continente Europeo poseen una gran variedad de productos desarrollados en este rubro, más de 10 años de trayectoria y distribuidores en otros países, sin dudas, características que hablan de una muy buena recepción del producto por parte de esos mercados. Por otra parte, se tuvo la posibilidad de hablar con áreas comerciales de la EPESF, y mencionaron el notable incremento de los reclamos de clientes solicitando se verifique el correcto funcionamiento de sus medidores, debido a que no creían viables los montos de las facturas recibidas. Esta situación es sin dudas propicia para la instalación del producto a desarrollar.

Capítulo 6: Conclusión y líneas futuras.

A modo de conclusión podría decirse los resultados del proyecto fueron muy satisfactorios y que de hecho superaron las expectativas planteadas al inicio del mismo. La clase de los medidores de energía resulto ser muy buena, el tamaño de medidor conseguido resulta cómodo para poder ser utilizado en instalaciones de todo tipo y además, por medio del uso de herramientas como códigos QR y generación de puntos de acceso, se consiguió que tanto la instalación como la puesta en marcha y utilización del producto sean sumamente simples para cualquier usuario.

Otro punto a resaltar de los logros conseguidos es el protocolo de autodescubrimiento, mediante el cual los medidores de energía y el concentrador de datos pueden encontrarse para establecer una comunicación independientemente del IP que les haya asignado el router del establecimiento en que se instalen. Sin dudas es una de las características principales que lo vuelven aplicable y comercializable.

Sin embargo, como en todo proyecto, siempre existe la posibilidad de realizar mejoras. En este caso algunas de las ideas que pueden ser desarrolladas en líneas futuras, podrían ser:

- Realizar una versión del producto preparado para tomacorrientes de 20 [A]. Para ello se debería cambiar el transductor de corriente por uno que admita una corriente de 20 [A] y además debería conseguirse una carcasa dotada de un enchufe y tomacorriente de esta característica.
- Realizar los mismos dispositivos medidores de energía, pero enfocado a la industria, medidores trifásicos, de mayor amperaje, con sistemas de comunicación adaptados para la industria. En éste caso se podría cambiar hasta la estrategia de mercado, se podría ofrecer un servicio, con un análisis detallado de los resultados del mismo, pudiendo ofrecer a la industria que maquinaria renovar primero, que tipo de energía alternativa se pudiese utilizar, etc.
- Se podrían replantear, ampliar y mejorar el acceso a las funcionalidades que brinda la web del concentrador de datos al usuario
- Podría sumarse la posibilidad de encender y apagar un electrodoméstico conectado al medidor de energía, desde la web.

Literatura Citada.

- [1] "Wabee". [En línea]. Disponible en: <http://www.wabee.com.ar>
[Accedido: 02-Marzo- 2018].
- [2] "Efergy". [En línea]. Disponible en: <http://www.efergy.com>
[Accedido: 02-Marzo- 2018].
- [3] "CS5463". [Hoja de datos]. Se anexa al final del informe
[Accedido: Marzo/Abril/Mayo- 2018].
- [4] "NodeMcu". [En línea]. Disponible en: http://nodemcu.com/index_en.html
[Accedido: 16-Marzo- 2018].
- [5] "ArduinoIDE". [En línea]. Disponible en:
<https://www.arduino.cc/en/main/software>
[Accedido: 23-Marzo- 2018].
- [6] "MQTT". [En línea]. Disponible en: <http://mqtt.org/documentation>
[Accedido: 01-Abril- 2018].
- [6] "Eficiencia en el uso de la energía eléctrica". [Libro]. 2011, CIRCUTOR SA
Autores: Jordi Autoneel, Josep Balcells, Vicente Barra, Francesc Foniales, Bernat Garcia y Joan Ros.
- [7] "MariaDB". [En línea].
Disponible en: <https://mariadb.com/kb/es/what-is-mariadb-51/>
[Accedido: 06-Abril- 2018].
- [8] "Apache". [En línea]. Disponible en: <https://www.apache.org>
[Accedido: 06-Abril- 2018].

[9] “Mosquitto install”. [En línea].

Disponible en: <https://jpmens.net/2013/09/01/installing-mosquitto-on-a-raspberry-pi/>
[Accedido: 08-Abril- 2018].

[10] “Mosquitto”. [En línea]. Disponible en: <https://mosquitto.org/>

[Accedido: 08-Abril- 2018].

[11] “HTTP”. [En línea]. Disponible en: <http://neo.lcc.uma.es>

[Accedido: 11-Abril- 2018].

[12] HTML”. [En línea]. Disponible en: <http://developer.mozilla.org/es/docs/Web/HTML>

[Accedido: 13-Abril- 2018].

[13] “PHP”. [En línea]. Disponible en: <http://php.net/manual/es/intro-what-is.php>

[Accedido: 16-Abril- 2018].

[14] “JavaScript”. [En línea]. Disponible en: <https://www.aprenderaprogramar.com>

[Accedido: 16-Abril- 2018].

[15] “Codigo QR”. [Tesis].

Disponible en: <https://gredos.usal.es/jspui/bitstream/10366/129651/1/>

[/TFG_InfyDoc_LiuCan_SI_95_2014-2015.pdf](#) [TFG_InfyDoc_LiuCan_SI_95_2014-2015.pdf](#)

[Accedido: 16-Abril- 2018].

[16] “Raspberry Pi”. [En línea].

Disponible en: <https://www.raspberrypi.org/downloads/raspbian>

[Accedido: 24-Abril- 2018].

[17] “Raspberry Pi”. [En línea].

Disponible en: <https://www.enmimaquinafunciona.com/pregunta/57673/como-saber-que-servicios-ejecutan-al-iniciar-en-frambuesa-pi-raspbian>

[Accedido: 28-Abril- 2018].

[18] “Secretaria de Comercio Interior, Metrología Legal”. [En línea].

Disponible en: <https://www.economia.gob.ar/concursos/biblio/RES%2090-12%20SECR%20COMERCIO%20INTERIOR.pdf>

[Accedido: 01-Mayo- 2018].

[19] “UDP”. [En línea].

Disponible en: <http://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/udp.html>

[Accedido: 04-Mayo- 2018].

[20] “Norma IRAM 2420”. [En línea/pago].

Disponible en: <http://www.iram.org.ar/>

“Medidores estáticos de energía eléctrica activa para corriente alterna (Clases 1 y 2)”

[Accedido: 25-julio- 2018].

ANEXOS: