



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MEDICIÓN DE
CONSUMO ELÉCTRICO INALÁMBRICO PARA ARTÍCULOS DEL HOGAR
MONITOREADO MEDIANTE UNA PÁGINA WEB

AUTOR

Christian Gustavo López Calvachi

AÑO

2018



FACULTAD DE INGENIERÍAS Y CIENCIAS AGROPECUARIAS

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE MEDICIÓN DE
CONSUMO ELÉCTRICO INALÁMBRICO PARA ARTÍCULOS DEL HOGAR
MONITOREADO MEDIANTE UNA PÁGINA WEB.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Electrónica y Redes de
Información.

Profesor Guía

MSc. Héctor Fernando Chinchero Villacis

Autor

Christian Gustavo López Calvachi

Año

2018

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido el trabajo, diseño e implementación de un prototipo de medición de consumo eléctrico inalámbrico para artículos de hogar monitoreado mediante una página web, a través de reuniones periódicas con el estudiante Christian Gustavo López Calvachi, en el semestre 2018-1, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Héctor Fernando Chinchero Villacis

Master en Domótica y Hogar Digital

CI: 1715451330

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, diseño e implementación de un prototipo de medición de consumo eléctrico inalámbrico para artículos de hogar monitoreado mediante una página web, del estudiante Christian Gustavo López Calvachi, en el semestre 2018-1, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”.

Luis Santiago Criollo Caizaguano

Magister en Redes de Comunicación

CI: 1717112955

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”.

Christian Gustavo López Calvachi

CI: 1724406309

AGRADECIMIENTOS

Agradezco a Dios por darme la vida y la facultad de realizar este proyecto de titulación. A mis padres por ser mi mayor bendición desde el día en que me trajeron al mundo y son el pilar fundamental para mi proceso de formación: académica y profesional, a mi gran amigo y tutor guía Hector por su: paciencia, sabiduría y empeño a lo largo de todo este proceso que me ha encaminado al éxito y a todas las personas a lo largo de mi carrera que han estado presentes en los momentos más importantes de mi vida. Gracias infinitas.

DEDICATORIA

Dedico este trabajo de titulación a mi abuelito paterno Antonio (+) de manera muy especial por sus sabios y oportunos consejos; por siempre creer en mí, por motivarme cada día a ser mejor, a superar mis miedos y salir victorioso de cualquier reto.

RESUMEN

El presente trabajo de titulación está destinado al desarrollo de un prototipo inalámbrico medidor de consumo energético; los resultados son almacenados en una plataforma Raspberry Pi y monitoreados mediante una página web con acceso directo a la plataforma mediante la dirección IP. La metodología que se utiliza es la de obtención de datos de potencia por medio del medidor de consumo eléctrico; todos estos valores son enviados al servidor Raspberry Pi que almacena y despliega los datos obtenidos en tiempo real en una interfaz gráfica amigable al usuario final.

El proyecto consta de artefactos tales como: medidor de potencia wifi, Raspberry Pi, Router y dispositivos finales como: smartphones, tables pc y/o laptops en los que se realiza toda la parte que corresponde al monitoreo del prototipo.

En el capítulo I se definen algunos términos eléctricos de utilidad, se expone el análisis de tecnologías a utilizar y varios de los dispositivos tradicionales y Smart existentes que permiten realizar mediciones en el entorno residencial.

El capítulo II corresponde al desarrollo e implementación del prototipo a partir de los objetivos planteados y las investigaciones realizadas.

En el capítulo III se detalla todo lo referente a los dispositivos de prueba que se utilizan, y los resultados que se obtienen después de realizar varias mediciones de varios artefactos del hogar en distintos escenarios.

En el capítulo IV se exponen las conclusiones y recomendaciones obtenidas a partir del desarrollo y ejecución de pruebas del prototipo.

La sección de anexos corresponde a toda la documentación de los datasheets de los dispositivos utilizados en el desarrollo del prototipo, el código de programación de cada elemento y el código fuente de la página web que se describe como la interfaz final del usuario.

ABSTRACT

The present titling work is destined to the development of a wireless meter prototype of energetic consumption; the results are stored on a Raspberry Pi platform and monitored through a web page with direct access to the platform through the IP address. The methodology used is the obtaining of power data through the electric consumption meter. These values are sent to the Raspberry Pi server that stores and displays the data obtained in real time in a friendly graphical interface.

The project consists of devices such as: Wi-Fi power meter, Raspberry Pi, Router and final devices such as: smartphones, tablets, pc and / or laptops in which all the part corresponding to the monitoring of the prototype is performed.

Chapter I defines some useful electrical terms, exposes the analysis of technologies to be used and several of the existing traditional and Smart devices that allow measurements in the residential environment.

Chapter II corresponds to the development and implementation of the prototype based on the objectives and research carried out.

Chapter III details everything related to the test devices used, and the results obtained after making several measurements of various household appliances in different scenarios.

Chapter IV presents the conclusions and recommendations obtained after the development and execution of prototype tests.

The section of annexes corresponds to all the documentation of the datasheets of the devices used in the development of the prototype, the programming code of each element and the source code of the web page that is described as the user's final interface.

ÍNDICE

1. CAPITULO I. INTRODUCCIÓN.....	1
1.1 Alcance	2
1.2 Justificación	2
1.3 Objetivos.....	3
1.3.1 General	3
1.3.2 Específicos	3
2. CAPÍTULO II. ANÁLISIS DE TECNOLOGÍAS Y MEDIDORES DE CONSUMO ELÉCTRICO.....	4
2.1 Definición de Términos Eléctricos	4
2.1.1 Corriente Eléctrica	4
2.1.1.1 Corriente Directa (DC)	5
2.1.1.2 Corriente Alterna (AC)	5
2.1.2 Potencia.....	5
2.1.2.1 Relación entre voltaje (V), corriente (I) y potencia (P)	6
2.1.3 Voltaje.....	7
2.2 Tecnologías de transferencia de información.....	8
2.2.1 Tecnología ADSL.....	8
2.2.2 Tecnología por Radio Frecuencia.....	9
2.2.3 Tecnología Wifi	10
2.2.3.1 Definición y Funcionamiento.....	10
2.2.3.2 Protocolos de Wifi IEEE 802.11	10
2.2.3.3 Ventajas y Desventajas de Wifi	12
2.2.4 Protocolos de Comunicación	13

2.2.4.1	Protocolo Serial Peer to peer.....	13
2.2.4.2	Comunicación en Serie RS 485.....	13
2.2.4.3	Protocolo M-BUS.....	14
2.2.4.3.1	Relación con el modelo OSI.....	14
2.2.4.4	IP.....	15
2.2.4.5	TCP/IP.....	16
2.2.4.5.1	TCP.....	16
2.2.4.6	UDP.....	17
2.2.4.7	HTML.....	17
2.2.4.7.1	Estructura de HTML.....	17
2.2.4.8	MQTT.....	18
2.3	Medidores de consumo eléctrico tradicionales.....	18
2.3.1	El Contador.....	19
2.3.2	Contador electrónico.....	19
2.3.3	Contador inteligente.....	20
2.3.4	Contador de prepago.....	21
2.3.5	Interruptor horario.....	21
2.3.6	Pinza Amperimétrica.....	22
2.4	Medidores de consumo eléctrico smart.....	23
2.4.1	MeterBUS-4C.....	23
2.4.1.1	Datos Técnicos.....	24
2.4.2	Contador de Pulsos Eléctricos KAMSTRUP 382.....	24
2.4.2.1	Definición y Aplicación.....	24
2.4.2.2	Datos Técnicos.....	25
2.4.3	Conmutador inalámbrico wifi Sonoff POW.....	26
2.4.3.1	Características Virtuales.....	26

2.4.3.2 Características Técnicas	27
2.4.3.3 Diagrama de Conexión	27
2.4.3.4 Ventajas del medidor Sonoff Pow vs otros medidores smart.....	28
2.4.3.5 ESP 8266	29
2.4.3.5.1 Modelos.....	29
2.4.3.5.2 Usos y Aplicaciones	30
2.4.3.5.3 Conexión Serial a TTL.....	31
2.5 Lectura automática de medidores (AMR)	32
2.5.1 Origen	32
2.5.2 Principio de Funcionamiento.....	32
2.5.3 Componentes Principales de AMR	33
2.5.3.1 Modulo de Interfaz del medidor	33
2.5.3.2 Sistema de Comunicaciones	34
2.5.3.3 Sistema de Equipamiento de la Oficina Central	34
2.5.4 Métodos de Comunicación	34
2.5.4.1 Líneas Telefónicas	34
2.5.4.2 Portadora de línea Eléctrica	35
2.5.4.3 Radio Frecuencia	35
2.5.4.4 Comunicación por televisión por cable	35
2.5.5 Tablas Comparativas	35
2.6 Entidades y Datos Estadísticos del consumo eléctrico en Ecuador	37
2.6.1 Arconel.....	37
2.6.2 Consumo per cápita.....	37
2.7 Norma ISO 50001: Gestión de Energía	38
2.7.1 Propósito.....	38

2.7.2 Modelo de Sistema de Gestión de la Energía.....	39
2.8 Highcharts	40
2.9 Raspberry Pi.....	41
2.9.1 Definición e Historia.....	41
2.9.2 Raspberry Pi 3 Model B	42
2.9.3 Componentes de la placa Raspberry.....	43
2.9.4 Componentes Externos	44
3. CAPÍTULO III. DESARROLLO E IMPLEMENTACIÓN	
DEL PROTOTIPO.....	46
3.1 Planteamiento del Sistema Global	46
3.1.2. Especificaciones del Diagrama del Sistema	47
3.1.3 Especificaciones Técnicas del Sistema	48
3.2 Prototipo	49
3.2.1 Diseño del Prototipo.....	50
3.2.2 Diagrama del Prototipo	51
3.2.3 Descripción Técnica del Prototipo	51
3.2.4 Capas de interacción con el modelo OSI.....	53
3.3 Desarrollo del prototipo.....	53
3.3.1 Determinar el módulo que posee el integrado ESP8266	54
3.3.2 Determinar el medidor ideal para el prototipo	54
3.3.3 Búsqueda de las librerías para la obtención de potencia.....	55
3.3.4 Escritura del firmware al Sonoff POW.....	55
3.3.5 Programación de Sonoff POW.....	56
3.3.6 Verificar conexión entre Sonoff y un localhost.	57
3.3.7 Configuración Inicial de Raspberry Pi 3.....	58

3.3.7.1	Descarga del Sistema Operativo	58
3.3.7.2	Instalación del Sistema Operativo	59
3.3.7.3	Integración de tarjeta Micro SD con Raspberry Pi 3.....	60
3.3.7.4	Conexión de Raspberry Pi a la red.....	61
3.3.7.5	Habilitación de Interfaces	61
3.3.7.6	Descarga e Instalación de VNC Viewer.....	62
3.3.8	Instalación del Servidor Web	65
3.3.8.1	Actualización de la Raspberry	65
3.3.8.2	Instalación de Apache	65
3.3.8.3	Instalación de PHP	66
3.3.9	Desarrollo del Entorno Web.....	67
3.3.9.1	DataDetecEsp.php	68
3.3.9.2	EncerarEsp.php.....	69
3.3.9.3	LeerValEsp.php.....	70
3.3.10	Escoger Plantilla de HighCharts	71
3.3.11	Interfaz de Presentación	72
3.4	Prototipo Final	72
4.	CAPÍTULO IV. PRUEBAS Y RESULTADOS	74
4.1	Pruebas de Conexión.....	74
4.2	Pruebas de Funcionamiento.....	75
4.2.1	Prueba 1: Node MCU con datos aleatorios.....	75
4.2.2	Prueba 2: Prototipo con bobilla incandescente.....	75
4.2.2	Prueba 3: Prototipo con bombilla ahorradora	77
4.2.4	Prueba 4: Smart TV LG	78
4.2.5	Prueba 5: Plancha eléctrica	81

4.3 Dispositivos Finales de Monitoreo.....	83
4.4 Resultados Finales.....	83
5. CONCLUSIONES Y RECOMENDACIONES.....	85
5.1 Conclusiones.....	85
5.2 Recomendaciones.....	87
REFERENCIAS.....	88
ANEXOS.....	91

ÍNDICE DE FIGURAS

Figura 1. Representación de la Corriente Directa.	5
Figura 2. Representación de la Corriente Alterna.	5
Figura 3. El Voltaje en un circuito eléctrico.	7
Figura 4. Frecuencias Usadas en ADSL	8
Figura 5. Funcionamiento de la tecnología RFID.	9
Figura 6. Monitorización Remota de Medidores.	14
Figura 7. Partes que conforman una dirección IP.	16
Figura 8. Código Fuente de la página web de la Universidad de las Américas.	18
Figura 9. El Contador.	19
Figura 10. El contador electrónico.	20
Figura 11. El contador Inteligente.	20
Figura 12. El contador de prepago.	21
Figura 13. El Interruptor Horario.	22
Figura 14. La pinza amperimétrica	22
Figura 15. MeterBus de 4 Canales.	23
Figura 16. El Kamstrup 382.	25
Figura 17. Medidor de Potencia Wifi (Sonoff POW).	26
Figura 18. Funcionamiento del medidor Sonoff Pow.	28
Figura 19. ESP 8266.	29
Figura 20. Modelos del ESP.	29
Figura 21. Convertidor de USB a TTL.	32
Figura 22. Bosquejo del Funcionamiento de AMR.	33
Figura 23. Lectura Automática de Medidores.	34
Figura 24. Modelo del Sistema de Gestión de Energía para ISO 50001	39
Figura 25. Diagrama de Línea Básica.	40
Figura 26. Diagrama de Tiempo con líneas de intervalo.	41
Figura 27. Placa de pruebas (Alfa) Raspberry Pi.	42
Figura 28. Raspberry Pi 3 Model B V1.2.	42
Figura 29. Bosquejo de la placa Raspberry Pi.	43
Figura 30. Raspberry Pi con periféricos conectados	45

Figura 31. Planteo Inicial del Sistema.	46
Figura 32. Componentes que constituyen el diagrama del sistema	47
Figura 33. Descripción del diseño del prototipo.....	50
Figura 34. Bosquejo de los componentes del prototipo.....	51
Figura 35. Node MCU funcionando en el terminal de Arduino.	54
Figura 36. El Sonoff POW.	55
Figura 37. Diagrama de Flujo de Programación Sonoff Pow	56
Figura 38. Respuestas de ejecución exitosa entre Sonoff y el servidor local ...	57
Figura 39. Diagrama de la Configuración Inicial de Raspberry Pi 3	58
Figura 40. Página Oficial de Descargas Raspberry.....	59
Figura 41. USB Image Tool.	60
Figura 42. Mensaje Bienvenida al Escritorio de Raspberry Pi.....	60
Figura 43. Asignación de Dirección IP estática	61
Figura 44. Habilitación de servicios para conexión mediante SSH y VNC	62
Figura 45. Página Oficial RealVNC.	62
Figura 46. Establecimiento de Nueva Conexión.....	63
Figura 47. Autenticación para conexión por VNC.....	64
Figura 48. Interfaz del Sistema Operativo Raspbian	64
Figura 49. Servidor Apache corriendo sobre la Raspberry.....	66
Figura 50. PHP en Raspberry Pi.	67
Figura 51. Archivos del Entorno Web en la Raspberry Pi.	67
Figura 52. Diagrama de Flujo del Archivo “DataDetectEsp.php”	68
Figura 53. Diagrama de Flujo del Archivo “Encerar.php”	69
Figura 54. Diagrama de Flujo del Archivo “LeerValEsp.php”	70
Figura 55. Plantilla Modelo HighChart empleado.	71
Figura 56. Grafico Interfaz WEB.....	72
Figura 57. Presentación Final del Prototipo	73
Figura 58. Pruebas de Conexión hacia el Router y Raspberry Pi	74
Figura 59. Resultados del Node MCU.....	75
Figura 60. Medición de potencia con prototipo y bombilla incandescente.....	76
Figura 61. Resultados Gráficos Prueba 2	77
Figura 62. Medición de potencia con prototipo y foco ahorrador.....	77

Figura 63. Resultados Gráficos Prueba 3	78
Figura 64. Medición de potencia con prototipo y Smart TV LG	79
Figura 65. Resultados Gráficos Prueba 4	80
Figura 66. Medición de potencia con prototipo y plancha eléctrica	81
Figura 67. Resultados Gráficos Prueba 5	82
Figura 68. Pruebas de concurrencia en varios dispositivos finales.	83

ÍNDICE DE TABLAS

Tabla 1 Protocolos de Wi-Fi IEEE 802.11	12
Tabla 2 Ventajas y Desventajas de la tecnología Wifi.....	13
Tabla 3 Mapeo de la estructura M-Bus en las capas del modelo OSI.....	15
Tabla 4 Comparación de los modelos OSI y TCP/IP.....	16
Tabla 5 Datos Técnicos del Meter Bus de 4 Canales.....	24
Tabla 6 Datos técnicos del Kamstrup 382.....	25
Tabla 7 Características Virtuales del Sonoff POW.....	26
Tabla 8 Características Técnicas del Sonoff POW.....	27
Tabla 9 Ventajas del Sonoff POW frente a otros medidores Smart.....	28
Tabla 10 Aplicaciones del ESP 8266	30
Tabla 11 Descripción de los pines de Conexión.....	31
Tabla 12 Comparación de los métodos de medición.....	35
Tabla 13 Comparación de los medidores tradicionales y Smart.....	36
Tabla 14 Consumo per Capita Multianual Ecuador.	37
Tabla 15 Especificaciones Técnicas de los componentes del Sistema.....	48
Tabla 16 Especificaciones técnicas de los componentes del prototipo.	51
Tabla 17 Relación entre las capas del modelo OSI y el prototipo.	53
Tabla 18 Resultados Prueba 2	76
Tabla 19 Resultados Prueba 3	78
Tabla 20 Características Del Televisor Smart TV LG.....	79
Tabla 21 Resultados Prueba 4	80
Tabla 22 Resultados Prueba 5	81

1. CAPITULO I. INTRODUCCIÓN

El termino medición de consumo eléctrico se refiere principalmente al vocablo energía; esto abarca 2 componentes primarios: potencia activa y demanda o potencia aparente. La potencia activa es aquella que se mide comúnmente en vatios o kilovatios e indica la cantidad de energía que se consume en un segundo. Los vatios-hora indican el total de energía que se ha consumido en un periodo de tiempo como los registros mensuales de consumo de energía que realizan las compañías eléctricas. La demanda o potencia aparente consta de los requerimientos totales que el usuario demanda de la compañía eléctrica para que se proporcione la tensión y la corriente, independientemente de la eficiencia de los equipos del usuario o de si estos realizan un trabajo real o útil como el de calefacción o ventilación de edificios, etc.

En la actualidad existen diferentes unidades en la cuales se puede obtener la medición eléctrica tales como: voltio que es la unidad de potencial eléctrico, culombio que corresponde a la unidad de carga eléctrica dentro de un circuito, ohmio que es la unidad que indica la resistencia eléctrica, faradio que muestra la capacidad eléctrica, henrio que corresponde a la inductancia de un circuito, tesla que indica la densidad de flujo magnético e inductividad magnética, entre otros. Consecuentemente para medir las magnitudes se utilizan los instrumentos de medición eléctricos entre los dispositivos más utilizados están: el galvanómetro, amperímetro, voltímetro, óhmetro, multímetro, osciloscopio, analizador de espectro, entre otros.

Entre algunos medidores que ya han sido implementados y puestos a prueba se destaca la empresa Adler que ha desarrollado el medidor Fluke 1735 utilizado para registrar el consumo de energía de un refrigerador y determinar la eficiencia del equipo. Otro ejemplo de medidores eléctricos corresponde al sistema de medición, monitoreo y control de carga eléctrica para aplicaciones domesticas de la Universidad Politécnica Salesiana de Cuenca (Contreras, 2015). También se destaca el diseño e implementación de un sistema de monitoreo energético en la Escuela Politécnica del Ejercito de Latacunga (Ortiz y Rodríguez, pp 1-2).

1.1 Alcance

En el proyecto de titulación se estudian los distintos métodos de medición de consumo eléctrico, las tecnologías más utilizadas, así como también las aplicaciones en el ambiente residencial. Posteriormente se implementará un medidor inalámbrico de lectura del consumo eléctrico utilizando medidor de corriente comercial conjuntamente con todas las adaptaciones electrónicas necesarias para que el sistema lea los datos y se pueda comprobar las mediciones realizadas mediante acceso remoto; el dispositivo envía la información obtenida por conexión inalámbrica a un sistema central Raspberry que contiene los resultados de las mediciones eléctricas. El acceso al Raspberry se realiza mediante servicios web ya que se encuentra conectado mediante un cable Ethernet a un Access Point el que permite que el sistema central Raspberry obtenga una dirección IP de acceso; todos estos datos son monitoreados mediante acceso remoto a través de la dirección IP del Raspberry en una página web donde se despliegan todos los resultados de potencia que contenga el sistema central Raspberry en una interfaz amigable y sólida. Finalmente se realizará un análisis de pruebas y resultados con las mediciones obtenidas en un escenario real con electrodomésticos y circuitos de consumo en el entorno residencial.

1.2 Justificación

La motivación para la realización de este proyecto de titulación es que en el siglo XXI el incremento de hogares en donde existe un excesivo consumo de energía eléctrica cada vez es mayor, y el aumento de pago mes a mes que deben realizar los habitantes de un hogar involucran directamente al número de los dispositivos electrónicos conectados y a la cantidad de energía que están consumiendo. Con consumos de energía eléctrica considerablemente altos; los costos energéticos también son considerablemente elevados y crecientes, lo que se desea lograr es reducir el consumo eléctrico, mostrando al usuario evidencias físicas y en escenarios reales cuanto están gastando de energía eléctrica los dispositivos del hogar, con estos datos se puede generar en el usuario la concientización acerca del desperdicio de energía eléctrica por parte de ciertos dispositivos del hogar y

como beneficio el usuario pueda tomar decisiones adecuadas y acertadas para reducir el consumo eléctrico en su hogar y de esa manera se optimicen los recursos y se abaraten costos. Esto ayudara a los habitantes a controlar de mejor manera su consumo de energía y a ahorrar dinero.

1.3 Objetivos

1.3.1 General

Implementar un medidor digital inalámbrico de consumo eléctrico para dispositivos en el hogar a través de un sistema Raspberry y monitoreado en una página web mediante acceso remoto.

1.3.2 Específicos

- Analizar los diferentes métodos de medición de consumo eléctrico, tecnologías utilizadas y las aplicaciones en el entorno residencial.
- Implementar un prototipo de medición de consumo eléctrico sobre un sistema Raspberry, utilizando conexiones inalámbricas que desplieguen los resultados de las mediciones en un web browser mediante acceso remoto IP.
- Realizar un análisis de pruebas y resultados utilizando las mediciones obtenidas en un escenario real con electrodomésticos y circuitos de consumo en el hogar.

2. CAPÍTULO II. ANÁLISIS DE TECNOLOGÍAS Y MEDIDORES DE CONSUMO ELÉCTRICO.

En este capítulo se detallan todos los fundamentos teóricos en base a los que se desarrolla el proyecto.

2.1 Definición de Términos Eléctricos.

Se detallan algunos de los términos eléctricos que son los fundamentos teóricos de los medidores de consumo energético.

2.1.1 Corriente Eléctrica

“Es la velocidad de cambio de carga respecto al tiempo, medida en amperes (A)”
(Alexander y Sadiku, 2013, p. 6)

En términos matemáticos, se define a la corriente eléctrica “i” como la variación de una carga “q” en un intervalo de tiempo “t”.

$$i = \Delta \frac{dq}{dt} \quad (\text{Ecuación 1})$$

La corriente eléctrica se mide en amperios y la equivalencia de 1 amperio es de un culombio sobre segundo

$$1 \text{ ampere} = 1 \text{ coulomb/segundo} \quad (\text{Ecuación 2})$$

La carga “Q” que se transfiere en el tiempo inicial “t₀” y el tiempo “t” se obtiene de la integración de ambos parámetros en una ecuación.

$$Q = \Delta \int_{t_0}^t i. dt \quad (\text{Ecuación 3})$$

2.1.1.1 Corriente Directa (DC)

La corriente que permanece constante y no cambia con el tiempo se denomina corriente directa



Figura 1. Representación de la Corriente Directa.

Tomado de Alexander y Sadiku, 2013.

2.1.1.2 Corriente Alterna (AC)

Esta corriente es aquella que varía senoidalmente con el tiempo; comúnmente es la corriente que se emplea para accionar la mayoría de los artefactos eléctricos y electrónicos existentes en un entorno residencial.

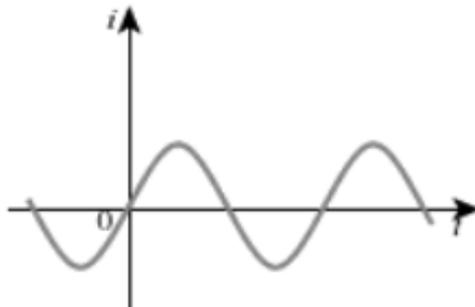


Figura 2. Representación de la Corriente Alterna.

Tomado de Alexander y Sadiku, 2013.

2.1.2 Potencia

La potencia se define como: “la razón de cambio de la energía con respecto al tiempo en términos del voltaje y de la corriente” (Grainger y Stevenson, 1996, p, 24).

La potencia instantánea “p” es una cantidad que varía con el tiempo y se define como la variación del trabajo “U” en un intervalo de tiempo. Y en términos de energía como la variación de la energía “w” en un determinado intervalo de tiempo.

$$p = \Delta \frac{dU}{dt} = \Delta \frac{dw}{dt} \quad (\text{Ecuación 4})$$

La unidad utilizada en el sistema internacional de unidades para la potencia es el watt representado con la letra mayúscula (W); donde un watt equivale a un joule por segundo.

$$1W = 1J/s \quad (\text{Ecuación 5})$$

Existe otra manera de representar la potencia eléctrica en el caso de motores en donde se utiliza la unidad denominada caballos de fuerza y se representa con la abreviatura (hp); en donde:

$$1hp = 745.7W \quad (\text{Ecuación 6})$$

2.1.2.1 Relación entre voltaje (V), corriente (I) y potencia (P).

La fórmula para obtener la potencia es el resultado de la multiplicación el voltaje por la corriente. En términos matemáticos:

$$P = V \cdot I \quad (\text{Ecuación 7})$$

Bajo estos términos existen 3 tipos de potencia que están definidas en circuitos eléctricos:

La potencia aparente:

Se representa con la letra “S” y es el resultado de la multiplicación del voltaje “V” por la corriente aparente “I_{ap}”. Su unidad de medida son los voltios amperios (VA) o kilovoltios amperios (kVA)

$$S = V \cdot I_{ap} \quad (\text{Ecuación 8})$$

La potencia real:

Se utiliza la letra “P” para su representación y se obtiene de la multiplicación entre el voltaje y la corriente. Si se tiene la corriente aparente se puede realizar una operación matemática por el coseno del ángulo que forma “ φ ”. Se mide en watts (W) o Kilowatts (kW).

$$P = V \cdot I_{ac} = V \cdot I_{ap} \cdot \cos(\varphi) \quad (\text{Ecuación 9})$$

La potencia reactiva:

Se representa con la letra “Q” y corresponde a la multiplicación del voltaje por la corriente reactiva. También es posible obtenerla mediante la multiplicación de la corriente aparente por el seno del ángulo que lo forma “ φ ”. Se utilizan los voltios-ampereos reactivos (Var) o kilovoltios-ampereos reactivos (KVA_r) como unidades de medida.

$$Q = V \cdot I_r = V \cdot I_{ap} \cdot \sin(\varphi) \quad (\text{Ecuación 10})$$

2.1.3 Voltaje

“En electricidad, la definición de diferencia de potencial es el voltaje” (Arce, J). Se trata de una magnitud física medida en voltios que impulsa el flujo de electrones a lo largo de un conducto. Se encarga de que la energía eléctrica se conduzca con mayor o menor potencia.

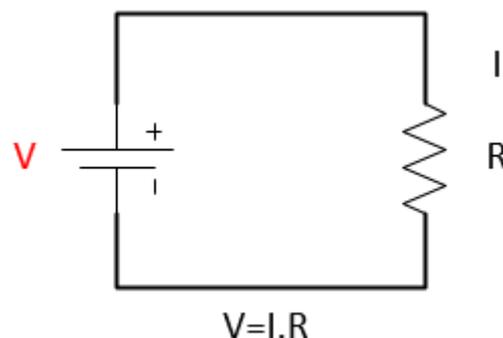


Figura 3. El Voltaje en un circuito eléctrico.

El voltaje en términos matemáticos y según la ley de Ohm está definido como la diferencia entre la intensidad de corriente eléctrica por la resistencia en un circuito eléctrico.

$$V = I \cdot R \quad (\text{Ecuación 11})$$

2.2 Tecnologías de transferencia de información

2.2.1 Tecnología ADSL

Es la tecnología que transmite información sobre accesos de cobre, plantea el uso más eficiente del medio de transmisión de datos; pretende que el par de cobre que se utiliza en los conectores RJ45 sea un medio para transmitir señales digitales a altas velocidades, con el objetivo de brindar a los usuarios, servicios que necesiten forzosamente de un consumo macro de ancho de banda como, por ejemplo: las teleconferencias, video en streaming, Internet, entre otros. ADSL se desarrolla con el objetivo de la creciente necesidad de ancho de banda que genera en el usuario respecto a nivel de acceso, y que hoy por hoy es una tecnología muy utilizada en distintas aplicaciones alrededor del mundo.

La transmisión es asimétrica por sus siglas en inglés "*Asymmetric Digital Subscriber Line*"; por ende, la descarga se transmite a mayor velocidad, que la carga de información.

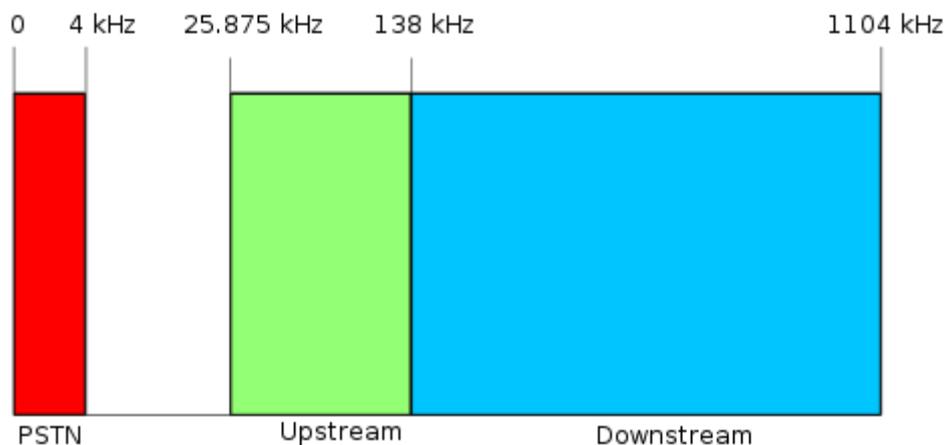


Figura 4. Frecuencias Usadas en ADSL

Tomado de Wikipedia, s.f.

El área roja de la figura representa la voz en una red de telefonía conmutada (PSTN), la sección verde conforma el upstream (subida de datos) y la de color celeste se emplea para el downstream (descarga de datos).

2.2.2 Tecnología por Radio Frecuencia

Radio Frequency IDentification, por sus siglas en inglés, es una de las tecnologías más utilizadas para la captura automática de datos e identificación electrónica de: artículos, componentes mecánicos y hasta personas, mediante el uso de instrumentales denominados “etiquetas”.

Un sistema RFID se conforma de cuatro elementos indispensables que son:

Etiquetas. _ Es la antena, que contiene un receptor y un transmisor de ondas de radio, con un modular que sirve para enviar señales de respuesta.

Lectores. _ Se encarga de la transmisión continua de pulsos de energía mediante ondas de radio, que son recibido por las etiquetas. Cuando la etiqueta detecta energía, devuelve una señal de respuesta hacia el lector, Esta información guardado en el chip de las etiquetas es por lo general un número de serie.

Antenas y Radios. _ Corresponde a la capa física de RFID y es usada para la transferencia de información entre los lectores y las etiquetas.

Hardware de procesamiento. _ Es el repositorio de datos que se usa para procesar la información que ha sido obtenida por los lectores.



Figura 5. Funcionamiento de la tecnología RFID.

2.2.3 Tecnología Wifi

2.2.3.1 Definición y Funcionamiento

Proviene de termino en inglés “Wireless Fidelity” con su traducción al español “Fidelidad Inalámbrica”. Es un servicio de conexión a la red de Internet de forma inalámbrica, satelital o mediante ondas radiales sin necesidad alguna de cables, en la actualidad es la más utilizada alrededor del mundo.

Para que un equipo o dispositivo pueda conectarse a la red mediante Wifi es necesario disponer de un modem o un ISP (Internet Service Provider) que cuenten con la tecnología necesaria para decodificar esta señal.

Opera dentro del rango de 2 a 5 GHz y su señal se fortalece mientras menor sea la distancia entre el dispositivo conectado y el modem. En entornos de mayor demanda de usuarios o dispositivos un solo servidor permite la conexión simultanea de equipos tales como: smartphones, tablets, laptops, entre otros. Para ampliar la cobertura de la señal es necesario el uso de repetidoras.

2.2.3.2 Protocolos de Wifi IEEE 802.11

Wifi posee varios protocolos en base a los cuales se define dentro de que rango operan y la velocidad máxima que puedan alcanzar entre otras características.

802.11ac wave2

Se lanzó al mercado en junio de 2016, con nuevas características tales como: Multiusuario MIMO y canales de 160 MHZ. (INTEL, 2017)

802.11ac wave1

Se implemento en enero de 2014, con velocidades de datos distintas. Posee 24 canales de infraestructura de información nacional sin licencia (UNNI) no superpuesto en banda de frecuencia de 5 GHz. (INTEL, 2017)

802.11n

Velocidades de datos con distintos tipos de modulación. Multiplexión de división de frecuencia ortogonal (OFDM) utilizando entrada múltiple y salida múltiple (MIMO) y unión de canales (CB) (INTEL, 2017)

802.11g

Posee multiplexación por división de frecuencia ortogonal (OFDM) con 52 canales de subportadora; compatibilidad retroactiva con 802.11b utilizando DSSS y CCK y tres canales no superpuestos en banda de frecuencia industrial, científica, médica (ISM) a 2,4 GHz (INTEL, 2017)

802.11a

Multiplexión de división de frecuencia ortogonal (OFDM) con 52 canales de subportadora y con 12 canales de infraestructura de información nacional sin licencia (UNII) no superpuestos en banda de frecuencia de 5 GHz (INTEL, 2017)

802.11b

Posee un espectro de difusión de secuencia directa de alta velocidad (HR-DSSS) y tres canales no superpuestos en banda de frecuencia industrial, científica, médica (ISM) a 2,4 GHz (INTEL, 2017)

802.11

Posee dos velocidades de datos sin procesar de 2 y 1 Mbps y con acceso múltiple de detección de portadora (CSMA-CA) (INTEL, 2017)

Tabla 1

Protocolos de Wi-Fi IEEE 802.11

Protocolo	Frecuencia	Ancho del canal	MIMO	Velocidad de datos máxima (en teoría)
802.11ac wave2	5 GHz	80, 80+80, 160 MHz	Usuario múltiple (MIMO-MU)	1,73 Gbps ¹
802.11ac wave1	5 GHz	80 MHz	Un solo usuario (SU-MIMO)	866,7 Mbps ¹
802.11n	2,4 o 5 GHz	20, 40 MHz	Un solo usuario (SU-MIMO)	450 Mbps ²
802.11g	2,4 GHz	20 MHz	No se aplica	54 Mbps
802.11a	5 GHz	20 MHz	No se aplica	54 Mbps
802.11b	2,4 GHz	20 MHz	No se aplica	11 Mbps
Tradicional 802.11	2,4 GHz	20 MHz	No se aplica	2 Mbps

Tomado de INTEL, 2017.

*Nota 1: ¹ 2 transmisiones espaciales con modulación 256-QAM.**Nota 2: ² 3 transmisiones espaciales con modulación 64-QAM.*

2.2.3.3 Ventajas y Desventajas de Wifi

Se exponen algunas ventajas y desventajas del protocolo wifi frente a otras tecnologías de transmisión de información.

Tabla 2

Ventajas y Desventajas de la tecnología Wifi

Ventajas	Desventajas
No necesita de cables, la conexión es inalámbrica.	Es de fácil acceso a los datos y hackeo de redes.
Soporta altas tasas de velocidad de transmisión de datos	Posee cobertura limitada y es necesario una repetidora para amplificar la señal de cobertura.
Se puede conectar desde cualquier lugar y varios dispositivos simultáneamente.	Posibilidad de fallos en la conexión

2.2.4 Protocolos de Comunicación

Corresponden a las tecnologías para envío y recepción de información más utilizada en el mundo, se destacan algunos protocolos como:

2.2.4.1 Protocolo Serial Peer to peer

En el protocolo peer to peer, la comunicación es full dúplex y se realiza solo entre transmisor y receptor. Todos los nodos son clientes y servidores al mismo tiempo, todo tienen la misma funcionalidad, este tipo de intercambio de ficheros supera al tráfico generado por la web.

2.2.4.2 Comunicación en Serie RS 485

Es una comunicación entre maestro y esclavo de hasta 1200 metros de longitud pueden alcanzar velocidades desde 2mbits por segundo mientras mayor sea la distancia es menor la velocidad de transmisión.

2.2.4.3 Protocolo M-BUS

Es el sistema desarrollado para la lectura remota y automática de contadores residenciales, como la medición del consumo de un hogar de agua o electricidad. Este protocolo posee requerimientos de sistema controlados a distancia, entre los cuales se toma como ejemplo a los contadores públicos de consumo. Algunas de las aplicaciones de Meter Bus es el control de temperatura, sistemas de alarma, iluminación entre otras.

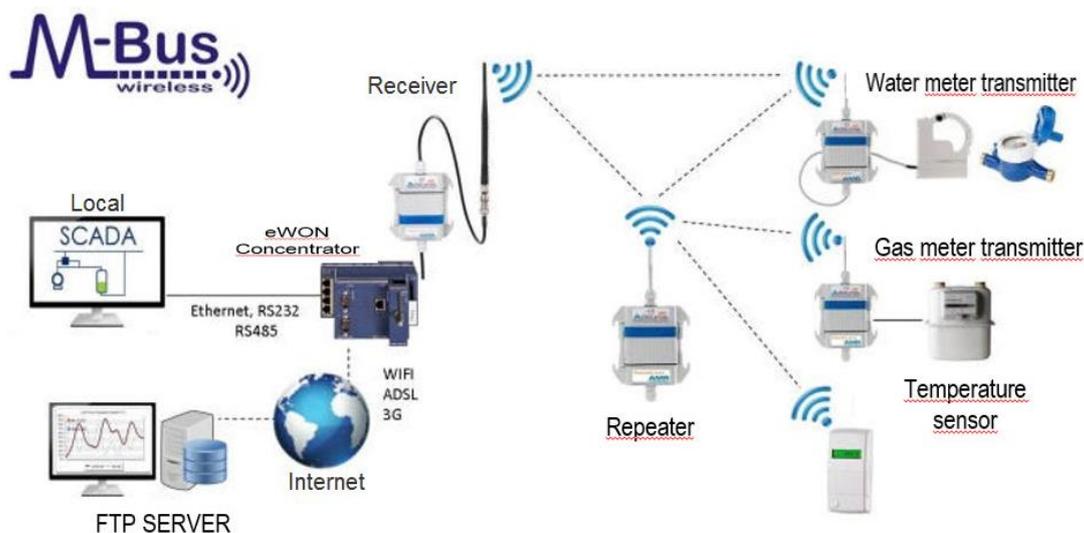


Figura 6. Monitorización Remota de Medidores.

Tomado de Airicom, s.f

Cada medidor envía sus datos inalámbricamente a una repetidora o directamente a un receptor dependiendo de la distancia que se encuentren; esta información a su vez es enviada a un concentrador central y por medio de cable Ethernet o Wifi todos estos datos son monitoreados por un servidor Local o FTP.

2.2.4.3.1 Relación con el modelo OSI

La tecnología M-Bus se basa en el modelo OSI con el propósito de conseguir un sistema abierto que pueda utilizar la mayoría de los protocolos. M-Bus no es una red y por ende no se define en las capas de transporte o sesión, provee funciones solamente en las capas restantes que corresponden a: física, enlace de datos, red y aplicación.

Tabla 3

Mapeo de la estructura M-Bus en las capas del modelo OSI

Capa	Funciones	Estándar
Aplicación	Estructura de datos, tipos de datos y acciones	EN 1434-3
Presentación	-	-
Sesión	-	-
Transporte	-	-
Red	Direccionamiento extendido	-
Enlace de Datos	Parámetros de transmisión, direccionamiento, integridad de los datos.	IEC 870
Física	Cable, representación a nivel de bit, extensiones de bus, topología, especificaciones eléctricas	M- Bus

Adaptado de López, 2011.

2.2.4.4 IP

Compone la capa 3 en el paquete de Internet. “Además del enrutamiento de internetwork, IP proporciona informe de errores y fragmentación y reensamblado de unidades de información llamadas datagramas para transmisión a través de redes con diferentes tamaños máximo de unidades de datos”. (Cisco, 2005)

Cada dispositivo que es asignado una dirección IP; es exclusivas en todo el mundo y lo conforma un numero de 32 bits asignados por el centro de información de red. El beneficio de cada dirección sea única es que permiten que las redes IP en cualquier parte del mundo se puedan comunicar entre sí y de ese modo se evitan los duplicado de IPs entre equipos.

Se divide en 2 partes. La primera corresponde a la dirección de red y la segunda parte pertenece a la dirección de host.

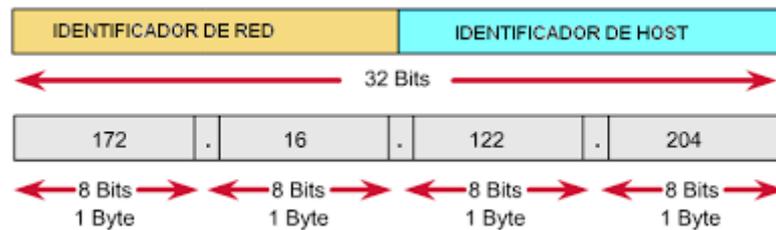


Figura 7. Partes que conforman una dirección IP.

Tomado de Cisco, 2005.

2.2.4.5 TCP/IP

2.2.4.5.1 TCP

“Es un protocolo de transporte orientado a la conexión que envía datos como un flujo no estructurado de bytes” (Cisco, 2005). TCP brinda la opción de retransmisión de datos hasta que la entrega de información sea exitosa. Adicionalmente este protocolo identifica mensajes que estén duplicados y los descarta. Puede enviar la información de entrega a capas superiores. Estas características definen a TCP como un protocolo de transmisión confiable de extremo a extremo.

Tabla 4

Comparación de los modelos OSI y TCP/IP

Modelo OSI	Modelo TCP/IP
Aplicación	
Presentación	Aplicación
Sesión	
Transporte	Transporte
Red	Internet
Enlace de Datos	
Física	Acceso a la Red

Adaptado de Emaze, s.f.

2.2.4.6 UDP

El protocolo de datagrama de usuario pertenece a la capa de transporte del modelo OSI sin conexión. Es un interfaz entre IP y procesos de capas superiores. No posee corrección de errores, es poco fiable; mediante el uso de UDP los programas de ordenadores en red pueden enviar mensajes cortos conocido como datagramas mediante el uso de sockets.

2.2.4.7 HTML

The Hypertext Markup Language por sus siglas en ingles. Es en lenguaje más utilizado a nivel mundial para el diseño y programación de páginas web. Se destaca principalmente porque con html es posible insertar hipervínculos, imágenes y además es compatible con otros lenguajes de programación como PHP para establecer conexiones con servidores.

2.2.4.7.1 Estructura de HTML

HTML inicialmente se compone de etiquetas limitadas por corchetes angulares (<,>) que puede incluir un tipo de referencia hacia un programa denominado script. Cada elemento de HTML tiene 2 propiedades básicas: atributo y contenido. El orden de la plantilla de HTML es el siguiente:

“<html>” Se trata del documento inicial, sin esta etiqueta no existe código html, esta línea le indica al navegador web que debe interpretar información de carácter html.

“<script>” Esta etiqueta es opcional y permite insertar un script en la web.

“<head>” Esta parte define la cabecera del documento HTML, contiene información sobre el documento que no es visible directamente al usuario como el título que tiene la ventana de navegador.

“<body>” Esta es la sección donde se incluye todo el cuerpo o el contenido de la página web, y se puede insertar colores, márgenes, tablas, estilos, imágenes, salto de líneas y todo lo que sea permitido dentro del lenguaje html.



```

1 <!DOCTYPE html>
2 <html lang="es-ES" xmlns:og="http://opengraphprotocol.org/schema/"
  xmlns:fb="http://www.facebook.com/2008/fbml" prefix="og: http://ogp.me/ns#">
3 <head>
4 <meta charset="UTF-8" />
5
6 <!--
7 /* @license
8  * font icons by icomoon.com
9  *
10 * © 2013 Udla Quito
11 */
12 -->
13
14 <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0,
  user-scalable=no," />
15 <title>Universidad de Las Américas - UDLA Quito</title>
16
17 <meta name="apple-mobile-web-app-capable" content="yes" />

```

Figura 8. Código Fuente de la página web de la Universidad de las Américas. Tomado de Udla, 2018.

2.2.4.8 MQTT

Se define como “Message Queue Telemetry Transport” por sus siglas en inglés. Es un protocolo pensado por la empresa multinacional de tecnología IBM orientado a la conectividad (M2M) Machine to Machine. Se enfoca al envío de datos en aplicaciones donde la necesidad de ancho de banda es mínimo, por su bajo consumo es idóneo por que utiliza pocos recursos para su funcionamiento. El protocolo se aplica en la comunicación de sensores y por su exiguo consumo principalmente en el Internet de las Cosas.

2.3 Medidores de consumo eléctrico tradicionales

Son aquellos dispositivos destinados a medir el consumo de energía eléctrica de los artefactos de un: hogar, fabrica, edificio, entre otros.

2.3.1 El Contador

Se trata de un dispositivo diseñado para medir la energía que se consume en un suministro (Cornejo, 2014, p. 93). Posee bobinas de intensidad y tensión que provocan un flujo magnético como respuesta al paso de corriente que alimenta a una carga. Esas corrientes generan un flujo magnético en el disco interno del contador lo que origina su giro. Las vueltas que genera el disco son directamente proporcionales al campo magnético que se induce, y también depende de la tensión e intensidad que consume una carga.



Figura 9. El Contador.

Tomado de Yaencontre, s.f.

2.3.2 Contador electrónico

Se trata de un contador que posee circuitos electrónicos que emiten una señal proporcional a la energía consumida. Esta señal es almacenada en una memoria electrónica. No posee partes móviles y es más sencillo de ubicar. (Cornejo, 2014, p. 94)



Figura 10. El contador electrónico.

Tomado de Wikipedia, s.f.

2.3.3 Contador inteligente

Se denomina contador inteligente al contador electrónico que, además de ser capaz de discriminar la energía en varios periodos horarios, también es capaz de controlar la potencia demandada y están integrados en el sistema de teledeteción y telegestión de cada distribuidor. (Cornejo, 2014, p. 94)



Figura 11. El contador Inteligente.

Tomado de INESEM, s.f.

2.3.4 Contador de prepago

Este contador necesita de monedas o una tarjeta para funcionar. Está basado en un sistema por el cual va descontando del crédito el importe de la energía que se consume. Cuando ese crédito haya llegado a 0, la instalación se habrá desconectado. Comúnmente posee un visor donde aparece el crédito disponible que es utilizado para controlar el gasto. (Cornejo, 2014, p. 94).



Figura 12. El contador de prepago.

Tomado de Arelia, 2017.

2.3.5 Interruptor horario

Es un dispositivo programable en el cual se puede contralar varias cargas eléctricas, activando o desactivando el medidor, además permite medir el consumo en función del momento en el cual se produce. Estos pueden ser ciclos diarios, semanales, entre otros, dependiendo de aquellos ciclos el interruptor debe poseer una programación diferente en base al día del año que se utilice. (Cornejo, 2014, p. 95)



Figura 13. El Interruptor Horario.

Tomado de Panasonic, s.f.

2.3.6 Pinza Amperimétrica

Se trata de un comprobador eléctrico que contiene un voltímetro conjuntamente con un medidor de corriente similar a una pinza. De la misma forma que el multímetro, esta pinza permite medir elementos digitales. Además, tienen incorporados funciones de medida con más exactitud y en algunas de ellas funciones de medida específicos. Estos sujetadores amperimétricos cuentan con las funciones elementales de un multímetro digital, con la diferencia que tiene incorporado un transformador de corriente en su diseño.



Figura 14. La pinza amperimétrica

Tomado de Wikipedia, s.f.

2.4 Medidores de consumo eléctrico smart

Estos medidores son aquellos que utilizan tecnologías inalámbricas y comúnmente son utilizados en el desarrollo de proyectos inmóticos y servicios inteligentes, usualmente son empleados en instalaciones domóticas y casas futuristas.

2.4.1 MeterBUS-4C

Es un dispositivo para medición de consumo eléctrico con 4 canales físicos que soporta 25 Amperios de carga eléctrica por canales con la posibilidad de visualización de los canales desde interfaces Busing compatibles con la plataforma Ingenium. Permite mostrar de manera gráfica el consumo acumulado semanalmente. (Ingenium, 2018)

Es posible la programación en escenarios desde el sistema de desarrollo Busing: SIDE O SIDE KITS.



Figura 15. MeterBus de 4 Canales.

Tomado de Ingenium, 2018.

2.4.1.1 Datos Técnicos

Tabla 5

Datos Técnicos del Meter Bus de 4 Canales.

Atributo	Descripción
Rango de Medida	0 – 131000 KW/h
Tiempo de Refresco	5 seg
Alimentación	12 Vdc (BUS)
Consumo	75 Ma (BUS)
Montaje en carril DIN	6 módulos

Adaptado de Ingenium, 2018.

2.4.2 Contador de Pulsos Eléctricos KAMSTRUP 382

2.4.2.1 Definición y Aplicación

Se trata de un contador de energía eléctrica de conexión directa. No posee piezas móviles por lo que es netamente electrónico. Es inmune a campos electromagnéticos y corrientes directas. Posee una pantalla de lectura que se puede desplazar con botones. Los registros son configurables y visibles.

Los datos leídos en pantalla se encuentran disponibles mediante la lectura a distancia o el puerto óptico. El dispositivo permite una conexión con una red de AMR (Automatic Meter Reading).

Se puede realizar configuraciones de fábrica para medir energía exportada así también como energía importada por fase. Está compuesto por 3 circuitos independiente y galvánicamente separados, puede almacenar hasta 36 meses de históricos. Una función muy importante es el modo de verificación en el cual se mejora la resolución de las indicaciones de energía, como consecuencia se reduce la duración de escenarios de pruebas y verificaciones.



Figura 16. El Kamstrup 382.

Tomado de Kamstrup, s.f

2.4.2.2 Datos Técnicos

Tabla 6

Datos técnicos del Kamstrup 382.

Atributo	Descripción
Intensidad	Monofásica medida por shunt
Tensión	Monofásica medida por divisor de tensión
Tensión Nominal	3x230V ± 10% (Sólo Aron) 1x230V ± 10% 2x230/400V ± 10% 3x230/400V ± 10%
Frecuencia Nominal	50Hz ± 2%
Desviación de fase	Sin límite, no para los contadores Aron
Temperatura de operación	-40°C - +70°C
Temperatura de almacenamiento	-40°C - +70°C

Adaptado de Kamstrup, s.f.

2.4.3 Conmutador inalámbrico wifi Sonoff POW

Se trata de un conmutador eléctrico capaz de medir la potencia consumida por un dispositivo. Lo innovador de este elemento es que la captura de información de los datos se puede visualizar en una aplicación en tiempo real siempre y cuando el dispositivo medidor se conecte a una red wifi. Este tipo de medidores son comúnmente utilizados para edificios inteligentes y servicios domóticos.



Figura 17. Medidor de Potencia Wifi (Sonoff POW).

Tomado de Itead, s.f.

2.4.3.1 Características Virtuales

Tabla 7

Características Virtuales del Sonoff POW.

Serie	Descripción
1	Control de Uso de Energía, 1% de Precisión.
2	Verificación del consumo diario
3	Verificación del consumo mensual de energía
4	Verificación de la potencia en tiempo real
5	Configuración SSID y configuración de contraseña través de la aplicación "EWelink"
6	Control remoto
7	Configuración de conteo regresivo, simple y repetido.
8	Instrucción de Cableado

Adaptado de Itead, s.f.

2.4.3.2 Características Técnicas

Tabla 8

Características Técnicas del Sonoff POW

Atributo	Descripción
Modelo	Sonoff POW
Rango de Voltaje	90 a 250V AC
Potencia Máxima	3500 watts
Corriente Máxima	16 amperios
Seguridad	WPA/WPA2
Color	Blanco
Dimensiones	114*51*32 (mm)
Temperatura Recomendada	0°C-40°C
Frecuencia Wireless	80 – 160 MHz
Estándar Wireless	IEEE 802.11 b/g/n

Adaptado de OpenHacks, s.f.

2.4.3.3 Diagrama de Conexión

El medidor de corriente eléctrica Wifi Sonoff Pow está basado en una placa del integrado ESP8266 el cual permite inalámbricamente conectarse a la red y obtener los datos reales de potencia eléctrica consumida de uno a varios dispositivos que se encuentren conectados, estos datos son (enviados y se pueden obtener) reflejados en una aplicación previamente desarrollada denominada “Ewelink”, en la cual se pueden visualizar los valores de potencia, además de brindar varios servicios tales como: establecer una alarma para el dispositivo conectado, visualizar la consumición diaria y un parte promedio de cuantos KWh se ha consumido en cada mes, también cuenta con la posibilidad de compartir el control del dispositivo hacia otro usuario que tenga la aplicación, temporizador y cuenta atrás programada ya sea para apagar o encender el dispositivo.



Figura 18. Funcionamiento del medidor Sonoff Pow.

2.4.3.4 Ventajas del medidor Sonoff Pow vs otros medidores smart

Tabla 9

Ventajas del Sonoff POW frente a otros medidores Smart

Sonoff POW	Otros medidores SMART
De carácter económico.	Demandan altos costos.
No requiere de licencias.	Se requiere de licencias para utilizar software.
Posee el integrado wifi ESP8266 para realizar las conexiones inalámbricas.	No posee el integrado ESP8266 para conexiones inalámbricas.
Información en la web para desarrollar aplicaciones.	Poca o mediana información en la web para desarrollar aplicaciones.
Es un switch y medidor al mismo tiempo	Únicamente sirve como medidor.
La principal es que posee 4 pines de programación en los cuales se puede añadir o sobrescribir código sobre el firmware de fábrica.	No se puede insertar código

El Sonoff Pow es el dispositivo medidor de corriente eléctrica predilecto para el desarrollo de este proyecto porque a diferencia de otros medidores, posee 4 ranuras para soldar pines para programación en su placa y mediante una conexión serial USB a TTL permite grabar y sobrescribir otro firmware sobre el firmware

original de fabrica del dispositivo; esto lo convierte en el elemento ideal para aplicaciones smart y proyectos domóticos.

2.4.3.5 ESP 8266

Es un microcontrolador de bajo costo desarrollado por la empresa Espressif Systems, internamente posee un módulo wifi que permita realizar conexiones TCP/IP. Es ideal para aplicaciones de Internet of Things. El objetivo principal es el de dar acceso a cualquier microcontrolador a la red. Se puede configurar como cliente, servidor y Access Point.



Figura 19. ESP 8266.

Tomado de Wikipedia, s.f.

2.4.3.5.1 Modelos

Este integrado está presente en varios módulos, entre los modelos figuran los siguientes.



Figura 20. Modelos del ESP.

Tomado de Aprendiendo Arduino, s.f.

Características

- Tecnología Wifi 802.11 b / g / n
- Tiene integrado un MCU in de baja potencia y 32 bits
- Pila de rotocolo TCP / IP integrado
- Admite la diversidad de antenas
- WiFi 2.4 GHz, compatible con WPA / WPA2
- Soporta modos de operación STA / AP / STA + AP
- Admite la función Smart Link para dispositivos Android e iOS
- SDIO 2.0, (H) SPI, UART, I2C, I2S, Control remoto IR, PWM, GPIO
- STBC, 1 × 1 MIMO, 2 × 1 MIMO
- Consumo de energía en espera de <1.0mW (DTIM3) • Potencia de salida de +20 dBm en modo 802.11b
- Rango de temperatura de funcionamiento -40C ~ 125C
- Certificación FCC, CE, TELEC, WiFi Alliance y SRRC
- ESP8266 no posee una memoria ROM y usa una ROM externa SPI y soporta hasta 16MB. (AprendiendoArduino, s.f)

Ventajas

- Es económico.
- Posee un entorno basado en Arduino.
- En la web existente librerías útiles ya programadas para Arduino.
- Permite la integración del código Arduino con el hardware o la placa del Sonoff POW.

2.4.3.5.2 Usos y Aplicaciones

Tabla 10

Aplicaciones del ESP 8266

Aplicación	Descripción
1	Automatización del Hogar
2	Domótica

3	Aparatos eléctricos
4	Cámaras IP
5	Redes de Sensores
6	Wearables
7	Internet of Things

Adaptado de AprendiendoArduino, s.f

2.4.3.5.3 Conexión Serial a TTL

La placa del Sonoff POW viene integrada con 4 pines de conexión serial TTL que permite sobrescribir firmware sobre el código original de fábrica. Inicialmente estos pines no están soldados por lo que si se requiere insertar el código de programación se deben soldar los 4 pines de conexión previamente.

Tabla 11

Descripción de los pines de Conexión.

Pin	Descripción
1	Alimentación de 3,3 voltios
2	TX
3	RX
4	Conexión a Tierra

Se comunica el RX del programador con el TX de la placa del Sonoff Pow y viceversa. Como primer paso se aplasta el pulsador y luego se alimenta los 3,3 voltios al pin correspondiente a la alimentación para que quede en modo de programación; posteriormente se compila y se envía a grabar el código del programa. Y no se dejar de presionar el pulsador hasta que la grabación finalice.



Figura 21. Convertidor de USB a TTL.

Tomado de INVEN, s.f.

2.5 Lectura automática de medidores (AMR)

2.5.1 Origen

La lectura automática de medidores se probó por primera vez en experimentos realizados por la compañía de origen estadounidense AT&T en conjunto con la colaboración de la compañía Westinghouse Electric. Con ensayos se prueba exitosos la empresa AT&T decidió brindar servicio de AMR que se basaban en sistemas de telefonía celular, adicionalmente que el precio propuesto por estos servicios era económicamente factible.

1985 se denomina como la era moderna para AMR en donde se implementaron varios proyectos de impacto a gran escala. Las compañías Equitable Gas Go LLC y Hackensack Water Co fueron los pioneros al involucrarse a gran escala en la implementación de AMR en medidores de agua y gas.

2.5.2 Principio de Funcionamiento

Esta tecnología recopila de manera automática los datos de consumo, diagnóstico y el estado de dispositivos medidores de: agua, gas y electricidad y toda esta información es transferida a una base de datos central que se utiliza para: análisis, solución de inconvenientes, facturación, entre otras. AMR envuelve tecnologías móviles, portátiles y de red que son basadas en plataforma

de telefonía inalámbricas y cableadas, radiofrecuencia o transmisión de líneas eléctricas.

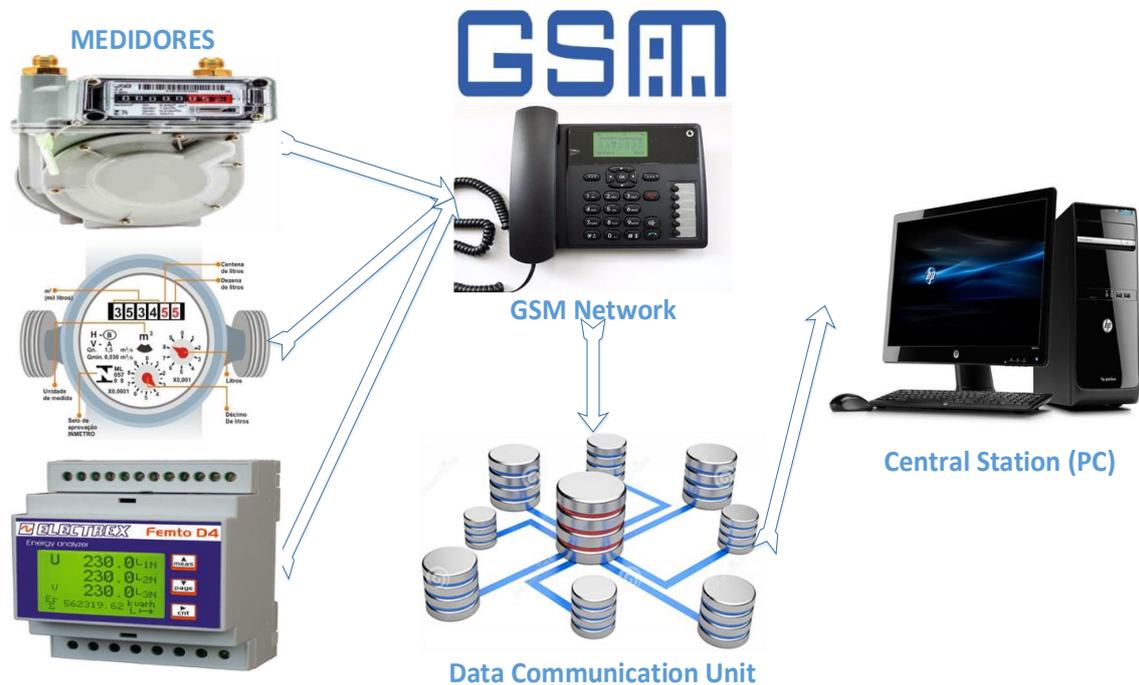


Figura 22. Bosquejo del Funcionamiento de AMR.

“El sistema ARM comienza en el medidor. Algunos medios para traducir la lectura de los diales rotativos del medidor o diales del medidor del estilo del ciclómetro, en forma digital son necesarios para enviar la medición digital de los datos del sitio del cliente a un punto central.” (Tamarakin, 1992)

2.5.3 Componentes Principales de AMR

2.5.3.1 Modulo de Interfaz del medidor

Se compone de la fuente de alimentación, sensores propios del medidor, electrónica de control y una interfaz de comunicación que da la posibilidad que la información se transmita desde el dispositivo ARM remoto a cualquier ubicación central. La interfaz de comunicación es bidireccional y permite que las señales de oficina sean también recibidas remotamente. Es necesario que cualquier medidor, de: agua, gas o eléctrico tenga incorporada esta unidad de interfaz para que pueda ser leída remotamente.

2.5.3.2 Sistema de Comunicaciones

Se utilizan para la transmisión o telemetría de datos y control de las señales de envío que se generan entre las unidades de interfaz del medidor y la oficina central. Estos sistemas generalmente son: radiofrecuencia (RF), la línea de transmisión portadora (plc), o televisión por cable.

2.5.3.3 Sistema de Equipamiento de la Oficina Central

Entre estos sistemas se destacan los ordenadores, controladores, módems, datos concentradores, enlaces de carga de host, receptores, entre otros.



Figura 23. Lectura Automática de Medidores.

Tomado de Toronto SUN, s.f.

2.5.4 Métodos de Comunicación

En los sistemas AMR existen 4 métodos de comunicación que se usan para la lectura del medidor.

2.5.4.1 Líneas Telefónicas

La mayor parte de la población poseen el servicio telefónico lo que lo hace un método económico y rentable. El sistema telefónico brinda una infraestructura de comunicación ideal para AMR por la facilidad de operación, alta inmunidad al ruido, fiabilidad, calidad de los datos, bajo costo entre otros parámetros.

2.5.4.2 Portadora de línea Eléctrica

Este tipo de comunicación demanda un costo considerablemente alto debido al estrecho ancho de banda que ocupan los datos al realizar la transmisión y recepción. Se ejecuta en las líneas que entregan electricidad. Implica la inyección de una portadora de CA de alta frecuencia en la línea eléctrica y una modulación de esta portadora con los datos que son propios del medidor remoto o la estación central.

2.5.4.3 Radio Frecuencia

Estos sistemas usan transceptores o transmisores de radio frecuencia de baja potencia situados en el controlador. Aprovechan porciones del espectro de radio frecuencia y la potencia radia efectiva del transmisor, así como las distancias que puede atravesar y varía en función de los términos de potencia y frecuencia de transmisores y receptores remotos.

2.5.4.4 Comunicación por televisión por cable

Este tipo de comunicación usa líneas de televisión por cable ya posicionadas para transmitir la información. Es de carácter costoso y complejo. Demanda mucho dinero la actualización de estos sistemas con amplificadores, utilizan gran ancho de banda. Existen algunas empresas eléctricas que se adaptan a las necesidades de AMR. Como meta a futuro se plantea que el cable incluya una transmisión de señal digital dirección con un ancho de banda más extenso con fibra óptica, que como finalidad lo convierta en un medio de comunicación ideal.

2.5.5 Tablas Comparativas

Tabla 12

Comparación de los métodos de medición.

Medidor Tradicional	Medidor Inteligente
Contador Mecánico	Contador Estático
Desgaste con el tiempo	Precisión continua
Lectura manual	Capacidad de lectura remota
Datos Limitados	Visión General de todos los datos
No existen alarmas	Alarmas Inteligentes

Tomado de Kamstrup, s.f.

Por sus características que son más adaptables al usuario, los medidores inteligentes brindan mayores capacidades frente a medidores tradicionales y al ser inteligentes ofrecen más servicios

Tabla 13

Comparación de los medidores tradicionales y Smart

	Medidor Tradicional	Meter Bus 4 C	KAMSTRUP 382	Sonoff POW
Inalámbrico				X
Smart		X	X	X
Monitoreo en Tiempo Real		X	X	X
Compatibilidad con cualquier sistema operativo			X	X
Conmutador Wifi (On/Off)				X
Control remoto desde aplicación			X	X
Reporte de consumos de potencia	x		X	X
4 pines de programación para sobrescribir firmware				X

Por todas las características y servicios que ofrece el Smart wifi Sonoff POW se define como el medidor de consumo eléctrico ideal para el diseño e implementación de este proyecto de titulación.

2.6 Entidades y Datos Estadísticos del consumo eléctrico en Ecuador.

Las entidades reguladoras son creadas por el Estado para realizar el control de las industrias o sectores que ofrecen los servicios invisibles para la ciudadanía, tales como: energía, obras públicas, comunicaciones, etc.

2.6.1 Arconel

Es la institución encargada de regular y controlar las actividades relacionadas a la energía eléctrica, precautelando el bienestar de los ciudadanos y el desarrollo sustentable del país. La ARCONEL tiene como objetivo: “Ser un modelo como organismo de regulación y control del servicio eléctrico de energía eléctrica a nivel latinoamericano, por nuestra solvencia técnica y jurídica, contribuyendo al buen vivir de los ecuatorianos y la sostenibilidad global” (ARCONEL, 2017).

2.6.2 Consumo per cápita

Es un término proviene del vocablo “de cada cabeza” y es muy utilizado en el campo de las estadísticas, que pueden ser de carácter: económica, social, etc. También se aplica para referencias a tipos de distribución o divisiones entre grupos o comunidades de personas. En este caso representa la relación que hay entre el consumo de energía y la población desde el año 1999 hasta el 2016 e término de (KWH/hab). El número de habitantes es obtenido mediante los censos realizado por el INEC.

Tabla 14

Consumo per Capita Multianual Ecuador.

Consumo Per Cápita Multianual (KWH/hab)			
Año	Consumo de Energía (GWh)	Población * (miles)	Consumo Per Cápita
1999	7730.69	12121	637.79
2000	7885.08	12990	607.01

2001	7965.60	12480	638.27
2002	8097.05	12661	639.53
2003	8362.31	12843	651.12
2004	8693.34	13027	667.33
2005	9044.38	13771	656.79
2006	9549.78	14021	681.12
2007	10063.95	14271	705.21
2008	11146.68	14521	767.63
2009	12740.80	14771	862.55
2010	13769.73	15012	917.23
2011	14931.12	15266	978.04
2012	15847.99	15521	1021.07
2013	16742.94	15775	1061.38
2014	17958.30	16027	1120.47
2015	18926.51	16279	1162.64
2016	18897.43	16529	1143,31

Tomado de Datos INEC, s.f

2.7 Norma ISO 50001: Gestión de Energía.

Esta norma se preparó por el comité de Proyectos ISO/PC 242 como: Gestión de la Energía.

2.7.1 Propósito

Esta norma Internacional tiene como objetivo brindar a las organizaciones el establecimiento de los sistema y procesos que son fundamentales para el desempeño energético, consumo de la energía, eficiencia entra otros parámetros. Se destina a reducir de emisión de gases de efecto invernadero, así como también otras problemáticas ambientales. La norma se aplica a toda organización sin tomar en cuenta su tipo, tamaño, condiciones sociales, geográficas o culturales.

2.7.2 Modelo de Sistema de Gestión de la Energía



Figura 24. Modelo del Sistema de Gestión de Energía para ISO 50001

Adaptado de ISO, 2017.

El proceso de mejora continua de la norma ISO 50001 es un proceso recurrente que tiene como finalidad la mejora en el desempeño energético y en el sistema de gestión de la energía. En el proceso se establece objetivos y se pretende encontrar oportunidades de mejora. Cuando se detecta un incumplimiento de alguna cláusula de la norma se denomina como “no conformidad” en la cual se aplica una acción correctiva para brindar una solución, esta acción se toma para prevenir que algún incidente vuelve a producirse.

2.8 Highcharts

Es un software diseñado para la creación de gráficas y diagramas; escrita en el lenguaje de JavaScript, brinda un método dinámico y sencillo al usuario para la creación, inserción y modificación de graficas en un sitio web o aplicación web, es absolutamente compatible con todos los navegadores existentes hasta la actualidad, lo más importante es que es de código abierto y el mismo se puede modificar y es posible visualizar los cambios en tiempo real. Es una plataforma adaptable y flexible por que las características de las plantillas y los demos existentes en su software y se pueden personalizar según las necesidades del usuario ya que existen innumerables opciones para la creación de diagramas, cada uno de ellos es único porque tiene algo que lo distingue de los otros. Para acceder a estos servicios en la web se escribe la URL “<https://www.highcharts.com/>”. A nivel empresarial se debe adquirir licencias. Algunos de los JavaScript disponibles en la página oficial del sitio web son:

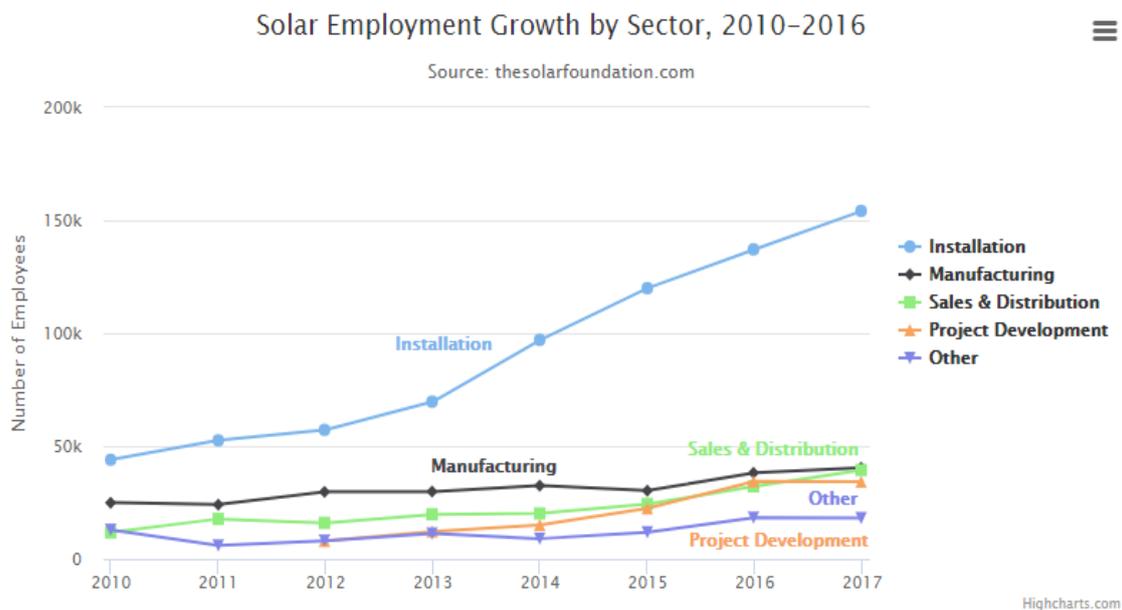


Figura 25. Diagrama de Línea Básica.

Tomado de Highcharts, 2018.

El diagrama de demostración tomado de la página oficial muestra el crecimiento del empleo solar por sectores del año 2010 al 2017; en el eje “x” se aprecia los años, mientras que en el eje “y” se distingue el número de empleados, cada línea con su respectivo color representa una categoría.

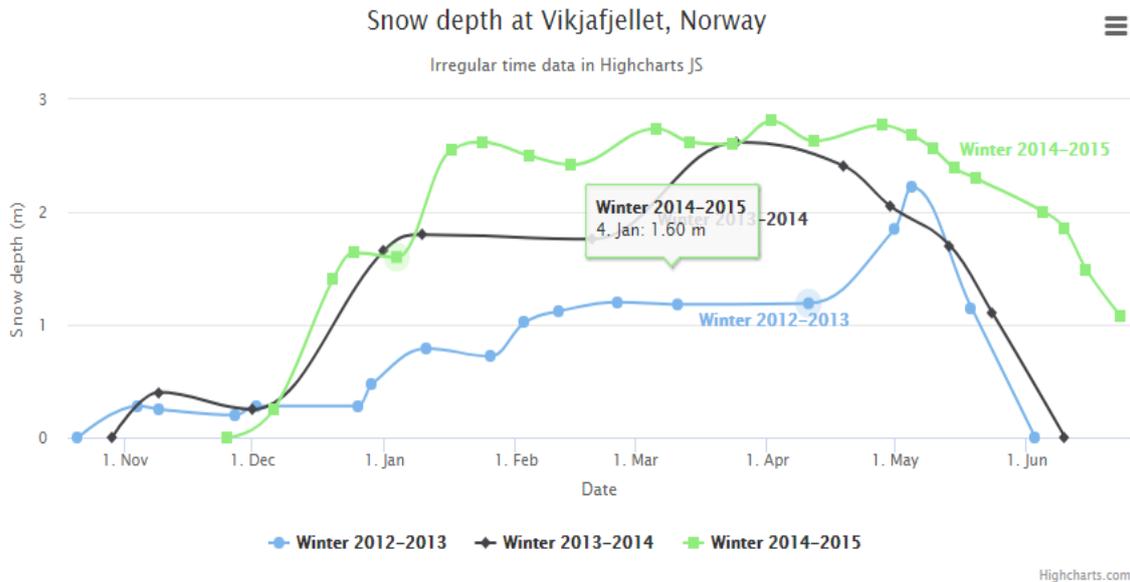


Figura 26. Diagrama de Tiempo con líneas de intervalo.

Tomado de Highcharts, 2018.

Este diagrama indica valores de profundidad en cada intervalo de tiempo ideal para realizar comparaciones entre mediciones de sensores.

2.9 Raspberry Pi

2.9.1 Definición e Historia

Se trata de una placa similar a una mini computadora, pero portátil con similares características y su sistema operativo estaba basado en Linux, específicamente su distribución Raspbian para ordenadores, permite desarrollar aplicaciones de diversos usos tales como: escaneo en 3D, media center, estaciones meteorológicas, procesamiento de imágenes, captura de datos e información, etc.

En el año 2011 y frente a la carencia de estudios de la ciencia de la computación por parte de estudiantes en el ámbito informático; basada en la arquitectura de un microprocesador ARM se da origen al primer prototipo "Raspberry Pi" creado por la fundación que lleva su mismo nombre. La placa se creó con el objetivo de "Promover la educación de los adultos y los niños, particularmente en el campo de las computadoras, ciencias de la computación y temas relacionados". []

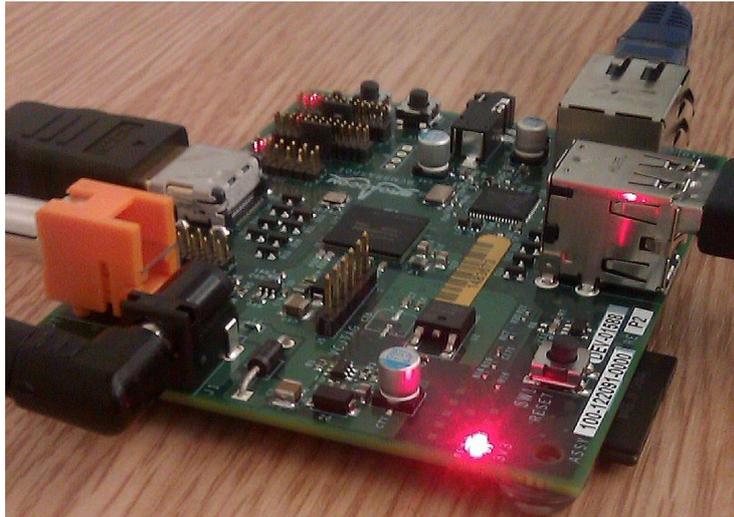


Figura 27. Placa de pruebas (Alfa) Raspberry Pi.

Tomado de Wikipedia, s.f.

2.9.2 Raspberry Pi 3 Model B

Es un minicomputador de placa simple creado por la organización Raspberry Pi con tecnología Wifi 802.11n, Bluetooth 4.1; está conformado un procesador ARMv8 de 64 bits con cuatro núcleos que trabaja a una velocidad de 1.2 GHz, 1 GB de memoria RAM; conjuntamente con una entrada HDMI para configuración inicial, 4 puertos USB, un puerto para cable ethernet, un puerto para dispositivos de audio, un puerto para la alimentación y el lector de tarjetas USB donde se realiza la instalación del sistema operativo Raspbian una versión adaptada de Debian.



Figura 28. Raspberry Pi 3 Model B V1.2.

Tomado de Raspberry, s.f.

2.9.3 Componentes de la placa Raspberry

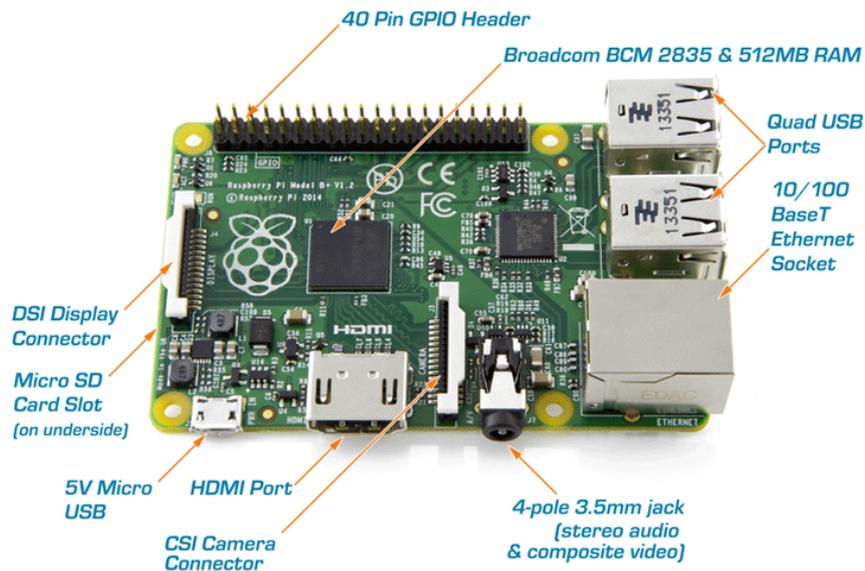


Figura 29. Bosquejo de la placa Raspberry Pi.

Tomado de Cumulations Technologies, s.f.

Procesador Broadcom BCM 2835

Es el cerebro de la Raspberry Pi encargado de ejecutar los programas desde el sistema operativo hasta las aplicaciones de usuario, ejecuta lenguajes de bajo nivel, controla el flujo de información, el acceso a memoria, etc.

Puertos Quad USB

Son 4 puertos USB designados para conectar los periféricos a la Raspberry tales como: mouse, teclado, flash memory, entre otras cosas.

Ethernet Socket

Es la ranura para el cable Ethernet que permite la conexión alámbrica desde un Router o un Access Point hacia la Raspberry para la asignación de una IP dinámicamente.

CSI Camera

Este puerto es específicamente para una cámara Pi.

4 pole 3.5mm jack

Ranura utilizada para la conexión de audio se puede conectar dispositivos tales como audífonos.

HDMI

Uno de los más importantes, es la salida de video conectado hacia un monitor HDMI o cualquier elemento que posea una entrada HDMI de la interfaz de la Raspberry Pi necesaria para la configuración inicial.

5V micro USB

La fuente de alimentación de la Raspberry la cual se enchufa generalmente con un cargador similar al de un teléfono que posea de salida 5V y 1,5A.

Ranura para Micro SD

La ranura en donde se inserta la tarjeta micro SD que contiene el sistema operativo Raspbian.

DSI Display Connector

Es un puerto de 15 pines que se utilizan para la conexión con pantallas LCD, esto permite la visualización en pantalla táctiles.

GPIO Header

En total es un conjunto de 40 pines que se utilizan para varias aplicaciones, como: ser configurados para entradas o salidas, pines de voltaje, tierra, conexión serial (TX, RX), etc.

2.9.4 Componentes Externos

Micro SD

La micro SD es en donde se almacena el sistema operativo para el arranque de la Raspberry, la más utilizada es la micro SD de 16Gb clase 10, de modo que al ser clase 10 garantiza una velocidad optima a la cual va a trabajar la Raspberry

Pi. El sistema operativo Raspbian ocupa aproximadamente 4,6 Gigas por lo que el requerimiento mínimo es de una micro SD de 8Gb clase 10 en adelante.

Fuente de Alimentación

La Raspberry Pi se alimenta con un cargador similar al de un celular con una salida de 5V 1.5A.

Cable HDMI

Permite la conexión con un monitor o televisor HDMI

Teclado USB

Utilizado para la configuración inicial.

Mouse USB

Utilizado para la configuración inicial



Figura 30. Raspberry Pi con periféricos conectados

3. CAPÍTULO III. DESARROLLO E IMPLEMENTACIÓN DEL PROTOTIPO

En este capítulo se describe el alcance que se puede llegar a desarrollar con el prototipo medidor de potencia wifi inalámbrico y el diseño con su implementación que constituye la base para poder integrar un sistema de medidores de potencia wifi basados en el dispositivo Sonoff POW y esta integración podrá ser desarrollada por posteriores estudiantes que incursionen en el campo del control IOT teniendo como base esta información.

3.1 Planteamiento del Sistema Global

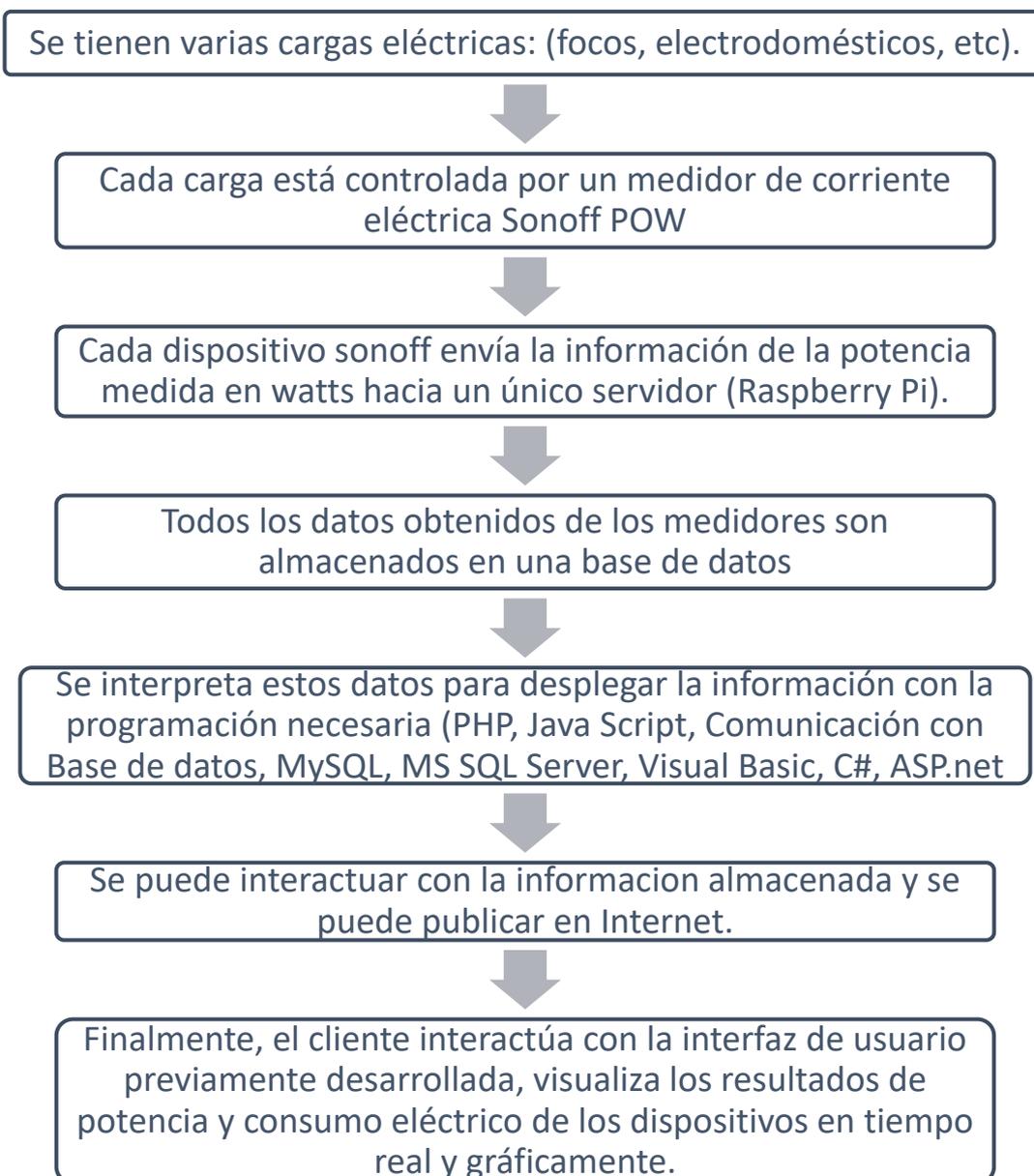


Figura 31. Planteo Inicial del Sistema.

3.1.2 Especificaciones del Diagrama del Sistema



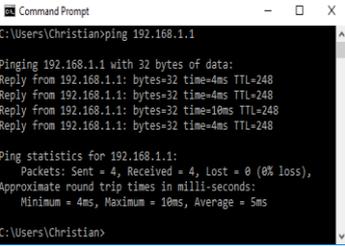
Figura 32. Componentes que constituyen el diagrama del sistema

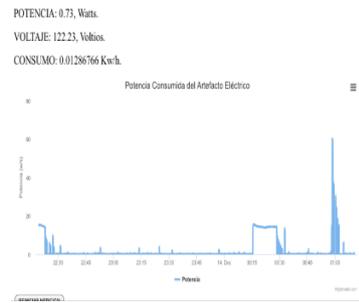
Adaptado de (Daraz, s.f.) (Raspberry, s.f.)

3.1.3 Especificaciones Técnicas del Sistema

Tabla 15

Especificaciones Técnicas de los componentes del Sistema.

Nombre	Función	Equipo
Sonoff POW 1,2,3	<ul style="list-style-type: none"> -Medidor de Potencia -Tecnología wifi 802.11 a/b/g/n Soportas cargas de hasta 16 Amperios -Pines de programación para insertar firmware -Sistema Operativo Raspbian 	
Servidor (Raspberry Pi)	<ul style="list-style-type: none"> -Se integra con otros lenguajes de programación. -Tecnología cableada e inalámbrica. 	
Router	<ul style="list-style-type: none"> -Asigna todas direcciones IP a los dispositivos que conforman la red. 	
Conexión entre Dispositivos	<ul style="list-style-type: none"> -Verifica que no haya caídas de enlaces y que los tiempos de latencia no sean elevados. -Ping entre dispositivos 	 <pre> C:\Users\Christian>ping 192.168.1.1 Pinging 192.168.1.1 with 32 bytes of data: Reply from 192.168.1.1: bytes=32 time=4ms TTL=248 Reply from 192.168.1.1: bytes=32 time=4ms TTL=248 Reply from 192.168.1.1: bytes=32 time=10ms TTL=248 Reply from 192.168.1.1: bytes=32 time=4ms TTL=248 Ping statistics for 192.168.1.1: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 4ms, Maximum = 10ms, Average = 5ms C:\Users\Christian> </pre>

Bases de Datos	Es el almacenamiento de los datos puede ser: MySQL, MS SQL Server, Oracle entre otros.	
Programas de Integración	Visual Basic, Java Script se integran con servicios de Internet.	
Interfaz del Cliente	Pantalla grafica que se muestra al cliente, en donde se encuentran toda la información de potencia, voltaje, consumo y diagramas	
Cliente	Son los dispositivos finales como: smartphones, tables, ordenadores; los cuales interactúan con el sistema y despliega toda la información	

Adaptado de (Itead, s.f) (Raspberry, s.f.)

3.2 Prototipo

Está integrado con varios dispositivos electrónicos como: Raspberry Pi, Sonoff POW, placa de pruebas Node MCU, regleta, router, además todas las adaptaciones eléctricas necesarias para la implementación del proyecto; los elementos se encuentran conectados inalámbricamente y para energizar es necesario un solo tomacorriente de 120 voltios. En lo que respecta a software se desarrolla con distintos lenguajes de programación y entorno web; todo en conjunto: el hardware y software dan como resultado el éxito del prototipo.

3.2.1 Diseño del Prototipo

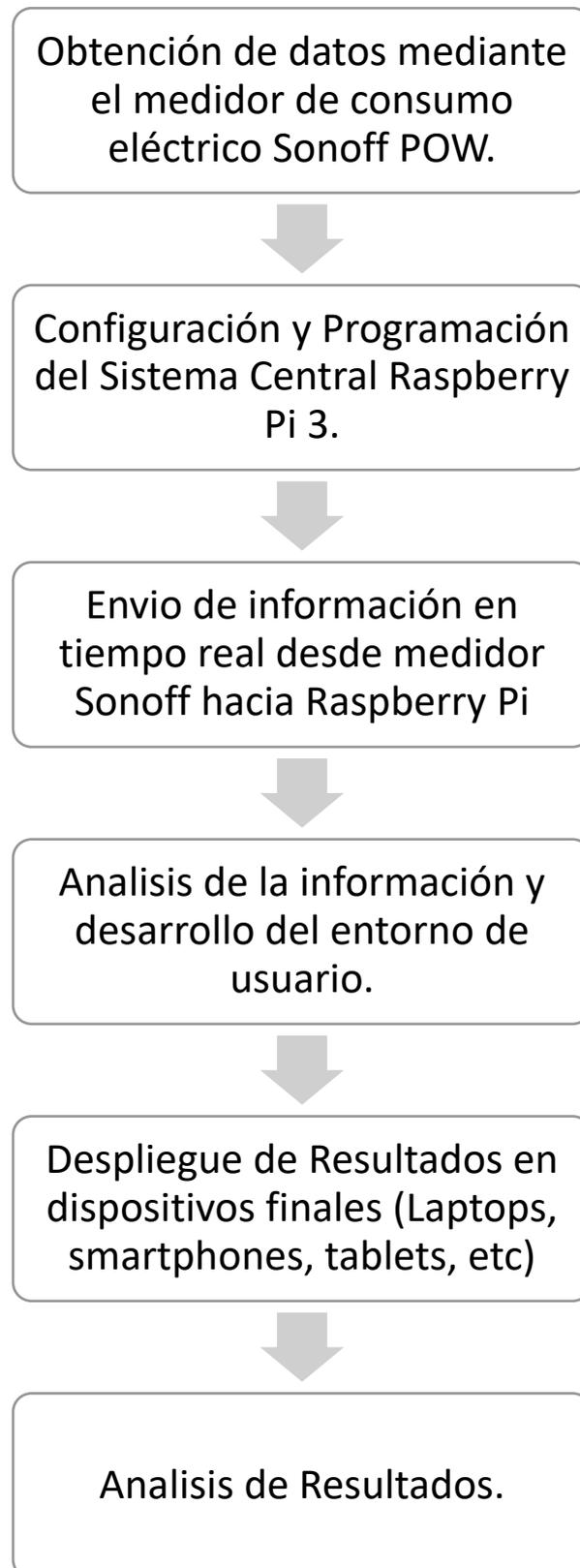


Figura 33. Descripción del diseño del prototipo.

3.2.2 Diagrama del Prototipo



Figura 34. Bosquejo de los componentes del prototipo.

Adaptado de Future Electronics, s.f.

3.2.3 Descripción Técnica del Prototipo

Tabla 16

Especificaciones técnicas de los componentes del prototipo.

Nombre	Función	Equipo
Sonoff POW	<ul style="list-style-type: none"> -Medidor de Potencia -Tecnología wifi 802.11 a/b/g/n Soporta cargas de hasta 16 Amperios 	

NODE MCU

- Pines de programación para insertar firmware
- Tecnología wifi 802.11 a/b/g/n.



Raspberry Pi 3

- Posee integrado ESP8266.
- Permite programación
- Servidor
- Almacenamiento de los datos de potencia eléctrica



Integración (Programación)

- Ejecución de los programas para almacenar los datos en un archivo ".txt", lee y despliega en el widget con javascript.

```

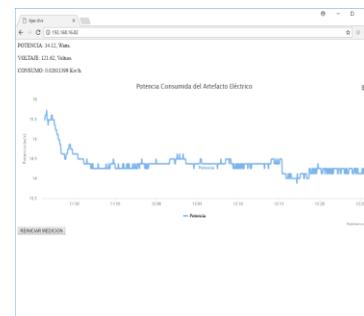
MyRaspberryPiPhuaRouter (raspberrypi) VNC Viewer
index.php
File Edit Search Options Help
<html>
<head>
<title>Ajax divs</title>
<script language="javascript" type="text/javascript">
function refreshDivs(divid,secs,url)
{
// define our vars
var divid,secs,url,fetch_unix_timestamp;

// Chequeamos que las variables no esten vacias..
if(divid == ""){ alert('Error: escribe el id del div que qu
else if(!document.getElementById(divid)){ alert('Error: el
else if(secs == ""){ alert('Error: indica la cantidad de s
else if(url == ""){ alert('Error: la URL del documento que

```

Presentación Interfaz Web

- Presentación de la interfaz final.
- Despliegue de los valores de potencia, consumo, voltaje, y graficas de monitoreo.



Router LBL-LINK

- Dispositivo master para la conexión de todos los dispositivos
- Asigna IPs a cada uno de los dispositivos que conforman el prototipo.



**Dispositivos
Finales**

-Conforma todos los dispositivos que puedan acceder a la red y que se puedan conectar a la IP fija asignada a la Raspberry como: smartphones, tablets, laptops, ordenadores, entre otros.



Adaptado de (Future Electronics, s.f.) (Raspberry, s.f.)

3.2.4 Capas de interacción con el modelo OSI

Tabla 17

Relación entre las capas del modelo OSI y el prototipo.

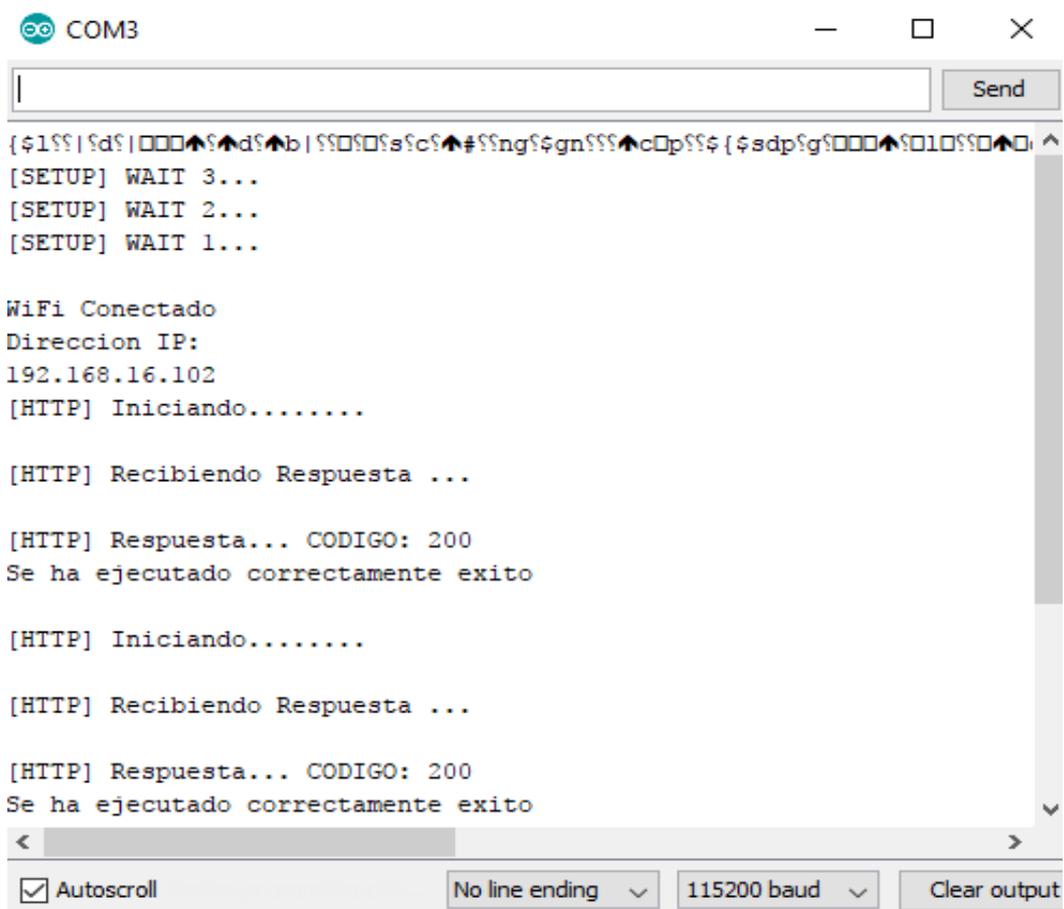
Capa	Descripción
Física	Conforma los integrados los módulos ESP8266 de Sonoff POW y la Raspberry.
Enlace de Datos	Protocolos de comunicación para este proyecto HTML.
Red de Conexión	Corresponde al Router y el direccionamiento Verificación de conexión entre todos los dispositivos de la red.
Presentación	Desarrollo del entorno de usuario.
Aplicación	Despliegue de la interfaz gráfica en cualquier dispositivo final que pueda acceder a la IP de la Raspberry.

3.3 Desarrollo del prototipo

A continuación, se detallan todos los pasos que se desarrollan para implementar el prototipo.

3.3.1 Determinar el módulo que posee el integrado ESP8266

El NODE MCU es un dispositivo basado en el ESP8266 corresponde a la placa de pruebas inicial, se programan datos aleatorios para comprobar la comunicación entre este dispositivo y el servidor (Raspberry PI). Este dispositivo permite optimizar el número de veces que se realiza la escritura de firmware y pruebas de funcionamiento hasta escribir el programa final en el medidor de corriente eléctrica.



```

COM3
[SETUP] WAIT 3...
[SETUP] WAIT 2...
[SETUP] WAIT 1...

WiFi Conectado
Direccion IP:
192.168.16.102
[HTTP] Iniciando.....

[HTTP] Recibiendo Respuesta ...

[HTTP] Respuesta... CODIGO: 200
Se ha ejecutado correctamente exito

[HTTP] Iniciando.....

[HTTP] Recibiendo Respuesta ...

[HTTP] Respuesta... CODIGO: 200
Se ha ejecutado correctamente exito
  
```

Figura 35. Node MCU funcionando en el terminal de Arduino.

3.3.2 Determinar el medidor ideal para el prototipo

En base a las capacidades y ventajas del conmutador eléctrico Sonoff POW frente a todos los demás medidores, y su característica principal de permitir la programación de código y sobrescribir sobre el firmware original de fábrica, se concluye que el medidor de corriente eléctrica Sonoff POW es el dispositivo electo para el desarrollo de este proyecto.



Figura 36. El Sonoff POW.

Tomado de OpenHacks, s.f.

3.3.3 Búsqueda de las librerías para la obtención de potencia

La información con respecto al módulo Sonoff POW se encuentra en Internet en los siguientes enlaces

- 1) <https://github.com/arendst/Sonoff-Tasmota>
- 2) <https://github.com/KmanOz/Sonoff-HomeAssistant>

Dentro de estos archivos se encuentran varias librerías; algunas de ellas son las que permiten capturar los datos correspondientes a potencia del dispositivo que se encuentre conectado al medidor de corriente eléctrica Sonoff POW. [Las Librerías se adjuntan en los Anexos 5 y 6]

3.3.4 Escritura del firmware al Sonoff POW

Una vez encontradas las librerías correspondientes a la obtención de potencia existentes en la web para el medidor Sonoff POW; se procede a escribir el código. El paso inicial es soldar los 4 pines para la programación que de fábrica viene sin soldar, después con el dispositivo conector serial USB a TTL se procede a grabar el código hacia la placa del medidor.

3.3.5 Programación de Sonoff POW

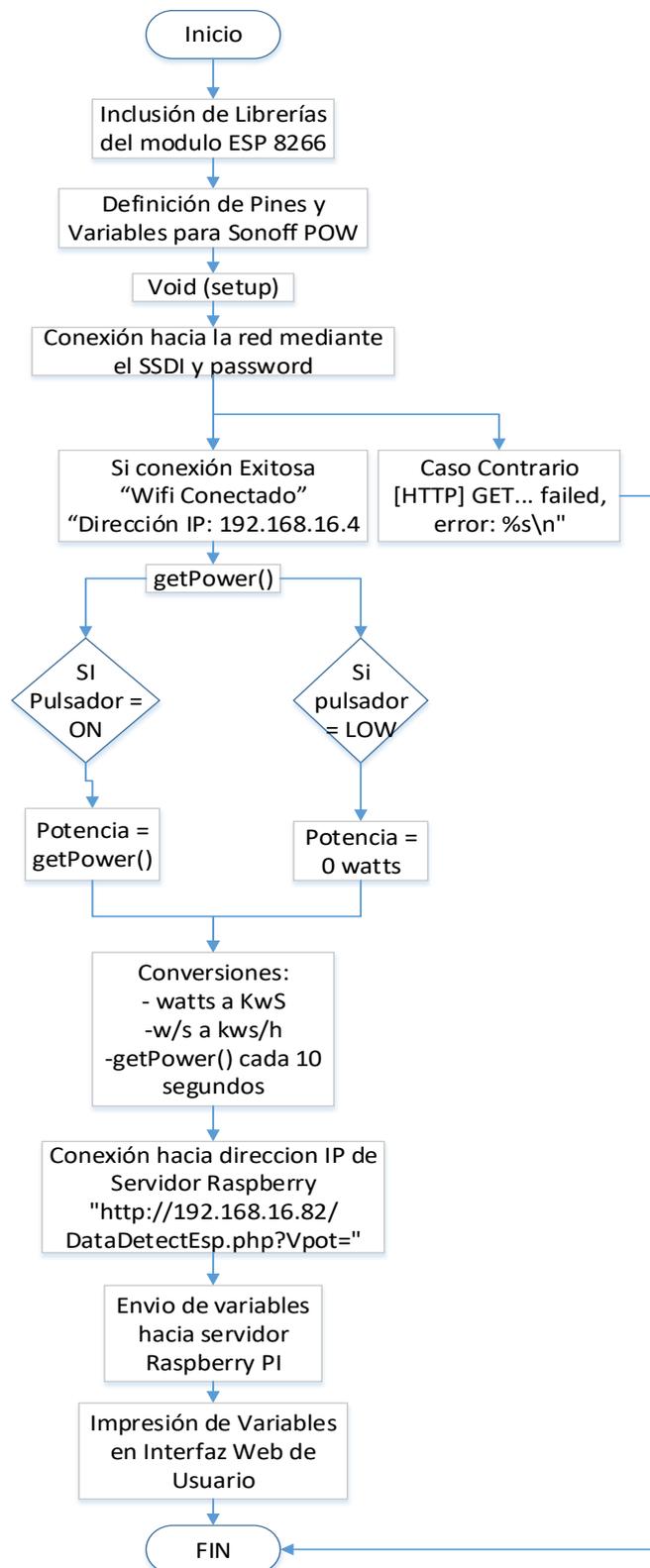


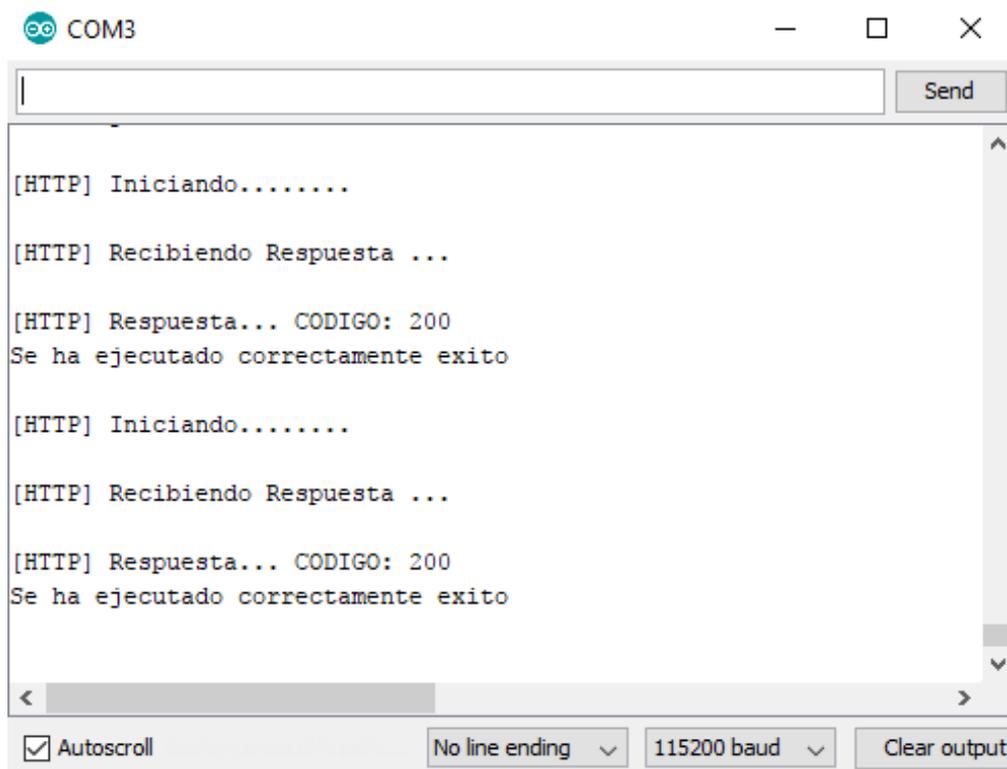
Figura 37. Diagrama de Flujo de Programación Sonoff Pow

En el ID de Arduino se programa todo el código necesario para obtener los datos de potencia del medidor y enviarlos a Raspberry Pi.

El código correspondiente a la programación se encuentra en el Anexo 3.

3.3.6 Verificar conexión entre Sonoff y un localhost.

Después de haber escrito el código, resta probar la conexión entre el medidor y un servidor local de la Raspberry, se tiene un resultado exitoso puesto que el medidor envía datos de potencia hacia la Raspberry y se verifica que no existe errores y hay total comunicación entre todos los dispositivos de la red.



```
COM3

[HTTP] Iniciando.....

[HTTP] Recibiendo Respuesta ...

[HTTP] Respuesta... CODIGO: 200
Se ha ejecutado correctamente exito

[HTTP] Iniciando.....

[HTTP] Recibiendo Respuesta ...

[HTTP] Respuesta... CODIGO: 200
Se ha ejecutado correctamente exito

Autoscroll No line ending 115200 baud Clear output
```

Figura 38. Respuestas de ejecución exitosa entre Sonoff y el servidor local

El conmutador Wifi Sonoff POW está integrada con una programación en la cual se declaran las variables de potencia voltaje y consumo; estas variables mediante las librerías de captura de información son obtenidas por el medidor y enviadas al servidor de Raspberry pi, si todo este proceso es de carácter exitoso se muestra la información en conjunto con las gráficas de monitoreo en tiempo

real en la interfaz en caso contrario se desplegará un mensaje de error y no desplegará información alguna.

3.3.7 Configuración Inicial de Raspberry Pi 3

Para realizar la configuración inicial de la Raspberry PI se deben seguir los siguientes pasos:

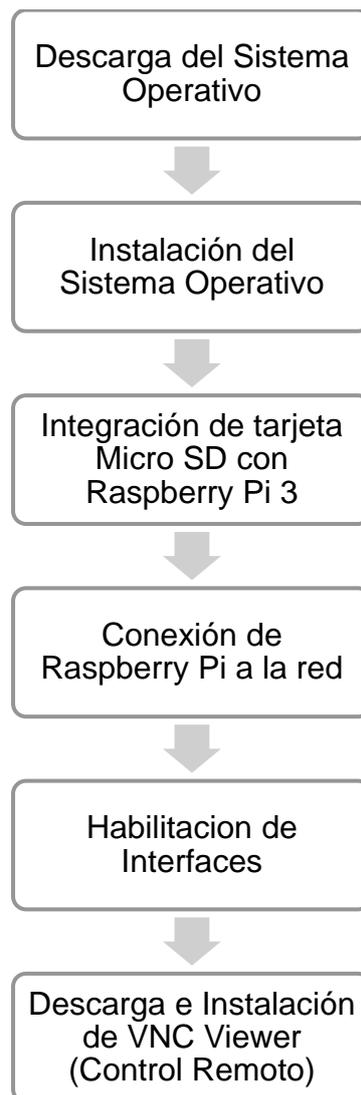


Figura 39. Diagrama de la Configuración Inicial de Raspberry Pi 3

3.3.7.1 Descarga del Sistema Operativo

En la página oficial Raspberry están disponibles las opciones de descarga en formato .ISO de las distribuciones Linux (Debian); la más utilizada es Raspbian

por su adaptabilidad. La imagen ISO se copia directamente a la micro SD. Cuando esta insertado en la ranura, se alimenta a la Raspberry Pi y automáticamente empieza a correr el sistema de arranque para dar el primer enfoque del sistema operativo Raspbian.

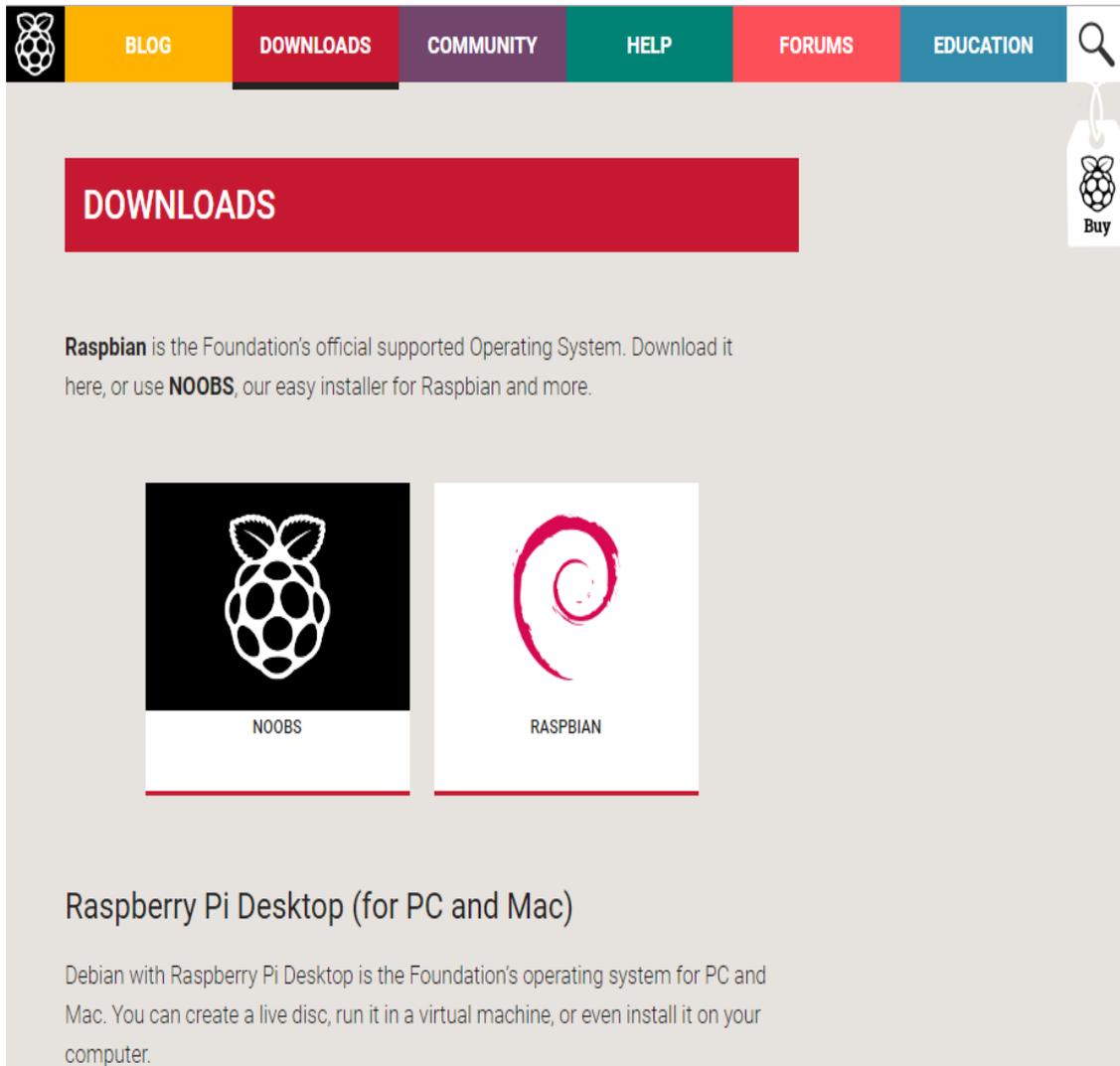


Figura 40. Página Oficial de Descargas Raspberry.

Adaptado de Raspberry, s.f

3.3.7.2 Instalación del Sistema Operativo

Con la ayuda del programa USB Image Tool se procede a grabar la imagen .ISO en la micro SD, de tal forma que después de realizado este proceso la micro SD se convierte en un dispositivo booteable para la Raspberry Pi; cargado con el sistema operativo Raspbian.

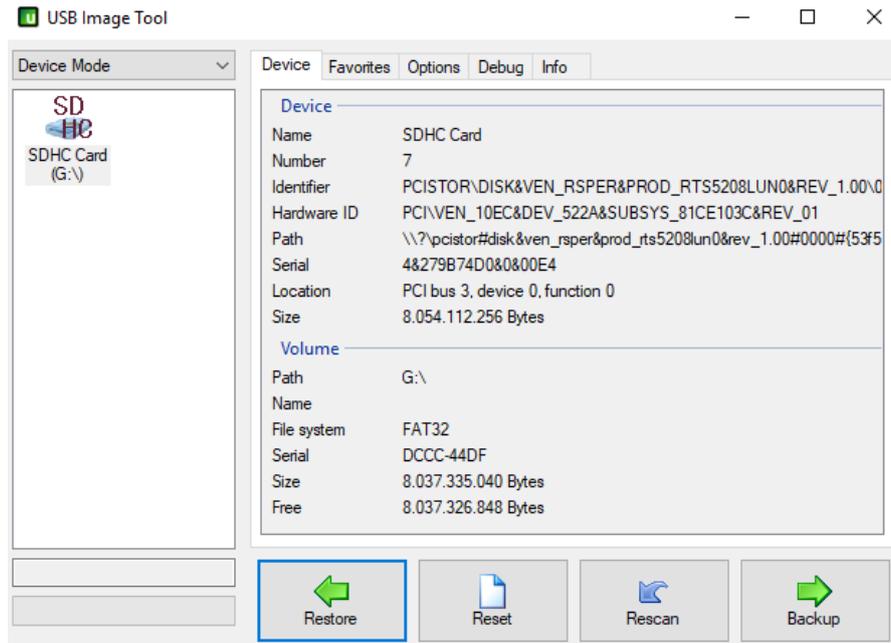


Figura 41. USB Image Tool.

3.3.7.3 Integración de tarjeta Micro SD con Raspberry Pi 3

La placa Raspberry Pi 3 conjuntamente con la tarjeta Micro SD previamente cargada con el sistema operativo se alimenta con un cargador de 120 voltios 1.5 amperios y se conecta a un televisor o monitor que posea entrada HDMI y sus respectivos periféricos: mouse y teclado.



Figura 42. Mensaje Bienvenida al Escritorio de Raspberry Pi.

3.3.7.4 Conexión de Raspberry Pi a la red

En la interfaz inicial del sistema operativo Raspbian el primer paso es establecer la conexión a la red posteriormente la asignación de una dirección IP estática que permita realizar una conexión VNC para acceso remoto desde cualquier ordenador, de tal forma que luego solo sea necesario energizar la placa Raspberry para el acceso remoto.

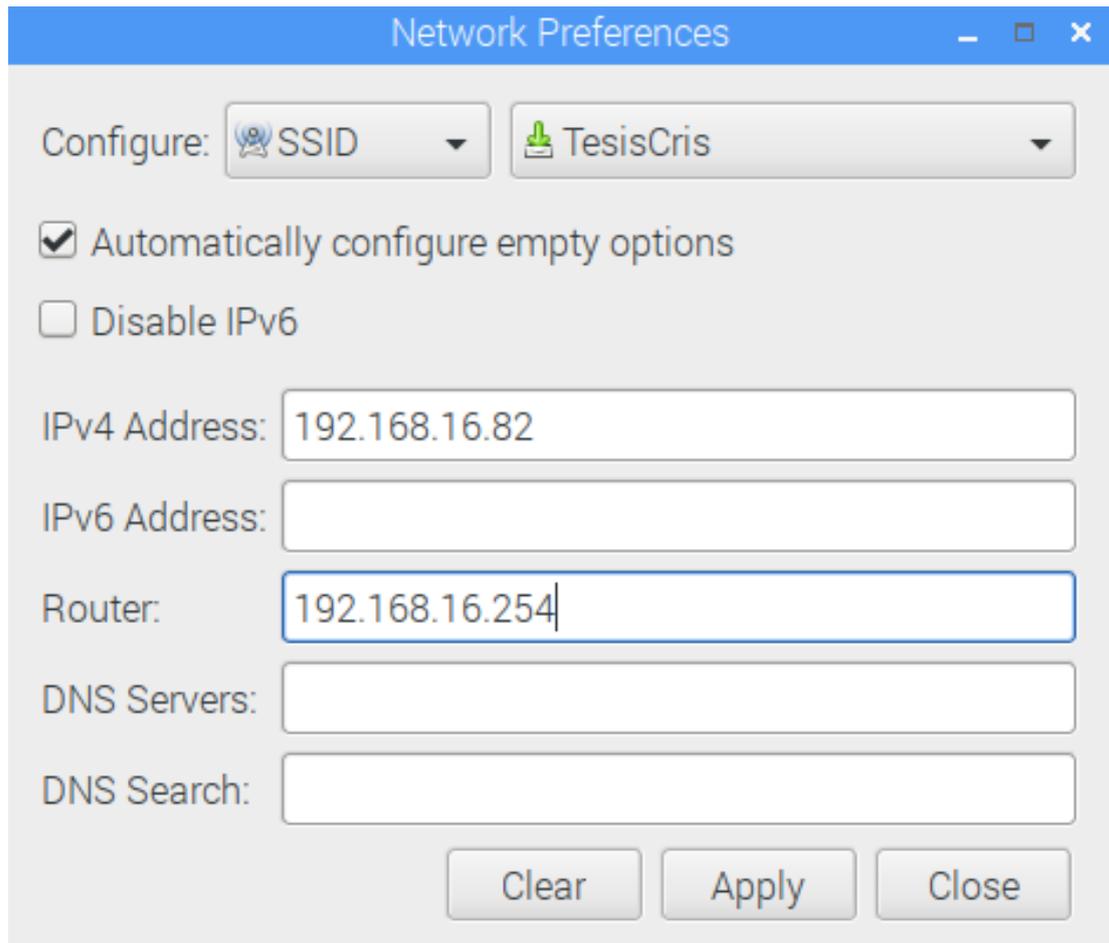


Figura 43. Asignación de Dirección IP estática

3.3.7.5 Habilitación de Interfaces

Para poder acceder a la interfaz de la Raspberry pi desde cualquier ordenador es necesario habilitar algunas de las interfaces que posee como: SSH y VNC de tal manera luego como en una conexión remota se accede mediante el usuario y la contraseña y ya no es necesario utilizar los periféricos para la conexión y tampoco disponer un televisor o monitor HDMI.

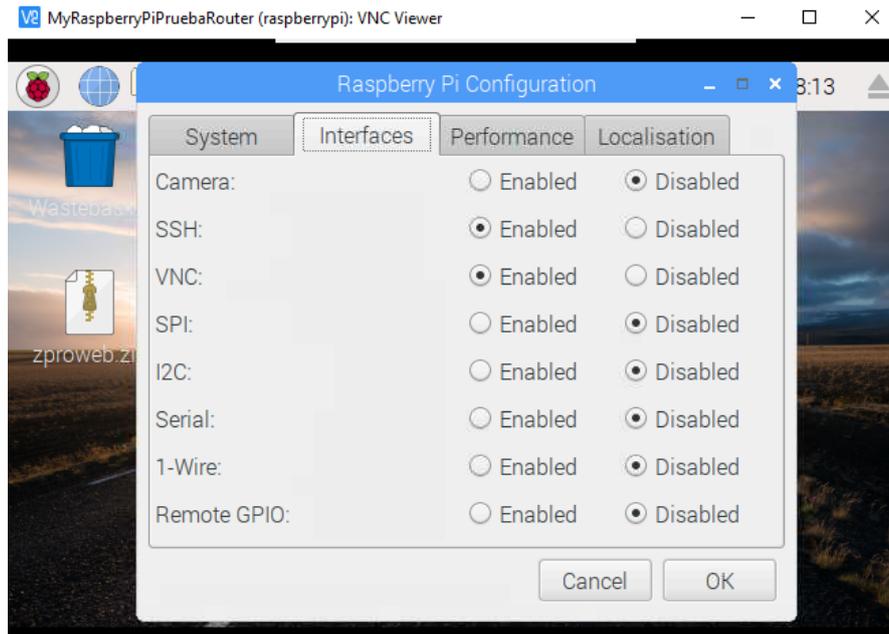


Figura 44. *Habilitación de servicios para conexión mediante SSH y VNC*

3.3.7.6 Descarga e Instalación de VNC Viewer

En la página oficial de Real VNC está disponible la opción para descargar el archivo VNC Viewer, una vez descargado se instala normalmente en el ordenador con un click en sí a todo.

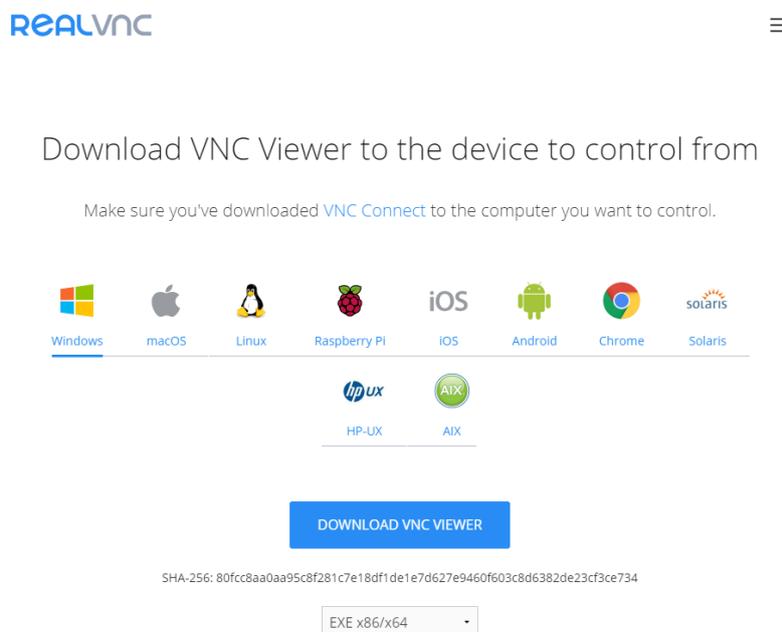


Figura 45. *Página Oficial RealVNC.*

Adaptado de Real VNC, 2017

Una vez terminada la instalación se ejecuta y se debe establecer una nueva conexión en la cual se deben ingresar algunos campos tales como: la dirección IP del servidor al cual se desea establecer una conexión y un nombre de identificación para esa sesión de VNC.

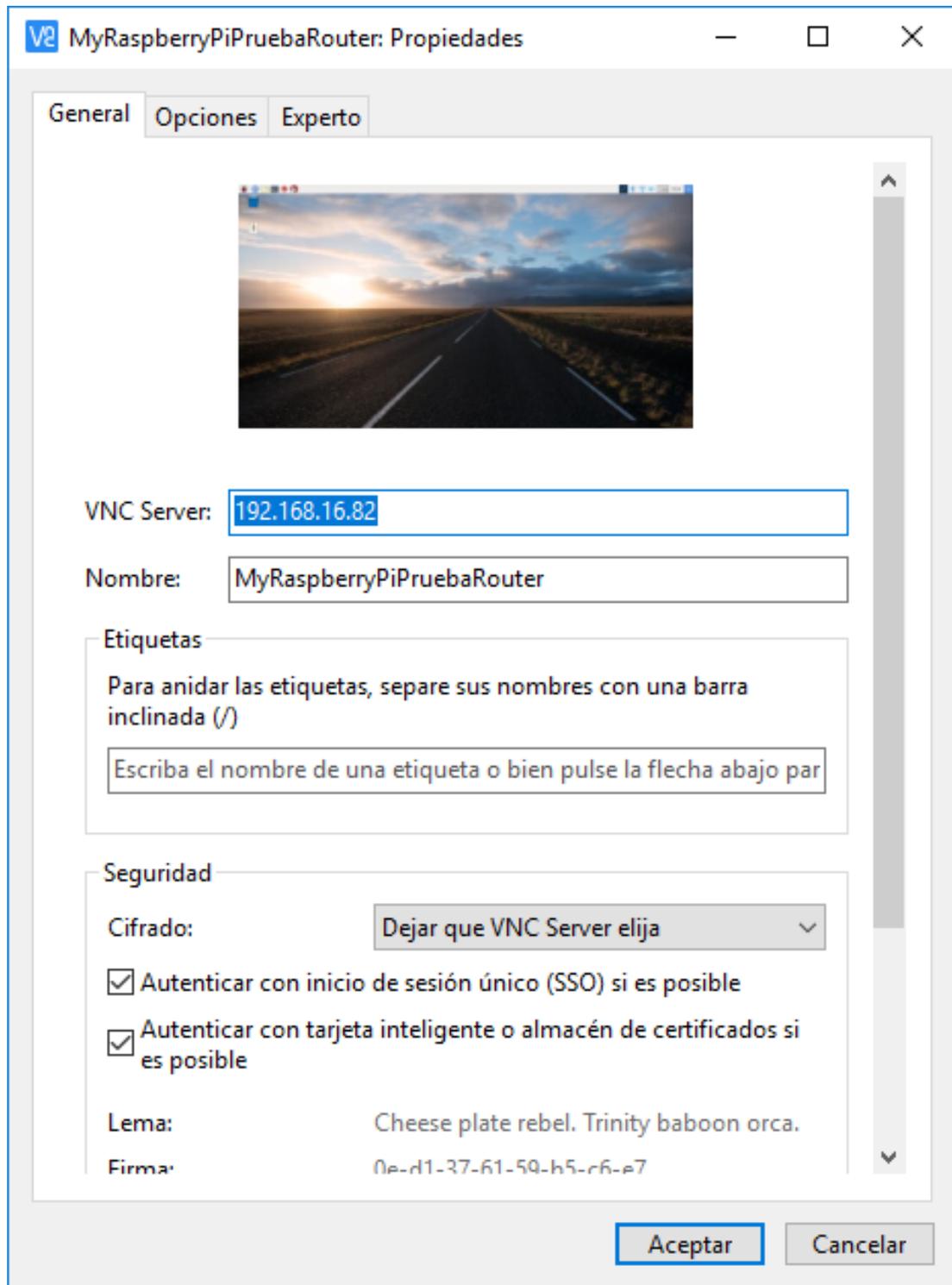


Figura 46. Establecimiento de Nueva Conexión.

Posteriormente el sistema pedirá las credenciales de usuario que corresponden al nombre y la contraseña para poder ingresar; una vez que la autenticación sea exitosa, se accederá a la interfaz de la Raspberry Pi.

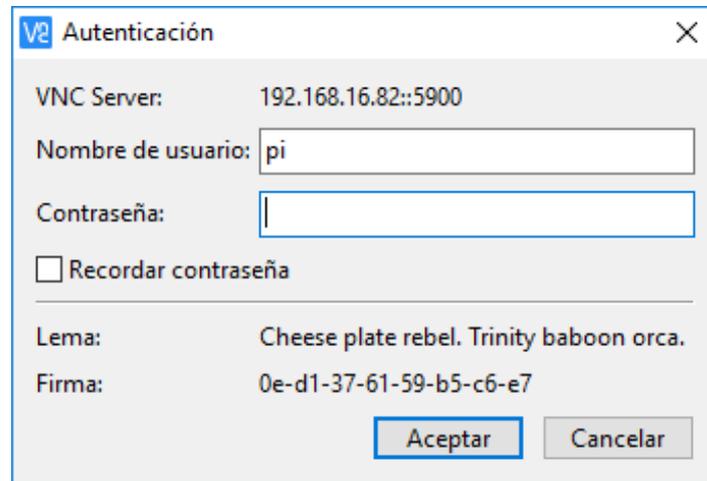


Figura 47. Autenticación para conexión por VNC

Finalmente, la Raspberry Pi 3 está lista para usar y empezar la programación mediante línea de comandos en un terminal.

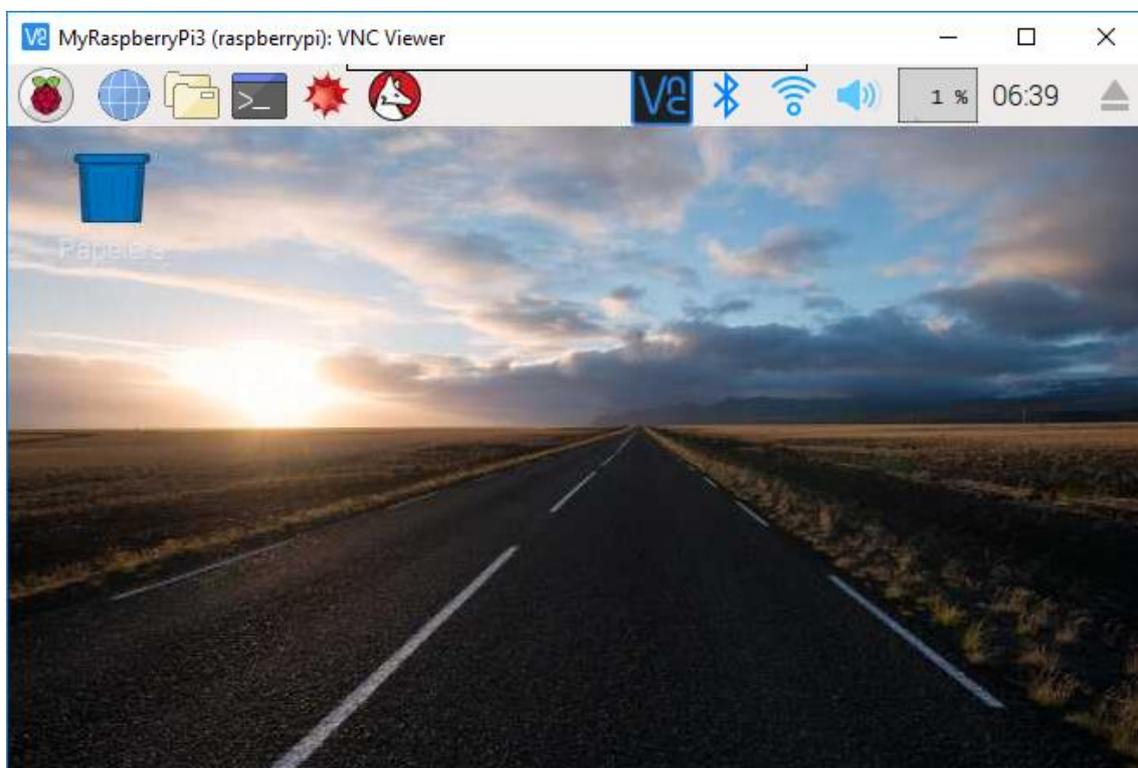


Figura 48. Interfaz del Sistema Operativo Raspbian

Una vez terminada la configuración es necesario programar la Raspberry para los fines necesarios, se empieza con las actualizaciones de la placa y sistema operativos, para ello se utiliza el terminal en el cual se introducen sentencias de comandos propios de Linux.

3.3.8 Instalación del Servidor Web

En esta sección se detalla todas las actualizaciones y los comandos que son necesarios para la correcta instalación del servidor web que se ejecuta sobre la Raspberry Pi 3.

3.3.8.1 Actualización de la Raspberry

Para actualizar la Raspberry se realiza mediante línea de comandos en el terminal de la plataforma con los siguientes códigos:

- `sudo apt-get update`
- `sudo apt-get upgrade`

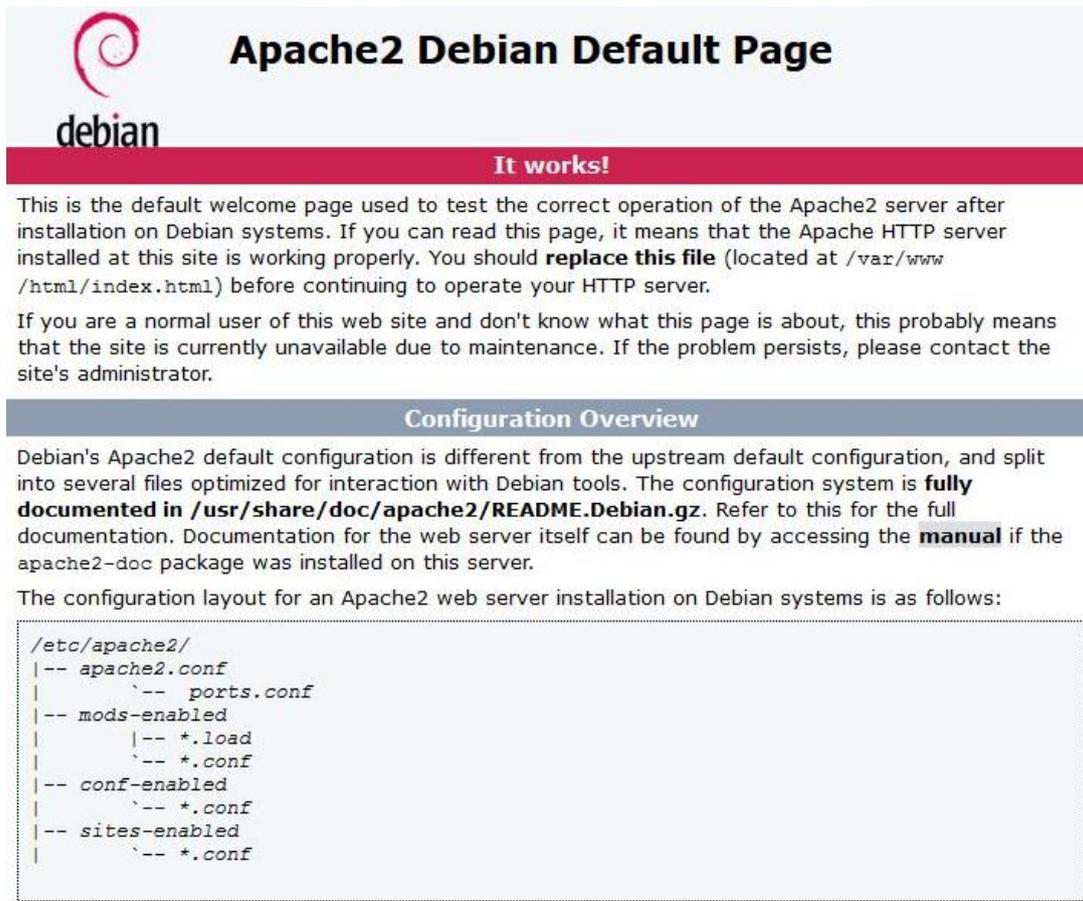
Lo que hacen estos comandos es descargar e instalar todas las actualizaciones disponibles en la Raspberry Pi 3.

3.3.8.2 Instalación de Apache

Para instalar Apache en la Raspberry se inserta el comando de instalación en el terminal y posteriormente se reinicia el servicio para que los cambios se apliquen.

- 1) `sudo apt-get install apache2`
- 2) `sudo service apache2 restart`

La comprobación que el servidor Apache se ha instalado correctamente es digitar la dirección IP de la Raspberry y en el navegador se debe observar los datos correspondientes al servidor Apache2 Debian.



Apache2 Debian Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Debian's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Debian tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Debian systems is as follows:

```

/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf

```

Figura 49. Servidor Apache corriendo sobre la Raspberry

3.3.8.3 Instalación de PHP

El soporte para contenido dinámico PHP y algunas librerías necesarias se ejecuta con la siguiente línea de comando. Los siguientes comandos indican como instalar php7.0

```
sudo apt-get install php7.0 php7.0-curl php7.0-gd php7.0-imap php7.0-json
php7.0-mcrypt php7.0-mysql php7.0-opcache php7.0-xmlrpc libapache2-mod-
php7.0 -y
```

Se reinicia la Raspberry con el comando: `sudo reboot`

Ahora la Raspberry tiene soporte para PHP y para realizar la comprobación es necesario dirigirse al directorio: `/var/www/html`

Se ejecuta el comando "ls-1" y se observa que existe el archivo `index.html` el mismo que se procede a removerlo con la sentencia: `sudo rm index.html`

Se añade otro fichero con el comando: `sudo nano index.php`. Y dentro de las etiquetas de PHP se procede a escribir el código.

Una vez realizados todos estos cambios se guardan y se apunta a la IP de la Raspberry en el navegador y se debe observar la información de php.

PHP Version 5.6.27-0+deb8u1 	
System	Linux RaspLuis 4.4.36-v7+ #933 SMP Fri Dec 2 22:03:01 GMT 2016 armv7l
Build Date	Oct 24 2016 17:26:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-curl.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20131106
PHP Extension	20131226
Zend Extension	220131226
Zend Extension Build	API20131226,NTS
PHP Extension Build	API20131226,NTS
Debug Build	no

Figura 50. PHP en Raspberry Pi.

3.3.9 Desarrollo del Entorno Web

Se levantan todos los servicios (php, apache). Con estos archivos se desarrolla la página web que va a visualizar el cliente.

 DataDetectEsp.php	6/11/2017 22:09	PHP File	2 KB
 DatosGrafico	6/11/2017 22:09	Text Document	1 KB
 EncerarEsp.php	6/11/2017 22:09	PHP File	2 KB
 EspValln.json	6/11/2017 22:09	JSON File	1 KB
 exporting	6/11/2017 22:09	JavaScript File	10 KB
 FechayHora	6/11/2017 22:09	Text Document	1 KB
 highcharts	6/11/2017 22:09	JavaScript File	189 KB
 index.php	6/11/2017 22:09	PHP File	4 KB
 LeerValEsp.php	6/11/2017 22:09	PHP File	1 KB
 series-label	6/11/2017 22:09	JavaScript File	7 KB

Figura 51. Archivos del Entorno Web en la Raspberry Pi.

3.3.9.1 DataDetecEsp.php

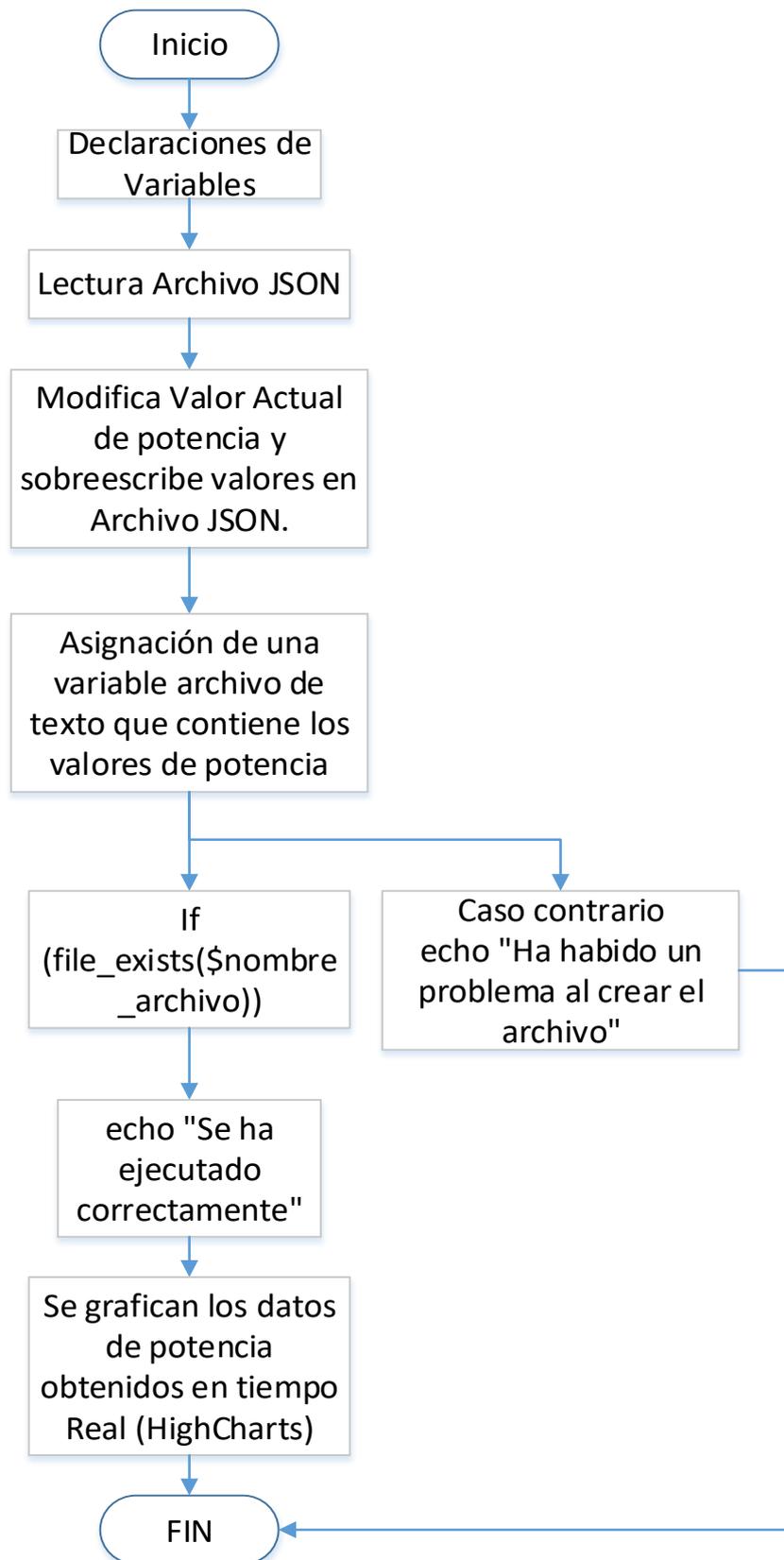


Figura 52. Diagrama de Flujo del Archivo "DataDetectEsp.php"

Este archivo corresponde a la parte de lectura de datos y grafica en tiempo real mediante los valores obtenidos de potencia en función del tiempo. Si todo es correcto se bosqueja un diagrama de líneas en el cual se pueden visualizar todos los valores obtenidos de potencia de cualquier artefacto eléctrico que haya sido conectado, de ser negativa la respuesta se despliega un mensaje de error.

El código de programación se encuentra en el Anexo 7.

3.3.9.2 EncerarEsp.php

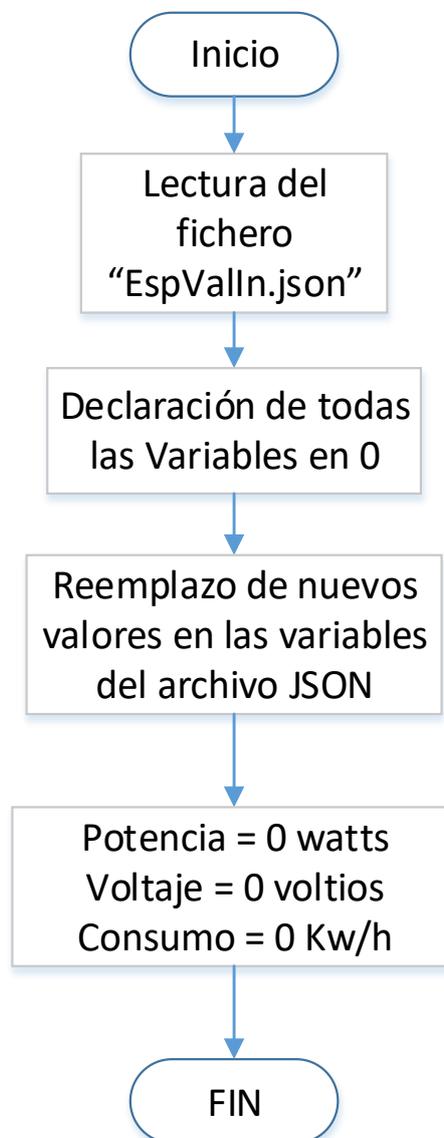


Figura 53. Diagrama de Flujo del Archivo "Encerar.php"

Este archivo corresponde al reinicio de la medición de potencia encerrando todos los valores correspondientes a: voltaje, consumo y potencia para comenzar una nueva grafica de potencia en función del tiempo.

El código de programación se encuentra en el Anexo 8.

3.3.9.3 LeerValEsp.php

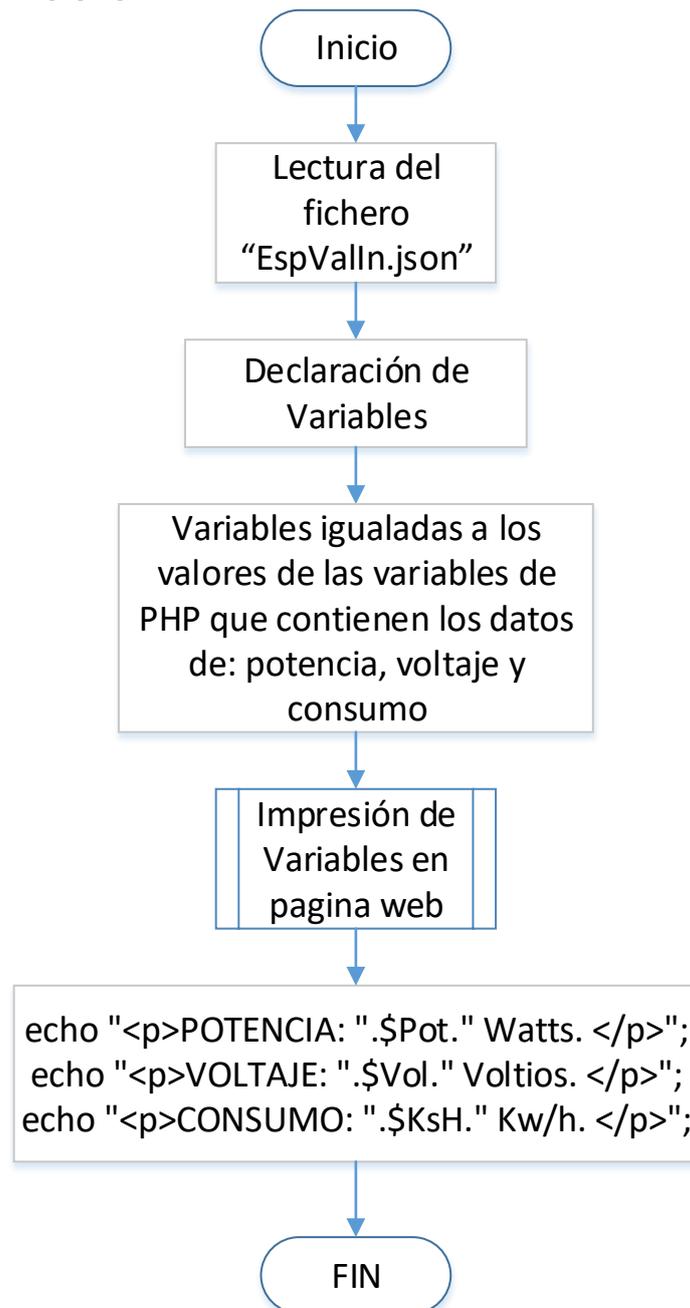


Figura 54. Diagrama de Flujo del Archivo "LeerValEsp.php"

Es el más importante se definen las variables para almacenar todos los datos de potencia, voltaje y consumo recopilados anteriormente y realiza la impresión en la interfaz web, para que el usuario puede visualizar los resultados en tiempo real textual y gráficamente. El código de programación se adjunta en el Anexo 9.

Los demás archivos corresponden a ficheros JavaScripts descargados de las plantillas “demos” del sitio oficial correspondiente a Highcharts necesarios para mostrar la gráfica y archivos de texto plano en donde se almacenan los datos de fecha y hora.

3.3.10 Escoger Plantilla de HighCharts

El primer paso es encontrar un modelo Highchart de utilidad. Después entender cómo funciona su código JavaScript y posteriormente integrar el modelo Highchart a la página web con los datos de potencia obtenidos por el medidor de corriente eléctrica Sonoff POW, el mismo que envía la información cada 10 segundos a la Raspberry y estos datos son almacenados en un archivo “.txt”. Posteriormente toda esa información es desplegada en el modelo Highchart. La información del grafico utilizado proviene del siguiente link: <https://www.highcharts.com/demo/spline-plot-bands>

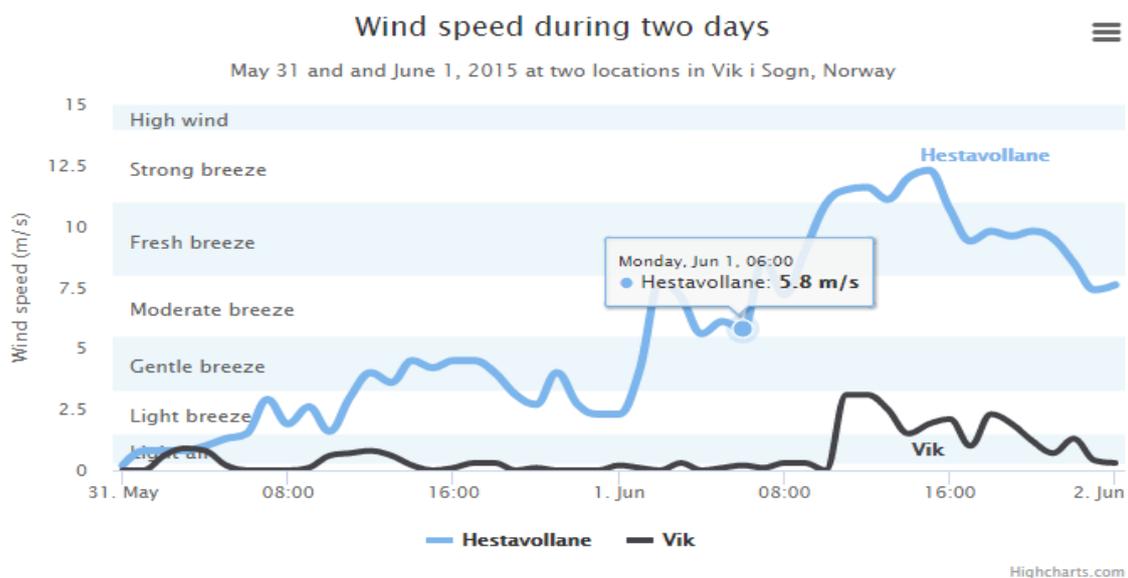


Figura 55. Plantilla Modelo HighChart empleado.

Tomado de Highcharts, 2018.

3.3.11 Interfaz de Presentación

Corresponde a la interfaz final de usuario que se conforma por los valores obtenidos de: potencia, voltaje y consumo que son actualizados automáticamente cada 10 segundos y la gráfica de monitoreo en donde se bosquejan todos los valores de potencia en un diagrama en tiempo real. En la gráfica al pasar el dedo o el clic por la línea azul en cualquier instante de tiempo se puede observar el: año, mes, día, hora, minuto y segundo exactos en los cuales se obtiene el dato de potencia.

El código de programación está disponible en el Anexo 4.

POTENCIA: 0.73, Watts.

VOLTAJE: 122.23, Voltios.

CONSUMO: 0.01286766 Kw/h.



Figura 56. Grafico Interfaz WEB.

3.4 Prototipo Final

El sistema prototipo se presenta en su versión final con todos sus componentes y un funcionamiento exitoso. El prototipo final es portátil y se encuentra en la Universidad de las Américas, en conjunto con otros proyectos de innovación tecnológica.



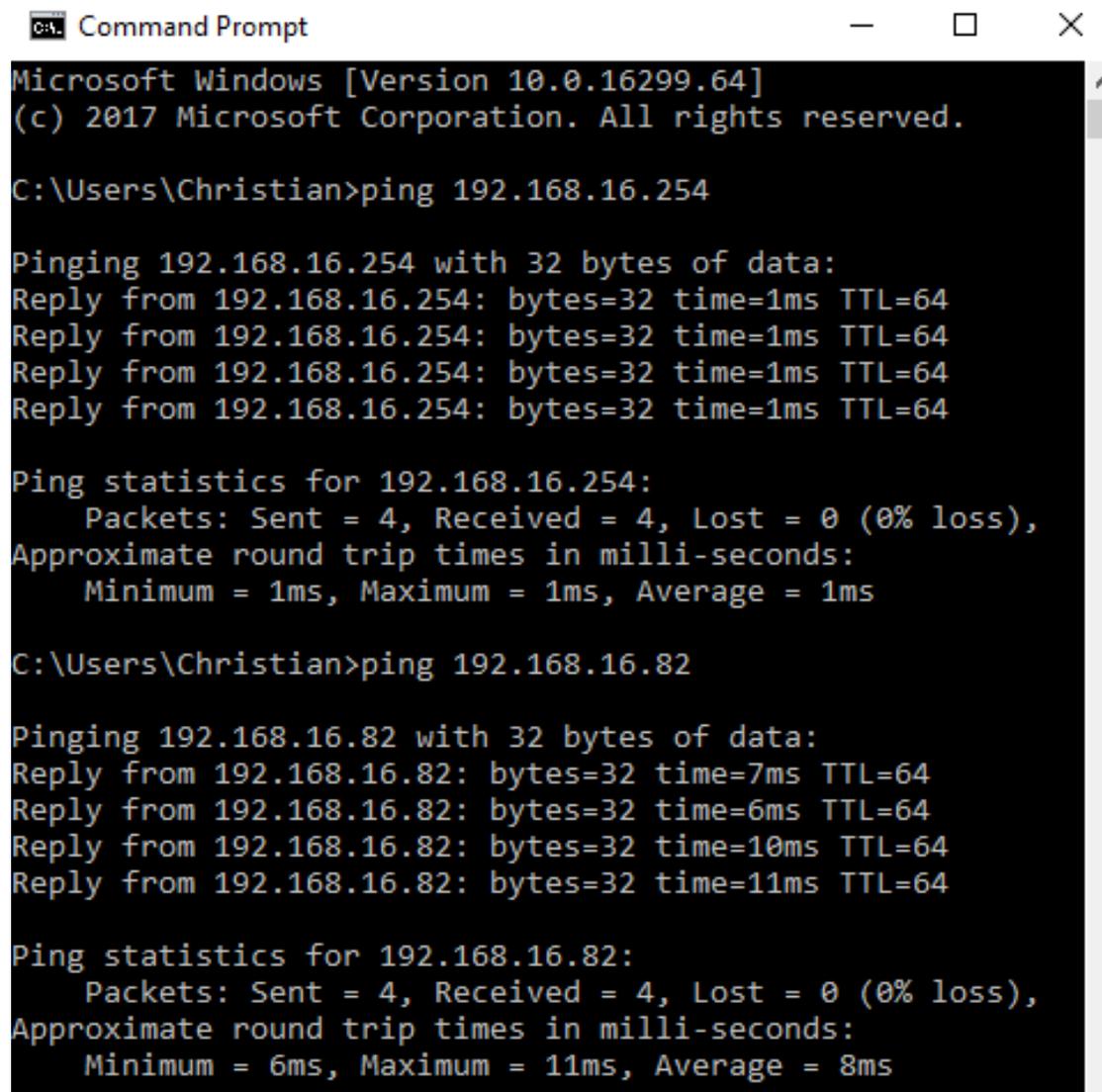
Figura 57. Presentación Final del Prototipo

4. CAPÍTULO IV. PRUEBAS Y RESULTADOS

En este capítulo se especifican todos los dispositivos de prueba y los distintos resultados que se obtienen de cada artefacto eléctrico.

4.1 Pruebas de Conexión

Las primeras pruebas que se deben realizar son las de conexión de todos los dispositivos a la red. Mediante el símbolo del sistema y con la sentencia “ping” se comprueba que todos los elementos envíen y reciban paquetes de manera satisfactoria.



```
Command Prompt
Microsoft Windows [Version 10.0.16299.64]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Christian>ping 192.168.16.254

Pinging 192.168.16.254 with 32 bytes of data:
Reply from 192.168.16.254: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.16.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\Christian>ping 192.168.16.82

Pinging 192.168.16.82 with 32 bytes of data:
Reply from 192.168.16.82: bytes=32 time=7ms TTL=64
Reply from 192.168.16.82: bytes=32 time=6ms TTL=64
Reply from 192.168.16.82: bytes=32 time=10ms TTL=64
Reply from 192.168.16.82: bytes=32 time=11ms TTL=64

Ping statistics for 192.168.16.82:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 6ms, Maximum = 11ms, Average = 8ms
```

Figura 58. Pruebas de Conexión hacia el Router y Raspberry Pi

4.2 Pruebas de Funcionamiento

Se realizan las pruebas de funcionamiento con varios electrodomésticos del entorno residencial para medir los datos correspondientes a potencia eléctrica consumida de cada uno de los dispositivos y la carga máxima que es soportada por el prototipo medidor.

4.2.1 Prueba 1: Node MCU con datos aleatorios.

El Node MCU corresponde a la placa de prueba para verificar el envío y recepción de información entre el dispositivo y la Raspberry Pi. Inicialmente los valores enviados por el kit de desarrollo de código abierto basado en el chip ESP8266 son aleatorios en un rango de 10 a 230 watts de potencia.

Resultados

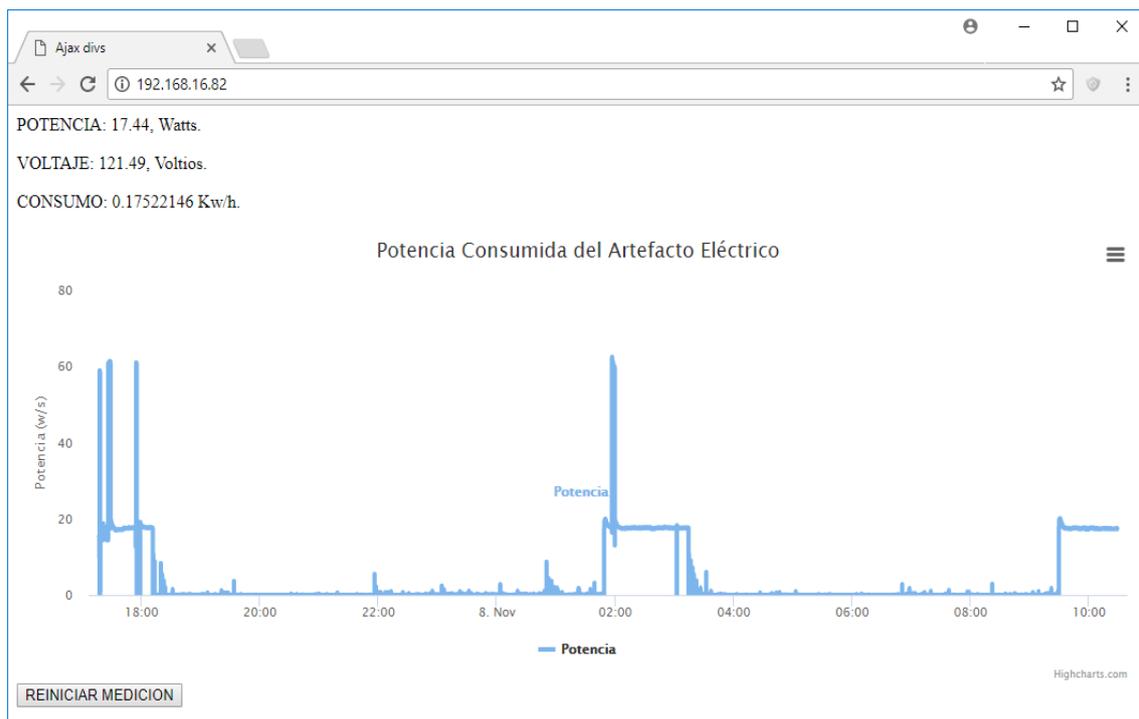


Figura 59. Resultados del Node MCU

4.2.2 Prueba 2: Prototipo con bobilla incandescente

El dispositivo de prueba corresponde a una bombilla incandescente que se encuentra conectado directamente al medidor de potencia.



Figura 60. Medición de potencia con prototipo y bombilla incandescente

Tabla 18

Resultados Prueba 2

Atributo	Descripción
Potencia	60 watts
Pico más bajo	58,84 watts
Pico más alto	62,9 watts
Voltaje	120-122 voltios
Consumo	0,12067535 Kw/h
Tiempo Encendido	2 horas

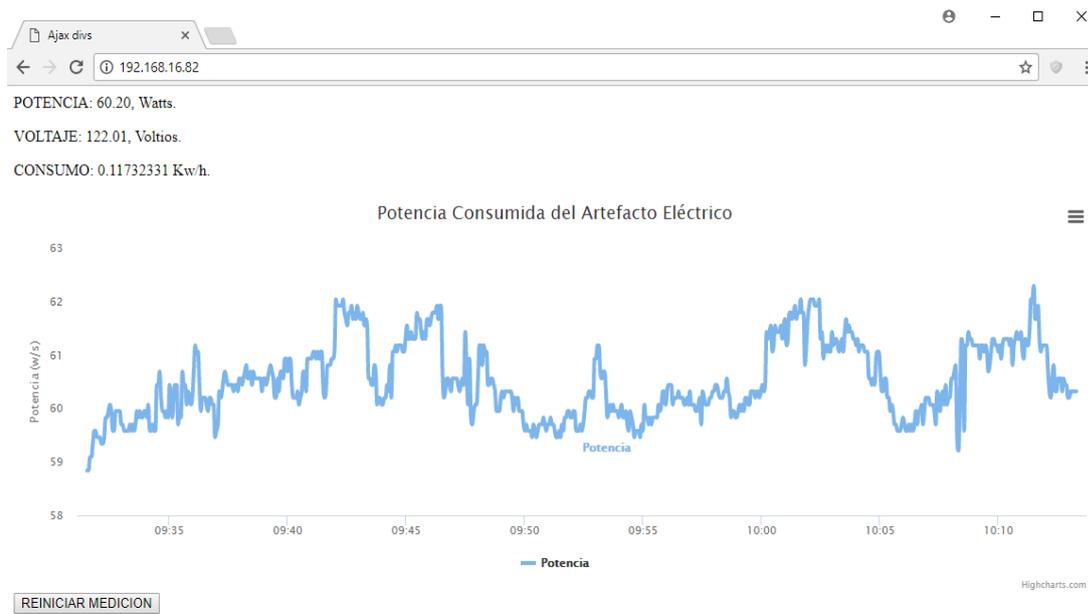


Figura 61. Resultados Gráficos Prueba 2

4.2.2 Prueba 3: Prototipo con bombilla ahorradora

La prueba siguiente consta de la integración de un foco ahorrador al prototipo.

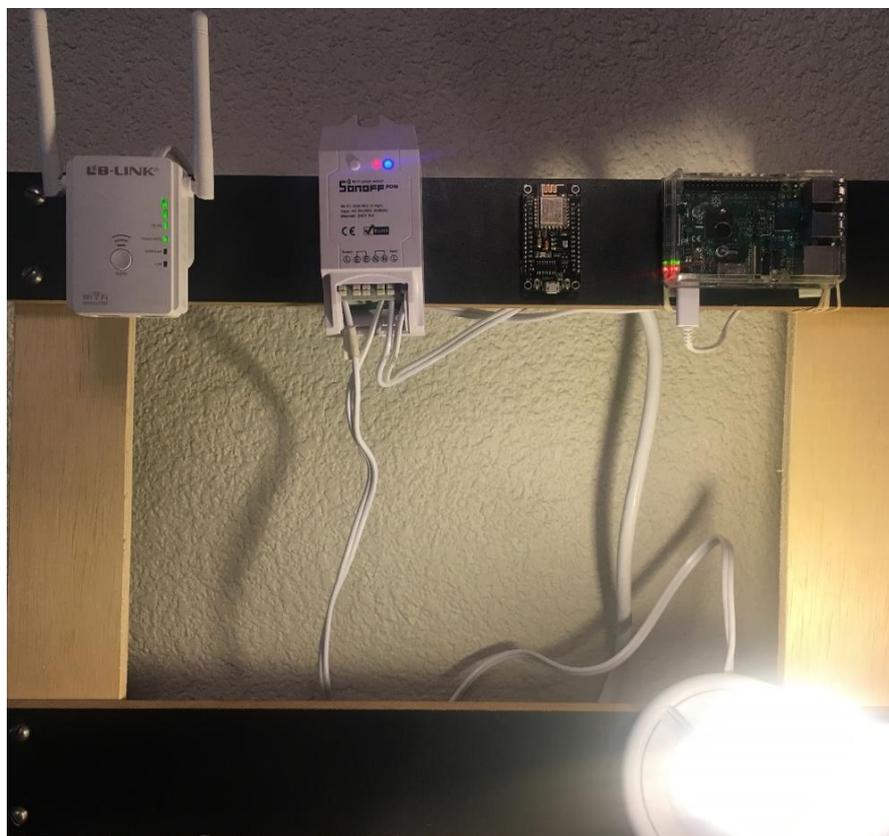


Figura 62. Medición de potencia con prototipo y foco ahorrador

Tabla 19

Resultados Prueba 3

Atributo	Descripción
Potencia	15 watts
Pico más bajo	15,7 watts
Pico más alto	14,13 watts
Voltaje	120-122 voltios
Consumo	0,0607535 Kw/h
Tiempo Encendido	2 horas

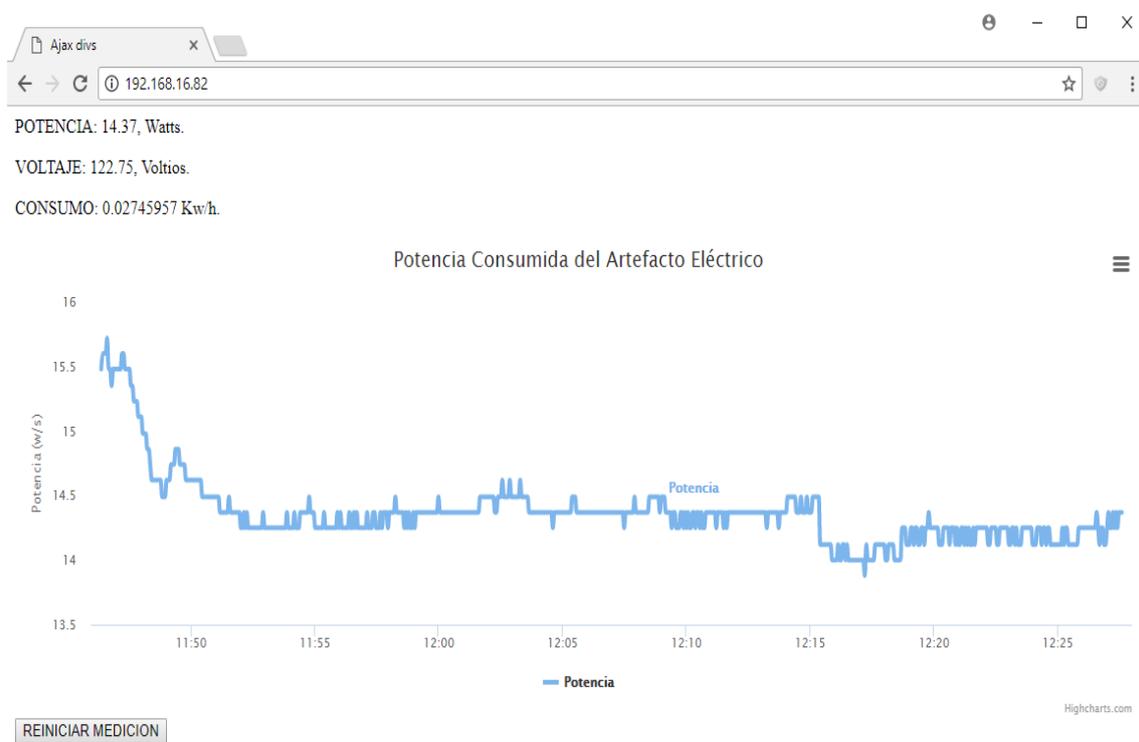


Figura 63. Resultados Gráficos Prueba 3

4.2.4 Prueba 4: Smart TV LG

Para esta prueba se conecta a un televisor LG al cortapicos asociado al prototipo para determinar la potencia que consume, así como los picos que ocurren cuando se realiza el encendido del dispositivo y luego su comportamiento mientras se encuentra prendido.



Figura 64. Medición de potencia con prototipo y Smart TV LG

Tabla 20

Características Del Televisor Smart TV LG

Atributo	Descripción
Tipo	Smart TV
Tecnología	Wifi
Tamaño	55 pulgadas
Alimentación	110V

Grafico



Tabla 21

Resultados Prueba 4

Atributo	Descripción
Potencia	22 watts
Pico más bajo	59,32 watts
Pico más alto	20,12 watts
Voltaje	120-122 voltios
Consumo	0,04324334 Kw/h
Tiempo Encendido	2 horas

*Figura 65. Resultados Gráficos Prueba 4*

Se puede apreciar que en el instante de tiempo que se realiza el encendido del televisor existe un pico de aproximadamente 59 watts. Después de un corto periodo de tiempo el dispositivo se normaliza y mientras se encuentra prendido muestra una línea continua de consumo que oscila entre los 20 y 23 watts que corresponde a la potencia original que el elemento debería consumir. De ese modo se verifica que solo al instante de tiempo que se realiza el primer encendido el televisor tiene un pico alto de consumo de potencia.

4.2.5 Prueba 5: Plancha eléctrica

Este es uno de los artefactos que más potencia eléctrica consume, se realiza un análisis de cuantos watts estaría consumiendo mientras esta encendida y se utiliza.



Figura 66. Medición de potencia con prototipo y plancha eléctrica

Tabla 22

Resultados Prueba 5

Atributo	Descripción
Potencia	800 watts
Pico más bajo	0 watts
Pico más alto	1048 watts
Voltaje	120-122 voltios
Consumo	0,19067535 Kw/h
Tiempo Encendido	2 horas

Grafico

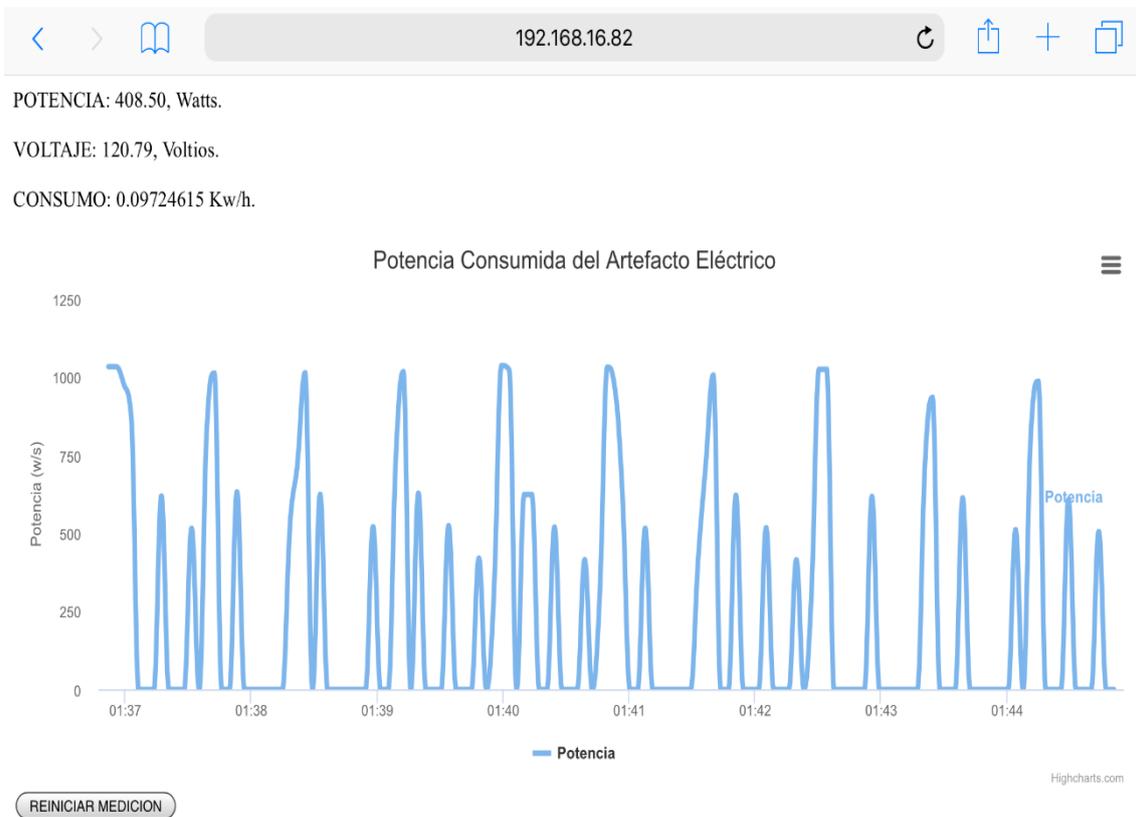


Figura 67. Resultados Gráficos Prueba 5

La plancha es uno de los artefactos eléctricos que más potencia consumen dentro de un hogar; es por tal motivo que su uso es por cortos lapsos de tiempo. Al mantener conectadas varias planchas dentro de un entorno residencial el aumento de consumo de potencia sería considerable y sería necesario utilizar otro tipo de maquinaria que optimice los recursos.

4.3 Dispositivos Finales de Monitoreo

Los dispositivos finales corresponden a los dispositivos que puede conectar a la red y desplegar la información del artefacto a verificar en una interfaz gráfica como: ordenadores, tablets, smartphones, entre otros.

La interface de ingreso para verificar la información del prototipo admite a varios dispositivos conectados simultáneamente a la Raspberry.

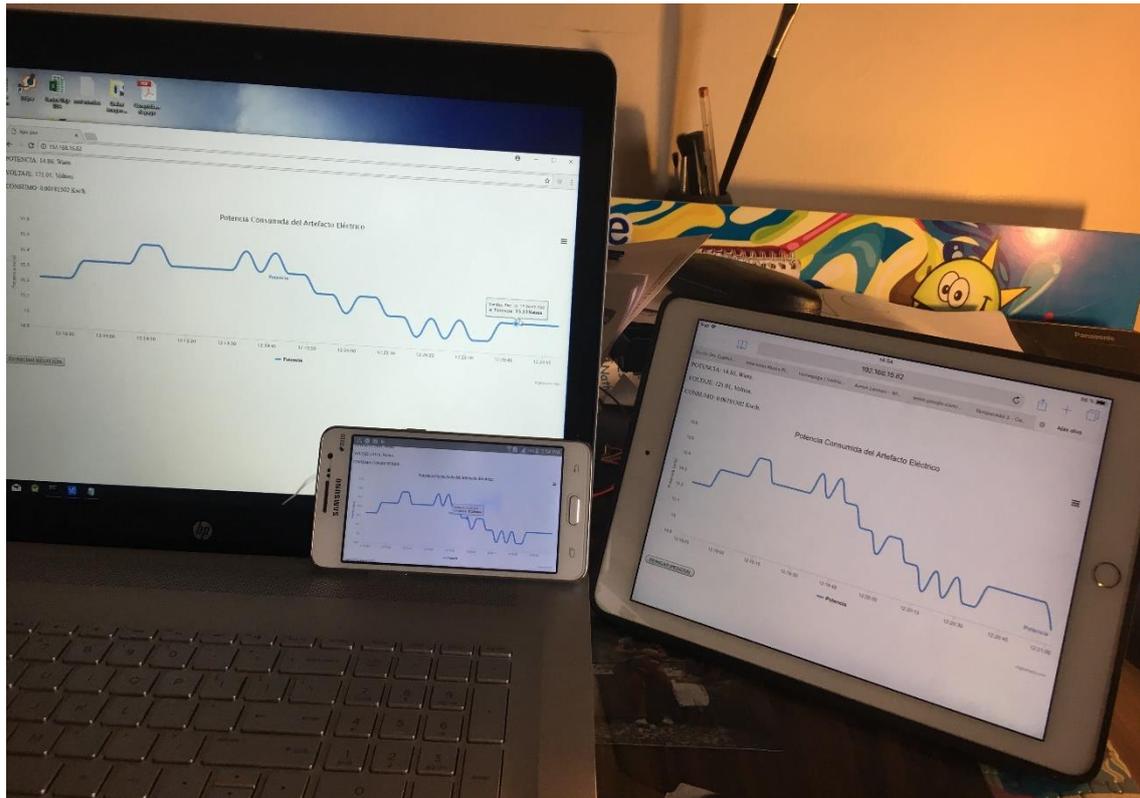


Figura 68. Pruebas de concurrencia en varios dispositivos finales.

4.4 Resultados Finales

Se obtiene los siguientes resultados:

Efectividad de funcionamiento del 90% del prototipo y un margen de error de +/- 4.5 watts en cada una de las mediciones de potencia eléctrica que se realizan.

Es posible medir cargas simultáneamente, conectar, desconectar, añadir o conectar electrodomésticos y el prototipo determina la potencia eléctrica que corresponde a suma total de la potencia consumida por todos los elementos conectados a la regleta.

Éxito de envío y recepción de información entre el medidor Sonoff Pow y servidor Rasberry Pi dando como resultado el cumplimiento de los objetivos que se plantean.

El sistema prototipo puede almacenar datos de potencia de hasta 7 días, encendido por 24 horas; después de este tiempo seguirá almacenando la información, pero la gráfica se verá distorsionada o poco visible al ojo humano.

En las pruebas de concurrencia se comprueba que varios dispositivos finales pueden estar conectados simultáneamente a la Raspberry desplegando datos de potencia y gráficos en tiempo real sin afectar al rendimiento del prototipo.

La visualización de la información es de carácter multiplataforma al ser compatible con cualquier sistema operativo: IOS, Android, Linux entre otros. Ya que solo es necesario el ingreso al explorador web.

En el caso de apagar el dispositivo medidor del prototipo; el sistema seguirá funcionando y almacenando los datos de potencia que será de 0 watts hasta que se vuelva accionar nuevamente.

Finalmente, el prototipo medidor de consumo eléctrico puede permanecer conectado las 24 horas y los 7 días de la semana, sin ninguna restricción o limitación mientras todos los dispositivos se encuentren conectados en red para su correcto funcionamiento.

5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

Al poseer las siguientes características: Smart, control remoto, conmutador wifi, compatibilidad con cualquier sistema operativo, reportes de consumo y 4 puertos de programación diseñados especialmente para grabar firmware externo convierten al Sonoff POW en el elemento ideal para el desarrollo e implementación del prototipo en conjunto con la plataforma Raspberry Pi.

El análisis de tecnologías determino que la manera más optima de realizar transferencia de información es utilizado el protocolo de comunicación wifi y los lenguajes de programación: Arduino y PHP que permite la integración del módulo medidor con Raspberry Pi.

En su versión final físicamente el sistema prototipo es portable y fácilmente integrable en un rack de comunicaciones mediante un módulo de 6 unidades en el cual quedan fijos los elementos de medición; todas las comunicaciones son inalámbricas y únicamente se requiere de una conexión a un tomacorriente de 120 voltios.

La Raspberry Pi es un dispositivo que permite actuar como un servidor sin necesidad de tener salida al internet, la programación e interconexión entre el medidor y la plataforma permite desplegar la interfaz de usuario en donde se visualizan todos los datos correspondientes a: potencia, voltaje, consumo y monitoreo mediante graficas en tiempo real.

En las pruebas correspondiente a los focos: incandescente y ahorrador con un consumo de potencia de: 60 y 15 watts respectivamente, con un margen de error de +/- 3.1 watts; se comprueba experimentalmente que se denomina bombillo ahorrador por que consume el 75% menos potencia eléctrica que un incandescente, ideal para generar ahorro energético en el entorno residencial.

La interfaz web de prototipo es accesible a cualquier sistema operativo y dispositivo final que se pueda conectar a la dirección IP de Raspberry sin la necesidad de una aplicación o licencia.

En la prueba numero 4 los resultados gráficos arrojan que al momento de encender un televisor automáticamente se tiene un pico de 58 watts de potencia, después de pasado este proceso, su consumo se normaliza y oscila entre los 19 y 22 watts; estos valores también varían dependiendo de las características de cada televisión.

En la prueba número 5 se concluye que el electrodoméstico (plancha eléctrica); en estado pasivo tiene un bajo consumo energético, mientras que cuando está al 100% en estado activo es donde genera su mayor consumo de potencia eléctrica que oscila entre los 70 y 85 watts; dependiendo de las características de cada plancha eléctrica estos valores pueden variar.

El prototipo medidor de potencia eléctrica admite la conexión de cualquier artefacto eléctrico o electrónico simultáneamente; siempre y cuando la suma de todas sus cargas no supere los 16 amperios o 3500 watts de potencia.

Cuando el código de la programación ha sido grabado en el integrado ESP 8266, automáticamente el firmware original de fabrica queda obsoleto y por tal motivo ya no es accesible con la aplicación que fue diseñado inicialmente, sin embargo, permite seguir ingresando nuevo código cuantas veces se requieran.

En la exposición y pruebas realizadas en la Universidad de las Américas interactuando con varios dispositivos a la vez: focos, batidoras, parlantes, cargadores de teléfonos, plancha, entre otros. El sistema prototipo brindo datos de potencia, voltaje y consumo precisos y un funcionamiento exitoso teniendo más de 7 dispositivos conectados simultáneamente sin afectar el rendimiento de la interfaz web.

5.2 Recomendaciones

Se recomienda no llegar ni exceder la carga de 16 amperios para el prototipo medidor de corriente eléctrica.

Verificar que la conexión de (línea, fase, neutro estén en su correcto lugar).

Tener fusibles para reemplazo en caso de cortocircuito.

Es recomendable que la información se guarde en una base de datos para mediante consultas que puedan determinar tramos por tramos la información.

Al momento de realizar la grabación del código a la placa del prototipo, en ningún caso la alimentación debe exceder los 3,3 voltios puesto que el dispositivo se quemará y quedará inservible.

Es posible añadir más gráficos al servidor web de Raspberry Pi tales como: barras, pasteles entre otros.

En caso de que se necesite conectar más dispositivos para su medición, se recomienda implementar una red de medidores distribuidos en cada uno de los lugares que conforma un entorno hogar para distribuir las cargas, optimizar el consumo de la energía y de esa manera tener una administración y control más preciso de cada medidor de potencia.

El Raspberry PI podrá leer a otros medidores, sensores de calor, temperatura, etc; mientras se cumplan las especificaciones del sistema en hardware y software.

Se recomienda realizar una correcta soldadura de los pines de programación en el dispositivo de medición puesto que si no se realiza de manera adecuada puede generar conflictos al momento de grabar el código.

REFERENCIAS

- Adler. (s.f). ¿Cómo medir consumo de energía? Recuperado el 22 de Diciembre de 2017 de: http://www.adler-instrumentos.es/imagenes_web/notas_aplicacion/C%C3%B3mo%20medir%20el%20consumo%20de%20energ%C3%ADa.pdf
- Alexander, C y Sadiku, M. (2013). Fundamentos de Circuitos Eléctricos. (5a, ed). McGraw Hill.
- Aprendiendo Arduino. (s.f). El ESP 8266. Recuperado el 15 De Diciembre de 2017 de: <https://aprendiendoarduino.wordpress.com/tag/esp32/>
- Arce, J. (s.f). Curso Electrónica Básica – 1
- Contreras, M (2015). Desarrollo e implementación de un sistema de medición, monitoreo y control de carga eléctrica para aplicaciones domésticas. Cuenca, Ecuador.
- Cornejo, J. (2014). La nueva facturación de energía eléctrica. Barcelona, España: Ediciones Experiencia
- Craemer, K y Deconinck G. (s.f). *Analysis of State-of-the-art Smart Metering Communication Standards*
- Culturacion, (s.f). Ventajas y desventajas del WIFI. Recuperado el 12 De Diciembre de 2017 de: <http://culturacion.com/ventajas-y-desventajas-del-wifi/>
- Cumulation Technologies (s.f). *Raspberry Pi & IoT Prototypes*. Recuperado el 31 de Octubre de 2017 de: <https://www.cumulations.com/blogs/42/RaspberryPilotPrototypes>
- Espressif (2017). ESP 8266. Recuperado el 5 de Diciembre de 2017 de: <http://espressif.com/>

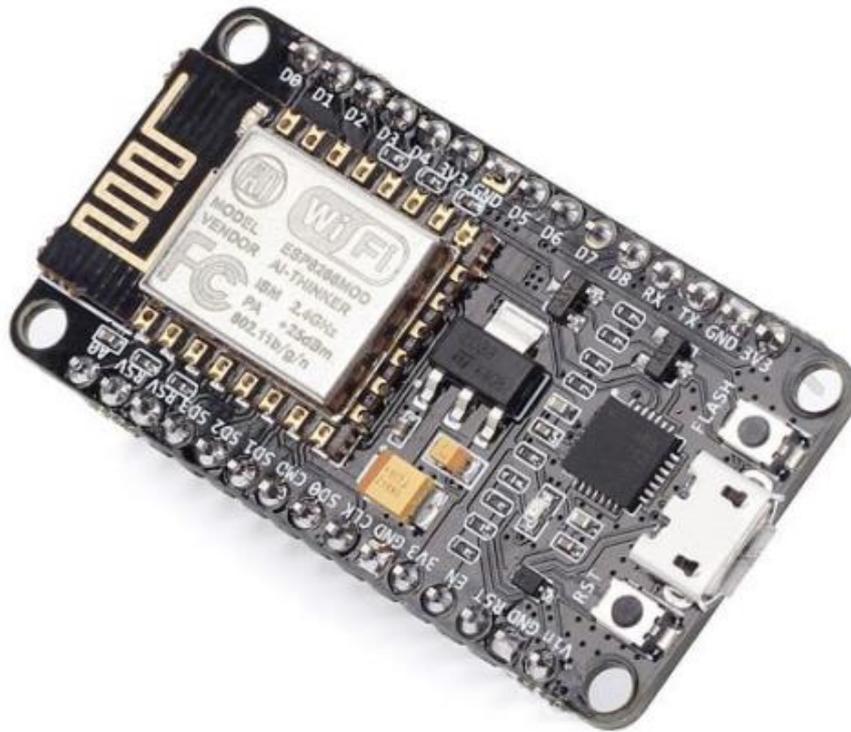
- Fabrica Digital, (s.f). Empezando a programar en Raspberry Pi con Python (s.f). Recuperado el 22 de Diciembre de 2017 de: <https://fabricadigital.org/2016/01/hola-pi-empezando-a-programar-en-raspberry-pi-con-python/>
- Forero, N. (2012). Normas de Comunicación en Serie: RS-232, RS-422 y Rs-485.
- Grainger, J y Stevenson, W. (1996). Análisis de Sistemas de Potencia. (1a, ed.). Mexico DF, Mexico: McGraw Hill.
- Highcharts, (2018). Recuperado el 22 de Diciembre de 2017 de: <https://www.highcharts.com/>
- Ingenium, (2018). Meter Bus 4 Canales. Recuperado el 12 de Diciembre de 2017 de: <http://ingeniumsl.com/website/productos/eficiencia-energetica/meterbus-4c/>
- Intel, (2017). Protocolos de Wifi y velocidad de datos. Recuperado el 26 de Noviembre de 2017 de <https://www.intel.la/content/www/xl/es/support/articles/000005725/network-and-i-o/wireless-networking.html>
- ISO, (s.f). Recuperado el 22 de Noviembre de 2017 de <https://www.iso.org/obp/ui/#iso:std:iso:50001:ed-1:v1:es>
- Itead. (2017). “*Sonoff POW*”. Recuperado el 31 de Octubre de 2017 de: <https://www.itead.cc/>
- Kamstrup (s.f). “*El kamstrup 382*”. Recuperado el 12 de Diciembre de 2017 de: <https://www.kamstrup.com/es-es/business-areas/electricity-metering>
- Leon, R (2002). Estudio de la tecnología ADSL para su implementación en la infraestructura de la red Andinatel S.A en la ciudad de Quito. Tomo 1. Quito, Ecuador.
- López, I. (2011). Introducción a Meter-Bus

- Mota, K., Rojas, Y (2015). Revista Ethos Venezolana, Ojeda, Venezuela: Ediciones Astro Data S.A
- Nasar, S. (1991). Sistemas Electrónicos de Potencia (1a, ed.). Kentucky, Estados Unidos: McGraw Hill.
- OpenHacks. (s.f). “*Sonoff Pow WiFi Switch with Power Consumption Measurement*”. Recuperado el 02 de Diciembre de 2017 de: [https://www.openhacks.com/page/productos/id/2460/title/Sonoff-Pow-WiFi-Switch-with-Power-Consumption-Measurement#lightbox\['galeria'\]/0/](https://www.openhacks.com/page/productos/id/2460/title/Sonoff-Pow-WiFi-Switch-with-Power-Consumption-Measurement#lightbox['galeria']/0/)
- Ortiz, D. y Rodríguez, O (s.f). Diseño e Implementación de un sistema de monitoreo energético en el Campus Rodríguez Lara – Espe Extensión Latacunga. Latacunga, Ecuador.
- Perez, J., Pineda, M., Puche, R y Sapena A. (2016). “*Introduction to Electric Installations and Technology*”. (1a, ed.). Valencia, España: Editorial Universitat Politècnica de Valencia.
- Raspberry Pi (s.f). Recuperado el 29 de Diciembre de 2017 de: <https://www.raspberrypi.org/>
- Real VNC (2018). Recuperado el 20 de octubre de 2017 de: <https://www.realvnc.com/en/connect/download/viewer/>
- Tamarkin, T. (1992). *Automatic Meter Reading*. (Volume 50)
- Tapia, D., Cueli, J., García, O., Corchado, J., Bajo, J., Saavedra, A. (2007) Identificación por Radio Frecuencia: Fundamentos y Aplicaciones. Salamanca, España

ANEXOS

ANEXO 1

Datasheet NodeMCU 1.0 (V2)



The NodeMcu is an open-source firmware and development kit that helps you to Prototype your IOT product within a few Lua script lines.

Features:

- Open-source
- Interactive
- Programmable
- Low cost
- Simple
- Smart
- WI-FI enabled

Arduino-like hardware IO

Advanced API for hardware IO, which can dramatically reduce the redundant work for configuring and manipulating hardware Code like arduino, but interactively in Lua script.

Nodejs style network API

Event-driven API for network applicaitons, which facilitates developers writing code running on a 5mm*5mm sized MCU in Nodejs style.

Greatly speed up your IOT application developing process.

Specification:

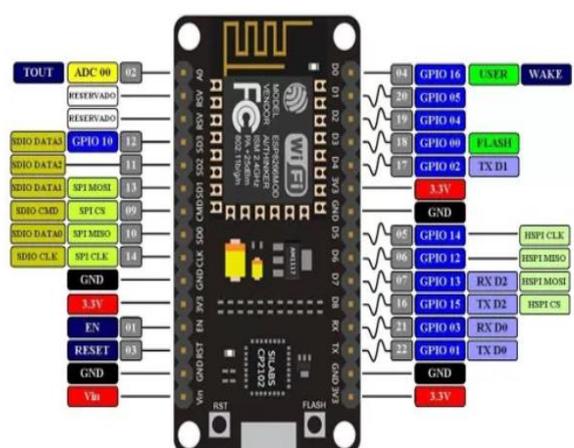
The Development Kit based on ESP8266, integrates GPIO, PWM, IIC, 1-Wire and ADC all in one board.

Power your development in the fastest way combining with NodeMCU Firmware!

- USB-TTL included, plug&play
- 10 GPIO, every GPIO can be PWM, I2C, 1-wire
- FCC CERTIFIED WI-FI module (Coming soon)
- PCB antenna

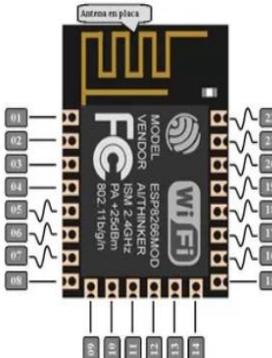
PLACA NodeMCU 1.0 (V2)

PINOUT



5V	ALIMENTACIÓN EXTERNA (de 5V a 10V).
3.3V	ALIMENTACIÓN INTERNA (desde la placa a dispositivos).
GND	TIERRA (GND Ground).
GPIO	PIN DE ENTRADA/SALIDA +3.3V (GPIO <i>General Purpose Input/Output</i>). Entrada digital \square Entrada analógica \sim . (Todas las salidas son digitales).
ADC	PIN DE SALIDA ANALÓGICA (el rango es entre +0V y +1V dividido en 1023 intervalos).
SPI	BUS SPI (<i>Serial Peripheral Interface</i>)
I2S/I	BUS I2S/I (<i>Hardware-Serial Peripheral Interface</i>).
SDIO	PINES PARA INICIO DEL ESP8266 DESDE UNA TARJETA SD. Para activar el modo SDIO el pin GPIO 15 debe estar en tensión cuando se enciende la placa.
TX/RX	COMUNICACIÓN SERIE TX/RX. Los pines GPIO01 y GPIO02 están conectados al puerto MicroUSB a través del convertor UART.

ESP8266 12E



NOTAS:

- ⚠ El voltaje de alimentación (V_{in}) debe estar comprendido entre 5 V y 10 V.
- ⚠ La intensidad de máxima de salida a un pin es de 12 mA. No se debe demandar mas intensidad para no quemar el procesador. La intensidad de salida normal será de 6 mA.
- Para activar el modo de reposo (*sleep mode*), unir los pines GPIO16 (D0) y RESET y poner el pin GPIO16 en tensión (*HIGH*). Para reactivar (*wakeup*), quitar la tensión en el pin GPIO16 (*LOW*). El sistema se reiniciará.
- ⚠ En *boot/reset/wakeup* (inicio/reseteo/reactivado), los pines GPIO00 (D3) ó GPIO15 (D8) **no** deben estar con tensión (+3.3V). **Tampoco** el pin GPIO2 (D4) debe estar conectado a tierra (+0.0V).
- ⚠ Los pines GPIO01 (TX) y GPIO03 (RX) se utilizan en el puerto MicroUSB, por lo que no se deben utilizar **simultáneamente** con otro dispositivo ya que la conexión se interfiere.
- ⚠ Los pines GPIO00 y GPIO02 **no** debe utilizarse para lectura (*input*). El pin GPIO09 **no** debe utilizarse ni para lectura ni para escritura (*input-output*).
- El pin GPIO02 (D4) controla el LED azul del ESP8266. Se enciende cuando no tiene tensión (+0.0V).
- El pin GPIO16 (D0) controla el LED azul de la placa. Se enciende cuando no tiene tensión (+0.0V). *(En la placa LoLin este LED no está disponible).*
- Para flashear, en el caso de que la placa quede bloqueada, se debe conectar el pin GPIO00 (D3) a tierra, el MicroUSB con el ordenador y ejecutar el flasher.



08-23-2016
V1.0

ANEXO 3

Código de Programación de Sonoff POW

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>

ESP8266WiFiMulti WiFiMulti;
#include "power.h"

#define BUTTON      0           // (Don't Change for Sonoff POW)
#define RELAY       12          // (Don't Change for Sonoff POW)
#define LED         15          // (Don't Change for Sonoff POW)

extern "C" {
    #include "user_interface.h"
}

int buttonState = 0;
ESP8266PowerClass power_read;

const double Seg10_Hora = 0.002777778; //tiempo de 10 segundos en horas

double VA_pot = 0;
double VA_vol = 0;
double VA_khr = 0;

int EstadoPulsador =0;
int SelectModo =0;
int ContadorEnvio =0;
```

```
String Str_datos ="";
//Ticker btn_timer;
void setup() {

    Serial.begin(115200);
    delay(200);

    pinMode(LED, OUTPUT);
    pinMode(RELAY, OUTPUT);
    pinMode(BUTTON, INPUT);

    digitalWrite(LED, LOW);
    digitalWrite(RELAY, LOW);
    power_read.enableMeasurePower();
    power_read.selectMeasureCurrentOrVoltage(VOLTAGE);
    power_read.startMeasure();

    ContadorEnvio=0;

    for(uint8_t t = 4; t > 0; t--)
    {
        Serial.printf("[SETUP] WAIT %d...\n", t);
        Serial.flush();
        digitalWrite(LED, HIGH);delay(500);
        digitalWrite(LED, LOW);delay(500);
    }
    digitalWrite(LED, LOW);

    WiFiMulti.addAP("TesisCris", "cris1234");
```

```
Serial.println("");
Serial.println("WiFi Conectado");
Serial.println("Direccion IP: ");
Serial.println(WiFi.localIP());

}

void loop()
{
  EstadoPulsador = digitalRead(BUTTON);
  if (EstadoPulsador == LOW)
  {
    SelectModo ++;
    ContadorEnvio=0;
    if (SelectModo ==1){

      digitalWrite(LED, HIGH);
      digitalWrite(RELAY, HIGH);
    }

    if (SelectModo ==2){

      digitalWrite(LED, LOW);
      digitalWrite(RELAY, LOW);
    }

    if (SelectModo >3){
      SelectModo=0;
      digitalWrite(LED, LOW);
      digitalWrite(RELAY, LOW);
    }
  }
}
```

```

    delay(1000);
}

/*.....*/
/*.....*/
/*.....*/

if ( ContadorEnvio == 0)
{
    // wait for WiFi connection
    if((WiFiMulti.run() == WL_CONNECTED)) {

        //-----
        getPower();

        //-----

        VA_pot= VA_pot/1000; //Paso de Watts a Kwss
        VA_khr = Seg10_Hora * VA_pot; // Kwatts * hora
        String Pot10Seg = String(VA_khr,8); //Recupero la potencia en 10
SEG
        //Serial.println("kh/h: ");Serial.println(Pot10Seg);
        VA_pot= VA_pot*1000;//RECUPERO el valor original de la potencia

        Serial.print("PotenciaSTR: ");Serial.println(VA_pot);
        Serial.print("VoltajeSTR: ");Serial.println(VA_vol);

        Str_datos = "";
        Str_datos = "http://192.168.16.82/DataDetectEsp.php?Vpot=";

```

```

Str_datos = Str_datos + String(VA_pot)+ "&Vvol=";
Str_datos = Str_datos + String(VA_vol)+ "&Vkhr=";
Str_datos = Str_datos + String(Pot10Seg);

//Serial.println(Str_datos);

HTTPClient http;
Serial.println("[HTTP] Iniciando.....\n");
http.begin(Str_datos);

Serial.println("[HTTP] Recibiendo Respuesta ...\n");
// start connection and send HTTP header
int httpCode = http.GET();

// httpCode will be negative on error
if(httpCode > 0) {
    // HTTP header has been send and Server response header has
    been handled
    Serial.printf("[HTTP] Respuesta... CODIGO: %d\n", httpCode);

    // file found at server
    if(httpCode == HTTP_CODE_OK) {
        String payload = http.getString();
        Serial.println(payload);
    }
} else {
    Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
}
http.end();
}

```

```
}

ContadorEnvio ++;
delay(1);
if (ContadorEnvio > 10000){ContadorEnvio=0;}
/*.....*/
/*.....*/
/*.....*/

}

void getPower()
{
  VA_pot = power_read.getPower();
  yield();
  Serial.print("Potencia: ");
  Serial.println(VA_pot);

  VA_vol = power_read.getVoltage();
  yield();
  Serial.print("Voltaje: ");
  Serial.println(VA_vol);
}
```

ANEXO 4

Código Fuente HTML de la interfaz web de Raspberry Pi 3 “index.php”

```
<html>
<head>
<title>Ajax divs</title>
<script language="Javascript" type="text/javascript">
function refreshDivs(divid,secs,url)
{

// define our vars
var divid,secs,url,fetch_unix_timestamp;

// Chequeamos que las variables no esten vacias..
if(divid == ""){ alert('Error: escribe el id del div que quieres refrescar'); return;}
else if(!document.getElementById(divid)){ alert('Error: el Div ID seleccionado no
esta definido: '+divid); return;}
else if(secs == ""){ alert('Error: indica la cantidad de segundos que quieres que
el div se refresque'); return;}
else if(url == ""){ alert('Error: la URL del documento que quieres cargar en el div
no puede estar vacia. '); return;}

// The XMLHttpRequest object

var xmlhttp;
try{
xmlhttp=new XMLHttpRequest(); // Firefox, Opera 8.0+, Safari
}
catch (e){
try{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP"); // Internet Explorer
```

```

}
catch (e){
try{
xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
}
catch (e){
alert("Tu explorador no soporta AJAX.");
return false;
}
}
}

// Timestamp para evitar que se cachee el array GET

fetch_unix_timestamp = function()
{
return parseInt(new Date().getTime().toString().substring(0, 10))
}

var timestamp = fetch_unix_timestamp();
var nocacheurl = url+"?t="+timestamp;

// the ajax call
xmlHttp.onreadystatechange=function(){
if(xmlHttp.readyState == 4 && xmlHttp.status == 200){
document.getElementById(divid).innerHTML=xmlHttp.responseText;
setTimeout(function(){refreshDivs(divid,secs,url);},secs*1000);
}
}
}

```

```
xmlHttp.open("GET",nocacheurl,true);
xmlHttp.send(null);
}
```

```
// LLamamos las funciones con los repectivos parametros de los DIVs que
queremos refrescar.
```

```
window.onload = function startrefresh(){
```

```
refreshDivs('div1',5,'LeerValEsp.php');
```

```
}
```

```
</script>
```

```
<script src="highcharts.js"></script>
```

```
<script src="series-label.js"></script>
```

```
<script src="exporting.js"></script>
```

```
</head>
```

```
<body>
```

```
<div id="div1"></div>
```

```
<div id="container" style="min-width: 310px; height: 400px; margin: 0
auto"></div>
```

```
<form action="/EncerarEsp.php" method="get" target="_blank">
```

```
<button type="submit">REINICIAR MEDICION</button>
```

```
</form>
```

```
<script type="text/javascript">
```

```
// Data retrieved from  
http://vikjavev.no/ver/index.php?spenn=2d&sluttid=16.06.2015.
```

```
Highcharts.chart('container', {  
  chart: {  
    type: 'spline'  
  },  
  title: {  
    text: 'Potencia Consumida del Artefacto Electrico'  
  },  
  subtitle: {  
    text: ''  
  },  
  xAxis: {  
    type: 'datetime',  
    labels: {  
      overflow: 'justify'  
    }  
  },  
  yAxis: {  
    title: {  
      text: 'Potencia (w/s)'  
    },  
    minorGridLineWidth: 0,  
    gridLineWidth: 0,  
    alternateGridColor: null,  
    plotBands: [],  
    tooltip: {  
      valueSuffix: ' Watios'  
    }  
  },  
  },
```

```

plotOptions: {
  spline: {
    lineWidth: 4,
    states: {
      hover: {
        lineWidth: 5
      }
    },
    marker: {
      enabled: false
    },
    pointInterval: 3600, // one hour
    pointStart: Date.UTC(<?php $fp = fopen("FechayHora.txt",
"r");while(!feof($fp)) {$linea = fgets($fp);echo $linea;}fclose($fp); ?>
//Date.UTC(año,mes[, dia[, hora[, minutos[, segundos, milisegundos]]]))
  }
},
series: [{
  name: 'Potencia',
  data: [
    <?php
      $fp = fopen("DatosGrafico.txt", "r");
      while(!feof($fp)) {
        $linea = fgets($fp);
        echo $linea;
      }
      fclose($fp);
    ?>
  ]
}],

```

```
navigation: {  
  menuItemStyle: {  
    fontSize: '10px'  
  }  
}  
});
```

```
</script>
```

```
</body>
```

```
</html>
```

ANEXO 5

Código Fuente de la librería Power.cpp

```
#include "power.h"
#include <Arduino.h>

uint32_t ESP8266PowerClass::power_freq_cnt = 0;
uint32_t ESP8266PowerClass::current_freq_cnt = 0;
uint32_t ESP8266PowerClass::voltage_freq_cnt = 0;
uint8_t ESP8266PowerClass::power_test_mode =
IOTGO_HLW8012_TEST_MODE_REPEAT;
uint8_t ESP8266PowerClass::current_test_mode =
IOTGO_HLW8012_TEST_MODE_REPEAT;
uint8_t ESP8266PowerClass::voltage_test_mode =
IOTGO_HLW8012_TEST_MODE_REPEAT;
bool ESP8266PowerClass::power_flag = false;
bool ESP8266PowerClass::current_flag = false;
bool ESP8266PowerClass::voltage_flag = false;
uint32_t ESP8266PowerClass::power_freq= 0;
uint32_t ESP8266PowerClass::voltage_freq= 0;
uint32_t ESP8266PowerClass::current_freq= 0;

void ESP8266PowerClass::measurePowerFreq(void)
{
    power_freq_cnt++;
}

void ESP8266PowerClass::measureCurrenFreq(void) // counts number of
interrupts CF1 pin
{
    current_freq_cnt++;
    if(voltage_freq_cnt != 0)
    {
```

```

        voltage_freq_cnt = 0;
    }
}

void ESP8266PowerClass::measureVoltageFreq(void) // counts number of
interrups CF1 pin
{
    voltage_freq_cnt++;
    if(current_freq_cnt != 0)
    {
        current_freq_cnt = 0;
    }
}

void ESP8266PowerClass::getAverageFreq(uint32_t
(&history_queue)[10],uint32_t &freq)
{
    uint8_t i = 0;
    for (i = 9; i >= 1; i--)
    {
        history_queue[i] = history_queue[i - 1];
    }
    history_queue[0] = freq;
    for (i = 1; i < 10; i++)
    {
        freq += history_queue[i];
    }
    freq = (uint32_t)((freq*1.0)/i + 0.5);
}

void ESP8266PowerClass::getFreq(uint32_t &cnt,uint32_t &freq_cnt,uint8_t
&test_mode,uint32_t &freq,

```

```

        uint8_t &hlw8012_bad_once_cnt,uint8_t
&hlw8012_bad_repeat_cnt,uint32_t (&history_queue)[10])
{

if (IOTGO_HLW8012_TEST_MODE_REPEAT == test_mode)
{
    if (cnt >= 1000)
    {
        if (freq_cnt > 10)
        {
            hlw8012_bad_repeat_cnt = 0;
            freq = freq_cnt * 100;
            getAverageFreq(history_queue,freq);
        }
        else
        {
            hlw8012_bad_repeat_cnt++;
            if (hlw8012_bad_repeat_cnt >= 2)
            {
                hlw8012_bad_repeat_cnt = 0;
                test_mode = IOTGO_HLW8012_TEST_MODE_ONCE;
            }
        }
        freq_cnt = 0;
        cnt = 0;

    }
}

else if (IOTGO_HLW8012_TEST_MODE_ONCE == test_mode)
{

```

```
if (1 == freq_cnt)
{
    if (cnt >= 100)
    {
        hlw8012_bad_once_cnt = 0;
        freq = (uint32_t)(100000.0/(cnt) + 0.5);
        getAverageFreq(history_queue,freq);
    }
    else
    {
        hlw8012_bad_once_cnt++;
        if (hlw8012_bad_once_cnt >= 2)
        {
            hlw8012_bad_once_cnt = 0;
            test_mode = IOTGO_HLW8012_TEST_MODE_REPEAT;
        }
    }
    cnt = 0;
    freq_cnt = 0;
}
```

```
if (cnt >= (10*1000 + 1000))
{
    test_mode = IOTGO_HLW8012_TEST_MODE_REPEAT;
    cnt = 0;
    freq_cnt = 0;
    freq = 0;
    os_memset(history_queue,0,10);
}
```

```
    }  
  }  
}
```

```
void ESP8266PowerClass::timerCallback(void)
```

```
{  
  static uint32_t power_1ms_cnt = 0;  
  static uint32_t curent_1ms_cnt = 0;  
  static uint32_t voltage_1ms_cnt = 0;  
  static uint8_t power_bad_once_cnt = 0;  
  static uint8_t power_bad_repeat_cnt = 0;  
  static uint8_t current_bad_once_cnt = 0;  
  static uint8_t current_bad_repeat_cnt = 0;  
  static uint8_t voltage_bad_once_cnt = 0;  
  static uint8_t voltage_bad_repeat_cnt = 0;  
  static uint32_t power_history_queue[10] = {0};  
  static uint32_t current_history_queue[10] = {0};  
  static uint32_t voltage_history_queue[10] = {0};  
  power_1ms_cnt++;  
  curent_1ms_cnt++;  
  voltage_1ms_cnt++;  
  if(power_flag)  
  {  
    getFreq(power_1ms_cnt,power_freq_cnt,power_test_mode,power_freq,  
  
power_bad_once_cnt,power_bad_repeat_cnt,power_history_queue); // fills all  
these variables with numbers, either continuous or once  
  }  
  if(current_flag)
```

```

    {
        getFreq(curent_1ms_cnt,current_freq_cnt,current_test_mode,current_freq,

current_bad_once_cnt,current_bad_repeat_cnt,current_history_queue);
    }
    if(voltage_flag)
    {

getFreq(voltage_1ms_cnt,voltage_freq_cnt,voltage_test_mode,voltage_freq,

voltage_bad_once_cnt,voltage_bad_repeat_cnt,voltage_history_queue);

    }
}

```

```

ESP8266PowerClass::ESP8266PowerClass() // Constructor

```

```

{
    this->power_param = {12.286,0};
    this->current_param = {14.818,0};
    this->voltage_param = {0.435,0};

    this->power_pin = 14; // CF
    this->current_voltage_pin = 13; //CF1
    this->sel_pin = 5; // Chip select
}

```

```

ESP8266PowerClass::ESP8266PowerClass(int8_t power_pin,int8_t
current_voltage_pin,int8_t sel_pin)

```

```

{
    ESP8266PowerClass();
    this->power_pin = power_pin;
    this->current_voltage_pin = current_voltage_pin;
}

```

```

    this->sel_pin = sel_pin;
}

void ESP8266PowerClass::enableMeasurePower(void)
{
    this->power_flag = true;
    pinMode(this->power_pin,INPUT_PULLUP);
    attachInterrupt(this->power_pin,measurePowerFreq,FALLING); // interrupt
on CF
}

void ESP8266PowerClass::selectMeasureCurrentOrVoltage(MEASURETYPE
dev_type)
{
    #if 1
        pinMode(this->sel_pin,OUTPUT); // CF1
        detachInterrupt(this->current_voltage_pin);
        voltage_freq = 0;
        current_freq = 0;
        pinMode(this->current_voltage_pin,INPUT_PULLUP);
        switch(dev_type)
        {
            case CURRENT:
                {
                    this->current_flag = true;
                    this->voltage_flag = false;
                    digitalWrite(this->sel_pin,HIGH);
                    attachInterrupt(this-
>current_voltage_pin,measureCurrenFreq,FALLING);
                }break;

```

```

    case VOLTAGE:
    {
        this->current_flag = false;
        this->voltage_flag = true;
        digitalWrite(this->sel_pin,LOW);
        attachInterrupt(this-
>current_voltage_pin,measureVoltageFreq,FALLING);
    }break;
    default:
    {
        /*
            do nothing
        */
    }
}
#endif
}

void ESP8266PowerClass::setPowerParam(double param_a,double param_b)
{
    this->power_param.param_a = param_a;
    this->power_param.param_b = param_b;
}

void ESP8266PowerClass::setCurrentParam(double param_a,double param_b)
{
    this->current_param.param_a = param_a;
    this->current_param.param_b = param_b;
}

void ESP8266PowerClass::setVoltageParam(double param_a,double param_b)

```

```
{
    this->voltage_param.param_a = param_a;
    this->voltage_param.param_b = param_b;
}
```

```
DEVPARAM ESP8266PowerClass::getPowerParam(void)
```

```
{
    return power_param;
}
```

```
DEVPARAM ESP8266PowerClass::getCurrentParam(void)
```

```
{
    return current_param;
}
```

```
DEVPARAM ESP8266PowerClass::getvoltageParam(void)
```

```
{
    return voltage_param;
}
```

```
void ESP8266PowerClass::startMeasure(void)
```

```
{
    os_timer_setfn(&myTimer, (os_timer_func_t
*)&ESP8266PowerClass::timerCallback, NULL);
    os_timer_arm(&myTimer, 1, true); // every ms
}
```

```
double ESP8266PowerClass::getPower(void)
```

```
{
    uint32_t power = 0;
    power = power_param.param_a * power_freq + power_param.param_b *
100;
```

```
        return power / 100.0;
    }

double ESP8266PowerClass::getCurrent(void)
{
    double current = (current_param.param_a * current_freq +
current_param.param_b*100.0);

    return current/100.0;
}

double ESP8266PowerClass::getCurrFrequency(void)
{
    uint32_t frequency = 0;
    frequency = current_freq ;
    return frequency;
}

double ESP8266PowerClass::getVoltage(void)
{
    uint32_t voltage = 0;
    voltage = voltage_param.param_a * voltage_freq + voltage_param.param_b *
100;
    return voltage / 100.0;
}
```

ANEXO 6

Código Fuente de la Librería power.h

```
#ifndef __POWER__H_
#define __POWER__H_

#include <functional>

extern "C" {
#include "user_interface.h"
}

typedef struct _DEVPARAM{
    double param_a;
    double param_b;
}DEVPARAM;

typedef enum{
    CURRENT,
    VOLTAGE,
}MEASURETYPE;

typedef enum
{
    IOTGO_HLW8012_TEST_MODE_REPEAT,
    IOTGO_HLW8012_TEST_MODE_ONCE,
}TESTMODE;

class ESP8266PowerClass{
public:
```

```

    ESP8266PowerClass();
    ESP8266PowerClass(int8_t power_pin,int8_t current_voltage_pin,int8_t
sel_pin);
    void enableMeasurePower(void);
    void selectMeasureCurrentOrVoltage(MEASURETYPE dev_type);
    void setPowerParam(double param_a,double param_b);
    void setCurrentParam(double param_a,double param_b);
    void setVoltageParam(double param_a,double param_b);
    DEVPARAM getPowerParam(void);
    DEVPARAM getCurrentParam(void);
    DEVPARAM getvoltageParam(void);
    void startMeasure(void);
    double getPower(void);
    double getCurrent(void);
    double getVoltage(void);
    double getCurrFrequency(void);
private:
    static void measurePowerFreq(void);
    static void measureCurrenFreq(void);
    static void measureVoltageFreq(void);
    static void timerCallback(void);
    static void getFreq(uint32_t &cnt,uint32_t &freq_cnt,uint8_t
&test_mode,uint32_t &freq,
                        uint8_t &hlw8012_bad_once_cnt,uint8_t
&hlw8012_bad_repeat_cnt,uint32_t (&history_queue)[10]);
    static void getAverageFreq(uint32_t (&history_queue)[10],uint32_t &freq);
    static uint32_t power_freq_cnt;
    static uint32_t voltage_freq_cnt;
    static uint32_t current_freq_cnt;
    static uint8_t power_test_mode;

```

```
static uint8_t current_test_mode;
static uint8_t voltage_test_mode;
static bool power_flag;
static bool current_flag;
static bool voltage_flag;
static uint32_t power_freq;
static uint32_t voltage_freq;
static uint32_t current_freq;
DEVPARAM power_param;
DEVPARAM current_param;
DEVPARAM voltage_param;
int8_t power_pin;
int8_t current_voltage_pin;
int8_t sel_pin;

os_timer_t myTimer;

};

#endif
```

ANEXO 7

Código de Programación Archivo "DataDetectEsp.php"

```
<?php

$VA_pot = $_GET['Vpot'];
$VA_vol = $_GET['Vvol'];
$VA_khr = $_GET['Vkhr'];

//Leemos el JSON
$datos_file_php= file_get_contents("EspValln.json");
$json_EspVal = json_decode($datos_file_php, true);

/* //VALORES IGUALADOS A VARIABLES PHP
$Pot = $json_EspVal["Pot"];
$Vol = $json_EspVal["Vol"];
$KsH = $json_EspVal["KsH"];
*/

// Modifica el valor, y escribe el fichero json de salida
$Pot = $VA_pot;
$Vol = $VA_vol;
$KsH = $json_EspVal["KsH"]+ $VA_khr;

$json_EspVal["Pot"] = $Pot;
$json_EspVal["Vol"] = $Vol;
$json_EspVal["KsH"] = $KsH;

// Escribimos una primera línea en fichero.txt
```

// fichero.txt tienen que estar en la misma carpeta que el fichero php

```
$nombre_archivo = "DatosGrafico.txt";
```

```
if(file_exists($nombre_archivo))
```

```
{
```

```
    $mensaje = $Pot;
```

```
}
```

```
else
```

```
{
```

```
    $mensaje = $Pot;
```

```
}
```

```
if($archivo = fopen($nombre_archivo, "a"))
```

```
{
```

```
    if(fwrite($archivo, $mensaje))
```

```
    {
```

```
        echo "Se ha ejecutado correctamente";
```

```
    }
```

```
    else
```

```
    {
```

```
        echo "Ha habido un problema al crear el archivo";
```

```
    }
```

```
    fclose($archivo);
```

```
}
```

```
$fh = fopen("EspValln.json", 'w') or die("Error al abrir fichero de salida");  
fwrite($fh, json_encode($json_EspVal,JSON_UNESCAPED_UNICODE));  
fclose($fh);
```

```
echo " exito";
```

```
?>
```

ANEXO 8

Código de Programación Archivo "EncerarEsp.php"

```
<?php
```

```
$nombre_archivo = "FechayHora.txt";
```

```
if(file_exists($nombre_archivo))
```

```
{
```

```
    $mensaje = "El Archivo $nombre_archivo se ha modificado";
```

```
}
```

```
else
```

```
{
```

```
    $mensaje = "El Archivo $nombre_archivo se ha creado";
```

```
}
```

```
//GUARDA LOS DATOS DE HORA Y FECHA PARA PONER EN EL  
WIGGET
```

```
if($archivo = fopen($nombre_archivo, "w"))
```

```
{
```

```
    //pointStart: Date.UTC(2017, 10, 6, 7, 35, 0)
```

```
    //Date.UTC(año,mes[, dia[, hora[, minutos[, segundos, milisegundos]]]])
```

```
    //d m Y H:m:s
```

```
    $hoy = getdate();
```

```
$horaLocal = $hoy[hours] -5;
```

```
if(fwrite($archivo, $hoy[year]. " ".$hoy[mon]. " ".$hoy[mday]. " ".$horaLocal.  
" ".$hoy[minutes]. " ".$hoy[seconds]) )
```

```
{  
    echo "1 Se ha ejecutado correctamente";  
}
```

```
else
```

```
{  
    echo "1 Ha habido un problema al crear el archivo";  
}
```

```
fclose($archivo);
```

```
}
```

```
//ENCERA LOS DATOS DEL ARCHIVO TXT
```

```
$nombre_archivo2 = "DatosGrafico.txt";
```

```
if(file_exists($nombre_archivo2))
```

```
{  
    $mensaje2 = "El Archivo $nombre_archivo se ha modificado";  
}
```

```
else
```

```
{  
    $mensaje2 = "El Archivo $nombre_archivo se ha creado";  
}
```

```
if($archivo2 = fopen($nombre_archivo2, "w"))
```

```
{
```

```
if(fwrite($archivo2, "" )
{
    echo "2 Se ha ejecutado correctamente";
}
else
{
    echo "2 Ha habido un problema al crear el archivo";
}

fclose($archivo2);
}
```

```
//Leemos el JSON
```

```
$datos_file_php= file_get_contents("EspValln.json");
$json_EspVal = json_decode($datos_file_php, true);
```

```
$json_EspVal["Pot"] = 0.000000;
$json_EspVal["Vol"] = 0.000000;
$json_EspVal["KsH"] = 0.000000;
```

```
$fh = fopen("EspValln.json", 'w') or die("Error al abrir fichero de salida");
fwrite($fh, json_encode($json_EspVal,JSON_UNESCAPED_UNICODE));
fclose($fh);
```

```
echo " exito";
echo "<script lenguaje='javascript'
type='text/javascript'>window.close();</script>";
?>
```

ANEXO 9

Código de Programación Archivo "LeerValEsp.php"

```
<?php

//Leemos el JSON
$datos_file_php= file_get_contents("EspValln.json");
$json_EspVal = json_decode($datos_file_php, true);

//VALORES IGUALADOS A VARIABLES PHP
$Pot = $json_EspVal["Pot"];
$Vol = $json_EspVal["Vol"];
$KsH = $json_EspVal["KsH"];

echo "<p>POTENCIA: ".$Pot." Watts. </p>";
echo "<p>VOLTAJE: ".$Vol." Voltios. </p>";
echo "<p>CONSUMO: ".$KsH." Kw/h. </p>";

?>
```

