

**UNIVERSIDAD NACIONAL DEL ALTIPLANO  
FACULTAD DE INGENIERIA MECÁNICA ELÉCTRICA  
ELECTRÓNICA Y SISTEMAS  
ESCUELA PROFESIONAL DE INGENIERÍA MECÁNICA  
ELÉCTRICA**



**CONTROL ENERGÉTICO EN VIVIENDAS CON REDES DE BAJA  
TENSIÓN MEDIANTE ASISTENCIA VIRTUAL PARA UNA MEJOR  
EFICIENCIA DE CONSUMO**

**TESIS**

**PRESENTADA POR:**

**CARLOS BRANTY CONZA MENDOZA**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO MECÁNICO ELECTRICISTA**

**PUNO – PERÚ**

**2020**

**UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO**  
FACULTAD DE INGENIERÍA MECÁNICA ELÉCTRICA ELECTRÓNICA Y  
SISTEMAS  
ESCUELA PROFESIONAL DE INGENIERÍA MECÁNICA ELÉCTRICA

**CONTROL ENERGÉTICO EN VIVIENDAS CON REDES DE  
BAJA TENSIÓN MEDIANTE ASISTENCIA VIRTUAL PARA UNA  
MEJOR EFICIENCIA DE CONSUMO**

TESIS PRESENTADA POR:

**CARLOS BRANTY CONZA MENDOZA**

PARA OPTAR EL TÍTULO PROFESIONAL DE:

**INGENIERO MECÁNICO ELECTRICISTA**

APROBADA POR EL JURADO REVISOR CONFORMADO POR:



PRESIDENTE :

  
M.Sc. MATEO ALEJANDRO SALINAS MENA

PRIMER MIEMBRO :

  
M.Sc. ANGEL MARIO HURTADO CHAVEZ

SEGUNDO MIEMBRO:

  
Ing. FELIPE CONDORI CHAMBILLA

DIRECTOR/ASESOR :

  
M.Sc. JOSE MANUEL RAMOS CUTIPA

ÁREA : Ingeniería Eléctrica

TEMA : Eficiencia de consumo por Asistencia Virtual.

FECHA DE SUSTENTACIÓN: 06 DE ENERO DEL 2020.

## DEDICATORIA

El presente trabajo de investigación va dedicado a mis padres Vicente y Rosa, a mis hermanos Saret y Fernando por su apoyo incondicional a lo largo de mi formación profesional, en los momentos difíciles iluminaron mi camino y me enseñaron a enfrentar con sabiduría y valentía los obstáculos de la vida.

## AGRADECIMIENTO

A Dios por darme la vida y permitirme cumplir mis metas, a mi familia por apoyarme y darme su comprensión y amor, a mi novia Samantha por apoyarme, impulsar mis metas y brindarme su amor.

Expresar mi sincero agradecimiento a los docentes de la Escuela Profesional de Ingeniería Mecánica Eléctrica por haberme brindado sus conocimientos y hacer de mi un profesional eficiente y competente.

Agradezco a la primera casa de estudios Universidad Nacional del Altiplano, por abrirme las puertas y acogerme en los años de estudios cursados.



## ÍNDICE GENERAL

<b>UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO .....</b>	<b>2</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>8</b>
<b>ÍNDICE DE TABLAS .....</b>	<b>10</b>
<b>ÍNDICE DE ANEXOS .....</b>	<b>11</b>
<b>ÍNDICE DE ACRÓNIMOS .....</b>	<b>12</b>
<b>RESUMEN.....</b>	<b>13</b>
<b>ABSTRACT .....</b>	<b>14</b>
<b>CAPÍTULO I.....</b>	<b>15</b>
INTRODUCCIÓN.....	15
1.1. Planteamiento del Problema de investigación .....	16
1.1.1. Problema general.....	17
1.1.2. Problemas específicos.....	17
1.2. Justificación del proyecto.....	17
1.2.1. Justificación técnica.....	17
1.2.2. Justificación social.....	18
1.2.3. Justificación económica.....	19
1.3. Objetivos de la investigación.....	19
1.3.1. Objetivos generales.....	19
1.3.2. Objetivos específicos.....	19
<b>CAPÍTULO II.....</b>	<b>21</b>
REVISIÓN DE LITERATURA .....	21
MARCO TEÓRICO.....	21
2.1. Antecedentes de la investigación.....	21
2.2. Sustento teórico.....	23
2.2.1. Concepto de smart grid.....	23
2.2.2. Energía eléctrica.....	23
2.2.2.1. <i>Tipos de energía eléctrica</i> .....	24
2.2.3 Domótica.....	25
2.2.4 Asistente virtual.....	26
2.2.5. Concepto de medidor de energía eléctrica.....	27
2.2.5.1. Contadores electromecánicos.....	28
2.2.5.2. <i>Contadores electrónicos</i> .....	30
2.2.6 Windows Presentation Foundation.....	30
2.2.6.1 <i>C sharp (c#)</i> .....	30

2.2.7. Conceptos basicos de arduino. ....	31
2.2.8. Concepto general de multímetro. ....	36
2.2.9. Concepto básico de amplificadores operacionales. ....	53
2.3 Circuito impreso. ....	56
2.3.1. Historia del circuito impreso. ....	57
2.3.2. Programas para el diseño de circuito impreso. ....	57
2.3.3. Fabricacion de circuitos impresos. ....	58
2.4. Hipótesis de la investigación ..... 77	
2.4.1. Hipótesis general. ....	77
2.4.2. Hipótesis específicas. ....	77
<b>CAPÍTULO III ..... 78</b>	
MATERIALES Y MÉTODOS. ....	78
DISEÑO DE LA METODOLOGÍA DE INVESTIGACIÓN ..... 78	
3.1. Método y diseño de la investigación. ....	78
3.1.1 Método. ....	78
3.1.2. Diseño. ....	78
3.2. Población y muestra. ....	78
3.3. Ubicación y descripción de la población. ....	78
3.3.1. Ubicación. ....	78
3.4. Materiales. ....	79
<b>CAPÍTULO IV ..... 82</b>	
RESULTADOS Y DISCUSIÓN. ....	82
4.1. Primer prototipo. ....	82
4.1.1. Modelo esquemático arduino uno. ....	82
4.1.2. Prueba de funcionamiento protoboard. ....	83
4.2. Segundo prototipo. ....	85
4.2.1. Segundo modelo esquemático arduino uno. ....	85
4.2.2. Prueba de funcionamiento de protipo n° 2. ....	85
4.3. Tercer prototipo. ....	87
4.3.1. Modelo esquemático para arduino due. ....	87
4.3.2. Mlaboración y prueba de funcionamiento de protipo n° 3. ....	88
4.4. Prototipo final. ....	90
4.4.1. Modelo esquemático para arduino due p.final. ....	91
4.4.2. Elaboración y puebas del prototipo final. ....	91
4.5. Creación de asistente virtual e interfaz gráfico. ....	93
4.5.1. Partes del proyecto en wpf. ....	93

4.5.2. CREACIÓN DEL INTERFAZ DE HANABI. ....	94
4.5.3. ELABORACION DE SISTEMA DE DETECCIÓN DE FUGA ELÉCTRICA A TIERRA Y CORTO CIRCUITO. ....	101
<b>CAPÍTULO V.....</b>	<b>109</b>
CONCLUSIONES .....	109
5.1 Conclusión 1. ....	109
5.2 Conclusión 2. ....	109
5.3 Conclusión 3. ....	109
<b>CAPÍTULO VI.....</b>	<b>110</b>
RECOMENDACIONES .....	110
<b>CAPÍTULO VII.....</b>	<b>111</b>
REFERENCIAS.....	111
<b>ANEXOS.....</b>	<b>114</b>

## ÍNDICE DE FIGURAS

Figura.2.1: Contador electromecánico.....	28
Figura 2.2: Representación esquemática de medidor electromecánico .....	29
Figura 2.3: Diagrama de funcionamiento de medidor electrónico Trifásico .....	30
Figura 2.4: El universo Arduino .....	31
Figura 2.5: Arduino Uno .....	32
Figura 2.6: Arduino MKR 1000 .....	33
Figura 2.7: Arduino Leonardo .....	34
Figura 2.8: Arduino Mega 2560.....	35
Figura 2.9: Arduino Due .....	36
Figura 2.10: Donald Macadie el primer multímetro .....	37
Figura 2.11: Multimetros.....	38
Figura 2.12: Partes de una Pinza Amperimétrica .....	39
Figura 2.13: Sensor Hall.....	42
Figura 2.14: Tipos de salida sensor AC SCT-13 .....	43
Figura 2.15: conexionado transformador de corriente SCT-13.....	45
Figura 2.16: Esquema sensor SCT-13. ....	45
Figura 2.17: componentes internos de sensor SCT-013. ....	46
Figura 2.18: Características SCT-013 .....	47
Figura 2.19: Conector de 3 núcleos.....	48
Figura 2.20: Curva Característica Voltio – Amperios SCT-013.....	49
Figura 2.21: Transformador de corriente ZMPT101B .....	50
Figura 2.22: Parámetros de ZMPT101B.....	52
Figura 2.23: Esquema de uso .....	52
Figura 2.24: Símbolo de Amplificador operacional. ....	54
Figura 2.25: Amplificador operacional LM358. ....	54
Figura 2.26: Características de fabricación. ....	56
Figura 2.27: Protoboard .....	58
Figura 2.28: Vista interna de un Protoboard.....	59
Figura 2.29: Diagrama Esquemático (schematic).....	59
Figura 2.30: Placa de fibra de vidrio para circuitos impresos. ....	60
Figura 2.31: Técnica de fabricación con tinta indeleble.....	61
Figura 2.32: Diseño de pistas de circuito a imprimir.....	62
Figura 2.33: Técnica de Serigrafiado .....	64
Figura 2.34: Esquema de proceso de fabricación de circuitos impresos. ....	64
Figura 2.35: Pre producción de circuito impreso. ....	65
Figura 2.36: Preparación de herramientas gráficas. ....	66
Figura 2.37: Imprimir capas internas. ....	66
Figura 2.38: Atacado de las capas internas.....	67
Figura 2.39: Inspección óptica automática. ....	68
Figura 2.40: Apilado y laminado.....	69
Figura 2.41: Taladrado de PCB. ....	70
Figura 2.42: Deposición química de cobre. ....	71
Figura 2.43: Proceso gráfico de capas exteriores.....	71
Figura 2.44: Proceso de plateado.....	72
Figura 2.45: Atacado de capas externas. ....	73
Figura 2.46: Inspección óptica externa. ....	73
Figura 2.47: Máscara de soldadura. ....	74

Figura 2.48; Acabado superficial. ....	74
Figura 2.49: Perfil y cortado. ....	75
Figura 2.50: Comprobación eléctrica. ....	76
Figura 2.51: Inspección final.....	77
Figura 3.52: Placa de cobre perforada. ....	80
Figura 4.53: Esquemático de conexión Arduino UNO.....	82
Figura 4.54: Rectificación de media onda con A.O. LM358. ....	83
Figura 4.55: Conexionado protoboard. ....	83
Figura 4.56: Medición de voltaje y corriente con Arduino UNO.....	84
Figura 4.57: Medición de voltaje y corriente por Multímetro digital. ....	84
Figura 4.58: Segundo esquemático Arduino UNO. ....	85
Figura 4.59: Prototipo en placa con perforaciones (Vista superior). ....	86
Figura 4.60: Prototipo en placa de cobre con perforaciones (Vista inferior) ....	86
Figura 4.61: Lecturas de medición del segundo prototipo. ....	87
Figura 4.62: Esquemático Arduino Due (Tercer Prototipo).....	88
Figura 4.63: Diseño de Circuito impreso para el 3er prototipo en EASYEDA.....	89
Figura 4.64: Circuito impreso para Arduino Due (Vista superior 3er prototipo) ....	90
Figura 4.65: Circuito impreso para Arduino Due (Vista inferior 3er prototipo).....	90
Figura 4.66: Esquemático para el prototipo final.....	91
Figura 4.67: Modelo de PCB elaborado en EasyEda (Vista superior e inferior). ....	92
Figura 4.68: Circuito impreso final fabricado por JLCPCB (Vista superior e inferior).....	92
Figura 4.69: Circuito impreso final ensamblado (Vista superior e inferior). ....	93
Figura 4.70: Partes del proyecto HanabiFinal. ....	94
Figura 4.71: Interfaz principal HanabiFinal. ....	95
Figura 4.72: Interfaz de configuración del sistema. ....	96
Figura 4.73: Interfaz del botón Gráficos (InterfazDash).....	96
Figura 4.74: Interfaz de conexión con puerto serial. ....	97
Figura 4.75: Comandos del Sistema. ....	98
Figura 4.76: Interfaz de Gráficos y Datos. ....	99
Figura 4.77: Datos de prueba de fuga a tierra. ....	104
Figura 4.78: Conexión modulo relé con Arduino UNO.....	107
Figura 4.79: Caja 3d para componentes.....	108

## ÍNDICE DE TABLAS

Tabla N° 1.1: Reclamos II semestre 2018 Electro Puno. ....	18.
Tabla N° 2.2: Modelos de sensores de corriente SCT-013 .....	46.
Tabla N° 2.3: Modelos de sensores de corriente SCT-013 .....	47.
Tabla N° 2.4: Parámetros de sensor SCT-013-030 .....	48.
Tabla N° 2.5: Parámetros de sensor SCT-013-000 .....	49.
Tabla N° 2.6: Parámetros de sensor ZMPT101B .....	51.
Tabla N° 4.7: Pliego tarifario OSINERGMIN .....	99.
Tabla N° 4.8: Resultados de prueba de fuga eléctrica a tierra .....	104.

## ÍNDICE DE ANEXOS

ANEXO 1 Código del 1er Prototipo Arduino Uno (Arduino IDE): .....	114
ANEXO 2 Código del 2do prototipo Arduino Uno en (Arduino IDE).....	114
ANEXO 3 Código del 3er prototipo Arduino Due en (Arduino IDE).....	114
ANEXO 4 Código de prototipo final Arduino Due en (Arduino IDE).....	115
ANEXO 5 Código de interfaz principal de HanabiFinal en (Xamarin Forms). .....	117
ANEXO 6 Código de Interfaz principal HanabiFinal en (C#).....	124
ANEXO 7 Código de Ventana de ajustes de HanabiFinal en (C#). .....	134
ANEXO 8 Código de interfaz de Visualización de Datos HanabiFinal en(C#).....	138
ANEXO 9 Código de interfaz de control de Relé y entrada serial en (C#). .....	139
ANEXO 10 Código de interfaz de Base de datos de HanabiFinal en (C#). .....	143
ANEXO 11 Parámetros de entrada y salida del prototipo final. ....	150
ANEXO 12 PARÁMETROS DE INTERFAZ DE SOFTWARE HANABI.....	153
ANEXO 13 PRESUPUESTO DE PROTOTIPO (FINAL).....	155
ANEXO 14 PRESUPUESTO TOTAL DE INVESTIGACIÓN.....	156

## ÍNDICE DE ACRÓNIMOS

<b>WPF</b>	: Windows Presentation Foundation.
<b>OSINERGMIN</b>	: Organismo Supervisor de la Inversión en Energía y Minas.
<b>KWH</b>	: Kilo Watt por Hora.
<b>WSN</b>	: Wireless Sensor Network.
<b>IDE</b>	: entorno de desarrollo integrado
<b>AVO</b>	: Amperio, Voltio, Ohmios.
<b>HET</b>	: Hall Effect.
<b>VAC</b>	: Voltios de Corriente Alterna.
<b>AC</b>	: Corriente Alterna.
<b>PCB</b>	: Printed Circuit Board (Placa de Circuito impreso).
<b>A.O.</b>	: Amplificador Operacional.



## RESUMEN

Un gran problema en cuanto a la eficiencia y ahorro de consumo energético es la mala monitorización de este mismo, así como el uso inconsciente de energía eléctrica. Casi el 50% de reclamos en las empresas de distribución de energía eléctrica del sur de nuestro país son por consumo excesivo y este a su vez en su mayoría es debido a un mal manejo y mal uso de energía eléctrica; otra de las causas más frecuentes de consumo excesivo, es el mal dimensionamiento de cables en el conexionado de un hogar, así como frecuentes fugas a tierra de electricidad. A consecuencia de varios factores entre ellos, los anteriormente mencionados se ha diseñado un Asistente Virtual Basado en Speech Recognition de Microsoft, que nos permite mediante un micrófono la conversión de voz a texto y de texto a voz en la plataforma de Windows 10. Para el la medición de energía eléctrica y su posterior monitorización se usó un Arduino Due que nos permite conectar a un transformador de corriente no invasivo SCT-13 así como un transformador de voltaje AC - ZMPT101B que nos permite medir la tensión, gracias a esto se pudo elaborar un medidor de consumo. Para visualizar nuestro medidor digitalmente y para interactuar con nuestro asistente virtual se creó un software en lenguaje de programación Visual Studio C# Windows Presentation Foundation.

**Palabras Clave:** Ahorro energético, Arduino Due, Visual Studio.

## ABSTRACT

A major problem in terms of efficiency and saving of energy consumption is its poor monitoring, as well as the unconscious use of electric energy. Almost 50% of claims in the electricity distribution companies in the south of our country are due to excessive consumption and this in turn is mostly due to mishandling and misuse of electricity; Another of the most frequent causes of excessive consumption is the poor sizing of cables in the connection of a home, as well as frequent leaks to the ground of electricity. As a result of several factors, the aforementioned has designed a Virtual Assistant based on Microsoft Speech Recognition, which allows us to convert the voice to text and text to speech on the Windows 10 platform using a microphone. The measurement of electrical energy and its subsequent monitoring was used an Arduino Due that allows us to connect to a non-invasive current transformer SCT-13 as well as an AC - ZMPT101B voltage transformer that allows us to measure the voltage, thanks to this it was possible to elaborate A consumption meter. To visualize our meter digitally and to interact with our virtual assistant, software was created in the programming language Visual Studio C # Windows Presentation Foundation.

**Keywords:** Energy saving, Arduino Due, Visual Studio.

## CAPÍTULO I

### INTRODUCCIÓN

Hoy en día la mayoría de medidores eléctricos del Perú son electrónicos, estos contadores de KWH, permite a la empresa comercializadora de energía y al consumidor visualizar su consumo total; permitiendo a la empresa comercializadora llevar el control de medición para poder hacer el respectivo cobro por la energía eléctrica consumida, todo este proceso se realiza según la ley del sector eléctrico y las tarifas establecidas según las características urbanas y características de consumo.

Para el mejor control de consumo eléctrico por parte del consumidor se realizó y diseñó el software de nombre “Hanabi” soportado en la plataforma Windows; este permite al usuario interactuar con su consumo eléctrico en tiempo real, así como realizar pruebas en prevención de fuga eléctrica y cortos circuitos.

El dispositivo encargado de la medición y procesamiento de datos de intensidad de corriente y voltaje es un Arduino due que, en conjunto a sus respectivos sensores, calcula la “Potencia”, “Intensidad de Corriente” y “Voltaje”; datos necesarios para el cálculo de consumo eléctrico en KWh.

Para el funcionamiento del software creado, se necesita un computador portátil con Windows 10 que nos permitirá almacenar los datos en tiempo real y que constantemente estará monitoreando algún tipo de falla en el conexionado eléctrico del hogar.

Capítulo I: Se describirá el planteamiento del Problema, los Objetivos planteados a los cuales se quiere llegar en el Proyecto de Investigación

Capítulo II: En el presente capítulo se desarrollará la revisión de literatura para dar a conocer los conceptos de Smart Grid, Inteligencia Artificial, Medidores inteligentes, eficiencia energética, Smart Home, WPF C#, Internet Of Things

Capítulo III: En el presente capítulo se describirá la metodología de investigación científica, tipo, diseño, ubicación, técnicas e instrumentos.

Capítulo IV: En el presente capítulo se presentará el análisis de los resultados, en tablas o Gráficos según corresponda.

Capítulo V: En el presente capítulo se darán a conocer las conclusiones obtenidas de acuerdo al estudio realizado.

### **1.1. Planteamiento del Problema de investigación**

En la actualidad la gran parte de los reclamos hacia el servicio de las empresas comercializadoras de energía eléctrica por parte de sus clientes son “Excesivo consumo”, “la no entrega de sus respectivos recibos de electricidad”, entre otros. En el primer caso, la gran mayoría no se debe a un problema en el medidor instalado, ya que las empresas periódicamente hacen inspección a sus dispositivos medidores para prevenir cualquier tipo de falla, esto se traduce en que el problema de excesivo consumo no se encuentra en el medidor, si no en el conexionado eléctrico interno del hogar, esto puede deberse a una fuga eléctrica a tierra que es causada por el deterioro del material aislante del conductor, o también puede ser a causa de un mal dimensionamiento de cables; otro gran causante, es la desinformación y la mala administración en el consumo de energía eléctrica haciendo un ineficiente uso de este mismo.

Debido a que los usuarios de energía eléctrica no tienen el fácil acceso a la información de consumo en tiempo real, pueden llegar a hacer un uso no eficiente de

energía eléctrica, además de no contar con el conocimiento para comprobar si el conexionado de su hogar tiene algún tipo de falla hace que el consumo energético y por consiguiente el costo de este aumentara proporcionalmente.

### **1.1.1. PROBLEMA GENERAL.**

El 49% de reclamos en la empresa Electro Puno S.A.A. es debido a excesivo consumo, lo que quiere decir que ese gran porcentaje de personas están descontentas y no saben el por qué existe un consumo desmedido en su hogar, a pesar de que ellos aseguran de no consumir tal potencia.

### **1.1.2. PROBLEMAS ESPECIFICOS**

- ✓ ¿Cómo identificar la causa del excesivo consumo de ese 49% de usuarios en su hogar?
- ✓ ¿Cómo mejorar la eficiencia de consumo desde un ordenador asistido por inteligencia artificial en un hogar y tener un mejor control de energía eléctrica?
- ✓ ¿Cuánto y que porcentaje pierde en energía eléctrica y monetaria un usuario al tener una fuga eléctrica y un consumo ineficiente?

## **1.2. JUSTIFICACION DEL PROYECTO**

En la actualidad gran parte de los reclamos y descontentos de los clientes de una empresa comercializadora de energía eléctrica proviene del excesivo consumo ya que los costos de este son elevados si aparente explicación, esto puede deberse al difícil acceso a la información de consumo en tiempo real.

### **1.2.1. JUSTIFICACION TÉCNICA.**

Los dispositivos que pueden interactuar con el alrededor en este caso entre mediciones de energía, asistente virtual y ser humano ha aumentado y se ha hecho posible y de mayor accesibilidad.

Muchos desarrolladores de diversos países han intentado incorporar este tipo de tecnología con éxito haciendo más eficiente el consumo energético; gracias a la tecnología que se aplicara mejorara la calidad y eficiencia en el uso de energía eléctrica.

### 1.2.2. JUSTIFICACION SOCIAL.

Hoy en día ya existen muchos hogares que aplican el concepto de Smart Home, Así como individualmente el concepto de asistencia virtual, la sociedad con la incorporación de nuevas tecnologías requiere de un crecimiento en todos los ámbitos y uno de ellos es el sector energético, gracias a la aplicación de estas nuevas tecnologías será más fácil e intuitivo para un usuario llevar con inteligencia y eficiencia el consumo eléctrico de su hogar; así mismo este proyecto busca resolver un problema social respecto al excesivo consumo ya que gran parte de los usuarios de energía eléctrica tienen este inconveniente como se muestra en la siguiente tabla.

Tabla N° 1.1 Reclamos II semestre 2018 Electro Puno.

TIPO DE RECLAMO	RECIBIDOS Y ATENDIDOS
Excesivo consumo de facturación	2563
Recibo no llega al usuario	1328
Daños y perjuicios	749
Calidad de producto	483
Corte y reconexión	213
Deuda de terceros	8
Calidad de suministro	12
Instalaciones defectuosas	4

Medidor dañado o malogrado	17
Alumbrado publico	35
Robo de medidor	3
Reubicación de medidores	6
<b>Total</b>	<b>5421</b>

Fuente: Electro Puno S.A.A.

### 1.2.3. JUSTIFICACIÓN ECONÓMICA.

Debido a un ineficiente control de consumo eléctrico o a un problema de conexionado, hace que el costo de su consumo sea mayor; el impacto que causara la asistencia virtual en el consumo eléctrico se verá reflejado en la economía del usuario mejorando y reduciendo los costos a pagar el fin de mes.

## 1.3. OBJETIVOS DE LA INVESTIGACIÓN

### 1.3.1. OBJETIVOS GENERALES.

Mejorar la calidad de uso de energía mediante la aplicación de domótica y asistencia virtual haciendo la experiencia para el usuario intuitiva y fácil de usar.

### 1.3.2. OBJETIVOS ESPECIFICOS.

- ✓ Mejorar la eficiencia de consumo energético así contribuir al ahorro de este.
- ✓ Controlar, monitorear y automatizar mediante la asistencia virtual el nivel de consumo eléctrico.
- ✓ Tener un comparativo en tiempo real de consumo eléctrico entre los diferentes circuitos del hogar en KWh y en S/.

- ✓ Enfocar este tipo de tecnología a la eficiencia de consumo eléctrico, así como habilitar la posibilidad a un usuario de detectar fugas eléctricas a tierra y corto circuito.



## CAPÍTULO II

### REVISIÓN DE LITERATURA

#### MARCO TEÓRICO

##### 2.1. ANTECEDENTES DE LA INVESTIGACIÓN.

En la actualidad existen diferentes tipos de investigación, aplicación de la domótica y la asistencia virtual; en cuanto al tema de ahorro energético, eficiencia y automatización se ha investigado en gran medida los siguiente:

- ❖ Según JunSung Kim el primer secretario de la embajada de la República de Corea en la Republica del Perú en su conferencia (12 de Abril 2019) que titula: “Las ciudades inteligentes de Corea” Explica que una ciudad inteligente utiliza una amplia gama de tecnologías electrónicas y digitales para:
  - Mejorar la provisión de servicios públicos y el bienestar de los ciudadanos.
  - **Reducir los costos y el consumo de recursos.**

Según la información proporcionada en dicha conferencia en Corea del sur ya existen servicios públicos inteligentes entre ellos sensores de monitoreo y control de consumo energético.

Figura 2.1: Control de consumo energético en Corea



Fuente: Conferencia JunSung Kim(2019)

- ❖ Según (Raihaan Kamarudin & Aiman Md Yusof, 2013) en su publicación de journal título:” Low Cost Smart Home Automation via Microsoft Speech Recognition”, es posible automatizar un hogar mediante la utilización de speech recognition de Microsoft y concluye que mediante el reconocimiento de voz y una monitorización por computadora permite y facilita a las personas especialmente a aquellas que sufren algún tipo de discapacidad como ceguera entre otros a manejar las funciones de un hogar de manera interactiva.  
  
El proyecto de automatización de hogar fue un éxito con muy bajos errores en cuanto al reconocimiento de voz.
- ❖ Según (Ramya & Poomurugan, 2015) se logró realizar un medidor inteligente en Wireless Sensor Network (WSN) mediante la aplicación de sensores inalámbricos, este proyecto dio como resultado mediciones aceptables en cuanto precisión para su uso industrial.
- ❖ Según (ZEGARRA PINTO, 2017) en su Tesis titula: ANALISIS DE NUEVO SISTEMA DE MEDICIÓN CENTRALIZADA DE ENERGÍA ELÉCTRICA CON MEDIDORES INTELIGENTES EN ÁREA DE LA REGIÓN AREQUIPA; Concluye: que para ser consistente con los análisis realizados hasta ahora y comparando con los resultados obtenidos en los últimos años, se percibe que el costo de implementar y operar nuevos sistemas de medición, junto con análisis de los beneficios que pudieran percibirse tanto por parte de la empresa distribuidora como de los usuarios son positivos. Además, se ha realizado un análisis de largo plazo, con el fin de estimar en qué medida la respuesta de los clientes ante nuevos estímulos podría afectar la operación del mercado eléctrico. En particular, se estudió la variación que se podría observar en los niveles de precio, costos y

descuentos que se podría considerar ya que se realiza trabajos de forma remota, ingresos. Las redes inteligentes en general son un conjunto de tecnologías que permiten una nueva forma de operación de los sistemas eléctricos, empezando desde la distribución, transmisión y generación.

- ❖ Según (SAMANIEGO IDROVO & VELESACA ORELLANA, 2016) en su trabajo de tesis Título: “Diseño e implementación de un medidor de energía electrónico para vivienda, con orientación y prevención de consumo y ahorro energético ” Concluye que: mediante el diseño de un medidor electrónico con Arduino Uno y RaspBerry Pie se logra crear un medidor con niveles de error muy bajos con un error aproximado de +- 2,5KWh mensuales lo cual es un resultado aceptable para la medición de energía eléctrica en un hogar.

## **2.2. SUSTENTO TEÓRICO.**

### **2.2.1. CONCEPTO DE SMART GRID.**

Las Smart Grid se basan en combinar las "tecnologías de información y comunicación" con la "automatización y control" desde la generación de la energía, el transporte y distribución, el almacenamiento eléctrico, hasta el consumo de la energía para conseguir lo que se domina también Redes Eléctricas Inteligentes.

Así mismo el Smart Grid se encarga de compartir información entre distribuidores, comercializadores y usuarios finales de electricidad de esta manera se logra equilibrar la oferta y demanda entre proveedor y consumidor (SmartGridsInfo, 2019).

### **2.2.2. ENERGÍA ELÉCTRICA.**

Hoy en día la energía eléctrica se ha convertido en una de las más grandes necesidades del ser humano para entender el concepto podemos decir que la Energía eléctrica es causada por el movimiento de las cargas eléctricas en el interior de los materiales

conductores. Esta energía produce, fundamentalmente, 3 efectos: luminoso, térmico y magnético. Ej.: La transportada por la corriente eléctrica en nuestras casas y que se manifiesta al encender una bombilla (Raffino, 2019).

### **2.2.2.1. TIPOS DE ENERGÍA ELÉCTRICA.**

Podemos hablar de distintas formas de electricidad:

**Básica.** La producida por el contacto de una carga positiva y una negativa, es el tipo de corriente que empleamos en nuestros aparatos de uso cotidiano y que permite el funcionamiento de los bombillos.

**Estática.** Aquella producida por la fricción de dos cuerpos susceptibles de cargarse eléctricamente, como ocurre al frotar un peine con un pañuelo de ciertos materiales, o al acercar un brazo a la pantalla de un televisor antiguo.

**Dinámica.** De la mano con otras formas de energía como la química, es la forma de electricidad manejable y controlable, a partir del flujo constante de electrones por medio de un conductor.

**Conductual.** Aquella corriente eléctrica que se mantiene en continuo movimiento por los conductores, como ocurre en los circuitos.

**Electromagnética.** La electricidad propia de los campos electromagnéticos, como en los grandes imanes industriales.

Por otro lado, se conocen dos formas de corriente eléctrica, gracias a las experiencias y estudios de Nikola Tesla: la corriente ordinaria y la alterna, siendo esta última la que varía cíclicamente en su magnitud y sentido (Raffino, 2019).

### 2.2.3 DOMÓTICA.

La domótica es un término que refiere a todo aquel elemento que pueden supervisarse y controlarse de manera sistemática; este tiene la finalidad de poder proporcionar un eficiente uso de la energía, así como aportar seguridad y comodidad; permitiendo de esta manera que exista una comunicación entre el beneficiario y el sistema (Conceptodefinicion.de, 2019).

La domótica nos permite recolectar información mediante sensores de los diferentes elementos de una casa, así como también nos permite controlarlos. La domótica ofrece a las personas una serie de factores que contribuyen con el mejoramiento de la calidad de vida por ejemplo nos permite el ahorro energético, ya que ejecuta de manera inteligente todo lo relacionado con la iluminación, los electrodomésticos, el agua caliente del baño, etc. Promueve la accesibilidad, facilitando el manejo de las piezas del hogar a las personas que sufren de alguna discapacidad. Aporta seguridad por medio de la vigilancia automática de personas, bienes y animales, a través de cámaras de vigilancia, alarmas personales, cierre automático de todas las ranuras o aberturas. Garantiza la comunicación por medio del control de supervisión remoto de la casa a través de su teléfono o computadora, que facilita el ingreso de avisos sobre alguna anomalía, así como la información sobre el funcionamiento de los equipos y sus instalaciones (Conceptodefinicion.de, 2019).

**Figura 2 Interfaz Domótico Smart Home**

Fuente: (Conceptodefinición.de, 2019).

#### **2.2.4 ASISTENTE VIRTUAL.**

Un asistente virtual es un software informático capaz de reconocer el lenguaje utilizado por un usuario permitiendo reconocer y descifrar a un computador el lenguaje humano, siendo este capaz de responder preguntas, así como obedecer comandos pre definidos de voz.

desarrolladores de software y hardware de la talla de Google, Apple o Microsoft han incorporado en sus sistemas operativos de forma nativa asistentes virtuales que mejoran la experiencia de uso de sus smartphones, tabletas electrónicas y demás dispositivos, disponiendo en todo momento y en todo lugar de nuestro propio asistente virtual con solo pulsar un botón o pronunciar una frase. Siri de Apple, Cortana de Microsoft o Google Now de Google son los exponentes de esta tecnología implementada en millones y millones de dispositivos.

Estos son algunos usos que se le ha dado a un Asistente virtual:

- Encender y apagar luminarias de un hogar.

- Indicarnos el clima y temperatura de un lugar.
- Apuntar una cita en nuestra agenda generando automáticamente una alarma de recordatorio.
- Indicar el recorrido más próximo de un punto a otro mediante el uso de una aplicación de mapas.
- Comprar entradas de cine, teatro o música.
- Reservar una mesa en un restaurante, una habitación en un hotel o un viaje de avión.
- Llamar a un taxi para que nos recoja en la ubicación en donde nos encontramos.
- Llamar o mandar un mensaje a cualquier contacto de la agenda al indicar su nombre.

El asistente virtual se compone de 2 elementos principales

- **Interfaz Externa:** Es el interfaz gráfico o físico mediante el cual interactuamos con el asistente ayudando al usuario a tener una experiencia más cómoda.
- **Interfaz Interna:** Es la programación que hay de tras de este, esta programación se encarga de dotar a nuestro asistente de una inteligencia virtual capaz de reconocer y procesar el lenguaje humano.

### 2.2.5. CONCEPTO DE MEDIDOR DE ENERGÍA ELÉCTRICA.

El medidor de energía, conocido también como contador, es un equipo que se emplea para medir la energía suministrada a los clientes. Aplicada una tarifa establecida por ley, posibilita a la Empresa realizar una facturación adecuada de la potencia y energía consumida.

El Medidor de energía Eléctrica consta de dos partes:

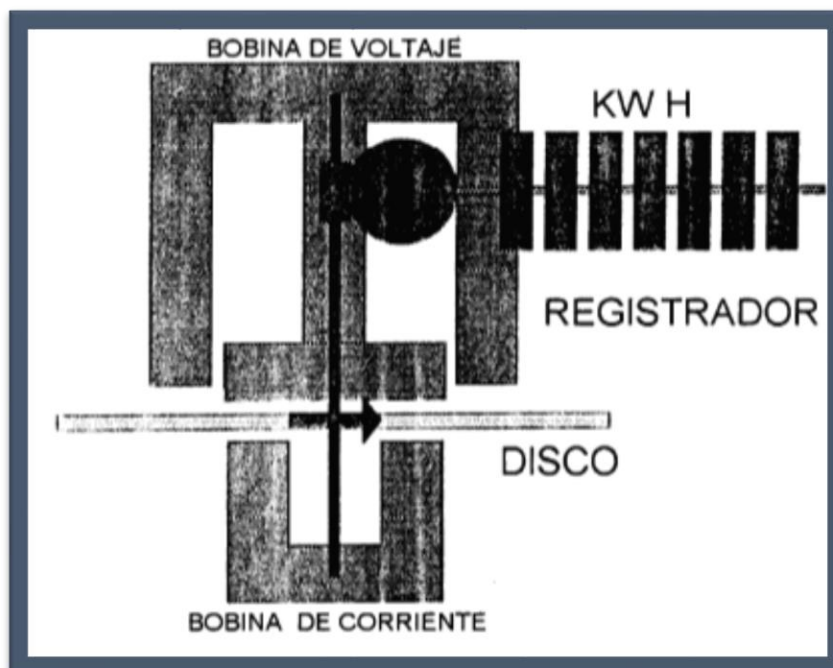
- **Caja principal o verificadora:** Es donde se encuentra el mecanismo del medidor, compuesta por: bobina de tensión, bobina de corriente, el disco giratorio y el numerador.

- **Caja de conexiones o bornera:** Como lo indica su nombre es donde se realizan las conexiones del medidor.

### 2.2.5.1. CONTADORES ELECTROMECAÑICOS

El contador electromecánico tiene un disco que gira a una velocidad proporcional a la corriente consumida. Está contenido por un núcleo de chapa magnética en el que van montados dos bobinas, una en serie por el conductor que circula la corriente denominada bobina de intensidad y la otra en derivación sobre los conductores denominado bobina de tensión.

Figura.2.1: Contador electromecánico.



Fuente: Redes eléctricas (Ríos, s. f.)

El medidor está constituido por las siguientes partes:

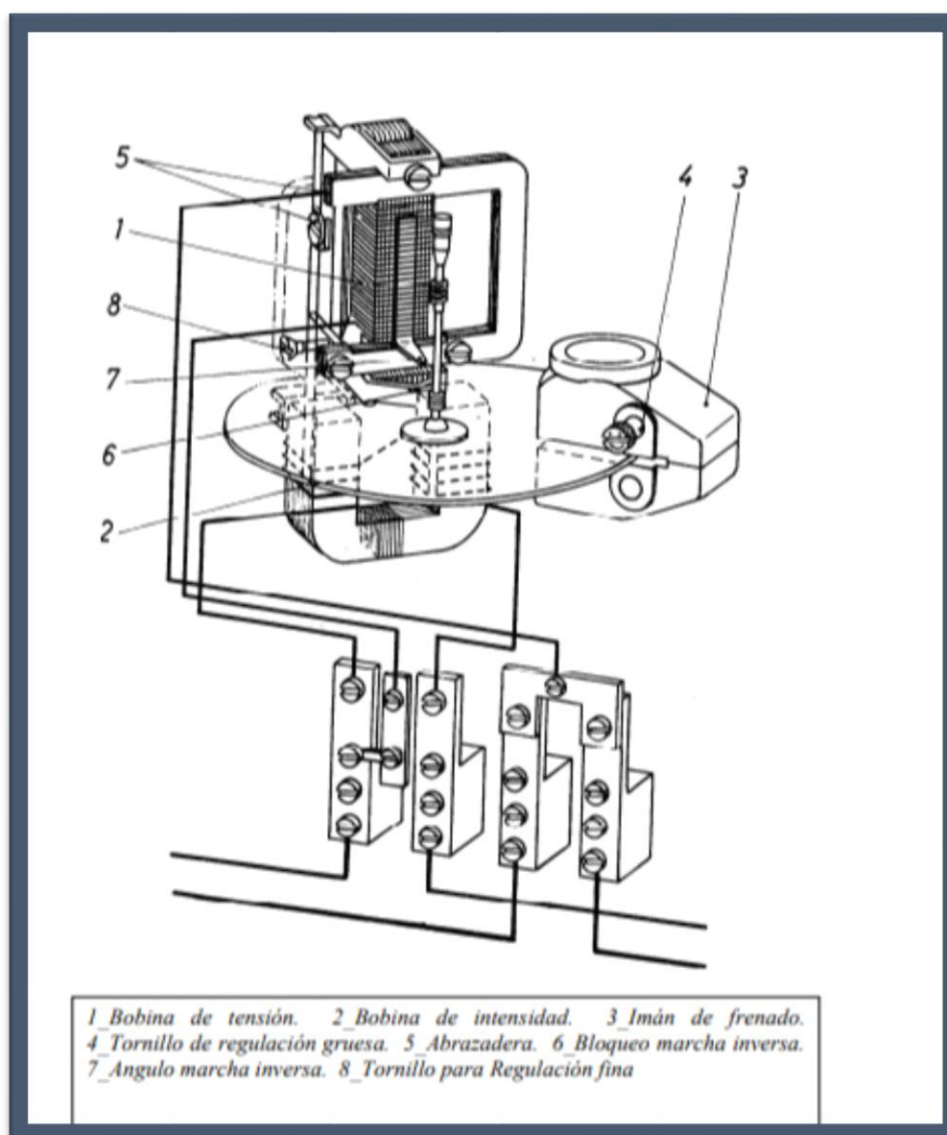
1. Bobina de Tensión
2. Bobina de Intensidad
3. Imán de frenado



- 4. Regulación fina
- 5. Regulación gruesa
- 6. Disco
- 7. Sistema de Transmisión
- 8. Terminales de conexión

La representación esquemática:

**Figura 2.2: Representación esquemática de medidor electromecánico**

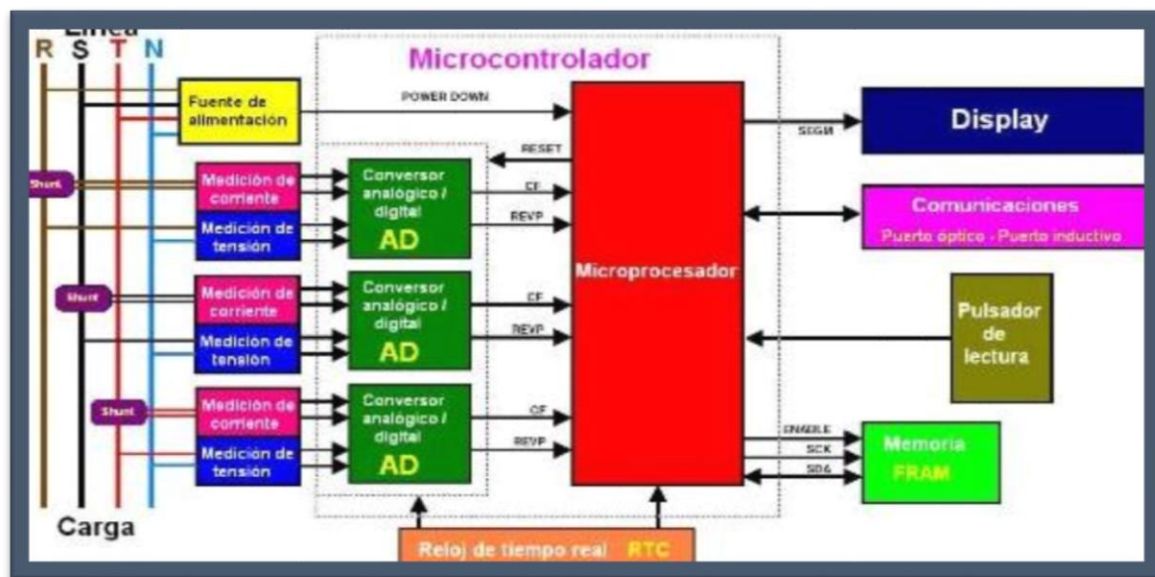


**Fuente:** Redes eléctricas (Ríos, s. f.).

### 2.2.5.2 CONTADORES ELECTRÓNICOS.

Un medidor electrónico posee un procesador que se encarga de convertir las señales analógicas de corriente y tensión a una señal digital; este mismo se encarga de calcular la potencia consumida por el cliente.

Figura 2.3: Diagrama de funcionamiento de medidor electrónico Trifásico



Fuente: disección de un medidor trifásico electrónico (Leandro, 2007).

### 2.2.6 WINDOWS PRESENTATION FOUNDATION.

Windows Presentation Foundation (WPF) es una tecnología de Microsoft que permite el desarrollo de interfaces en Windows, este facilita al desarrollador con la creación de animaciones e interfaces con la que el usuario final tendrá contacto.

#### 2.2.6.1 C SHARP (C#).

C# es un lenguaje de programación multiplataforma de Microsoft, este lenguaje de programación deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET.

Aunque C# forma parte de la plataforma .NET, esta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma.

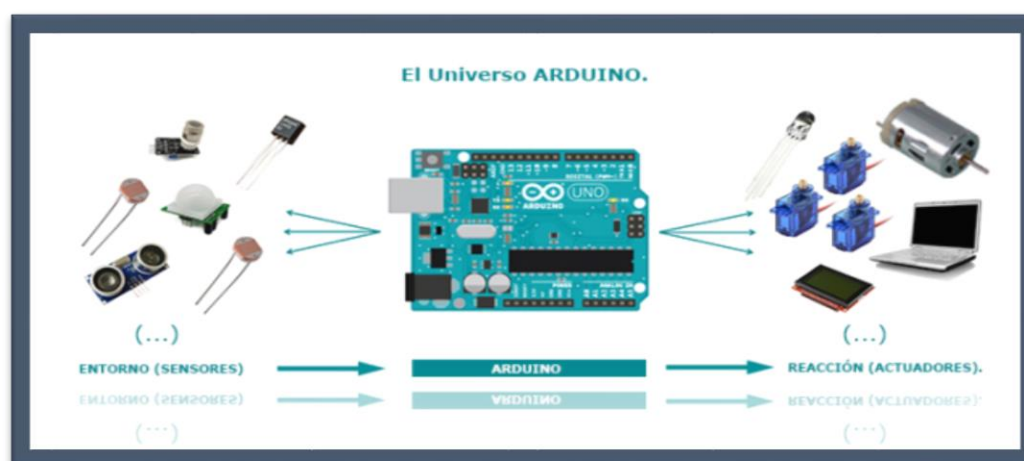
### 2.2.7 CONCEPTOS BASICOS DE ARDUINO.

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo (software), diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware de Arduino consiste en una placa con un microcontrolador generalmente Atmel AVR con puertos de comunicación y puertos de entrada/salida. Los microcontroladores más usados en las plataformas Arduino son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez, pero se está ampliando a microcontroladores Atmel con arquitectura ARM de 32 bits y también a microcontroladores de Intel (Aprendiendoarduino.wordpress.com, 2016).

Arduino tiene un software de entorno de desarrollo IDE que permite al desarrollador programar y cargar su código en desarrollo una de las características de este, es la sencillez y facilidad de uso.

Figura 2.4: El universo Arduino

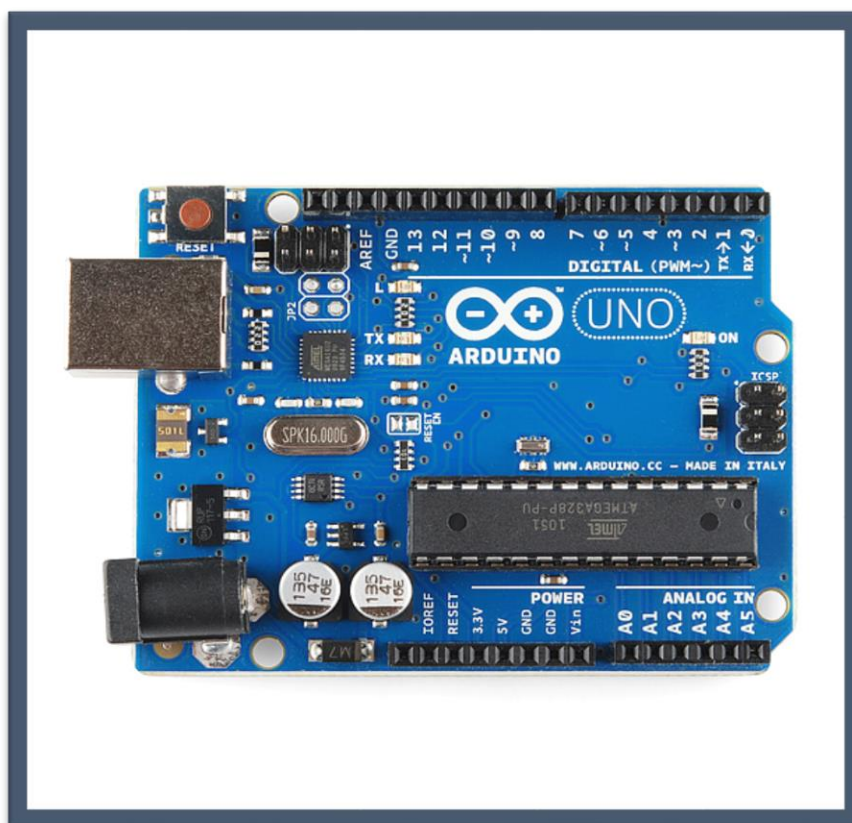


Fuente: (Aprendiendoarduino.wordpress.com, 2016).

### 2.2.7.1 TIPOS DE PLACA ARDUINO

- **Arduino UNO:** Es la primera placa que salió al mercado, siendo esta plataforma la más conocida; este se basa en un microcontrolador Atmel ATmega328 de 8 bits a 16 Mhz que funciona con un voltaje de 5v. La memoria flash que posee es de 32kb; 2KB de SRAM y 1KB de EEPROM. Contiene 14 pines digitales, 6 de ellos se pueden emplear como PWM. En cuanto a pines analógicos se cuenta con hasta 6. Estos pines pueden trabajar con intensidades de corriente de hasta 40mA.

Figura 2.5: Arduino Uno

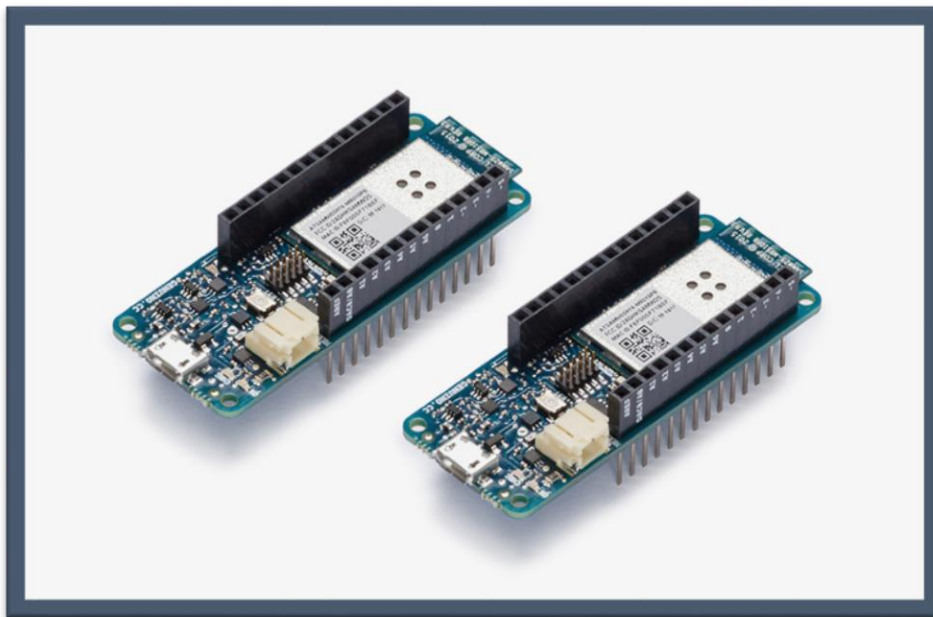


Fuente: (Oscar, 2017).

- **ARDUINO MKR 1000:** Fue diseñado con el propósito de añadir conexión WIFI a Arduino. Basado en ATSAMW25 SoC, cuenta con un microcontrolador de 32 bits ARM MCU SAMD21 Cortex-M0, se alimenta

usb/vin 5V, 48 MHz, 8 E/S digitales, 12 PWM, 8 analógicos, Memoria flash 256 KB, y SRAM de 32 KB.

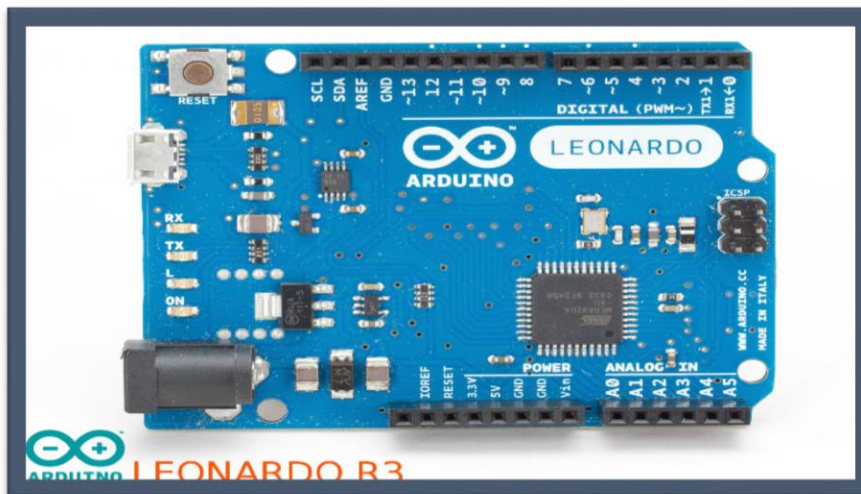
**Figura 2.6: Arduino MKR 1000**



**Fuente:** (Oscar, 2017).

- **ARDUINO LEONARDO:** Esta placa tiene un microcontrolador ATmega32U4 que trabaja a 16Mhz La memoria flash es de 32KB (4KB para el bootloader) y 2.5KB de SRAM. La EEPROM es de 1KB. Este es un modelo muy parecido al Arduino UNO sin embargo la diferencia entre estos es su salida de pines este cuenta con 20 pines digitales (7 como PWM) y 12 analógicos.

Figura 2.7: Arduino Leonardo



Fuente: (Oscar, 2017).

- **ARDUINO MEGA:** Esta placa de Arduino también trabaja a 16 Mhz y un voltaje de 5v; lleva un microcontrolador ATmega2560. Este microcontrolador de 8 bits cuenta con una SRAM de 8KB, 4KB de EEPROM y 256KB de flash. el número de pines es de 54 pines digitales (15 de ellos como PWM) y 16 pines analógicos.



Figura 2.8: Arduino Mega 2560



Fuente: (Oscar, 2017).

- **ARDUINO DUE:** Esta placa es muy parecida al Arduino Mega, sin embargo, la diferencia es que Arduino Due tiene un microcontrolador de 32 bits Atmel SAM3X8E ARM Cortex-M3, microcontrolador que trabaja a 3,3v con una frecuencia de 84 Mz. Esta placa dispone de 54 pines de E/S digitales, 12 como PWM y 12 analógicos, 4 UARTs y conexión USB OTG, dos conexiones DAC (conversión digital a analógico), 2 TWI, un power jack, SPI y JTAG.

Figura 2.9: Arduino Due



Fuente: (Oscar, 2017).

### 2.2.8. CONCEPTO GENERAL DE MULTÍMETRO.

Los multímetros digitales al paso del tiempo han ido mejorando en cuanto a versatilidad y precisión; estos han sido el remplazo de los anteriores multímetros analógicos.

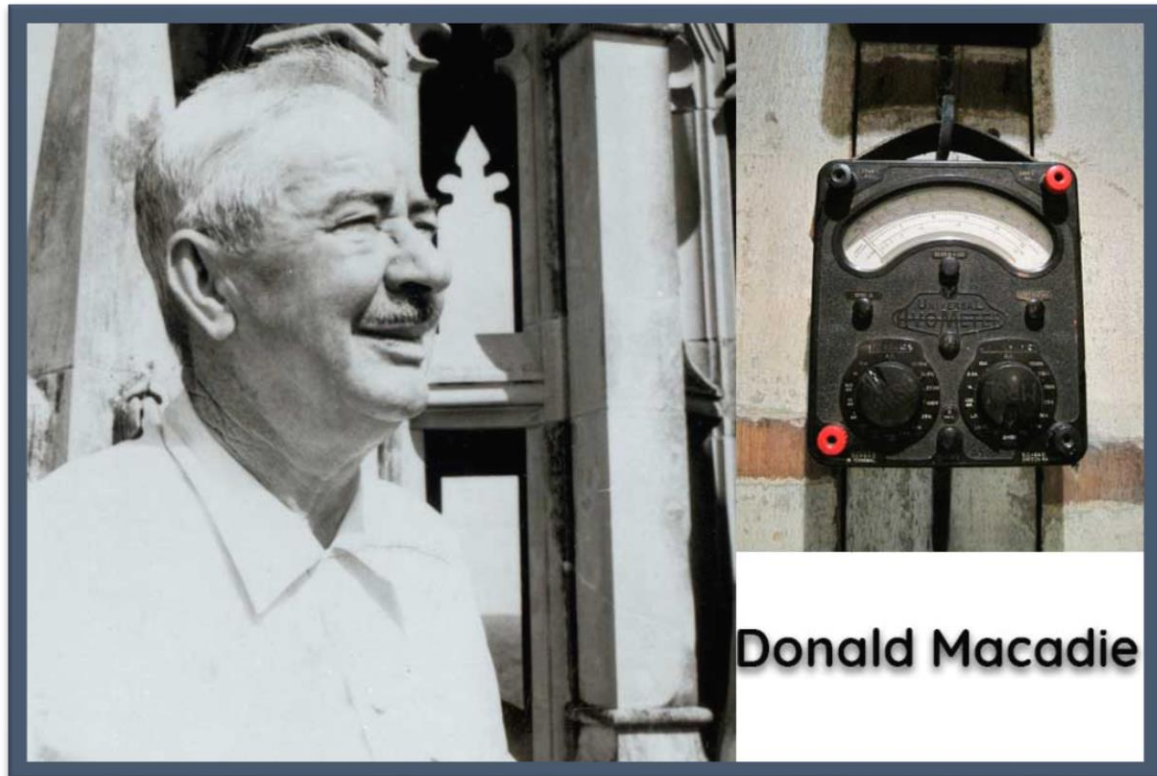
Esta herramienta de medición es usada para medir diferentes señales, en las más utilizadas tensiones (voltios), corriente (amperios) y resistencia (ohmios), el multímetro se ha estandarizado al paso del tiempo siendo usado en la industria e investigación electrónica y eléctrica.

El multímetro fue inventado por Donald Macadie, un ingeniero eléctrico que trabajaba en british post office. antes de adoptar el nombre de multímetro, se conocía como AVO (Amperios, voltios, ohmios). Este invento supuso una revolución en la época dentro de la industria y facilitó la actividad de muchos profesionales del sector.



El multímetro empezó a comercializarse en el año 1993 a través de Automatic Coil Winder and Eléctrica Equipment Company y en un inicio solo se podía medir corriente continua (Fluke, 2019a).

**Figura 2.10: Donald Macadie el primer multímetro**



**Fuente:** (Multimetrodigital.net, 2019).

Los componentes de un multímetro digital que interactúan con el usuario pueden incluir cuatro componentes:

- Pantalla: donde se observan las lecturas de medición.
- Botones: para seleccionar varias funciones; las opciones varían según el modelo.
- Selector (o conmutador giratorio): para seleccionar los valores de medición primarios (voltios, amperios, ohmios).
- Conectores de entrada: donde se insertan los cables de prueba.

Figura 2.11: Multimetros



Fuente: (Fluke, 2019a).

### 2.2.8.1 PINZA AMPERIMÉTRICA.

Es una herramienta eléctrica que combina un multímetro digital básico con un sensor de corriente.

Las pinzas amperimétricas tienen una tenaza batiente integrada en un instrumento eléctrico permite a los técnicos colocar las tenazas de la pinza alrededor de un alambre o cable, y el otro conductor en cualquier punto de un sistema eléctrico para medir la corriente en dicho circuito sin desconectarlo/desactivarlo (Fluke, 2019b).

Debajo de las molduras de plástico, la mordaza está hecha de hierro de ferrita y está diseñada para detectar, concentrar y medir el campo magnético que genera la corriente cuando fluye a través de un conductor (Fluke, 2019b).

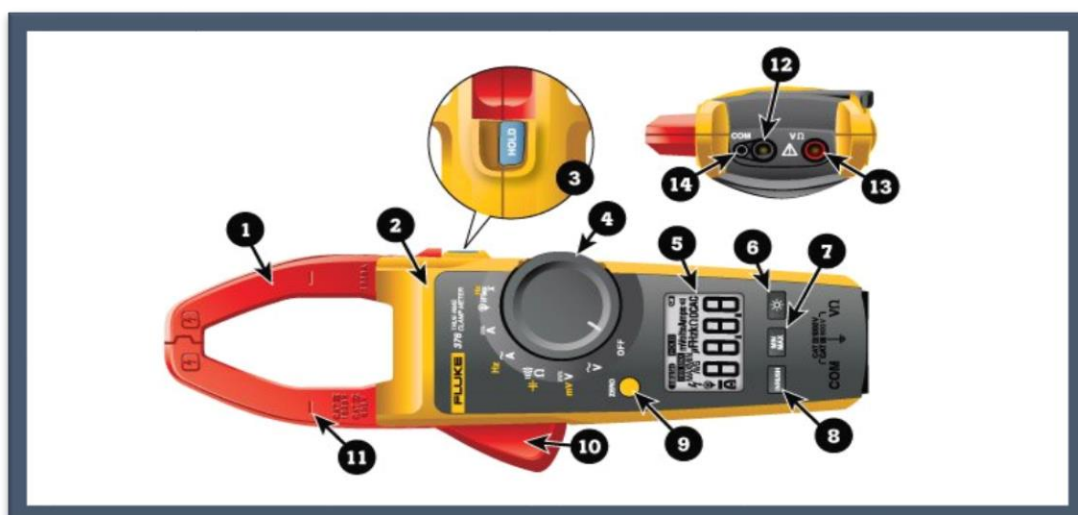
Las pinzas amperimétricas se han convertido en herramientas populares principalmente por dos razones:

- **Seguridad.** Las pinzas amperimétricas permiten a los electricistas omitir el antiguo método de cortar un cable e insertar un medidor de prueba de cables en el circuito para tomar una medición de corriente. Las mordazas de una pinza amperimétrica no necesitan tocar el conductor durante una medición.
- **Comodidad.** Durante la medición, no es necesario desconectar el circuito que transmite corriente: un gran avance en la eficiencia.

Hay tres tipos de pinzas amperimétricas:

- **Pinzas amperimétricas con transformador de corriente:** miden solo corriente alterna (CA).
- **Pinzas amperimétricas de efecto Hall:** miden tanto corriente alterna como corriente continua (CA y CC).
- **Pinzas amperimétricas flexibles:** usan una bobina Rogowski; solo miden la CA; son buenas para mediciones en espacios estrechos.

**Figura 2.12: Partes de una Pinza Amperimétrica.**



**Fuente:** (Fluke, 2019b).

1. Mordaza con detección de corriente.
2. Barreras táctiles (para proteger los dedos de descargas).
3. Botón de retención: Congela la lectura de la pantalla. La lectura es liberada cuando se presiona el botón una segunda vez.
4. Selector (también conocido como interruptor giratorio).
5. Pantalla.
6. Botón de retroiluminación.
7. Botón Mín.-Máx.: La primera vez que se presiona, la pantalla muestra la entrada máxima. Cuando se presiona varias veces, se muestran las entradas mínimas y promedio. Funciona en modos de corriente, tensión y frecuencia.
8. Botón de corriente de entrada.
9. Botón de cero (amarillo): Elimina la compensación de CC de las mediciones de corriente CC. También sirve como botón de funciones secundarias para seleccionarlas funciones en amarillo distribuidas en el selector.
10. Palanca de liberación de la mordaza.
11. Marcas de alineación: Para cumplir con las especificaciones de precisión, un conductor debe estar alineado con estas marcas.
12. Pin de entrada común.
13. Pin de entrada en voltios/ohmios.
14. Sonda para entrada de corriente flexible.

#### **2.2.8.1.2 EFECTO HALL.**

##### **✓ ANTECEDENTE HISTORICO.**

En octubre de 1879, el físico Edwin Helbert Hall descubrió el efecto que lleva su nombre. Encontró que si se aplica un campo magnético elevado a una fina lámina de oro por la que circula corriente, se produce un voltaje en

la lámina transversalmente a como fluye la corriente, este voltaje se llama voltaje Hall.

Si un conductor se encuentra en el seno por donde circula corriente eléctrica aparece una fuerza magnética en los portadores de carga o portadores de carga se desvían y agrupan a un lado del material conductor o semiconductor, apareciendo así un campo eléctrico («Efecto Hall - EcuRed», 2019).

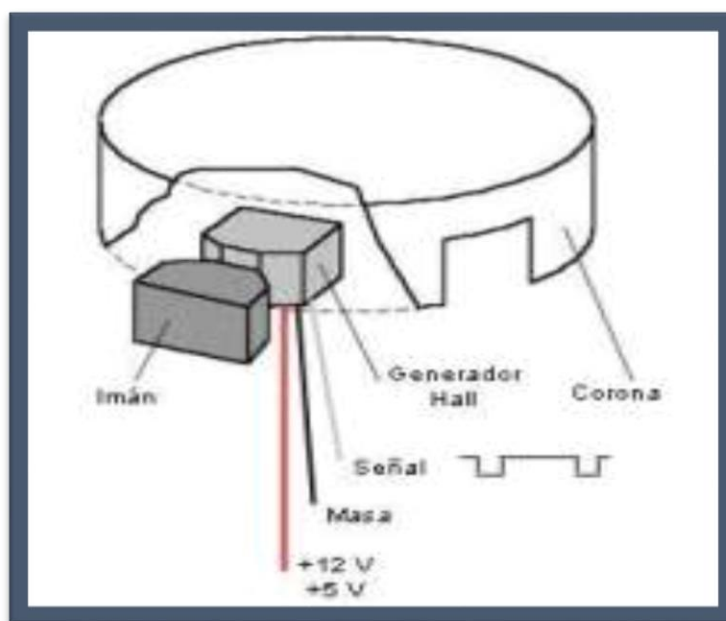
#### ✓ **APLICACIÓN DE EFECTO HALL.**

Un sensor de efecto Hall te permite medir los siguientes aspectos:

- La movilidad de una partícula cargada eléctricamente (electrones, lagunas, etc).
- Los campos magnéticos (Tslámetros)
- La intensidad de corrientes eléctricas (sensores de corriente de Efecto Hall)
- También permiten la elaboración de sensores o detectores de posición sin contacto, utilizados particularmente en el automóvil, para detectar la posición de un árbol giratorio (caja de cambios, paliers, etc.).
- Se encuentran también sensores de efecto Hall bajo las teclas de los teclados de los instrumentos de música modernos (órganos, órganos digitales, sintetizadores) evitando así el desgaste que sufren los contactos eléctricos tradicionales.
- Sensores de efecto Hall en el codificador de un motor de CD.

- Los motores de Efecto Hall (HET) son aceleradores de plasma de gran eficacia.

**Figura 2.13: Sensor Hall.**



**Fuente:** («Efecto Hall - EcuRed», 2019).

### 2.2.8.2. SENSOR DE CORRIENTE NO INVASIVO AC SCT-13.

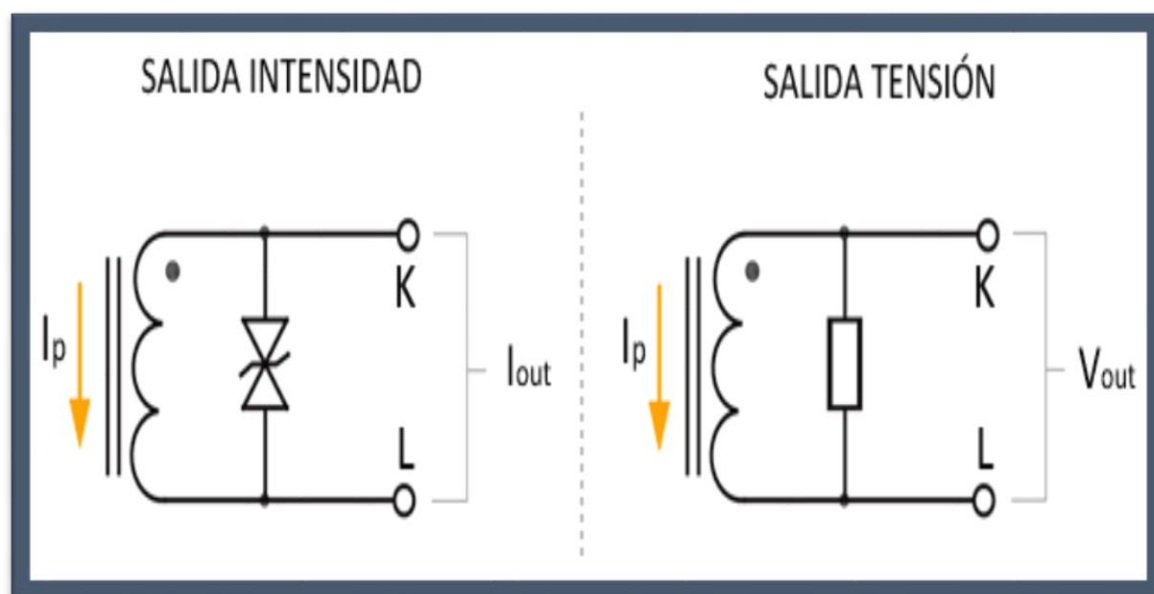
Los transformadores de intensidad son componentes frecuentes en el mundo de la distribución eléctrica que nos permite monitorizar el consumo en lugares y puntos donde sería imposible otro tipo de medición. También forman parte de múltiples instrumentos de medición, incluso en equipos portátiles como en pinzas perimétricas ya mencionadas anteriormente (Luis, 2017).

El sensor de corriente no invasivo de corriente alterna de modelo “SCT-13”, es un transformador de corriente que tiene un devanado primario un núcleo magnético un devanado secundario; en este transformador de corriente el devano primario es representado por el cable o conductor que desea medir actuando este como devanado primario; el devanado secundario puede tener hasta 2000 espiras, esta represente la

relación entre corriente que circula por el cable y la que el sensor nos entrega esta relación o proporción es la que diferencia entre los diferentes modelos de sensores SCT-013, adicionalmente pueden tener una resistencia de carga en la salida de esta forma en lugar de corriente se trabaja con una salida voltaje («Tutorial sensor de corriente AC no invasivo SCT-013», 2016).

existen modelos que proporcionan la medición como una salida de intensidad o de tensión.

**Figura 2.14: Tipos de salida sensor AC SCT-13**



**Fuente:** (Luis, 2017).

Como ya se había mencionado este transformador de corriente busca generar una intensidad en el secundario que sea proporcional a la intensidad que atraviesa el primario. Para ello se desea que el primario esté formado un número de espiras reducido.



Podemos emplear esto para construir un sensor de corriente no invasivo. En un sensor se corriente el núcleo ferromagnético puede estar dividido de forma que pueda abrirse y arrollar un conductor (Luis, 2017).

De esta forma, se tenemos un transformador en el que:

- El cable por el que circula la intensidad a medir constituye un devanado primario
- La "pinza" es el núcleo magnético
- El devanado secundario está integrado como parte de la sonda.

**Cuando la corriente alterna circula por el conductor se genera un flujo magnético en el núcleo ferromagnético, que a su vez genera una corriente eléctrica en el devanado secundario.**

La relación de transformación de intensidad depende de la relación entre el número de espiras.

$$\frac{I_s}{I_p} = \frac{v_p}{v_s} = \frac{N_p}{N_s}$$

#### Ec. N° 1 relación de transformación de intensidad

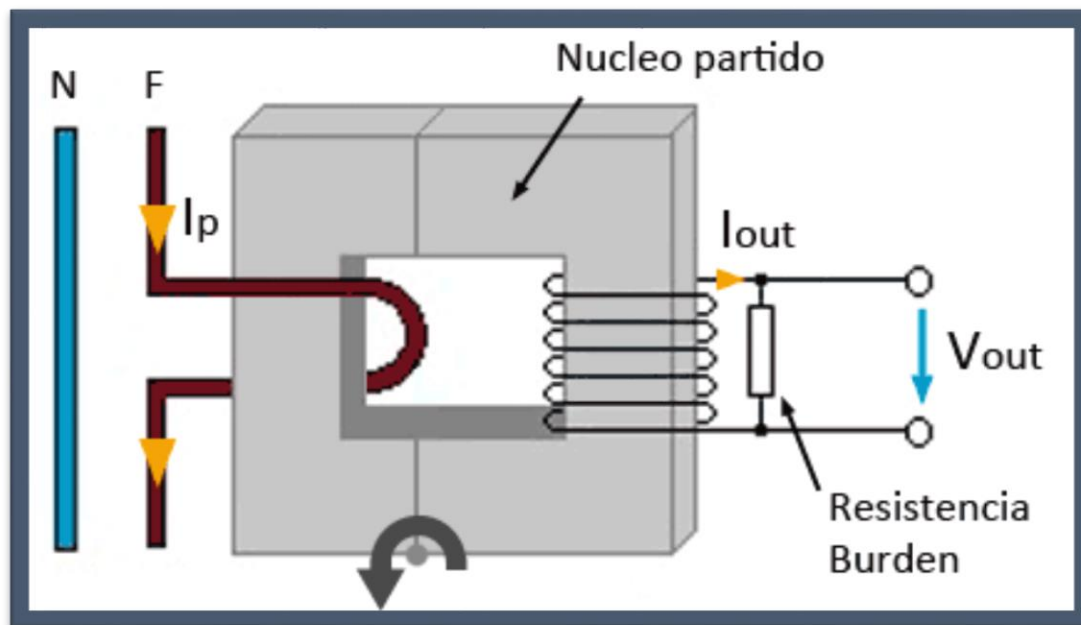
El primario generalmente está formado por una única espira formada por el conductor a medir. Aunque es posible enrollar el conductor haciendo que pase más de una vez por el interior de la "pinza"

A diferencia de los transformadores de tensión, en un transformador de intensidad el circuito secundario nunca debería estar abierto, porque las corrientes inducidas podrían llegar a dañar el componente. Por ese motivo, los sensores de SCT-13 disponen de



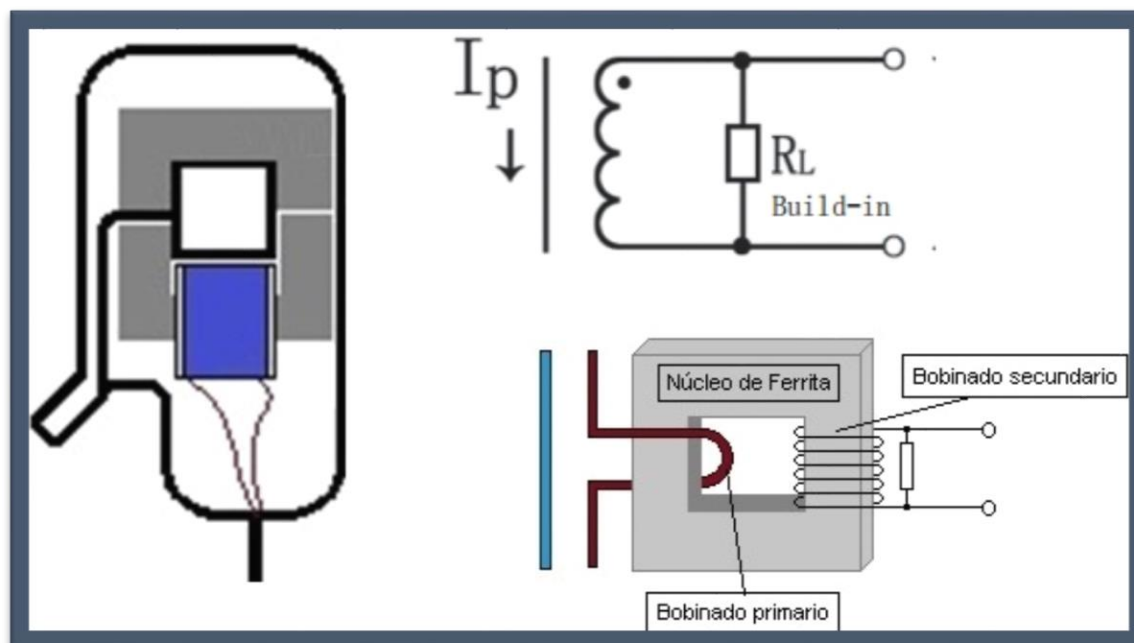
protecciones (resistencia “burden” en los sensores de salida por tensión, o diodos de protección en los sensores de salida por corriente).

Figura 2.15: conexionado transformador de corriente SCT-13.



Fuente: (Luis, 2017).

Figura 2.16: Esquema sensor SCT-13.

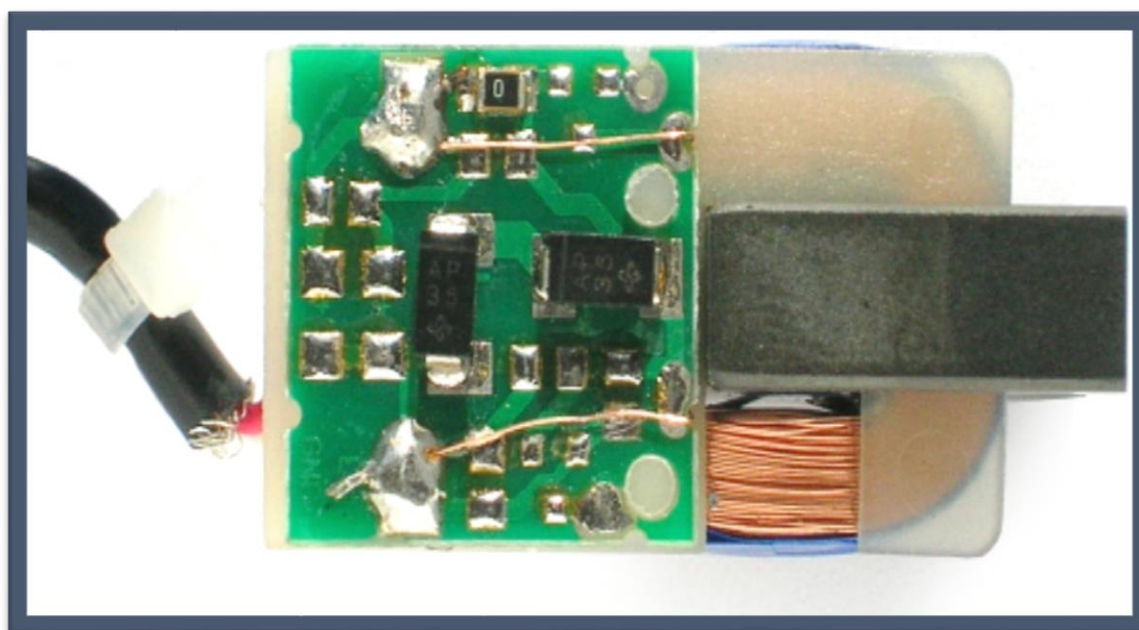


Fuente: («Tutorial sensor de corriente AC no invasivo SCT-013», 2016).

**2.2.8.2.1 MODELOS DE SENSORES SCT-013.**

Existen diferentes versiones y modelos del sensor de corriente SCT-13 los cuales podemos identificar y clasificar de acuerdo a la capacidad de entrada de intensidad en el devanado primario (cable) y la intensidad o voltaje de salida que será proporcional a la de entrada.

**Figura 2.17: componentes internos de sensor SCT-013.**



Fuente: (Robert, 2014).

**Tabla N° 2.2: Modelos de sensores de corriente SCT-013.**

MODELO	SCT-013-000	SCT-013-005	SCT-13-010	SCT-013-015	SCT-013-020
<b>Corriente de entrada</b>	0-100A	0-5A	0-10A	0-15A	0-20A
<b>Tipo de salida</b>	0-50mA	0-1V	0-1V	0-1V	0-1V

Elaboración Propia.

Tabla N° 2.3: Modelos de sensores de corriente SCT-013.

MODELO	SCT-013-025	SCT-013-030	SCT-013-050	SCT-013-060	SCT-013-000V
Corriente de entrada	0-25A	0-30A	0-50A	0-60A	0-100A
Tipo de salida	0-1V	0-1V	0-1V	0-1V	0-1V

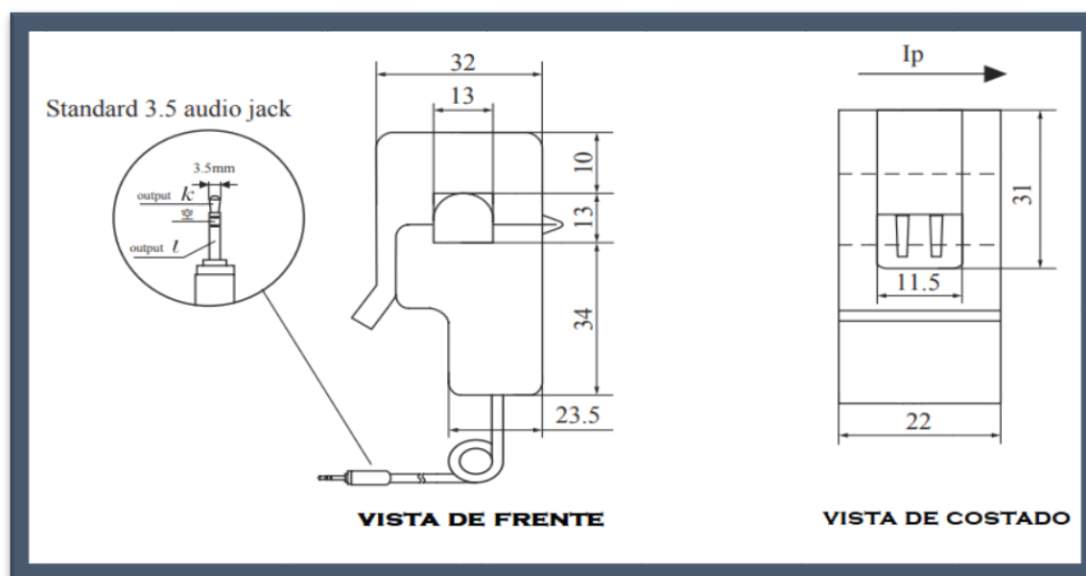
Elaboración Propia.

❖ **MODELO SCT-013-030 Y CST-013-000.**

**Características:**

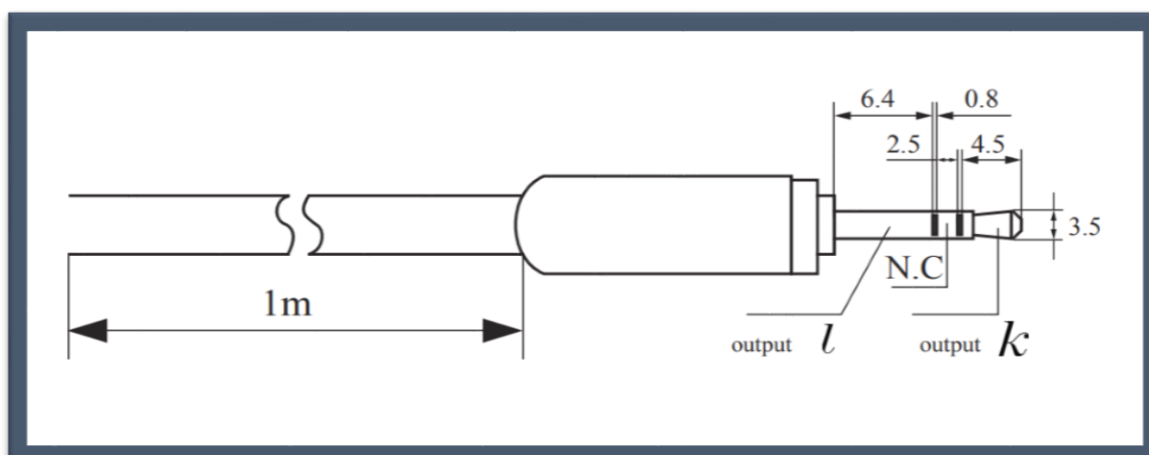
- Tamaño de apertura 13mm x 13mm (Para ambos modelos).
- Cable principal de 1m (Para ambos modelos).
- Plug de salida de  $\Phi$  3.5mm de 3 núcleos (Para ambos modelos).
- Tipo de salida Voltaje (para modelo de 30<sup>a</sup>).
- Tipo de salida Amperios (Para modelos de 100<sup>a</sup>).

Figura 2.18: Características SCT-013



Fuente: (PowerUc, 2019).

Figura 2.19: Conector de 3 núcleos



Fuente: (PowerUc, 2019).

❖ **INDICADORES TÉCNICOS.**

- Propiedad de resistencia al fuego: UL94-V0(Para ambos modelos).
- Temperatura de trabajo:  $-25^{\circ}\text{C} \sim +70^{\circ}\text{C}$  (Para ambos modelos).
- Temperatura de almacenamiento:  $-30^{\circ}\text{C} \sim +90^{\circ}\text{C}$  (Para ambos modelos).
- Voltaje máximo de trabajo: 600V (Para modelo de 30A).
- Voltaje máximo de trabajo 660V (Para modelo de 100A).
- Rango de frecuencia: 50HZ – 1KHz (Para ambos modelos).
- Resistencia dieléctrica: 3.5KV 50Hz 1min (Para ambos modelos).

❖ **PARÁMETROS ELECTRICOS.**

Tabla N° 2.4: Parámetros de sensor SCT-013-030.

<b>ENTRADA NOMINAL</b>	30A
<b>ENTRADA MÁXIMA</b>	60A
<b>SALIDA NOMINAL</b>	1V
<b>PRECISIÓN</b>	$\pm 1 \%$

**LINEALIDAD**  $\leq 0.2 \%$

**PESO** 50G

Elaboración propia.

Tabla N° 2.5: Parámetros de sensor SCT-013-000.

**ENTRADA NOMINAL** 100A

**ENTRADA MÁXIMA**

**SALIDA NOMINAL** 0-50mA

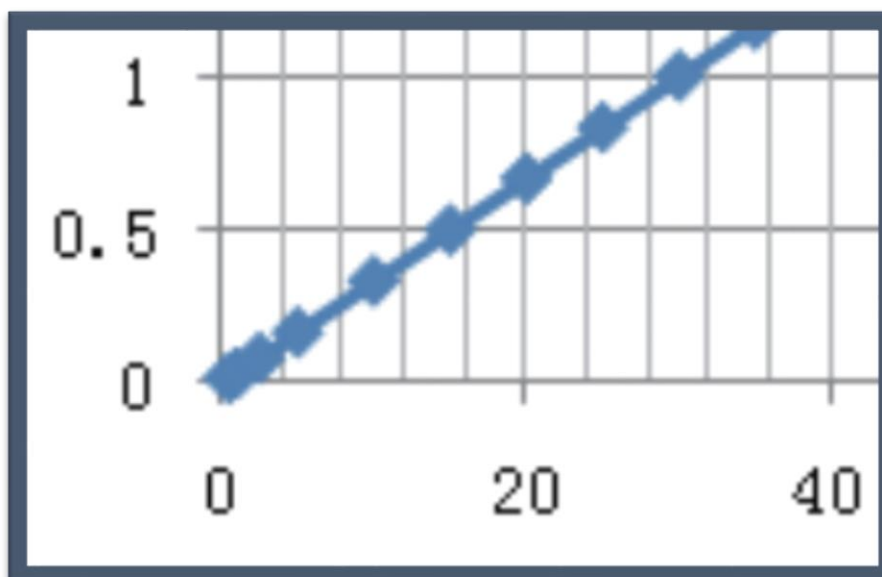
**PRECISIÓN**  $\pm 1 \%$

**LINEALIDAD**  $\pm 0.3 \%$

**PESO** 55G

Elaboración propia.

Figura 2.20: Curva Característica Voltio – Amperios SCT-013



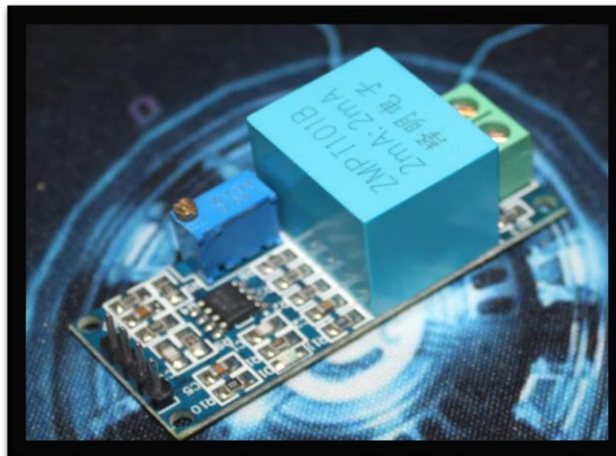
Fuente: (PowerUc, 2019).

### 2.2.8.3 SENSOR DE VOLTAJE ZMPT101B.

Este sensor es un transformador de que permite medir voltaje alterno de hasta 250VAC y entrega una onda senoidal de amplitud regulable por un potenciómetro en la placa. Esta onda esta desplazada positivamente para que no tenga voltajes negativos y así poder leer la onda por completo. Este desplazamiento depende del voltaje con el que alimentemos el módulo: si el voltaje de alimentación es de 5V el desplazamiento será de 2.5V y si alimentamos el módulo con 3.3V el desplazamiento será de 1.65V.

El circuito de acondicionamiento de señal permite que el voltaje de salida del módulo pueda ser leído por cualquier microcontrolador con entrada analógica (ADC), de esta forma es posible leer el voltaje instantáneo y realizar cálculos de energía, como: voltaje pico a pico ( $V_{pp}$ ) y voltaje eficaz ( $V_{rms}$ ) (Naylamp, 2019).

**Figura 2.21: Transformador de corriente ZMPT101B**



**Fuente:** (SurtrTech, 2019).

Tabla N° 2.6: Parámetros de sensor ZMPT101B

ZMPT101B	
<b>Corriente de entrada nominal</b>	2mA
<b>Corriente de salida nominal</b>	2mA
<b>Ratio</b>	1000:1000
<b>Diferencia de fase</b>	≤30 ' (50Ω)
<b>Rango linear</b>	0~3mA (50Ω)
<b>Linealidad</b>	0.1%
<b>Precisión</b>	0.2
<b>Tensión de aislamiento</b>	3KV
<b>Usos</b>	Medición
<b>Material de sellado</b>	epoxy
<b>Instalación</b>	PCB
<b>Temperatura de operación</b>	-40°C~+70°C

Elaboración propia

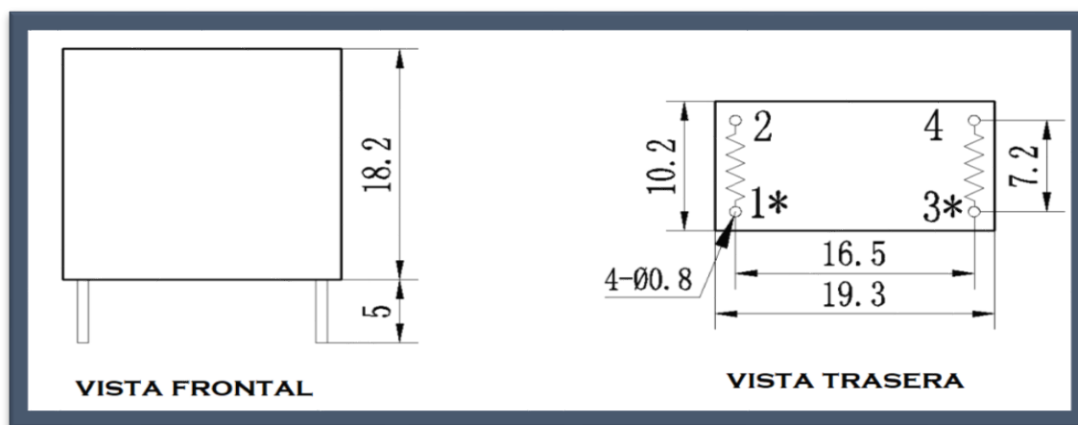
➤ **APLICACIONES.**

- ✓ Aplicable para tensiones de voltaje AC de baja tensión como 220VAC.
- ✓ Para monitoreo y protección de equipos AC.

- ✓ Retroalimentación para sistemas de control de voltaje AC.

➤ **Parámetros estructurales.**

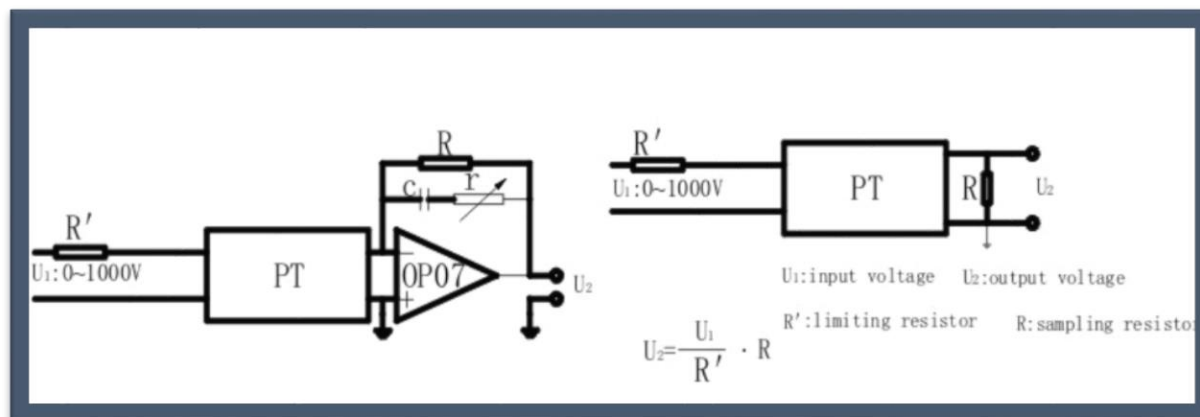
Figura 2.22: Parámetros de ZMPT101B.



Fuente: (ZMPT107 potential transformer, 2019).

➤ **EQUEMA DE USO.**

Figura 2.23: Esquema de uso



Fuente: (Zeming-e, 2013).



### **2.2.9. CONCEPTO BÁSICO DE AMPLIFICADORES OPERACIONALES.**

Un amplificador operacional, es un circuito integrado cuya función principal es amplificar el voltaje con una entrada de tipo diferencial para tener una salida amplificada y con referencia a tierra.

Estos circuitos integrados existen desde 1964 en donde los primeros modelos son el 702, 709 y 741 desarrollados por Fairchild, y 101 y 301 por National Semiconductor. La salida al mercado de los amplificadores operacionales solvento en gran medida la ardua tarea de amplificar señales con transistores.

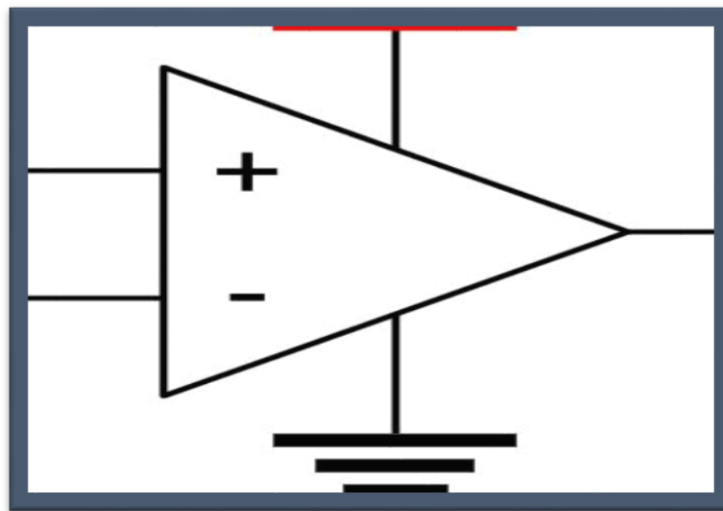
Hoy en día gracias a los amplificadores operacionales la amplificación es más barato, más rápido, en pequeños espacios a diferencia que su contraparte los transistores, usualmente MOSFET. Algunas de estas aplicaciones pueden ser: amplificador de instrumentación, amplificador diferencia, convertidor de corriente voltaje. Con un Amplificador Operacional, Gracias a esta tecnología podemos realizar temporizadores, comparadores o detectores de voltaje, acondicionar señales para ADCs y mucho más (Hector, 2017).

#### ***2.2.9.1. DIAGRAMA DE UN AMPLIFICADOR OPERACIONAL.***

El símbolo de un amplificador operacional incluye al conjunto de 2 entradas y una salida. Este tiene una entrada positiva y una negativa. Por ejemplo, en su forma de comparador, si la entrada positiva supera en voltaje a la entrada negativa, la salida se va a su voltaje de saturación.

El amplificador operacional puede ser alimentado con fuentes diferenciales (voltaje positivo y negativo) o fuentes simples (Voltaje positivo y GND y tiene distintas configuraciones, por ejemplo, inversor, no-inversor, amplificador o sumador, entre otras.

Figura 2.24: Símbolo de Amplificador operacional.

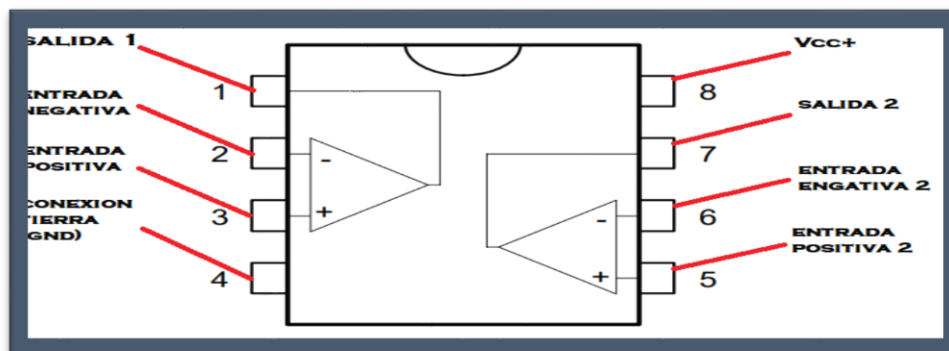


Fuente: (Hector, 2017).

➤ **AMPLIFICADOR OPERACIÓN LM358.**

Este amplificador operacional cuenta con 8 pines y trabaja con dos circuitos independientes que se encuentran dentro del encapsulado que compensan la frecuencia del amplificador operacional y cada uno opera como suplemento de poder que operan a diferente rango de voltaje.

Figura 2.25: Amplificador operacional LM358.



Elaboración propia.

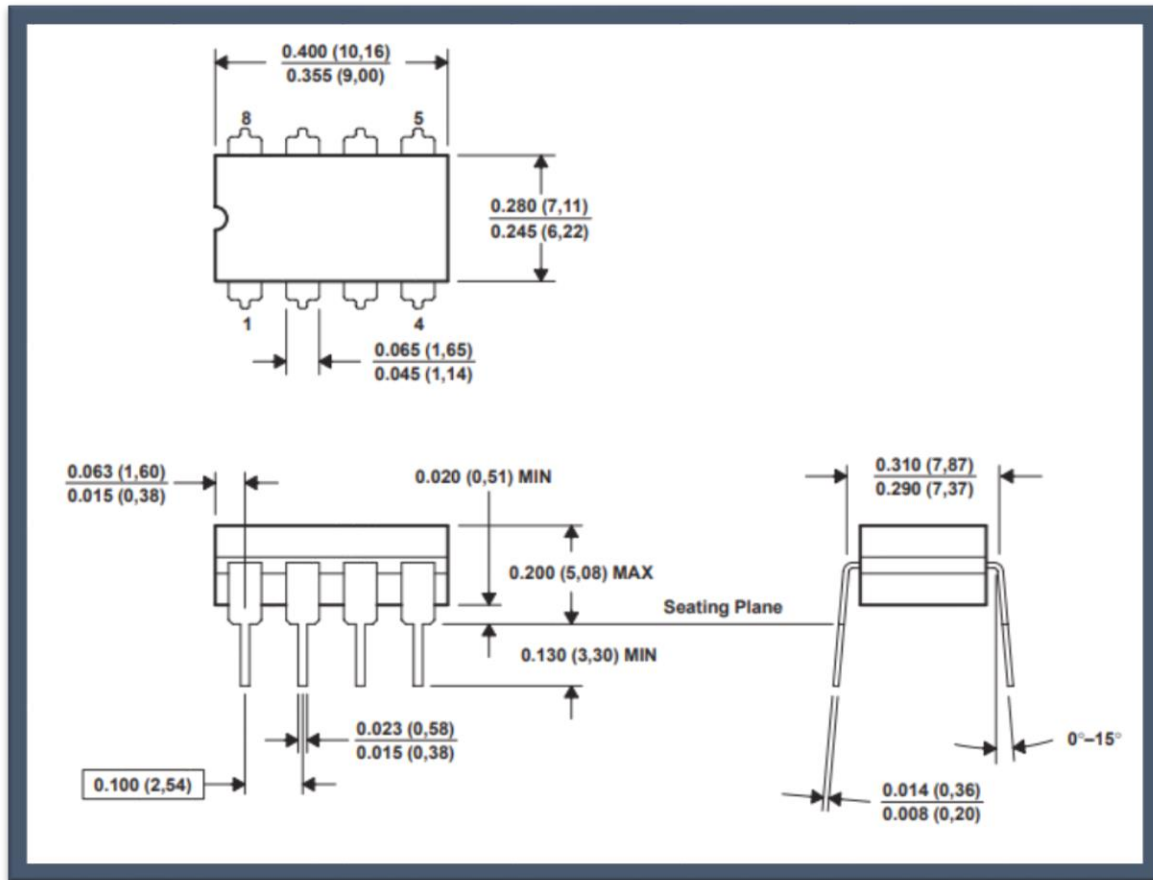
• **APLICACIONES FRECUENTES.**

- ✓ Radio frecuencias.
- ✓ Audio frecuencias.

- ✓ Generación de pulsos.
- ✓ Sensores.
- **CLASIFICACIONES MÁXIMAS ABSOLUTAS.**
  - ✓ Voltaje de aprovisionamiento VCC 16 a 32 V.
  - ✓ Voltaje de entrada 3 a 32 V.
  - ✓ Corto circuito de salida para GND es de VCC +-15V.
  - ✓ Temperatura máxima de operación 0 a 70°C.
  - ✓ Temperatura máxima de almacenamiento de -50 a 150°C.
- **CARACTERÍSTICAS ELÉCTRICAS.**
  - ✓ Compensación de entrada de voltaje: usual 2.9 y máximo 7.0mV.
  - ✓ Ganancia de voltaje: mínimo 25, usual 100 V/mV.
  - ✓ Oscilación de voltaje de salida (RL 10K $\Omega$ ) mínima 27 usual 28V.
- **CARACTERÍSTICAS DE FABRICACIÓN.**

En la figura se puede observar las características de dimensionamiento del amplificador operacional LM358.

Figura 2.26: Características de fabricación.



Fuente: (Texas, 2019).

### 2.3 CIRCUITO IMPRESO.

Un circuito impreso o por sus siglas en ingles PCB (Printed Circuit Board), es una superficie que está constituida por pistas o caminos de un material conductor de electricidad, como el cobre que esta laminada sobre una base no conductora que es en su mayoría resina de fibra de vidrio reforzada.

Un circuito impreso tiene la función de conectar eléctricamente a través de sus pistas o caminos y sostener mecánicamente por medio de la base los componentes electrónicos que contendrá dicho circuito.

### 2.3.1 HISTORIA DEL CIRCUITO IMPRESO.

No se sabe con exactitud cuál fue el primer circuito impreso, sin embargo, es muy probable que Paul Eisler, ingeniero Austriaco, fuera el inventor del circuito impreso mientras trabajaba en una radio en Inglaterra.

Aproximadamente en 1943, en Estados Unidos comenzaron a aplicar esta tecnología para fabricar radios a gran escala, con la finalidad de usarla en la segunda guerra mundial, después de la guerra en 1948, Estados Unidos liberó la invención para su uso comercial («Circuito impreso - Wikipedia», 2019).

### 2.3.2 PROGRAMAS PARA EL DISEÑO DE CIRCUITO IMPRESO.

Entre los programas más comunes tenemos:

- ExpressPCB
- OrCAD(en)
- Proteus Design Suite
- Cadence Allegro
- EasyEDA
- KiCad
- Altium
- Livewire
- PCBWiz
- DesignSpark PCB
- EAGLE
- gEDA

### 2.3.3. FABRICACION DE CIRCUITOS IMPRESOS.

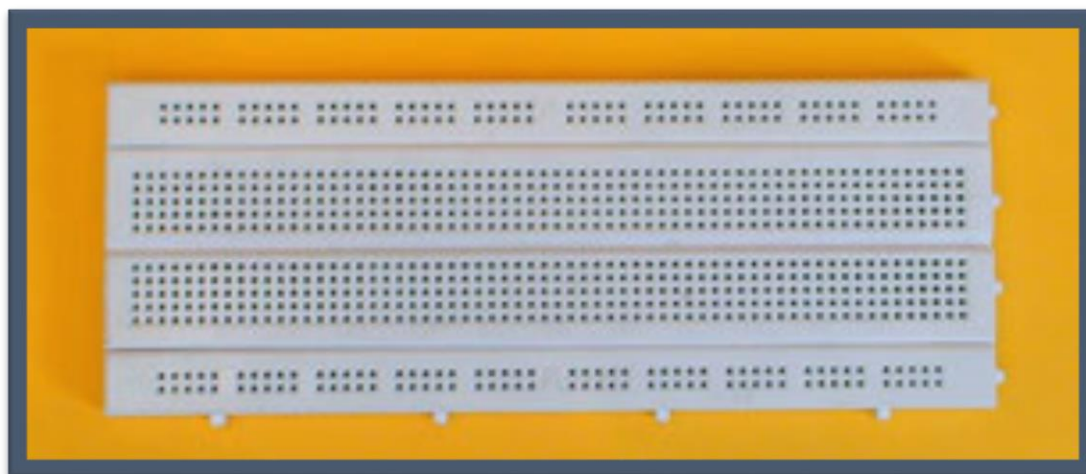
En un proyecto, al diseñar el prototipo electrónico primero se debe probar armándose en una placa de pruebas protoboard. Cuando este funciona correctamente se dibuja un diagrama esquemático, ya sea a mano o con ayuda de un programa de computador. Posteriormente se diseña y fabrica el circuito impreso («Fabricacion de Circuitos Impresos (PCB) | Video Rockola», 2019).

#### 2.3.3.1. PRUEBA DE CIRCUITO EN PROTOBOARD.

Un protoboard o placa de pruebas es una herramienta que permite interconectar los componentes electrónicos, como resistencias, condensadores, etc., sin necesidad de soldarlos en un circuito impreso; gracias a esto podemos realizar muchas pruebas así optimizando el resultado que se quiere lograr con el circuito final.

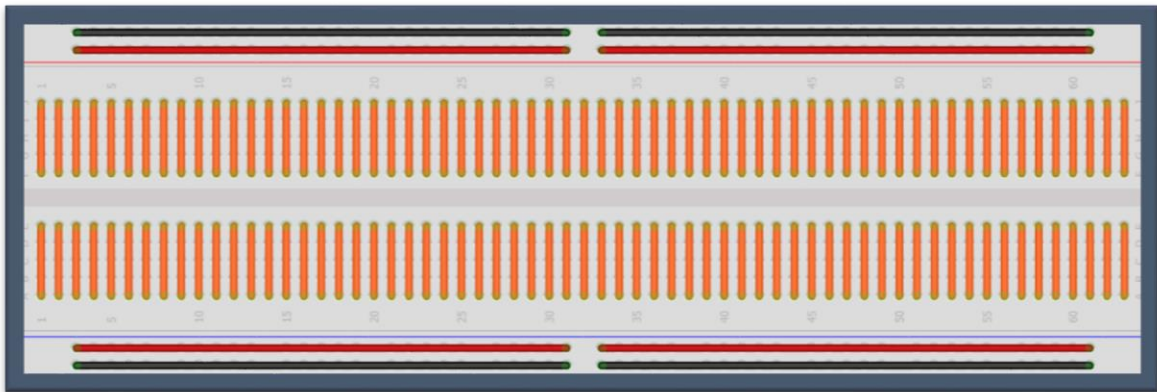
Una placa de pruebas está compuesta por segmentos plásticos que tienen perforaciones y laminas delgadas de una aleación de cobre, estaño y fósforo las cuales pasan por debajo de las perforaciones.

**Figura 2.27: Protoboard**



**Fuente:** («Fabricacion de Circuitos Impresos (PCB) | Video Rockola», 2019).

Figura 2.28: Vista interna de un Protoboard.

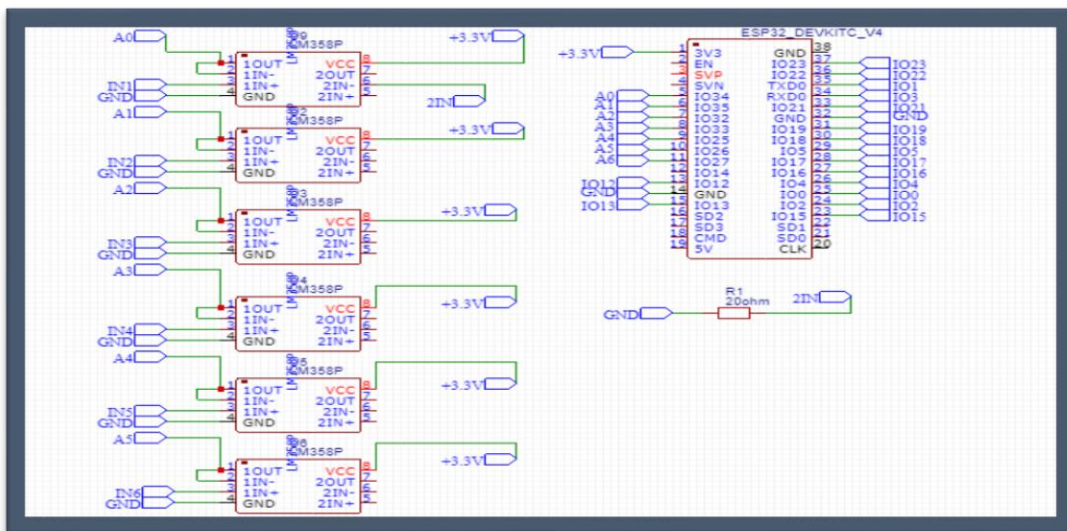


Fuente: («Cómo funciona la protoboard | Atlantis», 2017).

2.3.3.2 DIAGRAMA ESQUEMÁTICO (SCHEMATIC).

Cuando las pruebas en el protoboard han salido exitosamente, se procede a realizar un diagrama esquemático el cual consiste en dibujar el circuito utilizando símbolos electrónicos; este esquemático puede realizarse o dibujarse a mano o con ayuda de un software de computador como EasyEda, Eagle, Proteus, etc.

Figura 2.29: Diagrama Esquemático (schematic).



Elaboración propia

### 2.3.3.3. DISEÑO Y FABRICACIÓN DE CIRCUITOS IMPRESOS.

Gracias al material que usa un circuito impreso, en muchos casos fibra de vidrio con resina de poliéster, la calidad y presentación de un circuito impreso es de mayor calidad y útil debido a que estos llevan un baño de cobre en la superficie de ambas caras, lo que nos permitirá marcar los caminos que interconectarán los diversos componentes ya establecidos en el esquemático.

**Figura 2.30:** Placa de fibra de vidrio para circuitos impresos.



**Fuente:** («Fabricacion de Circuitos Impresos (PCB) | Video Rockola», 2019).

### 2.3.3.4. TÉCNICAS PARA LA FABRICACIÓN DE CIRCUITOS IMPRESOS.

Existen muchos métodos y técnicas para poder fabricar un circuito impreso, dependiendo del presupuesto se pueden elegir entre diferentes opciones, entre las cuales tenemos:

- Elaboración de circuitos impresos con **tinta indeleble**
- Elaboración de circuitos impresos con la técnica de **planchado** (papel termo transferible, impresión láser).
- Elaboración de circuitos impresos con la técnica de **serigrafía**.



#### ❖ TÉCNICA DE ELABORACIÓN CON TINTA INDELEBLE.

Esta técnica o manera de fabricar un circuito impreso es la mas económica de realizar, ya que para ello solo se necesita un marcador o plumón de tinta indeleble.

Este método consiste en dibujar las pistas del circuito con el marcador en la placa bañada de cobre para luego sumergirla en una solución corrosiva, (Cloruro Férrico), esta solución corroe la superficie de cobre no marcada por nuestro plumón creando así las pistas que conectaran los componentes del circuito.

**Figura 2.31: Técnica de fabricación con tinta indeleble**



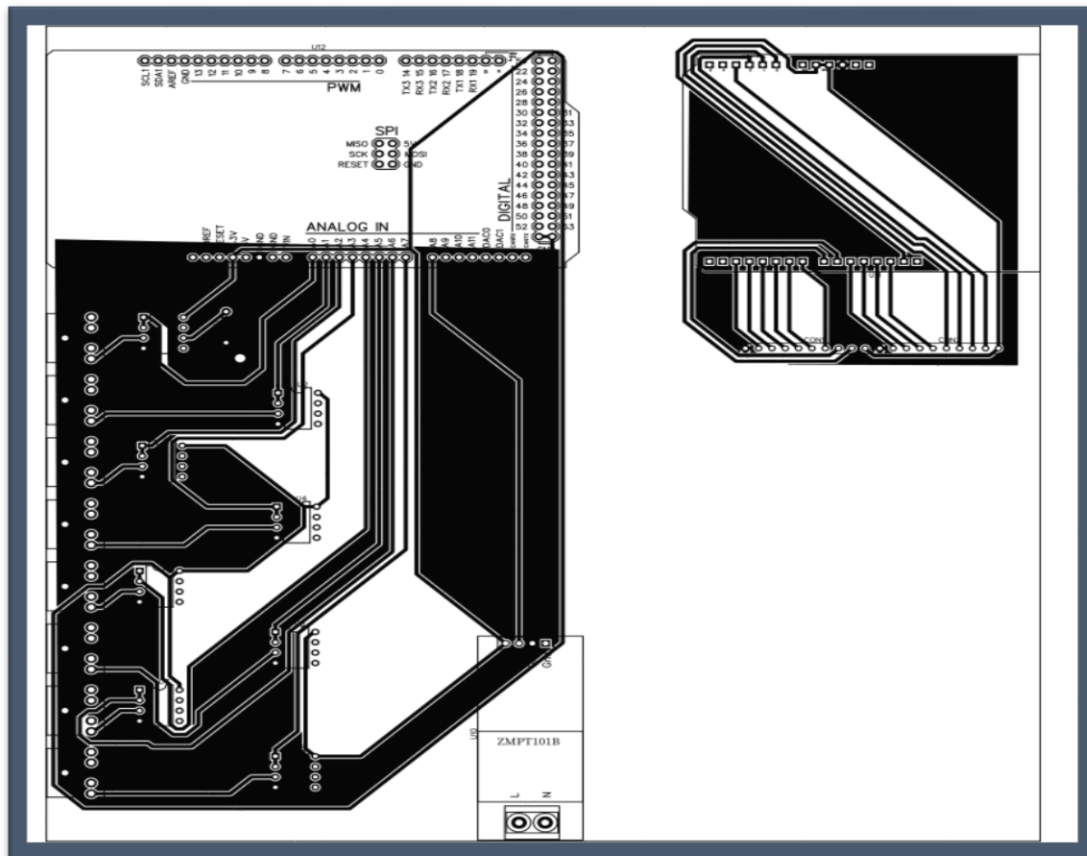
**Fuente:** («Fabricacion de Circuitos Impresos (PCB) | Video Rockola», 2019).

#### ❖ TÉCNICA DE ELABORACION POR EL MÉTODO DEL PLANCHADO.

Este método consiste en imprimir con una impresora a laser o de tóner las pistas del circuito en un papel termotransferible, estas pistas deberán haber sido ya diseñadas en computadora con ayuda de un software especializado; de esta manera el proceso se facilita y a la vez se realiza con mayor precisión y velocidad; este método es económico y de precisión mayor al de la técnica de

elaboración con tinta indeleble; gracias a este podemos realizar circuitos más avanzados.

**Figura 2.32: Diseño de pistas de circuito a imprimir.**



Elaboración propia

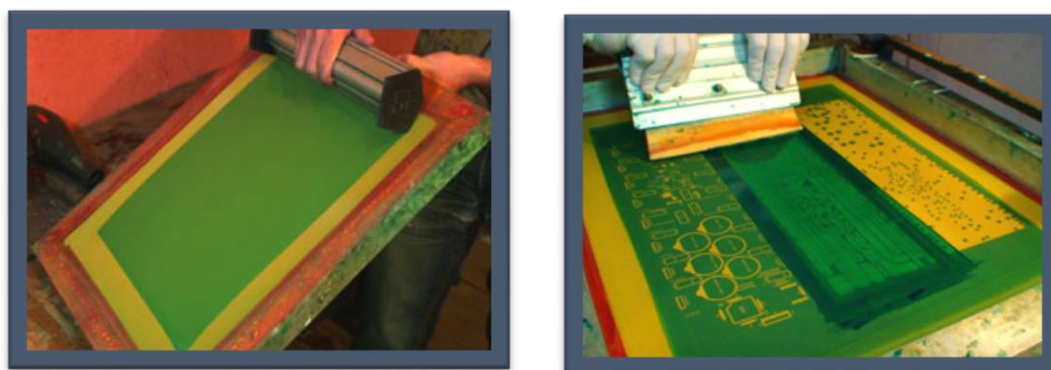
#### ❖ TÉCNICA DE ELABORACION MEDIANTE SERIGRAFÍA.

Esta técnica, permite imprimir imágenes sobre cualquier material, transfiere un tipo de tinta a través de una seda templada por un marco de madera. La seda debe ser tratada con antelación con una emulsión que bloquea el paso de la tinta en las áreas donde no habrá imagen quedando así libre la zona donde pasará la tinta. Esta técnica es usada industrialmente ya que se obtienen impresos de muy buena calidad a un bajo costo.

**MATERIALES NECESARIOS:**

- ✓ Seda No 90 y No 120 con su respectivo marco o bastidor –T300.
- ✓ 90% emulsión nacional universal o atlas verde y revelador 10%.
- ✓ Solvente (activador) o PREGASOL EP 3.
- ✓ Espuma floral (oasis).
- ✓ 1 vidrio de 4 o 5 milímetros con las mismas dimensiones que el interior del marco.
- ✓ Pintura antisoldante (solder mask UV).
- ✓ Pintura UV blanca y verde o tinta para metal o de polietileno screen line.
- ✓ 1 espátula de plástico (Racleta o rasero) (desecho de los acrílicos).
- ✓ Hipoclorito o Varsol para rendir la tinta.
- ✓ Cloruro férrico.
- ✓ Thinner.
- ✓ Resina Colofonia.
- ✓ Soda cáustica.
- ✓ Cloro domestico.
- ✓ Recipiente de plástico adecuado para el baño de las tarjetas.
- ✓ Bombillo de rayos UV o photoflood
- ✓ Trozos de tela y estopa.

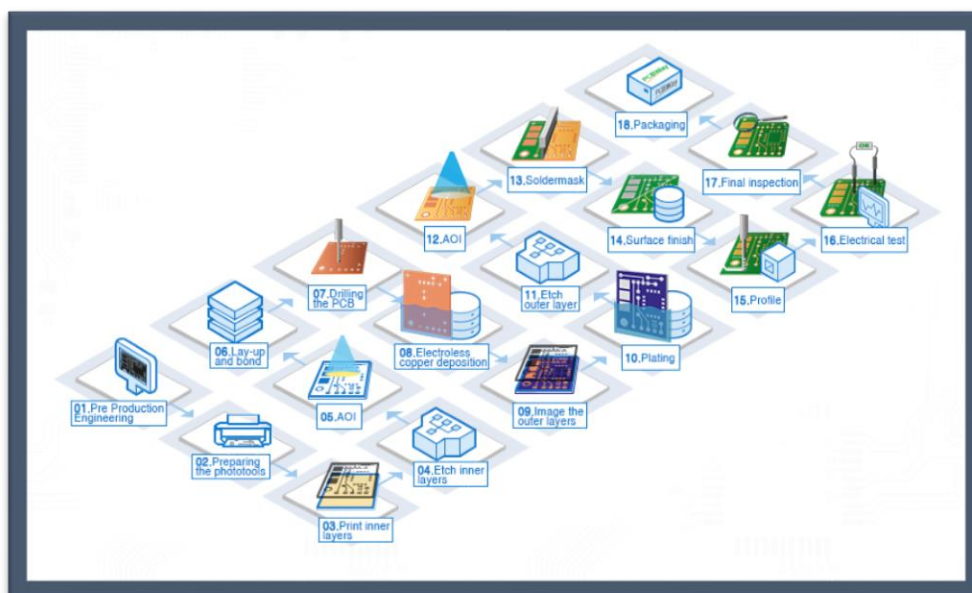
Figura 2.33: Técnica de Serigrafiado



Fuente: («Fabricacion de Circuitos Impresos (PCB) | Video Rockola», 2019).

❖ . **FABRICACIÓN INDUSTRIAL DE CIRCUITOS IMPRESOS.**

Figura 2.34: Esquema de proceso de fabricación de circuitos impresos.



Fuente: (PCBWay, 2019).

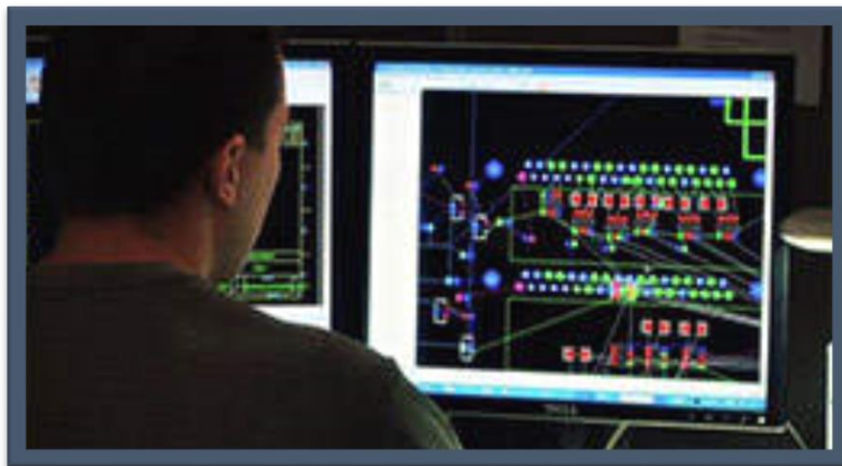
Hoy en día existen diferentes empresas que se encargan de fabricar circuitos impresos a un costo accesible, de esta manera se puede obtener circuitos de mayor calidad, hay varias empresas chinas que envían el producto a diferentes países del mundo, tales como PCBWAY, JLCPCB, PCBGOGO, etc.

El proceso de fabricación industrial de un PCB es el siguiente:

✓ **INGENIERÍA DE PREPRODUCCIÓN.**

El gerber suministrado por el cliente el comparado por los ingenieros de la empresa fabricante con las especificaciones y capacidades de producción, para así asegurar el cumplimiento y también determinar los pasos que seguirá a continuación dicho proceso.

**Figura 2.35: Pre producción de circuito impreso.**



Fuente:(PCBWay, 2019).

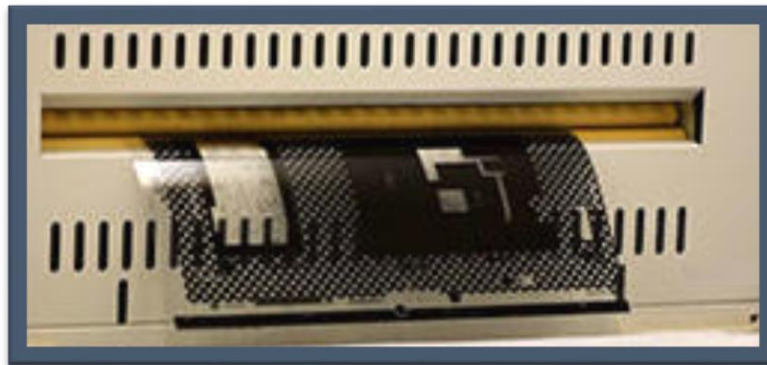
✓ **PREPARACION DE LAS HERRAMIENTAS GRÁFICAS.**

La preparación gráfica es elemental y un paso clave en la producción del circuito impreso, ya que este afecta directamente la calidad del producto final.

Gracias a una configuración de datos electrónicos que esta escalada de forma muy precisa se producen gráficos de alta calidad; el patrón que queda grabado en la pcb generalmente es en escala de 1:1.

Generalmente existen tres tipos de grafico en el PCB; el patrón conductivo, la máscara de soldadura y la serigrafía.

**Figura 2.36: Preparación de herramientas gráficas.**



**Fuente:** (PCBWay, 2019).

✓ **IMPRIMIR LAS CAPAS INTERNAS.**

El primer proceso o etapa consiste en transferir la imagen utilizando una filmina a la superficie de la placa de cobre, utilizando lámina seca fotosensible y luz ultravioleta, que polimerizará la lámina seca expuesta por la filmina; este proceso deberá realizarse en una locación limpia.

Se utiliza luz para transferir la imagen negativa del circuito al panel o filmina.

**Figura 2.37: Imprimir capas internas.**



**Fuente:** (PCBWay, 2019).

✓ **ATACADO DE LAS CAPAS INTERNAS.**



En esta etapa 2 se elimina el cobre no deseado del panel utilizando un atacado, Una vez que el cobre ha sido eliminado, se retira el resto de lámina seca quedando únicamente el circuito de cobre que coincide con el diseño.

Atacado- Es la eliminación química y/o química electrolítica, de porciones no deseadas de material conductor o resistivo.

**Figura 2.38: Atacado de las capas internas.**

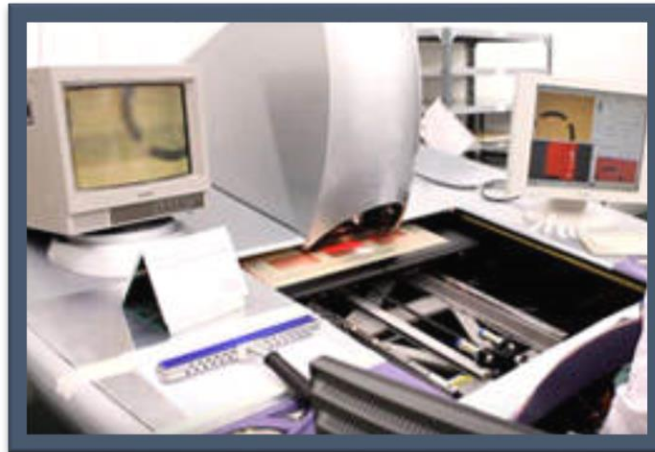


Fuente: (PCBWay, 2019).

✓ **INSPECCIÓN ÓPTICA AUTOMÁTICA DE LAS CAPAS.**

En esta etapa se inspecciona el circuito comparándolos con las imágenes digitales para verificar que coinciden y el diseño está libre de defectos. Se consigue a través de un escaneado de la placa que inspectores entrenados verificarán por si se detecta alguna anomalía.

**Figura 2.39: Inspección óptica automática.**



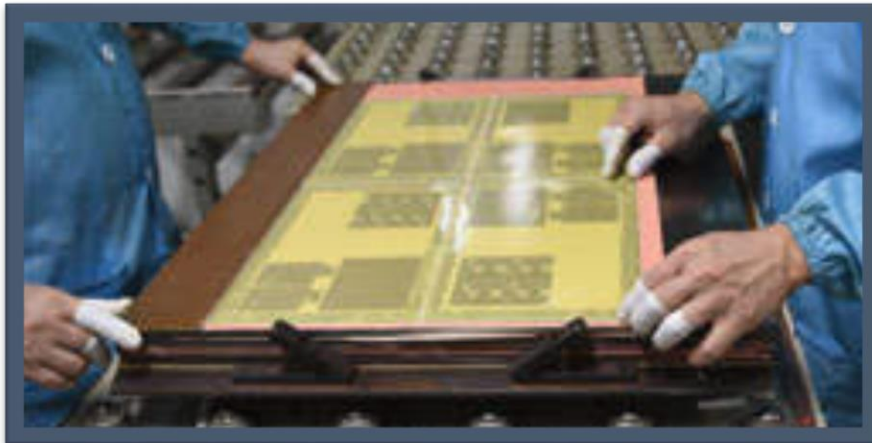
**Fuente:** (PCBWay, 2019).

✓ **APILADO Y PEGADO(LAMINADO).**

En este proceso se aplica una capa de óxido a las capas internas que luego son apiladas conjuntamente con pre-preg que proporciona aislamiento entre las capas y se añade una capa de cobre a las partes superior e inferior del apilado. El proceso de laminado consiste en someter a las capas internas a una temperatura (375 grados Fahrenheit) y presión (de 275 a 400 psi) extremas mientras se laminan con un aislante seco fotosensible. Se somete la PCB a una cura a alta temperatura, se disminuye suavemente la presión y, entonces, se enfría lentamente el material.



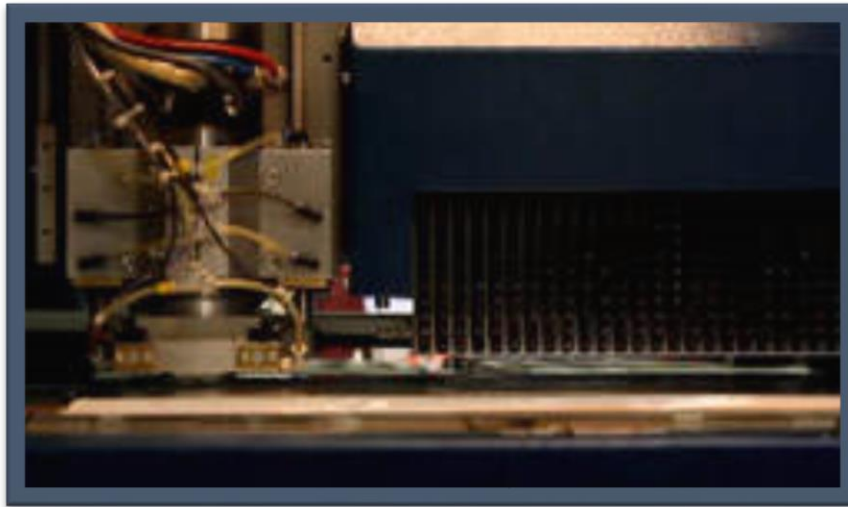
**Figura 2.40: Apilado y laminado.**



**Fuente:** (PCBWay, 2019).

✓ **TALADRADO DEL CIRCUITO IMPRESO.**

Ahora debemos taladrar los orificios que crearán conexiones entre las múltiples capas. Se trata de un proceso de taladrado mecánico que debe ser optimizado para poder conseguir un registro exacto para cada una de las conexiones internas entre capas. Los paneles pueden apilarse durante este proceso. El taladrado también puede realizarse mediante láser.

**Figura 2.41: Taladrado de PCB.**

**Fuente:** (PCBWay, 2019).

✓ **DEPOSICIÓN QUÍMICA DE COBRE.**

El primer paso en el proceso de plateado, es la deposición química de una capa muy delgada de cobre en las paredes del interior de los orificios. El plateado pasante proporciona un depósito muy fino de cobre que cubre las paredes de los orificios y el panel por completo. Un proceso químico complejo que debe estrictamente controlado para permitir un depósito eficaz de cobre incluso en la pared no metálica de los orificios. Siendo aún una cantidad de cobre insuficiente, ya tenemos continuidad eléctrica entre capas y a través de los orificios. Tras el plateado pasante, se procede al plateado del panel para conseguir un depósito de cobre más grueso sobre el anterior – típicamente de 5 a 8  $\mu\text{m}$ . Esta combinación se utiliza para optimizar la cantidad de cobre que ha de ser plateada y atacada con el fin de alcanzar las demandas de pistas y espaciado (PCBWay, 2019).

**Figura 2.42: Deposición química de cobre.**



**Fuente:** (PCBWay, 2019).

✓ **PROCESO GRÁFICO DE CAPAS EXTERIORES.**

De forma similar al proceso de las capas internas (transferencia de la imagen utilizando lámina seca fotosensible, exposición a luz ultravioleta y atacado), pero con una diferencia principal – eliminaremos la lámina seca de los lugares donde deseemos mantener el cobre/definir el circuito – así podremos platear más cobre posteriormente en el proceso. Este paso se realiza en una habitación limpia (PCBWay, 2019).

**Figura 2.43: Proceso gráfico de capas exteriores.**

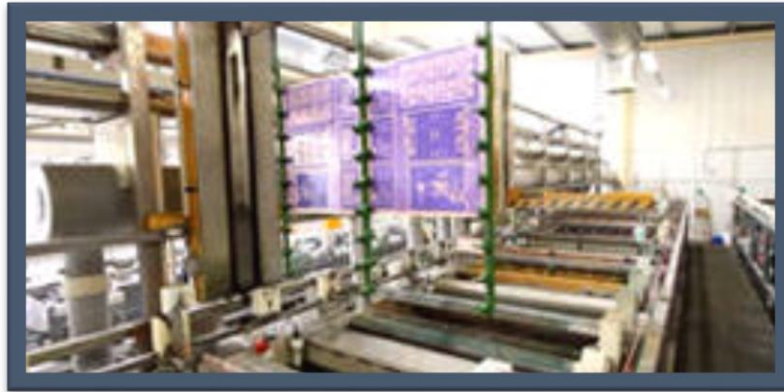


**Fuente:** (PCBWay, 2019).

✓ **PLATEADO.**

En esta etapa de plateado electrolítico, donde un plateado adicional se deposita en las áreas sin lámina seca /circuito). Una vez que se ha plateado el cobre, se aplica estaño para proteger el cobre.

**Figura 2.44: Proceso de plateado.**



**Fuente:** (PCBWay, 2019).

✓ **ATACADO DE CAPAS EXTERNAS.**

Se trata normalmente de un proceso de tres pasos. El primer paso es eliminar la lámina seca azul. El segundo paso es el atacado del cobre expuesto/no deseado mientras el estaño depositado actúa como protector del cobre que deseamos mantener. El tercer y último paso es eliminar químicamente el depósito de estaño dejando el circuito (PCBWay, 2019).

**Figura 2.45: Atacado de capas externas.**

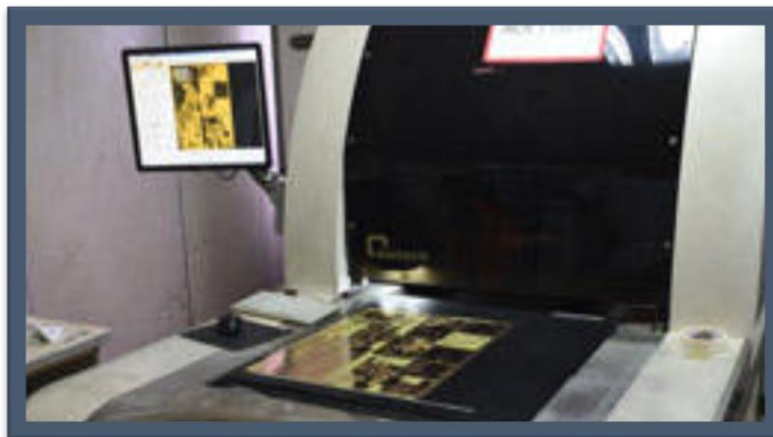


**Fuente:** (PCBWay, 2019).

✓ **INSPECCIÓN ÓPTICA AUTOMATIZADA DE LAS CAPAS EXTERNAS.**

De manera similar a las capas internas, se escanea el panel atacado para asegurar que el circuito cumple con el diseño y está libre de defectos.

**Figura 2.46: Inspección óptica externa.**



**Fuente:** (PCBWay, 2019).

✓ **MÁSCARA DE SOLDADURA.**

La tinta de la máscara de soldadura se aplica sobre toda la superficie de la PCB. Utilizando filminas y luz ultravioleta se exponen ciertas áreas y las no expuestas se eliminan posteriormente durante el proceso de



revelado químico –generalmente las áreas serán utilizadas como superficies para soldar la máscara de soldadura restante se cura completamente obteniéndose un acabado elástico.

**Figura 2.47: Máscara de soldadura.**

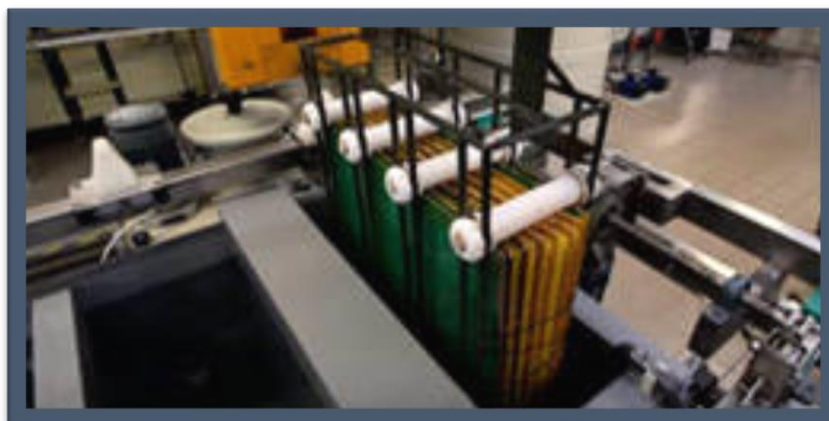


**Fuente:** (PCBWay, 2019).

✓ **ACABADO SUPERFICIAL.**

En este proceso se aplican varios acabados a la superficie expuesta del cobre. Esto se realiza con el fin de proteger la superficie y conseguir una soldadura óptima. Los diferentes acabados pueden incluir Electroless Nickel Immersion Gold, HASL, plata por inmersión, etc. Se realizan comprobaciones de grosor y de soldadura.

**Figura 2.48: Acabado superficial.**



**Fuente:** (PCBWay, 2019).

✓ **PERFIL.**

Este es el proceso de corte de los paneles fabricados a las medidas y formas especificadas basadas en el diseño del cliente como se definen en los archivos gerber. Hay tres opciones disponibles al proporcionar la matriz o vender el panel – marcado, enrutado o perforado. Todas las dimensiones se comparan con los gráficos aportados por el cliente para asegurarse de que el panel tiene las dimensiones adecuadas (PCBWay, 2019).

**Figura 2.49: Perfil y cortado.**



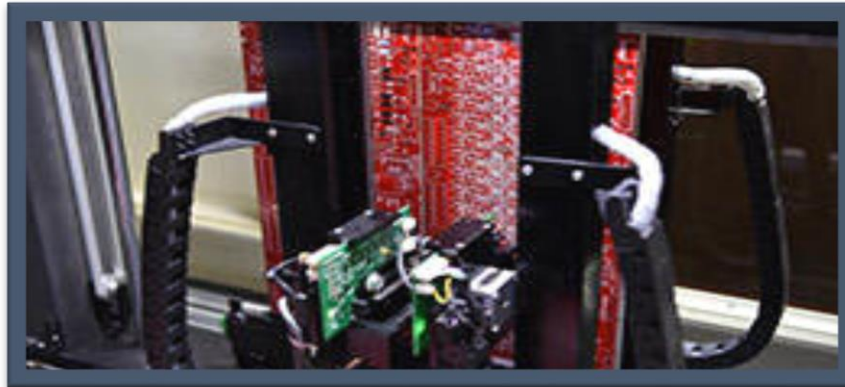
**Fuente:** (PCBWay, 2019).

✓ **COMPROBACIÓN ELÉCTRICA.**

Se utiliza para comprobar la integridad de las pistas y de las conexiones pasantes – comprobar que no existen circuitos abiertos o cortocircuitos en la placa terminada. Hay dos métodos de comprobación, sonda volante para pequeños volúmenes y para grandes volúmenes. Comprobamos eléctricamente cada PCB multicapa con los datos originales de la placa. Utilizando la sonda volante se comprueba cada

red para asegurarse de que está completa (sin circuitos abiertos) ni en corto con otras redes.

**Figura 2.50: Comprobación eléctrica.**



**Fuente:** (PCBWay, 2019).

✓ **INSPECCIÓN FINAL.**

En el último paso del proceso, un equipo de inspectores de precisa vista dan a cada placa una delicada comprobación añadida. Comprobando visualmente la PCB con los criterios de aceptación y utilizando inspectores “aprobados” por PCBWay. Utilizando inspección visual manual y automática– compara la PCB con los gerber y es más rápida que la vista humana, pero aún requiere verificación humana. Todos los pedidos están también sujetos a una inspección completa, incluyendo dimensionado, capacidad de soldadura, etc. (PCBWay, 2019).



**Figura 2.51: Inspección final.**



**Fuente:** (PCBWay, 2019).

## **2.4. HIPÓTESIS DE LA INVESTIGACIÓN**

### **2.4.1. HIPÓTESIS GENRAL.**

- Mediante la asistencia virtual y la monitorización de consumo eléctrico en tiempo real, un usuario de energía eléctrica podrá tener mayor eficiencia a la hora de consumir energía eléctrica, además de poder detectar anomalías en el conexionado de su sistema eléctrico de hogar.

### **2.4.2 HIPÓTESIS ESPECÍCAS.**

- Se mejorará la eficiencia de consumo energético.
- Se controlará y monitoreará de manera intuitiva el consumo eléctrico mediante la asistencia virtual.
- Gracias al comparativo de KWh en cada circuito se podrá limitar y tomar acciones respecto al consumo energético.
- Mediante el asistente virtual podremos verificar el conexionado de un circuito en tiempo real, de esta manera se podrá verificar si existe algún tipo de anomalía como cortos circuitos y fuga eléctrica a tierra.

## **CAPÍTULO III**

### **MATERIALES Y MÉTODOS**

#### **DISEÑO DE LA METODOLOGÍA DE INVESTIGACIÓN**

La metodología utilizada es de enfoque cuantitativo y de tipo aplicativo: debido a que está centrada en encontrar un mecanismo que permita lograr nuestro objetivo, el cual es la eficiencia energética mediante la asistencia virtual.

#### **3.1. MÉTODO Y DISEÑO DE LA INVESTIGACIÓN**

##### **3.1.1 MÉTODO**

El método utilizado es de carácter inductivo ya que se busca analizar datos particulares simulados y llevar a una conclusión general llevar los datos analizados en los prototipos creados para la lectura y visualización de datos requeridos y concluir a un entorno general de diferentes hogares del lugar.

##### **3.1.2. DISEÑO.**

Este plan se desarrollará experimentalmente, ya que se requieren datos precisos y compararlos con la realidad es por eso que se aplicara el método experimental ya que analizaremos las variables en diferentes lapsos de tiempo y se crearan prototipos y diseños para la simulación.

#### **3.2. POBLACIÓN Y MUESTRA.**

La población y muestra estará representada por los usuarios de energía eléctrica clientes de ELECTRO PUNO S.A.A.

#### **3.3. UBICACIÓN Y DESCRIPCIÓN DE LA POBLACIÓN.**

##### **3.3.1. UBICACIÓN.**

La ubicación del siguiente proyecto de investigación se realizó en:

- ✓ Región: PUNO.
- ✓ Departamento: PUNO
- ✓ Provincia: PUNO.
- ✓ Distrito: PUNO
- ✓ Altitud: 3,827 msnm.
- ✓ Latitud Sur: 13°00'66"00" y 17°17'30".
- ✓ Longitud Oeste: 71°06'57" y 68°48'46".

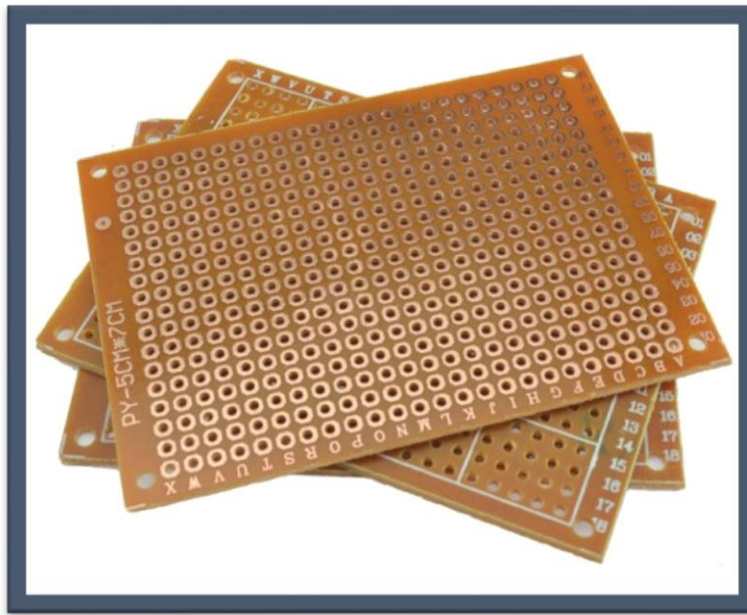
### 3.4. MATERIALES.

- ❖ **Protoboard.**
- ❖ **Placa de cobre perforada.**

Esta placa es ideal para la creación de prototipos y la construcción de circuitos, puede llevar mucho tiempo para utilizar con componentes con ciertas configuraciones de pines.

Esta es una placa para PCB soldable, tiene un solo lado y está hecha de puntos de cobre. Estas placas son especialmente útiles para conservar un prototipo o experimento que acaba de crear en una protoboard, soldando así todas las piezas en su lugar. .

Figura 3.52: Placa de cobre perforada.



Fuente: (MicroJPM, 2019).

- ❖ Arduino Uno.
- ❖ Arduino Due.
- ❖ Amplificador operacional Lm358P.
- ❖ Sensor de corriente SCT-013-030.
- ❖ Sensor de corriente SCT-013-000.
- ❖ Placa de fibra de vidrio bañada en cobre.
- ❖ Cables conectores para Arduino.
- ❖ Micrófono behringer C1.
- ❖ Placa de audio U Phoria UM2.
- ❖ Cable y conectores para micrófono XLR.
- ❖ Sensor de voltaje ZMPT101B.
- ❖ Computador portátil.
- ❖ Resistencia cerámica de 20 Ohm.
- ❖ Estaño de 0.8mm.

- ❖ Pistola de soldar de 70w.
- ❖ Pines conectores hembra y macho.
- ❖ Modulo relé de 16 canales.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1 PRIMER PROTOTIPO.

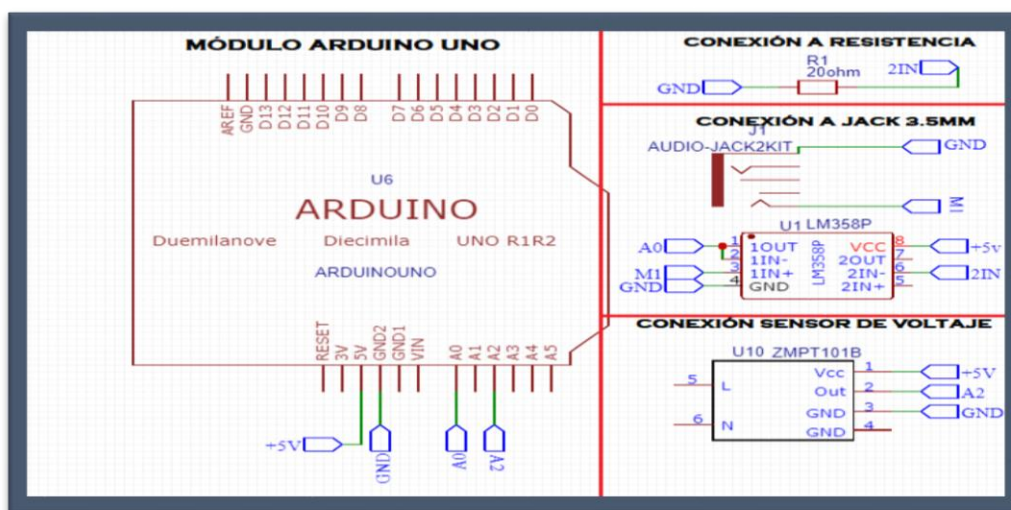
##### 4.1.1. MODELO ESQUEMÁTICO ARDUINO UNO.

Para el diseño y prototipo de los sensores que actuaran en conjunto con el Arduino, se desarrolló un esquemático con todos los componentes electrónicos para así de esa manera continuar haciendo las pruebas en un protoboard para el posterior diseño de un prototipo.

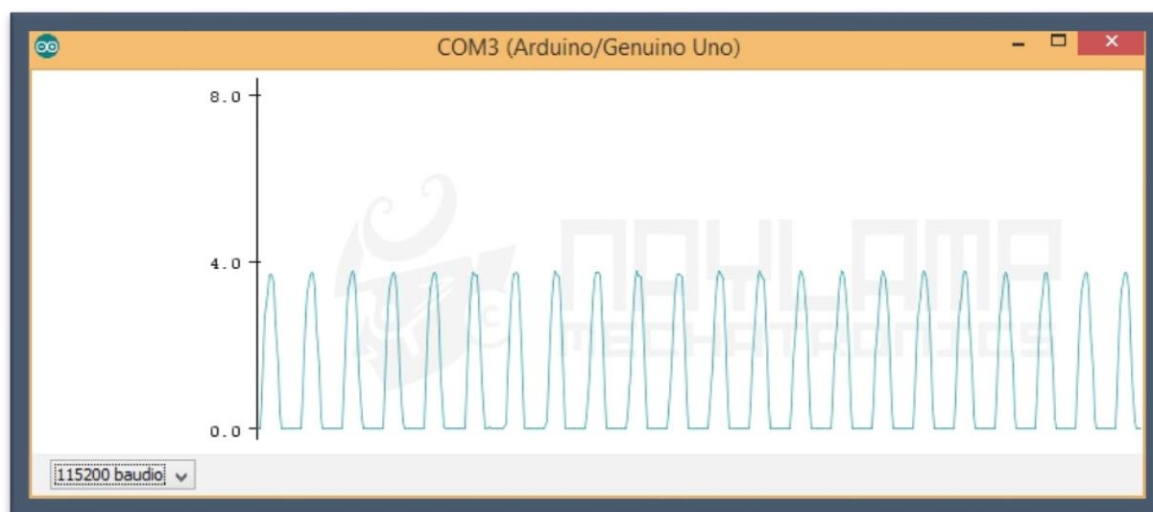
##### 4.1.1.1 CIRCUITO PARA ACONDICIONAR LA SEÑAL.

La salida de nuestro sensor es una señal alterna, cuyos valores analógicos están de entre 0 a +3.3V; el rango del sensor puede ser aún menor es por ello que la parte negativa de este, podría malograr el módulo de Arduino Due; para prevenir este posible fallo rectificaremos la señal de entrada y trabajaremos solo con las señales positivas de entrada; para ello usaremos el Amplificador Operacional LM358, que trabajara con polaridad positiva; de esta manera eliminaremos la parte negativa trabajando así con una rectificación de media onda.

Figura 4.53: Esquemático de conexión Arduino UNO



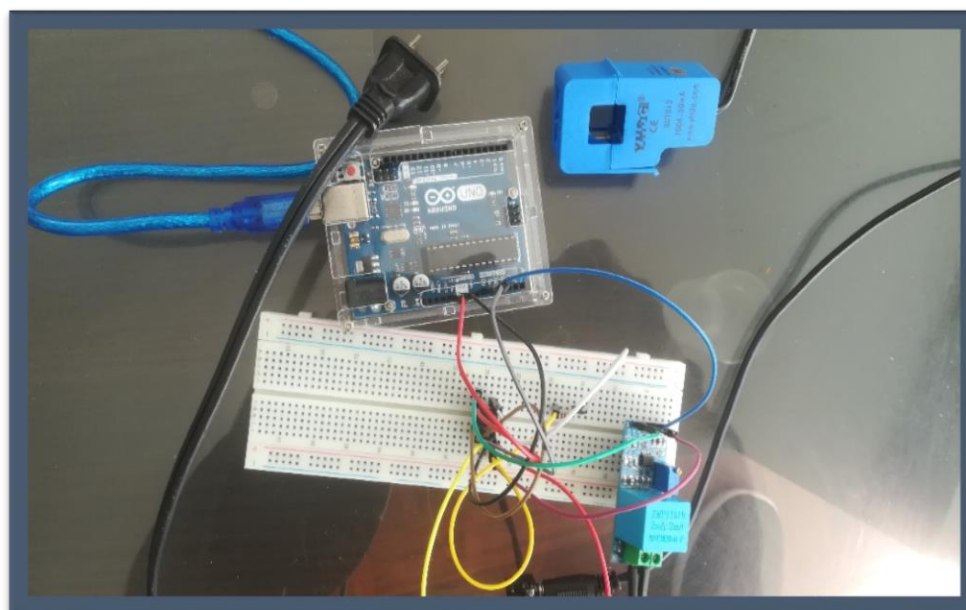
Elaboración propia

**Figura 4.54: Rectificación de media onda con A.O. LM358.**

**Fuente:** («Tutorial sensor de corriente AC no invasivo SCT-013», 2016).

#### **4.1.2. PRUEBA DE FUNCIONAMIENTO PROTOBOARD.**

Luego de realizado el esquemático iniciamos a hacer el conexionado y posterior prueba en protoboard, así como haciendo la respectiva programación en Arduino IDE ver Anexo 1.

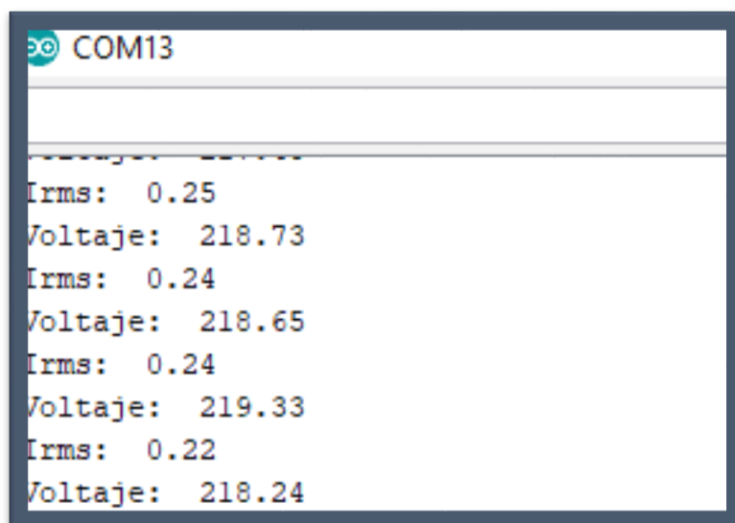
**Figura 4.55: Conexionado protoboard.**

**Elaboración propia.**



Una vez realizada la conexión respectiva del Sensor de corriente SCT-013-000 y el sensor de voltaje ZMPT101B con el módulo Arduino UNO, se realizó la prueba respectiva y la comparación de medición con un multímetro digital SAMWIN para ello se usó un foco incandescente de 50 watt; cómo se puede apreciar en la Figura 7.6 y 7.7 las lecturas hechas por los sensores y modulo Arduino se asemejan bastante al del multímetro digital pudiéndose calibrar en adelante el sensor de corriente para más precisión.

**Figura 4.56: Medición de voltaje y corriente con Arduino UNO.**



Elaboración propia.

**Figura 4.57: Medición de voltaje y corriente por Multímetro digital.**



Elaboración propia.

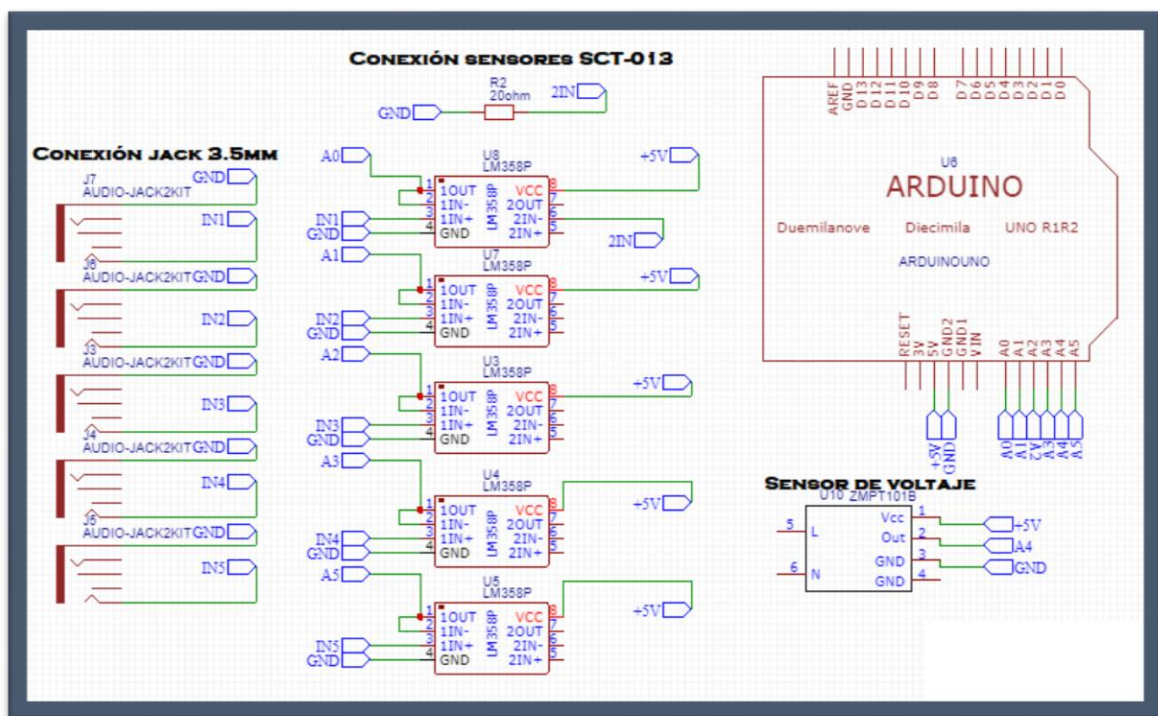


**4.2 SEGUNDO PROTOTIPO.**

**4.2.1. SEGUNDO MODELO ESQUEMÁTICO ARDUINO UNO.**

Luego de comprobar que el primer esquemático y conexionado funciona en el protoboard se realizó un segundo esquemático de conexión para 01 sensor de voltaje ZMPT101B, 01 sensor de corriente SCT-013-000 y 04 sensores de corriente SCT-013-030.

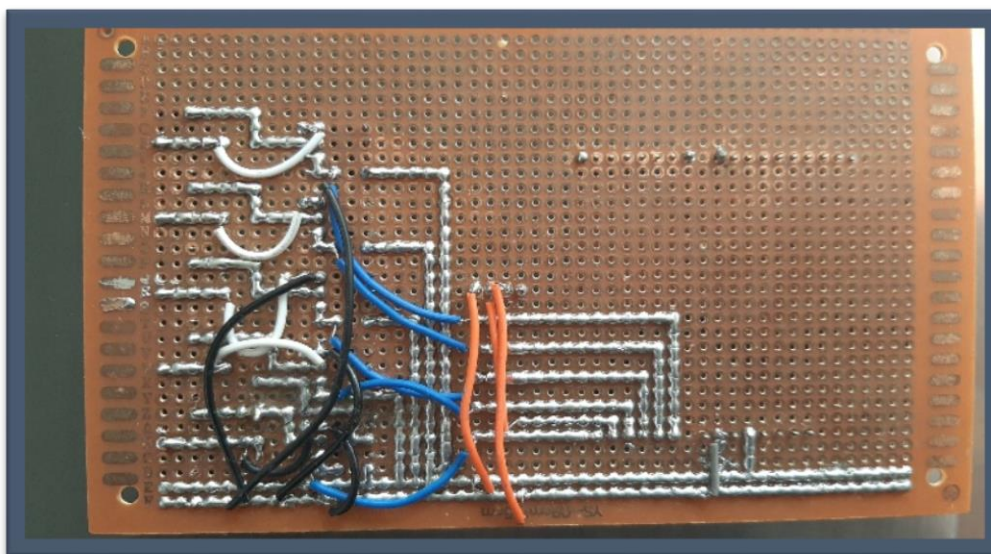
**Figura 4.58: Segundo esquemático Arduino UNO.**



Elaboración propia.

**4.2.2. PRUEBA DE FUNCIONAMIENTO DE PROTIPO N° 2.**

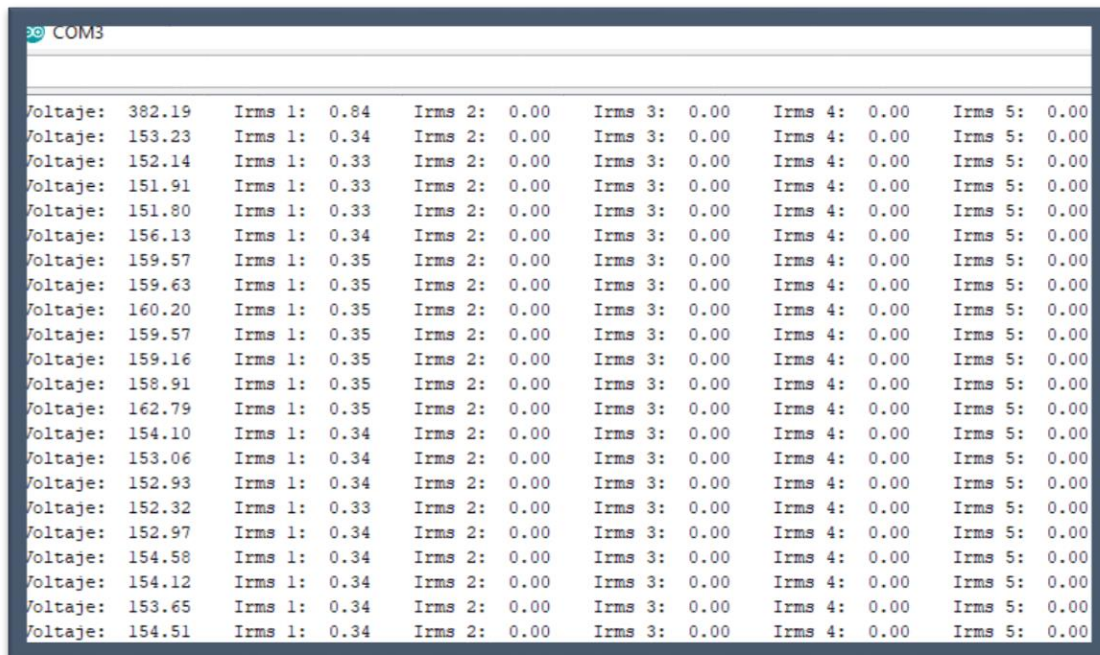
Para realizar las pruebas se utilizó una placa de cobre con perforaciones, para así poder soldar el conexionado ya establecido en el segundo esquemático en esta prueba se pretende analizar los valores y funcionamiento de los 6 sensores en conjunto tanto el de voltaje, así como los de intensidad de corriente.

**Figura 4.59: Prototipo en placa con perforaciones (Vista superior).****Elaboración propia.****Figura 4.60: Prototipo en placa de cobre con perforaciones (Vista inferior)****Elaboración propia.**

Luego de haberse soldado todos los componentes en la placa con perforaciones se realizó pruebas de medición para comprobar una vez más el funcionamiento del mismo, esta vez se está utilizando 05 sensores de corriente y 01 sensor de voltaje con el módulo Arduino UNO; para lo cual se modificó el anterior código de programación en Arduino IDE (véase Anexo 2).

Luego de realizada las pruebas pudimos observar que las lecturas eran correctas; sin embargo, el intervalo de entre cada lectura era de 2 -4 segundos, lo cual no es ventajoso ya que necesitamos lecturas mucho más rápidas para poder desarrollar nuestro sistema de detección de corto circuito y fuga a tierra más adelante.

**Figura 4.61: Lecturas de medición del segundo prototipo.**



Voltaje:	Irms 1:	Irms 2:	Irms 3:	Irms 4:	Irms 5:
382.19	0.84	0.00	0.00	0.00	0.00
153.23	0.34	0.00	0.00	0.00	0.00
152.14	0.33	0.00	0.00	0.00	0.00
151.91	0.33	0.00	0.00	0.00	0.00
151.80	0.33	0.00	0.00	0.00	0.00
156.13	0.34	0.00	0.00	0.00	0.00
159.57	0.35	0.00	0.00	0.00	0.00
159.63	0.35	0.00	0.00	0.00	0.00
160.20	0.35	0.00	0.00	0.00	0.00
159.57	0.35	0.00	0.00	0.00	0.00
159.16	0.35	0.00	0.00	0.00	0.00
158.91	0.35	0.00	0.00	0.00	0.00
162.79	0.35	0.00	0.00	0.00	0.00
154.10	0.34	0.00	0.00	0.00	0.00
153.06	0.34	0.00	0.00	0.00	0.00
152.93	0.34	0.00	0.00	0.00	0.00
152.32	0.33	0.00	0.00	0.00	0.00
152.97	0.34	0.00	0.00	0.00	0.00
154.58	0.34	0.00	0.00	0.00	0.00
154.12	0.34	0.00	0.00	0.00	0.00
153.65	0.34	0.00	0.00	0.00	0.00
154.51	0.34	0.00	0.00	0.00	0.00

**Elaboración Propia.**

### 4.3 TERCER PROTOTIPO.

Luego de analizar los resultados obtenidos con Arduino UNO, decidimos cambiar dicho módulo, ya que como antes mencionado necesitaremos mayor rapidez de lectura de las entradas analógicas, así como mayor rapidez de procesamiento de datos, para ello elegimos a Arduino DUE que tiene mayores capacidades de conversores de señal analógica a digital, así como mayor capacidad de procesamiento,

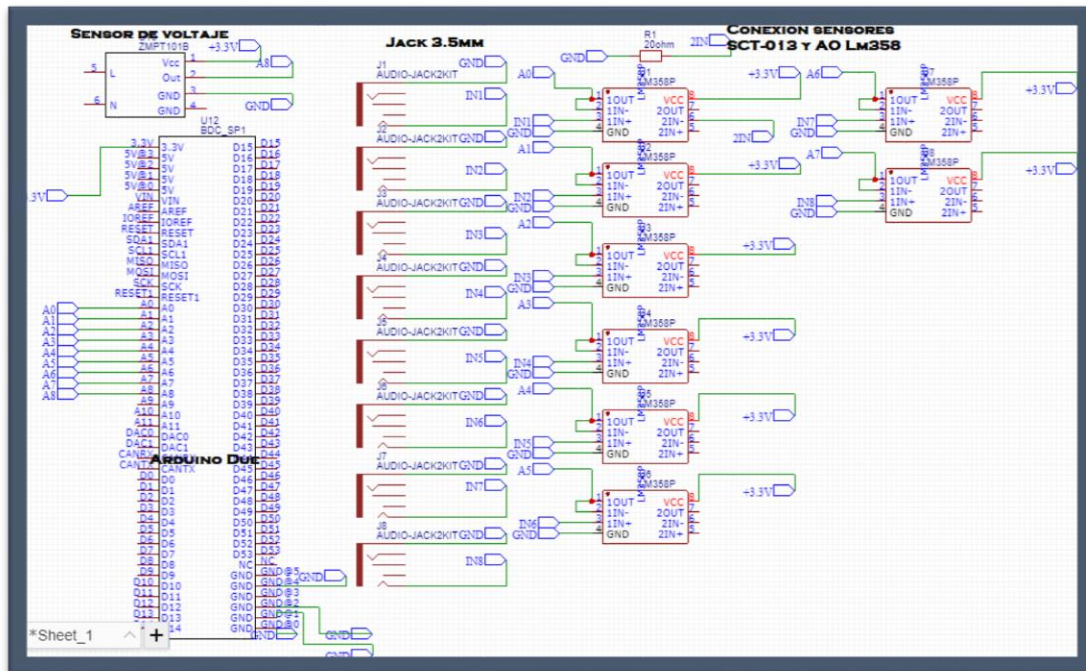
#### 4.3.1. MODELO ESQUEMÁTICO PARA ARDUINO DUE.

Se desarrollo un esquemático basándonos en el conexionado de Arduino DUE, para luego poder realizar el diseño del circuito impreso; ya que las pruebas con



Arduino UNO en cuanto a medición y precisión de lectura fueron satisfactorias; por ello nos basaremos en el esquemático anterior.

Figura 4.62: Esquemático Arduino Due (Tercer Prototipo).

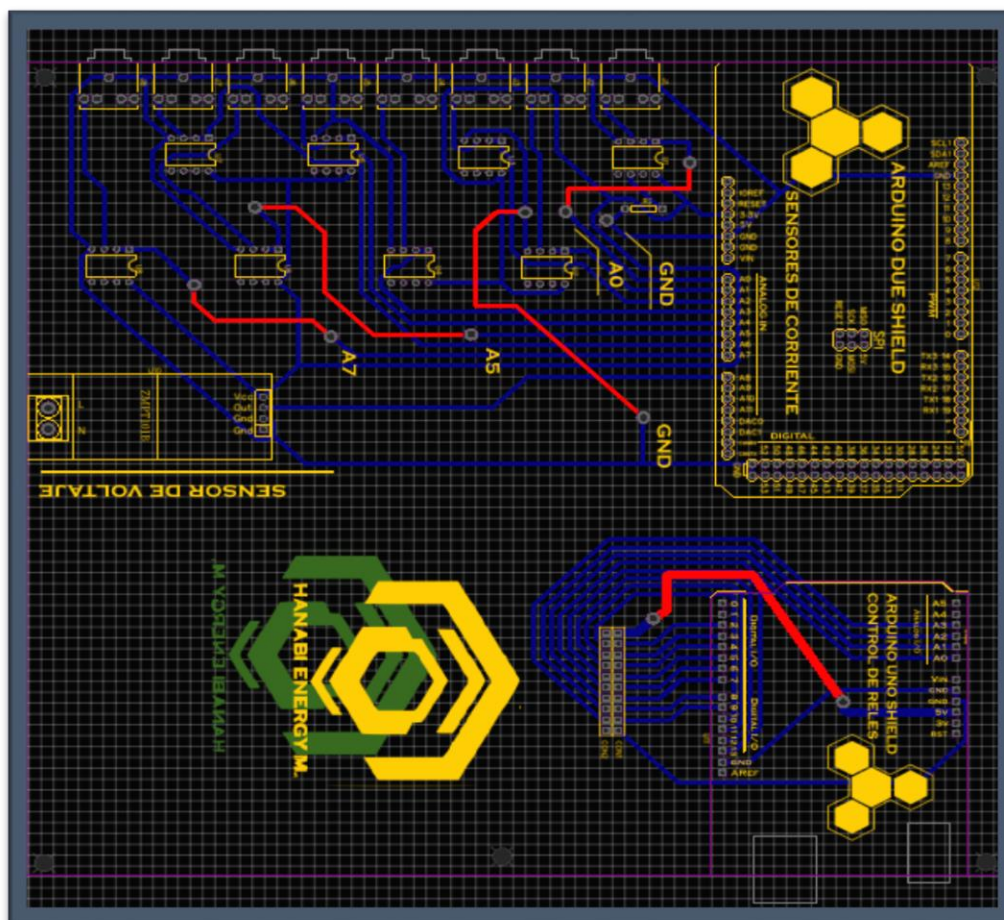


Elaboración Propia.

**4.3.2. ELABORACIÓN Y PRUEBA DE FUNCIONAMIENTO DE PROTIPO N° 3.**

Ya que sabemos las bases del conexionado y que tenemos la certeza de que funciona; el prototipo para esta prueba se realizó mediante el método del planchado, habiendo ya realizado el modelo del circuito con el software EASYEDA, para la posterior impresión del negativo en papel fotográfico y el grabado de este mismo en una placa de fibra de vidrio bañado en cobre.

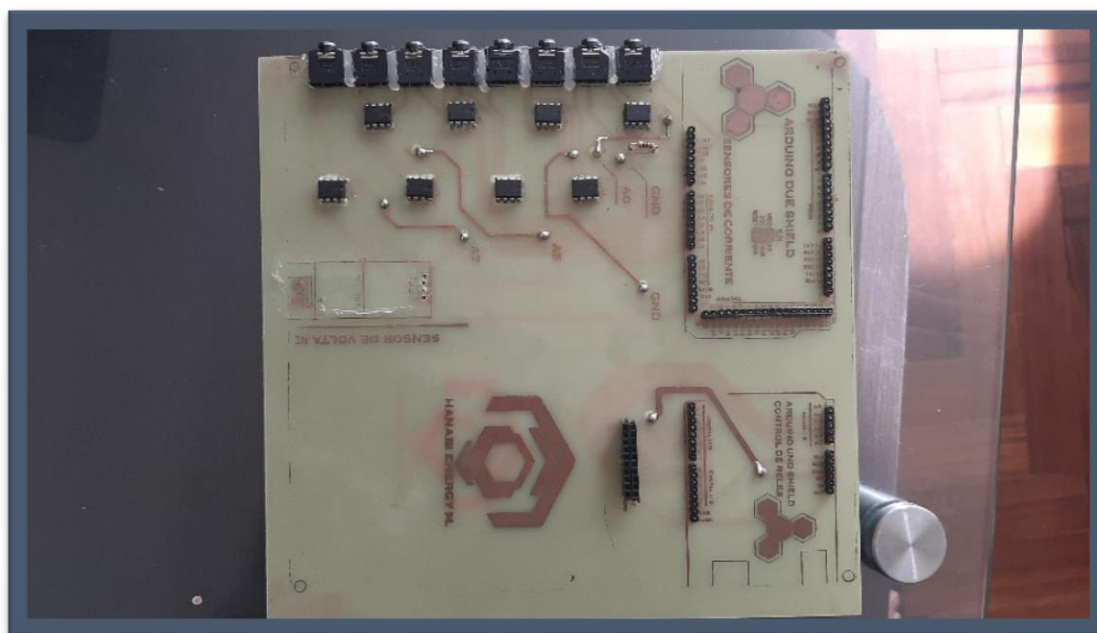
Figura 4.63: Diseño de Circuito impreso para el 3er prototipo en EASYEDA.



Elaboración propia.

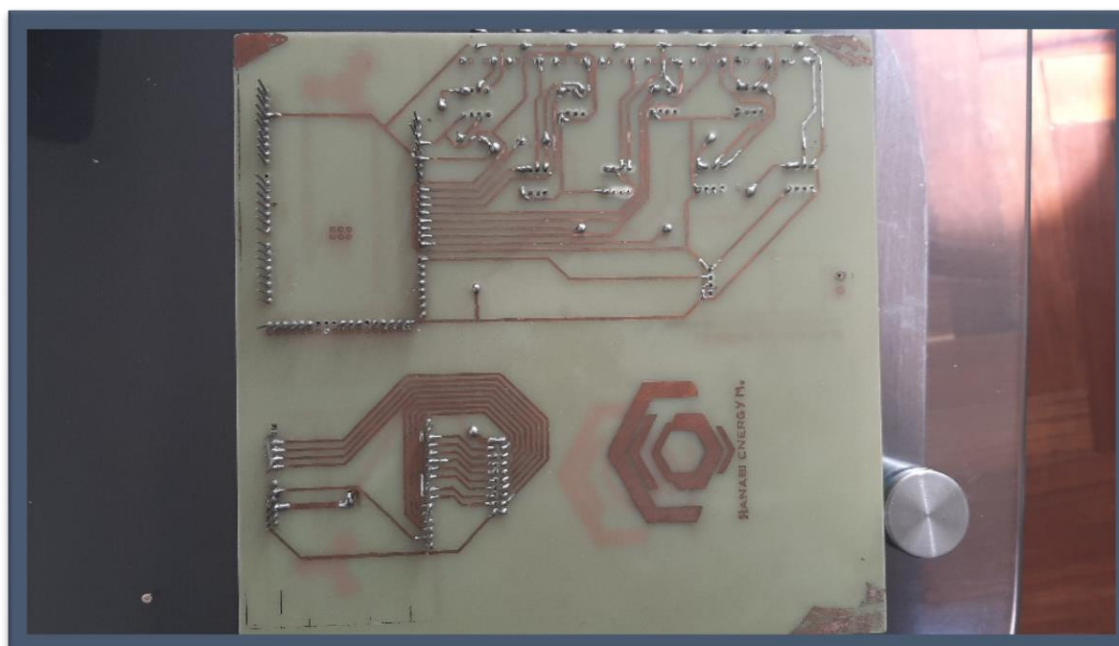
Luego de realizado el modelo del circuito impreso en EasyEda se procedió a elaborar el PCB por el método del planchado ya mencionado, obteniendo resultados mucho mejores, nuestra placa mide 19cm x 20cm, siendo esta considerablemente grande; sin embargo los resultados de intervalos de lectura de datos son los esperados, gracias a la capacidad de procesamiento y a la capacidad de conversión de señal analógica a digital de Arduino Due se lograron intervalos de 200 milisegundos siendo este intervalo suficiente para el posterior desarrollo de un mecanismo para prueba de cortocircuito y fuga a tierra.

**Figura 4.64: Circuito impreso para Arduino Due (Vista superior 3er prototipo)**



Elaboración propia

**Figura 4.65: Circuito impreso para Arduino Due (Vista inferior 3er prototipo)**



Elaboración propia

#### 4.4. PROTOTIPO FINAL.

El anterior prototipo nos trajo satisfactoria y eficientemente los datos requeridos, sin embargo, el tamaño de este es considerable y debido a que las pistas de la placa están descubiertas, tienden a corroerse pudiendo traernos problemas de precisión en el futuro,

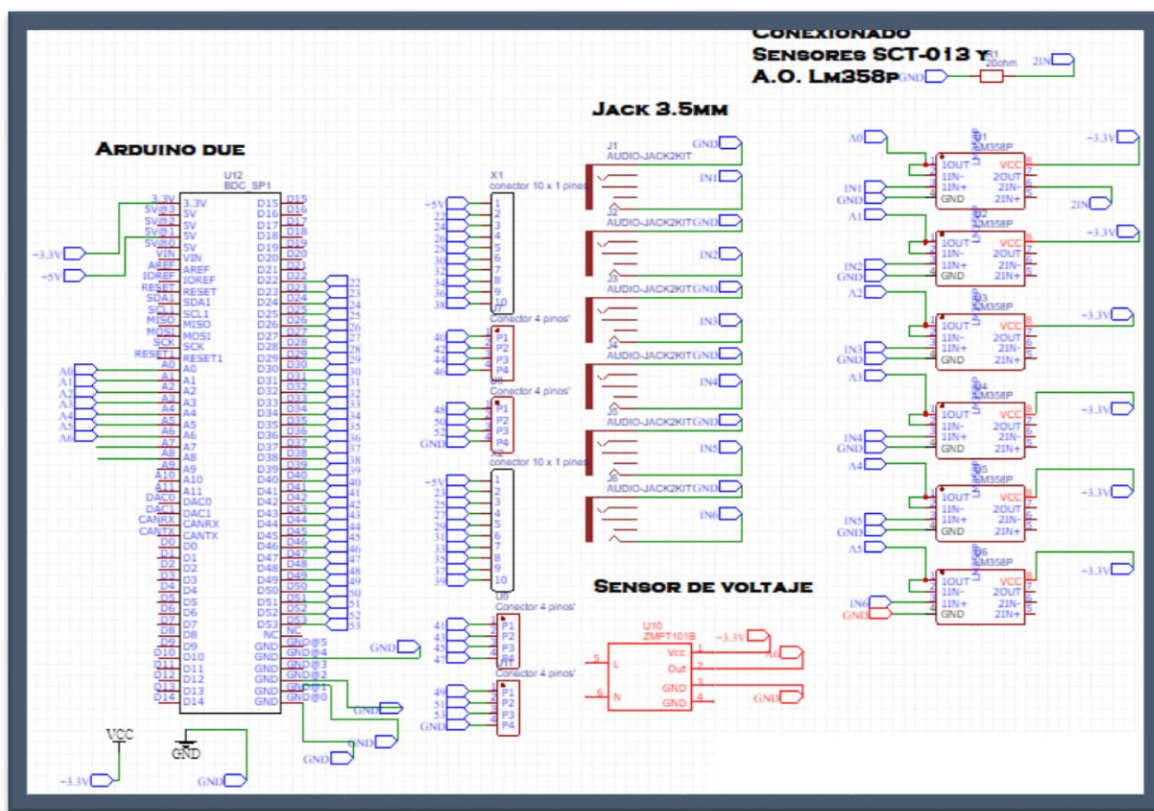


es por eso que se decidió mandar a fabricar nuestro prototipo a China con fabricantes especializados, en este caso JLCPCB.

#### 4.4.1. MODELO ESQUEMÁTICO PARA ARDUINO DUE P.FINAL.

Se realizo algunos cambios en el esquemático para este prototipo reduciendo el tamaño de este mismo a 10cm x 10 cm y teniendo, 06 entradas Jack 3.5mm para el conexionado de los sensores SCT-013.

Figura 4.66: Esquemático para el prototipo final.



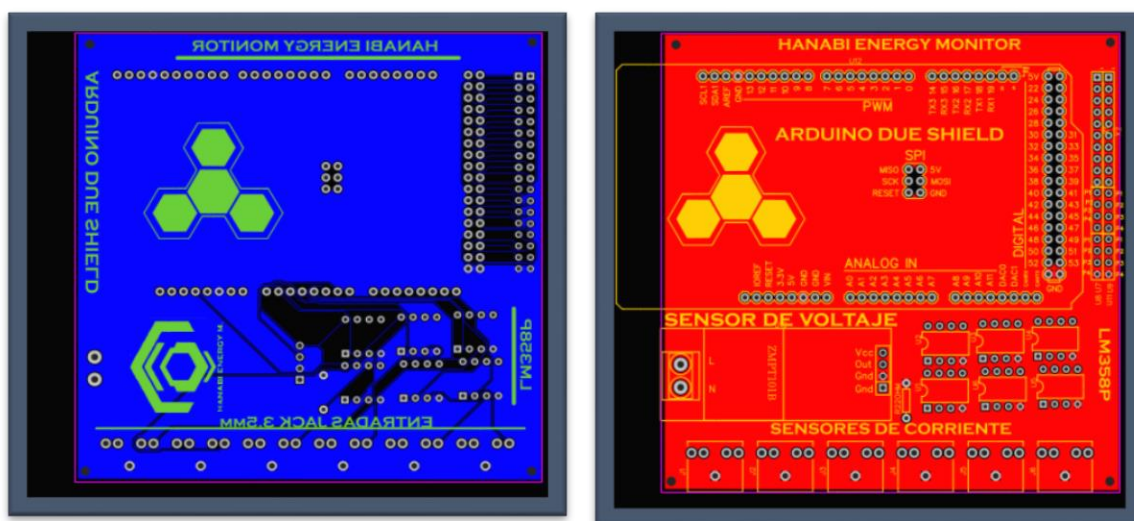
Elaboración propia.

#### 4.4.2. ELABORACIÓN Y PUEBAS DEL PROTOTIPO FINAL.

Luego de haber realizado nuestro esquemático se realizó el modelado del PCB en el software EasyEda obteniendo así un modelo de 10cm x 10 cm.

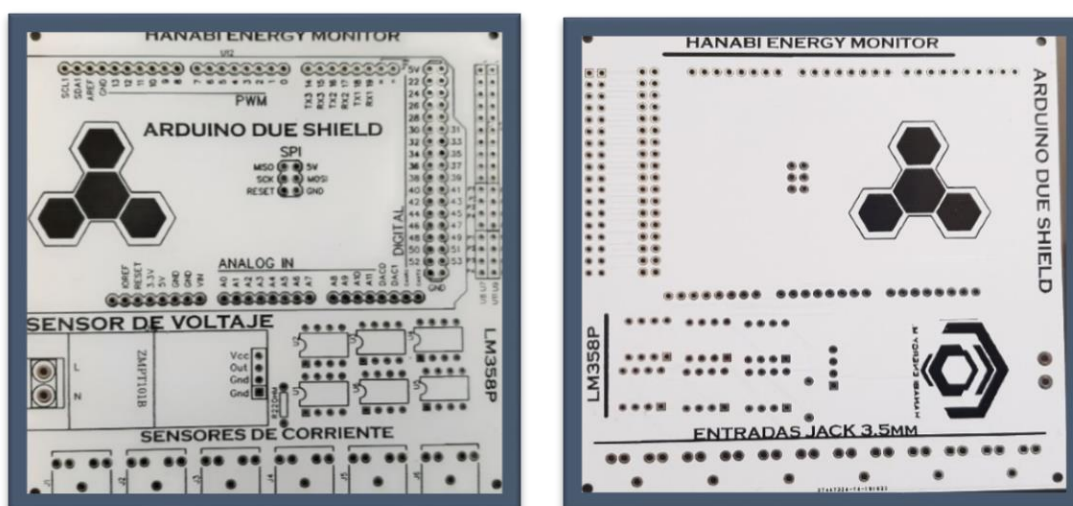
Como ya se había mencionado, el prototipo final se mandó a fabricar a China, con el fabricante JLCPCB teniendo un circuito impreso de mayor calidad.

Figura 4.67: Modelo de PCB elaborado en EasyEda (Vista superior e inferior).



Elaboración propia.

Figura 4.68: Circuito impreso final fabricado por JLPCB (Vista superior e inferior).



Elaboración Propia.



Figura 4.69: Circuito impreso final ensamblado (Vista superior e inferior).



Elaboración propia.

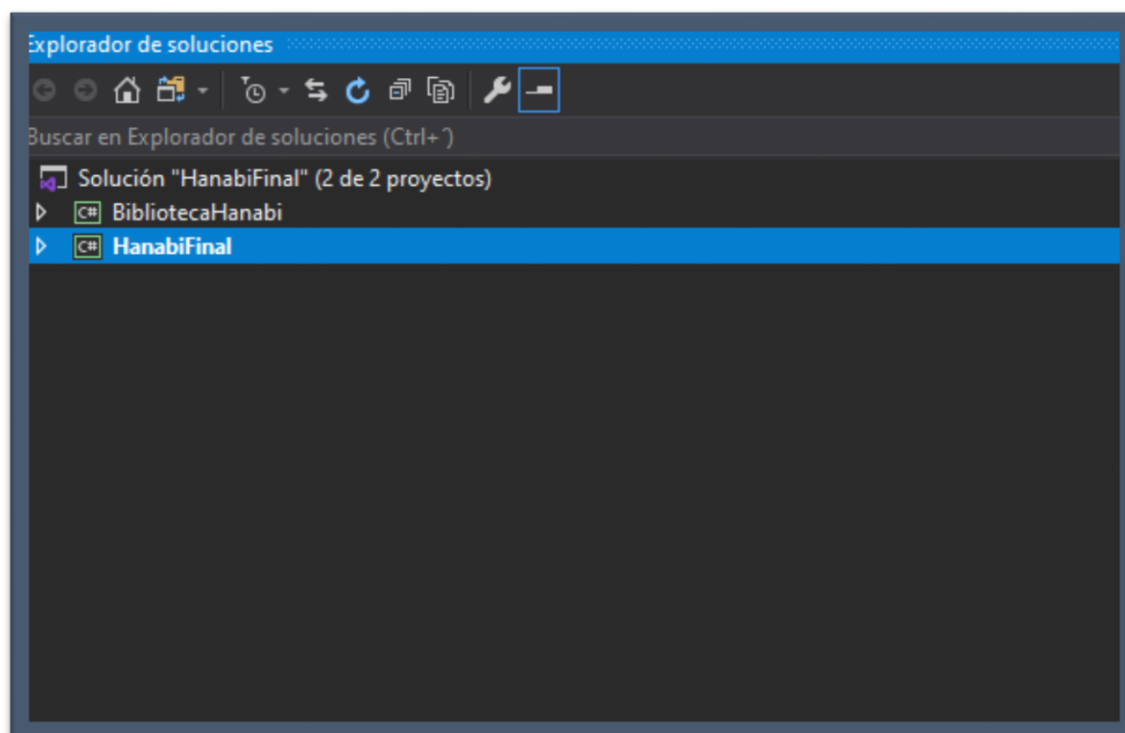
Las pruebas realizadas con el circuito impreso son satisfactorias como ya se había previsto el intervalo de lectura es idéntica al del prototipo 3, con la única diferencia que este es de mucho menor tamaño y a su vez tiene las pistas protegidas.

## 4.5. CREACIÓN DE ASISTENTE VIRTUAL E INTERFAZ GRÁFICO.

### 4.5.1. PARTES DEL PROYECTO EN WPF.

Para la creación del asistente virtual e interfaz gráfico se utilizará el lenguaje de programación C# WPF este lenguaje de programación es propio de Microsoft y en conjunto con la plataforma de Xamarin nos permitirá crear un interfaz amigable con el usuario.

Creemos un nuevo proyecto en la plataforma Visual estudio con el lenguaje de programación WPF C#, el cual llamaremos HanabiFinal, este proyecto estará dividido en dos partes, una que será la plataforma principal (HanabiFinal) y contendrá todas las formas para el interfaz y la 2da que será netamente para las gramáticas y base de datos de los comandos principales (BibliotecaHanabi).

**Figura 4.70: Partes del proyecto HanabiFinal.**

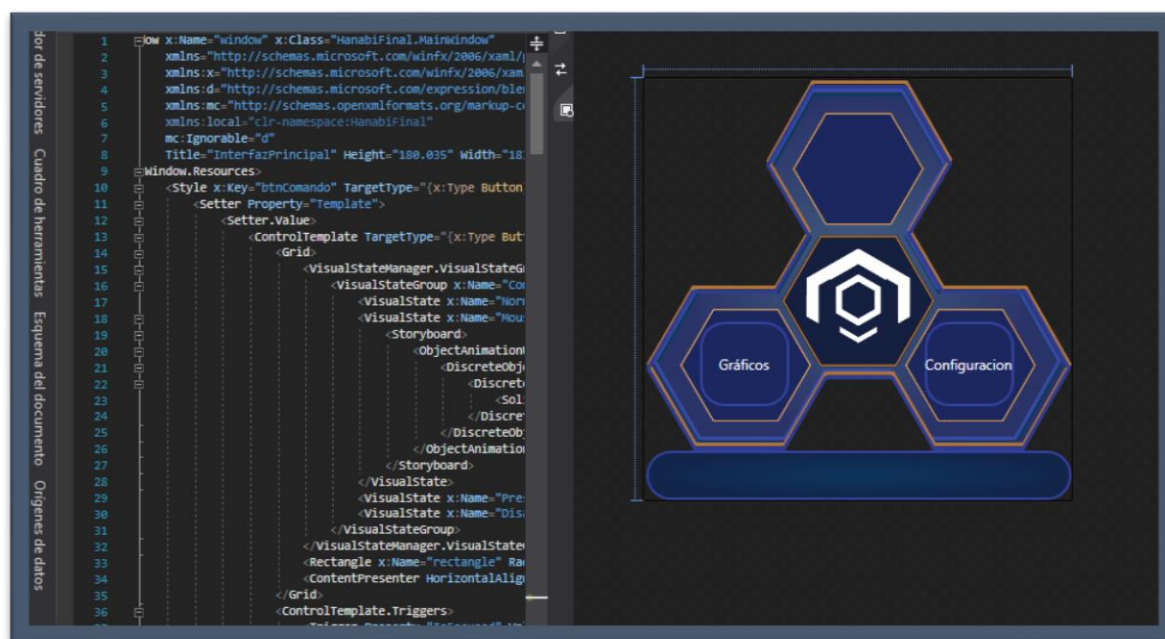
**Elaboración propia.**

#### **4.5.2. CREACIÓN DEL INTERFAZ DE HANABI.**

El interfaz principal constará de 02 botones, “Gráficos” y “Configuración”, ambos botones tienen la función de llevarnos a diferentes ventanas, el interfaz principal también nos mostrara la predicción de palabras del asistente en la parte inferior; este interfaz hablando gráficamente esta desarrollado con xamarin forms (Véase Anexo N°3).

En cuanto a la programación de las funciones de cada uno de estos todo está programado en lenguaje C# (Véase Anexo N° 4),

Figura 4.71: Interfaz principal HanabiFinal.



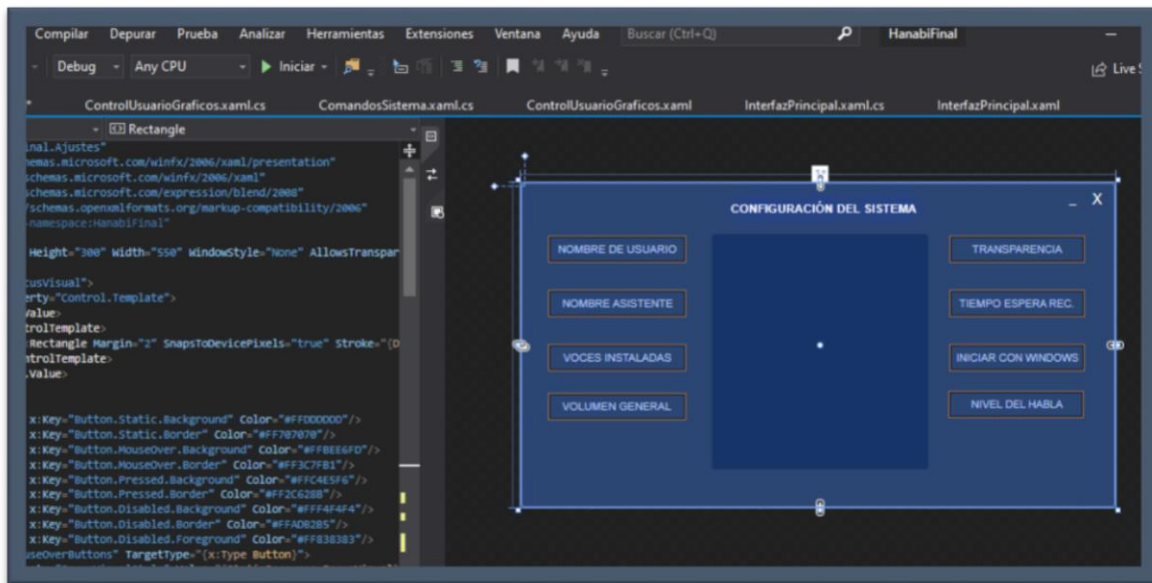
Elaboración propia.

En caso del botón configuración este nos llevara a las propiedades principales del sistema del asistente los cuales son:

- Nombre del Asistente.
- Nombre del usuario quien usara el asistente.
- Voz del asistente.
- El volumen de la voz del asistente.
- Nivel de precisión mínimo para que el asistente responda.
- Transparencia de la ventana.
- Tiempo de espera para desactivar el micrófono.
- Opción si desea iniciar con Windows.

En cuanto a la programación de las funciones de este mismo (Véase Anexo N° 5).

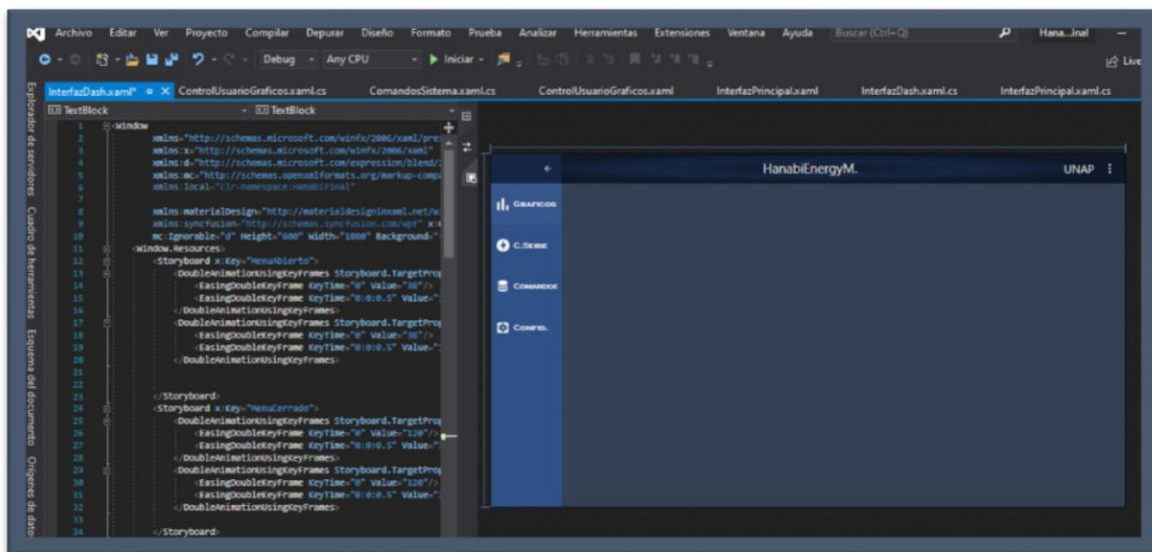
Figura 4.72: Interfaz de configuración del sistema.



Elaboración propia.

En cuanto al botón “Gráficos” este nos llevara a otra ventana que contendrá el interfaz Grafico (InterfazDash) el cual estará dividido en 3 botones, “Gráficos”, Comandos Serie “C.Serie”, “Comandos”; en cuanto a la programación de las funciones de este interfaz (Véase Anexo N° 6).

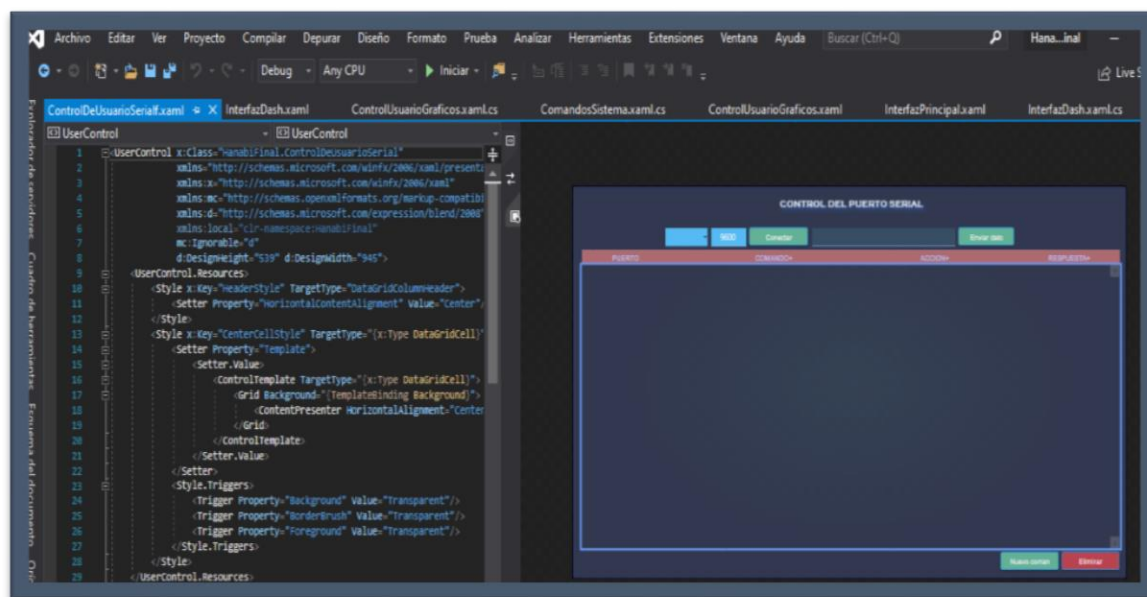
Figura 4.73: Interfaz del botón Gráficos (InterfazDash).



Elaboración Propia.

En el botón comandos serie no abrirá una pestaña que contendrá una tabla con todos los comandos que se conectarán con Arduino Due, estos estarán especializados en el control de los relés y otras funciones; para el código de las funcionalidades del interfaz de comandos del puerto serial (Véase Anexo N° 7).

**Figura 4.74: Interfaz de conexión con puerto serial.**

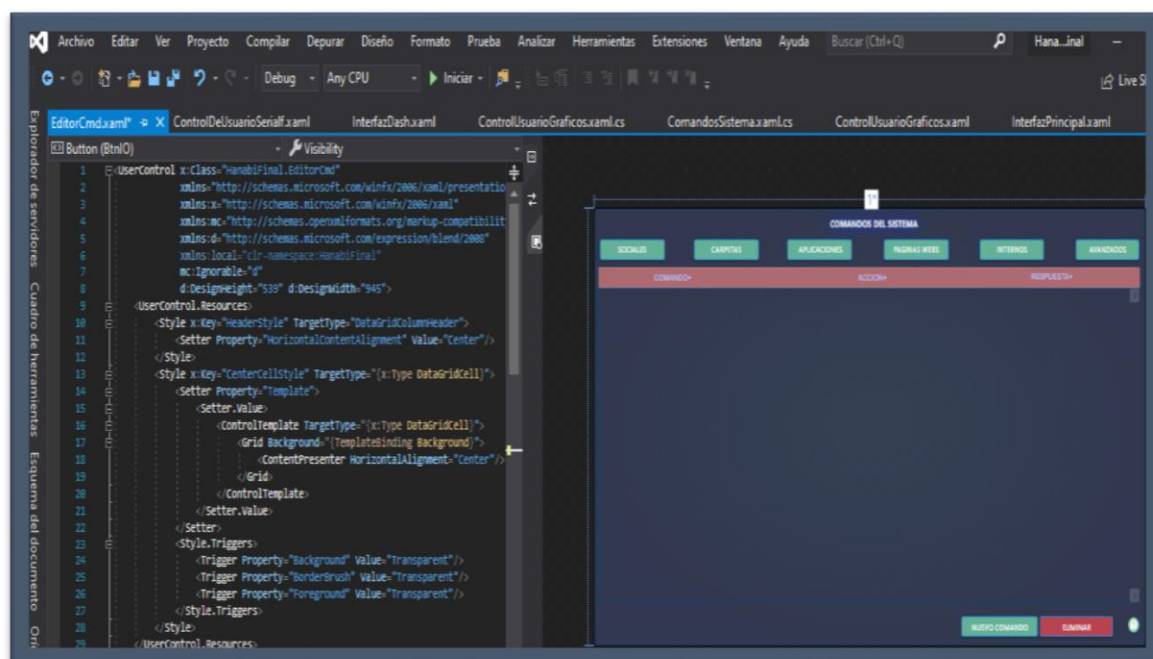


**Elaboración propia.**

el botón “Comandos” nos llevara a otra pestaña que contendrán los comandos del asistente, estos será comandos de uso personal y de ayuda con el sistema operativo del usuario, para la realización de este código nos basamos en el curso de programación de “El vínculo digital Tv” (ElvinculoDigitalTV, 2019); para más detalle del código de programación usado (Véase Anexo N° 8).



Figura 4.75: Comandos del Sistema.



Elaboración propia.

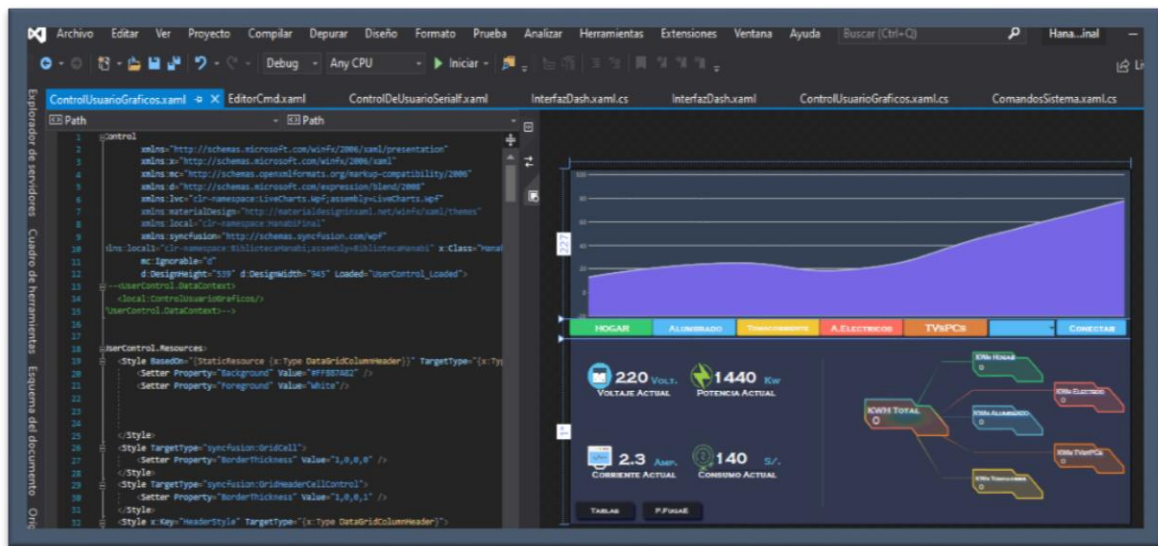
El botón “Gráficos”, contiene la parte más importante del software ya que este nos llevará a una pestaña que contendrá los siguientes datos.

- Combo que contiene el número de puerto.
- Botón para conectar el puerto serial.
- El voltaje actual.
- Intensidad de corriente.
- Potencia.
- Consumo KWh.
- Consumo en S/.
- Botón con una base de datos que contendrá en una tabla de dos pestañas los datos de potencia y de consumo KWh.
- Botón para la prueba de fuga eléctrica a tierra.

- Chart que muestra gráficamente la intensidad, voltaje y potencia actual según corresponda.

Para más detalle del código de programación usado en C# (Véase Anexo N° 9).

**Figura 4.76: Interfaz de Gráficos y Datos.**



Elaboración propia.

Para hallar el costo en soles aproximado según el KWh que se está midiendo en el momento, se empleó el pliego tarifario en la ciudad de puno declarada por OSINERGMIN, más específicamente la tarifa BT5B residencial.

**Tabla N° 4.7: Pliego tarifario OSINERGMIN.**

TARIFA	TARIFA CON DOBLE MEDICIÓN DE ENERGÍA 2E		
BT5A:			
	a) Usuarios con demanda máxima mensual de hasta 20kW en HP y HFP		
	Cargo Fijo Mensual	S./mes	11.65
		ctm.	194.6
	Cargo por Energía Activa en Punta	S./kW.h	6
		ctm.	
	Cargo por Energía Activa Fuera de Punta	S./kW.h	23.45

	Cargo por Exceso de Potencia en Horas Fuera de Punta	S./kW-mes	72.92	
	<b>b) Usuarios con demanda máxima mensual de hasta 20kW en HP y 50kW en HFP</b>			
	Cargo Fijo Mensual	S./mes	11.65	
	Cargo por Energía Activa en Punta	ctm. S./kW.h	231.3 3	
	Cargo por Energía Activa Fuera de Punta	ctm. S./kW.h	23.45	
	Cargo por Exceso de Potencia en Horas Fuera de Punta	S./kW-mes	72.92	
<b>TARIFA BT5B:</b>	<b>TARIFA CON SIMPLE MEDICIÓN DE ENERGÍA 1E</b>			
No Residencial	Cargo Fijo Mensual	S./mes	4.03	
	Cargo por Energía Activa	ctm. S./kW.h	73.95	
<b>TARIFA BT5B</b>	<b>TARIFA CON SIMPLE MEDICIÓN DE ENERGÍA 1E</b>			<b>MCTER-Sin IGV</b>
<b>Residencial</b>	<b>a) Para usuarios con consumos menores o iguales a 100 kW.h por mes</b>			
	0 - 30 kW.h			
	Cargo Fijo Mensual	S./mes	3.88	3.58
	Cargo por Energía Activa	ctm. S./kW.h	53.43	47.25
	31 - 100 kW.h			
	Cargo Fijo Mensual	S./mes	3.88	3.58
	Cargo por Energía Activa - Primeros 30 kW.h	S./mes	16.03	14.18
	Cargo por Energía Activa - Exceso de 30 kW.h	ctm. S./kW.h	71.24	63.00
	<b>b) Para usuarios con consumos mayores a 100 kW.h por mes</b>			
	Cargo Fijo Mensual	S./mes	4.03	3.72
	Cargo por Energía Activa	ctm. S./kW.h	73.95	65.39

Fuente:(OSINERGMIN, 2019).



### 4.5.3. ELABORACION DE SISTEMA DE DETECCIÓN DE FUGA ELÉCTRICA A TIERRA Y CORTO CIRCUITO.

#### 4.5.3.1. DETECCIÓN DE CORTO CIRCUITO.

- **Corto circuito:** El corto circuito es un fallo en un aparato o línea eléctrica por el cual la corriente pasa directamente del conductor activo o fase al neutro o a tierra, esto en sistemas monofásicos de corriente alterna.

El cortocircuito se produce normalmente por los fallos en el aislante de los conductores, cuando estos quedan sumergidos en un medio conductor como el agua o por contacto accidental entre conductores aéreos por fuertes vientos o rotura de los apoyos.

- **Ley de Ohm:** En un circuito cerrado el voltaje, corriente eléctrica y la resistencia deben tener valores debidamente controlados para un buen funcionamiento del sistema. Una condición de cortocircuito queda determinada al eliminarse desde el punto de vista práctico, la resistencia de consumo del circuito. Según la ley de Ohm se tiene que

$$I = \frac{V}{R}$$

**Ec. N° 2 Ley de ohm.**

En un corto circuito la resistencia se acerca a cero lo que ocasiona que la intensidad de corriente suba dramáticamente y tienda al infinito; sin embargo, se debe tomar en cuenta diferentes factores tales como:

- Resistencia Eléctrica en Ohmios, de la línea.
- Protecciones Térmicas de la línea.
- Protectores Termo Magnéticos.
- Disyuntores Diferenciales en la línea, entre otros.

Con esta información analizada podemos decir que, si la intensidad de corriente sube anormalmente, se puede tomar ese dato y concluir que ha ocurrido un cortocircuito de esta manera con tan solo poner una condicional de lectura en nuestra línea de código podremos detectar el corto circuito en tiempo real; para ver en detalle el código de programación de cortocircuito (Véase Anexo N° 9).

**Gráfico N° 2: Parte de código de detección de corto circuito (Anexo 9).**

```

if (StrIncomingSerial1.Contains("V"))
{
    int foundS1 = StrIncomingSerial1.IndexOf("V");
    StrIncomingSerial1 = StrIncomingSerial1.Remove(0, foundS1 + 1);
    string[] cut3 = StrIncomingSerial1.Split('W');
    PotenGHogar = Convert.ToDouble(cut3[0]);

    CostoHogar1mes = ((PotenGHogar / 1000) * 730) * 0.5343;
    Tbx_PHogar.Text = "" + PotenGHogar;

    if (PotenGHogar > 1870.2)
    {
        ia.habla_asistente.SpeakAsync("Alerta Corto Circuito, si el problema
perciste Rebice su conexionado Hogar.");
    }
  
```

Elaboración propia.

#### 4.5.3.2 DETECCIÓN DE FUGA ELÉCTRICA A TIERRA.

Las causas de una fuga eléctrica a tierra, puede ser la pérdida de aislamiento en los conductores en uno o más puntos del conexionado de un hogar, cuando este cable hace contacto con alguna estructura de la edificación ocasiona un flujo de corriente eléctrica hacia tierra que se considera pérdida energética; sabiendo este concepto podemos determinar una fuga a tierra de la siguiente manera:

- Desconectar cada artefacto electrónico del hogar.
- Apagar todas las luces del hogar.
- Luego de cumplirse estas condiciones la corriente en el circuito deberá ser cero, en caso de que no es una muy probable fuga a tierra.

Con estas bases podemos determinar si hay una fuga eléctrica más precisa gracias a que contamos con sensores de intensidad en cada circuito específico del hogar esto nos permitirá encontrar la falla con mayor precisión, mediante la implementación de un código de programación se simulará este proceso en conjunto con el asistente virtual.

**Gráfico N° 3: Parte del código de simulación de fuga eléctrica a tierra (Anexo 9).**

```

if (Btn_Prueba_de_Fuga.Content.Equals("Terminar"))
{
MessageBox.Show("Precione Aceptar para continuar con la Prueba");
if (PotenGHogar <4)
{
ia.habla_asistente.Speak("Se ha completado el analisis no existe fuga a tierra en el circuito Hogar");
}
else if (PotenGHogar > 2)
{
ia.habla_asistente.Speak("Se ha completado el análisis, existe fuga Electrica de" +
"" + PotenGHogar + "Watts en el circuito Hogar, Se recomienda Revisar los conductores del
circuito. " +
"Si la fuga continúa El costo mensual del circuito hogar aproximado sera de ..." +
"" + (CostoHogar1mes,0) + "Soles");
}
}

```

**Elaboración propia.**

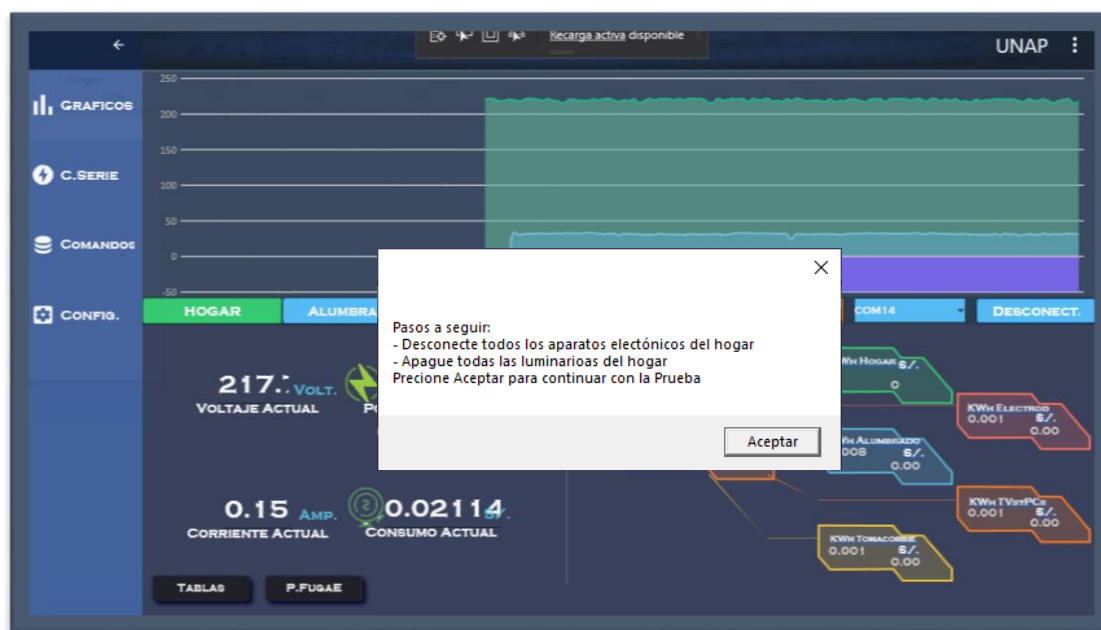
Este código es muy importante ya que como se había visto antes el porcentaje de reclamos en cuanto a excesivo consumo es uno de los problemas más grandes puestos en reclamo; de los cuales la gran mayoría se debe a un mal conexionado del sistema eléctrico del hogar, en este caso fuga eléctrica a tierra.

- **SIMULANDO UNA FUGA ELÉCTRICA A TIERRA.**

Se puso una carga de aproximadamente 0.15 Amp simulando una fuga eléctrica en el circuito de alumbrados; para lo cual se espera que la alerta de fuga que el asistente nos debe dar y decir es solamente en ella ya que los demás circuitos la carga será 0.

- ✓ **Paso 1:** Con el comando “Hanabi, Iniciar prueba de fuga eléctrica”, el asistente virtual dará inicio a la prueba, sugiriéndonos mediante una voz de computador lo siguiente: “A ingresado al modo Prueba de fuga eléctrica o fuga a tierra: siga los siguientes pasos”

Figura 4.77: Datos de prueba de fuga a tierra.



Elaboración propia.

Como podemos observar en la figura tenemos una carga que oscila entre 0.14 y 0.15 Amperios, valor premeditadamente puesto para la prueba de fuga a tierra, podemos ver que la los datos que el asistente de voz no dio son los siguientes:

Tabla N° 4.8: Resultados de prueba de fuga eléctrica a tierra.

	EXISTE FUGA ELECTRICA A TIERRA	POTENCIA DE LA FUGA ELÉCTRICA	COSTO MENSUAL QUE TENDRA ESA FUGA ELÉCTRICA
CIRCUITO HOGAR	NO	0	0
CIRCUITO DE ALMBRADO	SI	32.43 Watt	13.0206 soles aproximadamente
CIRCUITO DE TOMACORRIENTES	NO	0	0
CIRCUITO DE ELECTRODOMÉSTICOS	NO	0	0

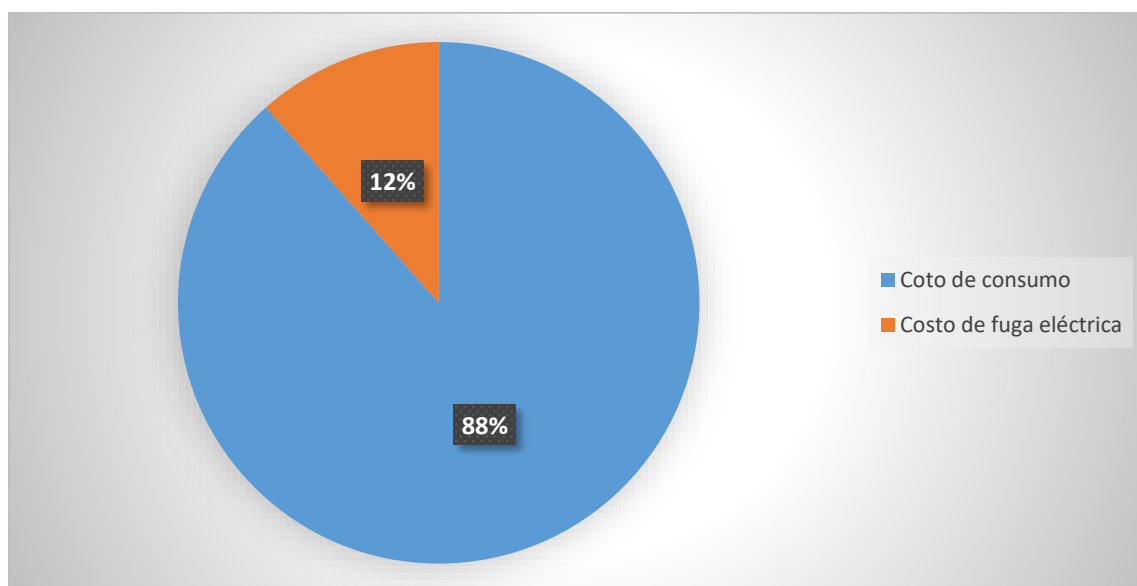
<b>CIRCUITO TV Y PCS</b>	<b>NO</b>	<b>0</b>	<b>0</b>
<b>DIFERENCIA ANTES Y DESPUES</b>	<b>SIN FUGA ELÉCTRICA A TIERRA</b>	<b>CON FUGA ELÉCTRICA A TIERRA</b>	<b>DIFERENCIA</b>
	<b>Potencia de uso (300 amp)</b>	<b>Potencia de uso + 0.15 amp (300.15 amp)</b>	Lectura de diferencia (0,15 amp)

Elaboración propia.

Con esto comprobamos que el sistema funciona correctamente calculándonos la potencia de la fuga, así como también el costo aproximado que tendrá en un mes en caso de que este persista.

Es decir, si en el mes con sumimos traducido a soles 100 soles de consumo eléctrico a este se adicionará la carga de fuga eléctrica que es de 13.0206 soles. Gracias al sistema podemos saber que la falla se encuentra en el circuito de alumbrados y reparar la falla mucho más rápidamente.

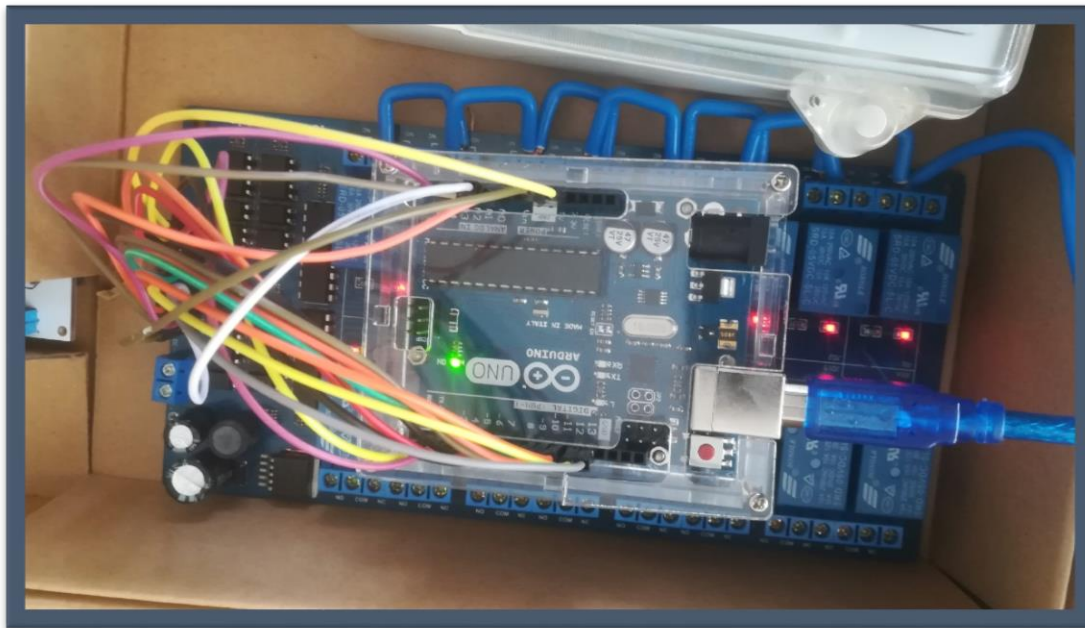
En el siguiente grafico podemos observar que la fuga eléctrica representa el 12% del total que se pagaría por consumo en el mes; es decir que gracias al sistema de detección podremos ahorrar de manera eficiente un 12% al reparar la fuga eléctrica a tierra.

**Gráfico N° 4: Representación gráfica del consumo electro y fuga a tierra.**

Elaboración propia.

#### ***4.5.3.3. CONEXIÓN MÓDULO RELÉ CON ARDUINO UNO.***

Para el control del alumbrado del hogar se utilizará un módulo relé de 16 canales, lo que nos permitirá controlar el alumbrado u otras cosas necesarias por comandos de voz; para ello se escribió un código en Arduino IDE; el módulo relé utilizara conectores para poder interactuar con los pines (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, A0, A1, A2, A3) de Arduino UNO.

**Figura 4.78: Conexión módulo relé con Arduino UNO.****Elaboración propia.**

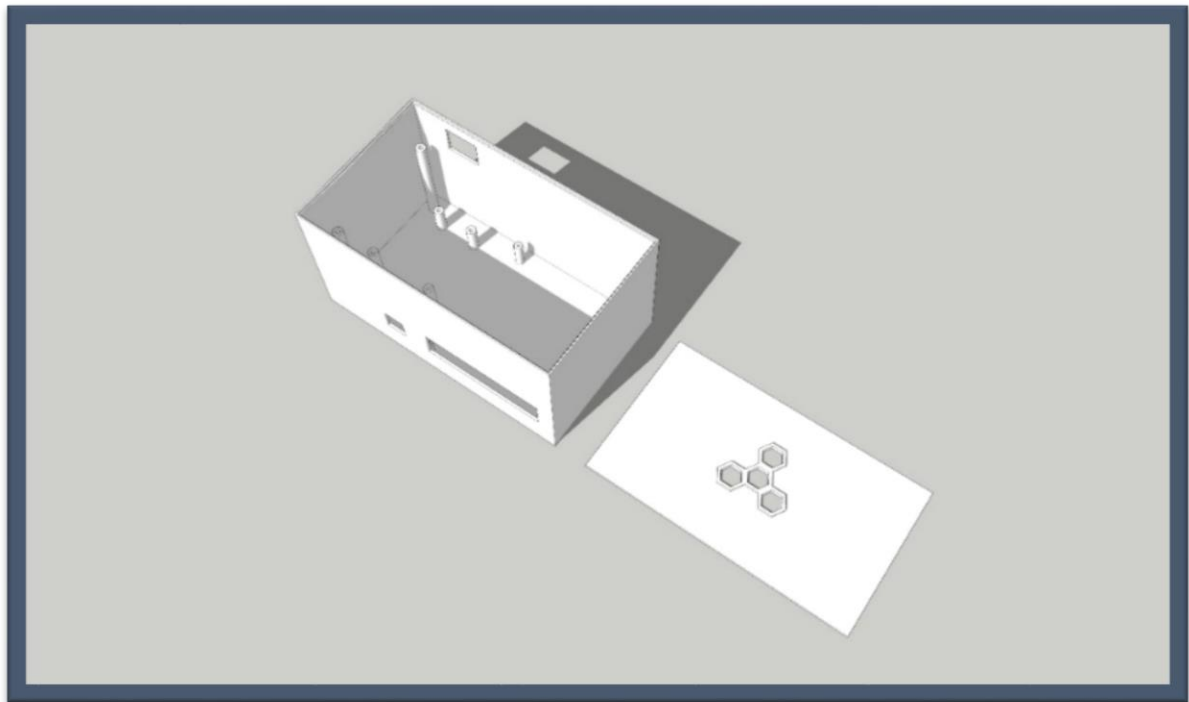
Con los problemas de fuga eléctrica a tierra solucionados pasamos a otro aspecto que también tiene relevancia y que puede ser usado de manera inteligente; al tener el asistente de voz ya programado podemos usarlo para automatizar las luminarias del hogar de esta manera, gracias al módulo relé y Arduino UNO prender y apagar las luces por comandos de voz; esto nos permitirá apagar las luces desde el ordenador es decir; si quisiera apagar las luces de una habitación o ambiente que este lejos de donde estoy; puedo hacer mediante un comando de voz sin moverme del sitio; de esta manera no solo ahorramos un 12% que se ha demostrado en la simulación, si no un porcentaje que contribuye al ahorro de energía eléctrica a su vez la economía de las familias u hogares.

***4.5.3.4. ELABORACIÓN DE CAJA PARA LOS COMPONENTES.***

Para almacenar todos los componentes y circuitos de modo seguro, se ha diseñado en 3d una caja que de manera precisa nos permite poner tornillos a los circuitos de esta manera todo el sistema estará mejor ordenado y más seguro.



**Figura 4.79: Caja 3d para componentes.**



**Elaboración propia.**

## CAPÍTULO V

### CONCLUSIONES

#### 5.1 CONCLUSIÓN 1.

En primera instancia concluimos que gracias a que podemos visualizar datos como la intensidad, voltaje, potencia, consumo KWh, y consumo traducido en Soles; todo ello en tiempo real, mejora la experiencia de usuario y hace que el consumo que se está teniendo en el momento sea mucho más consiente. El manejo de este interfaz y asistente virtual facilita mucho el procedimiento que se hubiera hecho antes para saber tu consumo en tiempo real, ahora se puede desde tu computador personal.

#### 5.2 CONCLUSIÓN 2.

Habiendo ya hablado del manejo e interfaz creado anteriormente; en cuanto a las funciones que conlleva este, como la detección de Fuga a tierra o la detección de un corto circuito, la simulación realizada tuvo mucho éxito siendo este el de mayor resultado en cuanto al ahorro energético gracias a su detección de prueba de fuga eléctrica.

#### 5.3 CONCLUSIÓN 3.

Como una conclusión final, los objetivos establecidos al inicio del proyecto se cumplieron en su totalidad y la hipótesis que se tenía se comprobó satisfactoriamente; ya que, en la simulación de prueba de fuga eléctrica a tierra con una carga controlada, se logra un ahorro de 12%, este puede variar según la potencia de dicha fuga. Además, gracias al interfaz, y al módulo relé que permite controlar las luces, se usa con mayor eficiencia la energía eléctrica, abaratando costos en facturas de energía eléctrica y haciendo el uso más consiente por parte de un usuario.

## CAPÍTULO VI

### RECOMENDACIONES

Las recomendaciones para futuros proyectos que tengan temática parecida a este tema o para implementar el sistema en un hogar. Se recomienda usar sensores SCT-013 de acuerdo al nivel promedio de amperaje en el hogar o circuito más específicamente; con el fin de tener mayor precisión al medir, es decir por ejemplo si el circuito alumbrado solo llega a 5 amperios entonces se debe elegir un sensor en ese rango para que la precisión sea mayor.

En cuanto al asistente virtual e interfaz de usuario Hanabifinal se recomienda introducir mayores funcionalidades para el hogar tales como sensores de temperatura. En caso el hogar tenga termostato hacer una conexión con HanabiFinal, en pocas palabras añadir más funciones domóticas del hogar.

## CAPÍTULO VII

### REFERENCIAS

- Aprendiendoarduino.wordpress.com. (2016). Qué es Arduino | Aprendiendo Arduino. Recuperado 26 de diciembre de 2019, de <https://aprendiendoarduino.wordpress.com/2016/12/11/que-es-arduino-2/>
- Circuito impreso - Wikipedia. (2019). Recuperado 26 de diciembre de 2019, de [https://es.wikipedia.org/wiki/Circuito\\_impreso](https://es.wikipedia.org/wiki/Circuito_impreso)
- Cómo funciona la protoboard | Atlantis. (2017). Recuperado 26 de diciembre de 2019, de <https://edutech.atlantistelecom.com/post/como-funciona-la-protoboard-166.php>
- Concepto definicion.de. (2019). ¿Qué es Domótica? » Su Definición y Significado. Recuperado 26 de diciembre de 2019, de <https://concepto definicion.de/domotica/>
- Efecto Hall - EcuRed. (2019). Recuperado 26 de diciembre de 2019, de [https://www.ecured.cu/Efecto\\_Hall](https://www.ecured.cu/Efecto_Hall)
- ElvnculoDigitalTV. (2019). curso de programación de El Vínculo Digital TV. Recuperado 26 de diciembre de 2019, de <https://www.elvinculodigital.com/cienciatecnologia/2019/sara-2-0-el-nuevo-curso-de-programacion-de-el-vinculo-digital-tv/>
- Fabricacion de Circuitos Impresos (PCB) | Video Rockola. (2019). Recuperado 26 de diciembre de 2019, de <http://www.videorockola.com/tutoriales/fabricacion-de-circuitos-impresos-pcb-2/>
- Fluke, C. (2019a). ¿Qué Es Un Multímetro Digital? | Fluke. Recuperado 26 de diciembre de 2019, de <https://www.fluke.com/es-mx/informacion/mejores-practicas/aspectos-basicos-de-las-mediciones/electricidad/que-es-un-multimetro-digital>
- Fluke, C. (2019b). ¿Qué Es Una Pinza Amperimétrica? | Fluke. Recuperado 26 de diciembre de 2019, de <https://www.fluke.com/es-ar/informacion/mejores-practicas/aspectos-basicos-de-las-herramientas-de-prueba/pinzas-amperimetricas/que-es-una-pinza-amperimetrica>
- Hector, T. (2017). Amplificador Operacional - qué es y sus configuraciones - HeTPro/Tutorial. Recuperado 26 de diciembre de 2019, de <https://hetpro-store.com/TUTORIALES/amplificador-operacional/>
- Leandro, K. (2007). Disección de un medidor electrónico trifásico. Recuperado 26 de diciembre de 2019, de <http://www.afinidadelectrica.com/articulo.php?IdArticulo=188>
- Luis, L. (2017). Sensor de corriente eléctrica no invasivo con Arduino y SCT-013. Recuperado 26 de diciembre de 2019, de <https://www.luisllamas.es/arduino-sensor-corriente-sct-013/>
- MicroJPM. (2019). Placa Cobre Perforada. Recuperado 26 de diciembre de 2019, de <http://m.microjpm.com/products/placa-de-cobre-perforada-70mm-x-95mm/>
- Multimetrodigital.net. (2019). ▷ ¿QUE ES UN MULTÍMETRO? ► (COMO SE USA). Recuperado 26 de diciembre de 2019, de <https://multimetrodigital.net/que-es-un-multimetro/>
- Naylamp. (2019). Transformador de voltaje AC - ZMPT101B - Naylamp Mechatronics - Perú. Recuperado 26 de diciembre de 2019, de <https://naylampmechatronics.com/sensores-corriente-voltaje/393-transformador-de-voltaje-ac-zmpt101b.html>
- Oscar, O. (2017). Arduino «Tipos de Placas». Recuperado 26 de diciembre de 2019, de

<https://roboticclassroom.blogspot.com/2017/04/arduino-tipos-de-placas.html>

- OSINERGMIN. (2019). Pliego Tarifario Máximo del Servicio Público de Electricidad. Recuperado 26 de diciembre de 2019, de <https://www.osinergmin.gob.pe/Tarifas/Electricidad/PliegosTarifariosUsuarioFinal.aspx?Id=210000>
- PCBWay. (2019). Proceso de fabricación de PCB y equipos. Recuperado 26 de diciembre de 2019, de <https://www.pcbway.es/pcb-service.html>
- PowerUc. (2019). Transformador de corriente de núcleo dividido SCT013 entrada nominal 5A 10A 15A 20A 25A 3 - PowerUC. Recuperado 26 de diciembre de 2019, de <https://www.poweruc.pl/collections/split-core-current-transformers2/products/split-core-current-transformer-sct013-rated-input-5a-100a?variant=6876754903084>
- Raffino, M. E. (2019). «Energía eléctrica». Recuperado 26 de diciembre de 2019, de 16 de enero de 2019 website: <https://concepto.de/energia-electrica/>
- Raihaan Kamarudin, M., & Aiman Md Yusof, M. F. (2013). Low Cost Smart Home Automation via Microsoft Speech Recognition. *International Journal of Engineering & Computer Science*. Recuperado de <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.657.3480&rep=rep1&type=pdf>
- Ramya, R., & Poomurugan, R. (2015). ENERGY MANAGEMENT SYSTEM BASED ON WIRELESS SENSOR NETWORKS IN INTELLIGENT BUILDINGS. *International Journal of modern trends in engineering and research*, 02(12). Recuperado de <https://pdfs.semanticscholar.org/1ff2/c10dd7797800a62fb81c7eca51422ee0ee94.pdf>
- Ríos, F. (s. f.). *Conceptos Generales de Redes Eléctricas*. Recuperado de <https://www.monografias.com/trabajos-pdf4/redes-electricas/redes-electricas.pdf>
- Robert, W. (2014). *Yhdc SCT-013-000 Current Transformer A Report on the properties of the Yhdc Current Transformer and its suitability for use with the OpenEnergy monitor*. Recuperado de <https://learn.openenergymonitor.org/electricity-monitoring/ct-sensors/files/YhdcCTReportIss6.pdf>
- SAMANIEGO IDROVO, D. I., & VELESACA ORELLANA, D. F. (2016). *DISEÑO E IMPLEMENTACIÓN DE UN MEDIDOR DE ENERGÍA ELECTRÓNICO PARA VIVIENDA, CON ORIENTACIÓN A LA PREVENCIÓN DE CONSUMO Y AHORRO ENERGÉTICO*. (UNIVERSIDAD POLITÉCNICA SALESIANA SEDE CUENCA). Recuperado de <https://dspace.ups.edu.ec/bitstream/123456789/12893/1/UPS-CT006717.pdf>
- SmartGridsInfo. (2019). Smart Grid • SMARTGRIDSINFO. Recuperado 26 de diciembre de 2019, de <https://www.smartgridsinfo.es/smart-grid>
- SurtrTech. (2019). Easy measure of AC Voltage using Arduino and ZMPT101B – SURTR TECHNOLOGY. Recuperado 26 de diciembre de 2019, de <https://surtrtech.com/2019/01/21/easy-measure-of-ac-voltage-using-arduino-and-zmpt101b/>
- Texas, I. (2019). *Industry-Standard Dual Operational Amplifiers*. Recuperado de <https://www.ti.com/lit/ds/symlink/lm258a.pdf>
- Tutorial sensor de corriente AC no invasivo SCT-013. (2016). Recuperado 26 de diciembre de 2019, de [https://naylampmechatronics.com/blog/51\\_tutorial-sensor-de-corriente-ac-no-invasivo-s.html](https://naylampmechatronics.com/blog/51_tutorial-sensor-de-corriente-ac-no-invasivo-s.html)

ZEGARRA PINTO, M. A. (2017). *Análisis de nuevo sistema de medición centralizada de energía eléctrica con medidores inteligentes en área de la Región Arequipa* (UNSA). Recuperado de <http://repositorio.unsa.edu.pe/bitstream/handle/UNSA/5097/ELzepima.pdf?sequence=1&isAllowed=y>

Zeming-e. (2013). *ZMPT101B(ZMPT107) voltage transformer operating guide 1, Wiring diagram*. Recuperado de [http://www.zeming-e.com/file/0\\_2013\\_10\\_18\\_093344.pdf](http://www.zeming-e.com/file/0_2013_10_18_093344.pdf)

*ZMPT107 potential transformer*. (2019). Recuperado de <https://p.globalsources.com/IMAGES/PDT/SPEC/210/K1051796210.pdf>

## ANEXOS

### ANEXO 1 Código del 1er Prototipo Arduino Uno (Arduino IDE):

```
#include "EmonLib.h"           //
Include Emon Library          void loop()
EnergyMonitor emon;           // {
Create an instance
void setup()
{
  Serial.begin(9600);
  emon.voltage(4, 554.26, 1.7); //
Voltage: input pin, calibration,
phase_shift
  emon.current(0, 1.21);       //
Current: input pin, calibration. }
}
```

### ANEXO 2 Código del 2do prototipo Arduino Uno en (Arduino IDE).

```
#include "EmonLib.h"           // float V = emon.Vrms;
Include Emon Library          float C1 = emon.Irms;
EnergyMonitor emon;           // float C2 = emon1.Irms;
Create an instance            float C3 = emon2.Irms;
EnergyMonitor emon1;          float C4 = emon3.Irms;
EnergyMonitor emon2;          float C5 = emon4.Irms;
EnergyMonitor emon3;          Serial.print("Voltaje: ");
EnergyMonitor emon4;          Serial.print(V);
                               Serial.print(" ");
void setup()                  Serial.print("Irms 1: ");
{                               Serial.print(C1);
  Serial.begin(9600);          Serial.print(" ");
  emon.voltage(4, 554.26, 1.7); // Serial.print("Irms 2: ");
Voltage: input pin, calibration, Serial.print(C2);
phase_shift                   Serial.print(" ");
  emon.current(0, 1.21);       // Serial.print("Irms 3: ");
Current: input pin, calibration. Serial.print(C3);
  emon1.current(1, 1.21);      Serial.print(" ");
  emon2.current(2, 1.21);      Serial.print("Irms 4: ");
  emon3.current(3, 1.21);      Serial.print(C4);
  emon4.current(5, 1.21);      Serial.print(" ");
}                               Serial.print("Irms 5: ");
                               Serial.println(C5);
void loop()                   //delay(500);
{                               }
  emon.calcVI(20,2000);
```

### ANEXO 3 Código del 3er prototipo Arduino Due en (Arduino IDE).

```
#include "EmonLib.h"           // Include EnergyMonitor emon3;
Emon Library                   EnergyMonitor emon4;
EnergyMonitor emon;           // Create an
instance                         void setup()
EnergyMonitor emon1;           {
EnergyMonitor emon2;           analogReadResolution(ADC_BITS);
                               Serial.begin(250000);
```



```

    emon.voltage(6, 434.26, 1.7); // Voltage:
input pin, calibration, phase_shift

    emon.current(0, 1.21); // Current:
input pin, calibration.

    emon1.current(1,1651.1);

    emon2.current(2,311.1);

    emon3.current(3,231.1);

    emon4.current(4,291.1);
}

void loop()
{
    emon.calcVI(20,2000); // Calculate
all. No.of wavelengths, time-out

    double Irms2 = emon1.calcIrms(1480);

    double Irms3 = emon2.calcIrms(1480);

    double Irms4 = emon3.calcIrms(1480);

    double Irms5 = emon4.calcIrms(1480);

    float V = emon.Vrms;

    float C1 = emon.Irms;

    float C2 = emon1.Irms;

    float C3 = emon2.Irms;

    float C4 = emon3.Irms;

    float C5 = emon4.Irms;

    Serial.print("T");

    Serial.println(V);

    Serial.print("A");

    Serial.println(C1);

    Serial.print("B");

    Serial.println(C2);

    Serial.print("C");

    Serial.println(C3);

    Serial.print("D");

    Serial.println(C4);

    Serial.print("E");

    Serial.println(C5);

    Serial.print("V");

    Serial.println(C1 * V);

    Serial.print("W");

    Serial.println(C2 * V);

    Serial.print("X");

    Serial.println(C3 * V);

    Serial.print("Y");

    Serial.println(C4 * V);

    Serial.print("Z");

    Serial.println(C5 * V);

    Serial.println("\r");

    delay(500);

}

```

**ANEXO 4 Código de prototipo final Arduino Due en (Arduino IDE).**

```

#include "EmonLib.h" // Include
Emon Library
EnergyMonitor emon; // Create an
instance
EnergyMonitor emon1;
EnergyMonitor emon2;

EnergyMonitor emon3;
EnergyMonitor emon4;
unsigned long startMillis;
unsigned long endMillis;
double KilosHogar = 0;
double KilosAlumbrado = 0;

```

```

double KiloTomacorriente = 0;

double KiloElectrodomesticos = 0;

double KiloTVsyPCs = 0;

void setup()
{

  analogReadResolution(ADC_BITS);

  Serial.begin(115200);

  emon.voltage(6,638.5, 1.7); // Voltage:
  input pin, calibration, phase_shift

  emon.current(0,1.40); // Current:
  input pin, calibration.

  emon1.current(1,63.5);

  emon2.current(2,105.5);

  emon3.current(3,68.5);

  emon4.current(4,60.7);

}

void loop()
{
  emon.calcVI(20,2000); // Calculate
  all. No.of wavelengths, time-out

  //emon.calclrms(1480);

  emon1.calclrms(1380);

  emon2.calclrms(1180);

  emon3.calclrms(1380);

  emon4.calclrms(1040);

  float V = emon.Vrms;

  float C1 = emon.Irms;

  float C2 = emon1.Irms;

  float C3 = emon2.Irms;

  float C4 = emon3.Irms;

  float C5 = emon4.Irms;

  endMillis = millis();

  unsigned long time = endMillis -
  startMillis;

  KiloHogar = KiloHogar + ((double)(V *
  C1) * ((double)time/60/60/1000000));

  KiloAlumbrado = KiloAlumbrado +
  ((double)(V * C2) *
  ((double)time/60/60/1000000)); //Calculate
  kilowatt hours used

  KiloTomacorriente = KiloTomacorriente
  + ((double)(V * C3) *
  ((double)time/60/60/1000000));

  KiloElectrodomesticos =
  KiloElectrodomesticos + ((double)(V * C4)
  * ((double)time/60/60/1000000));

  KiloTVsyPCs = KiloTVsyPCs + ((double)(V
  * C5) * ((double)time/60/60/1000000));

  startMillis = millis();

  Serial.print("T");

  Serial.print(V);

  Serial.print("A");

  Serial.print(C1);

  Serial.print("B");

  Serial.print(C2);

  Serial.print("C");

  Serial.print(C3);

  Serial.print("D");

  Serial.print(C4);

  Serial.print("E");

```

```

Serial.print(C5);
Serial.print("V");
Serial.print(C1 * V);
Serial.print("W");
Serial.print(C2 * V);
Serial.print("X");
Serial.print(C3 * V);
Serial.print("Y");
Serial.print(C4 * V);
Serial.print("Z");
Serial.print(C5 * V);
Serial.print("F");

Serial.print(KilosHogar, 3);

Serial.print("G");
Serial.print(KilosAlumbrado, 3);
Serial.print("H");
Serial.print(KilosTomacorriente, 3);
Serial.print("I");
Serial.print(KilosElectrodomesticos, 3);
Serial.print("J");
Serial.print(KilosTVsyPCs, 3);
Serial.println("\r");

//delay(400);
}
    
```

### ANEXO 5 Código de interfaz principal de HanabiFinal en (Xamarin Forms).

```

<Window x:Name="window"
x:Class="HanabiFinal.MainWindow"
xmlns="http://schemas.microsoft.com/w
infx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com
/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com
/expression/blend/2008"
xmlns:mc="http://schemas.openxmlforma
ts.org/markup-compatibility/2006"
xmlns:local="clr-
namespace:HanabiFinal"
mc:Ignorable="d"
Title="InterfazPrincipal"
Height="180.035" Width="183.339"
AllowsTransparency="True"
WindowStyle="None"
Background="{x:Null}"
Foreground="{x:Null}"
HorizontalAlignment="Center"
HorizontalContentAlignment="Center"
VerticalAlignment="Top"
VerticalContentAlignment="Top"
Visibility="Visible"
ResizeMode="NoResize"
ShowInTaskbar="False"
MouseLeftButtonDown="Window_MouseLeft
ButtonDown"
Activated="Window_Activated"
Closing="Window_Closing">
    <Window.Resources>
        <Style x:Key="btnComando"
TargetType="{x:Type Button}">
            <Setter
Property="Template">
                <Setter.Value>
                    <ControlTemplate
TargetType="{x:Type Button}">
                        <Grid>
                            <VisualStateManager.VisualStateGroups
                            >
                                <VisualStateGroup
x:Name="CommonStates">
                                    <VisualState x:Name="Normal"/>
                                    <VisualState x:Name="MouseOver">
                                        <Storyboard>
                                            <ObjectAnimationUsingKeyFrames
Storyboard.TargetProperty="(Shape.Fil
l)"
Storyboard.TargetName="rectangle">
                                                <DiscreteObjectKeyFrame KeyTime="0">
                                                    <DiscreteObjectKeyFrame.Value>
                                                        <SolidColorBrush Color="#FF101084"/>
                                                    </DiscreteObjectKeyFrame.Value>
                                                </DiscreteObjectKeyFrame>
                                            </Storyboard>
                                        </VisualState>
                                </VisualStateGroup>
                            </Grid>
                        </ControlTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
    </Window.Resources>
    <Content>
        <Button Content="Comando" Style="{StaticResource btnComando}">
    </Content>
</Window>
    
```

```

</ObjectAnimationUsingKeyFrames>
</Storyboard>
</VisualState>
<VisualState x:Name="Pressed"/>
<VisualState x:Name="Disabled"/>
</VisualStateGroup>
</VisualStateManager.VisualStateGroup
s>

<Rectangle x:Name="rectangle"
RadiusY="10" RadiusX="10"
Stroke="#303F9F"/>

<ContentPresenter
HorizontalAlignment="{TemplateBinding
HorizontalContentAlignment}"
RecognizesAccessKey="True"
SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"
VerticalAlignment="{TemplateBinding
VerticalContentAlignment}"/>
</Grid>

<ControlTemplate.Triggers>
    <Trigger
Property="IsFocused" Value="True"/>
    <Trigger
Property="IsDefaulted" Value="True"/>
    <Trigger
Property="IsMouseOver" Value="True"/>
    <Trigger
Property="IsPressed" Value="True"/>
    <Trigger
Property="IsEnabled" Value="False"/>
</ControlTemplate.Triggers>
</ControlTemplate>
    </Setter.Value>
</Setter>
</Style>
<Style x:Key="btnMedio"
TargetType="Button">
    <Setter
Property="Template">
        <Setter.Value>
            <ControlTemplate
TargetType="{x:Type Button}">
                <Grid>

<VisualStateManager.VisualStateGroups
>

<VisualStateGroup
x:Name="CommonStates">

<VisualState x:Name="Normal"/>

<VisualState x:Name="MouseOver">

<Storyboard>

<ObjectAnimationUsingKeyFrames
Storyboard.TargetProperty="(Shape.Fil
l)"
Storyboard.TargetName="rectangle1">

<DiscreteObjectKeyFrame KeyTime="0">

<DiscreteObjectKeyFrame.Value>

<SolidColorBrush Color="#FF101084"/>

</DiscreteObjectKeyFrame.Value>

</DiscreteObjectKeyFrame>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</VisualState>

<VisualState x:Name="Pressed"/>

<VisualState x:Name="Disabled"/>

</VisualStateGroup>

</VisualStateManager.VisualStateGroup
s>

<Rectangle x:Name="rectangle1"/>

<ContentPresenter
HorizontalAlignment="{TemplateBinding
HorizontalContentAlignment}"
RecognizesAccessKey="True"
SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"
VerticalAlignment="{TemplateBinding
VerticalContentAlignment}"
Margin="5.911,0.768,5.911,2.271"/>
</Grid>

<ControlTemplate.Triggers>
    <Trigger
Property="IsFocused" Value="True"/>
    <Trigger
Property="IsDefaulted" Value="True"/>
    <Trigger
Property="IsMouseOver" Value="True"/>
    <Trigger
Property="IsPressed" Value="True"/>
    <Trigger
Property="IsEnabled" Value="False"/>

```

```

</ControlTemplate.Triggers>

</ControlTemplate>
    </Setter.Value>
    </Setter>
</Style>
    <Storyboard x:Key="Inicio">

<DoubleAnimationUsingKeyFrames
Storyboard.TargetProperty="(UIElement
.Opaicity)"
Storyboard.TargetName="window">
    <EasingDoubleKeyFrame
KeyTime="0" Value="0"/>
    <EasingDoubleKeyFrame
KeyTime="0:0:1" Value="1"/>
</DoubleAnimationUsingKeyFrames>
</Storyboard>
    <Style x:Key="BtnLados"
TargetType="{x:Type Button}">
    <Setter
Property="Template">
        <Setter.Value>
            <ControlTemplate
TargetType="{x:Type Button}">
                <Grid>

<VisualStateManager.VisualStateGroups
>

<VisualStateGroup
x:Name="CommonStates">

<VisualState x:Name="Normal"/>

<VisualState x:Name="MouseOver">

<Storyboard>

<DoubleAnimationUsingKeyFrames
Storyboard.TargetProperty="(Shape.Str
okeThickness)"
Storyboard.TargetName="path">

<EasingDoubleKeyFrame KeyTime="0"
Value="0"/>

</DoubleAnimationUsingKeyFrames>

<DoubleAnimationUsingKeyFrames
Storyboard.TargetProperty="(UIElement
.RenderTransform).(TransformGroup.Chi
ldren)[3].(TranslateTransform.X)"
Storyboard.TargetName="path">

<EasingDoubleKeyFrame KeyTime="0"
Value="-0.167"/>

</DoubleAnimationUsingKeyFrames>

<DoubleAnimationUsingKeyFrames
Storyboard.TargetProperty="(UIElement
.RenderTransform).(TransformGroup.Chi
ldren)[3].(TranslateTransform.Y)"
Storyboard.TargetName="path">

<EasingDoubleKeyFrame KeyTime="0"
Value="0"/>

</DoubleAnimationUsingKeyFrames>

<ObjectAnimationUsingKeyFrames
Storyboard.TargetProperty="(Shape.Fil
l)" Storyboard.TargetName="path">

<DiscreteObjectKeyFrame KeyTime="0">

<DiscreteObjectKeyFrame.Value>

<SolidColorBrush Color="#FF101084"/>

</DiscreteObjectKeyFrame.Value>

</DiscreteObjectKeyFrame>

</ObjectAnimationUsingKeyFrames>

</Storyboard>

</VisualState>

<VisualState x:Name="Pressed"/>

<VisualState x:Name="Disabled"/>

</VisualStateGroup>

</VisualStateManager.VisualStateGroup
s>

                <Path
x:Name="path" Data="M343.35417,6.563
L331.10838,6.5572963
330.25032,6.5937558
329.50031,6.708045
328.66698,6.9686349
328.14582,7.2081117
327.78105,7.4063467
327.34381,7.6410173
327.01543,7.8593643
326.78068,8.0316392
326.54654,8.2194462
326.34328,8.3754717
326.13991,8.5476417
325.92075,8.7663169
325.6706,9.0315861
325.42045,9.3280582
325.12341,9.6719201
324.81088,10.093644
324.49873,10.578546
324.26432,11.016215
324.09195,11.453476

```

```

323.90418,11.938006
323.73253,12.531239
323.67041,12.859617&#xD;&#xA;323.6077
8,13.203767 323.54566,13.610178
323.51428,14.048115
323.49828,14.204259
323.49828,14.938863
323.51392,17.969862
323.52954,18.313383
323.57642,18.688923
323.65456,19.142354
323.76394,19.501772
323.84207,19.798684
323.96709,20.189228
324.12335,20.596039
324.31087,21.002093
324.54526,21.455278
324.84217,21.923701
325.24846,22.439252
325.63912,22.877308
326.06103,23.299238
326.43607,23.611651
326.90486,24.017569
327.43616,24.345345
328.02997,24.657615&#xD;&#xA;328.7019
,24.970389 329.35822,25.157905
330.10829,25.361298
330.7646,25.439057 331.3584,25.45518
343.37516,25.439562 z"
Fill="{x:Null}" Stretch="Fill"
Stroke="{x:Null}" Width="Auto"
Height="19" Margin="1.333,1,0,1"
RenderTransformOrigin="0.5,0.5">

<Path.RenderTransform>

<TransformGroup>

<ScaleTransform/>

<SkewTransform/>

<RotateTransform/>

<TranslateTransform/>

</TransformGroup>

</Path.RenderTransform>
</Path>

<ContentPresenter
HorizontalAlignment="{TemplateBinding
HorizontalContentAlignment}"
RecognizesAccessKey="True"
SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"
VerticalAlignment="{TemplateBinding
VerticalContentAlignment}"
Margin="8.938,1.644,4.132,3.395"/>
</Grid>
    
```

```

<ControlTemplate.Triggers>
    <Trigger
Property="IsFocused" Value="True"/>
    <Trigger
Property="IsDefaulted" Value="True"/>
    <Trigger
Property="IsMouseOver" Value="True"/>
    <Trigger
Property="IsPressed" Value="True"/>
    <Trigger
Property="IsEnabled" Value="False"/>
</ControlTemplate.Triggers>

</ControlTemplate>
    </Setter.Value>
</Setter>
    <Setter
Property="BorderBrush"
Value="{x:Null}"/>
    <Setter
Property="Foreground"
Value="{x:Null}"/>
    </Style>
    <Style x:Key="BtnLados2"
TargetType="{x:Type Button}">
    <Setter
Property="Template">
    <Setter.Value>
    <ControlTemplate
TargetType="{x:Type Button}">
    <Grid>

<VisualStateManager.VisualStateGroups
>

<VisualStateGroup
x:Name="CommonStates">

<VisualState x:Name="Normal"/>

<VisualState x:Name="MouseOver">

<Storyboard>

<DoubleAnimationUsingKeyFrames
Storyboard.TargetProperty="(Shape.Str
okeThickness)"
Storyboard.TargetName="path">

<EasingDoubleKeyFrame KeyTime="0"
Value="0"/>

</DoubleAnimationUsingKeyFrames>

<DoubleAnimationUsingKeyFrames
Storyboard.TargetProperty="(UIElement
.RenderTransform).(TransformGroup.Children)[3].(TranslateTransform.X)"
Storyboard.TargetName="path">
    
```

```

<EasingDoubleKeyFrame KeyTime="0"
Value="-0.167"/>
325.12341,9.6719201
324.81088,10.093644
324.49873,10.578546
324.26432,11.016215
324.09195,11.453476
323.90418,11.938006
323.73253,12.531239
323.67041,12.859617&#xA;323.60778,13.
203767 323.54566,13.610178
323.51428,14.048115
323.49828,14.204259
323.49828,14.938863
323.51392,17.969862
323.52954,18.313383
323.57642,18.688923
323.65456,19.142354
323.76394,19.501772
323.84207,19.798684
323.96709,20.189228
324.12335,20.596039
324.31087,21.002093
324.54526,21.455278
324.84217,21.923701
325.24846,22.439252
325.63912,22.877308
326.06103,23.299238
326.43607,23.611651
326.90486,24.017569
327.43616,24.345345
328.02997,24.657615&#xA;328.7019,24.9
70389 329.35822,25.157905
330.10829,25.361298
330.7646,25.439057 331.3584,25.45518
343.37516,25.439562 z"
Fill="{x:Null}" Stretch="Fill"
Stroke="{x:Null}" Width="Auto"
Height="19" Margin="1.333,1,0,1"
RenderTransformOrigin="0.5,0.5">
<Path.RenderTransform>
<TransformGroup>
<ScaleTransform/>
<SkewTransform/>
<RotateTransform/>
<TranslateTransform/>
</TransformGroup>
</Path.RenderTransform>
</Path>
<ContentPresenter
HorizontalAlignment="{TemplateBinding
HorizontalContentAlignment}"
RecognizesAccessKey="True"
SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"

```



```

VerticalAlignment="{TemplateBinding
VerticalContentAlignment}"
Margin="9.476,2.187,4.67,2.852"
Content="S"
RenderTransformOrigin="0.5,0.5">
<ContentPresenter.RenderTransform>
<TransformGroup>
<ScaleTransform ScaleY="1" ScaleX="-
1"/>
<SkewTransform AngleY="0"
AngleX="0"/>
<RotateTransform Angle="0"/>
<TranslateTransform/>
</TransformGroup>
</ContentPresenter.RenderTransform>
</ContentPresenter>
</Grid>
<ControlTemplate.Triggers>
<Trigger
Property="IsFocused" Value="True"/>
<Trigger
Property="IsDefaulted" Value="True"/>
<Trigger
Property="IsMouseOver" Value="True"/>
<Trigger
Property="IsPressed" Value="True"/>
<Trigger
Property="IsEnabled" Value="False"/>
</ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
<Setter
Property="BorderBrush"
Value="{x:Null}"/>
<Setter
Property="Foreground"
Value="{x:Null}"/>
</Style>
<ControlTemplate x:Key="BtnM"
TargetType="{x:Type Button}">
<Grid>
<VisualStateManager.VisualStateGroups
>
<VisualStateGroup
x:Name="CommonStates">
<VisualState
x:Name="Normal"/>
<VisualState
x:Name="MouseOver">
<Storyboard>
<ObjectAnimationUsingKeyFrames
Storyboard.TargetProperty="(Shape.Fil
1)"
Storyboard.TargetName="rectangle2">
<DiscreteObjectKeyFrame KeyTime="0">
<DiscreteObjectKeyFrame.Value>
<SolidColorBrush Color="#FF101084"/>
</DiscreteObjectKeyFrame.Value>
</DiscreteObjectKeyFrame>
</ObjectAnimationUsingKeyFrames>
</Storyboard>
</VisualState>
<VisualState
x:Name="Pressed"/>
<VisualState
x:Name="Disabled"/>
</VisualStateGroup>
</VisualStateManager.VisualStateGroup
s>
<Rectangle
x:Name="rectangle2"/>
<ContentPresenter
HorizontalAlignment="{TemplateBinding
HorizontalContentAlignment}"
RecognizesAccessKey="True"
SnapsToDevicePixels="{TemplateBinding
SnapsToDevicePixels}"
VerticalAlignment="{TemplateBinding
VerticalContentAlignment}"
Margin="4,1,5,2" Width="10"/>
</Grid>
</ControlTemplate>
</Window.Resources>
<Window.Triggers>
<EventTrigger
RoutedEvent="FrameworkElement.Loaded"
>
<BeginStoryboard
Storyboard="{StaticResource
Inicio}"/>
</EventTrigger>
</Window.Triggers>
<Grid>
<Image
VerticalAlignment="Top" Width="182"
HorizontalAlignment="Left"
Height="158" Margin="1,1,-1,0"

```

```

Source="HanabiInterfaz.png"
Stretch="Fill"/>
    <Rectangle x:Name="rec_onoff"
HorizontalAlignment="Left"
Height="21" Margin="1,159,-1,0"
Stroke="#FF303F9F"
VerticalAlignment="Top" Width="182"
RadiusX="10" RadiusY="10">
    <Rectangle.Fill>
        <RadialGradientBrush>
            <GradientStop
Color="#FF073057"/>
            <GradientStop
Color="#FF0C2047" Offset="1"/>
        </RadialGradientBrush>
    </Rectangle.Fill>
</Rectangle>
<ProgressBar
x:Name="pb_Audio" Margin="91,80,90,-
62" Orientation="Vertical"
Foreground="#FF177BBF"
Background="{x:Null}"
RenderTransformOrigin="0.5,0.5"
BorderBrush="{x:Null}" Width="2">
    <ProgressBar.RenderTransform>
        <TransformGroup>
            <ScaleTransform/>
            <SkewTransform/>
            <RotateTransform
Angle="90.15"/>
        </TransformGroup>
    </ProgressBar.RenderTransform>
</ProgressBar>
    <Button
x:Name="Btn_NomAsisstant"
Content="Configuracion"
HorizontalAlignment="Left"
Height="36" Margin="120,104,0,0"
Style="{DynamicResource btnComando}"
VerticalAlignment="Top" Width="38"
BorderBrush="#FF073651"
Foreground="White"
Click="Btn_NomAsisstant_Click"
ToolTip="Configuración"
Background="#FF073651"
BorderThickness="0"
ClipToBounds="True" FontSize="6"/>
        <Path Data="M343.52099,6.5"
Fill="#FFF4F4F5"
HorizontalAlignment="Left" Height="1"
Margin="343.521,6.5,0,0"
Stretch="Fill" Stroke="Black"
VerticalAlignment="Top" Width="1"/>
        <Label x:Name="lbl_pantalla"
Content=""
HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
HorizontalContentAlignment="Left"
Height="21" Margin="1,159,-1,0"
VerticalAlignment="Top" Width="182"
Background="{x:Null}"
BorderBrush="White" Opacity="1.5"
Content="Gráficos" Foreground="White"
HorizontalContentAlignment="Left"
Height="36" Margin="24,104,0,0"
Style="{DynamicResource btnComando}"
VerticalAlignment="Top" Width="38"
FontSize="6" BorderThickness="3"
BorderBrush="#FF073651"
HorizontalContentAlignment="Center"
Click="BtnDashboard_Click"/>
    </Grid>
</Window>

```

**ANEXO 6 Código de Interfaz principal HanabiFinal en (C#).**

```

using HanabiFinal.Properties;
using BibliotecaHanabi;
using System;
using System.ComponentModel;
using System.Diagnostics;
using System.Drawing;
using System.Globalization;
using System.Speech.Recognition;
using System.Speech.Synthesis;
using System.Threading;
using System.Windows;
using System.Windows.Input;
using System.Windows.Media;
using HanabiFinal;
using System.Windows.Controls;

namespace HanabiFinal
{
    public partial class MainWindow : Window
    {
        #region Variables
        EditorCmd editorCmd;
        ControlDeUsuarioSerial serial;
        Ajustes ajustes;
        ComandosSistema csis;
        Asistente ia = new Asistente();
        Gramaticas gramaticas = new
        Gramaticas();
        CargarGramaticas loadgramaticas = new
        CargarGramaticas();
        ControlUsuarioGraficos dashboard = new
        ControlUsuarioGraficos();
        InterfazDash interfazDash = new
        InterfazDash();
        #endregion

        public MainWindow()
        {
            InitializeComponent();
            CargarDatosIniciales();
        }

        #region Métodos
        void CargarDatosIniciales()
    
```

```

    {
        Btn_NomAsisstant.Content =
Settings.Default.NombreAsistente;

ia.habla_asistente.SelectVoice(Settings.Default.
VozDefault);

        ia.habla_asistente.Volume =
Settings.Default.VolumenIA;

        VistaOffStroke();

        DesactivarMicrófono();

        Top =
Settings.Default.PosicionVentanaX;

        Left =
Settings.Default.PosicionVentanaY;

        IdiomaEspañol();

        CargarDatosGramaticas();

Process.GetCurrentProcess().MaxWorkingSet =
Process.GetCurrentProcess().MinWorkingSet;
    }

void CargarDatosGramaticas()

{

    loadgramaticas.SeleccionarComandos();

ia.reconocer_voz_usuario.LoadGrammarAsync(
new
Grammar(loadgramaticas.ComandosInternos()))
; //todo esto tiene la información de los
comandos internos

ia.reconocer_voz_usuario.LoadGrammarAsync(
new
Grammar(loadgramaticas.CargarGramáticasCal
c())); //Carga las gramaticas de la calculadora

ia.reconocer_voz_usuario.LoadGrammarAsync(
new Grammar(new
Choices(loadgramaticas.CargarGramáticasBD()
)); //Carga las gramaticas del usuario

ia.reconocer_voz_usuario.LoadGrammarAsync(
new Grammar(new
Choices(Settings.Default.NombreAsistente)); //
Esto se eliminará en un futuro, pero carga el
nombre del asistente

ia.reconocer_voz_usuario.LoadGrammarAsync(
loadgramaticas.CargarGramáticasWebs()); //hast
a aquí carga todas las gramáticas

ia.reconocer_voz_usuario.RequestRecognizerU
pdate();

```

```

ia.reconocer_voz_usuario.AudioLevelUpdated
+=
Reconocer_voz_usuario_AudioLevelUpdated;

ia.reconocer_voz_usuario.SpeechRecognized
+= Reconocer_voz_usuario_SpeechRecognized;

    ia.habla_asistente.SpeakStarted +=
Habla_asistente_SpeakStarted;

    ia.habla_asistente.SpeakCompleted +=
Habla_asistente_SpeakCompleted;

ia.reconocer_voz_usuario.SetInputToDefaultAudioDevice();

ia.reconocer_voz_usuario.RecognizeAsync(RecognizeMode.Multiple);

    ia.tiempo_desactivar_microfono.Tick +=
Tiempo_desactivar_microfono_Tick;

    ia.liberarRAM.Tick +=
LiberarRAM_Tick;

    ia.liberarRAM.Start();
}

private void LiberarRAM_Tick(object sender, EventArgs e)
{
    Process.GetCurrentProcess().MaxWorkingSet =
Process.GetCurrentProcess().MinWorkingSet;
}

void GuardarPosiciónVentana()
{
    ia._PosiciónX = Top;

    ia._PosiciónY = Left;

    Settings.Default.PosicionVentanaX =
ia._PosiciónX;

    Settings.Default.PosicionVentanaY =
ia._PosiciónY;

    Settings.Default.Save();
}

void DesactivarMicrófono()
{
    Settings.Default.OnOffMicro = false;

    Settings.Default.Save();
}

void VistaOnStroke()
{
    rec_onoff.Stroke =
System.Windows.Media.Brushes.Green;
}

```

```

void VistaOffStroke()

{

    rec_onoff.Stroke =
System.Windows.Media.Brushes.Red;

}

void Activo(string speech)

{

    Random rnd = new Random();

    if (speech.Equals("Activar el
micrófono"))

    {

        Settings.Default.OnOffMicro = true;

        Settings.Default.Save();

        VistaOnStroke();

    }

    else if
(speech.Equals(Settings.Default.NombreAsisten
te))

    {

        int numran = rnd.Next(1, 5);

        ia.on_mic.Play();

        switch (numran)

        {

            case 1:

                ia.habla_asistente.SpeakAsync("Si " +
Properties.Settings.Default.NombreUsuario + ".
. . .");

                break;

            case 2:

                ia.habla_asistente.SpeakAsync("A sus órdenes "
+ Properties.Settings.Default.NombreUsuario +
". . . .");

                break;

            case 3:

                ia.habla_asistente.SpeakAsync("En qué puedo
ayudarle " +
Properties.Settings.Default.NombreUsuario + ".
. . .");

                break;

            case 4:

                ia.habla_asistente.SpeakAsync("Diga " +
Properties.Settings.Default.NombreUsuario + ".
. . .");

                break;

            case 5:

```

```

ia.habla_asistente.SpeakAsync("Esperando
instrucciones " +
Properties.Settings.Default.NombreUsuario + ".
...");

        break;
    }

    Settings.Default.OnOffMicro = true;

    Settings.Default.Save();

    VistaOnStroke();

    lbl_pantalla.Content = "Escuchando
instrucciones...";

    //Actualizar Gramaticas

loadgramaticas.SeleccionarComandos();

ia.reconocer_voz_usuario.LoadGrammarAsync(
new
Grammar(loadgramaticas.ComandosInternos()))
; //todo esto tiene la información de los
comandos internos

ia.reconocer_voz_usuario.LoadGrammarAsync(
new
Grammar(loadgramaticas.CargarGramáticasCal
c())); //Carga las gramaticas de la calculadora

```

```

ia.reconocer_voz_usuario.LoadGrammarAsync(
new Grammar(new
Choices(loadgramaticas.CargarGramáticasBD()
))); //Carga las gramaticas del usuario

ia.reconocer_voz_usuario.LoadGrammarAsync(
new Grammar(new
Choices(Settings.Default.NombreAsistente))); //
Esto se eliminará en un futuro, pero carga el
nombre del asistente

ia.reconocer_voz_usuario.LoadGrammarAsync(
loadgramaticas.CargarGramáticasWebs()); //hast
a aquí carga todas las gramáticas

    }

    else if (speech == "Desactivar el
micrófono" || speech.Equals("Desactivar
micrófono"))

    {

        DesactivarMicrófono();

        VistaOffStroke();

        lbl_pantalla.Content = "Para activar el
micrófono di \"" +
Settings.Default.NombreAsistente + "\"";

```



```

        pb_Audio.Value = e.AudioLevel;
    }

    private void
    Habla_asistente_SpeakStarted(object sender,
    SpeakStartedEventArgs e)
    {
        ia.habilitarReconocimiento = false;

        VistaOffStroke();

        ia.tiempo = 0;
    }

    private void
    Habla_asistente_SpeakCompleted(object
    sender, SpeakCompletedEventArgs e)
    {
        ia.habilitarReconocimiento = true;

        VistaOnStroke();

        lbl_pantalla.Content = "Escuchando
    instrucciones...";

        ia.tiempo = 0;

        ia.tiempo_desactivar_microfono.Start();
    }

    private void
    Tiempo_desactivar_microfono_Tick(object
    sender, EventArgs e)
    {

```

```

ia.tiempo_desactivar_microfono.Stop();

        ia.tiempo = 0;

        ia.off_mic.Play();
    }

}

private void IdiomaEspañol()
{
    Btn_NomAsisstant.ToolTip = "Ajustes";

    //BtnComando.ToolTip = "Comandos";

    //BtnSerial_Port.ToolTip = "Puerto
Serial";

    lbl_pantalla.Content = "Para activar el
micrófono di \" +
Settings.Default.NombreAsistente + "\"";
}

#endregion

#region Eventos Internos

private void
Reconocer_voz_usuario_AudioLevelUpdated(o
bject sender, AudioLevelUpdatedEventArgs e)
{

```

```

        ia.tiempo++;

        if
        (ia.tiempo.Equals(Settings.Default.OffMicro))

        {

            ia.off_mic.Play();

            DesactivarMicrófono();

            VistaOffStroke();

        ia.tiempo_desactivar_microfono.Stop();

            ia.tiempo = 0;

        }

    }

    public void
    Reconocer_voz_usuario_SpeechRecognized(obj
    ect sender, SpeechRecognizedEventArgs e)

    {

        ia.recogResult = e.Result;

        ia.speech = e.Result.Text;

        Activo(ia.speech);

        if (Settings.Default.OnOffMicro == true)

        {

            ia.tiempo++;

            if (e.Result.Confidence >=
            Settings.Default.ConfidenciaUsuario)

            {

                if (ia.habilitarReconocimiento ==
                true)

                {

                    string ejecutarcmdBD =
                    gramaticas.EjecutarComandosBD(ia.speech); //
                    Esto es igual a e.Result.Text (ia.speech)

                    string ejecutarcmdInternos =
                    gramaticas.SinonimoCmd(ia.speech);

                    string calculadora =
                    gramaticas.Calc(ia.recogResult);

                    string buscador =
                    gramaticas.Busc(ia.recogResult, ia.speech);

                    if (ejecutarcmdBD !=
                    string.Empty)

                    {

                        ia.habla_asistente.SpeakAsync(ejecutarcmdBD
                        + "      .");

                        lbl_pantalla.Content =
                        ejecutarcmdBD;

                    }

                    if (ejecutarcmdInternos !=
                    string.Empty)

```

```

{
    if (ejecutarcmdInternos ==
"Abriendo configuraciones del sistema")
    {
        if (ajustes == null)
        {
            ajustes = new Ajustes();
            ajustes.Show();

            ajustes.Closed +=
delegate (object a, EventArgs b)
            {
                ajustes = null;
            };
        }
    }
    else if (ejecutarcmdInternos
== "Abriendo el editor de comandos")
    {
        interfazDash.ListMenu_SelectionChanged();
    }
}
else if (ejecutarcmdInternos
== "Hasta pronto")
{
    ia.habla_asistente.Speak(ejecutarcmdInternos +
Properties.Settings.Default.NombreUsuario);
    Application.Current.Shutdown();
}
else if (ejecutarcmdInternos
== "Minimizado")
{
    Visibility =
Visibility.Hidden;
}
else if (ejecutarcmdInternos
== "Mostrado")
{
    Visibility =
Visibility.Visible;
}
else if (ejecutarcmdInternos
== "Mostrando todos los comandos")
}

```

```

        {
            ComandosSistema();
            csis.Show();
        }

        {
            csis = new ComandosSistema();
            csis.Show();
        }

        {
            ia.habla_asistente.SpeakAsync(ejecutarecmdInter
            nos + " ");
            lbl_pantalla.Content =
            ejecutarecmdInternos;
        }

        if (calculadora != string.Empty)
        {
            ia.habla_asistente.SpeakAsync(calculadora + "
            .");
            lbl_pantalla.Content =
            calculadora;
        }

        if (buscador != string.Empty)
        {
            ia.habla_asistente.SpeakAsync(buscador + "
            .");
        }
    }

    lbl_pantalla.Content =
    buscador;
}

}

}

#endregion

#region Controles WPF, Eventos WPF
private void
Btn_NomAsisstant_Click(object sender,
RoutedEventArgs e)
{
    if (ajustes == null)
    {
        ajustes = new Ajustes();
        ajustes.Show();
        ajustes.Closed += delegate (object a,
        EventArgs b)
        {
            ajustes = null;
        };
    }
}
}
}

```

```

//private void BtnComando_Click(object
sender, RoutedEventArgs e)

//{

// if (editorCmd == null)

// {

//     editorCmd = new EditorCmd();

//     editorCmd.Show();

//     editorCmd.Closed += delegate
(object a, EventArgs b)

//     {

//         editorCmd = null;

//     };

// }

//}

//private void BtnSerial_Port_Click(object
sender, RoutedEventArgs e)

//{

// if (serial == null)

// {

//     serial = new ControlSerial();

//     serial.Show();

//     serial.Closed += delegate (object a,
EventArgs b)

//     {

//         serial = null;

//     };

// }

//}

//private void Window_Activated(object
sender, EventArgs e)

{

ia.habla_asistente.SelectVoice(Settings.Default.
VozDefault);

ia.habla_asistente.Volume =
Settings.Default.VolumenIA;

Btn_NomAsisstant.Content =
Settings.Default.NombreAsistente;

ia.reconocer_voz_usuario.LoadGrammarAsync(
new Grammar(new
Choices(Settings.Default.NombreAsistente)));

Process.GetCurrentProcess().MaxWorkingSet =
Process.GetCurrentProcess().MinWorkingSet;

}

private void
Window_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)

{

```



```

tbx_NombreUsuario.Visibility =
Visibility.Hidden;

tbx_NombreAsistente.Visibility =
Visibility.Hidden;
cbx_Voces.Visibility =
Visibility.Hidden;
sld_Volumen.Visibility =
Visibility.Hidden;

sld_confidencia.Visibility =
Visibility.Hidden;
Sld_Opacidad.Visibility =
Visibility.Hidden;
lblinfovolumen.Visibility
= Visibility.Hidden;
lblnombreAI.Visibility =
Visibility.Hidden;
cbxOffMicro.Visibility =
Visibility.Hidden;
checkInicio.Visibility =
Visibility.Hidden;
    }
    public void CargarDatos()
    {

CargarVoces_tiempospera();
    lblnombreAI.Content =
Properties.Settings.Default.NombreAsi
stente;
    lbl_info.Content = "
Por favor, verifique los ajustes \n
para una mejor interacción\n
con el asistente";
    tbx_NombreUsuario.Text =
Properties.Settings.Default.NombreUsu
ario;
    tbx_NombreAsistente.Text
=
Properties.Settings.Default.NombreAsi
stente;
    cbx_Voces.Text =
Properties.Settings.Default.VozDefaul
t;
    sld_Volumen.Value =
Properties.Settings.Default.VolumenIA
;
    sld_confidencia.Value =
Properties.Settings.Default.Confidenc
iaUsuario;
    Sld_Opacidad.Value =
Properties.Settings.Default.OpacidadV
entana;
    cbxOffMicro.Text =
Properties.Settings.Default.OffMicro.
ToString();
    checkInicio.IsChecked =
Properties.Settings.Default.InicioWin
dows;
    }

        private void
CargarVoces_tiempospera()
        {
            foreach (InstalledVoice
voces in Janabi.GetInstalledVoices())
            {

cbx_Voces.Items.Add(voces.VoiceInfo.N
ame);
            }

cbxOffMicro.Items.Add("5");

cbxOffMicro.Items.Add("10");

cbxOffMicro.Items.Add("15");

cbxOffMicro.Items.Add("30");

cbxOffMicro.Items.Add("60");
        }

        private void Guardardatos()
        {

Properties.Settings.Default.NombreUsu
ario = tbx_NombreUsuario.Text;

Properties.Settings.Default.NombreAsi
stente = tbx_NombreAsistente.Text;

Properties.Settings.Default.VozDefaul
t = cbx_Voces.Text;

Properties.Settings.Default.VolumenIA
= (int)sld_Volumen.Value;

Properties.Settings.Default.Confidenc
iaUsuario =
Math.Round(sld_confidencia.Value, 1);

Properties.Settings.Default.OpacidadV
entana =
Math.Round(Sld_Opacidad.Value);

Properties.Settings.Default.OffMicro
= Convert.ToInt32(cbxOffMicro.Text);

Properties.Settings.Default.InicioWin
dows =
checkInicio.IsChecked.HasValue;

Properties.Settings.Default.Save();
        }

#endregion
#region Eventos de Control
private void
BtnMinimizar_Click(object sender,
RoutedEventArgs e)

```



```

        {
            WindowState =
            WindowState.Minimized;
        }
        private void
        BtnCerrar_Click(object sender,
        RoutedEventArgs e)
        {
            Guardardatos();

            System.Diagnostics.Process.GetCurrent
            Process().MaxWorkingSet =
            System.Diagnostics.Process.GetCurrent
            Process().MinWorkingSet;
            Close();
        }
        private void
        Window_MouseLeftButtonDown(object
        sender, MouseButtonEventArgs e)
        {
            DragMove();
        }
        private void
        BtnNombreUsuario_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content = "
            Ingrese su nombre con el que \n" + "
            " + "el asistente le hablará";

            OcultarControles();

            tbx_NombreUsuario.Visibility =
            Visibility.Visible;
        }
        private void
        BtnNombreAsistente_Click(object
        sender, RoutedEventArgs e)
        {
            lbl_info.Content = "
            Ingrese el nombre del asistente";

            OcultarControles();

            tbx_NombreAsistente.Visibility =
            Visibility.Visible;
        }
        private void
        BtnVoces_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content = "
            Seleccione una voz instalada";

            OcultarControles();
            cbx_Voces.Visibility =
            Visibility.Visible;
        }
        private void
        BtnVolumen_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content = "
            Nivel de volumen del asistente";
            OcultarControles();
            sld_Volumen.Visibility =
            Visibility.Visible;
            lblinfovolumen.Visibility
            = Visibility.Visible;
        }
        private void
        BtnConfidencia_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content = "
            Nivel de afirmación al hablar";

            OcultarControles();

            sld_confidencia.Visibility =
            Visibility.Visible;
        }
        private void
        BtnOpacidad_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content = "
            Opacidad de las ventanas";

            OcultarControles();
            Sld_Opacidad.Visibility =
            Visibility.Visible;
        }
        private void
        BtnOffMicro_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content =
            "Tiempo de espera para desactivar \n"
            +
            "
            el
            micrófono";

            OcultarControles();
            cbxOffMicro.Visibility =
            Visibility.Visible;
        }
        private void
        BtnInicioWindows_Click(object sender,
        RoutedEventArgs e)
        {
            lbl_info.Content = "
            Marque la casilla para iniciar\n
            con su sistema operativo";
            OcultarControles();
            checkInicio.Visibility =
            Visibility.Visible;
        }
    }

```

```
        private void
Sld_Opacidad_ValueChanged(object
sender,
RoutedPropertyChangedEventArgs<double
> e)
    {
        Opacity =
Sld_Opacidad.Value;
    }
        private void
Sld_Volumen_ValueChanged(object
sender,
RoutedPropertyChangedEventArgs<double
> e)
    {
```

```
        lblinfovolumen.Content =
Math.Round(sld_Volumen.Value);
    }
        private void
CbX_Voces_DropDownClosed(object
sender, EventArgs e)
    {

Janabi.SelectVoice(cbx_Voces.Text);
        Janabi.SpeakAsync("Voz
seleccionada.");
    }

        #endregion
    }
}
```

**ANEXO 8 Código de interfaz de Visualización de Datos HanabiFinal en(C#).**

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Microsoft.Win32;
using System.ComponentModel;

namespace HanabiFinal
{
    public class MeasureModel
    {
        public DateTime DateTime {
get; set; }
        public double Value { get;
set; }
    }
    public class MeasureModel2
    {
        public DateTime DateTime {
get; set; }
        public double Value { get;
set; }
    }
    public class MeasureModel3
    {
        public DateTime DateTime {
get; set; }
        public double Value { get;
set; }
    }
    public partial class InterfazDash
: Window
    {
        public InterfazDash()
        {
            InitializeComponent();
        }

        private void
BotonPoupSalir_Click(object sender,
RoutedEventArgs e)
        {
            Application.Current.Shutdown();
        }
    }
}

private void
BotonAbrirMenu_Click(object sender,
RoutedEventArgs e)
{
    BotonAbrirMenu.Visibility
= Visibility.Collapsed;

    BotonCerrarMenu.Visibility =
Visibility.Visible;
}

private void
BotonCerrarMenu_Click(object sender,
RoutedEventArgs e)
{
    BotonAbrirMenu.Visibility
= Visibility.Visible;

    BotonCerrarMenu.Visibility =
Visibility.Collapsed;
}

public void
ListMenu_SelectionChanged(object
sender, SelectionChangedEventArgs e)
{
    int index =
ListMenu.SelectedIndex;
    MoveCursorMenu(index);
    switch (index)
    {
        case 0:
            //GridP.Children.Clear();

            GridP.Children.Add(new
ControlUsuarioGraficos());
            break;
        case 1:
            //GridP.Children.Clear();

            GridP.Children.Add(new
ControlDeUsuarioSerial());
            break;
        case 2:
            //GridP.Children.Clear();

            GridP.Children.Add(new EditorCmd());
            break;
    }
}

```

```

        private void
MoveCursorMenu(int index)
    {

TracitioningContenidoSlide.OnApplyTem
plate();
        GridCursor.Margin = new
Thickness(0, (100 + (60 * index)), 0,
0);
    }

    private void
Grid_MouseDown(object sender,
MouseButtonEventArgs e)
    {
        DragMove();
    }

    internal void
ListMenu_SelectionChanged()

```

```

    {
        int index =
ListMenu.SelectedIndex;
        MoveCursorMenu(index);
        switch (index)
        {

                case 2:

GridP.Children.Clear();

GridP.Children.Add(new EditorCmd());
                break;
                default:
                break;
        }
    }
}

```

### ANEXO 9 Código de interfaz de control de Relé y entrada serial en (C#).

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Data;
using System.Data.SQLite;
using System.IO.Ports;
using BibliotecaHanabi;

namespace HanabiFinal
{
    /// <summary>
    /// Lógica de interacción para
    ControlDeUsuarioSerial.xaml
    /// </summary>
    public partial class
    ControlDeUsuarioSerial : UserControl
    {
        #region Variables
        private string strBufferOut;
        SerialPort SpPuertos;
        SQLiteConnection conection;
        DataSet ds;
        SQLiteDataAdapter adap;
        SQLiteCommandBuilder cmdb;
        Asistente ia = new
Asistente();
        #endregion

```

```

        public
ControlDeUsuarioSerial()
    {
        InitializeComponent();

        ia.verificarpuertoconectado.Tick +=
Verificarpuertos_Tick;

        ia.verificarpuertodesconectado.Tick
+= Verificarpuertodesconectado_Tick;

        ia.verificarpuertoconectado.Start();
        Refresh_ports();
        MostrarDatos();
        Ocultar_Campos();
        IdiomaEspañol();
    }
    #region Metodos
    void Refresh_ports()
    {
        string[]
Puertos_Disponibles =
SerialPort.GetPortNames();
        cb_ports.Items.Clear();

        foreach (string
puerto_simple in Puertos_Disponibles)
        {

            cb_ports.Items.Add(puerto_simple);
        }
        if (cb_ports.Items.Count
> 0)
        {

            cb_ports.SelectedIndex = 0;
            Btn_conect.IsEnabled
= true;

```

```

        ia.verificarpuertoconectado.Stop();
        ia.verificarpuertodesconectado.Start(
        );
            }
            else
            {
                cb_ports.Text = "
";
                strBufferOut =
string.Empty;
                Btn_conect.IsEnabled
= false;
                Btn_send_Data.IsEnabled = false;
            ia.verificarpuertodesconectado.Stop(
            );
            ia.verificarpuertoconectado.Start();
            }
        }
        void Ocultar_Campos()
        {
            txt_puerto.Visibility =
Visibility.Hidden;
            txt_comando.Visibility =
Visibility.Hidden;
            txt_acción.Visibility =
Visibility.Hidden;
            txt_respuesta.Visibility
= Visibility.Hidden;
            rect_fondo_add.Visibility
= Visibility.Hidden;
        }
        void Mostrar_Campos()
        {
            txt_puerto.Visibility =
Visibility.Visible;
            txt_comando.Visibility =
Visibility.Visible;
            txt_acción.Visibility =
Visibility.Visible;
            txt_respuesta.Visibility
= Visibility.Visible;
            rect_fondo_add.Visibility
= Visibility.Visible;
        }
        void Agregar_Comandos_BD()
        {
            try
            {
                if (txt_puerto.Text
!= string.Empty && txt_comando.Text
!= string.Empty && txt_acción.Text !=
string.Empty)
                {
                    conection = new
SQLiteConnection(Properties.Settings.
Default.ConexionBD);
                    conection.Open();
                    SQLiteCommand cmd
= new
SQLiteCommand(string.Format("INSERT
INTO CmdPuertoSerial (PuertoCOM,
Comando, Accion, Respuesta)
values('{0}','{1}','{2}','{3}')"
,txt_puerto.Text, txt_comando.Text,
txt_acción.Text, txt_respuesta.Text),
conection);
                    cmd.ExecuteNonQuery();
                    txt_comando.Text
= string.Empty;
                    txt_acción.Text =
string.Empty;
                    txt_respuesta.Text = string.Empty;
                }
            }
            else
            {
                System.Windows.MessageBox.Show("No se
puede agregar comandos vacíos o
incompletos", "Error AVSARA",
MessageBoxButton.OK,
MessageBoxImage.Error);
            }
        }
        catch
        {
            System.Windows.MessageBox.Show("Ha
ocurrido un problema", "Aviso",
MessageBoxButton.OK,
MessageBoxImage.Warning);
        }
    }
    void MostrarDatos()
    {
        try
        {
            conection = new
SQLiteConnection("Data Source =
|DataDirectory|\\DataBase\\Data.sqlit
e; Version = 3");
            conection.Open();
            adap = new
SQLiteDataAdapter("SELECT
ID,PuertoCOM,Comando, Accion,
Respuesta FROM CmdPuertoSerial",
conection);
            ds = new DataSet();
            adap.Fill(ds);
            DatagridBD.ItemsSource =
ds.Tables[0].DefaultView;
            conection.Close();
        }
        catch
    }
}

```

```

        {
            System.Windows.MessageBox.Show("Se ha
            encontrado un error en la base de
            datos", "Aviso", MessageBoxButton.OK,
            MessageBoxImage.Information);
        }
        finally
        {
            connection.Close();
        }
    }
    void OcultarColumnas()
    {
        DataGridView.Columns[0].Visibility =
        Visibility.Hidden;
    }
    void IdiomaEspañol()
    {
        lblTitle.Content =
        "CONTROLADOR DE PUERTO SERIAL";
        Btn_conect.Content =
        "Conectar";
        Btn_send_Data.Content =
        "Enviar Dato";
        cb_ports.ToolTip =
        "Puerto COM";
        tbx_BaudRate.ToolTip =
        "Baudio";
        txt_date.ToolTip =
        "Escribe un dato de prueba para
        enviar al Arduino";
        lblPuertoCOM.Content =
        "PUERTO";
        lblcomando.Content =
        "COMANDO+";
        lblaccion.Content =
        "ACCIÓN+";
        lblRespuesta.Content =
        "RESPUESTA+";
        Btn_add_comandos.Content
        = "Nuevo Comando";
        Btn_delet_cmd.Content =
        "Eliminar";
    }
    #endregion
    #region Eventos Internos
    private void
    Btn_conect_Click(object sender,
    RoutedEventArgs e)
    {
        try
        {
            if
            (Btn_conect.Content.Equals("Conectar"
            ))
            {
                SpPuertos = new
                SerialPort("COM13")
                {
                    BaudRate =
                    Int32.Parse(tbx_BaudRate.Text),
                    DataBits = 8,
                    Parity =
                    Parity.None,
                    StopBits =
                    StopBits.One,
                    Handshake =
                    Handshake.None,
                    PortName =
                    "COM13"
                };
            }
            try
            {
                SpPuertos.Open();
                Btn_conect.Content = "Desconectar";
                Btn_conect.ToolTip = "Conectado";
                Btn_send_Data.IsEnabled = true;
            }
            catch
            (FormatException ex)
            {
                System.Windows.MessageBox.Show(ex.Mes
                sage);
            }
            else if
            (Btn_conect.Content.Equals("Desconect
            ar"))
            {
                SpPuertos.Close();
                Btn_conect.Content = "Conectar";
                Btn_conect.ToolTip = "Sin conexión";
                Btn_send_Data.IsEnabled = false;
            }
            catch (FormatException
            ex)
            {
                System.Windows.MessageBox.Show(ex.Mes
                sage);
            }
        }
        private void
        Btn_send_Data_Click(object sender,
        RoutedEventArgs e)
        {
            try

```

```

        {
SpPuertos.DiscardOutBuffer();
        strBufferOut =
txt_date.Text;
SpPuertos.Write(strBufferOut);
        }
        catch (FormatException
ex)
        {

System.Windows.MessageBox.Show(ex.Mes
sage);
        }
        }
        private void
Btn_add_comandos_Click(object sender,
RoutedEventArgs e)
        {
            if
(Btn_add_comandos.Content.Equals("Nue
vo Comando"))
            {
                Mostrar_Campos();

Btn_add_comandos.Content = "Agregar";
            }
            else if
(Btn_add_comandos.Content.Equals("Agr
egar"))
            {

Agregar_Comandos_BD();
                cmdb = new
SQLiteCommandBuilder(adap);
                adap.Update(ds);
                MostrarDatos();
                OcultarColumnas();
                Ocultar_Campos();

Btn_add_comandos.Content = "Nuevo
Comando";
            }
        }

        private void
Window_Loaded(object sender,
RoutedEventArgs e)
        {
            strBufferOut =
string.Empty;
            Btn_conect.IsEnabled =
false;
            Btn_send_Data.IsEnabled =
false;
            OcultarColumnas();
        }
        private void
DatagridBD_SelectedCellsChanged(objec

```

```

t sender,
SelectedCellsChangedEventArgs e)
        {
            Btn_delet_cmd.Visibility
= Visibility.Visible;
        }
        private void
Btn_delet_cmd_Click(object sender,
RoutedEventArgs e)
        {
            List<DataRow> theRows =
new List<DataRow>();
            for (int i = 0; i <
DatagridBD.SelectedItems.Count; ++i)
            {
                Object o =
DatagridBD.SelectedItems[i];
                if (o !=
CollectionView.NewItemPlaceholder)
                {
                    DataRowView r =
(DataRowView)o;
                    theRows.Add(r.Row);
                }
            }

            foreach (DataRow r in
theRows)
            {
                int k =
ds.Tables[0].Rows.IndexOf(r);
                ds.Tables[0].Rows[k].Delete();
            }

            cmdb = new
SQLiteCommandBuilder(adap);
            adap.Update(ds);

            OcultarColumnas();
        }
        private void
Verificarpuertodesconectado_Tick(obje
ct sender, EventArgs e)
        {
            Refresh_ports();
        }
        private void
Verificarpuertos_Tick(object sender,
EventArgs e)
        {
            Refresh_ports();
        }
        private void
DatagridBD_MouseLeftButtonDown(object
sender, MouseButtonEventArgs e)
        {
            DatagridBD.UnselectAll();
            Btn_delet_cmd.Visibility
= Visibility.Hidden;
        }

```



```

        #endregion
    }
}

ANEXO 10 Código de interfaz de Base de datos de HanabiFinal en (C#).
using HanabiFinal.Properties;
using Microsoft.Win32;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.IO;
using System.Media;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Input;
using System.Runtime.InteropServices;
using System.ComponentModel;

namespace HanabiFinal
{
    public partial class EditorCmd :
    UserControl
    {
        #region Variables
        private SQLiteConnection con;
        private SQLiteDataAdapter
adap;
        private SQLiteCommand cmd;
        private SQLiteCommandBuilder
cmdb;
        private DataSet ds;
        private ComandosSistema
comandos;
        private ArrayList
listaComando = new ArrayList();
        private bool celda_ingresada
= false;
        private string tipo;
        #endregion
        public EditorCmd()
        {
            InitializeComponent();
            CargarCmdSociales();
            IdiomaEspañol();
        }
        #region Metodos internos
        private void IdiomaEspañol()
        {
            lblTitle.Content =
"COMANDOS DEL SISTEMA";
            BtnSociales.Content =
"SOCIALES";
            BtnCarpetas.Content =
"CARPETAS";
            BtnAplicaciones.Content =
"PROGRAMAS";
            BtnPaginasWebs.Content =
"PAGINAS WEBS";
            BtnInternos.Content =
"INTERNOS";
            BtnAyuda.Content =
"AYUDA";
            BtnIO.Content = "BASE DE
DATOS";
            BtnNuevoComando.Content =
"NUEVO COMANDO";
            BtnEliminarComando.Content =
"ELIMINAR";
            lblcomandoTitle.Content =
"COMANDO+";
            lblaccionTitle.Content =
"ACCIÓN+";
            lblRespuesta.Content =
"RESPUESTA+";
            lblbdio.Content = "BASE
DE DATOS";
            lbbdinfo.Content =
"Importar: Reemplaza la actual base
de datos por una nueva.\nExportar:
Guarde la base de datos en su
pc.\nNota: No cambiar el nombre a la
base de datos.";
            BtnImportar.Content =
"IMPORTAR";
            BtnExportar.Content =
"EXPORTAR";
        }
        private void
CargarCmdSociales()
        {
            tipo = "Sociales";
            try
            {
                con = new
SQLiteConnection(Settings.Default.Con
exionBD);
                con.Open();
                adap = new
SQLiteDataAdapter("SELECT ID,Comando,
Accion, Respuesta FROM CmdSociales",
con);
                ds = new DataSet();
                adap.Fill(ds);
                DataGridP.ItemsSource
= ds.Tables[0].DefaultView;
                con.Close();
            }
            catch
            {
                System.Windows.MessageBox.Show("Hay
un error en la base de datos, en la
tabla de CmdSociales, por favor
verifique el error.");
            }
        }
    }
}

```

```

    }
    private void
    CargarCmdCarpetas()
    {
        tipo = "Carpetas";
        try
        {
            con = new
            SQLiteConnection(Settings.Default.Con
            exionBD);
            con.Open();
            adap = new
            SQLiteDataAdapter("SELECT Id,
            Comando, Accion, Respuesta FROM
            CmdCarpetas", con);
            ds = new DataSet();
            adap.Fill(ds);
            DataGridView.ItemsSource
            = ds.Tables[0].DefaultView;
            con.Close();
        }
        catch
        {

```

```

            System.Windows.MessageBox.Show("Hay
            un error en la base de datos, en la
            tabla de CmdCarpetas, por favor
            verifique el error.");
        }
    }

```

```

    private void
    CargarCmdAplicaciones()
    {
        tipo = "Aplicaciones";
        try
        {
            con = new
            SQLiteConnection(Settings.Default.Con
            exionBD);
            con.Open();
            adap = new
            SQLiteDataAdapter("SELECT Id,
            Comando, Accion, Respuesta FROM
            CmdAplicaciones", con);
            ds = new DataSet();
            adap.Fill(ds);
            DataGridView.ItemsSource
            = ds.Tables[0].DefaultView;
            con.Close();
        }
        catch
        {

```

```

            System.Windows.MessageBox.Show("Hay
            un error en la base de datos, en la
            tabla de CmdAplicaciones, por favor
            verifique el error.");
        }
    }

```

```

    private void
    CargarCmdPaginasWebs()
    {
        tipo = "Paginas Webs";
        try
        {
            con = new
            SQLiteConnection(Settings.Default.Con
            exionBD);
            con.Open();
            adap = new
            SQLiteDataAdapter("SELECT Id,
            Comando, Accion, Respuesta FROM
            CmdPaginasWebs", con);
            ds = new DataSet();
            adap.Fill(ds);
            DataGridView.ItemsSource
            = ds.Tables[0].DefaultView;
            con.Close();
        }
        catch
        {

```

```

            System.Windows.MessageBox.Show("Hay
            un error en la base de datos, en la
            tabla de CmdPaginasWebs, por favor
            verifique el error.");
        }
    }

```

```

    private void
    CargarCmdInternos()
    {
        tipo = "Internos";
        try
        {
            con = new
            SQLiteConnection(Settings.Default.Con
            exionBD);
            con.Open();
            adap = new
            SQLiteDataAdapter("SELECT ID,
            Comando, Accion, Sinonimo FROM
            CmdInternos", con);
            ds = new DataSet();
            adap.Fill(ds);
            DataGridView.ItemsSource
            = ds.Tables[0].DefaultView;
            con.Close();
        }
        catch
        {

```

```

            System.Windows.MessageBox.Show("Vaya,
            esto no debería pasar con los
            comandos internos, pero igual hay un
            error.");
        }
    }

```

```

    private void
    BorrarComandoBD()

```

```

        {
            List<DataRow> theRows =
            new List<DataRow>();
            for (int i = 0; i <
            DataGridP.SelectedItems.Count; ++i)
            {
                object o =
                DataGridP.SelectedItems[i];
                if (o !=
                CollectionView.NewItemPlaceholder)
                {
                    DataRowView r =
                    (DataRowView)o;

                    theRows.Add(r.Row);
                }

                foreach (DataRow r in
                theRows)
                {
                    int k =
                    ds.Tables[0].Rows.IndexOf(r);
                    ds.Tables[0].Rows[k].Delete();
                }
            }

            private void
            ActualizarTabla()
            {
                cmdb = new
                SQLiteCommandBuilder(adap);
                adap.Update(ds);
            }

            private void AgregarCmdBD()
            {
                try
                {
                    if (tipo ==
                    "Sociales")
                    {
                        con = new
                        SQLiteConnection(Settings.Default.Con
                        exionBD);

                        con.Open();
                        cmd = new
                        SQLiteCommand(string.Format("INSERT
                        INTO CmdSociales (Comando, Accion,
                        Respuesta)
                        values('{0}','{1}','{2}']",
                        Tbx_Comando.Text, tbxAccion.Text,
                        TbxRespuesta.Text), con);

                        cmd.ExecuteNonQuery();
                        con.Close();

                        ActualizarTabla();

                        CargarCmdSociales();
                        LimpiarTextBox();
                    }
                    else if (tipo ==
                    "Carpetas")
                    {
                        con = new
                        SQLiteConnection(Settings.Default.Con
                        exionBD);

                        con.Open();
                        cmd = new
                        SQLiteCommand(string.Format("INSERT
                        INTO CmdCarpetas (Comando, Accion,
                        Respuesta)
                        values('{0}','{1}','{2}']",
                        Tbx_Comando.Text, tbxAccion.Text,
                        TbxRespuesta.Text), con);

                        cmd.ExecuteNonQuery();
                        con.Close();

                        ActualizarTabla();

                        CargarCmdCarpetas();
                        LimpiarTextBox();
                    }
                    else if (tipo ==
                    "Aplicaciones")
                    {
                        con = new
                        SQLiteConnection(Settings.Default.Con
                        exionBD);

                        con.Open();
                        cmd = new
                        SQLiteCommand(string.Format("INSERT
                        INTO CmdAplicaciones (Comando,
                        Accion, Respuesta)
                        values('{0}','{1}','{2}']",
                        Tbx_Comando.Text, tbxAccion.Text,
                        TbxRespuesta.Text), con);

                        cmd.ExecuteNonQuery();
                        con.Close();

                        ActualizarTabla();

                        CargarCmdAplicaciones();
                        LimpiarTextBox();
                    }
                    else if
                    (tipo.Equals("Paginas Webs"))
                    {
                        con = new
                        SQLiteConnection(Settings.Default.Con
                        exionBD);

                        con.Open();
                        cmd = new
                        SQLiteCommand(string.Format("INSERT
                        INTO CmdPaginasWebs (Comando, Accion,
                        Respuesta)
                        values('{0}','{1}','{2}']",
                        Tbx_Comando.Text, tbxAccion.Text,
                        TbxRespuesta.Text), con);
                    }
                }
            }
        }
    
```

```

cmd.ExecuteNonQuery();
        con.Close();

ActualizarTabla();

CargarCmdPaginasWebs();
        LimpiarTextBox();
    }
    catch
    {

System.Windows.MessageBox.Show("Aquí
hubo un problemón, posiblemente sea
el código o no encuentro la base de
datos.");
    }

    private void LimpiarTextBox()
    {
        Tbx_Comando.Text =
string.Empty;
        tbxAccion.Text =
string.Empty;
        TbxRespuesta.Text =
string.Empty;
    }

    private void OcultarBotones()
    {

BtnNuevoComando.Visibility =
Visibility.Hidden;

BtnEliminarComando.Visibility =
Visibility.Hidden;

    private void MostrarBotones()
    {

BtnNuevoComando.Visibility =
Visibility.Visible;

BtnEliminarComando.Visibility =
Visibility.Visible;
    }

    private void
Buscar_Archivos()
    {
        if (tipo ==
"Aplicaciones")
        {

Microsoft.Win32.OpenFileDialog
buscar_programa = new
Microsoft.Win32.OpenFileDialog();

        buscar_programa.ShowDialog();

        tbxAccion.Text =
        buscar_programa.FileName.ToString();
        }
        else if (tipo ==
"Carpetas")
        {

System.Windows.Forms.FolderBrowserDia
log buscar_carpetas = new
System.Windows.Forms.FolderBrowserDia
log();

        buscar_carpetas.ShowDialog();
        tbxAccion.Text =
        buscar_carpetas.SelectedPath;
        }
    }
    #endregion
    #region Eventos de controles
    private void
BtnSociales_Click(object sender,
RoutedEventArgs e)
    {
        CargarCmdSociales();
        MostrarBotones();

        DataGridViewP.Columns[0].Visibility =
        Visibility.Hidden;
        lblRespuesta.Content =
"RESPUESTA+";
    }
    private void
BtnCarpetas_Click(object sender,
RoutedEventArgs e)
    {
        CargarCmdCarpetas();
        MostrarBotones();

        DataGridViewP.Columns[0].Visibility =
        Visibility.Hidden;
        lblRespuesta.Content =
"RESPUESTA+";
    }
    private void
BtnAplicaciones_Click(object sender,
RoutedEventArgs e)
    {
        CargarCmdAplicaciones();
        MostrarBotones();

        DataGridViewP.Columns[0].Visibility =
        Visibility.Hidden;
        lblRespuesta.Content =
"RESPUESTA+";
    }
    private void
BtnPaginasWebs_Click(object sender,
RoutedEventArgs e)
    {
        CargarCmdPaginasWebs();
        MostrarBotones();
    }
    }
    #endregion
}

```

```

        DataGridP.Columns[0].Visibility =
        Visibility.Hidden;
        lblRespuesta.Content =
        "RESPUESTA+";
    }
    private void
    BtnInternos_Click(object sender,
    RoutedEventArgs e)
    {
        CargarCmdInternos();
        OcultarBotones();

        DataGridP.Columns[0].Visibility =
        Visibility.Hidden;

        DataGridP.Columns[1].IsReadOnly =
        true;

        DataGridP.Columns[2].IsReadOnly =
        true;
        lblRespuesta.Content =
        "SINONIMOS+";
    }
    private void
    BtnEliminarComando_Click(object
    sender, RoutedEventArgs e)
    {
        BorrarComandoBD();
        ActualizarTabla();

        DataGridP.Columns[0].Visibility =
        Visibility.Hidden;
    }
    private void
    BtnBuscar_Click(object sender,
    RoutedEventArgs e)
    {
        Buscar_Archivos();
    }

    private void
    Window_Loaded(object sender,
    RoutedEventArgs e)
    {
        DataGridP.Columns[0].Visibility =
        Visibility.Hidden;
    }

    private void
    BtnNuevoComando_Click(object sender,
    RoutedEventArgs e)
    {
        if
        (BtnNuevoComando.Content.Equals("NUEVO
        O COMANDO"))
        {
            Tbx_Comando.Visibility =
            Visibility.Visible;

            tbxAccion.Visibility
            = Visibility.Visible;
            BtnBuscar.Visibility
            = Visibility.Visible;

            TbxRespuesta.Visibility =
            Visibility.Visible;

            BtnNuevoComando.Content = "AGREGAR";
        }
        else
        {
            if (Tbx_Comando.Text
            == string.Empty && tbxAccion.Text ==
            string.Empty && TbxRespuesta.Text ==
            string.Empty)
            {
                System.Windows.MessageBox.Show("No se
                permiten comandos vacios", "Error de
                sistema", MessageBoxButton.OK,
                MessageBoxImage.Error);
            }
            else
            {
                AgregarCmdBD();

                DataGridP.Columns[0].Visibility =
                Visibility.Hidden;

                Tbx_Comando.Visibility =
                Visibility.Hidden;

                tbxAccion.Visibility =
                Visibility.Hidden;

                BtnBuscar.Visibility =
                Visibility.Hidden;

                TbxRespuesta.Visibility =
                Visibility.Hidden;

                BtnNuevoComando.Content = "NUEVO
                COMANDO";
            }
        }
    }
    private void
    DataGridP_CellEditEnding(object
    sender,
    DataGridCellEditEndingEventArgs e)
    {
        celda_ingresada = true;
    }
    private void
    DataGridP_SelectedCellsChanged(object
    sender, SelectedCellsChangedEventArgs e)
    {
        if (celda_ingresada ==
        true)
    
```

```

        {
            ActualizarTabla();
            if (tipo ==
"Sociales")
            {
                CargarCmdSociales();
            }
            else if (tipo ==
"Carpetas")
            {
                CargarCmdCarpetas();
            }
            else if (tipo ==
"Aplicaciones")
            {
                CargarCmdAplicaciones();
            }
            else if (tipo ==
"Paginas Webs")
            {
                CargarCmdPaginasWebs();
            }
            else
            {
                CargarCmdInternos();

                DataGridP.Columns[1].IsReadOnly =
                true;

                DataGridP.Columns[2].IsReadOnly =
                true;
            }

            DataGridP.Columns[0].Visibility =
            Visibility.Hidden;
            celda_ingresada = false;
        }
        private void
        BtnAyuda_Click(object sender,
        RoutedEventArgs e)
        {
            if (comandos == null)
            {
                comandos = new
                ComandosSistema() { WindowState =
                WindowState.Normal };
                comandos.Show();
                comandos.Closed +=
                delegate (object a, EventArgs b)
                {
                    comandos = null;
                }
            }
        }

        private void
        BtnIO_Click(object sender,
        RoutedEventArgs e)
        {
            if
            (Settings.Default.ingresóIE == true)
            {
                rectangle_Exp.Visibility =
                Visibility.Visible;
                lblbdio.Visibility =
                Visibility.Visible;
                lbbdinfo.Visibility =
                Visibility.Visible;

                BtnExportar.Visibility =
                Visibility.Visible;

                BtnImportar.Visibility =
                Visibility.Visible;

                Settings.Default.ingresóIE = false;
                Settings.Default.Save();
            }
            else if
            (Settings.Default.ingresóIE == false)
            {
                rectangle_Exp.Visibility =
                Visibility.Hidden;
                lblbdio.Visibility =
                Visibility.Hidden;
                lbbdinfo.Visibility =
                Visibility.Hidden;

                BtnExportar.Visibility =
                Visibility.Hidden;

                BtnImportar.Visibility =
                Visibility.Hidden;

                Settings.Default.ingresóIE = true;
                Settings.Default.Save();
            }
        }

        //private void
        BtnImportar_Click(object sender,
        RoutedEventArgs e)
        //{
        //    try
        //    {
        //        OpenFileDialog
        //        openFileDialog = new OpenFileDialog
        //        {
        //            Filter = "Base
        //            de datos de SQLite(.sqlite)|
        //            *.sqlite",
    
```

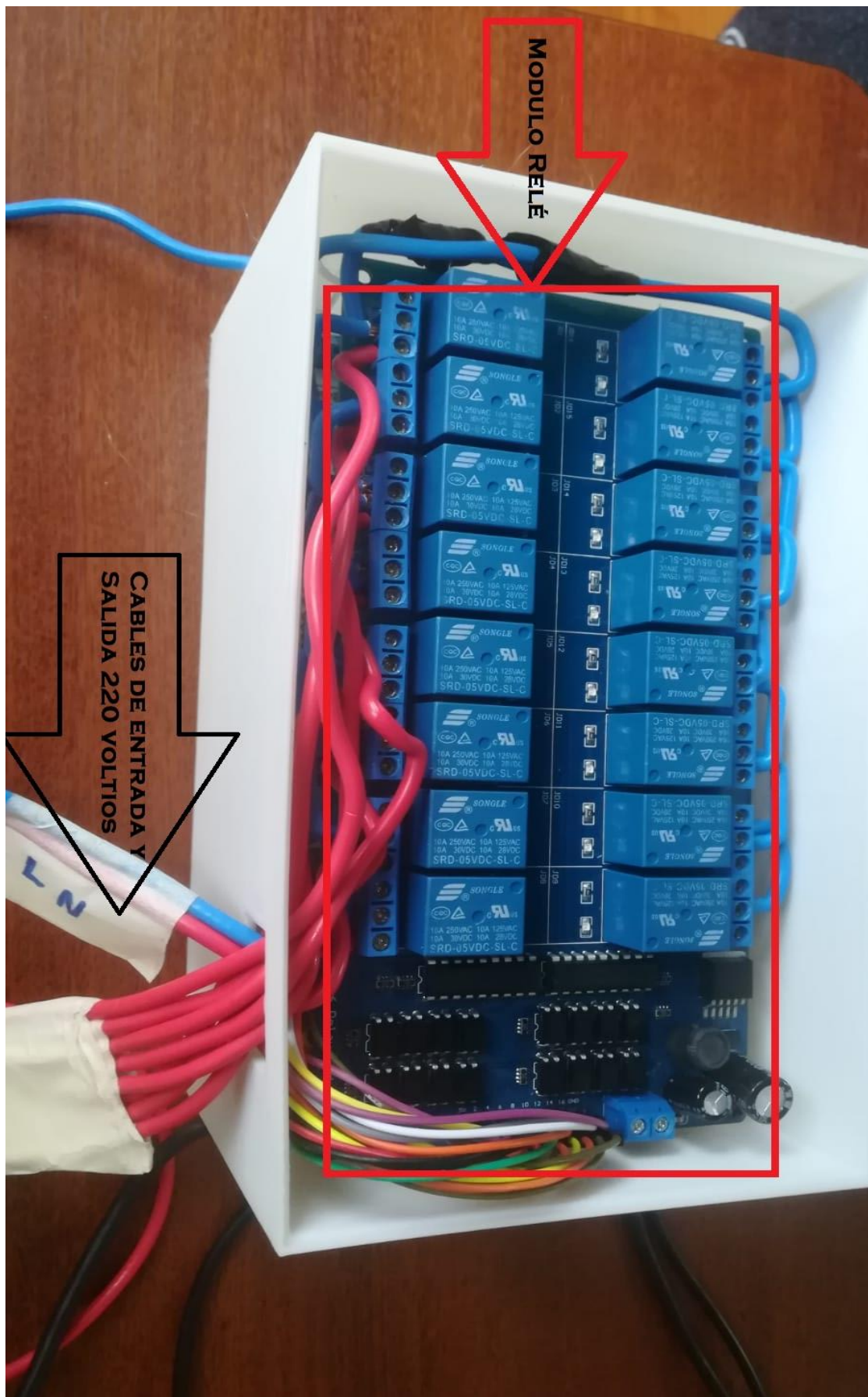
```

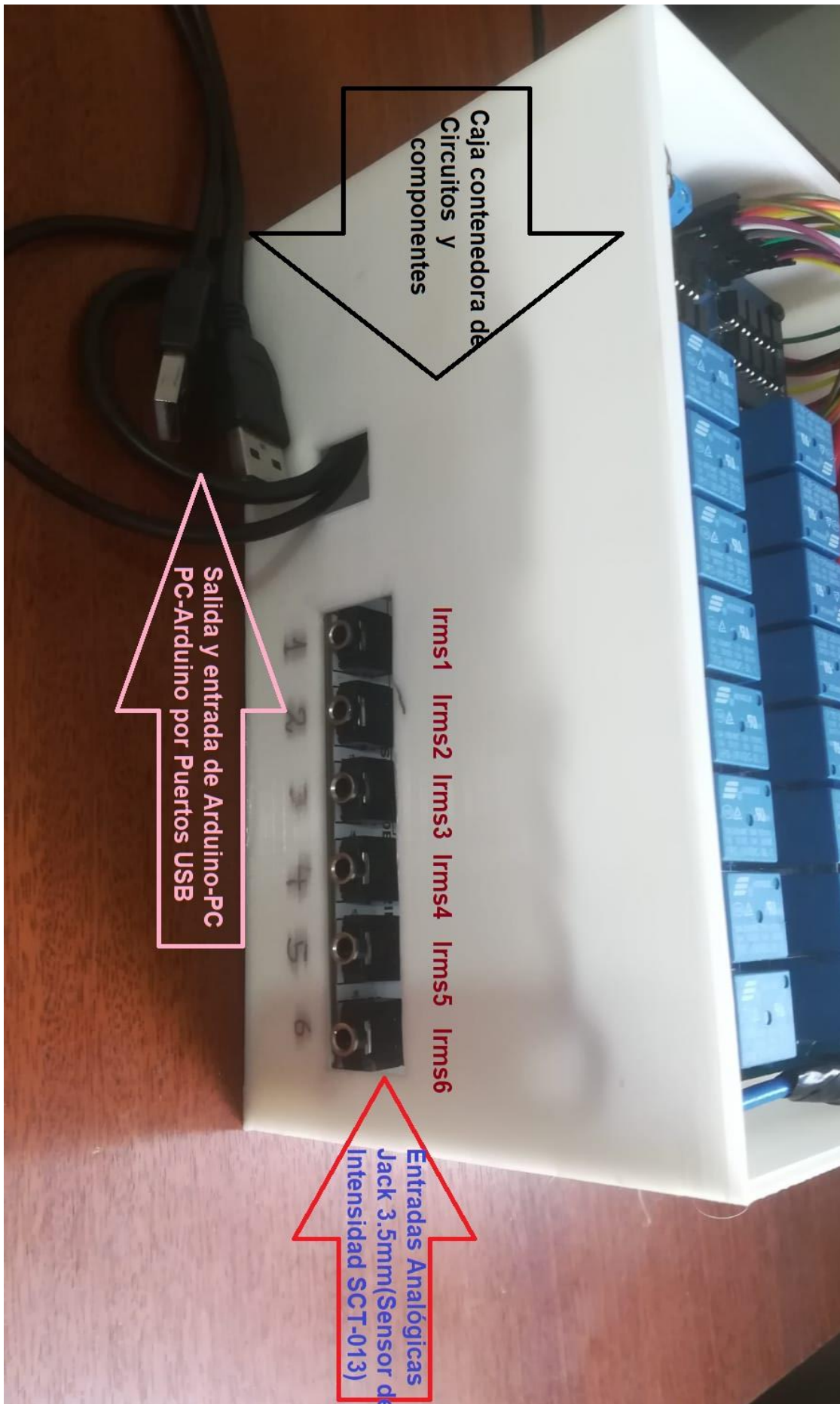
        //          Title = "Cargar
base de datos",
        //
InitialDirectory =
Environment.GetFolderPath(Environment
.SpecialFolder.Personal)
        //          };
        //          if
(openFileDialog.ShowDialog().ToString
()).Equals("OK"))
        //          {
        //          FileInfo
fileInfo = new
FileInfo(openFileDialog.FileName);
        //
fileInfo.CopyTo("C:\\HanabiFinal\\Dat
aBase\\Data" + fileInfo.Extension,
true);
        //
System.Windows.MessageBox.Show("Base
de datos importada con exito");
        //
SystemSounds.Asterisk.Play();
        //          }
        //
openFileDialog.Dispose();
        //          }
        //          catch (Exception ex)
        //          {
        //
System.Windows.MessageBox.Show(ex.Mes
sage);
        //          }
        //}
        //private void
BtnExportar_Click(object sender,
RoutedEventArgs e)
        //{
        //          try
        //          {
        //          SaveFileDialog
export_bd = new SaveFileDialog
        //          {
        //
InitialDirectory =
Environment.GetFolderPath(Environment
.SpecialFolder.MyComputer),
        //          FileName =
"Data",
        //          Title =
"Guardar base de datos",
        //          Filter = "Base
de datos de
SQLite(.sqlite)|*.sqlite",
        //
RestoreDirectory = true
        //          };
        //          if
        //          {
        //          FileInfo
fileInfo = new
FileInfo("C:\\HanabiFinal\\Data
Base\\Data.sqlite");
        //
fileInfo.CopyTo(export_bd.FileName,
true);
        //
rectangle_Exp.Visibility =
Visibility.Hidden;
        //
lblbdio.Visibility =
Visibility.Hidden;
        //
lbbdinfo.Visibility =
Visibility.Hidden;
        //
BtnExportar.Visibility =
Visibility.Hidden;
        //
BtnImportar.Visibility =
Visibility.Hidden;
        //
SystemSounds.Asterisk.Play();
        //
System.Windows.MessageBox.Show("La
base de datos se ha exportado
correctamente", "HanabiFinal",
MessageBoxButton.OK,
MessageBoxImage.Information);
        //          }
        //
export_bd.Dispose();
        //          }
        //          catch (Exception ex)
        //          {
        //
System.Windows.MessageBox.Show(ex.Mes
sage);
        //          }
        //}
        //private void
BtnAvanzados_Click(object sender,
RoutedEventArgs e)
        //
        //
System.Windows.MessageBox.Show("En
algún momento habrá una
característica por aquí.");
        //
        //endregion
    }
}

```

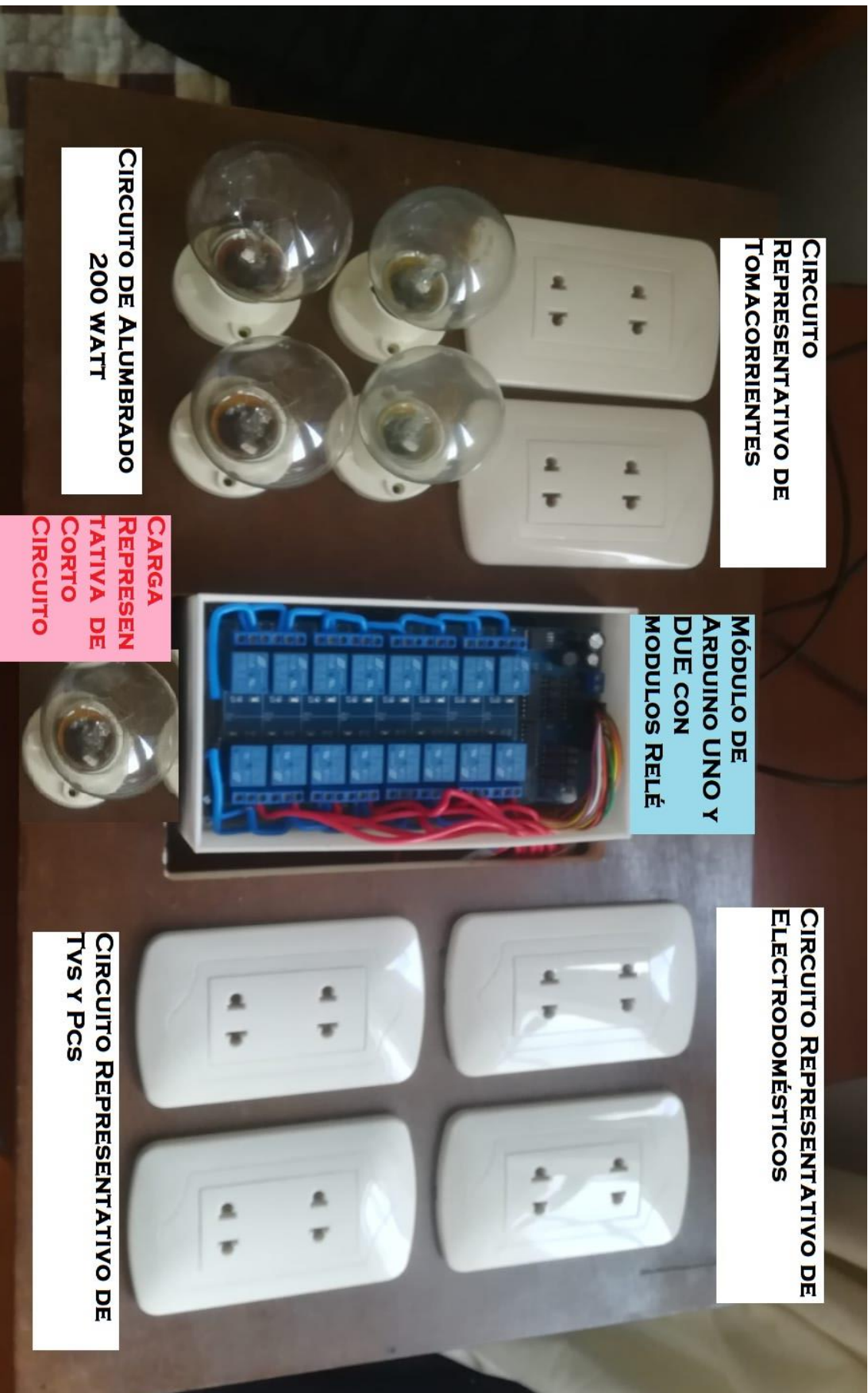


**ANEXO 11 Parámetros de entrada y salida del prototipo final.**









**CIRCUITO REPRESENTATIVO DE TOMACORRIENTES**

**MÓDULO DE ARDUINO UNO Y DUE CON MODULOS RELÉ**

**CIRCUITO REPRESENTATIVO DE ELECTRODOMÉSTICOS**

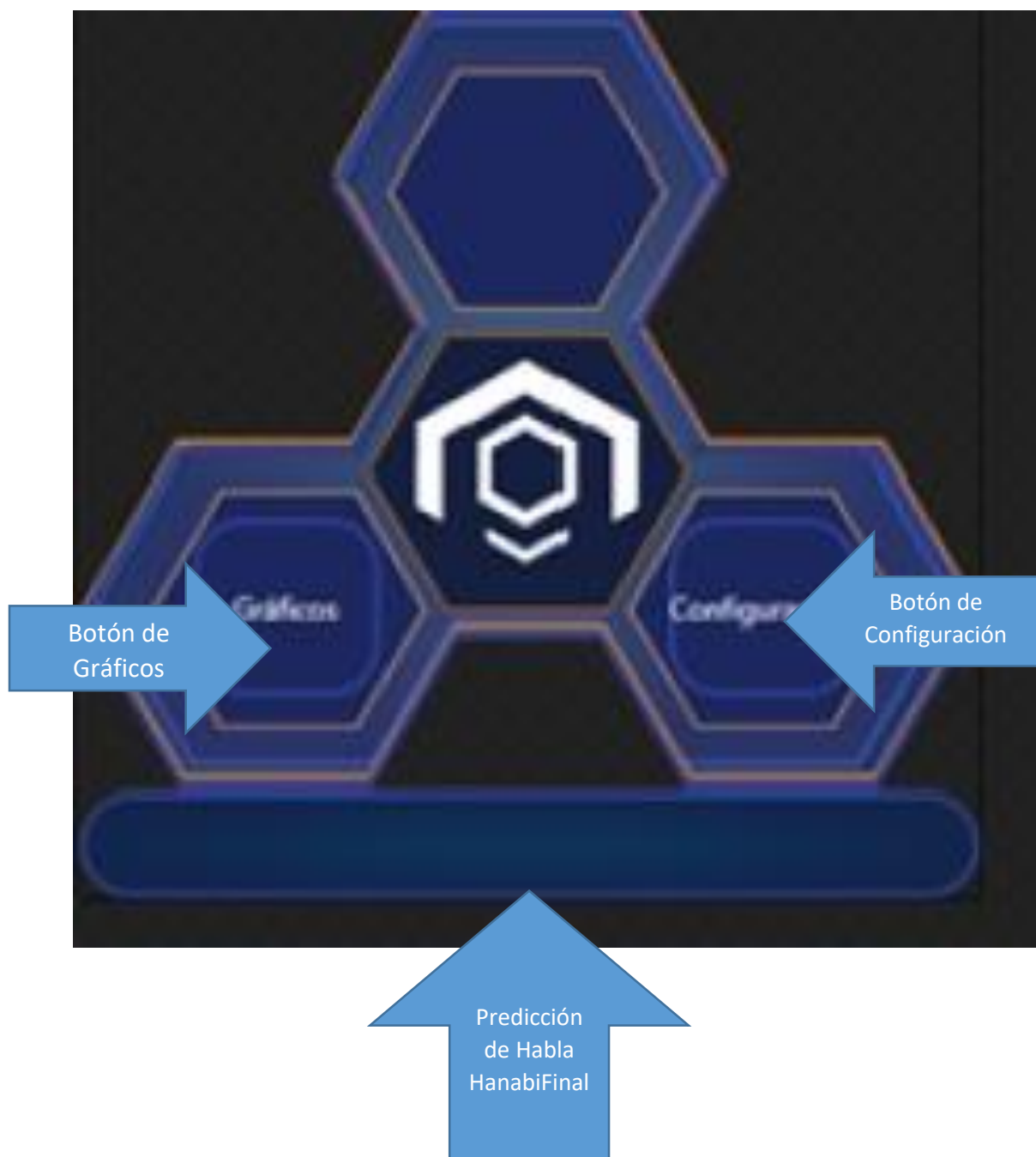
**CIRCUITO DE ALUMBRADO 200 WATT**

**CARGA REPRESENTATIVA DE CORTO CIRCUITO**

**CIRCUITO REPRESENTATIVO DE TVS Y PCS**

**ANEXO 12 PARÁMETROS DE INTERFAZ DE SOFTWARE HANABI.**

Paso N°1 Interfaz Principal:



**BOTONES (Comandos, Comandos y conexión Serie, Gráficos)**

The screenshot shows the HanabiEnergyM. mobile application interface. At the top, there are three menu items: GRAFICOS, C. SERIE, and COMANDOS. Below this, a central panel displays real-time energy data: 220 VOLT. (VOLTAGE ACTUAL), 1440 WATT (POTENCIA ACTUAL), and 2.3 AMP. (CORRIENTE ACTUAL). To the left of this panel are buttons for TABLAS and P.FUGAE. Below the data panel is a horizontal bar with five categories: HOGAR, ALUMBRADO, TOMACORRIENTE, A.ELECTRICOS, and TVSPCS. At the bottom, there is a tree diagram showing energy consumption in KWh for each category: KWh Hogar S./., KWh Aluminio S./., KWh TVSPCS S./., KWh Electrico S./., and KWh Tomacorriente S./., all showing a value of 0. A 'CONECTAR' button is located at the bottom right. The app name 'HanabiEnergyM.' and 'UNAP' are visible in the bottom right corner.

**Botones Tablas, Prueba De Fuga a Tierra** (Annotations pointing to TABLAS and P.FUGAE buttons)

**Botones de Circuitos conexión en el Hogar.** (Annotation pointing to the HOGAR button in the category bar)

**Visualizador KWh y Soles.** (Annotation pointing to the KWh consumption cards in the tree diagram)

**ANEXO 13 PRESUPUESTO DE PROTOTIPO (FINAL).**

COMPONENTE	CANTIDAD	COSTO IMPORTADO (CHINA envío incluido)	COSTO PERÚ
SENSOR DE CORRIENTE SCT- 013-000	01 und.	S/ 22.69 (XY Component Store Aliexpress)	S/ 50.00 (Naylampelectronics)
SENSOR DE CORRIENTE SCT- 013-030	04 und	S/ 77.51 (XY Component Store Aliexpress)	S/ 160.00 (Naylampelectronics)
SENSOR DE VOLTAE ZMPT101B	01 und.	S/ 5.42 (ZQC Module Store Aliexpress)	S/ 28.00 (Naylampelectronics)
ARDUINO DUE	01 und.	S/ 42.94 (GreatZT Store Aliexpress)	S/ 90.00 (Naylampelectrónicos)
PLACA SHIELD DE MEDICIÓN DE ENERGÍA (Elaboración Propia)	01 und.	S/ 39.96 (elaboración Propia)	S/ 39.96 (elaboración Propia)
PINES DE CONEXION		S/ 5.00	S/ 5.00
<b>TOTAL</b>		<b>S/ 193.52</b>	<b>S/ 372.96</b>

**ANEXO 14 PRESUPUESTO TOTAL DE INVESTIGACIÓN.**

COMPONENTE	CANTIDAD	COSTO IMPORTADO (CHINA envío incluido)	COSTO PERÚ (IGV incluido)
SENSOR DE CORRIENTE SCT- 013-000	01 und.	S/ 22.69 (XY Component Store Aliexpress)	S/ 50.00 (Naylampelectronics)
SENSOR DE CORRIENTE SCT- 013-030	04 und	S/ 77.51 (XY Component Store Aliexpress)	S/ 160.00 (Naylampelectronics)
SENSOR DE VOLTAE ZMPT101B	01 und.	S/ 5.42 (ZQC Module Store Aliexpress)	S/ 28.00 (Naylampelectronics)
ARDUINO DUE	01 und.	S/ 42.94 (GreatZT Store Aliexpress)	S/ 90.00 (Naylampelectronics)
PLACA SHIELD DE MEDICIÓN DE ENERGÍA (Elaboración Propia)	01 und.	S/ 39.96 (elaboración Propia)	S/ 39.96 (elaboración Propia)
PINES DE CONEXION		S/ 5.00	S/ 5.00
FOCO INCANDESCENTE DE 100 WATT	03 und	S/ 4.50	S/ 4.50
FOCO INCANDESCENTE DE 50 WATT	02 und.	S/ 3.00	S/ 3.00
INTERRUPTOR DIFERENCIA DE 32 AMPERIOS Y 33mamp de sensibilidad	01 und	S/ 50.00	S/ 50.00
MÓDULO DE PRUEBAS DE MADERA	01 und.	S/ 75.00	S/ 75.00
ARDUINO UNO	02 und.	S/ 50.11 (KuongShun elec. Makers Store Aliexpress)	S/ 80.00 (Naylampelectronics)
MÓDULO RELÉ 16 CH.	01 und.	S/ 43.23 (AliExpress CZB6721960 Store)	S/ 70.00 (MTLab S.A.C.)
MÓDULO RELÉ 8 CH.	01 und.	S/ 16.37 (ModuleFans Aliexpress)	S/ 45.00 (Naylampelectronics)



IMPRESIÓN A LACER BLANCO Y NEGRO.	15 und.	S/ 30.00	S/ 30.00
PLACA DE FIBRA DE VIDRIO BAÑADA EN COBRE	01 und (20 x 20 cm).	S/55.00	S/ 55.00
ÁCIDO FÉRRICO	01 botella 200ml	S/ 10.00	S/ 10.00
Amplificador operacional LM358P	30 und.	S/ 30.00	S/30.00
RESISTENCIA CERÁMICA DE 18 OHM	10 und.	S/ 0.50	S/ 0.50
CABLES DE CONEXIÓN	02 pack.	S/ 16.00	S/ 16.00
PLACA DE COBRE CON PERFORACIONES	01 und.	S/ 6.00	S/ 6.00
<b>TOTAL</b>		<b>S/ 583.23</b>	<b>S/ 847.96</b>