ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UN MEDIDOR ELECTRÓNICO DE ENERGÍA CON ENVÍO DE DATOS A INTERNET

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES

ALEJANDRO DAVID CHAMORRO CAPELO

alejandro.chamorro@epn.edu.ec

DIRECTOR: ING. PABLO ANDRÉS PROAÑO CHAMORRO

pablo.proano@epn.edu.ec

CODIRECTOR: ING. FABIO MATÍAS GONZÁLEZ GONZÁLEZ MSC.

fabio.gonzalez@epn.edu.ec

Noviembre, 2019

DECLARACIÓN

Yo, Alejandro David Chamorro Capelo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 114 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación -COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional. Entregaré toda la información técnica pertinente. En el caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

_____

Alejandro David Chamorro Capelo

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por Alejandro David Chamorro Capelo, bajo nuestra supervisión.

Ing. Pablo Andrés Proaño Chamorro	**Ing. Fabio Matías González González**
DIRECTOR DEL PROYECTO	**CODIRECTOR DEL PROYECTO**

AGRADECIMIENTO

Con mucho amor a mi madre, mi padre y mi hermano.

DEDICATORIA

A Hilda.

INDICE

INDICE DE FIGURAS

INDICE DE TABLAS.

RESUMEN

El presente proyecto detalla el diseño y construcción de un prototipo electrónico para la medición del consumo de energía eléctrica y la posterior consulta de la información del abonado desde cualquier lugar en el internet. El dispositivo está especialmente pensado para su instalación en el sector residencial y se esperaría que su implementación masiva colabore en la tarea de recopilar información de los clientes del sistema eléctrico. El proyecto está dividido en cuatro capítulos, los cuales se describen a continuación:

En el primer capítulo se presenta una corta introducción, el planteamiento del problema, el objetivo general, los objetivos específicos y la justificación del proyecto, finalmente, se realiza una breve descripción de los conceptos generales involucrados en el desarrollo del presente proyecto.

En el segundo capítulo se describe la metodología empleada para el cumplimiento de los objetivos planteados, a través de la investigación desarrollada en torno a los sistemas ya existentes en el mercado se hace una descripción general de las características que poseerá el prototipo.

El tercer capítulo presenta los resultados y discusiones. Se describe la elaboración de cada uno de los subsistemas que posee el prototipo, la fabricación de las placas de circuito impreso, el diseño de la parte lógica, la programación realizada y el ensamblaje de todas las partes individuales en un solo módulo compacto, funcional y de fácil transporte. Además, se realizaron los ensayos y pruebas pertinentes con el fin de corregir errores y verificar el correcto funcionamiento del prototipo.

En el cuarto capítulo se listan las conclusiones y recomendaciones conseguidas luego de finalizar la elaboración del proyecto y realizar las pruebas de funcionamiento pertinentes.

1. INTRODUCCIÓN

1.1 Planteamiento del Problema

Según las estadísticas de la Agencia de Regulación y Control de Electricidad hasta el 2017 el número total de clientes regulados fue 5.071.523, de los cuales 5.055.593 cuentan con medidores en todo el Ecuador (Agencia de Regulación y Control de Electricidad, 2017). Para el caso de la provincia de Pichincha, se contabilizaron un total de 1.112.876 unidades de las cuales 167740 corresponden a medidores monofásicos para el sector residencial. En el año 2011 se registró un total de 890.321 medidores en el Distrito Metropolitano (Consejo Nacional de Electricidad (CONELEC), 2011).

Con los datos anteriores se notó el crecimiento rápido de la demanda de energía eléctrica y por consiguiente el aumento de número de trabajadores que toman las medidas de consumo energético en cada uno de los medidores. La consecuencia directa en el aumento del número de medidores es el incremento de gastos económicos y la logística es más compleja. Para los empleados que recolectan las medidas se torna complicado y peligroso ingresar a localidades con altos niveles de delincuencia y criminalidad, haciéndose una actividad de alto riesgo, además llegar a zonas rurales de difícil acceso hace que su trabajo sea más pesado.

En diferentes ciudades del Ecuador, como Guayaquil, ya se ha instalado 104.000 medidores inteligentes con tecnología AMI (Infraestructura de Medición Avanzada) que disponen de capacidad de lectura, corte y reconexión remota (Corporación Nacional de Electricidad (CNEL EP), 2016). Ésta tecnología conecta la red de medidores, a través de transmisión inalámbrica, con un centro de cómputo que recepta datos, su aplicación proporcionar un adecuado y moderno servicio de distribución y comercialización para mejorar la eficiencia operativa. (Instituto Nacional de Electricidad y Energías Limpias de México (INEEL), 2015)

Por los motivos antes mencionados, se desarrolló de un prototipo electrónico de medición de energía eléctrica que sea económico y fácil de instalar en cualquier residencia, que a su vez utilice la red de comunicaciones de internet para el envío de datos y control remoto del dispositivo.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar de un medidor electrónico de energía con envío de datos a internet.

1.2.2 Objetivos Específicos

- Determinar los requerimientos del dispositivo a implementarse en base a las necesidades y exigencias del mercado.
- Implementar un circuito electrónico capaz de medir la potencia eléctrica en residencias.
- Comunicar el circuito de medición con una interfaz externa usando un módulo Wi-Fi.
- Realizar las pruebas de funcionamiento y análisis de resultados.

1.3 Justificación

Al finalizar la construcción e implementación del prototipo se espera beneficiar a los estudiantes de la Escuela de Formación Tecnológica ya que este dispositivo ayudaría a incursionar en temas relacionados con sensores, actuadores, instrumentación, aplicaciones con microcontroladores y comunicaciones, inclusive una breve introducción al Internet de las cosas.

En un futuro cercano la Empresa Eléctrica podría aplicar el prototipo para mejorar su proceso de toma de mediciones. Se reducirá costos por la disminución de la mano de obra, proporcionará más producción con menos trabajadores, se mejorará la calidad del servicio y la eficiencia, aumentará la seguridad del recurso humano y se agilitará el proceso. (Rose, Eldridge, & Chapin, 2015).

Además, se dispuso de un dispositivo de bajo costo, fácil de implementar y cambiar de ubicación, la puesta en marcha será mucho menor a los dispositivos existentes en el mercado, su configuración será sencilla y en caso de requerir algún repuesto se lo podrá encontrar en el mercado con mucha facilidad. (Office of Electricity Delivery and Energy Reability, 2016).

1.4 Conceptos Generales

1.4.1 Módulo WiFi NodeMCU ESP 8266

Node Microcontroller Unit ESP 8266, es una tarjeta de desarrollo de código abierto orientada al diseño de proyectos de Internet de las Cosas. Está basado en el System on Chip ESP8266 que integra un procesador de 32 bits con conectividad WI-FI. Utiliza el lenguaje de programación Lua pero también se puede emplear el IDE de Arduino y sus respectivas librerías para su programación. Admite la programación directa desde el puerto USB. Combina las características de punto de acceso y estación más microcontrolador. Es posible alojar un servidor web y conectarse a internet para buscar o cargar datos. En la Figura 1.1 se observa el NMCU ESP 8266. (Espressif, 2015).

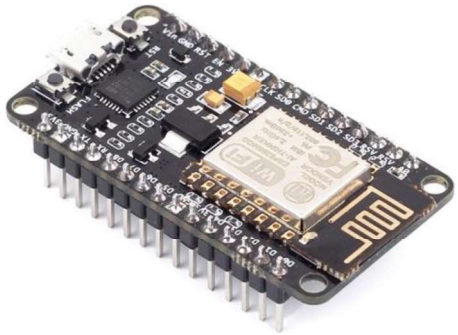

Figura 1.1. NodeMCU ESP 8266.
(Fuente: Future Electronics, 2018)

En la Tabla 1.1 se observan las características del módulo WiFi.

Procesador	Tensilica LX106 32 (bits) a 80 (MHz)
Memoria RAM	80 (KB)
Memoria Flash	4 (MB)
ROM	NO
Alimentación	3 (V_{DC}) – 3.6 (V_{DC})
Consumo de corriente	80 (mA_{DC})
WiFi	802.11 b/g/n
ADC	1

Tabla 1. 1. Características del módulo WiFi.
(Fuente: Autoría propia)

1.4.2 Infraestructura de Medición Avanzada

AMI (Advanced Metering Infrastructure) es un sistema integrado de medidores inteligentes (SM – Smart Meters), redes de comunicación y sistemas de administración de datos que habilita la comunicación de dos vías entre los servicios públicos y el consumidor final. El sistema proporciona una serie de funciones que antes no eran posibles o bebían realizarse manualmente, como la capacidad de medir de forma automática y remota el uso de la electricidad, conectar y desconectar el servicio, detectar alteraciones, etc. (Office of Electricity Delivery and Energy Reability, 2016).

En la Figura 1.2 se observa un esquema básico para la conexión de un sistema AMI.

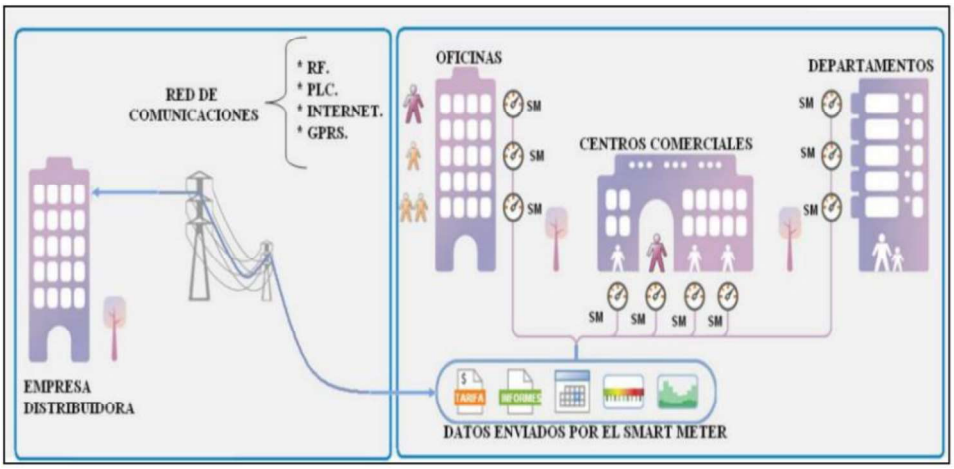

Figura 1.2. Esquema básico para la conexión de un sistema AMI.
(Fuente: Idrovo & Reinoso, 2012).

1.4.3 Medidor Inteligente

Es un dispositivo electrónico que registra el consumo de agua, electricidad o gas y envía esa información de forma automática hacia la compañía de servicios públicos para el monitoreo y facturación. (ETSI, 2018).

En la Figura 1.3 se observa un ejemplo de medidor inteligente comercial monofásico de 220 (V_{AC}) y 40 (A_{AC}) con comunicación wifi.

Figura 1.3. Medidor inteligente comercial.
(Fuente: XT Energy Meter Expert, 2019).

1.4.4 Servidor NTP

Network Time Protocol (NTP) es un protocolo de Internet (IP – Internet Protocol) estándar que ayuda a sincronizar en tiempo real la hora de los relojes de las computadoras en una red. Las marcas de tiempo se transmiten y reciben utilizando el Protocolo de Datagrama de Usuario (UDP) en el puerto 123.

La configuración más común en operar en modo unicast o cliente-servidor. El cliente transmite un paquete de solicitud al servidor y éste responde con un paquete de marca de tiempo. Cada paquete recibido tiene marcas de tiempo origen, recepción y transmisión para calcular los retrasos de propagación a través de la red, lo que permite que los clientes se sincronicen lo más estrechamente posible con el reloj del servidor. Todos los paquetes de marca de tiempo generados por el servidor NTP utilizan la hora UTC (Tiempo Universal Coordinado) que es un estándar de tiempo mundial y está relacionado con GTM (Greenwich Mean Time). UTC no varía, es el mismo alrededor de todo el mundo. El servicio NTP establece los relojes de las computadoras en UTC, y es el sistema operativo el que aplica cualquier compensación de zona horaria local, de esta forma, los clientes se sincronizan con los servidores independientemente de su ubicación y las diferencias de zona horaria. (Time Tools , 2018)

1.4.5 Formato JSON

JSON (JavaScript Object Notation) es un formato para el intercambio de datos. Los programadores lo pueden leer y escribir de forma relativamente rápida y las máquinas lo analizan y generan de forma rápida. Su sintaxis se basa en un subconjunto de funcionalidades de JavaScript. (Aprende a Programar, 2014)

La estructura de JSON puede estar constituida de las siguientes formas:

- Estructura nombre-valor o colecciones de parejas, que en diferentes lenguajes de programación se las conoce como registros, diccionarios, tablas, hash, arreglos asociativos o listas de claves. En la Figura 1.4 se observa un ejemplo del formato JSON en estructura nombre-valor.

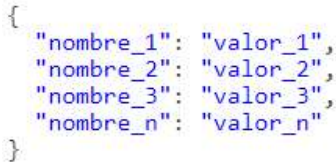

Figura 1.4. Formato JSON con estructura nombre-valor.
(Fuente: Autoría propia)

- Lista ordenada de valores, en varios lenguajes de programación se las conoce como vectores o arreglos. En la Figura 1.5 se observa una lista ordenada de valores en formato JSON.

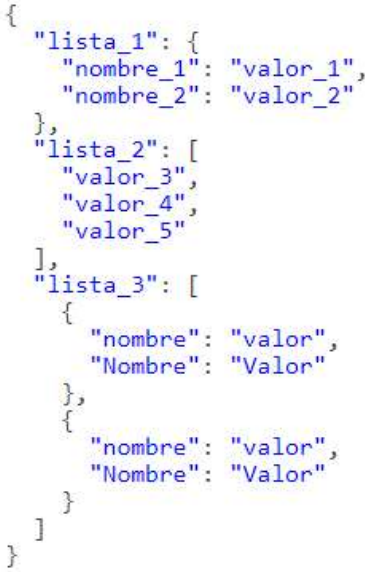

Figura 1.5. Formato JSON en lista ordenada de elementos.
(Fuente: Autoría propia)

2. METODOLOGÍA

2.1 Descripción de la Metodología

Antes de iniciar el desarrollo del proyecto se tuvo que determinar la forma en que se debían adquirir los datos de voltaje y corriente, necesarios para calcular la energía consumida, así como el factor de potencia. Para lo cual se estudió los diferentes sensores que se ofertan en el mercado y luego de su respectivo análisis se escogió los más adecuados para el desarrollo del proyecto.

Fue necesario diseñar los circuitos para la adquisición de datos de los sensores y ajustar sus señales de tal forma que sean reconocidas por el microcontrolador. Para la medición de la corriente se escogió un transformador de corriente (CTs) tipo pinza y para la medición del voltaje se empleó un transformador de voltaje común y corriente.

Se desarrolló un algoritmo capaz de interpretar las señales provenientes de las placas de adquisición de datos y se obtuvo como resultado el valor medio cuadrado para el voltaje y corriente, además, el factor de potencia, potencia activa, potencia reactiva y potencia aparente instantáneas. Para realizar esta tarea se optó por la placa Arduino Pro Mini, ya que su tamaño reducido permite su fácil implementación en la estructura.

A continuación, se envió la información de forma serial hacia el módulo wifi ESP8266 para proceder al cálculo de la energía consumida.

Además, en el módulo wifi ESP8266 fue necesario levantar un servidor web para transmitir los datos de consumo energético hacia el internet y se configuró el modem del ISP (Internet Service Provider – Proveedor del Servicio de Internet) de tal manera que el servidor web sea accesible desde cualquier lugar del internet. También se desarrolló una interfaz gráfica, creando una aplicación web para desplegar la información del consumo energético del abonado con la opción de suprimir el servicio en caso de ser necesario, para lo cual se dispone de un módulo relé encargado de realizar esta tarea.

Fue obligatorio configurar los parámetros de red, así como el usuario y contraseña para la conexión wifi del dispositivo. También se levantó un servidor NTP (Network Time Protocol – Protocolo de Tiempo de Red) con la finalidad de sincronizar la hora del dispositivo con la red.

Luego, se diseñaron y fabricaron las placas de circuito impreso para las fuentes de alimentación y las tarjetas de adquisición de datos.

Finalmente, tomando en cuenta las dimensiones de las tarjetas y los elementos electrónicos (fuentes de alimentación, placas de adquisición de datos, sensores, microcontroladores, LCD) se desarrolló la integración de los mismos en una sola estructura, de tal forma que se manipule el prototipo de forma segura y su instalación sea rápida.

Para realizar las pruebas de funcionamiento se dividió en dos partes, pruebas del circuito de potencia y pruebas del circuito de comunicaciones.

En las pruebas del circuito de potencia se conectaron diversas cargas al dispositivo, inductivas y resistivas, con el fin de contrastar los valores reales con los obtenidos por el dispositivo y posteriormente corregir los errores en los resultados de potencia y energía.

En las pruebas de comunicaciones, luego de haber conectado el circuito y las cargas de prueba, se procedió a acceder al servidor web desde fuera de la red y comprobar que los valores desplegados en la interfaz gráfica concuerdan con los valores desplegados en el LCD del módulo. Finalmente, se corrigió los errores presentados en la comunicación con el internet.

3. RESULTADOS Y DISCUCIONES

A continuación, se describe el diseño y construcción del prototipo electrónico para la medición del consumo de energía eléctrica y la posterior consulta de la información del abonado desde cualquier el internet. Se diseñó el sistema para desplegar una interfaz web cada vez que el administrador de servicio eléctrico desea conocer la información de un cliente, brindando además la capacidad de suspender el servicio si se presentase un posible incumplimiento en el pago de la factura de servicio.

En la Figura 3.1 se observa el dispositivo ya finalizado, el cual está listo para ser instalado en cualquier residencia.

Figura 3.1. Módulo finalizado.
(Fuente: Autoría propia)

En la Figura 3.2 se observa la estructura interna del prototipo y se describe cada una de las partes que lo constituyen.

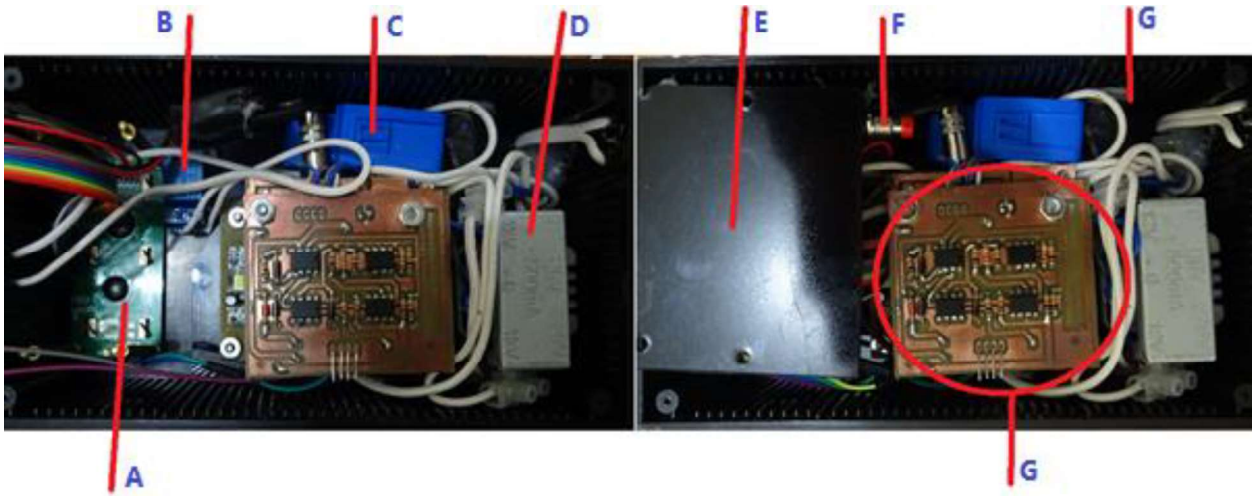

Figura 3.2. Disposición final de los elementos en la caja.
(Fuente: Autoría propia)

A. LCD.
B. Módulo relé.
C. Transformador de corriente.
D. Transformador de voltaje.
E. Módulo controlador – ESP8266.
F. Pulsador para la configuración inicial.
G. Fusible de protección.
H. Módulo PCBs.

3.1 Requerimientos Para el Diseño.

El módulo debe cubrir ciertas características y especificaciones básicas para que tenga una adecuada funcionalidad. A continuación, se describen los requerimientos básicos del módulo.

3.1.1 Requerimientos Técnicos.

Se pretende la instalación del prototipo en el sector residencial, por tal motivo, éste deberá ser capaz de medir voltajes de 120 (V_{AC}) y corrientes de hasta 20 (A_{AC}). Se deberá conectar mediante wifi a la red de área local, por lo tanto, debe cumplir las especificaciones del estándar 802.11 las cuales son: banda de frecuencias de 2.4 (GHz) y velocidad de transmisión máxima 11 (Mbps). Alcance de 10 (m).

3.1.2 Modular.

Los productos o proyectos se subdividen en partes más simples en el marco de una estructura modular. El establecimiento de esta estructura requiere criterios para repartir las funciones y establecer interfaces entre los módulos que transmitan adecuadamente la información entre ellos.

El prototipo se dividió en segmentos, cada uno de los cuales realizarán ula tarea de recolectar información, procesar la información y transmitir los datos, de esta forma se reconocerá rápidamente el lugar donde se genera una falla y en caso de ser necesario se cambiará la pieza o placa para su reparación o sustitución.

3.1.3 Confiable y Seguro.

Para un producto, es su capacidad de realizar una función requerida en condiciones establecidas durante un intervalo de tiempo fijo.

El medidor de energía a desarrollar deberá ser capaz de trabajar ininterrumpidamente durante períodos prolongados de tiempo sin presentar fallos, para de esta forma reducir tanto el número de ciclos de mantenimiento como las reparaciones, esto se reflejará directamente en la disminución de costes destinados a la manutención del sistema.

3.1.4 Duración.

Los distintos módulos y partes que conforman el prototipo deberán disponerse en el interior de una estructura de plástico cerrada para proteger las piezas de las inclemencias del clima y sobre todo que brinde soporte y cree resistencia a los golpes y caídas que pudieran producirse durante su manipulación. Además, deberá contar con protección en el caso de sobre corrientes o cortocircuito. Así se pretende prolongar la vida útil del dispositivo.

3.1.5 Facilidad de Uso.

El prototipo deberá instalarse de forma rápida en cualquier residencia, la configuración para la conexión de la red wifi se la realizará en corto tiempo y no se necesitará de conocimientos especializados, la visualización de la información de

cada medidor será posible desde cualquier navegador web con acceso a internet, finalmente, se pretende que el mantenimiento, reparación y cambio de placas o piezas sea sencillo.

3.1.6 Requerimientos Adicionales.

Además de los requerimientos presentados antes se procurará desarrollar un prototipo que sea susceptible a mejoras tanto en la parte lógica como en la parte del hardware. Al igual que el mantenimiento, las actualizaciones serán sencillas y se las realizará en poco tiempo. Así mismo, el precio del producto final será bajo, la fabricación será sencilla, de una presentación estética y elegante y a la vez sencillo y minimalista.

3.2 Selección de los Dispositivos

A continuación, se listan los componentes electrónicos elegidos en base a los requerimientos para el diseño.

3.2.1 Sensor de Corriente

En la Figura 3.3 se muestra el transformador de corriente.

Figura 3.3. Transformador de corriente (CTs).
(Fuente: YHDC, 2019)

Se utilizó un transformador de corriente (CTs) no invasivo con núcleo de ferrita, muy fácil de manejar y acoplar. (YHDC, 2019).

Se escogió este tipo de transformador principalmente por su capacidad de reducir altas corrientes a valores de corriente pequeños, también por el aislamiento galvánico que ofrece, su pequeño tamaño y reducido valor.

En la Tabla 3.1 se describen las características del sensor de corriente.

Tabla 3.1. Características del sensor de corriente

MODELO	SCT-013-020
Corriente de Entrada Máxima	20 (A_{AC})
Voltaje de Salida Máximo	1 (V_{AC})
Resistencia	62 (Ω)
Conector	Plug de 2.5 (mm) – macho
Dimensiones	13 x 13 (mm)

(Fuente: Autoría propia)

3.2.2 Transformador de Voltaje.

Para la medición del voltaje de línea se escogió un transformador de 12 (V_{AC}) y 0.5 (A_{AC}). En la Figura 3.4 se muestra el transformador usado en la elaboración del proyecto.

Figura 3.4. Transformador 12 V_{AC}.
(Fuente: Grupo Velasco, 2019)

3.2.3 Módulo de fuente de alimentación

Es un módulo de alimentación con conversión AC-DC. Posee protección de temperatura, contra cortocircuito y sobre corriente además de aislamiento de bajo y alto voltaje. En la Figura 3.5 se observa el módulo de fuente de alimentación. (Itead Studio, 2019).

Figura 3.5. Módulo de fuente de alimentación.
(Fuente: Itead Studio, 2019).

En la Tabla 3.2 se presenta las características técnicas generales del módulo.

Tabla 3.2. Características generales del módulo fuente de alimentación.

	Versión 5V	**Versión 12V**
Voltaje de entrada alterno	85 (V_{AC}) ~ 265 (V_{AC})	85 (V_{AC}) ~ 265 (V_{AC})
Voltaje de entrada continuo	100 (V_{DC}) ~ 370 (V_{DC})	100 (V_{DC}) ~ 370 (V_{DC})
Corriente de Salida Pico	0.6 (A_{DC})	0.3 (A_{DC})
Potencia de Salida	3 (W)	3 (W)
Eficiencia	97 (%)	97 (%)
Precisión de Salida	± 3 (%V)	± 3 (%V)
Frecuencia de Conmutación	50 (KHz) ~ 60 (KHz)	50 (KHz) ~ 60 (KHz)
Temperatura de Trabajo	-30 (°C) ~ +70 (°C)	-30 (°C) ~ +70 (°C)
Dimensiones	30 (mm) x 20 (mm) x 18 (mm)	30 (mm) x 20 (mm) x 18 (mm)

(Fuente: ebay, 2019).

Con estos módulos se generará las tensiones de +12 (V_{DC}) y -12 (V_{DC}) necesarias para polarizar los amplificadores operacionales, los cuales se encargarán de procesar las señales provenientes del sensor de voltaje y del sensor de corriente. Además, se tiene un módulo con una tensión de salida de +5 (V_{DC}) para alimentar el controlador y el LCD.

3.2.4 Arduino Pro Mini

Placa desarrollada mediante el microcontrolador ATmega328, pensado para aplicaciones en las que el espacio es reducido. (Arduino, 2019).

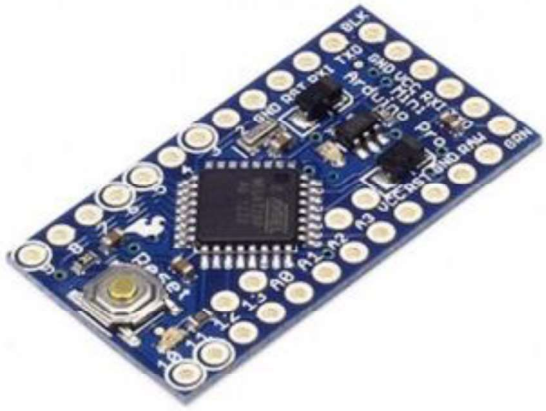

Figura 3.6. Arduino pro mini.
(Fuente: Arduino, 2019)

En la Tabla 3.3 se observan las características técnicas de la placa.

Tabla 3.3. Características técnicas Arduino pro mini.

Alimentación	5 (V_{DC})
Voltaje de Operación	5 (V_{DC})
Entradas Analógicas	6
Velocidad de Reloj	16 (MHz)
Entradas Digitales	14

(Fuente: Autoría propia)

3.2.5 Modulo relé

El módulo relé brinda la opción al administrador del servicio eléctrico de cortar y activar la energía en caso de falta de pago de la planilla del servicio. Las características técnicas del módulo se listan en la Tabla 3.7.

Figura 3.7: Módulo relé de 1 canal.
(Fuente: Naylampmechatronics, 2019)

Tabla 3.4. Características técnicas del módulo relé.

Canales	1
Voltaje de alimentación	5 (V_{DC})
Voltaje de activación	5 (V_{DC})
Corriente de activación	0.2 (A_{DC})
Tensión máxima AC	250 (V_{DC})
Tensión máxima DC	30 (V_{DC})
Corriente máxima	10 (A_{AC})

(Fuente: Autoría propia)

3.3 Diagrama de Bloques

El prototipo desarrollado consta de tres módulos, los cuales realizan las tareas de recolectar información, procesar ésta información y transmitirla.

La tarjeta de adquisición de datos toma la señal de voltaje del transformador de 12 (V_{AC}) y la señal de corriente del CTs, posteriormente se encarga de procesarlas y acoplarlas para que sean reconocidas por el microcontrolador.

El microcontrolador calcula la potencia instantánea mediante la información que recibe de la tarjeta de adquisición de datos y posteriormente envía los datos hacia el módulo wifi.

El módulo wifi ESP8266 realiza la mayor parte de las tareas, tales como, levantar el servidor web, configurar el cliente NTP, calcular la energía usando los datos provenientes del microcontrolador, conectar el sistema a una red wifi, enviar los datos al internet, desplegar los datos en el LCD, almacenar la página web, y controlar el estado del relé para activar o desactivar la energía eléctrica de acuerdo con los requerimientos del proveedor de servicio. En la Figura 3.8 se observa el diagrama de bloques del módulo desarrollado. En el que se resume lo expuesto anteriormente.

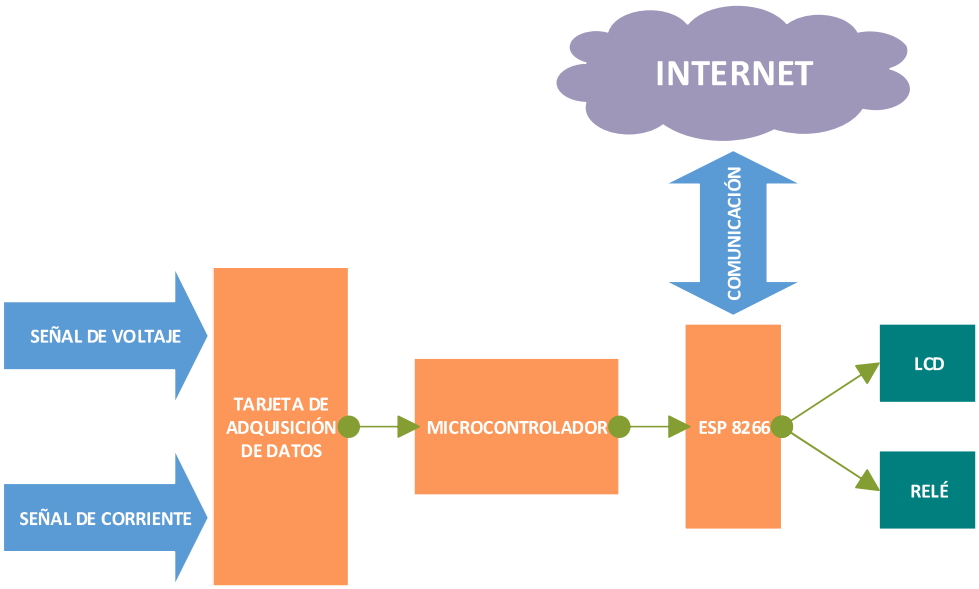

Figura 3.8. Diagrama de bloques para el desarrollo del prototipo.
(Fuente: Autoría propia)

3.4 Tarjeta de Adquisición de Datos

En la Figura 3.9 se observa el diagrama de bloques de la tarjeta de adquisición.

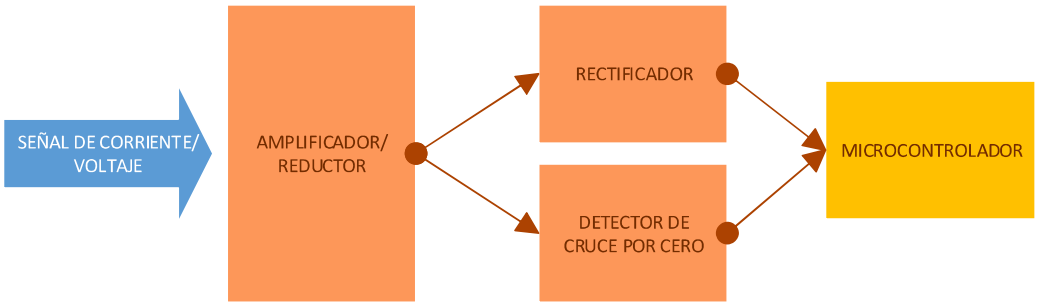

Figura 3.9. Diagrama de bloques de la tarjeta de adquisición.
(Fuente: Autoría propia)

La señal proveniente del transformador de 12 (V_{AC}), la cual mide el valor del voltaje de línea ingresa a un amplificador inversor con el fin de reducir la señal a 5 (V_{AC}). El circuito desarrollado se muestra en la Figura 3.10.

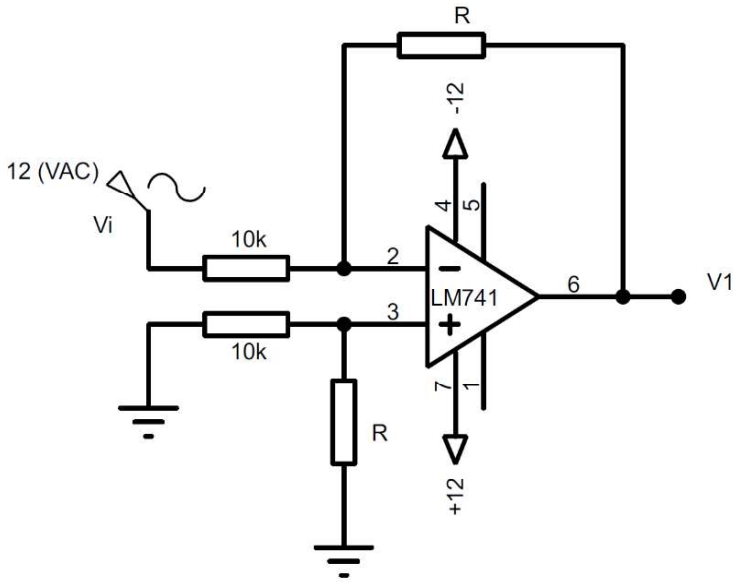

Figura 3.10. Amplificador Inversor.
(Fuente: Autoría propia)

Para calcular el valor de la resistencia de retroalimentación del circuito de voltaje (R_V) se procedió de la siguiente manera:

$$V_1 = -\frac{R_v}{10\ K\Omega} \times V_i \qquad \text{Ec. 3.1}$$

$$R_v = -\frac{-5}{12} \times 10000 \qquad \text{Ec. 3.2}$$

$$R_v \approx 3900\ (\Omega) \qquad \text{Ec. 3.3}$$

Para el caso de la señal de corriente se fabricó otra placa con el mismo circuito mostrado en la Figura 3.10 pero en este caso la señal del CTs debe ser amplificada hasta un valor máximo de 5 (V_{AC}). A continuación, se muestra el procedimiento para hallar R_i, hay que tomar en cuenta que el voltaje de entrada máximo (V_i) corresponde al voltaje de salida máximo del sensor de corriente, es decir, 1 (V_{pico}).

$$R_i = -\frac{-5}{1} \times 10000 \qquad \text{Ec. 3.4}$$

$$R_i \approx 47000\ (\Omega) \qquad \text{Ec. 3.5}$$

Con el fin de visualizar la forma de onda a la salida del sensor CTs, se colocó una carga que produjera una corriente de 10 (A_{AC}), con lo cual se obtuvo un valor de voltaje de 1 (V_{p-p}) en el secundario del transformador de corriente, esto se puede apreciar en la Figura 3.11.

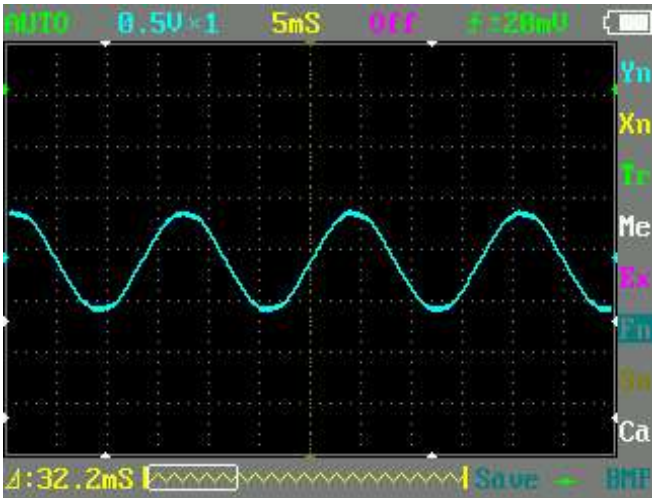

Figura 3.11. Forma de onda del sensor CTs
(Fuente: Autoría propia)

Luego la señal pasa a través del circuito de la Figura 3.7 para ser amplificada, aplicando la ecuación Ec. 3.1 se obtiene un voltaje de salida (V1) como se indica a continuación.

$$V_o = -\frac{47000}{10000} \times 0.5 \qquad \text{Ec. 3.6}$$

$$V_o = -2.35\ (V_{AC}) \qquad \text{Ec. 3.7}$$

La forma de onda en este punto se muestra en la Figura 3.12.

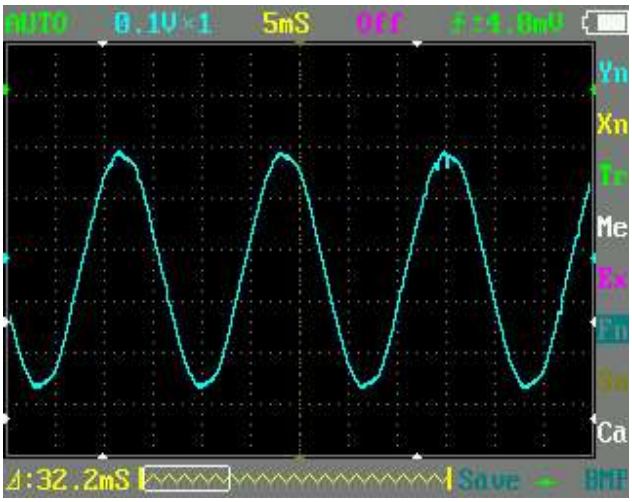

Figura 3.12. Forma de onda a la salida del amplificador inversor.
(Fuente: Autoría propia)

Debido a que las entradas analógicas del microcontrolador aceptan únicamente valores positivos de voltaje, primero se tiene que rectificar la señal.

Cuando se usan rectificadores de onda completa, tipo puente constituido por diodos de silicio, en ondas de señal alterna de 120 (V_{AC}) es posible despreciar la caída de tensión de 1.4 (V_{DC}) en los diodos sin que se genere un error significativo. Pero para el caso de una onda de 5 (V_{AC}), la caída de tensión en los diodos provoca errores grandes. Para solucionar este inconveniente se empleó un rectificador de precisión, el cual funciona como un rectificador ideal.

En la Figura 3.13. se observa el diagrama circuital del rectificador de precisión que emplea amplificadores operacionales.

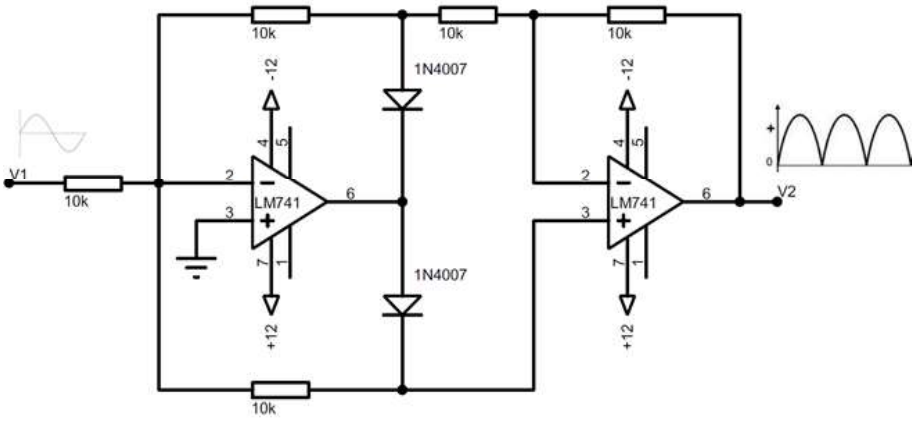

Figura 3.13. Rectificador de onda completa de precisión.
(Fuente: Autoría propia)

El amplificador operacional de la derecha trabaja como amplificador inversor o no inversor, dependiendo de cuál diodo esté activado. En consecuencia, el voltaje V2 de salida corresponde a una señal que no invierte el semi-ciclo positivo de entrada y si invierte el semi-ciclo negativo de V1. Todas las resistencias tienen el mismo valor por lo que su impedancia de entrada es igual a 10 (KΩ).

Finalmente, a la salida del circuito se tiene el voltaje V2, que corresponde a una señal rectificada en onda completa con 4.68 (V_p) para el caso de la señal de voltaje de 110 (V_{AC}) y 4.70 (V_p) para el caso de la señal de corriente de 20 (A_{AC}). En la Figura 3.14 se observa la señal de salida del rectificador de precisión para la señal proveniente del sensor CTs.

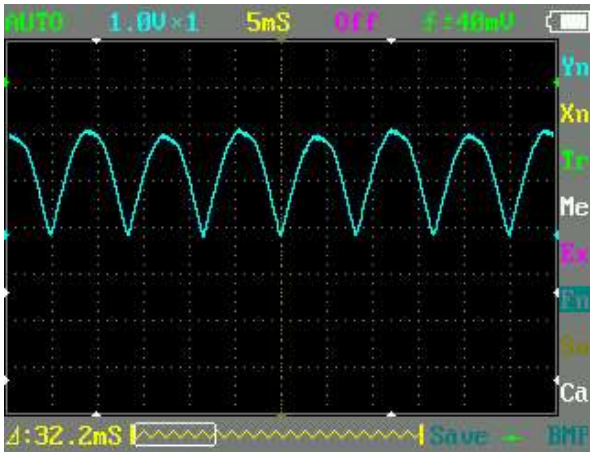

Figura 3.14. Forma de onda a la salida del rectificador de precisión.
(Fuente: Autoría propia)

Para encontrar el desfase entre el voltaje y la corriente, y posteriormente su factor de potencia, fue necesario desarrollar un circuito detector de cruce por cero para cada una de las señales de entrada. Este circuito es un comparador que contrasta el voltaje en la entrada inversora (V1) con el voltaje de referencia (0 V_{DC}) y luego entrega un tren de pulsos con la misma fase y frecuencia que la señal sinusoidal original. El diodo zener limita la señal de salida y entrega pulsos positivos de 5 (V_{DC}) los cuales ingresaran al microcontrolador para realizar los cálculos necesarios. En la Figura 3.15 se observa el circuito empleado.

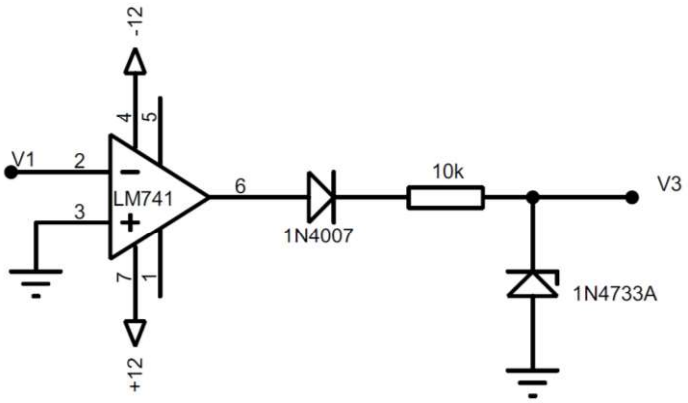

Figura 3.15. Detector de cruce por cero.
(Fuente: Autoría propia)

En la Figura 3.16 se aprecia el diagrama del circuito final de la tarjeta, la cual corresponde a la integración del amplificador inversor, el rectificador de precisión y el detector de cruce por cero.

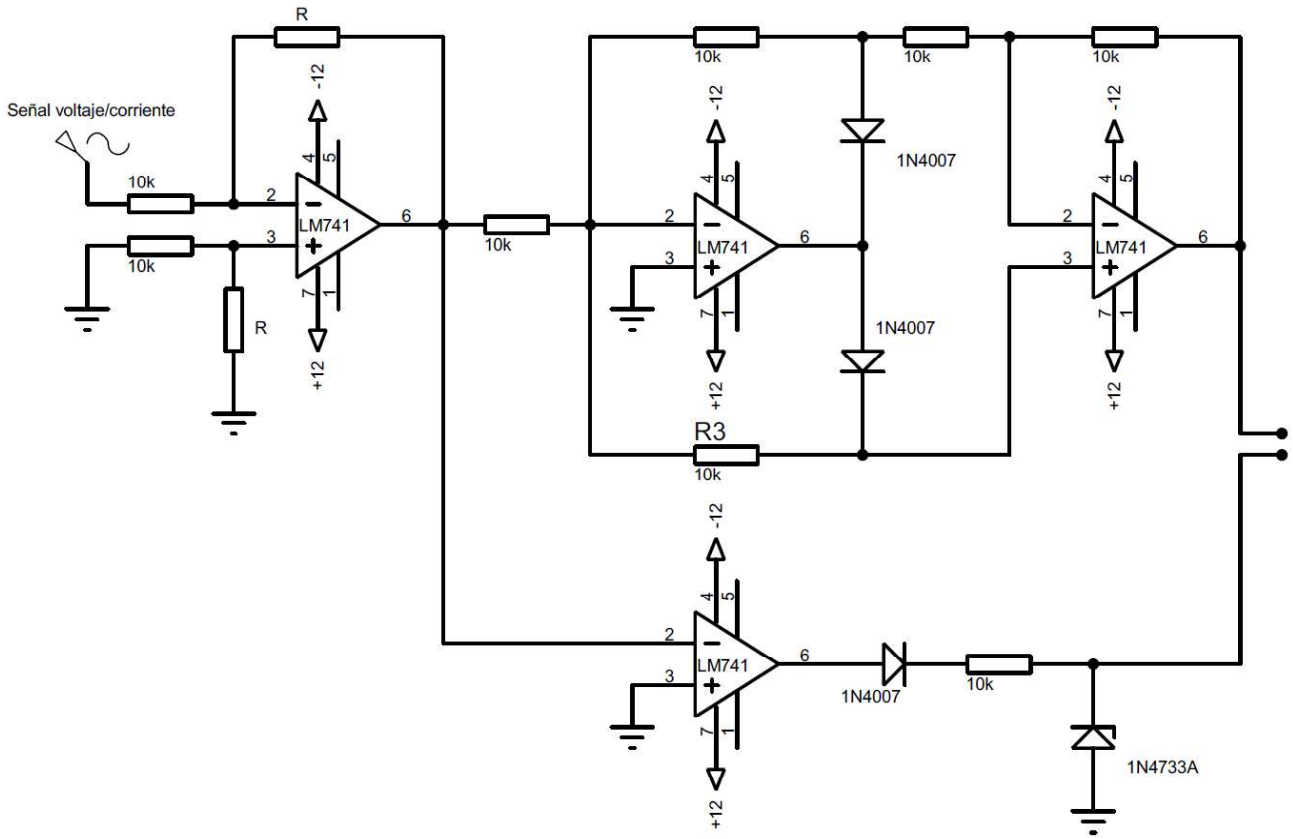

Figura 3.16. Diagrama circuital de la tarjeta de adquisición.
(Fuente: Autoría propia)

3.4.1 Fabricación de las Placas de Circuito Impreso.

Se usó la herramienta EAGLE de AutoDesk para diseñar y rutear las pistas de las tarjetas. Se desarrollaron las placas para el fácil montaje y desmontaje en el módulo y la rápida conexión de los sensores en las mismas. Se elaboraron dos PCBs, una para tomar la señal de voltaje y otra para tomar la señal de corriente. En la Figura 3.17 se observa el diseño de las placas tanto para la cara superior como para la cara inferior.

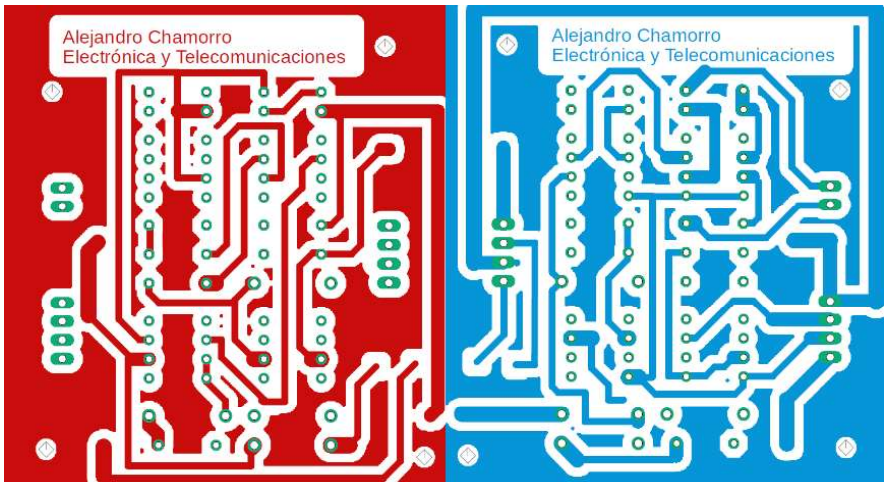

Figura 3.17. Pistas para la fabricación de la tarjeta de adquisición.
(Fuente: Autoría propia)

Finalmente, en la Figura 3.18 se observa el resultado final de la placa de circuito impreso con todos sus elementos soldados y lista para ser montada en el módulo con los demás componentes.

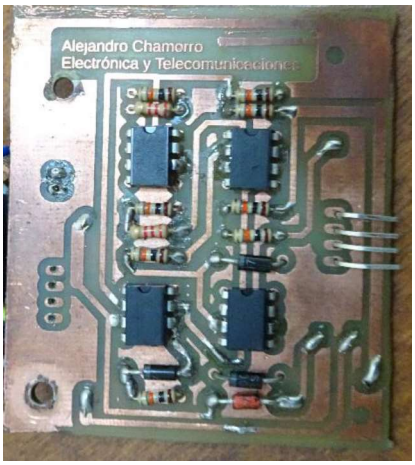

Figura 3.18. Tarjeta de adquisición de la señal de corriente.
(Fuente: Autoría propia)

Tanto la tarjeta de adquisición de la señal de voltaje como la tarjeta de adquisición de la señal de corriente son similares, con la única diferencia que la señal de entrada proviene de diferentes sensores.

3.5 Algoritmo del Microcontrolador.

3.5.1 Diagrama de Flujo

En la Figura 3.19 se observa el diagrama de flujo del algoritmo que permite calcular la potencia mediante las señales de voltaje, corriente y del circuito detector se cruce por cero provenientes de la etapa anteriormente descrita.

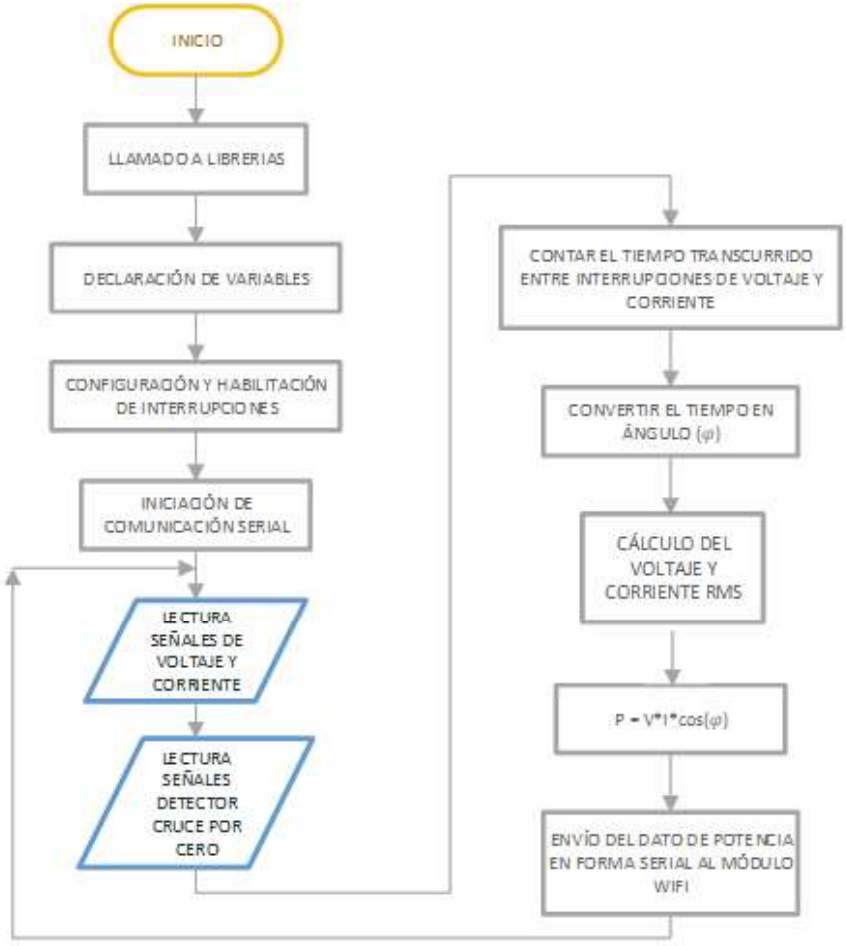

Figura 3.19. Diagrama de flujo del algoritmo de cálculo de la potencia eléctrica. (Fuente: Autoría propia)

Las señales que ingresan a las entradas analógicas del microcontrolador (entrada A0 para la señal de voltaje, entrada A1 para la señal de corriente) corresponden a señales alternas rectificadas en onda completa, tal como se muestra en la Figura 3.11. Posteriormente se procedió a realizar un escalamiento de las señales con el fin de calcular el valor medio cuadrático real de las dos señales.

Los pulsos provenientes del circuito detector de cruce por cero ingresan a los pines digitales D2 (pulsos de voltaje) y D3 (pulsos de corriente), luego, mediante el uso de las interrupciones del microcontrolador, se calcula el tiempo transcurrido desde el flanco de bajada del pulso de voltaje hasta el flanco de bajada del pulso de corriente, con este valor se hace la respectiva conversión a ángulo y finalmente se obtiene el $\cos\varphi$ (factor de potencia). Seguidamente, con los valores de voltaje, corriente y $\cos\varphi$ se calculó la potencia.

Para terminar, el valor de la potencia se envía de forma serial en formato JSON hacia el módulo wifi. es preciso enviar la información de esta forma ya que es de gran ayuda en el desarrollo de la interfaz web. Para realizar esta tarea se usó la librería JSON para Arduino.

3.6 Configuración de la Infraestructura de Red WiFi

3.6.1 Configuración del Cliente NTP.

Para configurar el cliente se usó la librería *NTPClient* de Arduino, la cual ofrece una interfaz simple para consulta del servicio desde el módulo wifi ESP8266. También proporciona funcionalidades de utilidad para traducir los valores de campo NTP a texto. Además de sincronizar en tiempo real la hora y fecha del dispositivo, la librería proporciona información sobre el tiempo *UNIX* o tiempo *POSIX*, el cual es un sistema para describir instantes de tiempo, indica la cantidad de segundos transcurridos desde la medianoche del 1 de enero de 1970 (medianoche *UTC/GMT*) hasta el instante en que se realiza la consulta de tiempo. Este parámetro es muy importante ya que permitirá calcular la energía consumida.

Luego de descargar la librería, a través del Gestor de Librerías del IDLE (Integrated DeveLopment Environment – Entorno de Desarrollo Integrado) de Arduino y hacer el llamado a la misma, se especificó la dirección del servidor NTP, en este caso se usó el servidor europe.pool.ntp.org para obtener la hora exacta del meridiano de Greenwich. Ya que la diferencia horaria entre el meridiano y Ecuador es de 5 horas, fue necesario definir un retardo de -18000 (s) con la finalidad de emparejar a la hora ecuatoriana. Finalmente, se especifica el intervalo de tiempo en que el cliente enviará solicitudes de actualización al servidor, para este caso se configuró un intervalo de 60000 (ms).

En la Figura 3.20 se muestra la configuración realizada para el cliente NTP.

```
NTPClient timeClient(ntpUDP, "europe.pool.ntp.org", -18000, 60000);
```

Figura 3.20. Configuración de la clase timeClient.
(Fuente: Autoría propia)

En la Tabla 3.5 se describen los métodos utilizados en el cálculo del consumo energético y para desplegar información de fecha en la interfaz gráfica.

Tabla 3.5. Métodos de la librería NTPClient.

Método	Descripción	Salida
update()	Actualiza el servidor.	----------
getEpochTime()	Tiempo UNIX	1569941905 (1 de octubre de 2019, 14:58:25)
getDay()	Día Actual	1
getHours()	Hora Actual	14
getMinutes()	Minuto Actual	58
getFormattedTime()	Fecha actual	14:58:25

(Fuente: Martín, 2019)

Se hace el llamamiento al método *update()* para enviar peticiones al servidor y recibir una respuesta. Con *getEpochTime()* se recibe el tiempo *UNIX* del servidor, este método es indispensable para el posterior cálculo de la energía. Los métodos *getDay(), getHours(), getMinutes()* y *getFormattedTime()* fueron necesarios para desplegar información del tiempo en la interfaz gráfica.

3.6.2 Cálculo del Consumo de Energía.

El valor de la potencia es recibido de forma serial en formato *JSON* y almacenada en una variable para proceder con los cálculos correspondientes.

Con el fin de obtener el valor de la energía consumida se empleó la Ecuación 3.4 que se muestra a continuación.

$$E = \frac{(V \times I \times \cos\varphi)}{3600000} \times t \qquad \text{Ec. 3.8}$$

Donde la Energía (E) está dada en KWh, V corresponde al voltaje medio cuadrático de línea, I es la corriente eficaz de línea y el tiempo (t) en segundos.

El numerador de la ecuación anterior corresponde a la potencia, por lo tanto, se tiene la ecuación Ec. 3.9.

$$E = \frac{P}{3600000} \times t \qquad \text{Ec. 3.9}$$

Donde la potencia (P) está dada en KWh y el tiempo (t) en segundos.

Para conseguir el valor del tiempo se usó la información ofrecida por el servidor NTP. El cliente actualiza su información de tiempo cada 60 segundos, por lo tanto, el valor del tiempo exacto será la diferencia entre el tiempo *UNIX* actual y la medición del tiempo *UNIX* anterior. Con este valor se decidió calcular la energía consumida en un intervalo de 60 (s), asumiendo que la variación de potencia entre intervalos de medición no sea muy elevada.

Con la finalidad de precautelar la integridad de los datos y evitar que éstos se pierdan, cuando exista una posible falla de energía, el valor del consumo se almacena en la memoria EEPROM del módulo wifi.

Se tomó en cuenta la medición mensual del servicio de energía eléctrica, por lo tanto, la fecha de corte es a las cero horas del día 25 de cada mes, momento en el cual se encera la variable de consumo energético y se borra la información almacenada en la memoria EEPROM.

En la Figura 3.21 se muestra el diagrama de flujo para el cálculo del consumo de energía eléctrica.

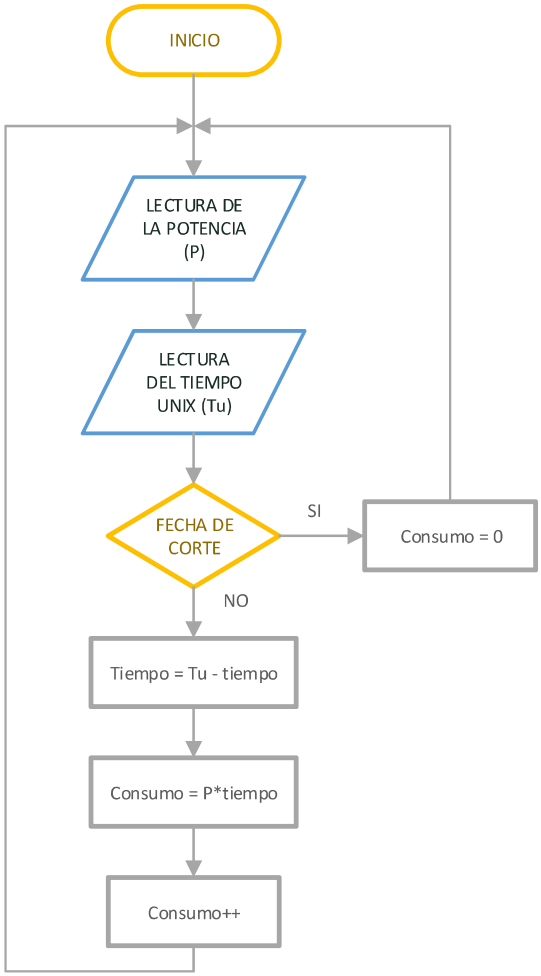

Figura 3.21. Diagrama de flujo para el cálculo del consumo de energía.
(Fuente: Autoría propia)

Con el valor del consumo energético se puede computar el valor de consumo. En la ecuación Ec. 3.11 se describe cómo hallar el precio.

$$\text{Valor Consumo} = \text{E} \times 0.0798 \qquad \text{Ec. 3.10}$$

Dónde Valor Consumo está dado en dólares y la Energía (E) en KWh, el valor de 0.0798 corresponde al precio en dólares de 1 (KWh).

3.6.3 Conexión con la Red WIFI

Una característica del dispositivo a desarrollar es que sea portable, de fácil instalación y la configuración sea sencilla, evitando la programación del código fuente para cambiar los parámetros de usuario y contraseña de la red wifi a la que

se conectará. Por estas razones la configuración de las credenciales de la red se las realiza a través de la interfaz web que proporciona las librerías *WiFiManager* y ESP8266WiFi.

La librería *WiFiManager* provee una interfaz web que facilita la detección de las redes cercanas y la configuración de las credenciales de red. Luego de haber descargado e instalado la librería y hacer su respectivo llamamiento, se debe proporcionar los parámetros de red. En la Figura 3.22 se muestra el script para la correcta utilización de la librería.

```
1  #include <ESP8266WiFi.h>
2  #include <ESP8266WebServer.h>
3  #include <WiFiManager.h>
4  void setup() {
5      WiFiManager wifiManager;
6      wifiManager.resetSettings();
7      wifiManager.autoConnect("CONFIGURACON DE RED", "123456789");
8  }
9  void loop() {
10 }
```

Figura 3.22. Script que muestra la configuración de la librería *WiFiManager*.
(Fuente: Autoría propia)

La línea 5 realiza la inicialización local de la librería, la línea 6 borra parámetros de red almacenados anteriormente en el dispositivo y la línea 7 brinda la característica de punto de acceso (Access Point – AP) al módulo wifi ESP8266, creando la red inalámbrica de nombre CONFIGURACIÓN DE RED y con la contraseña 123456789.

Es posible conectarse al punto de acceso desde cualquier host cercano. En la Figura 3.23 se observa la pantalla de Conexiones de Red de Windows 10, en la cual ya se ha reconocido la nueva red y se debe ingresar la contraseña para acceder.

Figura 3.23. Autenticación para acceder a la red CONFIGURACIÓN DE RED.
(Fuente: Autoría propia)

Por defecto se crea la red 192.168.4.0/24. Se debe acceder a la configuración interna del módulo wifi con el fin de cambiar las credenciales de red, para lo cual, desde cualquier navegador web se ingresa la dirección 192.168.4.1, que corresponde a su puerta de enlace, En la Figura 3.24 se observa el menú principal de la configuración.

Figura 3.24. Menú principal de configuración de conexiones de red.
(Fuente: Autoría propia)

Se debe ingresar en la opción Configure WiFi. En la Figura 3.25 se muestran todas las redes cercanas disponibles a las que se podría enlazar. Para este caso en particular se ha escogido la red con el nombre CHAMORRO y se debe escribir la contraseña necesaria.

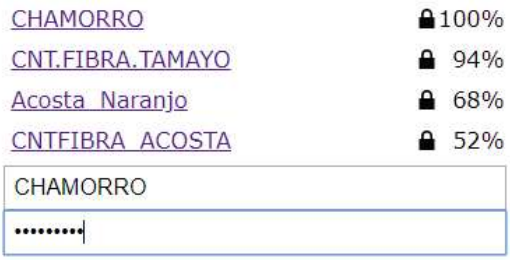

Figura 3.25. Pantalla de selección de red e ingreso de credenciales.
(Fuente: Autoría propia)

Los datos de las credenciales de la nueva red configurada se perderán si existe una falla de energía, por esto fue necesario cambiar el código fuente de la librería

WiFiManager, con el propósito de guardar las credenciales en la memoria EEPROM del módulo wifi. En la Figura 3.26 se observa los cambios realizados a la librería.

```
String contrasen;
contrasen = pass.c_str();
EEPROM.begin(512);
int longitudPass = 0;
longitudPass = contrasen.length();
if(contrasen.length() < 10){
  EEPROM.write(500, longitudPass);
  EEPROM.write(501, 111);
} else {
    EEPROM.write(500, longitudPass - int(longitudPass/10));
    EEPROM.write(501, int(longitudPass/10));
}
for(int n=0; n < contrasen.length(); n++){
  EEPROM.write(n, contrasen[n]);
}
  EEPROM.commit();
for(int n=0; n < contrasen.length(); n++){
}
EEPROM.write(350, 0);
```

Figura 3.26. Cambios realizados a la librería *WiFiManager*.
(Fuente: Autoría propia)

Estos cambios se realizaron dentro de la función connectWifi, la cual es la encargada de leer las credenciales del navegador web para posteriormente conectarse a la red wifi.

Finalmente, se debe asignar la dirección IP, la máscara de red y la puerta de enlace. La librería ESP8266WiFi proporciona esta funcionalidad. En la Figura 3.27 se observa el segmento de código para fijar las direcciones IP en el módulo wifi.

```
#include <ESP8266WiFi.h>
#include <WiFi.h>
IPAddress ip(192,168,0,222);
IPAddress gateway(192,168,0,1);
IPAddress subnet(255,255,255,0);
int status = WL_IDLE_STATUS;
void setup()
{
  // verifica la presencia del dispositivo
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("Módulo WiFi ausente");
    while(true);
  }
  // Asigna las direcciones al dispositivo
  WiFi.config(ip, gateway, subnet);
}
void loop () {}
```

Figura 3.27. Script para ingresar las direcciones IP.
(Fuente: Autoría propia)

3.6.4 Levantamiento Servidor WEB

El módulo wifi ESP8266 tiene que ser programado como Servidor Web con la finalidad de recibir las solicitudes de los diferentes clientes y enviar como respuesta el consumo de energía y valor a pagar, para lo cual se ha empleado la librería ESP8266WebServer, la cual se encarga de administrar las solicitudes HTTP como GET y POST, indispensables para el desarrollo de la comunicación.

Antes de levantar servicio en el módulo wifi fue necesario crear una interfaz gráfica, para lo cual se desarrolló una aplicación web en JavaScript, HTML y CSS en la cual se presenta la información relacionada al consumo de energía, los datos personales del abonado y la opción de cortar el servicio y activarlo.

En la Figura 3.28 se presenta el diseño simple de la página web.

Figura 3.28. Diseño web de la interfaz gráfica.
(Fuente: Autoría propia)

La interfaz gráfica tiene que ser cargada en la memoria flash del módulo wifi ESP8266, para lo cual fue necesario crear un fichero de cabecera y guardarlo con el nombre *index.h*, en el cual se almacenan las definiciones de librería. En el interior de

este fichero se define la clase *MAIN_page* que contiene el código íntegro de la aplicación web. Por último, el fichero de cabecera y el archivo con extensión ino, que corresponde al código del módulo wifi, deben ubicarse en la misma ruta para luego ser cargadas a la placa ESP8266 mediante el IDLE de Arduino.

Usando la librería *ESP8266WebServer* se inicia el servicio web y se carga la interfaz en la memoria flash. En la Figura 3.29 se observa el código que permite realizar estas acciones.

```
1  #include <ESP8266WebServer.h>
2  #include "index.h"
3  WebServer server(80);
4  void handleRoot() {
5   String s = MAIN_page;
6   server.send(200, "text/html", s);
7  }
8  void setup(void){
9    server.on("/", handleRoot);
10   server.begin();
11 }
12 void loop(void){
13   server.handleClient();
14 }
```

Figura 3.29. Código para configuración del servidor web.
(Fuente: Autoría propia)

La línea 2 hace el llamado y carga en la memoria flash del módulo wifi el fichero que contiene el código de la interfaz gráfica, la línea 3 crea el objeto server y asigna el puerto 80 al host del servidor, la función *handleRoot* se ejecuta al momento de ingresar la dirección IP del servidor en el navegador web y despliega la interfaz, en la función *setup* se inicia el servidor web y finalmente, se hace el llamado a la función *handleClient* con el objetivo de recibir las solicitudes de los clientes y enviar las respuestas correspondientes.

La información recibida en formato JSON desde el microcontrolador es desplegada en la interfaz gráfica cada vez que el servidor recibe una petición de un cliente, por lo tanto, para definir una respuesta a una solicitud, se tiene que hacer un llamado al método *send* en el objeto server. Los argumentos de éste método son el código de respuesta HTTP, el tipo de contenido y el contenido en sí. En la Figura 3.30 se aprecia la creación de la función *handleVoltaje*.

```
138  void handleVoltaje(){
139    server.send(400, "text/plane", String(voltaje));
140  }
```

Figura 3.30. Creación de la función *handleVoltaje*.
(Fuente: Autoría propia)

El código de respuesta HTTP 200 (OK) indica el estado ante una solicitud estándar correcta a la que puede responder sin problemas, el tipo de contenido es texto plano y el contenido corresponde al valor del voltaje.

Posteriormente, se debe indicar al servidor que hacer con cada solicitud recibida, por ejemplo, en el caso de recibir la solicitud */readVoltaje*, el servidor debe responder con el valor del voltaje el cual se especifica en la función *handleVoltaje*. Para realizar esta acción se debe invocar al método *on* dentro de la función *setup* del script. En la Figura 3.31 se muestra el método *on* para el valor del voltaje.

```
456    server.on("/readVoltaje", handleVoltaje);
```

Figura 3.31. Método *on* para el valor del voltaje.
(Fuente: Autoría propia)

De esta forma el servidor ejecutará automáticamente las funciones correctas de acuerdo a las peticiones recibidas por parte del cliente.

3.6.5 Configuración del Modem del ISP

Es preciso que el servidor web envíe la información fuera de la conexión de área local hacia el Internet, por lo tanto, fue necesario configurar el modem ADSL del proveedor de internet para abrir el puerto del host y así crear una puerta de enlace hacia el internet. En la pestaña Port *Mapping Configuration* del menú Forward Rules se debe agregar la dirección IP del servidor y en la pestaña *Port Trigger Configuration* se debe seleccionar el protocolo TCP (Transmission Control Protocol – Protocolo de Control de Transmisión) y colocar el número del puerto del servidor en todas las opciones, en este caso se asignó el puerto 1201 al host.

En la Figura 3.32 se observan las configuraciones realizadas en el módem.

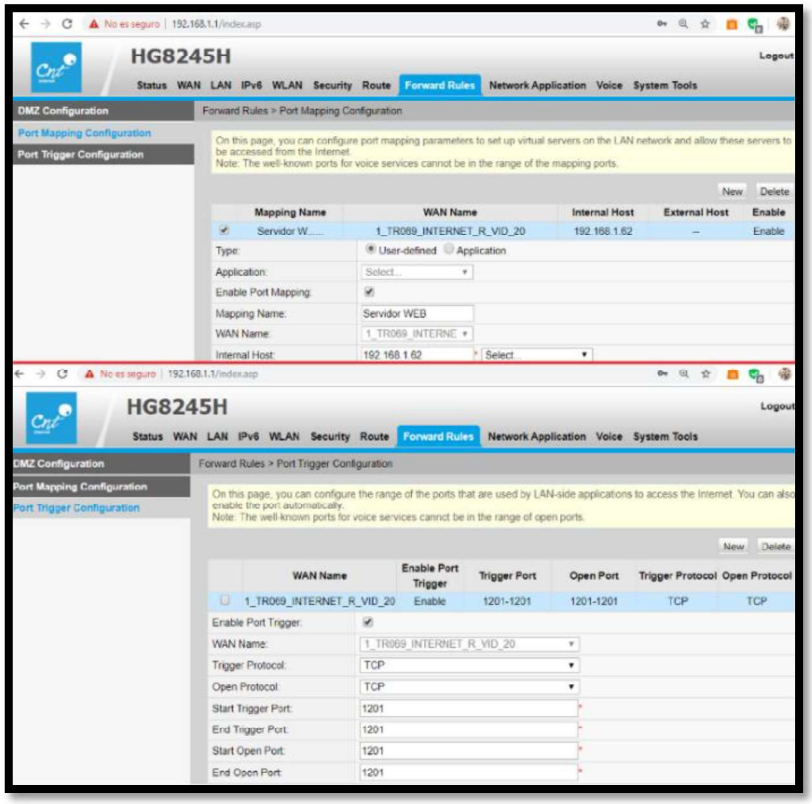

Figura 3.32. Configuraciones realizadas al modem.
(Fuente: Autoría propia)

Para poder acceder al servidor desde cualquier lugar en el Internet se debe hacerlo ingresando la dirección IP pública que el proveedor de internet asignó a nuestra red más el número de puerto del servidor.

3.6.6 Desarrollo de la Estructura.

En la Figura 3.33 se muestra el diagrama circuital final del prototipo desarrollado.

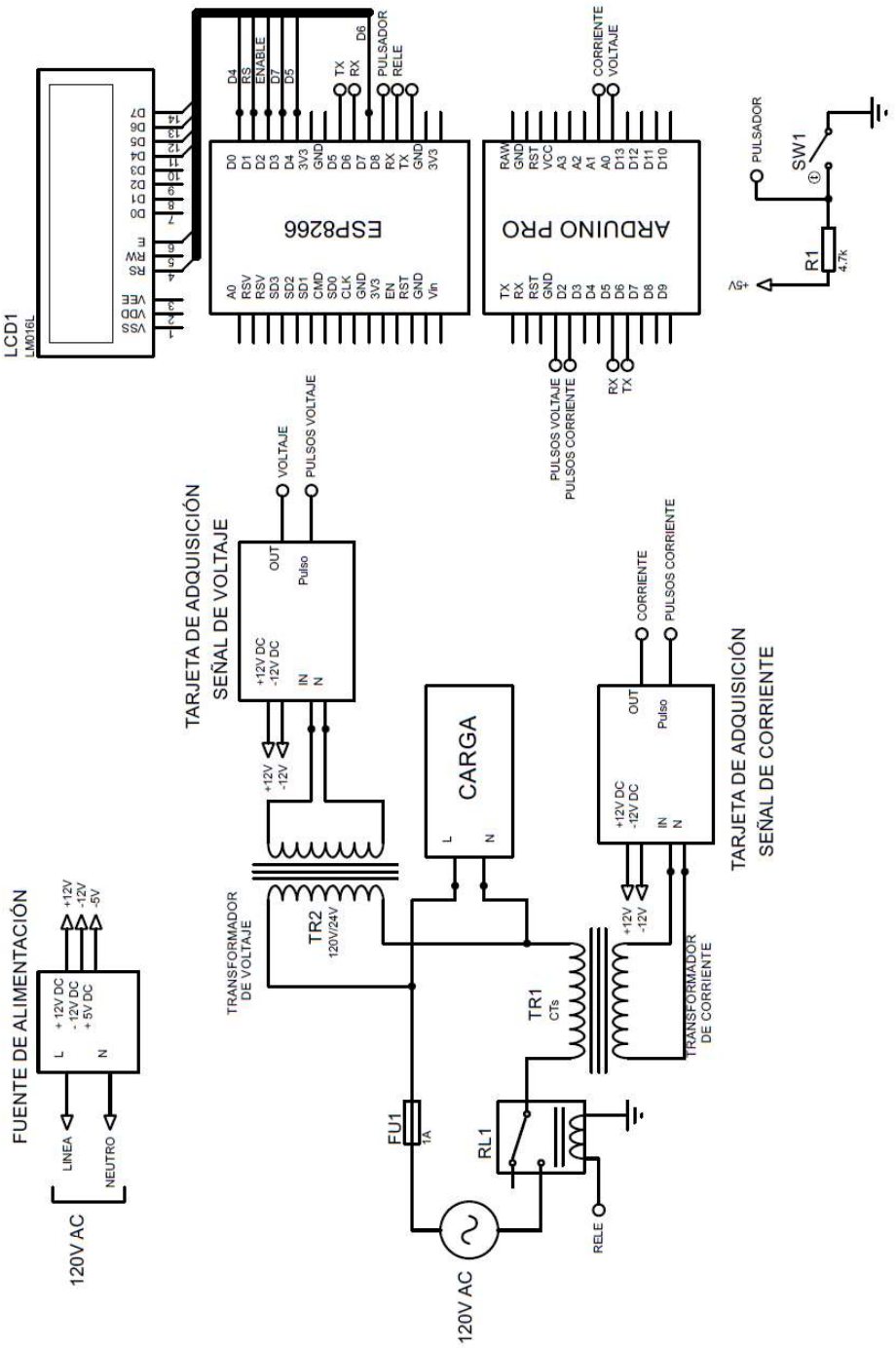

Figura 3.33. Diagrama circuital final del prototipo.
(Fuente: Autoría propia)

3.6.7 Montaje de Componentes

Luego de comprobar el funcionamiento de las tarjetas y de cada uno de los componentes electrónicos se procedió montarlos en una caja de plástico de 19.75

(cm) de largo, 11.2 (cm) de ancho y 5.8 (cm) de altura, de tal forma que se lo transporte de forma segura, su manipulación sea cómoda y se tome en cuenta las especificaciones descritas en la metodología.

Las fuentes de alimentación se dispusieron en una placa de circuito impreso con el fin de crear un solo módulo de alimentación con voltajes de -12 (V_{DC}) y +12 (V_{DC}) para polarizar los amplificadores operacionales de las placas de adquisición, además de una salida de +5 (V_{DC}) para polarizar el microcontrolador y el módulo wifi. La placa de circuito impreso se muestra en la Figura 3.34, se aprecia que en la parte superior se localiza la entrada de 120 (V_{AC}) y en la parte inferior las salidas DC.

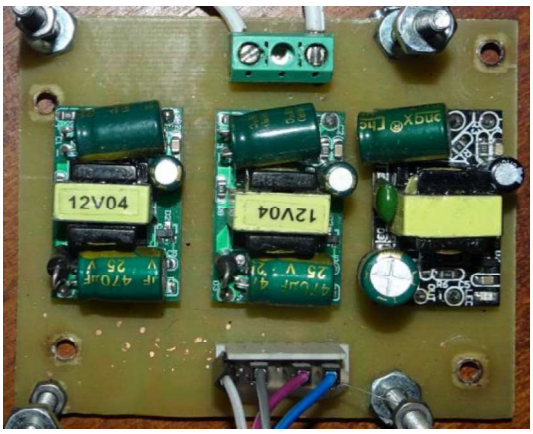

Figura 3.34. Fuente de alimentación.
(Fuente: Autoría propia)

Luego, las placas fabricadas se dispusieron en un arreglo vertical con el fin de crear un solo cuerpo y de esta manera sea cómoda su disposición en la caja de plástico. Este arreglo se observa en la Figura 3.35 (A).

A continuación, se colocó el transformador de voltaje y el transformador de corriente como se aprecia en la Figura 3.35 (B).

Posteriormente, se perforó una abertura en la parte superior de la caja de plástico para situar el LCD y se instaló el módulo relé, este procedimiento se ve en la Figura 3.35 (C) y la Figura 3.35 (D)

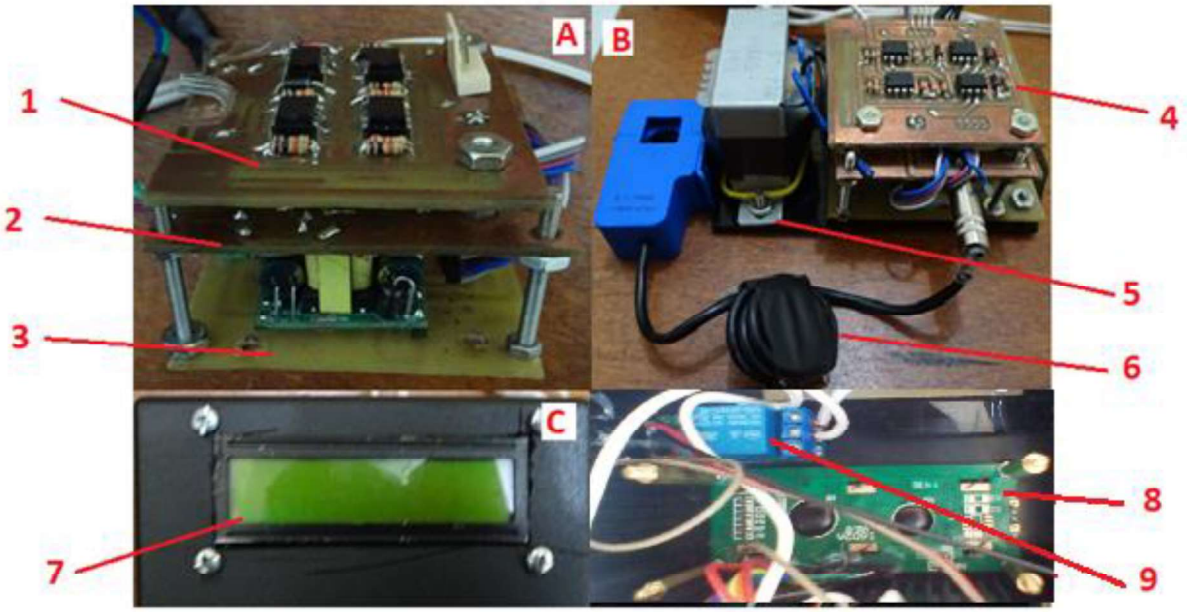

Figura 3.35. Montaje de los componentes.
(Fuente: Autoría propia)

1. Placa de adquisición de la señal de voltaje.
2. Placa de adquisición de la señal de corriente.
3. Placa de fuentes de alimentación.
4. Módulo de PCBs.
5. Transformador de voltaje.
6. Transformador de corriente (CTs).
7. LCD vista frontal.
8. LCD visto desde el interior de la caja de plástico.
9. Módulo relé.

En seguida, se situaron de forma ordenada todos los elementos en el interior de la caja de plástico, se instaló el fusible de protección, se realizaron las conexiones necesarias, se ajustaron los módulos con tornillos y añadieron las borneras de entrada y salida de voltaje del prototipo. En la Figura 3.36 (A) se observa la vista del interior de la caja con los elementos colocados.

Por último, se juntó el microcontrolador y el módulo wifi ESP8266 en un solo módulo, se realizaron las conexiones necesarias y se colocó en el interior de la caja.

En la Figura 3.36 (B) se ve el resultado final luego de juntar los módulos y elementos.

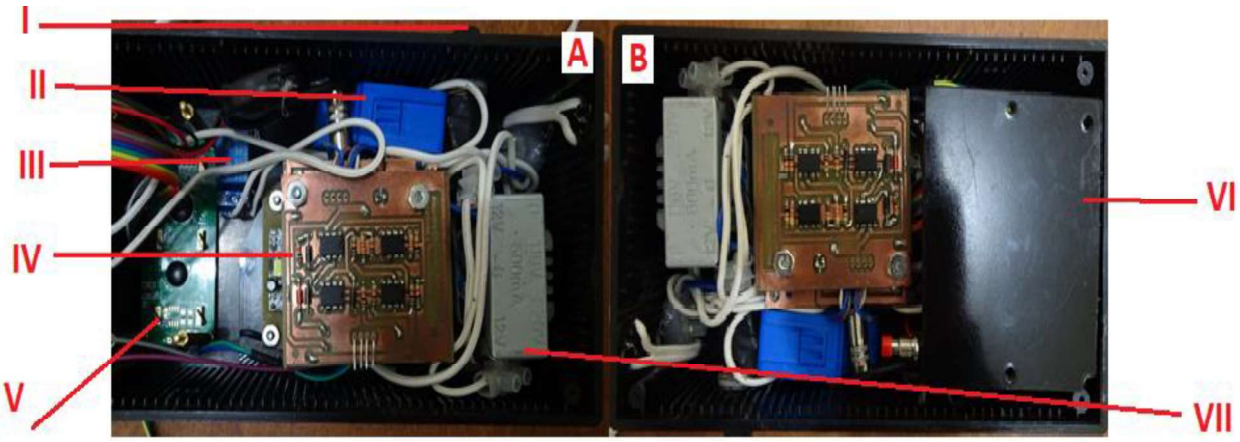

Figura 3.36. Disposición final de los elementos en la caja.
(Fuente: Autoría propia)

I. Fusible de protección.
II. Sensor de corriente CTs.
III. Relé.
IV. Módulo de PCBs.
V. LCD.
VI. Módulo controlador – ESP8266.
VII. Transformador de voltaje.

3.7 Pruebas De Funcionamiento

A continuación, se describen las pruebas realizadas para validar el funcionamiento del prototipo. Se verificó el funcionamiento tanto de la conectividad y comunicaciones como de la precisión del prototipo en la toma de datos.

3.7.1 Pruebas de Conectividad

Esta prueba se la realizó con la finalidad de verificar la accesibilidad hacia el dispositivo desde cualquier lugar del internet.

El proveedor de servicio de internet asigna direcciones IP públicas dinámicas a cada usuario, por lo tanto, es preciso conocer cuál dirección IP está asignada en ese instante. Existen muchas aplicaciones en el Internet las cuales proporcionan

información sobre la dirección IP pública y el estado de los puertos. Para este caso se usó el servicio proporcionado por la página www.yougetsignal.com.

El puerto 1201 corresponde al puerto del servidor web configurado en el prototipo, el cual se encuentra abierto, lo que quiere decir que se puede acceder al prototipo desde cualquier lugar del internet. La Figura 3.37 muestra la dirección IP pública y el estado del puerto 1201.

Figura 3.37. Verificación de la IP pública.
(Fuente: You get signal)

Posteriormente se comprobó el acceso hacia el dispositivo desde diferentes IPs públicas para verificar la conectividad y controlar el corte y activación del servicio de energía eléctrica. En la Tabla 3.6 se muestra el resultado de este proceso.

Tabla 3.6. Verificación de la conectividad con diferentes direcciones IP

IP PÚBLICA	ACCESO AL PROTOTIPO	CONTROL DE CORTE/ACTIVACIÓN DE ENERGÍA
200.63.215.238	SI	SI
186.71.179.84	SI	SI
200.63.215.234	SI	SI
190.155.198.148	SI	SI

(Fuente: Autoría propia)

1.7.2 Pruebas de Medición

Para realizar esta prueba se colocó distintas cargas a la salida del módulo y se procedió a tomar mediciones tanto de voltaje y de corriente, posteriormente se calculó el error entre las mediciones reales y aquellas obtenidas por el prototipo.

En la Figura 3.38 se observa el montaje para realizar las mediciones, el cual dispone de un multímetro para la medición del voltaje y otro para la toma de datos de corriente.

Figura 3. 38. Esquema para la toma de datos.
(Fuente: Autoría propia)

en la Tabla 3.7 se observan los resultados.

Tabla 3.7. Verificación de la toma de las mediciones.

UNIDAD	VALOR REAL	VALOR DEL PROTOTIPO	ERROR
Voltaje	120 (V_{AC})	118 (V_{AC})	1.67 (%)

Corriente	4 (A_{AC})	4.22 (A_{AC})	5.50 (%)
Corriente	6 (A_{AC})	6.30 (A_{AC})	5.00 (%)
Corriente	8 (A_{AC})	8.36 (A_{AC})	4.50 (%)
Corriente	12 (A_{AC})	12.59 (A_{AC})	4.90 (%)
Corriente	16 (A_{AC})	16.85 (A_{AC})	5.30 (%)
		ERROR PROMEDIO	4.48 (%)

(Fuente: Autoría propia)

En la Tabla 3.7 se observa que el error promedio es de 4.48 (%), lo cual quiere decir que los resultados obtenidos en la práctica son cercanos a la realidad.

Además, hay que indicar que el prototipo cumple con las especificaciones del estándar 802.11, esto debido a que en el módulo wifi ESP8266 se encuentran integradas las características del estándar.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se desarrolló un prototipo electrónico para la medición del consumo de energía eléctrica y la posterior consulta de la información del abonado desde cualquier lugar en el internet. El prototipo cumple con todos los objetivos planteados y además es susceptible de mejoras tanto en la parte de hardware como en la parte de software.
- Se realizó la investigación referente a la forma en que se recopila los datos de consumo energético en el sector residencial del Ecuador y se comprobó que este es un proceso que puede ser modernizado mediante la implantación de una Infraestructura de Medición Avanzada usando Medidores Inteligentes. También se observó que, aunque la inversión inicial fuese elevada, se esperaría que a largo plazo ahorre recursos económicos y humanos a las empresas eléctricas locales.
- Se obtuvo la potencia de consumo energético, para lo cual se empleó un circuito sencillo pero capaz de procesar las señales de los sensores de tal forma que puedan ingresar sin problemas a los pines analógicos y digitales del microcontrolador.
- Para comunicar el prototipo con el Internet fue necesario realizar la conexión a una red wifi de área local ingresando sus credenciales y asignando las direcciones IP, en esta tarea se usaron las librerías del módulo wifi ESP8266, las cuales están disponibles en el repositorio GitHub. Finalmente, para crear una puerta de enlace entre la red LAN y el Internet se tuvo que abrir los puertos en el modem del proveedor de servicios.
- La interfaz gráfica desarrollada permite la lectura rápida de la información del cliente desde cualquier navegador web con acceso a internet, además cuenta con la opción de corte manual del servicio eléctrico en caso de que el abonado no haya cancelado la planilla de luz a tiempo.
- Para la construcción y armado de la estructura se tomó en cuenta los posibles escenarios en los que podría estar involucrado el prototipo, para lo cual se dispuso los componentes en el interior de una estructura de

plástico cerrada para protección contra el clima y caídas, se pensó en una estructura modular para el fácil reconocimiento de fallas y rápido intercambio de piezas, se puso énfasis en la capacidad de realizar su trabajo por períodos prolongados de tiempo y también se tomó en cuenta su fácil instalación y configuración.

4.2 Recomendaciones

- Para abrir los puertos en el modem basta con llamar a servicio técnico del proveedor de Internet y solicitar la apertura de los puertos. En caso de hacerlo por uno mismo hay que crear respaldo de la configuración actual antes de realizar cualquier cambio, de esta forma, es posible reestablecer los parámetros anteriores.
- Para no cometer ningún error en el uso de las librerías del módulo wifi se recomienda revisar los ejemplos que vienen con las mismas.

BIBLIOGRAFÍA

Agencia de Regulación y Control de Electricidad. (2017). Estadística Anual y Multianual del Sector Eléctrico Ecuatoriano. Recuperado el 23 de abril de 2019, de https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2018/10/estadistica%20reducida.pdf

Aprende a Programar. (2014). ¿Qué es y para qué sirve JSON? Obtenido de Tutorial básico del programador web: Ajax desde cero: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=956:ique-es-y-para-que-sirve-json-especificacion-oficial-javascript-object-notation-diferencia-de-xml-cu01213f&catid=83&Itemid=212

Arduino. (2019). ARDUINO PRO MINI. Obtenido de https://store.arduino.cc/usa/arduino-pro-mini

Arduino. (2019). Arduino Store. Recuperado el octubre de 2019, de ARDUINO PRO MINI: https://store.arduino.cc/usa/arduino-pro-mini

Consejo Nacional de Electricidad (CONELEC). (2011). Estadística del Sector Eléctrico Ecuatoriano. Folleto Resumen. Recuperado el 23 de abril de 2019, de https://www.regulacionelectrica.gob.ec/wp-content/uploads/downloads/2015/11/Folleto-Resumen-Estad%C3%ADsticas-2011.pdf

Corporación Nacional de Electricidad (CNEL EP). (6 de diciembre de 2016). Medición inteligente al servicio de clientes residenciales. Recuperado el 23 de abril de 2019, de https://www.cnelep.gob.ec/tag/cambio-de-medidores/

ebay. (2019). Mini AC-DC 3V 5V 9V 12V 15V 24V Step Down Power Supply Converter Constant Switch. Recuperado el 11 de agosto de 2019, de https://www.ebay.com/itm/Micro-AC-DC-3V-24V-Buck-Step-Down-Power-Supply-Converter-Constant-Switch-Module-/283289635816?_ul=AR

Espressif. (2015). ESP8266EX Datasheet. Version 4.3. (E. S. Team, Ed.) Recuperado el 5 de agosto de 2019, de Espressif Systems IOT Team:

https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf

ETSI. (2018). Smart Metering. Recuperado el 11 de agosto de 2019, de https://www.etsi.org/technologies/internet-of-things/smart-metering

Future Electronics. (2018). NodeMCU (ESP8266 WiFi Programming & Development Kit). Recuperado el 10 de agosto de 2019, de https://store.fut-electronics.com/products/nodemcu-esp8266-programming-and-development-kit

Grupo Velasco. (2019). Transformaor 12V-500mA. Recuperado el septiembre de 2019, de http://www.velasco.com.ec/velasco/producto.php?id=1154

Idrovo, D., & Reinoso, S. (2012). Análisis de la Factibilidad para la Implementación de un Sistema AMI (Advanced Metering Infrastructure) Mediante Contadores Inteligentes por Parte de la Elmpresa Eléctrica Azogues C. A. (J. Bermeo, Ed.) Cuenca, Ecuador: Universidad Politécnica Salesiana, Ingeniería Electrónica. Recuperado el 9 de agosto de 2019, de https://dspace.ups.edu.ec/bitstream/123456789/1933/12/UPS-CT002400.pdf

INDRAMAT PRODUCTS. (2014). 5 Key Benefits of Industrial Automation. Recuperado el 21 de abril de 2019, de http://www.indramat-us.com/5-key-benefits-of-industrial-automation/

Instituto Nacional de Electricidad y Energías Limpias de México (INEEL). (octubre de 2015). Aplicación de Tecnologías de Medición Avanzada (AMI) como Instrumento para Reducción de Pérdidar. Recuperado el 23 de abril de 2019, de https://www.ineel.mx/boletin042015/tecni1.pdf.

Itead Studio. (2019). MX141219001 - AC-DC Power Module 5V 700mA V2. Recuperado el 9 de agosto de 2019, de Mixtrónica: https://mixtronica.com/alimentacao-modulos/9842-mx141219001-ac-dc-power-module-5v-700ma-v2-mx141219001.html

Martín, G. (octubre de 2019). NtpClient. Obtenido de GitHub.com: https://github.com/gmag11/NtpClient

Naylampmechatronics. (2019). Módulo Relay 1CH 5VDC. Recuperado el octubre de 2019, de https://naylampmechatronics.com/drivers/297-modulo-relay-1-canal-5vdc.html

Office of Electricity Delivery and Energy Reability. (2016). Advanced Metering Infrastructure and Customer Systems. Recuperado el 21 de abril de 2019, de U. S. Department of Energy: https://www.energy.gov/sites/prod/files/2016/12/f34/AMI%20Summary%20Report_09-26-16.pdf

Rose, K., Eldridge, S., & Chapin, L. (2015). Internet de las Cosas - una Breve Reseña. Recuperado el 2019 de abril de 2019, de https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf

Time Tools . (6 de diciembre de 2018). What is NTP. Obtenido de Time Tools Ltd.: https://timetoolsltd.com/ntp/what-is-ntp/

U. S. Department of Energy. (septiembre de 2016). Advan.

XT Energy Meter Expert. (2019). Single phase prepaid smart energy meter wifi . Recuperado el 11 de agosto de 2019, de alibaba: https://www.alibaba.com/product-detail/Single-phase-prepaid-smart-energy-meter_60291103825.html?spm=a2700.7724838.2017115.14.54df757c0mUBYy&s=p

YHDC. (2019). Split-Core Current Transformer. Obtenido de https://nicegear.nz/obj/pdf/SCT-013-datasheet.pdf

You get signal. Port forwarding test. Obtenido de https://www.yougetsignal.com/tools/open-ports/

5. ANEXOS

5.1 ANEXO A: Manual de Operación

5.1.1 Diagrama de Conexión

En la parte inferior se encuentran dos borneras en las cuales se conecta el neutro y la fase de la red eléctrica pública. Las borneras no tienen polaridad, por lo tanto, se puede conectar indistintamente el neutro y la fase en cualquiera de ellas.

En la parte superior se encuentran dos borneras que corresponden a la salida hacia la carga, igualmente, no existe polaridad en los bornes de salida para la fase y el neutro.

Para realizar la conexión del módulo se lo debe realizar tal como se indica en la Figura 1.

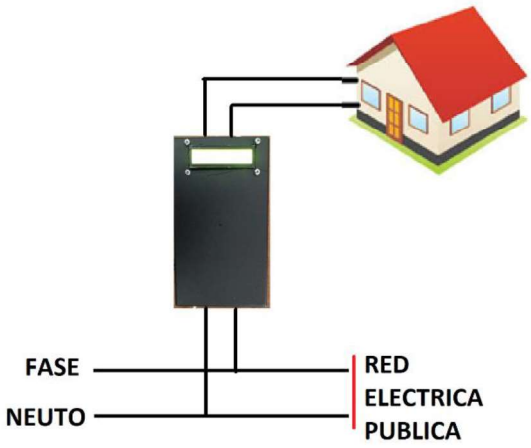

Figura 5.1. Conexión del módulo a la red eléctrica.
(Fuente: Autoría propia)

5.1.2 Configuración Inicial

Luego de realizar la conexión es necesario configurar las credenciales de la red WiFi a la cual se conectará el módulo. Cuando el dispositivo se enciende se muestra en la pantalla LCD el mensaje que indica la figura 5.2.

Figura 5.2. Mensaje de bienvenida.
(Fuente: Autoría propia)

En éste momento se debe mantener presionado el pulsador que se encuentra en el interior de la caja, tal como se muestra en la Figura 5.3.

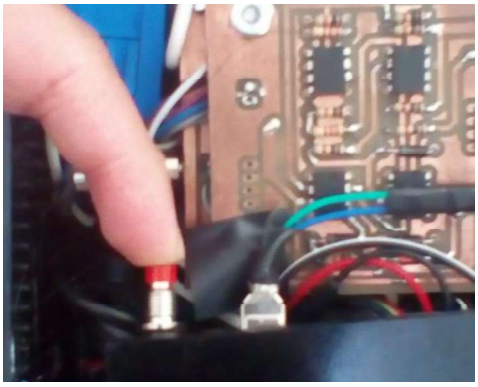

Figura 5.3. Pulsador para ingresar al modo de configuración
(Fuente: Autoría propia)

De esta forma se accede al modo configuración, se observa en la pantalla LCD los mensajes de la Figura 5.4 y Figura 5.5.

Figura 5.4. Mensaje de ingreso al modo configuración
(Fuente: Autoría propia)

Figura 5.5. Mensaje que indica que se debe proceder a configurar la red.
(Fuente: Autoría propia)

Una vez ingresado en el modo de configuración el dispositivo funciona como punto de acceso y se crea la red 192.168.4.0/24 con el nombre “Configuración de Red”. Desde cualquier computador o teléfono móvil es posible conectarse a la red, la cual no necesita contraseña.

Una vez conectado a la red se procede a ingresar la dirección 192.168.4.1 en cualquier navegador web, y se despliega la pantalla de configuración que se muestra en la Figura 5.6.

Figura 5.6. Menú principal pantall de configuración de red.
(Fuente: Autoría propia)

Accediendo en la opción “Configure WiFI” se despliegan todas las redes cercanas, se debe seleccionar la red a la que se desea acceder y luego ingresar la contraseña de la misma.

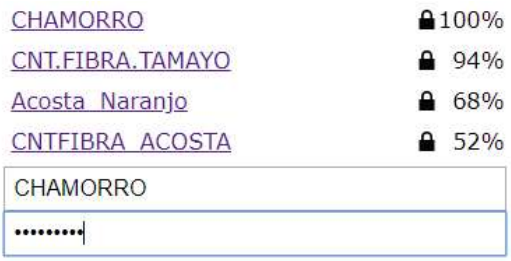

Figura 5.7. Selección de red WiFi.
(Fuente: Autoría propia)

En la opción “Configure WiFi (No Scan)” es posible ingresar manualmente las credenciales de red.

La opción “Info” despliega la información de la red a la cual se encuentra conectada el módulo y finalmente la opción “Reset” borra el usuario y contraseña de la red actual.

Una vez configurado el módulo es necesario apagarlo y volverlo a encender para que los cambios se guarden satisfactoriamente. Cuando el módulo se encienda, se visualizará nuevamente el mensaje de bienvenida, si no se presiona el pulsador el

dispositivo se conecta a la red anteriormente configurada y se despliega el mensaje mostrado en la Figura 5.8, en este caso se ha conectado exitosamente a la red llamada CHAMORRO.

Figura 5.8. Módulo conectado a la red de nombre CHAMORRO.
(Fuente: Autoría propia)

Ya conectado a la red, el modulo entra a trabajar en el modo normal de funcionamiento y se visualizará en la pantalla LCD información referente al consumo energético. En la Figura 5.9 se observa la pantalla con la información de consumo de energía.

Figura 5.9. Pantalla que indica el modo normal de funcionamiento del módulo.
(Fuente: Autoría propia)

La información desplegada corresponde al voltaje, en este caso 120 (V_{AC}), la corriente instantánea, 8.5 (A_{AC}), la energía consumida desde la fecha de corte, día 5 de cada mes, (9.99 Kwh) y el valor a cancelar para ese valor de consumo, 1.00 dólares.

5.1.3 Visualización de la Información

Se puede visualizar la información en cualquier navegador web, ya sea desde la red LAN o desde el Internet. En caso de querer acceder desde la red LAN, basta con ingresar la dirección IP privada asignada al dispositivo más el número de puerto, por ejemplo: 192.168.0.4:1201.

En la figura 5.10 se observa la interfaz gráfica web.

Figura 5. 10. Interfaz web.
(Fuente: Future Electronics, 2018)

En caso de querer acceder desde el Internet, es preciso conocer la IP pública asignada por el proveedor de servicio más el número de puerto, por ejemplo 200.63.215.234:1201. Se visualizará la interfaz mostrada en la Figura 5.10.

Como se puede observar existen dos botones: ACTIVAR SERVICIO para el caso que se desee suspender el servicio eléctrico y CORTAR SERVICIO para reanudar el servicio eléctrico.

5.1.4 Características Técnicas

En la Tabla 5.1 se observan las características técnicas básicas.

Voltaje de entrada	120 (V_{AV})
Voltaje de salida	120 (V_{AV})
Corriente máxima	20 (A_{AV})
Fusible de protección	2 (A)
Red WiFi	802.11

Tabla 5. 1. Características técnicas básicas.
(Fuente: Autoría propia)

5.2 ANEXO B: Código del Microcontrolador

```
#include <SoftwareSerial.h>
#include <ArduinoJson.h>
#include <LiquidCrystal.h>
SoftwareSerial s(5,6); //Rx, Tx
int RS=13, EN=12,d4=11,d5=10,d6=9,d7=8;
LiquidCrystal lcd(RS, EN, d4, d5, d6, d7);
int ana0=0;
int ana1=0;
int ana0_max=0;
int ana1_max=0;
int cuenta=0;
int aux=0;
float V1;
float V2;
float V3;
float V4;
float V5;
float V=0;
float I=0;
float th=0;
float S=0;
float P=0;
float Q=0;
float fp=0;

void setup() {
  s.begin(9600);
  Serial.begin(9600);

  attachInterrupt(digitalPinToInterrupt(2),int_volt, FALLING);
  attachInterrupt(digitalPinToInterrupt(3),int_cor, FALLING);
  lcd.begin(20, 4);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("V:");
  lcd.setCursor(0,1);
  lcd.print("I:");
  lcd.setCursor(0,2);
  lcd.print("th:");
  lcd.setCursor(10,0);
  lcd.print("S:");
  lcd.setCursor(10,1);
  lcd.print("P:");
  lcd.setCursor(10,2);
  lcd.print("Q:");
  lcd.setCursor(10,3);
  lcd.print("fp:");
  TCCR2A=0b00000000;
  TCCR2B=0b00000111;
  TIMSK2=0b00000010;
  TCNT2=0;
  TCCR1A=0;
  TCCR1B=0b00000010;
}

void int_volt(){
  TCNT1=0;
```

```
}

void int_cor(){
  cuenta=TCNT1;
}

ISR(TIMER2_COMPA_vect)
{
  if(aux==10)
  {
    aux=0;
    calcular();
    presentar();
    ana0_max=0;
    ana1_max=0;
  }
  else
  {
    aux=aux+1;
  }
  TCNT2=0;
}

void calcular(){
  V=(V1+V2+V3+V4+V5)/5;
  I=ana1_max*0.022447;
  th=cuenta*0.0108;
  S=V*I;
  fp=cos(th*0.01745);
  P=V*I*cos(th*0.01745);
  Q=V*I*sin(th*0.01745);
  V1=V2;
  V2=V3;
  V3=V4;
  V4=V5;
  V5=ana0_max*0.23327;

}

void presentar(){
  StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.createObject();
  lcd.setCursor(3,0);
  lcd.print(V);
  lcd.setCursor(3,1);
  lcd.print(I);
  lcd.setCursor(4,2);
  lcd.print(th);
  lcd.setCursor(13,0);
  lcd.print(S);
  lcd.setCursor(13,1);
  lcd.print(P);
  lcd.setCursor(13,2);
  lcd.print(Q);
  lcd.setCursor(14,3);
  lcd.print(fp);
  Serial.print("VOLTAJE:   ");
  Serial.println(V);
  Serial.print("CORRIENTE:   ");
  Serial.println(I);
```

```
  Serial.print("FP:   ");
  Serial.println(fp);

  root["voltaje"] = V;
  root["corriente"] = I;
  root["S"] = S;
  root["P"] = P;
  root["Q"] = Q;
  root["fp"] = abs(fp);
  float prueba = 0.1234;
  root["prueba"] = prueba;
  root.printTo(s);

}

void loop() {
  ana0=analogRead(0);
  ana1=analogRead(1);
  if (ana0_max<ana0)
  {
    ana0_max=ana0;
  }
    if (ana1_max<ana1)
  {
    ana1_max=ana1;
  }
}
```

5.3 ANEXO C: Código del Módulo WiFi ESP8266

```
#include <EEPROM.h>
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <WiFiManager.h>
#include <LiquidCrystal.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <TimeLib.h>
#include "index.h"
#define LED D10
#define selec D8
int anterior1=0, actual1=0, anterior2=0, actual2=0, compara1=0, compara2=0,
imprimir1=0, imprimir2=0, voltaje=0, mesCorte=0, seconds=0, secondsAnte=0,
secondsAct=0;
float corriente=0.00, fp=1.00, S=0.00, P=0.00, Q=0.00, energia=0.0,
consumoW, consumoWAnt=0.0, aPagar=0;
const char* meses[14] = {"hola", "ENERO", "FEBRERO", "MARZO", "ABRIL",
"MAYO", "JUNIO", "JULIO", "AGOSTO", "SEPTIEMBRE", "OCTUBRE", "NOVIEMBRE",
"DICIEMBRE", "ENERO"};
#include <SoftwareSerial.h>
SoftwareSerial s(14,12); //Rx, Tx --> A5 Arduino con 12 ESP --- A6 Arduino
con 14 ESP ...........
#include <ArduinoJson.h>
LiquidCrystal lcd(5, 4, 16, 2, 13 ,0);
IPAddress ip(192,168,0,222);
IPAddress gateway(192,168,0,1);
IPAddress subnet(255,255,255,0);
ESP8266WebServer server(1201); //Server on port 80
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "time.windows.com", -18000, 60000);

void handleRoot() {
 String s = MAIN_page; //Read HTML contents
 server.send(200, "text/html", s); //Send web page
}
void handleADC() {
 int a = analogRead(A0);
 String adcValue = String(9991);
 server.send(200, "text/plane", String("ALEJANDRO"));
}
void handleFecha() {
  timeClient.update();
  delay(1);
  seconds = timeClient.getEpochTime();
  String fecha = String(day(seconds)) + " de " + meses[month(seconds)] + "
de " + year(seconds);
  server.send(200, "text/plane", fecha);
}
void handleFechaCorte() {
  timeClient.update();
  delay(1);
  seconds = timeClient.getEpochTime();
  int indice = int(month(seconds))+1;
  String fechaCorte = "5 de " + String(meses[indice]) + " de " +
String(year(seconds));
  server.send(200, "text/plane", fechaCorte);
}
```

```
void handleLED() {
 String ledState = "OFF";
 String t_state = server.arg("LEDstate");
 if(t_state == "1")
 {
  digitalWrite(LED,LOW);
  ledState = "ACTIVO";
 }
 else
 {
  digitalWrite(LED,HIGH);
  ledState = "DESACTIVADO";
 }
 server.send(200, "text/plane", ledState);
}
void handleVoltaje(){
  server.send(400, "text/plane", String(voltaje));
}
void handleCorriente(){
  server.send(200, "text/plane", String(corriente));
}
void handleFP(){
  server.send(200, "text/plane", String(fp));
}
void handleConsumo(){
  energia = voltaje*corriente*fp/1000;
  consumoW = consumoW + energia;
  consumoWAnt = energia
  EEPROM.write(350, int(consumoW));
      server.send(200, "text/plane", String(consumoW));
}
void handleValor(){
  //aPagar = float(char(EEPROM.read(350))*7.95);
  aPagar = consumoW*0.0795;
  server.send(200, "text/plane", String(aPagar));
}
void setup(void){
  timeClient.begin();
  Serial.begin(115200);
  EEPROM.begin(512);
  s.begin(9600);
  while (!Serial) continue;
  Serial.println("serial 115200");
  WiFiManager wifiManager;
  wifiManager.resetSettings();
  lcd.begin(16,2);
  lcd.clear();
  pinMode(15, INPUT);  //D8
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("BIENVENIDO !!");
  delay(10000);
  if (digitalRead(15) == 1) {
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MODO");
    lcd.setCursor(0,1);
    lcd.print("CONFIGURACION");
    delay(3000);
    lcd.clear();
```

```
    lcd.setCursor(0,0);
    lcd.print("CONFIGURANDO RED");
    lcd.setCursor(0,1);
    lcd.print(".............");
    delay(2000);
    wifiManager.autoConnect("Alejandro_Temp");
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("CONECTADO A:");
    lcd.setCursor(0,1);
    lcd.print(WiFi.SSID());
    delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("IP:");
    lcd.setCursor(0,1);
    lcd.print(ip);
    delay(3000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Puerto:");
    lcd.setCursor(0,1);
    lcd.print(1201);
    delay(3000);
    WiFi.config(ip, gateway, subnet);
    if(WiFi.SSID().length() < 10){
      EEPROM.write(505, WiFi.SSID().length());
      EEPROM.write(506, 111);
    } else {
        EEPROM.write(506, int(WiFi.SSID().length()/10));
        EEPROM.write(505, WiFi.SSID().length() -
int(WiFi.SSID().length()/10));
    }
    for(int n=0; n < WiFi.SSID().length(); n++){
      EEPROM.write(n+50, WiFi.SSID()[n]);
    }
      EEPROM.commit();
  } else {
    int Longitud = 0, longi = 0;
    String Pwd = "";
    Longitud = EEPROM.read(501) - 47;
    //Serial.println(Longitud);
    if(Longitud > 0 && Longitud < 11){
      longi = (Longitud - 1)*10 + EEPROM.read(500) - 48;
    } else {
        longi = EEPROM.read(500);
    }
    for(int n=0; n <= longi - 1; n++){
      //Serial.print(char(EEPROM.read(n)));
      Pwd = Pwd + char(EEPROM.read(n));
    }
    String SSIDprom = "";
    int SSIDpromLen = 0;
    if(EEPROM.read(506) == 111){
      SSIDpromLen = EEPROM.read(505);
    } else {
      SSIDpromLen = EEPROM.read(505) + EEPROM.read(506)*10;
    }
    for(int n=0; n < SSIDpromLen; n++){
      //Serial.print(char(EEPROM.read(n+50)));
```

```
      SSIDprom = SSIDprom + char(EEPROM.read(n+50));
    }
    WiFi.begin(SSIDprom, Pwd);
    WiFi.config(ip, gateway, subnet);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("CONECTADO A:");
    lcd.setCursor(0,1);
    lcd.print(WiFi.SSID());
    delay(3000);
  }
  pinMode(LED, OUTPUT);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  server.on("/", handleRoot);
  server.on("/setLED", handleLED);
  server.on("/readADC", handleADC);
  server.on("/readFecha", handleFecha);
  server.on("/readFechaCorte", handleFechaCorte);
  server.on("/readCorriente", handleCorriente);
  server.on("/readFP", handleFP);
  server.on("/readConsumo", handleConsumo);
  server.on("/readValor", handleValor);
  server.on("/readVoltaje", handleVoltaje);
  server.begin();
}


void loop(void){
  server.handleClient();          //Handle client requests
  float cada = voltaje*corriente*fp;
Serial.println("INICIO JSON");
 StaticJsonBuffer<1000> jsonBuffer;
  JsonObject& root = jsonBuffer.parseObject(s);
  if (root == JsonObject::invalid())
    return;
  int data1=root["data1"];
  int data2=root["data2"];
  voltaje = root["Voltaje"];
  corriente = root["Corriente"];
  fp = root["FP"];
  float ggg = root["voltaje"];
lcd.clear();
lcd.setCursor(0,0);
lcd.print("V:");
lcd.setCursor(2,0);
lcd.print(voltaje);
lcd.setCursor(6,0);
lcd.print("I:");
lcd.setCursor(8,0);
lcd.print(corriente);
lcd.setCursor(0,1);
lcd.print("kwh:");
lcd.setCursor(4,1);
lcd.print(consumoW);
lcd.setCursor(9,1);
lcd.print("$:");
lcd.setCursor(11,1);
lcd.print(aPagar);
```

```
if(EEPROM.read(350)==255){
    EEPROM.write(350, 0);
  }
anterior1 = data1;
anterior2 = data2;
delay(3);
}
```

5.4 ANEXO D: Código de la Interfaz Web

```
<html lang="en">
<head>
        <title>Medidor ESFOT</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
</head>
<style>
 {
        margin: 0px;
        padding: 0px;
        box-sizing: border-box;
}
body, html {
        height: 100%;
        font-family: sans-serif;
}
a {
        margin: 0px;
        transition: all 0.4s;
        -webkit-transition: all 0.4s;
  -o-transition: all 0.4s;
  -moz-transition: all 0.4s;
}
a:focus {
        outline: none !important;
}
a:hover {
        text-decoration: none;
}
h1,h2,h3,h4,h5,h6 {margin: 0px;}
p {margin: 0px;}
ul, li {
        margin: 0px;
        list-style-type: none;
}
input {
  display: block;
        outline: none;
        border: none !important;
}
textarea {
  display: block;
  outline: none;
}
textarea:focus, input:focus {
  border-color: transparent !important;
}
button {
        outline: none !important;
        border: none;
        background: transparent;
}
button:hover {
        cursor: pointer;
}
iframe {
        border: none !important;
```

```
}
.limiter {
  width: 100%;
  margin: 0 auto;
}
.container-table100 {
  width: 100%;
  min-height: 100vh;
  background: #c4d3f6;
  display: -webkit-box;
  display: -webkit-flex;
  display: -moz-box;
  display: -ms-flexbox;
  display: flex;
  align-items: center;
  justify-content: center;
  flex-wrap: wrap;
  padding: 33px 30px;
}
.wrap-table100 {
  width: 960px;
  border-radius: 10px;
  overflow: hidden;
}
.table {
  width: 100%;
  display: table;
  margin: 0;
}
@media screen and (max-width: 768px) {
  .table {
    display: block;
  }
}
.row {
  display: table-row;
  background: #fff;
}
.row.header {
  color: #ffffff;
  background: #6c7ae0;
}
@media screen and (max-width: 768px) {
  .row {
    display: block;
  }
  .row.header {
    padding: 0;
    height: 0px;
  }
  .row.header .cell {
    display: none;
  }
  .row .cell:before {
    font-family: Poppins-Bold;
    font-size: 12px;
    color: #808080;
    line-height: 1.2;
    text-transform: uppercase;
    font-weight: unset !important;
```

```
    margin-bottom: 13px;
    content: attr(data-title);
    min-width: 98px;
    display: block;
  }
}
.cell {
  display: table-cell;
}
@media screen and (max-width: 768px) {
  .cell {
    display: block;
  }
}
.row .cell {
  font-family: Poppins-Regular;
  font-size: 15px;
  color: #666666;
  line-height: 1.2;
  font-weight: unset !important;

  padding-top: 20px;
  padding-bottom: 20px;
  border-bottom: 1px solid #f2f2f2;
}
.row.header .cell {
  font-family: Poppins-Regular;
  font-size: 18px;
  color: #fff;
  line-height: 1.2;
  font-weight: unset !important;
  padding-top: 19px;
  padding-bottom: 19px;
}
.row .cell:nth-child(1) {
  width: 360px;
  padding-left: 40px;
}
.row .cell:nth-child(2) {
  width: 160px;
}
.row .cell:nth-child(3) {
  width: 250px;
}
.row .cell:nth-child(4) {
  width: 190px;
}
.table, .row {
  width: 100% !important;
}
.row:hover {
  background-color: #ececff;
  cursor: pointer;
}

@media (max-width: 768px) {
  .row {
    border-bottom: 1px solid #f2f2f2;
    padding-bottom: 18px;
```

```
    padding-top: 30px;
    padding-right: 15px;
    margin: 0;
  }
  .row .cell {
    border: none;
    padding-left: 30px;
    padding-top: 16px;
    padding-bottom: 16px;
  }
  .row .cell:nth-child(1) {
    padding-left: 30px;
  }
  .row .cell {
    font-family: Poppins-Regular;
    font-size: 18px;
    color: #555555;
    line-height: 1.2;
    font-weight: unset !important;
  }
  .table, .row, .cell {
    width: 100% !important;
  }
}
.button {
  display: inline-block;
  background-color: #008CBA;
  border: none;
  border-radius: 4px;
  color: white;
  padding: 16px 40px;
  text-decoration: none;
  font-size: 30px;
  margin: 2px;
  cursor: pointer;
}
.button2 {
  background-color: #f44336;
}
</style>
<body>
        <div class="limiter">
        <div class="container-table100">
        <div class="wrap-table100">
        ESCUELA POLITECNICA NACIONAL
        <br> </br>
        <div class="table">
                ESCUELA DE FORMACION DE TECNOLOGOS
                <br> </br>
        <!-- <br> </br>
        <br> </br>
        <br> </br>
        <br> </br>
        <br> </br>
                <br> </br> -->
        </div>
        ELECTRONICA Y TELECOMUNICACIONES
        <br> </br>
        <div class="table">
```

```
                <div class="row header">
                        <div class="cell">
                                DESCRIPCION
                        </div>
                        <div class="cell">
                                VALOR
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Full Name">
                                NOMBRE
                        </div>
                        <div class="cell" data-title="Age">
                                <span id="ADCValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Fecha">
                                FECHA ACTUAL
                        </div>
                        <div class="cell" data-title="Valor Fecha">
                                <span id="FechaValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Fecha corte">
                                FECHA DE CORTE
                        </div>
                        <div class="cell" data-title="Valor fecha corte">
                                <span id="FechaCorteValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="voltaje">
                                VOLTAJE
                        </div>
                        <div class="cell" data-title="valor voltaje">
                                <span id="voltajeValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Corriente">
                                CORRIENTE
                        </div>
                        <div class="cell" data-title="Valor corriente">
                                <span id="corrienteValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Factor Potencia">
                                Factor de Potencia
                        </div>
                        <div class="cell" data-title="Valor Factor
Potencia">
                                <span id="FPValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Consumo">
                                CONSUMO HASTA LA FECHA
```

```
                        </div>
                        <div class="cell" data-title="Valor consumo">
                                <span id="consumoValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Pagar">
                                VALOR A PAGAR
                        </div>
                        <div class="cell" data-title="Valor pagar">
                                <span id="valorValue">0</span>
                        </div>
                </div>
                <div class="row">
                        <div class="cell" data-title="Estado">
                                ESTADO DEL SERVICIO
                        </div>
                        <div class="cell" data-title="Valor Estado">
                                <span id="LEDState">0</span>
                        </div>
                </div>
        </div>
        <div id="demo">
                <p><button class="button" onclick="sendData(1)">ACTIVAR
SERVICIO</button></a></p>
                <p><button class="button button2"
onclick="sendData(0)">CORTAR SERVICIO</button></a></p>
        </div>
        </div>
        </div>
        </div>
        <script>
                setInterval(function() {
                  getData();
                }, 250);
                setInterval(function() {
                  getData2();
                }, 250);
                setInterval(function() {
                  getData3();
                }, 250);
                setInterval(function() {
                  getData4();
                }, 250);
                setInterval(function() {
                  getData5();
                }, 250);
                setInterval(function() {
                  getData6();
                }, 250);
                setInterval(function() {
                  getData7();
                }, 250);
                setInterval(function() {
                  getData8();
                }, 250);
                setInterval(function() {
                  getData9();
                }, 250);
}
```

```
                function getData() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {
                          document.getElementById("ADCValue").innerHTML =
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readADC", true);
                  xhttp.send();
                }
                function getData2() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {
                          document.getElementById("FechaValue").innerHTML =
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readFecha", true);
                  xhttp.send();
                }
                function getData3() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {

document.getElementById("FechaCorteValue").innerHTML =
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readFechaCorte", true);
                  xhttp.send();
                }
                function getData4() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 400) {
                          document.getElementById("voltajeValue").innerHTML
=
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readVoltaje", true);
                  xhttp.send();
                }
                function getData5() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {

document.getElementById("corrienteValue").innerHTML =
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readCorriente", true);
                  xhttp.send();
                }
```

```
                function getData6() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {
                          document.getElementById("FPValue").innerHTML =
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readFP", true);
                  xhttp.send();
                }
                function getData7() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {
                          document.getElementById("consumoValue").innerHTML
=
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readConsumo", true);
                  xhttp.send();
                }
                function getData8() {
                  var xhttp = new XMLHttpRequest();
                  xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {
                          document.getElementById("valorValue").innerHTML =
                          this.responseText;
                        }
                  };
                  xhttp.open("GET", "readValor", true);
                  xhttp.send();
                }
                function sendData(led) {
                        var xhttp = new XMLHttpRequest();
                        xhttp.onreadystatechange = function() {
                        if (this.readyState == 4 && this.status == 200) {
                                document.getElementById("LEDState").innerHTML
=
                                this.responseText;
                        }
                        };
                        xhttp.open("GET", "setLED?LEDstate="+led, true);
                        xhttp.send();
                }
                </script>
</body>
</html>
```