



TECNOLÓGICO
NACIONAL DE MÉXICO



**TRABAJO PROFESIONAL
COMO REQUISITO PARA OBTENER EL TITULO DE:
INGENIERO ELECTRONICO**

QUE PRESENTA:

ALBERTO DANIEL GÓMEZ MORALES

CON EL TEMA:

**“DETECCIÓN Y MONITOREO DE FALLAS EN
TRANSFORMADORES ELECTRICOS DE
DISTRIBUCIÓN VÍA WIFI”**

MEDIANTE:

**INFORME TECNICO DE RESIDENCIA
PROFESIONAL**

AGRADECIMIENTOS

Sabiendo que nunca podré pagar todo lo que hacen por mi les agradezco **a mis padres, mi hermana y mi abuelita** por ser el motor principal para cumplir mis sueños, por que gracias a su cariño, guía y apoyo he llegado a realizar uno de mis anhelos mas grandes en la vida, fruto del inmenso apoyo, amor y confianza que en mi depositaron y con los cuales he logrado terminar mis estudios profesionales siendo la mejor herencia que pude recibir y por lo cual les viviré eternamente agradecido.

Gracias **a dios** por la vida que me ha dado, y por estar presente no solo en esta etapa tan importante de mi vida, sino en todo momento ofreciéndome lo mejor para mi persona.

Gracias **al departamento de distribución** de Comisión Federal de Electricidad, por haberme permitido realizar mi residencia profesional en sus instalaciones, y aportar a mi formación profesional.

Gracias al **Instituto Tecnológico de Tuxtla Gutiérrez** por haberme formado como ingeniero, y por la educación que me dio para ser un mejor ser humano.

RESUMEN

El presente documento de Informe técnico de residencia profesional, denominado “Detección y monitoreo de fallas en transformadores eléctricos de distribución vía Wifi” en él se detallan la metodología y las actividades realizadas para llevar a cabo el proyecto, Dicho sistema se desarrolló con microcontroladores y se ocuparon diferentes lenguajes de programación ya que, en los requisitos del proyecto pedía el apoyo de un foro web, para la visualización y representación de los datos adquiridos.

La problemática se genera, cuando un transformador deja de funcionar en una cierta área y las personas se ven afectadas, al momento de que el transformador deja de funcionar para darle solución a este problema esperan a que el usuario reporte esta falla y los encargados de restablecer el servicio tienen que ir buscando cual transformador es el afectado, ya que no tienen un indicador que les permita identificarlo.

En el documento se plasma el proceso y las actividades que realice en un periodo de casi 5 meses, donde tuve que aplicar los conocimientos adquiridos en el ámbito escolar, donde analice, investigue y saque conclusiones de la problemática para dar la mejor solución.

INDICE DE CONTENIDO

CAPITULO I INTRODUCCION

CAPITULO I INTRODUCCIÓN	1
1.1 INTRODUCCIÓN	1
1.2 DESCRIPCIÓN DE LA EMPRESA	2
1.3 MISIÓN	2
1.4 VISIÓN	2
1.5 LOCALIZACIÓN	2
1.6 DESCRIPCIÓN DEL ÁREA DONDE SE REALIZÓ EL PROYECTO	3
1.7 PROBLEMAS A RESOLVER	3
1.8 OBJETIVOS	4
1.8.1 Objetivo General	4
1.8.2 Objetivos específicos	5
1.9 JUSTIFICACIÓN	5

CAPITULO II MARCO TEÓRICO

CAPITULO II MARCO TEÓRICO	6
2.1 MONITOREO EN LINEA	6
2.2 TRANSFORMADORES ELECTRICOS DE DISTRIBUCIÓN	7
2.3 TIPOS DE TRANSFORMADORES	8
2.4 ARDUINO UNO	9
2.5 COMO FUNCIONA ARDUINO	10
2.6 POR QUÉ USAR ARDUINO	10
2.7 ENTORNO DE DESARROLLO EN ARDUINO	11
2.8 ESTRUCTURA GENERAL DE UN SKETCH	12
2.9 MÓDULO SHIELD ETHERNET	14
2.10 CHIP WIZNET W5100	15
2.11 MÓDULO SIM800L GSM/GPRS	16
2.12 MÓDULO GPS SERIAL NEO-6M	17
2.13 SENSOR DE CORRIENTE SCT-013-000 100A 50mA	17

2.14 IMPRESIÓN 3D.....	18
2.15 SOLIDWORKS.....	19
2.16 PANTALLA LCD 20x4 CON I2C.....	20

CAPITULO III PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

CAPITULO III PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.	21
3.1 BUSQUEDA BIBLIOGRAFICA.....	21
3.2 DISEÑO DEL SISTEMA ELECTRÓNICO PARA RECABAR LA INFORMACIÓN NECESARIA	28
3.2.1 SENSOR DE MEDICIÓN.....	28
3.2.2 OBTENCIÓN DE COORDENADAS GEOGRÁFICAS.....	31
3.2.3 COMUNICACIÓN EN CASO DE FALLA.	34
3.2.4 CARCASA DE SENSORES DE MEDICIÓN.....	38
3.3 ELEGIR EL FORO (PÁGINA WEB) MÁS VIABLE PARA LAS NECESIDADES DEL SISTEMA Y EL ACONDICIONAMIENTO DE LAS SEÑALES PARA TENER UNA BUENA LECTURA DE ESTAS.	43
3.3.1 PROCESO PARA LA VISUALIZACIÓN DE DATOS EN UBIDOTS.....	44
3.3.2 ACONDICIONAMIENTO DE LAS SEÑALES.....	47
3.4 PROGRAMACIÓN DEL SISTEMA, PARA LA PRESENTACIÓN Y OBTENCIÓN DE LOS DATOS RECABADOS, ASÍ COMO ESTABLECER EL PROTOCOLO DE COMUNICACIÓN.....	51
3.4.1 PROTOCOLO DE COMUNICACIÓN.....	51
3.4.2 PROGRAMACIÓN DEL CIRCUITO RECEPTOR DE DATOS (Rx).....	51
3.4.3 PROGRAMACIÓN DEL CIRCUITO TRANSMISOR (Tx).....	54

CAPITULO IV RESULTADOS Y CONCLUSIONES

CAPITULO IV RESULTADOS Y CONCLUSIONES	61
COMPETENCIAS DESARROLLADAS	65
FUENTES DE INFORMACION	66
ANEXOS	68

INDICE DE FIGURAS

FIGURA 1.1	2
FIGURA 1.2	3
FIGURA 2.1	7
FIGURA 2.2	8
FIGURA 2.3	9
FIGURA 2.4	11
FIGURA 2.5	12
FIGURA 2.6	14
FIGURA 2.7	15
FIGURA 2.8	15
FIGURA 2.9	16
FIGURA 2.10	17
FIGURA 2.11	18
FIGURA 2.12	19
FIGURA 2.13	19
FIGURA 2.14	20
FIGURA 2.15	20
FIGURA 3.1	21
FIGURA 3.2	22
FIGURA 3.3	27
FIGURA 3.4	27
FIGURA 3.5	28
FIGURA 3.6	29
FIGURA 3.7	30
FIGURA 3.8	30
FIGURA 3.9	33
FIGURA 3.10	34
FIGURA 3.11	34
FIGURA 3.12	37

FIGURA 3.13	38
FIGURA 3.14	38
FIGURA 3.15	38
FIGURA 3.16	39
FIGURA 3.17	39
FIGURA 3.18	39
FIGURA 3.19	40
FIGURA 3.20	40
FIGURA 3.21	40
FIGURA 3.22	41
FIGURA 3.23	41
FIGURA 3.24	42
FIGURA 3.25	42
FIGURA 3.26	44
FIGURA 3.27	45
FIGURA 3.28	45
FIGURA 3.29	45
FIGURA 3.30	46
FIGURA 3.31	46
FIGURA 3.32	46
FIGURA 3.33	46
FIGURA 3.34	47
FIGURA 3.35	48
FIGURA 3.36	48
FIGURA 3.37	49
FIGURA 3.38	49
FIGURA 3.39	50
FIGURA 3.40	50
FIGURA 3.41	50
FIGURA 3.42	53
FIGURA 3.43	60

FIGURA 3.44	60
FIGURA 4.1	61
FIGURA 4.2	62
FIGURA 4.3	62
FIGURA 4.4	63
FIGURA 7.1	72
FIGURA 7.2	73
FIGURA 7.3	73
FIGURA 7.4	74
FIGURA 7.5	74
FIGURA 7.6	75
FIGURA 7.7	75
FIGURA 7.8	76
FIGURA 7.9	76

INDICE DE TABLAS

TABLA 1.1	3
TABLA 3.1	21
TABLA 3.2	23
TABLA 3.3	29
TABLA 5.1	65
TABLA 7.1	68
TABLA 7.2	69
TABLA 7.3	70
TABLA 7.4	72

CAPITULO I INTRODUCCIÓN

1.1 INTRODUCCIÓN

La etapa final de nuestra formación académica es la residencia profesional, esta es una parte primordial para nuestra formación ya que, al ser próximos a egresar, no tenemos experiencia laboral como verdaderos profesionistas en una empresa real, donde se enfrentan a desafíos diariamente, al realizar la residencia en una empresa nos trae ciertos beneficios ya que obtenemos experiencia para relacionarnos con diferentes personas en el ámbito laboral.

El *monitoreo* es una herramienta de gestión para controlar el avance de los proyectos en ejecución, La idea básica de esta herramienta es comparar su desempeño efectivo con lo planeado y medir los resultados reales en función de los previstos.

Una vez establecido, el monitoreo le ahorra tiempo y trabajo al encargado del proyecto, y facilita el control y preparación de los reportes correspondientes.

El monitoreo nos ayuda a comprender y explicar a los demás lo que sucede en el proyecto y por qué suceden los resultados obtenidos con esto obtenemos mayores oportunidades para ofrecer un mejor servicio a la población beneficiada con este servicio.

El presente proyecto trata sobre el desarrollo, construcción e implementación de un sistema de monitoreo de fallas en transformadores eléctricos de distribución, el cual tendrá como principal característica detectar cuando haya una falla al momento de la distribución eléctrica en la población.

El presente documento se divide en 4 etapas:

En la primera etapa se realizará una búsqueda bibliográfica para conocer las fallas eléctricas que más ocurren en los transformadores de distribución y así tomar consideraciones en la realización del proyecto.

En la segunda etapa se diseñará el sistema electrónico para recabar la información necesaria, utilizando principalmente sensores para la detección de fallas, tendremos que tener diferentes consideraciones como el voltaje, la corriente y la temperatura ya que por ser transformadores de distribución manejan altas cantidades de estas y por ultimo la intemperie ya que el dispositivo estará expuesto a diferentes condiciones climatológicas.

En la tercera etapa se elegirá la página web más viable para las características de nuestro sistema, donde busquemos tener una buena visualización de nuestros datos recibidos y que sean fáciles de interpretar.

En la cuarta etapa se realizará la programación del sistema, para la presentación de los datos hacia el foro, así como se establecerá el protocolo de comunicación que habrá entre el microcontrolador y el foro web, para la presentación de los datos.

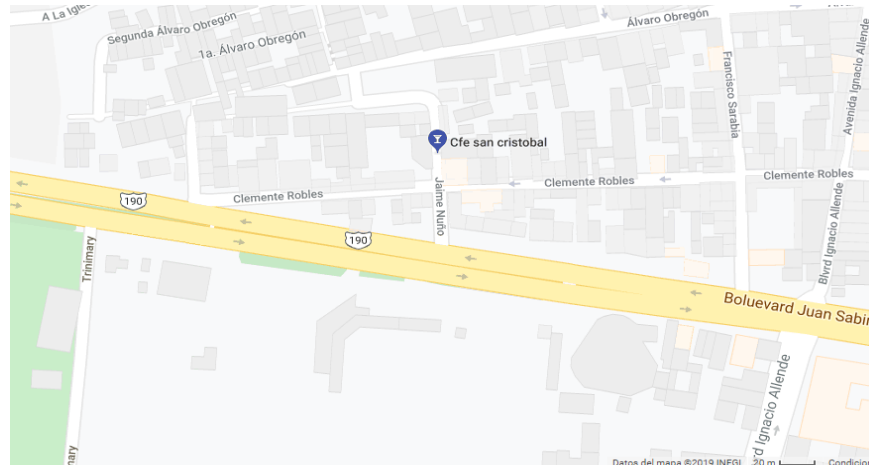


Figura 1.2 Localización CFE Distribución y Mantenimiento.

1.6 DESCRIPCIÓN DEL ÁREA DONDE SE REALIZÓ EL PROYECTO

El proyecto detección y monitoreo de fallas en transformadores eléctricos de distribución vía wifi, se realizó en el centro de Distribución y Mantenimiento de San Cristóbal de las casas, Chiapas de Comisión Federal de Electricidad División sureste, el cual cuenta con oficinas amplias donde es posible trabajar cómodamente, el cual cuenta con el equipo para realizar las pruebas necesarias, y teniendo herramientas como el acceso a internet para poder hacer las pruebas necesarias para el desarrollo del proyecto, este se pudo realizar de buena manera ya que se tenía un ambiente de trabajo bueno y cómodo.

1.7 PROBLEMAS A RESOLVER

Tabla 1.1 Cronograma de actividades a realizar.

Actividad	Semana															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1 Búsqueda Bibliográfica																
2 Diseño del sistema electrónico para recabar la información necesaria.																
3 Elegir el foro (página Web) más viable para las necesidades del sistema y el acondicionamiento de las señales para tener una buena lectura de estas.																

1.8.2 Objetivos específicos.

- Diseñar el sistema electrónico para recabar la información necesaria, utilizando el microcontrolador más viable.
- Diseñar el circuito que será capaz de acoplar la información que nos entregaran los sensores.
- Diseñar el circuito que será el encargado de acoplar las señales para la buena lectura de la información que llegará al microcontrolador.
- Implementar el protocolo de comunicación Wifi que se establecerá entre el microcontrolador y la página Web para la presentación de los datos.
- Programar el sistema electrónico para la presentación y obtención de los datos recabados.
- Fortalecer conocimientos en el área de microcontroladores, programación y comunicación de datos ya que es de vital importancia para la realización de un sistema de monitoreo.
- Fortalecer conocimientos en el área de distribución eléctrica para conocer los riesgos y limitaciones a los que el sistema será sometido.

1.9 JUSTIFICACIÓN

Este proyecto ha sido diseñado para aplicar las nuevas herramientas tecnologías y establecer nuevos protocolos de comunicación para la presentación de datos en foros web de forma remota, un transformador eléctrico su principal tarea es suministrar energía eléctrica de forma ininterrumpida a los usuarios como son las industrias, sector público, hogares, etc. Así como ofrecer calidad en el servicio eléctrico. Por estos motivos, debe existir un monitoreo constante y que sea el adecuado para el transformador de distribución.

Los transformadores eléctricos de distribución son elementos de suma importancia para el sistema eléctrico en los diferentes niveles de tensión, esta es una de las razones que me motiva a diseñar un sistema con nuevas técnicas que sea capaz de interpretar los datos entregados por el transformador, con el propósito de ofrecer un mejor servicio a los usuarios y de ocurrir alguna falla en el transformador poder restablecer la energía eléctrica en el menor tiempo posible ya que si la falla no es atendida en un margen de tiempo adecuado, traerá consigo pérdidas significativas en lo económico a la empresa e incluso ambientales y en caso de ser necesario aplicar acciones de mantenimiento correctivas al transformador que se esté monitoreando, ya que muchas veces por su ciclo de vida o por acciones de la naturaleza suceden fallas imprevistas.

CAPITULO II MARCO TEÓRICO

Durante su operación, los transformadores de potencia están sujetos a esfuerzos térmicos, eléctricos y mecánicos, los cuales degradan el sistema de aislamiento aceite/papel. Las causas principales de la degradación del sistema aislante aceite/papel son temperatura excesiva, la presencia de oxígeno y humedad combinadas con los esfuerzos eléctricos, los cuales actúan como acelerados del proceso de degradación. Las causas secundarias que aceleran el envejecimiento del sistema aislante son los esfuerzos mecánicos (originados por la excitación del núcleo y cortos circuitos) y los productos de descomposición del aceite (ácidos y lodos).

El proceso de degradación del sistema aislante aceite/papel evoluciona gradualmente hasta presentarse la falla en el transformador de potencia, que en ocasiones puede ser catastrófica. La degradación de transformadores de potencia presenta ciertos síntomas que indican la presencia de problemas. La detección oportuna de una degradación incipiente puede ser la diferencia entre someter el transformador a una reparación mayor o sustituir una pieza dañada.

La detección de degradación incipiente se efectúa mediante el monitoreo de ciertos parámetros clave que sirven para diagnosticar la condición del sistema aceite/papel. El monitoreo de equipo de potencia puede realizarse en línea o fuera de línea. La tendencia mundial en el ámbito de detección de degradación incipiente en equipo de potencia es la aplicación de técnicas de monitoreo y diagnóstico en línea.

En México, la estadística de fallas de transformadores de potencia de Comisión Federal de Electricidad, Indica que el 49% de las fallas se refieren a problemas de aislamiento en devanados, 26% a boquillas, 10% al cambiador de derivaciones, 3% a explosiones con incendio, 2% al núcleo y 10% a otras causas.

2.1 MONITOREO EN LINEA

Un sistema de monitoreo en línea consta principalmente de sensores, un sistema de adquisición de datos, un protocolo de comunicación el cual será el medio de comunicación entre el sistema de monitoreo y la presentación de datos y una foro web, El monitoreo es la base para realizar el diagnóstico de la condición de los transformadores de potencia, el monitoreo en línea permite diagnosticar problemas en transformadores de potencia de manera efectiva teniendo acceso de forma remota a esta información. Los parámetros importantes a monitorear continuamente en un transformador de potencia son la carga, ya que estos deben de operar ininterrumpidamente, El monitoreo de la carga y de las condiciones de operación consiste básicamente en la medición de las tensiones y corrientes a la que están diseñados.

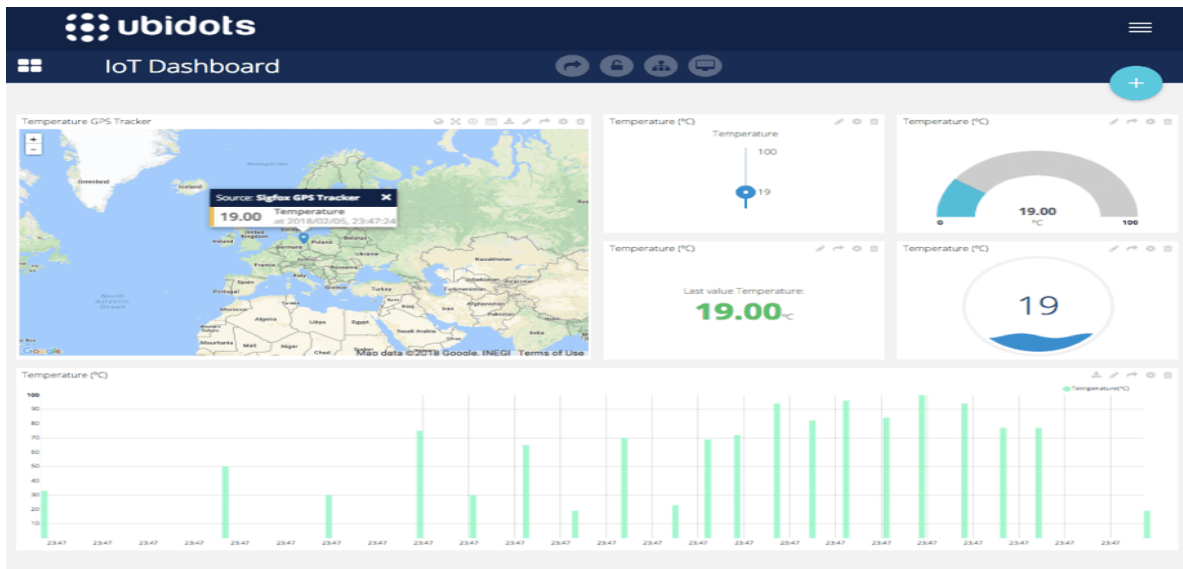


Figura 2.1 Ejemplo de foro web donde se monitorea la temperatura con diferentes presentaciones de los datos adquiridos.

2.2 TRANSFORMADORES ELECTRICOS DE DISTRIBUCIÓN.

El transformador es un dispositivo que permite elevar o disminuir el voltaje en un circuito por medio de un campo magnético, manteniendo una misma potencia, el funcionamiento de un transformador se basa en el principio de inducción electromagnética. El transformador se compone de dos bobinas, con distintas cantidades de vueltas. Ambas bobinas están unidas por un material ferromagnético para disminuir las pérdidas del transformador.

Se aplica un voltaje de corriente alterna al devanado primario, lo que genera en este un campo magnético, que se traslada a través del material ferromagnético al devanado secundario. Al ser un campo magnético variable (debido a la corriente alterna) genera en el devanado secundario una Fem (Fuerza electromotriz). Este voltaje va depender de 3 factores:

1. La cantidad de vueltas que tiene el devanado primario (N1)
2. La cantidad de vueltas que tiene el devanado secundario (N2)
3. El voltaje aplicado al devanado primario

Es muy probable que en todos lados donde encontremos energía eléctrica, haya previamente un transformador que nos esté proporcionando la energía con el potencial necesario, es por eso que el uso de un transformador es prácticamente universal, de igual forma a continuación detallaremos alguno de los usos más comunes de estos:

- **Para distribución de energía:** Es mucho más eficiente transportar la energía con alto potencial y baja intensidad. Es por eso que se utilizan los transformadores para elevar el potencial a alta tensión. Sin embargo, en nuestros hogares tenemos corriente de baja tensión. Por lo que también se necesitan transformadores para pasar de alta a media y baja tensión.

- **Para protección de maquinaria eléctrica:** En las industrias, los transformadores son muy utilizados para proteger y aislar los equipos eléctricos, controlando los pulsos de energía.
- **Para generar altos voltajes:** Los transformadores son muy utilizados en el ámbito ferroviario para hacer mover las maquinarias que necesitan de un voltaje alto para funcionar correctamente.



Figura 2.2 Transformador para distribución de energía eléctrica.

2.3 TIPOS DE TRANSFORMADORES.

Existen distintos tipos de transformadores según la aplicación que se le dé. A continuación, detallaremos cada uno de ellos.

- **Transformador elevador/Reductor de tensión:** Son utilizados por las empresas generadoras de electricidad para transportar a altas tensiones y que las casas pueden recibir bajas tensiones.
- **Transformador variable:** Para una entrada de tensión fija, se puede variar la tensión de salida.
- **Transformador de aislamiento:** Contiene un aislamiento galvánico, y se utilizan para proteger equipos que están conectados directamente a la red.
- **Transformador de alimentación:** Puede tener una o más bobinas secundarias. Incorporan un fusible térmico que permite proteger los equipos de sobrecargas.
- **Transformador de pulsos:** Transformador que trabaja en régimen de pulsos. Tiene una rápida velocidad de respuesta. Se utiliza para transferir impulsos a elementos de control.
- **Transformador diferencial de variación lineal:** Es un transformador utilizado para medir desplazamientos que son lineales. Son utilizados en los servomecanismos para dar una retroalimentación de la posición.
- **Transformador con diodo dividido:** Compuesto por diodos repartidos por todo el bobinado conectados en serie que permiten proporcionar una tensión continua.

- **Transformador de impedancia:** Es el transformador encargado de adaptar antenas y líneas de transmisión. Son encargados de disminuir o aumentar la impedancia.
- **Estabilizador de tensión:** Este funciona normalmente mientras la tensión no supere un valor nominal. En caso de que la tensión supere este valor, el núcleo se satura sin dejar pasar el exceso de tensión. Protege a los equipos de sobrecargas.

Partes de un transformador: Las partes que componen un transformador son.

- **Bobinado primario:** Encargado de recibir la tensión a transformar y convertirla en un flujo magnético.
- **Núcleo del transformador:** Encargado de transportar el flujo magnético a la bobina secundaria.
- **Bobina secundaria:** Encargada de transformar el flujo magnético en una diferencia de potencial requerida.

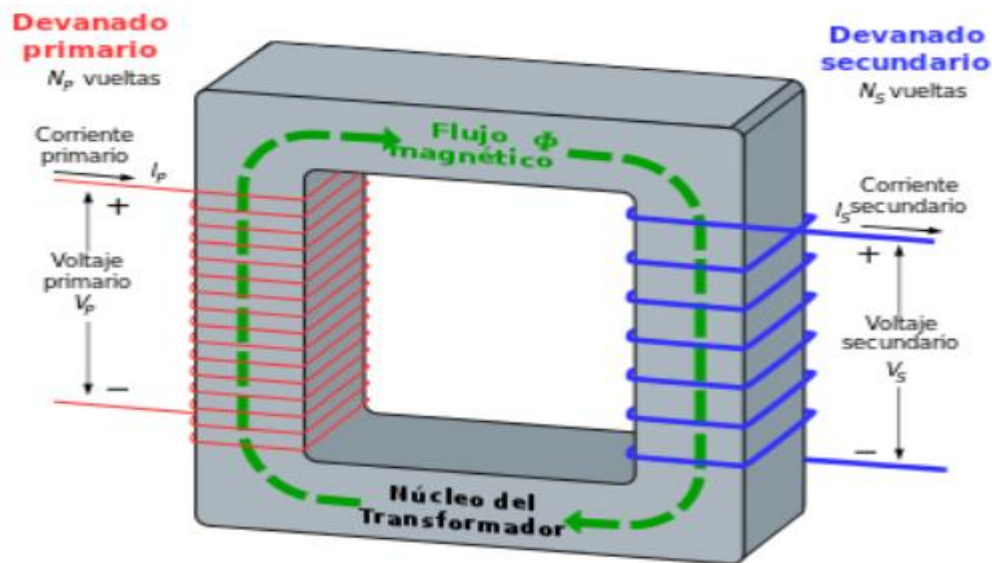


Figura 2.3 Diagrama básico de un transformador.

2.4 ARDUINO UNO.

Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas, pero igualmente funcionales al partir de la misma base

El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

Arduino nació en 2005, cuando varios estudiantes del Instituto de Diseño Interactivo de Ivrea, Italia, con el fin de facilitar el acceso y uso de la electrónica de programación. Lo hicieron para que los estudiantes de electrónica tuviesen una alternativa más económica a las populares BASIC STAMP.

2.5 COMO FUNCIONA ARDUINO.

El Arduino es una placa basada en un microcontrolador ATMEL, los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación que puedes utilizar en el entorno Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

El Arduino posee lo que se llama una interfaz de entrada, que es una conexión en la que podemos conectar en la placa diferentes tipos de periféricos. La información de estos periféricos que se conecten se trasladara al microcontrolador, el cual se encargara de procesar los datos que le lleguen a través de ellos.

También cuenta con una interfaz de salida, que es la que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos periféricos pueden ser pantallas donde se puedan presentar los datos procesados, pero también pueden ser otras placas o controladores.

Arduino es un proyecto y no un modelo concreto de placa, lo que quiere decir que compartiendo su diseño básico uno se puede encontrar con diferentes tipos de placas. Las hay de varias formas, tamaños y colores para las necesidades del proyecto en el que estés trabajando, las hay sencillas o con características mejoradas.

2.6 POR QUÉ USAR ARDUINO.

- Arduino es libre y extensible, esto quiere decir que cualquiera que desee ampliar y mejorar el diseño hardware de las placas como el entorno de desarrollo, puede hacerlo sin problemas. Su entorno es multiplataforma, se puede instalar y ejecutar en sistemas operativos Windows, Mac OS y Linux.
- Lenguaje de programación de fácil comprensión, su lenguaje de programación basado en C++ es de fácil comprensión que permite una entrada sencilla a los nuevos programadores y a la vez con una capacidad tan grande, que los programadores más avanzados pueden expresar todo el potencial de su lenguaje y adaptarlo a cualquier situación.

- Reusabilidad y versatilidad porque una vez terminado el proyecto es muy fácil poder desmontar los componentes externos a la placa y empezar con un nuevo proyecto, de igual manera todos los pines del microcontrolador están accesibles a través de conectores hembra, lo cual permite sacar partido de todas las bondades del microcontrolador con un riesgo muy bajo de hacer conexión errónea.

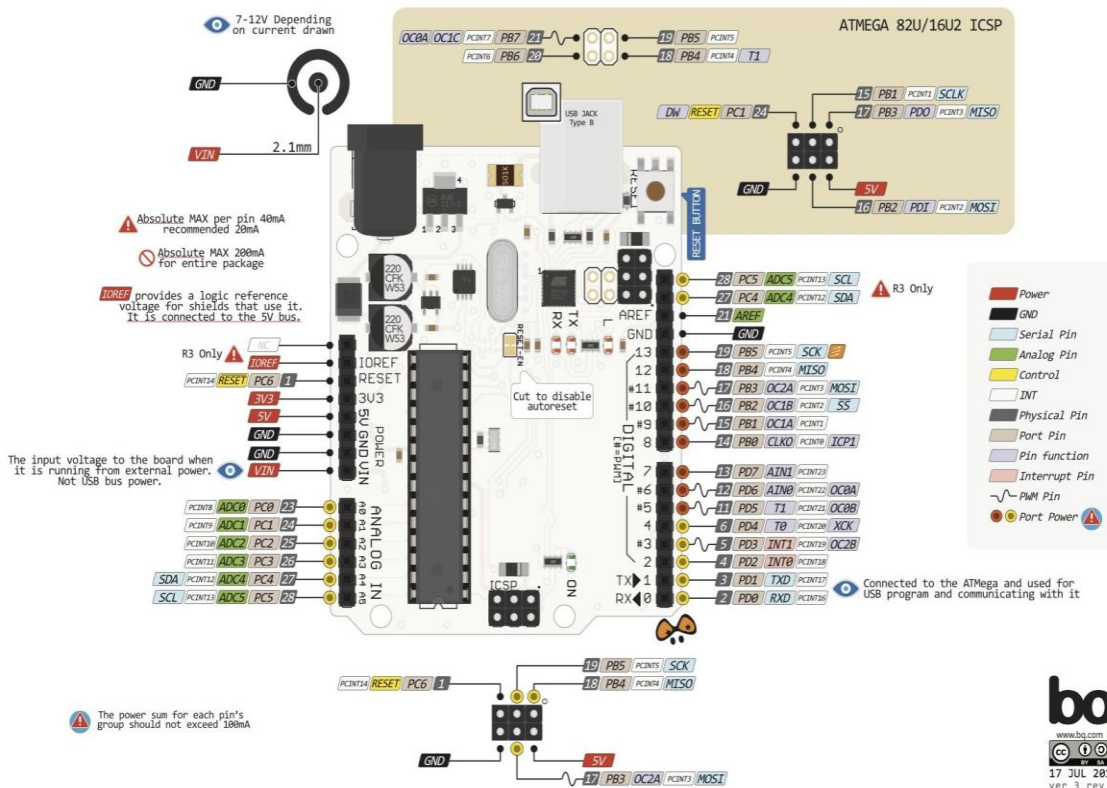


Figura 2.4 Diagrama esquemático del Arduino Uno.

2.7 ENTORNO DE DESARROLLO EN ARDUINO.

El entorno de desarrollo en Arduino (IDE) es el encargado de la gestión de la conexión entre el PC y el hardware de Arduino con el fin de establecer una comunicación entre ellos por medio de la carga de programas. El IDE de Arduino se compone de:

Un editor de texto: donde escribir el código de programa.

Un área de mensajes: a través del cual el usuario tendrá constancia en todo momento de los procesos que se encuentren en ejecución, errores en código, problemas de comunicación, etc.

Una consola de texto: mediante la que podremos comunicarnos con el hardware Arduino y viceversa.

Una barra de herramientas: donde podremos acceder a una serie de menús y a los botones con acceso directo a las principales funcionalidades de Arduino.

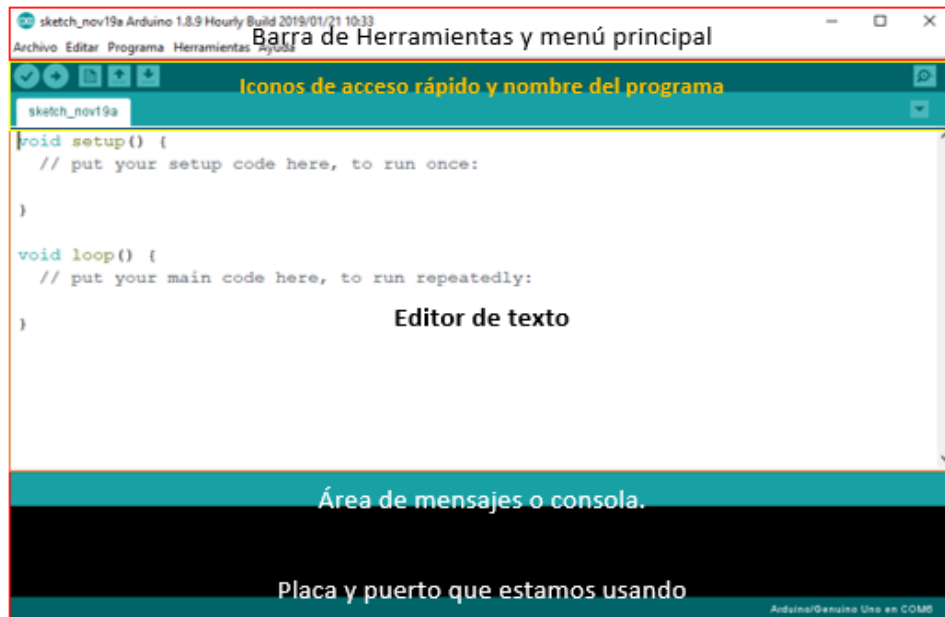


Figura 2.5 Entorno de desarrollo de Arduino.

2.8 ESTRUCTURA GENERAL DE UN SKETCH.

Un programa diseñado para ejecutarse sobre un Arduino, un sketch se compone de tres secciones:

La sección de declaraciones de variables globales: Ubicada directamente al principio del código.

La sección llamada Void setup (): Delimitada por llaves de apertura y cierre, donde las instrucciones escritas dentro de esta sección se ejecutan una única vez, en el momento de encender la placa Arduino.

La sección llamada Void Loop (): Delimitada por llaves de apertura y cierre. Las instrucciones escritas dentro se ejecutan infinitas veces hasta que la placa se apague (o se resetee). Es en este punto donde está el programa en sí y que funciona continuamente.

Es decir, el contenido de “Void Loop ()” se ejecuta desde la 1ª instrucción hasta la última, para seguidamente ejecutarse desde la 1ª instrucción hasta la última, y así una y otra vez.

Por tanto, las instrucciones escritas en la sección “Void setup ()” normalmente sirven para realizar ciertas preconfiguraciones iniciales y las instrucciones del interior de “Void Loop ()”.

Las entradas y salidas digitales.

La placa Arduino dispone de 14 pines-hembra de entradas o salidas (según lo que convenga) digitales, numeradas desde la 0 hasta la 13. A veces a estos pines hembra digitales de “Propósito general” se les llama pines GPIO (de “General Purpose Input/Output”).

Las funciones que nos ofrece el lenguaje Arduino para trabajar con entradas y salidas digitales son:

pinMode (): Configura un pin digital (cuyo número se ha de especificar como primer parámetro) como entrada o como salida de corriente, según si el valor de su segundo parámetro es la constante predefinida INPUT o bien OUTPUT, respectivamente. Esta función es necesaria porque los pines digitales a priori pueden actuar como entrada o salida, pero en nuestro sketch hay que definir previamente si queremos que actúen de una forma u de otra. Es por ello que esta función se suele escribir dentro del “setup ()”. No teniendo valor de retorno.

digitalWrite (): Envía un valor ALTO (HIGH) o bajo (LOW) a un pin digital; es decir, tan solo es capaz de enviar dos valores posibles. Por eso, de hecho, hablamos de salida “digital”. El pin al que se le envía la señal se especifica como primer parámetro (escribiendo su número) y el valor concreto de esta señal se especifica como segundo parámetro (escribiendo la constante predefinida HIGH o bien la constante predefinida LOW, ambas de tipo “int”).

digitalRead (): Devuelve el valor leído del pin digital (configurado como entrada mediante pinMode ()) cuyo número se haya especificado como parámetro. Este valor de retorno es de tipo “int” y puede tener dos únicos valores: el valor HIGH o el valor LOW.

Las funciones que tenemos para trabajar con entradas y salidas analógicas son:

Tenemos 6 pines de E/S digitales numeradas del A0 al A5, al ser estos analógicos, siempre son de lectura (No necesita la instrucción pinMode).

analogRead (): Lee el valor del pin analógico especificado. La placa Arduino contiene 6 canales (8 canales en el Mini y Nano, 16 en el Mega), analógicos con convertidor analógico a digital de 10 bit. Esto significa que mapeara tensiones de entre 0 y 5 voltios en valores enteros entre 0 y 1023. Esto produce una resolución entre las lecturas de $5 \text{ volts}/1024 = 0.0049 \text{ volts}$. El rango de entrada y la resolución se pueden cambiar con **analogReference ()**. Se tarda unos 100 microsegundos para leer una entrada analógica, por lo que la velocidad de lectura máxima es de alrededor de 10,000 veces por segundo.

analogWrite (): La función de Arduino analogWrite permite hacer uso del módulo de PWM, este módulo puede generar una señal de una frecuencia de 980 Hz y modular el ciclo de trabajo (CT). El propósito de esta función es generar una señal de 980 Hz al CT configurado. De hecho, a señal generada se mantendrá en el pin seleccionado hasta cualquiera de las siguientes condiciones: se ejecute otra instrucción analogWrite, se utilice digitalWrite o digitalRead sobre el mismo pin.

Pines de alimentación: La forma más sencilla de alimentar el Arduino UNO es a través del puerto USB. Sin embargo, una vez que deseamos colocar nuestra tarjeta Arduino en su aplicación final, el puerto USB puede no ser la forma más óptima de alimentarla.

La placa Arduino admite varias formas para ser alimentada como serían:

- **Alimentación por USB**
- **Alimentación por medio del Jack externo**
- **Alimentación por medio del pin Vin.**

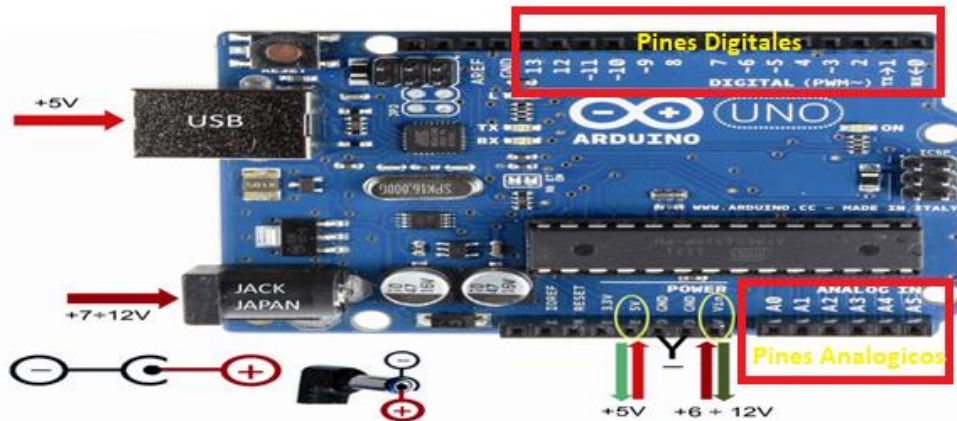


Figura 2.6 Configuración de los pines del Arduino Uno.

2.9 MÓDULO SHIELD ETHERNET.

El módulo Arduino Shield Ethernet conecta la placa Arduino a internet rápidamente, Se basa en el chip de ethernet Wiznet W5100, El chip Wiznet W5100 ofrece una red (IP) capaz de usar TCP y UDP, soporta hasta cuatro conexiones de sockets simultáneos. El módulo Arduino Shield Ethernet tiene una conexión RJ-45 estándar, con un transformador de línea integrado cuenta también con una ranura para tarjeta micro-SD, que se puede utilizar para almacenar archivos para servir a través de la red.

Arduino se comunica tanto con el chip W5100 como con la tarjeta SD usando el bus SPI (a través de la cabecera ICSP). Para ello utiliza los pines digitales 10,11,12 y 13 en las placas Arduino/Genuino Uno y los pines 50,51 y 52 en el Mega. En ambas placas, el pin 10 se utiliza para seleccionar el chipW5100 y el pin 4 para la tarjeta SD. Estos pines no se pueden utilizar para otras entradas o salidas.

El módulo Shield Ethernet contiene una serie de leds informativos:

- **PWR:** Indica que la placa y el módulo están encendidos.
- **LINK:** Indica la presencia de un enlace de red y parpadeo cuando el módulo transmite o recibe datos.
- **FULL:** Indica que la conexión de red es full dúplex.
- **100M:** Indica la presencia de una conexión de red de Mb/s 100 (en contraposición a 10 Mb/s).
- **RX:** Parpadea cuando la Shield recibe datos.
- **TX:** Parpadea cuando la Shield envía datos.
- **COLL:** Parpadea cuando se detectan colisiones de red.



Figura 2.7 Conexión del Módulo Shield Ethernet a la placa Arduino.

2.10 CHIP WIZNET W5100.

Es un chip controlador de Ethernet, el que incluye la pila TCP/IP implementada por hardware, su principal campo de aplicación son sistemas embebidos con conectividad hacia internet.



Figura 2.8 Chip WIZnet W5100

Características:

- Implementa protocolos TCP, UDP, ICMP, IPV4, ARP, IGMPv2, PPPoE, Ethernet
- Soporta 8 sockets independientes simultáneos
- 10/BaseT/100BaseTx Ethernet PHY
- Auto negociación (Full-duplex/Half-duplex)
- Ancho de bus 16/8 bit
- Memoria interna de 128kbytes para comunicación de datos Tx/Rx
- Memoria SPI (Serial Peripheral Interface)
- Opera a 3.3V.

2.11 MÓDULO SIM800L GSM/GPRS.

En la parte principal del módulo hay un chip celular SIM800L GSM de SimCom. El voltaje de funcionamiento del chip es de 3.4V a 4.4V, lo que lo convierte en una buena opción para integrarse en proyectos sin mucho espacio.



Figura 2.9 Modulo SIM800L GSM/GPRS

Todos los pines de datos necesarios del chip SIM800L GSM se dividen en encabezados de paso de 0.1. Esto incluye los pines necesarios para la comunicación con un microcontrolador a través de UART. El módulo admite una velocidad de transmisión de 1200 bps a 115200 bps con detección automática de baudios.

En la parte posterior esta la entrada de SIM cualquier tarjeta micro SIM 2G activada funcionaria perfectamente. La dirección correcta para insertar la tarjeta SIM normalmente está grabada en la superficie del zócalo SIM.

Este modulo mide solo 1 pulgada cuadrada, pero incluye una sorprendente cantidad de características como son:

- Admite cuatribanda: GSM850, EGSM900, DCS1800 y PCS 1900
- Conectarse a cualquier red GSM global con cualquier SIM 2G
- Recibe y realiza llamadas de voz con un altavoz externo de 8Ω y un micrófono electret
- Enviar y recibir mensajes SMS
- Enviar y recibir datos GPRS (TCP/IP, HTTP, etc.)
- Escanea y reciba transmisiones de radio FM
- Conjunto de comandos AT basado en serie
- Conectores FL para antenas celulares
- Acepta tarjeta micro SIM.

Fuente de alimentación para el módulo SIM800L

Una de las partes más importantes para hacer funcionar el módulo SIM800L es suministrarle suficiente energía. Dependiendo de en qué estado se encuentre, el SIM800L puede ser un dispositivo que consume bastante energía. El consumo de corriente máximo del módulo es de alrededor de 2A durante la transmisión, Dado que el módulo SIM800L no viene con un regulador de voltaje incorporado, se requiere una fuente de alimentación externa ajustada al voltaje entre 3.4V a 4.4V. La fuente también debe poder generar 2A de sobretensión, de lo contrario, el módulo seguirá apagándose, Una opción para la alimentación del módulo.

2.12 MÓDULO GPS SERIAL NEO-6M.

El módulo GPS NEO-6M V2 esta hecho en base al receptor GPS NEO-6 de la marca Ublox, es usado con sistemas como Arduino, Raspberry Pi y otros microcontroladores. La board posee una memoria EEPROM que permite guardar los datos que se hayan configurado, además incluye una antena ceramica que se conecta directamente a la board por lo tanto el módulo se puede usar al instante sin necesidad de mas accesorios, con este módulo se pueden medir parámetros como latitud, longitud y velocidad, si solamente se desean recibir los datos NMEA basta con conectar el pin Tx con el Rx del Arduino y así recibir los datos que envía el módulo las características principales de este módulo son:

- Voltaje de alimentación: 3-5 VDC
- Interface: Serial
- Antena Cerámica
- EEPROM para guardar datos de configuración
- Indicador de señal con LED
- Tamaño de la antena: 25mm X 25mm
- Tamaño del módulo: 25mm X 35mm
- Velocidad de Transmisión: 9600bps.
- Compatibilidad con la mayoría de las librerías GPS

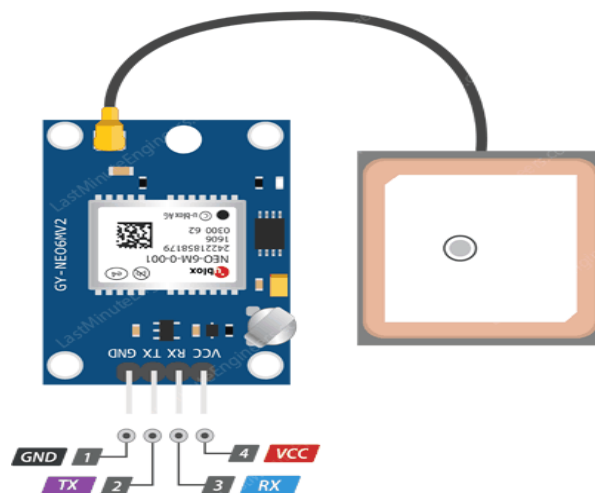


Figura 2.10 Modulo GPS Serial NEO-6M

2.13 SENSOR DE CORRIENTE SCT-013-000 100A 50mA.

Sensor de corriente SCT-013-000 100A 50mA. Este sensor tiene una salida de voltaje proporcional a la corriente que pasa por la bobina. Las transformadoras de corriente (TC) son sensores que se utilizan para medir la corriente alterna. Son particularmente útiles para medir el consumo de electricidad de un edificio entero (o generación según sea el caso). El tipo de núcleo dividido, es particularmente adecuado para el uso en proyectos, ya que se puede enganchar directamente a

cualquier cable de alta tensión o neutral sin tener que hacer ningún trabajo de alta tensión eléctrica, estas son algunas de las características con las que cuenta en el sensor:

- Corriente de entrada: 0~100A AC
- Modo de salida: 0~50mA
- No linealidad: $\pm 3\%$
- Relación: 100A:0.05A
- Resistencia de Grado: Grado B
- Temperatura de trabajo: -25°C ~ $+70^{\circ}\text{C}$
- Longitud del cable: 100cm
- Plug: 3.5mm
- Dimensiones: 36X57X21 mm



Figura 2.11 Sensor SCT-013-000 100A 50mA

2.14 IMPRESIÓN 3D.

La impresión 3D, también conocida como manufactura por adición, es un proceso por el cual se crean objetos físicos colocando un material por capas en base a un modelo digital. Todos los procesos de impresión 3D requieren que el software, el hardware y los materiales trabajen en conjunto.

La tecnología de impresión 3D puede utilizarse para crear todo tipo de cosas, desde prototipos y piezas simples hasta productos finales altamente técnicos, como piezas para aeronaves, edificios ecológicos, implantes médicos que pueden salvar vidas e incluso órganos artificiales que se producen con capas de células humanas (AUTODESK).

La mas importante diferencia que podemos encontrar entre las formas tradicionales de construcción de objetos y la impresión 3D es que mientras en el primer método el objeto a modelar se obtiene quitando el material sobrante, en la impresión 3D sólo se utiliza estrictamente el material a utilizar, lo que produce importantes ahorros, además de menor contaminación.

Con respecto a los materiales que emplean las impresoras 3D para crear los objetos, estos básicamente se dividen en termoplásticos PLA o ABS, los mas utilizados en las impresoras 3D más económicas del mercado, pero también podemos encontrar impresoras que modelen objetos tales como resinas, fotopolímeros y hasta incluso metal, pero el costo de este tipo de impresoras 3D es tan alto que es prácticamente imposible adquirir alguna de ellas para ser usada fuera del ámbito industrial.

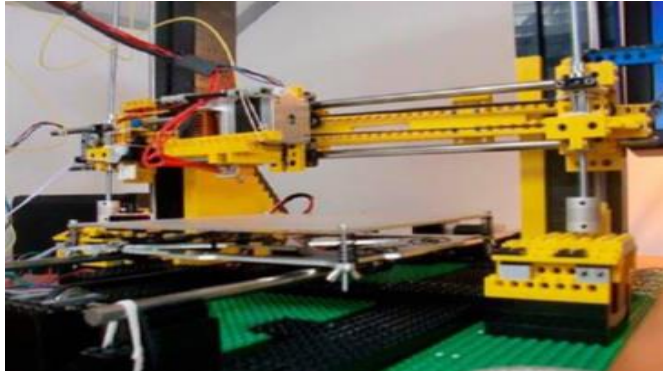


Figura 2.12 Ejemplo de una impresora 3D.

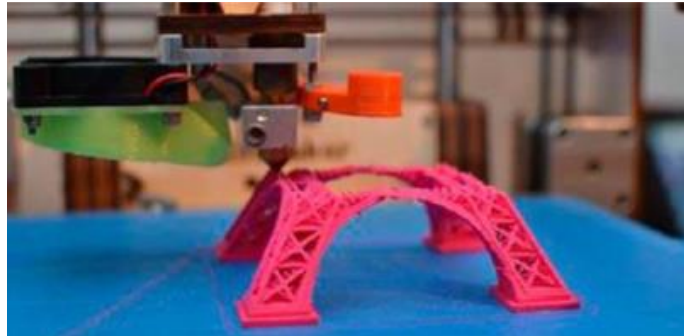


Figura 2.13 Impresión de una pieza en PLA.

2.15 SOLIDWORKS.

SolidWorks es un programa de diseño mecánico en 3D con el que puedes crear geometría 3D usando sólidos paramétricos, la aplicación está enfocada a diseño de producto, diseño mecánico, ensamblajes y dibujos para taller. SolidWorks diseña de forma que va dejando un historial de operaciones para que puedas hacer referencia a ellas en cualquier momento.

Con SolidWorks puedes diseñar piezas mecánicas en 3D, evaluar ensamblajes de varias piezas y producir dibujos de fabricación para el taller, además podrás manejar los datos de diseño en su sistema de administración PDM y llevar un control de las versiones de dibujos.

Al diseñar puedes evaluar el impacto ambiental del diseño, simular virtualmente las condiciones y análisis del diseño en situaciones reales y optimizar su desempeño. El programa está basado en un motor de modelado de sólidos y también contiene comandos de creación, edición de superficies complejas, además es asociativo entre los modelos 3D y sus dibujos.

Técnicas de diseño de superficies:

Las Herramientas de diseño de superficies avanzadas de SolidWorks aceleran el proceso de diseño, por lo que ahorran tiempo y costes de desarrollo, y aumentan la productividad.

SolidWorks simplifica el diseño de superficies avanzadas para hacer que la creación de partes sofisticadas con formas complejas sea más fácil. Dichos dispositivos son

habituales en una variedad de sectores entre los que se incluyen productos de consumo, vehículos, moldes y dispositivos médicos. SolidWorks nos proporciona:

- Herramientas para: Superficies extruidas, revueltas, barridas, recubiertas, equidistantes, planas.
- Herramientas para: Superficie limitante de formas orgánicas y estilizadas, relleno de superficies, superficies radiadas de forma libre y superficies medias.
- Herramientas de edición: mover/copiar cara, Instant3D (Haciendo Clic y arrastrar la geometría), copiar superficie, reemplazar cara, reparar, coser, recortar y extender.
- Herramientas de análisis de superficies: ángulo, corte sesgado, espesor, curvatura, radio mínimo, franjas de cebra, análisis de desviación y comprobación de simetría.
- Las herramientas de simulación de piezas de plástico y llenado de moldes están disponibles en SolidWorks Plastics.



Figura 2.14 Logotipo del Software de diseño SolidWorks

2.16 PANTALLA LCD 20x4 CON I2C.

Es una pantalla alfanumérica ideal para interactuar con un microcontrolador, ya que el microcontrolador nos entrega información en forma binaria, la pantalla posee un procesador capaz de interpretar este código e imprimirlo de manera alfanumérica. La pantalla esta compuesta por 4 filas y 20 columnas.

La comunicación con Arduino se lleva a través del módulo adaptador de LCD a I2C, el cual es un expansor de entradas y salidas digitales a través de la comunicación serial. La LCD se puede controlar con 2 pines del Arduino (SDA y SCL), el contraste de la pantalla entre los dígitos y el fondo es controlado a través del potenciómetro incluido en el módulo, basta con ajustarlo para obtener el contraste deseado, la luz de fondo se controla por software desde Arduino.

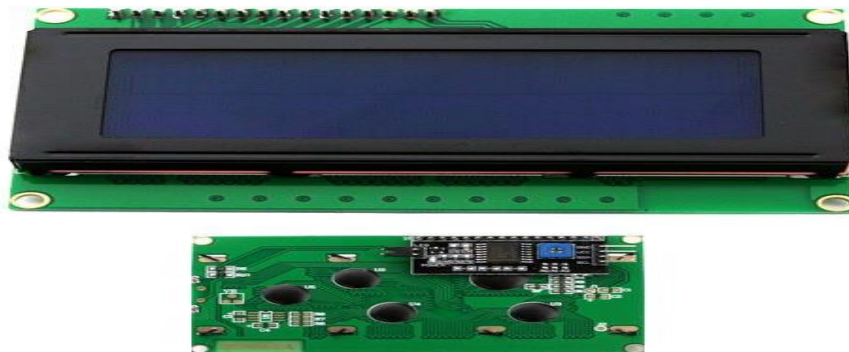


Figura 2.15 Pantalla LCD 20x4 con I2C.

CAPITULO III PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS.

3.1 BUSQUEDA BIBLIOGRAFICA.

Para el diseño del sistema electrónico que será el encargado de verificar cuando haya una falla en el transformador de distribución, fue necesario que realizara una búsqueda bibliográfica, ya que es necesario saber cuáles son las fallas más comunes en estos y el comportamiento de un transformador de distribución.

En base a la investigación realizada se obtuvieron los siguientes datos:

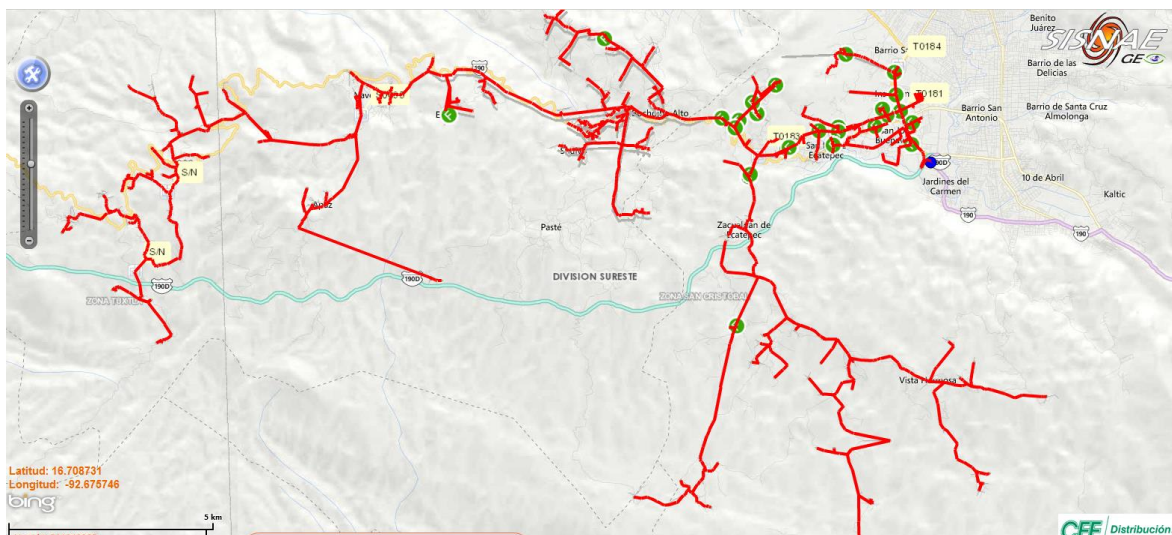


Figura 3.1 Fallas en el CRI04030 (San Cristóbal de las casas).

Esta imagen fue extraída del Sistema Nacional de Emergencias (SISNAE), donde se registran las fallas ocurridas en los circuitos de Comisión Federal de Electricidad, donde registra fallas en ramales y troncales, la línea marcada con rojo es la trayectoria que sigue el circuito y los puntos verdes con una F, indican donde ésta la falla.

Tabla 3.1 Tabla de Fallas en Troncales en el circuito 04030.

FALLAS EN DISTRIBUCION ELECTRICA Periodo (2019-01-01 a 2019-10-19)						
Zona	S.E	Circuito	Tipo	Causa	Daños	Observaciones
03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Transformador de distribución	LIC 203 RIM 087 (1-15/3), CAUSA ARBOL SOBRE LA LINEA DE MEDIA TENSION POR LOS FUERTES VIENTOS POR FRENTE FRIO NUM. 26 SE SECCIONA RAMAL FALLADO Y CIERRA RESTAURADOR LA RAZA SE RECUPERA 99% DE LA CARGA.
03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Transformador de distribución	LIC. 886 RIM 4439 RAMA SOBRE LA LINEA POR FUERTES VIENTOS ORIGINADOS POR FRENTE FRIO #40, SE SECCIONA EL TRAMO FALLADO SE ABREN CUCHILLAS T5000 DE SAN FELIPE, SE RETIRA ARBOL Y SE PROCEDE A CORREGIR TRANSF.

03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Transformador de distribución	LIC. 1532 RIM 0181 K0303330755, RAMA SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR EL FRENTE FRIO # 49, AVENIDA REVOLUCION, BARRIO DE FATIMA.
03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Transformador de distribución	LIC 1346 Y 1347 RIM 121 50ABC 2976AMP ARBOL SOBRE LINEAS POR FUERTES VIENTOS ORIGINADOS POR EL FRENTE FRIO #46, SE RETIRO ARBOL Y NORMALIZA RAMAL, CALZADA DEL CBTIS RAMAL A SAN FELIPE.
03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Transformador de distribución	LIC 302 RIM 100 RAMA SOBRES LINEA POR FUERTES VIENTOS ORIGINADOS POR FRENTE FRIO #29, SE RETIRA RAMAS Y NORMALIZA EL RAMAL, RESTAURANT LAS FLORES, NACHIG NAVENCHAUC CTO. CRI 4030.
03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Conductor ACSR	LIC 1418 Y 1422 RIM 180 50C 1428 AMP, ARBOL SOBRE LINEAS POR FUERTES VIENTOS ORIGINADOS POR EL FF48, D0057 ENL CRI4030_4060 RECUPERERA 99% DEL CTO; SE LEVANTA CONDUCTOR ACSR CON CONECTOR 3/0 CALZADA DEL CBTIS RAMAL A SAN FELIPE.
03-SAN CRISTOBAL	CRI	04030	TRONCAL	Frente Frio	Empalme Preformado	LIC. 4117 RIM 45454 RAMA SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR LA ONDA TROPICAL NUM #38, ENTRE LAS EST. 44 TIPOPS30/VS30/RD Y EST. 45 TD30/TD30, PERIFERICO NORTE, SE ABREN LAS T0031 Y SE CIERRA, RECUPERARND0 99% CRI 4030.

En la tabla 3.1 Se detallan los puntos importantes de cuando ocurre una falla, donde se pide la Zona, Subestación eléctrica (S.E), Circuito, Tipo, Causa, Daños, y Observaciones, en la tabla se observan 7 eventos ocurridos en un periodo de 10 meses, donde al ser un troncal el afectado el tiempo de respuesta debe ser inmediato ya que esta es una conexión primaria para la distribución eléctrica, para prevenir estas fallas se recurre al patrullaje del circuito en busca de ramas, fisuras en postes o degradación del herraje del transformador, ya que si el transformador esta en una zona rural, tienen 6 horas para restablecer el suministro de energía eléctrica y evitar sanciones por parte de la CENACE (Centro Nacional de Control de Energía).

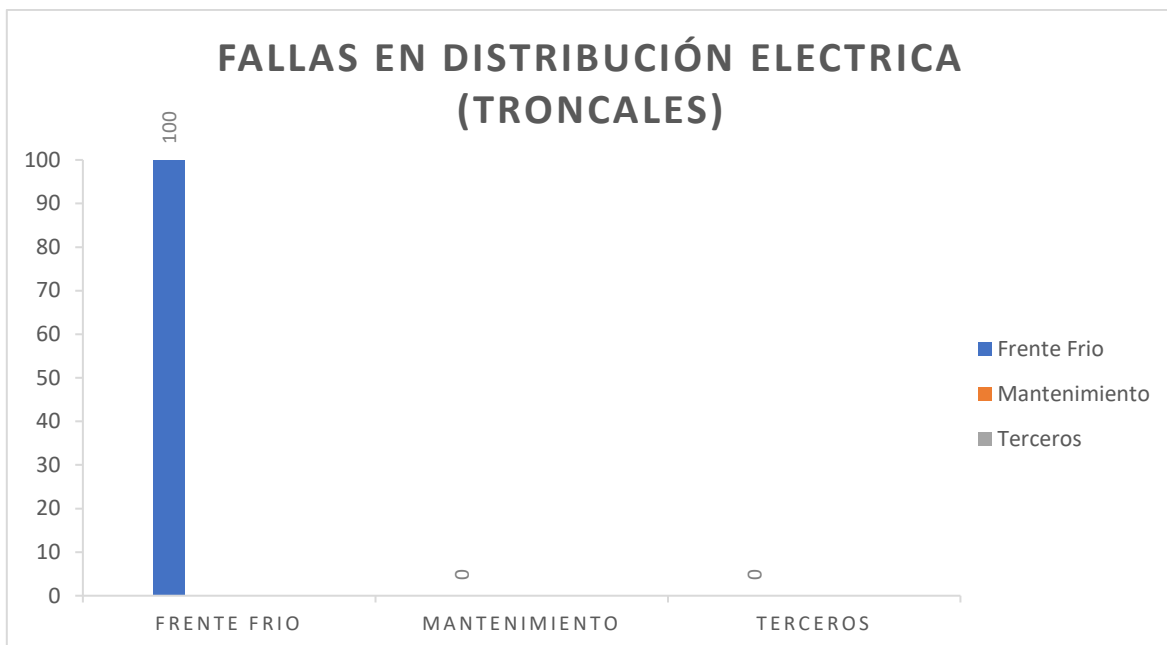


Figura 3.2 Grafico donde se muestra el porcentaje de la causa de fallas en troncales. Como podemos observar en la Figura 3.2, la mayor amenaza en cuanto a las fallas de distribución son los frentes fríos ya que con ellos, arrastran laminas, ramas, y algún otro objeto extraño, que son cosas que no podemos evitar ya que estas dependen mas del comportamiento del frente frio.

Tabla 3.2 Tabla de Fallas en Ramales en el circuito 04030.

FALLAS EN DISTRIBUCION ELECTRICA Periodo (2019-01-01 a 2019-10-19)						
Zona	S.E	Circuito	Tipo	Causa	Daños	Observaciones
03-SAN CRISTOBAL	CRI	04030	RAMAL	Operación Inad. Fusible de Bco. de Distribución.	Transformador de distribución	LIC 4161 RIM 406 (1-5/3) CAUSATRANSFORMADOR QUEMADO DEJO PASAR LA FALLA SE REPUSO 1 PZ FUSIBLE 5 AMP DEJANDO POTENCIAL AL 100% CALLE ORIZONTE SAN FELIPE CRI-4030, ATENDIO.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba.	Amarre	LIC. 2731 LINEA DE MEDIA TENSION SUELTA AMARRES DAÑADOS CAUSA RAMAS SOBRE LA LINEA DE M.T. POR LOS FUERTES VIENTOS Y LLUVIAS DEL CANAL DE BAJA PRESION SE CORRIGEN AMARRES CON ALUMINIO AS4 DL RAMAL 1F-1H, CALLE PEQUEÑO SOL CTO. CRI 04030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Conductor ACSR	LIC 2503 RIM 4304 (1-5/3) RAMA SOBRE POR FUERTES VIENTOS PROVOCADOS POR UN CANAL DE BAJA PRESION SE LEVANTA CONDUCTOR CON CONECTOR 1/0 CLARO M.T., EST. TIPO PS10, VISTA HERMOSA HUITEPEC CTO. CRI 04030.

03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Conductor ACSR	LIC 1418 Y 1422 RIM 180 50C 1428 AMP, ARBOL SOBRE LINEAS POR FUERTES VIENTOS ORIGINADOS POR, SE LEVANTA CONDUCTOR ACSR CON CONECTOR 3/0 SE INTERCONECTA RAMAL SN FELIPE CON EL CTO. CRI 4040, RECUPERA 100% DEL CTO. RAMAL A SAN FELIPE.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Conductor ACSR	LIC.2891, IN-R0036 ZACUALPA, ARBOL SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR LA ONDA TROPICAL NUM. 8, SE LEVANTA CONDUCTOR CON CONECTOR #2, CARRETERA ZACUALPA CTO. CRI 4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC 861 RIM 112 RAMAS SOBRE LA LINEA ORIGINADOS POR LOS FUERTES VIENTOS POR EL FRENTE FRIO #40, SE RETIRA RAMAS Y SE RETENSIONA FASE A Y C, NORMALIZANDO EL RAMAL LAS FLORES DEL CRI 4030
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC 307 RIM 24 (3-25/3) RAMA SOBRE LINEA POR FUERTE VIENTOS ORIGINADOS POR EL FRENTE FRIO #29, SE RETIRA RAMA Y NORMALIZA RAMAL ANITA HUIITEPEC CTO CRI-4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC 302 RIM 100 (3-25 3) RAMA SOBRES LINEA POR FUERTES VIENTOS ORIGINADOS POR FRENTE FRIO #29, SE RETIRA RAMAS Y NORMALIZA EL RAMAL, ANITA HUIITEPEC CTO. CRI-4030 CERRO HUIITEPEC.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC: 2154, RIM: 43, (1-5/2) EN RAMAL SAN FELIPE, DERIVADO A DESCARGA ATMOSFERICA, SE RESTABLECE FUSIBLE RESTABLECIENDO EL 100% DEL POTENCIAL. CIRCUITO CRI-4030, ATENDIO ZM-131.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Transformador de distribución	LIC 4155 RIM 42 129 (2-12/3) RAMA SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR LA ONDA TROPICAL NUM. 39, SE RETIRA RAMA Y NORMALIZA EL RAMAL 3F-3H CALLE RIO GRIJALVA CTO. CRI-4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Descarga atmosférica	Transformador de distribución	LIC 2749 RIM 142 (3-10/3) OPERO FUSIBLE POR FUERTES DESCARGAS ATMOSFERICAS SE

						RESPUSO FUSIBLE QUEDANDO CON POTENCIAL ZOOLOGICO RAMAL SN JOSE BOCOTONELTE ATENDIO ZM42 CRI-4030
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC 1205 RIM 149 (1-5/3) RAMA SOBRE LA LINEA POR FUERTES VIENTOS ORIGINADOS POR EVENTO DE NORTE EN EL ISTMO Y GOLFO DE T, SE RETIRO RAMAS SOBRE LAS LINEAS Y SE REPUSO FUSIBLES CALZADA HUITEPEC CTO CRI-4030 ATENDIO ZM132
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Transformador de distribución	LIC. 3063 RIM 52 N-R0002 CRI4030 ZACUALPA, RAMA SOBRE LINEAS POR FUERTES VIENTOS ORIGINADOS POR UN CANAL DE BAJA PRESION, SE RETIRA RAMAL ZACUALPA, CARRETERA A ZACUALPA, CTO. CRI 4030,
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Transformador de distribución	LIC. 3097 RIM 04474, ARBOL SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR LA ONDA TROPICAL NUM. 14, SE RETIRA ARBOL DE LA FASE C Y NORMALIZA RAMAL CTO. CRI 4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Transformador de distribución	LIC.3179 RIM 0460 (2-15/3) RAMA SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR UNA ONDA TROPICAL NUM. 16, SE RETIRA RAMA Y NORMALIZA RAMAL CTO. CRI 4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Transformador de distribución	LIC.3241 RIM.0472 (1-8/3) RAMA SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR UNA ONDA TROPICAL SE RETIRA RAMA Y NORMALIZA RAMAL 3F-3H HUITEPEC CTO.CRI-4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Tromba	Transformador de distribución	LIC 3299 RIM 85 (3-15/3) RAMA SOBRE LINEA POR FUERTES VIENTOS ORIGINADOS POR LA ONDA TROPICAL NUM. 19, SE RETIRA RAMA Y NORMALIZA RAMAL JOIGEL 3F-3H CARRETERA A VISTA HERMOSA HUITEPEC CTO. CRI-4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Descarga atmosférica	Sin Daño	LIC 3945 RIM 182 (1-25/3) POR DESCARGAS ATMOSFERICAS OPERO RAMAL SE REPUSO FUSIBLE QUEDANDO CORRECTO CRI-4030 ATENDIO ZM131 CALZ DANIEL SARMIENTO ROJAS.

03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC 1236 RIM 42 68 (1-8 3) SE ENCONTRO RAMAS SOBRE LAS LINEAS DERIVADO DE FUERTES VIENTOS ORIGINADOS POR EVNTO DE NORTE EN EL ISTMO Y GOLF. DE T. SE REPUSO FUSIBLES Y SE RETIRO RAMAS VISTA HERMOSA HUIITEPEC CRI-4030 ATENDIO ZM42
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Sin Daño	LIC. 1246 RIM. 157 (1-5 1-2/3) RAMAS SOBRE LA LINEA DERIVADO DE FUERTES VIENTOS ORIGINADOS POR EL FRENTE FRIO #44 SE RETIRA RAMA, SE REPONEN FUSIBLES, SE RESTABLECE LA TOMA DEL RAMAL EN HUIITEPEC OCOTAL Y VISTA HERMOSA HUIITEPEC
03-SAN CRISTOBAL	CRI	04030	RAMAL	Descarga atmosférica	Transformador de distribución	LIC.4137 RIM 213 (2-5/2) DESCARGAS ATMOSFERICAS CON FUERTES LLUVIAS PROVOCAN OPERACIÓN DE SECCIONAMINETO DE RAMAL SAN FELIPE CALLE ORIZABA CTO. CRI 4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Descarga atmosférica	Transformador de distribución	LIC. 4131 RIM 0622 (3-40/3) DESCARGAS ATMOSFERICAS Y FUERTES LLUVIAS PROVOCAN OPERACIÓN DE PROTECCION DEL RAMAL SE REPONEN FUSIBLES Y NORMALIZA RAMAL CTO. CRI 4030.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Frente Frio	Grapa de Remate	LIC 203 RIM 087 (1-15/3) ARBOL SOBRE LA LINEA DE MEDIA TENSION POR LOS FUERTES VIENTOS POR FRENTE FRIO NUM. 26, SE LEVANTA CONDUCTOR CON CONECTOR 1/0 SE NORMALIZA RAMAL ANITA HUIITEPEC.
03-SAN CRISTOBAL	CRI	04030	RAMAL	Terceros	Poste	SE PROCEDE A LIBRAR EL RAMAL DE SAN FELIPE PARA CAMBIAR UN POSTE DE 11:500 YA QUE FUE GOLPEADO POR UN CAMION DESCONOCIDO TIPO RD30/RD3/AD10 SE INSTALA UNO DE CONCRETO 11:500 CTO CRI 04040 APOYA ZM44

En la table 3.2 Se detallan los puntos importantes que se recaban para el restablecimiento de la energía eléctrica, en cuanto a ramales y troncales se recaban los mismos datos, estas fallas son en un periodo de 10 meses obteniendo 21 fallas en ramales, el tiempo de atención a ramales es más rápido ya que estos tienen un tiempo de restablecimiento máximo de 4 Hrs. Por estar en la Zona urbana, los ramales son derivaciones de troncales, para el suministro de energía eléctrica, los

ramales también tienen su propio transformador de distribución que es el encargado de regular la carga en el ramal para una correcta distribución.

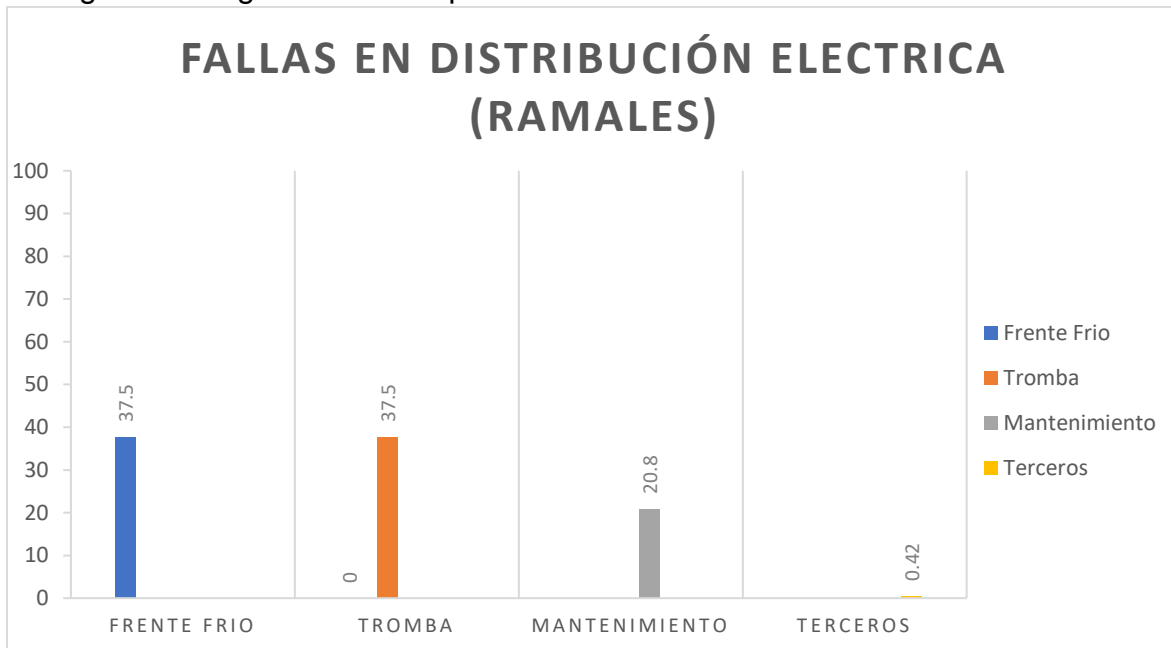


Figura 3.3 Grafico donde se muestra el porcentaje de la causa de fallas en ramales. Como podemos observar en la Figura 3.3 las amenazas a fallas en transformadores de ramales, son mayores y de diferentes tipos, los de mayor porcentaje son frentes fríos y tromba, le sigue el mantenimiento y por ultimo terceros, de estos el único punto que si es responsabilidad directa de CFE son las de mantenimiento ya que lo que ocasiona esta son descargas atmosféricas que dañan significativamente a los transformadores, las demás como Frente Frio, Tromba y Terceros, son eventos que no se pueden predecir, particularmente la de Terceros ya que uno nunca llega a pensar que alguna persona llegue a dañar algún poste, transformador o la instalación.

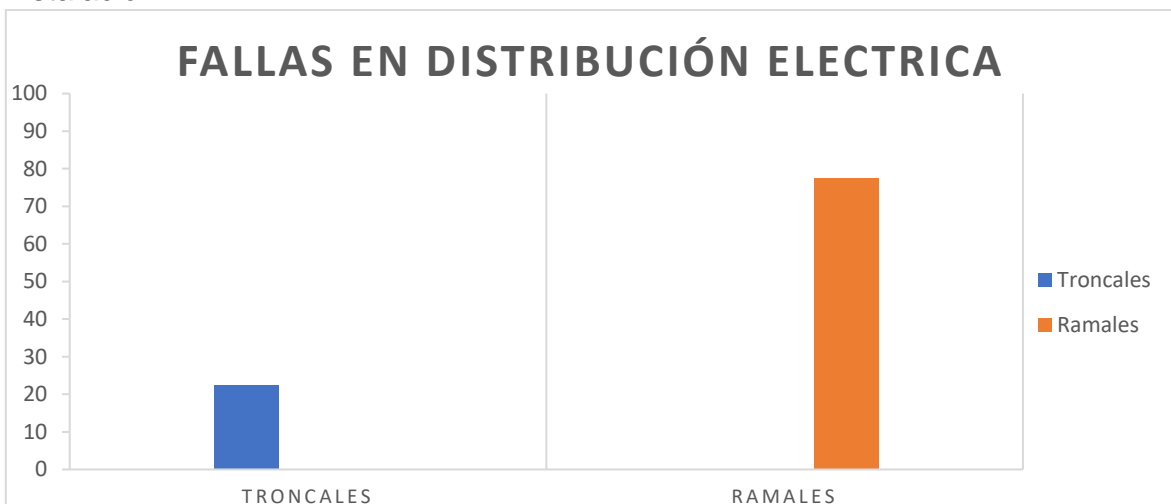


Figura 3.4 Grafico donde se muestra el porcentaje de fallas en distribución.

Como podemos observar en la Figura 3.4 mostramos cuales son los porcentajes de las fallas en distribución, donde podemos observar que tenemos 31 Fallas, de las cuales las fallas ocurren más en ramales con 77.41% y teniendo un menor porcentaje los troncales con un 22.59%, esto debido al número de estos, las condiciones del medio ambiente, el mantenimiento y la intervención de terceros.

Esto fue uno de los puntos que nos motivo a querer buscar una solución a este problema ya que cuando una falla ocurre, se espera a que el usuario lo reporte, preparan una orden de trabajo y la guardia en turno, debe recorrer el circuito para saber donde esta la falla e inspeccionar que tipo de falla fue la que ocurrió, en estos pasos se pierde mucho el tiempo con este sistema se espera tener una solución más rápida, obteniendo de la falla que fase fue la dañada y la latitud y longitud de donde se ubica, con esto obtendríamos una mejor solución, para el restablecimiento de energía eléctrica, ya que iríamos a la falla con la latitud y longitud y sabríamos que fase esta operada por una sobrecarga o alguna rama que se metió en la conexión de la Fase.

Diagrama Eléctrico de la conexión del Transformador:

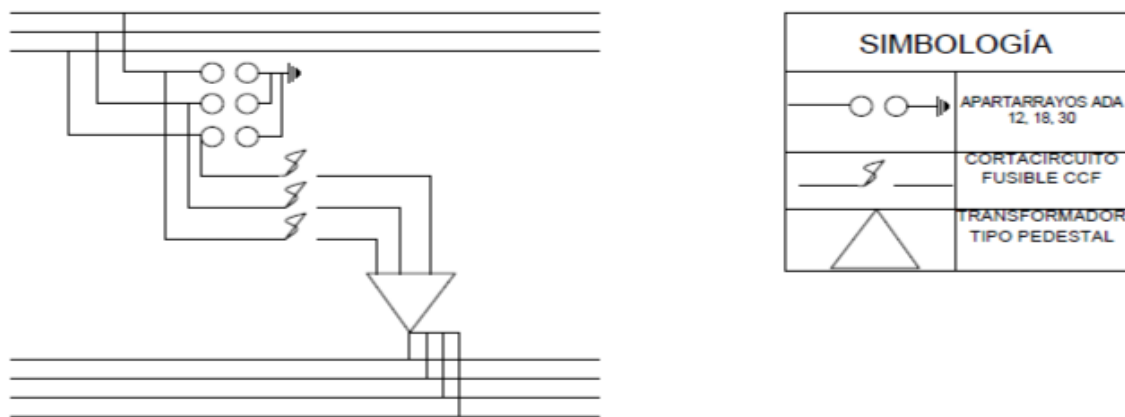


Figura 3.5 Diagrama eléctrico para Instalación de un transformador.

Para tener una visión mas clara de como es la conexión del transformador, se realizo el plano de como es la instalación de un transformador tipo poste o pedestal, esto se hizo con la intención de tener una mejor interpretación del transformador, así como en la tabla de la derecha se muestra la simbología del equipo que necesita.

3.2 DISEÑO DEL SISTEMA ELECTRÓNICO PARA RECABAR LA INFORMACIÓN NECESARIA

El microcontrolador que emplearemos será el Arduino uno, ya que nos ofrece muchas ventajas al monitorear sensores y por sus entradas analógicas ya que estas serán las que recibirán los datos de los sensores, recibiendo la información el Arduino podrá realizar las verificaciones correspondientes e identificar si los valores que reciben son de alguna falla o el transformador esta operado correctamente.

3.2.1 SENSOR DE MEDICIÓN.

El Sensor que usaremos para medir el correcto funcionamiento del transformador será el SCT-013-000, este sensor tiene la capacidad para medir corrientes de 50

mA hasta los 100 A, este sensor nos da un rango bastante útil para verificar los transformadores, ya que varios de los transformadores que se utilizan, en media y baja tensión, están dentro de ese rango de corriente.

Al adquirir el sensor SCT-013-000 inmediatamente no podemos empezar a utilizarlo, ya que para que las mediciones que obtendremos sean correctas necesitamos calibrarlo, para esto, necesitamos conectarlo a nuestro microcontrolador y tomar como referencia los valores que nos entregan los fabricantes como la potencia y la corriente, para esto necesitamos conectar algún aparato con una potencia alta para que la obtención de datos así sea mas precisa.

La calibración se realizo con un Cautín que tiene una potencia de 30 W, si hacemos los cálculos para obtener el valor de la corriente seria:

$$P = V * I$$

Si despejamos la corriente que sería lo que buscamos, quedaría de la siguiente manera:

$$I = \frac{P}{V} \rightarrow I = \frac{30\text{ W}}{127\text{ V}} \rightarrow I = 0.2362\text{ Amperes.}$$

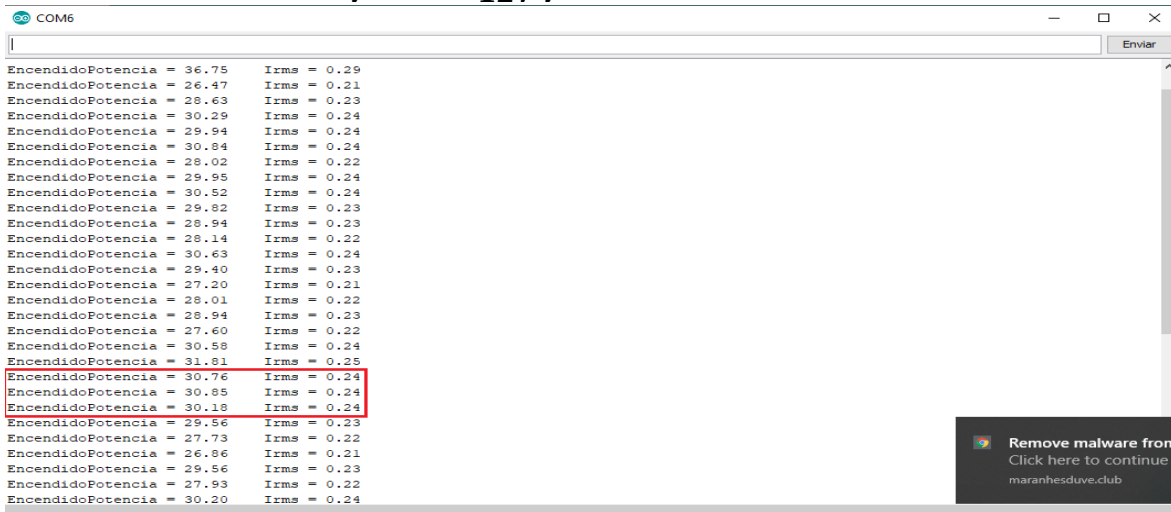


Figura 3.6 Datos obtenidos en la calibración de sensores.

Como podemos observar en la captura de pantalla, al obtener los datos para la calibración, los datos de la corriente que debimos obtener si son muy cercanos a los calculados, con esto podemos decir que nuestro sensor ya esta calibrado y lo podemos emplear en el monitoreo de los transformadores.

En el cuadro rojo marcamos los valores obtenidos con una potencia de 30 Watts.

Tabla 3.3 Tabla comparativa del valor calculado y el valor medido.

Valor calculado	Valor medido
0.2362 Amperes	0.24 Amperes

En esta tabla se compara el valor calculado, con el valor medido con el sensor, al hacer la medición de corriente del cautín se obtuvo una medición muy cercana a la precisa.

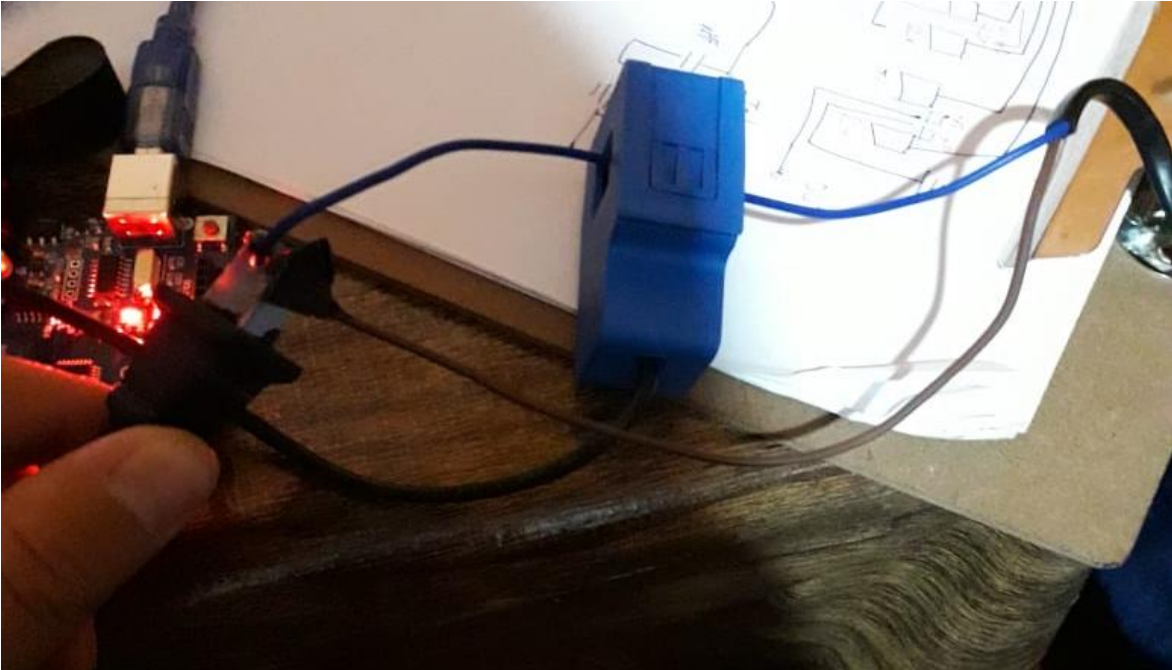


Figura 3.7 Medición de corriente con el Sensor SCT-013-000.

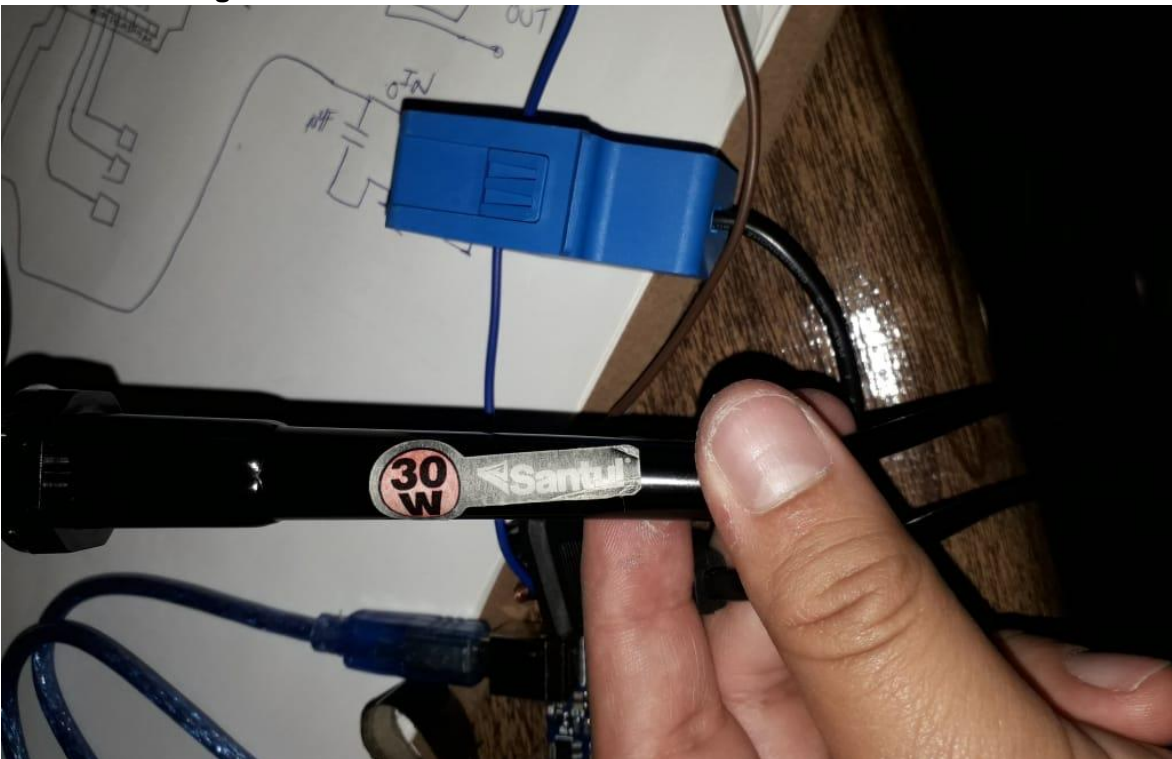


Figura 3.8 Cautín empleado para la calibración del sensor.

Programación empleada para la calibración:

```
#include "EmonLib.h"
```

En la parte superior de nuestro sketch, debemos incluir la librería “Emonlib.h” ya que esta es la librería que necesitamos para el monitoreo de la energía eléctrica.

EnergyMonitor energyMonitor;

Creamos una variable donde guardaremos los datos obtenidos por el sensor.

float voltajeRed = 127.0;

Declaramos una variable de valor flotante, con el valor del voltaje que nos suministran.

```
void setup () {  
  Serial.begin(9600);  
  energyMonitor.current(0, 0.7);}
```

En la parte del void setup () inicializamos el Serial con una comunicación de 9600 baudios, así como declaramos el pin Analógico que usaremos, que en este caso será el pin A0, y el valor de la calibración que en este caso es de 0.7, ya que con este valor obtenemos el valor correcto en las mediciones.

```
void loop () {  
  double Irms = energyMonitor.calcIrms(1484);  
  double potencia = Irms * voltajeRed;  
  Serial.print("Encendido");  
  Serial.print("Potencia = ");  
  Serial.print(potencia);  
  Serial.print("  Irms = ");  
  Serial.println(Irms);}
```

En la parte del void loop (), obtenemos el valor de la corriente eficaz (Irms), llamando la función calcIrms(número de muestras), el número de muestras es importante ya que indica el número de veces que se va leer la señal para obtener el valor de la corriente eficaz, las muestras son en ciclos completos, según la velocidad de acceso a los pines analógicos con Arduino uno, aproximadamente podemos medir 106 muestras en cada ciclo, Necesitamos medir 14 ciclos que es el valor recomendado para emplearse en este sensor, así que obtendríamos el siguiente valor.

Muestras en cada ciclo X Ciclos que queremos medir = 106 x 14 = 1484

También obtenemos el valor de la potencia que sería multiplicando la corriente obtenida por el voltaje de red que nos suministran, e imprimimos los valores obtenidos en el serial, para ver si son correctos, con el nombre de "Potencia" y "Irms".

3.2.2 OBTENCIÓN DE COORDENADAS GEOGRÁFICAS

Para obtener las coordenadas geográficas, usaremos el Modulo GPS Serial NEO-6M, ya que es un módulo de bajo consumo, y muy preciso al obtener las coordenadas y demas datos que necesitemos, así como su tamaño pequeño, que podemos incluir en nuestros circuitos.

Programación empleada para la calibración:

```
#include <TinyGPS.h>
```

En la parte superior del sketch debemos incluir la librería “TinyGPS.h” que es la encargada de obtener los datos por medio del módulo.

```
float latitud, longitud;
```

Declaramos una variable del tipo float, donde guardaremos los datos de latitud y longitud.

```
TinyGPS gps;
```

Declaramos el objeto “gps” donde guardaremos los datos obtenidos por el módulo.

```
int year;
```

```
byte month, day, hour, minute, second, hundredths;
```

```
unsigned long chars;
```

```
unsigned short sentences, failed_checksum;
```

Declaramos las variables de las cuales obtendremos los datos, como son el año, mes, día, hora, minuto, segundo, entre otras variables.

```
void setup () {  
  Serial.begin(9600);  
  Serial.println("");  
  Serial.println("GPS GY-GPS6MV2 Leantec");  
  Serial.println(" ---Buscando senal--- ");  
  Serial.println("");}
```

En la parte del void setup inicializamos el serial con una comunicación de 9600 baudios, así como escribimos unos mensajes, mientras obtenemos los datos que nos interesan por medio del módulo.

```
void loop() {  
  int c = Serial.read();  
  if(gps.encode(c))  
  {  
    gps.f_get_position(&latitud, &longitud);  
    Serial.print("Latitud/Longitud: ");  
    Serial.print(latitud,5);  
    Serial.print(", ");  
    Serial.println(longitud,5);  
    gps.crack_datetime(&year,&month,&day,&hour,&minute,&second,&hundredths);  
    Serial.print("Fecha: "); Serial.print(day, DEC); Serial.print("/");  
    Serial.print(month, DEC); Serial.print("/"); Serial.print(year);  
    Serial.print(" Hora: "); Serial.print(hour, DEC); Serial.print(":");
```



```

Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
Serial.print("."); Serial.println(hundredths, DEC);
Serial.print("Altitud (metros): "); Serial.println(gps.f_altitude());
Serial.print("Rumbo (grados): "); Serial.println(gps.f_course());
Serial.print("Velocidad(kmph): "); Serial.println(gps.f_speed_kmph());
Serial.print("Satelites: "); Serial.println(gps.satellites());
Serial.println();
gps.stats(&chars, &sentences, &failed_checksum);
}
}
}

```

En la parte del void loop se establece la comunicación por medio del serial, donde los datos obtenidos se envían a la variable c, para que esos valores puedan ser usados por el objeto gps, después de obtener los datos por el gps, los escribiremos en el monitor serial, para ver cuáles son los datos que se obtuvieron, los datos obtenidos son latitud, longitud, año, mes día, la hora, fecha, altitud, rumbo, satelites, etc.

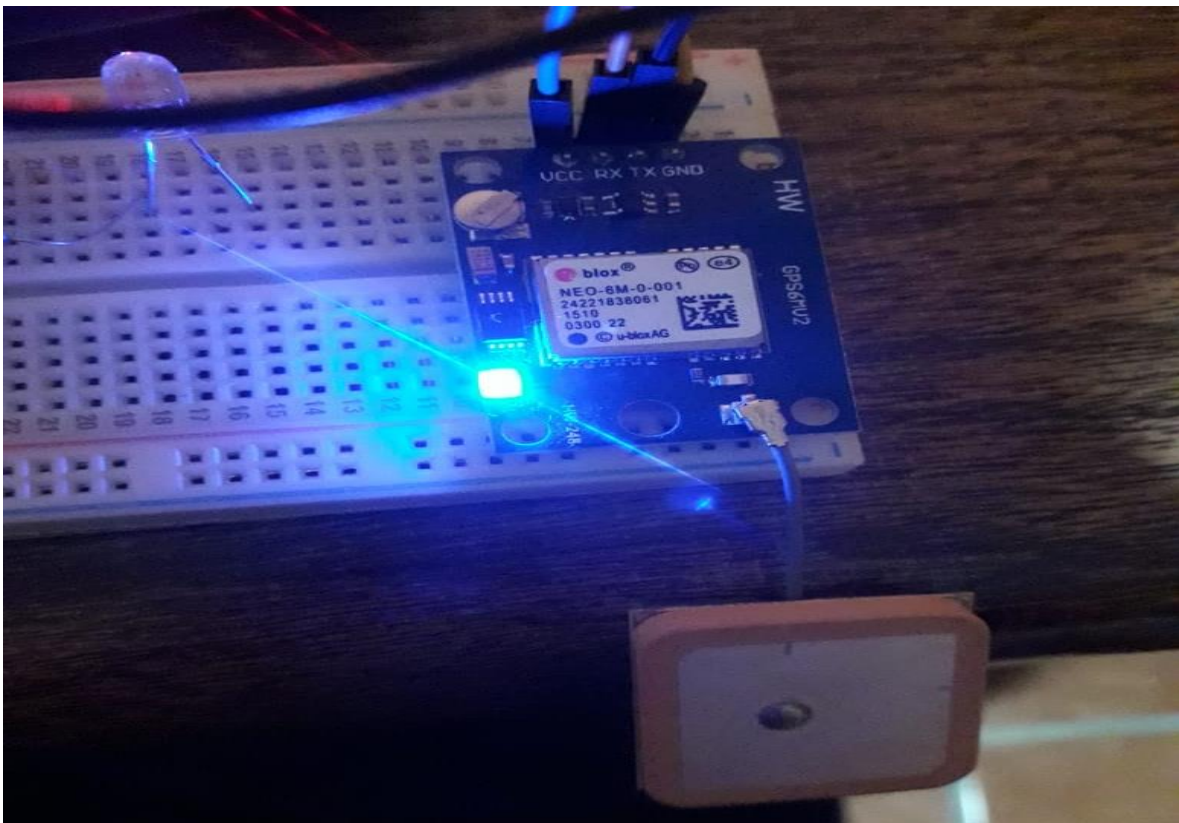


Figura 3.9 Módulo GPS en funcionamiento.

Al programar nuestro módulo debemos ver que se haya programado correctamente, ya que cuando obtuvo los datos, el led azul parpadea cada segundo.

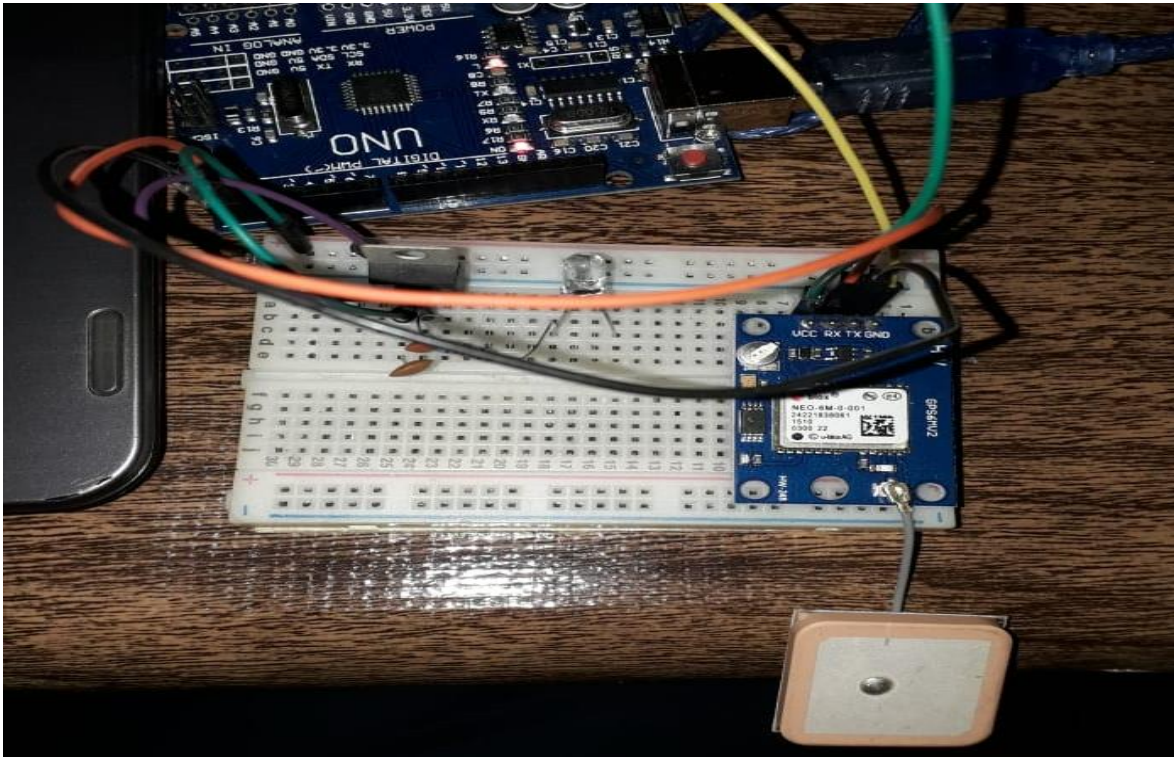


Figura 3.10 Conexión del Módulo GPS con Arduino.

```

COM6
GPS GY-GPS6MV2 Leantec
---Buscando señal---

Latitud/Longitud: 16.71700, -92.64090
Fecha: 16/12/2019 Hora: 7:39:55.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.61
Satelites: 255

Latitud/Longitud: 16.71700, -92.64089
Fecha: 16/12/2019 Hora: 7:39:56.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.24
Satelites: 255

Latitud/Longitud: 16.71700, -92.64089
Fecha: 16/12/2019 Hora: 7:39:57.0
Altitud (metros): 1000000.00
Rumbo (grados): 0.00
Velocidad(kmph): 0.41
Satelites: 255

```

Figura 3.11 Datos obtenidos por el módulo GPS, escritos en el monitor Serial.

3.2.3 COMUNICACIÓN EN CASO DE FALLA.

En nuestro sistema electrónico tendremos un Receptor (Rx) y un Transmisor (Tx), la comunicación la haremos con los módulos SIM800L, este es un módulo que nos sirve para enviar y recibir mensajes de texto, ya que el transmisor lo colocaremos en el transformador y el receptor lo tendremos en nuestro centro de trabajo, el receptor se encargará de subir los datos recibidos a Ubidots y mostrarnos a partir de la latitud y longitud donde está ubicada la falla.

Transmisor:

En la parte del transmisor el módulo será el encargado de comunicar cuando haya una falla en el transformador, el sistema hará las verificaciones correspondientes para identificar cual falla ocurrió, y el módulo enviara el mensaje con la latitud y longitud de donde se encuentra el transformador para que lo reparen, en caso de no haber alguna falla, el módulo no enviara ningún mensaje y el sistema seguirá con su monitoreo.

Programación del Transmisor:

```
#include <SoftwareSerial.h>
```

Para la programación del Transmisor, necesitamos incluir la librería "SoftwareSerial.h", ya que la comunicación serial, ya la tenemos ocupada por el módulo GPS, con esto tendremos un transmisor y receptor por software, que será un puerto virtual.

```
int contadora = 0;
```

Declaramos la variable "contadora", esto para que cada cuando una falla ocurra, incremente el valor de la variable, la variable será condicionada para que cuando llegue a un cierto valor envíe el mensaje con la latitud y longitud.

```
SoftwareSerial MOD_SIM800L (2,3);
```

Declaramos los pines que nos servirán para la recepción y transmisión de datos, el pin 2 será el receptor (Rx), el pin 3 será el transmisor (Tx).

```
Void setup () {  
  MOD_SIM800L.begin(115200);}
```

En el Void setup, únicamente inicializamos el módulo con una comunicación de 115200 baudios.

```
Void loop () {  
  if((contadora>180) && (contadora<185)) {  
    contadora=0;
```

En el Void loop la función del módulo será condicionada por el contador, ya que cuando llegue al valor indicado, enviará el mensaje de la falla.

```
    MOD_SIM800L.write("AT+CMGF=1\r\n");  
    Delay (100);
```

Con este comando establecemos el formato SMS (Mensaje de texto) en ASCII, Con un descanso de 100 milisegundos.

```
    MOD_SIM800L.write("AT+CMGS=\"+529671387727\"\r\n");  
    Delay (100);
```

Con este comando indicaremos un nuevo SMS al numero indicado, es importante colocar la extensión, ya que sin esta no enviara el mensaje y con un descanso de 100 milisegundos.

```
MOD_SIM800L.println("Latitud/Longitud: ");
MOD_SIM800L.print(latitud,5);
MOD_SIM800L.print(" ");
MOD_SIM800L.println(longitud,5);
Delay (1000);
```

Estos comandos indican el contenido que ira en el mensaje, en este caso enviaremos la latitud y longitud y un descanso de 1 segundo.

```
MOD_SIM800L.write((char)26);
Serial.println("Latitud/Longitud: ");
Serial.print (latitud,5);
Serial.print(" ");
Serial.println (longitud,5);
Serial.println(¡" Mensaje enviado!");}}
```

Por último, escribimos el comando final para que envíe el mensaje y escribimos los datos en el puerto serial, y la confirmación de que el mensaje fue enviado.

Receptor:

El Receptor será el encargado de recibir la latitud y longitud de las fallas, al recibir las fallas hará comparaciones para ver si el mensaje que recibió, coincide con la ubicación del transformador, si es correcta transfiere el valor a la variable "posición", y la variable posición creara un punto en el mapa para ver la ubicación de la falla.

Programación del Transmisor:

```
#include <SoftwareSerial.h>
```

Para la programación del Receptor, necesitamos incluir la librería "SoftwareSerial.h", con esto tendremos un transmisor y receptor por software, que será un puerto virtual.

```
SoftwareSerial SIM800L (8,9);
```

Declaramos los pines que nos servirán para la recepción y transmisión de datos, el pin 8 será el receptor (Rx), el pin 9 será el transmisor (Tx).

```
Void setup () {
SIM800L.begin(9600);
SIM800L.println("AT+CMGF=1");
Delay (100);
SIM800L.println("AT+CNMI=1,2,0,0,0");}
```

Con estos comandos en el Void setup, iniciamos la comunicación serial que será de 9600 Baudios, y con los comandos AT le indicaremos que utilizaremos los SMS (mensajes de texto) y le indicaremos que leerá los mensajes de texto.

```
Void loop () {  
if (SIM800L.available(>0) {  
  ubicacion = SIM800L.readString();  
  Serial.println(ubicacion);  
}
```

En el Void loop, se hace la primera verificación donde indicamos que, si el módulo SIM800L está habilitado, transfiera los valores a la variable ubicación, e imprima los valores en el monitor serial.

```
if(ubicacion.indexOf("16.73149-92.64257")>=0) {  
digitalWrite (6,HIGH);  
falla=16.73149-92.64257;  
Delay (1000);  
}}
```

Se tiene una segunda verificación donde con los comandos indicamos, que si el módulo, recibe la latitud y longitud donde se encuentre el transformador, encienda el indicador y envíe los datos a la plataforma, para ubicar la falla en el mapa con un descanso de 1 segundo.

Indicador LED de Estado

Hay un led en lado superior derecho del módulo celular SIM800L que indica el estado de su red celular. Parpadea a diferentes velocidades para mostrar en qué estado se encuentra.



Parpadea cada segundo: El módulo se está ejecutando, pero aún no se ha conectado a la red celular.

Parpadea cada dos segundos: La conexión de datos GPRS que solicito está activada.

Parpadea cada tres segundos: El módulo se ha puesto en contacto con la red celular y puede enviar / recibir voz y SMS.

Figura 3.12 Estados del LED Indicador.

Para no tener inconvenientes es necesario saber como funciona el módulo SIM800L, ya que si no se sabe cómo funciona, no obtendremos los resultados que buscamos, el led indicador será el que nos dirá el estado del módulo.



Figura 3.13 Datos obtenidos sobre el conteo del tiempo de la falla.

En esta imagen mostramos los datos obtenidos por el GPS, y enviamos un mensaje con las coordenadas de la falla a un numero de celular, esto se hizo para ver si el envío de mensajes era correcto.



Figura 3.14 Mensaje de texto recibido en numero de celular establecido.

3.2.4 CARCASA DE SENSORES DE MEDICIÓN.

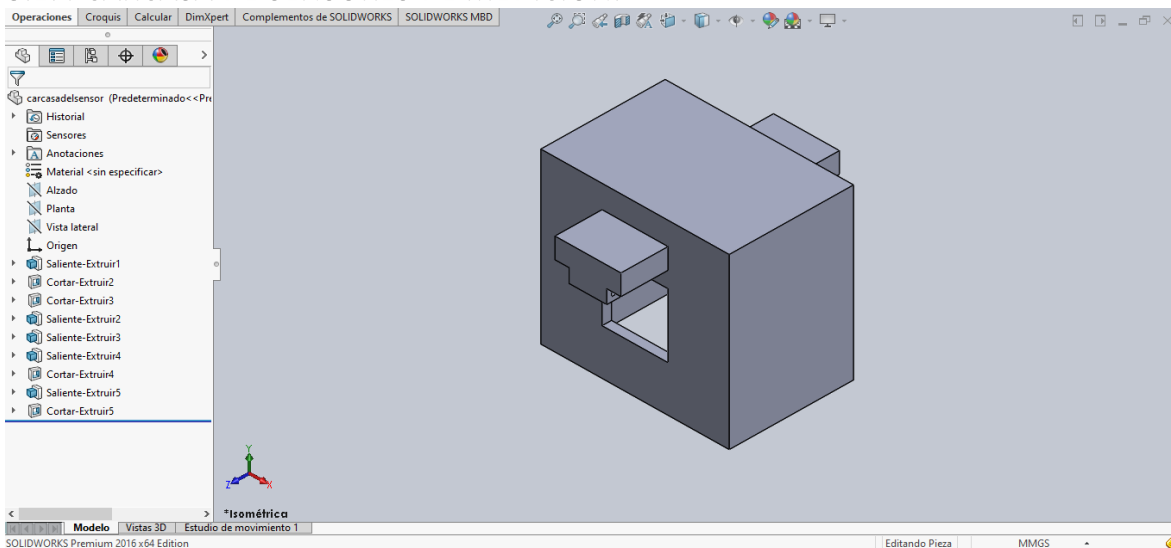


Figura 3.15 Vista Isométrica de la carcasa del sensor.

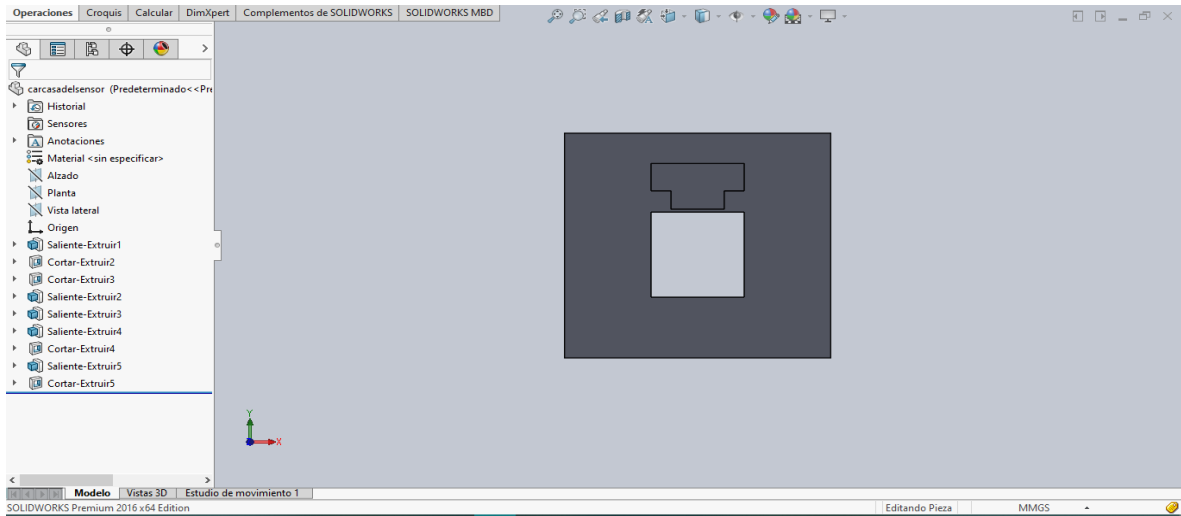


Figura 3.16 Vista Frontal de la carcasa del sensor.

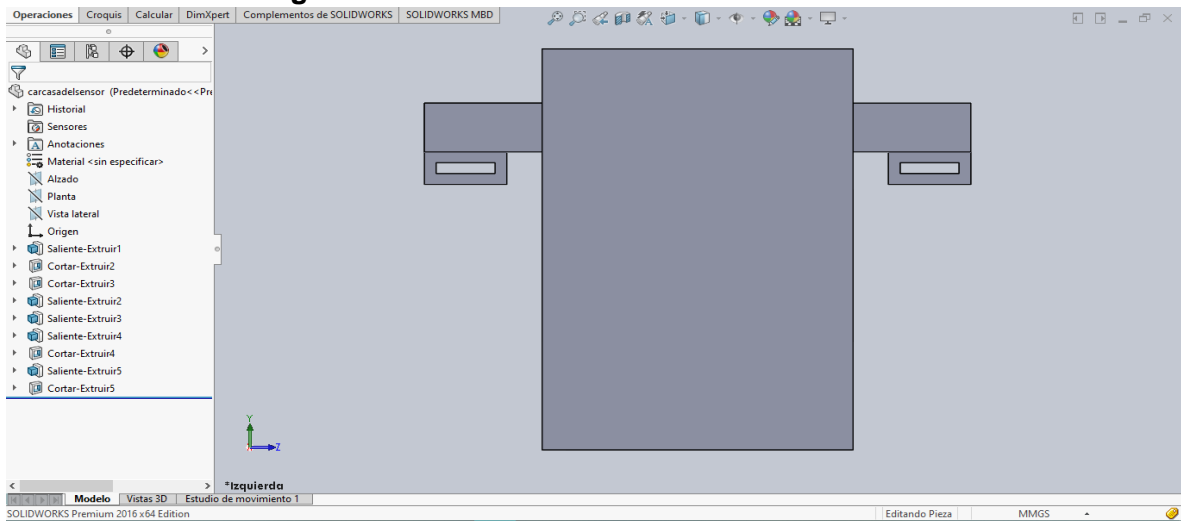


Figura 3.17 Vista Derecha de la carcasa del sensor.

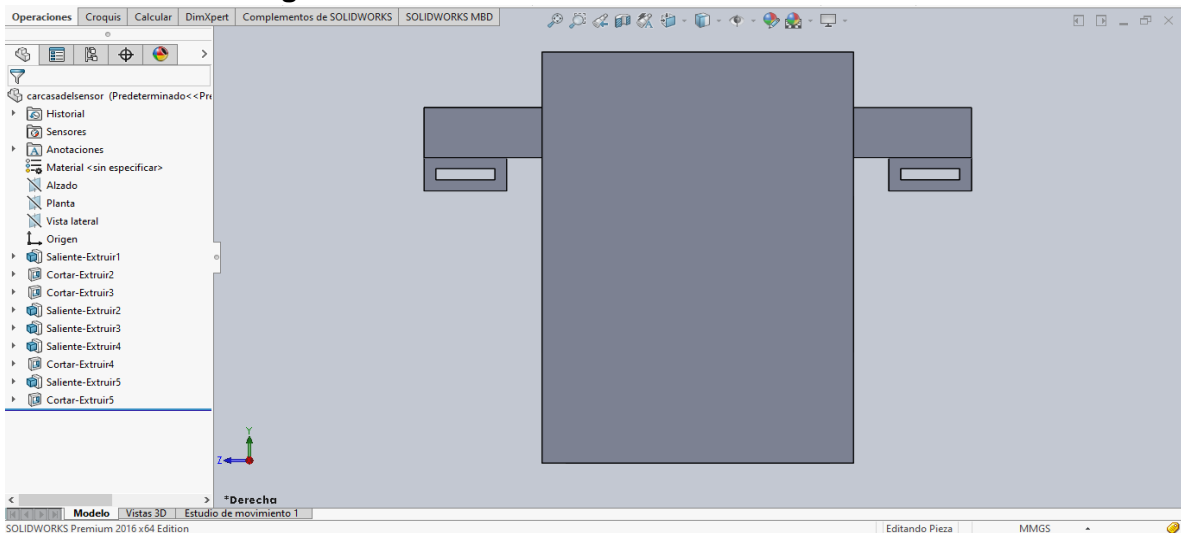


Figura 3.18 Vista Izquierda de la carcasa del sensor.

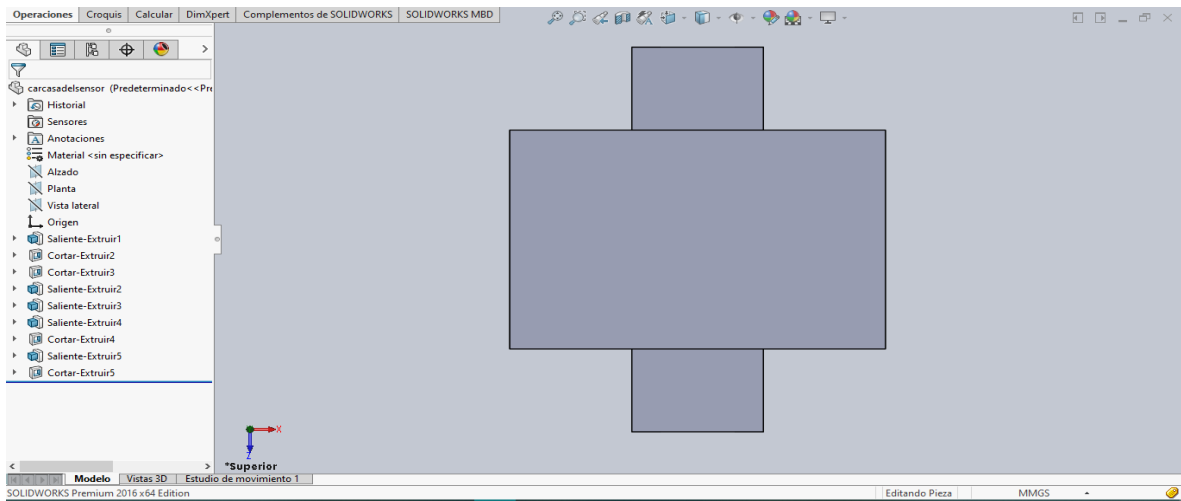


Figura 3.19 Vista Superior de la carcasa del sensor.

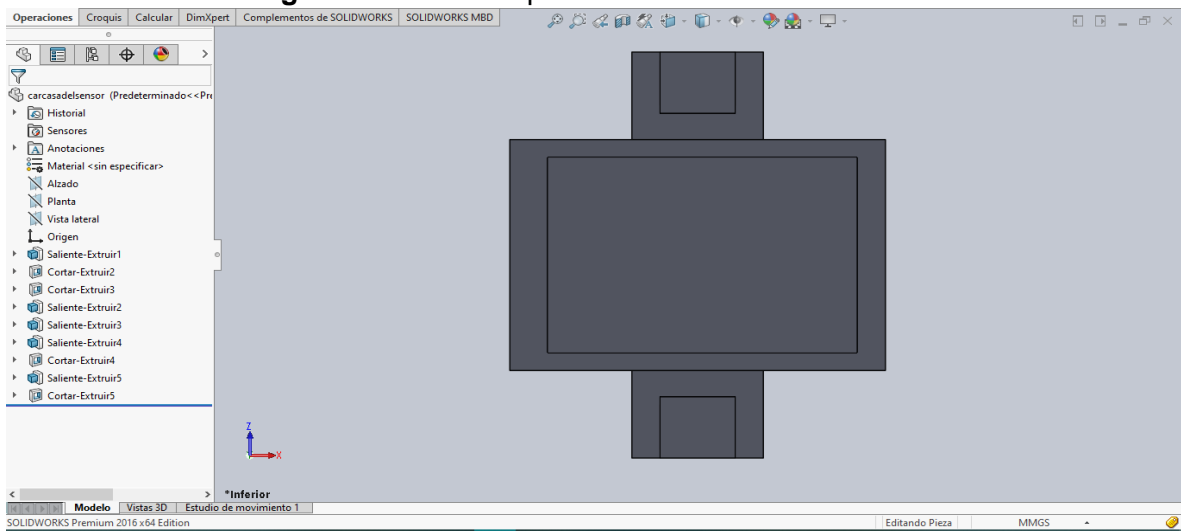


Figura 3.20 Vista Inferior de la carcasa del sensor.

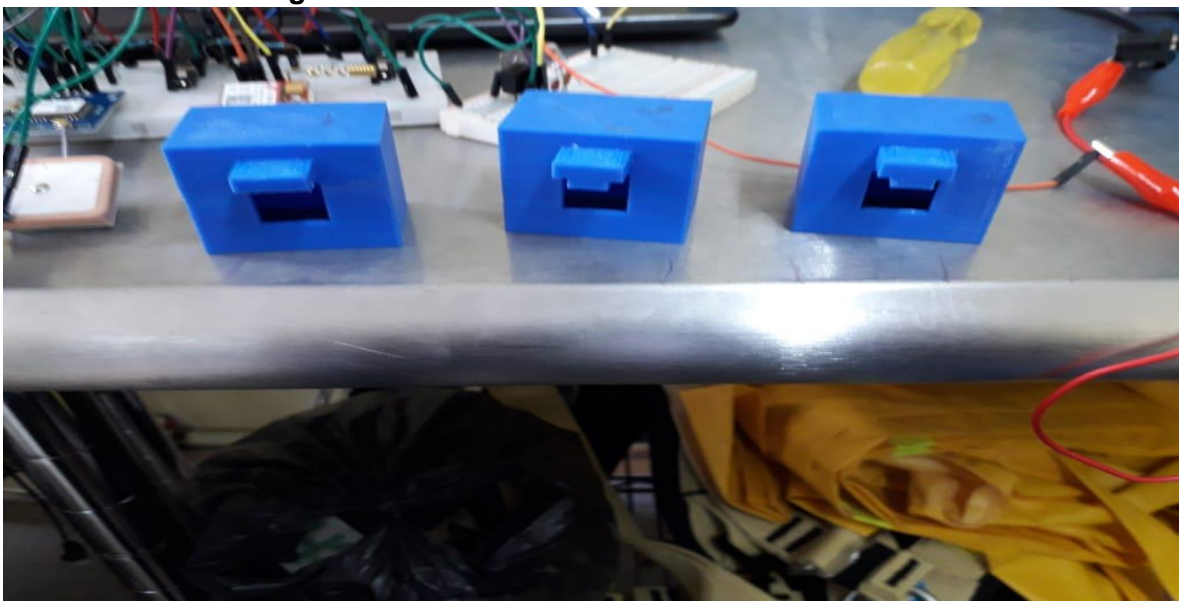


Figura 3.21 Pieza final hecha por una impresora 3D.

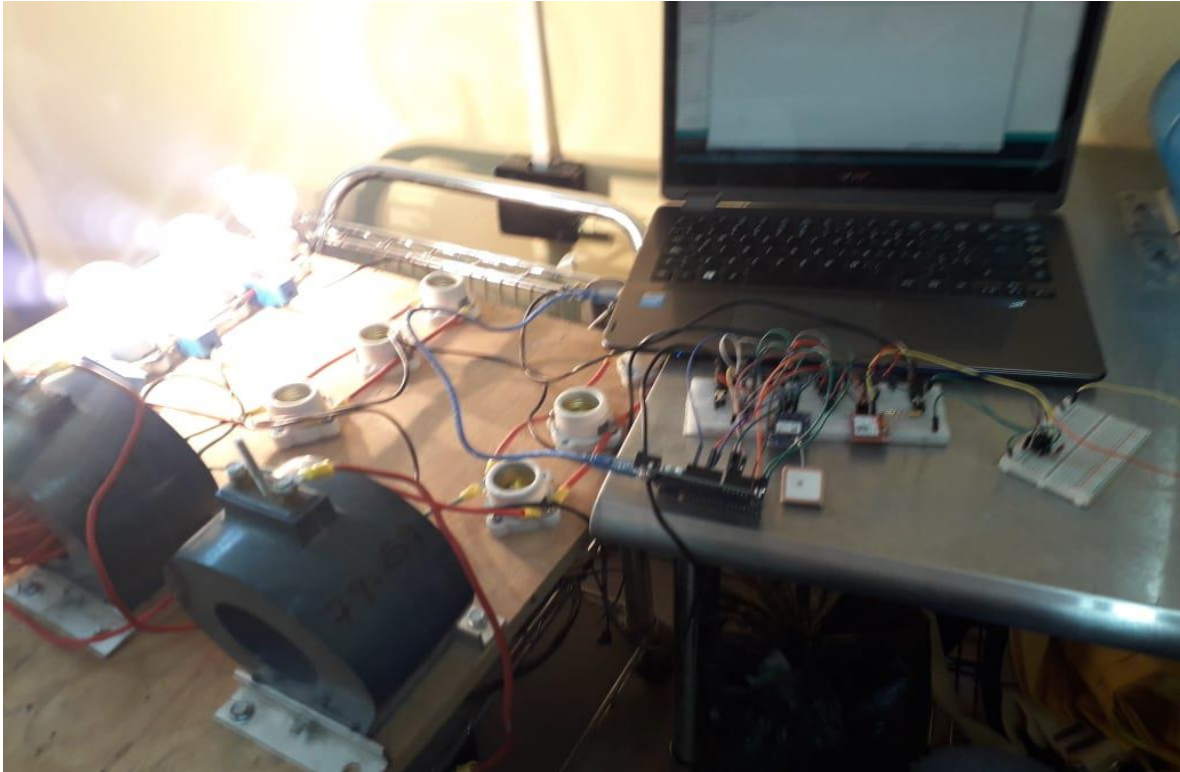


Figura 3.22 Pruebas en el laboratorio de mediciones.



Figura 3.23 Medición con amperímetro para ver si la lectura es correcta.

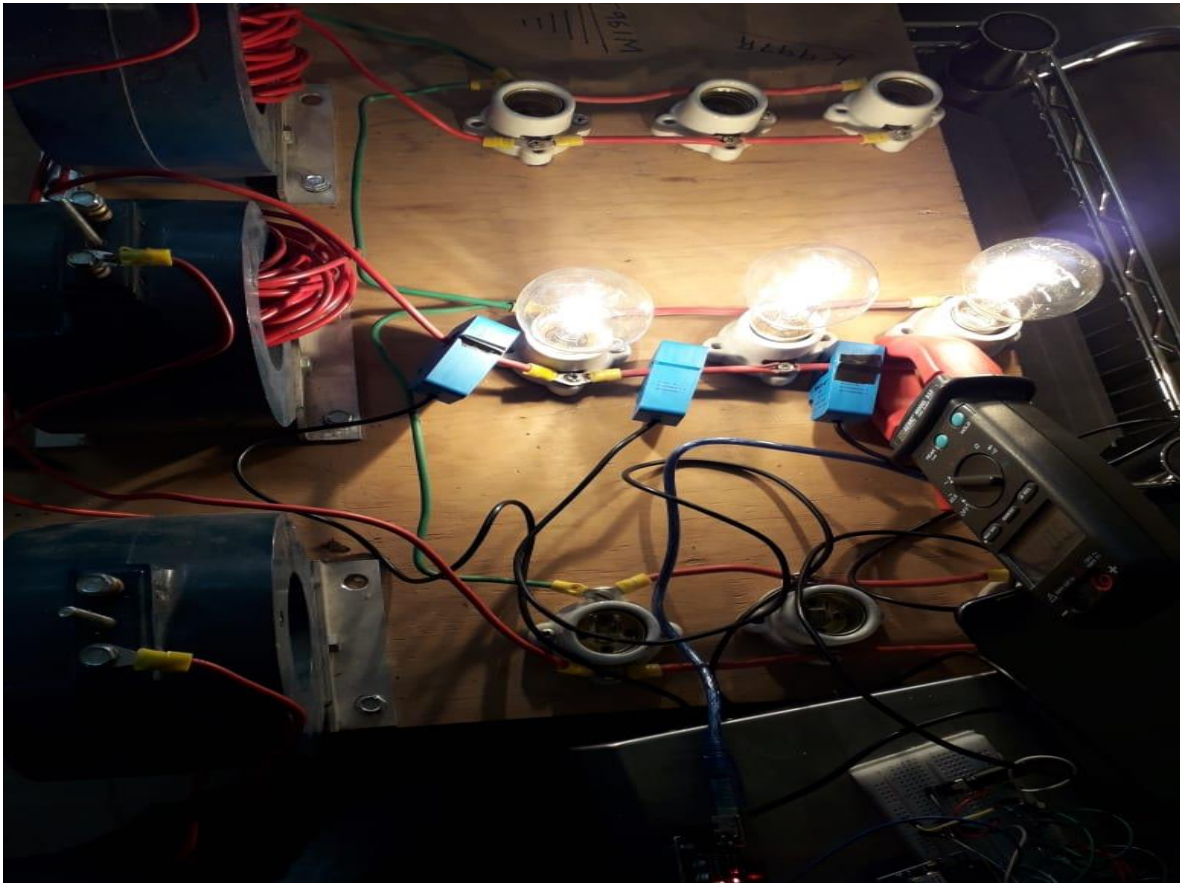


Figura 3.24 Medición con sensores y amperímetro.



Figura 3.25 Lugar de trabajo donde se realizaban las pruebas.

3.3 ELEGIR EL FORO (PÁGINA WEB) MÁS VIABLE PARA LAS NECESIDADES DEL SISTEMA Y EL ACONDICIONAMIENTO DE LAS SEÑALES PARA TENER UNA BUENA LECTURA DE ESTAS.

Para realizar la implementación del foro donde visualizaremos los datos de las fallas en los transformadores, fue necesario recabar información para ver cuál es la opción mas viable y las consideraciones a tomar en cuenta.

Consideraciones:

- 1._ El personal que trabaja en CFE, son personas que no todos saben el manejo de paginas web o saben interpretar datos como latitud y longitud, por eso mismo se busca una solución para que todos puedan ver los datos recabados.
- 2._ No todos tienen acceso a una computadora y menos a internet.
- 3._ El manejo de páginas Web o foros en CFE es restringido y debe ser verificada para ser ingresada a los servidores, es ingresada a los servidores cuando ya haya sido revisada y cumpla con los parámetros de seguridad.
- 4._ Considerar la interpretación de los datos, ya que debe ser un foro donde podamos mostrar los datos de manera correcta, para que a partir de estos tomar las acciones necesarias.

A partir de estas consideraciones debemos investigar cual es el foro o pagina web mas viable, para desarrollar el monitoreo, obtuvimos diferentes foros donde podríamos obtener una buena comunicación, que se presentan a continuación:

- **ThingSpeak:** Es una plataforma de análisis IoT (Internet of Things), que te permite agregar, ver y analizar flujos de datos, de una manera muy sencilla, una de las grandes ventajas de la plataforma ThingSpeak es que nos permite visualizar los datos enviados por nuestros dispositivos en tiempo real, pero también la posibilidad de analizar los mismos recurriendo a Matlab.
- **Ubidots:** Es una plataforma más dirigida al mundo IoT. En la práctica, con esta plataforma, podemos visualizar de forma sencilla e intuitiva la información de sensores conectados a dispositivos IoT en nuestra red, la ventaja de esta plataforma es que ofrece aplicaciones móviles para que podamos tener en nuestro dispositivo móvil toda la información necesaria.
- **Emoncms.org:** Es una plataforma de código abierto dirigido a IoT, para procesamiento y registro de variables como la temperatura, energía, etc. Emoncms es parte del proyecto OpenEnergyMonitor.
- **ThingWorx:** Esta es una de las plataformas más populares para el mundo de internet de las cosas, esta plataforma, la mayoría con vistas al segmento de negocios, incluye las tecnologías y herramientas que permiten a las empresas desarrollar e implementar soluciones innovadoras de la IoT y experiencias de realidad aumentada de forma rápida y sencilla.
- **Kaa:** Esta es una plataforma de código abierto para el desarrollo de soluciones IoT y dispositivos inteligentes. Kaa ofrece una amplia gama de herramientas de IoT que pueden ser fácilmente interconectadas.

- **Thinger.io:** Esta plataforma de código abierto soporta múltiples tipos de protocolos, sensores y actuadores. Esta plataforma ofrece una buena interfaz y altamente interactiva.

Dentro de todas estas plataformas la que mas se adecua a las necesidades de nuestro sistema y por lo que nos ofrece es **Ubidots**, por eso el proyecto lo desarrollaremos en esta plataforma.

Acerca de Ubidots:

Ubidots es una empresa de análisis y visualización de datos de Internet de las cosas (IoT). Convertimos los datos de los sensores en información importante para las decisiones empresariales, las interacciones de máquina a máquina, la investigación educativa y aumentamos la economización de los recursos globales. Ubidots existe como un medio fácil y accesible para integrar el poder de IoT en su negocio o investigación. La tecnología de Ubidots y la pila de ingeniería se desarrollaron para ofrecer una experiencia y de guante blanco para nuestros usuarios. Las API amigables para dispositivos (a las que se accede a través de los protocolos HTTP/MQTT/TCP/UDP) proporcionan una conexión simple y segura para enviar y recuperar datos desde y hacia nuestro servicio en la nube en tiempo real.

Nacida como una empresa de servicios de ingeniería en 2012, Ubidots entregó soluciones de IoT de extremo a extremo en conjunto con su socio y compañía fundadora Netux, para monitorear, controlar y automatizar de forma remota los procesos para clientes de atención médica, Entre 2012 y 2014, Ubidots realizó innumerables proyectos conectados a Internet en Salud, Energía / Servicios públicos, Fabricación, Transporte y Venta minorista, aprendiendo las muchas características pequeñas de IoT y habilitación de la nube con las que los expertos en transformación digital no pueden hablar a menos que se hayan ensuciado las manos en el campo. Ubidots se unió al Boston MassChallenge Accelerator en 2013, convirtiéndose en una startup tecnológica global, ganando aventuras y apoyo en los Estados Unidos, y dejando atrás el negocio de servicios locales.



Figura 3.26 Logotipo de la plataforma “Ubidots”

3.3.1 PROCESO PARA LA VISUALIZACIÓN DE DATOS EN UBIDOTS.

Paso 1 Registro:

Para tener un lugar en la plataforma de Ubidots es necesario registrarnos con un correo electrónico y una contraseña, la plataforma tiene diferentes versiones

dependiendo de las necesidades de nuestro sistema, ya que depende del numero de variables, la cantidad de información que se enviara y el tipo de valores que se enviara.



Figura 3.27 Proceso de registro en la plataforma “Ubidots”

Paso 2 Panel de demostración:

Al registrarnos correctamente en la página de Ubidots nos encontramos con el panel de demostración, donde encontraremos una barra de Herramientas, con la que podremos interactuar, para el monitoreo del dispositivo.

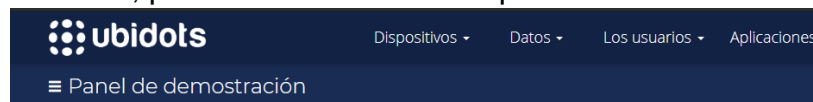


Figura 3.28 Barra de Herramientas de “Ubidots”

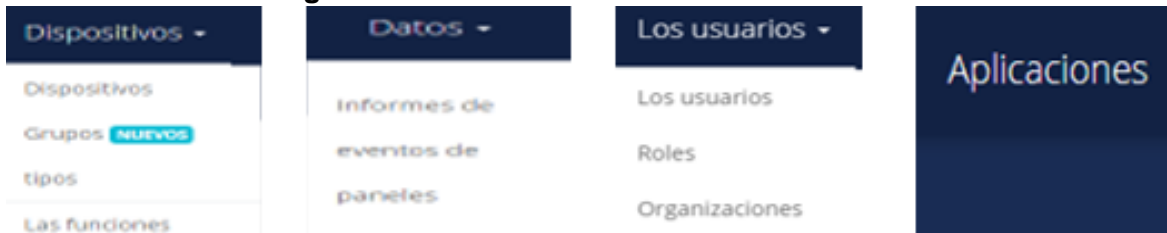


Figura 3.29 Secciones que nos ofrece la barra de Herramientas.

Dispositivos:

- Dispositivos
- Grupos
- Tipos
- Funciones

Datos:

- Informes de
- Eventos de
- Paneles

Usuarios:

- Usuarios
- Roles
- Organizaciones

Aplicaciones:

En este apartado nos muestra información de Ubidots para aplicaciones.

Paso 3 Dar de alta el dispositivo y presentación de Datos:

Seleccionamos “Dispositivos” y nos dirigimos en “Agregar un nuevo Dispositivo”.



El botón para agregar un nuevo dispositivo, lo encontramos en la parte superior derecha, de la sección Dispositivos.

Figura 3.30 Sección para agregar un nuevo dispositivo.

Cuando presionamos el botón de Agregar un nuevo dispositivo, tendremos que seleccionar el tipo de conectividad, la opción que nos interesa es la de “Ethernet” ya que es la shield que usara el Arduino, la opción está marcada con un rectángulo rojo.

Así mismo tendremos que agregar el tipo de dispositivo, Donde seleccionaremos “Dispositivo en Blanco”, marcado con un cuadro azul.

Figura 3.31 Selección del tipo de dispositivo.



Al presionar “Dispositivo en Blanco”, tendremos que llenar los siguientes campos, donde nos aparecen dos opciones, El Nombre del dispositivo y Etiqueta del dispositivo, que nos ayudara a identificarlo.

Figura 3.32 Configuración del dispositivo.

Después de haber creado nuestro Dispositivo, nos aparecerá una lista de los dispositivos que tenemos.

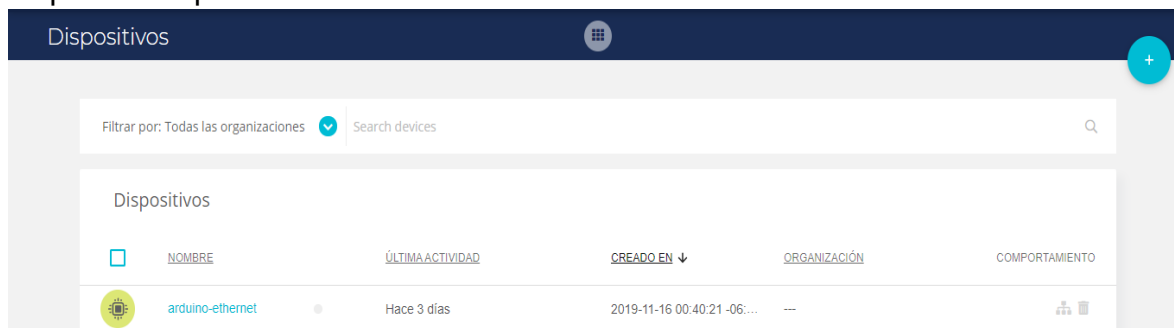


Figura 3.33 Dispositivos agregados.

Como podemos observar cree un nuevo dispositivo llamado “arduino-ethernet”, el cual será el dispositivo que enviara los datos al foro, en esa sección nos aparece más información como son la ultima actividad, la fecha de creación del dispositivo, organización y comportamiento del dispositivo.

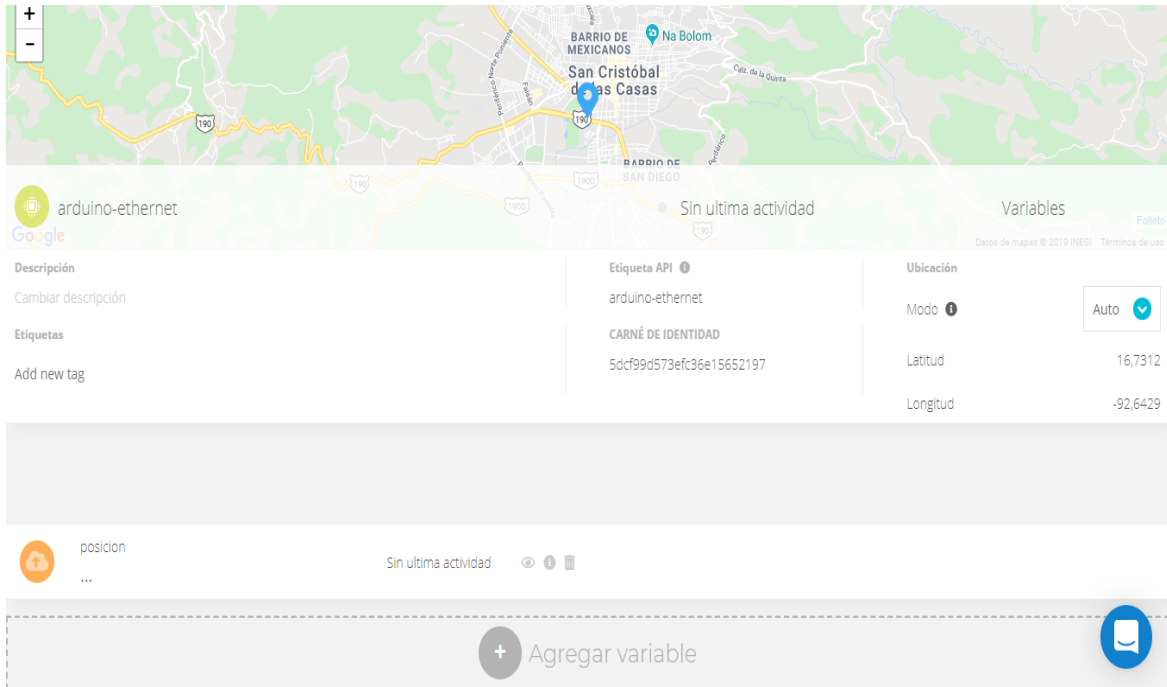


Figura 3.34 Entorno de desarrollo de nuestro proyecto.

Al ya crear nuestro dispositivo, obtendremos nuestro espacio de trabajo, donde tendremos que crear las variables a monitorear, en este caso e creado la variable llamada “posición”, esta variable, recibirá la latitud y longitud enviada por el microcontrolador para ubicar la falla.

En la Sección de Ubicación tenemos el Modo “Auto”, este modo creara un punto en el mapa, con las coordenadas que recibirá la variable “posición”, este punto que se creara a partir de las coordenadas, será donde se encuentre la falla para un transformador de distribución.

3.3.2 ACONDICIONAMIENTO DE LAS SEÑALES.

Los sensores SCT-013-000 están construidos como un transformador de corriente, que proporciona una medición proporcional a la corriente que atraviesa el circuito. Este tipo de medición indirecta, se le conoce como inducción electromagnética.

Cuando una corriente circula por el devanado primario, gracias al efecto de la inducción magnética, en el devanado secundario se produce una intensidad de corriente proporcional a la que pasa por el devanado primario

Un efecto importante dentro de los transformadores de corriente es el número de espiras o vueltas que da el cable al núcleo ferromagnético. Conociendo estos datos y la corriente que circula por uno de los devanados podemos calcular la corriente por el otro devanado.

Esto debido a la relación que tienen:

$$\frac{N_p}{N_s} = \frac{I_s}{I_p} = \frac{V_p}{V_s}$$

A esta fórmula se le llama relación de transformación. Relaciona el número de espiras del primario N_p , del secundario N_s , las intensidades del primario I_p , del secundario I_s , el voltaje primario V_p y del secundario V_s .

En el caso del sensor de corriente alterna SCT-013-000 el devanado primario es el cable del aparato que queremos medir y el número de vueltas sería uno.

El devanado secundario tiene 2,000 vueltas.

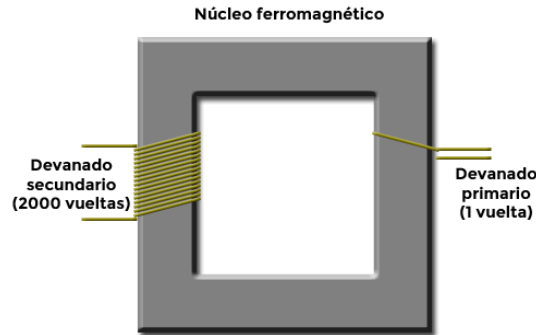


Figura 3.35 Núcleo ferromagnético

Si aplicamos la relación de transformación a esta situación y suponiendo que queremos medir una intensidad de 10 Amperes, el resultado sería el siguiente.

$$\frac{N_p}{N_s} = \frac{I_s}{I_p} = I_s = \frac{N_p \times I_p}{N_s} = \frac{1 \times 10}{2000} = 0.005A$$

Como podemos comprobar se produce una transformación de energía. Hemos bajado de 10 A hasta los 0.005 A.

Al final de cuenta es como si estuviéramos haciendo un cambio de escala para poder medir mejor.

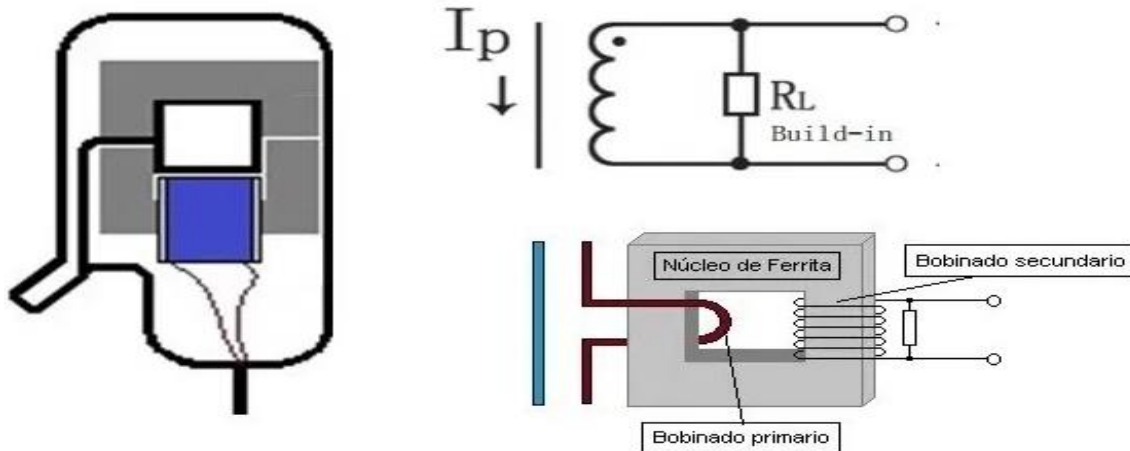


Figura 3.36 Composición del sensor SCT-013-000

El SCT-013-000 es el único que nos proporciona una corriente y no tiene resistencia de carga. Puede medir una corriente de entre 50 mA y 100 A, con Arduino solo podemos medir voltajes. Si Además ese voltaje varía entre un mínimo y máximo, solo podremos hacerlo a través de una entrada analógica.

Por otro lado, para los cálculos correspondientes debemos conocer el consumo aproximado del aparato a medir. Esto nos permitirá ajustar la precisión del sensor.

Cálculos para el acoplamiento de las señales:

- **Cálculo de la corriente pico que vamos a medir:**

Si estamos trabajando con una potencia de 100W y un voltaje típico de 127V, como ya sabemos la formula de potencia, calculamos la corriente pico

$$P = V \times I = I = \frac{P}{V} = \frac{100W}{127V} = 0.7874 \text{ Ampers}$$

- **Convertir la corriente pico en corriente eficaz:**

La corriente que hemos obtenido en el paso interior es la corriente pico, ahora hay que convertirla en corriente eficaz.

$$I_{RMS} = \frac{I}{\sqrt{2}} = \frac{0.7874 A}{1.4142} = 0.5567 \text{ Ampers}$$

- **Calcular la corriente pico en el devanado secundario:**

Con la formula de relación de transformación que hemos visto antes, podemos calcular la corriente pico en el devanado secundario.

$$\frac{N_p}{N_s} = \frac{I_s}{I_p} = I_s = \frac{N_p \times I_p}{N_s} = \frac{1 \times 0.5567}{2000} = 278.35 \mu A$$

- **Desplazamiento de la señal:**

Sin el offset en DC tendríamos nuestra señal senoidal que varía de 2,5V a -2.5V.

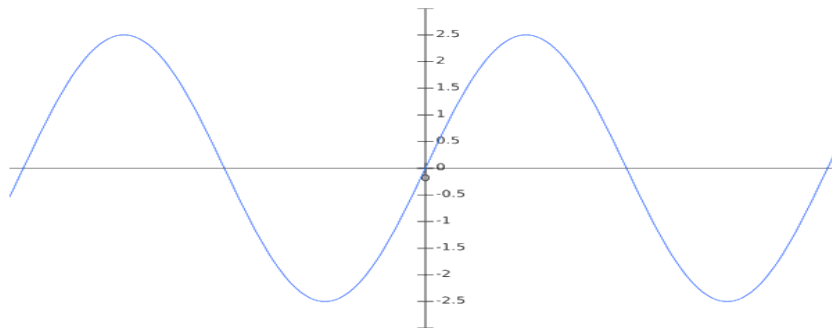


Figura 3.37 Comportamiento de la señal sin un offset en DC

El problema que tenemos es que ningún Arduino lee voltajes negativos. Así que tenemos que modificar la señal para que pase a estar en el rango de 0V a 5V. Esto se hace añadiendo un offset en DC a nuestra señal.

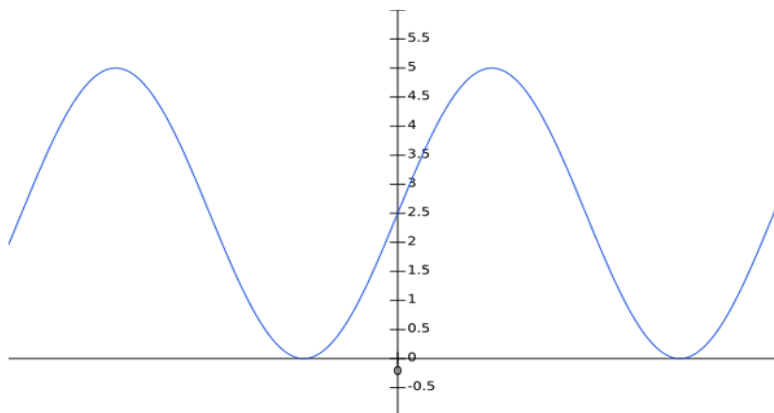


Figura 3.38 Comportamiento de la señal con offset en DC

Para realizar esto debemos sumar 2.5V para que el problema quede resuelto. Esto lo haremos con un circuito sencillo conocido como circuito offset en DC. El circuito consiste en poner un divisor de tensión y un condensador. El condensador tiene que ser de $10\mu\text{F}$. Esto hace que la reactancia sea baja y la corriente alterna evite la resistencia. Como el dispositivo que usaremos sera el arduino usaremos resistencias de $470\text{K}\Omega$ para que el consumo sea minimo.

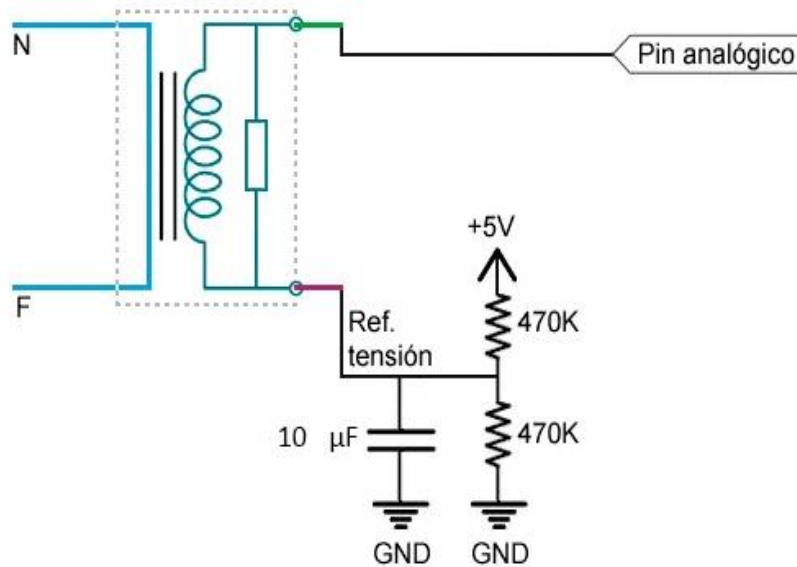


Figura 3.39 Circuito Offset DC empleado al sensor.

Colocación del sensor:

Para este paso no es posible colocarlo de cualquier forma, hay que hacerlo de una manera determinada, ya que, de no hacerlo, no tendremos mediciones correctas. Tendremos que verificar que los cables de lo que vayamos a medir estén al descubierto que serian la fase y el neutro, el sensor se debe colocar únicamente en la fase, esto debido a que por uno de los cables la corriente fluye en un sentido y por el otro cable fluye en sentido contrario. Si se introducen ambos cables en el sensor un flujo magnético compensará al otro flujo magnético y se anulará.

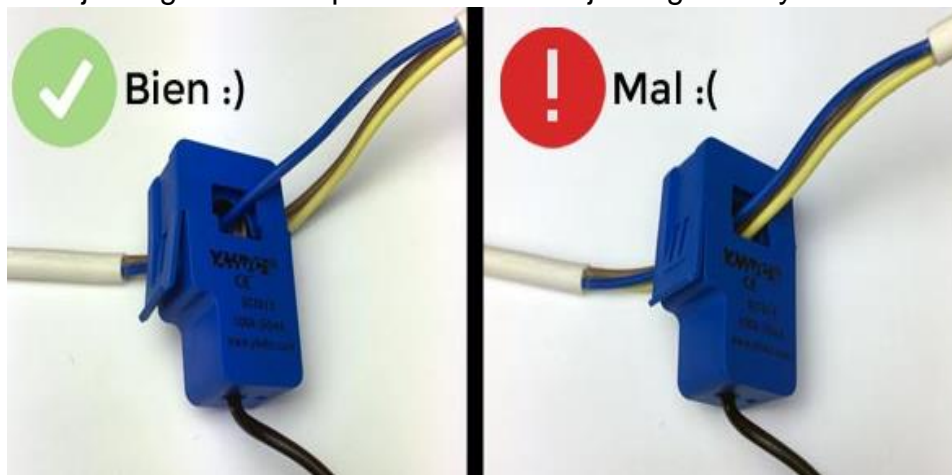


Figura 3.40 Como colocar el Sensor SCT-013-000.

3.4 PROGRAMACIÓN DEL SISTEMA, PARA LA PRESENTACIÓN Y OBTENCIÓN DE LOS DATOS RECABADOS, ASÍ COMO ESTABLECER EL PROTOCOLO DE COMUNICACIÓN.

3.4.1 PROTOCOLO DE COMUNICACIÓN.

Con la Shield Ethernet se abren innumerables opciones para controlar el Arduino a través de Internet. Domótica, automatización, Internet de las cosas (IoT), control y monitoreo remoto, etc. Son algunos de los campos donde se puede utilizar esta Shield. La Shield posee un conector RJ45 estándar para ethernet, al mantener una conexión estable con internet es mas viable que una comunicación inalámbrica que puede fallar o no ser estable en la transmisión y recepción de datos.

3.4.2 PROGRAMACIÓN DEL CIRCUITO RECEPTOR DE DATOS (Rx).

```
#include <Ethernet.h>
#include <SPI.h>
#include <UbidotsEthernet.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <SoftwareSerial.h>
SoftwareSerial SIM800L (8,9);
String ubicacion = " ";
float falla;
LiquidCrystal_I2C lcd(0x27,20,4);
```

En la parte superior de nuestro sketch, debemos declarar las librerías para los componentes que usaremos, en este caso la de la Shield Ethernet, la de la LCD, el módulo receptor y la librería para la comunicación con la plataforma web.

```
char const * TOKEN = "BBFF-NY7uRFfrTDVLYrh2qgSScaHwTiX6cb";
char const * VARIABLE_LABEL_1 = "posicion";
```

Estos son los datos de la plataforma web, donde enviaremos los datos.

```
byte mac [] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
Ubidots client (TOKEN);
```

Esta es la dirección MAC de la Shield Ethernet.

```
void setup () {
  lcd.init ();
  lcd.backlight ();
  lcd.clear ();
  lcd.begin (20,4);
  lcd.setCursor (0,0);
  lcd.print ("Monitoreando...");
  pinMode (6,OUTPUT);
  pinMode (5,INPUT);
  Serial.begin (9600);
```

```

Serial.print(F("Iniciando conexion ethernet..."));
if(!Ethernet.begin(mac)){ Serial.println(F("Hubo un problema"));
  } else {Serial.println(Ethernet.localIP());}
  delay (2000);
  Serial.println(F("Placa Ethernet Lista"));
  SIM800L.begin(9600);
  SIM800L.println("AT+CMGF=1");
  delay(100);
  SIM800L.println("AT+CNMI=1,2,0,0,0");}

```

En el void Setup preparamos nuestra LCD para que sea capaz de mostrarnos los datos que recibimos, declaramos las entradas y salidas, iniciamos la comunicación serial de 9600 Baudios, preparamos nuestra Shield ethernet y verificamos que la comunicación sea correcta y le indicamos a nuestro módulo SIM800L que recibirá mensajes de texto.

```

void loop () {
  Ethernet.maintain();
  if (SIM800L.available(>0) {
    ubicacion = SIM800L.readString();
    Serial.println(ubicacion);}

```

En el void loop iniciamos nuestro módulo SIM800L he indicamos que los valores que reciba el módulo, se envíen a la variable ubicación, ya que es la variable que enviaremos a la plataforma web.

```

if(ubicacion.indexOf("16.73149-92.64257")>=0) {
  digitalWrite(6,HIGH);
  falla=16.71700-92.64086;
  delay (1000);}

```

En esta condición verificamos los datos recibidos por el transmisor, si las coordenadas coinciden con la posición del transformador, le otorga las coordenadas a la falla y enciende un led indicador.

```

lcd.clear ();
lcd.setCursor (0,1);
lcd.print(ubicacion);
delay(500);
client.add(VARIABLE_LABEL_1, falla);
client.sendAll();
delay (4000);}

```

Por último, limpiamos nuestra LCD, y escribimos los datos recibidos, como son la latitud, longitud, hora que recibimos el mensaje, y del dispositivo que lo recibimos, para tener mas información de la falla.

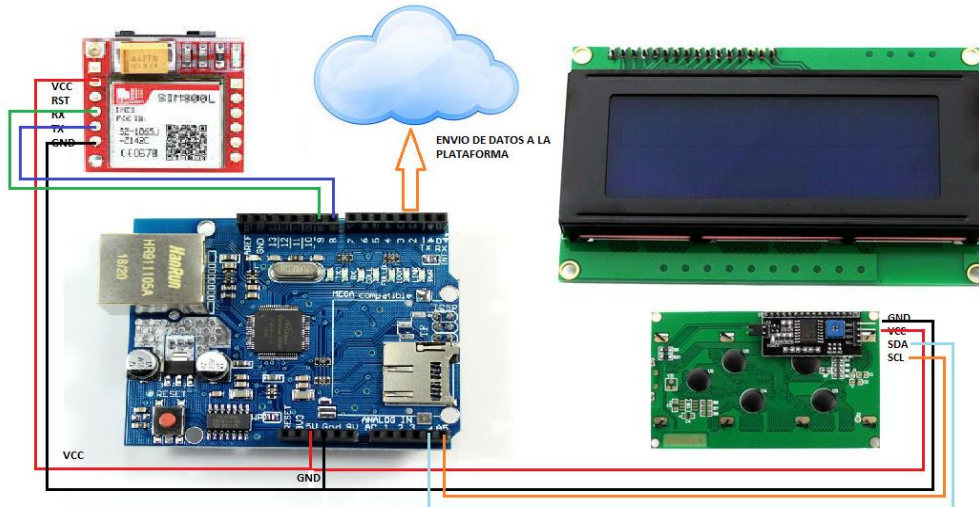


Figura 3.41 Diagrama esquemático del circuito Receptor (Rx).

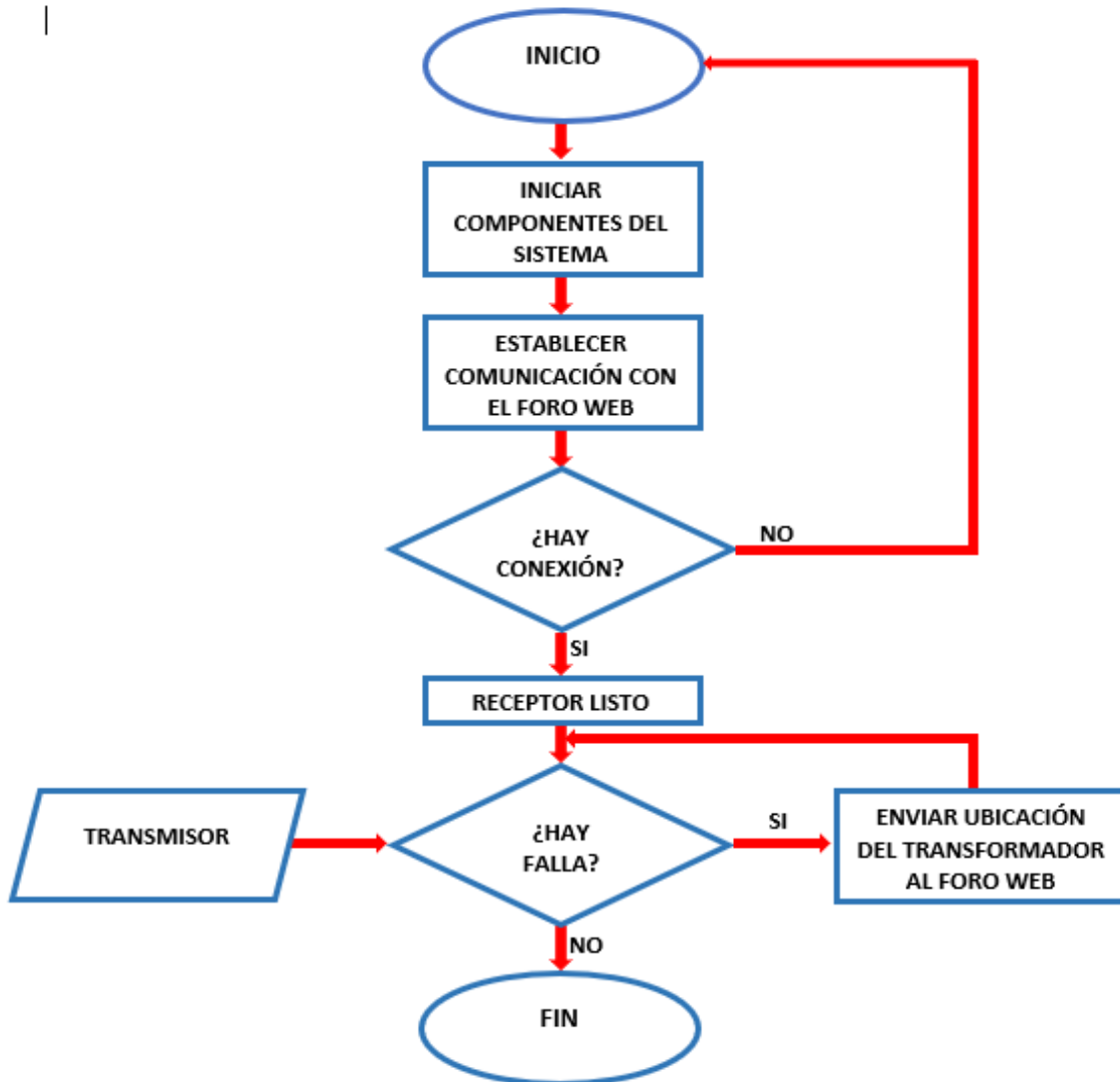


Figura 3.42 Diagrama de flujo en la programación del Receptor (Rx).

3.4.3 PROGRAMACIÓN DEL CIRCUITO TRANSMISOR (Tx).

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include "EmonLib.h"
```

En la parte superior del sketch declaramos las librerías que necesitaremos, que en este caso serían las del GPS, la del módulo SIM800L y la de los sensores de medición.

```
EnergyMonitor energyMonitora;
EnergyMonitor energyMonitorb;
EnergyMonitor energyMonitorc;
float voltajeRed = 127.0;
float latitud, longitud;
int contadora = 0;
int contadorb = 0;
int contadorc = 0;
int contadord = 0;
int fasea = analogRead(0);
int faseb = analogRead(1);
int fasec = analogRead(2);
```

En esta parte declaramos el voltaje con el que estaremos trabajando que será de 127 volts, así como las variables latitud y longitud, necesitamos declarar cuatro contadores, estos nos servirán para que cada segundo aumente su valor en uno, al llegar a un valor especificado se realizara una acción, por último, declaramos los pines analógicos al que estarán conectados nuestros sensores que en este caso van desde el analógico 0 hasta el analógico 2.

```
TinyGPS gps;
SoftwareSerial MOD_SIM800L (2,3);
```

Declaramos el objeto "gps" que será donde guardaremos los datos obtenidos por el GPS, usaremos un puerto serial por software, por lo que tendremos que declarar el receptor (rx) y el transmisor (tx), en este caso el pin 2 será el receptor y el pin 3 el transmisor.

```
int year;
byte month, day, hour, minute, second, hundredths;
unsigned long chars;
unsigned short sentences, failed_checksum;
```

Declaramos las variables de las cuales obtendremos los datos, como son el año, mes, día, hora, minuto, segundo, entre otras variables.

```

void setup () {
Serial.begin(9600);
MOD_SIM800L.begin(115200);
While (!Serial);
pinMode (0,INPUT);
pinMode (1,INPUT);
pinMode (2,INPUT);
energyMonitora.current(0 ,0.7);
energyMonitorb.current(1 ,0.7);
energyMonitorc.current(2 ,0.7);
Serial.println("");
Serial.println("GPS GY-GPS6MV2 Leantec");
Serial.println(" ---Buscando señal--- ");
Serial.println("Monitoreando...");
Serial.println("");}

```

En la parte del void setup iniciamos la comunicación serial que será de 9600 baudios, declaramos que los pines analógicos 0,1 y 2 serán entradas, así como el valor de calibración que será de “0.7”, este valor es muy importante ya que será el que hará que las mediciones sean con mayor precisión.

```

void loop () {while (Serial.available())
{int c = Serial.read();
if(gps.encode(c)
{gps.f_get_position (&latitud, &longitud);
Serial.print("Latitud/Longitud: ");
Serial.print(latitud,5);
Serial.print(" ");
Serial.println(longitud,5);
gps.crack_datetime (&year, &month, &day, &hour, &minute, &second,
&hundredths);
Serial.print("Fecha: "); Serial.print(day, DEC); Serial.print("/");
Serial.print(month, DEC); Serial.print("/"); Serial.print(year);
Serial.print(" Hora: "); Serial.print(hour, DEC); Serial.print(":");
Serial.print(minute, DEC); Serial.print(":"); Serial.print(second, DEC);
Serial.print("."); Serial.println(hundredths, DEC);
Serial.print("Altitud (metros): "); Serial.println(gps.f_altitude());
Serial.print("Rumbo (grados): "); Serial.println(gps.f_course());
Serial.print("Velocidad (kmph): "); Serial.println(gps.f_speed_kmph());
Serial.print("Satelites: "); Serial.println(gps.satellites());
Serial.println();
gps.stats(&chars, &sentences, &failed_checksum);
}}

```

En el void Loop tendremos diferentes tareas una de estas serán los datos que obtendremos por medio del GPS, para posteriormente ser enviados en dado caso que haya alguna falla al receptor.

```
double Irmsa = energyMonitora.calcIrms(1484);  
double potenciaa = Irmsa * voltajeRed;  
double Irmsb = energyMonitorb.calcIrms(1484);  
double potenciab = Irmsb * voltajeRed;  
double Irmsc = energyMonitorc.calcIrms(1484);  
double potenciac = Irmsc * voltajeRed;
```

En esta parte es donde se hacen las mediciones, donde obtenemos el valor de la corriente eficaz y la potencia en cada fase, aquí es donde se hará la medición para ver si los datos obtenidos representan una falla.

```
//Condicion 1  
If (Irmsa <1) {contadora++;  
Serial.println("Falla en la Fase a");  
Serial.print("Corrientea: ");  
Serial.println(Irmsa);  
Serial.print("Corrienteb: ");  
Serial.println(Irmsb);  
Serial.print("Corriente c: ");  
Serial.println(Irmsc);  
Serial.print("Tiempo de Falla: ");  
Serial.println(contadora);  
Delay (1000);  
if(contadora>180){  
  contadora=0;}}
```

Esta es nuestra primera condición, que será cuando haya alguna falla en la fase a, al haber incrementa en uno el valor del contadora.

```
//Condicion 2  
If (Irmsb <1) {contadorb++;  
Serial.println("Falla en la Fase b");  
Serial.print("Corrientea: ");  
Serial.println(Irmsa);  
Serial.print("Corrienteb: ");  
Serial.println(Irmsb);  
Serial.print("Corriente c: ");  
Serial.println(Irmsc);  
Serial.print("Tiempo de Falla: ");  
Serial.println(contadorb);
```



```
Delay (1000);  
if(contadorb>180) {  
  contadorb=0;}}
```

Esta es nuestra segunda condición, que será cuando haya alguna falla en la fase b, al haber incrementa en uno el valor del contadorb.

```
//Condicion 3  
If (Irmisc <1) {contadorc++;  
  Serial.println("Falla en la Fase c");  
  Serial.print("Corrientea: ");  
  Serial.println(Irmsa);  
  Serial.print("Corrienteb: ");  
  Serial.println(Irmsb);  
  Serial.print("Corriente c: ");  
  Serial.println(Irmisc);  
  Serial.print("Tiempo de Falla: ");  
  Serial.println(contadorc);  
  Delay (1000);  
  if(contadorc>180) {  
    contadorc=0;}}
```

Esta es nuestra tercera condición, que será cuando haya alguna falla en la fase c, al haber incrementa en uno el valor del contadorc.

```
//Condicion 4  
if ((Irmisc<1) && (Irmsb<1) && (Irmsa<1)) {contadord++;  
  Serial.println("Falla en la Fase c");  
  Serial.print("Corrientea: ");  
  Serial.println(Irmsa);  
  Serial.print("Corrienteb: ");  
  Serial.println(Irmsb);  
  Serial.print("Corriente c: ");  
  Serial.println(Irmisc);  
  Serial.print("Tiempo de Falla: ");  
  Serial.println(contadord);  
  Delay (1000);  
  if(contadord>180) {  
    contadord=0;}}
```

Esta es nuestra cuarta condición, que será cuando haya alguna falla en las tres fases, al haber incrementa en uno el valor del contadord.

```
//Envió de Mensaje de Falla en condicion 1  
if((contadora>180) &&(contadora<183)) {
```

```

contadora=0;
MOD_SIM800L.write("AT+CMGF=1\r\n");
Delay (100);
MOD_SIM800L.write("AT+CMGS=\"+529671387727\"\r\n");
Delay (100);
MOD_SIM800L.println("Latitud/Longitud: ");
MOD_SIM800L.print(latitud,5);
MOD_SIM800L.print(" ");
MOD_SIM800L.println(longitud,5);
Delay (1000);
MOD_SIM800L.write((char)26);
Delay (200);
Serial.println("Latitud/Longitud: ");
Serial.print (latitud,5);
Serial.print(" ");
Serial.println (longitud,5);
Serial.println(¡" Mensaje enviado!");}

```

Esta condición será la que envíe el mensaje de falla, establecemos el formato, que será para que envíe mensajes de texto, el numero de celular, y el contenido del mensaje, en este caso enviaremos la latitud y longitud de la ubicación de la falla, la falla se enviara cuando el contadora llegue a un valor de 180, esto por que 180 es igual a 3 min y el valor incrementa en uno cada segundo, con ese tiempo podemos saber si realmente es una falla, o fue alguna rama que entró y salió de la línea de distribución.

```

//Envió de mensaje de falla en condicion 2
if((contadorb>180) &&(contadorb<183)) {
contadorb=0;
MOD_SIM800L.write("AT+CMGF=1\r\n");
Delay (100);
MOD_SIM800L.write("AT+CMGS=\"+529671387727\"\r\n");
Delay (100);
MOD_SIM800L.println("Latitud/Longitud: ");
MOD_SIM800L.print(latitud,5);
MOD_SIM800L.print(" ");
MOD_SIM800L.println(longitud,5);
Delay (1000);
MOD_SIM800L.write((char)26);
Delay (200);
Serial.println("Latitud/Longitud: ");
Serial.print (latitud,5);
Serial.print(" ");

```

```
Serial.println (longitud,5);  
Serial.println(j;"Mensaje enviado!");}
```

Esta condición será la que envíe el mensaje de falla, en la fase b.

```
//Envió de mensaje de falla en condicion 3  
if((contadorc>180) &&(contadorc<183)) {  
contadorc=0;  
MOD_SIM800L.write("AT+CMGF=1\r\n");  
Delay (100);  
MOD_SIM800L.write("AT+CMGS=\"+529671387727\"\r\n");  
Delay (100);  
MOD_SIM800L.println("Latitud/Longitud: ");  
MOD_SIM800L.print(latitud,5);  
MOD_SIM800L.print(" ");  
MOD_SIM800L.println(longitud,5);  
Delay (1000);  
MOD_SIM800L.write((char)26);  
Delay (200);  
Serial.println("Latitud/Longitud: ");  
Serial.print (latitud,5);  
Serial.print(" ");  
Serial.println (longitud,5);  
Serial.println(j;"Mensaje enviado!");}
```

Esta condición será la que envíe el mensaje de falla, en la fase c.

```
//Envió de mensaje de falla en condicion 4  
if((contadord>180) &&(contadord<183)) {  
contadord=0;  
MOD_SIM800L.write("AT+CMGF=1\r\n");  
Delay (100);  
MOD_SIM800L.write("AT+CMGS=\"+529671387727\"\r\n");  
Delay (100);  
MOD_SIM800L.println("Latitud/Longitud: ");  
MOD_SIM800L.print(latitud,5);  
MOD_SIM800L.print(" ");  
MOD_SIM800L.println(longitud,5);  
Delay (1000);  
MOD_SIM800L.write((char)26);  
Delay (200);  
Serial.println("Latitud/Longitud: ");  
Serial.print (latitud,5);  
Serial.print(" ");
```

```
Serial.println (longitud,5);
Serial.println(¡"Mensaje enviado!");}
```

Esta condición será la que envíe el mensaje cuando tengamos falla en las tres fases.

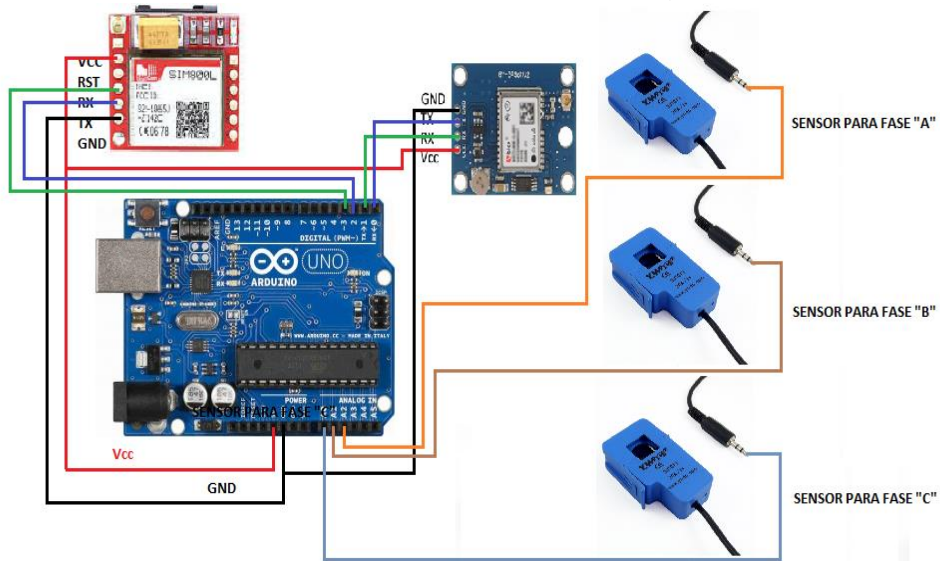


Figura 3.43 Diagrama esquemático del circuito transmisor (Tx).

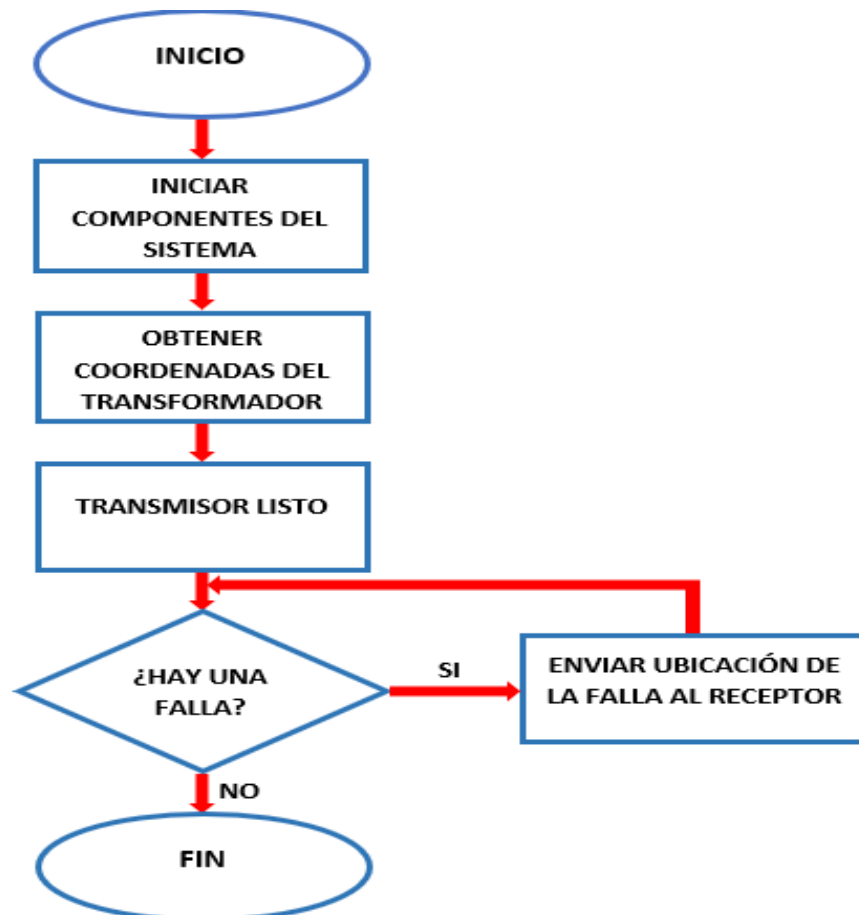
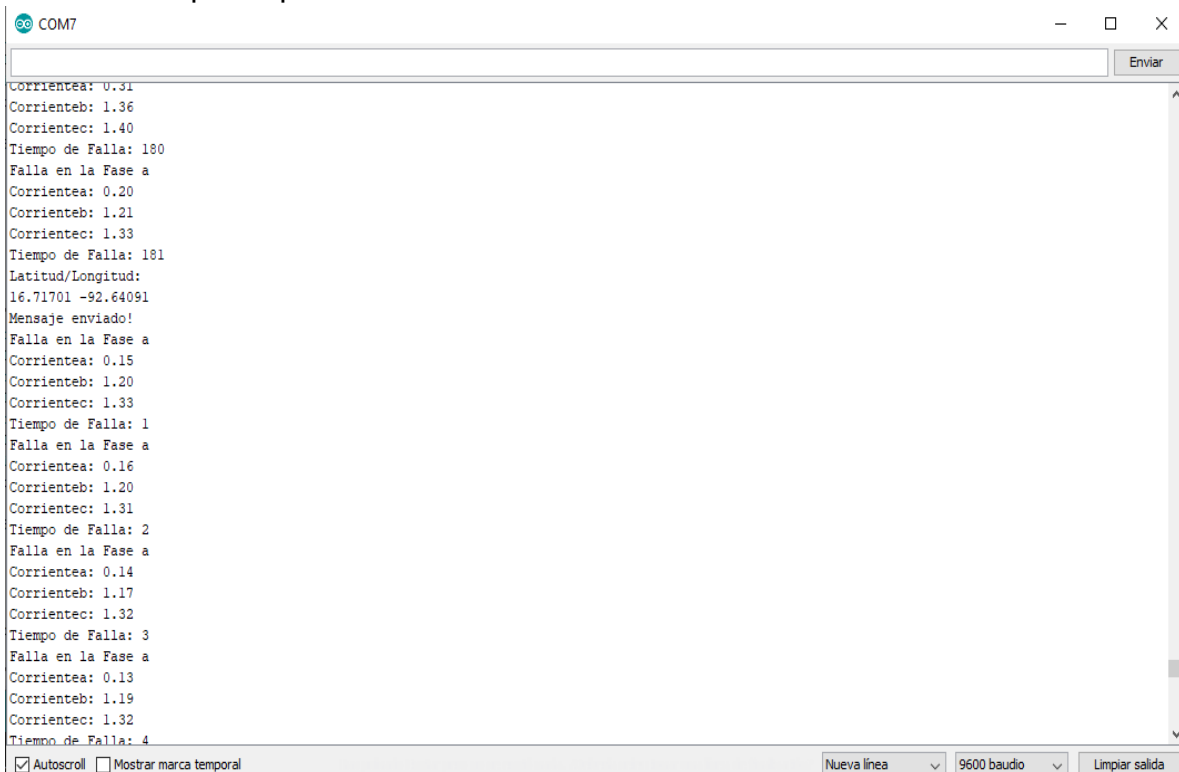


Figura 3.44 Diagrama de flujo en la programación del Receptor (Rx).

CAPITULO IV RESULTADOS Y CONCLUSIONES

Al final del proyecto se lograron cumplir los objetivos que se plantearon al inicio de la residencia, ya que a base de pruebas se logró el objetivo que es la detección de la falla, con esto logramos resolver un problema que cada vez requiere de mayor importancia, para la correcta realización de este proyecto fue necesario volver a repasar materias y fortalecer conocimientos en diferentes áreas ya que el proyecto demandaba estudiar en diferentes aspectos, como parte de esta residencia se seguirá trabajando con este proyecto ya que se espera que esto sea implementado en los servidores de comisión federal de electricidad, y que todas las personas encargadas del mantenimiento puedan llegar más rápido a las fallas, y poder corregirlas, sin estar tanto tiempo buscando donde está la falla, los circuitos, las carcasas y la programación fue una parte fundamental, en el proyecto ya que estos me daban la pauta para la realización del circuito.



```
COM7
Corrientea: 0.31
Corrienteb: 1.36
Corrientec: 1.40
Tiempo de Falla: 180
Falla en la Fase a
Corrientea: 0.20
Corrienteb: 1.21
Corrientec: 1.33
Tiempo de Falla: 181
Latitud/Longitud:
16.71701 -92.64091
Mensaje enviado!
Falla en la Fase a
Corrientea: 0.15
Corrienteb: 1.20
Corrientec: 1.33
Tiempo de Falla: 1
Falla en la Fase a
Corrientea: 0.16
Corrienteb: 1.20
Corrientec: 1.31
Tiempo de Falla: 2
Falla en la Fase a
Corrientea: 0.14
Corrienteb: 1.17
Corrientec: 1.32
Tiempo de Falla: 3
Falla en la Fase a
Corrientea: 0.13
Corrienteb: 1.19
Corrientec: 1.32
Tiempo de Falla: 4
```

Figura 4.1 Resultado obtenido cuando ocurre una falla.

Decidí hacer una prueba, desconectando la carga de la fase “a”, para que el sistema identificara una falla en esa fase e incrementara el valor del contador, como podemos observar en la imagen el tiempo de falla llega a 180 y envía el mensaje, con la latitud y longitud, después el contador regresa a cero, si la falla sigue el contador seguirá incrementando y nos enviara de nuevo el mensaje hasta que la falla haya sido atendida.

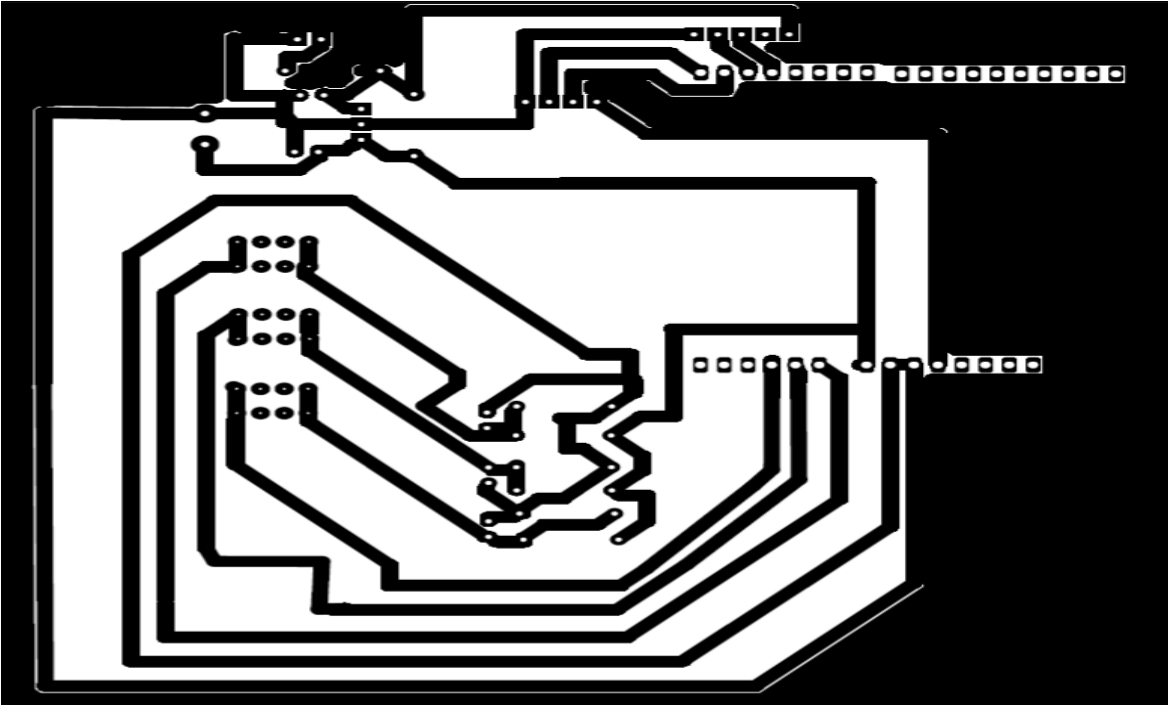


Figura 4.2 Placa PCB del circuito transmisor.

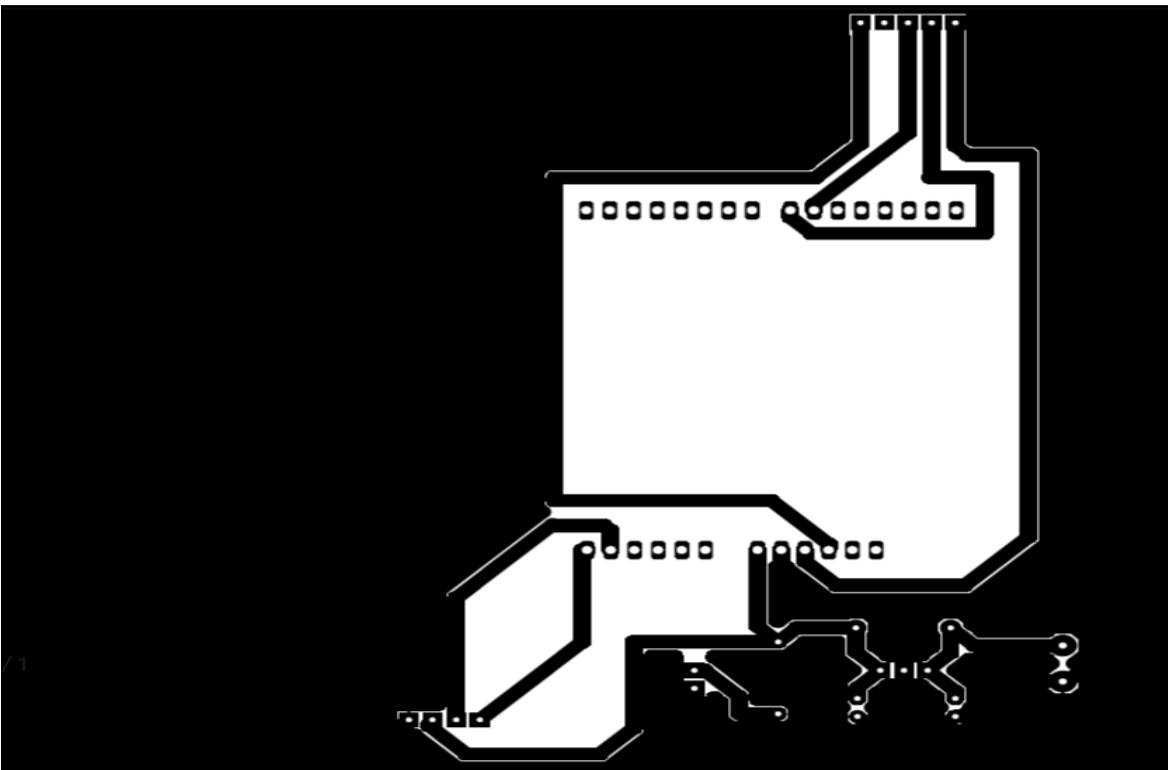


Figura 4.3 Placa PCB del circuito Receptor.

Para el circuito Transmisor y Receptor se hicieron placas PCB, para tener un trabajo con mejor calidad, en el cual buscamos reducir el ruido exterior ya que muchas veces, eso provoca mediciones erróneas por parte del microcontrolador y sea un trabajo más presentable.

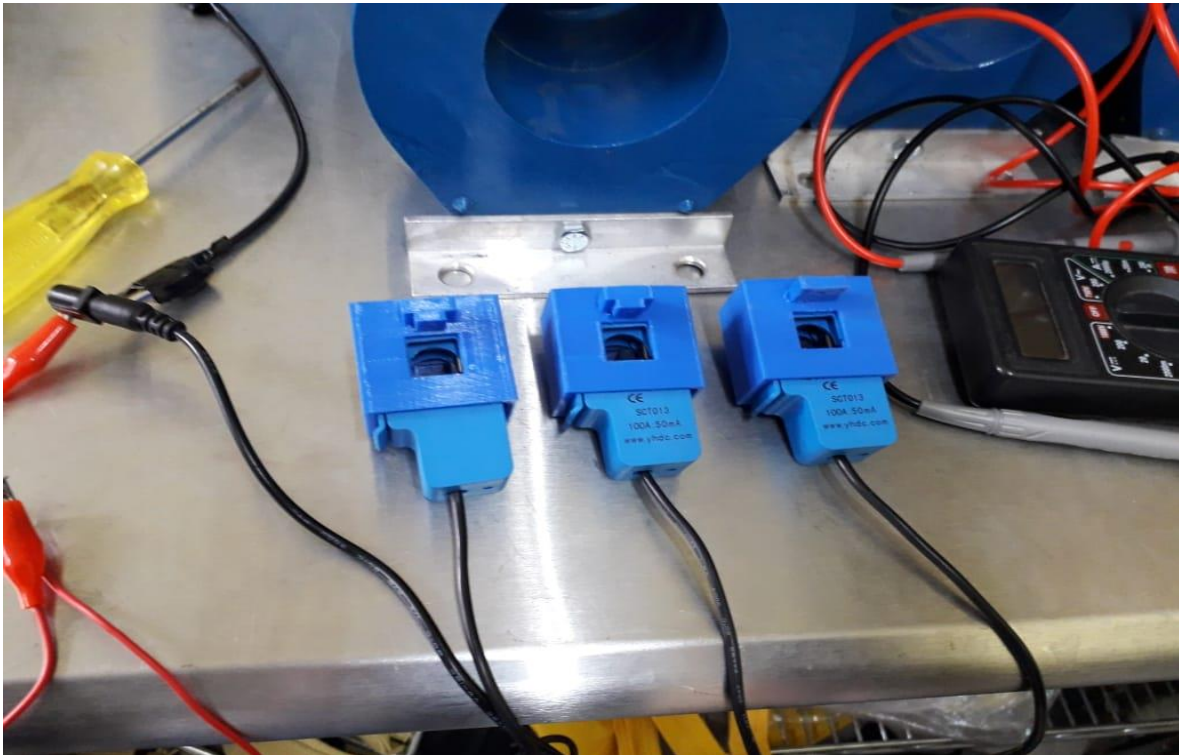


Figura 4.4 Carcasa puesta en sensores.

Se optó por diseñar unas carcasas para los sensores, ya que serán expuestos a diferentes ambientes de trabajo como son lluvia, el sol, polvo, vientos, etc. Y estos pueden hacer que los sensores tengan mediciones erróneas o se descompongan.

CONCLUSIONES:

El monitoreo de Fallas en transformadores de distribución cada día es más importante, ya que una falla trae consigo diferentes problemas, como podrían ser multas económicas, contaminación al medio ambiente, o afectar a personas enfermas que necesitan de la energía eléctrica para estar bien de salud, por eso cada vez se buscan nuevas soluciones para atender estos problemas lo más rápido posible.

Cada día la electrónica es usada para tener nuevas soluciones, ya que se presta para trabajar en cualquier ambiente de trabajo, desde los más robustos hasta los más sencillos, con esto la electrónica nos da diferentes soluciones a nuestros problemas, en este caso para monitorear el comportamiento de cada fase en los transformadores de distribución.

Cuando uno comienza a trabajar un proyecto desde cero, si resulta ser algo complicado ya que se tiene que investigar y tomar cada punto como una consideración, ya que, si no lo tomamos en cuenta, más adelante nos retrasaría, al momento de investigar uno tendrá diferentes opciones para solucionar el problema, pero ahí entra nuestra capacidad de análisis para saber cual es la mejor solución a nuestro problema.

En la residencia profesional me di cuenta que cada materia que cursé a través de mi formación académica tiene que algo que aportar, para mi formación profesional, por eso es fundamental darle la importancia necesaria.

Estar en una empresa Nacional como Comisión Federal de Electricidad aporta mucho a nuestra formación profesional, ya que nos forja para que seamos personas responsables, ya que tenemos un horario y tareas que cumplir, eso va forjando nuestro carácter para ser unos buenos profesionistas.

Participar en un proyecto de esta magnitud y para una empresa como esta, me dejó un gran sabor de boca, ya que es un estímulo anímico para seguir dando lo mejor de mí, y esforzándome cada día más para obtener mejores resultados, como sugerencia me gustaría que más residentes siguieran trabajando en este proyecto, con el trabajo que hice sería una primera etapa, donde aún se pueden hacer mejoras a los circuitos electrónicos, dando mejores resultados para el monitoreo y tener mejores análisis de las fallas en los ramales o troncales con los que cuenta el circuito, y buscar estudiantes de otras carreras como sistemas computacionales, para tener registros de las fallas en diferentes clasificaciones.

COMPETENCIAS DESARROLLADAS

Esta residencia profesional me permitió reafirmar y afinar los conocimientos adquiridos durante la carrera de Ingeniería Electrónica, demostrando la capacidad que tengo para analizar, diseñar, investigar e implementar la electrónica para ayudar a las empresas a encontrar nuevas soluciones a los problemas a los que se enfrentan, en este caso me toco ayudar al departamento de mantenimiento y distribución de comisión federal de electricidad en San Cristóbal de las casas, para ofrecer un mejor servicio a los usuarios. De este modo pude aplicar la ingeniería de manera directa en la parte de diseño y análisis ya que este trabajo es la primera vez que se está realizando y se planea seguir trabajándolo para tener un mejor control en los transformadores eléctricos.

Tabla 5.1 Tabla de las Competencias Desarrolladas y/o aplicadas.

Competencias Desarrolladas y/o Aplicadas	¿Dónde se aplicó?
Analizar	Antes de trabajar en el proyecto fue necesario hacer el análisis de las fallas en transformadores, como son en troncales o ramales, ya que esto nos indica cuales son las fallas a las que estemos expuestas.
Diseñar	Como el sensor que se usará estará expuesto a la intemperie, fue necesario diseñar unas carcasas para proteger el sensor y que haga malas mediciones, los circuitos se diseñaron para tener una placa más profesional y que no tenga fallas.
Investigar	En la búsqueda bibliográfica, ya que antes de realizar el prototipo fue necesaria conocer las características del transformador, así como cuál es su comportamiento para poder detectar la falla.
Implementar.	En la realización del proyecto ya que tuve que aplicar los conocimientos adquiridos en la carrera, ya que nos preparan para solucionar los problemas con que nos encontremos, así como hacer pruebas para ver si lo que estamos haciendo es correcto.

FUENTES DE INFORMACION

GEEKY THEORY. (s.f.). *INTERNET DE LAS COSAS [PARTE 2]* Recuperado de: <https://geekytheory.com/internet-de-las-cosas-parte-2->

DASSAULT SYSTEMES. (s.f.). *Operaciones de Superficie* Recuperado de: http://help.solidworks.com/2013/spanish/solidworks/sldworks/c_Surface_Features_Folder.htm

Tecnología-Informática. (s.f.). *Impresoras 3D ¿Qué son? ¿Cómo Funcionan?* Recuperado de: <https://tecnologia-informatica.com/impresoras-3d-que-son-como-funcionan-impresion-3d/>

AFINIDAD ELÉCTRICA. (s.f.). *Diseción de un medidor electrónico trifásico* Recuperado de: <http://www.afinidadelectronica.com/articulo.php?IdArticulo=188>

Electgpl. (s.f.). *Medidor de potencia AC sin Amplificador Operacional* Recuperado de: <http://electgpl.blogspot.com/2017/01/medidor-de-potencia-ac-sin-amplificador.html>

Electgpl. (s.f.). *Medidor de potencia TrueRMS para 220v* Recuperado de: <http://electgpl.blogspot.com/2016/06/medidor-de-potencia-truerms-para-220vac.html>

Todoproductividad. (s.f.). *Transductores para medir altas corrientes eléctricas AC o DC* Recuperado de: <http://todoproductividad.blogspot.com/2010/03/transductores-para-medir-altas.html>

SciELO. (s.f.). *Diseño y construcción de bobinas de Rogowski modificadas de bajo costo y alto desempeño* Recuperado de: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S2007-07052014000200011

Electgpl. (s.f.). *Sensor de Corriente Alterna* Recuperado de: <http://electgpl.blogspot.com/2017/01/sensado-de-corriente-alterna.html>

RogerBit. (s.f.). *Subiendo datos de temperatura y humedad a Ubidots con módulo ethernet y Arduino* Recuperado de: <http://rogerbit.com/wprb/2019/01/subiendo-datos-de-temperatura-y-humedad-a-ubidots-con-modulo-ethernet-y-arduino/>

Ubidots. (s.f.). *Conecte un Arduino UNO + Ethernet Shield a Ubidots a través de HTTP* Recuperado de: <https://help.ubidots.com/en/articles/513315-connect-an-arduino-uno-ethernet-shield-to-ubidots-over-http>

RogerBit. (s.f.). *Medidor de Corriente no Invasivo IoT, Sensor SCT-013-000* Recuperado de: <http://rogerbit.com/wprb/2017/09/medidor-de-corriente-no-invasivo-iot-sensor-sct-013-030video-colaborativo/>

Programarfacil.com. (s.f.). *SCT-013 mide el consumo eléctrico con Arduino* Recuperado de: <https://programarfacil.com/blog/arduino-blog/sct-013-consumo-electrico-arduino/>

Patagoniatec. (s.f.). *Medidor de corriente AC SCT-013* Recuperado de: <https://saber.patagoniatec.com/2016/08/medidor-de-corriente-30a-sct013-030/>

GeekFactory. (s.f.). *Enviar SMS con Arduino y módulo GSM SIM800L* Recuperado de: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/enviar-sms-con-arduino-y-sim800l/>

Last Minute ENGINEERS. (s.f.). *Enviar Recibir SMS y Llamadas con el módulo SIM800L GSM y Arduino* Recuperado de: <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>

HETPRO. (s.f.). *Sensor de Corriente SCT-013-000 100A 50MA* Recuperado de: <https://hetpro-store.com/sensor-de-corriente-sct-013-000-100a/>

RogerBit. (s.f.). *Medidor de Corriente, frecuencia de red y potencia no invasivo* Recuperado de: <http://rogerbit.com/wprb/2018/08/medidor-de-corriente-frecuencia-y-potencia-aparente-de-red-no-invasivo/>

NAYLAMPmechatronics. (s.f.). *Módulo GPS con Arduino* Recuperado de: https://naylampmechatronics.com/blog/18_Tutorial-M%C3%B3dulo-GPS-con-Arduino.html

Luis Llamas. (s.f.). *Localización GPS con Arduino y los módulos GPS NEO-6* Recuperado de: <https://www.luisllamas.es/localizacion-gps-con-arduino-y-los-modulos-gps-neo-6/>

Programarfacil.com. (s.f.). *GPS Arduino localiza tus proyectos en todo el mundo* Recuperado de: <https://programarfacil.com/podcast/gps-arduino-geolocalizacion/>

Untitled.es. (s.f.). *Enviar SMS con Arduino y SIM800L* Recuperado de: <http://untitled.es/sms-arduino-sim800l/>

ANEXOS

ANEXO 1: Capacidad de Transformadores de distribución en Comisión Federal de electricidad en delta abierta

Tabla 7.1 Tabla de transformadores de distribución en CFE.

CAPACIDAD DE TRANSFORMADORES PARA USARSE EN BANCOS CONECTADOS EN ESTRELLA ABIERTA O DELTA ABIERTA PARA PROPORCIONAR SERVICIOS TRIFASICOS EN LINEAS SECUNDARIAS DE 120/240V												
CARGA TRIFASICA EN KVA												
TRANSFORMADORES FUERZA (KVA)												
TRANSFORMADORES DE ALUMBRADO (KVA)												
CARGA MONOFASICA (KVA)	3	5	10	10	15	15	25	25	25	25	25	37
0	3	5	10	10	15	15	25	25	25	25	25	37
5	5	10	10	15	25	25	25	25	25	25	37	37
10	10	10	15	15	25	25	25	25	25	37	37	50
15	15	15	25	25	25	25	25	37	37	37	50	50
20	15	15	25	25	25	25	25	37	37	37	50	50
25	25	25	25	25	37	37	37	37	37	37	50	50
30	25	37	37	37	37	37	37	37	37	37	50	50
35	37	37	37	37	37	37	37	50	50	50	50	75
40	37	37	37	37	37	50	50	50	50	50	75	75
45	37	37	50	50	50	50	50	50	50	50	75	75
50	50	50	50	50	50	50	50	75	75	75	75	75
60	50	50	75	75	75	75	75	75	75	75	75	100
75	75	75	75	75	75	75	75	75	75	75	75	100
100	100	100	100	100	100	100	100	100	100	100	100	167

El Tamaño de los transformadores es el mínimo para la carga indicada, la tabla está basada en valores del factor de potencia de 0.80 para la carga trifásica y de 0.95 para la carga monofásica.

Estos son algunos de los valores en cargas de transformadores de distribución en Comisión Federal de Electricidad.

ANEXO 2: Conductores para conexión de Transformadores

Tabla 7.2 Conductores para conexión de Transformadores Monofásicos.

TRANSFORMADORES MONOFÁSICOS				
KVA	Corriente Nominal (Amperes)	No. Y Calibre de Conductores de Fase (AWG o KCM)		Conductor Neutro (AWG)
		1 conductor	2 conductores	
10	41	6	-	6
15	63	4	2 N°6	6
25	104	2	2 N°6	4
37.5	156	1/0	2 N°4	2
50	208	3/0	2 N°2	1/0
75	312	250	2 N°1/0	3/0
TRANSFORMADORES TRIFÁSICOS				
KVA	Corriente Nominal (Amperes)	No. Y Calibre de Conductores de Fase (AWG o KCM)		Conductor Neutro (AWG)
		1 conductor	2 conductores	
15	36	6	-	6
30	79	4	2 N 6	6
45	118	2	2 N°6	4
75	197	3/0	2 N°2	1/0
112.5	295	250	2 N°1/0	3/0
150	394	400	2 N°3/0	250

Consideraciones a tomar en cuenta:

- Todas las conexiones de la línea primaria a los apartarrayos, cortacircuitos fusible y de estos al banco de transformación, así como las conexiones para tierra de los apartarrayos y la bajante de tierra, serán con alambre de cobre semiduro desnudo, calibre N°4 AWG.
- Todas las conexiones de los bornes secundarios de las fases del transformador a la línea secundaria será con cable aislado CF-600.
- La conexión del borne neutro del transformador al neutro de la línea secundaria será con cable de cobre semiduro, desnudo y de un calibre normalizado inmediato inferior al de las fases.
- En caso de cambiar un transformador, verifique que los conductores de conexión a la línea secundaria sean los indicados para la capacidad del nuevo transformador.
- La conexión de las salidas del transformador a la línea secundaria se hará como sigue.
- Si la salida del transformador esta formada por un solo conductor, utilice doble conector por cada salida.
- Si la salida del transformador esta formada por dos conductores por fase, utilice un conector por cada cable.

Tabla 7.3 Capacidades de corriente para distintos calibres del cable.

CAPACIDAD DE CONDUCCIÓN DE CORRIENTE DEL CABLE CF-600	
CALIBRE (AWG o KCM)	AMPERES
6	95
4	125
2	170
1/0	230
3/0	310
250	405
300	445
400	545

ANEXO 3: Codificación en Transformadores

La codificación de los bancos de transformación consta de cinco campos:

- 1._ En el primer campo se indica el número de unidades que componen el transformador.
- 2._ En el segundo y tercer campo se indica el tipo de equipo requerido (TR).
- 3._ En el cuarto campo se indica el número de fases a las que está conectado el transformador.
- 4._ En el quinta campo se indica el sistema de distribución de media tensión: A, B o R.

a) Un Transformador trifásico, conectado a un sistema 3F-4H se codificará:

1	T	R	3	A
----------	----------	----------	----------	----------

Esta tabla indica que se cuenta con (1) transformador, trifásico (TR), Conectado a un sistema de Tres Fases (3), con neutro Corrido (A).

b) Un banco de Transformadores Monofásicos conectados en Delta-Abierta-Delta Abierta en un sistema 3F-3H-

2	T	R	3	B
----------	----------	----------	----------	----------

La Codificación Anterior Indica un banco con dos (2), Transformadores (TR) Monofásicos en un sistema de tres (3) Fases, Tres Hilos (B).

- 5._ El complemento para su total definición es la tensión de utilización del sistema y la capacidad de los transformadores.

ANEXO 4: Conexión de un transformador trifásico (BANCO 1TR3B)

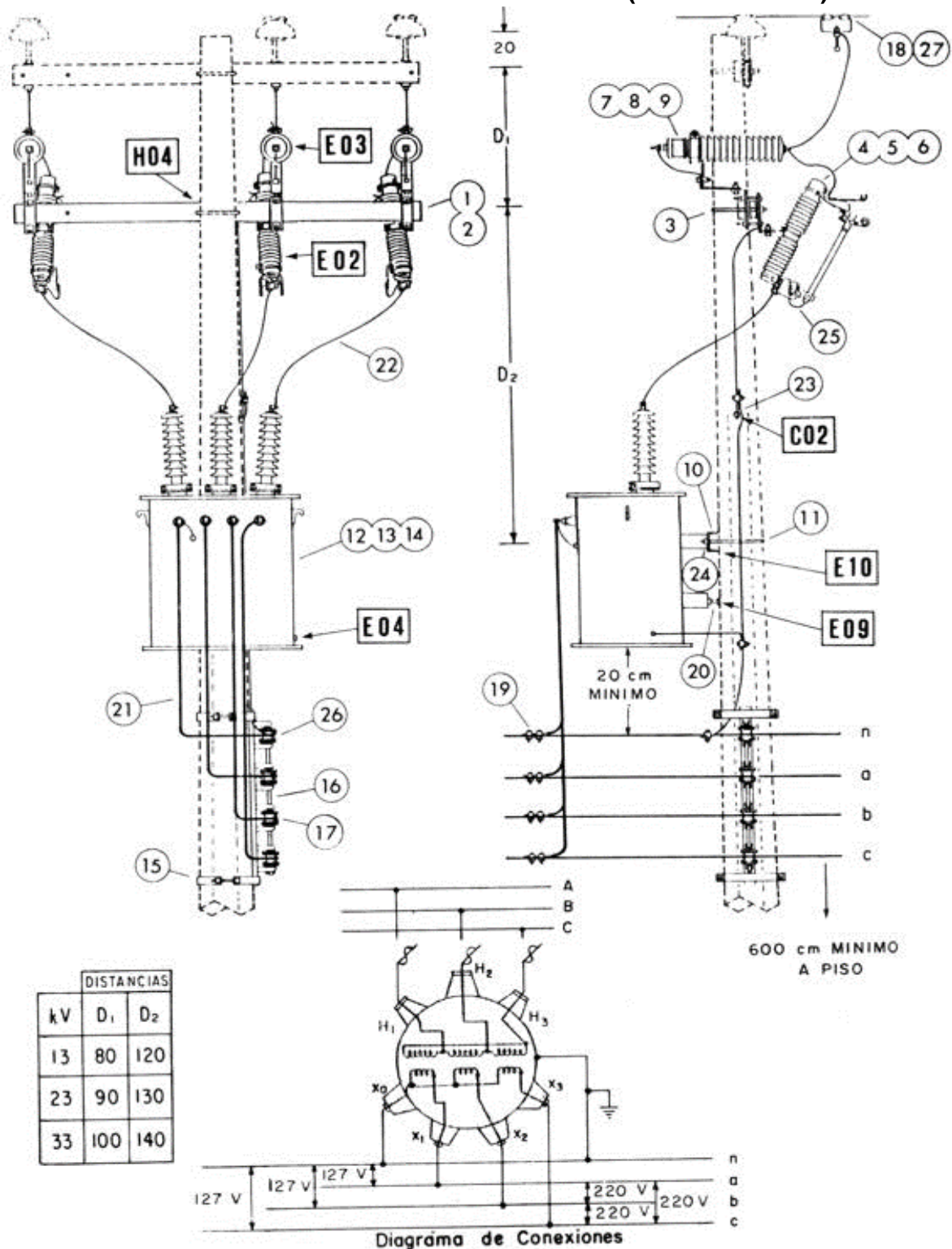


Figura 7.1 Conexión de un transformador trifásico (BANCO 1TR3B)

En esta imagen se muestra el diagrama de conexión que se requiere para la instalación de un transformador, incluyendo el herraje a utilizarse y su respectiva acotación en centímetros para tener una instalación apropiada y eficiente.

Tabla 7.4 Lista de Materiales para la instalación de un transformador trifásico.

LISTA DE MATERIALES						
No.	C.P.I.E.	Unidad	Descripción Corta	Cantidad		
				13KV	23KV	33KV
1	1058A1	Pieza	Cruceta PT200	1	1	-
2	1062A1	Pieza	Cruceta PT250	-	-	1
3	1010A4	Pieza	Abrazadera UC	1	1	1
4	8029A1	Pieza	Cortacircuito Fusible CCF-15-100-95-8000	3	-	-
5	8029A4	Pieza	Cortacircuito Fusible CCF-27-100- 95-8000	-	3	-
6	8029A7	Pieza	Cortacircuito Fusible CCF-38-100-150-2000	-	-	3
7	8003A2	Pieza	Apartarrayos ADA 12	3	-	-
8	8003A4	Pieza	Apartarrayos ADA 18	-	3	-
9	8003A8	Pieza	Apartarrayos ADA 30	-	-	3
10	1162A2	Pieza	Soporte CV1 (2)	1	1	1
11	1010A5	Pieza	Abrazadera UL	1	1	1
12	7024A*	Pieza	Transformador D3*-13200-220Y/127	1	-	-
13	7026A*	Pieza	Transformador D3*-23000-220Y/127	-	1	-
14	7028A*	Pieza	Transformador D3*-33000-220Y/127	-	-	1
15	1008A2	Pieza	Abrazadera 2BS	2	2	2
16	1023A4	Pieza	Bastidor B4	1	1	1
17	200700	Pieza	Aislador 1C	3	3	3
18		Pieza	Conectador Estribo	3	3	3
19	3014A	Pieza	Conectador S/N 07 C0 09	8	8	8
20	9026A2	Pieza	Tornillo Maquina 16x76	2	2	2
21	4022A	Metros	Cable CF-600 S/N 08 TR 07	12	12	12
22	4004A1	Kg	Alambre Cu 4	2	2	2
23		LOTE	Bajante de Tierra S/N 09 00 02	1	1	1
24	1135A1	Pieza	Placa 1PC	3	3	3
25	20**A*	Pieza	Fusible Universal S/N 08 TR 03	3	3	3
26	1030A1	Pieza	Carrete H	1	1	1
27		Pieza	Conectador perico	3	3	3

En la tabla 7.4 se muestra y se marca el equipo necesario para la instalación del transformador tipo poste, este equipo es el necesario para tener un correcto funcionamiento del transformador.

ANEXO 5: Diseño de placas PCB para el circuito Transmisor y Receptor

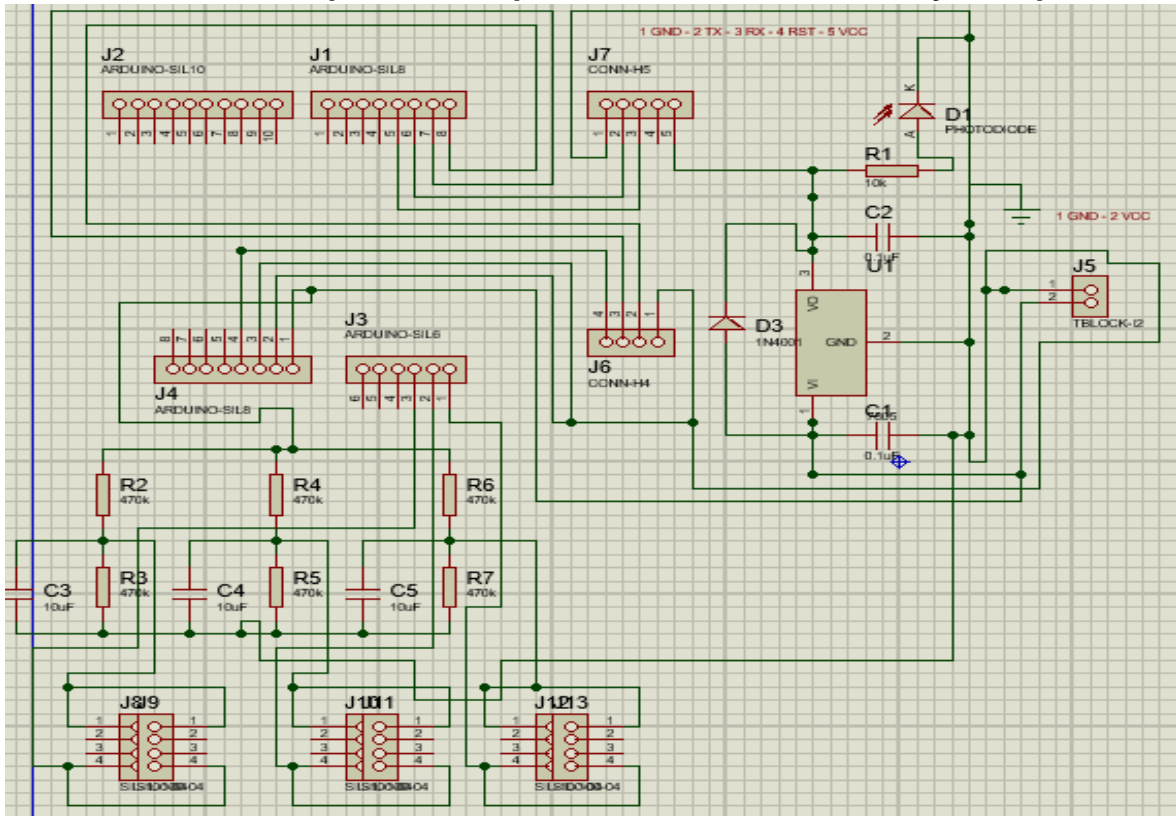


Figura 7.2 Diagrama esquemático para el diseño del circuito transmisor.

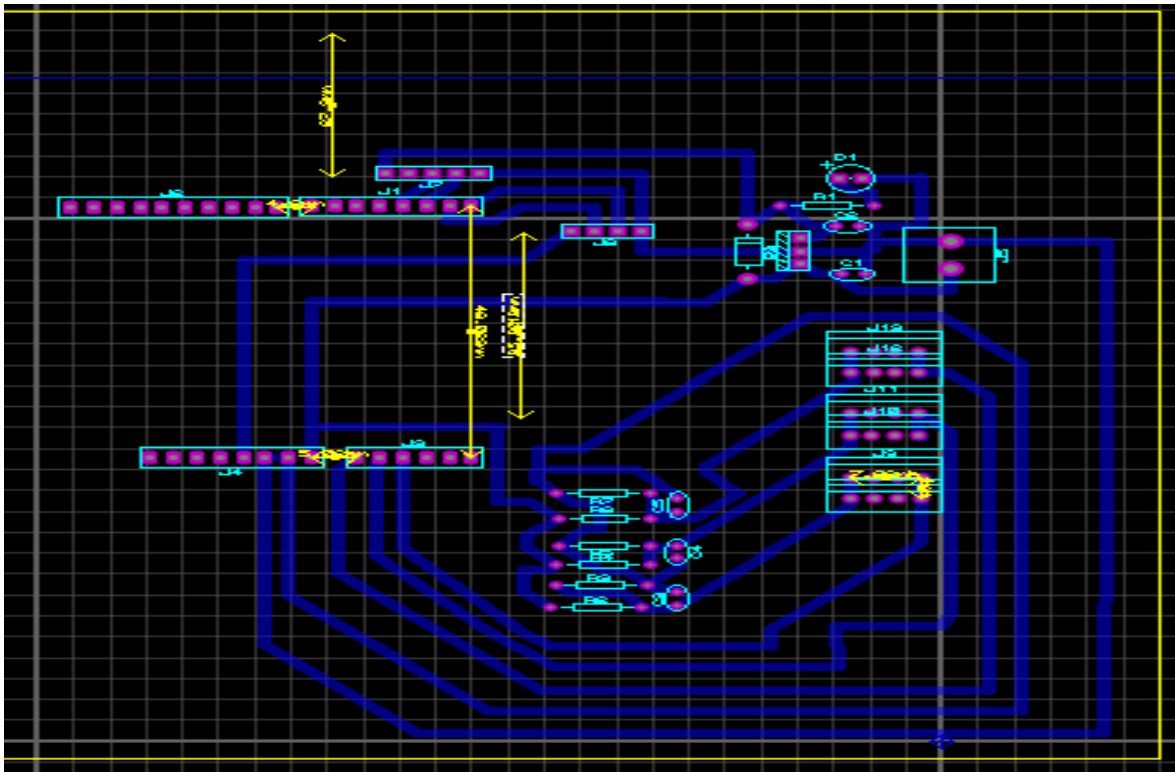


Figura 7.3 Diagrama PCB del circuito transmisor.

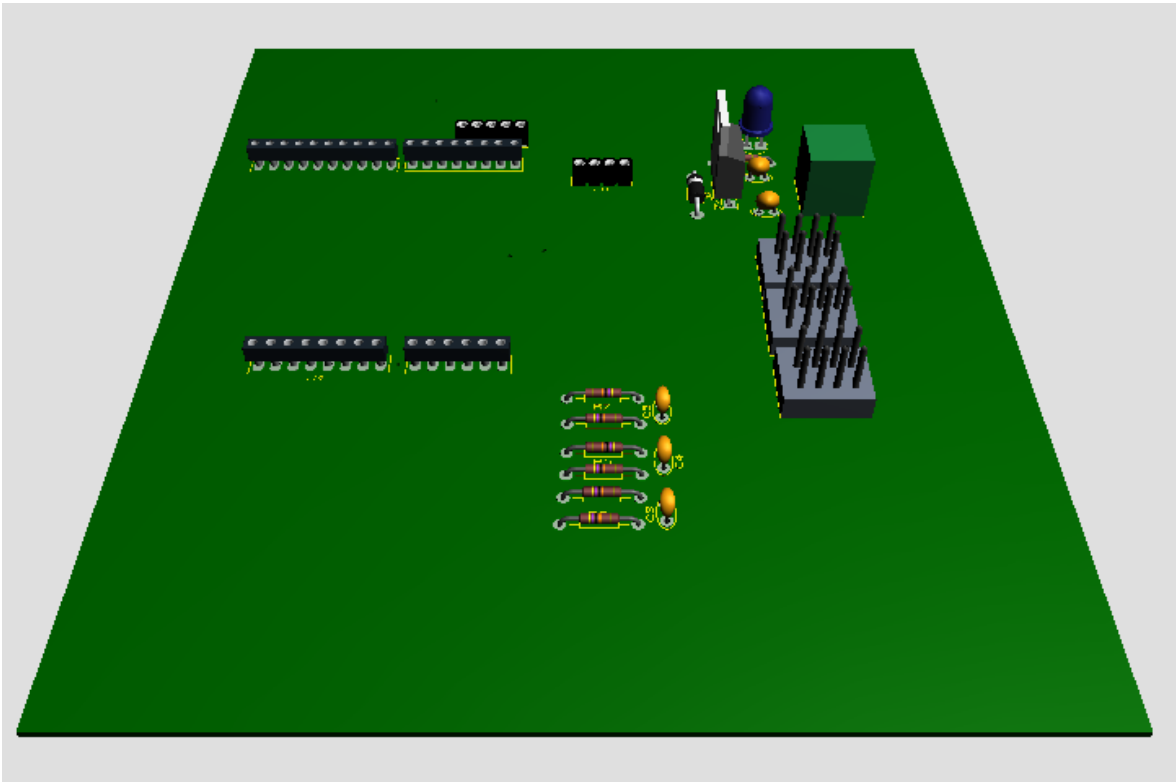


Figura 7.4 Circuito 3D del circuito transmisor vista superior.

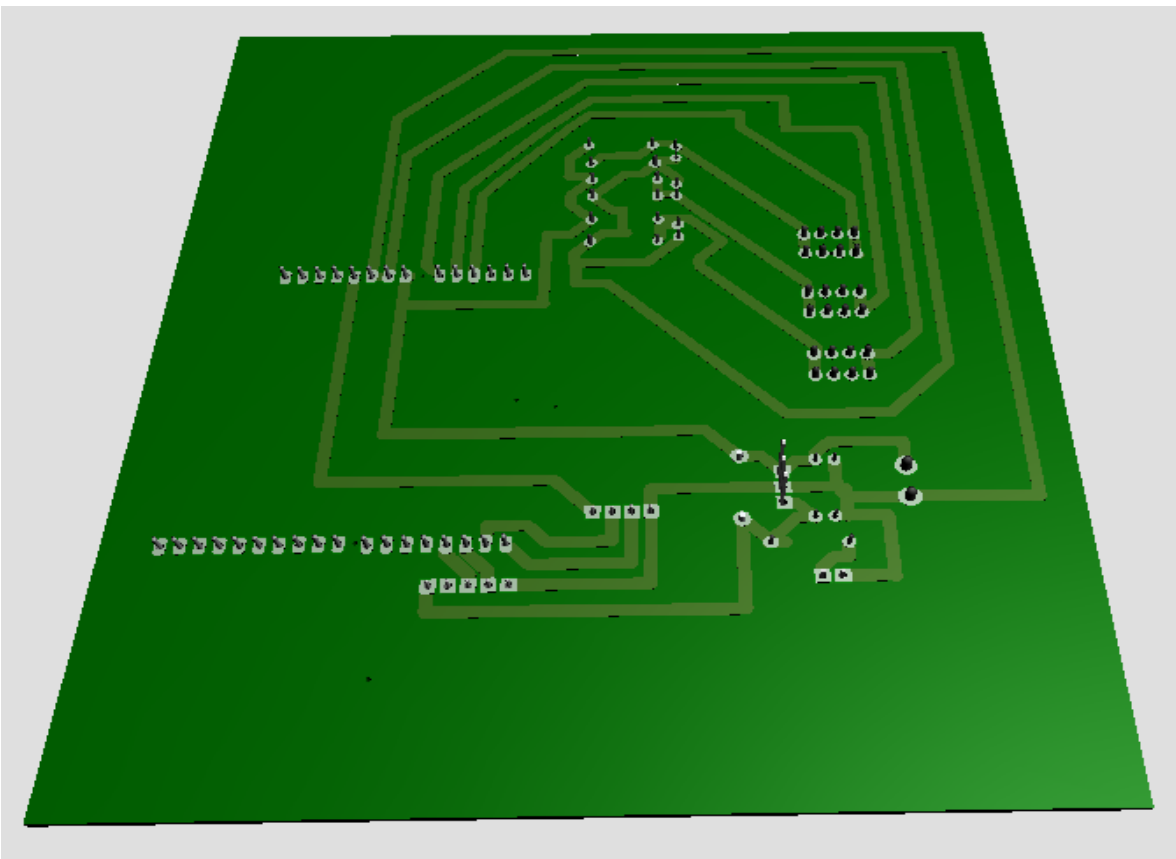


Figura 7.5 Circuito 3D del circuito transmisor vista inferior.

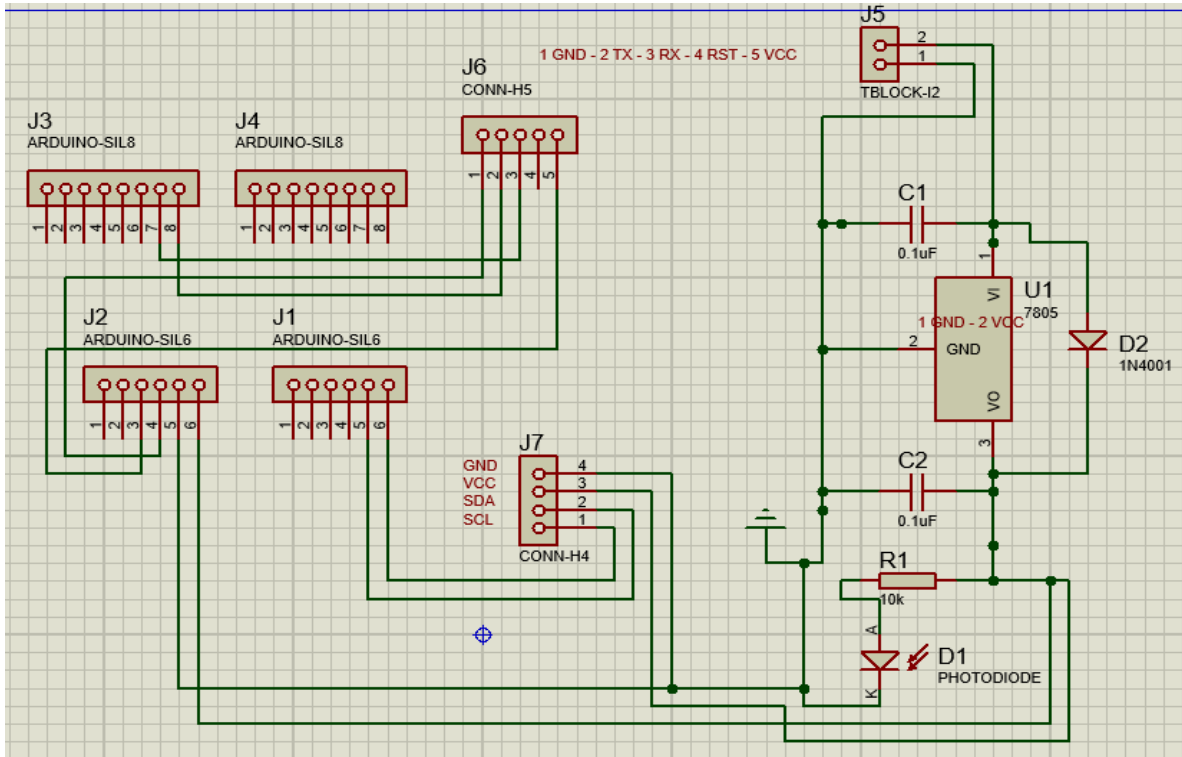


Figura 7.6 Diagrama esquemático para el diseño del circuito Receptor.

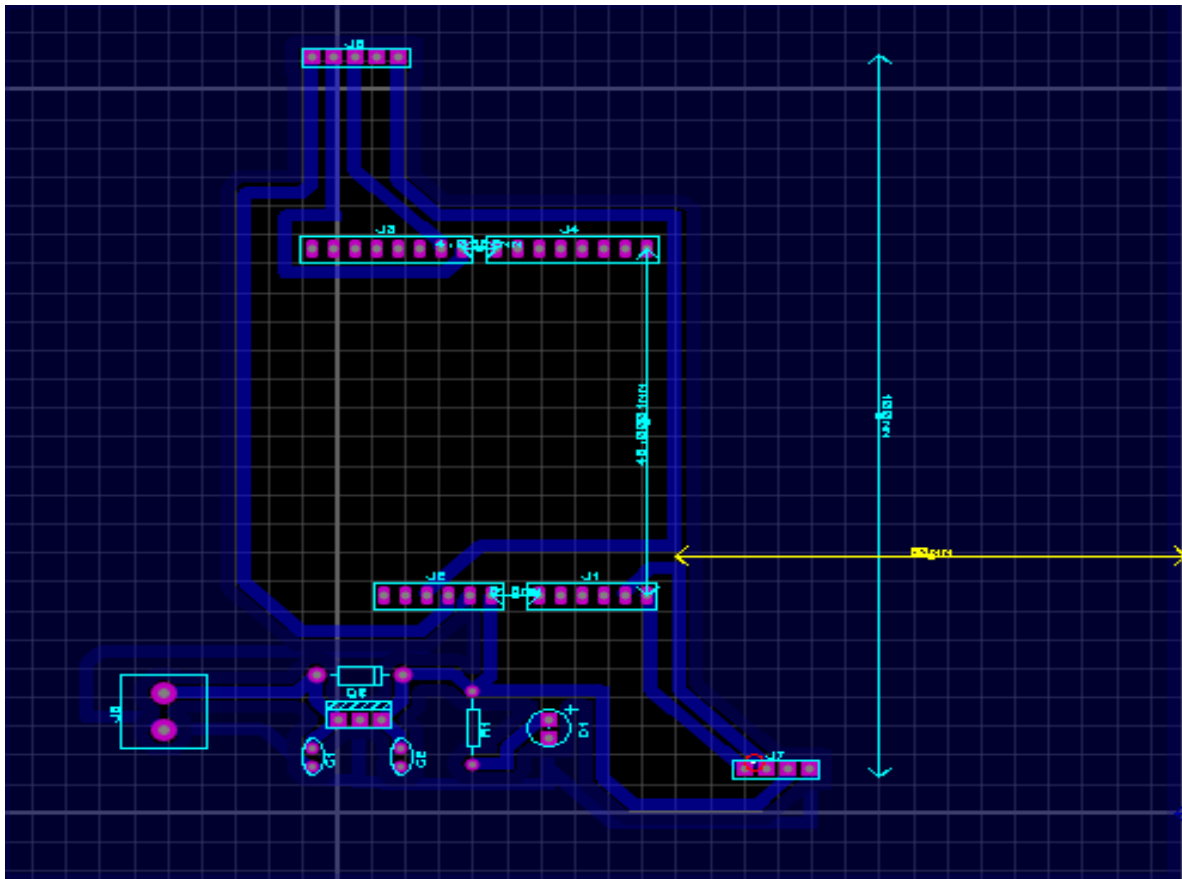


Figura 7.7 Diagrama PCB del circuito Receptor.

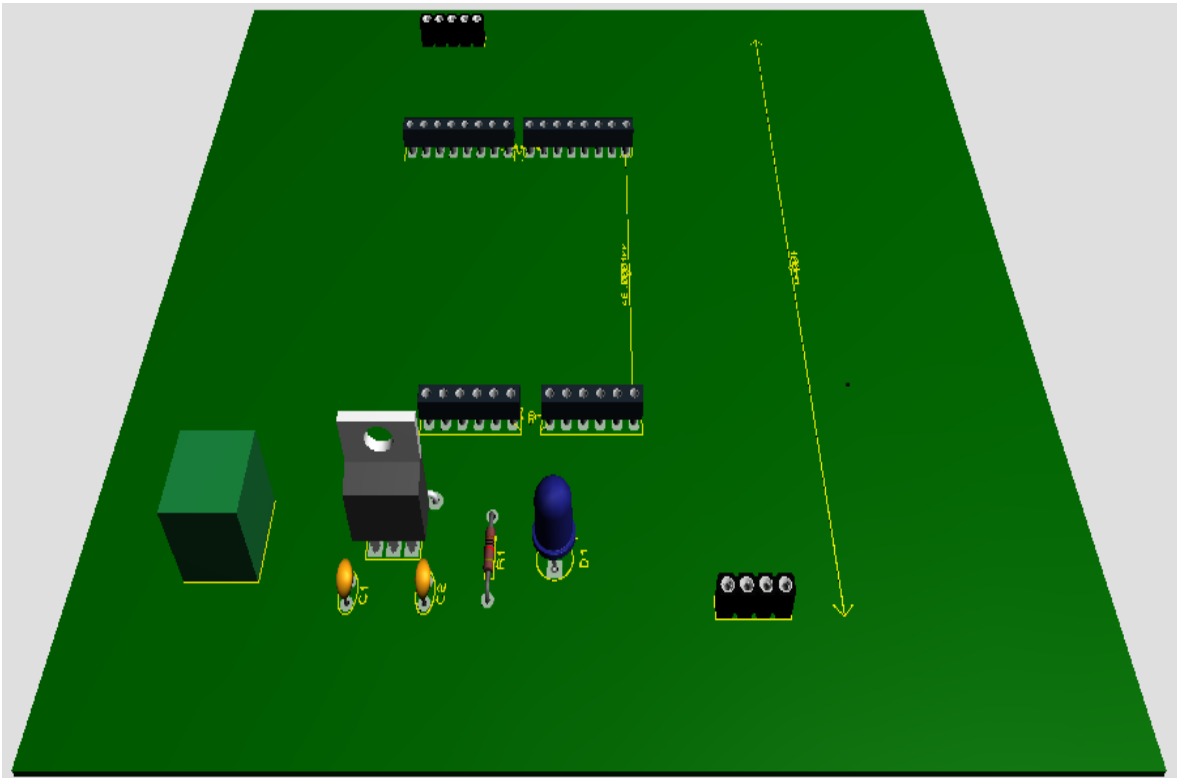


Figura 7.8 Circuito 3D del circuito Receptor vista superior.

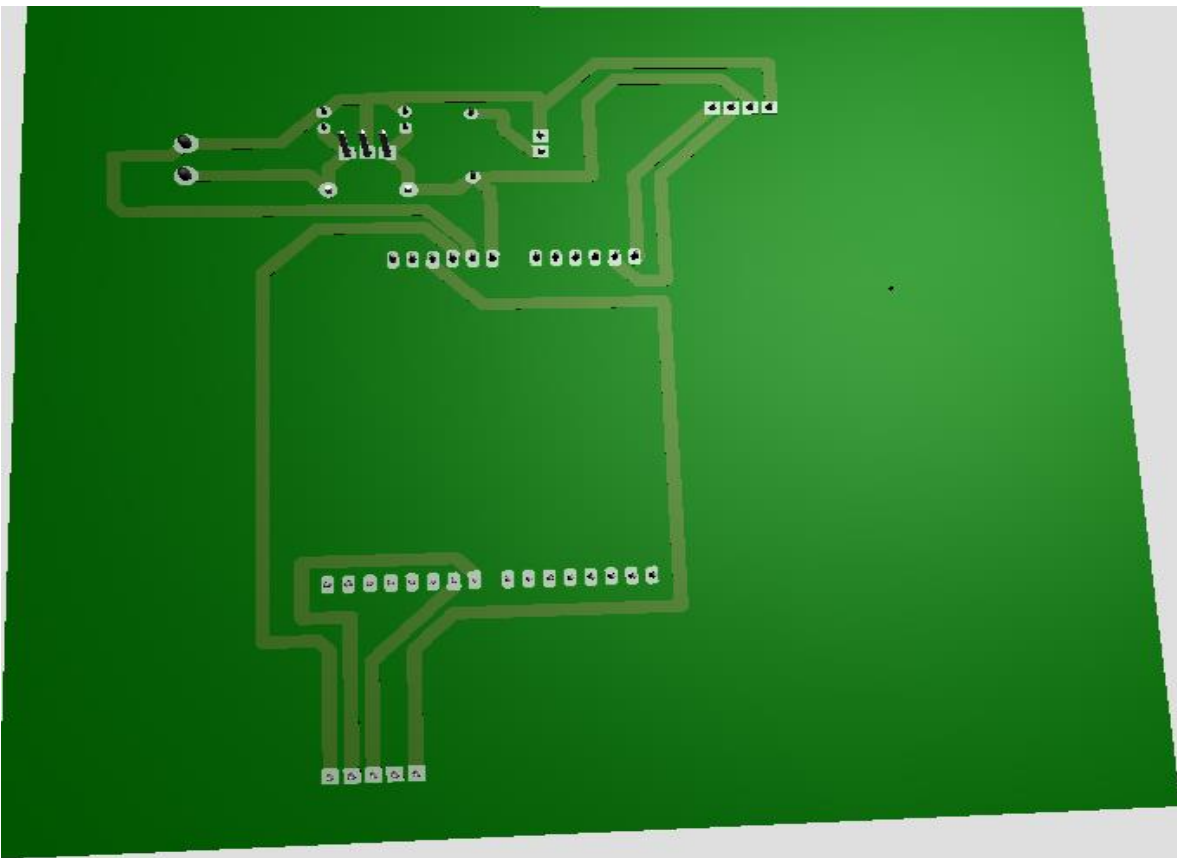


Figura 7.9 Circuito 3D del circuito Receptor vista inferior.