



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**

---

---

**CENTRO UNIVERSITARIO UAEM ECATEPEC**

“SISTEMA DE CONTROL Y MONITOREO DE CONSUMO  
ELECTRICO A TRAVES DEL INTERNET DE LAS COSAS”

# **T E S I S**

Que para obtener el título de

**MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A:**

**EL ING. ALEJANDRO ANDRÉS SERAPIO CARMONA**

**Asesor: DR. EN C. RODOLFO ZOLÁ GARCÍA LOZANO**

**Asesores adjuntos: Dr. Ismael Díaz Rangel  
M. en I.S.C Cuauhtémoc Hidalgo Cortés.**

**Ecatepec de Morelos, Estado de México Junio 2020**





# Índice

|  |    |
|--|----|
| Capítulo 1 Introducción .....                      | 1  |
| 1.1 Objetivo general .....                         | 2  |
| 1.2 Objetivos particulares .....                   | 2  |
| 1.3 Actividades .....                              | 2  |
| 1.4 Justificación .....                            | 3  |
| 1.5 Descripción del capitulado.....                | 4  |
| Capítulo 2 Marco teórico y antecedentes.....       | 5  |
| 2.1 Estado del arte .....                          | 5  |
| 2.2 La energía eléctrica en México.....            | 6  |
| 2.3 Medidores de consumo eléctrico.....            | 8  |
| 2.3.1 Medidores analógicos.....                    | 9  |
| 2.3.2 Medidores digitales .....                    | 11 |
| 2.4 Internet de las cosas .....                    | 12 |
| 2.4.1 Origen .....                                 | 13 |
| 2.4.2 Aplicaciones del internet de las cosas ..... | 13 |
| 2.5 Tarjetas de desarrollo.....                    | 15 |
| 2.5.1 Dispositivos lógicos programables.....       | 15 |
| 2.5.2 Sistemas basados en microprocesadores .....  | 16 |
| 2.5.2.1 Raspberry Pi .....                         | 17 |
| 2.5.2.2 Intel Galileo .....                        | 18 |
| 2.5.2.3 Arduino UNO.....                           | 19 |





|  |    |
|--|----|
| 2.5.2.4 Wemos, NodeMCU .....   | 20 |
| 2.6 Sensores y dispositivos electrónicos.....  | 21 |
| 2.6.1 Sensor de corriente ACS-712 .....  | 22 |
| 2.6.2 Módulo MicroSD .....   | 24 |
| 2.6.3 Pantalla LCD.....  | 25 |
| 2.6.4 Sensor PIR .....   | 26 |
| 2.6.5 Módulo relevador .....   | 27 |
| Relevador SLA-05VDC-SL-A .....   | 27 |
| Relevador SDR-05VDC-SL-C .....   | 28 |
| 2.6.6 Fuente de alimentación AC/DC .....   | 29 |
| 2.7 Entorno de programación Arduino.....   | 30 |
| 2.7.1 Importar bibliotecas.....  | 33 |
| 2.7.2 Biblioteca WiFiManager .....   | 35 |
| 2.7.3 Biblioteca CayenneMQTTESP8266.....   | 36 |
| 2.8 Circuitos impresos .....   | 37 |
| 2.8.1 Transferencia por planchado .....  | 37 |
| 2.8.2 Eagle.....   | 38 |
| 2.8.2 Relación entre la dimensión de una pista con la cantidad de corriente<br>..... | 44 |
| 2.9 Diseño e impresión en tres dimensiones.....                                      | 46 |
| 2.9.1 SolidWorks.....  | 46 |
| 2.9.2 Ultimaker Cura .....   | 52 |
| 2.9.3 Impresora Anycubic i3 Mega. ....   | 57 |





|   |     |
|---|-----|
| 2.10 Plataformas para el Internet de las Cosas.....               | 58  |
| 2.10.1 Amazon Web Services.....                                   | 58  |
| 2.10.2 Android Things de Google .....                             | 59  |
| 2.10.3 Microsoft Azure IoT.....                                   | 60  |
| 2.10.4 myDevices Cayenne .....                                    | 61  |
| Capítulo 3 Desarrollo experimental.....                           | 66  |
| 3.1 Diagrama a bloques .....                                      | 66  |
| 3.2 Diagramas de flujo .....                                      | 67  |
| 3.2.1 Diagrama de flujo para obtener la potencia instantánea..... | 67  |
| 3.2.2 Diagrama de flujo para la comunicación inalámbrica.....     | 69  |
| 3.2.3 Diagrama de flujo configuración Cayenne .....               | 69  |
| 3.3 Medición de la corriente eléctrica.....                       | 71  |
| 3.4 Código en el entorno de Arduino.....                          | 74  |
| 3.5 Desarrollo de la aplicación en Cayenne .....                  | 83  |
| 3.6 Desarrollo y construcción del prototipo final.....            | 90  |
| 3.6.1 Medidor consumo general .....                               | 93  |
| Construcción del circuito esquemático en Eagle.....               | 95  |
| Tarjeta de circuito impreso .....                                 | 98  |
| 3.6.2 Medidor consumo individual (contacto).....                  | 99  |
| Construcción del circuito esquemático en Eagle.....               | 101 |
| Tarjeta de circuito impreso .....                                 | 103 |
| 3.7 Diseño e impresión 3D del contenedor .....                    | 107 |
| 3.7.1 Contenedor medidor general .....                            | 107 |





|   |     |
|---|-----|
| 3.7.2 Contenedor medidor individual .....   | 112 |
| Capítulo 4 Pruebas y resultados .....   | 118 |
| 4.1 Prueba convertidor analógico digital. ....                                    | 118 |
| Resultados de la prueba con el ADC .....  | 119 |
| Prueba de divisor de tensión .....  | 120 |
| Resultados prueba divisor de tensión.....   | 120 |
| Prueba salida sensor de corriente ACS-712.....                                    | 121 |
| Resultados de la prueba del sensor de corriente ACS-712. ....                     | 121 |
| Prueba para el algoritmo para la medición de corriente. ....                      | 122 |
| Resultados obtenidos de la prueba para el algoritmo de medición de corriente..... | 122 |
| Prueba para el algoritmo para la obtención de potencia en Watt-hora. ....         | 123 |
| Resultados obtenidos de la medición de Wh por intervalos de tiempo. ....          | 124 |
| 4.2 Pruebas realizadas al primer prototipo .....                                  | 125 |
| Prueba de medición de voltaje de alimentación.....                                | 127 |
| Resultados prueba de encendido .....  | 127 |
| Prueba de comunicación con computadora.....                                       | 128 |
| Resultados obtenidos de la comunicación con la computadora .....                  | 128 |
| 4.3 Pruebas realizadas al segundo prototipo. ....                                 | 129 |
| Prueba de alimentación del prototipo. ....  | 131 |
| Resultados de la prueba de alimentación .....                                     | 132 |
| Prueba de medición de corrientes con cargas resistivas.....                       | 132 |
| Resultados de la prueba medición de corrientes con cargas resistivas.....         | 133 |





|   |     |
|---|-----|
| 4.4 Pruebas y resultados del tercer prototipo .....                   | 134 |
| Prueba a la fuente de alimentación .....                              | 134 |
| Resultados fuente de alimentación .....                               | 134 |
| Prueba de las borneras de conexión .....                              | 135 |
| Resultados de las borneras de conexión .....                          | 135 |
| 4.5 Pruebas realizadas con el entorno Cayenne .....                   | 135 |
| Prueba de conexión con Cayenne .....                                  | 135 |
| Resultados obtenidos conexión con Cayenne .....                       | 136 |
| Prueba notificación fuera de línea .....                              | 136 |
| Resultados obtenidos notificación fuera de línea.....                 | 137 |
| Prueba notificación rangos de consumo .....                           | 137 |
| Resultados notificación rangos de consumo.....                        | 138 |
| Prueba visualización de información en widgets. ....                  | 138 |
| Resultados visualización de información en plataforma y aplicación... | 139 |
| 4.6 Pruebas realizadas al sistema final .....                         | 141 |
| Prueba de medición de consumo eléctrico .....                         | 141 |
| Resultados prueba de consumo eléctrico .....                          | 141 |
| Conclusiones .....  | 143 |
| Trabajo Futuro .....  | 144 |
| Referencias.....  | 145 |
| Referencias de figuras .....  | 149 |
| Anexos.....   | 157 |
| A) Código.....  | 157 |



|                            |     |
|----------------------------|-----|
| B) Plan de negocios.....   | 171 |
| C) Constancias .....       | 177 |
| D) Artículo publicado..... | 187 |

## Índice de figuras

|  |    |
|--|----|
| FIGURA 2.1. MEDIDOR ENERGÍA ELÉCTRICA ANALÓGICO. ....                            | 9  |
| FIGURA 2.2. MEDIDOR DE ENERGÍA ELÉCTRICA DIGITAL. ....                           | 11 |
| FIGURA 2.3. TARJETAS DE DESARROLLO. ....   | 15 |
| FIGURA 2.4. FPGA. ....   | 16 |
| FIGURA 2.5. RASPBERRY PI. ....   | 17 |
| FIGURA 2.6. TARJETA INTEL GALILEO. ....  | 18 |
| FIGURA 2.7. TARJETA ARDUINO UNO. ....  | 19 |
| FIGURA 2.8. TARJETA NODEMCU. ....  | 20 |
| FIGURA 2.9. SENSOR DE CORRIENTE ACS - 712. ....                                  | 22 |
| FIGURA 2.10. COMPORTAMIENTO DE LA SALIDA DEL SENSOR.....                         | 23 |
| FIGURA 2.11. TERMINALES ACS-712.....   | 23 |
| FIGURA 2.12. MÓDULO PARA MEMORIA MICROSD .....                                   | 24 |
| FIGURA 2.13. PANTALLA LCD 16x2.....  | 25 |
| FIGURA 2.14. SENSOR PIR .....  | 26 |
| FIGURA 2.15 RELEVADOR 5700 W.....  | 27 |
| FIGURA 2.16. MÓDULO RELEVADOR.....   | 28 |
| FIGURA 2.17. ENTORNO PROGRAMACIÓN ARDUINO.....                                   | 30 |
| FIGURA 2.18. CARGAR BIBLIOTECAS ENTORNO ARDUINO.....                             | 34 |
| FIGURA 2.19. ADMINISTRADOR DE BIBLIOTECAS. ....                                  | 34 |
| FIGURA 2.20. PANEL DE CONTROL EAGLE .....  | 39 |
| FIGURA 2.21. VENTANA DISEÑO DIAGRAMA ESQUEMÁTICO. ....                           | 40 |
| FIGURA 2.22. VENTANA DISEÑO PCB .....  | 43 |
| FIGURA 2.23. SOLIDWORKS. ....  | 46 |
| FIGURA 2.24. INTERFAZ SOLIDWORKS.....  | 47 |
| FIGURA 2.25. CREAR NUEVO DOCUMENTO. ....   | 49 |
| FIGURA 2.26. VENTANA ARCHIVO ACTIVO SOLIDWORKS. ....                             | 52 |
| FIGURA 2.27. ULTIMAKER CURA.....   | 52 |
| FIGURA 2.28. INTERFAZ CURA.....  | 53 |
| FIGURA 2.29. CREAR CUENTA AWS .....  | 59 |
| FIGURA 2.30. CONSOLA ANDROID THINGS .....  | 60 |
| FIGURA 2.31. VENTANA DE REGISTRO CAYENNE .....                                   | 62 |
| FIGURA 2.32. INTERFAZ USUARIO CAYENNE. ....                                      | 63 |
| FIGURA 3.1. DIAGRAMA A BLOQUES GENERAL .....                                     | 66 |
| FIGURA 3.2. DIAGRAMA DE FLUJO PARA OBTENER POTENCIA INSTANTÁNEA Y ACUMULADA..... | 67 |
| FIGURA 3.3. DIAGRAMA PARA COMPLETAR 1 SEGUNDO. ....                              | 68 |
| FIGURA 3.4. DIAGRAMA DE FLUJO CONFIGURACIÓN VALORES INICIALES. ....              | 69 |





|  |     |
|--|-----|
| FIGURA 3.5. DIAGRAMA DE FLUJO FUNCIONAMIENTO CAYENNE. ....                                     | 70  |
| FIGURA 3.6. SEÑAL DE SALIDA DEL SENSOR CON CARGA 2300W ALIMENTADA CON AC. ....                 | 71  |
| FIGURA 3.7. SEGMENTO CÓDIGO "DECLARACIÓN DE BIBLIOTECAS" .....                                 | 74  |
| FIGURA 3.8. SEGMENTO CÓDIGO "VARIABLES GLOBALES 1" .....                                       | 75  |
| FIGURA 3.9. SEGMENTO DE CÓDIGO "VARIABLES GLOBALES 2" .....                                    | 76  |
| FIGURA 3.10 SEGMENTO DE CÓDIGO "VARIABLES GLOBALES 3" .....                                    | 77  |
| FIGURA 3.11 SEGMENTO DE CÓDIGO "CONFIGURACIÓN O SETUP" .....                                   | 77  |
| FIGURA 3.12 SEGMENTO DE CÓDIGO SETUP 2 .....   | 78  |
| FIGURA 3.13 LECTURA VALORES EEPROM. ....   | 80  |
| FIGURA 3.14 SEGMENTO DE CÓDIGO "COMPARACIÓN DE LECTURAS" .....                                 | 80  |
| FIGURA 3.15. SEGMENTO DE CÓDIGO "CÁLCULO DE POTENCIA Y ACUMULATIVO DE ENERGÍA CONSUMIDA" ..... | 81  |
| FIGURA 3.16 OBTENCIÓN PRECIO POR EL CONSUMO REALIZADO. ....                                    | 82  |
| FIGURA 3.17 SEGMENTO DE CÓDIGO "ESPERA" .....  | 82  |
| FIGURA 3.18. SEGMENTO DE CÓDIGO "ENVÍO DE DATOS A CAYENNE" .....                               | 83  |
| FIGURA 3.19 SEGMENTO DE CÓDIGO "OBTENCIÓN DEL ESTADO DE LA CARGA" .....                        | 83  |
| FIGURA 3.20 CONFIGURACIÓN WIDGET GAUGE.....  | 85  |
| FIGURA 3.21 CONFIGURACIÓN CASILLA DE TEXTO.....  | 86  |
| FIGURA 3.22 CONFIGURACIÓN DÍAS DEL PERIODO. ....   | 86  |
| FIGURA 3.23 CONFIGURACIÓN BOTÓN ON/OFF .....   | 87  |
| FIGURA 3.24 CONFIGURACIÓN SLIDER TARIFA.....   | 87  |
| FIGURA 3.25 CONFIGURACIÓN BOTÓN SALVAR. ....   | 88  |
| FIGURA 3.26 CONFIGURACIÓN FUERA DE LÍNEA. ....   | 89  |
| FIGURA 3.27 CONFIGURACIÓN NOTIFICACIÓN TARIFAS. ....   | 90  |
| FIGURA 3.28. RESULTADOS CALCULO ANCHO DE PISTA. ....   | 92  |
| FIGURA 3.29. FUENTE DE ALIMENTACIÓN COMERCIAL .....  | 93  |
| FIGURA 3.30 DIAGRAMA MEDIDOR GENERAL. ....   | 94  |
| FIGURA 3.31 SÍMBOLO RELEVADOR. ....  | 96  |
| FIGURA 3.32 HUELLA RELEVADOR. ....   | 97  |
| FIGURA 3.33 TARJETA DE CIRCUITO IMPRESO MEDIDOR GENERAL .....                                  | 99  |
| FIGURA 3.34. DIAGRAMA ESQUEMÁTICO MEDIDOR DE CONSUMO INDIVIDUAL. ....                          | 100 |
| FIGURA 3.35. DISEÑO TARJETA DE CIRCUITO IMPRESO MEDIDOR INDIVIDUAL. ....                       | 105 |
| FIGURA 3.36. TARJETA MEDIDOR DE CONSUMO INDIVIDUAL.....  | 106 |
| FIGURA 3.37. POSICIÓN DE COMPONENTES. ....   | 106 |
| FIGURA 3.38 PRISMA RECTANGULAR. ....   | 107 |
| FIGURA 3.39 CAJA. ....   | 108 |
| FIGURA 3.40 POSTES PARA COLOCAR LA TARJETA. ....   | 109 |
| FIGURA 3.41 APERTURAS PARA INGRESAR LOS CONDUCTORES. ....                                      | 110 |
| FIGURA 3.42 CONTENEDOR DEL MEDIDOR GENERAL. ....   | 111 |
| FIGURA 3.43 TAPA CONTENEDOR MEDIDOR GENERAL. ....  | 112 |
| FIGURA 3.44. PRISMA RECTANGULAR. ....  | 113 |
| FIGURA 3.45. PRISMA RECTANGULAR HUECO. ....  | 113 |
| FIGURA 3.46. ORIFICIO PARA CLAVIJA DE ACCESO RÁPIDO. ....                                      | 114 |
| FIGURA 3.47. ORIFICIO ENCHUFE.....   | 115 |
| FIGURA 3.48 MODELO TRIDIMENSIONAL EN CURA. ....  | 115 |





|   |                                      |
|---|--------------------------------------|
| FIGURA 3.49. ACOMODO DEL MODELO TRIDIMENSIONAL. ....  | 116                                  |
| FIGURA 3.50 SISTEMA MONTADO EN EL CONTENEDOR. ....  | 117                                  |
| FIGURA 3.51. SISTEMA MEDIDOR GENERAL MONTADO EN LA CARCASA. ....  | 117                                  |
| FIGURA 4.1. DIAGRAMA CONEXIONES ADC. ....   | 118                                  |
| FIGURA 4.2. DIAGRAMA DE FLUJO PRUEBA ADC. ....  | 119                                  |
| FIGURA 4.3. DIAGRAMA CONEXIONES PRUEBA SALIDA SENSOR ACS 712. ....  | 121                                  |
| FIGURA 4.4. DIAGRAMA PARA MEDICIÓN DE CORRIENTE. ....   | 122                                  |
| FIGURA 4.5. DIAGRAMA CONEXIÓN CARGA. ....   | 123                                  |
| FIGURA 4.6 DIAGRAMA ESQUEMÁTICO PRIMER PROTOTIPO. ....  | 126                                  |
| FIGURA 4.7 PRIMER PROTOTIPO. ....   | 129                                  |
| FIGURA 4.8. DIAGRAMA ESQUEMÁTICO SEGUNDO PROTOTIPO. ....  | 130                                  |
| FIGURA 4.9 MENSAJES NOTIFICACIÓN FUERA DE LÍNEA. ....   | 137                                  |
| FIGURA 4.10 NOTIFICACIONES TARIFAS. ....  | 138                                  |
| FIGURA 4.11 RESULTADOS MEDICIONES. ....   | 141                                  |
| FIGURA 4.12 RESULTADOS MÉTODO BLAND-ALTMAN. ....  | 142                                  |
| FIGURA 7.1 LOGO EMPRESA. ....   | 171                                  |
| FIGURA 7.2 ANÁLISIS FODA. ....  | <b>¡ERROR! MARCADOR NO DEFINIDO.</b> |
| FIGURA 7.3 CERTIFICADO DE PARTICIPACIÓN EN EL TALLER "INTERNET OF THINGS – A PRACTICAL APPROACH". ....                      | 178                                  |
| FIGURA 7.4 CONSTANCIA DE PARTICIPACION RALLY LATINOAMERICANO DE INNOVACIÓN. ....  | 179                                  |
| FIGURA 7.5 CONSTANCIA POR PONENCIA EN "COLOQUIO DE LA MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN" CU UAEM VALLE DE MÉXICO. .... | 180                                  |
| FIGURA 7.6 RECONOCIMIENTO POR IMPARTIR EL TALLER "DESARROLLO DE SISTEMAS CON JAVA". ....                                    | 181                                  |
| FIGURA 7.7 CONSTANCIA POR IMPARTIR EL TALLER "DISEÑO DE CIRCUITOS PCB" ....   | 182                                  |
| FIGURA 7.8 CONSTANCIA 1ER LUGAR EN INNOVACIÓN NIVEL SEDE RALLY LATINOAMERICANO DE INNOVACIÓN. ....                          | 183                                  |
| FIGURA 7.9 RECONOCIMIENTO POR PONENCIA EN CU UAEM ECATEPEC. ....  | 184                                  |
| FIGURA 7.10 CONSTANCIA DE PARTICIPACIÓN EN TALLER "MODELO DE NEGOCIOS CANVAS". ....   | 185                                  |
| FIGURA 7.11 CERTIFICADO DE PARTICIPACIÓN COMO PONENTE EN EL CONGRESO CORE19. ....   | 186                                  |

## Índice de tablas

|   |     |
|---|-----|
| TABLA 2.1. TARIFA CONTRA TEMPERATURA. ....                    | 7   |
| TABLA 2.2. TARIFA APLICADA A CDMX. ....                       | 7   |
| TABLA 2.3. TARIFA DOMÉSTICA DE ALTO CONSUMO. ....             | 8   |
| TABLA 3.1. RELACIÓN WIDGET CON CANAL. ....                    | 84  |
| TABLA 3.2 COMPONENTES MEDIDOR GENERAL. ....                   | 95  |
| TABLA 3.3. COMPONENTES TERCER PROTOTIPO. ....                 | 101 |
| TABLA 4.1. RESULTADOS PRUEBA ADC. ....                        | 120 |
| TABLA 4.2. RESULTADOS DIVISOR DE TENSIÓN. ....                | 120 |
| TABLA 4.3. RESULTADOS VOLTAJE SALIDA SENSOR EFECTO HALL. .... | 122 |
| TABLA 4.4. RESULTADOS ALGORITMO MEDICIÓN CORRIENTE. ....      | 123 |
| TABLA 4.5 VALORES OBTENIDOS DE MANERA EXPERIMENTAL. ....      | 124 |
| TABLA 4.6 VALORES CALCULADOS. ....                            | 124 |





|  |     |
|--|-----|
| TABLA 4.7. NIVEL DE VOLTAJE EN LA TARJETA. ....                      | 127 |
| TABLA 4.8. RESULTADOS PRUEBA ALIMENTACIÓN SEGUNDO PROTOTIPO. ....    | 132 |
| TABLA 4.9. RESULTADOS MEDICIÓN DE CORRIENTES SEGUNDO PROTOTIPO. .... | 133 |
| TABLA 4.10 MEDICIÓN DE VOLTAJE. ....                                 | 134 |
| TABLA 7.1 MODELO CANVAS .....  | 174 |
| TABLA 7.2 PRESUPUESTO DE INVERSIÓN. ....                             | 175 |
| TABLA 7.3 PROYECCIÓN DE INGRESOS Y EGRESOS. ....                     | 176 |





## Capítulo 1 Introducción

La energía eléctrica es un recurso que ha brindado el apoyo para el desarrollo de las sociedades. Cuando el uso de la electricidad comenzó a brindar mayores comodidades a las personas, su uso se fue incrementando; tanto, que en la actualidad se encuentra en casi todas las actividades diarias de las personas: en el transporte, en la preparación de la comida, en la iluminación, comunicaciones, entretenimiento, trabajo, etc. Su uso es tan intenso, que se ha tenido la necesidad de buscar e invertir en tecnologías que produzcan mayores cantidades de potencial eléctrico a partir del uso de recursos renovables, con la finalidad de poder abastecer la creciente demanda de suministro eléctrico, pues al crecer, lo hacen también las actividades relacionadas a su producción; aun así, es necesario cuidar de los recursos energéticos que se utilizan para la producción de energía, por lo que los usuarios deberían utilizar eficientemente la energía eléctrica proporcionada. Para ayudar a gestionar de manera eficaz la energía que utiliza, se pretende brindar una herramienta que ayude a los usuarios a conocer de manera anticipada a la fecha de corte, el estado del consumo energético y un estimado de cuanto deberá pagar. Se presentan dos variantes, una está enfocada hacia el consumo total de una vivienda, y la segunda hacia cargas individuales –electrodomésticos, etc.-. Adicionalmente, el sistema brindará al usuario información acerca de la tarifa en la que se encuentra, el monto estimado a pagar por el consumo, el tiempo restante para la fecha de corte, y la opción de cortar el suministro eléctrico o restaurarlo de manera remota. La información se muestra en una aplicación móvil, que se comunica a través de Internet.





## 1.1 Objetivo general

Desarrollar un sistema electrónico que cuantifique la energía eléctrica consumida en un periodo, y proporcione al usuario mediante una aplicación móvil información útil para la gestión racional del consumo, así como la capacidad de interrupción del suministro eléctrico, usando el paradigma del Internet de las cosas.

## 1.2 Objetivos particulares

- Cuantificar el consumo de energía eléctrica de manera fiable mediante la herramienta creada.
- Diseñar una aplicación móvil con interfaz que muestre la cantidad de consumo de la energía eléctrica, monto a cubrir en el periodo actual, fecha de corte, y los KWh faltantes para pasar a la siguiente tarifa.
- Desarrollar y diseñar un módulo de medición de energía eléctrica.
- Enviar y recibir información del módulo de medición a la aplicación móvil mediante el paradigma de internet de las cosas.
- Activar o desactivar la alimentación de las cargas conectadas al sistema mediante la aplicación móvil.
- Notificar al usuario cuando se haya superado el rango de cada tarifa.
- Desarrollar un sistema con la capacidad para soportar la carga total de un hogar y uno que soporte cargas individuales de contactos.

## 1.3 Actividades

- Realizar una búsqueda acerca del estado de cobro de la energía eléctrica en México.
- Desarrollar un algoritmo que permita cuantificar la energía consumida.
- Crear un sistema de medición de consumo eléctrico, que cuantifique la energía consumida de manera individual (por toma de alimentación) y de manera general (por toma principal de alimentación eléctrica).



- Caracterizar un sensor de corriente para su uso con corriente alterna.
- Implementar el sistema de control basado en un microcontrolador.
- Incorporar un módulo de comunicación inalámbrica vía WiFi.
- Incorporar una interfaz que comunique al dispositivo y a el usuario desde cualquier lugar con acceso a Internet.
- Agregar una etapa de activación o desactivación de alimentación eléctrica vía remota.
- Estudiar métodos para determinación de confiabilidad en mediciones.

### **1.4 Justificación**

La energía eléctrica es uno de los servicios más utilizados en la vida diaria de las personas en el mundo, tan solo en el año 2016, el consumo del potencial eléctrico en el mundo, fue de 23106.8 TWh [1], además, es uno de los recursos sobre los que se ha desarrollado y en cierta medida ha permitido que la vida de las personas mejore, pero en la actualidad, el número de personas crece de manera tan acelerada que el abastecimiento del potencial eléctrico tarda en cubrir la demanda tan grande, además, se presenta un problema que no se había contemplado, el desaprovechamiento de la energía eléctrica por parte de los usuarios.

La Comisión Federal de Electricidad (CFE), es la empresa encargada de cubrir el servicio de abastecimiento de energía eléctrica en México. Su forma de cobro es a partir del uso de tarifas por cada zona del país, cada zona, posee sus propias tarifas y rangos de consumo de energía eléctrica. Existe para el consumo doméstico, una tarifa doméstica de alto consumo (tarifa DAC), la cual se aplica, si rebasas los 250 KWh (KWh, unidad de medición del consumo de energía eléctrica), en un periodo, es decir, si rebasas por 1 KWh (251KWh), el ultimo rango de la tarifa normal, pasas automáticamente a la tarifa DAC, y en esta, el monto a cubrir por el consumo realizado, puede incrementarse hasta 10 veces más con respecto a la tarifa normal.



Por ejemplo: si el consumo realizado en un periodo es de una carga de 2 focos de 100W funcionando las 24 horas durante un periodo, se tendrá un total de 144KWh consumidos, el cobro que se realizará será de **\$135 pesos**". Pues se mantiene dentro de la tarifa 1. Pero si se excede el consumo normal se aplica la tarifa DAC.

Por ejemplo: sí el consumo equivale a 4 lámparas incandescentes de 100 W conectadas 24 horas, durante todo el periodo, el consumo final de energía es igual a 288 KWh. El cobro que se realizará es **\$1486.5**". Lo cual es diez veces mayor que el precio cubierto en la tarifa 1 con la mitad del consumo.

### **1.5 Descripción del capitulado**

Capítulo 2. Se describe el marco teórico de esta propuesta, el cual hablará de: estado del arte, la energía eléctrica en México, el Internet de las cosas, los programas utilizados para el desarrollo de la propuesta, y la información de los dispositivos utilizados.

Capítulo 3. Se describe el funcionamiento del sistema a través de diagramas de flujo, y la construcción de las etapas del sistema propuesto, medición de la corriente eléctrica, la construcción del código del programa, la configuración de la plataforma en Internet, la construcción de los prototipos del sistema y del contenedor.

Capítulo 4. Se habla acerca de las pruebas realizadas y resultados obtenidos de los prototipos que se desarrollaron a lo largo de la propuesta.

Para finalizar se mencionan las conclusiones del presente trabajo, y se habla acerca de las mejoras pendientes a implementarse como trabajo futuro.



## Capítulo 2 Marco teórico y antecedentes

### 2.1 Estado del arte

Existen trabajos, que han propuesto una solución para conocer el estado del consumo eléctrico a través del uso de medidores digitales. En [2] se describe un prototipo de medidor electrónico que considera las pérdidas de energía eléctrica ocasionadas por robo de energía a través del uso de sensores de corriente, colocados en segmentos específicos del tendido eléctrico; adicionalmente, el sistema puede establecer una comunicación a través de internet con la empresa que suministra el servicio, donde envía información acerca del estado del consumo; además, se puede activar o desactivar el suministro de energía de manera remota. El sistema utiliza un microcontrolador como sistema de procesamiento y de control, una pantalla LCD para mostrar la información al usuario y un Ethernet Shield para establecer una comunicación a través de internet. El autor refiere que las pruebas realizadas mostraron valores que se pueden considerar aceptables. Una desventaja de este prototipo, es el tendido que se debe realizar para enviar la información por internet a través de una red alámbrica; además, en caso de que el usuario no cuente con una conexión a internet, él tendrá que encargarse de cubrir este servicio. Otro aspecto a resaltar, es que el usuario para poder conocer el estado del consumo, debe de ir hasta donde se encuentra el sistema para visualizar la información en la pantalla.

Un sistema que posee como objetivo brindar información acerca del estado del consumo antes de que llegue la fecha de corte, se encuentra descrito en [3], el trabajo describe el desarrollo de un sistema de monitoreo basado en un microcontrolador como dispositivo de procesamiento, un medidor de energía utilizando una bobina, un ESP-8622 como medio de comunicación inalámbrica WiFi, y una interfaz web para visualizar la información obtenida. Una desventaja que





posee este sistema, es que solamente puede medir el consumo eléctrico, no realiza control sobre la alimentación de la carga.

Por otro lado, el sistema propuesto cuenta con una interfaz de teléfono inteligente que se comunica con el módulo de monitoreo a través de Wifi, con un servidor a través de internet, el sistema puede activar o desactivar la alimentación de la energía eléctrica. Además, el usuario podrá obtener mediante una app información del consumo, estimado del monto a pagar y días restantes para la fecha de corte. Adicionalmente, la interfaz de la app permite configurar fecha de corte y montos de tarifas y estado de la carga (activada o desactivada). Las mediciones son almacenadas en una base de datos para que el usuario pueda revisar el histórico del consumo que ha realizado.

## **2.2 La energía eléctrica en México**

La energía eléctrica, es uno de los recursos más empleados en todo el mundo, dado a que su utilización ha traído mejoras significativas en la calidad de vida de las personas. Está presente en la vida diaria de las personas y facilita muchas de las labores cotidianas que llevamos a cabo, se encuentra en el transporte, en los dispositivos móviles, aparatos médicos.

La energía eléctrica, además de ser un servicio, también es un recurso, puesto que se deben realizar una serie de procesos para la generación de la energía eléctrica. Los procesos para la generación, pueden utilizar recursos renovables, como el viento o el sol, y no renovables, como los combustibles fósiles, es por ello, que se requiere de una remuneración que cubra los recursos utilizados para su generación.

En México existe una empresa gubernamental que se dedica exclusivamente a la generación, elevación, transmisión, distribución y cobro de la energía eléctrica, esa empresa es la Comisión Federal de Electricidad (CFE). Ahora bien, CFE, es la encargada de realizar un cobro monetario equivalente al consumo eléctrico realizado. Para ello, aplica un esquema dividido en tres categorías: hogar, negocio





e industria. Para el esquema hogar, se utiliza un sistema de cobro categorizado por tarifas, y cada tarifa posee sus propias cuotas. El tipo de tarifa dependerá de las condiciones y características de la zona geográfica donde se encuentre. Por ejemplo, la tarifa para la zona del desierto de Sonora, es diferente a la tarifa de San Cristóbal en Chiapas, puesto que las condiciones climatológicas, y del entorno son diferentes.

Las características para cada tarifa dependerán de la temperatura promedio mínima en verano que se tenga en la entidad, las tarifas y las temperaturas [4] están mostradas en la tabla 2.1.

Tabla 2.1. Tarifa contra temperatura

| Tarifa | Temperatura |
|--------|-------------|
| 1      | < 25 °C     |
| 1A     | 25 °C       |
| 1B     | 28 °C       |
| 1C     | 30 °C       |
| 1D     | 31 °C       |
| 1E     | 32 °C       |
| 1F     | 33 °C       |

Cada una de las tarifas tiene sus características propias, como rango de consumo y precio por cada rango de energía consumida.

Para la Ciudad de México y área metropolitana, se aplica la tarifa 1, la cual esta descrita en la tabla 2.2.

Tabla 2.2. Tarifa aplicada a CDMX

| Tarifa 1           |        |                                 |
|--------------------|--------|---------------------------------|
| Consumo básico     | \$0.81 | 0 – 75 KWh                      |
| Consumo intermedio | \$0.98 | 75 – 140 KWh                    |
| Consumo excedente  | \$2.87 | 140 KWh – en adelante hasta 250 |





Por ejemplo: si el consumo realizado en un periodo es de una carga de 2 focos de 100W funcionando las 24 horas durante un mes, se tendrá un total de 144KWh consumidos, el cobro se realizará de la siguiente manera: **“(75KWh x \$0.81)+(65KWh x \$0.98)+(4KWh x \$2.87) = \$135 pesos”**.

Ahora bien, si al final del periodo, se llegarán a rebasar los 250 KWh, el esquema de cobro de la tarifa 1 ya no se aplica; sino que se aplica un esquema llamado tarifa doméstica de alto consumo (Tarifa DAC); la cual, tiene el esquema de cobro descrito en la tabla 2.3.

Tabla 2.3. Tarifa doméstica de alto consumo.

| Tarifa DAC        | Cargo fijo | Precio x KWh     |
|-------------------|------------|------------------|
| Consumo > 250 KWh | \$107      | \$ 4.79 (\$/KWh) |

Cuando la tarifa DAC es aplicada, se debe pagar un cargo fijo mensual de \$107, además de todo lo consumido con un precio por unidad de \$4.79 por cada KWh. Si el consumo al término del mes, por ejemplo, equivale a 4 lámparas incandescentes de 100 W conectados 24 horas, durante 30 días, el consumo final de energía es igual a 288 KWh. El cobro que se realizará es el siguiente: **“\$107 + (288KWh x \$4.79) = \$1486.5”**. Lo cual es diez veces mayor que el precio cubierto en la tarifa 1 con la mitad del consumo.

### 2.3 Medidores de consumo eléctrico

Un medidor de consumo eléctrico tiene como finalidad cuantificar la energía que se ha consumido en un periodo de tiempo. Estos medidores de energía se encuentran instalados en las viviendas, empresas, industrias y en todos aquellos lugares en los que se utilice la energía eléctrica.

Las empresas que proveen el servicio del suministro eléctrico utilizan los medidores para conocer cuanta energía se ha consumido y en base a la cantidad, realizan un cobro monetario por dicho consumo.





Los medidores de energía eléctrica pueden medir conexiones monofásicas, bifásicas y trifásicas, dependerá del tipo de conexión con la que cuente el usuario, además existen diversos tipos de medidores.

### 2.3.1 Medidores analógicos.



Figura 2.1. Medidor energía eléctrica analógico.

Los medidores con diales, se caracterizan principalmente por poseer un registro mecánico (analógico), donde se almacena el valor del consumo realizado, este tipo de medidor continúa ocupándose en algunas viviendas de la ciudad de México, se caracteriza por tener indicadores circulares con agujas, las cuales muestran el valor del consumo que se ha realizado. Su funcionamiento se basa en el principio de la inducción electromagnética, donde al interactuar dos campos magnéticos se genera una fuerza entre ellos, esa fuerza es la que se encarga de mover las agujas de los diales. Cada dial, representa una unidad de cantidad y dependiendo de la ubicación del dial, será el multiplicativo, el dial de la extrema izquierda, será el de la cantidad más significativa, y el de la extrema derecha será el menos significativo.

El proceso para poder realizar una lectura del valor medido es el siguiente:

- Se debe de tomar una primera lectura del medidor. Para capturar una lectura es necesario fijarse en el sentido en el que giran las agujas, éstas, siempre irán en dirección del dígito con mayor valor.



- Si la aguja se encuentra entre dos números, siempre se toma el menor.
- Cada dial, representa un múltiplo de cantidad, de derecha a izquierda (viendo la caratula del medidor de frente), se encuentra con el primer dial, el de la extrema derecha, corresponde a las unidades, el siguiente a la izquierda corresponde a las decenas, el próximo será el de las centenas, y el siguiente será el de las unidades de millar.
- Las cifras reunidas formaran un número que representará la cantidad de energía consumida desde que el medidor se conectó (Aunque también puede haber dado varios giros sobre sí mismo, es decir llegado a 9999, y vuelve a empezar en 0). Por lo tanto, para conocer la cantidad de energía que se consumió en 1 día, lo que debemos hacer es tomar una nueva lectura después de la primera lectura, habiendo pasado 24 horas, se captura el nuevo digito y se realiza una resta, la lectura actual menos la lectura pasada, el resultado de la resta será la cantidad de energía que se consumió desde la captura de la primera lectura.

Una desventaja de los medidores con diales, es justamente que las lecturas se deben realizar por una persona que conozca el método para obtener las lecturas, pero, si por alguna razón, se llegara a medir de manera incorrecta, a causa de una equivocación o distracción, no se conocería si hubo un error, dado que no existe ningún registro donde se pueda revisar un histórico del consumo.



### 2.3.2 Medidores digitales



Figura 2.2. Medidor de energía eléctrica digital.

Este tipo de medidores, cuenta con elementos electrónicos incorporados, como registros digitales que son los encargados de almacenar la información del consumo eléctrico de manera digital, así procesar la información, mostrarla en una pantalla electrónica, o ser transmitida por algún medio de comunicación inalámbrica se vuelve una tarea más sencilla.

Una ventaja que presentan estos medidores, con respecto a sus antecesores (los medidores analógicos), es que presentan un mayor número de funciones y características que son atractivas para los usuarios y los suministradores del servicio eléctrico. Por esa razón, se ha incursionado en el desarrollado y la comercialización de medidores digitales para el consumo eléctrico, tanto, que las empresas encargadas de brindar el servicio de energía eléctrica, han invertido en la producción e instalación de estos nuevos medidores, pues ofrecen mayores funcionalidades y comodidades, algunas de las ventajas que presentan son:

- Pantalla digital en la que se muestra la cantidad de energía consumida.
- Indicador de estado del estado del consumo. – Consiste en un LED (diodo emisor de luz) que cambia de color dependiendo del estado, si está activo el servicio de suministro de energía es de un color verde.



- Terminales de comunicación especial. – Son terminales para la transmisión de información, las cuales tienen una entrada especial, por lo que solo quien cuente con la entrada correcta, podrá comunicarse con el medidor.

## 2.4 Internet de las cosas

Internet of Things (IoT, por sus siglas en inglés) o Internet de las cosas, es un acontecimiento tecnológico que marca el inicio de la nueva revolución industrial, o mejor conocida como la revolución 4.0. [5] En donde, la búsqueda de esta revolución es la de integrar la tecnología de las comunicaciones inalámbricas, el procesamiento de la información, en dispositivos de uso cotidiano, con la finalidad de que estos dispositivos generen información acerca del uso de los recursos, para que pueda ser analizada por algoritmos computacionales para la toma de decisiones en cuanto al aprovechamiento de los recursos. Esta nueva revolución plantea lo siguiente: Que los objetos comunes de la vida diaria estén conectados entre sí con otros objetos, con el fin de obtener una red de objetos interconectados para adquirir información de cada objeto y lograr automatizar, eficientar y mejorar la calidad de vida de las personas. Por ejemplo: si cosas comunes, como los productos que se mantienen dentro de un refrigerador, logran interactuar entre sí, se podría generar la información suficiente para conocer cuántos productos se encuentran dentro del refrigerador, cuántos de ellos han sido abiertos, o están por terminarse; con esa información, se podrá realizar una lista con los nombres de los productos que están por terminar, y que se agreguen a la lista de compras de la semana. Esta es la proyección y el alcance que busca obtener la revolución industrial 4.0, pues no solo se busca con refrigeradores o electrodomésticos, sino con cualquier objeto o cosa, como las partes de un automóvil, una motocicleta, las partes de un tren; prendas que tomen información del cuerpo del usuario; elementos disponibles en un almacén, etcétera. Busca incorporarse en cualquier cosa, por eso el nombre de internet de las cosas.



### 2.4.1 Origen

El origen del internet de las cosas se remonta desde la concepción del termino mismo; históricamente la definición nace a partir del entendimiento de su concepto, en el que se pretendía incorporar comunicación a los objetos, con el fin de poder obtener información acerca de los mismos, como desgaste, identificación de piezas, si están en uso, detenidos, si requieren mantenimiento, etcétera.

La creación del concepto se le atribuye a Kevin Ashton [6], cuando se encontraba trabajando para Procter & Gamble (P&G). Tenía una propuesta para ellos, en la que proponía mejorar el negocio de P&G mediante la incorporación de identificadores por radiofrecuencia (RFID), y algunos otros sensores en cada uno de los productos de la cadena comercial. Para que éstos proporcionaran información acerca de la ubicación de los productos, si ya habían sido escaneados en un almacén o colocados en un estante, o si ya habían sido adquiridos por un consumidor. La idea de Ashton era nueva, complicada por el desarrollo tecnológico de la época (antes del año 2000), y muy potente. Tanto, que después de presentar su innovadora idea al grupo P&G y al grupo Gillette, fue Gillette quien interesado por la naturaleza del proyecto lo financió, y en colaboración con el Instituto Tecnológico de Massachusetts (MIT), creó el laboratorio Auto-ID, donde se llevaron a cabo los trabajos necesarios que sirvieron como base para el Internet de las cosas. Pero fue hasta un punto entre el 2008 y el 2009, cuando el IoT tomo fuerza. En algún punto intermedio entre esos años, hubo más dispositivos conectados a internet, que personas, este acontecimiento es el parteaguas donde se considera que nace el IoT.

### 2.4.2 Aplicaciones del internet de las cosas

Para que el IoT funcione, es necesario poder contar con una manera de comunicar los dispositivos entre sí, para lograr formar una gran red de cosas interconectadas,



con el fin de poder obtener información de los dispositivos y en base a la información proporcionada poder actuar de la manera que mejor convenga. El medio que permite la comunicación de las cosas, es el Internet, una gran red de servidores que reciben y transmiten paquetes de información [7].

En la actualidad, una gran parte de la población en el mundo utiliza Internet, tan solo desde el año 2017, un 49.7% de la población mundial lo utiliza [8], casi la mitad de la población en el planeta. Por lo tanto, al menos la mitad de las personas en el planeta, tienen la posibilidad de incorporar en sus vidas el IoT.

La principal aplicación que justifica el uso y la creación del IoT, es la interacción de objetos, con el fin de obtener información para llevar a cabo procesos automatizados, sin que el ser humano este presente. Una de las características del internet de las cosas, es que cada objeto debe poseer un identificador único que los diferencie de los demás, por esa razón pensaron en utilizar identificadores de radiofrecuencia como el elemento que le diera un identificador a cada objeto. Una práctica visible de esta implementación, se encuentra en los productos disponibles en algunas tiendas de autoservicio, donde los códigos RFID son leídos por unas paletas colocadas antes de la entrada a la caja de pago, las paletas se encargan de leer todos los códigos de los productos colocados en el carrito de compras, así, cuando llegas a la caja, los productos ya se encuentran escaneados y solamente queda pagarlos. Otra situación en la que se puede presentar el IoT es, si todas las partes de un vehículo tuvieran un identificador, y contaran con la capacidad de obtener información acerca del tiempo de uso, estado del mismo y se lograrán comunicar con escáner conectado a una computadora, se podría analizar la información recibida e identificar si alguna pieza o parte se encuentra dañada, – a partir del tiempo de uso – y de esa manera, notificar que es necesario realizar un reemplazo de dicho componente. Donde, el sistema se encargaría de apartar la pieza en almacén y una cita con el mecánico para realizar el reemplazo. En el



ejemplo anterior, se menciona un proceso automático que ya no requiere interacción humana, pero si necesita contar con una conexión a Internet. Por lo tanto, en el IoT se pretende que tareas sencillas se realicen de manera automática, con el fin de brindar una mayor comodidad, y un mejor aprovechamiento de los recursos.

## 2.5 Tarjetas de desarrollo

Son tarjetas electrónicas, conformadas por circuitos integrados y componentes electrónicos de alta escala de integración (figura 2.3), que se encargan de realizar funciones específicas, y son empleadas para el desarrollo de prototipos funcionales a corto plazo.



Figura 2.3. Tarjetas de desarrollo.

En el mercado existen una gran variedad de tarjetas electrónicas con diversas características y funcionalidades; las cuales se emplean para el desarrollo de prototipos; para seleccionar la tarjeta correcta, es necesario conocer la aplicación para la que se destinará, pues eso resaltarán las características que debe tener, así se seleccionará aquella que se ajuste a las necesidades de la tarea que se llevará a cabo.

Algunas de las tarjetas disponibles en el mercado son:

### 2.5.1 Dispositivos lógicos programables

Los Dispositivos Lógicos Programables (PLD, por sus siglas en inglés *Programmable Logic Device*) son dispositivos que están compuestos por matrices



lógicas programables, son redes interconectadas de compuertas lógicas, las cuales se pueden configurar para que realicen tareas específicas, o funciones especializadas.



Figura 2.4. FPGA.

El dispositivo de mayor auge dentro de los PLD, son los FPGA (figura 2.4), *Field Programmable gate array* o matriz de compuertas programables [9]. Son muy utilizados en la industria del diseño de sistemas electrónicos, sobre éstas, se pueden configurar sistemas que le den solución a un problema planteado; la ventaja, es que una vez que finalice el diseño, se puede realizar un conteo de los elementos que utilizó la tarjeta y la configuración que tenían, con esos valores es posible crear circuitos integrados que se encarguen de realizar esa función en específico, lo que genera que el desarrollo del sistema sea más especializado, pues realiza solamente lo que se necesita, ninguna función más; el costo por el circuito integrado obtenido, es más bajo que implementar el sistema resultante en los FPGA, por lo que se puede decir, que abarata los costos de producción, pero requiere de un mayor tiempo de preparación.

### 2.5.2 Sistemas basados en microprocesadores

Un microcontrolador es un dispositivo de alta escala de integración que está conformado por un conjunto de dispositivos necesarios para la realización de una



tarea con un microprocesador. Un microprocesador es un dispositivo electrónico de alta escala de integración que es capaz de realizar funciones para las que el programador lo diseñó.

Estos sistemas tienen la característica de que pueden realizar diversas tareas para las que fueron diseñados, además tienen incorporados elementos específicos para comunicación, procesamiento, control de señales, etc.

Los dispositivos electrónicos basados en microprocesadores son los que más abundan, tan solo con las computadoras, y los celulares abarcan un gran sector de consumibles del mercado. Ahora bien, no son los únicos en el mercado, pues existe una amplia variedad de dispositivos, complejos, sencillos, eficientes, que realizan funciones específicas, e incluso soportan el procesamiento de un sistema operativo. Estos dispositivos, son las tarjetas de desarrollo, basadas en microprocesadores.

Algunas tarjetas son:

### 2.5.2.1 Raspberry Pi



Figura 2.5. Raspberry Pi.

Es una de las primeras tarjetas en ser lanzadas que posee un gran poder de cómputo, y cabe en la palma de la mano. Esta tarjeta (figura 2.5), cuenta con los elementos necesarios para usarla como una computadora, sus características son (Para la versión Pi 3 B+) [10]:



- Procesador: Quad Core 1.2GHz Broadcom BCM2837 64bit.
- RAM: 1GB DDR2.
- Puertos:
  - 4 x USB 2.0.
  - 1 x 40 GPIO pin.
  - 1 X HDMI.
  - 1 x Ethernet.
  - 1 x Combo audio/mic.
  - 1 x Interfaz de cámara (CSI).
  - 1 X Interfaz de Pantalla (DSI).
  - 1 x Micro SD.
  - Módulo Bluetooth.
  - Módulo de Wi-Fi b/g/n en la banda de 2.4GHz.

Dadas sus características, esta tarjeta ha tenido mucha fama por la cantidad de procesos y tareas que puede realizar; además, reúne todo el hardware necesario para una computadora.

### 2.5.2.2 Intel Galileo



Figura 2.6. Tarjeta Intel Galileo.

Una tarjeta lanzada por la compañía Intel, tiene como objetivo, ser una tarjeta de desarrollo de proyectos, con la característica de tener compatibilidad con módulos para funciones específicas, como WiFi, Bluetooth, almacenamiento de datos, entre otros. Sus principales características son (Para la segunda generación) [11]:

- Procesador: Intel Quark SoC X1000 400MHz.



- RAM: 256MB DDR3.
- Puertos:
  - PCI Express.
  - 3 X USB 2.0
  - Ethernet.
  - Reloj de Tiempo Real.
  - Ranura MicroSD.

El fabricante establece, que la tarjeta se comunica con el entorno de programación Arduino, y presenta compatibilidad para trabajar con él.

### 2.5.2.3 Arduino UNO

El Arduino uno; es una de las tarjetas electrónicas utilizadas por personas que quieren desarrollar algún proyecto de manera rápida; utiliza el entorno de desarrollo Arduino, un entorno amigable que es sencillo de utilizar para nuevos usuarios, además, cuenta con una gran comunidad, por lo que, información acerca del funcionamiento, propuestas de solución a problemas y soluciones es muy basta.

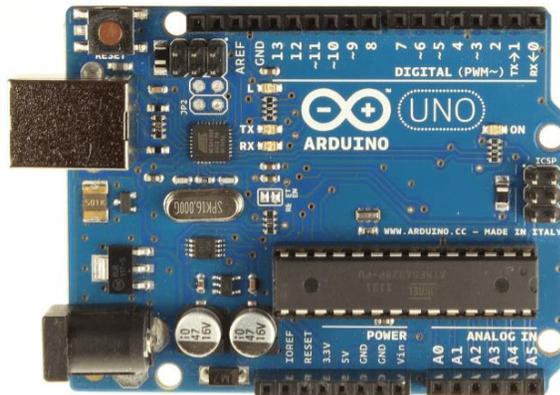


Figura 2.7. Tarjeta Arduino uno.

La tarjeta utiliza un microcontrolador AVR 328P como elemento encargado de llevar a cabo las funciones del sistema, por lo que las características de la tarjeta son las del microcontrolador [12]:



- Voltaje de operación: 1.8 - 5.5 VDC.
- Arquitectura de CPU: 8 bit AVR
- Memoria flash: 32 KB.
- Memoria RAM: 2 KB.
- EEPROM: 2 KB.
- Frecuencia de operación: 20 Mhz.
- Pines de I/O: 23
- Canales ADC: 10.
- Interfaces: UART, TWI, SPI.
- Temperatura de Operación: -40° a 85° C.

#### 2.5.2.4 Wemos, NodeMCU

Esta familia de tarjetas (figura 2.8) está basada en el microcontrolador ESP-8266 de la empresa Espressif System, la cual, reúne el hardware necesario para establecer una comunicación a través de Wifi, utilizando el protocolo de comunicación IPv4/TCP/UDP/HTTP. Lo cual hace de ellas la herramienta adecuada para desarrollar proyectos enfocados al Internet de las cosas.

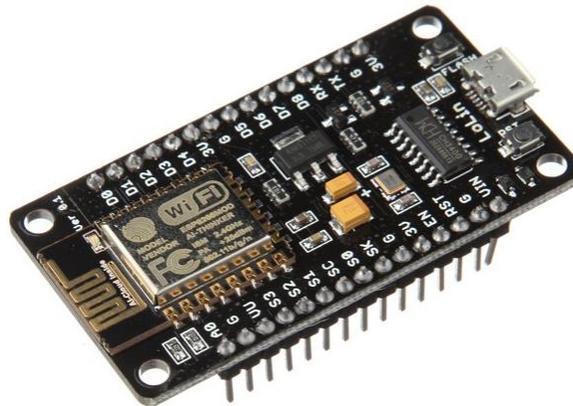


Figura 2.8. Tarjeta NodeMCU.

Las características del ESP-8266 son [13]:

- 32-bit RISC CPU: Tensilica.



- Voltaje de operación 2.5V – 3.6V.
- IEEE 802.11 b/g/n Wi-Fi.
- 16 pines GPIO.
- Protocolos de comunicación SPI e I2C.
- Convertidor Analógico-Digital (ADC) de 10 bit.
- Módulo PWM.

## **2.6 Sensores y dispositivos electrónicos.**

Los sensores son dispositivos electrónicos que cuantifican una variable física. Los sensores son utilizados para adquirir información acerca del comportamiento de algún fenómeno físico, donde, dependiendo de su comportamiento, es la acción que se puede realizar. Por ejemplo, si se desea monitorear la temperatura que presenta una máquina para que no rebase un valor en específico, se puede colocar un sensor que obtenga la temperatura de dicha máquina, y cuando la temperatura este por alcanzar el límite previsto emita una alerta, active un mecanismo de enfriamiento, etc. La respuesta dependerá de la aplicación.

Para este trabajo, se presentan algunos de ellos.



### 2.6.1 Sensor de corriente ACS-712

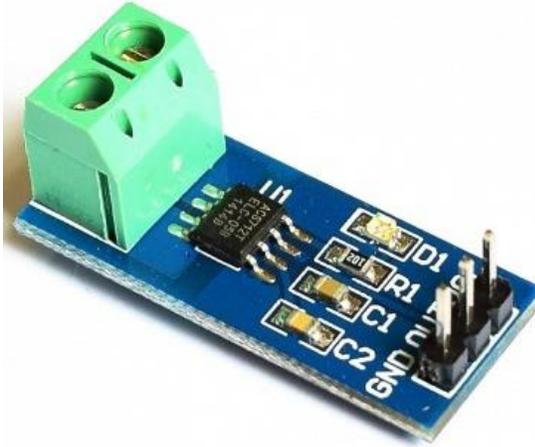


Figura 2.9. Sensor de corriente ACS - 712.

El sensor de corriente ACS-712 [14], (figura 2.9) es un dispositivo electrónico fabricado por la empresa Allegro; el cual, cuantifica la cantidad de corriente que se encuentra circulando entre sus terminales, puede cuantificar corrientes en directa y en alterna.

Es un sensor de tipo invasivo, para su funcionamiento es necesario que se conecte en serie con la carga. Su funcionamiento se basa en el efecto Hall; donde la cantidad de corriente que se encuentra circulando entre sus terminales produce un voltaje que es equivalente a la corriente en una escala de 0.5 hasta 4.5 Volts.

En la figura 2.10 se muestra la gráfica proporcionada por el fabricante, donde se muestra el comportamiento de la salida del sensor en términos de Voltaje ( $V_{IOUT}$ ) – Corriente ( $I_p$ ).

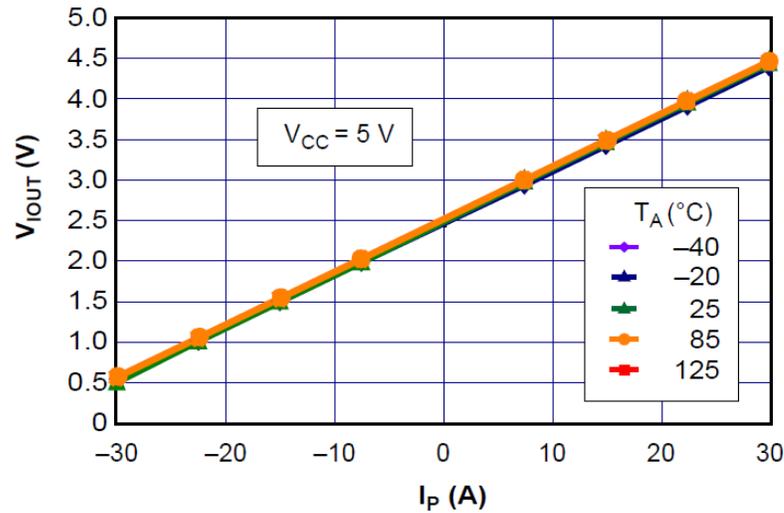


Figura 2.10. Comportamiento de la salida del sensor

Las terminales del sensor, se muestran en la figura 2.11, donde se aprecia una recomendación de implementación por parte del fabricante.

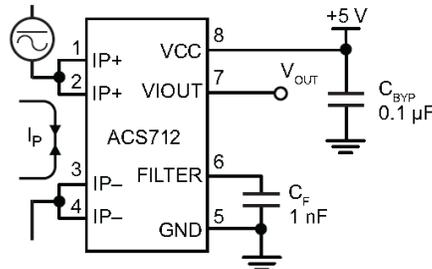


Figura 2.11. Terminales ACS-712

Las características del sensor son:

- 5V en DC de alimentación.
- Funcionamiento para corriente AC y DC hasta 30 Amper.
- Temperatura de operación – 40 °C a 85 °C.
- Salida de voltaje en directa, proporcional a la corriente cuantificada.
- Rango de voltaje en DC en la salida 0.5 – 4.5 V<sub>DC</sub>.
- Soporta hasta 3800 W



## 2.6.2 Módulo MicroSD

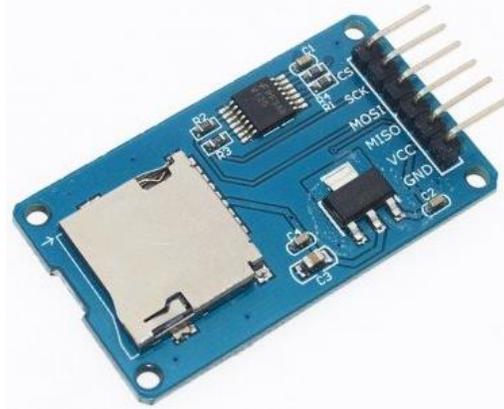


Figura 2.12. Módulo para memoria MicroSD

Es un dispositivo electrónico (figura 2.12) que permite almacenar información en una memoria de tipo MicroSD, la cual, se puede extraer del módulo, para analizar la información almacenada en ella. Este módulo permite incorporar la capacidad de almacenamiento al sistema por desarrollar, logrando almacenar en ella los resultados de operaciones, valores obtenidos de procesos, valores calculados o resultados de mediciones realizadas. El módulo MicroSD [15] se comunica mediante el protocolo Serial Peripheral Interface (SPI, por sus siglas en inglés). Lo que facilita la transferencia de información entre el módulo y el sistema. El módulo posee 6 terminales para su conexión, donde VCC y GND son terminales de alimentación.

SCK. – Es donde se coloca la entrada del pulso de reloj.

MOSI. – Terminal por la que se envía la información.

MISO. – Terminal en la que el dispositivo recibe la información proveniente del sistema de control.

CS. – Terminal identificadora, ayuda a poder distinguir entre diferentes dispositivos conectados al mismo tiempo.



### 2.6.3 Pantalla LCD



Figura 2.13. Pantalla LCD 16x2.

Una pantalla es un dispositivo electrónico que muestra información de manera visual. Existen muchos tipos de pantallas, pantallas CRT (tubo de rayos catódicos), pantallas VGA (Matriz de gráficos), pantallas led, pantallas plasma, pantallas Oled y pantallas con display de cristal líquido (Liquid Crystal Display, LCD, por sus siglas en inglés) que se encuentran disponibles en el mercado con precios accesibles, además existe una amplia variedad de información acerca de cómo utilizarlas. En las LCD, la versión más común que se puede encontrar en una tienda de electrónica es la de 16 columnas por 2 filas (16x2), la cual establece una comunicación en paralelo. En cada una de sus 32 casillas, puede mostrar una amplia gama de caracteres, entre los que se encuentran, letras, números, símbolos. etc. Las terminales que se utilizan son [16]:

- VSS. – Voltaje a surtidor, es decir, esta terminal va a tierra.
- VDD. – Voltaje de drenado, esta terminal va a 5V.
- V0. – Se usa para controlar el brillo de la pantalla (generalmente se coloca un potenciómetro de 10Komhs para variar el brillo).
- Las terminales D4, D5, D6 y D7 son las que se usan para la comunicación entre el controlador de la pantalla y la tarjeta de desarrollo.



A y K. – Son terminales de alimentación de la iluminación de fondo (+, -, respectivamente).

#### 2.6.4 Sensor PIR



Figura 2.14. Sensor PIR

El sensor infrarrojo pasivo (PIR, por sus siglas en ingles), es un dispositivo que cuantifica luz infrarroja y cuando existe una perturbación en el espectro infrarrojo, que puede ser causada por el movimiento de un objeto o cuerpo dentro del rango de visión del sensor, éste envía una señal digital por la terminal de salida [17]. El módulo mostrado en la figura 2.14, incluye el sensor de presencia PIR, además de los componentes electrónicos necesarios para trabajar con un voltaje de alimentación de 5 Volts, poder ajustar la intensidad de la radiación infrarroja con la que detecta movimiento y la duración del pulso de salida cuando existe una interrupción del espectro infrarrojo. La luz infrarroja puede ser emitida por objetos que se encuentren por encima de la temperatura del cero absoluto. Usualmente, este sensor se emplea para detectar movimiento, y la aplicación más usual, es para sistemas de alarmas.



### 2.6.5 Módulo relevador

Un relevador es un dispositivo electromecánico, funciona a través de una bobina, y un interruptor magnético, el cual se desplaza entre dos posiciones, posición de encendido, cuando la bobina esta energizada, y posición de apagado, cuando no pasa energía por la bobina, el uso más común para este dispositivo es activar o desactivar la alimentación de una carga que se encuentre conectada a él. Un relevador permite acoplar las etapas de control y de potencia a manera de que ambas se encuentren aisladas entre sí. La parte de control es la que se encarga de energizar a la bobina, y en las terminales de conexión del relé se conecta la etapa de potencia. Un relevador puede activar una carga determinada con un valor de corriente y voltaje máximo, estos parámetros dependen del tipo de relevador y fabricante.

#### Relevador SLA-05VDC-SL-A

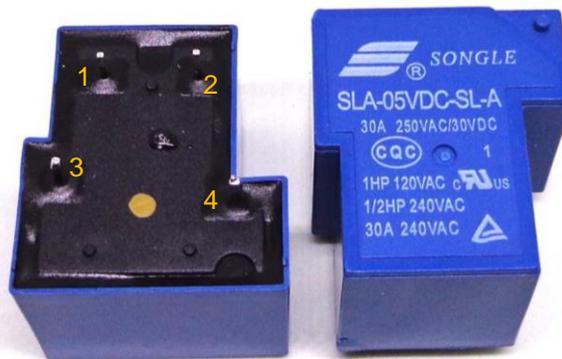


Figura 2.15 Relevador 5700 W.

El relevador mostrado en la figura 2.15, es un relevador electromecánico de la empresa SONGLE, puede soportar en sus terminales una carga de hasta 5700 W a 30 A como máximo [18], las terminales del relevador son:

1 y 2. – Corresponden a las terminales de alimentación de la bobina.



3. – Terminal de entrada, se conecta la señal que se desea activar o desactivar, comúnmente se conecta la señal de VCC.
4. – Terminal de activación o de salida, se conecta la alimentación de la carga que se desea alimentar, usualmente, corresponde a la terminal de alimentación positiva o VCC de la carga.

#### Relevador SDR-05VDC-SL-C

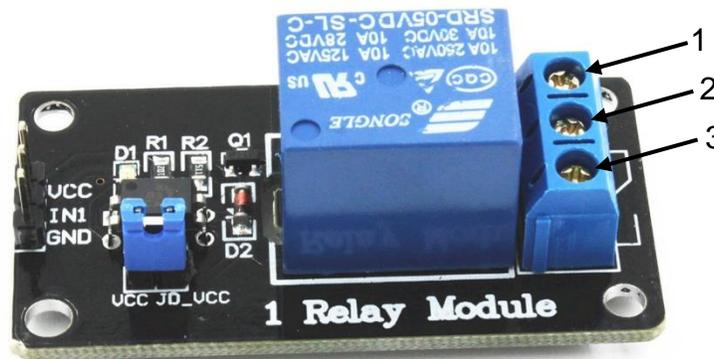


Figura 2.16. Módulo relevador.

Este módulo con relevador (figura 2.16), se alimenta con 5V y la señal de control puede estar entre un rango de voltaje entre 3.3 V – 5 V, además, cuenta con tres terminales de conexión en el lado de la carga, para poder realizar una alimentación de la carga, donde el interruptor se encuentre normalmente abierto, o normalmente cerrado. Este módulo posee la implementación de sistemas acopladores de protección, así se separan las etapas digitales y de potencia. Sus terminales son [19]:

- VCC. – Terminal donde se conecta el potencial positivo.
- IN1. – Terminal donde se conecta la señal para activar o desactivar el relé.
- GND. – Terminal donde se conecta la terminal negativa o común de la fuente.
- 1. – Terminal cerrada, cuando el relé esta desactivado, hay energía en la terminal.



2. – Terminal común, se conecta la señal necesaria para cerrar el circuito.
3. – Terminal abierta, cuando el relé esta desactivado no hay energía en la terminal.

### **2.6.6 Fuente de alimentación AC/DC**

Las fuentes de alimentación son conjuntos de componentes que se encargan de brindar un voltaje estable diferente al voltaje que se tiene en la entrada de alimentación. Una fuente AC/DC convierte el voltaje variante en el tiempo (AC) a un voltaje directo en el tiempo (DC), y la intensidad del voltaje convertido dependerá de la aplicación para la cual se destine su uso.

Las fuentes de alimentación se encuentran en la mayoría de los aparatos electrónicos, como estéreos, televisiones, grabadoras, reproductores de video, consolas de videojuego, etc. En los años 90 usualmente, las fuentes de alimentación se encontraban dentro de los aparatos, y en algunas ocasiones, el tamaño de la fuente influía directamente en el tamaño del aparato, por lo que, los aparatos eran grandes o muy voluminosos; en la actualidad, la reducción del tamaño de los componentes electrónicos ha permitido que el tamaño de algunos aparatos se haya reducido, y en algunos casos, se ha modificado el lugar de la fuente de alimentación, incluso se ha extraído y colocado por fuera. Tal es el caso de los teléfonos celulares, donde se colocó una batería como medio de almacenamiento de energía y la fuente se encuentra de manera externa.

Una fuente de alimentación muy común en la actualidad, es la que se utiliza en los cargadores para los celulares, la cual se alimenta con un voltaje de  $127 V_{AC}$ , el nivel de tensión que es suministrado por la CFE, y entrega a su salida  $5 V_{DC}$ , con los cuales se alimenta las baterías de los celulares. Además, muchos de los circuitos electrónicos digitales, funcionan a un nivel de  $5 V_{DC}$ , como los utilizados en este trabajo.

Las características para la fuente de alimentación seleccionada para este trabajo son las siguientes [20]:



- Voltaje de operación: 110 - 240 V<sub>AC</sub> a 60Hz 15 W máximo.
- Voltaje de salida: 5 V<sub>DC</sub>.
- Corriente máxima de salida: 2000 mA con una sola carga conectada.

## 2.7 Entorno de programación Arduino

Es un programa de acceso libre, cualquier persona puede descargarlo desde la página oficial y utilizarlo para la programación de dispositivos electrónicos que sean compatibles con el entorno Arduino. Posee una amplia variedad y compatibilidad con una gran cantidad de tarjetas electrónicas.

En la figura 2.17 se muestra la interfaz del entorno Arduino.



Figura 2.17. Entorno programación Arduino.

Dónde:

1. Barra de configuraciones. – En esta sección de la interfaz se muestran las pestañas con opciones y configuraciones que tiene disponible el entorno.
  - a. **Archivo.** – En esta pestaña se encuentran las opciones referentes a los archivos del entorno, se pueden crear, abrir, cerrar, guardar, imprimir los archivos, y en preferencias se encuentra la configuración



referente al archivo, la dirección donde se almacenan, la apariencia del entorno, etc.

- b. **Editar.** – Se encuentran las herramientas necesarias para la edición del texto escrito, como comentar, aumentar o disminuir la sangría, etc.
- c. **Programa.** – Aquí se encuentran las opciones referentes al programa que se está creando, como la compilación del código, la carga del programa a la tarjeta electrónica de destino, incluir bibliotecas para el funcionamiento de sensores o de alguna función especial para el programa.
- d. **Herramientas.** – Se encuentran configuraciones del entorno, y del programa, como el tipo de tarjeta sobre la que se va a programar, en que puerto serial se encuentra conectada, también se puede administrar las bibliotecas que se encuentran cargadas en el entorno, se puede inicializar el monitor serial, que es una pantalla que muestra lo que está transmitiendo y recibiendo por el puerto serial donde se encuentra conectada la tarjeta, se pueden modificar las opciones de como grabar el programa en la tarjeta destino.
- e. **Ayuda.** – Se encuentran opciones que describen soluciones a algunas problemáticas con el entorno, en referencias se encuentra una descripción de los componentes del entorno, como funciona, cuales son las tareas que puede realizar el entorno, etc.

2. Se muestran botones de acceso rápido para funciones específicas del entorno.

- a. . – Botón de compilación del programa, al presionarlo, se realiza una revisión del código del programa y se emite el resultado de la revisión en el área de mensajes.



- b. . – Botón de carga, al seleccionarlo, comienza una compilación del código del programa, si no presenta errores, se carga el programa en la tarjeta de destino.
  - c. . – Botón nuevo documento, al presionarlo se abre una nueva ventana con un documento en blanco donde se puede escribir el código para un nuevo programa.
  - d. . – Botón abrir, al hacer clic en él, se abre una ventana donde se muestran los programas que se han realizado.
  - e. . – Botón guardado, sirve para salvar los avances y cambios realizados en el código del programa.
  - f. . – Botón puerto serial, el presionarlo abre el monitor serial, comunicando con el puerto en el que se encuentra conectada la tarjeta destino.
3. Zona de trabajo. – En esta sección de la interfaz se escribe el código para el funcionamiento del programa. Existen dos funciones que se encuentran cargadas al abrir un nuevo archivo, son las funciones que se ejecutaran siempre en todo programa hecho en Arduino, si una de las funciones no se encuentra descrita en el entorno, el compilador del programa emitirá un mensaje de error y no se podrá cargar el programa a la tarjeta de destino. Las funciones son:
- a. **Void Setup()**. – Es la primera función que se encuentra en zona de trabajo, en ésta se suelen colocar las funciones o tareas, que solo se ejecutaran una vez, por lo que en esta función se escriben configuraciones iniciales, como las de puertos de entrada o salida, inicializaciones de pantalla, transmisión serial, etc.
  - b. **Void Loop()**. – Es la segunda función que aparece en la zona de trabajo, esta función es en la que se coloca el código que se ejecutará



cíclicamente; por lo que, en esta función se encuentran las tareas principales que realizará el programa.

- c. **Zona en blanco.** – En esta sección, que se encuentra fuera de las funciones se colocan las bibliotecas y variables globales que utilizará el programa, por lo general, se escriben encima de la función void setup().
4. Zona de mensajes del entorno. – En esta sección de la interfaz se muestran los mensajes del entorno, que pueden ser avisos del resultado de la compilación del programa, información acerca del consumo de los recursos de la tarjeta de destino una vez que se compilo y cargo correctamente el programa.
5. Información de la tarjeta. – En esta parte de la interfaz se muestra la información de la tarjeta destino, nombre de la tarjeta y puerto en el que se encuentra conectada.

### **2.7.1 Importar bibliotecas**

Una biblioteca, es un conjunto de archivos que contienen descrito las instrucciones necesarias para realizar un proceso en específico. Son utilizadas para desarrollar tareas como, la utilización de un sensor, la configuración para el uso de pantallas, protocolos de comunicación, etc. Algunas de las bibliotecas básicas disponibles para la utilización de ciertos dispositivos, están disponibles desde la página oficial del entorno de desarrollo Arduino. Es necesario descargar la biblioteca y realizar una serie de pasos para cargarla al entorno de programación.

Para poder cargar la biblioteca, el primer paso es buscarla, para posteriormente agregarla a las bibliotecas disponibles en el entorno de programación. El formato del archivo de descarga, usualmente se puede encontrar en formato .zip un formato para archivos comprimidos. Una vez que se cuenta con la biblioteca, es necesario,



abrir el entorno, y en la barra de configuraciones, en la casilla de programa, existe una opción que dice, incluir biblioteca (figura 2.18).

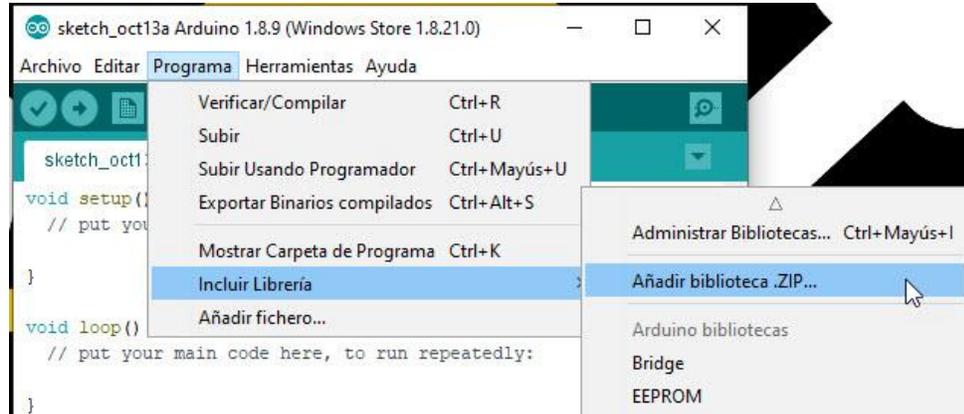


Figura 2.18. Cargar bibliotecas entorno Arduino.

Existen dos opciones, la primera, en una casilla dice administrar bibliotecas, esa opción, ejecuta el administrador de bibliotecas con el que cuenta el entorno, y en una nueva ventana (figura 2.19) aparecen las bibliotecas disponibles que se pueden cargar al entorno, se puede realizar una búsqueda, para hallar la biblioteca, una vez hallada, se selecciona y se da clic en la opción instalar.

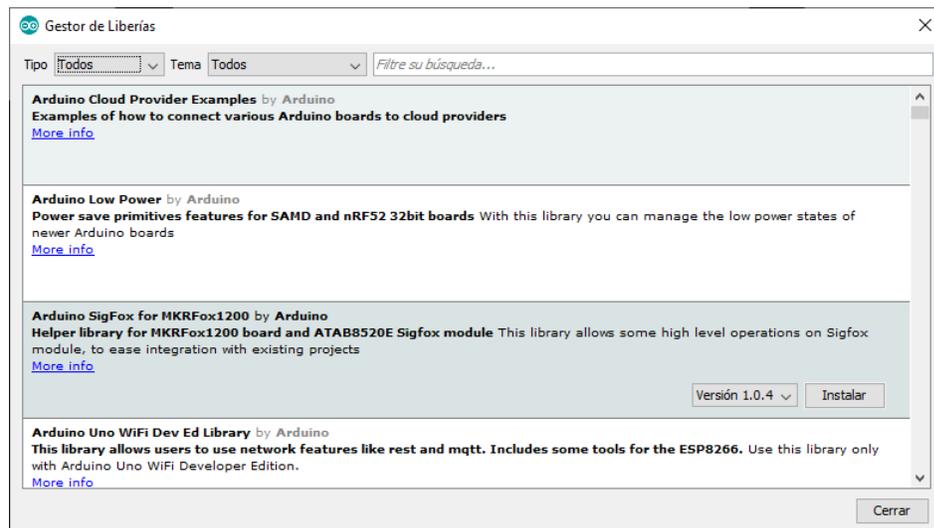


Figura 2.19. Administrador de bibliotecas.





La segunda opción, la segunda casilla de la opción de instalar bibliotecas, dice, añadir biblioteca .zip (para la cual, el archivo debe estar descargado en alguna parte del equipo), abre una ventana del explorador de archivos, donde se debe seleccionar el archivo, para ello se busca entre los archivos del equipo y se selecciona el archivo comprimido de la biblioteca. Nota importante, el archivo debe encontrarse en formato .zip para que la herramienta de añadir biblioteca funcione.

En el caso de que no se cuente con la librería en formato .zip, sino en un formato diferente, .rar por ejemplo, es necesario hacer la instalación de la biblioteca de manera manual. Para ello, es necesario descomprimir la biblioteca, y la carpeta que contiene los archivos de la biblioteca se debe de colocar en la carpeta de librerías del entorno de programación, para llegar a la carpeta, es necesario ir a archivos de programa, en la unidad C, se busca la carpeta con nombre Arduino, se hace clic en ella, y en la carpeta libraries se coloca la carpeta descomprimida.

### 2.7.2 Biblioteca WiFiManager

La biblioteca WiFiManager, está desarrollada para las tarjetas basadas en el microcontrolador ESP8266, pues permite configurar dos modos de operación de la tarjeta, en modo punto de acceso (acces point) y en modo estación (station). Con el objetivo de poder ingresar las credenciales de una red con internet, para que la tarjeta pueda vincularse con el modem, y así tener una salida de comunicación hacia internet. Además, esta biblioteca permite configurar más parámetros a la tarjeta, como las credenciales para que acceda a una cuenta de correo, a una plataforma en internet, etc. Estos parámetros pueden almacenarse en la memoria no volátil de la tarjeta.

**En modo punto de acceso.** – La tarjeta configura una red inalámbrica local – sin salida a Internet – con nombre de AutoConnectAP, la cual aparece visible para los dispositivos cercanos. La red tiene como objetivo, poder establecer una



comunicación con un dispositivo cliente, para que mediante un formulario HTML, que es visible desde cualquier navegador, se puedan ingresar los valores de las credenciales. Para ello, una vez que se ha realizado la conexión a la red AutoConnectAP, es necesario ingresar la dirección 192.168.4.1 en cualquier navegador, una vez que se ingresa a la dirección, se muestra un formulario en donde se pide el ingreso del nombre de la red, y la contraseña de la red a la que se desea conectar la tarjeta. Una vez que los datos son ingresados, la tarjeta realiza una prueba de comunicación con la red, si la conexión se ha realizado de manera exitosa, cambia el modo punto de acceso, a modo estación, donde es posible realizar solicitudes como cliente. Si los valores no son los correctos, la tarjeta se queda funcionando en modo punto de acceso.

**En modo estación.** – La tarjeta puede realizar solicitudes como cliente, por lo que es posible conectarse a una dirección web o IP, y realizar solicitudes, llenar formularios, enviar y/o recibir información. En este modo de funcionamiento, se lleva a cabo la ejecución del programa que se encuentra descrito en la sección de loop() de la tarjeta.

### 2.7.3 Biblioteca CayenneMQTTESP8266

Esta librería, permite establecer una comunicación entre una tarjeta basada en el ESP8266 y la plataforma Cayenne. La biblioteca administra las credenciales que son necesarias para establecer una conexión con la plataforma en internet. Los valores que deben de ser ingresados se obtienen desde la plataforma en línea, son tres, *nombre de usuario*, *contraseña*, *id del cliente*, estos valores se cargan en el programa de la tarjeta, una vez que la plataforma identifica las credenciales, permite el acceso a la plataforma y la información que es transmitida entre la tarjeta y el entorno puede ser mostrada en los diferentes widgets con los que cuenta la plataforma.



## 2.8 Circuitos impresos

Un circuito impreso (*Printed Circuit Board, PCB*, por sus siglas en inglés) es el diseño de las conexiones de los elementos de un circuito en una tarjeta fenólica, que por uno a ambos lados, posee una capa de cobre, que funciona como la superficie de contacto y conexiones entre componentes. Para poder realizar una tarjeta de circuito impreso, es necesario tener el diagrama esquemático (diseño de las conexiones entre los componentes), una tarjeta fenólica y un método de transferencia del diseño del circuito, se puede realizar el diseño del circuito a mano alzada, utilizando solamente reglas, escuadras y plumones permanentes, también se puede hacer uso de los softwares de diseño de PCB, los cuales facilitan la elaboración de las tarjetas fenólicas.

Una vez que se cuenta con el diseño del circuito en el programa de diseño de circuitos impresos, es necesario transferir el diseño de la tarjeta de circuito impreso a la placa fenólica, para ello, existen varios métodos, de transferencia por planchado y grabado con una maquina CNC (Control Numérico Computarizado).

### 2.8.1 Transferencia por planchado

La transferencia por planchado es uno de los métodos más utilizados para la elaboración de tarjetas de circuito impreso, donde el proceso consiste en transferir el diseño de la PCB a una tarjeta fenólica, a través de tinta tóner, la cual, al hacer presión y calentarse se adhiere a las superficies

La transferencia del diseño del circuito a la tarjeta fenólica se realiza a través de la impresión del diseño del circuito impreso, en un papel específico. La impresión debe realizarse en una impresora láser, por la tinta en polvo, o mejor conocida como tóner, pues esta sustancia una vez que se calienta se puede desprender de la superficie sobre la que se imprimió. Con respecto al papel, existen varios tipos de papel que pueden ser utilizados, como el papel couche, el acetato y papel transfer.



Los papeles couche y transfer se pueden encontrar en una papelería, y el papel transfer es vendido usualmente en las tiendas donde venden dispositivos electrónicos.

Una vez que se cuenta con el papel, lo siguiente es realizar la impresión del diseño de la PCB en el papel utilizando una impresora láser. Ya que se cuenta con la impresión sobre el papel, es necesario recortar el área que ocupara el espacio del circuito, que debe coincidir con el tamaño de la tarjeta fenólica. Para realizar la transferencia debe aplicarse calor mediante el contacto, además, se debe aplicar un poco de fuerza con el calor, para que la transferencia se logre. Para transferir el calor, se ocupa una plancha, la cual se coloca sobre el papel con la cara impresa viendo hacia el cobre de la tarjeta fenólica. Y la cara no impresa entra en contacto con la superficie de la plancha. Se debe de transferir el calor de manera constante, presionando el papel y la tarjeta con la plancha, y transcurridos 10 minutos se puede retirar la plancha para conocer el estado de la transferencia, el papel transfer y el acetato pueden levantarse de una esquina para comprobar si la tinta fue transferida, de no ser así, se vuelve a aplicar calor. Y con el papel couche, este se adhiere a la tarjeta, por lo que si no se puede despegar, quiere decir que la transferencia está hecha, solo que es necesario sumergir la tarjeta en agua, para que se pueda retirar el papel.

### 2.8.2 Eagle

Es un software de diseño asistido por computadora que se enfoca en el diseño de circuitos impresos o PCB (*Printed Circuit Boards*). Es un programa de uso comercial, por lo que hay que pagar una licencia para su uso. Existe una versión de prueba gratuita, la cual presenta restricciones con respecto a la versión con licencia, pero la versión de prueba es utilizable para proyectos que no requieran tamaños superiores a 80 cm<sup>2</sup> [21], con un máximo de hasta dos capas.



Los programas de dibujo asistidos por computadora para la creación de PCB, utilizan el diseño de un diagrama esquemático, el cual muestra las conexiones que existen entre los elementos electrónicos que conforman el circuito, como base para la creación de la tarjeta de circuito impreso. Por lo que el programa ofrece las herramientas necesarias para la creación del diagrama esquemático, para posteriormente utilizar el diagrama para el diseño y creación del circuito impreso.

En la figura 2.20, se muestra el panel de control de Eagle, en el cual se pueden visualizar las librerías añadidas al programa, las reglas de diseño, y los proyectos almacenados.

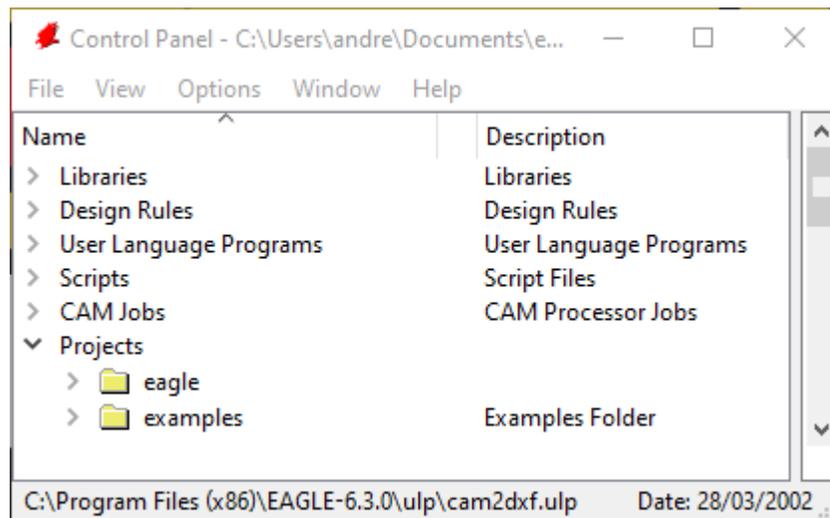


Figura 2.20. Panel de control Eagle

Para comenzar un nuevo proyecto es necesario hacer clic en la pestaña de *File*, en la opción de *New*, se selecciona *Project*. En la pestaña de *projects*, en el panel de control, aparecerá una nueva carpeta que lleva el nombre *new project*, es la carpeta que acabamos de crear, para cambiar el nombre de la carpeta se presiona el botón secundario encima de la carpeta, y en la ventana que aparece se selecciona la opción de *rename*, se borra el nombre actual y se escribe el nuevo nombre. Para comenzar un nuevo diagrama esquemático se selecciona la carpeta del nuevo





proyecto, y sobre ella se presiona el botón secundario del ratón y en el menú que aparece se selecciona la opción de *new, schematic*, y en la carpeta aparecerá el nuevo diagrama esquemático. Se abrirá una nueva ventana (figura 2.21) con el lienzo en donde se realizará el diagrama esquemático.

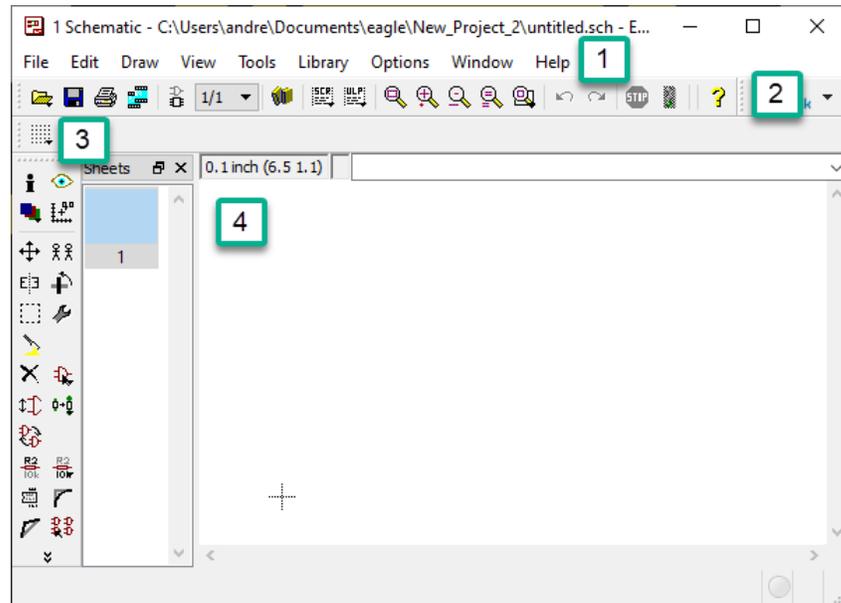


Figura 2.21. Ventana diseño diagrama esquemático.

Dónde:

En 1. – Se muestran la barra de configuración del programa, en ella se encuentran los ajustes del programa.

- *File.* – En esta pestaña se encuentran las funciones referentes al archivo, como abrir, guardar, imprimir, cambiar de ventana, importar o exportar el archivo, etc.
- *Edit.* – En esta pestaña se muestran las funciones para editar los componentes que se encuentran en la zona de trabajo. Como mover, girar, agregar, cambiar, eliminar, etc.



- *Draw.* – Se encuentran las funciones de dibujo de las conexiones entre los componentes y de la tarjeta, como colocar etiquetas de texto a los elementos, las dimensiones de los mismos, etc.
- *View.* – Se muestran las funciones relacionadas a la vista del diagrama, como es la vista de la tarjeta, si se ve de manera completa, o una región en específico de la misma.
- *Tools.* – Se muestran las herramientas de análisis del programa, como la comprobación de que no existan pistas de alimentación y tierra conectadas de manera directa, además, se analiza que se respeten las reglas de diseño.
- *Library.* – Se muestran las funciones para la utilización de las bibliotecas de los componentes, como abrir, usar y actualizar.
- *Options.* – Se muestran las configuraciones de la ventana de trabajo, como los controles y funciones de teclas que posee el programa.
- *Window.* – Muestra las ventanas actualmente abiertas del proyecto.
- *Help.* – Se muestra información relacionada al funcionamiento del programa.

En 2. – Es la barra de iconos de acceso rápido, en ella se muestran iconos de funciones del programa.

- . – Realizan un cambio en la vista de la zona de trabajo. Como una ampliación de la vista sobre un área en específico de la zona de trabajo, como centrar el diseño de la tarjeta en la zona de trabajo.
- . – Funciones para abrir, salvar, imprimir el archivo.
- . – Exporta el diagrama esquemático para el diseño de la PCB.
- . – Abre el menú para el uso de bibliotecas.

En 3. – En este panel se muestra la paleta con las herramientas disponibles para la creación y edición del diagrama esquemático.

- . – Agrega componentes a la zona de trabajo.



- . – Agrupa los componentes en un grupo.
- . – Permite mover los componentes por la zona de trabajo.
- . – Duplica un componente de la zona de trabajo.
- . – Elimina el componente seleccionado de la zona de trabajo.
- . – Herramienta para la conexión de terminales de los componentes.
- . – Coloca un indicador de conexión para 2 o más elementos.
- . – Muestra las conexiones de un elemento, pista o componente.
- . – Genera un nuevo doblez en la línea que conecta a dos elementos.
- . – Modifica el ángulo recto de una línea de conexión.
- . – Inserta texto.
- . – Herramienta para trazar un polígono.
- . – Muestra la dimensión que posee un componente.

En 4. – Se muestra la zona de trabajo, es el espacio destinado para la creación del diagrama esquemático. En esta área se colocan los componentes que formaran el circuito.

Por lo que, para comenzar a dibujar el circuito, es necesario colocar los elementos que formaran el circuito, posteriormente, se realizaran las conexiones entre las terminales de los componentes. Una vez que el diagrama esquemático está finalizado, se utiliza para crear el archivo del circuito impreso, para ello es necesario presionar el botón  que se encuentra en la barra de iconos de acceso rápido.

El programa genera de manera automática un nuevo archivo con los elementos y conexiones del diagrama. Se abre una nueva ventana (figura 2.22) donde se realizara el PCB.

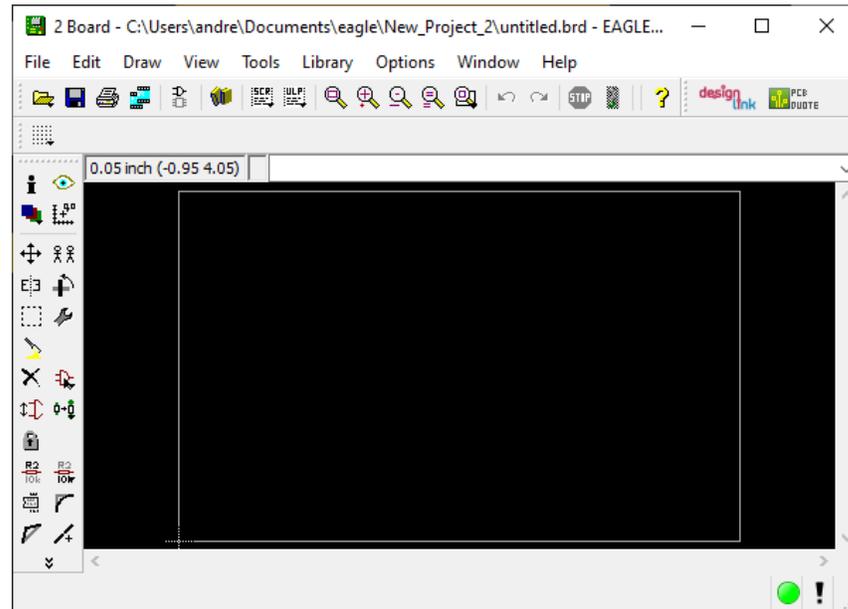


Figura 2.22. Ventana diseño PCB

En la figura 2.22 se puede apreciar que la nueva ventana es muy parecida a la ventana de la figura 2.21. Puesto que la barra de configuraciones, iconos de acceso rápido, y el panel de herramientas, son casi los mismos, cambia la configuración de la zona de trabajo, aparecen nuevos elementos en el panel de herramientas y los iconos que se muestran en la zona de trabajo, son diferentes a los mostrados en la ventana del diagrama esquemático.

Algunos de los nuevos elementos que aparecen en la paleta de herramientas son:

- . – Herramienta que ayuda en la creación de tierras.
- . – Ejecuta el asistente para comenzar con el trazado de pistas automático.
- . – Analiza que se cumpla con las reglas eléctricas en busca de fallas.
- . – Ajusta las reglas de diseño del PCB, como distancias, tamaños, tolerancias entre pistas, tierras, pads, etc.
- . – Muestra los errores encontrados en el PCB.



Los cuales representan funciones enfocadas al diseño de la tarjeta de circuito impreso.

### 2.8.2 Relación entre la dimensión de una pista con la cantidad de corriente

Uno de los aspectos importantes que se debe tomar en cuenta a la hora de crear una tarjeta de circuito impreso, es el tamaño de las pistas, donde, el tamaño es proporcional a la corriente y al voltaje que utiliza el circuito. A continuación, se describen las consideraciones que se deben contemplar.

Antes de pasar a los tamaños de las pistas, es necesario conocer algunas características de la placa fenólica que se va a utilizar.

Existen diferentes tipos de placas fenólicas, donde, la principal diferencia, es la cantidad de cobre con la que se manufactura la tarjeta (el grosor de la hoja de cobre). Existen tres tipos de grosores de la capa de cobre:

- 35 micras de cobre = 1 onza/pie<sup>2</sup>.
- 70 micras de cobre = 2 onza/pie<sup>2</sup>.
- 105 micras de cobre = 3 onza/pie<sup>2</sup>.

El grosor de la capa, ayudará a definir el tamaño y anchura de las pistas.

Ahora bien, existe una expresión matemática que describe la relación que debe existir entre el tamaño de la pista con respecto a la cantidad de corriente, la expresión es:

$$Ancho = \frac{\left(\frac{I}{k1 \times \Delta T^{k2}}\right)^{1/k3}}{L * 1.378} \quad (\text{ec. 2.1})$$

Donde:

I. – Es el valor de la corriente máxima en Amperes.

ΔT. – Es el incremento de temperatura máxima expresada en °C.



L. –Es el grosor de la capa de cobre de la tarjeta fenólica, está expresado en onza/pie<sup>2</sup> (oz/ft<sup>2</sup>).

K1. – Corresponde a una constante, definida por la ubicación de la pista, interior = 0.015, exterior = 0.0647.

K2. – Corresponde a una constante, definida por la ubicación de la pista, interior = 0.5453, exterior = 0.4281.

K3. – Corresponde a una constante, definida por la ubicación de la pista, interior = 0.7349, exterior = 0.6732.

El resultado obtenido esta expresado en mil (milésimas de pulgada), (donde, 1mm = 2.54 mil).

Así, para conocer el tamaño que debe de tener la pista, solo se deben sustituir los valores en la ecuación 1.1.

Ahora bien, la mayor parte de las tarjetas fenólicas disponibles en el mercado son de 35 micras. También, se puede ingresar a un sitio web<sup>1</sup>, en el que se encuentra disponible una calculadora para obtener el grosor de las pistas.

Ahora bien, las constantes  $K_n$ , fueron obtenidas a partir de un estándar creado para el diseño y manufactura de circuitos impresos, el estándar es el ANSI-IPC-2221, el cual establece las características necesarias que debe contener y respetar un circuito impreso. En el capítulo seis del estándar, apartado dos, se encuentra el tema de requerimientos de los materiales conductivos, donde establece la relación de distribución de la corriente, tiene que ver con tamaños de la pista, el incremento de la temperatura, y las tolerancias entre pistas.

---

<sup>1</sup> <http://circuitcalculator.com/wordpress/2006/01/31/pcb-trace-width-calculator/>



El incremento de la temperatura está definido como la diferencia entre la temperatura máxima segura que alcanza la tarjeta impresa y la temperatura segura que debe alcanzar la pista del circuito.

## 2.9 Diseño e impresión en tres dimensiones

Para el desarrollo de esta propuesta, se utilizaron programas de diseño en tres dimensiones para generar los modelos de los contenedores para los prototipos desarrollados. Para ello, también fue necesario utilizar un programa que permita “rebanar” el modelo en capas, para poder imprimirlo con la utilización de una impresora de tres dimensiones. A continuación, se describen los programas utilizados para esta parte del desarrollo de la presente propuesta.

### 2.9.1 SolidWorks



Figura 2.23. SolidWorks.

Solidworks es un programa de dibujo asistido por computadora, el cual permite la elaboración de modelos en dos y tres dimensiones. Es un programa propietario, por lo que es necesario contar con la licencia para poder utilizarlo. Cuenta con herramientas que facilitan el diseño de sólidos en tres dimensiones, por lo que la complejidad dependerá del diseño que se desee realizar.



Para comenzar a utilizar SolidWorks versión 2015 es necesario tenerlo instalado en un equipo de cómputo, los requerimientos con los que debe cumplir el equipo son [22]:

- Sistemas operativos: Windows 7 x 64 bits/8.0 x 64 bits/ 8.1 x 64 bits/ 10 x 64 bits.
- Hardware: RAM 8Gb/ 3Gb espacio en disco/ Procesador Intel o AMD compatible con Streaming Single Multiple Data Extensions 2 (SSE2).

Una vez que se cuenta con el programa instalado, se ejecuta y se muestra la interfaz del programa (figura 2.24).



Figura 2.24. Interfaz Solidworks.

Donde:

1. Barra de iconos de acceso rápido.
  - a. . – Botón nuevos archivos. Al presionarlo se ejecuta el asistente para la creación de un nuevo archivo.



- b. . – Botón abrir archivos. Abre una ventana en la que se muestran los últimos archivos guardados.
- c. . – Botón guardar. Se guardan los cambios realizados en el archivo con el que se está trabajando.
- d. . – Botón imprimir. Se ejecuta el asistente para imprimir la vista del diseño actual, impresión en dos dimensiones.

Los demás iconos se encuentran desactivados porque no hay un archivo en uso.

2. Barra de recursos. En esta barra, se muestran botones que contienen funciones y herramientas del programa.

- a. . – Botón para comenzar. Al presionarlo se muestran algunas funciones que son necesarias para comenzar a utilizar el programa, como, la creación de nuevos documentos, abrir documentos, tutoriales, algunas noticias acerca de nuevas funcionalidades, una guía introductoria al uso del programa e información general del programa.
- b. . – Botón bibliotecas de diseño. Al seleccionarlo se muestran las bibliotecas de diseño con las que cuenta el programa.
- c. . – Botón explorador de archivos. Se muestran los archivos disponibles en el equipo y en el entorno de programación.
- d. . – Botón de apariencias, escenas y calcomanías. Se muestran las apariencias de los objetos con los que cuenta el programa, se pueden visualizar objetos en diferentes apariencias, como madera, plástico, metal, vidrio, entre otros.
- e. . - Botón propiedades personalizadas. Permite crear y editar páginas de propiedades para archivos de una pieza. de modelos de apariencias, vistas y etiquetas que se pueden aplicar a los archivos.



3. Es la zona de trabajo, cuando no existe un documento sobre el que se esté trabajando, se muestra la pantalla en gris.

Para comenzar a trabajar con el programa es necesario crear un nuevo archivo, al presionar el botón de nuevo archivo, se ejecuta un asistente para crear nuevos archivos (figura 2.25).

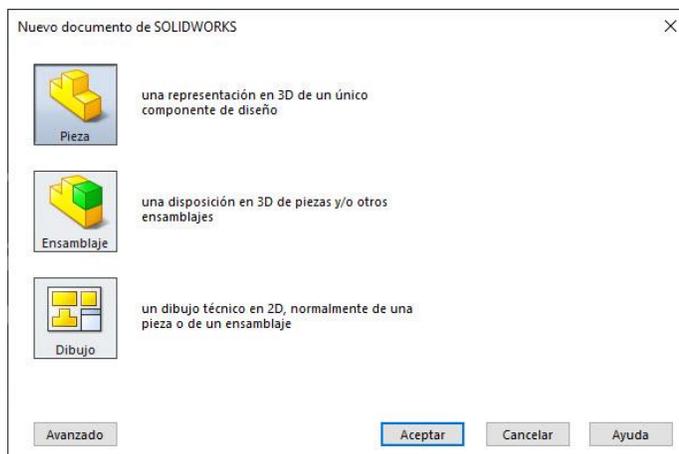


Figura 2.25. Crear nuevo documento.

Para comenzar a diseñar modelos en tres dimensiones, es necesario crear una pieza, pues es la representación de un único objeto; por lo tanto, se selecciona la casilla de pieza, se da clic en aceptar. Se crea el nuevo archivo y se muestran herramientas y funciones en el área de trabajo de la interfaz (figura 2.26). Las nuevas funciones están relacionadas a herramientas para la creación y modificación del objeto que se desea diseñar.

Donde:

1. Panel de herramientas. – En esta sección de la interfaz se muestran las herramientas disponibles con respecto a la categoría seleccionada en la pestaña de herramientas.
2. Pestaña de herramientas. – Se muestran las categorías que contienen los diferentes tipos de herramientas.



- a. Operaciones. – En esta categoría, se encuentran las herramientas de diseño con aplicación en tres dimensiones, por ejemplo:
  - i. Extruir saliente/base. – Esta herramienta permite proyectar una forma definida desde algún plano del dibujo.
  - ii. Revolución de saliente/base. – Esta herramienta, toma una forma definida sobre algún plano y lo gira sobre un eje.
  - iii. Extruir corte. – Esta es una herramienta de corte, toma la forma definida sobre algún plano y realiza un corte con la silueta de la forma.
  - iv. Redondeo. – Esta herramienta permite modificar esquinas rectas del objeto, dejándolas curvas o redondeadas.
  - v. Geometría de referencia. – Esta herramienta permite realizar un plano con respecto a una de las caras o superficies de la pieza.
  
- b. Croquis. – Esta categoría contiene las herramientas para trazar figuras o formas en dos dimensiones. Estas herramientas son las que se utilizan para trazar los croquis que sirven como base para la creación tridimensional de la pieza. Algunas de las herramientas son:
  - i. Croquis. – Esta herramienta es como una hoja en blanco donde se puede trazar un dibujo, línea, forma o figura, es la herramienta básica para crear una forma.
  - ii. Cota inteligente. – Esta herramienta muestra el tamaño que posee el dibujo, figura o forma con respecto a una dimensión.
  - iii. Herramientas para el trazado de figuras. – En esta sección de las herramientas, se muestran aquellas para poder realizar el trazado de un dibujo, a través de trazar figuras geométricas.
  - iv. Recortar entidades. - Es una herramienta que realiza un corte sobre una forma o silueta definida.



- v. Equidistanciar entidades. – Permite obtener un duplicado de una forma o silueta definida, con mayor o menor tamaño, a través de una relación entre los perímetros que es un valor en distancia entre ellos.
  - c. Calcular. – En esta categoría se encuentran las herramientas para simulaciones de tensión, fuerza, flujo y torsión en el material o diseño realizado.
  - d. DimXpert. – En esta categoría se encuentran las herramientas para colocar acotaciones y tolerancias de la pieza realizada.
  - e. Complementos de Solidworks. – En esta pestaña se muestran los iconos de las diferentes aplicaciones con las que cuenta el programa, donde cada aplicación tiene una utilidad específica.
3. Barra de iconos visuales rápida. – En esta barra de iconos se muestran herramientas que ayudan de manera visual a mover, desplazar, centrar, etc. la pieza.
    - a. . – Sirve para centrar la vista de la pieza en la pantalla de trabajo.
    - b. . – Realiza una ampliación sobre la zona seleccionada.
    - c. . – Regresa a la penúltima vista en la que se tenía colocada la pieza.
    - d. . – Se acomoda la pieza a manera de que se muestre la vista frontal de la pieza, también se puede modificar el tipo de vista, puede ser lateral izquierda, lateral derecha, posterior etc.
  4. Panel de propiedades de la pieza. En este panel se muestran las propiedades y las operaciones realizadas para la elaboración de la pieza, los croquis, extrusiones, cortes y las relaciones entre ellos. Además, se muestran las acciones realizadas con respecto a las vistas que posee el entorno.

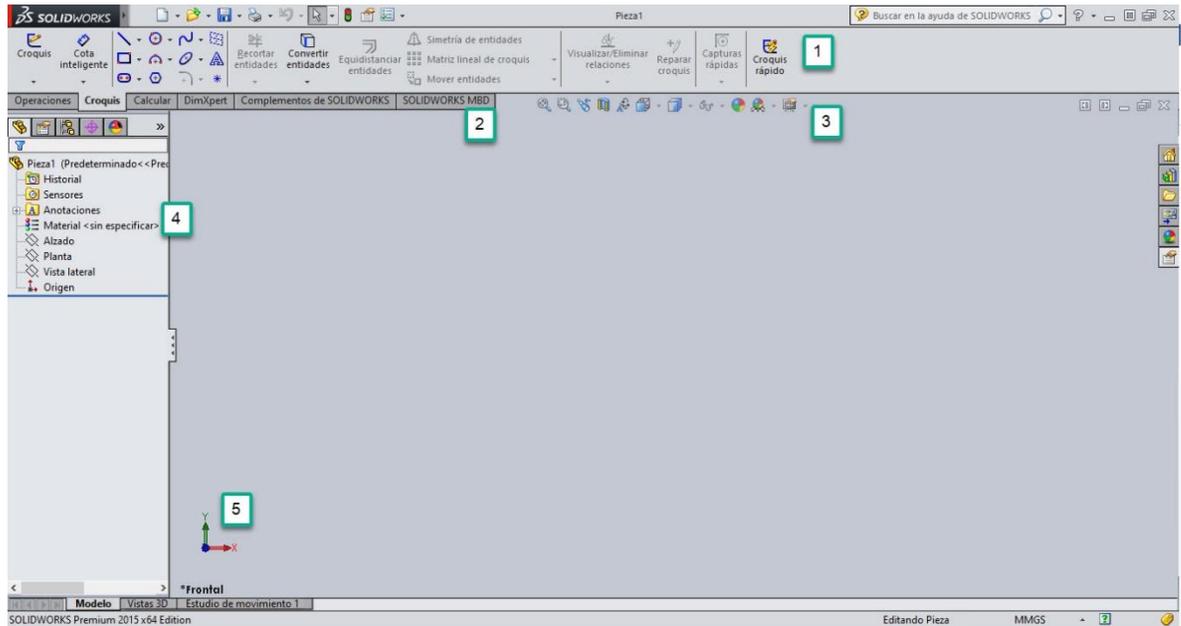


Figura 2.26. Ventana archivo activo SolidWorks.

## 2.9.2 Ultimaker Cura



Figura 2.27. Ultimaker Cura

Es un programa que procesa modelos en tres dimensiones para reestructurarlos mediante la descomposición por capas, para que mediante la deposición fundida aditiva de filamento se forme un modelo físico en tres dimensiones. Se utiliza para procesar un archivo en formato stl de un modelo en 3D, para traducirlo a indicaciones que seguirá una impresora 3D.



La descomposición en capas, consiste en tomar el modelo y rebanarlo en capas con una resolución (tamaño grosor de cada capa) predefinida que se puede ajustar, dependiendo de la resolución con la que se desee tener el modelo. La resolución es directamente proporcional al número de capas en las que se rebana el modelo, y por lo tanto influye directamente en el tiempo de impresión. Una vez que se rebana el modelo, se realiza el trazado como si fuera una ruta, la cual, indica las posiciones y desplazamientos que se deben realizar para formar cada capa. Este proceso es necesario, pues así se logra convertir el modelo en instrucciones que seguirá una impresora 3D para poder imprimirlo.

El entorno del programa se muestra en la figura 2.28, donde se pueden observar las zonas principales que componen la interfaz con el usuario.

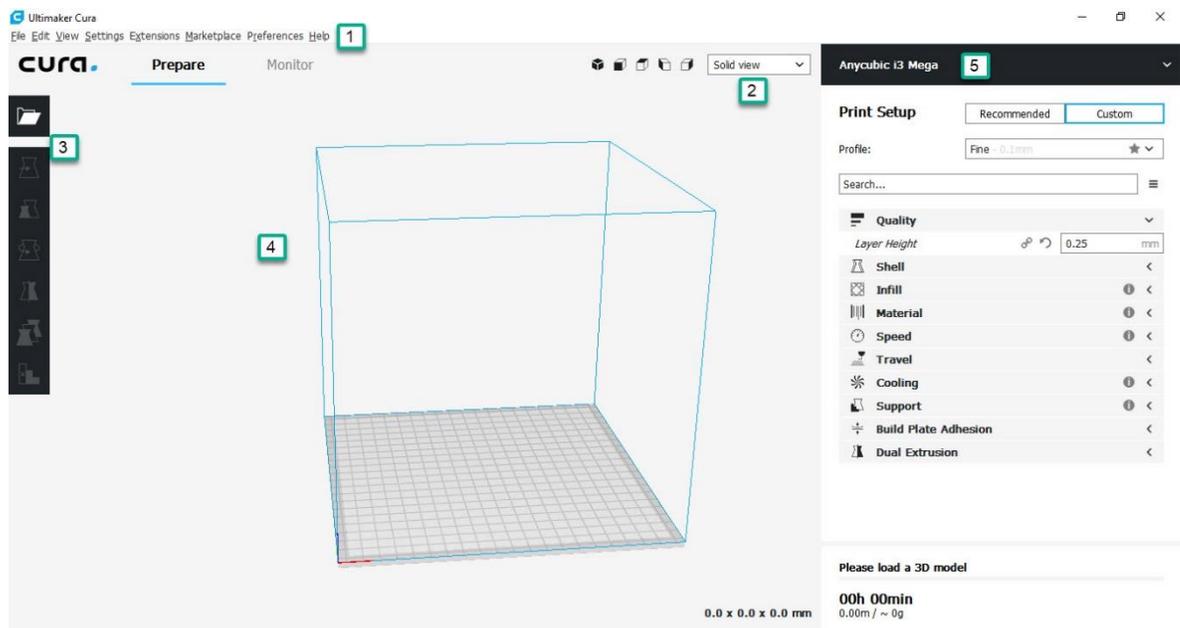


Figura 2.28. Interfaz Cura

Donde:

1. Barra de opciones. – En ella se muestran las opciones para configurar parámetros del programa.



- a. File. – En esta pestaña se encuentran las opciones para crear un nuevo proyecto, abrir uno existente, guardar el avance, exportar el modelo y salir del programa.
  - b. Edit. – En esta pestaña se encuentran las funciones para la colocación de los modelos, donde se puede regresar a una acción previa.
  - c. View. – En esta categoría se encuentran las opciones para las diferentes vistas del modelo, vista de sólido, rayos x, y vista de capas, así como se puede modificar la orientación de la vista del modelo.
  - d. Settings. – En esta pestaña se encuentran las configuraciones de la impresora y los perfiles de impresión.
  - e. Extensions. – Contiene los ajustes del código generado después de procesar el modelo, también se puede ejecutar un asistente para realizar la búsqueda de actualizaciones del programa.
  - f. Marketplace. – Es una función del programa que ejecuta una nueva ventana en donde se pueden seleccionar funcionalidades extra al programa, las cuales pueden ir desde los materiales, vistas, etc.
  - g. Preferences. – En esta pestaña se muestran las configuraciones del programa, como idioma, opciones que puede visualizar el usuario, perfiles y características de la impresora.
  - h. Help. – En esta pestaña se encuentra la documentación referente al programa, además se encuentra una guía de ayuda para el uso del programa.
2. Barra de vista, en esta sección de la interfaz se muestran los iconos para las funciones de vista del modelo.
- a.  . – Coloca la plataforma de impresión con el origen orientado en perpendicular.
  - b.  . – Muestra la vista frontal de la plataforma de impresión.
  - c.  . – Muestra la vista superior de la plataforma de impresión.



- d. . – Muestra la vista lateral derecha de la plataforma de impresión.
  - e. . – Muestra la vista lateral izquierda de la plataforma de impresión.
3. Es una barra vertical con herramientas para la colocación del modelo sobre la plataforma de impresión.
- a. Abrir archivo. – Es un icono que abre una ventana en la que se puede seleccionar el archivo con el que se va a trabajar, una vez seleccionado, se carga el modelo en la plataforma de impresión.
  - b. Mover. – Con esta herramienta se puede desplazar el modelo sobre cualquier eje (x, y, z), por lo usual, se emplea para acomodar de una manera eficiente el modelo.
  - c. Escalar. – Esta herramienta permite aumentar o disminuir las dimensiones del modelo, puede realizar reducciones o ampliaciones en proporciones constantes y también modificar las dimensiones de un solo eje.
  - d. Rotar. – Con esta herramienta se puede girar con respecto a un eje el modelo.
  - e. Espejo. – Permite aplicar un efecto de espejo al modelo, invirtiendo sus lados, como si se visualizaran en un espejo.
4. Zona de impresión. – Se muestra la zona que corresponde al área de impresión disponible que posee la impresora que estamos ocupando. En esta zona, se muestra como se verá el modelo impreso con respecto al tamaño de la plancha de impresión.
5. Ajustes de impresión. – En esta sección del programa se muestran los ajustes y configuraciones que pueden realizarse para la impresión del modelo.
- a. Quality. – Se muestra el valor de la resolución que tendrá el modelo, se pueden colocar valores entre 0.1 y 0.3, siendo el primero el de mayor resolución y el segundo el de menor resolución.



- b. Shell. – Muestra los ajustes de impresión de la superficie del modelo, como el número de líneas que conformaran el perímetro exterior y cuantas el interior, el número de capas que debe haber debajo y encima del modelo.
- c. Infill. – Contiene los ajustes del relleno que tendrá el modelo, se puede modificar el porcentaje que tendrá el relleno, y la forma del mismo, si será un diseño de malla cuadrada, triangular, zigzag, etc.
- d. Material. – Se configuran los parámetros del material utilizado, los cuales son, temperatura del material, temperatura de la plancha de impresión y existe (según el modelo de la impresora) la retracción de material.
- e. Speed. – Ajusta la velocidad de los traslados y la creación de los acabados de las capas, este ajuste va relacionado con las características de la impresora e influye en el tiempo de impresión y calidad de la misma.
- f. Support. – Genera soportes para partes del modelo que las necesite.
- g. Build Plate Adhesion. – Modifica la adhesión que tiene el modelo con la plancha de impresión, el tipo de fijación varía dependiendo de las características del modelo.

Por último, se encuentra el botón de *prepare*, el cual procesa el modelo con la configuración establecida, y muestra el tiempo y cantidad de filamento aproximado para la impresión del modelo. Si se realizan cambio en la configuración para la impresión, es necesario volver a presionar el botón de *prepare*, para que cargue la información y procese el modelo.



### **2.9.3 Impresora Anycubic i3 Mega.**

Una impresora en tres dimensiones o 3D, es una máquina que puede imprimir modelos tridimensionales. Su funcionamiento es con base a la deposición aditiva de material plástico fundido, realizando recorridos desde coordenadas en tres dimensiones, de esta manera forma hilos que en conjunto conforman capas y las capas forman el modelo. La impresora Anycubic i3 Mega es una impresora por deposición aditiva de material, sus características son [23]:

- Resolución 0.05-0.3.
- Exactitud en los ejes x/y 0.0125 mm, z 0.002mm.
- Imprime en material PLA, ABS, HIPS, Wood.
- Diámetro del extrusor 0.4mm/1.75mm.
- Velocidad de impresión: 20-100 mm/s.
- Temperatura máxima del extrusor: 260 °C.
- Temperatura máxima de la plancha: 110 °C.
- Soporta archivos: .stl, .obj, .dae, .amf.
- Terminales de comunicación Ranura tarjeta SD y conexión USB.

La impresión por deposición aditiva de material, consiste en formar un hilo de material que es colocado sobre una plancha caliente, la cual mantiene adherido el material a la misma. El dispositivo que se encarga de depositar el material fundido se llama extrusor, consiste en una pieza cuyo diámetro se ve reducido a la salida y la entrada del material tiene un diámetro mayor (normalmente 1.7mm a la entrada y 0.4mm a la salida), el filamento, es el material que pasa por el extrusor, el cual al calentarse lo derrite para poder reducirlo a un diámetro de 0.4mm. El proceso de formación consiste en desplazar el extrusor a manera de que forme una capa del modelo, en cuanto la termina, sube a la siguiente capa superior, realiza el mismo procedimiento hasta tener completo el modelo.



## **2.10 Plataformas para el Internet de las Cosas**

Las plataformas para el internet de las cosas proporcionan un espacio en el que se pueden desarrollar proyectos funcionales dotándolos de las características de comunicación, conectividad, alojamiento de información y procesamiento de datos.

Una plataforma, reúne y brinda los requerimientos de software necesarios para incorporar en un proyecto las características de comunicación e interacción necesarias para el Internet de las cosas.

A continuación se describen algunas de las plataformas disponibles para desarrollar proyectos dentro del paradigma del internet de las cosas, en donde se describen sus características y funcionalidades. Para el desarrollo de este proyecto, se utilizó la plataforma Cayenne, debido a las características que ofrece para el desarrollo de proyectos y a las funcionalidades que posee, pues ofrece bibliotecas compatibles con el entorno de programación Arduino, además es compatible con la tarjeta de desarrollo ESP-8266.

### **2.10.1 Amazon Web Services**

AWS (por sus siglas en inglés) es una plataforma privada en la nube, la cual ofrece más de 175 servicios integrales de centros de datos a nivel global. Ofrece tecnologías de infraestructura como cómputo, almacenamiento, bases de datos, así como, aprendizaje automático e inteligencia artificial, lagos de datos, y análisis de internet de las cosas. Con el fin de obtener el mejor costo y rendimiento.

Cuenta con 69 zonas de disponibilidad en 22 regiones geográficas de todo el mundo, lo que la hace una de las compañías con más servidores. [24]

Para poder utilizar la plataforma en la versión de prueba gratuita durante 1 año es necesario crear una cuenta en el portal de internet de la empresa (figura 2.29).



Figura 2.29. Crear cuenta AWS

### 2.10.2 Android Things de Google

Es una plataforma que brinda a los usuarios un entorno de programación para desarrollar aplicaciones para dispositivos del Internet de las cosas. Permite construir aplicaciones en las más populares tarjetas de desarrollo, como la Raspberry Pi 3 [25], pues cuenta con una paquetería administrada por Google, que provee el soporte para las diferentes tarjetas que existen en el mercado. La plataforma, es una consola de programación (figura 2.30) que se instala en el dispositivo para el internet de las cosas, se comunica con el entorno de programación de Android Studio, desde donde se muestran los dispositivos conectados y es posible realizar cambios en las aplicaciones, cargando imágenes nuevas de las mismas, como si fueran actualizaciones de la App. La comunicación de la consola-programador-dispositivo, se logra a través del uso de los servicios de Google.

Para poder comenzar a trabajar con Android Things, es necesario contar con uno de los kits que recomienda el entorno de Android, ya sea una Raspberry Pi, o una tarjeta NXP i.MX7D. Para poder trabajar con las tarjetas, se debe descargar la imagen de Android Things y después instalarla en la tarjeta del kit que hayamos adquirido.



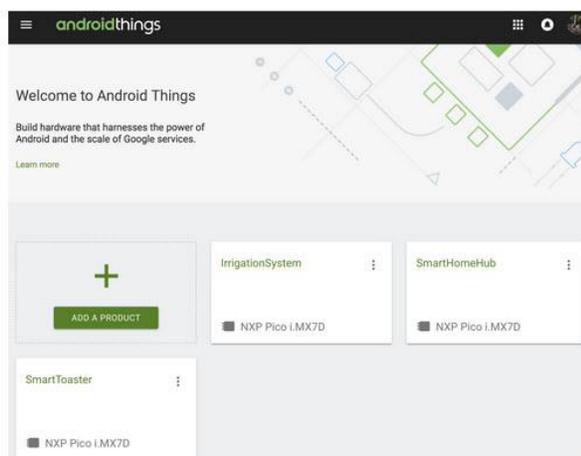


Figura 2.30. Consola Android Things

### 2.10.3 Microsoft Azure IoT

Es una plataforma creada por Microsoft para brindar un entorno para la creación de aplicaciones enfocadas al Internet de las cosas, brinda, servicios de conectividad, acceso a la nube, blockchain, inteligencia artificial, bases de datos, entre otros, además ofrece plantillas de aplicaciones específicas para minoristas, enfocadas a distintas áreas de aplicación del IoT, como supervisión de la calidad del agua, consumo de agua, monitoreo de energía eléctrica, servicios de logística para empresas. [26]

Para acceder a la plataforma, es necesario contratar un plan que se adapte a la cantidad de dispositivos que se deban tener conectados a la plataforma, dependiendo del número de dispositivos será el precio que se cobre al mes por cada dispositivo que utilice la plataforma. Adicionalmente, cada dispositivo, cuenta con 50000 mensajes durante 1 mes, cada mensaje, no debe exceder el tamaño de 1Kb, de hacerlo, se cobrará el excedente con mensajes, es decir, si un mensaje tiene como tamaño final 3.5 Kb, ese mensaje contará como 4 mensajes.

Posee un catálogo de dispositivos certificados que pueden ser utilizados en la plataforma Azure. Existen diferentes tipos de dispositivos, cada uno de ellos,





enfocado a una tarea distinta. Cuenta con sensores de presencia, actuadores, medidores, interruptores, entre muchos otros dispositivos más. Entre las tarjetas electrónicas disponibles, se encuentran:

- Raspberry Pi 3.
- Sparkfun-ThingDev
- Adafruit feather M0 Wifi.
- Intel NUC

Solo por mencionar algunos.

#### **2.10.4 myDevices Cayenne**

myDevices es una empresa que se encarga de brindar soluciones a situaciones cotidianas a través del uso de dispositivos de IoT. Posee un amplio catálogo de componentes electrónicos, sensores, actuadores y soluciones comerciales que se pueden adquirir desde su portal de Internet. Las soluciones comerciales son dispositivos creados para llevar a cabo una función, como un botón de pánico que envía la posición actual dentro de una instalación, un sistema de monitoreo y control de la temperatura dentro de un refrigerador para carne, etc. Cuenta con una amplia variedad de integraciones con aplicaciones existentes creadas con otras plataformas, como Amazon Web Services, Azure, Google Cloud, entre otros. [27]

Para poder probar el funcionamiento de la plataforma y hacer uso de las funciones que posee myDevices, es necesario crear una cuenta con usuario y contraseña para acceder a la plataforma llamada Cayenne, la cual, es una plataforma compatible con una gran cantidad de tarjetas electrónicas, sensores, actuadores, etc. Además, posee una gran cantidad de características que la hacen atractiva, cuenta con una plataforma para acceder con computadora y tiene aplicación móvil para acceder al portal desde un teléfono inteligente. Permite la interacción con varios dispositivos, conectados de manera simultánea, donde se puede visualizar mediante el portal o



la aplicación móvil la información obtenida por cada dispositivo, adicionalmente, el entorno puede llevar a cabo acciones y emitir alertas, programadas previamente en la plataforma.

Para poder utilizar Cayenne se debe ingresar a la página desarrolladores de myDevices<sup>2</sup> donde se describen las características de la plataforma y se puede llevar a cabo el registro para poder tener acceso al entorno. Para ello, es necesario crear una cuenta como se muestra en la figura 2.31, es necesario escribir el primer nombre, el primer apellido, un correo electrónico y una contraseña.

The image shows the Cayenne registration page. At the top left is the Cayenne logo. To its right are links for 'FEATURES' and 'SUPPORTED HARDWARE'. The main heading is 'Start Building Today'. Below this are four input fields: 'First Name', 'Last Name', 'Email Address', and 'Password'. Under the 'Email Address' field is a checkbox with the text 'I agree to the myDevices Cayenne terms.'. At the bottom center is a blue button labeled 'GET STARTED FREE'.

Figura 2.31. Ventana de registro Cayenne

Una vez hecho el registro, se puede acceder a la plataforma y a la ventana del usuario (figura 2.32).

<sup>2</sup> <https://developers.mydevices.com/cayenne/features/>

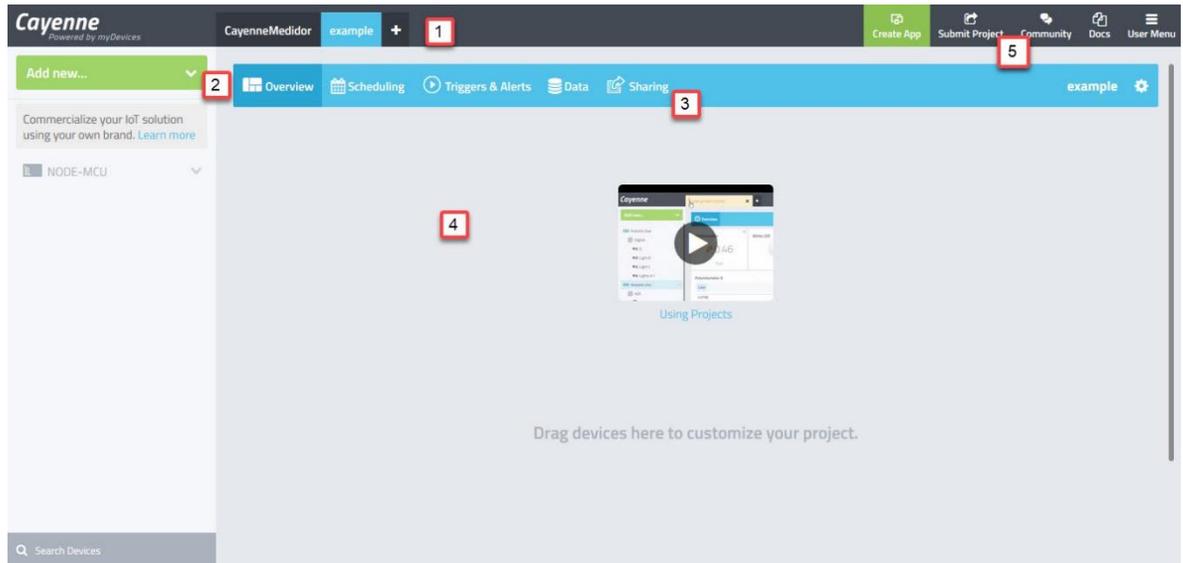


Figura 2.32. Interfaz usuario Cayenne.

Donde:

1. – Es la barra de proyectos, en ella se muestran los proyectos creados, adicionalmente, en la parte derecha de las ventanas de proyecto, se encuentra el botón para crear nuevos proyectos, tiene un signo de adición +, al presionarlo, se abre una nueva pestaña en la barra de proyectos.
2. – Paleta agregar componentes, en esta zona de la interfaz, se pueden agregar nuevos componentes al actual proyecto, para ello, se presiona el botón que dice, *add new*, y se mostraran las cosas que se pueden agregar:
  - Device/Widget. – En esta categoría se encuentran las tarjetas o dispositivos con los que la plataforma puede interactuar, algunas de las tarjetas son: la raspberry pi, tarjeta Arduino con Ethernet shield, y algunas tarjetas basadas en el ESP8266. En cuanto a los dispositivos que se pueden agregar, se encuentran algunos sensores, de temperatura, luminosidad, presión, entre otros. Los widgets son



- **Event.** – En esta categoría se puede establecer la ejecución de eventos con acciones y notificaciones, con la condición de ejecutarse en un momento en específico, para ello, es necesario colocar la hora con minutos y segundos, el día, mes y año. Adicionalmente se puede configurar que el evento se repita cada día, mes, o año, así como cada vez que se ejecute, se puede enviar una notificación por mensaje de texto o correo electrónico. En cuanto a las acciones que se pueden configurar, dependerá de las funciones que pueda llevar a cabo el dispositivo con el que se está trabajando.
  - **Triggers.** – Son acciones que se pueden ejecutar cuando se cumple una condición, es decir, si el evento A se lleva a cabo, entonces, ejecuta la acción b. las acciones pueden ser llevadas a cabo por dispositivos, sensores o actuadores, adicionalmente, se pueden configurar notificaciones vía mensaje de texto y/o correo electrónico, las cuales son enviadas cuando se cumple la condición.
  - **Project.** – Se puede crear un nuevo proyecto.
3. – Se encuentra la barra de vistas, donde se pueden visualizar las diferentes pantallas que conforman el proyecto, las vistas son:
- **Overview.** – En esta vista se muestran los widgets que forman parte del proyecto, se puede interactuar con ellos, se puede modificar su nombre, la forma en la que se muestran los datos, el tamaño que ocupan en pantalla, entre otros.
  - **Scheduling.** – Se muestran los eventos programados del proyecto, la vista se muestra como si fuera un calendario. Desde esta misma vista se pueden crear nuevos eventos y gestionar los anteriores.
  - **Triggers & alerts.** – Se visualizan las diferentes condiciones/acciones programadas en el proyecto. Se pueden editar las existentes o crear nuevas.



- **Data.** – Muestra los valores enviados a la plataforma, almacena el valor, el dispositivo del cual proviene, el canal por el que fue recibido, el nombre del sensor, el identificador y las unidades. Los valores se almacenan conforme su recepción, la vista permite visualizar los datos en una línea de tiempo, que se puede mostrar de forma diaria, semanal, mensual y anual.
  - **Share.** – Pestaña para compartir, en esta ventana se puede compartir el proyecto mediante un enlace, donde se puede mostrar solo la pantalla principal del proyecto, donde se encuentran los widgets, O se pueden dar privilegios sobre la vista *overview*, para que puedan interactuar con los widgets.
  - **Ajustes.** – Sobre la misma barra, hasta el extremo derecho se encuentra un botón de ajustes, con el cual, se puede modificar el nombre del proyecto, o se puede eliminar, una vez eliminado el proyecto, no se puede recuperar.
4. – Se muestra el área de trabajo, en esta zona, dependiendo de la vista que está activa, se mostrará la información de cada pantalla.
5. – Se encuentran los accesos para las funciones adicionales de la plataforma:
- **Crear aplicación móvil.** – Aunque está en la versión de prueba, se puede crear una aplicación móvil en la que se incluya la funcionalidad del entorno de Cayenne, para ello, en esta opción, se brindan las credenciales para poder acceder a la API a partir del proyecto creado.
  - **Publicar el proyecto.** – En esta opción se puede publicar el proyecto en la comunidad de usuarios de Cayenne.
  - **Comunidad Cayenne.** - Muestra los proyectos recientes subidos por usuarios de Cayenne y las conversaciones o preguntas realizadas por otros usuarios.

Adicionalmente, esta plataforma cuenta con una aplicación para teléfonos inteligentes, la cual se puede descargar desde la Play Store para dispositivos Android o desde iTunes, para dispositivos iOS.



## Capítulo 3 Desarrollo experimental

Para el desarrollo de este trabajo, se realizaron algoritmos para la realización de tareas específicas de cada parte del sistema. A continuación, mediante diagramas de bloques y diagramas de flujo, se muestran las funciones del sistema de la presente propuesta.

### 3.1 Diagrama a bloques

El sistema está compuesto por los módulos que se presentan en el diagrama de bloques de la figura 3.1.

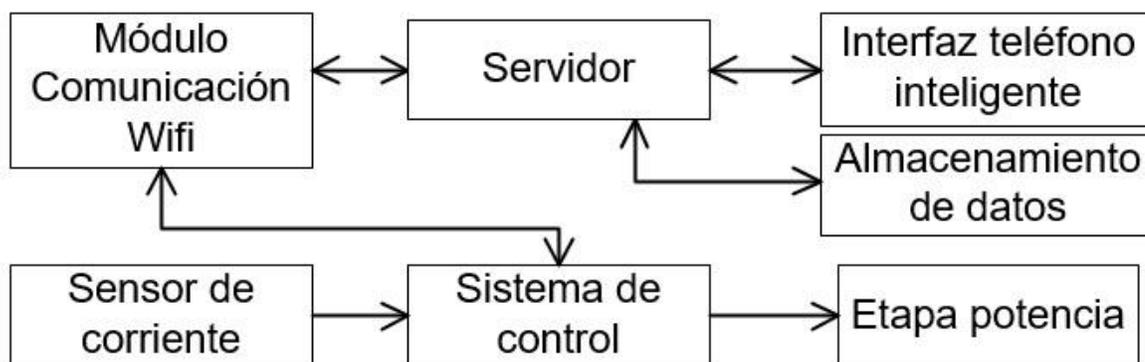


Figura 3.1. Diagrama a bloques general

Donde cada uno de los bloques representa una funcionalidad diferente del sistema, los cuales se describen a continuación:

**Sistema de control:** Es el microcontrolador que se encarga de realizar todas las funciones lógicas, comunicación, adquisición, procesamiento del sistema, se utilizó un microcontrolador ESP-8266.

**Sensor de corriente:** Se utilizó el sensor de corriente de efecto Hall ASC-712 versión para 30 A, este obtiene información de la corriente utilizada, la cual es almacenada en una variable.



**Almacenamiento de datos:** La información de fecha, hora y consumo, es almacenada en una base de datos. Al acceder a través de la aplicación, el usuario podrá revisar el histórico del consumo realizado.

**Módulo comunicación Wifi:** Realiza una comunicación inalámbrica con un servidor a través de la Internet para enviar y recibir información.

**Interfaz para teléfono inteligente:** Es una aplicación desarrollada para dispositivos móviles, la cual funciona como interfaz visual que permite comunicar al usuario y al sistema.

**Etapas de potencia:** Con este componente se realiza el control para la activación o desactivación de la energía en la carga sobre la que está conectado el sistema.

### 3.2 Diagramas de flujo

Un diagrama de flujo, es la representación por tareas, pasos o actividades para poder realizar una función en específico. A continuación, se muestran los diagramas de flujo que ilustran funciones particulares del sistema.

#### 3.2.1 Diagrama de flujo para obtener la potencia instantánea

Para obtener la potencia instantánea se desarrolló un algoritmo que cuantifica la energía que se encuentra consumiendo la carga que está conectada al sistema.

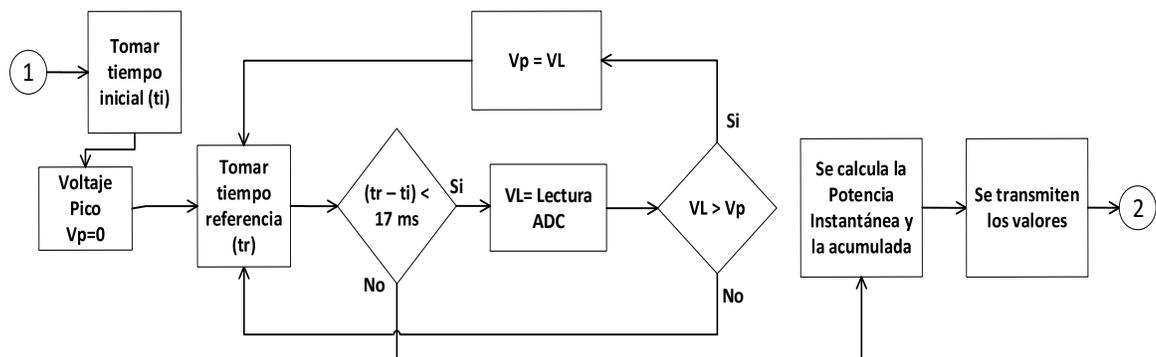


Figura 3.2. Diagrama de flujo para obtener potencia instantánea y acumulada.



Lo que realiza el algoritmo es obtener la medición máxima de corriente obtenida por el sensor, para ello, realiza muestras consecutivas de los valores obtenidos por el sensor de corriente durante 17 milisegundos, debido a que es aproximadamente la duración de un periodo completo de un ciclo de la señal sinusoidal, de esta forma, se analiza un ciclo de manera completa, cada lectura realizada, es comparada con la lectura anterior, de esa manera, al analizar lecturas consecutivas se almacena la de mayor valor en una variable. Con el valor mayor de las lecturas, se calcula, primero, la corriente instantánea, el valor de la corriente se multiplica con el valor del voltaje que hay en la línea, el resultado del producto es la potencia instantánea. A partir de la potencia instantánea, se obtiene la potencia acumulada, para ello, se desarrolló un algoritmo (figura 3.3) que realiza una espera entre una medición y otra hasta completar un segundo, para que el resultado obtenido sea proporcional a una fracción  $1/3600$  parte de lo consumido en 1 hora. Así, después de 3600 mediciones, habrá transcurrido una hora, y la cantidad acumulada será equivalente al consumo realizado en términos de Wh.

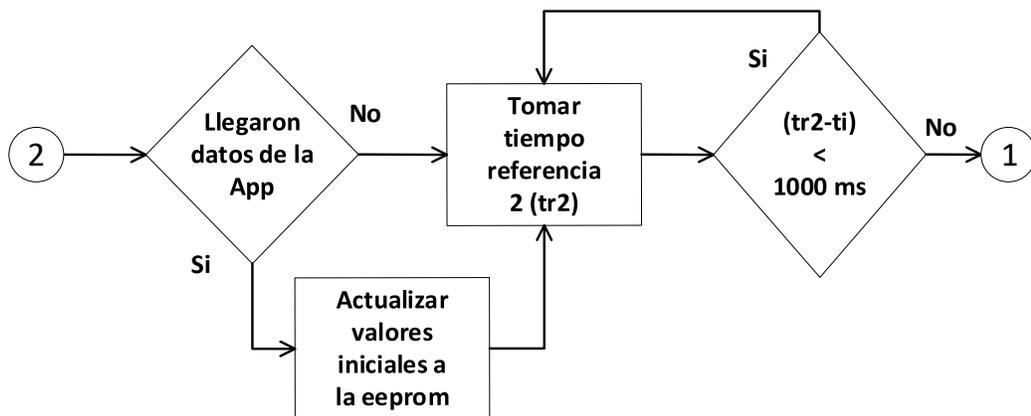


Figura 3.3. Diagrama para completar 1 segundo.





### 3.2.2 Diagrama de flujo para la comunicación inalámbrica

A continuación, se describe el procedimiento realizado para poder establecer una comunicación inalámbrica con el sistema y un servidor a través de internet.

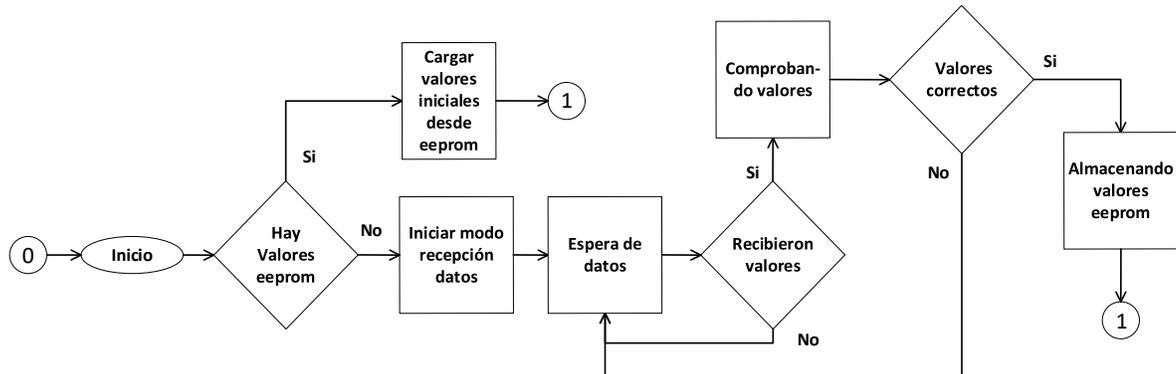


Figura 3.4. Diagrama de flujo configuración valores iniciales.

En la figura 3.4 se muestra el diagrama de flujo que describe la configuración de los valores iniciales para la comunicación inalámbrica, por lo que, la primera evaluación que realiza es verificar que existan valores cargados en la memoria EEPROM, de no ser así, se inicia un modo en que la tarjeta enciende el módulo de comunicación inalámbrica Wifi y su configuración de trabajo, entra en el modo *acces point*, donde es posible conectarse a la tarjeta, para enviar la información para conectarse a una red con acceso a internet. Por ello, recibe la información, intenta conectarse, si lo logra, almacena la información, de lo contrario, pide nuevos valores.

### 3.2.3 Diagrama de flujo configuración Cayenne

Para utilizar el entorno de Cayenne, se realizaron configuraciones y se establecieron notificaciones para cuando ciertos eventos tengan lugar. Para ello, se describe a continuación el funcionamiento de la aplicación desarrollada en Cayenne (figura 3.5).



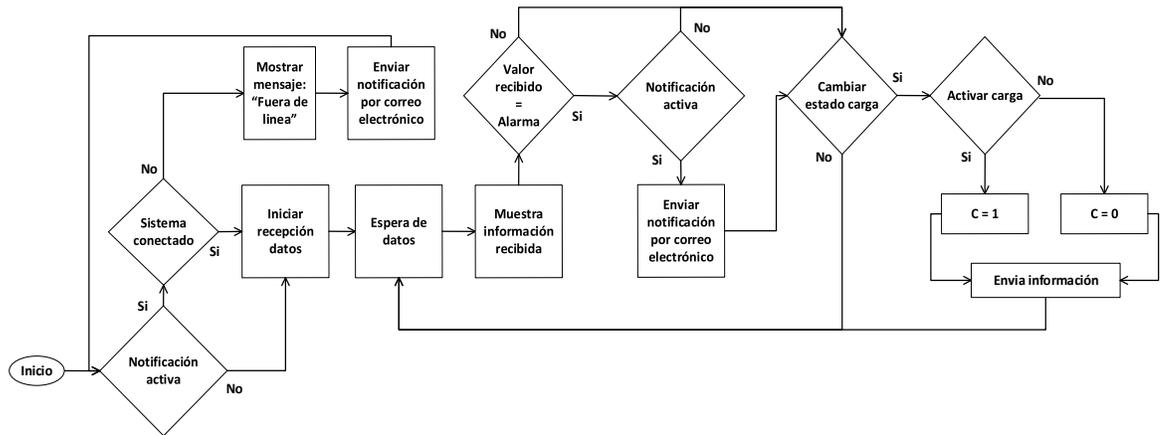


Figura 3.5. Diagrama de flujo funcionamiento Cayenne.

Las configuraciones realizadas para el entorno de Cayenne, permiten identificar si el sistema se encuentra en línea, de no ser así, se le notifica al usuario que se encuentra fuera de línea. Una vez que el sistema se encuentra enviando información al entorno de Cayenne, éste muestra la información donde corresponde, el canal por el que se está enviando la información definirá a donde pertenece. Una vez que la información recibida fue enviada, se realiza una comparación para conocer si los valores enviados son iguales a los valores definidos como alertas, si es así, se envía una notificación, si se encuentran activas, al usuario, donde se le indica la variable que ha alcanzado o superado el valor de la alerta. Adicionalmente, si el usuario modifica el estado de la carga, encendido o apagado, este valor es enviado al sistema para que realice los cambios pertinentes.

En cuanto al funcionamiento de las notificaciones, cabe mencionar que el entorno de Cayenne permite al usuario activar o desactivar las notificaciones; ya que si las condiciones de las notificaciones siguen cumpliéndose, la aplicación seguirá enviando notificaciones al usuario hasta que la condición deje de cumplirse o en su caso, el usuario apague la notificación.





### 3.3 Medición de la corriente eléctrica.

Para la obtención del consumo, se realizaron experimentos para interpretar el comportamiento de la salida del sensor en AC (Corriente Alterna), la gráfica presentada en la figura 3.6 muestra el comportamiento en DC (Corriente Directa). Para la caracterización en AC, se analizó la salida del sensor, con la ayuda de un osciloscopio, y una carga de 2300 W y se obtuvo la señal mostrada en la figura 3.6, donde se puede apreciar que se entrega un voltaje en directa, pero presenta variaciones que van desde un voltaje mínimo 880 mV a un voltaje máximo 4.08 V.



Figura 3.6. Señal de salida del sensor con carga 2300W alimentada con AC.

Para calcular el valor de la corriente, fue necesario conocer el modelo matemático que describe el comportamiento de la salida del sensor con respecto a la entrada. Por lo que se realizó un experimento en el que se conectaron diferentes cargas resistivas y a través de un análisis matemático utilizando la ecuación de la recta, se obtuvo el modelo del comportamiento del sensor. Para este análisis, se trabajó con los valores máximos de la señal observada y primero se empleó la ecuación de la recta, definida por la expresión:

$$V = mI + B \quad (\text{ec. 3.1})$$

Donde:





$V$  = Voltaje de salida del sensor.

$I$  = Corriente que circula por el sensor.

$B$  = Voltaje cuando la corriente es cero.

$m$  = pendiente o razón de cambio.

Donde  $m$  está definida como:

$$m = \frac{V_2 - V_1}{I_2 - I_1} \quad (\text{ec. 3.2})$$

De los valores mostrados por el fabricante, se sustituyen en la ecuación 3.2 y se obtiene:

$$m = \frac{4.5 - 2.5}{30 - 0} = \frac{2}{30} = \frac{1}{15} \quad (\text{ec. 3.3})$$

Sustituyendo el valor de  $m$  en la ecuación 3.1 y utilizando la ordenada al origen, encontramos la expresión:

$$V = \frac{1}{15}I + 2.5 \quad (\text{ec. 3.4})$$

Ahora bien, el valor del voltaje, es un valor conocido, pero el valor de la corriente es el valor desconocido, por lo tanto, se realiza un despeje de la variable  $I$  de la ecuación 3.4.

$$I = 15V - \frac{75}{2} \quad (\text{ec. 3.5})$$

Donde, al realizar experimentos con cargas resistivas conocidas, se obtuvieron valores que se compararon con la ecuación 3.5, donde se observó que los valores



obtenidos son valores pico, y se necesita de valores eficaces (RMS). Para obtener el Voltaje RMS que entrega el sensor, se implementó un algoritmo que toman lecturas consecutivas durante 17 milisegundos del convertidor analógico digital (dado a que es la duración de un ciclo completo de una señal de 60 Hz), y cada lectura realizada es comparada con la anterior, con el fin de obtener el valor mayor, el cual representa al  $V_p$  (Voltaje de Pico). Para el cálculo de la corriente RMS se utiliza (ec. 3.6).

$$I_{RMS} = \frac{I_P}{\sqrt{2}} \quad (\text{ec. 3.6})$$

Sustituyendo (ec. 3.6) en (ec. 3.4) e indicado en lugar  $V$  al  $V_P$  tenemos:

$$V_P = \frac{1}{15} (I_{RMS} * \sqrt{2}) + 2.5$$

Despejando a  $I_{RMS}$ .

$$I_{RMS} = \frac{15(V_P - 2.5)}{\sqrt{2}} \quad (\text{ec. 3.7})$$

Con la expresión mostrada en (ec. 3.7), se obtiene la corriente eficaz que se encuentra midiendo el sensor de corriente.

Para obtener la energía consumida, se debe realizar un acumulativo de la potencia instantánea, la cual está definida por (ec. 3.8).

$$P_{ins} = V_{RMS} \times I_{RMS} \quad (\text{ec. 3.8})$$

Dónde:

$P_{ins}$  = Potencia instantánea.

$V_{RMS}$  = Voltaje eficaz de la fuente de alimentación.

$I_{RMS}$  = Corriente eficaz obtenida en (ec. 3.7).





El valor de la potencia instantánea se almacena en un registro.

Para el cálculo de la energía consumida, se realizó un algoritmo que se describe en la figura 3.2, la cual almacena en un registro la potencia obtenida. La obtención de la potencia instantánea se realiza cada segundo, para poder convertir P instantánea a Potencia consumida se utiliza la ecuación 3.9.

$$P_{consumida} = \frac{P_{ins}}{\Delta t} \quad (\text{ec. 3.9})$$

Donde  $\Delta t$  es igual a 3600, se debe a que cada lectura es tomada en el espacio de 1 segundo. Por la tanto, corresponde a 1/3600 parte de lo consumido en 1 hora.

### 3.4 Código en el entorno de Arduino

El código creado para el funcionamiento del sistema de la presente propuesta está basado en los diagramas de flujo de las figuras 3.2, 3.3 y 3.4.

La captura de código mostrada en la figura 3.7 describe las bibliotecas utilizadas:

```
#include <FS.h> //se utiliza para la gestion de los valores almacenados en SPIFFS
#include <ESP8266WiFi.h> //se utiliza para realizar la comunicacion inalambrica
#include <DNSServer.h> //se utiliza para implementar un servidor en modo STA y AP
#include <ESP8266WebServer.h> //se utiliza para la creacion de un servidor web
#include <WiFiManager.h> //se utiliza para gestionar el arranque en modo STA y AP
#include <ArduinoJson.h> //se utiliza para manejar datos del tipo Json
#define CAYENNE_DEBUG //declaracion de variable constante
#define CAYENNE_PRINT Serial // eclaracion de variable constante
#include <CayenneMQTTESP8266.h> //comunica al sistema con cayenne
#include <EEPROM.h> //permite gestionar el contenido de la eeprom
```

Figura 3.7. Segmento código "Declaración de bibliotecas".

Las bibliotecas se colocaron al inicio del programa, para que sean las primeras en cargarse y estén disponibles a la hora que se utilice una función de ellas. Las primeras bibliotecas fueron agregadas debido a que realizan funciones de comunicación inalámbrica, y la última librería, pertenece al entorno de Cayenne.



El siguiente segmento de código (figura 3.8) corresponde a las variables globales que utiliza el programa, las variables almacenan valores o resultados de procesos del programa principal, y como pueden ser utilizados en cualquier parte del programa, fue necesario que se definieran como globales.

```
char cayenneUserName[40] = ""; // almacena usuario para Cayenne
char cayennePassword[40] = ""; // almacena contraseña para Cayenne
char cayenneClientID[40] = ""; // almacena identificador para Cayenne

float lecturaVp = 0.0; //almacena el valor obtenido de la digitalizacion del adc
float vpMax = 0.0; //almacena el mayor valor de la digitalizacion del adc
float Vrms = 0.0; //guarda el valor en voltaje de la digitalizacion del adc
```

Figura 3.8. Segmento código “variables globales 1”.

Las variables de tipo *char*, corresponden a los registros donde se almacenan las credenciales para comunicarse con la plataforma Cayenne, para que se identifique el dispositivo. Las siguientes variables de tipo *float*, en ellas se almacenan los resultados obtenidos de la cuantificación de la corriente instantánea, en donde se deben de tomar lecturas consecutivas durante un periodo de tiempo de 17 mili segundos, y cada resultado obtenido compararlo con el anterior para obtener el valor más grande.

El próximo segmento (figura 3.9), describe las variables utilizadas para el almacenamiento del valor de las tarifas. En las variables “*float* tarifaf1” se almacenan el valor en decimal de la primera tarifa, lo mismo es el caso para las otras dos tarifas y la de alto consumo doméstico.



```
float tarifaf1 = 0.0; //almacena el valor de la tarifa rango 1
float tarifaf2 = 0.0; //almacena el valor de la tarifa rango 2
float tarifaf3 = 0.0; //almacena el valor de la tarifa rango 3
float DACf = 0.0; //almacena el valor de la tarifa DAC

int salvar = 0; //variable que indica cuando es necesario almacenar nueva informacion.

byte ht1; //almacena el byte mas significativo
byte lt1; //almacena el byte menos significativo
byte ht2; //almacena el byte mas significativo
byte lt2; //almacena el byte menos significativo
byte ht3; //almacena el byte mas significativo
byte lt3; //almacena el byte menos significativo
byte hDAC; //almacena el byte mas significativo
byte lDAC; //almacena el byte menos significativo

float dinero = 0.0; //almacena la cantidad a pagar por el consumo realizado
```

Figura 3.9. Segmento de código "variables globales 2".

La variable *salvar*, es una bandera que indica cuando se debe almacenar la información de las tarifas que establezca el usuario, la variable depende del valor de un botón que se encuentra programado en la interfaz en Cayenne.

Las variables de tipo *byte* almacenan los 8 bits más y menos significativos de cada tarifa, de esta forma se pueden almacenar en la memoria no volátil de la tarjeta.

En la figura 3.10, se muestran las variables tipo *unsigned long* e *int*, las cuales son utilizadas para cuantificar los tiempos de ejecución del programa, las variables tipo *float*, almacenan los resultados del cálculo de la potencia instantánea, y el acumulado de la energía consumida.



```
unsigned long tmpRef = 0; //almacena el tiempo de referencia para duracion de procesos
unsigned long tmpIni = 0; //almacena el tiempo cuando el ciclo esta en el comienzo
unsigned long fin = 0; //almacena el tiempo cuando el ciclo llega al final

int d = 0; //almacena el resultado de un contador
int dias = 0; //lleva el registro de cuantos dias han transcurrido.
int seg = 0; //lleva la cuenta de los segundos transcurridos para el envio de la informacion
int tiempo = 5; //tiempo en segundos que el sistema envia informacion a la plataforma

float ampRms = 0.0; // almacena la corriente instantanea cuantificada por el sensor
float voltLinea = 123.5; //valor del voltaje AC de linea
float potIns = 0.0; // guarda el valor de la potencia instantanea.
float potAcum = 0.0; // almacena el valor de la potencia consumida
float potWh = 0.0; // guarda el valor de la energia consumida
float potKWh = 0.0; // valor de la enrgia consumida en kWh

bool shouldSaveConfig = false; //bandera utilizada para almacenar nuevos valores en modo AP
```

Figura 3.10 Segmento de código "variables globales 3".

El siguiente segmento de código, corresponde a la configuración del programa, a las configuraciones que solamente se ejecutan 1 vez durante la ejecución del programa (figura 3.11).

```
void setup() { //funcion setup, se ejecuta solamente 1 vez
  Serial.begin(115200); //establece una velocidad de comunicacion a 115200 bits por segundo
  Serial.println("abriendo sistema de archivos."); //imprime un mensaje por el puerto serial

  /////////////// funcion para leer los valores almacenados en la memoria SPIFFS////////////////////
  if (SPIFFS.begin()) { //inicia el sistema de archivos
    Serial.println("Sistema de archivos abierto"); //imprime un mensaje por el puerto serial
    if (SPIFFS.exists("/config.json")) { // condicion, si existen los ficheros se leen y cargan
      Serial.println("leyendo archivos de configuracion"); //imprime un mensaje por el puerto serial
      File configFile = SPIFFS.open("/config.json", "r"); //abre el archivo configFile de la memoria SPIFFS
      if (configFile) { //condicion si el archivo existe
        Serial.println("opened config file"); //imprime mensaje por el puerto serial
        size_t size = configFile.size(); //mide el tamaño del archivo
        std::unique_ptr<char[]> buf(new char[size]); // asigna un buffer para almacenar el contenido del archivo
        configFile.readBytes(buf.get(), size); // lee la informacion del buffer considerando el tamaño del archivo
        DynamicJsonBuffer jsonBuffer; // declaracion de una variable perteneciente de la clase dynamycJsonBuffer
        JsonObject json = jsonBuffer.parseObject(buf.get()); // asigna el contenido del buffer
        json.printTo(Serial); //imprime el contenido de json
        if (json.success()) { //condicion si el contenido fue cargado correctamente
          Serial.println("\nparsed json"); // imprime por el puerto serial
          strcpy(cayenneUserName, json["cayenneUserName"]); //almacena en la variable, el contenido de json.
        } else { // en caso de que no se cumpla con la condicion
          Serial.println("error al cargar valores almacenados"); // imprime un mensaje por el puerto serial
        }
      }
    }
  }
} else { //en caso de que la condicion no se cumpla
  Serial.println("failed to mount FS"); // imprime un mensaje por el puerto serial
}
```

Figura 3.11 Segmento de código "configuración o setup".





La configuración inicial del dispositivo, primero se establece una velocidad de comunicación a 115200 baudios, enseguida se utilizan las funciones de la librería *WifiManager*, donde, se evalúa si existen valores almacenados en la memoria del dispositivo, si los hay, los carga en la configuración, de no ser así, se ejecuta el asistente *AutoConnectAP*, donde se cambia el modo de operación de la tarjeta a acces point, este modo configura una red inalámbrica en la que se permite la conexión a dispositivos externos, para que a través de una dirección IP, se pueda ingresar a un formulario html donde se configuran los parámetros de red, nombre de red y contraseña de la misma. Para poder ingresar a la red que genera la tarjeta, se puede colocar una contraseña para que solo el usuario correcto pueda ingresar. Una vez que se ha conectado al sistema, se puede ingresar la información del nombre de red y de la contraseña para conectarse a ella.

```
WiFiManagerParameter custom_cayenne_user("cayenne_user", "username", "", 40); //asignacion de parametros
WiFiManagerParameter custom_cayenne_pass("cayenne_pass", "Password", "", 40); //asignacion de parametros
WiFiManagerParameter custom_cayenne_id("cayenne_id", "user_id", "", 40); //asignacion de parametros

WiFiManager wifiManager; //declaracion de una variable

wifiManager.setSaveConfigCallback(saveConfigCallback); // se llama a la funcion de salvar datos

wifiManager.addParameter(&custom_cayenne_user); //agrega un parametro en el formulario html
wifiManager.addParameter(&custom_cayenne_pass); //agrega un parametro en el formulario html
wifiManager.addParameter(&custom_cayenne_id); //agrega un parametro en el formulario html

wifiManager.autoConnect("AutoConnectAP"); // muestra el nombre de la red
Serial.println("connected...yeey :)"); // imprime un mensaje por el monitor serial

strcpy(cayenneUserName, custom_cayenne_user.getValue()); //asigna los valores introducidos a variables
strcpy(cayennePassword, custom_cayenne_pass.getValue()); //asigna los valores introducidos a variables
strcpy(cayenneClientID, custom_cayenne_id.getValue()); //asigna los valores introducidos a variables

pinMode(D3, OUTPUT); //configura el pin D3 como salida

Serial.println("Connected"); //imprime un mensaje por el puerto serial
Serial.println("Your IP is"); //imprime un mensaje por el puerto serial
Serial.println(WiFi.localIP().toString()); //imprime un mensaje por el puerto serial
Cayenne.begin(cayenneUserName, cayennePassword, cayenneClientID); //da de alta credenciales, usuario, contraseña e ID
```

Figura 3.12 Segmento de código setup 2

Adicionalmente, la biblioteca permite configurar parámetros adicionales, como otras credenciales para acceder a una plataforma, configuraciones de valores iniciales,



etc. Se configuró tres parámetros más (figura 3.12), estos parámetros corresponden a las credenciales que se deben ingresar para poder utilizar la plataforma Cayenne. Una vez que se han ingresados los valores de manera correcta, se corroboran los valores introducidos, y si no hay error en las credenciales, el asistente *AutoConnectAP* finaliza.

El siguiente segmento de código figura 3.13 se encarga de obtener los valores almacenados en la memoria eeprom de la tarjeta, éstos corresponden a los valores de cada una de las tarifas con la que se cobra el consumo eléctrico realizado, para obtener la información se lee cada una de las direcciones de la memoria, el valor, se encuentra separado en dos rangos, los 8 bits más significativos y los 8 bits menos significativos, por ello, es necesario unir ambos rangos de bits en uno solo, por lo que, primero se agregan los bits más significativos, se hace un desplazamiento de 8 bits hacia la izquierda – se desplazan los bits cargados a la posición de los bits más significativos – para cargar los menos significativos, se realiza mediante una operación OR, así solo se cargan los bits menos significativos y los más significativos no sufren modificaciones.



```
////////////////Lectura de valores almacenados en EEPROM////////////////////////////////////  
//lectura de valor tarifa 1  
ht1=EEPROM.read(4); //almacena el contenido de 1 localidad eeprom en la variable  
lt1=EEPROM.read(5); //almacena el contenido de 1 localidad eeprom en la variable  
tarifall= ht1; //carga los bits mas significativos en la posicion menos significativa  
tarifall= tarifall <<8; //realiza un desplazamiento de 8 bits hacia la derecha  
tarifall= tarifall|lt1; //aplica una operacion or para almacenar los menos significativos  
  
//lectura de valor tarifa 2  
ht2=EEPROM.read(6); //almacena el contenido de 1 localidad eeprom en la variable  
lt2=EEPROM.read(7); //almacena el contenido de 1 localidad eeprom en la variable  
tarifa22= ht2; //carga los bits mas significativos en la posicion menos significativa  
tarifa22= tarifa22 <<8; //realiza un desplazamiento de 8 bits hacia la derecha  
tarifa22= tarifa22|lt2; //aplica una operacion or para almacenar los menos significativos  
  
//lectura de valor tarifa 3  
ht3=EEPROM.read(8); //almacena el contenido de 1 localidad eeprom en la variable  
lt3=EEPROM.read(9); //almacena el contenido de 1 localidad eeprom en la variable  
tarifa33= ht3; //carga los bits mas significativos en la posicion menos significativa  
tarifa33= tarifa33 <<8; //realiza un desplazamiento de 8 bits hacia la derecha  
tarifa33= tarifa33|lt1; //aplica una operacion or para almacenar los menos significativos
```

Figura 3.13 Lectura valores eeprom.

En la función loop, se encuentra el proceso de obtención de la corriente instantánea y a partir de su medición, se calcula la energía consumida.

```
void loop()  
{  
  Cayenne.loop(); // indica el inicio de Cayenne  
  tmpIni = millis(); //se almacena el tiempo inicio  
  vpMax = 0.0; //se limpia el registro del voltaje pico maximo  
  tmpRef = millis(); // se toma el tiempo en este punto  
  while ((millis() - tmpRef) < 17) // ciclo mientras que la condicion se cumpla  
  {  
    lecturaVp = analogRead(A0); //realiza una lectura del ADC  
    if (lecturaVp >= vpMax) //compara la lectura obtenida si es mas grande, se almacena el valor  
    {  
      vpMax = lecturaVp ; //almacena el mayor valor leido  
    }  
  }  
  // fin del ciclo mientras
```

Figura 3.14 Segmento de código "Comparación de lecturas".

En la figura 3.14 se muestra el segmento de código que realiza lecturas sucesivas de los valores obtenidos por el sensor de corriente, donde, una vez que obtiene un





valor, lo compara con el anterior, y en caso de que el anterior sea menor, guarda el valor acabo de obtener, este proceso lo realiza durante 17 mili segundos, pues es la duración de un periodo de la señal sinusoidal de la energía alterna. Con ello, se obtiene el valor pico de la señal, lo que se considera como el valor pico de la corriente. El valor obtenido es utilizado para el siguiente segmento de código.

En la figura 3.15 se muestra el código empleado para el cálculo de la potencia instantánea a partir de la obtención de la corriente instantánea, y mediante la suma de los valores obtenidos de la potencia instantánea, se halla la energía consumida. Para poder visualizar la energía consumida en KWh, es necesario hacer una conversión, donde cada lectura obtenida, corresponde a 1/3600 parte de lo consumido en 1 hora. Por lo que después de 3600 lecturas, se obtendrá el equivalente a lo consumido en Wh.

```
Vrms = vpMax * (0.004882); //convierte la lectura del adc en un valor de voltaje
ampRms = ((15 * (Vrms)) / 1.4142); //aplicacion de la formula para obtener corriente instantanea
potIns = ampRms * voltLinea; //operacion para obtener la potencia instantanea
potAcum = potAcum + potIns; //Operacion donde se almacena la suma de la potencia
potWh = potAcum * 0.0003; //Convierte la potencia acumulada en Watts-hora
potKWh = potWh / 1000; //convierte los Wh a KWh
```

Figura 3.15. Segmento de código "cálculo de potencia y acumulativo de energía consumida".

Una vez que se cuenta con el valor de la energía que se está consumiendo, es posible calcular la cantidad a pagar por el consumo realizado, para ello, en la figura 3.16 se muestra el segmento de código que corresponde al cálculo del precio. Para saber cuál es la tarifa que se aplica, se compara la cantidad alcanzada del consumo, y se compara con el valor superior del primer rango, si es menor, se utiliza la primera tarifa. Si el valor superior de la primera tarifa es superado, se comienza a cobrar el excedente de la primera tarifa – a partir de los primeros 75 KWh, el 76 – se cobra al precio de la segunda tarifa. La misma mecánica se aplica para la tercera tarifa. Si la tercera tarifa es rebasada, el esquema de cobro que se aplica, es el de alto consumo doméstico – DAC – y se cobra todo lo consumido a 1 solo precio.





```
//Codigo que muestra el precio por el consumo realizado
if (0.0 < potKWh <= 75.0) //comparacion para saber si esta en la primera tarifa
{
    dinero = (potKWh * tarifal); //obtiene el precio
    //Serial.println("tarifa 1");
}
else if (75.0 < potKWh <= 140.0) //comparacion para saber si esta en la segunda tarifa
{
    dinero = ((potKWh-75.0) * tarifa2 + (tarifal * 75.0)); //obtiene el precio
    //Serial.println("tarifa 2");
}
else if (140.0 < potKWh <= 250.0) //comparacion para saber si esta en la tercera tarifa
{
    dinero = ((potKWh-140.0) * tarifa3 + (tarifal * 75.0) + (tarifa2 * 140.0)); //obtiene el precio
    //Serial.println("tarifa 3");
}
else //(PotenciaFaux>250.0) // si esta en la tarifa DAC
{
    dinero = ( potKWh * DAC); //obtiene el precio
    //Serial.println("tarifa DAC");
}
```

Figura 3.16 Obtención precio por el consumo realizado.

En la figura 3.17 se muestra el segmento de código que corresponde a la espera que realiza el sistema para que haga una espera hasta que se complete 1 segundo, para ello, se tomó una lectura del tiempo al inicio del loop, y hasta este punto se tomó una nueva lectura del tiempo, al restarlas se obtiene el tiempo que duro la ejecución del programa, se resta de 1 segundo, y la cifra resultante, es el tiempo que realiza la espera.

```
d = d + 1; //se incrementa en 1 cada que el programa se vuelve a ejecutar
seg = seg + 1; //se incrementa en 1 cada que el programa se vuelve a ejecutar

fin = (millis() - tmpIni); // se obtiene el tiempo de ejecucion del programa principal

tmpRef = millis(); // se toma el tiempo como valor de referencia
do
{
} while ((millis() - tmpRef) < 1000 - fin); //mantiene una espera para completar 1 segundo

if (seg >= 60) //3600 //condicion que evalua el tiempo transcurrido
{
    dias = dias + 1; // se incrementa en 1 el registro
    seg = 0; //reinicia el contador
}
```

Figura 3.17 Segmento de código "Espera"





En la figura 3.18 se muestra el código que realiza el envío de información hacia la plataforma de Cayenne, donde se muestra el canal por el que se enviará la información y el valor de la variable que se enviará.

```
if (d >= tiempo) // tiempo que debe transcurrir para enviar valores a Cayenne
{
  Cayenne.virtualWrite(0, potKWh); //envia el valor de la variable por el canal 0 a cayenne
  Cayenne.virtualWrite(1, potWh); //envia el valor de la variable por el canal 0 a cayenne
  Cayenne.virtualWrite(2, dias); //envia el valor de la variable por el canal 0 a cayenne
  d = 0; //reinicia el contador
}
```

Figura 3.18. Segmento de código "envío de datos a Cayenne".

En la figura 3.19 se muestra el código que verifica el estado que debe poseer la carga que se encuentra conectada al sistema, este valor es enviado desde la plataforma de Cayenne, por lo que el sistema solo obtiene el dato, y dependiendo del valor, activa o desactiva la carga, a través de la activación o desactivación del módulo con relevador.

```
CAYENNE_IN(2) //cuando llegue un valor por el canal 2
{
  int OnOff = getValue.asInt(); //se almacena en una variable el valor enviado
  digitalWrite(D3, OnOff); //Se activa pone en estado alto el pin D3
}
```

Figura 3.19 Segmento de código "Obtención del estado de la carga".

### 3.5 Desarrollo de la aplicación en Cayenne

Para crear la aplicación en el entorno de Cayenne, el primer paso fue darse de alta en la plataforma. Una vez que se cuenta con usuario y contraseña, se ingresa a la página de inicio de Cayenne<sup>3</sup> para ingresar los datos de la cuenta y poder entrar a la plataforma.

<sup>3</sup> <https://developers.mydevices.com/cayenne/signin/>



Una vez dentro, en la interfaz para el usuario, se creó un nuevo proyecto, y se nombró como CayenneMedidor. Se agregó al proyecto un nuevo dispositivo, basado en el microprocesador ESP-8266. Para ello, en el panel de agregar nuevos componentes, se seleccionó la opción de *device & widgets*, y en la opción de *microcontroller*, se escogió el dispositivo *SparkfunESP8266*. Al seleccionarlo el entorno nos brinda las credenciales necesarias para que se comunique con la plataforma, estas credenciales son tres, *MQTTUSERNAME*, *MQTTPASSWORD* Y *CLIENTID*, adicionalmente se le dio el nombre al dispositivo como *node-mcu*. Una vez que se brindaron las contraseñas, se ingresaron las contraseñas al sistema medidor. El sistema se conecta con la plataforma a través de internet y la plataforma, recibe la información que es enviada por el sistema y la muestra en el área de trabajo, *overview*.

Para hacer que el entorno tenga una funcionalidad aún mayor, se seleccionaron algunos de los widgets que proporciona el entorno, se escogió de la pestaña de widgets, una casilla con valor numérico, un medidor de aguja, un tanque y un botón. La casilla con valor numérico, contiene la información del precio a cubrir por la energía que ha consumido la carga. El medidor de aguja contiene la información de la energía que se ha consumido, el tanque, muestra la cantidad de días que el sistema lleva trabajando y el botón, define el estado en el que se encuentra la carga. Para configurar los widgets, se empleó el canal por el que la información se está transmitiendo (tabla 3.1).

Tabla 3.1. Relación Widget con canal

| Widget                 | Canal |
|------------------------|-------|
| Medidor de aguja       | 0     |
| Casilla valor numérico | 1     |
| Tanque                 | 2     |
| Botón On/Off           | 3     |
| Slider Tarifa1         | 4     |
| Slider Tarifa2         | 5     |
| Slider Tarifa3         | 6     |





|                  |   |
|------------------|---|
| Casilla precio   | 7 |
| Slider TarifaDAC | 8 |
| Botón salvar     | 9 |

Con la información enviada por el sistema, siendo visible en la zona de trabajo, se configuraron widgets, para que muestren la información. Para ello, fue necesario seleccionar el widget, y en las propiedades para su configuración establecer los parámetros para cada uno.

**Medidor de aguja** (figura 3.20). Se le dio por nombre, consumo, el canal por el que obtiene la información es el canal 1, las unidades son KWh. Adicionalmente, se establecieron los rangos en los que cambiara de color el indicador si la cantidad mostrada se encuentra dentro del rango. Los rangos corresponden a las tarifas.

Figura 3.20 Configuración widget gauge.

**Casilla valor numérico** (figura 3.21). – Se le asigno como nombre precio a cubrir, recibe información por el canal 1.





Figura 3.21 Configuración casilla de texto.

**Tanque** (figura 3.22). – Representa de manera gráfica el tiempo que hace falta para que el periodo este cubierto, recibe información por el canal 2.

Figura 3.22 Configuración días del periodo.

**Botón On/Off** (figura 3.23). – Modifica el estado de la alimentación de la carga que se encuentra conectada al sistema, envía información al sistema por el canal 3.



Figura 3.23 Configuración botón On/Off.

**Slider Tarifa1** (figura 3.24). – Representan el valor de la tarifa, se configuraron cuatro sliders de la misma forma, cada uno envía la información por un canal distinto, el valor que envían es entero, por lo que no se deben considerar puntos decimales, solo los dígitos, el sistema se encarga de convertirlos con punto decimal, dividiéndolos entre mil.

Figura 3.24 Configuración slider tarifa.



**Botón salvar** (figura 3.25). – Se utiliza como bandera que indica el momento en el que la información enviada por los sliders debe ser almacenada en la memoria no volátil del sistema, envía la información al sistema por el canal 9.

Figura 3.25 Configuración botón salvar.

Una vez que se tuvieron los widgets configurados, se realizó la configuración de las notificaciones, para ello, en la venta de *add new*, se seleccionó *trigger*.

Para la primera notificación, se configuro el estado del medidor, si se encuentra apagado o fuera de línea, envía una notificación mediante un mensaje por correo electrónico y como mensaje de texto para un número móvil. Para configurar la notificación, se asignó como nombre *fuera de línea*, se seleccionó el dispositivo NODE-MCU como dispositivo sobre el que se evalúa la condición “*Si se encuentra fuera de línea*”, y en la sección “*entonces se notifica mediante*”, se colocó el correo electrónico al que llega la notificación, junto con el número de teléfono móvil. Las configuraciones realizadas se muestran en a figura 3.26.

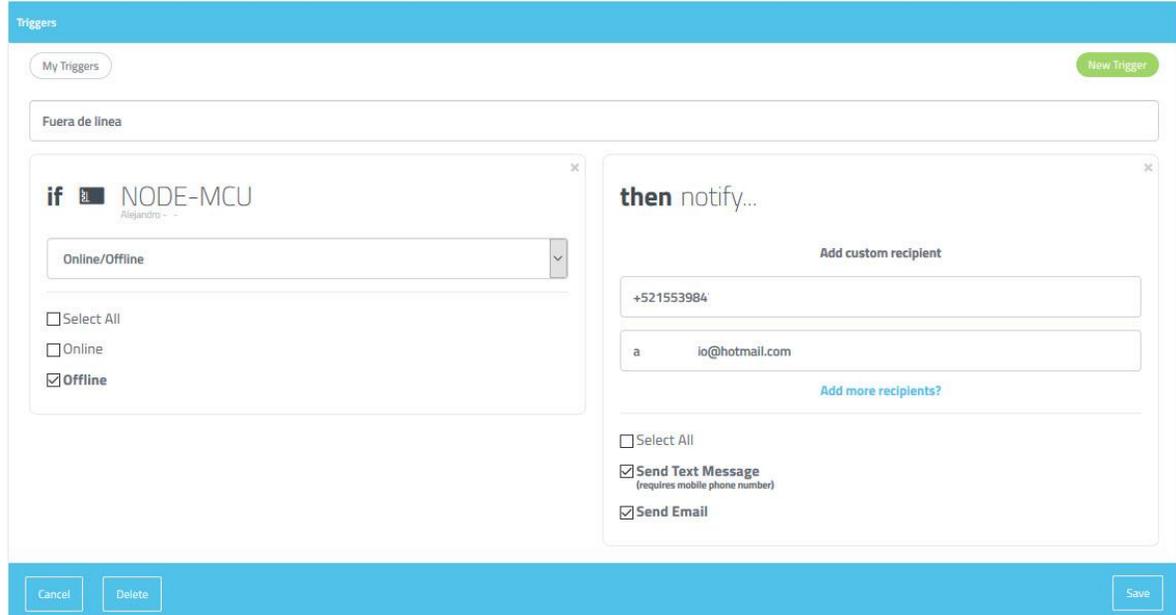


Figura 3.26 Configuración fuera de línea.

Para la segunda notificación se realizó para el medidor de aguja, la notificación se realizó en tres partes, cada parte corresponde a uno de los rangos que tiene configurado el medidor. En la ventana de triggers se configuró la notificación, a modo que se envía una notificación cuando un intervalo del medidor es rebasado. Para la primera notificación que corresponde con el primer rango, se estableció que, si rebasan los 75 KWh en un KWh, entonces envía una notificación por correo electrónico y por mensaje de texto a una dirección de correo electrónico y número móvil, los ajustes se muestran en la figura 3.27.

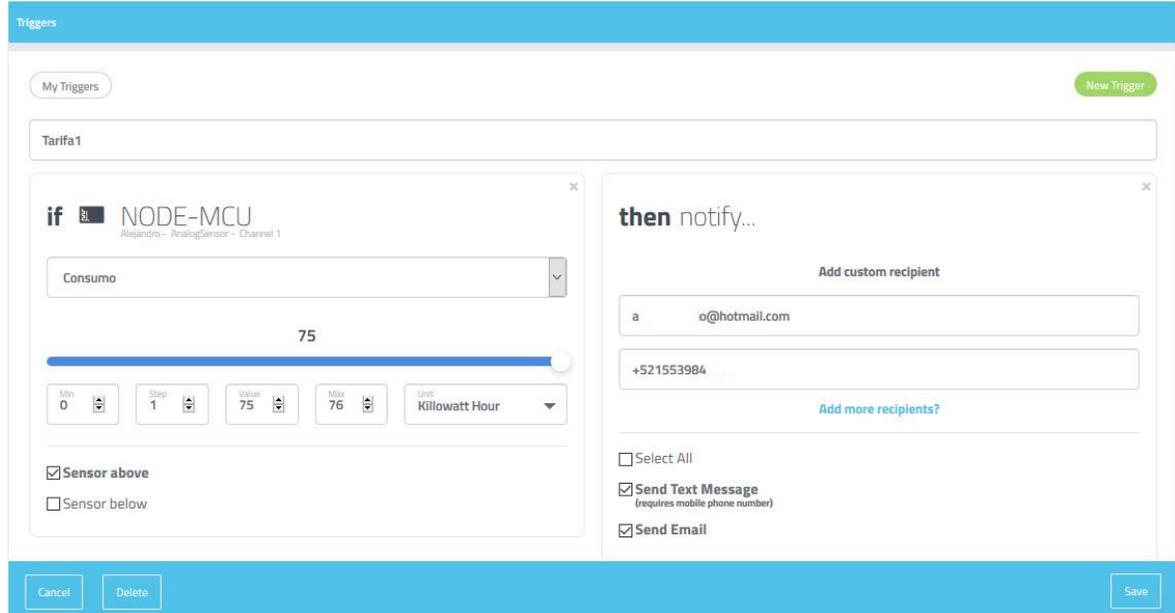


Figura 3.27 Configuración notificación tarifas.

Para las siguientes notificaciones se realizó el mismo procedimiento, solamente se modificó el intervalo y el valor que al rebasarlo activará la notificación. El siguiente rango, es de los 76 – 140, si rebasa 140, envía una notificación, y el último si llega a rebasar los 250, envía otra notificación.

### 3.6 Desarrollo y construcción del prototipo final

Para el desarrollo del medidor de consumo se realizaron varios prototipos, en los cuales se realizaron pruebas de funcionamiento, y conforme a los resultados obtenidos se realizaba un nuevo prototipo, hasta alcanzar el prototipo final.

El prototipo final se creó a partir de los resultados obtenidos de los experimentos de funcionamiento del primer y segundo prototipo, donde se consideraron los ajustes necesarios para el mejoramiento del diseño del sistema propuesto. Además, cabe señalar que, para el prototipo final, se desarrollaron dos propuestas, la primera, el medidor de consumo general, tiene una capacidad para soportar una potencia de



hasta 3750 W. El segundo, el medidor individual o por contacto, tiene una capacidad de potencia más reducida, soporta hasta 1200 W.

Para los prototipos finales, se consideró utilizar un tamaño definido para la anchura de las pistas según la cantidad de corriente máxima que debe soportar. Para ello, se estudiaron las reglas de diseño para los circuitos impresos, y mediante el uso de las ecuaciones que describen las dimensiones de las pistas según la cantidad de energía máxima. Estas ecuaciones consideran:

- La cantidad de corriente máxima que circulara por las pistas.
- La longitud de la pista.
- El grosor de la capa de cobre de la tarjeta fenólica que se está empleando.
- La temperatura máxima que debe soportar la pista y los componentes.

Por lo que se definió una cantidad máxima de corriente hasta 15 Amper y un límite de temperatura de 60 °C, la cantidad de cobre de la tarjeta, fue un dato del fabricante, la cual tiene un espesor de 35 micras de cobre o 1 onza/pie<sup>2</sup> [28]. Utilizando las ecuaciones para el cálculo del grosor de las pistas – véase el subcapítulo 2.8.2 de este trabajo – se obtuvo un valor de 11 mm para el ancho de las pistas. (Figura 3.28).



Inputs:

|           |    |      |
|-----------|----|------|
| Current   | 15 | Amps |
| Thickness | 35 | um   |

Optional Inputs:

|                     |    |       |
|---------------------|----|-------|
| Temperature Rise    | 60 | Deg C |
| Ambient Temperature | 25 | Deg C |
| Trace Length        | 15 | mm    |

Results for Internal Layers:

|                      |          |       |
|----------------------|----------|-------|
| Required Trace Width | 11.0     | mm    |
| Resistance           | 0.000815 | Ohms  |
| Voltage Drop         | 0.0122   | Volts |
| Power Loss           | 0.183    | Watts |

Figura 3.28. Resultados calculo ancho de pista.

Con el valor del ancho de la pista se procedió a rediseñar el diagrama esquemático del segundo prototipo, considerando los resultados obtenidos de los experimentos realizados al mismo.

Para los prototipos finales, se consideró incorporar una fuente de alimentación, que se alimente con 127 V<sub>AC</sub> y entregue 5 V<sub>DC</sub> y que incluya entre sus componentes un transformador a la entrada de alimentación. Se decidió emplear un producto comercial dado el funcionamiento, precio, espacio reducido que posee. La fuente seleccionada para incorporarla en el tercer prototipo, se describe su diagrama de componentes en la figura 3.29.

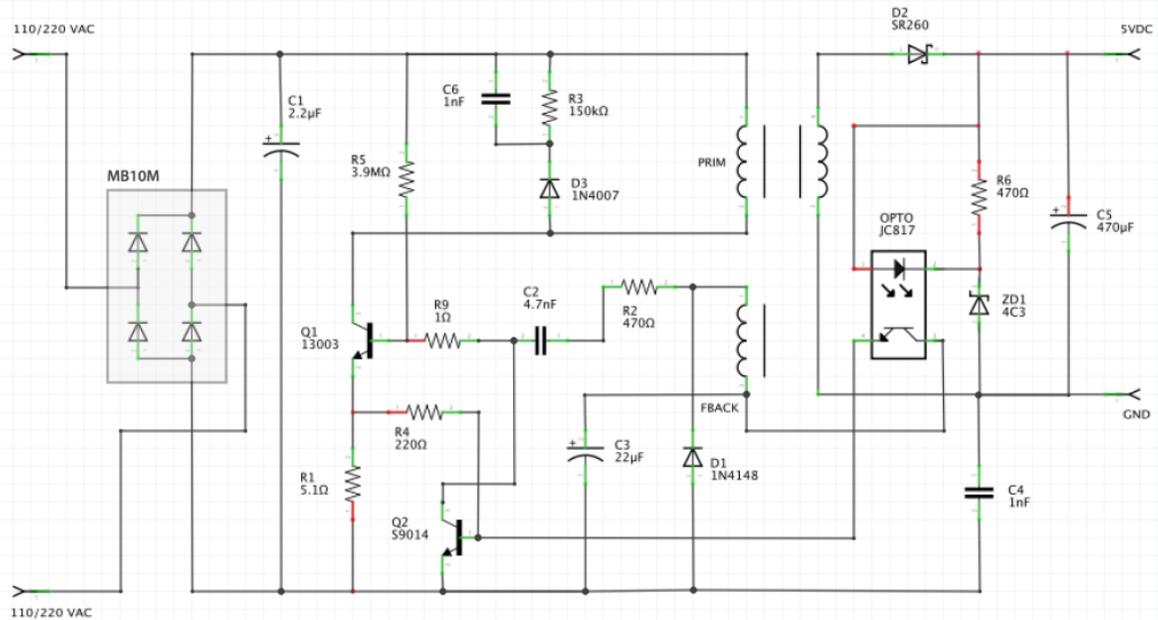


Figura 3.29. Fuente de alimentación comercial.

La fuente de alimentación se encuentra construida en dos tarjetas de circuito impreso, por lo que, para incorporarla en el tercer prototipo, solo se conectaron las terminales de alimentación y de salida de voltaje al diseño del prototipo final.

### 3.6.1 Medidor consumo general

Para el desarrollo del medidor de consumo general, se consideró tomar en cuenta para su diseño, el que sea capaz de soportar la demanda total de energía de una casa habitación completa, por lo que para su diseño y construcción fue fundamental esta característica. La creación de la versión final tomó como referencia los resultados obtenidos de los primeros prototipos creados.

Para el desarrollo del medidor de consumo general se diseñó un diagrama electrónico (figura 3.30) que muestra las conexiones realizadas entre los componentes del medidor.



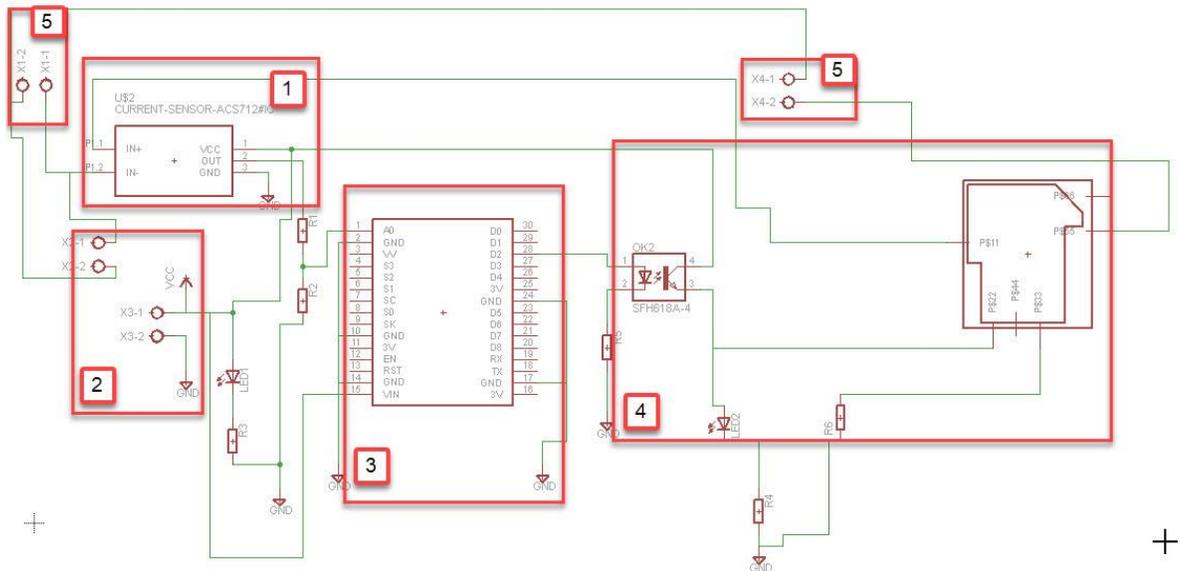


Figura 3.30 Diagrama medidor general.

Donde:

1. – Corresponde al sensor de corriente que cuantifica la corriente instantánea que se encuentra consumiendo la carga.
2. – Son las terminales de conexión para la fuente de alimentación con salida de 5 V<sub>DC</sub>, que es la encargada de alimentar al sistema de control, sensor de corriente y la activación del relevador.
3. – Es el sistema de control, está basado en el microcontrolador ESP8266 y es el encargado de realizar las funciones operativas y de comunicación del sistema.
4. – Acoplador y relevador, como acoplador, se colocó un dispositivo optoelectrónico que conecta la etapa de potencia con la etapa de control y las mantiene aisladas una de otra. Se utilizó un relevador como dispositivo de potencia para activar o desactivar la alimentación de la carga.





5. – Para las terminales de entrada y salida del sistema, se implementó tornillería de bronce, se adquirieron 4 tornillos con cuerda estándar con un grosor de  $\frac{1}{4}$  de pulgada – 6 mm de diámetro – y con una longitud de 1 pulgada – 2.54 cm de largo– Se seleccionaron tornillos de bronce debido a sus características dieléctricas.

### Construcción del circuito esquemático en Eagle

Del diseño del diagrama esquemático, se procedió a realizar el diseño en el entorno del programa Eagle. Para ello, se creó una nueva carpeta con nombre *medidor general*, en ella, se creó un nuevo archivo esquemático, se le dio el mismo nombre que a la carpeta que lo contiene. En él, se colocaron los componentes del medidor general, están descritos en la tabla 3.2.

Tabla 3.2 Componentes medidor general.

| Componente              | Cantidad |
|-------------------------|----------|
| Borneras dos terminales | 4        |
| NodeMCU                 | 1        |
| ACS – 712               | 1        |
| Relevador 30 A          | 1        |
| Resistor 1K $\Omega$    | 1        |
| Resistor 500 $\Omega$   | 1        |
| Resistor 330 $\Omega$   | 1        |
| Led                     | 1        |

Para el diseño del diagrama esquemático del sistema, se seleccionaron uno a uno los componentes con la herramienta  siguiendo el diagrama mostrado en la figura 3.31. El primer componente colocado fue la tarjeta NodeMCU, en la barra de búsqueda de componentes, se escribió el nombre de la tarjeta, y en los resultados se seleccionó la tarjeta y se colocó en la zona de trabajo. La tarjeta NodeMCU se colocó al centro del diseño como sistema de control. Los siguientes componentes fueron resistores el primero de 500  $\Omega$ , el segundo de 1k  $\Omega$  y el sensor de corriente ASC-712, se seleccionaron los componentes y se colocaron en la zona de trabajo.



Los siguientes componentes colocados fueron un acoplador y un relevador que soporta una potencia de 5700 W.

Como el relevador no se encontraba dentro de los componentes que incluye el entorno de Eagle, se creó la biblioteca del componente, con la herramienta de creación de bibliotecas. En el panel de control del entorno de Eagle, en la carpeta del proyecto, se creó una nueva biblioteca, se le nombro SLA-T90. En la ventana de creación, se seleccionó la vista de símbolo, en esta vista, se dibujó el símbolo del dispositivo con la herramienta de trazado – *draw* – se dibujó el contorno del componente, y se asignaron las terminales de conexión que tendrá el dispositivo, fueron 5 terminales (figura 3.31), la terminal P\$22 y P\$33, son los pines de alimentación de la bobina. Los pines P\$11 y P\$55 son las terminales de activación, entre ellas se abre o se cierra el interruptor. En P\$11 se conecta la alimentación y en P\$55 se conecta una terminal de alimentación de la carga.

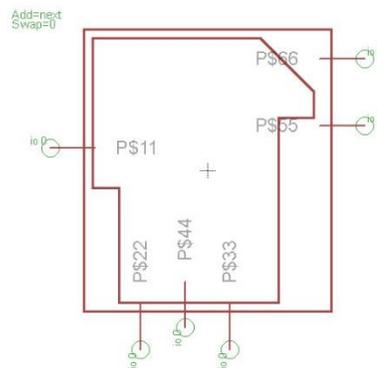


Figura 3.31 Símbolo relevador.

Con el símbolo diseñado, se realizó el diseño del paquete, el cual describe la distribución de los pines del componente. Se dibujó la silueta del relevador y mediante la información de los pines de la hoja de datos del fabricante [18] se colocaron en la ubicación adecuada, se ajustó el entorno para que la malla de la zona de trabajo estuviera escalada en 0.1 milímetros. La figura 3.32 muestra el diseño de la huella del componente.



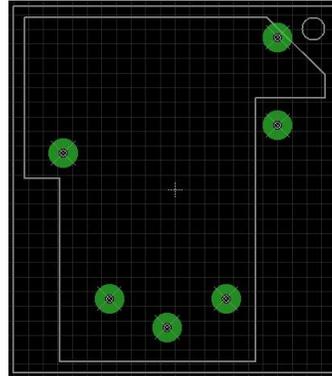


Figura 3.32 Huella relevador.

Una vez que se cuenta con los archivos de símbolo y de la huella del dispositivo, en la vista de *dispositivo*, se relacionan las terminales de conexión del símbolo, con los pines diseñados en la huella del dispositivo. De esta forma se asocian los pines y se completa la biblioteca. Una vez que se cuenta con la biblioteca, solo se agrega al entorno de Eagle, mediante la pestaña de *libraries*, en la opción *add*, se selecciona la ubicación del archivo, y se agrega al entorno, solo es cuestión de buscar el componente por su nombre SLA-T90, al aparecer, se agrega a la zona de trabajo, se colocó del lado derecho de la tarjeta NodeMCU. Los componentes agregados fueron, un diodo emisor de luz ubicado a un costado de la tarjeta, es el indicador que nos muestra cuando la tarjeta se encuentra energizada. Los últimos componentes agregados fueron 4 borneras con nombre AK300/2, 3 se colocaron a la izquierda de la tarjeta, pues es la sección donde se encuentra la alimentación del sistema. La bornera 1 corresponde a la alimentación con  $127 V_{CA RMS}$ , la segunda bornera se conectó en paralelo a la primera, se encarga de suministrar energía a la fuente de alimentación. La tercera bornera corresponde a las terminales de salida de la fuente de alimentación, las cuales suministran un voltaje de  $5 V_{DC}$ . La última bornera se colocó a la salida del sistema, en caso Una vez que se colocaron todos los componentes en la zona de trabajo, con la herramienta . Las primeras conexiones realizadas fueron las de alimentación. De la tercera bornera, de la



terminal 1, se conectaron las terminales “+” de alimentación de la tarjeta NodeMCU y el sensor de corriente. En la terminal 2 de la bornera, se conectaron las terminales comunes o “gnd” de la tarjeta, el sensor, el embobinado del relevador, y la resistencia con el led. Las siguientes conexiones realizadas fueron las de la tarjeta hacia los demás componentes, de la terminal A0, se conectó a un divisor de tensión realizado con las resistencias de 500  $\Omega$  y de 1k  $\Omega$ . La salida del sensor de corriente se conectó a la primera terminal de la resistencia de 500  $\Omega$ . De la terminal D2 de la tarjeta, se conectó a la terminal de activación del acoplador, y en la salida, se enchufó el embobinado del relevador.

### *Tarjeta de circuito impreso*

Con el diseño del diagrama esquemático realizado, se procedió a realizar el diseño de la tarjeta de circuito impreso.

Para ello, se exporto el diseño del circuito esquemático con la herramienta , en la nueva ventana del programa se configuró un tamaño de la zona de trabajo de 135 mm x 100 mm. Para redimensionar el área, con la herramienta  se seleccionó una esquina de la zona de trabajo, y con el botón secundario se accedió a las propiedades de la línea, y se le dio un tamaño de 100. Para la otra línea que forma la esquina se le dio un tamaño de 135 mm. Con la herramienta  y  se seleccionaron todos los componentes y se movieron dentro de la zona de trabajo. Una vez dentro, se colocaron los componentes a manera de que su ubicación coincidiera con la ubicación en el diagrama esquemático, cuidando que la orientación fuera la más adecuada, siendo las líneas amarillas las que definen la mejor orientación de los componentes, si las líneas, se cruzan, se re-orienta el componente, hasta que las líneas estén lo menos cruzadas posible. Para las pistas sobre las que viajan 127 V<sub>CA RMS</sub> se definió un tamaño de 11 mm para que pueda circular por ellas una corriente de hasta 30 Amperes. Estas pistas se colocaron en la parte superior de la tarjeta, para evitar que interfirieran con las demás pistas. Las



pistas por las que se aplican 5 V<sub>DC</sub>. Se definió para un tamaño de 1 mm, dado a que solo transportan señales digitales. El diseño de la tarjeta se muestra en la figura 3.33.

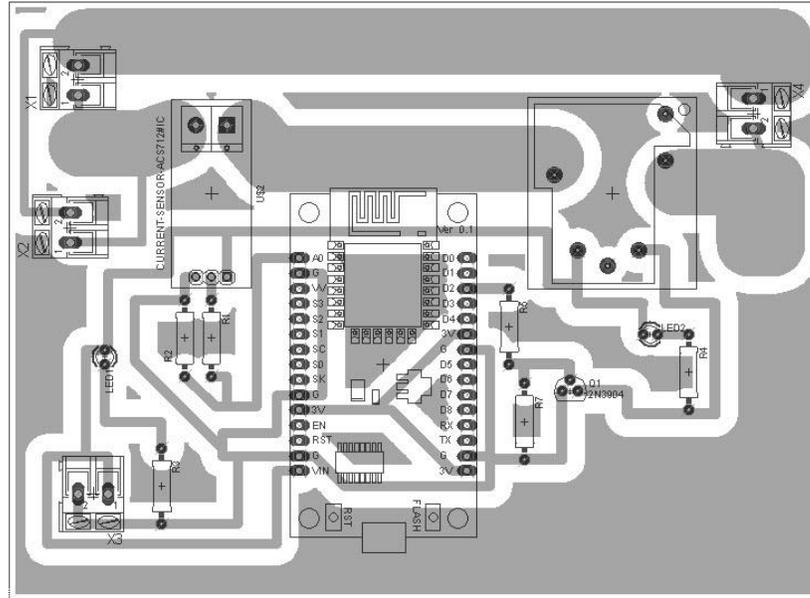


Figura 3.33 Tarjeta de circuito impreso medidor general.

### 3.6.2 Medidor consumo individual (contacto)

Para el desarrollo del medidor de consumo individual, se crearon y modificaron prototipos que cumplieran con características y funcionalidades específicas, con el fin de probarlas en un prototipo, evaluar su funcionamiento, y de ser necesario realizar los ajustes necesarios para mejorar las funciones del modelo.

Para el diseño de este prototipo, se creó un diagrama esquemático en el que retomaron los diseños de los prototipos previos y se incorporaron los nuevos componentes, quedando como se muestra en la figura 3.34.

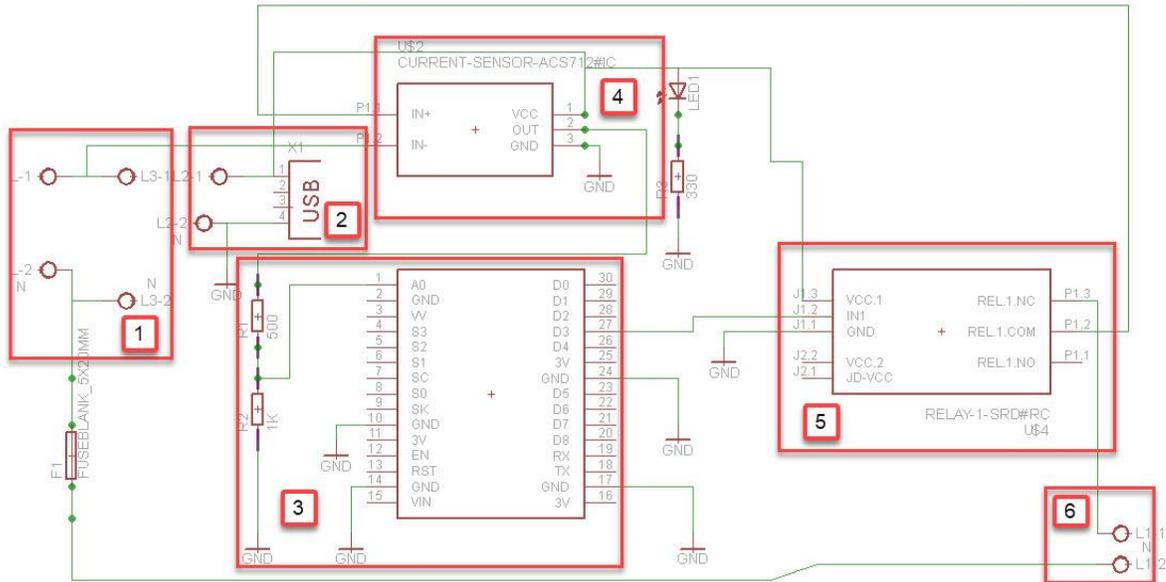


Figura 3.34. Diagrama esquemático medidor de consumo individual.

Donde:

1. Terminales de alimentación AC. En esta sección del diagrama se colocaron los bornes de conexión para la alimentación de la fuente, y la carga, en esta parte del circuito, se conecta el sistema a la toma de energía de la red eléctrica que posee un nivel de tensión de 120 V<sub>AC</sub>.
2. Terminales de alimentación DC. Las terminales corresponden a la salida de voltaje de la fuente de alimentación, donde se entrega un nivel de voltaje de 5 V<sub>DC</sub> el cual, alimenta a los componentes electrónicos y al sistema de control.
3. Sistema de control. Esta conformado por una tarjeta electrónica que tiene como microcontrolador un ESP-8266, el cual se encarga de realizar las tareas de procesamiento, comunicación y control que debe llevar a cabo el sistema.
4. Sensor de corriente. Es el encargado de cuantificar la corriente instantánea que se encuentra consumiendo la carga conectada al sistema. Esta conformado por el sensor de corriente ACS-712.





5. Etapa de potencia. Esta parte del sistema controla la activación o desactivación de la energía eléctrica con la que se alimenta la carga. Se realiza a través del uso de un módulo con relevador electromecánico que soporta hasta 20 A.

6. Terminales de salida. En estas terminales se conecta la entrada de alimentación de la carga, con el fin de poder cuantificar el consumo y controlar la alimentación de la misma las borneras indican la zona de contacto para la alimentación del sistema y de la carga.

### Construcción del circuito esquemático en Eagle

De los resultados obtenidos de las pruebas aplicadas al segundo prototipo, se incluyeron arreglos y mejoras al diseño del actual prototipo. Por lo tanto, tomando como referencia y base el diseño del segundo prototipo, se realizaron cambios que hicieron necesario que se modificara el diagrama, pues se colocaron nuevos componentes y se reemplazaron otros, los componentes actuales se muestran en la tabla 3.3.

Tabla 3.3. Componentes tercer prototipo.

| Componente              | Cantidad |
|-------------------------|----------|
| Conector hembra USB A   | 1        |
| Borneras dos terminales | 2        |
| NodeMCU                 | 1        |
| ACS – 712               | 1        |
| Módulo con relevador    | 1        |
| Fusible 15 A            | 1        |
| Resistor 1K $\Omega$    | 1        |
| Resistor 500 $\Omega$   | 1        |
| Resistor 330 $\Omega$   | 1        |
| Led                     | 1        |

Para comenzar con el armado del prototipo, se creó un nuevo archivo *schematic*, donde se comenzó a colocar en la zona de trabajo los componentes que se van a emplear para el tercer prototipo, para ello, con la herramienta  se agregaron uno





a uno a la zona de trabajo, acomodandolos con respecto al diagrama diseñado. Los primeros componentes colocados, fueron las borneras de conexión para la entrada de alimentación de AC. Las borneras empleadas tienen como nombre en la biblioteca de Eagle AK300/2. Se colocaron cuatro borneras, dos a la entrada para alimentación en alterna y otra para la salida, para conectar la carga. La última, se colocó como terminales de alimentación en directa. Los siguientes componentes agregados, fueron los resistores, se colocaron tres resistores, los seleccionados fueron los que la huella mide 10 mm. Después se colocó un diodo emisor de luz, seguido de una resistencia. También se colocó un fusible, a la entrada de la alimentación en AC de la carga.

El siguiente componente, fue la tarjeta NodeMCU, para colocarlo, se utilizó la herramienta  y se colocó el nombre *node* en la casilla de búsqueda. Una vez que apareció el componente se agregó a la zona de trabajo, colocándolo a continuación de la bornera de alimentación. Con la tecla de escape del teclado, se suelta el componente que se había seleccionado, y se vuelve a abrir la ventana para agregar otro componente. El siguiente componente fue el sensor de corriente, ACS-712, se buscó en la barra de búsqueda de componentes, y se colocó arriba de la tarjeta, siguiendo el orden del diagrama. El último componente, fue el módulo con relevador, se buscó el componente en la ventana de agregar componentes y se colocó a la izquierda de la tarjeta NodeMCU.

Una vez que se colocaron todos los componentes, se procedió a realizar las conexiones de las terminales de los mismos. Para ello se utilizó la herramienta . Las conexiones se realizaron siguiendo el diseño del diagrama esquemático. Las primeras conexiones realizadas fueron las de alimentación alterna, que se conectan a la primera bornera de la entrada, al sensor de corriente cuya conexión se realiza en serie, enseguida se conectó la terminal común del módulo relevador y por último la bornera de salida. Las siguientes conexiones realizadas fueron las del sistema de



control hacia los demas componenetes, a la terminal A0 del la tarjeta NodeMCU, se conectó la salida del divisor de tension de las dos resistencias que se encuentran debajo del sensor de corriente, la salida de este sensor, se conectó a la primera resistencia del divisor, y la terminal libre de la segunda resistencia se conectó a tierra. Posteriormente, se conectó la terminal D3 de la tarjeta NodeMCU a la terminal IN1 del módulo relevador. Las ultimas conexiones fueron las de alimentación en directa, para ello, de la tercera bornera, a la terminal superior se conectaron las terminales de alimentacion VCC, y a la terminal inferior de la bornera, se conectó a GND. Cabe mencionar, que para no realizar tantas lineas de conexión, se utilizaron en eagle, los componentes GND, los cuales son puntos que indican que la terminal conectada a ellos se conecta a tierra.

#### *Tarjeta de circuito impreso*

Una vez que todas las conexiones estuvieron realizadas en el archivo *schematic*, se almaceó el trabajo realizado, y se exportó el diseño a un nuevo archivo *board*, utilizando la herramienta . Al abrirse en nuevo archivo, aparecieron las huellas de todos los componentes colocados en el *schematic*, con las conexiones realizadas entre las terminales.

El primer paso para el diseño de la PCB fue seleccionar todos los componentes para colocarlos dentro de la zona de trabajo, para ello, primero se seleccionó la herramienta  y enseguida la herramienta , con ello, se arrastró el puntero a manera que todos los componentes quedaran dentro de la zona, para despues moverlos todos al mismo tiempo hacia la zona de trabajo.

Los componentes se acomodaron a manera que quedaran lo mas parecidos al diagrama esquematico, por lo que se dejó el sistema de control en el centro, el sensor de corriente se colocó en la parte superior de la zona de trabajo, el módulo con relevador, se ubico hacia la derecha del la tarjeta NodeMCU. Las borneras se colocaron tres en la parte izquierda del área de trabajo, donde, dos de ellas, en las



que se conecta la alimentación alterna, se colocaron más hacia el borde, y la tercera, donde se conecta la alimentación directa, se conectó más hacia el centro de la tarjeta, dado a que son las terminales donde se alimentan los componentes. La última bornera, se colocó en la parte derecha de la zona de trabajo, pues es donde se conectará la carga. Los siguientes componentes a colocar, fueron el fusible, que se colocó en la parte izquierda de la zona de trabajo, bajo las borneras de alimentación alterna, las dos resistencias que conforman el divisor de tensión, se colocaron del lado izquierdo de la tarjeta NodeMCU, el diodo emisor de luz y el último resistor se conectaron del lado derecho de la tarjeta NodeMCU.

Para realizar el trazado de las pistas, en la configuración de reglas del diseño de la tarjeta de circuito impreso, en la pestaña de *sizes*, se definió un tamaño de 1.27 mm para el ancho de las pistas. Con el ancho establecido, se procedió a iniciar el asistente de autoruta, presionando el icono , una vez que apareció la ventana de configuración de autoruta, se configuró para que el trazado lo realizará en la capa *bottom* y con un espacio de malla para el trazado de las pistas de 0.3 mm. Una vez insertados los valores, se ejecuta el asistente de autoruta, dejando que realice el trazado de las pistas de manera automática. Una vez que finalizó, se procedió a incrementar de manera manual, el tamaño del ancho de las pistas de la alimentación alterna, dado a los resultados obtenidos de los experimentos realizados al segundo prototipo, mostraron que era necesario incrementar la anchura de las pistas, para que puedan soportar la cantidad de energía que debe pasar por ellas, aproximadamente 10 Amper. Por lo que se tuvo que seleccionar, pista por pista, y cambiar de manera manual la propiedad de *width*, dejando un valor de 11 mm. Las pistas que se ampliaron, fueron las que se conectan con las borneras de alimentación alterna, el sensor de corriente, el módulo con relé y la bornera de salida. El resultado de la ampliación se muestra en la figura 3.35.

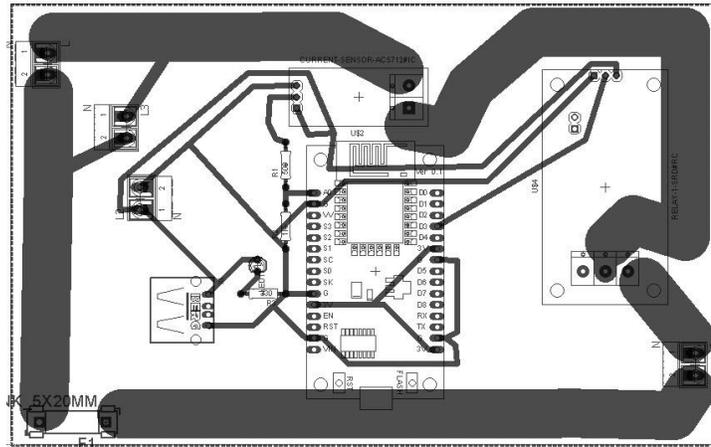


Figura 3.35. Diseño tarjeta de circuito impreso medidor individual.

Con el diseño de la tarjeta de circuito impreso, se procedió a transferir el diseño a una placa fenolica, por lo que se imprimió el diseño en una hoja de papel transfer con la ayuda de una impresora láser. Para poder transferir el diseño a la placa fenolica, primero se limpió la superficie de la tarjeta fenolica, utilizando una lija suave, a manera que quedara libre de imperfecciones, posteriormente se limpió la superficie de la tarjeta con un trapo y un poco de alcohol isopropilico, para realizar la transferencia fue necesario utilizar una plancha para ropa, colocada en la temperatura maxima, para ello, se colocó la tarjeta fenolica con la cara de cobre hacia arriba, para poder colocar la impresión encima de la cara, se fijo con cinta adhesiva para que no se movieran, y se colocó una franela encima para que el calor directo de la plancha no quemara el papel. Se colocó la plancha encima de la franela, aplicando fuerza contra la tarjeta se mantuvo un planchado constante alrededor de 10 minutos. Se retiró la plancha y la franela, dejando enfriar la tarjeta, para poderla sujetar y revisar el estado de la transferencia. Se transfirió completamente el diseño a la tarjeta fenolica, por lo que se sumergió la tarjeta en cloruro ferrico, dejando que la solución eliminara el cobre no deseado. El resultado de la transferencia y el revelado, se muestra en la figura 3.36.

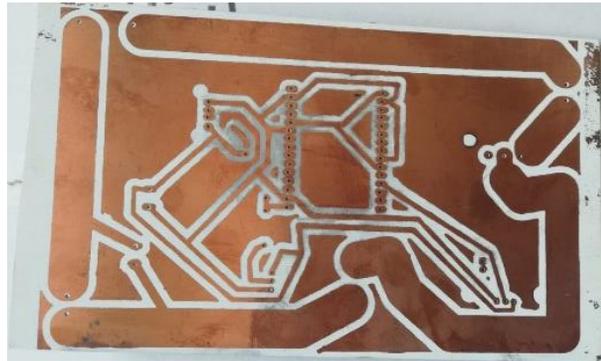


Figura 3.36. Tarjeta medidora de consumo individual.

Con la tarjeta revelada, el siguiente paso fue realizar perforaciones en cada pad donde se debe insertar la terminal de un componente. Para ello, con ayuda de un taladro y una base para mantenerlo verticalmente, se hicieron las perforaciones debidas. Para los componentes pasivos, se utilizó una broca de 0.8 mm. Una vez que estuvieron hechas las perforaciones, en la cara blanca de la tarjeta fenólica, con ayuda de un plumón indeleble se indicó donde corresponde cada componente (figura 3.37).

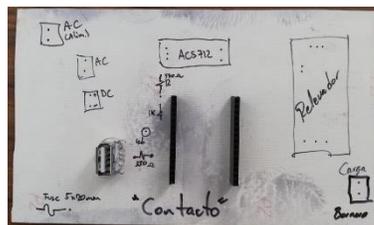


Figura 3.37. Posición de componentes.

Para fijar los componentes a la tarjeta, se colocaron las terminales de los componentes en su lugar respectivo, con ayuda de segmentos de cinta adhesiva, se mantuvieron los componentes lo más fijo posible para girar la tarjeta, y del lado del cobre, estañar las terminales y fijarlas al cobre. Una vez que todos los componentes fueron soldados a la tarjeta, realizaron pruebas con la misma.



### 3.7 Diseño e impresión 3D del contenedor

Para mantener aislado el sistema de la intemperie y las condiciones del exterior, se diseñó una carcasa que lo cubre del contacto con el exterior. La carcasa se diseñó utilizando SolidWorks, es un software para el diseño de objetos tridimensionales.

#### 3.7.1 Contenedor medidor general

Para la creación del contenedor, se tomó como referencia una caja que se pareciera a la caja de fusibles que se encuentra en la zona de control de cargas eléctricas.

Así que, tomando como modelo un contenedor cuadrado, se procedió a realizar el diseño del contenedor en un nuevo archivo en el programa SolidWorks.

Para la base del contenedor se selecciona la vista de planta como referencia para la creación de un croquis, y con la herramienta de dibujo de rectángulo – que se encuentra en barra de herramientas, pestaña de croquis – se trazó un rectángulo con dimensiones de 160 mm x 120 mm, con la herramienta de cota inteligente se ajustó la altura a 160 mm cerrados, lo mismo se realizó para el ancho. Con la herramienta de extruir desde base – se encuentra en la pestaña de operaciones – se le dio una altura – con respecto al eje z – de 45 mm. Se formó un prisma rectangular como se muestra en la figura 3.38.

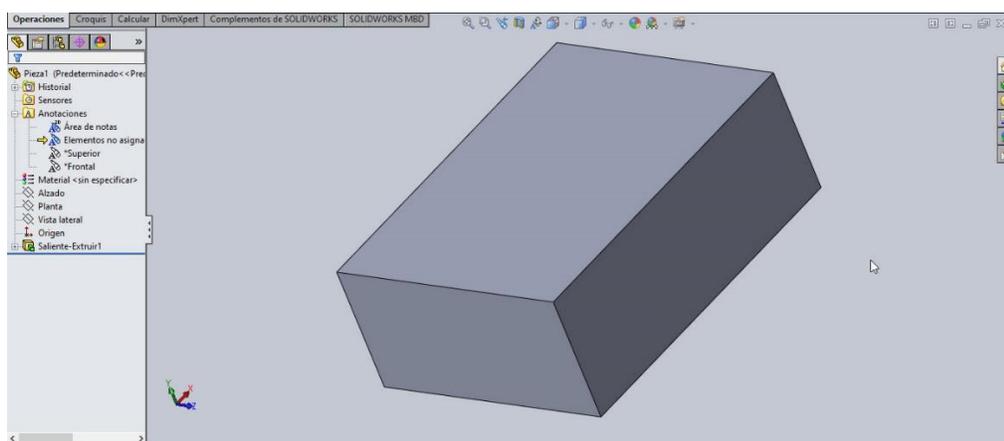


Figura 3.38 Prisma rectangular.



Para la hacer el vaciado de la caja, se seleccionó la superficie superior del prisma y con la herramienta de croquis se dibujó un nuevo rectángulo con dimensiones de 154 mm x 114 mm, con la herramienta de cota, se colocaron los valores cerrados para cada lado del rectángulo. Con la herramienta de extruir corte se realizó un corte con las dimensiones del nuevo rectángulo, y se configuro una profundidad de corte de 42 mm. La caja resultante se muestra en la figura 3.39.

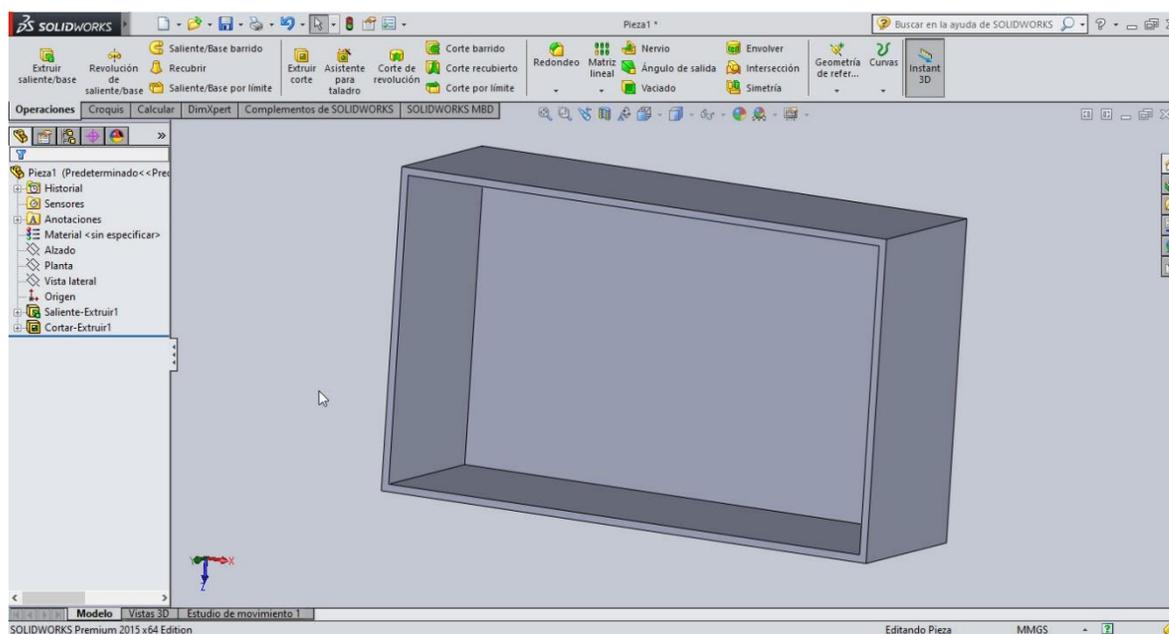


Figura 3.39 Caja.

Para colocar la tarjeta del circuito, se colocaron cuatro postes circulares con un diámetro de 8 mm. Se colocaron en los centros de cada lado de un rectángulo con dimensiones 105 mm x 140 mm, el rectángulo es concéntrico a la caja mostrada en la figura 3.25. Para realizar el dibujo del rectángulo, se seleccionó la superficie inferior interior de la caja, se dibujó el rectángulo creando un nuevo croquis y con la herramienta para dibujar rectángulos con el centro y una de sus esquinas, se realizó el trazado, con la herramienta de cota, se estableció el tamaño exacto de los lados del rectángulo. Para trazar los puntos medios de cada lado del rectángulo, se trazó con la herramienta de línea, desde una esquina del rectángulo hasta el punto medio,





el cual es sencillo de hallar, gracias al asistente del programa, nos muestra de manera automática donde se encuentra el punto medio. El procedimiento anterior se realizó para cada lado del rectángulo. Con los centros de cada lado, se trazó una circunferencia con la herramienta círculo con centro y diámetro. Con la herramienta extruir desde base, se seleccionaron los 4 círculos, y se estableció una altura de 6 mm. Para terminar con el diseño de los postes, solo hace falta realizar el trazado del conducto por el que se insertara el tornillo que sujetara a la tarjeta, para ello, se seleccionó la superficie de superior de un poste, y con un croquis, se trazó un nuevo círculo con diámetro de 3 mm, y con la herramienta de extruir corte, se realizó un corte con profundidad de 4 mm, el procedimiento se realizó para cada poste. Los postes creados se muestran en la figura 3.40.

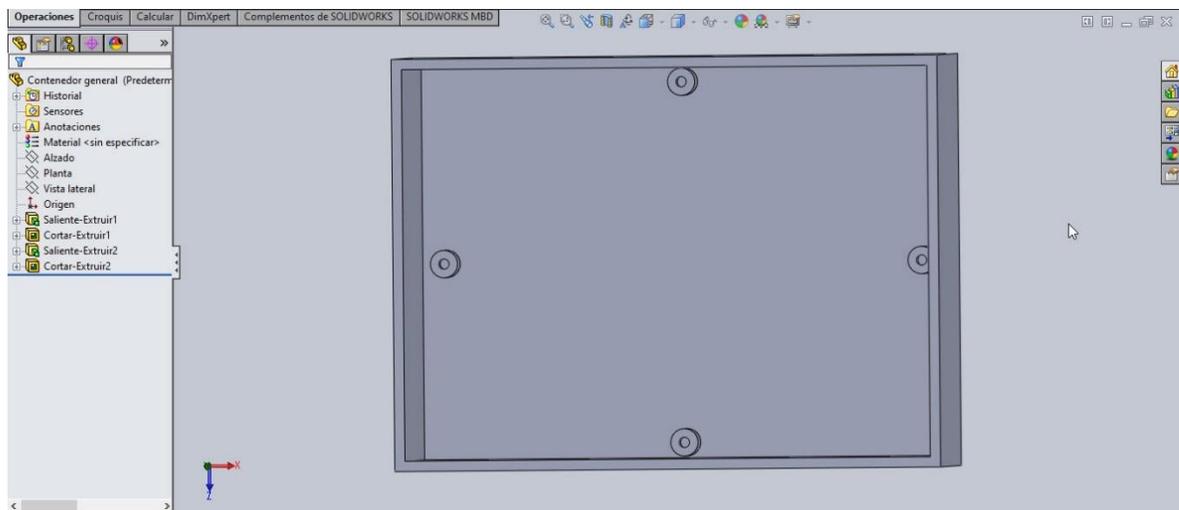


Figura 3.40 Postes para colocar la tarjeta.

Para poder realizar las conexiones del medidor al sistema eléctrico, se realizó un corte de la pared derecha de la caja, el corte realizado tuvo un tamaño de 12 mm x 7 mm, mediante la herramienta de croquis, se seleccionó la pared derecha del modelo, o bien, la cara que se encuentra seleccionando el eje x, con la herramienta de línea, se trazó un segmento vertical – paralelo al eje z – de 10 mm, y en la punta de la primer línea, se dibujó una horizontal – paralela el eje y – de 25 mm, y en el





punto final de la línea, se trazó el área para el corte, con la herramienta de dibujo de un rectángulo con una esquina y el centro, se trazó un rectángulo de 7 mm x 12 mm. Con la herramienta de extrusión de corte, se realizó un corte con una profundidad de 5 mm, para que atravesara la pared.

El mismo proceso fue repetido para el lado izquierdo de la caja, solo que la segunda línea, tuvo una longitud de 20 mm. La figura 3.41 muestra las aperturas para el ingreso de los conductores al sistema.

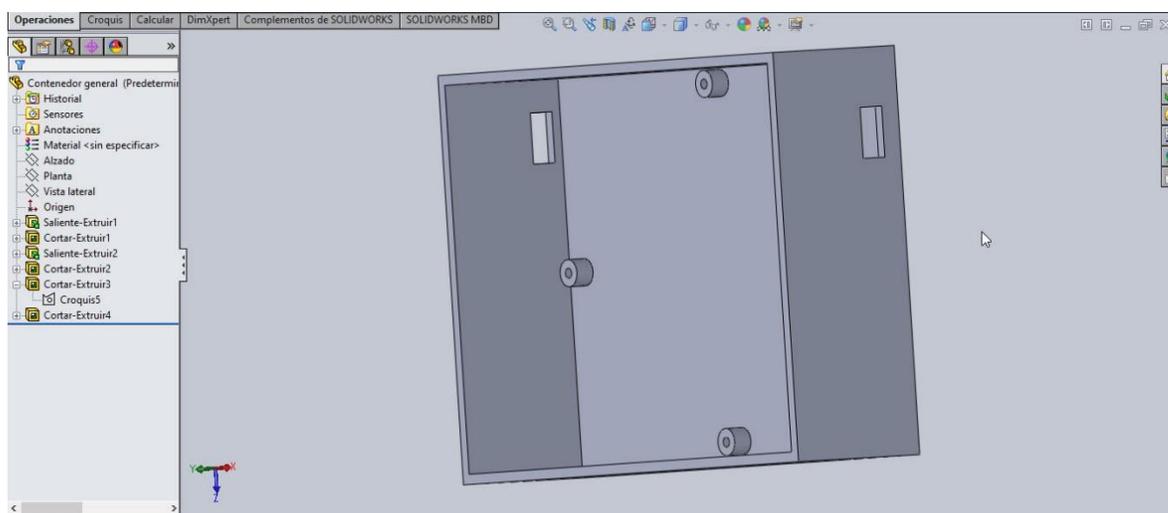


Figura 3.41 Aperturas para ingresar los conductores.

Para poder colocar una tapa, y que esta cierre al contenedor, se diseñaron 4 soportes para que la tapa de la caja por fuera se pueda atornillar al contenedor, los postes se realizaron de la siguiente manera: primero se seleccionó la superficie inferior de la caja, como superficie para la creación de un nuevo croquis, en el que se dibujaron 4 cuadrados con la herramienta de rectángulo con dos esquinas, y se dibujaron tomando como una esquina la esquina de la caja, los cuadrados tuvieron una longitud de 8 mm, para establecer el valor exacto, se utilizó la herramienta de cota inteligente. Con la herramienta de extrusión desde base, se les dio una altura – con respecto al eje z – de 40 mm. Ahora para hacer los orificios para los tornillos de la tapa se seleccionó la superficie superior de los soportes, y con un croquis y la



herramienta para dibujar círculos, se trazó un círculo perimetral, seleccionando tres lados del cuadrado, y dentro de ese círculo, se trazó uno más desde el centro, con diámetro de 3 mm, se repitió el mismo proceso con los otros tres soportes, y con la herramienta de extrusión de corte, se realizó un corte con una profundidad de 5 mm. En la figura 3.42, se muestra el recipiente para el sistema.

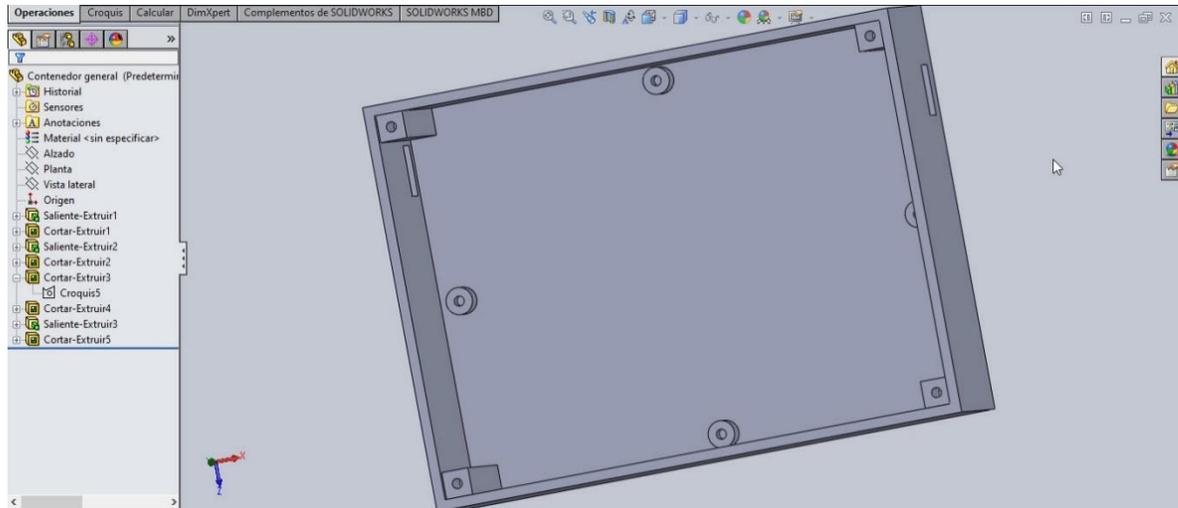


Figura 3.42 Contenedor del medidor general.

Para el diseño de la tapa de la caja fue más simple, una tapa con una muesca que coincide con la altura de los soportes para la caja y 4 orificios por los que pasan los tornillos. Para la construcción de la tapa, se creó un nuevo archivo en donde se tomó la vista de planta como referencia para crear un nuevo croquis, y se dibujó un rectángulo con dimensiones de 160 mm x 120 mm, y con la herramienta de extrusión desde base, se le dio una altura de 2 mm. Se tomó la superficie superior y se trazó otro rectángulo con dimensiones de 154 mm x 114 mm y con la herramienta de extrusión desde base, se le dio una altura de 2 mm. Para los orificios, se trazaron cuadrados de 6 mm en cada esquina del segundo rectángulo, y mediante el dibujo de círculos de 3 mm de diámetro dentro de los cuadrados se realizó la extrusión de corte, dejando así los espacios para que los tornillos pasen.

La figura 3.43 muestra la tapa del contenedor para el medidor general terminada.

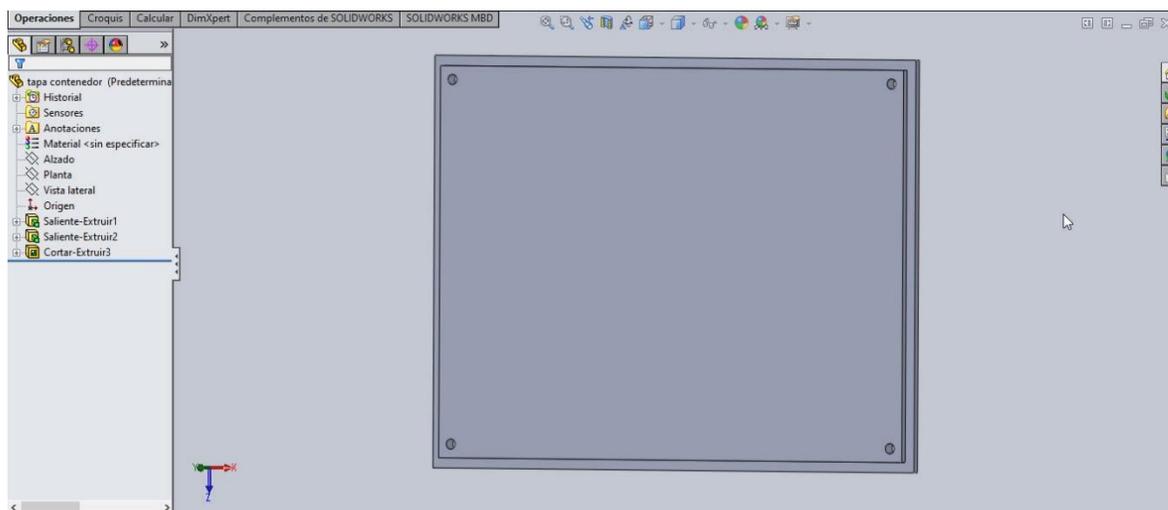


Figura 3.43 Tapa contenedor medidor general.

### 3.7.2 Contenedor medidor individual

Para crear el modelo de la carcasa, se tomó como modelo una caja de zapatos, una versión de un prisma rectangular hueco. Para poder llevarlo a cabo, se inició el programa de SolidWorks, y se creó un nuevo archivo de pieza. Para comenzar a dibujar el prisma, primero se seleccionó la *vista de planta*, haciendo doble clic en el icono de planta en el panel de propiedades de la pieza, enseguida en la barra de herramientas se seleccionó *croquis* y la herramienta de dibujo de prismas, se dibujó con respecto a la planta un rectángulo, las dimensiones fueron 105 mm x 175 mm, con la herramienta de cota inteligente, se seleccionó el lado más largo del rectángulo dibujado y se ingresó el valor de 175 mm. Se realizó el mismo procedimiento para el lado más corto. Una vez que quedo definido el rectángulo, en la pestaña de operaciones, se seleccionó la herramienta, Extruir saliente/ base, y se tomó como referencia el rectángulo dibujado, se hizo una extrusión de 55 mm, por lo que la figura resultante quedo como se muestra en la figura 3.44



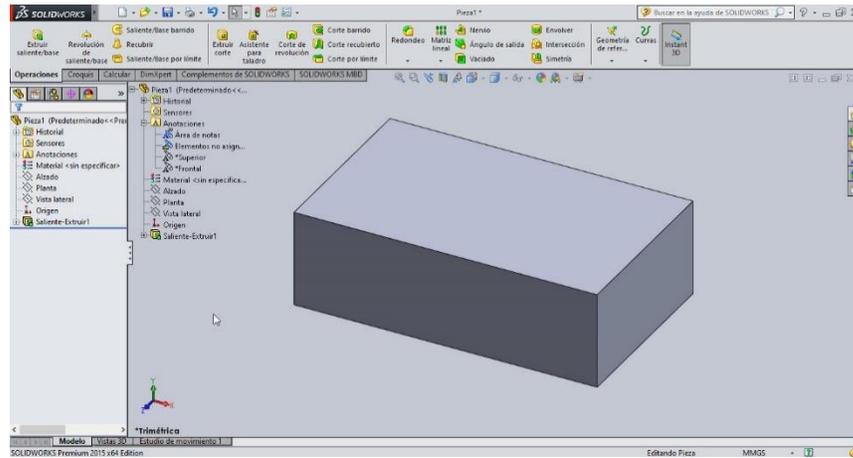


Figura 3.44. Prisma rectangular.

Se seleccionó la cara superior del prisma, se creó un nuevo *croquis*, con la herramienta de dibujo de prismas, se dibujó otro rectángulo, pero este de dimensiones de 100 mm x 170 mm, en la pestaña de operaciones con la herramienta extruir corte, se tomó como referencia el segundo rectángulo, y se le dio al corte una profundidad de 52mm, quedando el prisma como se muestra en la figura 3.45.

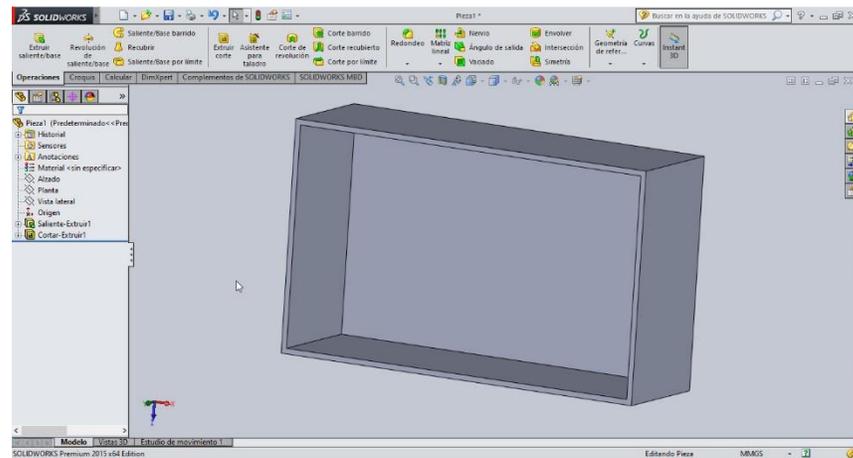


Figura 3.45. Prisma rectangular hueco.

Una vez que se cuenta con la caja, fue necesario realizar los cortes en las paredes de la caja por donde quedara la terminal para la alimentación del sistema una clavija





de conexión rápida y un enchufe eléctrico normal. Se obtuvieron las dimensiones de la clavija, y del contacto. Para la clavija de acceso rápido, se tomaron las dimensiones de 25 mm x 55 mm, y para el enchufe fueron de 30 mm x 15 mm. Para realizar los cortes, se seleccionaron las caras más cortas de la caja y tomando como referencia la cara externa de los lados más cortos. Para la clavija de acceso rápido, se dibujó mediante un croquis en una de las caras, un rectángulo de 25 mm x 55 mm, para que las dimensiones fueran exactas se utilizó la herramienta de cota inteligente para ingresar los valores deseados. Una vez que el rectángulo estuvo listo, en el panel de operaciones, con la herramienta de extruir corte, se hizo un corte con respecto al rectángulo, con una profundidad de 5 mm (figura 3.46).

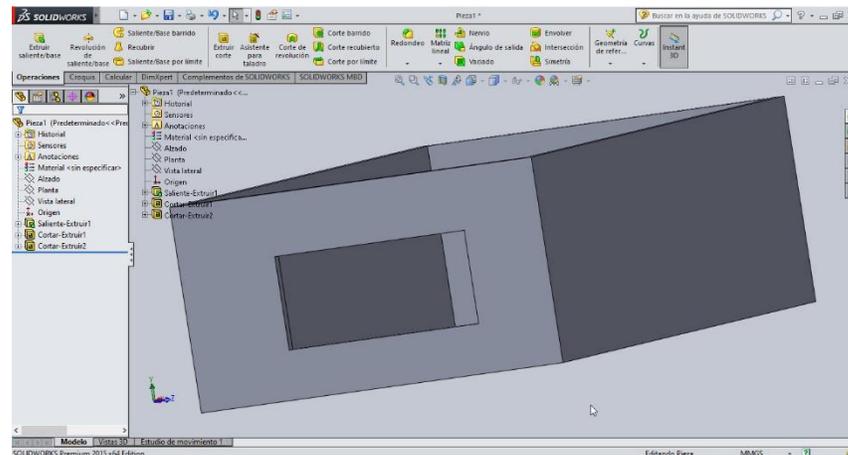


Figura 3.46. Orificio para clavija de acceso rápido.

El mismo procedimiento se realizó en la otra cara corta de la caja, siendo el rectángulo resultante más pequeño (figura 3.47).



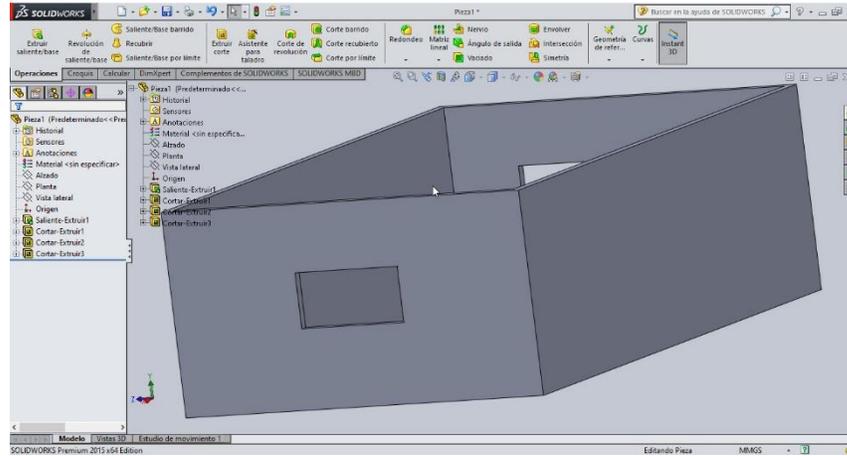


Figura 3.47. Orificio enchufe.

Una vez que estuvo listo el diseño del modelo tridimensional, se guardó el archivo de la pieza en formato .stl para que pueda ser interpretado por el software Cura para rebanar sólidos.

Una vez que se tuvo listo el archivo, se abrió el programa Cura 3.6.0. y se cargó al programa, arrastrando el archivo desde una ventana en el explorador de archivos hasta el programa, al cargarlo, este apareció en la superficie de impresión de la impresora (figura 3.48).

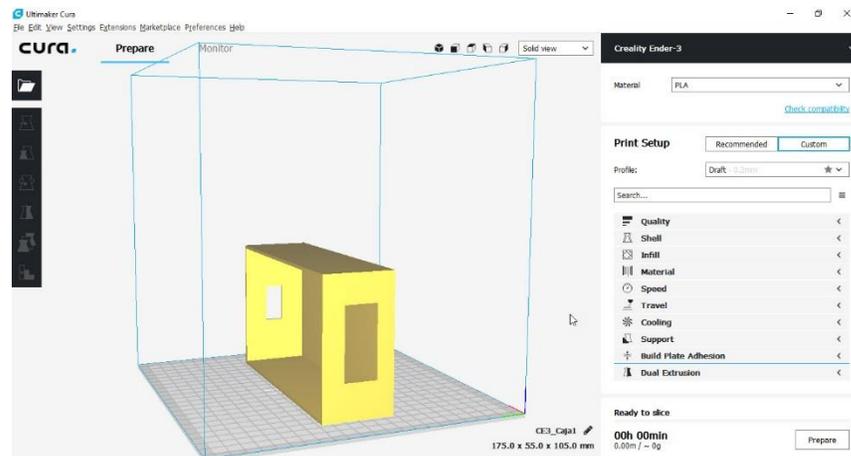


Figura 3.48 Modelo tridimensional en Cura.





Para poder disminuir el tiempo de impresión y ahorrar un poco de material, se acomodó el modelo para que fuera más eficiente la impresión del mismo, por lo que se roto el modelo 90° hacia la derecha, para moverlo, se seleccionó el modelo, y con la herramienta de giro se roto, hasta quedar como se muestra en la ilustración 3.49.

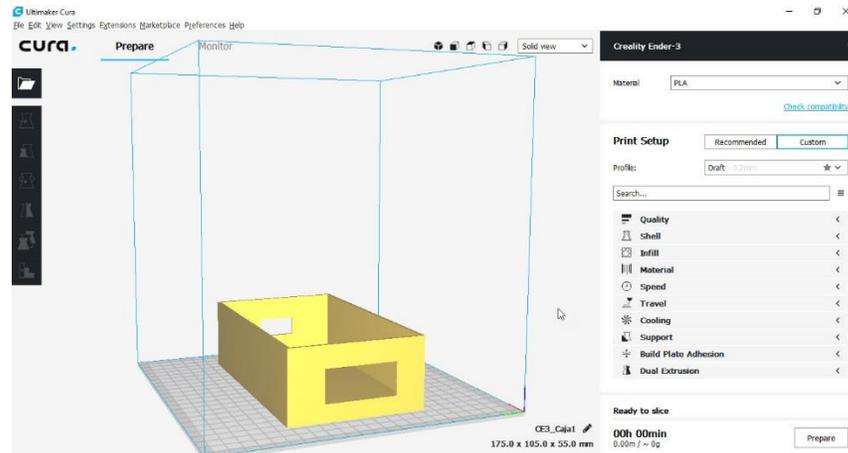


Figura 3.49. Acomodo del modelo tridimensional.

Una vez que esta acomodado el modelo de la manera más eficiente, se presiona el botón de preparar y se espera a que el software termine de hacer el rebanado del modelo para observar el estimado del tiempo que tardara en realizarse la impresión. Considerando ajustes de calidad media, donde cada capa tiene un tamaño de 0.2 mm.

El tiempo que tardó el modelo en imprimirse fue alrededor de 8 horas 40 minutos.

Para poder imprimir el modelo, se guardó el archivo generado por Cura en una memoria microSD, y se colocó la memoria en la ranura de la impresora Anycubic i3 mega. Se estableció una temperatura de boquilla de 190 °C para poder cargar el filamento hasta la boquilla y que este comenzará a salir ya derretido. Con el filamento cargado, se limpió la cama utilizando alcohol isopropílico y se seleccionó el modelo de la caja para que comenzara la impresión.





El resultado final del modelo diseñado e impreso y colocado en el sistema de la presente propuesta, se muestra en la figura 3.50 el medidor individual. En la 3.51 se muestra el medidor general.

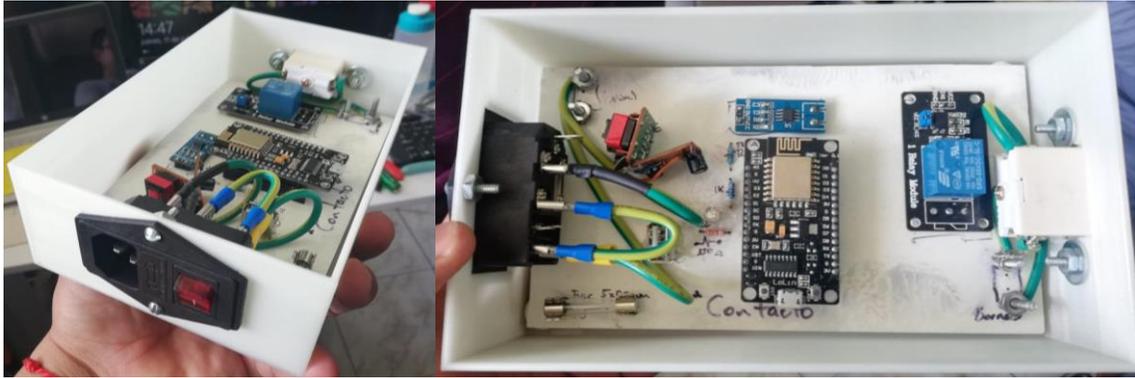


Figura 3.50 Sistema montado en el contenedor.



Figura 3.51. Sistema medidor general montado en la carcasa.



## Capítulo 4 Pruebas y resultados

Las pruebas realizadas tuvieron como objetivo comprobar el correcto funcionamiento de una o más funciones del sistema propuesto.

### 4.1 Prueba convertidor analógico digital.

La prueba consistió en comprobar el funcionamiento del ADC, donde se evaluó, la resolución máxima que alcanza, así como el voltaje máximo de lectura.

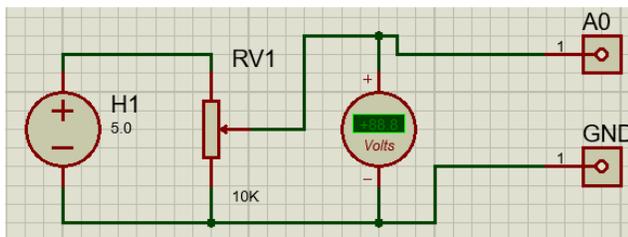


Figura 4.1. Diagrama conexiones ADC

El procedimiento realizado consistió en:

- Conectar a una fuente de 5V, una resistencia variable de carga, un voltímetro y el Pin A0 de la tarjeta electrónica, el diagrama mostrado en la figura 4.1, describe las conexiones realizadas.
- A la tarjeta se cargó un programa explicado en la figura 4.2, donde, en resumen, toma una lectura del voltaje del resistor y entrega un valor numérico correspondiente a la intensidad de voltaje.

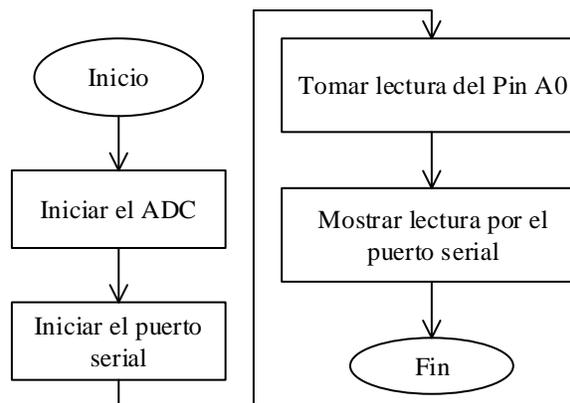


Figura 4.2. Diagrama de flujo prueba ADC.

- Se encendió la fuente, se conectaron las terminales de la fuente en las terminales del potenciómetro. El resistor variable se colocó en su valor más alto de resistencia, se tomó lectura del valor del voltímetro y del resultado del ADC.
- Los resultados obtenidos se muestran en la tabla 4.1

### Resultados de la prueba con el ADC

Los resultados obtenidos del experimento se muestran en la tabla 4.1, donde se puede apreciar que el valor máximo al que llega la resolución del ADC es hasta 1023, por lo que, si los valores van de 0 a 1023, quiere decir que existen 1024 valores, por lo tanto, el ADC, posee una resolución de 10 bits. En cuanto al voltaje de operación, al pasar los 3.5 V, el valor del ADC no se modificó, continuó siendo 1023, por lo que el voltaje máximo que logra medir, se encuentra entre 3.0 y 3.5 volts. Continuando con la prueba, se modificó el valor de la resistencia variable, para que pasara entre 3.0 y 3.5 volts, y se encontró, que el voltaje máximo donde se alcanza el valor 1023 es a los 3.3 volts.



Tabla 4.1. Resultados prueba ADC.

| Lectura ADC | Valor voltímetro (Volts) |
|-------------|--------------------------|
| 0           | 0                        |
| 310         | 1.0                      |
| 465         | 1.5                      |
| 621         | 2.0                      |
| 776         | 2.5                      |
| 931         | 3.0                      |
| 1023        | 3.5                      |
| 1023        | 4.0                      |

### Prueba de divisor de tensión

Se diseñó un divisor de tensión para acoplar los voltajes de la salida del sensor Hall, y del ADC, a manera de que se mantengan en un rango que vaya de 0V a 3.3 V.

La prueba consistió en:

- Conectar a una fuente variable de directa de 5 volts, el divisor de tensión diseñado.
- Conectar un voltímetro a la salida de voltaje del divisor de tensión.
- Realizar variaciones en el voltaje de salida de la fuente, menores a 5 Volts.
- Anotar la lectura del voltaje de la fuente y del valor obtenido por el voltímetro.

### Resultados prueba divisor de tensión.

Los resultados obtenidos se muestran en la tabla 4.2, donde se puede apreciar que el divisor de tensión diseñado para acoplar los voltajes cumple con su propósito, mantiene una relación de voltaje de 5V a 3.3V.

Tabla 4.2. Resultados divisor de tensión.

| Voltaje Fuente | Lectura Voltímetro |
|----------------|--------------------|
| 5              | 3.3                |
| 4.5            | 2.97               |
| 4.0            | 2.64               |
| 3.5            | 2.31               |
| 3.0            | 1.98               |
| 2.5            | 1.65               |
| 1.0            | 1.32               |
| 0.5            | 0.99               |
| 0              | 0.66               |





## Prueba salida sensor de corriente ACS-712

El sensor de efecto Hall ACS 712, es capaz de medir corrientes alternas y directas, según datos obtenidos de la hoja de datos del fabricante, por lo que la prueba consistió en conocer el comportamiento de la salida del sensor en cuanto a corrientes alternas.

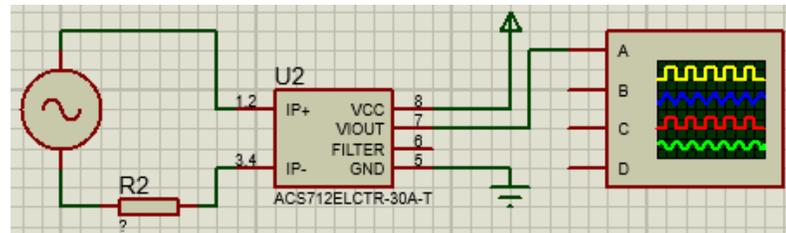


Figura 4.3. Diagrama conexiones prueba salida sensor ACS 712.

La prueba consistió en:

- Alimentar al sensor con un voltaje en directa de 5 Volts (figura 4.3).
- Conectar diferentes cargas de tipo resistivo en las terminales de medición del sensor.
- Acoplar la salida del sensor con una de las puntas de prueba del osciloscopio.
- Encender el osciloscopio, y colocar un ajuste de 1V/div, con una velocidad de muestro de 100ms.
- Medir la señal obtenida del sensor con cada una de las cargas utilizadas.

Las cargas empleadas para esta prueba, consistieron en cargas resistivas, como focos incandescentes, parrillas, etc. de diversas potencias.

## Resultados de la prueba del sensor de corriente ACS-712.

En la tabla 4.3 se muestran los resultados obtenidos, donde se observa que la salida del sensor de efecto Hall, es una señal de voltaje en directa, pero es variante, posee puntos máximos y mínimos de voltaje, los cuales son proporcionales a la corriente que es consumida por la carga; además, mientras más corriente consume la carga, el valor en voltaje se acerca más a 5 Volts.



Tabla 4.3. Resultados voltaje salida sensor efecto Hall.

| Carga                  | Voltaje Máximo |
|------------------------|----------------|
| Foco 40 watts          | 2.6            |
| Foco 100 watts         | 2.64           |
| Plancha 1100W          | 3.32           |
| Estación de soldado    | 3.08           |
| Planchas 2300W         | 4.04           |
| Parrilla eléctrica     | 3.32           |
| 2 parrillas eléctricas | 4.12           |

### Prueba para el algoritmo para la medición de corriente.

Se diseñó un algoritmo que obtiene la cantidad de corriente que se encuentra consumiendo una carga, a través del sensor de efecto hall.

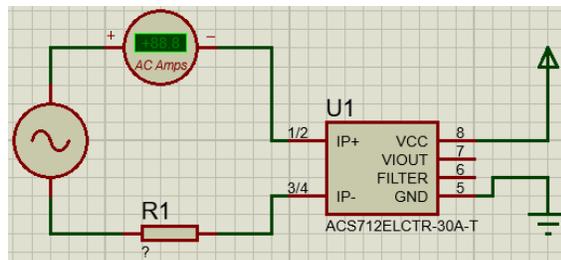


Figura 4.4. Diagrama para medición de corriente.

La prueba consistió en:

- Conectar en las terminales de medición del sensor con la carga a medir como se muestra en la figura 4.4.
- Acoplar un amperímetro para tomar lectura de la corriente.
- Cargar a la tarjeta el programa para obtener la medición de la corriente.
- Conectar cargas resistivas diferentes y capturar mediciones del amperímetro y del sensor de efecto Hall.

### Resultados obtenidos de la prueba para el algoritmo de medición de corriente.

Los resultados obtenidos se muestran en la tabla 4.4.





Tabla 4.4. Resultados algoritmo medición corriente

| Carga               | Amperímetro (A) | Algoritmo (A) | % error absoluto |
|---------------------|-----------------|---------------|------------------|
| Foco 40 watts       | 0.336           | 0.38          | 13.1             |
| Foco 100 watts      | 0.58            | 0.6           | 3.45             |
| Plancha 1100W       | 8.65            | 8.48          | 1.97             |
| Estación de soldado | 5.82            | 5.46          | 6.19             |
| Planchas 2300W      | 17.21           | 16.99         | 1.28             |
| Parrilla eléctrica  | 8.65            | 8.36          | 3.35             |

Donde después de un análisis de error absoluto entre los valores obtenidos, se obtuvo un error máximo de 13.1%, para corrientes que son inferiores a medio Amper.

### Prueba para el algoritmo para la obtención de potencia en Watt-hora.

Se diseñó un algoritmo que cuantifica la energía eléctrica que se encuentra consumiendo una carga, entrega un valor en unidades de Watt-hora (Wh). La prueba consistió en obtener lecturas por tiempo, donde se obtienen los resultados para diferentes cargas resistivas. Además, se establecieron intervalos de tiempo en el que se tomarían lecturas, y se compararían con los valores teóricos.

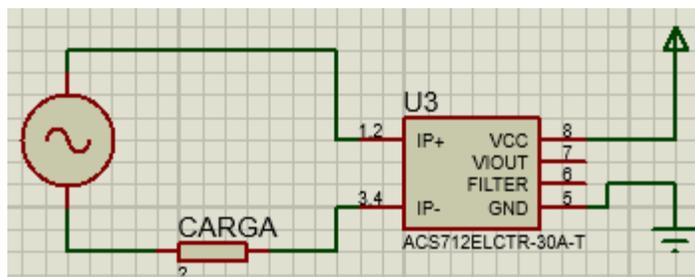


Figura 4.5. Diagrama conexión carga.

Para esta prueba se realizaron las siguientes actividades:

- Se conectó una carga resistiva a las terminales del sensor de efecto Hall (figura 4.5).
- Se cargó en la tarjeta el algoritmo para la obtención de Wh.





- El proceso de intervalos por tiempo tomara lecturas del consumo de energía eléctrica de la carga y una vez que termine el tiempo se mostrara la cantidad de energía consumida que cuantifico el sistema.
- El proceso se repite para cada intervalo y carga.

### Resultados obtenidos de la medición de Wh por intervalos de tiempo.

Los resultados obtenidos de manera experimental se muestran en la tabla 4.5. Los valores calculados se muestran en la tabla 4.6. Y en la tabla 4.7 se muestran los porcentajes de error obtenidos en las mediciones.

Tabla 4.5 Valores obtenidos de manera experimental.

| Aparato   | Consumo 1Hr (Wh) | 5 min (Wh) | 60 s (Wh) | 30 s (Wh) | 15 s (Wh) | 5 s (Wh) | 1 s (Wh) |
|-----------|------------------|------------|-----------|-----------|-----------|----------|----------|
| Foco 15W  | 15.5             | 1.31       | 0.26      | 0.1       | 0.05      | 0.01     | 0        |
| Foco 30W  | 30.2             | 2.46       | 0.49      | 0.27      | 0.12      | 0.04     | 0        |
| Foco 40W  | 41.1             | 3.89       | 0.75      | 0.35      | 0.17      | 0.06     | 0.01     |
| Foco 70W  | 74.14            | 6.51       | 1.25      | 0.64      | 0.33      | 0.11     | 0.02     |
| Foco 100W | 94.76            | 8.2        | 1.6       | 0.8       | 0.4       | 0.12     | 0.1      |

Tabla 4.6 Valores calculados.

| Aparato   | Consumo 1Hr (Wh) | 5 min (Wh) | 60 s (Wh) | 30 s (Wh) | 15 s (Wh) | 5 s (Wh) | 1 s (Wh) |
|-----------|------------------|------------|-----------|-----------|-----------|----------|----------|
| Foco 15W  | 15               | 1.25       | 0.25      | 0.125     | 0.0625    | 0.021    | 0.004    |
| Foco 30W  | 30               | 2.5        | 0.5       | 0.25      | 0.125     | 0.042    | 0.008    |
| Foco 40W  | 40               | 3.33       | 0.67      | 0.33      | 0.17      | 0.056    | 0.011    |
| Foco 70W  | 70               | 5.83       | 1.17      | 0.58      | 0.29      | 0.097    | 0.019    |
| Foco 100W | 100              | 8.33       | 1.67      | 0.83      | 0.42      | 0.139    | 0.098    |





Tabla 4.7 Porcentajes de variación.

| Aparato   | Consumo 1Hr (Wh) | 5 min (Wh) | 60 s (Wh) | 30 s (Wh) | 15 s (Wh) | 5 s (Wh) | 1 s (Wh) |
|-----------|------------------|------------|-----------|-----------|-----------|----------|----------|
| Foco 15W  | 3.33             | 4.8        | 4         | 20        | 20        | 52.38    | 100      |
| Foco 30W  | 0.67             | 1.6        | 2         | 8         | 4         | 4.76     | 100      |
| Foco 40W  | 2.75             | 16.82      | 11.94     | 6.06      | 0         | 7.14     | 9.09     |
| Foco 70W  | 5.91             | 11.66      | 6.84      | 10.34     | 13.79     | 13.4     | 5.26     |
| Foco 100W | 5.24             | 1.56       | 4.19      | 3.61      | 4.76      | 13.67    | 2.04     |

Los resultados obtenidos, muestran que son valores muy cercanos entre sí, pero existe una diferencia entre ellos, en la tabla 4.7 se presenta las diferencias, donde podemos observar que para cargas menores de 40 W, existe una mayor variación. Para las cargas que son mayores a 40 W, se obtuvieron mejores resultados, debido a que las diferencias, no superan el 20 por ciento. Una de las posibles causas que causaron los resultados obtenidos, es que se consideró de manera ideal la potencia de las cargas, siendo que, por los métodos de fabricación, las potencias reales pueden llegar a variar, debido a los procesos de manufactura, la calidad de los materiales, etc.

## 4.2 Pruebas realizadas al primer prototipo

Las pruebas realizadas al primer prototipo fueron realizadas con el propósito de probar el funcionamiento del prototipo y en base a los resultados obtenidos, poder evaluar si es necesario realizar ajustes o cambios en el siguiente prototipo.

El primer prototipo se creó a partir del diagrama esquemático (figura 4.6), donde se muestran las conexiones realizadas entre los distintos módulos que lo conforman.



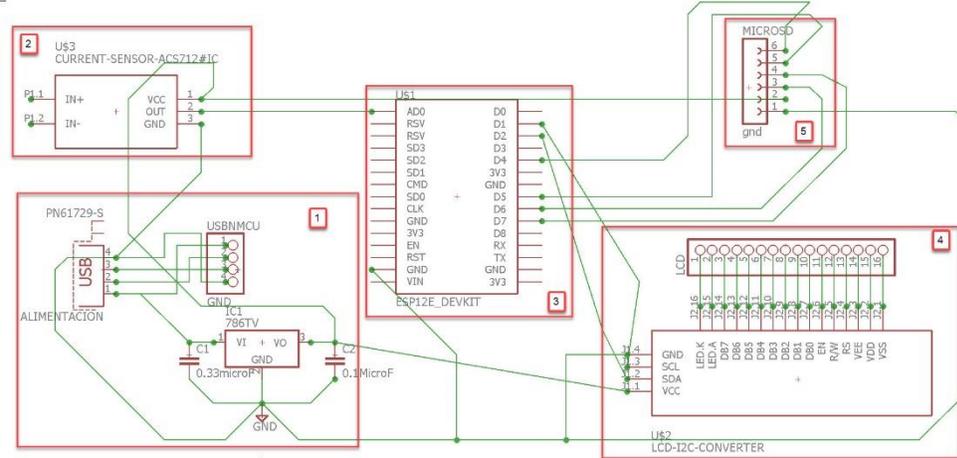


Figura 4.6 Diagrama esquemático primer prototipo.

Donde:

1. Fuente de alimentación. Esta sección del diagrama corresponde a una fuente de alimentación, es la encargada de suministrar el nivel correcto de voltaje para que las demás partes del sistema funcionen correctamente. Está compuesta por un conector USB tipo B y un regulador de voltaje, un LM 7805, el cual mantiene un voltaje estable a un valor de 5 V.
2. Sensor de corriente. Se encuentra el sensor de corriente ACS-712, el cuantifica la corriente instantánea que está circulando entre sus terminales, por lo que se coloca en serie con la carga, así la corriente que mida, será la misma corriente que consume la carga.
3. Sistema de control. Está conformado por la tarjeta Node-mcu, cuyo microcontrolador es el esp8266, el cual se encarga de llevar a cabo las funciones para la obtención de consumo eléctrico, así como las funciones lógicas, de control y comunicación que llevara a cabo el sistema
4. Pantalla. Esta sección del diagrama ilustra las conexiones realizadas con una pantalla de cristal líquido, la cual es una pantalla de cristal líquido (LCD), de 16 columnas por 2 filas, en la que se puede mostrar hasta 32 caracteres simultáneos.





Esta parte del diagrama está compuesta por la pantalla y por un controlador de la misma que se comunica a través del protocolo de comunicación I<sup>2</sup>C con el fin de emplear menos terminales del sistema de control.

5. Memoria microSD. En esta región se coloca el módulo para almacenar información en una memoria microSD, de esta forma, se puede extraer la memoria del módulo, insertarla en una computadora y se pueden visualizar los valores almacenados en ella.

### Prueba de medición de voltaje de alimentación

El experimento consistió en probar el funcionamiento de la fuente del primer prototipo, el procedimiento realizado fue energizar la tarjeta y medir los voltajes de operación de la misma, con el fin de verificar que los niveles de energía fueran los adecuados.

Para ello, se utilizó un multímetro una fuente de 9 V en DC. El procedimiento fue, mediante el conector USB de la placa se energizo la tarjeta y con un multímetro, se midieron los niveles de voltaje que llegaban a los componentes.

### Resultados prueba de encendido

Los resultados de la prueba fueron satisfactorios, los valores obtenidos están mostrados en la tabla 4.8.

Tabla 4.8. Nivel de voltaje en la tarjeta.

| Nivel de voltaje ( V <sub>DC</sub> ) | Componente                      |
|--------------------------------------|---------------------------------|
| 9                                    | Entrada regulador de voltaje    |
| 5                                    | Salida regulador de voltaje     |
| 5                                    | Entrada de alimentación NodeMCU |
| 5                                    | Sensor de corriente             |
| 5                                    | Pantalla LCD                    |





Cabe mencionar que los valores obtenidos demuestran que el diseño de la tarjeta fue el adecuado, pues los niveles de voltaje llegaron a las terminales correctas y con un nivel de voltaje equivalente a  $5 V_{DC}$ .

Aunque, cabe resaltar, que el brillo de fondo en la pantalla LCD no era visible, por lo que se encontró, que la corriente máxima que salía de los pines de la tarjeta, no era suficiente para mantener el brillo en la pantalla, por lo tanto, se considera utilizar una fuente de alimentación externa para los próximos prototipos, para que se aproveche el voltaje de línea  $127 V_{AC}$  y se cuente con una salida de  $5 V_{DC}$ .

### **Prueba de comunicación con computadora**

La prueba consistió en comunicar a la tarjeta con una computadora para realizar cambios en la programación del prototipo sin tener que retirar el sistema de control (NodeMCU) de la placa base del prototipo.

Para ello, se conectó el prototipo a una computadora mediante un cable USB y se cargó un programa de prueba al prototipo.

### **Resultados obtenidos de la comunicación con la computadora**

Al conectar el sistema a la computadora, no se logró reconocer la tarjeta NodeMCU, por lo que la comunicación a través del prototipo no se logró, se revisaron las conexiones del NodeMCU con el prototipo, y las líneas de comunicación para la transferencia de datos estaban interrumpidas, debido a que el cable que se soldó a la tarjeta se rompió internamente debido al juego que tenía con la tarjeta, por lo tanto, es necesario eliminar el cable, se consideró para la realización del segundo prototipo.

El primer prototipo, mostró los primeros inconvenientes en el diseño de la tarjeta, además brindo mejoras que se implementaron en el segundo prototipo, como un



mayor espacio entre los componentes, e indicadores para la alimentación, cambio del adaptador USB tipo B, por un tipo A.

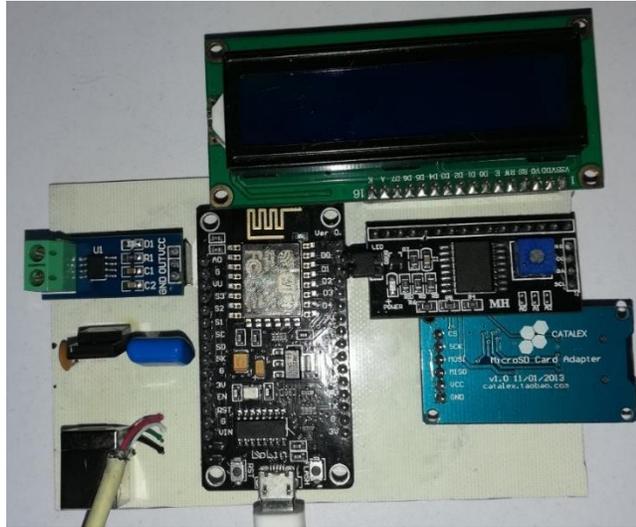


Figura 4.7 Primer prototipo.

Cabe mencionar, que el primer prototipo solo se encarga de medir el consumo eléctrico de una carga conectada al sistema a través de la bornera del sensor de corriente.

### 4.3 Pruebas realizadas al segundo prototipo.

El segundo prototipo, fue creado a partir de los resultados obtenidos de las pruebas realizadas al primer prototipo, donde los resultados notables fueron:

- Incorporación de una fuente de alimentación externa, para aprovechar el voltaje de línea monofásico a 127 V<sub>AC</sub>.
- Mejorar el espacio entre los componentes del sistema para que no se encimen unos con los otros.
- Modificar la conexión para la comunicación con la computadora.





que las demás partes del sistema funcionen correctamente. Está compuesta por un puente rectificador de diodos, que se encargan de convertir un voltaje de alterna en directa, la salida convertida está conectada a una resistencia la cual, decrementa el nivel del voltaje y con un diodo Zener se mantiene un voltaje de 5 V.

2. Sensor de corriente. Esta parte del diagrama, considera el espacio para el sensor de corriente, y con respecto al prototipo anterior, se consideraron las terminales de entrada para la conexión de la fuente de alimentación.

3. Sistema de control. Está conformado por la tarjeta electrónica NodeMCU, la cual posee un microcontrolador ESP8266, el cual, funciona como sistema de control y de comunicaciones, pues establece una comunicación inalámbrica mediante WiFi para poder establecer una comunicación con la Internet.

4. Entrada de alimentación y comunicación para el sistema de control. Esta sección del diagrama muestra la entrada de alimentación para los módulos y el sistema de control, está conformada por un regulador de voltaje que mantendrá un nivel de voltaje constante, además se incluyen dos terminales USB para la alimentación y comunicación del sistema de control, la otra terminal es para la comunicación con una computadora.

5. Etapa de potencia. En esta parte del diagrama se encuentra el módulo con relevador para la activación y desactivación de la carga que se conecte al sistema.

### **Prueba de alimentación del prototipo.**

Para esta prueba se analizó el funcionamiento de la fuente de alimentación del segundo prototipo, para ello, con ayuda de un voltímetro se midieron los voltajes de operación de la fuente de alimentación, el voltaje de entrada y el voltaje de salida.

El procedimiento fue:



- Conectar la entrada de la fuente del prototipo a la toma de alimentación de energía alterna.
- Con la ayuda del multímetro, medir el voltaje en alterna en la entrada del puente de diodos.
- Con el voltímetro en directa, medir el voltaje de salida del puente rectificador.
- Medir los voltajes de alimentación de los módulos del prototipo.

### Resultados de la prueba de alimentación.

Las mediciones obtenidas se muestran en la tabla 4.9

Tabla 4.9. Resultados prueba alimentación segundo prototipo.

| Nivel de voltaje    | Componente                              |
|---------------------|---|
| 125 V <sub>AC</sub> | Entrada fuente de alimentación          |
| 100 V <sub>AC</sub> | Terminal de salida resistencia de carga |
| 20 V <sub>DC</sub>  | Salida del puente de diodos             |
| 5 V <sub>DC</sub>   | Salida del diodo Zener                  |
| 5 V <sub>DC</sub>   | Terminal de entrada regulador 7805      |
| 3.7 V <sub>DC</sub> | Terminal de salida regulador 7805       |
| 3.5 V <sub>DC</sub> | Sensor de corriente                     |
| 3.5 V <sub>DC</sub> | Entrada alimentación NodeMCU            |

Los resultados obtenidos mostraron que la fuente de alimentación presentó problemas para alimentar el circuito con 5V, por lo tanto, se consideró rediseñar completamente la fuente de alimentación, por una que entregue un nivel constante de 5 V<sub>DC</sub> y no presente caídas de tensión.

### Prueba de medición de corrientes con cargas resistivas

Esta prueba consistió en conectar al sistema cargas resistivas hasta que se aproximaran a los valores máximos que puede cuantificar el sistema, con el fin de





comprobar la capacidad del sistema para la medición de grandes cantidades de energía.

El procedimiento fue:

- Conectar las terminales de entrada de alimentación del prototipo a la toma de alimentación doméstica.
- Con ayuda de un amperímetro, medir la cantidad de energía que circulaba por el circuito cuando se incrementaba la carga conectada al sistema.
- Para las cargas, primero se conectó una lámpara incandescente de 40 W, después se otra lámpara de 100W.
- La siguiente carga fue una plancha de 1100 W.
- La ultima carga fue una plancha de 2300 W.

Se anotaron los valores de la corriente medida.

### Resultados de la prueba medición de corrientes con cargas resistivas

Los resultados obtenidos se muestran en la tabla 4.7.

Tabla 4.10. Resultados medición de corrientes segundo prototipo.

| Carga             | Corriente medida [A] |
|-------------------|----------------------|
| Lámpara 40 watts  | 0.38                 |
| Lámpara 100 watts | 0.6                  |
| Plancha 1100W     | 8.48                 |
| Plancha 2300W     | -                    |

Donde se pueden apreciar las corrientes obtenidas por el segundo prototipo, donde, una vez que se conectó la carga de 2300 W la cual demando una mayor cantidad de corriente al ser energizada, presento una falla en el diseño de la tarjeta, las pistas del segundo prototipo no soportaron el nivel de corriente, se quemaron por la cantidad de energía, debido a lo delgado de las pistas. Estos resultados fueron considerados para la realización del tercer prototipo. Además, se incorporaran





elementos de protección como un fusible, para que las cargas y el sistema no deban soportar cargas para las que están diseñados.

#### 4.4 Pruebas y resultados del tercer prototipo

En este apartado se describen las pruebas realizadas al tercer prototipo de la presente propuesta. El tercer prototipo, está diseñado para soportar una potencia de hasta 3800W, además incorpora el módulo para la activación o desactivación de la carga. El diseño del tercer prototipo fue la base para la construcción de los sistemas medidor general y medidor individual de la presente propuesta.

##### Prueba a la fuente de alimentación

La prueba consistió en conectar el sistema a la alimentación de energía alterna. Para comprobar que la fuente se encuentre funcionando de manera correcta.

##### Resultados fuente de alimentación

La prueba realizada tuvo como objetivo comprobar el correcto funcionamiento de la fuente de alimentación, para ello, se energizó la tarjeta, y con ayuda de un multímetro se obtuvieron mediciones del voltaje a la entrada de la fuente y a la salida de la misma, adicionalmente, se midió el voltaje en las entradas de alimentación de los módulos que componen al sistema. Los resultados obtenidos se muestran en la tabla 4.11.

Tabla 4.11 Medición de voltaje.

| Medición de voltaje | Etapas                            |
|---------------------|-----------------------------------|
| 125 V <sub>AC</sub> | Entrada fuente de alimentación    |
| 5V <sub>DC</sub>    | Salida fuente de alimentación     |
| 5V <sub>DC</sub>    | Entrada alimentación Wemos        |
| 5V <sub>DC</sub>    | Entrada alimentación ACS712       |
| 5V <sub>DC</sub>    | Entrada de alimentación relevador |



Los resultados obtenidos, muestran el correcto funcionamiento de la fuente de alimentación, dado a que los voltajes se mantuvieron constantes en el valor adecuado para los componentes electrónicos. Por lo tanto, esta fuente de alimentación fue la seleccionada para implementarla en los sistemas de la presente propuesta.

### **Prueba de las borneras de conexión**

En esta prueba se conectaron diversas cargas resistivas a las borneas empleadas en el diseño del tercer prototipo. La prueba consistió en conectar cargas diferentes a la bornera de salida.

### **Resultados de las borneras de conexión**

Los resultados obtenidos de la conexión de cargas a las borneras mostraron que existe una dificultad al conectar los segmentos de cable conductor a las borneras, por lo que estar cambiando de carga, retirando y colocando los conductores de cada carga en la bornera es una tarea muy tediosa, pues se debe comprobar que los conductores deben siempre de mantenerse aislados entre sí, pues en una de las conexiones, uno de los hilos de un cable conductor, hizo contacto con la otra terminal lo que causo un corto en la prueba. Por lo tanto, dados los resultados obtenidos, se retiró la bornera como terminal de contacto con la carga, y en su lugar se optó por colocar un contacto eléctrico para que sea sencillo conectar una clavija al sistema.

## **4.5 Pruebas realizadas con el entorno Cayenne**

### **Prueba de conexión con Cayenne**

Esta prueba consistió en ingresar los valores de la cuenta creada en Cayenne, en dos dispositivos a la vez, para conocer el comportamiento del entorno y conocer la respuesta del mismo.



Un dispositivo fue una computadora y el segundo fue un dispositivo móvil con sistema operativo Android, a través del uso de la aplicación del entorno.

### **Resultados obtenidos conexión con Cayenne**

Los resultados obtenidos mostraron que es posible mantener dos sesiones iniciadas en dos dispositivos diferentes. Pues el funcionamiento del entorno muestra los cambios realizados con un retraso. Es decir, si se movieron ajustes en la computadora, los cambios también se ven reflejados en la aplicación en el teléfono móvil y viceversa. Si se activa el botón simultáneamente en ambos dispositivos, el comportamiento varía, pues el dispositivo que envíe más rápido la información a través de internet, será el que realice el cambio. Solo se puede realizar un cambio a la vez.

### **Prueba notificación fuera de línea**

Se configuró una notificación en la plataforma de Cayenne y cada vez que el dispositivo se quede fuera de línea – ya sea por una interrupción en su alimentación o por una desconexión de Internet – se envié una notificación por mensaje vía correo electrónico y mensaje de texto a número móvil.

La prueba consistió en interrumpir el funcionamiento del sistema, desactivando y activando la alimentación del mismo.



## Resultados obtenidos notificación fuera de línea.

Los mensajes de texto al número móvil, y al correo electrónico se realizaron de manera correcta, en cuanto la plataforma detectaba que la tarjeta se encontraba fuera de línea, enviaba una notificación al correo electrónico y al teléfono móvil, los mensajes se muestran en la figura 4.9.

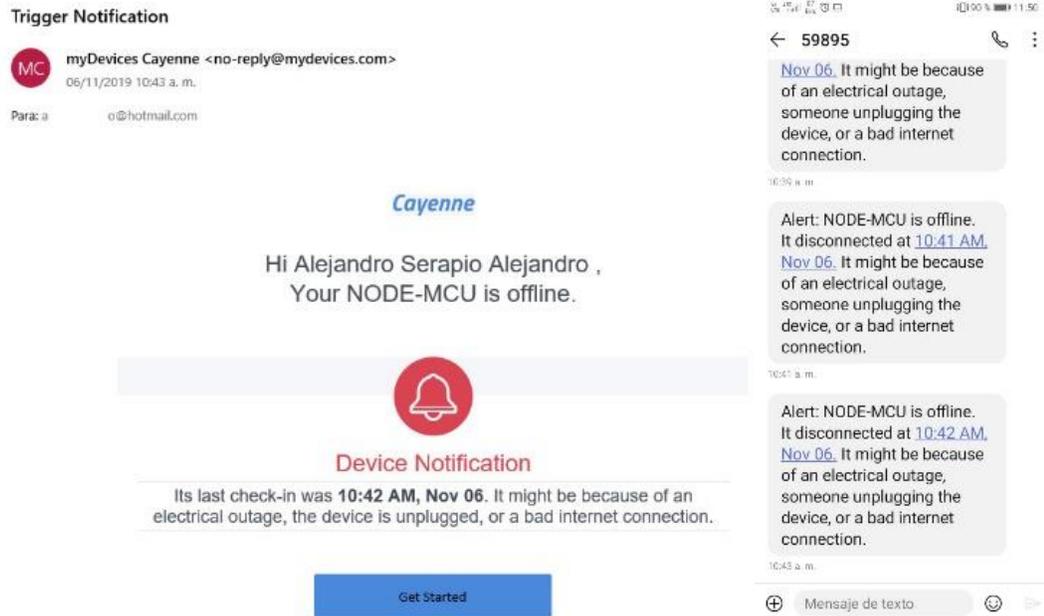


Figura 4.9 Mensajes notificación fuera de línea.

## Prueba notificación rangos de consumo

Se configuraron notificaciones en la plataforma para que cada vez que el medidor supere un intervalo, se envié una notificación. La prueba consistió en enviar valores superiores a los valores máximos de los intervalos, para comprobar el funcionamiento de la notificación.





## Resultados notificación rangos de consumo

Las notificaciones por mensaje de texto al número móvil y los correos electrónicos, se recibieron en el teléfono y en la cuenta de correo electrónico, por lo que el entorno, envió la información cuando el valor máximo de cada una de los rangos de del medidor de aguja fueron superados. En la figura 4.10 se muestran las notificaciones enviadas por la plataforma.

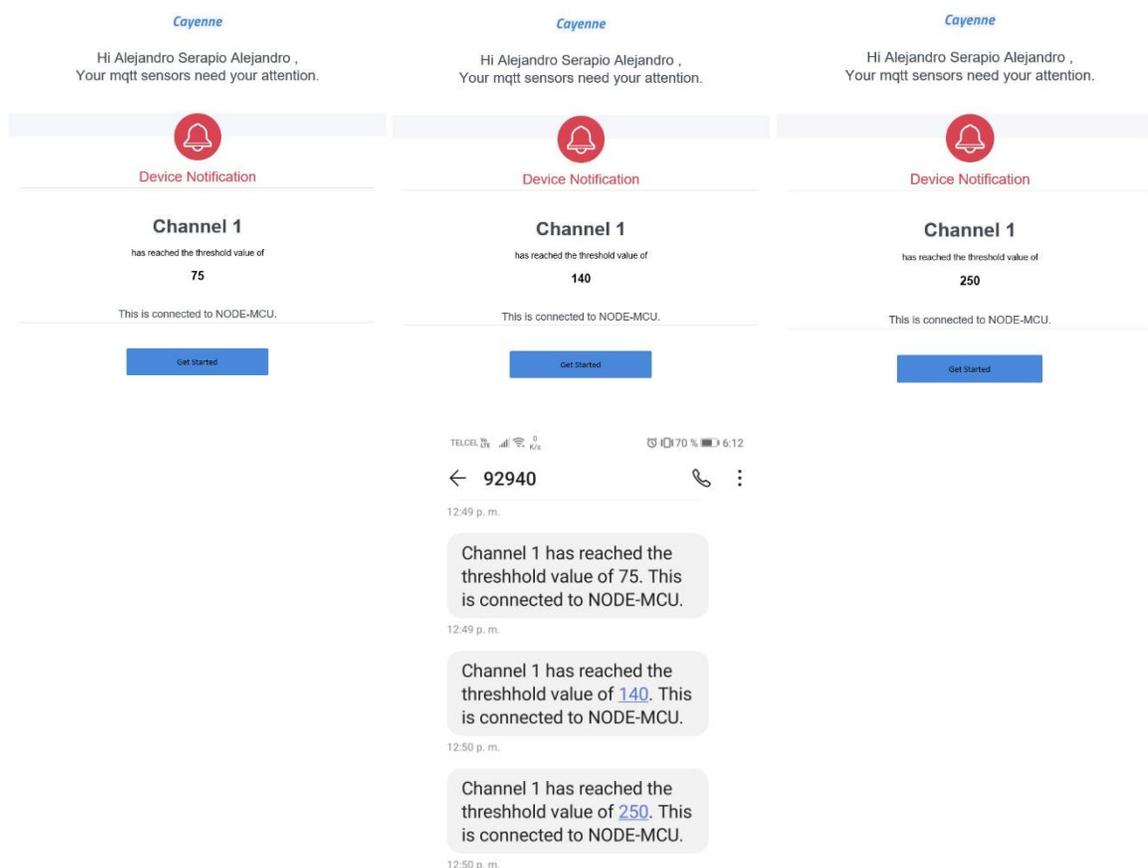


Figura 4.10 Notificaciones tarifas.

## Prueba visualización de información en widgets.

La prueba consistió en comprobar que la información enviada por el sistema es mostrada en el widget específico para cada valor. Se puso en funcionamiento el





sistema para poder visualizar como se muestra la información en la plataforma y en la aplicación.

### Resultados visualización de información en plataforma y aplicación

En la figura 4.11 se muestra una captura de los componentes en la plataforma Cayenne, donde se puede observar la información que cada widget se encuentra mostrando:

- OnOff. – Muestra el estado en el que se encuentra la alimentación de la carga, si el símbolo de la bombilla se encuentra encendido – como se ve en la figura 4.11 – quiere decir que la carga se encuentra activa.
- Potencia instantánea. – Muestra el valor de la potencia instantánea que se encuentra consumiendo la carga.
- Consumo. – Este widget muestra la cantidad de energía consumida por la carga, además, el color del indicador de aguja muestra la tarifa en la que se encuentra:
  - Si es de color **verde** se encuentra en la tarifa de consumo básico, que va hasta los primeros 75 KWh.
  - Si es de color **amarillo** se encuentra en la tarifa de consumo intermedio, la cual parte de los 75 KWh y va hasta los 140 KWh.
  - Si es de color **rojo**, se encuentra en la tarifa de consumo excedente, la cual parte de los 140 KWh hasta los 250KWh.
- Tarifas. – En los recuadros tarifa 1, 2, 3 y DAC se pueden insertar los valores de las tarifas básica, intermedia, excedente, y DAC respectivamente. El valor ingresado es un valor entero, no debe llevar puntos decimales, una vez que se han seleccionado los valores, para enviarlos en necesario presionar el botón salvar.
- Precio. – Muestra la cantidad monetaria que se debe cubrir por el consumo de energía eléctrica realizado. Se muestra la cantidad en valor numérico.



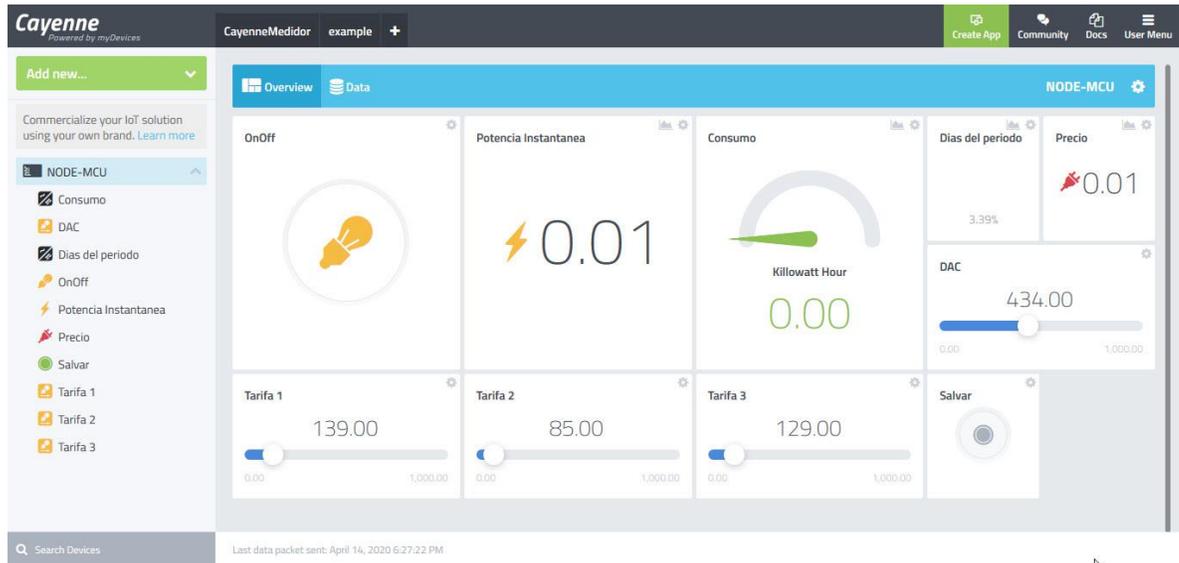


Figura 4.11 Componentes plataforma Cayenne.

Para la aplicación, se utilizaron los mismos componentes, pero el acomodo de los mismos y la presentación es diferente, en la figura 4.12 se muestra el aspecto de la aplicación.

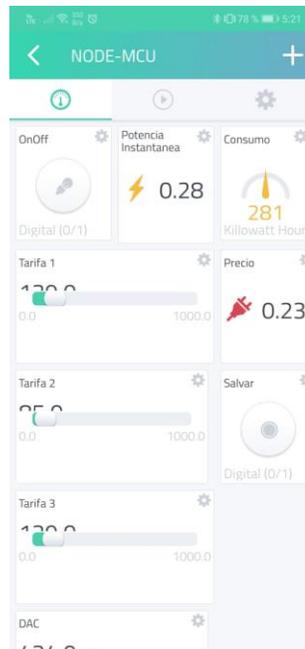


Figura 4.12 Componentes aplicación Cayenne.





Donde, los componentes se llaman de la misma forma, pero el acomodo de los mismos varia, pues se muestran acomodados de manera vertical, además, las barras deslizadoras no aparecen completas, se debe al acomodo automático que la aplicación le da a cada componente.

## 4.6 Pruebas realizadas al sistema final

### Prueba de medición de consumo eléctrico

La prueba, consistió en realizar lecturas simultáneas con dos instrumentos de medición de consumo eléctrico, sobre una misma carga. Se realizaron mediciones durante 7 días y se obtuvieron casi tres mil valores. Los resultados obtenidos fueron comparados bajo el método de concordancia de Bland-Altman, donde se compararon los resultados de las mediciones obtenidas por el medidor de esta propuesta, contra las mediciones obtenidas por el medidor proporcionado por la CFE.

### Resultados prueba de consumo eléctrico

Los resultados obtenidos de las mediciones se muestran en la figura 4.13.

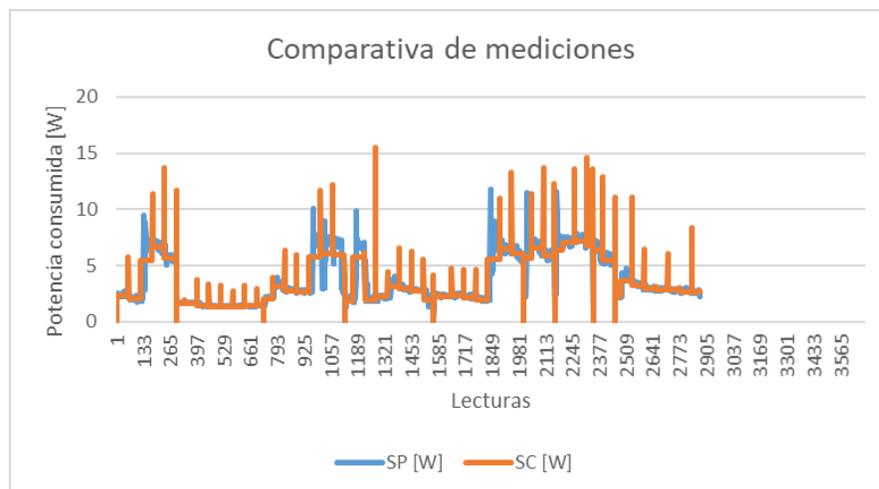


Figura 4.13 Resultados mediciones.



Donde SP, se refiere al sistema propuesto, y SC a un sistema comercial (medidor comercial de energía eléctrica).

Los resultados obtenidos de la aplicación del método de Bland-Altman, se muestra en la figura 4.14

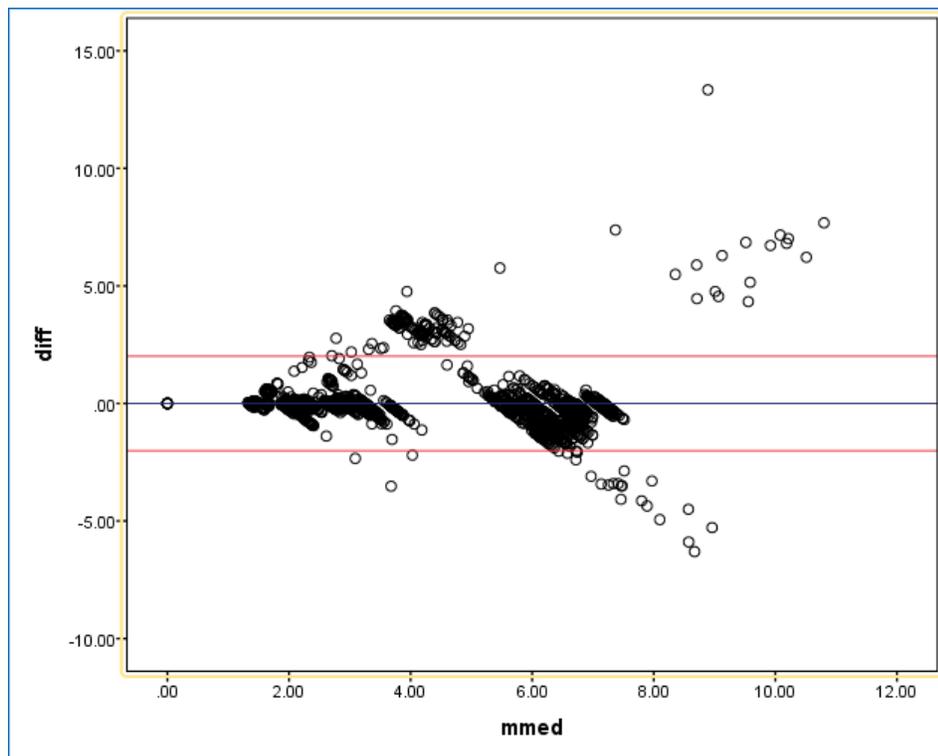


Figura 4.14 Resultados Método Bland-Altman.

En la figura 4.12 se puede apreciar los resultados obtenidos del método Bland-Altman, donde se puede apreciar que la mayoría de los puntos, se encuentran dentro de los márgenes rojos, los cuales indican un 95% de confianza, lo que quiere decir, que mientras más puntos de la gráfica, se encuentren entre los márgenes, los resultados tendrán una mayor concordancia entre sí, y mediante los resultados obtenidos de esta prueba se puede determinar que la calidad de la mediciones del sistema propuesto son equivalentes en un 95% al del medidor de la CFE.





## Conclusiones

Se llevó a cabo un análisis de funcionamiento del sensor de corriente ACS-712; con lo cual, se desarrolló un algoritmo mediante el cual se obtiene la medición del consumo eléctrico, el proceso parte de la obtención de la corriente instantánea, con el valor obtenido se calcula la potencia instantánea, y mediante la acumulación de la potencia instantánea, se obtiene la cantidad de energía consumida, expresada en Wh.

Se implementó el sistema en una tarjeta basada en el ESP-8266, donde se crearon las funciones para la obtención del consumo de la energía eléctrica, para la comunicación inalámbrica a través de Internet con la plataforma Cayenne, para la activación o desactivación de la carga a través de la plataforma Cayenne y para poder ingresar las credenciales y contraseñas de la red de acceso a Internet sin la necesidad de modificar el código.

Se diseñó una aplicación en la plataforma Cayenne, la cual funciona como interfaz visual y de configuración, para que el usuario logre visualizar la información proporcionada por el sistema y pueda configurar cambios en el mismo. La plataforma posee la capacidad para almacenar la información obtenida por el sistema y mostrar un histórico de los valores registrados. Adicionalmente se configuraron notificaciones para informar al usuario cuando se haya superado el rango de cada tarifa.

Se incorporó en el sistema un módulo con relevador que se utiliza para la activación o desactivación de la alimentación de la carga, la cual puede gestionarse desde la interfaz desarrollada en Cayenne.

Se realizó un análisis de concordancia del sistema propuesto con respecto a un instrumento de medición de la CFE, los resultados obtenidos muestran que existe un 95% de concordancia entre los valores de ambos sistemas.





## **Trabajo Futuro**

Como trabajo por realizar, aún quedan algunas características que se podrían implementar al sistema actual. Una de esas nuevas funciones sería que el sistema de manera automática, obtuviera el precio de la tarifa desde la página de CFE.

Otra característica por desarrollar sería crear un servidor web propio, dado a que las necesidades y requerimientos particulares del sistema se podrían cubrir mejor.

Trabajar en el tamaño del dispositivo, para reducirlo a manera de que sea más compacto.

Incursionar en mejores materiales para la carcasa de los dispositivos, que soporten las condiciones adversas del clima y de la exposición diaria al sol.





## Referencias

- [1] Agencia Internacional de Energía, «International Energy Agency,» OECD/IEA, 01 01 2019. [En línea]. Available: <https://www.iea.org/statistics/?country=WORLD&year=2016&category=Electricity&indicator=undefined&mode=chart&dataTable=INDICATORS>. [Último acceso: 08 02 2019].
- [2] E.A. Gaibor Edwin, «Implementacion de un prototipo de medidor de energia electrica residencial considerando la reduccion de perdidas no tecnicas por hurto.,» *Publicando*, vol. 1, n° 15, pp. 66-82, 2018.
- [3] P. D. Pérez, «Sistema embebido basado en el monitoreo de consumo de energía eléctrica de electrodomésticos para el control del gasto.,» *Revista Tecnologica Digital*, vol. 8, n° 1, pp. 69-80, 2018.
- [4] CFE, «Comision Federal de Electricidad,» 14 05 2019. [En línea]. Available: <https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/TarifasCRECasa/Casa.aspx>.
- [5] M. B. Andrés, INTERNET DE LAS COSAS, Madrid: REUS, 2018.
- [6] M. L. i. Seuba, Internet de las cosas. La transformacion digital de la sociedad, Madrid: RA-MA, 2019.
- [7] M. Lopez i Seuba, Internet de las cosas. La transformacion digital de la sociedad, Madrid: RA-MA, 2019, p. 46.



- [8] Mundial, Banco, «BANCO MUNDIAL,» [En línea]. Available: <https://datos.bancomundial.org/indicador/it.net.user.zs>. [Último acceso: 24 05 2019].
- [9] G. M. C. D. M. F. J. Q. Joaquín Olivares, «researchgate,» Julio 2003. [En línea]. Available: [https://www.researchgate.net/publication/268253760\\_GUIA\\_PARA\\_PROGRAMACION\\_DE\\_FPGAS](https://www.researchgate.net/publication/268253760_GUIA_PARA_PROGRAMACION_DE_FPGAS). [Último acceso: 15 10 2018].
- [10] RaspeberryPi, «Raspberrypi.org,» Raspberry Pi Foundation, [En línea]. Available: <https://www.raspberrypi.org/products/raspberrypi-3-model-b/>. [Último acceso: 2017 junio 2019].
- [11] Intel, «Intel,» Intel Corporation, [En línea]. Available: <https://ark.intel.com/content/www/es/es/ark/products/83137/intel-galileo-gen-2-board.html>. [Último acceso: 07 06 2019].
- [12] Atmel, «Microchip,» 01 01 2015. [En línea]. Available: [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf). [Último acceso: 17 05 2019].
- [13] Espressif System, «espressif,» 2019. [En línea]. Available: [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). [Último acceso: 7 09 2019].
- [14] Allegro MicroSystem, «allegromicro,» 2019. [En línea]. Available: <https://www.allegromicro.com/en/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs/ACS712.aspx>. [Último acceso: 31 05 2019].



- [15] datalogger, «datalogger,» [En línea]. Available: <http://datalogger.pbworks.com/w/file/fetch/89507207/Datalogger%20-%20SD%20Memory%20Reader%20Datasheet.pdf>. [Último acceso: 05 11 2018].
- [16] Xiamen amotec display, «sparkfun,» 29 10 2008. [En línea]. Available: <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>. [Último acceso: 15 11 2018].
- [17] L. Ada, «adafruit,» 25 01 2020. [En línea]. Available: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>. [Último acceso: 30 01 2020].
- [18] Songle, «songle,» [En línea]. Available: <http://www.songle.com/Public/Uploads/20161104/581c7d75b71b0.pdf>. [Último acceso: 10 02 2020].
- [19] FEC, «fecegypt,» [En línea]. Available: [https://www.fecegypt.com/uploads/dataSheet/1522335719\\_relay%20module.pdf](https://www.fecegypt.com/uploads/dataSheet/1522335719_relay%20module.pdf). [Último acceso: 10 05 2019].
- [20] Steren, «Steren,» [En línea]. Available: <https://www.steren.com.mx/pub/media/wysiwyg/doctosMX/ELI-713-V7.1-instr.pdf>. [Último acceso: 15 06 2019].
- [21] AUTODESK, «autodesk,» [En línea]. Available: <https://www.autodesk.com/products/eagle/compare>. [Último acceso: 15 04 2019].



- [22] Solidworks, «solidworks,» Dassault Systems SolidWorks Corporation, [En línea]. Available: <https://www.solidworks.es/sw/support/SolidWorks-Composer-SystemRequirements.html>. [Último acceso: 22 07 2019].
- [23] Anycubic, «anycubic,» [En línea]. Available: <https://www.anycubic.com/products/anycubic-i3-mega>. [Último acceso: 4 08 2019].
- [24] Amazon web services, «aws,» Amazon web services, 2019. [En línea]. Available: <https://aws.amazon.com/es/about-aws/global-infrastructure/?pg=WIAWS>. [Último acceso: 15 08 2019].
- [25] developers, «developers,» Google Developers, [En línea]. Available: <https://developer.android.com/things/get-started>. [Último acceso: 25 09 2019].
- [26] Microsoft Azure, «Microsoft Azure,» Microsoft, [En línea]. Available: <https://azure.microsoft.com/es-mx/services/iot-central/>. [Último acceso: 15 08 2019].
- [27] mydevices, «myDevices,» myDevices Inc, [En línea]. Available: <https://mydevices.com/integrations/>. [Último acceso: 8 11 2019].
- [28] CIF, «RS,» [En línea]. Available: <https://docs.rs-online.com/5b4e/0900766b80e3615b.pdf>. [Último acceso: 20 10 2019].
- [29] A. i. d. I. energia, «BANCO MUNDIAL,» 17 10 2017. [En línea]. Available: [https://datos.bancomundial.org/indicador/EG.ELC.ACCS.ZS?year\\_high\\_de\\_sc=true](https://datos.bancomundial.org/indicador/EG.ELC.ACCS.ZS?year_high_de_sc=true). [Último acceso: 01 12 2017].
- [30] Secretaria de Energia, «Sistema de informacion energetica,» secretaria de energia, 1 agosto 2017. [En línea]. Available:



<http://sie.energia.gob.mx/bdiController.do?action=cuadro&cvecua=IIIA1C0>

5. [Último acceso: 23 Octubre 2017].

- [31] Smappee, «smappee,» 2017. [En línea]. Available: [https://www.smappee.com/eu\\_es/home](https://www.smappee.com/eu_es/home). [Último acceso: 18 08 2017].
- [32] Wattio, «Wattio,» Mixpanel mobil analytics, [En línea]. Available: <https://wattio.com/es/>. [Último acceso: 18 08 2017].
- [33] L. d. J. R. G. y. M. M. Frago, «La generacion de energia electrica en México,» *Tecnología y Ciencias del Agua* , vol. III, nº 4, pp. 197-211, 2012.
- [34] S. Ian, Ingeniería de software, Madrid: Pearson, 2005.
- [35] R. S. Pressman, Ingeniería de software un enfoque practico, Ciudad de México: MC Graw Hill , 2010.
- [36] Sonoff, «Sonoff,» ITEAD Intelligent Systems Company, 01 01 2019. [En línea]. Available: <https://sonoff.itead.cc/en/products/residential/sonoff-s26-smart-plug>. [Último acceso: 07 02 2019].
- [37] tp-link, «tp-link,» TP-Link Technologies Company, 01 01 2019. [En línea]. Available: [https://www.tp-link.com/mx/products/details/cat-5258\\_HS110.html#](https://www.tp-link.com/mx/products/details/cat-5258_HS110.html#). [Último acceso: 07 02 2019].

## Referencias de figuras

Figura 2.1. Medidor energía eléctrica analógico. – recuperado el 04 de julio de 2019 de Autoría.

Figura 2.2. Medidor de energía eléctrica digital. – recuperado el 05 de julio 2019 de



<https://www.cfe.mx/industria/PublishingImages/Medidorelectr%C3%B3nico20cas a.jpg>

Figura 2.3. Tarjetas de desarrollo. – Recuperado el 07 de julio 2019 de [https://149362262.v2.pressablecdn.com/wp-content/uploads/2014/06/m36\\_boardscover-toc.jpg](https://149362262.v2.pressablecdn.com/wp-content/uploads/2014/06/m36_boardscover-toc.jpg)

Figura 2.4. Fpga. – Recuperado el 12 de julio 2019 de [https://cdn.shopify.com/s/files/1/0176/3274/products/451-02\\_1024x1024.jpg?v=1543335530](https://cdn.shopify.com/s/files/1/0176/3274/products/451-02_1024x1024.jpg?v=1543335530)

Figura 2.5 Raspberry Pi. – Recuperado el 12 de julio 2019 de [https://images-na.ssl-images-amazon.com/images/I/61iMPVK7X%2BL.\\_SL1500\\_.jpg](https://images-na.ssl-images-amazon.com/images/I/61iMPVK7X%2BL._SL1500_.jpg)

Figura 2.6. Intel Galileo. – Recuperado el 12 de julio de 2019 de [https://static.generation-robots.com/4709-large\\_default/arduino-intel-galileo-board.jpg](https://static.generation-robots.com/4709-large_default/arduino-intel-galileo-board.jpg)

Figura 2.7. Arduino Uno. – Recuperado el 13 de julio 2019 de [https://cdn.shopify.com/s/files/1/0557/2945/products/Arduino\\_UNO\\_-\\_ElectroCrea\\_x700.png?v=1546432033](https://cdn.shopify.com/s/files/1/0557/2945/products/Arduino_UNO_-_ElectroCrea_x700.png?v=1546432033)

Figura 2.8. NodeMCU. – Recuperado el 14 de julio 2019 de [https://www.makerlab-electronics.com/my\\_uploads/2017/02/NodeMCU-V3-ESP8266-ESP-12E-1.jpg](https://www.makerlab-electronics.com/my_uploads/2017/02/NodeMCU-V3-ESP8266-ESP-12E-1.jpg)

Figura 2.9. Sensor de corriente ACS - 712. – Recuperado el 14 de julio 2019 de [http://www.naylampmechatronics.com/946-large\\_default/modulo-ac712t.jpg](http://www.naylampmechatronics.com/946-large_default/modulo-ac712t.jpg)

Figura 2.10. Comportamiento de la salida del sensor. – Recuperado el 14 de julio de 2019 de <https://www.allegromicro.com/en/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs/ACS712.aspx>

Figura 2.11. Terminales ACS-712. – Recuperado el 14 de julio de 2019 de <https://www.allegromicro.com/en/Products/Current-Sensor-ICs/Zero-To-Fifty-Amp-Integrated-Conductor-Sensor-ICs/ACS712.aspx>

Figura 2.12. Módulo para memoria MicroSD. – Recuperado el 18 de julio de 2019 de [https://www.tiendatec.es/3286-large\\_default/modulo-adaptador-microsd-para-arduino.jpg](https://www.tiendatec.es/3286-large_default/modulo-adaptador-microsd-para-arduino.jpg)

Figura 2.13. Pantalla LCD 16x2. – Recuperado el 18 de julio 2019 de [https://cdn.shopify.com/s/files/1/1040/8806/products/AR-LCD1602\\_600x600.jpg?v=1568225278](https://cdn.shopify.com/s/files/1/1040/8806/products/AR-LCD1602_600x600.jpg?v=1568225278)

Figura 2.14. Sensor PIR. – Recuperado el 18 de julio de 2019 de <https://www.elektor.com/media/catalog/product/cache/2b4bee73c90e4689bbc4ca8391937af9/a/x/axirissensorc-web.jpg>



Figura 2.15. Módulo relevador 5700 W. – Recuperado el 29 de febrero 2020 de <https://ae01.alicdn.com/kf/HTB1bvO4MG6qK1RjSZFmq6x0PFXav/Power-relays-SLA-05VDC-SL-A-5V-30A-4PIN-T90.jpg>

Figura 2.16. Módulo Relevador. – Recuperado el 19 de julio 2019 de <https://electronilab.co/wp-content/uploads/2017/01/Modulo-Rel%C3%A9-De-1-Canal-Salida-Optoacoplada-5V-1.jpg>

Figura 2.17. Entorno de programación Arduino. – Recuperado el 20 de julio de 2019 de Autoría.

Figura 2.18. Cargar bibliotecas entorno Arduino. – Recuperado el 20 de julio de 2019 de Autoría.

Figura 2.19. Administrador de bibliotecas. – Recuperado el 20 de Julio de 2019 de Autoría.

Figura 2.20. Panel de control Eagle. – Recuperado el 21 de julio de 2019 de Autoría.

Figura 2.21. Ventana de diseño diagrama esquemático. – Recuperado el 21 Julio de 2019 de Autoría.

Figura 2.22. Ventana de diseño PCB. – Recuperado el 21 de Julio de 2019 de Autoría.

Figura 2.23. Solidworks. – Recuperado el 3 de agosto de 2019 de Autoría.

Figura 2.24. Interfaz Solidworks. – Recuperado el 3 de agosto de 2019 de Autoría.

Figura 2.25. Crear nuevo documento. – Recuperado el 3 de agosto de 2019 de Autoría.

Figura 2.26. Ventana de archivo activo SolidWorks. – Recuperado el 3 de agosto 2019 de Autoría.

Figura 2.27. Cura. – Recuperado el 4 de agosto 2019 de Autoría.

Figura 2.28. Interfaz Cura. – Recuperado el 4 de agosto de 2019 de Autoría.

Figura 2.29. Crear cuenta AWS. – Recuperado el 5 de agosto de 2019 de [https://portal.aws.amazon.com/billing/signup?nc2=h\\_ct&src=default&redirect\\_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirmation&language=es\\_es#/start](https://portal.aws.amazon.com/billing/signup?nc2=h_ct&src=default&redirect_url=https%3A%2F%2Faws.amazon.com%2Fregistration-confirmation&language=es_es#/start)

Figura 2.30. Consola Android Things. – Recuperado el 6 de agosto de 2019 de <https://developer.android.com/things/get-started>



Figura 2.31. Ventana de registro Cayenne. – Recuperado el 8 de noviembre de 2019 de <https://developers.mydevices.com/cayenne/features/#sign-up>

Figura 2.32. Interfaz usuario Cayenne. – Recuperado el 8 de noviembre de 2019 de Autoría.

Figura 3.1. Diagrama a bloques general. – Recuperado el 7 de Septiembre 2019 de Autoría.

Figura 3.2. Diagrama de flujo para obtener la potencia instantánea y acumulada. – Recuperado el 21 de Septiembre de 2019 de Autoría.

Figura 3.3. Diagrama para completar 1 segundo. – Recuperado el 22 de Septiembre de 2019 de Autoría.

Figura 3.4. Diagrama de flujo configuración valores iniciales. – Recuperado el 23 de septiembre de Autoría.

Figura 3.5. Diagrama de flujo funcionamiento Cayenne. – Recuperado el 23 de Septiembre de 2019 de Autoría.

Figura 3.6. Señal de salida del sensor con carga en AC. – Recuperado el 22 de Septiembre 2019 de Autoría.

Figura 3.7. Segmento de código “declaración de bibliotecas”. – Recuperado el 2 de Noviembre de 2019 de Autoría.

Figura 3.8. Segmento de código “variables globales 1”. – Recuperado el 2 de Noviembre de 2019 de Autoría.

Figura 3.9. Segmento de código “variables globales 2”. – Recuperado el 2 de Noviembre de 2019 de Autoría.

Figura 3.10. Segmento de código “variables globales 3”. – Recuperado el 2 de Noviembre de 2019 de Autoría.

Figura 3.11. Segmento de código “configuración setup”. – Recuperado el 2 de Noviembre de 2019 de Autoría.

Figura 3.12. Segmento de código “configuración setup 2”. – Recuperado el 3 de Noviembre de 2019 de Autoría.

Figura 3.13 Lectura valores eeprom. – Recuperado el 03 de Marzo de 2020 de Autoría.

Figura 3.14. Segmento de código “comparación de lecturas”. – Recuperado el 3 de Noviembre de 2019 de Autoría.

Figura 3.15. Segmento de código “cálculo de potencia y acumulativo de energía consumida”. - Recuperado el 3 de Noviembre de 2019 de Autoría.



Figura 3.16 Obtención precio por el consumo realizado. – Recuperado el 03 de Marzo de 2020 de Autoría.

Figura 3.17. Segmento de código “espera”. – Recuperado el 3 de Noviembre de 2019 de Autoría.

Figura 3.18. Segmento de código “envió de datos a Cayenne”. – Recuperado el 3 de Noviembre de 2019 de Autoría.

Figura 3.19. Segmento de código “obtención del estado de la carga”. - Recuperado el 3 de Noviembre de 2019 de Autoría.

Figura 3.20 Configuración widget gauge. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.21 Configuración casilla de texto. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.22 Configuración días del periodo. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.23 Configuración botón On/Off. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.24 Configuración slider tarifa. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.25 Configuración botón salvar. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.26 Configuración fuera de línea. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.27 Configuración notificación tarifas. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.28. Resultados calculo ancho de pista. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.29. Fuente de alimentación comercial. – Recuperado el 15 de Enero de 2020 de Autoría.

Figura 3.30. Diagrama medidor general. – Recuperado el 25 de Enero de 2020 de Autoría.

Figura 3.31 Símbolo relevador. – Recuperado el 25 de Enero de 2020 de Autoría.

Figura 3.32 Huella relevador. – Recuperado el 25 de Enero de 2020 de Autoría.

Figura 3.33 Tarjeta de circuito impreso medidor general. – Recuperado el 25 de Enero de 2020 de Autoría.



Figura 3.34. Diagrama esquemático medidor de consumo individual. – Recuperado el 5 de Octubre de 2019 de Autoría.

Figura 3.35. Diseño tarjeta de circuito impreso medidor individual. – Recuperado el 7 de Octubre de 2019 de Autoría.

Figura 3.6. Tarjeta medidora de consumo individual. – Recuperado el 11 de Octubre de 2019 de Autoría.

Figura 3.37. Posición de componentes. – Recuperado el 11 de Octubre de 2019 de Autoría.

Figura 3.38. Prisma rectangular. – Recuperado el 1 de febrero de 2020 de Autoría.

Figura 3.39. Caja. – Recuperado el 1 de febrero de 2020 de Autoría.

Figura 3.40. Postes para colocar la tarjeta. – Recuperado el 1 de febrero de 2020 de Autoría.

Figura 3.41 Aperturas para ingresar los conductores. – Recuperado el 1 de febrero de 2020 de Autoría.

Figura 3.42 Contenedor del medidor general. – Recuperado el 1 de febrero de 2020 de Autoría.

Figura 3.43 Tapa contenedor medidor general. – Recuperado el 1 de febrero de 2020 de Autoría.

Figura 3.44. Prisma rectangular. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.45. Prisma rectangular hueco. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.46. Orificio para clavija de acceso rápido. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.47. Orificio enchufe. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.8 Modelo tridimensional en Cura. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.49. Acomodo del modelo tridimensional. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.50 Sistema montado en el contenedor. – Recuperado el 5 de febrero de 2020 de Autoría.

Figura 3.51 Sistema medidor general montando en la carcasa. – Recuperado el 20 de febrero 2020 de Autoría.



Figura 4.1. Diagrama conexiones ADC. – Recuperado el sábado 7 de Septiembre 2019 de Autoría.

Figura 4.2. Diagrama de flujo prueba ADC. – Recuperado el sábado 7 de Septiembre 2019 de Autoría.

Figura 4.3. Diagrama conexiones prueba salida sensor ACS 712. – Recuperado el 8 de Septiembre 2019 de Autoría.

Figura 4.4. Diagrama para medición de corriente. – Recuperado el 8 de Septiembre 2019 de Autoría.

Figura 4.5. Diagrama conexión carga. – Recuperado el 10 de Septiembre 2019 de Autoría.

Figura 4.6. Diagrama esquemático primer prototipo. - Recuperado el 10 de Septiembre 2019 de Autoría.

Figura 4.7 Primer prototipo. – Recuperado el 10 de Septiembre 2019 de Autoría.

Figura 4.8. Diagrama esquemático segundo prototipo. - Recuperado el 12 de Septiembre 2019 de Autoría.

Figura 4.9 Mensajes notificación fuera de línea. – Recuperado el 6 de Noviembre de 2019 de Autoría.

Figura 4.10 Notificaciones tarifas. – Recuperado el 25 de Noviembre de 2019 de Autoría.

Figura 0.1 Componentes plataforma Cayenne. – Recuperado el 14 de Abril de 2020 de Autoría.

Figura 4.12 Componentes aplicación Cayenne. – Recuperado el 14 de Abril de 2020 de Autoría.

Figura 4.13 Resultados mediciones. – Recuperado el 8 de Octubre de 2019 de Autoría.

Figura 4.14 Resultados Método Bland-Altman. – Recuperado el 8 de Octubre de 2019 de Autoría.

Figura B.1 Logo empresa. – Recuperado el 21 de Octubre de 2019 de Autoría.

Figura C.1 Certificado de participación en el taller “Internet of Things – a practical approach”. – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.2 Constancia de participación Rally latinoamericano de innovación. – Recuperado el 15 de Diciembre de 2019 de Autoría.



Figura C.3 Constancia por ponencia en “coloquio de la maestría en ciencias de la computación” CU UAEM Valle de México. – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.4 Reconocimiento por impartir el taller “desarrollo de sistemas con Java”. – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.5 Constancia por impartir el taller "diseño de circuitos PCB". – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.6 Constancia 1er lugar en innovación nivel sede rally latinoamericano de innovación. – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.7 Reconocimiento por ponencia en CU UAEM Ecatepec. – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.8 Constancia de participación en taller "modelo de negocios canvas". – Recuperado el 15 de Diciembre de 2019 de Autoría.

Figura C.9 Certificado de participación como ponente en el congreso CORE19. – Recuperado el 15 de Diciembre de 2019 de Autoría.





## Anexos

### A) Código

```
#include <FS.h> //se utiliza para la gestión de los valores almacenados en SPIFFS
#include <ESP8266WiFi.h> //se utiliza para realizar la comunicación inalámbrica
#include <DNSServer.h> //se utiliza para implementar un servidor en modo STA y
AP
#include <ESP8266WebServer.h> //se utiliza para la creación de un servidor web
#include <WiFiManager.h> //se utiliza para gestionar el arranque en modo STA y
AP
#include <ArduinoJson.h> //se utiliza para manejar datos del tipo Json
#define CAYENNE_DEBUG //declaración de variable constante
#define CAYENNE_PRINT Serial // declaración de variable constante
#include <CayenneMQTTESP8266.h> //comunica al sistema con Cayenne

char cayenneUserName[40] = ""; // almacena usuario para Cayenne
char cayennePassword[40] = ""; // almacena contraseña para Cayenne
char cayenneClientID[40] = ""; // almacena identificador para Cayenne

float lecturaVp = 0.0; //almacena el valor obtenido de la digitalización del adc
float vpMax = 0.0; //almacena el mayor valor de la digitalización del adc
float Vrms = 0.0; //guarda el valor en voltaje de la digitalización del adc
float tarifa1 = 0.83;
float tarifa2 = 1.01;
float tarifa3 = 2.95;
float DAC = 4.36;
```





```
int tarifa11 = 0;
```

```
int tarifa22 = 0;
```

```
int tarifa33 = 0;
```

```
int DAC11 = 0;
```

```
float tarifaf1 = 0.0; //almacena el valor de la tarifa rango 1
```

```
float tarifaf2 = 0.0; //almacena el valor de la tarifa rango 2
```

```
float tarifaf3 = 0.0; //almacena el valor de la tarifa rango 3
```

```
float DACf = 0.0; //almacena el valor de la tarifa DAC
```

```
int salvar = 0; //variable que indica cuando es necesario almacenar nueva información.
```

```
byte ht1; //almacena el byte más significativo
```

```
byte lt1; //almacena el byte menos significativo
```

```
byte ht2; //almacena el byte más significativo
```

```
byte lt2; //almacena el byte menos significativo
```

```
byte ht3; //almacena el byte más significativo
```

```
byte lt3; //almacena el byte menos significativo
```

```
byte hDAC; //almacena el byte más significativo
```

```
byte lDAC; //almacena el byte menos significativo
```

```
float dinero = 0.0; //almacena la cantidad a pagar por el consumo realizado
```

```
unsigned long tmpRef = 0; //almacena el tiempo de referencia para duración de procesos
```

```
unsigned long tmpIni = 0; //almacena el tiempo cuando el ciclo está en el comienzo
```



```
unsigned long fin = 0; //almacena el tiempo cuando el ciclo llega al final

int d = 0; //almacena el resultado de un contador
int dias = 0; //lleva el registro de cuantos días han transcurrido.
int seg = 0; //lleva la cuenta de los segundos transcurridos para el envío de la
información
int tiempo = 5; //tiempo en segundos que el sistema envía información a la
plataforma

float ampRms = 0.0; // almacena la corriente instantánea cuantificada por el sensor
float voltLinea = 123.5; //valor del voltaje AC de línea
float potIns = 0.0; // guarda el valor de la potencia instantánea.
float potAcum = 0.0; // almacena el valor de la potencia consumida
float potWh = 0.0; // guarda el valor de la energía consumida
float potKWh = 0.0; // valor de la energía consumida en KWh

bool shouldSaveConfig = false; //bandera utilizada para almacenar nuevos valores
en modo AP

////////función para notificar cuando los nuevos datos deben ser salvados////////

void saveConfigCallback () { // declaración de la función para salvar información
  Serial.println("Should save config"); // muestra un mensaje por el puerto serial
  shouldSaveConfig = true; //activa la bandera para salvar los datos
}

////////////////////////////////////////función setup o de configuraciones////////////////////////////////////////

void setup() { //funcion setup, se ejecuta solamente 1 vez
```





```
Serial.begin(115200); //establece una velocidad de comunicación a 115200 bits
por segundo

Serial.println("abriendo sistema de archivos."); //imprime un mensaje por el puerto
serial

////////// funcion para leer los valores almacenados en la memoria
SPIFFS//////////

if (SPIFFS.begin()) { //inicia el sistema de archivos

    Serial.println("Sistema de archivos abierto"); //imprime un mensaje por el puerto
serial

    if (SPIFFS.exists("/config.json")) { // condición, si existen los ficheros se leen y
cargan

        Serial.println("leyendo archivos de configuración"); //imprime un mensaje por el
puerto serial

        File configFile = SPIFFS.open("/config.json", "r"); //abre el archivo configFile de
la memoria SPIFFS

        if (configFile) { //condicion si el archivo existe

            Serial.println("opened config file"); //imprime mensaje por el puerto serial

            size_t size = configFile.size(); //mide el tamaño del archivo

            std::unique_ptr<char[]> buf(new char[size]); // asigna un buffer para almacenar
el contenido del archivo

            configFile.readBytes(buf.get(), size); // lee la información del buffer
considerando el tamaño del archivo

            DynamicJsonBuffer jsonBuffer; // declaración de una variable perteneciente
de la clase dynamycJsonBuffer

            JsonObject& json = jsonBuffer.parseObject(buf.get()); // asigna el contenido
del buffer

            json.printTo(Serial); //imprime el contenido de json

            if (json.success()) { //condición si el contenido fue cargado correctamente

                Serial.println("\nparsed json"); // imprime por el puerto serial
```





```
strcpy(cayenneUserName, json["cayenneUserName"]); //almacena en la
variable, el contenido de json.

    } else { // en caso de que no se cumpla con la condición

        Serial.println("error al cargar valores almacenados"); // imprime un mensaje
por el puerto serial

        }

    }

} else { //en caso de que la condición no se cumpla

    Serial.println("failed to mount FS"); // imprime un mensaje por el puerto serial

}

WiFiManagerParameter custom_cayenne_user("cayenne_user", "username", "",
40); //asignacion de parametros

WiFiManagerParameter custom_cayenne_pass("cayenne_pass", "Password", "",
40); //asignacion de parametros

WiFiManagerParameter custom_cayenne_id("cayenne_id", "user_id", "", 40);
//asignacion de parametros

WiFiManager wifiManager; //declaración de una variable

wifiManager.setSaveConfigCallback(saveConfigCallback); // se llama a la funcion
de salvar datos

wifiManager.AddParameter(&custom_cayenne_user); //agrega un parámetro en
el formulario html

wifiManager.AddParameter(&custom_cayenne_pass); //agrega un parámetro en
el formulario html

wifiManager.AddParameter(&custom_cayenne_id); //agrega un parámetro en el
formulario html
```





```
wifiManager.autoConnect("AutoConnectAP"); // muestra el nombre de la red
Serial.println("connected...yeey :)"); // imprime un mensaje por el monitor serial

strcpy(cayenneUserName, custom_cayenne_user.getValue()); //asigna los
valores introducidos a variables

strcpy(cayennePassword, custom_cayenne_pass.getValue()); //asigna los
valores introducidos a variables

strcpy(cayenneClientID, custom_cayenne_id.getValue()); //asigna los valores
introducidos a variables

/////////////////////////////////salvar nuevos valores/////////////////////////////////

if (shouldSaveConfig) {
    Serial.println("saving config"); //imprime un mensaje por el monitor serial
    DynamicJsonBuffer jsonBuffer; //declaracion de una variable
    JsonObject& json = jsonBuffer.createObject(); //crea una variable de tipo objeto
    json["cayenneUserName"] = cayenneUserName; // se copia el contenido del
paramatro

    File configFile = SPIFFS.open("/config.json", "w"); //abre el archivo config.json
para escribir
    if (!configFile) { // condicion si el archivo no existe
        Serial.println("failed to open config file for writing"); // imprime un mensaje por
el monitor serial
    }

    json.printTo(Serial); //imprime el contenido de json por el serial
    json.printTo(configFile); //imprime el contenido de json en configFile
    configFile.close(); // cierra el archivo configFile
```





```
}  
  
pinMode(D3, OUTPUT); //configura el pin D3 como salida  
//////////////////Lectura de valores almacenados en EEPROM//////////////////  
    //lectura de valor tarifa 1  
    ht1=EEPROM.read(4); //almacena el contenido de 1 localidad eeprom en la  
variable  
    lt1=EEPROM.read(5); //almacena el contenido de 1 localidad eeprom en la  
variable  
    tarifa11= ht1; //carga los bits más significativos en la posición menos  
significativa  
    tarifa11= tarifa11 <<8; //realiza un desplazamiento de 8 bits hacia la derecha  
    tarifa11= tarifa11|lt1; //aplica una operación or para almacenar los menos  
significativos  
  
    //lectura de valor tarifa 2  
    ht2=EEPROM.read(6); //almacena el contenido de 1 localidad eeprom en la  
variable  
    lt2=EEPROM.read(7); //almacena el contenido de 1 localidad eeprom en la  
variable  
    tarifa22= ht2; //carga los bits más significativos en la posición menos  
significativa  
    tarifa22= tarifa22 <<8; //realiza un desplazamiento de 8 bits hacia la derecha  
    tarifa22= tarifa22|lt2; //aplica una operación or para almacenar los menos  
significativos  
  
    //lectura de valor tarifa 3
```





```
ht3=EEPROM.read(8); //almacena el contenido de 1 localidad eeprom en la
variable
lt3=EEPROM.read(9); //almacena el contenido de 1 localidad eeprom en la
variable
tarifa33= ht3; //carga los bits más significativos en la posición menos
significativa
tarifa33= tarifa33 <<8; //realiza un desplazamiento de 8 bits hacia la derecha
tarifa33= tarifa33|lt1; //aplica una operación or para almacenar los menos
significativos

//lectura de valor alarmas
hDAC=EEPROM.read(10); //almacena el contenido de 1 localidad eeprom en
la variable
IDAC=EEPROM.read(11); //almacena el contenido de 1 localidad eeprom en la
variable
DAC11= hDAC; //carga los bits más significativos en la posición menos
significativa
DAC11= DAC11 <<8; //realiza un desplazamiento de 8 bits hacia la derecha
DAC11= DAC11|IDAC; //aplica una operación or para almacenar los menos
significativos
Serial.println("Connected"); //imprime un mensaje por el puerto serial
Serial.println("Your IP is"); //imprime un mensaje por el puerto serial
Serial.println((WiFi.localIP().toString())); //imprime un mensaje por el puerto serial
Cayenne.begin(cayenneUserName, cayennePassword, cayenneClientID); //da de
alta credenciales, usuario, contraseña e ID

}
////////////////////Programa principal////////////////////////////////////
void loop()
{
```





```
Cayenne.loop(); // indica el inicio de Cayenne
tmpIni = millis(); //se almacena el tiempo inicio
vpMax = 0.0; //se limpia el registro del voltaje pico máximo
tmpRef = millis(); // se toma el tiempo en este punto
while ((millis() - tmpRef) < 17) // ciclo mientras que la condición se cumpla
{
    lecturaVp = analogRead(A0); //realiza una lectura del ADC
    if (lecturaVp >= vpMax) //compara la lectura obtenida si es más grande, se
almacena el valor
    {
        vpMax = lecturaVp ; //almacena el mayor valor leído
    }
} // fin del ciclo mientras

Vrms = vpMax * (0.004882); //convierte la lectura del adc en un valor de voltaje
ampRms = ((15 * (Vrms)) / 1.4142); //aplicación de la fórmula para obtener
corriente instantánea
potIns = ampRms * voltLinea; //operación para obtener la potencia instantánea
potAcum = potAcum + potIns; //Operación donde se almacena la suma de la
potencia
potWh = potAcum * 0.0003; //Convierte la potencia acumulada en Watts-hora
potKWh = potWh / 1000; //convierte los Wh a KWh

//Código que muestra el precio por el consumo realizado
if (0.0 < potKWh <= 75.0) //comparación para saber si está en la primera tarifa
{
    dinero = (potKWh * tarifa1); //obtiene el precio
}
```



```
if (75.0 < potKWh <= 140.0) //comparación para saber si está en la segunda tarifa
{
    dinero = ((potKWh-75.0) * tarifa2 + (tarifa1 * 75.0)); //obtiene el precio
}
if (140.0 < potKWh <= 250.0) //comparación para saber si está en la tercera tarifa
{
    dinero = ((potKWh-140.0) * tarifa3 + (tarifa1 * 75.0) + (tarifa2 * 140.0)); //obtiene
el precio
}
else //(PotenciaFaux>250.0) // si está en la tarifa DAC
{
    dinero = ( potKWh * DAC); //obtiene el precio
}
d = d + 1; //se incrementa en 1 cada que el programa se vuelve a ejecutar
seg = seg + 1; //se incrementa en 1 cada que el programa se vuelve a ejecutar

fin = (millis() - tmpIni); // se obtiene el tiempo de ejecución del programa principal

tmpRef = millis(); // se toma el tiempo como valor de referencia
do
{

} while ((millis() - tmpRef) < 1000 - fin); //mantiene una espera para completar 1
segundo

if (seg >= 60) //3600 //condición que evalúa el tiempo transcurrido
{
```





```
dias = dias + 1; // se incrementa en 1 el registro
seg = 0; //reinicia el contador
}

if (d >= tiempo) // tiempo que debe transcurrir para enviar valores a Cayenne
{
    Cayenne.virtualWrite(0, potKWh); //envía el valor de la variable por el canal 0 a
cayenne
    Cayenne.virtualWrite(1, potWh); //envía el valor de la variable por el canal 0 a
cayenne
    Cayenne.virtualWrite(2, dias); //envía el valor de la variable por el canal 0 a
cayenne
    Cayenne.virtualWrite(7, dinero); //envía el valor de la variable por el canal 2 a
cayenne
    d = 0; //reinicia el contador
}
//Código para almacenar los cambios de las tarifas.
if (salvar == 1)
{
    tarifaf1 = (tarifa11 * 0.001); //Convierte el valor enviado por la plataforma
tarifaf2 = (tarifa22 * 0.001); //Convierte el valor enviado por la plataforma
tarifaf3 = (tarifa33 * 0.01); //Convierte el valor enviado por la plataforma
DACf = (DAC11 * 0.01); //Convierte el valor enviado por la plataforma

    tarifa1 = tarifaf1; //Asigna el valor obtenido a la variable usada en el código
    tarifa2 = tarifaf2;
    tarifa3 = tarifaf3;
    DAC = DACf;
```





```
//Guardar información en EEPROM
//tarifa1
//Se guarda el byte más significativo
ht1 = highByte(tarifa11); //almacena los bytes más significativos
lt1 = lowByte(tarifa11); //almacena los bytes menos significativos
EEPROM.write(4, ht1); //almacena el byte en una localidad de la eeprom
EEPROM.write(5, lt1); //almacena el byte en una localidad de la eeprom

//tarifa2

ht2 = highByte(tarifa22); //almacena los bytes más significativos
lt2 = lowByte(tarifa22); //almacena los byte menos significativos
EEPROM.write(6, ht2); //almacena el byte en una localidad de la eeprom
EEPROM.write(7, lt2); //almacena el byte en una localidad de la eeprom

//tarifa3

ht3 = highByte(tarifa33); //almacena los bytes más significativos
lt3 = lowByte(tarifa33); //almacena los bytes menos significativos
EEPROM.write(8, ht3); //almacena el byte en una localidad de la eeprom
EEPROM.write(9, lt3); //almacena el byte en una localidad de la eeprom

//DAC

hDAC = highByte(DAC11); //almacena los bytes más significativos
IDAC = lowByte(DAC11); //almacena los bytes menos significativos
```



```
EEPROM.write(10, hDAC); //almacena el byte en una localidad de la eeprom
EEPROM.write(11, IDAC); //almacena el byte en una localidad de la eeprom
}
}
```

```
CAYENNE_IN(3) //Información proveniente del botón para encendido y apagado
{
  int OnOff = getValue.asInt(); //se almacena en una variable el valor enviado
  digitalWrite(D2, OnOff); //Se activa pone en estado alto el pin D2
}
```

```
CAYENNE_IN(4) //Información proveniente del slider para el valor de tarifa 1
{
  tarifa11 = getValue.asInt();
}
```

```
CAYENNE_IN(5) //Información proveniente del slider para el valor de tarifa 2
{
  tarifa22 = getValue.asInt();
}
```

```
CAYENNE_IN(6) //Información proveniente del slider para el valor de tarifa 3
{
  tarifa33 = getValue.asInt();
}
```

```
CAYENNE_IN(8) //Información proveniente del slider para el valor de tarifa DAC
{
```



```
DAC11 = getValue.asInt();  
}  
CAYENNE_IN(9) //Información proveniente del botón para salvar información  
{  
  salvar = getValue.asInt();  
}
```



## B) Plan de negocios

### Idea de negocio

Una empresa dedicada a la venta de sistemas de monitoreo y control de consumo eléctrico aplicados a casa-habitación, para que los usuarios puedan conocer el estado del consumo y así logren tener un control a distancia de la alimentación eléctrica de su hogar.

### Nombre y razón social

CEM (Care-Energy-Monitor);

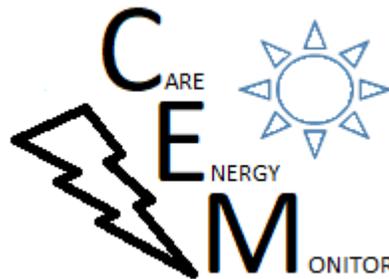


Figura B.1 Logo empresa.

Diseñadora y comercializadora de sistemas de monitoreo energético A.C.

### Eslogan

Cuidando tu consumo y la energía del mundo.

### Gama de productos

- Módulo medidor general. –Se encarga de medir el consumo eléctrico de una casa habitación de manera total.
- Módulo contacto. –Se encarga de medir el consumo eléctrico de los aparatos conectados, pudiendo activar o desactivar el consumo del mismo.
- Módulo socket. –Mide el consumo eléctrico de las luces o focos brindando información acerca del tiempo de uso, consumo realizado y puede activar o desactivar el consumo eléctrico.
- Servicio de monitoreo.



## **Beneficios que ofrece el producto**

Indica el estado del consumo de la energía eléctrica, brindando información de la tarifa actual, monto a pagar por el consumo realizado y días restantes para el término del periodo actual, además puede activar o desactivar la alimentación. Para que con la información mostrada el usuario pueda tomar acciones que le ayuden a aprovechar mejor el consumo eléctrico.

Comodidad. – El sistema está pensado para que el usuario solo monte los módulos sobre a instalación existente, además podrá activar o desactivar aparatos, aunque este fuera de casa.

Control. – El usuario podrá establecer horarios de encendido u apagado, para tener un mayor control de los aparatos conectados al sistema, incluso podría hacer parecer que se encuentra en casa sin estarlo.

## **Misión**

Somos una empresa que se dedica a la creación de dispositivos de monitoreo y control de la energía eléctrica, ofrecemos planes personalizados que se ajustan a las necesidades particulares de cada cliente. Los dispositivos que ofrecemos están diseñados por especialistas, los materiales utilizados son de gran calidad, durables, confiables y están desarrollados bajo normas de calidad internacional ISO.

## **Visión**

En 5 años la empresa será reconocida nivel nacional como una de las mejores empresas en desarrollo tecnológico, personal calificado y certificado. Los productos serán considerados de primera calidad. Además, los productos estarán disponibles en cualquier tienda de autoservicio y será una marca conocida nacionalmente.

## **Objetivos**



Primer año: Producir un total de 200 dispositivos apegándonos a los estándares de calidad para la producción de los mismos y Vender 15 dispositivos al mes, ofreciéndolos al mercado local.

Segundo año: Alcanzar las ventas proyectadas por mes, para pasar el punto de equilibrio en al menos un 25% y cubrir en al menos un 35% el préstamo realizado.

Tercer año: Expandir el área de venta, abarcando la zona metropolitana, aumentar el índice de ventas y abrir una segunda sucursal.

Cuarto año: Vender el sistema en el estado de México, incorporar nuevos módulos del sistema que monitoreen la temperatura, el consumo del agua y del gas.

Quinto año: Distribuir el sistema en toda la república mexicana.

### Análisis FODA

Tabla B.1 Análisis FODA

|  |  |
|--|--|
| <b>FORTALEZAS</b><br>Diseño del sistema.<br>Comunicación a través de Internet de las cosas.<br>Módulos para cada parte de la casa.<br>Paquetes personalizados.<br>Servicio de instalación.<br>Soporte técnico. | <b>DEBILIDADES</b><br>Empresa pequeña.<br>Mejorar presentación del producto.<br>Evitar errores en funcionalidades.   |
| <b>OPORTUNIDADES</b><br>Incremento en el precio de la energía.<br>Mercado.<br>Ahorro.<br>Ayuda a mostrarles a las personas cuanta energía consumen.  | <b>AMENAZAS</b><br>Competidores comerciales.<br>Avance y desarrollo tecnológico.<br>Incremento del precio del dólar.<br>Inflación<br>Políticas de importación.<br>Aranceles. |





## Modelo Canvas

Tabla B.2 Modelo Canvas

| <b>Aliados Clave</b>   | <b>Actividades Clave</b>  | <b>Propuesta de Valor</b>   | <b>Relación con el Cliente</b>  | <b>Segmentos de Clientes</b>   |
|--|---|---|---|--|
| <p>Servicios de paquetería.</p> <p>Proveedores de partes electrónicas (venden piezas).</p> <p>Tiendas departamentales.</p> <p>Socios comerciales maquinadores y ensambladores (China). (Así como elabora sus productos Apple).</p> <p>Servidores.</p> <p>Paquetería.</p> | <p>Diseño del sistema</p> <p>Instalación de sistema</p> <p>Mantenimiento del sistema</p> <p>Reparaciones del sistema</p> <p>Servicio de asistencia telefónica</p> | <p>Mi propuesta, ofrece a los usuarios el monitoreo y control eléctrico de su casa a distancia, siendo ellos mismos quienes deciden que partes de su casa desean controlar.</p> | <p>Es de manera directa, se apoya dando asistencia y se ofrecen planes que se ajusten a sus necesidades.</p> <p>Capacitación en línea.</p> <p>Soporte</p>   | <p>Personas físicas y morales interesadas en el aprovechamiento de la energía eléctrica y consumo sustentable.</p> |
|  | <b>Recursos Clave</b>   |   | <b>Canales</b>  |  |
|  | <p>Personal de venta y distribución.</p> <p>Programadores, técnicos para dar asistencia.</p> <p>Renta espacio</p> <p>Internet</p> <p>Habilidad de diseño</p>      |   | <p>Internet.</p> <p>Redes sociales.</p> <p>Medios impresos:</p> <p>Espectaculares.</p> <p>Sitio Web.</p> <p>Radio.</p>  |  |
| <b>Estructura de Costes</b>  |   |   | <b>Estructura de Ingresos</b>   |  |
| <p>Costos fijos: Renta del espacio de venta, publicidad, internet, líneas telefónicas, Sueldos y salarios, electricidad, equipo de cómputo.</p> <p>Costos variables: Precio paquetería, insumos, servicio de servidores.</p>   |   |   | <p>Venta de módulos y paquetes personalizados para cada cliente.</p> <p>Servicio de instalación y mantenimiento.</p> <p>Pago directo, transferencias bancarias y depósitos a cuenta bancaria.</p> |  |





## Estructura de costos

Se realizó un estudio acerca del presupuesto de inversión necesario para comenzar las operaciones en la empresa, y se obtuvo un presupuesto de inversión de \$81,708 pesos. Para realizar el estudio se consideraron los gastos e insumos necesarios para el primer año de operaciones.

Tabla B.3 Presupuesto de inversión.

| Conceptos                         | Unidad      | Cantidad | Costo unitario | Costo total       |
|-----------------------------------|-------------|----------|----------------|-------------------|
| <b>Activo fijo</b>                |             |          |                |                   |
| Computadoras                      | unidad      | 4        | \$9,163.00     | \$36,652.0        |
| Sillas ejecutivas                 | unidad      | 6        | \$800.00       | \$4,800.00        |
| Teclado y Mouse                   | unidad      | 4        | \$549.00       | \$2,196.00        |
| Teléfono IP                       | unidad      | 2        | \$1,040.00     | \$2,080.00        |
| Monitor                           | unidad      | 4        | \$3,000.00     | \$12,000.0        |
| Sillón 3P                         | unidad      | 1        | \$9,000.00     | \$9,000.00        |
| Mesas trabajo                     | unidad      | 2        | \$2,500.00     | \$5,000.00        |
| Agua 500 ml                       | unidad      | 50       | \$2.50         | \$125.00          |
| Laptop                            | unidad      | 2        | \$14,700.00    | \$29,400.0        |
| Escritorio                        | unidad      | 4        | \$1,200.00     | \$4,800.00        |
| Silla directiva ejecutiva         | unidad      | 1        | \$2,500.00     | \$2,500.00        |
| Escritorio Principal              | unidad      | 1        | \$5,000.00     | \$5,000.00        |
| Escritorio ejecutivo              | unidad      | 1        | \$6,000.00     | \$6,000.00        |
| Lápices                           | unidad      | 20       | \$58.50        | \$1,170.00        |
| Plumas                            | Paquete     | 12       | \$46.50        | \$558.00          |
| Hojas                             | Paquete     | 40       | \$80.00        | \$3,200.00        |
| Impresora                         | unidad      | 1        | \$5,494.00     | \$5,494.00        |
| Acondicionamiento del local       | Presupuesto | 1        | \$8,500.00     | \$ 8,500.00       |
| Requerimiento de herramientas     | Lote        | 1        | \$17,763.00    | \$17,763.0        |
| <b>Subtotal</b>                   |             |          |                | <b>\$26,263.0</b> |
| <b>Activo diferido</b>            |             |          |                |                   |
| Capacitación y asistencia técnica | Presupuesto | 1        | \$25,000.00    | \$25,000.0        |





| <b>Capital de trabajo</b> |          |     |            |                   |
|---------------------------|----------|-----|------------|-------------------|
| Materias primas           | Lote     | 1   | \$25,548   | \$25,548          |
| Mano de obra              | jornal   | 140 | \$112.00   | \$15,680.0        |
| Insumos                   | Lote     | 1   | \$1,217.00 | \$1,217.00        |
| Administración            | Servicio | 1   | \$9,000.00 | \$9,000.00        |
| Agua                      | Servicio | 1   | \$1,000.00 | \$1,000.00        |
| Luz                       | Servicio | 1   | \$2,000.00 | \$2,000.00        |
| Teléfono                  | Servicio | 1   | \$1,000.00 | \$1,000.00        |
| <b>Subtotal</b>           |          |     |            | <b>\$55,445</b>   |
| <b>Total</b>              |          |     |            | <b>\$81,708.0</b> |

### Proyección de ingresos y egresos

Tabla B.4 Proyección de ingresos y egresos.

| Concepto     | Venta al mes | Precio  | Año 1              | Año 2              | Año 3              | Año 4              | Año 5              |
|--------------|--------------|---------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Paquete 1    | 20           | \$3,500 | \$840,000          | \$882,000          | \$926,100          | \$972,405          | \$1,021,025        |
| Paquete 2    | 20           | \$1,750 | \$420,000          | \$441,000          | \$463,050          | \$486,202          | \$510,512          |
| Paquete 3    | 20           | \$2,625 | \$630,000          | \$661,500          | \$694,575          | \$729,303          | \$765,768          |
| <b>Total</b> |              |         | <b>\$1,890,000</b> | <b>\$1,984,500</b> | <b>\$2,083,725</b> | <b>\$2,187,911</b> | <b>\$2,297,306</b> |

### Punto de equilibrio

| Concepto            | Año1        | Año2        | Año3        | Año4        | Año5        |
|---------------------|-------------|-------------|-------------|-------------|-------------|
| Ventas              | \$1,890,000 | \$1,984,500 | \$2,083,725 | \$2,187,911 | \$2,297,306 |
| Costos fijos        | \$400,056   | \$420,058   | \$441,061   | \$463,114   | \$486,270   |
| Costos variables    | \$321,180   | \$337,239   | \$354,100   | \$371,806   | \$390,396   |
| Costos totales      | \$721,236   | \$757,297   | \$795,162   | \$834,920   | \$876,666   |
| Punto equilibrio \$ | \$481,958   | \$506,056   | \$531,359   | \$557,926   | \$585,823   |
| Punto equilibrio%   | 26%         | 26%         | 26%         | 26%         | 26%         |

El punto de equilibrio indica el porcentaje de ventas que se debe tener para cubrir los costos totales, sin que se tenga ganancias, es lo mínimo que se debe vender para no tener pérdidas. El punto de equilibrio, para el primer año arroja un 26%, lo





que indica que con el 26% de las unidades vendidas por mes (15 unidades por mes), la empresa cubre los costos totales.

### **C) Constancias**

En este anexo se encuentran las constancias obtenidas de las distintas actividades llevadas a cabo a lo largo de la maestría.



## D) Artículo publicado

En este apartado se adjunta el artículo publicado en la revista Research in Computing Science<sup>4</sup>; Volumen 48, tomo 10, páginas de la 279 a la 290, publicada el 1 de octubre de 2019.

### Research in Computing Science

#### Series Editorial Board

##### Editors-in-Chief:

Grigori Sidorov, CIC-IPN, Mexico  
Gerhard X. Ritter, University of Florida, USA  
Jean Serra, Ecole des Mines de Paris, France  
Ulises Cortés, UPC, Barcelona, Spain

##### Associate Editors:

Jesús Angulo, Ecole des Mines de Paris, France  
Jihad El-Sana, Ben-Gurion Univ. of the Negev, Israel  
Alexander Gelbukh, CIC-IPN, Mexico  
Ioannis Kakadiaris, University of Houston, USA  
Petros Maragos, Nat. Tech. Univ. of Athens, Greece  
Julian Padget, University of Bath, UK  
Mateo Valero, UPC, Barcelona, Spain  
Rafael Guzmán, Univ. of Guanajuato, Mexico

##### Editorial Coordination:

Alejandra Ramos Porras  
Carlos Vizcaino Sahagún

*Research in Computing Science*, Año 18, Volumen 148, No. 10, octubre de 2019, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. <https://www.rcs.cic.ipn.mx>. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 30 de agosto de 2019.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda estrictamente prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización del Instituto Politécnico Nacional.

*Research in Computing Science*, year 18, Volume 148, No. 10, October 2019, is published monthly by the Center for Computing Research of IPN.

The opinions expressed by the authors does not necessarily reflect the editor's posture.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research of the IPN.

---

Volume 148(10)

---

<sup>4</sup>[https://rcs.cic.ipn.mx/2019\\_148\\_10/Sistema%20de%20monitoreo%20de%20consumo%20electrico%20con%20interfaz%20para%20telefono%20inteligente.html](https://rcs.cic.ipn.mx/2019_148_10/Sistema%20de%20monitoreo%20de%20consumo%20electrico%20con%20interfaz%20para%20telefono%20inteligente.html)





**ISSN: 1870-4069**

---

---

Copyright © Instituto Politécnico Nacional 2019

Instituto Politécnico Nacional (IPN)  
Centro de Investigación en Computación (CIC)  
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal  
Unidad Profesional “Adolfo López Mateos”, Zacatenco  
07738, México D.F., México

<http://www.rcs.cic.ipn.mx>

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

The editors and the publisher of this journal have made their best effort in preparing this special issue, but make no warranty of any kind, expressed or implied, with regard to the information contained in this volume.

All rights reserved. No part of this publication may be reproduced, stored on a retrieval system or transmitted, in any form or by any means, including electronic, mechanical, photocopying, recording, or otherwise, without prior permission of the Instituto Politécnico Nacional, except for personal or classroom use provided that copies bear the full citation notice provided on the first page of each paper.

Indexed in LATINDEX, DBLP and Periodica

Electronic edition





## Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente

Alejandro Andrés Serapio Carmona<sup>1</sup>, Ismael Díaz Rangel<sup>2</sup>,  
Rodolfo Zola García Lozano<sup>1</sup>, Cuauhtémoc Hidalgo Cortés<sup>1</sup>

<sup>1</sup>Universidad Autónoma del Estado de México, Centro Universitario Ecatepec, México

<sup>2</sup>Universidad Nacional Autónoma de México, Facultad de Estudios Superiores Aragón, México  
andreserapio@hotmail.com, zolagarcia@yahoo.com,  
chidalgoc@uaemex.mx, ismael1099@hotmail.com

**Resumen.** El consumo eléctrico, es la cantidad de energía eléctrica que se ha utilizado a lo largo de un periodo definido, por la cual se debe pagar. El uso de la energía eléctrica está determinado por el estilo de vida de cada usuario, así como de la actividad para la que se destine su uso [1]. La empresa encargada de proveer la energía eléctrica cuenta con un sistema de cobro de tarifas que se incrementa a mayor consumo [2]. Esta información, se le da a conocer al usuario hasta que termina el periodo y se aplica el cobro. En este trabajo se describe el desarrollo de un sistema, cuya función principal es ser una herramienta que ayude a los usuarios a conocer fácilmente el estado del consumo eléctrico de manera anticipada a la fecha de corte; de esa manera, ellos podrán tomar las medidas pertinentes para utilizar racionalmente la energía eléctrica. Adicionalmente, el sistema brindará un estimado de la cantidad a pagar por el consumo realizado desde el inicio del corte hasta el momento actual. El módulo de monitoreo se comunica con un teléfono inteligente por medio de bluetooth, el control es realizado por un microcontrolador; para la medición de la corriente eléctrica, se utiliza el sensor ACS712 de efecto hall, el cual fue caracterizado para la medición de corriente alterna; además, se guarda un histórico de las mediciones para su análisis.

**Palabras clave:** consumo eléctrico, monitoreo, sensor de corriente.

### Power Consumption Monitoring System with Smartphone Interface

**Abstract.** Electricity consumption is the amount of electrical energy that has been used over a defined period, for which it must be paid. The use of electrical energy is determined by the lifestyle of each user, as well as the activity for which its use is intended [1]. The company responsible for providing electricity has a system for charging fees that increases to higher consumption [2]. This information is made known to the user until the end of the period and the payment is applied. This work describes the development of a system, whose main function is to be a tool that helps users to easily know the state of electricity consumption in advance of the cutoff date; that way, they can take the appropriate measures to rationally use electric power. Additionally, the system will provide an estimate of the amount to be paid for the consumption made from the





*Alejandro Andrés Serapio Carmona, Ismael Díaz Rangel, Rodolfo Zola García Lozano, et al.*

beginning of the cut until the current time. The monitoring module communicates with a smartphone via bluetooth, the control is carried out by a microcontroller; for the measurement of electric current, the hall effect ACS712 sensor is used, which was characterized for the measurement of alternating current; In addition, a history of the measurements is kept for analysis.

**Keywords:** electricity consumption, monitoring, current sensor.

## 1. Introducción

La energía eléctrica es un recurso que ha brindado apoyo para el desarrollo de las sociedades. Cuando el uso de la energía eléctrica comenzó a brindar mayores comodidades a las personas, su uso se fue incrementando; tanto, que en la actualidad la energía eléctrica se encuentra en casi todas las actividades diarias de las personas: en el transporte, en la preparación de la comida, en la iluminación, comunicaciones, entretenimiento, trabajo, etc. Su uso es tan intenso, que se ha tenido la necesidad de buscar e invertir en tecnologías que produzcan energía eléctrica a partir del uso de recursos renovables, con la finalidad de poder abastecer la creciente demanda de energía eléctrica, pues al crecer, lo hacen también las actividades relacionadas a su producción; aun así, es necesario cuidar de los recursos energéticos que se utilizan para la producción de energía, por lo que los usuarios deberían utilizar eficientemente la energía eléctrica proporcionada [3]. Para que el usuario tenga información y con ello tomar medidas de gestión del uso de energía, este trabajo propone el desarrollo de una herramienta para informar a los usuarios en cualquier momento, el estado del consumo energético. Adicionalmente, el sistema proporciona al usuario información acerca de la tarifa en que se encuentra, el precio a pagar por dicha tarifa, y el tiempo restante para la fecha de corte. La interfaz está desarrollada en una aplicación móvil, que se enlaza fácilmente con el módulo de monitoreo (o adquisición) mediante bluetooth.

## 2. Estado del arte

Existen trabajos, que han propuesto una solución para conocer el estado del consumo eléctrico a través del uso de medidores digitales. En [4] se describe un prototipo de medidor electrónico que considera las pérdidas de energía eléctrica ocasionadas por robo de energía a través del uso de sensores de corriente, colocados en segmentos específicos del tendido eléctrico; adicionalmente, el sistema puede establecer una comunicación a través de internet con la empresa que suministra el servicio, donde envía información acerca del estado del consumo; además, se puede activar o desactivar el suministro de energía de manera remota. El sistema utiliza un microcontrolador como sistema de procesamiento y de control, una pantalla LCD para mostrar la información al usuario y un Ethernet Shield para establecer una comunicación a través de internet. El autor refiere que las pruebas realizadas mostraron valores que se pueden considerar aceptables. Una desventaja de este prototipo, es el tendido que se debe realizar para enviar la información por internet a través de una red alámbrica; además, en caso de que el usuario no cuente con una conexión a internet, él tendrá que encargarse de cubrir este servicio. Otro aspecto a resaltar, es que el usuario para poder conocer el estado del





*Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente*

consumo, debe de ir hasta donde se encuentra el sistema para visualizar la información en la pantalla.

Un sistema que posee como objetivo brindar información acerca del estado del consumo antes de que llegue la fecha de corte, se encuentra descrito en [5], el trabajo describe el desarrollo de un sistema de monitoreo basado en un microcontrolador como dispositivo de procesamiento, un medidor de energía utilizando una bobina, un ESP-8622 como medio de comunicación inalámbrica WiFi, y una interfaz web para visualizar la información obtenida. Una desventaja que presenta el sistema es la necesidad de contar con una conexión a internet todo el tiempo; por lo que, si el usuario no cuenta con servicio de internet, no podrá visualizar la información del consumo en la página web.

Por otro lado, el sistema propuesto cuenta con una interfaz de teléfono inteligente que se comunica con el módulo de monitoreo a través de bluetooth, por lo que no se necesita contar con internet; de esta manera, el usuario podrá obtener mediante una app información del consumo, estimado del monto a pagar y días restantes para la fecha de corte. Adicionalmente, la interfaz de la app permite configurar fecha de corte y montos de tarifas. Las mediciones son almacenadas en una memoria MicroSD a razón de 5 segundos para que el usuario pueda revisar el histórico de su consumo.

### 3. Tarifas

En México la Comisión Federal de Electricidad (CFE) es quien suministra la energía eléctrica, y el esquema de cobro está definido la Comisión Reguladora de Energía (CRE), quien determina la metodología del cálculo de las tarifas para el suministro básico (Artículo 139 de la Ley de la Industria Eléctrica) [1]. Para la Ciudad de México y zona metropolitana el esquema de cobro se muestra en la tabla 1.

**Tabla 1.** Tarifas aplicadas a CDMX y zona metropolitana.

| Consumo                | Precio x KWh (MXN)    | Rango[KWh] |
|------------------------|-----------------------|------------|
| Consumo básico         | \$0.81                | 0 < 75     |
| Consumo intermedio     | \$0.98                | 75 < 140   |
| Consumo excedente      | \$2.87                | 140 < 250  |
| Alto Consumo Doméstico | \$107 (fijo) + \$4.79 | > 250      |

Para dar un contexto, si se tienen dos cargas de 100 W (como un par de luminarias) durante las 24 horas durante un mes, el consumo total es de **144 KWh** (200 W x 24 h x 30 días), por lo cual se deberá pagar (según la tabla 1) lo siguiente: (75 KWh x \$0.81) + (65 KWh x \$0.98) + (4 KWh x \$2.87) = **\$135.00**

Si se rebasa el rango de 250 KWh, se aplica el cobro de tarifa doméstica de alto consumo (DAC); tomando el ejemplo anterior, pero con cuatro cargas de 100W, el consumo mensual sería de 288 KWh por lo que se debe pagar: \$107 + (288 KWh x \$4.79) = **\$1486.50**, lo que equivale a 10 veces más el monto con respecto al caso anterior. De ahí la importancia de tener la información referente al consumo de manera fácil y puntual, y con ello elegir las medidas pertinentes para gestionar el consumo.





Alejandro Andrés Serapio Carmona, Ismael Díaz Rangel, Rodolfo Zola García Lozano, et al.

### 4. Metodología

El sistema propuesto está conformado por varias etapas que se pueden apreciar en el diagrama de bloques de la figura 1. En la figura 2 se presenta el diagrama esquemático del sistema.

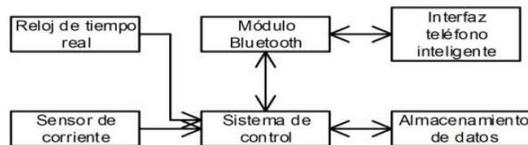


Fig. 1. Diagrama de bloques.

**Sistema de control:** Es el microcontrolador que se encarga de realizar todas las funciones lógicas del sistema, se utiliza un microcontrolador Atmega328 [6].

**Sensor de corriente:** Se utilizó el sensor de corriente de efecto Hall ASC712 versión para 30 A, este bloque obtiene la información de la corriente utilizada, la cual se almacena en una variable [7].

**Almacenamiento de datos:** La información de fecha, hora y consumo, es almacenada en una memoria MicroSD, utilizando un MicroSD shield. Extrayendo la memoria, el usuario podrá revisar el histórico de su consumo, abriendo el archivo generado mediante un programa de hoja de cálculo.

**Módulo Bluetooth:** Realiza la comunicación con un teléfono inteligente para enviar y recibir datos, utilizando un módulo HC05 [8].

**Interfaz para teléfono inteligente:** Es una aplicación desarrollada para dispositivos Android mediante appInventor2 [9], la cual es la interfaz visual que permite comunicar al usuario y al sistema.

**Reloj de Tiempo Real:** De este componente se obtiene la información de la fecha, sirve para mantener un registro por fecha y hora de los valores obtenidos, utilizando un módulo RTC DS1307 [10].

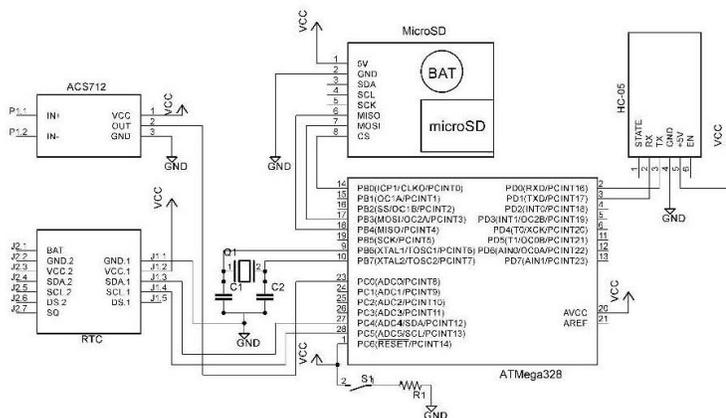


Fig. 2. Diagrama esquemático.





#### 4.1. Sensor de corriente ACS712

Es un sensor de corriente de efecto Hall, es de tipo invasivo, su funcionamiento consta de cuantificar la corriente a través de sus terminales (IN+, IN-), y utiliza el efecto Hall para producir en la salida un voltaje proporcional a la corriente que se encuentra circulando.

El valor en voltaje de la salida se observa en la figura 3, y muestra el comportamiento lineal del sensor.

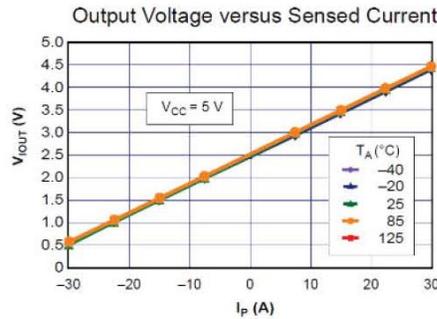


Fig. 3. Gráfica de voltaje vs corriente del ACS712.

#### 4.2. Medición de la corriente eléctrica

Para la obtención del consumo, se realizaron experimentos para caracterizar el comportamiento de la salida del sensor en AC (Corriente Alterna). Para la caracterización, se analizó la salida del sensor con la ayuda de un osciloscopio. Aplicando una carga de 2300 W se obtuvo la señal mostrada en la figura 4, donde se puede apreciar que entrega un voltaje en directa, pero presenta variaciones que van desde un voltaje mínimo 880 mV a un voltaje máximo 4.08 V.

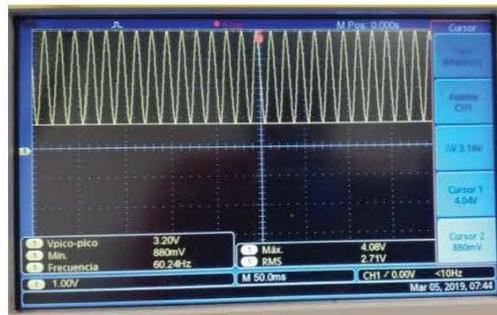


Fig. 4. Señal de salida del sensor con carga 2300W alimentada con AC.

Para calcular el valor de la corriente, se realizó un experimento en el que se conectaron diferentes cargas resistivas, y conociendo la linealidad del sensor, se empleó la ecuación de la recta para aplicar un análisis quedando definida como:





Alejandro Andrés Serapio Carmona, Ismael Díaz Rangel, Rodolfo Zola García Lozano, et al.

V = mI + B, (1)

donde:

- V = Voltaje de salida del sensor.
I = Corriente que circula por el sensor.
B = Voltaje cuando la corriente es cero.
m = pendiente o razón de cambio, estando definido por:

m = (V2 - V1) / (I2 - I1), (2)

De los valores mostrados por el fabricante y verificados experimentalmente, se extrajeron valores que se sustituyen en 2 y se obtiene:

m = (4.5-2.5) / (30-0) = 2/30 = 1/15, (3)

Ahora sustituyendo el valor de m en (1) y utilizando la ordenada al origen, encontramos la expresión:

V = (1/15)I + 2.5, (4)

Para obtener la potencia consumida es necesario trabajar con valores eficaces (RMS). Para obtener el voltaje RMS que entrega el sensor, se necesita obtener el voltaje de pico (Vp) máximo de un ciclo de la senoidal, por lo que se implementó un algoritmo que toma lecturas consecutivas durante 17 milisegundos con el convertidor analógico digital (dado que es la duración de un ciclo completo de una señal de 60 Hz), y cada lectura realizada es comparada con la anterior, con el fin de obtener el valor mayor, el cual representa al Vp. Para el cálculo de la corriente RMS por definición se utiliza:

IRMS = Ip / sqrt(2) despejando Ip = IRMS \* sqrt(2), (5)

Sustituyendo (5) en (4) e indicado en lugar V al Vp, tenemos:

Vp = (1/15)(IRMS \* sqrt(2)) + 2.5 despejando IRMS = (15(Vp-2.5)) / sqrt(2), (6)

Para calcular la energía consumida de manera instantánea (que en realidad será la potencia consumida en un ciclo de la senoidal) se utiliza:

Pins = VRMS x IRMS, (7)

donde:

- Pins = Potencia instantánea.
VRMS = Voltaje eficaz de la fuente de alimentación.
IRMS = Corriente eficaz obtenida en (6).

Para conocer el acumulativo de energía consumida se calcula con:

Pconsumida = Pins / delta t, (8)

donde delta t es igual a 3600, se debe a que se definió que cada lectura fuera tomada en intervalos de un segundo. Por la tanto, corresponde a 1/3600 parte de lo consumido en 1 hora.





Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente

El algoritmo programado para la carga de valores iniciales (tarifas y fecha de corte) almacenados en la eeprom del microcontrolador, estimación de la potencia instantánea, la transmisión de datos hacia la aplicación se observa en el diagrama de la figura 5.

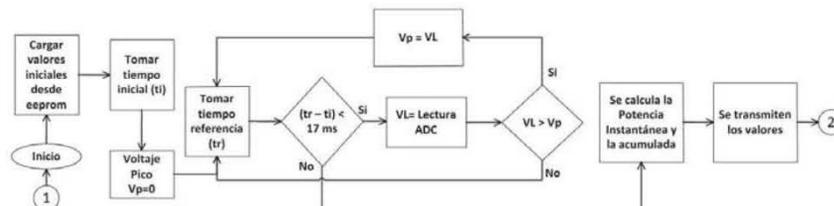


Fig. 5. Diagrama de flujo para obtener la potencia instantánea y su transmisión.

Para poder realizar las lecturas cada segundo, se desarrolló un algoritmo (figura 6) que asegura que haya transcurrido un segundo entre cada lectura, así se toman 3600 lecturas y sumatorias en el intervalo de una hora. Además, en esta sección, se monitorea la recepción de datos provenientes de la aplicación para la reconfiguración de parámetros y su almacenamiento en la eeprom.



Fig. 6. Diagrama de flujo para tomar lectura cada segundo y reconfiguración de parámetros.

4.3. Aplicación móvil

La aplicación tiene como objetivo servir de interfaz visual y de configuración entre el usuario y el sistema, el funcionamiento de la aplicación se muestra en el diagrama de flujo de la figura 7.

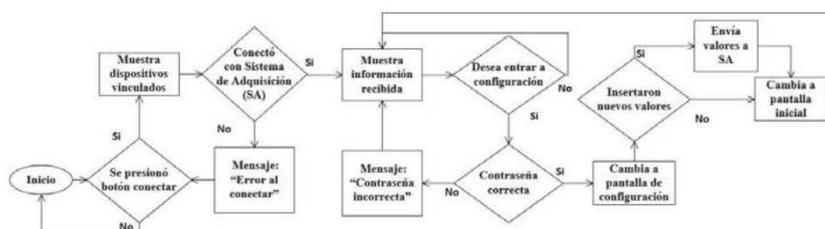


Fig. 7. Diagrama de flujo de la interfaz de la aplicación móvil.

Al ejecutar la aplicación (ver figura 8), la interfaz muestra al usuario la información del consumo, cuánto se va a pagar y cuántos días hacen falta para la fecha de corte.





Alejandro Andrés Serapio Carmona, Ismael Díaz Rangel, Rodolfo Zola García Lozano, et al.

Para visualizar la información, es necesario pulsar el botón conectar, con el que se enlaza la aplicación con el módulo de monitoreo a través de una comunicación por bluetooth. Cuando se presiona el botón de configuración se solicitará el ingreso de un código de seguridad, con lo que se accederá a la pantalla de la figura 9, los campos corresponden a las tarifas vigentes y la fecha de corte; el botón enviar actualiza la información del sistema, lo cual es importante para establecer la información que se presentará en la pantalla inicial referente al consumo.



Fig. 8. Pantalla inicial.



Fig. 9. Pantalla de configuraciones.

## 5. Pruebas y resultados

A continuación, se muestran los resultados de medir corrientes eficaces sobre diferentes cargas (tabla 2), todas ellas utilizando un voltaje de alimentación nominal de 127 VRMS, utilizando el sistema propuesto (SP) y un multímetro calibrado:

Tabla 2. Resultados de mediciones de corrientes.

| Carga              | Amperímetro [A] | SP [A]            | Diferencia [%] |
|--------------------|-----------------|-------------------|----------------|
| Lámpara 40 watts   | 0.336           | 0.38              | 14             |
| Lámpara 100 watts  | 0.58            | 0.6               | 4.5            |
| Plancha 1100W      | 8.65            | 8.48              | -1.9           |
| Estación de soldar | 5.82            | 5.46              | -6.5           |
| Plancha 2300W      | 17.21           | 16.99             | -1.25          |
| Parilla eléctrica  | 8.65            | 8.36              | -3.25          |
|                    |                 | <b>Promedio*:</b> | <b>-1.68</b>   |

\* Diferencia promedio sin considerar cargas pequeñas

Se pudo comprobar que para cargas superiores a 0.3 Amper los resultados son bastante aproximados, no tanto así para cargas inferiores; al ser un sistema para su implementación en hogares, no se considera una situación relevante, ya que los consumo suelen ser muy superiores a 0.3 A. Además, se debe considerar que existen fluctuaciones en la red de energía de la compañía proveedora y que eso también es





Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente

causa de las diferencias, aunque se trató de realizar las lecturas de manera simultánea, ese factor de error es a considerarse.

Para las pruebas de consumo instantáneo, se conectó el sistema propuesto en el centro de cargas de un hogar en serie con un sistema comercial [SC] de medición de consumo eléctrico con la capacidad de recolección de mediciones. Las mediciones se realizaron durante 7 días, obteniéndose casi tres mil muestras, la figura 10 muestra los resultados comparativos de los sistemas. Los resultados entregaron un valor promedio de mediciones de potencia instantánea para el sistema propuesto de 3.72 W, y para el sistema comercial de 3.7235 W, lo que implica una diferencia aproximada del 0.1 %.

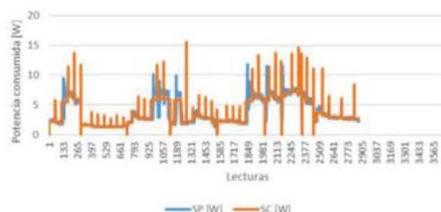


Fig. 10. Comparativa de mediciones.

Para determinar la equivalencia estadística, se aplicó a los datos el método de concordancia propuesto por Bland-Altman [11], el cual muestra una gráfica (figura 11) que indica niveles de correlación entre dos métodos de medición que observan un mismo fenómeno.

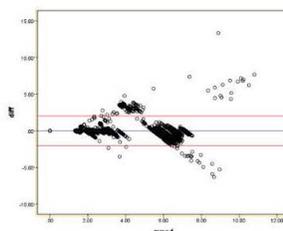


Fig. 11. Gráfica de Bland-Altman de las mediciones entre los dos sistemas.

Se puede observar como la gran mayoría de los datos se encuentran entre las líneas rojas (intervalo de confianza), lo cual permite determinar que los sistemas estadísticamente son equivalentes en sus mediciones.

Tabla 3. Resultados prueba medición consumo.

| Fecha      | CFE [kWh] | SP [kWh] |
|------------|-----------|----------|
| 17/06/2019 | 1         | 1        |
| 18/06/2019 | 0         | 0        |
| 19/06/2019 | 1         | 1        |
| 20/06/2019 | 1         | 1        |
| 21/06/2019 | 0         | 0        |
| 24/06/2019 | 1         | 1        |
| 25/06/2019 | 1         | 1        |





*Alejandro Andrés Serapio Carmona, Ismael Díaz Rangel, Rodolfo Zola García Lozano, et al.*

Para el consumo acumulado, se conectó el sistema a la salida del centro de cargas de un hogar, y se realizaron lecturas de manera manual durante 7 días con respecto al medidor de la CFE (mismo horario entre cada día), se calculó el consumo registrado en términos de KWh, ya que es la unidad de medida del medidor, y se compararon con el sistema propuesto, la tabla 3 muestra los resultados.

Se puede observar que los resultados son idénticos, por lo que se comprueba la eficacia del sistema para medir el consumo eléctrico en KWh.

## **6. Conclusiones**

Se desarrolló el sistema de medición con base a un microcontrolador Atmega328, donde se logró la implementación de algoritmos para la adquisición y procesamiento de datos, así como del control de almacenamiento y transmisión de datos. Cumpliendo de manera precisa con las cotas de tiempos necesarias para la adecuada cuantificación de consumo eléctrico.

Además, se diseñó una aplicación para el sistema operativo Android, la cual funciona como interfaz visual y de configuración, para que el usuario pueda observar la información del consumo proporcionada por el sistema, y también permite configurar algunas variables como tarifas y fecha de corte.

Se llevó a cabo el estudio del funcionamiento en AC del sensor de corriente; esto, a partir del análisis de cargas resistivas conocidas alimentadas con AC, se modeló un algoritmo para la interpretación de las lecturas cuando se trabaja con AC, con esa información, se calculó la potencia instantánea y a partir de esta última, se obtuvo el valor del consumo eléctrico, expresado en KWh.

Se realizó un análisis estadístico de la fiabilidad del sistema con respecto a un sistema comercial, obteniendo valores dentro del intervalo de confianza que indican la equivalencia de mediciones entre ambos sistemas, lo que indica que el sistema posee un grado de aproximación similar a un medidor de tipo comercial.

## **Referencias**

1. Comisión Federal de Electricidad (CFE): [https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/Tarifas/Tarifas\\_casa.asp?Tarifa=DACTAR1&anio=2018](https://app.cfe.mx/Aplicaciones/CCFE/Tarifas/Tarifas/Tarifas_casa.asp?Tarifa=DACTAR1&anio=2018). (Último acceso: 22 enero 2019)
2. Escoto Castillo, A., Sánchez Peña, L., Pérez Guardián, G.: Hogares y energía eléctrica en México. Revista Espinhaço, vol. V, pp. 30–43 (2017)
3. Riera, P.: Manual de economía ambiental y de los recursos naturales. Madrid, Parainfo (2005)
4. Gaibor, E.R. et al: Implementación de un prototipo de medidor de energía eléctrica residencial considerando la reducción de pérdidas no técnicas por hurto. Publicando, pp. 66–82 (2018)
5. Díaz Pérez, P., Clemente Arce, V.G., Ortiz Muñoz, J.A.: Sistema embebido basado en el monitoreo de consumo de energía eléctrica de electrodomésticos para el control del gasto. Tecnología Digital 8(1), 69–80 (2018)
6. Microchip: ATmega328 Datasheet. [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P\\_Datasheet.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf) (Último acceso: 14 noviembre 2018)





*Sistema de monitoreo de consumo eléctrico con interfaz para teléfono inteligente*

7. Allegro MicroSystems, Inc., «Allegro MicroSystems,»  
[https://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwig2Ohr9Y\\_jAhXDXc0KHXINACEQFjABegQIABAC&url=https%3A%2F%2Fwww.allegromicro.com%2F~%2Fmedia%2Ffiles%2Fdatasheets%2Facs712-datasheet.ashx&usg=AOvVaw1LkWwxiA9ZbTsL\\_UhbCLAp](https://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwig2Ohr9Y_jAhXDXc0KHXINACEQFjABegQIABAC&url=https%3A%2F%2Fwww.allegromicro.com%2F~%2Fmedia%2Ffiles%2Fdatasheets%2Facs712-datasheet.ashx&usg=AOvVaw1LkWwxiA9ZbTsL_UhbCLAp). (Último acceso: 5 octubre 2018)
8. Makezine: Make Magazine. 1 marzo 2014. [https://cdn.makezine.com/uploads/2014/03/he\\_he-05-user-instructions-bluetooth.pdf](https://cdn.makezine.com/uploads/2014/03/he_he-05-user-instructions-bluetooth.pdf). (Último acceso: 18 febrero 2019)
9. MIT: MIT APP INVENTOR. <https://appinventor.mit.edu/explore/ai2/tutorials.html>. (Último acceso: 24 enero 2019)
10. Maxim integrated: <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>. (Último acceso: 14 noviembre 2018)
11. Altman, D., Bland, J., Altman, D.G., Bland, J.M.: Measurement in medicine: the analysis of method comparison studies. *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 32, n° 3, pp. 307–317 (1983)

