

**SISTEMA DE GESTIÓN DE ENERGÍA ELÉCTRICA PARA EL SECTOR
RESIDENCIAL A TRAVÉS DE UNA APLICACIÓN MÓVIL.**



**JEISON SNEIDER RUIZ ZEMANATE
PABLO ALEXANDER VIDAL ARARAT**

**CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
GESTIÓN ENERGÉTICA
POPAYÁN
2019**

**SISTEMA DE GESTIÓN DE ENERGÍA ELÉCTRICA PARA EL SECTOR
RESIDENCIAL A TRAVÉS DE UNA APLICACIÓN MÓVIL.**



**JEISON SNEIDER RUIZ ZEMANATE
PABLO ALEXANDER VIDAL ARARAT**

**Trabajo de grado presentado para optar al título de
INGENIERO ELECTRÓNICO**

**Director
M. Sc. ING JULIO ANDRES MOSQUERA BOLAÑOS**

**CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
GESTIÓN ENERGÉTICA
POPAYÁN
2019**

NOTA DE ACEPTACIÓN

El director y los jurados del trabajo de grado titulado “**Sistema de Gestión de Energía Eléctrica para el Sector Residencial a través de una Aplicación Móvil**”, realizado por los estudiantes **Jeison Sneider Ruiz Zemanate** y **Pablo Alexander Vidal Ararat**. Una vez revisado el informe final y aprobado la sustentación, autorizan para que se hagan los trámites concernientes para optar por el título profesional de **Ingeniero Electrónico**.

M. Sc. ING JULIO ANDRES MOSQUERA BOLAÑOS
Director de trabajo de grado

Jurado 1

Jurado 2

Popayán, abril

AGRADECIMIENTOS

Primeramente, queremos agradecer a Dios por habernos acompañado y guiado a lo largo de nuestra carrera, por darnos la fortaleza necesaria para superar los obstáculos y dificultades, por brindarnos una vida llena de aprendizaje, experiencias y sobre todo felicidad.

A nuestros padres y familia, por los valores que nos han inculcado, por el amor, la confianza y el apoyo brindado en todo momento; por ser nuestro ejemplo para salir adelante y siempre hacer lo correcto y por habernos dado la oportunidad de tener una excelente educación en el transcurso de nuestra vida.

A los ingenieros, Julio Mosquera y Juan Pablo Diago por su apoyo y capacidad para guiar nuestras ideas, por la orientación y motivación para la elaboración de este trabajo de grado y la culminación satisfactoria de nuestra carrera profesional.

Al cuerpo de docentes de la Facultad de ingenierías de la Corporación Universitaria Autónoma del Cauca por sus enseñanzas, exigencias y motivaciones para impulsarnos en el desarrollo de nuestra formación profesional.

Finalmente queremos agradecer a nuestros compañeros y amigos que estuvieron con nosotros en el transcurso de la carrera, especialmente a Ingrid Rodríguez y Mónica Casas por su amistad y apoyo incondicional.

DEDICATORIA

A Dios por ser el pilar fundamental de nuestra vida, por permitirnos llegar a este momento tan especial, y por haber puesto en nuestro camino personas valiosas que nos apoyaron durante nuestra formación profesional.

A nuestros padres que con su amor, paciencia y esfuerzo nos han enseñado a no desfallecer y a siempre perseverar a través de sus sabios consejos.

RESUMEN

El siguiente trabajo de investigación tiene como propósito implementar un sistema electrónico que permita monitorear el consumo de electricidad de los hogares a través de una aplicación móvil. Este se diseñó a base de productos comerciales de fácil consecución que establecen la medición, comunicación y transmisión de datos de las cargas eléctricas domiciliarias y las presenta en un dispositivo móvil mediante una página web o aplicación, proporcionando información sobre el consumo que se da en horas, días y semanas obteniendo resultados similares a los que presenta la factura suministrada por la empresa prestadora del servicio de energía eléctrica.

Palabras claves: Eficiencia energética, consumo eléctrico, ahorro energético, servicios web, Android Studio, aplicación móvil.

ABSTRACT

The following research work aims to implement an electronic system that allows monitoring of the electricity in homes through a mobile application. This is a base of commercial products of easy handling of the communication, communication and transmission of data of the electric and domiciliary charges and the presentation in a mobile device through a web page or an application, information about the that occurs in hours. days and weeks get similar results.

Keywords: Energy efficiency, electric consumption, energy saving, web services, Android Studio, mobile app.

TABLA DE CONTENIDO

INTRODUCCION	15
CAPITULO I	17
1. PROBLEMA	17
1.1. PLANTEAMIENTO DEL PROBLEMA	17
1.2. JUSTIFICACIÓN	18
1.3. OBJETIVOS	19
1.3.1. Objetivo General	19
1.3.2. Objetivos específicos	19
1.4. ORGANIZACIÓN DEL DOCUMENTO	19
CAPITULO II	21
2. MARCO TEÓRICO	21
2.1. ANTECEDENTES	21
2.1.1. Núcleo Temático de Medición	21
2.1.2. Núcleo Temático de Aplicación Móvil	23
2.1.3. Núcleo Temático Plan Energético	25
2.2. BASES TEORICAS	26
2.2.1. Plataformas hardware	26
2.2.2. Módulo Wi-Fi NodeMcu	29
2.2.3. Medidores de energía eléctrica	29
2.2.4. Servicio Web	36
2.2.5. Servidor Web	36
2.2.6. Protocolo de transferencia de hipertexto	37
2.2.7. Protocolos de comunicación inalámbricos	37
2.2.8. PHP (Preprocesador de hipertexto)	39
2.2.9. JSON	39
2.2.9. Plataformas software	40
2.2.10. Stand-by	41
2.3. BASES LEGALES	43

2.3.1. Legislación Colombiana.....	44
CAPITULO III	46
3. METODOLOGÍA.....	46
3.1. FASE PREPARATORIA	46
3.2. FASE DESCRIPTIVA.....	46
3.3. FASE INTERPRETATIVA POR NÚCLEOS TEMÁTICOS.....	47
3.4. FASE DE CONSTRUCCIÓN TEÓRICA GLOBAL	48
3.4. FASE DE EXTENSIÓN Y PUBLICACIÓN	49
CAPITULO IV.....	50
4. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA	50
4.1. DISEÑO DE LOS SISTEMAS	50
4.1.1. Medición.	52
4.1.1.1. Módulo RS485	¡Error! Marcador no definido.
4.1.1.2. Protocolo I2C	56
4.1.1.3. Librería Wire.....	57
4.1.1.4. Programación OTA (Over the Air)	61
4.1.1.5. IFastNet.....	65
4.1.1.6. Base de Datos MySQL.....	66
4.1.1.7. Lógica del servidor	67
4.1.2. Conexión a Servicios Web.....	68
4.1.2.1. CloudMQTT.....	68
4.1.2.2. WebSocket.....	71
4.1.2.3. Thingspeak.....	72
4.1.2.4. IFTTT	74
4.1.2.5. Google Spreadsheets (Hojas de cálculo de Google)	77
4.1.3. Aplicación Móvil.	77
4.1.4. Plan Energético.....	91
4.1.4.1. Firebase	92
4.1.5. Página y Servicio Web.....	94
4.2. DESARROLLO HARDWARE.....	96

CAPITULO V	100
5. RESULTADOS	100
CAPITULO VI	112
6. CONCLUSIONES Y RECOMENDACIONES	112
6.1. CONCLUSIONES	112
6.2. RECOMENDACIONES	113
7. BIBLIOGRAFIA	115

LISTA DE TABLAS

Tabla 1-2: Especificaciones Generales del Medidor Monofásico ED610D2.	30
Tabla 2-2. Especificaciones Generales del Medidor de Potencia PM1200.	31
Tabla 2-3. Entorno del Medidor de Potencia PM1200.	32
Tabla 2-4. Especificaciones Generales del Medidor Eastron sdm230.	34
Tabla 2-5. Exactitud del Medidor Eastron sdm230.	35
Tabla 2-6. Entorno del Medidor Eastron sdm230.	35
Tabla 2-7. Consumo en stand-by de Aparatos electrónicos.	41
Tabla 2-8. Bases legales.	43
Tabla 4-1. Características del módulo Rs485.	53
Tabla 5-1. Mediciones del Sistema principal.	101
Tabla 5-2. Mediciones del Sistema portátil.	102

LISTA DE FIGURAS

Figura 2-1: Metodología de Desarrollo.....	24
Figura 2-2: Arquitectura de Android.....	25
Figura 2-3: Raspberry Pi 2 modelo B.....	27
Figura 2-4: Arduino mega 2560.....	28
Figura 2-5: Módulo Wi-Fi NodeMcu.....	29
Figura 2-6: Medidor Monofásico ED610D2.....	30
Figura 2-7: Medidor de Potencia PM1200.....	31
Figura 2-8: Medidor Eastron sdm230.....	34
Figura 2-9: Funcionamiento del protocolo HTTP.....	37
Figura 2-10: Consumo de Electrodomésticos según Tipo de Equipamiento.....	42
Figura 2-11: Potencias en stand-by de los principales equipos domésticos.....	43
Figura 4-1: Diagrama de bloques del diseño de sistemas desarrollados..	51
Figura 4-2: Módulo Rs485 a Serial Ttl.....	53
Figura 4-3: Transmisión Diferencial en comunicación Modbus por pines A y B para eliminación de ruido.....	55
Figura 4-4: Esquema de conexión Arduino mega y medidor Eastron SDM230....	55
Figura 4-5: Conexión entre maestro y esclavos.....	57
Figura 4-6: Comunicación I2C entre Nodemcu ESP8266 (Master Device) y Arduino Mega2560 (Slave Device).....	58
Figura 4-7: Diagrama de flujo código Arduino Mega2560 Dispositivo Esclavo.....	59
Figura 4-8: Esquema Clientes – Servidor Externo.....	60
Figura 4-9: Diagrama de flujo código Nodemcu Dispositivo Maestro.....	62
Figura 4-10: Medición de Corriente no invasiva y frecuencia con dispositivo.....	63
Figura 4-11: Diagrama de flujo conexión a servidor externo IFastNet. Nodemcu (Master Device).....	65
Figura 4-12: Parámetros eléctricos Nodemcu, BD MYSQL. IFasNet.....	66
Figura 4-13: Estructura Base de Datos MySQL.....	67
Figura 4-14: Petición a Servidor IFasNet y BD.....	68
Figura 4-15: Broker MQTT.....	69
Figura 4-16: Diagrama de flujo conexión a servidor CloudMQTT. Nodemcu (Master Device).....	70
Figura 4-17: Topic, Broker MQTT y clientes publicación/suscripción.....	71
Figura 4-18: Gráfica de Corriente en ThingSpeak.....	73
Figura 4-19: Gráfica de Potencia Activa en ThingSpeak.....	73
Figura 4-20: Gráfica de Energía y Potencia Reactiva en ThingSpeak.....	74
Figura 4-21: Creación de nueva receta IFTTT.....	75
Figura 4-22: Configuración ThingHTTP con URL de Webhook IFTTT.....	76

Figura 4-23: Hoja de cálculo. Google Spreadsheets.	77
Figura 4-24: Estructura carpeta Java Proyecto. Android Studio.....	79
Figura 4-25: Layout activity_login.xml. Android Studio.....	80
Figura 4-26: Diagrama de flujo Activity Login.java.....	81
Figura 4-27: Diagrama de flujo MainActivity.java. Android Studio.	83
Figura 4-28: Layout Main Activity (Navigation drawer). Android Studio.....	84
Figura 4-29: Diagrama de flujo DataFragment.java. Android Studio.	85
Figura 4-30: Diagrama de flujo CorrienteFragment.java. Android Studio.	86
Figura 4-31: Grafica de Corriente. Android Studio.	87
Figura 4-32: Diagrama de flujo Activity Menu2.java. Android Studio.....	88
Figura 4-33: Layout Fragment_menu_2.xml (Menu2.java). Android Studio.	89
Figura 4-34: Diagrama de flujo fragment Menu3.java. Android Studio.....	90
Figura 4-35: Pantalla Gestión de E.E (Menu3.java). Android Studio.....	91
Figura 4-36: Diagrama de flujo configuración Push Notifications Firebase. Android Studio	93
Figura 4-37: Página de inicio SIMEE.....	94
Figura 4-38: Página de login SIMEE.	95
Figura 4-39: Página Consultas por Fecha y hora SIMEE.....	95
Figura 4-40: Diseño PCB Sistema principal.	96
Figura 4-41: Diseño PCB Sistema portátil.....	97
Figura 4-42: Parte inferior y superior Sistema principal.....	97
Figura 4-43: Parte superior e inferior Sistema portátil.....	98
Figura 4-44: Estructura Final del Sistema principal.....	98
Figura 4-45: Estructura Sistema portatil.	99
Figura 5-1: Conexión del sensor de corriente no invasivo.....	100
Figura 5-2: Implementación de la conexión del sensor de corriente no invasivo.	101
Figura 5-3: Medición- secador, computador.	103
Figura 5-4: Aplicación móvil- secador, computador.....	103
Figura 5-5: Página web- secador, computador.....	104
Figura 5-6: Medición- nevera, secador, bombillo ahorrador, smart Tv.	104
Figura 5-7: Aplicación móvil- nevera, secador, bombillo ahorrador, smart Tv....	105
Figura 5-8: Página web- nevera, secador, bombillo ahorrador, smart Tv.	105
Figura 5-9: Medición- modem, bombillo incandescente, secador.....	106
Figura 5-10: Aplicación móvil- modem, bombillo incandescente, secador	106
Figura 5-11: Página web- modem, bombillo incandescente, secador	107
Figura 5-12: Medición de pulsos-secador.....	107
Figura 5-13: Página web-Medición de pulsos-secador.....	108
Figura 5-14: Facturación del servicio de energía eléctrica (Febrero 2019)	110

Figura 5-15: Datos de consumo eléctrico Medidor EastronSDM230(Febrero 2019).
..... 110

Figura 5-16: Facturación del servicio de energía eléctrica (Enero 2019) 111

Figura 5-17: Datos de consumo eléctrico Medidor EastronSDM230(Enero 2019).
..... 111

Figura 5-18: Facturación del servicio de energía eléctrica (Diciembre 2019)..... 112

Figura 5-19: Datos de consumo eléctrico Medidor EastronSDM230 (Diciembre
2019)..... 113

LISTA DE ANEXOS

Anexo 1. Link del Manual de usuario del Medidor Monofásico ED610D2.	118
Anexo 2. Link del Manual de usuario del Medidor de Potencia PM1200.	118
Anexo 3. Link del Manual de usuario del Medidor Eastron sdm230.	118
Anexo 4. Librerías Arduino Mega2560	118
Anexo 5. Código Arduino mega 2560	118
Anexo 6. Librerías Nodemcu ESP8266	129
Anexo 7. Configuración de variables para Nodemcu ESP8266.....	129
Anexo 8. Envío de variables a BD MYSQL de IFasNet (archivo <i>enviardatos.php</i>)	130
Anexo 9. Estructura HTML-conexión con WebSocket CloudMQTT	130
Anexo 10. Script conexión a topic CloudMQTT	131
Anexo 11. Configuración-Reacción ThingSpeak	131
Anexo 12. Estructura archivo AndroidManifest.xml. Android Studio	132
Anexo 13. Estructura archivo AndroidManifest.xml. Android Studio	134
Anexo 14. Código XML Layout activity_login.xml. Android Studio	135
Anexo 15. Código XML Layout Fragment_menu_2.xml (Menu2.java). Android Studio	135

INTRODUCCIÓN

La energía eléctrica siempre ha sido un tema importante en el desarrollo de los países, ya que además de posibilitar el desarrollo industrial (utilización de nuevas tecnologías que requieren de electricidad para su funcionamiento) el cual aporta a la economía de la sociedad, esta conforma uno de los factores que permite el acceso a los servicios de telecomunicaciones que facilitan la transmisión de conocimientos y la educación a distancia; asimismo suministra recursos para la generación e implementación de proyectos tecnológicos en los diferentes sectores productivos que conllevan a mejorar e innovar procesos, productos y servicios haciendo cada vez más que la calidad de vida de las personas aumente.

En la ley 143 de la legislación colombiana se establece alcanzar una cobertura en los servicios de electricidad a las diferentes regiones y sectores del país, que garantice la satisfacción de las necesidades básicas de los usuarios, puesto que este servicio está relacionado con el bienestar de las personas y condiciones elementales de comodidad, varias de las actividades de la vida cotidiana que hoy se dan por dadas y parecen naturales solo pueden llevarse a cabo por el acceso a las redes de energía eléctrica; participar de la riqueza económica, cultural, informática, vivir en un espacio con la adecuada calefacción, conservar y refrigerar los alimentos, entre otras.

Atendiendo estas consideraciones, este proyecto se plantea con la finalidad de realizar el monitoreo del consumo energético y control de cargas en los hogares de una manera sencilla y practica por medio de una aplicación móvil, la cual permita al usuario del sector residencial realizar las siguientes tareas:

- Ver las mediciones de los parámetros eléctricos de las cargas a través de gráficas que permiten interpretar mejor determinada información, haciéndola más entendible y proporcionando una idea generalizada de los resultados.
- Realizar consultas de los datos en diferentes fechas y horas del día.
- Enterarse de cómo ahorrar energía eléctrica por medio de consejos e imágenes explicativas.
- Recibir notificaciones push (mensajes instantáneos) de noticias o nuevos consejos y tips de ahorro energético.
- Visualizar el estado y mediciones de los sistemas de gestión portátiles que tenga asociados.
- Consultar el gasto monetario en pesos (\$) en relación al consumo en kWh.

- Proporcionar el control del encendido o apagado de las cargas que estén conectadas a las salidas relé de los sistemas desarrollados.

CAPITULO I: PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

La preocupación de los países desarrollados por la seguridad del suministro energético, el cambio climático y la producción de dióxido de carbono (CO₂) como aportante principal a la concentración atmosférica de gases de efecto invernadero, ha hecho que se presente la necesidad de incorporar elementos de competencia en la entrega y manejo de los servicios energéticos hacia una mayor eficiencia [1]. La Eficiencia Energética (EE) es el conjunto de acciones que permiten mejorar la relación entre la cantidad de energía utilizada para producir una cantidad determinada de productos y servicios finales, además se encarga de asegurar el abastecimiento energético, puesto que se sustenta en la adopción de nuevas tecnologías y buenos hábitos de consumo, con el fin de mejorar el manejo y uso de los recursos energéticos disponibles. Por lo tanto, se hace necesario implementar las políticas y estrategias contempladas en la ley 697 de 2001 y en el Plan de Acción Indicativo de Eficiencia Energética 2017 - 2022, para lo cual se requiere enfocar la forma como los usuarios finales gestionan el uso de la energía sin disminuir su calidad de vida. [1], [2], [3].

El ahorrar energía eléctrica equivale a disminuir el uso de los combustibles fósiles (fuentes susceptibles de agotamiento y que deterioran el medio ambiente) para la generación de electricidad, evitando de esta manera la emisión de gases contaminantes hacia la atmósfera; ahorrar y usar eficientemente la energía eléctrica así como contribuir con el cuidado del medio ambiente, no son sinónimo de sacrificar o reducir el nivel de bienestar o grado de satisfacción de las necesidades cotidianas, por el contrario, un cambio de hábitos y actitudes pueden favorecer una mayor eficiencia en el uso de este servicio, el empleo racional de los recursos, la protección de la economía familiar y la preservación del entorno natural [3].

Actualmente en Colombia los contadores sólo ofrecen un resumen pormenorizado del consumo general del hogar, y su objetivo es meramente administrativo: rendir cuenta a la compañía eléctrica de cuál es la factura en función del consumo de todo el hogar. A mayor consumo, mayor el importe pagado por el usuario, por lo que se hace necesario un sistema de medición de consumo de energía eléctrica mediante una aplicación móvil que muestre la electricidad que se gasta en cada momento y ayude a tomar decisiones para concienciar y, por ende, a ser más respetuosos con la electricidad, colaborando así con la reducción de la contaminación y emisión de

gases a la atmósfera, además de cooperar con el ahorro en la factura eléctrica, ayudar a controlar el gasto innecesario y aumentar la eficiencia energética.

¿Cómo podemos reducir la demanda energética en los hogares colombianos con la ayuda del control electrónico y las TIC (tecnologías de la información y comunicación)?

1.2. JUSTIFICACIÓN

Debido a que la energía eléctrica posibilita el funcionamiento y dinamismo del entorno físico, su utilidad es esencial para el desarrollo de la sociedad. En la actualidad la demanda energética de Colombia está constituida principalmente por tres sectores, residencial con un consumo de electricidad que se mantiene alrededor del 45%, seguido del industrial que se sostiene en el 29% y el comercial con el 22% de la energía eléctrica total [4]; por consiguiente es importante el uso eficiente de esta demanda, puesto que además de reducir el impacto ambiental, incrementa la competitividad del sector industrial contribuyendo al aumento del producto interno bruto del país (PIB) [5].

Cabe considerar que muchos son los beneficios tangibles de la implementación de proyectos de EE, entre los cuales se destaca la contribución a la seguridad energética y alivio de precios; mejora de la competitividad de la economía; aumento en la productividad nacional; introducción de nuevas tecnologías; creación de nuevos empleos; reducción de las facturas de energía de los usuarios finales y de los subsidios del gobierno a la energía; disminución de los costos de generación mediante la reducción de la inversión diferida en generación de energía y la mitigación de impactos ambientales en la cadena energética [5].

La implementación de un sistema de monitoreo de consumo de energía eléctrica a través de una aplicación móvil permitirá a los caucanos tomar decisiones inteligentes para ahorrar energía y dinero en su factura. Todo gracias a que el sistema les posibilitara saber exactamente cuánto consumen y su valor. También podrán monitorear, racionalizar y programar su consumo eléctrico. Además, Cabe señalar que, durante los últimos años con el desarrollo del IoT (internet de las cosas) se están implementando tecnologías para lograr la digitalización del mundo físico con el fin de mejorar la productividad y competitividad en todos los sectores productivos, estableciendo una relación entre objetos cotidianos y personas e incluso entre objetos que se conectan entre sí y proporcionan datos en tiempo real, lo que permite optimizar el uso de recursos escasos como la energía y el agua, gestionar aspectos

ambientales como los residuos, la calidad del aire y reducir emisiones de CO₂. De igual manera facilita el control de diferentes dispositivos electrónicos dentro del hogar [6], [7].

La realización de este proyecto contribuye con el grupo de investigación en tecnologías y ambiente GITA de la Corporación Universitaria Autónoma del Cauca, en aspectos que fortalecerán significativamente las líneas de ingeniería de software en cuanto a las tendencias actuales, el desarrollo de interfaces, la gestión de proyectos de software y hardware, así como también, la línea de gestión energética.

Finalmente tenemos que el aporte del sistema de gestión para el fortalecimiento de conocimiento esta dado en la mejora de los hábitos de consumo eléctrico y la reducción de demanda de electricidad, ya que el artículo 1 de la Ley 697 de 2001 establece que el uso racional y eficiente de la energía ha sido declarado como un asunto de interés social, público y de conveniencia nacional.

1.3 OBJETIVOS

1.3.1 Objetivo General

Implementar un sistema electrónico de comunicación que permita monitorear el consumo de energía eléctrica para usuarios regulados de la red de distribución secundaria por medio de una aplicación móvil.

1.3.2 Objetivos Específicos

- Implementar un sistema de comunicación para dispositivos móviles y los equipos de medida.
- Desarrollar una aplicación móvil para la gestión de la información enviada desde los equipos de medida.
- Realizar pruebas de campo que permitan establecer el comportamiento del sistema electrónico y de comunicación desarrollado.

1.4 ORGANIZACIÓN DEL DOCUMENTO

Este documento se encuentra seccionado en los siguientes capítulos:

Capítulo 1: Presenta el planteamiento del problema del proyecto de investigación, su respectiva justificación, objetivo general y objetivos específicos.

Capítulo 2: Expone todas las fases, bases legales, bases teóricas y antecedentes relacionados con los componentes temáticos para el desarrollo del proyecto.

Capítulo 3: Muestra el proceso metodológico dividido en 5 fases de desarrollo, en los cuales se encuentra las actividades propuestas para la elaboración del proyecto de investigación.

Capítulo 4: Describe los procesos técnicos y tecnológicos implementados para el desarrollo del sistema de gestión de energía eléctrica.

Capítulo 5: Expone los resultados obtenidos a partir de las pruebas de funcionamiento de las tecnologías y del sistema.

Capítulo 6: Presenta las conclusiones y recomendaciones obtenidas.

CAPITULO II: MARCO TEÓRICO

2.1. ANTECEDENTES

Para la descripción de los antecedentes se presentan los trabajos relacionados con los núcleos temáticos divididos en tres secciones de la siguiente forma: La sección 2.1.1 está relacionada con el núcleo temático de Medición, la sección 2.1.2 muestra relación con el núcleo temático de Aplicación móvil y por último, la sección 2.1.3 expone la relación con el núcleo temático de Plan Energético. A continuación, se mencionan trabajos significativos que abarcan procesos de implementación de sistemas de medición inteligente para la gestión de la energía eléctrica en el hogar. Es de gran importancia conocerlos porque estos aportan conocimientos previos y útiles para el desarrollo del mismo, en ellos se detalla las principales fases de desarrollo, componentes, características, implementación de prototipos y sistemas de medición, entre otros.

2.1.1. Antecedentes del Núcleo Temático de Medición

Betancor Caro en su investigación “Sistema doméstico de monitorización del consumo eléctrico” del año 2014 la cual se llevó a cabo en la ciudad de San Cristóbal de La Laguna - España [8], tiene como finalidad el diseño e implementación de un prototipo funcional, sencillo, de pequeño tamaño y bajo coste que permita monitorizar el consumo energético de las diferentes secciones de una instalación eléctrica y/o de diferentes dispositivos conectados a la red de suministro en el ámbito del hogar. La metodología se ajusta indicando los requisitos del proyecto (funcional y no funcional), diseño del sistema, comunicaciones del sistema (física y lógica), diseño electrónico, diseño de la plataforma web y el modelo de datos. Dicho sistema recoge la información de consumo proveniente de uno o varios sensores conectados al dispositivo. Esta información se procesa y se envía mediante un punto de acceso wifi a una base de datos alojada en la nube. Para la configuración del dispositivo y consulta de los datos almacenados se utiliza una plataforma web que se ha desarrollado de manera que se adapte a cualquier dispositivo, ordenador, tablet o móvil. Esta plataforma contiene toda la información estadística e histórica de los datos almacenados desde su puesta en funcionamiento.

En el año 2016 fue presentado en la Universidad Politécnica Salesiana sede Cuenca, el trabajo de investigación [9] que lleva como nombre “Diseño e Implementación de un Medidor de Energía Electrónico para Vivienda, con Orientación a la Prevención de Consumo y Ahorro Energético.” por los autores D. I

Samaniego Idrovo y D.F Velesaca Orellana, su objetivo es diseñar e implementar un medidor electrónico para una vivienda cuyo fin tiene la prevención de consumo y ahorro energético, monitoreando desde un sitio web o a su vez mediante un mensaje de texto el cual se lo puede consultar a través de un teléfono móvil. Este proyecto utiliza un ordenador de placa reducida (Raspberry Pi) y una plataforma Arduino. El sistema Arduino se encarga de procesar los datos provenientes de los elementos de medición, además a través de un módulo GSM puede enviar al usuario lecturas de la energía consumida y su respectivo costo en dólares. Por otra parte, el ordenador Raspberry cuenta con una base de datos que permite almacenar información en periodos de tiempo sobre el consumo energético, también incorpora un servidor web para que el usuario pueda hacer consultas de la energía consumida y la vez pueda acceder a la base de datos a través de reportes y gráficas. De acuerdo con los resultados una de las conclusiones formuladas es que con el sistema de monitoreo de consumo energético se optimiza el proceso de la lectura periódica de los medidores. Lo cual repercute directamente con la facturación mensual por servicio eléctrico puesto que se esto conllevaría a una recaudación más eficiente, reducción y/o eliminación de reclamos por errores en facturas y mayor satisfacción al cliente.

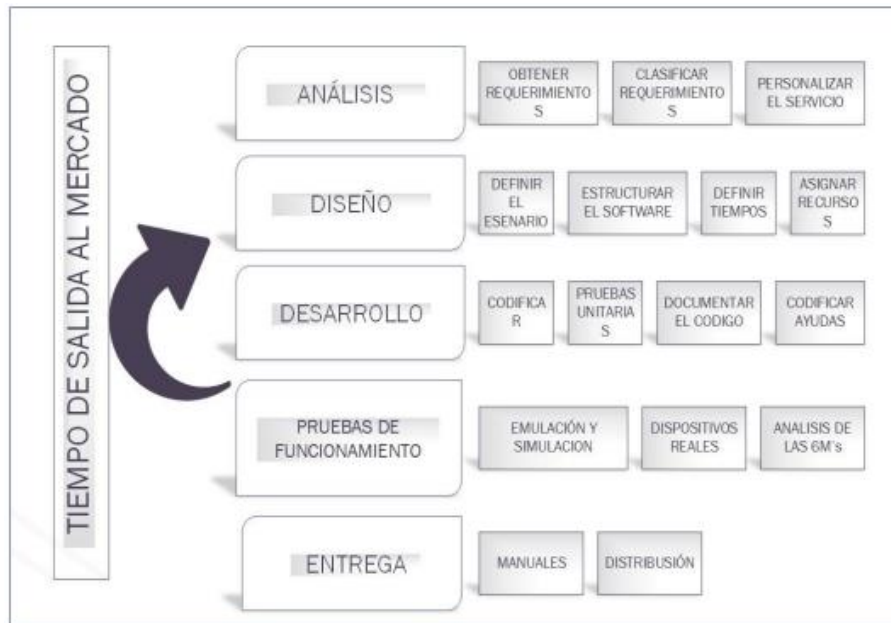
En el año 2015 el estudio [10] realizado en la Universidad Pública de Navarra, ciudad de Pamplona-España por el autor, V.E Carbonero, muestra el trabajo de investigación que tiene como nombre “Monitorización del consumo eléctrico de un hogar: Procesado de datos mediante Arduino”, tiene como objetivo diseñar y construir un contador de consumo eléctrico inteligente que permita al usuario conocer en cualquier instante el consumo de electricidad del hogar mediante la representación de los datos medidos, permitiendo implementar medidas de ahorro energético. Su metodología se centra en el procesado de la señal y la subida de datos a una hoja de cálculo de Google Drive mediante las herramientas que facilita la plataforma Arduino para ello se divide en tres partes, sensado y acondicionamiento, procesado y subida de datos y por último diseño de la página web. En este proyecto se emplea un Arduino Mini para realzar el sensado cada milisegundo, se envía cada segundo y medio mediante el módulo FS1000A, emisor de radiofrecuencia, un dato a un Arduino Uno conectado a una red de internet mediante cable Ethernet. El Arduino Uno recibe el dato a través del módulo receptor XY-MK-5V y lo sube a la Nube cada 10 minutos. En la Nube se almacena mediante una hoja de cálculo.

2.1.2. Antecedentes del Núcleo Temático de Aplicación Móvil

R.J. Castro Tenorio en su investigación “Desarrollo e Implementación de una Aplicación para Dispositivos Móviles, con Sistema Operativo Android, para el Control de Luminarias y Monitoreo de Consumo de Energía Eléctrica de una Vivienda” del año 2016 la cual se llevó a cabo en la ciudad de Guayaquil- Ecuador [11], tiene como finalidad desarrollar un sistema Android, que con la ayuda de circuitos integrados como Arduino Uno y sensores, que se conecta en el cuadro general de mando y protección de la vivienda para obtener una serie de parámetros (tensión, corriente, potencia, consumo, factor de potencia) que, una vez analizados, permitan hacer un uso más eficiente de los recursos energéticos eléctricos de una vivienda. Además, se utiliza una base de datos, que será la encargada de almacenar los datos que son emitidos por los dispositivos electrónicos, con la finalidad de conocer el historial de consumo de energía eléctrica de la vivienda. La comunicación entre la aplicación android y la base de datos, tiene como intermediario un servicio web, que valida el origen de las peticiones a la base de datos y la autenticación a la misma. Su metodología Ágil XP y MOOMH (Metodologías Orientada a Objetos para la producción de software Multimedia e Hipermedia) se centra en cuatro fases, Planificación del sistema, diseño del sistema, desarrollo del sistema, y pruebas e Implementación. El autor aporta conocimientos sobre tecnologías que en la actualidad están directamente relacionadas con el sistema operativo Android.

En el año 2015 fue presentado en la Universidad Politécnica Salesiana sede Quito, el trabajo de investigación [12] que lleva como nombre “Aplicación Android para Medición Inteligente de Energía Eléctrica en Usuarios Residenciales Urbanos” por el autor L.G de la Cruz Puebla, el cual tiene como finalidad acercar la empresa distribuidora hacia los usuarios por medio de información oportuna del consumo de energía. Los datos que requiere la aplicación para mostrar al usuario deben ser tomados de un medidor inteligente el cual alimenta una base de datos, los datos son recuperados por medio de un web server y mostrados en la aplicación. La metodología empleada para el desarrollo de la aplicación permite verificar cinco pasos ordenados secuencialmente que ayudan a la concepción, desarrollo, verificación de errores y corrección de los mismos, como se aprecia en la figura 2-1. Como conclusión este trabajo proporciona un conocimiento amplio del sistema operativo Android, su arquitectura, sus componentes y características, así como el funcionamiento y posibilidades ofrecidas por este sistema.

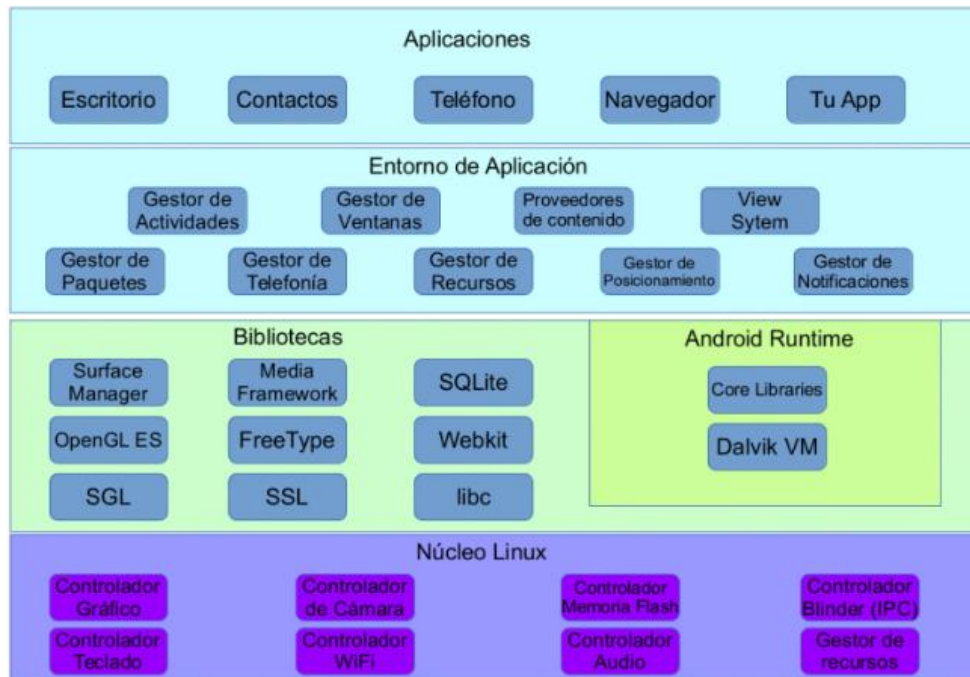
Figura 2-1: Metodología de Desarrollo.



Fuente: [12]

S. Alemany Ibor, en su trabajo [13] que tiene como nombre “Desarrollo de una Aplicación para la consulta de consumo eléctrico a través de dispositivos móviles” presentado en la ciudad de Valencia-España ha desarrollado una aplicación para dispositivos móviles y tabletas Android que permite a los clientes de una determinada empresa de comercialización eléctrica poder acceder tanto a los detalles de su tarifa y consumos, así como a las últimas facturas emitidas. También se aportan consejos sobre cómo ahorrar en la factura eléctrica. Todo esto ha sido desarrollado con el programa informático Android Studio. En el trabajo se presenta los Componentes principales de una aplicación Android y la arquitectura del sistema operativo Android (figura 2-2). Su metodología se basa en la Implementación del diseño mediante Layouts; conexión a la base de datos y tratamiento de mensajes obtenidos en formato JSON; hilos de ejecución para evitar cuelgues en las conexiones a la base de datos; simulación de tarifas eléctricas; creación de gráficas de consumo; y otras funciones secundarias como pueden ser implementar mapas de Google o añadir listas dinámicas. Las consultas a la base de datos se realizan a través de Internet accediendo a una serie de archivos PHP que realizan consultas predeterminadas sobre la base de datos. Reciben unas variables mediante la codificación HTTP y devuelven los datos obtenidos por la consulta sobre la base de datos codificados en el formato JSON.

Figura 2-2: Arquitectura de Android.



Fuente: [13]

2.1.3. Antecedentes del Núcleo Temático Plan Energético

En el año 2017 el estudio [14] realizado en la Universidad Nacional de Colombia por los autores, O. Gonzales, A. Pavas y S. Sánchez, muestran el artículo científico que tiene como nombre “Cuantificación del ahorro de energía eléctrica en clientes residenciales mediante acciones de gestión de demanda”, el cual presenta el modelado de los clientes de la electricidad y las estrategias de gestión de demanda propuestas. Su metodología consta de cuatro pasos progresivos, Condiciones iniciales del modelo (Población y tipo de carga), estrategias de gestión (Sustitución de tecnología, cambio de hábito de consumo y modelado de cargas), composición individual de la carga y combinaciones de cargas. Una de las conclusiones que aporta este trabajo es que, pueden lograrse reducciones en el consumo entre 6 y 26% mediante sustitución de dispositivos y proporcionar hasta un 19% de ahorro en energía eléctrica, sobre el consumo base con el uso de las estrategias indicadas. Lo anterior depende de la cantidad de estrategias adoptadas por los usuarios, y de los electrodomésticos presentes en el hogar sobre los cuales se aplicarían dichas estrategias.

J.P Lozano Celis y W.C Guzmán Espitia en su investigación “Evaluación de Demanda de Energía Eléctrica según Hábitos de Consumo Actuales en la Ciudad

de Bogotá” del año 2016 la cual se llevó a cabo en la Universidad Distrital Francisco José de Caldas [15], muestran que la metodología implementada permite realizar la identificación de hábitos de consumo en un grupo aleatorio de usuarios, con el fin de obtener patrones de consumo característicos dentro de los mismos. En esta se interactúa con el usuario final, para conocer la clase de prácticas empleadas en el hogar con respecto al consumo de energía eléctrica, teniendo en cuenta las siguientes variables: número de habitantes, edad, nivel educacional y ocupación, consumo eléctrico, conciencia de gasto del servicio, equipamiento eléctrico presente en las viviendas, nivel cultural, presencia de fuentes alternativas de generación y de aparatos con consumos elevados, y adicionalmente, se recopila las opiniones de los usuarios con relación a sus hábitos de consumo. Finalmente, la investigación concluye que a partir de las correlaciones realizadas en el proyecto sobre las categorías comparadas con el consumo eléctrico promedio, como el ingreso económico, nivel educativo y género, son factores que no deberían ser tenidos en cuenta, debido a que presentan un bajo impacto sobre el dicha demanda; mientras que el número de habitantes y el área de la vivienda son variables de gran influencia, por lo deberían ser características de mayor relevancia en estudios de esta naturaleza.

2.2. BASES TEÓRICAS

2.2.1. Plataformas hardware

- **Raspberry Pi**

Es un ordenador de tamaño reducido, placa simple y de bajo costo (figura 2-3) desarrollado en Reino Unido por la fundación Raspberry Pi. Está diseñada para ejecutar el sistema operativo GNU/Linux el cual posibilita descargar su código fuente por completo y realizar cambios. Tiene un procesador multimedia Broadcom system-on-chip (SoC) que permite su funcionamiento con tan sólo una fuente de alimentación de 5V-2A suministrada por el puerto micro-USB a bordo, además, evita el uso de disipadores térmicos sobre el dispositivo ya que el bajo consumo de energía del chip se traduce directamente en muy poco calor residual, incluso durante las tareas de procesamiento más complejas. El sistema operativo recomendado para uso normal es Raspbian, pero permite la instalación de otros sistemas operativos como Windows 10 IoT Core [16].

Figura 2-3: Raspberry Pi 2 modelo B.

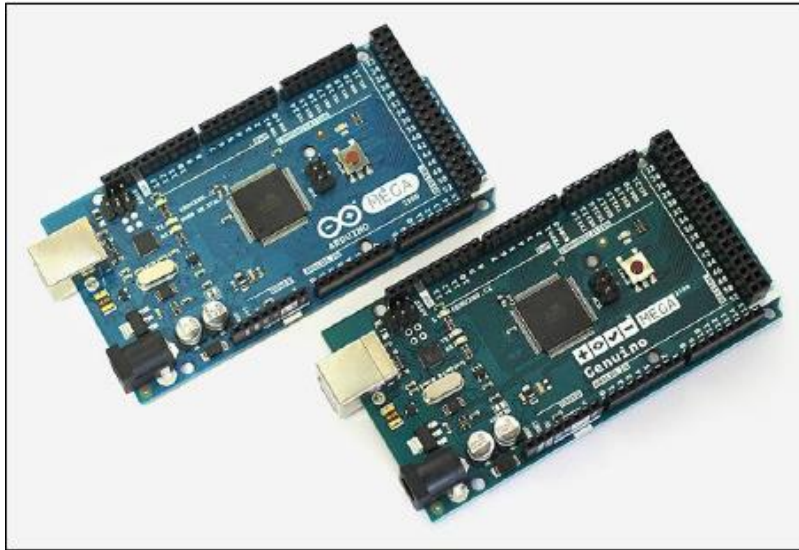


Fuente [16]

- **Arduino**

Es una plataforma hardware de desarrollo libre, está basado en los microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Una de sus características más sobresaliente es su bajo costo, sus placas son relativamente baratas comparadas con otras plataformas microcontroladoras. Cuenta con software multiplataforma, opera en los sistemas operativos Windows, Macintosh OSX y GNU/Linux. Su entorno de programación es simple y claro para facilitar el uso de la electrónica en proyectos multidisciplinarios. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. En la figura 2-4 se presenta la tarjeta de desarrollo Arduino Mega [9].

Figura 2-4: Arduino mega 2560.



Fuente [9]

Arduino Mega es una tarjeta de desarrollo de código abierto que implementa el lenguaje Processing/Wiring, la cual adicionalmente a todas las características de su sucesor utiliza un microcontrolador ATmega8U2 en vez del circuito integrado FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo INF es necesario para Windows) [9].

El Arduino mega 2560 posee las siguientes características [9]:

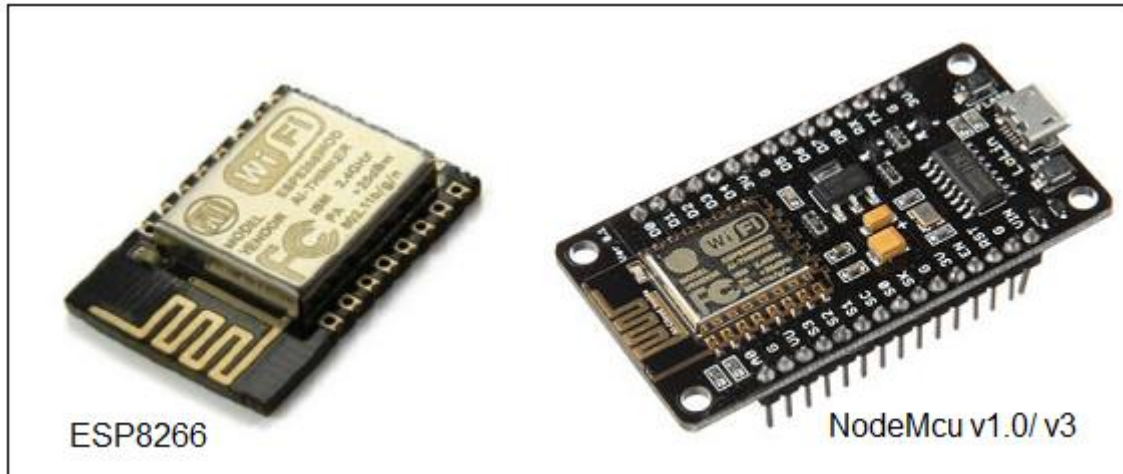
- Microcontrolador Atmega2560
- Voltaje de entrada 7-12V
- 4 puertos serial por hardware
- 16 entradas análogas.
- 54 pines digitales E/S (14 de ellos son salidas PWM).
- Tensión de funcionamiento 5V
- Memoria SRAM 8KB
- Memoria EEPROM 4KB
- Memoria flash 256 KB, 8 KB utilizado por el gestor de arranque
- Velocidad de reloj 16 MHz

2.2.2. Módulo Wi-Fi NodeMcu

NodeMcu es la placa de desarrollo basada en ESP8266 (Este chip tiene incorporado el módulo Wi-Fi), es totalmente abierta, a nivel de software y de hardware (figura 2-5). Sus características principales son [17]:

- Conversor Serie-USB para poder programar y alimentar a través del USB
- Fácil acceso a los pines
- Pines de alimentación para sensores y componentes
- LEDs para indicar estado
- Botón de reset

Figura 2-5: Módulo Wi-Fi NodeMcu.



Fuente: [17]

2.2.3. Medidores de energía eléctrica.

- **Medidor Monofásico ED610D2**

Este medidor (figura 2-6) con pantalla LCD para montaje en riel DIN pertenece a la nueva generación de medidores electrónicos monofásicos. El aparato utiliza técnicas micro electrónicas, circuitos LSI y tecnologías de montaje superficial digitales y avanzadas. Cumple con los requisitos técnicos de la clase 1, la clase 2 y IEC62052-11, IEC62053-21. En la tabla 2-1 se puede ver las especificaciones generales de este dispositivo.

Figura 2-6: Medidor Monofásico ED610D2



Fuente: [Anexo 1]

Tabla 2-1: Especificaciones Generales del Medidor Monofásico ED610D2

Voltaje Ac	120v
Corriente	5(80)
Clase de precisión	1.0
Frecuencia	50/60 Hz
Modo de visualización	LCD 5+1 = 99999.9 kWh
Consumo	≤ 0.4 W (8 VA)
Rango de temperatura	-20 a 65 °C
LED	1 parpadeo por impulso
Salida de impulsos	1000 imp/kWh tensión: de 12 a 27 V

	corriente: 27 mA
	anchura de impulso: 90 ms

Fuente: [anexo 1]

- **Medidor de Potencia PM1200**

Este dispositivo (figura 2-7) ofrece todas las características necesarias que requieren monitoreo, además de ser de fácil identificación y lo suficientemente resistente para ambientes comerciales e industriales. En la tabla 2-2 y 2-3 se pueden observar las especificaciones técnicas de este dispositivo.

Figura 2-7: Medidor de Potencia PM1200



Fuente: [anexo 2]

Tabla 2-2: Especificaciones Generales del Medidor de Potencia PM1200

Tensión de alimentación	44...277 V CC 44...277 V CA (50/60Hz)
Frecuencia de red	50/60 Hz
consumo de energía	3 VA en 240 V AC

Tipo de pantalla	LED
Tasa de muestreo	20 muestras/ciclo
Corriente de medición	50...6000mA
Tipo de entrada	Corriente 0...5 A (Impedancia $\leq 0.1\text{ohm}$) Corriente 0...1 A (Impedancia $\leq 0.1\text{ohm}$)
Tensión de medida	80...480 V CA (Fase a Fase)
Precisión de medida	$\pm 1\%$ potencia $\pm 1\%$ corriente $\pm 1\%$ tensión $\pm 1\%$ energía $\pm 0.1\%$ frecuencia $\pm 1\%$ factor de potencia $\pm 1\%$ potencia aparente
Clase de precisión	Clase 2 (energía reactiva según IEC 62053-23) Clase 1 (energía activa según IEC 62053-21)
Protocolo de puerto de comunicación	Modbus: 19.2 kbaudios
Soporte del puerto de comunicación	RS485

Fuente: [anexo 2]

Tabla 2-3: Entorno del Medidor de Potencia PM1200

Soporte de montaje	Panel
--------------------	-------

Categoría de sobretensión	III
Grado de protección IP	IP51 (frontal) acorde a IEC 60529 IP40 (negro) coordinación IEC 60529
Humedad relativa	5...95 %
Temperatura ambiente de funcionamiento	-10...60 °C
Temperatura ambiente de almacenamiento	-25...70 °C
Altitud máxima de funcionamiento	0...2000 m
Normas	IEC 62053-21 IEC 62052-11

Fuente: [anexo 2]

- **Medidor Eastron sdm230**

Los medidores de energía "con una pantalla LCD retroiluminada azul para una buena lectura" se utilizan para medir aplicaciones residenciales, de servicios públicos y aplicaciones industriales monofásicas (figura 2-8). La unidad mide y muestra varios parámetros eléctricos importantes y proporciona un puerto de comunicación para lectura y monitoreo remoto. La medición de energía es bidireccional. En las tablas 2-1, 2-2 y 2-3 se muestran las especificaciones técnicas.

Figura 2-8: Medidor Eastron sdm230.



Fuente: [anexo 3]

Tabla 2-4: Especificaciones Generales del Medidor Eastron sdm230.

Voltaje Ac	120v
Corriente de Base (Ib)	10 A
Corriente Máxima (Imax)	100 A
Corriente Mínima (Imin)	0.5 A
Corriente de Arranque	0.4% de Ib.
Consumo de Energía	<2W/10 VA
Frecuencia	50/60Hz ($\pm 10\%$)
Resistencia de Tensión AC	4KV por 1 Minuto
Resistencia a Impulso De Tensión	6kv-1.2uS Forma de Onda
Resistencia de Sobrecorriente	30 A Imax para 0.01s

Velocidad de salida del pulso	1000 imp/KWh (Predeterminado) 1000/100/10/1imp/KWh/KVarh (Configurable)
Display	Lcd Con Retroiluminación Azul
Lectura Máxima	999999.9kWh

Fuente: [anexo 3]

Tabla 2-5: Exactitud del Medidor Eastron sdm230.

Voltaje	0.5% del rango máximo
Corriente	0.5% del nominal
Frecuencia	0.2% de frecuencias medias
Factor de potencia	1% de Unidad
Potencia activa	1% del rango máximo
Potencia reactiva	1% del rango máximo
Potencia aparente	1% del rango máximo
Energía activa	Clase 1 IEC62053-21 Clase B EN50470-3
Energía Reactiva	1% del rango máximo

Fuente: [anexo 3]

Tabla 2-6: Entorno del Medidor Eastron sdm230.

Temperatura de funcionamiento	-25°C a +55°C
Temperatura de almacenamiento y transporte	-40°C a +70°C
Temperatura de referencia	23°C ±2°C

Humedad relativa	0 a 95%, sin condensación
Altitud	hasta 2500m
Tiempo de calentamiento	10s
Categoría de instalación	CAT III
Ambiente Mecánico	M1
Entorno electromagnético	E2
Grado de contaminación	2

Fuente: [anexo 3]

2.2.4. Servicio Web

Es una interfaz de software que describe un conjunto de operaciones a las cuales se puede acceder por la red a través de mensajería XML estandarizada, además sirve para intercambiar datos entre aplicaciones, permite manejarlos con mucha más facilidad y establecer una comunicación más libre entre los software, ya que funciona a un nivel más abstracto que faculta reevaluarlos y modificarlos dinámicamente mediante solicitud [13].

2.2.5. Servidor Web

Es un programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (Lenguaje de Marcas de Hipertexto), su principal función es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios proporcionándoles los recursos que solicitan mediante el protocolo HTTP o el protocolo HTTPS (la versión segura, cifrada y autenticada) [11].

Cuando un usuario entra en una página de Internet, su navegador se comunica con el servidor enviando y recibiendo datos que determinan lo que se ve en la pantalla. Esto quiere decir, que para una página web sea accesible en cualquier momento, el servidor debe estar permanentemente en línea [11].

Un servidor web básico tiene un esquema de funcionamiento sencillo, ejecutando de forma infinita el bucle siguiente: [12].

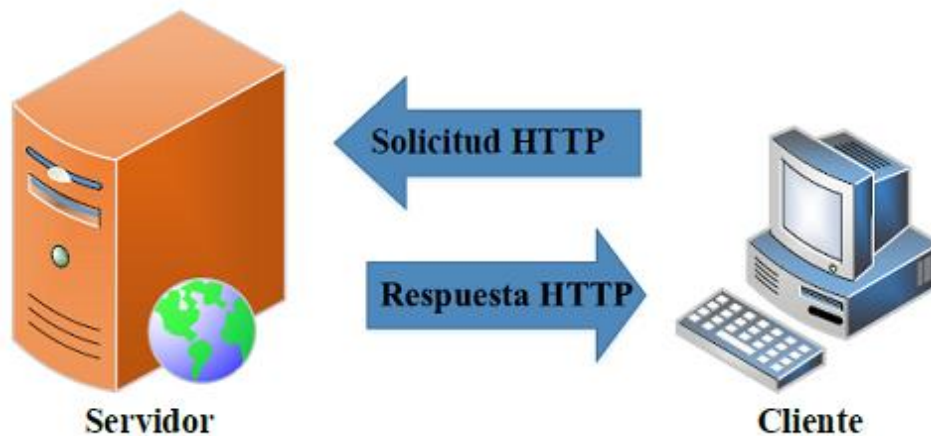
- Espera las peticiones en el puerto TCP (puerto 80).
- Recibe la petición.
- Busca el recurso en la cadena de petición.
- Envía la respuesta por la misma conexión por la que recibió la petición.
- Regresa al punto dos y realiza la secuencia nuevamente.

Algunos servidores web importantes son Apache, Nginix, Microsoft IIS (Servicios de Información de Internet) y Google Web Server [12].

2.2.6. Protocolo de transferencia de hipertexto

El Protocolo HTTP permite las transferencias de información en la red informática mundial y se basa en sencillas operaciones de solicitud/respuesta como se observa en la figura 2-9. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones pueden adjuntar un objeto o recurso sobre el que actúan; cada objeto Web (documento HTML, fichero multimedia o aplicación CGI) es conocido por su URL. [9]

Figura 2-9: Funcionamiento del protocolo HTTP.



Fuente [9]

2.2.7. Protocolos de comunicación inalámbricos

Los estándares Wi-Fi (IEEE 802.11), Bluetooth (norma IEEE 802.15.1), ultra-wideband (UWB - ultra banda ancha, norma IEEE 802.15.3) y ZigBee (IEEE 802.15.4), son cuatro de los protocolos de comunicación inalámbrica de corto

alcance más importantes y con bajo consumo de energía [17], es por esto que desde la perspectiva de aplicación de cada uno se diferencian en que *“Wi-Fi está dirigido a conexiones de equipo a equipo como una extensión o sustitución de las redes cableadas, Bluetooth está destinado a periféricos como ratones inalámbricos, teclados, auriculares, manos libres, controles para consolas de videojuegos, entre otros. UWB está orientado a enlaces multimedia de banda ancha mientras que ZigBee está diseñado para el monitoreo y control de redes inalámbricas confiables”*. [18].

- **Wi-Fi**

La conectividad Wi-Fi se basa en el estándar IEEE 802.11 el cual define los dos primeros niveles del modelo OSI, es decir, la capa física y la capa de enlace de datos, compuesta a su vez por el control de enlace lógico y control de acceso al medio. Wi-Fi es el estándar con mayor capacidad instalada, sus módulos para desarrollo, que hasta hace poco eran escasos, se encuentran hoy a precios similares a los de otras normas. Entre otras ventajas, permite el uso de menos infraestructura al no requerir de puertas de entrada adicionales para acceder a Internet, alcances de señal adecuados para su uso en ambientes interiores (hogares con múltiples habitaciones y pisos). Opera en las bandas de 2,4 y 5 GHz, alcanzando velocidades teóricas de 600 Mbps para el estándar 802.11n y de 7Gbps para el estándar 802.11ac. Las distancias de transmisión de datos varían según la frecuencia utilizada, pudiéndose alcanzar distancias de 35 m en interiores y 100 m en exteriores. Desde el punto de vista de la seguridad, Wi-Fi implementa la encriptación WPA y WPA-2, que provee una razonable seguridad para aplicaciones domésticas [19].

- **Bluetooth Low Energy**

El bluetooth pertenece al estándar de comunicación IEEE 802.15.1, el cual posibilita la transmisión de información inalámbrica de corto alcance entre dos dispositivos interconectados. El Bluetooth de bajo consumo opera en la frecuencia de 2,4 GHz, con una distancia de señal entre los 10 y 100m. Las velocidades alcanzadas con esta tecnología son de 1Mbps [19].

- **Zigbee**

Es un conjunto de protocolos de alto nivel de comunicación basados en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (Wireless Personal Área

Network, WPAN), se utiliza para la radiodifusión digital de datos buscando ahorrar lo máximo posible en energía. La tecnología de comunicación inalámbrica ZigBee utiliza la banda ISM y por lo general, adopta la banda 2.4GHz para comunicarse con el resto de dispositivos ya que esta se adopta en todo el mundo. ZigBee proporciona bajo consumo con una fácil implementación (larga duración a las baterías de los dispositivos mediante dos modos, uno activo y otro de latencia), Costo bajo tanto para el dispositivo y la instalación como para el mantenimiento (sus dispositivos permiten utilizar baterías no recargables que gracias a su bajo consumo, pueden durar años) y también ofrece alta densidad de nodos por red (El uso de las capas físicas y de acceso al medio del estándar IEEE 802.15.4 permite a las redes tener cualquier número de dispositivos). Este es un protocolo asíncrono y half dúplex, en el que los paquetes de datos se direccionan de manera dinámica, ya que la red se organiza y repara de forma automática. Se trata de un protocolo seguro, ya que permite la implementación de encriptación y autenticación. Las velocidades de transmisión varían entre 20, 40 y 250 kbit/s [19].

2.2.8. PHP (Preprocesador de hipertexto)

PHP es un lenguaje de programación de código abierto, flexible, potente y de alto rendimiento diseñado para el desarrollo web de contenido dinámico con acceso a información almacenada en una base de datos. El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Una ventaja del lenguaje de programación PHP es que permite ejecutar sentencias de SQL facilitando el manejo de la información en bases de datos. Tiene Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL [12].

2.2.9. JSON (JavaScript Object Notation - Notación de Objetos de JavaScript)

Es un formato ligero de intercambio de datos basado en un subconjunto del Lenguaje de Programación JavaScript, estándar ECMA-262 3rd Edición - diciembre 1999. Este formato de texto es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros [20].

2.2.10. Plataformas software

- **Thingspeak**

Es una plataforma de código abierto que permite desarrollar aplicaciones de IoT. Este servicio web gratuito diseñado por Mathworks cuenta con una Interfaz de programación de aplicaciones (API) para almacenar y recuperar datos de sensores usando el protocolo HTTP a través de Internet o a través de una LAN (Local Área Network), además proporciona aplicaciones que permiten analizar y visualizar sus datos en el software MATLAB y luego actuar sobre ellos. Se debe a agregar que esta plataforma faculta la integración con Arduino, Raspberry Pi, IoBridge/ RealTime.io, Electric Imp, Móviles / Aplicaciones web y Redes Sociales [20], [21] y [22].

- **CloudMQTT**

Son servidores Mosquitto gestionados en la nube, los cuales implementan el protocolo MQTT (Message Queue Telemetry Transport) que proporciona métodos simples y ligeros para llevar a cabo mensajes utilizando un modelo de cola de mensajes de publicación / suscripción. Este protocolo es ideal para la comunicación maquina a máquina (M2M), diseñado para dispositivos con restricciones y redes de bajo ancho de banda, alta latencia o poco confiables. Los dispositivos pueden enviar (publicar) o recibir (suscribirse) mensajes asociándoles un “topic” (tema). Cuenta con una arquitectura cliente/servidor donde el cliente se conecta a un Bróker, el cual se encarga de gestionar los mensajes y distribuirlos entre todos los dispositivos conectados para interactuar en la red [20] y [21].

2.2.11. Stand-by

Se produce cuando un aparato electrónico o electrodoméstico está enchufado a la red, aunque no se utilice. Este se encuentra conectado a la espera de recibir órdenes permitiendo el encendido de forma rápida. Los aparatos en modo de espera o stand by siguen consumiendo electricidad, y debido al incremento del número de dispositivos eléctricos ese consumo supone con el tiempo una cifra significativa.

La Organización de Consumidores y Usuarios de España la cual ofrece información objetiva y veraz para tomar buenas decisiones sobre el consumo en el hogar presenta los siguientes datos en la tabla 2-7:

Tabla 2-7: Consumo en stand-by de Aparatos electrónicos.

	Potencia Stand-by	Consumo anual
TV CRT 32"	6	53
TV LCD 37"	2	18
TV plasma 42"	3	26
DVD	4	35
Video	8	70
Decodificador satélite	7	61
Decodificador digital	5	44
Consola	4	35
Teléfono inalámbrico	4	35
Radio despertador	7	61
Cadena de música	6	53
Ordenador	5	44
Portátil	4	35
Monitor CRT	3	26
Monitor LCD	1	9
Router	8	70
Impresora	8	70
Altavoces PC	3	26
Microondas	4	35
Máquina del café	5	44

Total	97 W	850 kWh
-------	------	---------

Fuente: [24]

El Instituto para la Diversificación y Ahorro de la Energía (IDAE) la cual está adscrita al Ministerio de Industria, Energía y Turismo a través de la Secretaría de Estado de Energía en España, estima que el 10.7% de todo el consumo energético de los electrodomésticos corresponde al Stand-by, como se puede ver en la figura 2-10. [25]

Figura 2-10: Consumo de Electrodomésticos según Tipo de Equipamiento.

Frigorífico:	30,6%
Congeladores:	6,1%
Lavadora:	11,8%
Lavavajillas:	6,1%
Secadoras:	3,3%
Horno:	8,3%
TV:	12,2%
Ordenadores :	7,4%
Standby:	10,7%
Otros:	3,5%

Fuente: [25]

En Colombia, La Unidad de Planeación Minero Energética (UPME) realizó un análisis sobre el consumo de stand-by mediante la revisión de referentes internacionales e información de mercado con lo cual pudo determinar que un equipo domestico puede tener potencias de stand by de hasta 23W como se muestra en la Figura 2-11 y pueden llegar a representar hasta el 10% del consumo de energía eléctrica en un hogar [1].

Figura 2-11: Potencias en stand-by de los principales equipos domésticos.

Ítem	Equipo	Potencia en stand by (W)
1	TV CRT	Hasta 23
2	TV LCD	Hasta 4
3	Horno microondas	12
4	Decodificador de TV	12 – 22
5	Módem	10

Fuente: [1]

2.3. BASES LEGALES

2.3.1. Legislación Colombiana

El presente trabajo está sustentado con elementos jurídicos que se relacionan con el tema de energía eléctrica y eficiencia energética, debido a esto se apoya en la constitución nacional de Colombia y las diferentes leyes, decretos y resoluciones que se indican en la tabla 2-8.

Tabla 2-8: Bases legales.

Ley	Fecha	Descripción
Ley 143	11 de julio de 1994	Por la cual se establece el régimen para la generación, interconexión, transmisión, distribución y comercialización de electricidad en el territorio nacional, se conceden unas autorizaciones y se dictan otras disposiciones en materia energética.
Ley 697	3 de octubre de 2001	Mediante la cual se fomenta el uso racional y eficiente de la energía, se promueve la utilización de energías alternativas y se dictan otras disposiciones.

Decreto 3683	19 de diciembre de 2003	Por el cual se reglamenta la Ley 697 de 2001 y se crea una Comisión Intersectorial
Resolución 563	21 de diciembre de 2012	Por la cual se establecen el procedimiento y los requisitos para evaluar y conceptuar sobre las solicitudes que se presenten ante el Ministerio de Ambiente y Desarrollo Sostenible con miras a obtener la exclusión del impuesto sobre las ventas IVA y/o deducción en la renta de elementos, equipos y maquinaria destinados a proyectos, programas o actividades de reducción en el consumo de energía y eficiencia energética.
Ley 1715	13 de mayo de 2014	Regula la integración de las energías renovables no convencionales al Sistema Energético Nacional e igualmente contiene aspectos relacionados con la eficiencia energética y respuesta de la demanda, así como incentivos tributarios que permitan maximizar los beneficios para el país por la ejecución de este tipo de proyectos y programas.
Resolución 1283	3 de agosto de 2016	Por la cual se establece el procedimiento y requisitos para la expedición de la certificación de beneficio ambiental por nuevas inversiones en proyectos de fuente no convencionales de energías renovables - FNCER y gestión eficiente de la energía, para obtener los beneficios tributarios de que tratan los artículos 11, 12,13 y 14 de la ley 1715 de 2014 y se adoptan otras determinaciones.

Resolución 1286	30 de diciembre de 2016	Por la cual se adopta el Plan de Acción Indicativo 2017-2022 para el desarrollo del programa de Uso Racional y Eficiente de la Energía, PROURE, que define objetivos y metas indicativas de eficiencia energética, acciones y medidas sectoriales y estrategias base para el cumplimiento de metas y se adoptas otras disposiciones al respecto.
Resolución 585	2 de octubre de 2017	Por la cual se establece el procedimiento para conceptuar sobre los proyectos de eficiencia energética/gestión eficiente de la energía que se presenten para acceder a beneficios tributarios.

CAPITULO III: METODOLOGÍA

La metodología aplicada para la elaboración de este proyecto se divide en 5 fases de desarrollo con sus respectivas actividades, estas son:

3.1. FASE PREPARATORIA

En esta fase se alcanza las actividades encaminadas a la organización y planificación del estudio de las áreas temáticas. Las actividades específicas de esta fase son:

- Tipificación de las unidades conceptuales en el área de estudio planteada.
- Identificación de los componentes o núcleos temáticos comprendidos en la descripción de los objetivos.
- Definición por orden jerárquico de las áreas temáticas y sus respectivos núcleos.

Para este proyecto se ha definido los siguientes componentes temáticos:

- **Componente Temático de Medición:** Componente encargado del estudio, desarrollo e implementación del hardware y software para la obtención, lectura y transmisión de las variables eléctricas.
- **Componente Temático de Aplicación Móvil:** Componente enfocado en el desarrollo de la aplicación móvil donde se visualizará datos estadísticos del consumo energético.
- **Componente Temático Plan Energético:** Componente encargado de las notificaciones y recomendaciones para el uso racional y eficiente de la energía eléctrica.

3.2. FASE DESCRIPTIVA

Comprende aquellas actividades encaminadas a la recolección, identificación y estudio de los elementos clave de cada área temática. Las actividades específicas de cada uno de los componentes temáticos son:

Componente Temático de Medición

- Identificación de los dispositivos electrónicos de medición de consumo de energía en la actualidad.
- Definición de los componentes necesarios para el diseño del sistema electrónico.
- Verificación del correcto funcionamiento de los sensores, módulos de comunicación e instrumentos de medición de la electricidad.
- Comprobación de la exactitud de los datos medidos por el dispositivo.
- Estudio y clasificación de los sistemas de recopilación, almacenamiento, clasificación y distribución de datos.
- Estudio y clasificación de diferentes protocolos de comunicación.

Componente Temático de Aplicación Móvil

- Estudio y clasificación de dispositivos móviles y sus sistemas operativos.
- Identificación y clasificación de las diferentes alternativas en aplicaciones móviles para sistemas de monitoreo energético.

Componente Temático Plan Energético

- Estudio e identificación de comportamientos de consumo de energía residencial con el objetivo de preparar acciones correctivas por medio de alarmas.

3.3. FASE INTERPRETATIVA POR NÚCLEOS TEMÁTICOS

Comprende las actividades orientadas al análisis y síntesis de los elementos clave en cada uno de los componentes temáticos para la interpretación del conocimiento. Las actividades específicas de cada uno de los componentes son:

Componente Temático de Medición

- Definición de las variables a tratar en el sistema de monitoreo energético.
- Definición de la estructura de almacenamiento y distribución de la información. (servidor)
- Elección del protocolo con la estructura de trama de datos más factible para la transmisión de estos.

- Detección de posibles fallas o errores a la hora de transmitir o recibir las tramas de datos.

Componente Temático de Aplicación Móvil

- Elección de un entorno viable para el desarrollo de la aplicación móvil.
- Definición de los requerimientos mínimos de los equipos móviles donde se instalará la aplicación.
- Definición del entorno gráfico y lógico de la aplicación móvil.

Componente Temático Plan Energético

- Organización y visualización de los datos obtenidos mediante la medición para generar reportes de consumo diario de energía
- Generación de alarmas pertinentes de acuerdo a un consumo energético específico.

3.4. FASE DE CONSTRUCCIÓN TEÓRICA GLOBAL

Especifica las actividades que enmarcan la generación de nuevos conocimientos para la elaboración del contenido científico del proyecto, teniendo como base el área y los componentes temáticos. Las actividades específicas de cada uno de los componentes son:

Componente Temático de Medición

- Implementación de los módulos de captura y envío de variables eléctricas a la base de datos.

Componente Temático de Aplicación Móvil

- Desarrollo, identificación y corrección lógica y grafica de la aplicación móvil.

Componente Temático Plan Energético

- Elaboración de un sistema de notificación de alertas.
- Presentación grafica de tips para el ahorro energético en el hogar.

3.5. FASE DE EXTENSIÓN Y PUBLICACIÓN

Comprende las actividades orientadas a la presentación de los resultados en forma escrita. Las actividades específicas son:

- Análisis y síntesis de los resultados de las actividades planteadas.
- Comunicación con el director del proyecto para entrega de avances e informes.
- Elaboración del documento final.
- Revisión de los entregables

CAPITULO IV: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se describe el proceso utilizado para el diseño e implementación de los diferentes sistemas que hacen parte del proyecto, sus procesos de adquisición datos, comunicación de dispositivos y representación gráfica para el usuario de las variables monitoreadas y controladas, además, se dan a conocer el software y hardware que hace parte del desarrollo, teniendo en cuenta la metodología planteada para cada componente temático de medición, aplicación móvil y plan energético.

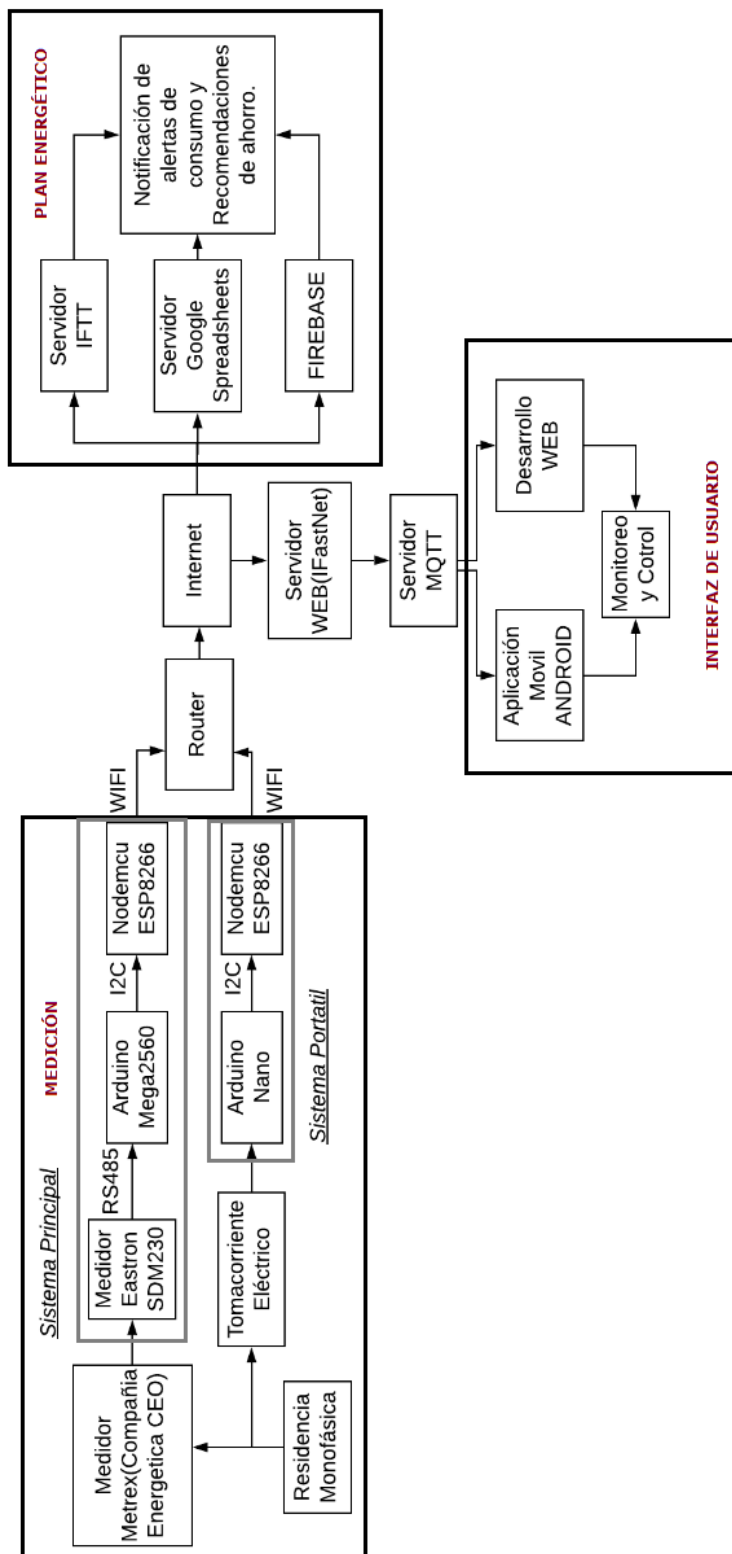
4.1. DISEÑO DE LOS SISTEMAS.

Se realizo el diseño de dos sistemas, uno principal que va conectado directamente a la caja de breakers de la vivienda, después del medidor instalado por la compañía prestadora del servicio de energía (Compañía Energética de Occidente) con el objetivo de conocer y comparar el consumo eléctrico total de las cargas conectadas (Luces, Electrodomésticos, etc.), con el consumo cobrado en la factura de energía eléctrica mes a mes.

Por otra parte, se diseñó un sistema portátil con las mismas características de programación hardware y software, además de la conexión con servicios web lot, aunque se cambió el dispositivo esclavo por una tarjeta Arduino Nano que no obtiene las mediciones de los parámetros eléctricos por parte de un medidor comercial, sino que las realiza utilizando diferentes componentes electrónicos y la librería "EmonLib.h" disponible para las tarjetas de desarrollo Arduino. Este sistema permite monitorear el consumo eléctrico discriminado por las cargas específicas que sean enchufadas.

A continuación, se presentan los sistemas según el siguiente diagrama de bloques de la figura 4-1.

Figura 4-1: Diagrama de bloques del diseño de sistemas desarrollados.



Fuente: [Elaboración propia]

4.1.1. Medición

Inicialmente el sistema cuenta con una etapa de medición, mediante la utilización de un medidor monofásico de la marca Easton, el SDM230 el cual fue escogido para este proyecto debido a 3 factores: primeramente por ser un medidor de precio bastante asequible para cualquier persona del sector residencial de estrato medio-alto a quienes va dirigido este sistema, además de esto proporciona las mediciones de varios parámetros eléctricos importantes y ofrece una salida de comunicación para lectura y monitoreo remoto mediante el protocolo de comunicación RS485.

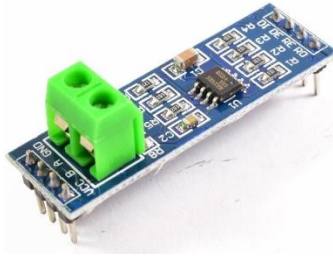
La medición de energía implementada es bidireccional utilizando el protocolo de comunicación Modbus para enviar estos datos a nuestro controlador wifi que se encargara de la transmisión de toda la información a la nube.

La adquisición de los datos del medidor Easton se hace mediante la interacción entre dos partes, hardware y software. El primero está integrado por el módulo rs485, la tarjeta de desarrollo Arduino mega 2560, el módulo wifi esp8266 y demás componentes electrónicos pasivos (son aquellos que no tienen la capacidad de controlar la corriente por medio de otra señal eléctrica, estos son: condensadores, resistencias, inductores, transformadores y diodos) y activos (pueden controlar el flujo de electricidad, estos son: transistores, tubos de vacío, rectificadores controlados de silicio). El segundo está conformado por librerías, programas de computadora y plataformas web que más adelante se mencionaran y que hacen parte tanto del Frontend (parte del sitio web que interactúa con los usuarios la cual está del lado del cliente), como del backend (es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios) el cual se conecta a la base de datos y hace parte del lado del servidor del proyecto planteado).

El medidor Easton de acuerdo a sus características permite configurar la velocidad de transmisión entre 1200, 2400,4800 y 9600 baudios, en este caso se configuro esta opción a 9600 Bd ya que la implementación realizada permitió garantizar la máxima velocidad de transmisión, porque no hay requerimientos de distancia entre equipos, por lo tanto no hay pérdida o retardo de información debido a la corta distancia entre los dispositivos, también se estableció paridad impar serial_801 tamaño de datos 8 bits y bits de parada 1 bit ya que se comprobó mediante experimentación con los equipos adquiridos que era la configuración que mejor funcionalidad tenía, además del Tiempo de salida de datos en la pantalla, tiempo de

escaneo de datos enviados y el pin de recepción y transmisión para el módulo RS485 (Figura 4-2 y Tabla 4-1).

Figura 4-2: Módulo Rs485 a Serial Ttl.



Fuente: [26]

Tabla 4-1: Características del módulo Rs485.

DESCRIPCIÓN DEL MODULO	ESPECIFICACIONES TÉCNICAS
<p>Este módulo permite construir una interfaz RS485 a Serial TTL. El protocolo RS485 es un protocolo industrial ampliamente utilizado por su robustez, fácil implementación y buenas prestaciones. Utilizando RS485 como capa física es posible implementar un bus de campo industrial tipo MODBUS y DMX para el control de luces.</p> <p>El protocolo RS485 está definido como un sistema en bus de transmisión multipunto diferencial, es ideal para transmitir a altas velocidades sobre largas distancias (35 Mbit/s hasta 10 metros y 100 kbit/s en 1200 metros) y a través de canales ruidosos, ya que reduce los ruidos que aparecen en los voltajes producidos en la línea de transmisión. El medio físico de transmisión es un par entrelazado que admite hasta 32 estaciones en 1 solo</p>	<ul style="list-style-type: none"> -Voltaje de Operación: 5 V. -Consumo Corriente: 500 uA (máx). -Chip principal: MAX485. -Tipo de Comunicación: Half-Duplex. -Velocidad máxima de 10 Mbit/s (a 12 metros). -Longitud máxima de alcance de 1200 metros (a 100 kbit/s).

hilo, con una longitud máxima de 1200 metros operando entre 300 y 19 200 bit/s y la comunicación half-duplex (semiduplex).	
--	--

Fuente: [26]

En el entorno de desarrollo integrado (IDE) de Arduino es de suma importancia instalar las siguientes librerías: ESP8266WiFi.h, necesaria para el nodemcu; SimpleModbusMasterSDM120.h, permite la comunicación modbus; Wire.h, posibilita la comunicación I2C y configuración de las direcciones de los registros de lectura para las variables eléctricas que entrega el medidor.

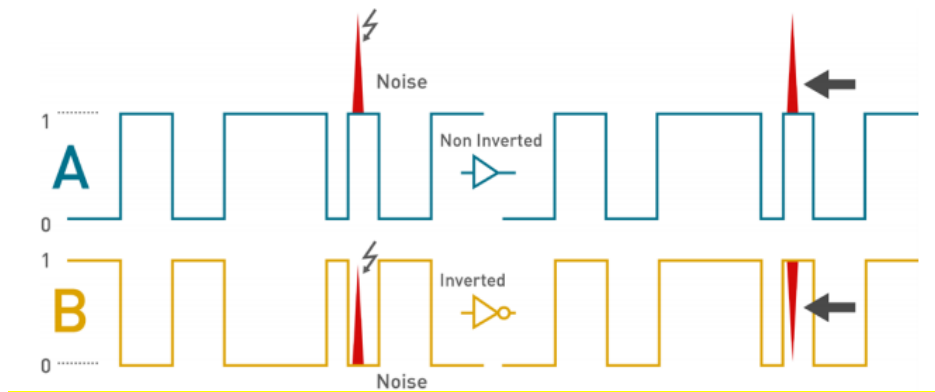
En el anexo 4 se puede observar las librerías requeridas para el correcto funcionamiento del sistema y en el anexo 5 se encuentra el código de programación para el Arduino Mega 2560.

4.1.1.1 Módulo RS485

Para la implementación del proyecto se utilizará el módulo RS485 Serial TTL (Figura 4-2) el cual que es un transceptor (un dispositivo que cuenta con un transmisor y un receptor que comparten parte de la circuitería) de baja potencia y es normalmente usado para comunicación RS-485.

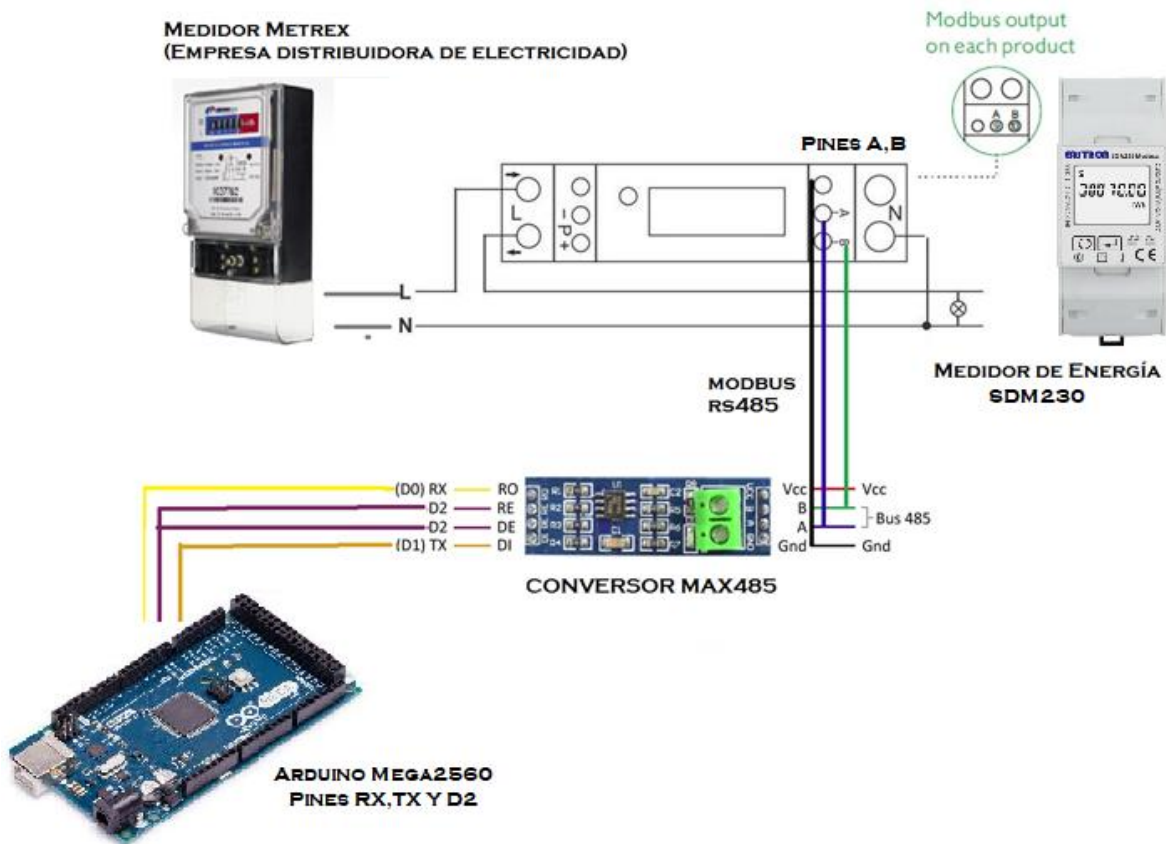
Los pines A y B del módulo RS485 Serial TTL son por donde se envían y reciben los datos, B es la negación en voltaje de A, de esta forma se pueden restar ambas señales y eliminar el ruido y quedarnos solo con la señal de datos como se muestra en la figura 4-3. El encargado de esta transmisión diferencial es el chip MAX485 el cual se comunica con la tarjeta Arduino Mega 2560 utilizando la comunicación serial, mediante los pines RX y TX además de ser necesaria la conexión de los pines DE y RE del módulo RS485 al pin 2 del Arduino como se puede ver en la Figura 4-4, esto con el fin de poder establecer al módulo como transmisor o receptor. [26]

Figura 4-3: Transmisión Diferencial en comunicación Modbus por pines A y B para eliminación de ruido.



Fuente: [26]

Figura 4-4: Esquema de conexión Arduino mega y medidor Eastron SDM230.



Fuente: [Elaboración propia]

Mediante la conexión del anterior circuito electrónico y la configuración de los parámetros (Anexo 5), registros y funciones adecuados para lectura de los datos

con la librería SimpleModbusMasterSDM120.h (Anexo 5) se pueden obtener las variables de: voltaje, corriente, potencia activa, potencia reactiva, potencia aparente, frecuencia, factor de potencia y energía tanto activa como reactiva.

Adicionalmente en este proyecto se implementa con el Arduino Mega2560 una medición alternativa de voltaje y corriente no invasiva, utilizando un transformador de corriente y las librerías EmonLib.h" y "FreqPeriodCounter.h" (Anexo 5) por lo tanto, se establecen dos fuentes de medición en el sistema principal, ambas en el mismo punto de conexión con el objetivo de poder comparar los resultados obtenidos con los datos que proporciona el medidor Eastron SDM230 y de esta manera poder hacer implementaciones futuras sin necesidad del medidor comercial.

Para él envió de toda la información a internet es necesaria la utilización de un dispositivo como el módulo Wifi nodemcu esp8266 que nos permita esta conexión, por lo que se hace pertinente la comunicación entre este módulo y el Arduino mega2560, el protocolo de comunicación escogido para este fin es el protocolo I2C.

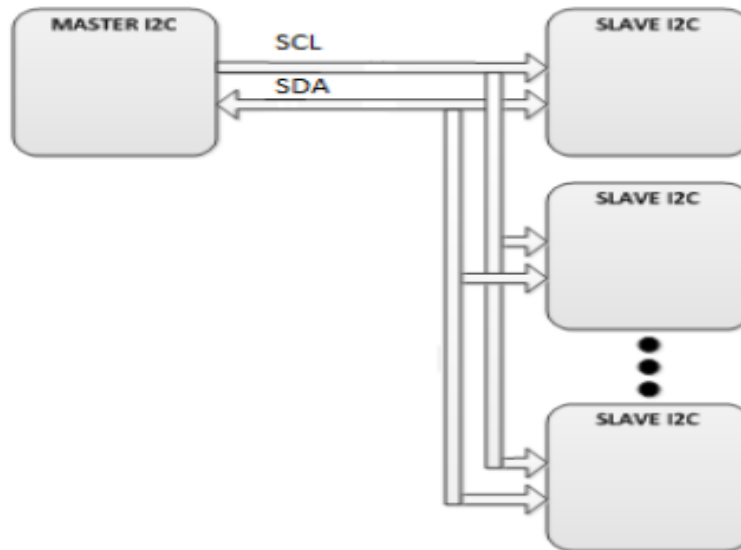
4.1.1.2 Protocolo I2C

Fue desarrollado por Philips Semiconductor en el año 1992 como un método para intercambiar información entre diferentes circuitos integrados. Se trata de un bus serie síncrono que emplea únicamente 2 líneas para las transmisiones entre los sistemas maestro y esclavos presentes en la comunicación, la transferencia de datos se realiza de forma secuencial, enviándose un único dato en cada periodo del reloj de sincronización.

En el bus cada dispositivo dispone de una dirección única que permite acceder a ellos de forma individual. Esta dirección puede ser fijada por hardware o totalmente por software [26].

En la Figura 4-5 se puede ver el esquema de conexión entre dispositivos maestro y esclavos que se comunican mediante el protocolo I2C.

Figura 4-5: Conexión entre maestro y esclavos.



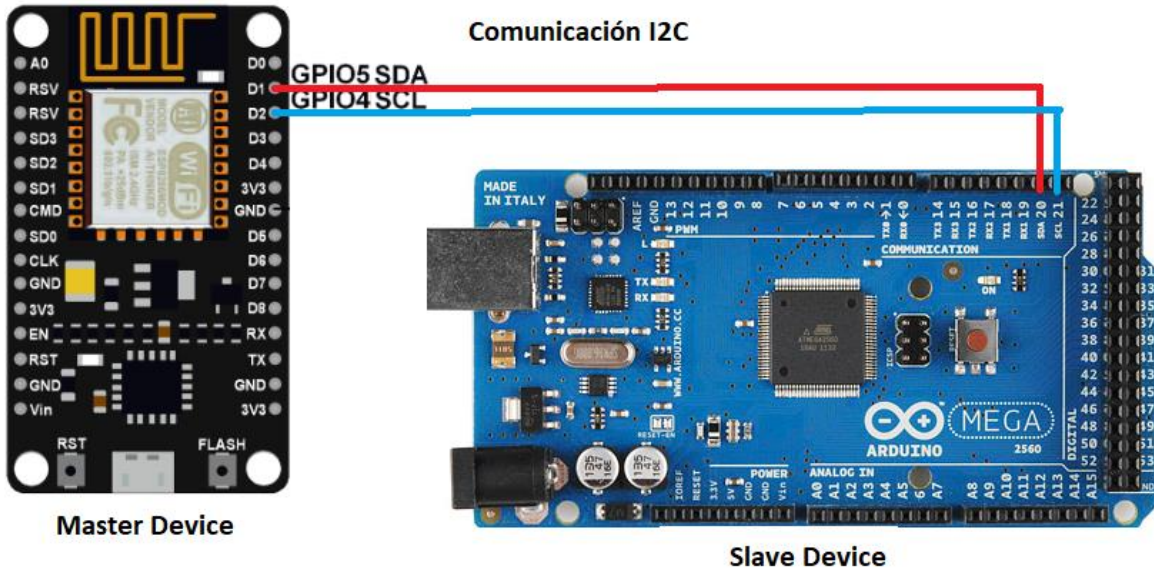
Fuente: [26]

El dispositivo maestro inicia la comunicación con los esclavos y puede enviar o recibir datos de ellos, pero los esclavos no pueden iniciar la comunicación (el maestro tiene que preguntarles), ni interactuar entre sí directamente [26].

4.1.1.3 Librería Wire

Esta librería permite la comunicación de la placa Arduino con dispositivos externos, u otras placas mediante el protocolo I2C. La comunicación se establece a través de los pines SDA (línea de datos) y SCL (línea de reloj) como se puede ver en la figura 4-6.

Figura 4-6: Comunicación I2C entre Nodemcu ESP8266 (Master Device) y Arduino Mega2560 (Slave Device).

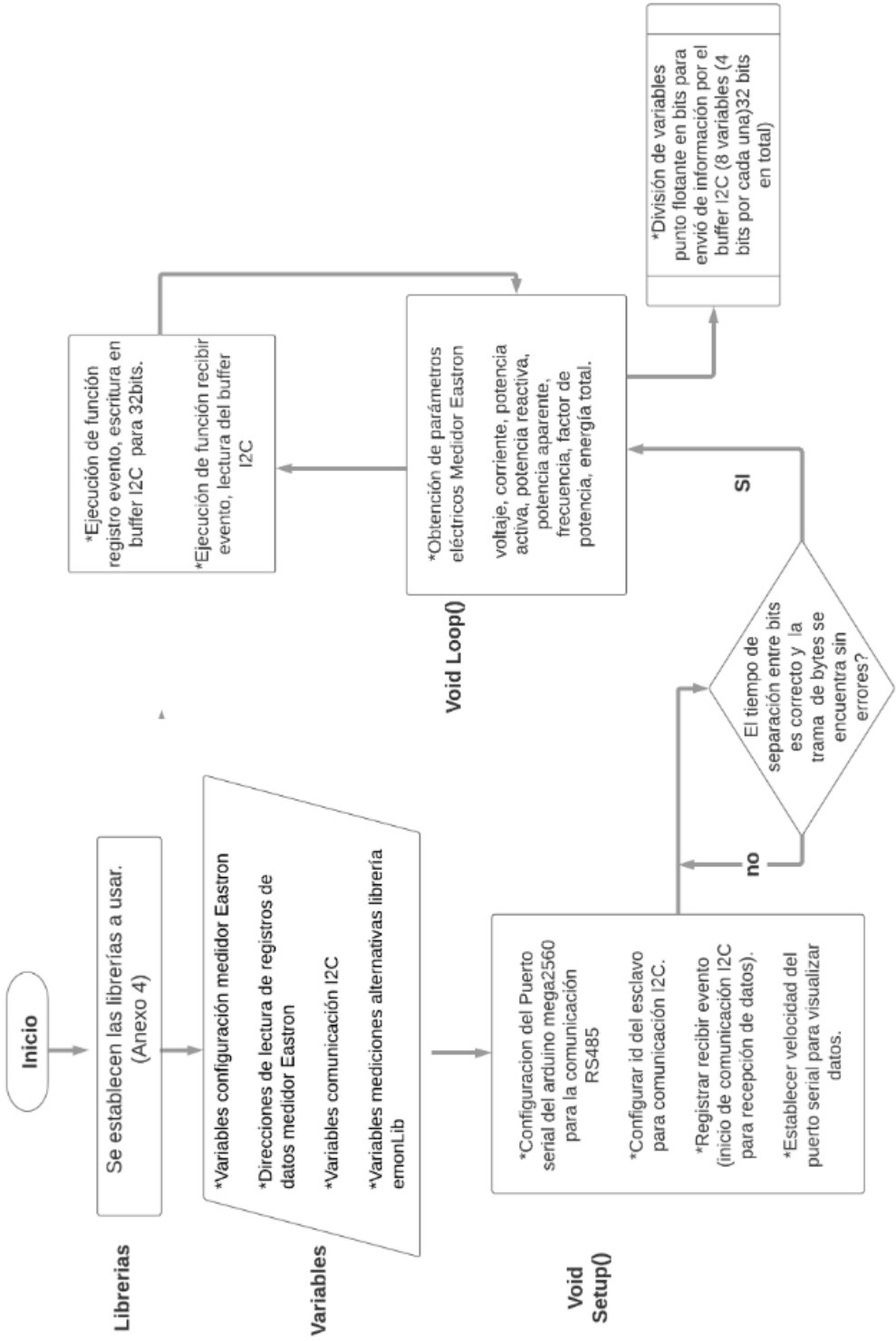


Fuente: [Elaboración propia]

Para el envío de la información mediante el protocolo I2c se hace necesario configurar el id del esclavo e inicializar las funciones `Wire.onRequest ()`, `Wire.onReceive ()` como se muestra en el diagrama de flujo de la Figura 4-7, también, enviar en el Void Loop (función que se ejecuta un número infinito de veces) cada una de las variables (8 parámetros eléctricos) divididas en diferentes bits debido a que son variables tipo flotante, por lo que ocupamos el tamaño total del buffer de comunicación I2C que máximo acepta 32 bits.

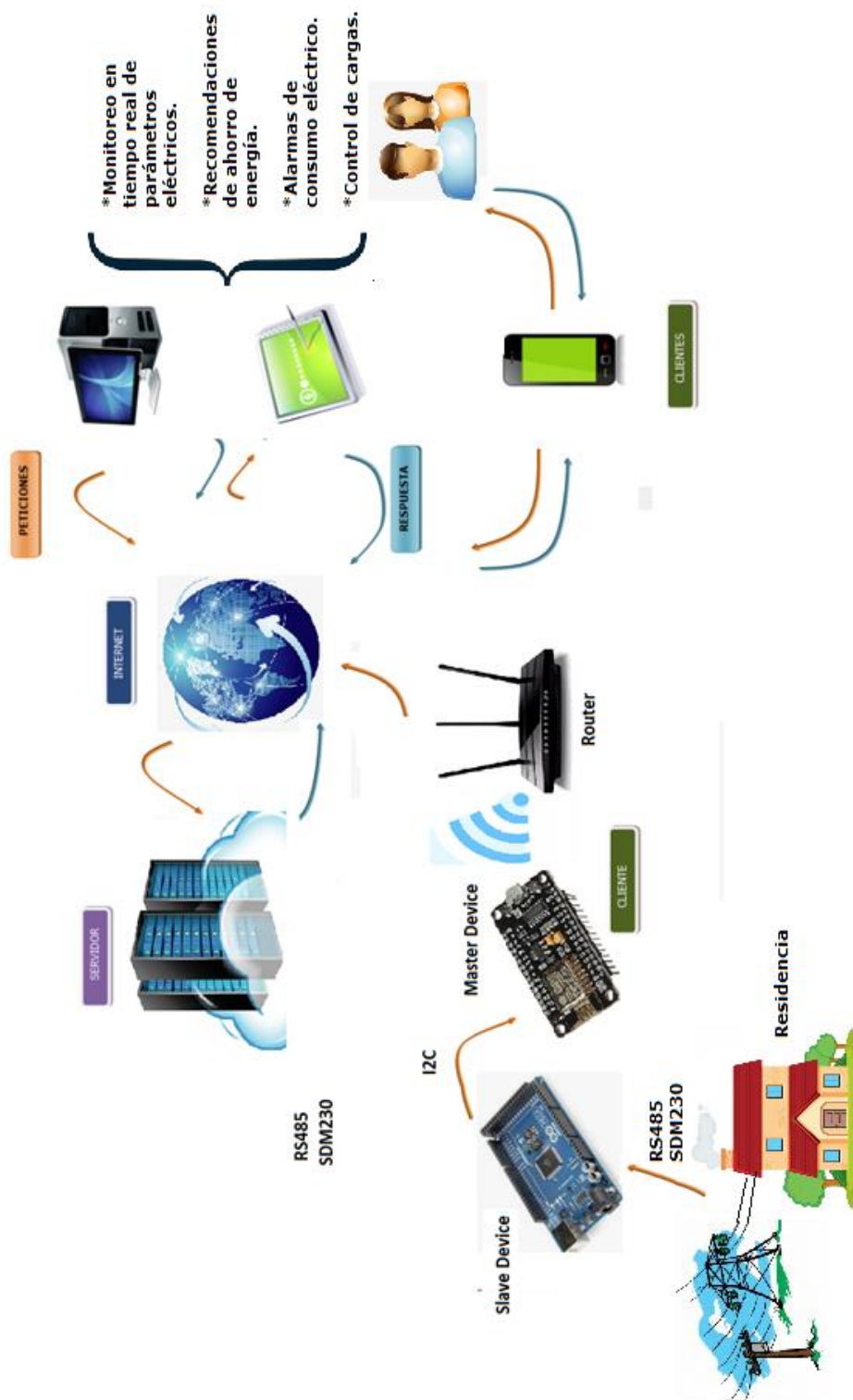
El esquema de conexiones entre el dispositivo maestro (la placa Nodemcu), el esclavo (Arduino Mega 2569) y la comunicación entre dispositivos clientes y el servidor externo se pueden ver en la figura 4-7 y Figura 4-8 respectivamente.

Figura 4-7: Diagrama de flujo código Arduino Mega2560 Dispositivo Esclavo.



Fuente: [Elaboración propia]

Figura 4-8: Esquema Clientes – Servidor Externo.



Fuente: [Elaboración propia]

Para la programación de la placa NodeMcu ESP8266 se requiere las librerías que se muestran en el anexo 6, ESP8266WIFI.h y WiFiClient.h permiten la conexión a internet; Wire.h comunica la placa arduino con dispositivos que trabajan mediante el protocolo I2C; EmonLib.h y FreqPeriodCounter.h posibilitan la medición de corriente no invasiva y frecuencia, además de la lectura de salida de pulsos (pulsos/kWh) de medidores comerciales con fines de aprendizaje; ArduinoOTA.h permite programar los Sketch de Arduino en modo inalámbrico, lo que abre muchas posibilidades para el IOT (Internet de las Cosas), porque no siempre es fácil poder ir a ciertos lugares y utilizar un cable para reprogramarlos; PubSubClient.h proporciona un cliente para enviar y recibir mensajes simples de publicación / suscripción con un servidor que admite MQTT, protocolo de mensajería ligero ideal para dispositivos pequeños e IOT.

Adicionalmente se configuran las variables del anexo 7 para la correcta conexión a internet y transmisión de los datos, para esta configuración se debe tener en cuenta:

- Las credenciales de autenticación de la red wifi, la direcciones ip del servidor externo, en este caso se utilizaron los servicios de hosting de ifastnet.com
- La configuración para la conexión con la plataforma thingspeak, la cual se utilizó con fines de aprendizaje y verificación de datos.
- La conexión con el broker CloudMQTT.

Cabe mencionar, que se realizó la conexión con el servicio web de IFTTT (plataforma software que se utiliza para gestionar y comunicar dispositivos IoT entre sí) y con otro servicio web, el de hojas de cálculo de Google (Spreadsheets) para almacenar los datos sin utilizar ningún complemento de plataformas de IoT en la nube.

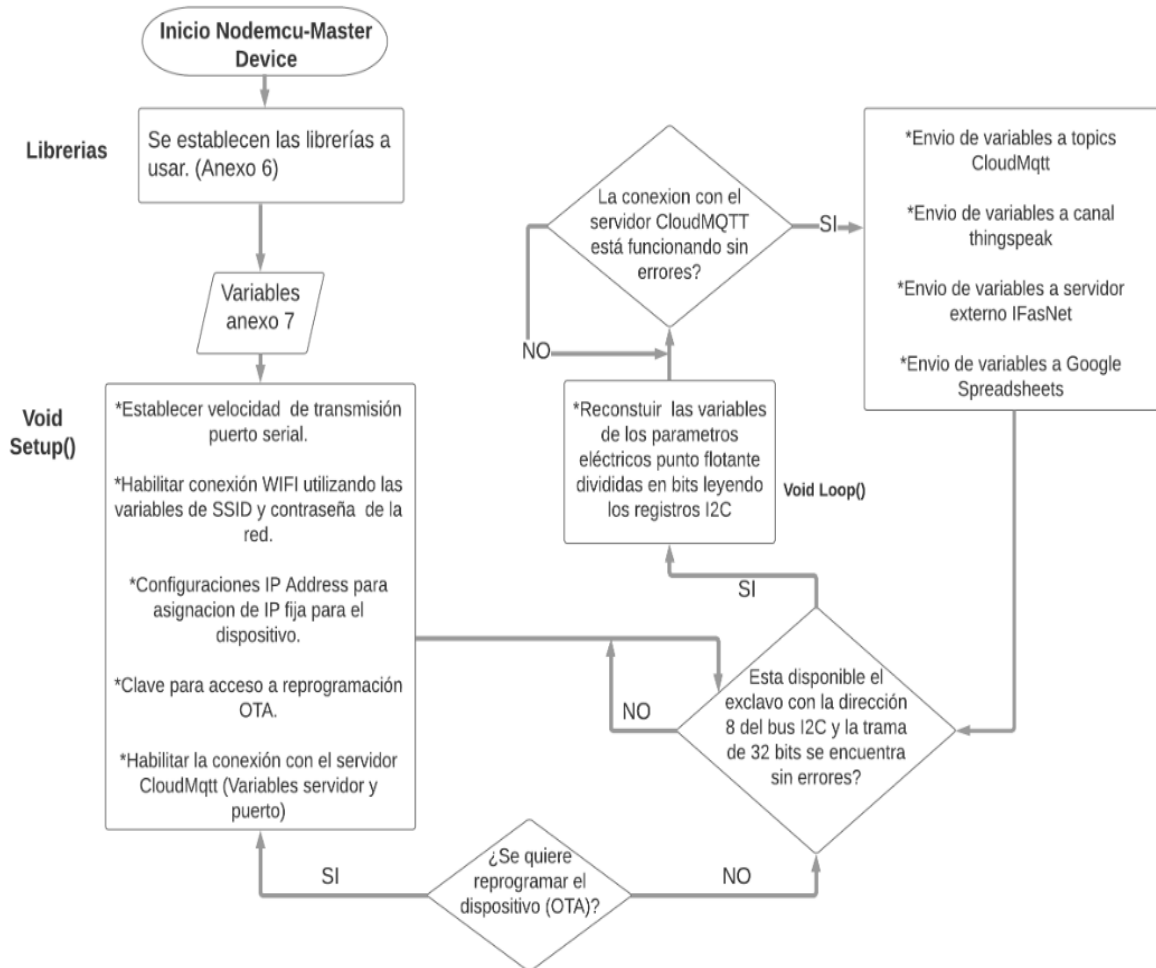
En el VOID SETUP (primera función en ejecutarse dentro de un programa en Arduino, solo se ejecuta una vez) del dispositivo Maestro (Nodemcu) se realiza la configuración para la conexión wifi y las credenciales de acceso, proporcionando una dirección IP fija para el dispositivo, lo cual permite identificarlo al momento de programarlo inalámbricamente mediante OTA (Over The Air, Figura 4-9).

4.1.1.4 Programación OTA (Over the Air)

La programación por aire se refiere a una actualización de software que se distribuye en Wi-Fi o banda ancha móvil usando una función incorporada en el sistema operativo, con el aspecto "OTA" refiriéndose a su uso de internet

inalámbrico en lugar de que el usuario requiera conectar el dispositivo a un ordenador a través de USB para realizar la actualización [27].

Figura 4-9: Diagrama de flujo código Nodemcu Dispositivo Maestro



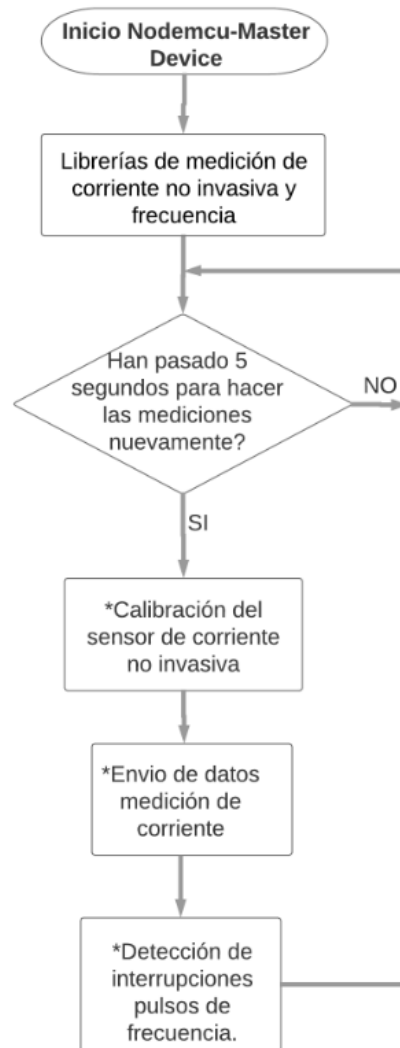
Fuente: [Elaboración propia]

En el dispositivo maestro se establece la dirección del esclavo al cual se quiere consultar la información y se procede a recuperar cada uno de los bytes del arreglo para estructurar nuevamente las variables de los parámetros eléctricos (Figura 4-9).

Se llevo a cabo la medición de corriente y frecuencia en el dispositivo maestro (Nodemcu esp8266) utilizando las librerías “emonlib.h” y “FreqPeriodCounter.h” como se ve en el diagrama de flujo de la Figura 4-10. Esto permite hacer una comparación entre los parámetros eléctricos proporcionados por el Medidor Eastron

Sdm230 y las mediciones obtenidas por la librería emonlib para de esta manera poder realizar a futuro mediciones más exactas con un desarrollo propio de hardware para este fin.

Figura 4-10: Medición de Corriente no invasiva y frecuencia con dispositivo maestro.



Fuente: [Elaboración propia]

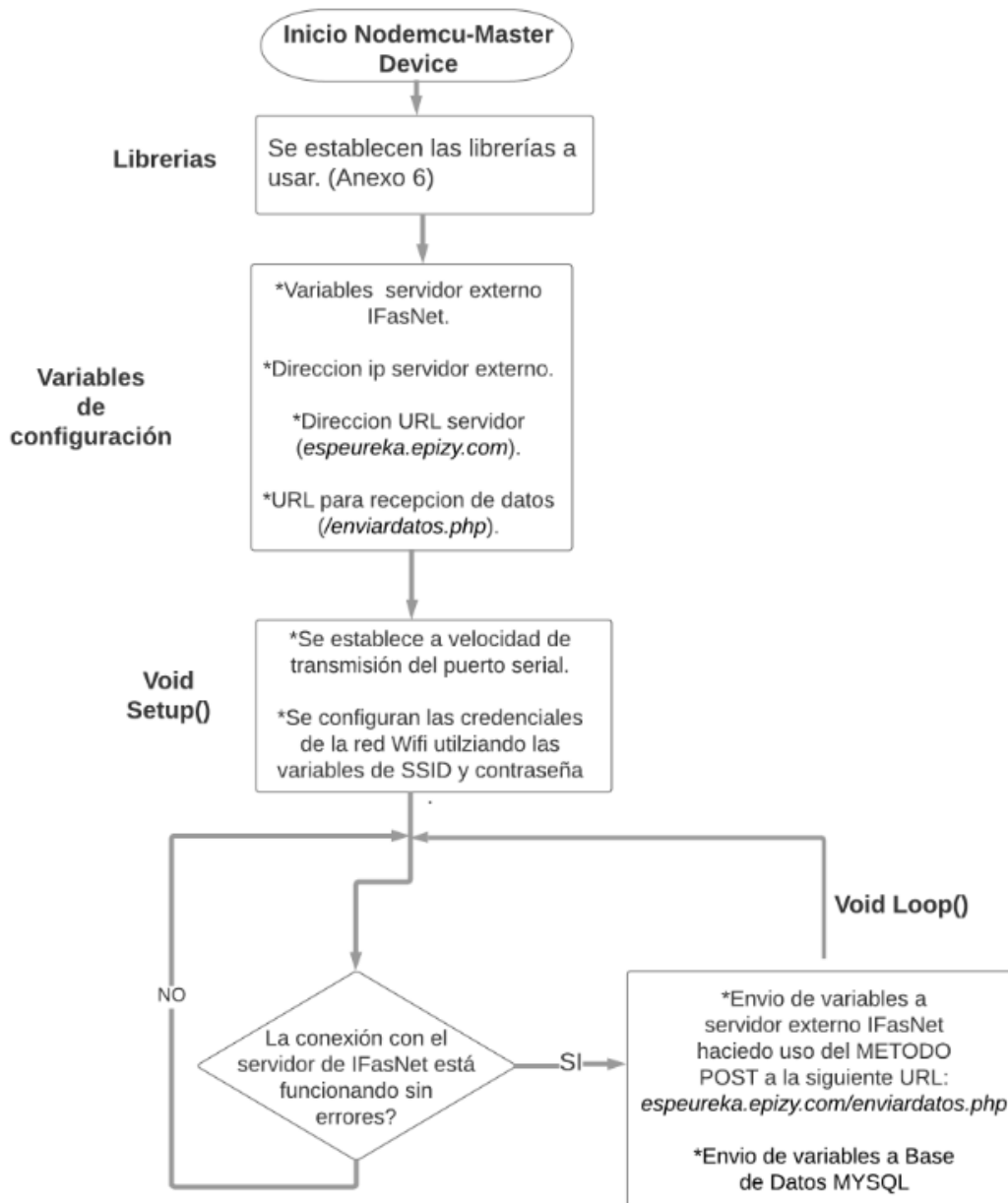
Se utilizó el Nodemcu ESP8266 como cliente, lo cual permite al dispositivo conectarse a un servidor externo y poder enviar o pedir información, en este caso se utilizaron los servicios de hosting de la plataforma IFastNet para alojar todos los recursos necesarios para el funcionamiento de la página web y la Api Rest para el consumo de los datos desde la aplicación Android.

Usar el Nodemcu ESP8266 como cliente significa que esta placa de desarrollo debe conectarse al servidor de IFastNet y hacer uso del protocolo HTTP, de la misma forma en la que se comunican los navegadores web (Chrome, Mozilla, etc) con los servidores.

Existen dos formas fáciles de enviar datos junto con una petición HTTP, estas son el método GET y POST, La diferencia principal entre estos ellos es que en el primero envía los datos en la URL, quedando visible para el usuario, mientras que en el segundo los datos quedan ocultos. Ambos son muy usados y casi siempre están implementados en las páginas webs en las que se navega [9].

En este caso en particular para la comunicación entre el hardware implementado y el servidor alojado en IFasNet se utiliza el Método POST a la siguiente url: *"espeureka.epizy.com/enviardatos.php"* para guardar la información en la base de datos MYSQL que nos proporciona esta plataforma (Figura 4-11).

Figura 4-11: Diagrama de flujo conexión a servidor externo IFastNet. Nodemcu (Master Device).



Fuente: [Elaboración propia]

4.1.1.5 IFastNet

Es una plataforma diseñada para administrar de manera segura los servicios de alojamiento de información. Para contar con el servicio se debe pagar un costo

monetario mensual, el cual es bajo comparado con las otras plataformas de hosting web.

En el anexo 8 se muestra la estructura del archivo php que se encuentra alojado en la plataforma nombrada y permite guardar la información en la Base de Datos MYSQL (Figura 4-12).

Figura 4-12: Parámetros eléctricos Nodemcu, BD MYSQL. IFasNet.



The screenshot shows a MySQL database interface. At the top, there are navigation buttons: Examinar, Estructura, SQL, Buscar, Insertar, Exportar, Importar, Operaciones, and Disparadores. Below this, a green status bar indicates 'Mostrando filas 2412750 - 2412770 (total de 2412771, La consulta tardó 0.1350 segundos.)'. The SQL query 'SELECT * FROM `tabla`' is visible. Below the query, there are options for 'Perfilando', 'Editar en línea', 'Editar', 'Explicar SQL', and 'Crear código PHP'. A navigation bar shows '96511' in a dropdown, 'Número de filas: 25', 'Filtrar filas: Buscar en esta tabla', and 'Ordenar según la clave: Ninguna'. Below this is a table with columns: id, chipid, fecha, temperatura, corriente, potenciaaparente, and potenciareactiva. The table contains three rows of data.

	id	chipid	fecha	temperatura	corriente	potenciaaparente	potenciareactiva
<input type="checkbox"/>	2412804	10783191	2018-11-18 12:57:53	0	1.47	191.06	-51.7
<input type="checkbox"/>	2412805	10783191	2018-11-18 12:57:59	0	1.47	191.06	-49.66
<input type="checkbox"/>	2412806	10783191	2018-11-18 12:58:05	0	1.75	160.07	-49.66

Fuente: [Elaboración propia]

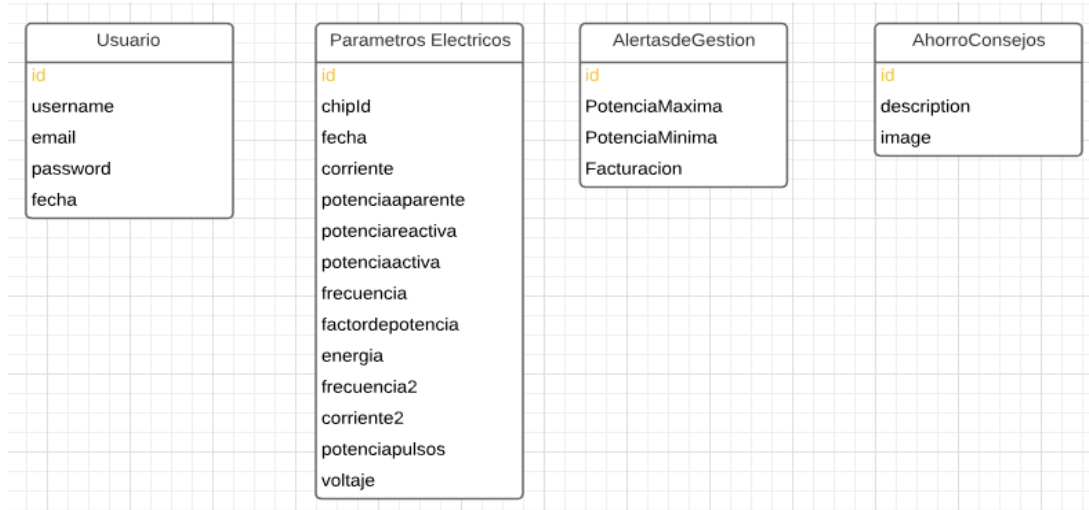
4.1.1.6 Base de Datos MySQL

Las bases de datos son bancos de información que contienen datos relativos a diversas temáticas, es decir un sistema informático a modo de “almacén” que permite guardar grandes volúmenes de información de forma organizada para que se pueda encontrar y utilizar fácilmente. constituyen una parte fundamental de los sistemas de información en las que están integrados [9].

MySQL, es un sistema de gestión de bases de datos relacionales de código abierto el cual soporta gran cantidad de estos mismo (algunas tienen hasta 50 millones de registros), se utiliza en aplicaciones que requieren centralizar información, con datos que se pueden estructurar en tablas, con campos definidos, con índices y llaves que relacionan las tablas entre sí. Gran parte de su lenguaje de programación es realizado en SQL. Provee gran seguridad con la conectividad de la base de datos ya que ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host, además el tráfico de contraseñas está cifrado al conectarse a un

servidor. Se puede instalar prácticamente en cualquier sistema operativo: Unix, Linux, Windows, MacOS; solo o acompañado de otros programas del entorno de programación como el servidor web Apache, PHP y Phpmysql; en las distribuciones WAMP, LAMP o XAMPP [12]. En la Figura 4-13 se puede ver la estructura de la base de datos MySQL la cual es gestionada directamente por el servidor.

Figura 4-13: Estructura Base de Datos MySQL.

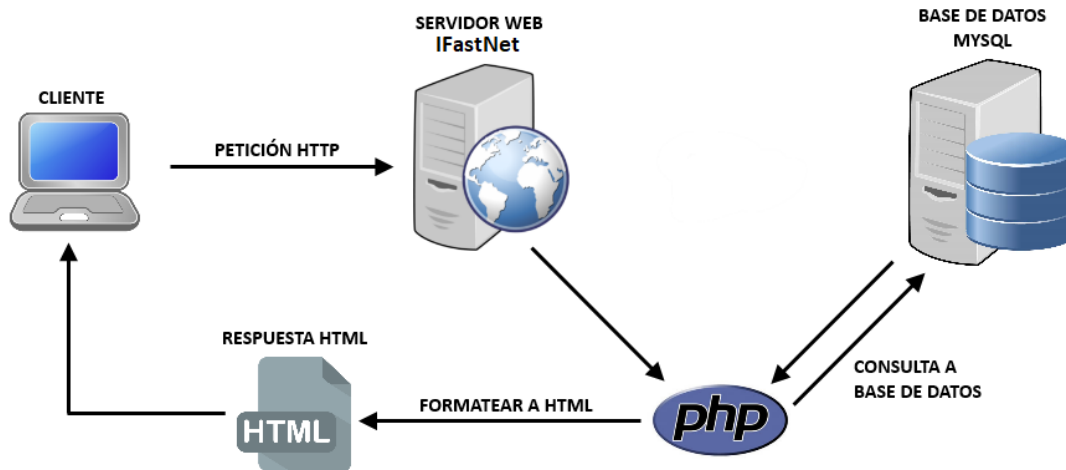


Fuente: [Elaboración propia]

4.1.1.7. Lógica del servidor

Como ya se había mencionado anteriormente, los datos enviados desde el hardware desarrollado para este proyecto son recibidos, procesados y almacenados por el servidor externo IFastNet en la base de datos MySQL, mediante el método POST al archivo enviardatos.php (anexo 8). Además, el servidor se encarga de gestionar las peticiones de los clientes a las diferentes rutas para acceder a la página web como se ve en la figura 4-14, por lo que se diseñan las funcionalidades de registro y login de usuarios, dashboard y visualización de parámetros eléctricos (estos se actualizan cada 3 segundos).

Figura 4-14: Petición a Servidor IFasNet y BD.



Fuente: [27]

4.1.2. CONEXIÓN A SERVICIOS WEB

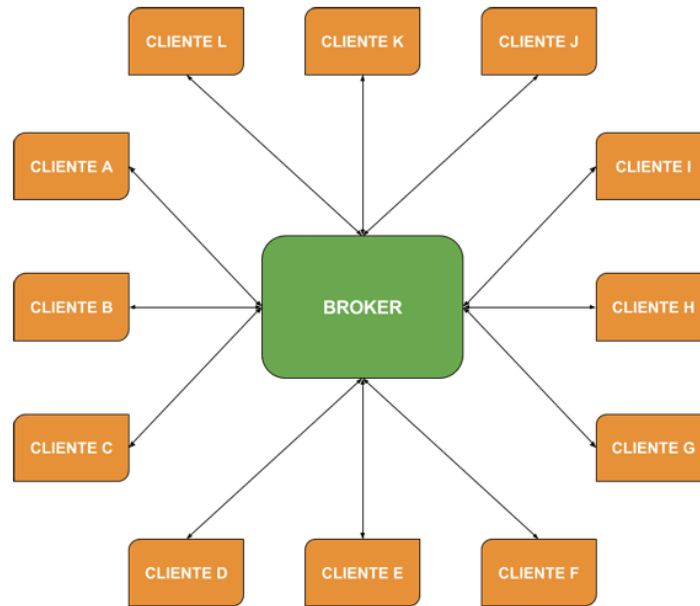
Adicionalmente, se realizó la conexión del hardware con diferentes plataformas software que se utilizan para gestionar y comunicar dispositivos IoT, entre las cuales inicialmente está el broker CloudMQTT, los servicios web de IFTTT, la plataforma thingspeak y el servicio web de hojas de cálculo de Google (Google Spreadsheets).

4.1.2.1 CloudMQTT.

La utilización del protocolo MQTT permite que las máquinas se comuniquen entre ellas (M2M), tiene la ventaja de consumir muy poco ancho de banda y puede tener una comunicación bidireccional, por lo que los dispositivos pasan de tener una iteración punto a punto a una más sofisticada donde se establecen verdaderos diálogos entre las máquinas mediante publicación/suscripción [22].

Los dispositivos se conectan utilizando una topología en estrella, es decir, todos los clientes se conectan directamente a un punto central que hace de servidor llamado Broker (Figura 4-15).

Figura 4-15: Broker MQTT.



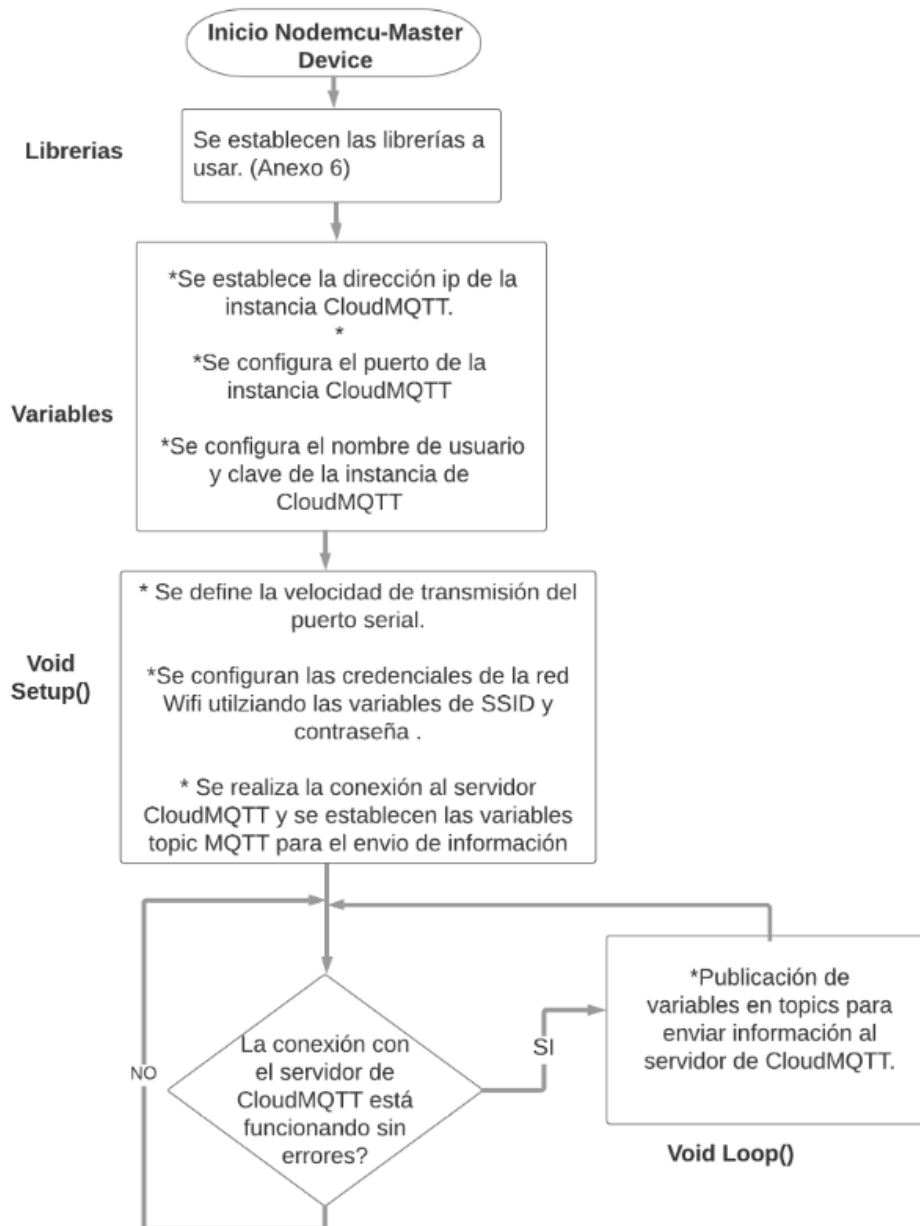
Fuente: [22]

El Broker es la parte central de la arquitectura y todos los mensajes pasan por él. debido a que MQTT utiliza una arquitectura publicación/suscripción, es necesario que el Nodemcu Esp8266 primeramente se conecte con el servidor externo que nos servirá de bróker, en este caso un servidor Mosquito administrado en la nube [19], [22].

La plataforma CloudMQTT nos permite crear instancias y escoger un centro de datos de Amazon web service (AWS) para alojarlas, además, nos proporciona las credenciales tanto de dirección del servidor, usuario, contraseña y puertos para realizar la comunicación con el hardware utilizado [19], características que además del bajo costo mensual de \$15.000 pesos comparado con otros bróker en la nube, permitieron escoger este servicio como canal para la transmisión de datos utilizando el protocolo MQTT.

La tarjeta Nodemcu ESP8266 se conecta con las credenciales que proporciona CloudMQTT (Variables Figura 4-16) y envía los datos de los parámetros eléctricos a diferentes topic a los que publica como cliente para enviar la información (Void Loop () Figura 4-16).

Figura 4-16: Diagrama de flujo conexión a servidor CloudMQTT. Nodemcu (Master Device).

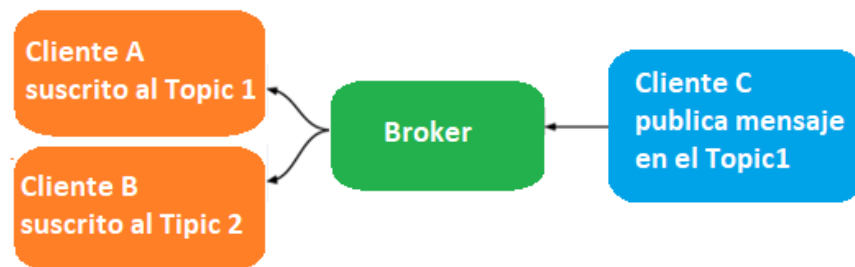


Fuente: [Elaboración propia]

Cabe mencionar que un topic es el tema, a quién va dirigido el mensaje, pero este no tiene destinatario concreto, puede haber uno, muchos o ninguno. El emisor, el que envía el mensaje, no sabe a quién va dirigido dicho mensaje, sólo el Broker lo sabe. Cuando llega un nuevo mensaje se lo envía a aquellos que se han suscrito al topic como se ve en la figura 4-17, el mensaje puede ser cualquier cosa, el formato

es totalmente libre, lo decide el programador y el tamaño máximo depende de la implementación del protocolo MQTT.

Figura 4-17: Topic, Broker MQTT y clientes publicación/suscripción.



Fuente: [22]

La plataforma también permite mediante su WebSocket UI poder probar y depurar la conexión con la tarjeta Nodemcu ESP8266 para así visualizar la información (cada 3 segundos se actualiza) suministrada por el dispositivo, además CloudMQTT admite MQTT en WebSockets y permite que cada navegador sea un cliente MQTT, por lo tanto se optó por desarrollar en los lenguajes de programación HTML, CSS y JavaScript un cliente de esta tipo, de esta manera se envía mensajes desde el navegador al dispositivo Maestro para generar eventos y recibir los datos que se están censando.

Para lograr la conexión con el websocket es necesario utilizar en el código HTML la librería mqttws31.js y de esta manera poder hacer el enlace con el servidor de CloudMQTT (Anexo 9), en donde se debe utilizar necesariamente el puerto WebSocket TLS.

4.1.2.2. WebSocket.

“Es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor. El estándar WebSocket simplifica gran parte de la complejidad en torno a la comunicación web bidireccional y la gestión de la conexión” [22].

En el código HTML se deben utilizar las siguientes funciones para recibir los datos y suscribirse a los diferentes topic que se han definido para el desarrollo (Anexo 10).

En el desarrollo web tal como se ve en la cabecera del código HTML(<head> Anexo 10) se utiliza la librería de Bootstrap la cual facilita la maquetación de sitios web, además de ofrecer diferentes herramientas para que el contenido web se vea bien en toda clase de dispositivos para así poder dar un aspecto visual más llamativo y cómodo para el usuario, también se utiliza la librería highcharts que permite realizar las gráficas en de los parámetros eléctricos, estas se actualizan cada 3 segundos.

Adicionalmente gracias a la comunicación bidireccional mediante el Websocket el usuario puede controlar fácilmente mediante dos botones en la interfaz de usuario web y móvil el accionamiento de cargas de hasta 30 amperios utilizando un relé Sla, y cargas de hasta 75 amperios utilizando un relé de estado sólido SSR, todo esto el fin de poder cortar el suministro de energía eléctrica a determinadas cargas y reducir el consumo eléctrico del hogar.

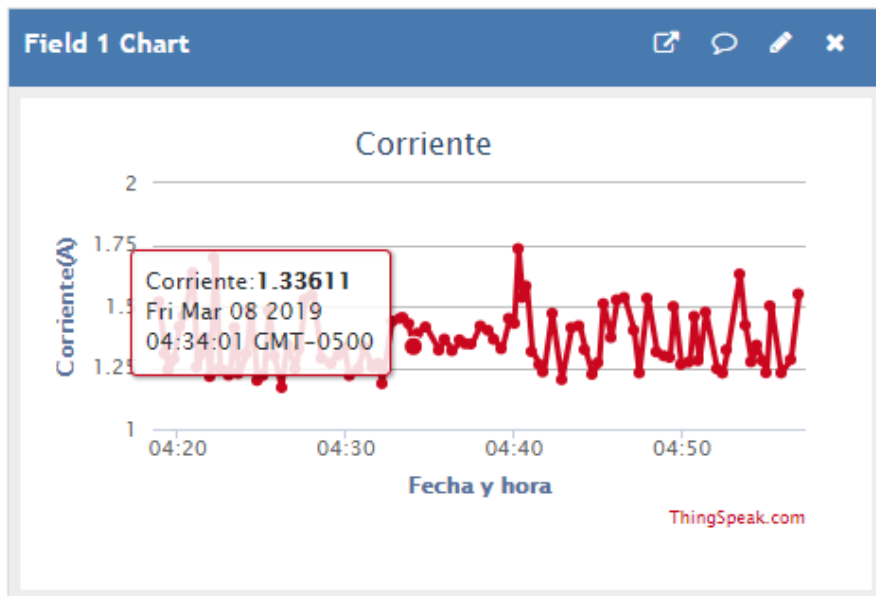
4.1.2.3. Thingspeak.

Permite desarrollar una aplicación visual para mostrar datos en pocos minutos, para lo cual utiliza una cuenta de usuario y un canal, en el cual se envía (cada 15 segundos) y almacena la información; puede tener hasta 8 campos de datos, de ubicación y uno de estado [19] y [20]. Estas características y su fácil configuración hardware permitieron escoger este servicio como plataforma rápida de verificación de datos.

En el desarrollo de este proyecto se utilizó esta plataforma con fines de aprendizaje y verificación de datos obtenidos en la conexión con el broker CloudMQTT, y para su funcionamiento se debe configurar en la programación del hardware las credenciales de acceso, y mediante una petición HTTP utilizando el método POST se realiza el envío de los datos a este servidor web. En la figura 4-31 se presenta la gráfica de los parámetros eléctricos del Nodemcu en la plataforma.

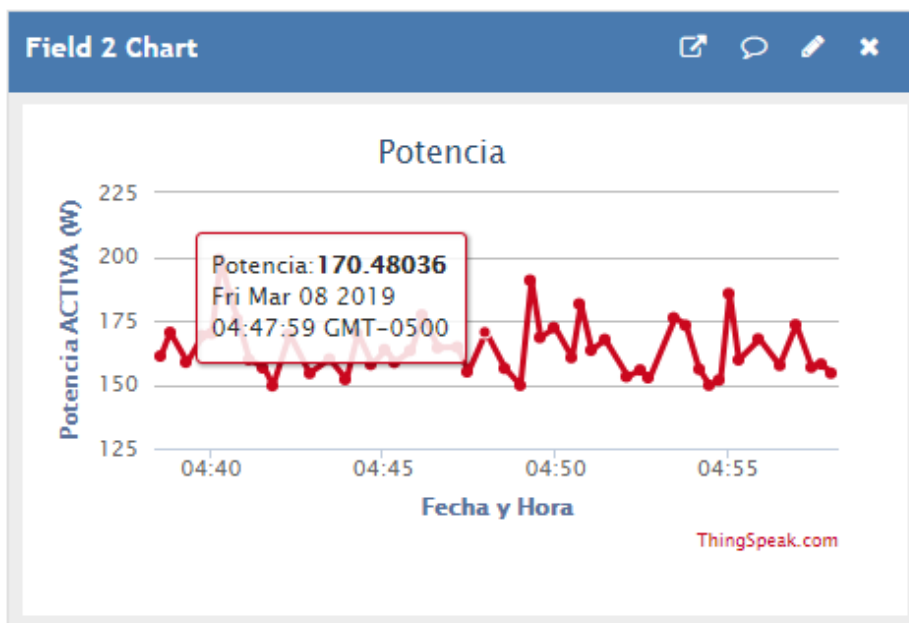
En las figuras 4-18, 4-19 y 4-20 se presentan las gráficas en la plataforma ThingSpeak de algunos de los parámetros eléctricos (Corriente, potencia activa, energía y potencia reactiva) obtenidos por el sistema principal implementado.

Figura 4-18: Gráfica de Corriente en ThingSpeak.



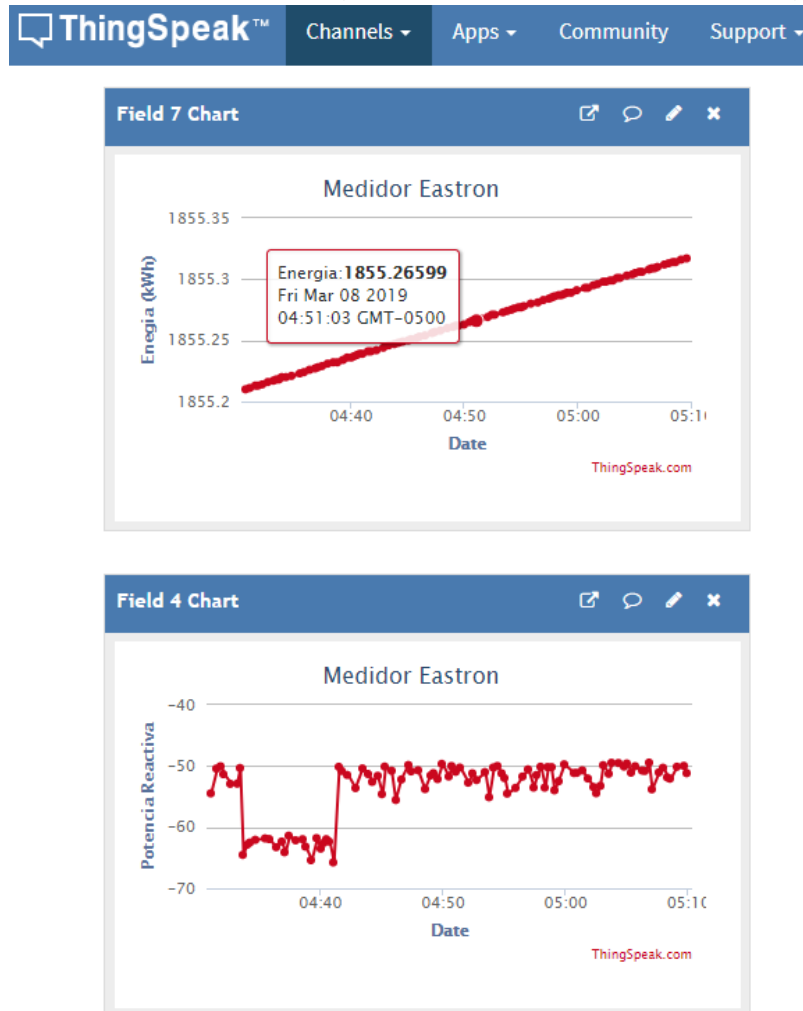
Fuente: [Elaboración propia]

Figura 4-19: Gráfica de Potencia Activa en ThingSpeak.



Fuente: [Elaboración propia]

Figura 4-20: Gráfica de Energía y Potencia Reactiva en ThingSpeak.



Fuente: [Elaboración propia]

4.1.2.4. IFTTT

IFTTT es una plataforma que permite conectar diferentes aplicaciones web o servicios de Internet de manera muy sencilla, a través de la lógica de “If This Then That (IFTT)”, es decir “Si ocurre esto, haz esto otro” [28]. Esta plataforma permite configurar un disparador o un evento que desencadenará una reacción, por ejemplo, imagina que cuando llueva en tu ciudad, quieres que te avise a través de un email o incluso que te cambie la hora del despertador para despertarte más tarde o más pronto, Todo gracias a los servicios web [28].

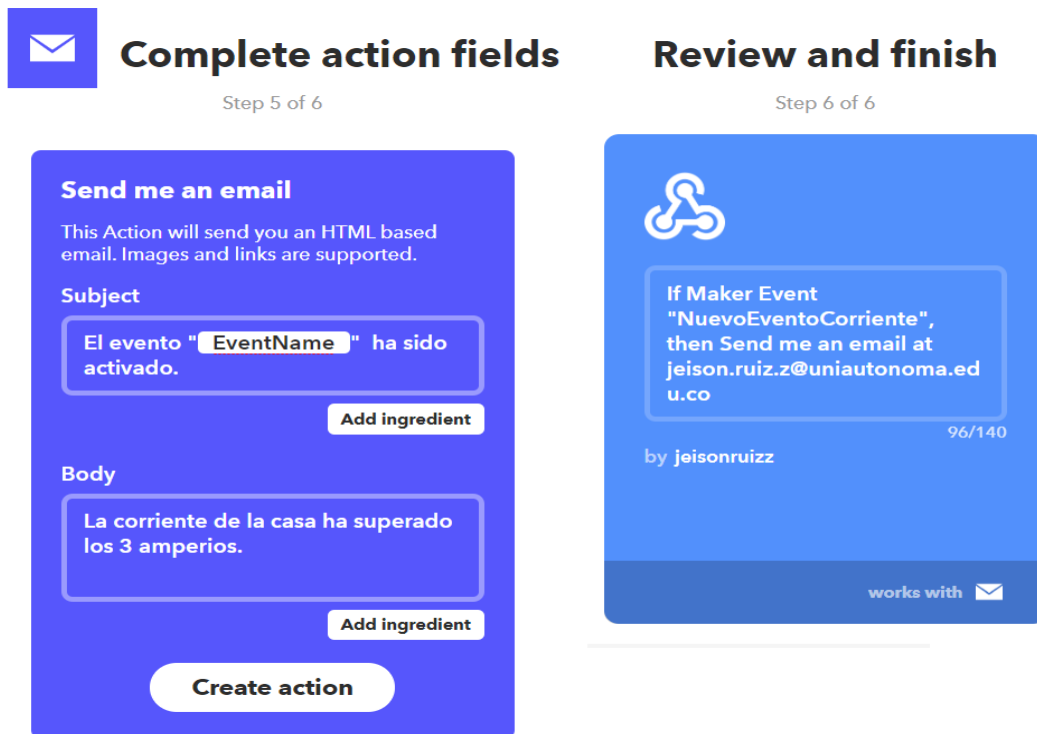
En IFTTT hay servicios específicos que se pueden utilizar para tener un enlace con Facebook, Instagram, Apps de Google, para Android, para iOS, Alexa, Google

Home, Adafruit y un sin fin de aplicaciones más. Solo tienes que pensar qué quieres hacer y buscar el servicio que mejor se ajuste a tus necesidades [28].

Para la implementación de los servicios de IFTTT en este proyecto se realizó una conexión con la plataforma Thingspeak la cual mediante su aplicación de ThingHTTP permite la comunicación entre dispositivos, sitios y servicios web sin tener que implementar el protocolo en el nivel del dispositivo, por lo que inicialmente se configura en la cuenta de IFTTT una nueva receta de automatización (If +this then that) utilizando como primer parámetro (If +this) el servicio de Webhooks (Función que invoca un Web Service) y como segundo parámetro los servicios tanto de Gmail como de Facebook (Then that).

Al momento de crear la nueva receta y escoger como primer parámetro el Webhook es necesaria la creación de un disparador para el nuevo evento y también configurar el asunto, mensaje y correo electrónico al cual se enviarán las notificaciones de esta acción IFTTT para el segundo parámetro del servicio de Gmail como lo muestra la figura 4-21.

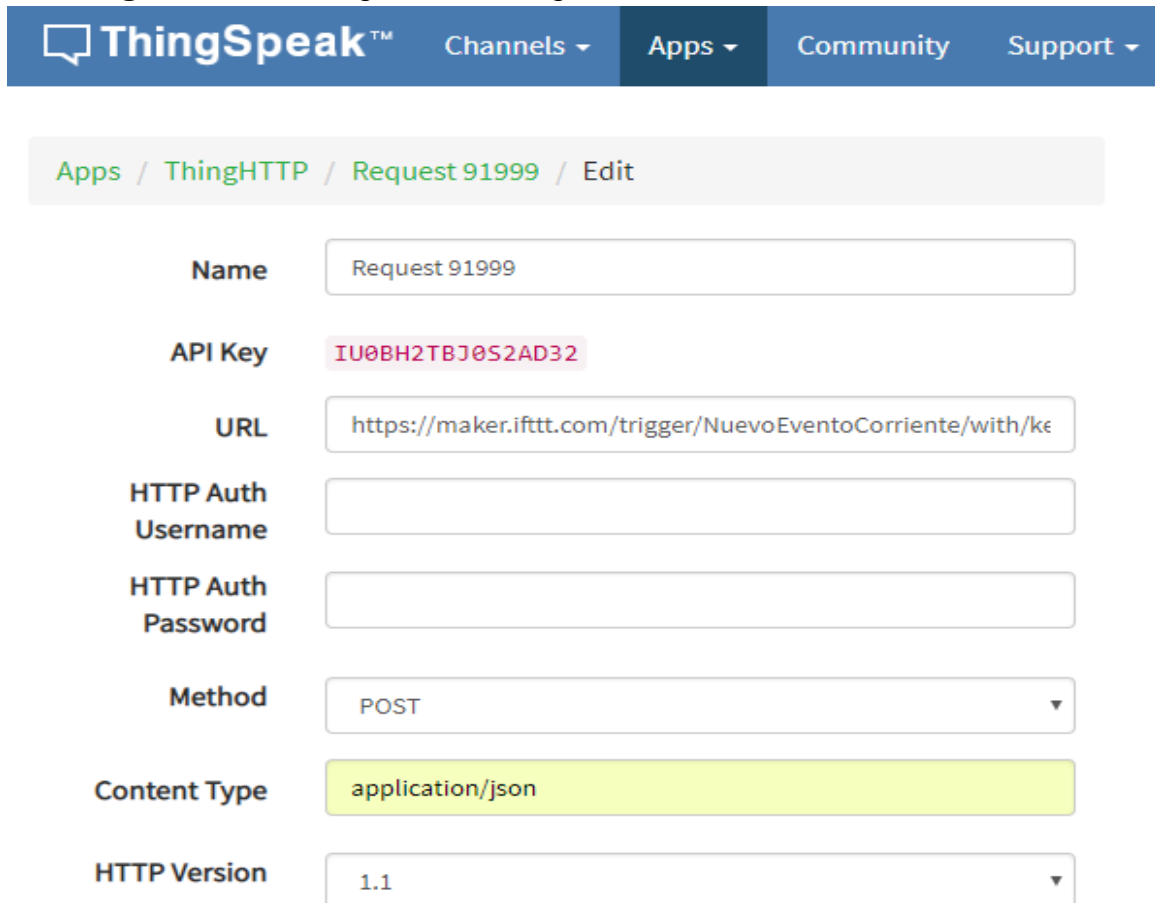
Figura 4-21: Creación de nueva receta IFTTT.



Fuente: [Elaboración propia]

Posterior a la creación de la nueva receta el Webhook brinda la siguiente URL <https://maker.IFTTT.com/trigger/NuevoEventoCorriente/with/key/cham6tQ8s8B0f8eeu1SISZ> para ser invocada en ThingHTTP mediante una petición HTTP utilizando el método POST configuración que se puede ver en la figura 4-22.

Figura 4-22: Configuración ThingHTTP con URL de Webhook IFTTT.



The image shows the configuration interface for a ThingHTTP request on the ThingSpeak platform. The navigation bar at the top includes 'ThingSpeak™', 'Channels', 'Apps', 'Community', and 'Support'. The breadcrumb trail indicates the current page is 'Apps / ThingHTTP / Request 91999 / Edit'. The configuration fields are as follows:

Name	Request 91999
API Key	IU0BH2TBJ0S2AD32
URL	https://maker.ifttt.com/trigger/NuevoEventoCorriente/with/ke
HTTP Auth Username	
HTTP Auth Password	
Method	POST
Content Type	application/json
HTTP Version	1.1

Fuente: [Elaboración propia]

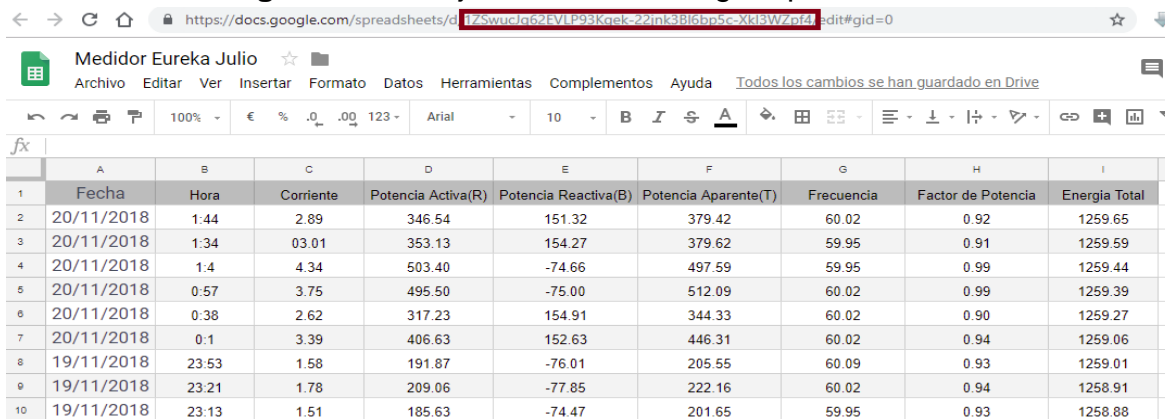
Adicionalmente, para que el hardware implementado tenga interacción con los servicios de IFTTT se configura una reacción en ThingSpeak (Anexo 11) para generar alarmas cuando los datos del canal cumplan determinada condición. En esta reacción se configura el tipo de dato de la condición, la frecuencia de prueba, el canal, el campo, el valor de umbral que se va a evaluar para la condición, la referencia del ThingHTTP que se va a utilizar y la frecuencia de ejecución de la reacción.

4.1.2.5. Google Spreadsheets (Hojas de cálculo de Google)

La integración de Google Spreadsheets con el dispositivo maestro (Nodemcu ESP8266) proporciona la posibilidad de almacenar los datos de los parámetros eléctricos obtenidos sin necesidad de utilizar ningún complemento de plataformas de IoT en la nube o bases de datos, permitiendo ahorrar recursos o gastos adicionales en mantener un servidor para guardar la información. Básicamente esta integración consta de dos partes, la primera parte es configurar el hardware ESP8266 con las credenciales de acceso para Google Spreadsheets utilizando como host= "script.google.com" y puerto 443, además de la huella digital del certificado SHA1 y el GAS_ID que es único dependiendo del id del script de Google que se vaya a ejecutar y la función encargada de enviar los datos mediante una petición HTTP utilizando el método GET.

La segunda parte consiste en crear una hoja de Google (Figura 4-23) y configurarla para aceptar los datos enviados desde el dispositivo maestro Nodemcu ESP8266 a través de Internet utilizando un script de aplicación de Google (Anexo 12), cabe resaltar que para la ejecución de este script es necesario el Id de la hoja de cálculo creada y este se encuentra en su dirección URL (<https://docs.google.com/spreadsheets/d/idUnico>).

Figura 4-23: Hoja de cálculo. Google Spreadsheets.



	A	B	C	D	E	F	G	H	I
1	Fecha	Hora	Corriente	Potencia Activa(R)	Potencia Reactiva(B)	Potencia Aparente(T)	Frecuencia	Factor de Potencia	Energía Total
2	20/11/2018	1:44	2.89	346.54	151.32	379.42	60.02	0.92	1259.65
3	20/11/2018	1:34	03.01	353.13	154.27	379.62	59.95	0.91	1259.59
4	20/11/2018	1:4	4.34	503.40	-74.66	497.59	59.95	0.99	1259.44
5	20/11/2018	0:57	3.75	495.50	-75.00	512.09	60.02	0.99	1259.39
6	20/11/2018	0:38	2.62	317.23	154.91	344.33	60.02	0.90	1259.27
7	20/11/2018	0:1	3.39	406.63	152.63	446.31	60.02	0.94	1259.06
8	19/11/2018	23:53	1.58	191.87	-76.01	205.55	60.09	0.93	1259.01
9	19/11/2018	23:21	1.78	209.06	-77.85	222.16	60.02	0.94	1258.91
10	19/11/2018	23:13	1.51	185.63	-74.47	201.65	59.95	0.93	1258.88

Fuente: [Elaboración propia]

4.1.3. Aplicación Móvil

La aplicación móvil desarrollada para este sistema de gestión de energía eléctrica permite al usuario del sector residencial realizar las siguientes tareas:

- Ver las mediciones de los parámetros eléctricos de las cargas a través de gráficas que permiten interpretar mejor determinada información, haciéndola más entendible y proporcionando una idea generalizada de los resultados.
- Realizar consultas de los datos en diferentes fechas y horas del día.
- Enterarse de cómo ahorrar energía eléctrica por medio de consejos e imágenes explicativas.
- Recibir notificaciones push (mensajes instantáneos) de noticias o nuevos consejos y tips de ahorro energético.
- Visualizar el estado y mediciones de los sistemas de gestión portátiles que tenga asociados.
- Consultar el gasto monetario en pesos (\$) en relación al consumo en kWh.
- Proporcionar el control del encendido o apagado de las cargas que estén conectadas a las salidas relé de los sistemas desarrollados.

Esta aplicación móvil fue desarrollada con el entorno de Android Studio, el cual que posee potentes herramientas de edición de código, además, contiene una interfaz de usuario que es construida o diseñada previamente, con variados modelos de pantalla, donde en ella los elementos existentes pueden ser desplazados; también, consta de un análisis de código que destaca los errores de forma inmediata, y proporciona una solución más rápida para estos [29].

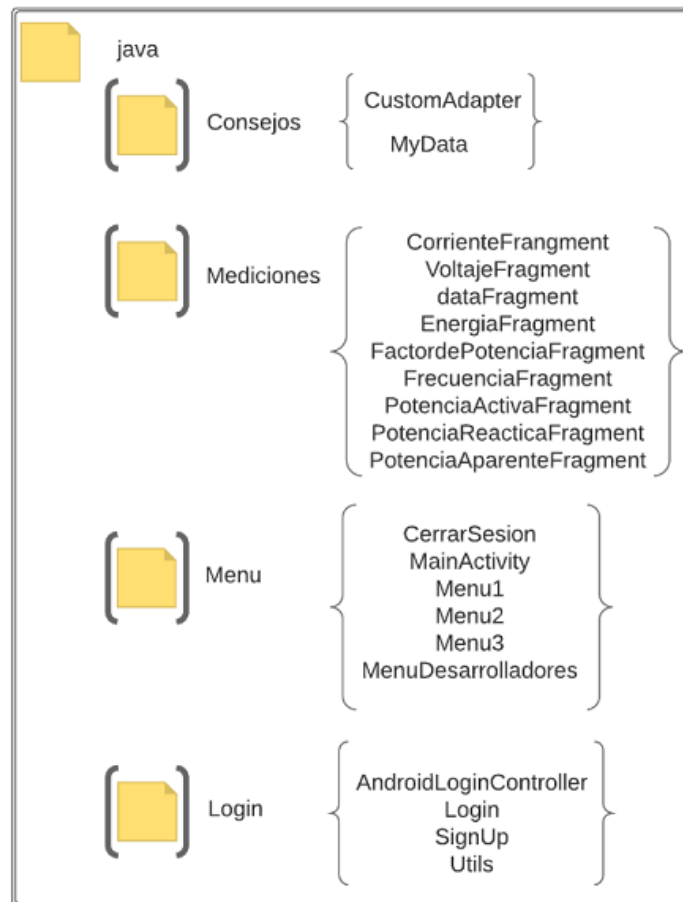
Para poder desarrollar la aplicación es necesario tener ciertos conocimientos sobre el lenguaje XML(eXtensible Markup Language), el cual es utilizado en este entorno de desarrollo para la elaboración de la parte del diseño de la interfaz, toda la apariencia gráfica, imágenes, audios, videos, botones, etc., son estructurados en un fichero XML e invocados posteriormente en una clase(.java) donde se establece la funcionalidad, esto es debido a que Android utiliza la estructura de modelo Vista Controlador(MVC) para el desarrollo de interfaces y de esta manera se separan los datos de la apariencia visual.

La estructura del proyecto Android está definida en una serie de carpetas divididas en los archivos del código fuente de la aplicación, así como sus recursos (Carpeta app, carpeta java y carpeta res) y el resto de archivos que corresponden a la configuración del proyecto (Carpeta Gradle Scripts), archivos de la herramienta Gradle la cual se encarga de la compilación automática del proyecto y demás archivos para el sistema de control de versiones [29].

El archivo AndroidManifest.xml como se ve en la Anexo 13 es el que contiene la información sobre las características generales de la aplicación y sus componentes, aquí se encuentran la estructura de ejecución de las actividades, services, intents, permisos y librerías externas necesarias para el correcto funcionamiento.

La carpeta java contiene todos los archivos de código fuente (.java) que se encargan de integrar el funcionamiento de la aplicación, aquí están todas las clases que permiten poder realizar la navegación e interacción de las diferentes vistas y actividades (Activities) desarrolladas para el proyecto, esto se puede ver en la figura 4-24.

Figura 4-24: Estructura carpeta Java Proyecto. Android Studio.



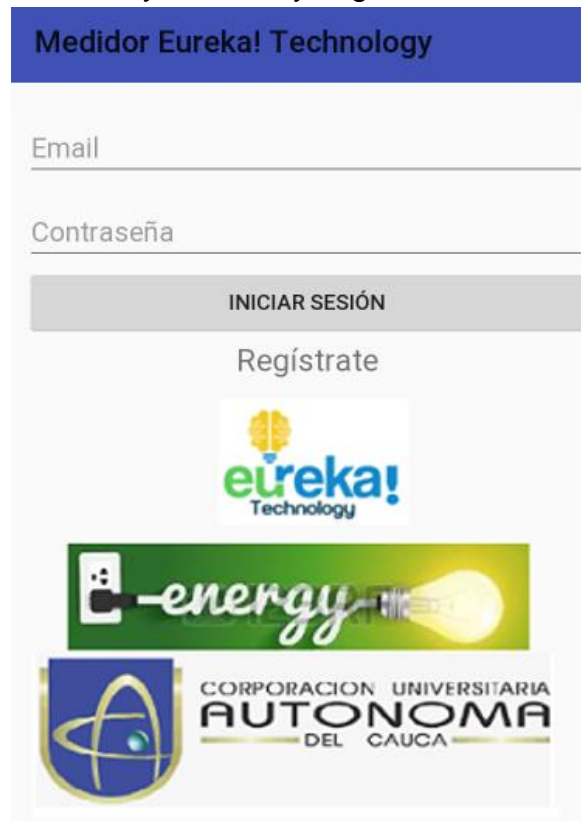
Fuente: [Elaboración propia]

La actividad (activity) principal determinada en el AndroidManifest es la de Login, esta se ejecuta solamente al abrir la aplicación y permite tener control en el acceso de los usuarios, un activity es una clase de java que hereda sus métodos de una

clase más genérica y su ciclo de vida puede estar en 4 diferentes estados: activa, visible, pausada o destruida los cuales necesitan diferentes métodos para inicializarse.

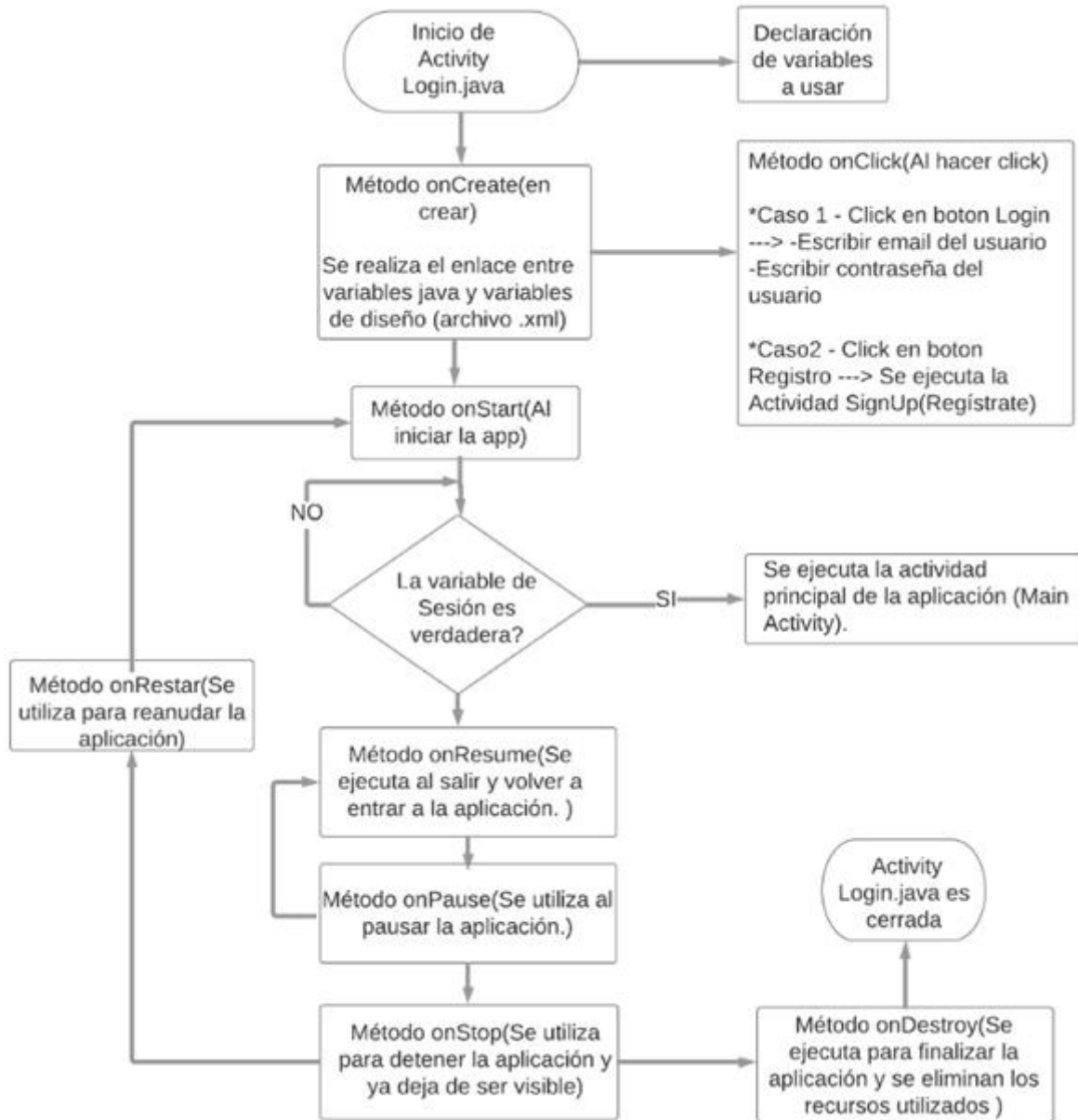
En esta aplicación solamente se usará el método onCreate() para definir la vista de las actividades con sus layouts (archivos XML que establecen el diseño de la interfaz de usuario de una Activity (Figura 4-25), en ellos se establecen las dimensiones, distribución y posiciones de los elementos que componen la interfaz, como se puede ver en el código XML del anexo 14), aquí se configuran las variables necesarias para el funcionamiento mediante la utilización de algunos elementos como lo son EditText que servirá para almacenar las variables de correo y contraseña del usuario, TextView para mostrar una línea de texto y Button que permitir generar la acción de iniciar la sesión correspondiente por medio del método onClick() el cual se llama cuando el usuario selecciona este elemento, el siguiente diagrama de flujo de la figura 4-26 demuestra el funcionamiento de la actividad Login.java.

Figura 4-25: Layout activity_login.xml. Android Studio.



Fuente: [Elaboración propia]

Figura 4-26: Diagrama de flujo Activity Login.java.



Fuente: [Elaboración propia]

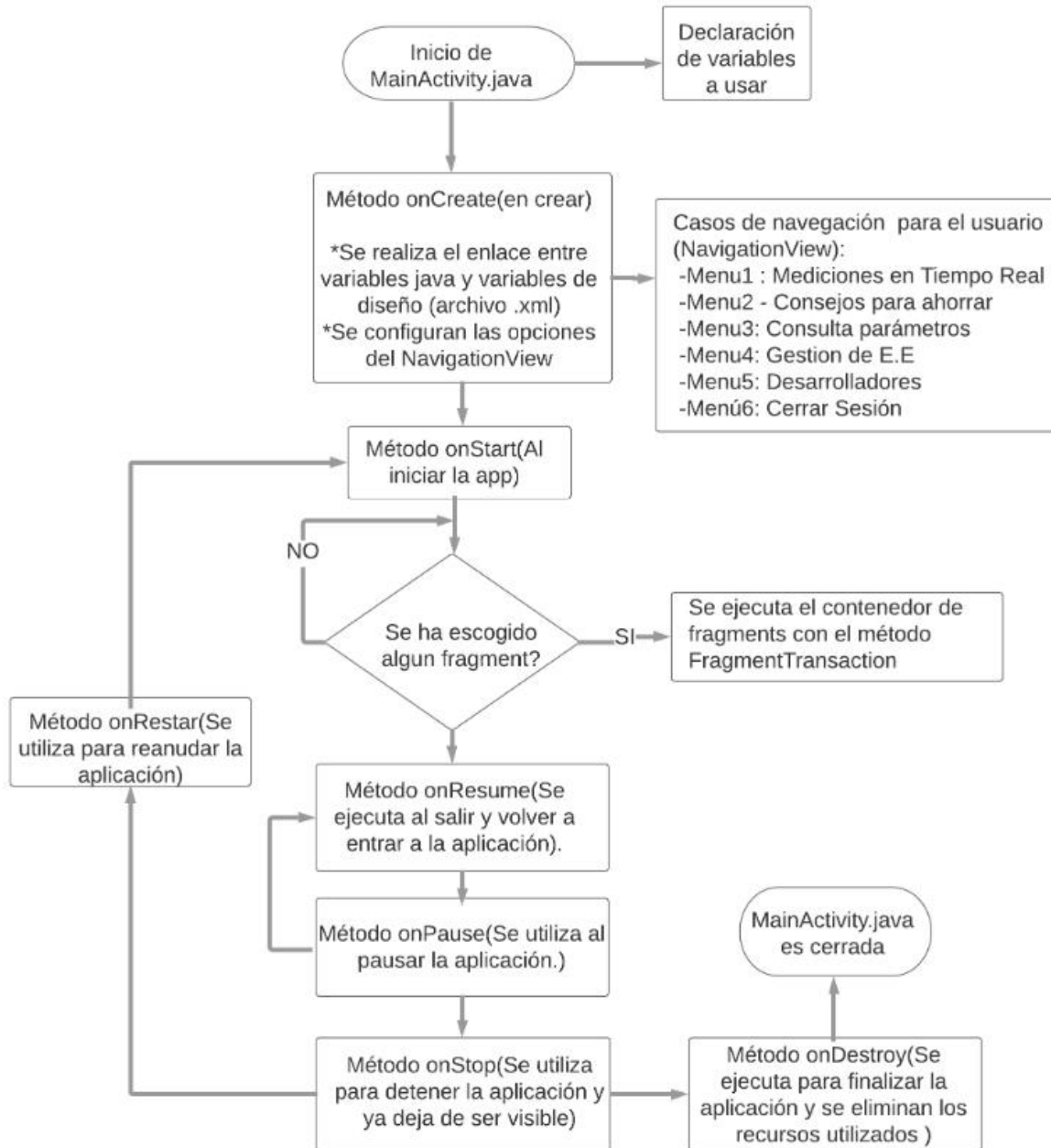
Al momento de ser llamado el método onClick, se instancia la función correspondiente y se verifican las credenciales de acceso para proceder a iniciar una nueva actividad (MainActivity) a través de un intent, que sirve para realizar invocaciones a otros componentes de android.

Cabe destacar que en este método se hace uso de la librería Volley para android studio, que permite utilizar cualquier API REST desde nuestra aplicación, haciendo llamadas HTTP y es aquí donde se comunica inicialmente la app con un servicio web propio implementado en IFastNet, para esta actividad se utiliza la petición HTTP mediante el método POST a la siguiente url: <http://espeureka.epizy.com/MysqlListView/Login/login.php>.

Después de que el usuario inicia sesión en la aplicación, es dirigido hacia la actividad MainActivity.java que se compone de un menú lateral deslizante o Navigation Drawer con fragments y contiene el menú principal de navegación de la app, para ello se hace uso de la clase Navigation View. Los fragments permiten versatilidad, optimización de diseño y recursos, reusar código y ahorrar tiempo a la hora de desarrollar una aplicación, también facilitan el despliegue de las aplicaciones en cualquier tipo de tamaño de pantalla y orientación.

En el método onCreate () de la actividad MainActivity.java se configura el layout que se utilizara para el menú de navegación y se instancia la clase Navigation View, además de esto, en la función displaySelectedScreen se establecen los diferentes casos de navegación para el usuario con el respectivo llamado a los diferentes fragments para cada opción, en la figura 4-27 se puede ver el diagrama de flujo que explica el funcionamiento de esta actividad y en la figura 4-28 se encuentra el diseño XML asociado a esta actividad.

Figura 4-27: Diagrama de flujo MainActivity.java. Android Studio.



Fuente: [Elaboración propia]

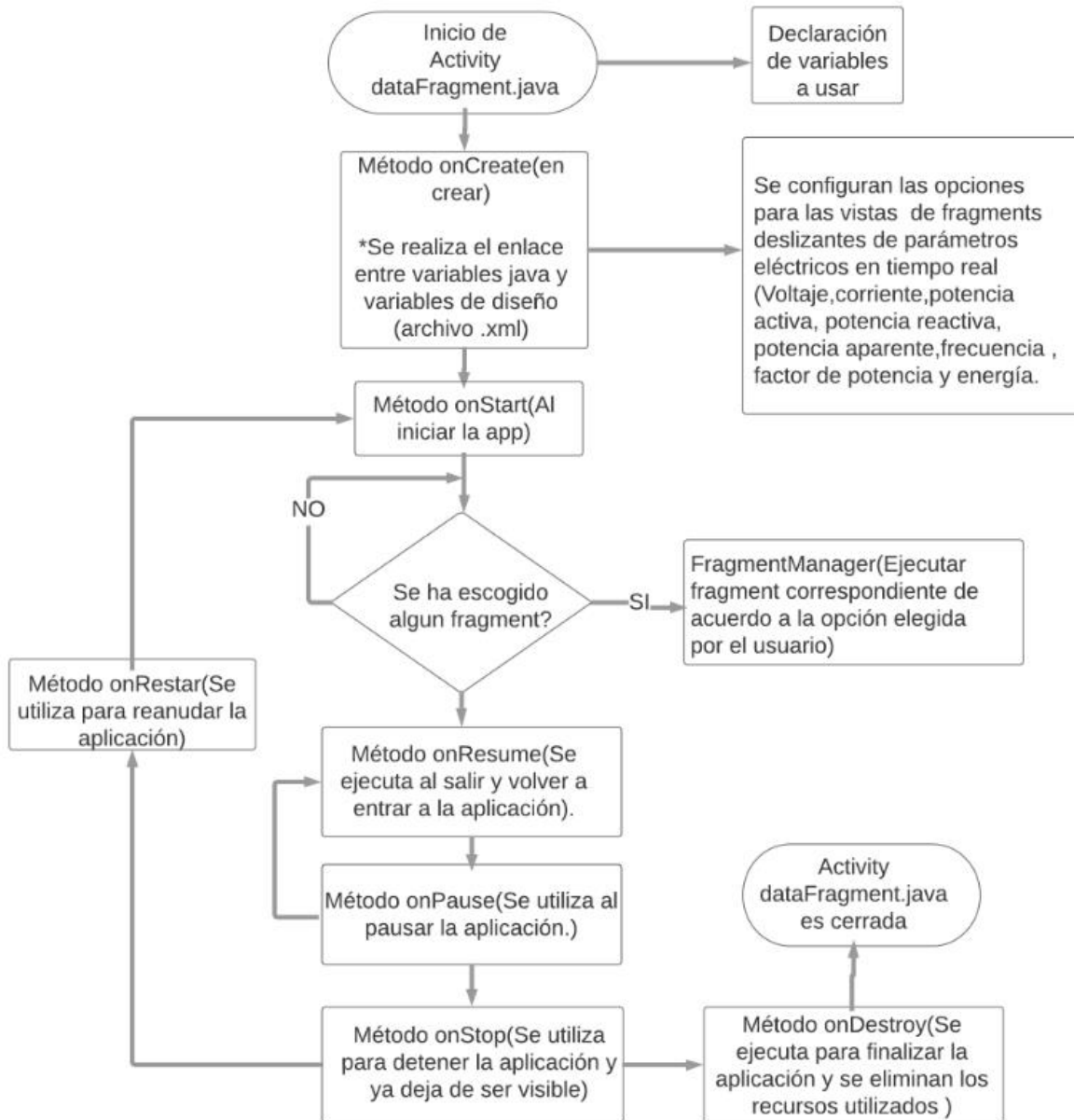
Figura 4-28: Layout Main Activity (Navigation drawer). Android Studio



Fuente: [Elaboración propia]

En cuanto a las opciones de navegación inicialmente se encuentra la de mediciones dentro de la cual se utiliza un fragment y a su vez el componente ViewPager, que permite mostrar distintas pantallas desplazando la página actual con el dedo a la izquierda o derecha para de esta manera poder presentar al usuario los datos(se actualizan cada 3 segundos) obtenidos desde el hardware implementado, en la figura 4-29 se encuentra el diagrama de flujo de la actividad que contiene este View Pager.

Figura 4-29: Diagrama de flujo DataFragment.java. Android Studio.



Fuente: [Elaboración propia]

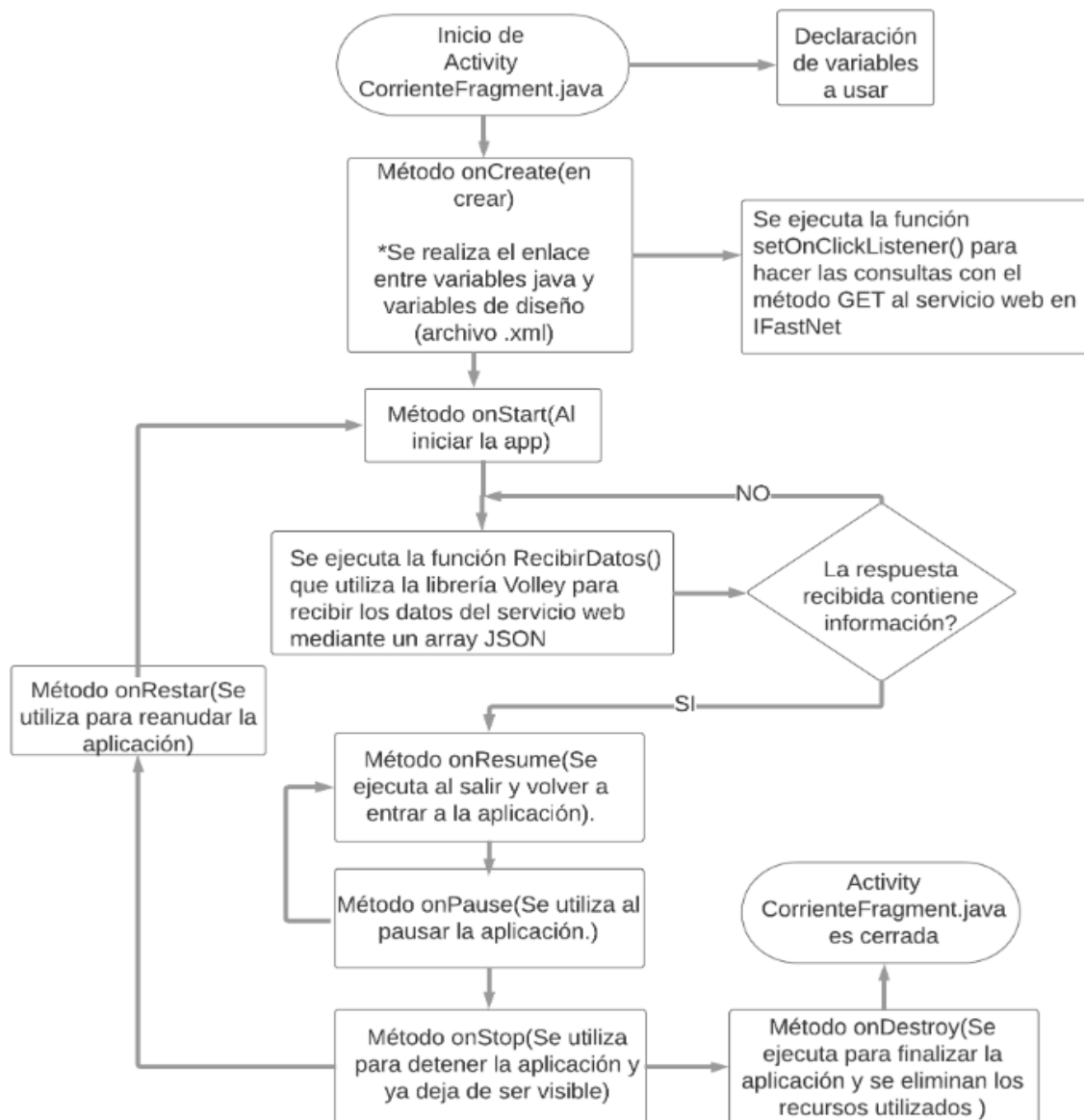
Nota: La opción de navegación “Consultar parámetros” utiliza la misma lógica de programación anterior con la diferencia de que en esta se le permite al usuario consultar en diferentes fechas y horas las mediciones obtenidas.

En los fragments que aparecen en la función sliderAdapter, por ejemplo CorrienteFragment(), también se hace uso de la librería Volley para android studio,

permitiendo la comunicación con el servicio web propio implementado en IFastNet , para esta actividad se utiliza la petición HTTP mediante el método GET a la siguiente url: <http://espeureka.epizy.com/MySQLListview/respuestaCorriente.php>.

En la figura 4-30 se encuentra el diagrama de flujo para este fragment que realiza las peticiones al servicio web para obtener y graficar los parámetros eléctricos en la aplicación móvil como se ve en la figura 4-31 en donde se está demostrando la corriente medida por el sistema.

Figura 4-30: Diagrama de flujo CorrienteFragment.java. Android Studio.



Fuente: [Elaboración propia]

Figura 4-31: Grafica de Corriente. Android Studio.

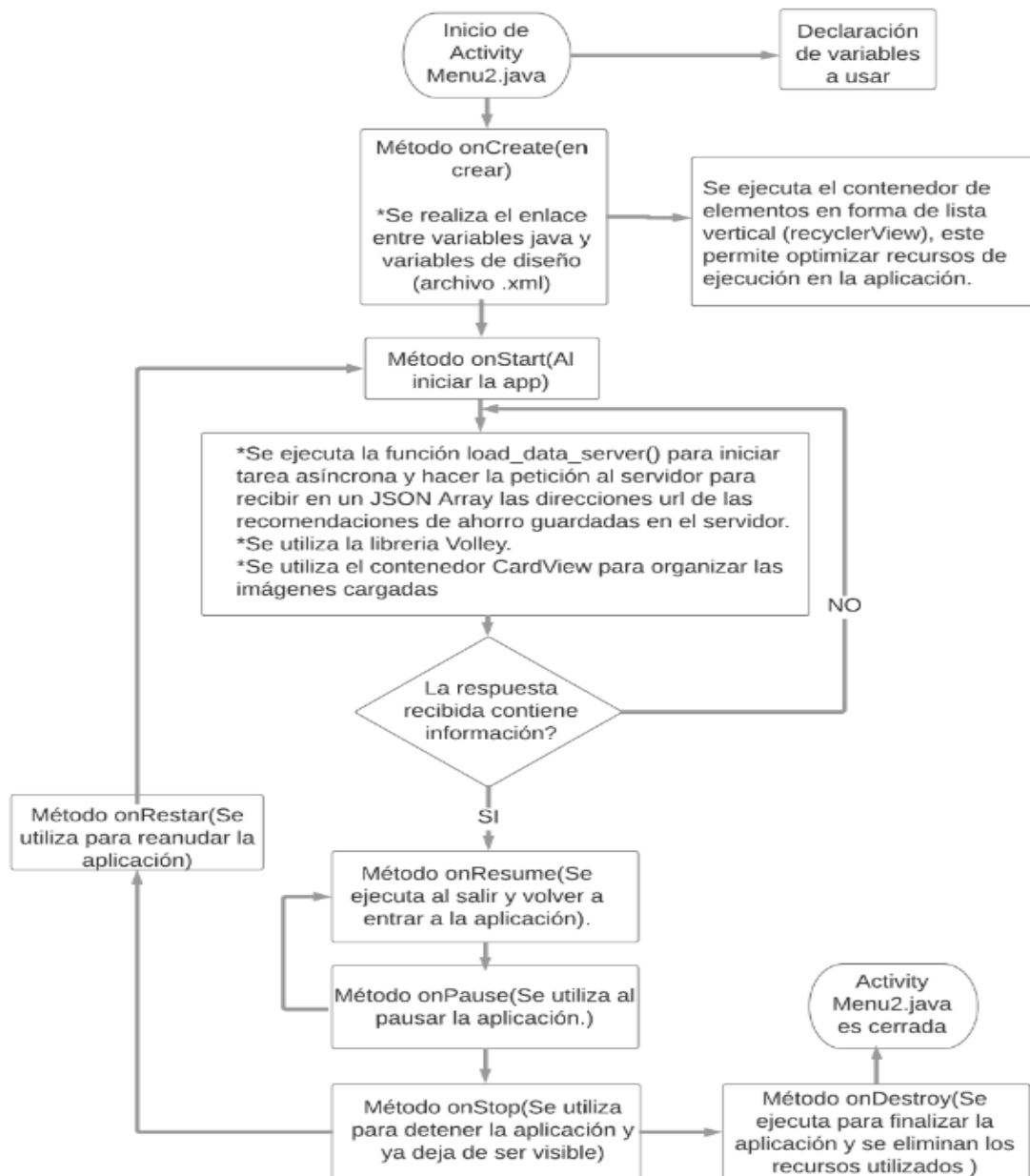


Fuente: [Elaboración propia]

En la actividad Consejos para ahorrar(Menu2.java) se utiliza un fragment y se hace la implementación en el método onCreateView () de un recyclerView, el cual proporcionan la posibilidad de listar elementos como un ListView pero además de esto “reciclar” los ítems que ya no son visibles por el usuario al scrolling, esto es ideal para proyectos en los cuales se manejan grandes volúmenes de ítems que se actualizan constantemente limitando la visibilidad de los elementos. El recyclerView también permite configurar animaciones para la creación, desplazamiento o eliminación de nuevos elementos de forma rápida.

En esta actividad Menu2.java se hace uso de la librería Volley para android studio, permitiendo la comunicación con el servicio web propio implementado en IFastNet, para esta actividad se utiliza la petición HTTP mediante el método GET a la siguiente url: <http://espeureka.epizy.com/MysqlListView/Login/CardView/script.php> para mostrar diferentes imágenes de consejos de ahorro de energía eléctrica distribuidos en una lista de CardView como se ve en el diagrama de flujo de la figura 4-32.

Figura 4-32: Diagrama de flujo Activity Menu2.java. Android Studio.



Fuente: [Elaboración propia]

En el anexo 15 se puede observar el código de diseño XML del fragment_Menu2 y en la figura 4-33 el resultado de su implementación en la aplicación móvil.

Figura 4-33: Layout Fragment_menu_2.xml (Menu2.java). Android Studio.



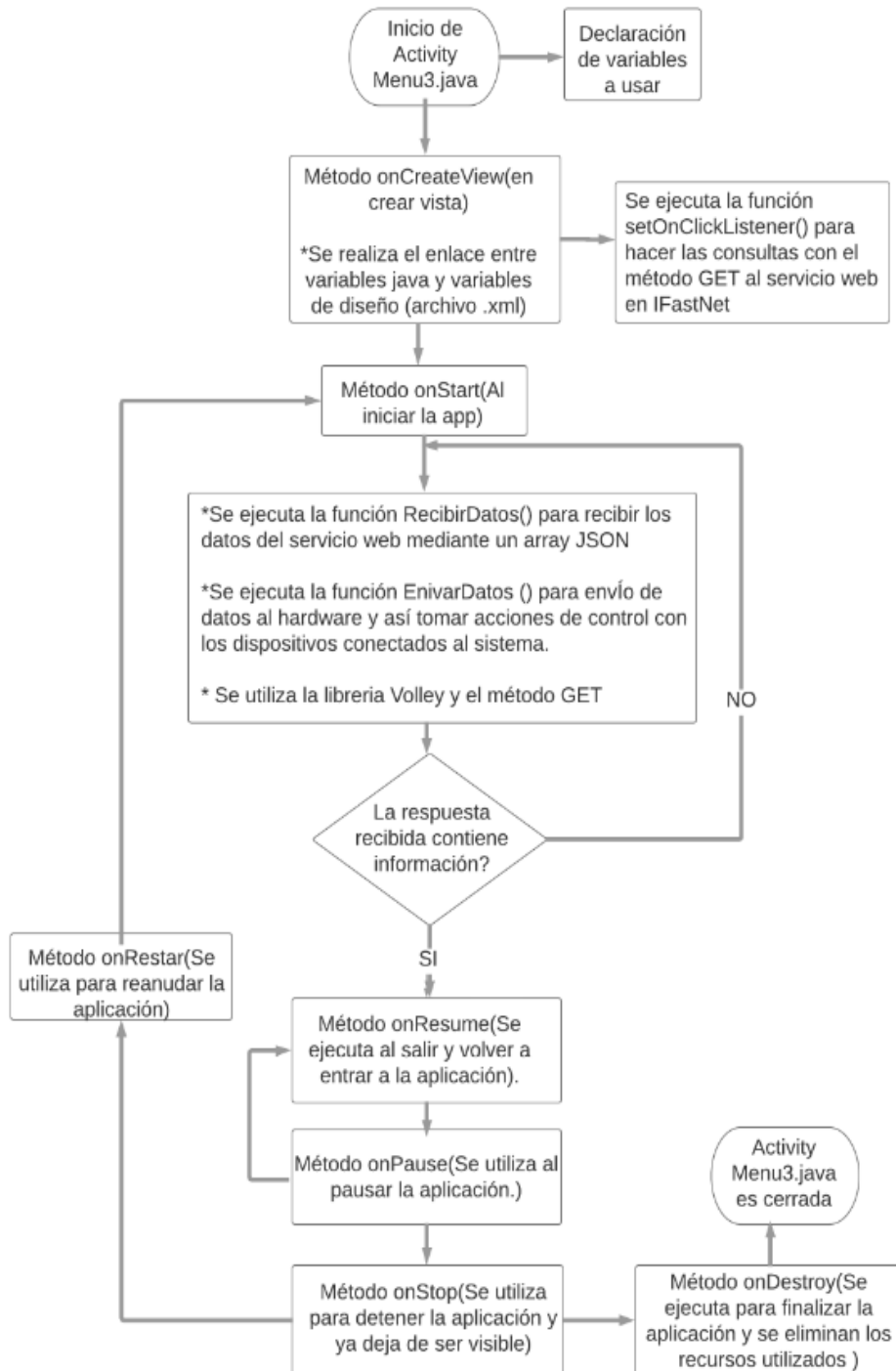
Fuente: [Elaboración propia]

La actividad Gestión de Energía Eléctrica (Menu3.java) utiliza un fragment y en ella se encuentran diferentes elementos como EditText que servirá para almacenar las variables de consulta de fechas y horas, TextView para mostrar los textos correspondientes y Buttons que permiten generar las acciones de selección de horas, fechas y además realizar la consulta correspondiente mediante una petición HTTP utilizando el método GET.

En esta actividad el usuario puede tener el control de las salidas relé del hardware desarrollado mediante el accionamiento de un botón en la interaz de usuario de la aplicación móvil para de esta manera encender o apagar algunas cargas y por ende reducir su consumo eléctrico en determinados momentos.

En la figura 4-34 se encuentra el diagrama de flujo para el fragment menu3.java y en la figura 4-35 está el diseño de la pantalla implementada para este menú.

Figura 4-34: Diagrama de flujo fragment Menu3.java. Android Studio.



Fuente: [Elaboración propia]

Figura 4-35: Pantalla Gestión de E.E (Menu3.java). Android Studio.



Fuente: [Elaboración propia]

4.1.4. Plan Energético.

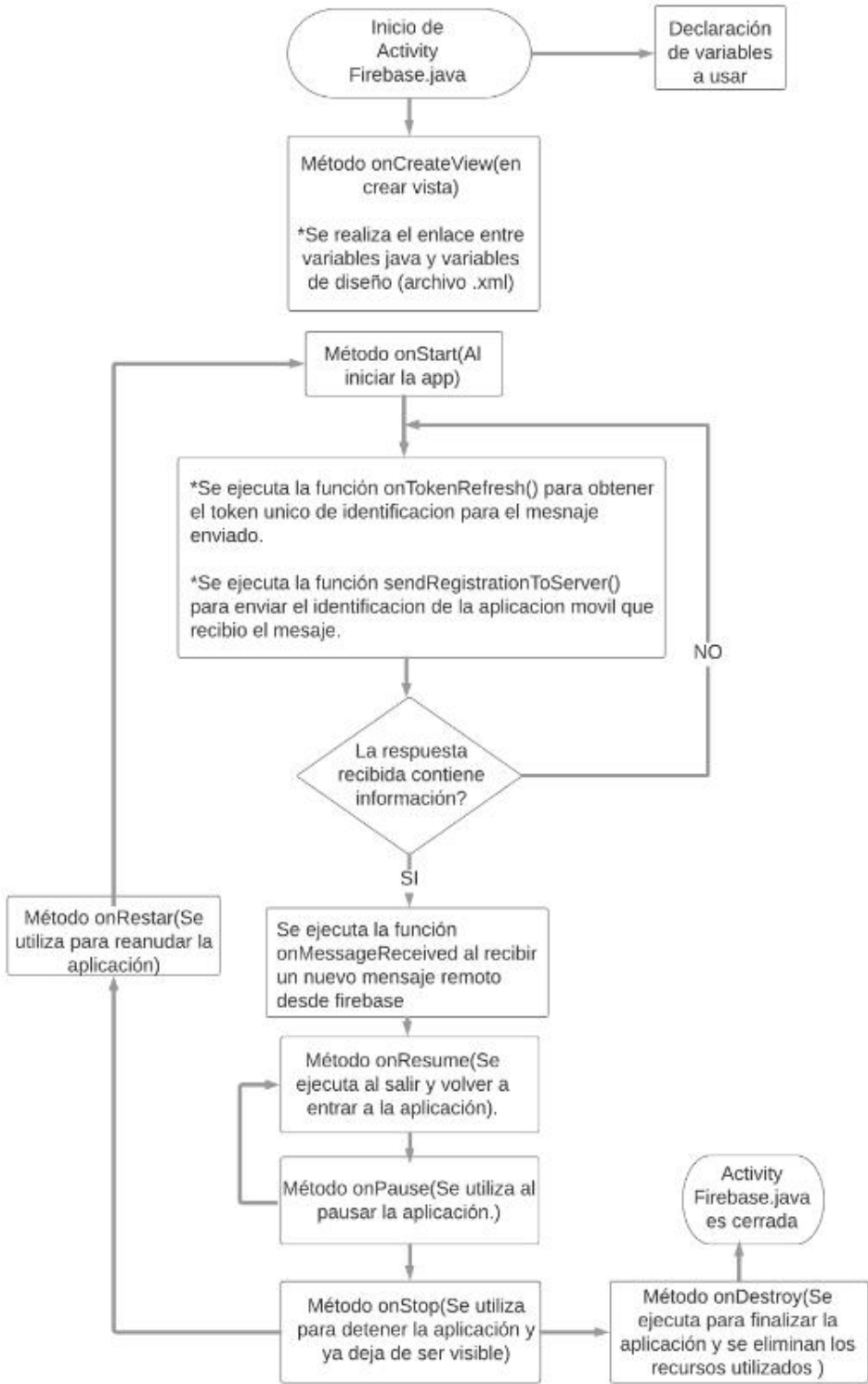
El desarrollo de este componente está determinado en la implementación de notificaciones push de noticias, nuevos consejos y tips de ahorro de energía (esto se realiza mediante la plataforma de Firebase), además de la activación de alarmas de consumo cuando se superan diferentes valores de umbral, que son enviadas mediante correo electrónico usando las recetas de automatización configuradas en IFTTT, para lo cual se usa servicios de Gmail; también son publicadas en una página de Facebook utilizando estas mismas recetas.

Finalmente, también se realiza la implementación en el hardware (Figura 4-56) de una medición alternativa de potencia y acumulado de kWh mediante la lectura de salida de pulsos (pulsos/kWh) de los medidores actuales instalados por la compañía prestadora del servicio de energía eléctrica, esto con el fin de comprobar las lecturas mensuales que arrojan las facturas de energía eléctrica.

4.1.4.1 Firebase.

Firestore es un conjunto de herramientas orientadas a la creación de aplicaciones de alta calidad y al crecimiento de los usuarios, entre sus servicios está el de Notificaciones (Firestore Cloud Messaging) el cual permite gestionar el envío de notificaciones a los usuarios de la aplicación móvil, con la diferencia de que estas pueden ser programadas acorde a diferentes parámetros. El siguiente diagrama de flujo de la figura 4-36 muestra la configuración para las notificaciones push mediante firestore en la aplicación móvil.

Figura 4-36: Diagrama de flujo configuración Push Notifications Firebase. Android Studio.

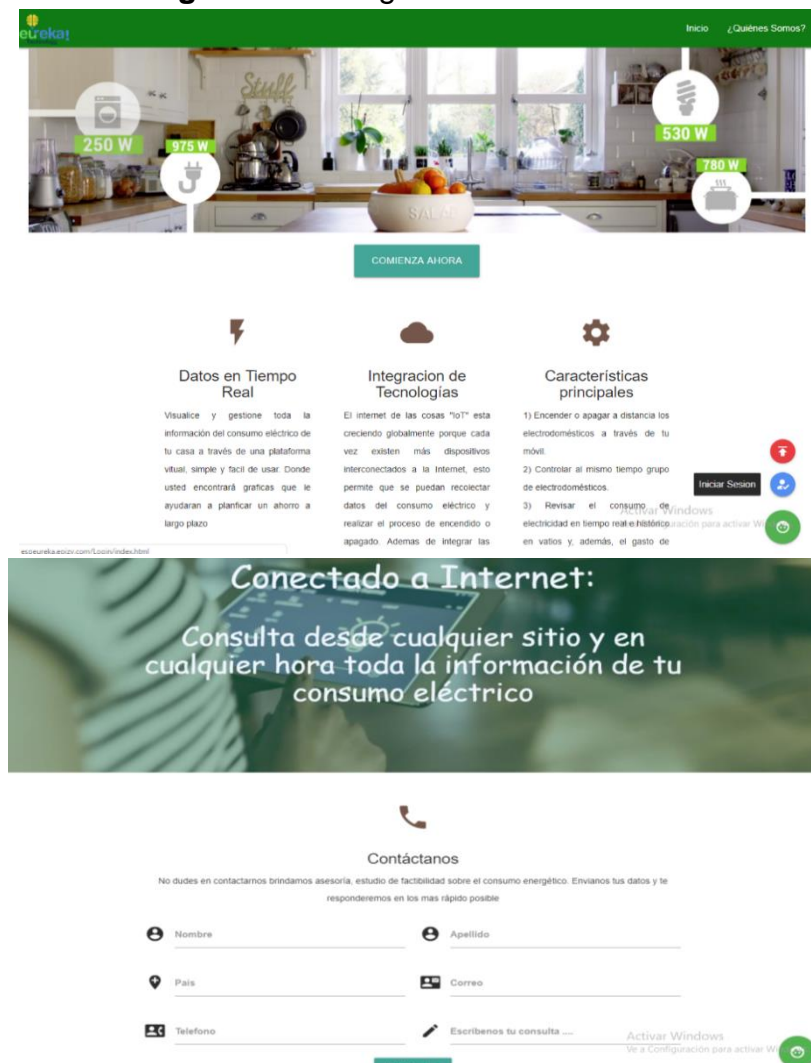


Fuente: [Elaboración propia]

4.1.5. Página y Servicio Web.

La página (figura 4-37) y servicio web que se desarrolló se denominó SIMEE (Sistema Inteligente de Medición de Energía Eléctrica), la cual es resultado de la implementación de diferentes tecnologías y software de desarrollo web como lo son JavaScript, Html, Css, Php, Visual Studio Code, Github, tanto para el Frontend como el Backend y se puede ingresar desde la siguiente URL: <http://espeureka.epizy.com/>. En la figura 4-38 se muestra la página de inicio de sesión y en la figura 4-39 está la página de Consultas por fecha y hora.

Figura 4-37: Página de inicio SIMEE.



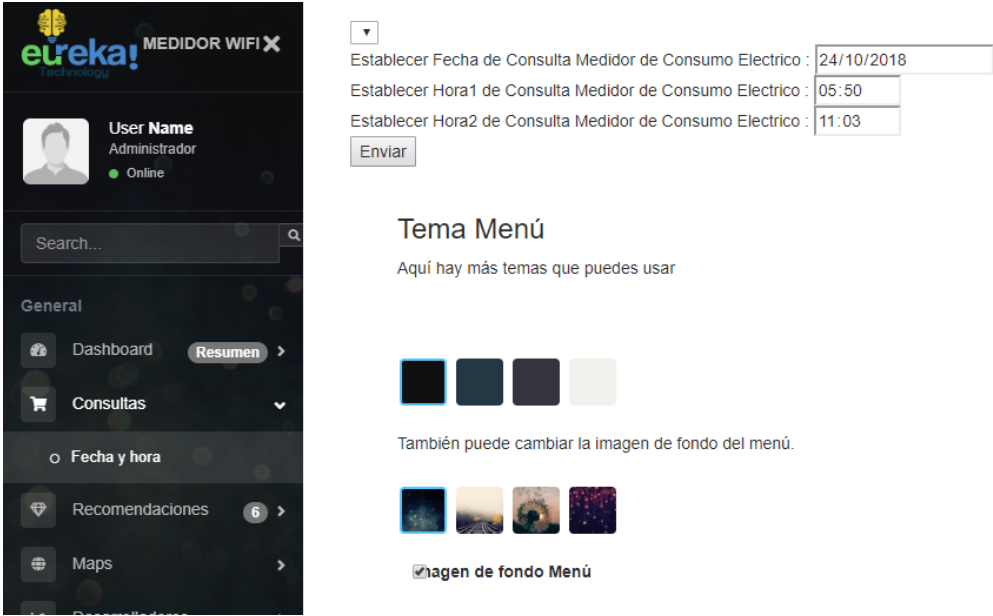
Fuente: [Elaboración propia]

Figura 4-38: Página de login SIMEE.



Fuente: [Elaboración propia]

Figura 4-39: Página Consultas por Fecha y hora SIMEE.



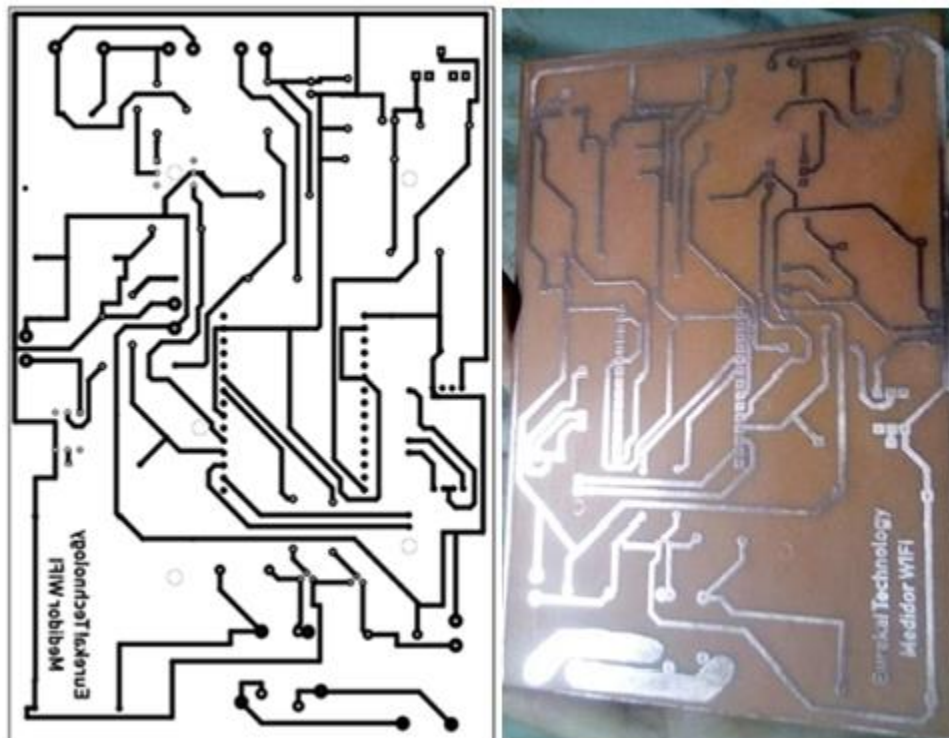
Fuente: [Elaboración propia]

4.2. DESARROLLO HARDWARE

Los sistemas diseñados basan su funcionamiento en elementos electrónicos de bajo costo y con base en ellos se realizó el Diseño PCB (placa de circuito impreso) de los dos sistemas en el software Proteus. Posteriormente, se procede a la elaboración de las placas circuitales mediante el uso de fabricación de circuitos impresos conocido como el método de la plancha, en donde se imprime el diseño en papel fotográfico, después se calienta para transferir la tinta a una váquela virgen de cobre y se realiza el atacado de cobre con Cloruro Férrico, el cual se disuelve en agua caliente junto con la placa para finalmente pasar a brillarla y finalmente realizar las perforaciones correspondientes usando un motortool.

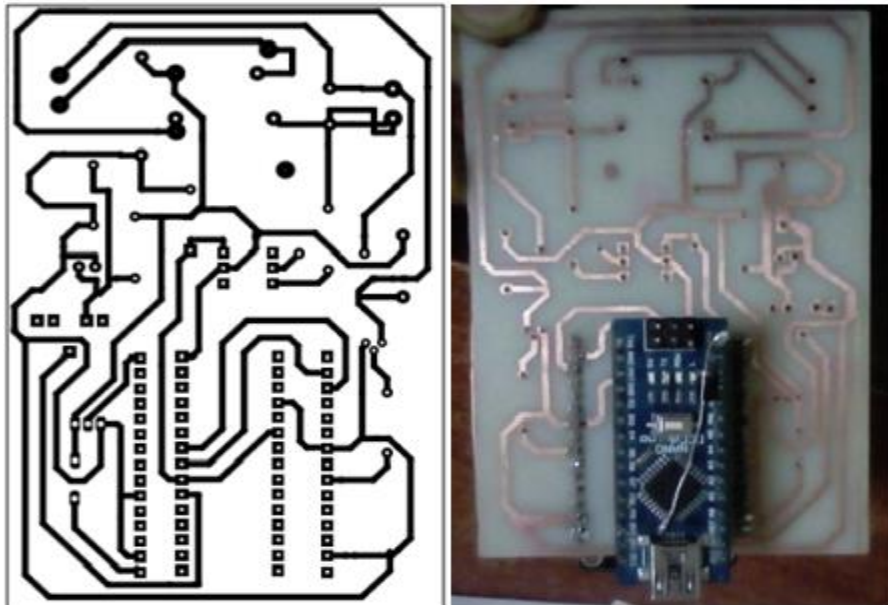
El diseño PCB del sistema principal y del portátil se presentan en la figura 4-40 y 4-41 respectivamente.

Figura 4-40: Diseño PCB Sistema principal.



Fuente: [Elaboración propia]

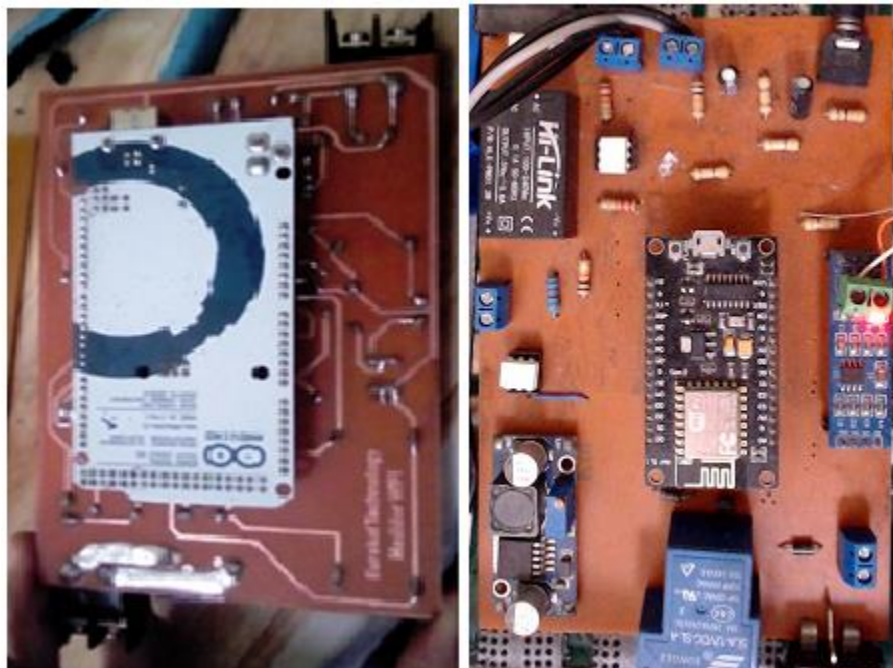
Figura 4-41: Diseño PCB Sistema portátil



Fuente: [Elaboración propia]

Después de elaboradas las PCB se procede al ensamble de sus componentes electrónicos, como se observa en la figura 4-42 y 4-43.

Figura 4-42: Parte inferior y superior Sistema principal.



Fuente: [Elaboración propia]

Figura 4-43: Parte superior e inferior Sistema portátil



Fuente: [Elaboración propia]

Cabe resaltar que las estructuras para los dos sistemas fueron realizadas utilizando la tecnología de impresión 3D. En la figura 4-44 y 4-45 se observa la estructura de cada sistema.

Figura 4-44: Estructura Final del Sistema principal.



Fuente: [Elaboración propia]

Figura 4-45: Estructura Sistema portatil.



Fuente: [Elaboración propia]

CAPITULO V: RESULTADOS

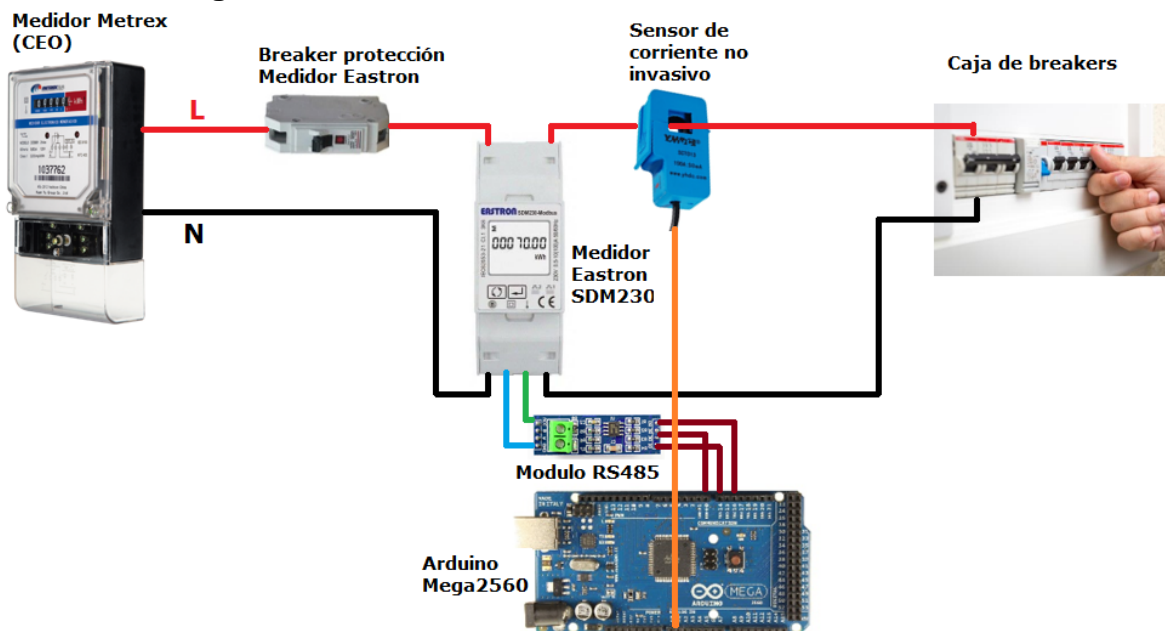
A continuación, se presentan los resultados obtenidos mediante la implementación hardware y software del sistema de gestión de energía eléctrica para los usuarios del sector residencial.

Los sistemas (principal y el portátil) se instalaron en una vivienda familiar del barrio colina campestre de la ciudad de Popayán, la casa es de dos pisos con instalación eléctrica monofásica, cuenta con 3 habitaciones, 2 baños, sala, comedor, cocina, patio y garaje.

La casa cuenta con alrededor de 10 elementos electrónicos:

- 2 televisores, 2 computadores de escritorio, 1 equipo de sonido, 1 nevera, 1 impresora 3d, 1 lavadora, 1 router, 1 mini componente.
- 10 bombillos (Lámparas incandescentes y tubos led), unos constantemente funcionando y otros esporádicamente conectados, pero de igual manera se les hizo el reconocimiento de consumo energético mediante el medidor Eastron y el sensor de corriente no invasivo utilizando la conexión de las figuras 5-1,5-2 y haciendo el siguiente procedimiento de experimentación:

Figura 5-1: Conexión del sensor de corriente no invasivo.



Fuente: [Elaboración propia]

Figura 5-2: Implementación de la conexión del sensor de corriente no invasivo.



Fuente: [Elaboración propia]

Procedimiento 1: Apagar todos los dispositivos electrónicos y testear cuánto consume cada uno elemento con ambos sistemas. En las tablas 5-1, 5-2 se presentan las mediciones que se obtuvieron.

Tabla 5-1: Mediciones del Sistema principal.

	Medidor Sdm230 (A)	Sensor no Invasivo (A)	Potencia Activa(W)
Modem	0,12	0,10	12
Bombillo incandescente	0,44	0,41	52
Bombillo ahorrador	0,30	0,29	35
2 bombillos	0,66	0,65	81
Computador	0,65	0,63	79
Plancha	8,51	8,29	1040

Secador	2,76	2,74	342
Fotocopiadora	10,58	10,41	1299
Nevera	1,80	1,76	215
Smart Tv 55"	1,50	1,47	181

Fuente: [Elaboración propia]

Tabla 5-2: Mediciones del Sistema portátil.

	Sensor no Invasivo (A)	Potencia Activa(W)
Bombillo incandescente	0,43	50
Bombillo Ahorrador	0,30	35
2 bombillos	0,65	81
Computador	0,63	78
Plancha	8,13	1025
Secador	2,73	334
Fotocopiadora	10,63	1303
Nevera	1,78	211
Smart Tv 55"	1,47	179

Fuente: [Elaboración propia]

Los resultados de ambos sistemas tienen correlación y se obtiene un porcentaje de similitud del 95,7% con un error del 4,3%, resultados obtenidos tomando como referencia los datos del medidor comercial EastronSDM230, comparándolos con las mediciones del sensor no invasivo y realizando el promedio de las 10 opciones de la tabla 5-1.

Adicionalmente se realizó la comparación entre las mediciones del sistema principal VS el sistema portátil haciendo uso de los datos de las tablas 5-1 y 5-2 en donde se puede concluir que se tiene un porcentaje de similitud del 98% y un porcentaje de error del 2% entre los dos sistemas.

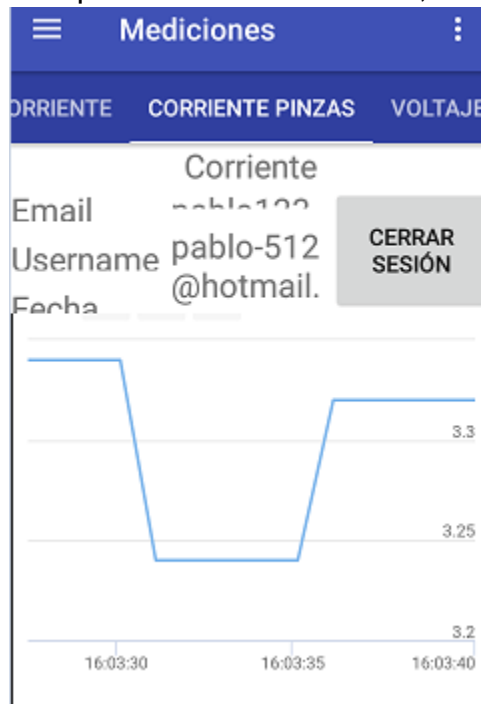
Procedimiento 2: Tomar mediciones de diferentes cargas de forma simultánea. En las figuras 5-3, 5-4, 5-5 se muestra la medición del secador y el computador.

Figura 5-3: Medición- secador, computador.



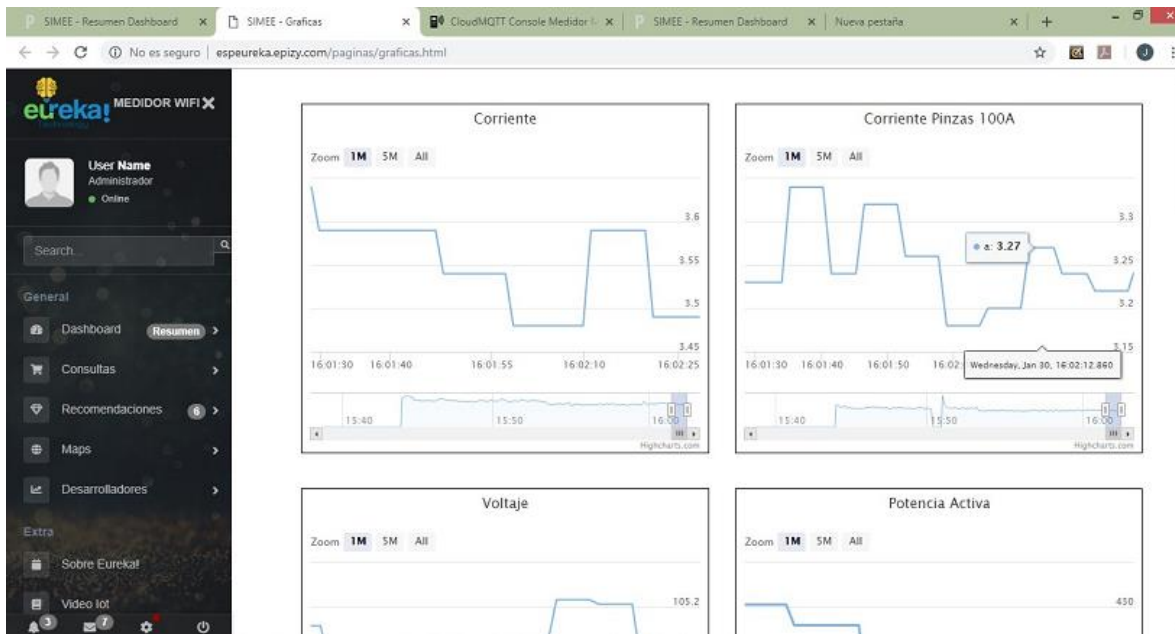
Fuente: [Elaboración propia]

Figura 5-4: Aplicación móvil- secador, computador.



Fuente: [Elaboración propia]

Figura 5-5: Página web- secador, computador.



Fuente: [Elaboración propia]

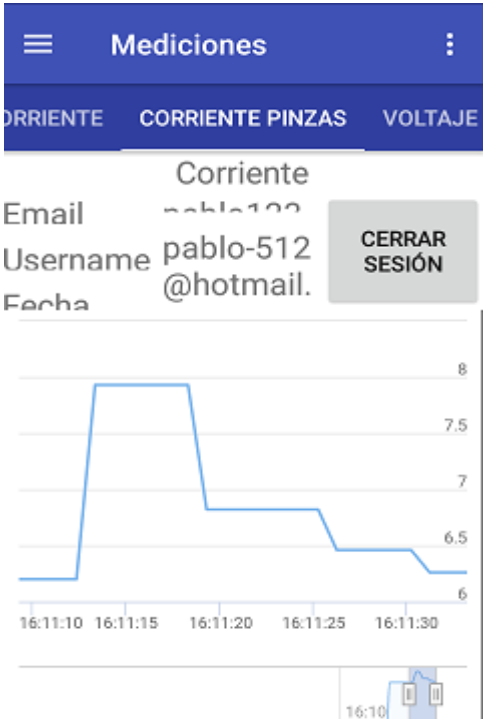
En las figuras 5-6,5-7,5-8 se enseña la medición de la nevera, secador, bombillo ahorrador y smart Tv de forma simultánea.

Figura 5-6: Medición- nevera, secador, bombillo ahorrador, smart Tv.



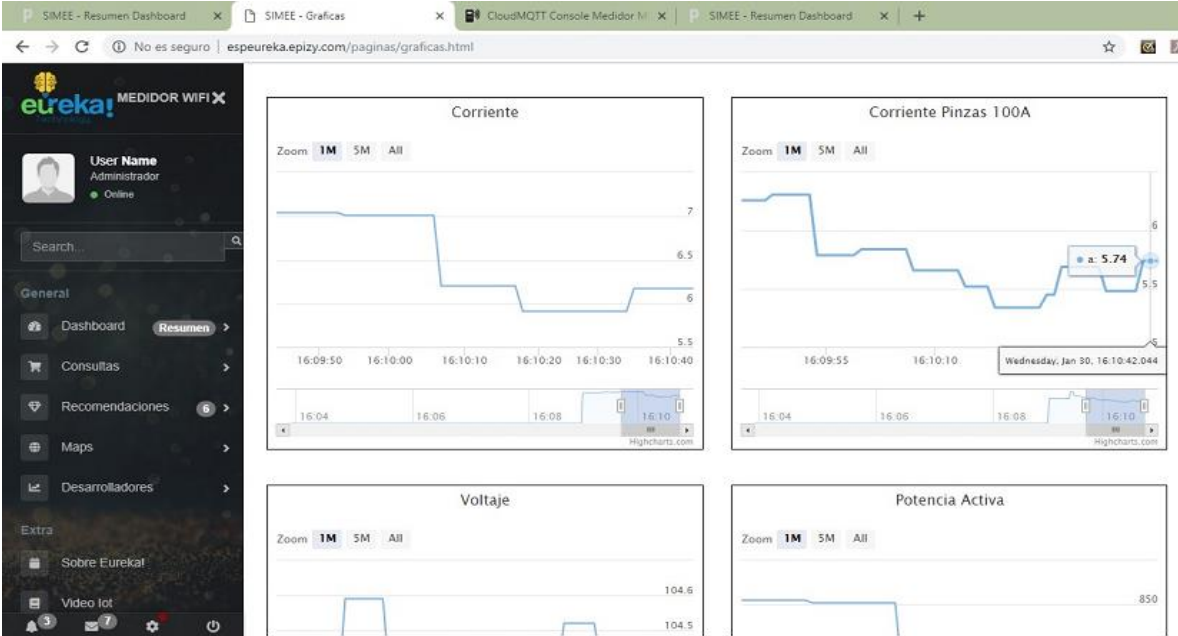
Fuente: [Elaboración propia]

Figura 5-7: Aplicación móvil- nevera, secador, bombillo ahorrador, smart Tv.



Fuente: [Elaboración propia]

Figura 5-8: Página web- nevera, secador, bombillo ahorrador, smart Tv.



Fuente: [Elaboración propia]

En las figuras 5-7,5-8,5-9 se presenta la medición del modem, bombillo incandescente y secador de forma simultánea.

Figura 5-9: Medición- modem, bombillo incandescente, secador.



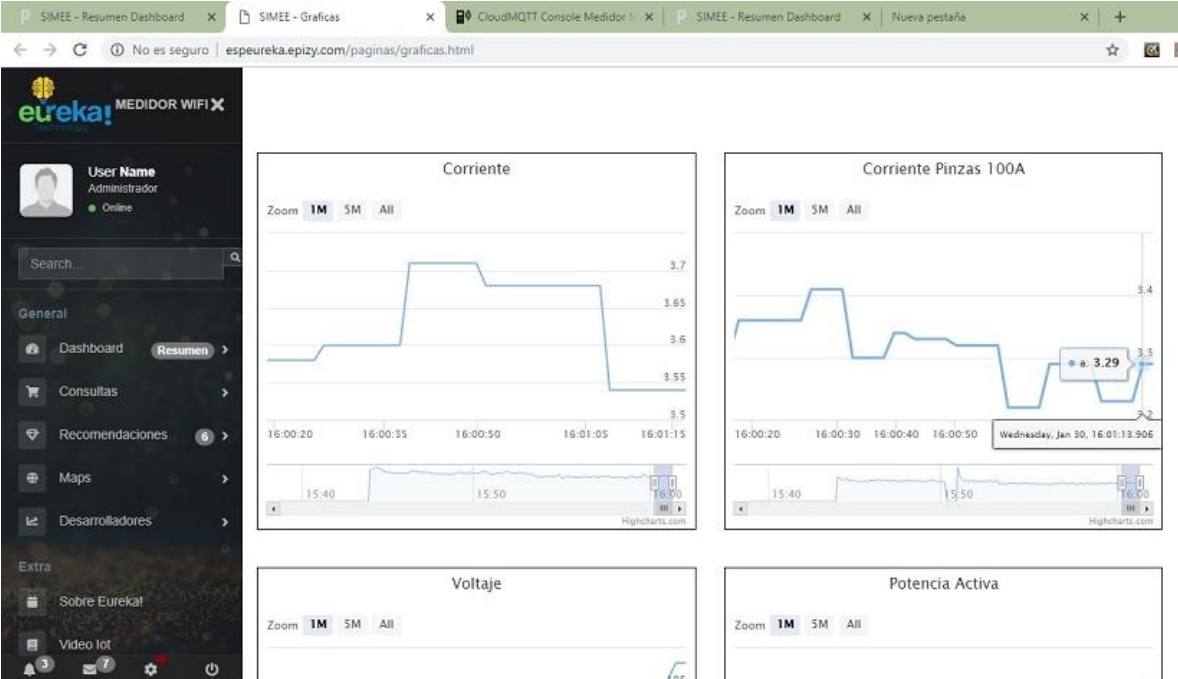
Fuente: [Elaboración propia]

Figura 5-10: Aplicación móvil- modem, bombillo incandescente, secador



Fuente: [Elaboración propia]

Figura 5-11: Página web- modem, bombillo incandescente, secador



Fuente: [Elaboración propia]

Las Figuras 5-12, 5-13 demuestran las mediciones obtenidas por la salida de pulsos del contador de la compañía energética de occidente al conectar como carga un secador, arrojando un resultado similar con una diferencia del 1% al que se obtiene por parte del medidor Eastron SDM230 como se evidencia en la tabla 5-1.

Figura 5-12: Medición de pulsos-secador.



Fuente: [Elaboración propia]

Figura 5-13: Página web-Medición de pulsos-secador.



Fuente: [Elaboración propia]

Procedimiento 3: Se realiza la comparación entre las mediciones obtenidas mes a mes por la factura eléctrica de la empresa prestadora del servicio, la compañía energética de occidente y el sistema principal de la siguiente manera:

- 1) Se compara la facturación del servicio de energía eléctrica del periodo de consumo comprendido entre las fechas 03 de febrero del año 2019 y 04 de marzo del año 2019 y los datos del sistema principal implementado, dando como resultado una diferencia aceptable de 11Kwh de más en la factura eléctrica como se puede verificar en las figuras 5-14 y 5-15.

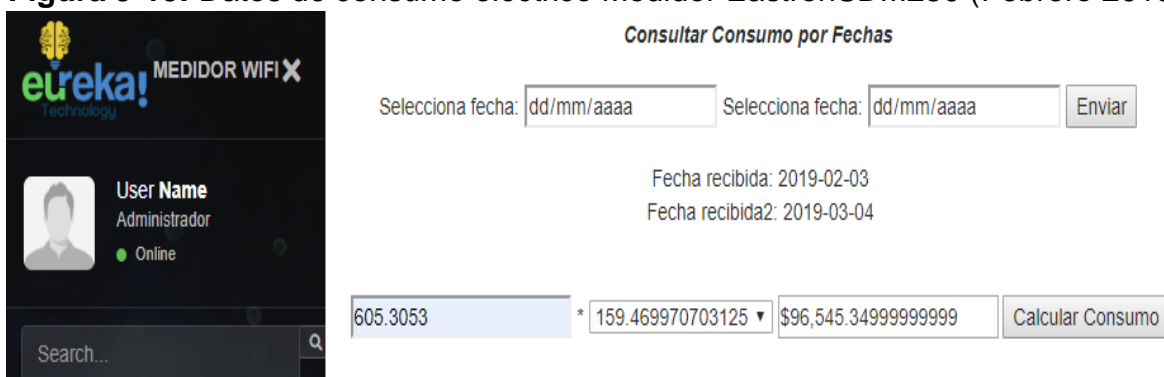
De acuerdo a la factura del mes de Febrero del 2019 el consumo eléctrico es de 159 Kwh por un valor de \$96,700, por lo contrario de acuerdo a los datos del medidor Eastron SDM230 el consumo eléctrico es de 159.47 Kwh por un valor de \$96,545, lo cual da la diferencia de 0.47 Kwh por un valor de \$284,5.

Figura 5-14: Facturación del servicio de energía eléctrica (Febrero 2019)



Fuente: [Elaboración propia]

Figura 5-15: Datos de consumo eléctrico Medidor EastronSDM230 (Febrero 2019)



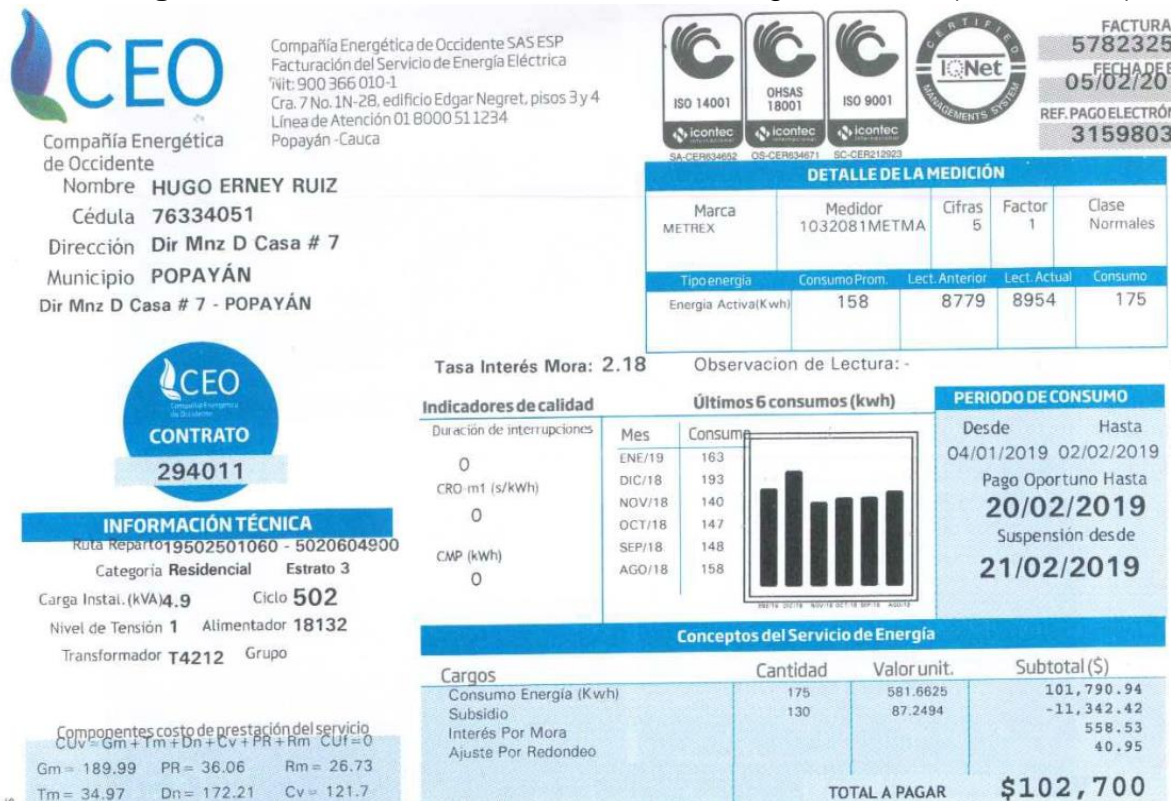
Fuente: [Elaboración propia]

- 2) Se compara la facturación del servicio de energía eléctrica del periodo de consumo comprendido entre las fechas 04 de enero del año 2019 y 02 de febrero del año 2019 y los datos del sistema principal implementado, dando como

resultado una diferencia aceptable de 11Kwh de más en la factura eléctrica como se puede verificar en las figuras 5-16 y 5-17.

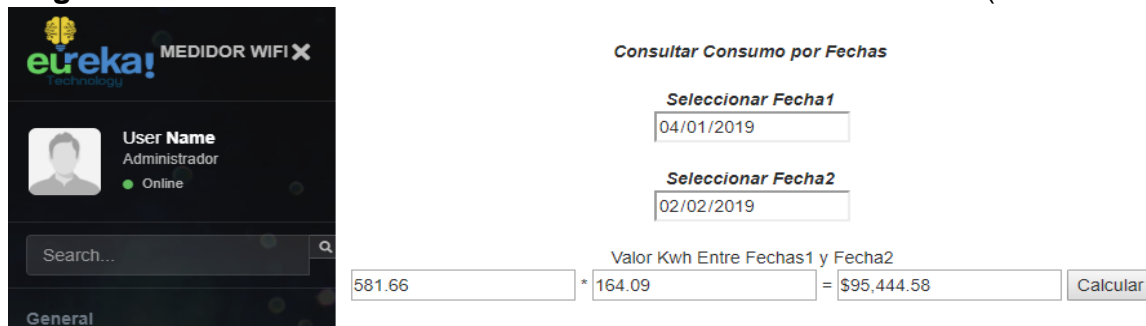
De acuerdo a la factura del mes de Enero del 2019 el consumo eléctrico es de 175 Kwh por un valor de \$102,700, por lo contrario de acuerdo a los datos del medidor Eastron SDM230 el consumo eléctrico es de 164.09 Kwh por un valor de \$95,444.58, lo cual da la diferencia de 11Kwh por un valor de \$6,398.

Figura 5-16: Facturación del servicio de energía eléctrica (Enero 2019)



Fuente: [Elaboración propia]

Figura 5-17: Datos de consumo eléctrico Medidor EastronSDM230 (Enero 2019)



Fuente: [Elaboración propia]

- 3) Se compara la facturación del servicio de energía eléctrica del periodo de consumo comprendido entre las fechas 04 de diciembre del año 2018 y 03 de enero del año 2019 y los datos del sistema principal implementado, dando como resultado una diferencia aceptable de 7Kwh de más en la factura eléctrica como se puede verificar en las figuras 5-18 y 5-19.

De acuerdo a la factura del mes de Diciembre del 2018 el consumo eléctrico es de 163 Kwh por un valor de \$97,600, por lo contrario de acuerdo a los datos del medidor Eastron SDM230 el consumo eléctrico es de 156.90 Kwh por un valor de \$93,377, lo cual da la diferencia de 7Kwh por un valor de \$4,165.

Figura 5-18: Facturación del servicio de energía eléctrica (Diciembre 2018)



Fuente: [Elaboración propia]

Figura 5-19: Datos de consumo eléctrico Medidor EastronSDM230 (Diciembre 2018)

The screenshot shows the Eureka! Medidor WiFi interface. On the left, there is a sidebar with the Eureka! logo, a user profile for 'Administrador' (Online), a search bar, and navigation buttons for 'Dashboard' and 'Resumen'. The main area is titled 'Consultar Consumo por Fechas' and contains two date selection fields: 'Seleccionar Fecha1' with the value '04/12/2018' and 'Seleccionar Fecha2' with the value '03/01/2019'. Below these is a calculation area with the text 'Valor Kwh Entre Fechas1 y Fecha2'. The calculation is displayed as: $595.1406 * 156.90 = \$93,377.56$. A 'Calcular' button is located to the right of the result.

Fuente: [Elaboración propia]

CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Se implemento un sistema electrónico para el monitoreo remoto de los parámetros eléctricos presentes en un hogar residencial, esto mediante una aplicación móvil que permite al usuario además de tener esta supervisión realizar un control ON/OFF de las cargas conectadas lo cual permite reducir considerablemente el consumo de standby-by entre el 5% y 10% en los hogares.
- Los protocolos de comunicación utilizados para el presente proyecto demuestran la capacidad de comunicación bidireccional de los dispositivos con las plataformas de visualización y control desarrolladas, esto permite tener una buena flexibilidad para el manejo de estos datos y la información presentada al usuario.
- La utilización tanto de la plataforma web como aplicación móvil han permitido corroborar los datos de consumo de energía eléctrica de diferentes dispositivos electrónicos, por ejemplo, plancha de ropa, bombillos, televisores, nevera, lavadora, impresora 3d y fotocopiadora, como se muestra en los resultados del presente proyecto, para de esta manera poder tomar acciones de ahorro energético en determinados momentos del día.
- Se desarrolló e implemento la base de una plataforma IOT propia para la recepción y visualización de los datos obtenidos del hardware, lo cual permite integrar muchos más dispositivos y escalar el proyecto a diferentes soluciones inmersas en las tendencias tecnológicas del internet de las cosas, esto puede ser evidenciado ingresando a la siguiente URL: <http://espeureka.epizy.com/Dasboard.html>.
- La aplicación móvil desarrollada permite al usuario del sector residencial realizar las siguientes tareas: ver las mediciones de los parámetros eléctricos de las cargas a través de gráficas que permiten interpretar mejor determinada información, haciéndola más entendible y proporcionando una idea generalizada de los resultados; realizar consultas de los datos en diferentes fechas y horas del día; enterarse de cómo ahorrar energía eléctrica por medio de consejos e imágenes explicativas; recibir notificaciones push (mensajes instantáneos) de noticias o nuevos consejos y tips de ahorro energético;

visualizar el estado y mediciones de los sistemas de gestión portátiles que tenga asociados; consultar el gasto monetario en pesos (\$) en relación al consumo en kWh y proporcionar el control del encendido o apagado de las cargas que estén conectadas a las salidas relé de los sistemas desarrollados.

- La verificación de los parámetros eléctricos ha permitido personalmente tomar mejores decisiones y configurar de una manera más precisa y real el costo por hora del servicio de impresión 3d ya que las plataformas desarrolladas permiten verificar el tiempo de trabajo y consumo eléctrico para de esta manera ofrecer precios competitivos y concretos sin incurrir en pérdidas monetarias.
- Con el desarrollo de este proyecto se aprendió a utilizar diferentes tecnologías de programación hardware y software para la implementación del Frontend y Backend del proyecto, esto permite ampliar nuestros conocimientos y capacidades personales y profesionales para la creación de nuevas soluciones tecnológicas web y móvil.
- La programación OTA abre la posibilidad de programar los dispositivos inalámbricamente, un punto a favor para el desarrollo de soluciones de IOT que tengan que ser instaladas en lugares difíciles de acceder para su programación manual.
- La plataforma IFTTT abre un mundo de posibilidades de interconexión y automatización mediante su configuración de recetas con diferentes servicios web, esto genera oportunidades para la ejecución de tareas y acciones predeterminadas o reacciones a eventos obtenidos por parte de los desarrollos hardware.

6.2. RECOMENDACIONES

- Es sumamente importante que se utilicen a futuro tecnologías para el análisis y tratamiento de los datos obtenidos, esto con ayuda del BigData y machine learning para de esta manera poder realizar mejores recomendaciones de ahorro energético y poder presentar pronósticos futuros de consumo con base en las mediciones actuales.
- Se recomienda utilizar para el backend de proyectos futuros una base de datos no relacional preferiblemente en MongoDB y montar un servidor propio

utilizando la tecnología de node.js para poder tener un mejor manejo y rapidez en las consultas realizadas a la base de datos, para mejorar la experiencia del usuario ya que en desarrollos de IOT se suele manejar una gran cantidad de datos que se actualizan constantemente.

- Se recomienda socializar los beneficios y funcionalidades de los sistemas implementados a la compañía prestadora del servicio de energía eléctrica y de igual manera a los usuarios residenciales buscando masificar la solución para de esta manera contribuir tanto en los procesos internos de la empresa en relación con la obtención de mediciones de los usuarios como el impacto ambiental y económico al reducir el consumo eléctrico mediante el control y monitoreo de cargas.
- Para trabajos enfocados a este tipo de investigación en el campo de la eficiencia energética, se debe tener en cuenta la legislación y normativa que los abarcan ya que el gobierno establece incentivos para su desarrollo al ser proyectos que aportan muchos beneficios para la sociedad.
- Para trabajos futuros es recomendable implementar más funcionalidades a la aplicación móvil como lo son la configuración de diferentes tarifas y control de cargas y dispositivos asociados.

BIBLIOGRAFIA

- [1] Ministerio de Minas y Energía, Unidad de Planeación Minero-Energética-, «UPME,»[Enlínea].Available:http://www1.upme.gov.co/DemandaEnergetica/MarcoNormatividad/PAI_PROURE_2017-2022.pdf#search=PAI_PROURE_2017-2022. [Último acceso: 08 05 2018].
- [2] Congreso de Colombia, «Colciencias,» 03 10 2001. [En línea]. Available: <http://www.colciencias.gov.co/sites/default/files/upload/reglamentacion/ley-697-2001.pdf>. [Último acceso: 06 07 2017].
- [3] C.G Zapata Garza, A. Uribe Durán y M. Demmler «La eficiencia energetica como ventaja competitiva empresarial sostenible en Mexico,» *Daena: International Journal of Good Conscience*, vol. 2, nº 12, pp. 67-89, 2017.
- [4] Ministerio de Minas y Energía-Unidad de Planeación Minero-Energética, Proyección de la Demanda de Energia Electrica y Potencia Maxima en Colombia, Bogota, 2016.
- [5] Unidad de Planeación Minero-Energética, «UPME,» 01 2015. [En línea]. Available: http://www.upme.gov.co/Docs/PEN/PEN_IdearioEnergetico2050.pdf. [Último acceso: 28 03 2018].
- [6] M. Zito, «La Sustentabilidad de Internet de las Cosas,» Centro de estudios en diseño y comunicacion , nº 7, pp. 37-44, 2018.
- [7] D. Meana Llorián, C. González García, B. C. Pelayo García-Bustelo, J. M. Cueva Lovelle y V. H. Medina Garcia «IntelliSenses: Sintiendo Internet de las Cosas,» *CISTI*, vol. 1, nº 6, pp. 226-231, 2016.
- [8] Y. Betancor Caro, «“Sistema doméstico de monitorización del consumo,» San Cristobal de la Laguna- España, 2014.
- [9] D. I Samaniego Idrovo y D.F Velesaca Orellana, «Diseño e Implementación de Un Medidor De Energía Electrónico Para Vivienda, Con Orientación A La Prevención De Consumo Y Ahorro Energético.,» Cuenca-Ecuador, 2016.
- [10] V.E Carbonero, «Monitorización del consumo eléctrico de un hogar:Procesado de datos mediante Arduino,» Pamplona-España, 2015.

- [11] R. J. Castro Tenorio, «Desarrollo e implementación de una aplicación para dispositivos móviles, con sistemas operativo android, para el control de luminarias y monitoreo de consumo de energía eléctrica de una vivienda,» Guayaquil-Ecuador, 2016.
- [12] L.G de la Cruz Puebla, «Aplicación android para medición inteligente de energía eléctrica en usuarios residenciales urbanos,» Quito-Ecuador, 2015.
- [13] IBM, «IBM Knowledge Center,» 25 04 2015. [En línea]. Available: https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac55710_.htm. [Último acceso: 14 06 2018].
- [14] O. Gonzales, A. Pavas y S. Sánchez, «Cuantificación del ahorro de energía eléctrica en clientes residenciales mediante acciones de gestión de demanda,» UIS Ingenierias, vol. 16, nº 2, pp. 217-226, 2017.
- [15] J.P Lozano Celis y W.C Guzmán Espitia, «Evaluación de demanda de energía eléctrica según hábitos de consumo actuales en la ciudad de Bogotá,» Bogota-Colombia , 2016.
- [16] A.A Roman Párraga y J.W Vega Sánchez , «Implementacion de un sistema de administracion energetica mediante Raspberry Pi, bajo las condiciones de la norma Iso 50001 aplicado a casas domesticas,» Guayaquil-Ecuador, 2017.
- [17] I. M. Ferreras Astorqui, «Sensor lot para monitorización de consumo de energía en continua,» Madrid-España, 2016.
- [18] G. Guacaneme Valbuena y D. A Pardo Agudelo, «Diseño e Implementación de un sistema de medición de consumo de energía eléctrica y agua potable remoto con interacción al usuario basado en el concepto “internet de las cosas”,» Bogota-Colombia, 2016.
- [19] A. Marín Parra, «Desarrollo de herramientas software para facilitar la gestión de la demanda de energía eléctrica: aplicación a usuarios residenciales,» Cartagena-Colombia, 2014.
- [20] D. M. Ruiz Martínez, Diseño de un sistema en cloud para controlar dispositivos iot vía internet, Bogotá, 2016.
- [21] R.Loureiro Garrido, PFC Estudio plataformas IoT, 2015

- [22] MathWorks, «Thingspeak,» MathWorks, 1994-2018. [En línea]. Available: <https://www.mathworks.com/help/thingspeak/>. [Último acceso: 20 06 2018].
- [23] CloudMQTT, «CloudMQTT,» [En línea]. Available: <https://www.cloudmqtt.com/docs-faq.html>. [Último acceso: 10 06 2018].
- [24] Organización de Consumidores y Usuarios, «OCU.ORG,» [En línea]. Available: <https://www.ocu.org/vivienda-y-energia/nc/calculadora/consumo-en-stand-by>. [Último acceso: 08 05 2018].
- [25] Instituto para la Diversificación y Ahorro de la Energía, «IDAE,» [En línea]. Available: http://www.idae.es/uploads/documentos/documentos_Documentacion_Basica_Residencial_Unido_c93da537.pdf. [Último acceso: 08 05 2018].
- [26] J. J. Rabadán Barastegui, «Diseño y desarrollo de una red modbus Rtu basada en arduino,» sevilla-españa, 2017.
- [27] C. Garcia Salvador, «Implementación de analizadores de protocolos de comunicaciones SPI, I2C,» Madrid-España, 2014.
- [28] D. Lizárraga Osuna, «Sistema de administración de energía utilizando Internet de las Cosas (IoT),» Monterrey-Mexico , 2018.
- [29] J. D. Serrano Andrés, «Diseño, implementación e integración de un sistema de medición de variables de entorno en un sistema iot con software y hardware libre,» Valencia-España, 2016.
- [30] Android, «Android Studio,» [En línea]. Available: <https://developer.android.com/studio/intro/>. [Último acceso: 20 08 2018].

ANEXOS

Anexo 1. Link del Manual de usuario del Medidor Monofásico ED610D2

<https://www.electronicaembajadores.com/datos/manuales/ed/ed61/ed610d2.pdf>

Anexo 2. Link del Manual de usuario del Medidor de Potencia PM1200

<https://www.schneider-electric.es/es/product/METSEPM1200/medidor-de-potencia-pm1200-con-energ%C3%ADa-y-demanda%2C-con-comunicaci%C3%B3n/?range=7206-pm1000-series&node=166388355-productos#pdp-download>

Anexo 3. Link del Manual de usuario del Medidor Eastron sdm230

http://www.eastrongroup.com/data/uploads/Eastron_SDM230Modbus_user_manual_V1_4_2015.pdf

Anexo 4. Librerías Arduino Mega2560

```
#include < SimpleModbusMasterSDM120.h >  
#include <Wire.h>  
#include <EmonLib.h>  
//#include <FreqPeriodCounter.h>
```

Anexo 5. Código Arduino Mega2560

```
//////////medidores emonlib  
//#include <LiquidCrystal.h>  
#include "EmonLib.h" // Include Emon Library  
EnergyMonitor emon1;  
EnergyMonitor emon2;  
#include "FreqPeriodCounter.h"  
#define MainPeriod 100  
//Inicializa o LCD  
//LiquidCrystal lcd(12, 11, 5, 4, 3, 6);// Create an instance  
FreqPeriodCounter counter(3, micros, 0);  
static float realPower;  
static float aparentPower;  
static float powerFACTOR;  
static float Vrms;  
static float Irms;  
static float reativPower;  
static float kwhTotal;  
static float kWh;
```

```
boolean pulso;
long pulsos;
float Freq;
float Freq2;
float fi;
float fi2;
float potenciaM;
float consumo;
float consumo_aux;
long previousMillis = 0;
int lastSampleTime, sampleDuration;
volatile unsigned long duration=0; // accumulates pulse width
volatile unsigned int pulsecount=0;
volatile unsigned long duration2=0; // accumulates pulse width
volatile unsigned int pulsecount2=0;
volatile unsigned long previousMicros=0;
unsigned long previousMillisem = 0;
unsigned long previousMilliscorr = 0;
//////////
/* SDM120C Modbus RTU Protocol con Arduino.
 * version 0.0.4 by @cosmopaco.
 *
 * Materiales
 * Arduino Mega
 * Modulo RS485
 * Conexiones (Configuracion por defecto) Puerto serial 1
 * Arduino pin Modulo RS485 pin
 * 19 RO (receive out)
 * 18 DI (data in)
 * 17 DE/RE (data enable/receive enable).
 *
 */
char prueba;
int32_t prueba2;
////////Comunicacion I2C Cn modulo wifi ---(MAESTRO--->MODULO WIFI
ESCALVO-->Arduino Mega)
#include <Wire.h>
int table[32]={};
volatile int Work=0;
#include <SimpleModbusMasterSDM120.h>
```



```
// Direcciones registros de datos solo lectura. Valores tipo float.
// Utilizar funcion 04 lectura, numero de bytes 4.
// Las marcadas en mayusculas son las de el este ejemplo.
#define VOL_ADR 0X0000 // VOLTAJE.
#define CUR_ADR 0X0006 // CORRIENTE.
#define POW_ADR 0X000C // POTENCIA ACTIVA.
#define VAM_ADR 0X0012 // Potencia Aparente.
#define PRE_ADR 0X0018 // Potencia Reactiva.
#define PFA_ADR 0X001E // Factor de potencia.
#define FRE_ADR 0X0046 // Frecuencia.
#define ENE_ADR 0X0048 // ENERGIA IMPORTADA KWH
#define POE_ADR 0X004A // Energia exportada.
#define NEE_ADR 0X0156 // Energia activa Total.
//
#define SDM120C_METER_NUMBER 1
#define SDM120C_BAUDRATE 9600
#define SDM120C_BYTEFORMAT SERIAL_8O1 //Prty n
//
#define TIMEOUT 1000 //TIEMPO DE SALIDA DE DATOS EN PANTALLA
#define POLLING 200 // the scan rate
#define RETRYCOUNT 10 // numero de reintentos, para volver set the
"connection" variable to true.
#define TXENPIN 2 // Pin cambio recibir/trasmiste para el driver RS485
enum
{
PACKET1,
PACKET2,
PACKET3,
PACKET4,
PACKET5,
PACKET6,
PACKET7,
PACKET8,
PACKET9,
PACKET10,
TOTAL_NO_OF_PACKETS // leave this last entry
};

// Create an array of Packets to be configured
Packet packets[TOTAL_NO_OF_PACKETS];
// Create a packetPointer to access each packet
```

```
// individually. This is not required you can access
// the array explicitly. E.g. packets[PACKET1].id = 2;
// This does become tedious though...
packetPointer volPacket = &packets[PACKET1];
packetPointer curPacket = &packets[PACKET2];
packetPointer powPacket = &packets[PACKET3];
packetPointer enePacket = &packets[PACKET4];
packetPointer potapaPacket = &packets[PACKET5];
packetPointer potrePacket = &packets[PACKET6];
packetPointer ftpPacket = &packets[PACKET7];
packetPointer frePacket = &packets[PACKET8];
packetPointer eneexPacket = &packets[PACKET9];
packetPointer eneatPacket = &packets[PACKET10];
// Union
union datas{
byte b[4];
float f;
unsigned int Array[2];
}voltage, current ,power,
energy,potapa,potre,ftp,fre,eneex,eneat,VrmsFloatA,aparentPoweFloatA,IrmsFloat
A,realPowerFloatA,powerFactorFloatA,reativPowerFloatA,FreqFloatA,kwhTotalFlo
atA,VrmsFloatA2,aparentPoweFloatA2,IrmsFloatA2,realPowerFloatA2,powerFacto
rFloatA2,reativPowerFloatA2,FreqFloatA2,kwhTotalFloatA2;;

union Float_Byte
{
float datoF;
byte datoB[4];
}CorrienteFloat,potenciaFloat,energiaFloat,paparenteFloat,preactivaFloat,fpFloat,fr
ecFloat,VrmsFloat,aparentPoweFloat,IrmsFloat,realPowerFloat,powerFactorFloat,r
eativPowerFloat,FreqFloat,kwhTotalFloat,VrmsFloat2,aparentPoweFloat2,IrmsFloa
t2,realPowerFloat2,powerFactorFloat2,reativPowerFloat2,FreqFloat2,kwhTotalFloa
t2;

void setup() {
// modbus_construct(packet, id, function, address, data, register array)
// For functions 1 & 2 data is the number of points
// For functions 3, 4 & 16 data is the number of registers
// For function 15 data is the number of coils
modbus_construct_SDM120(volPacket, SDM120C_METER_NUMBER,
VOL_ADR, voltage.Array);
```

```
modbus_construct_SDM120(curPacket, SDM120C_METER_NUMBER,  
CUR_ADR, current.Array);  
modbus_construct_SDM120(powPacket, SDM120C_METER_NUMBER,  
POW_ADR, power.Array);  
modbus_construct_SDM120(enePacket, SDM120C_METER_NUMBER,  
ENE_ADR, energy.Array);  
modbus_construct_SDM120(potapaPacket, SDM120C_METER_NUMBER,  
VAM_ADR, potapa.Array);  
modbus_construct_SDM120(potrePacket, SDM120C_METER_NUMBER,  
PRE_ADR, potre.Array);  
modbus_construct_SDM120(ftpPacket, SDM120C_METER_NUMBER, PFA_ADR,  
ftp.Array);  
modbus_construct_SDM120(frePacket, SDM120C_METER_NUMBER, FRE_ADR,  
fre.Array);  
modbus_construct_SDM120(eneexPacket, SDM120C_METER_NUMBER,  
POE_ADR, eneex.Array);  
modbus_construct_SDM120(eneatPacket, SDM120C_METER_NUMBER,  
NEE_ADR, eneat.Array);
```

```
// Iniciamos comunicación modbus SERIAL1 Arduino Mega.  
modbus_configure(&Serial3, SDM120C_BAUDRATE, SDM120C_BYTEFORMAT,  
TIMEOUT, POLLING, RETRYCOUNT, TXENPIN, packets,  
TOTAL_NO_OF_PACKETS);  
Wire.begin(8);          // Slave id #8  
Wire.onRequest(requestEvent); // register event  
Wire.onReceive(receiveEvent);  
//Iniciamos puerto serial"0" Arduino Mega para visualizar datos.  
Serial.begin(9600);  
Serial.println(F("Test SDM120-Modbus"));  
pinMode(13, OUTPUT); // Led pin de Arduino Mega;  
digitalWrite(13, LOW);  
  emon1.voltage(1, 110, 1.7); // Voltage: input pin, calibration, phase_shift  
//VOLTAJE TRANSFORMADOR  
}  
void loop() {  
  static float voltaje,potencia,energia,paparente,preactiva,fp,frec,v;  
  static float corriente;  
  static boolean b = false ;  
  static unsigned long oldMillis;  
  modbus_update();  
  if( (millis()-oldMillis) > POLLING){
```

```
unsigned long temp;
digitalWrite(13,b);
b++;
oldMillis=millis();
Serial.println("  MEDIDAS:");
  voltaje = (float)voltage.f;
Serial.print(F(" Voltaje="));
Serial.println(voltaje, 1);
  corriente= (float)current.f;
Serial.print(F(" Corriente="));
  Serial.print("FLOAAT corriente ");
  CorrienteFloat.datoF =corriente;
Serial.println(CorrienteFloat.datoF);
  potencia= (float)power.f;
Serial.print(F(" Potencia="));
Serial.println(potencia, 2);
  Serial.print("FLOAAT potencia ");
  potenciaFloat.datoF =potencia;
Serial.println(potenciaFloat.datoF);
  energia= (float)energy.f;
  Serial.print(F(" Energia="));
  Serial.println(energia, 2);
  Serial.print("FLOAAT energia ");
  energiaFloat.datoF =energia;
Serial.println(energiaFloat.datoF);
  paparente= (float)potapa.f;
  Serial.print(F(" Potencia Aparente="));
// node.writeSingleRegister(6,paparente);
  Serial.println(paparente, 2);
  Serial.print("FLOAAT paparente ");
  paparenteFloat.datoF =paparente;
Serial.println(paparenteFloat.datoF);
  preactiva= (float)potre.f;
  Serial.print(F(" Potencia Reactiva="));
//node.writeSingleRegister(7,preactiva);
  Serial.println(preactiva);
  Serial.print("FLOAAT reactiva");
  preactivaFloat.datoF =preactiva;
Serial.println(preactivaFloat.datoF);
  fp= (float)ftp.f;
  Serial.print(F("Factor de potencia="));
```

```
Serial.println(fp, 2);
Serial.print("FLOAAT fp");
fpFloat.datoF =fp;
Serial.println(fpFloat.datoF);

frec= (float)fre.f;
Serial.print(F("Frecuencia="));
//node.writeSingleRegister(8,frec);
Serial.println(frec, 2);
Serial.print("FLOAAT frec");
frecFloat.datoF =frec;
Serial.println(frecFloat.datoF);
v= (float)eneex.f;
Serial.print(F("Energia Exportada="));
Serial.println(v, 2);
v= (float)eneat.f;
Serial.print(F("Energia ACTIVA TOTAL="));
Serial.println(v, 2);
//////////medidores emonlib
  unsigned long currentMillisem = millis();
  if (currentMillisem - previousMillisem >= 5000) { //envia la temperatura cada 15
segundos
    previousMillisem = currentMillisem;
emon1.calcVI(20,2000); // Calculate all. No.of half wavelengths (crossings),
time-out
  Vrms      = emon1.Vrms; //extract Vrms into Variable
  unsigned long currentMillis = millis();
  if (currentMillis - previousMillis >= MainPeriod)
  {
    previousMillis = currentMillis;
    // need to bufferize to avoid glitches
    unsigned long _duration = duration;
    unsigned long _pulsecount = pulsecount;
    duration = 0; // clear counters
    pulsecount = 0;
    Freq = 1e6 / float(_duration);
    Freq *= _pulsecount; // calculate F
    // output time and frequency data to RS232
    Serial.print(currentMillis);
    Serial.print(" "); // separator!
    Serial.print(Freq);
```

```
    Serial.print(" "); // separator!  
    Serial.print(_pulsecount);  
    Serial.print(" ");  
    Serial.println(_duration);  
    }  
    kWhTotal += ((realPower*1)/3600); // Cálculo de la Energía total consumida (En  
kWh)  
    kWhTotal2 += ((realPower2*1)/3600); // Cálculo de la Energía total consumida  
(En kWh)  
    lcdUpdate();  
    }  
}  
table[0]=CorrienteFloat.datoB[0];  
table[1]=CorrienteFloat.datoB[1];  
table[2]=CorrienteFloat.datoB[2];  
table[3]=CorrienteFloat.datoB[3];  
table[4]=potenciaFloat.datoB[0];  
table[5]=potenciaFloat.datoB[1];  
table[6]=potenciaFloat.datoB[2];  
table[7]=potenciaFloat.datoB[3];  
table[8]=energiaFloat.datoB[0];  
table[9]=energiaFloat.datoB[1];  
table[10]=energiaFloat.datoB[2];  
table[11]=energiaFloat.datoB[3];  
table[12]=paparenteFloat.datoB[0];  
table[13]=paparenteFloat.datoB[1];  
table[14]=paparenteFloat.datoB[2];  
table[15]=paparenteFloat.datoB[3];  
table[16]=preactivaFloat.datoB[0];  
table[17]=preactivaFloat.datoB[1];  
table[18]=preactivaFloat.datoB[2];  
table[19]=preactivaFloat.datoB[3];  
  
table[20]=fpFloat.datoB[0];  
table[21]=fpFloat.datoB[1];  
table[22]=fpFloat.datoB[2];  
table[23]=fpFloat.datoB[3];  
table[24]=frecFloat.datoB[0];  
table[25]=frecFloat.datoB[1];  
table[26]=frecFloat.datoB[2];  
table[27]=frecFloat.datoB[3];
```

```
table[28]=VrmsFloat.datoB[0];  
table[29]=VrmsFloat.datoB[1];  
table[30]=VrmsFloat.datoB[2];  
table[31]=VrmsFloat.datoB[3];  
Serial.print("Soy El NODEMCU-->: ");  
  Serial.print(Work);  
  digitalWrite(6,Work);  
  
}
```

```
/////funciones I2C  
// function that executes whenever data is requested from master  
void requestEvent() {  
  uint8_t Buffer[32];  
  Buffer[0]=table[0];  
  Buffer[1]=table[1];  
  Buffer[2]=table[2];  
  Buffer[3]=table[3];  
  Buffer[4]=table[4];  
  Buffer[5]=table[5];  
  Buffer[6]=table[6];  
  Buffer[7]=table[7];  
  Buffer[8]=table[8];  
  Buffer[9]=table[9];  
  Buffer[10]=table[10];  
  Buffer[11]=table[11];  
  Buffer[12]=table[12];  
  Buffer[13]=table[13];  
  Buffer[14]=table[14];  
  Buffer[15]=table[15];  
  Buffer[16]=table[16];  
  Buffer[17]=table[17];  
  Buffer[18]=table[18];  
  Buffer[19]=table[19];  
  Buffer[20]=table[20];  
  Buffer[21]=table[21];  
  Buffer[22]=table[22];  
  Buffer[23]=table[23];  
  Buffer[24]=table[24];  
  Buffer[25]=table[25];  
  Buffer[26]=table[26];
```

```
Buffer[27]=table[27];
Buffer[28]=table[28];
Buffer[29]=table[29];
Buffer[30]=table[30];
Buffer[31]=table[31];
Wire.write(Buffer,32);
}
void receiveEvent(int howMany)
{Work = Wire.read();
}
void lcdUpdate(){
// lcd.clear();
if (isnan(Vrms)){// retorna "true" se o valor não for numérico
  Vrms = 0.00;
}
if (isnan(Irms)){// retorna "true" se o valor não for numérico
  Irms = 0.00;
}
if (isnan(realPower)){// retorna "true" se o valor não for numérico
  realPower = 0.00;
}
if (isnan(reativPower)){// retorna "true" se o valor não for numérico
  reativPower = 0.00;
}
if (isnan(powerFActor)){// retorna "true" se o valor não for numérico
  powerFActor = 0.00;
}
if (isnan(Freq)){// retorna "true" se o valor não for numérico
  Freq = 0.00;
}
if (isnan(kwhTotal)){// retorna "true" se o valor não for numérico
  kwhTotal = 0.00;
}
if (isnan(Vrms2)){// retorna "true" se o valor não for numérico
  Vrms2 = 0.00;
}
if (isnan(Irms2)){// retorna "true" se o valor não for numérico
  Irms2 = 0.00;
}
if (isnan(realPower2)){// retorna "true" se o valor não for numérico
  realPower2 = 0.00;
}
```



```
}  
if (isnan(reativPower2)){// retorna "true" se o valor não for numérico  
    reativPower2 = 0.00;  
}  
if (isnan(powerFActor2)){// retorna "true" se o valor não for numérico  
    powerFActor2 = 0.00;  
}  
if (isnan(kwhTotal2)){// retorna "true" se o valor não for numérico  
    kwhTotal2 = 0.00;  
}  
  
*/ Serial.print("NUEVA LINEA ");  
Serial.print("\n");  
  
    Serial.print(Vrms,2);  
    VrmsFloat.datoF=Vrms;  
  
Serial.print("V---");  
Serial.print(aparentPower ,2);  
aparentPoweFloat.datoF=aparentPower;  
Serial.print("VA---");  
Serial.print(Irms,2);  
IrmsFloat.datoF=Irms;  
Serial.print("A---");  
Serial.print(abs(realPower),2);  
realPowerFloat.datoF=realPower;  
Serial.print("W---");  
Serial.print("FP:");  
Serial.print(abs(powerFActor),2);  
powerFActorFloat.datoF=powerFActor;  
Serial.print("---");  
Serial.print(reativPower,2);  
reativPowerFloat.datoF=reativPower;  
Serial.print("VAr--");  
Serial.print("Fr:");  
Serial.print(Freq,0 );  
FreqFloat.datoF=Freq;  
Serial.print("Hz--");  
Serial.print(abs(kwhTotal),2);  
Serial.print(" kWh");  
kwhTotalFloat.datoF=kwhTotal;
```

```
}  
//=====CALCULOS DE FRECUENCIA //  
void myinhandler() // interrupt handler  
{  
  unsigned long currentMicros = micros();  
  duration += currentMicros - previousMicros;  
  previousMicros = currentMicros;  
  pulsecount++;  
}
```

Anexo 6. Librerías Nodemcu ESP8266

```
#include <ESP8266WiFi.h>  
#include <Wire.h>  
#include <ArduinoOTA.h>  
#include <WiFiClient.h>  
#include <PubSubClient.h>  
#include <EmonLib.h>  
//#include <FreqPeriodCounter.h>
```

Anexo 7. Configuración de variables para Nodemcu ESP8266.

- *Variables de conexión Wifi (SSID y contraseña).
- *Variables servidor externo (Dirección IP servidor, Dirección URL servidor, URL para recepción de datos).
- *Variables Thingspeak (URL servidor Thingspeak, apiKey, canal Thingspeak).
- *Variables CloudMQTT (Dirección IP instancia CloudMQTT, puerta de enlace instancia MQTT, nombre de usuario, contraseña).
- *Variables interrupciones.
- *Variables Google Spreadsheets (URL Script Google, puerto HTTP Script).

Anexo 8. Envío de variables a BD MYSQL de IFasNet (archivo *enviardatos.php*)

```
Archivo PHP

//Conexión a Base de datos Mysql
(mysql_connect)

-Nombre del servidor (servername).
-Usuario (username).
-Contraseña (password).

//Obtención de variables (parámetros
eléctricos)

//Insertar variables a la BD MYSQL
(mysql_query)
-intertar en (INSERT INTO)
-nombre BD.
-nombre tabla y columnas
```

Anexo 9. Estructura HTML-conexión con WebSocket CloudMQTT.

```
<html>
  <head>
    //CDN framework Bootstrap
    //CDN Highcharts
  </head>

  <body>
    <script src='mqttws31.js'>
    </script>

    <script>
    //Configuracion
    credenciales CloudMqtt

    //Obtención de parámetros
    eléctricosy gráfica con
    Highcharts </script>
  </body>

</html>
```

Anexo 10. Script conexión a topic CloudMQTT.

```
<script>  
  
function onConnect()  
//función para ejecutar conexion como  
cliente mqtt  
function onConnectionLost()  
//se ejecuta cuando se pierde la  
conexion con el WebSocket  
function onMessageArrived()  
//es llamada cuando llega un mensaje,  
dentro de esta función se obtienen los  
parámetros eléctricos mediante los  
topic mqtt.  
function onFailure()  
//se ejecuta en el momento en el que  
hay un error con el servidor websocket.  
</script>
```

Anexo 11. Configuración-Reacción ThingSpeak.

Apps / React / New

React Name

Condition Type

Test Frequency

Condition

field

Action

Options Run action only the first time the condition is met
 Run action each time condition is met

Anexo 12. Script para configuración de Google Spreadsheets

```
/**  
 * Function doGet: Parse received data from GET request,  
 get and store data which is corresponding with header row in Google  
 Spreadsheet  
 */  
function doGet(e) {  
  Logger.log( JSON.stringify(e) ); // view parameters  
  var result = 'Ok'; // assume success  
  if (e.parameter == 'undefined') {  
    result = 'No Parameters';  
  }  
  else {  
    var sheet_id = '1ZSwucJq62EVLp93Kgek-22jnk3Bl6bp5c-Xkl3WZpf4';  
      // Spreadsheet ID  
    var sheet = SpreadsheetApp.openById(sheet_id).getActiveSheet(); //  
get Active sheet  
    var newRow = sheet.getLastRow() + 1;  
    var rowData = [];  
    rowData[0] = new Date();  
      // Timestamp in column A  
  
    var ahora=new Date();  
    var hora=ahora.getHours();  
    var minutos=ahora.getMinutes();  
    var horatotal;  
    horatotal=hora + ":" + minutos;  
  
    rowData[1] = horatotal;  
  
    for (var param in e.parameter) {  
      Logger.log('In for loop, param=' + param);  
      var value = stripQuotes(e.parameter[param]);  
      Logger.log(param + ':' + e.parameter[param]);  
      switch (param) {  
  
        case 'Corriente': //Parameter  
          rowData[2] = value; //Value in column B  
          result = 'Written on column B';  
          break;
```

```
case 'potencia': //Parameter
    rowData[3] = value; //Value in column C
    result += ' ,Written on column C';
    break;

case 'preactiva': //Parameter
    rowData[4] = value; //Value in column D
    result = 'Written on column D';
    break;

case 'paparente': //Parameter
    rowData[5] = value; //Value in column E
    result += ' ,Written on column E';
    break;

    case 'frec': //Parameter
    rowData[6] = value; //Value in column F
    result = 'Written on column F';
    break;

case 'fp': //Parameter
    rowData[7] = value; //Value in column G
    result += ' ,Written on column G';
    break;

case 'energia': //Parameter
    rowData[8] = value; //Value in column H
    result = 'Written on column H';
    break;

default:
    result = "unsupported parameter";
}
}
Logger.log(JSON.stringify(rowData));
// Write new row below
var newRange = sheet.getRange(newRow, 1, 1, rowData.length);
newRange.setValues([rowData]);
}
// Return result of operation
return ContentService.createTextOutput(result);
```

```
}  
/**  
 * Remove leading and trailing single or double quotes  
 */  
function stripQuotes( value ) {  
    return value.replace(/^["]|[""]$/g, "");  
}  
//-----  
// End of file
```

Anexo 13. Estructura archivo AndroidManifest.xml. Android Studio.

```
<manifest>  
  
<configuración de permisos para usar internet >  
  
    <application  
        //nombre de la app  
        //icono de la app  
        //titulo que se va a mostrar en la app  
        //tema de fondo para la app >  
  
        -Configuración de actividades a usar en la app  
        <activity .Menu,MainActivity //actividad menu />  
        <activity .SignUp //actividad registro usuario />  
        <activity .Login //actividad login usuario (Actividad  
            principal de la app) />  
  
    </application>  
  
</manifest>
```

Anexo 14. Código XML Layout activity_login.xml. Android Studio.

```
<DrawerLayout> //Contenedor Menu Lateral

    android:id //nombre identificador menu lateral
    android:layout_width //ancho contenedor menú
    android:layout_height //altura contenedor menú

    <android.support.design.widget.NavigationView
        //Contenedor de opciones de navegación
        //identificador navigationView
        //ancho de contenedor
        //alto de contenedor
        //posición para opciones de menú />

</DrawerLayout>
```

Anexo 15. Código XML Layout Fragment_menu_2.xml (Menu2.java). Android Studio.

```
<RelativeLayout> //Contenedor posición
relativa con otros elementos

    android:id //nombre identificador contenedor
    android:layout_width //ancho contenedor menú
    android:layout_height //altura contenedor menú

    <android.support.v7.widget.RecyclerView
        //Contenedor de elementos en forma de lista
        //identificador RecyclerView
        //ancho de contenedor
        //alto de contenedor
        //margen de CardView para ordenar los
        elementos
        //orientación para scrollbar />

</RelativeLayout>
```