



UNIVERSIDAD NACIONAL DE LOJA

**FACULTAD DE LA ENERGÍA LAS INDUSTRIAS Y
LOS RECURSOS NATURALES NO RENOVABLES**

**CARRERA DE INGENIERÍA EN
ELECTRÓNICA Y
TELECOMUNICACIONES**



**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
MONITOREO Y CONTROL DE VARIABLES FÍSICAS DE UN
GENERADOR ELÉCTRICO Y SISTEMA DE BACKUP DE
ALIMENTACIÓN A TRAVÉS DE UNA APLICACIÓN WEB**

*TESIS DE GRADO PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y TELECOMUNICACIONES*

AUTOR:

DANNY SIXTO ARMIJOS LOYOLA.

DIRECTOR:

ING. RENATO BENJAMÍN TORRES CARRIÓN

**Loja – Ecuador
2022**

CERTIFICACIÓN



CARRERA DE ELECTRÓNICA
Y TELECOMUNICACIONES Y DE
TELECOMUNICACIONES

Certificado No. RTC-UNL-01-2021

ING. RENATO BENJAMÍN TORRES CARRIÓN. Mg.Sc

CERTIFICO:

Que el estudiante señor **Danny Sixto Armijos Loyola**, portador de la cédula de identidad Nro: **1104141377**; ha obtenido un avance del **100%**, respecto a su Trabajo de Titulación denominado: **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE VARIABLES FÍSICAS DE UN GENERADOR ELÉCTRICO Y SISTEMA DE BACKUP DE ALIMENTACIÓN A TRAVÉS DE UNA APLICACIÓN WEB"**, por lo que es **PERTINENTE** dar continuidad a los procesos académicos y administrativos necesarios una vez que es estudiante ha cumplido completamente con el desarrollo de su tesis.

Loja, 11 de Marzo del 2021



RENATO BENJAMIN
TORRES CARRION

Ing. Renato Benjamin Torres Carrión Mg.Sc.
DIRECTOR DE TESIS

AUTORÍA

Yo, **DANNY SIXTO ARMIJOS LOYOLA**, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente, acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional-Biblioteca Virtual.

Firma: _____.

Cédula: 1104141377

Fecha: 14 de enero del 2022

CARTA DE AUTORIZACIÓN

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LA AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, DANNY SIXTO ARMIJOS LOYOLA, declaro ser autor de la tesis titulada: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE VARIABLES FÍSICAS DE UN GENERADOR ELÉCTRICO Y SISTEMA DE BACKUP DE ALIMENTACIÓN A TRAVÉS DE UNA APLICACIÓN WEB”, como requisito para optar al grado de: INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la reproducción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los catorce días del mes de enero del dos mil veintidós.

Firma: _____.

Autor: **Danny Sixto Armijos Loyola**

Cédula: 1104141377

Dirección: Catamayo (Parroquia María Auxiliadora)

Correo electrónico: danny.s.armijos@unl.edu.ec

Celular: 0983902544

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Renato Benjamín Torres Carrión, Mg. Sc.

Tribunal de Grado: Ing. Diego Vinicio Orellana Villavicencio, MSc

Ing. Luis Eduardo Rodríguez Montoya

Ing. Santiago Abraham Medina León

DEDICATORIA

Este trabajo va dedicado primeramente a Dios, Padre Celestial que me ha brindado amor y fortaleza en cada paso que he dado, por no abandonarme en ningún instante y mostrarme aún en mis momentos más desolados su inmenso amor por mí.

A mis amados padres: Yaneth Loyola y Sixto Armijos, por ser un pilar fundamental para mí y lo que soy, a mi Padre por su apoyo económico y por sus consejos que me guía en mi diario vivir, y mi mamá por ser mi ejemplo de vida, por su dedicación, determinación y su lucha, los tengo en mi corazón.

A mis hermanos, especialmente a Richar Armijos, por su apoyo incondicional de y sus palabras de motivación para seguir adelante pese a cualquier dificultad.

A mis compañeros, por ser parte del camino de vida estudiantil afrontando unidos cualquier adversidad por la que me siento afortunado por contar con su amistad.

Danny Sixto Armijos Loyola

AGRADECIMIENTO

Quiero expresar mi sincero agradecimiento a la Universidad Nacional de Loja, a la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables y a la Carrera de Ingeniería en Electrónica y Telecomunicaciones, quienes participaron activamente de mi formación académica y para la finalización de la misma.

De manera especial, agradezco al Ing. Renato Benjamín Torres Carrión, director del presente trabajo, por su guía y asesoramiento profesional el cual fue pilar fundamental para cumplir con la meta propuesta, mi profundo agradecimiento por el tiempo dedicado.

De igual manera hago extensivo este agradecimiento especialmente a mi padre y madre que me apoyaron en todo momento y a cada una de las personas que de alguna u otra manera me orientaron en forma oportuna y desinteresada.

Danny Sixto Armijos Loyola

TABLA DE CONTENIDO

CERTIFICACIÓN.....	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN	IV
DEDICATORIA.....	V
AGRADECIMIENTO	VI
TABLA DE CONTENIDO	VII
INDICE DE TABLAS.....	X
INDICES DE FIGURAS	XI
1. TITULO	1
2. RESUMEN.....	2
3. INTRODUCCIÓN	4
4. REVISIÓN LITERARIA	6
4.1 ESTADO DEL ARTE	6
4.2 GENERADOR ELÉCTRICO	7
4.2.1 Tipos de generadores.....	7
4.2.2 Clasificación de los generadores según el tipo de arranque.....	8
4.2.3 Principio de funcionamiento de un generador A.C.....	9
4.3 SISTEMA DE ALIMENTACIÓN ININTERRUMPIDA (SAI).....	9
4.3.1 Topologías básicas de SAI.....	10
4.3.2 Componentes comunes a todos los tipos de SAI.....	11
4.3.3 Clasificación de SAI.....	11
4.3.4 Banco de Baterías.....	14
4.4 SENSORES Y ACTUADORES	17
4.4.1 Sensores de Nivel.....	18
4.4.2 Relés electromagnéticos.....	20
4.5 SISTEMAS DE MONITOREO	22
4.5.1 Rendimiento del sistema de monitoreo	23
4.5.2 Arquitectura de un sistema de monitoreo.....	24

4.6	APLICACIÓN WEB	25
4.6.1	Lenguaje de Marcas de Hipertexto (HTML)	27
4.6.2	Lenguajes de programación	29
4.7	PLATAFORMA DE IOT.....	32
4.7.1	Amazon AWS IoT.....	34
4.7.2	Google Cloud IoT.....	35
4.7.3	Microsoft Azure IoT.....	36
4.7.4	ThingSpeak.....	36
4.7.5	Thinger.io	37
4.7.6	Ubidots	38
4.7.7	Analices de plataforma IoT a utilizar	38
4.8	HARDWARE DE CONTROL PARA MONITOREO DE VARIABLES	39
4.8.1	Circuito Integrado Programable PIC.....	39
4.8.2	Controlador lógico programable PLC	40
4.8.3	Hardware Arduino.....	41
4.8.4	Arduino en la Industria 4.0.....	43
4.8.5	Aspectos relevantes costo / beneficio.....	48
5.	MATERIALES Y MÉTODOS	51
5.1	MÉTODOS	51
5.2	MATERIALES.....	52
5.2.1	Arduino Mega 2560	52
5.2.2	Arduino Ethernet Shield W5100 + Lector De SD	53
5.2.3	Generador de energía HISA LT3990E.....	54
5.2.4	Sensor nivel de combustible Marine Boat RV	55
5.2.5	Relé Arduino	56
5.2.6	Módulo de fuente de alimentación CA, 110 V, 220 V a CC, 5 V.....	57
5.2.7	Sensor de nivel de energía.....	57
5.2.8	Reloj de tiempo real RTC 3231.....	58
5.3	ARQUITECTURA DEL SISTEMA	59
5.3.1	Software Seleccionado	61
5.3.2	Diagrama de conexiones del sistema	63
6.	RESULTADOS.....	65

6.1	PLATAFORMA DE MONITOREO Y CONTROL DEL SISTEMA CON ACCESO VÍA LAN.	
	65	
6.1.1	Nivel de combustible.....	65
6.1.2	Nivel de batería	67
6.1.3	Funcionamiento de la red y del generador	68
6.1.4	Control de encendido y apagado	69
6.1.5	Sistema de advertencia.....	72
6.1.6	Visualización y registro de datos en aplicación web.....	73
6.2	PLATAFORMA DE MONITOREO Y CONTROL DEL SISTEMA A TRAVÉS DE UN SERVIDOR IoT EL CUAL PERMITE EL ACCESO VÍA WAN	75
6.2.1	Programación en el ATmega2560.....	75
6.2.2	Configuración en la plataforma Ubidots	78
7.	DISCUSIÓN.....	85
8.	CONCLUSIONES	87
9.	RECOMENDACIONES	89
10.	REFERENCIAS	91
11.	ANEXOS.....	99
	ANEXO 1. DISEÑO DEL CIRCUITO DE LA PCB EN EL SOFTWARE PROTEUS	99
	ANEXO 2. INSTRUCCIONES PROGRAMADAS EN ARDUINO IDE PARA EL SERVIDOR WEB LOCAL.....	102
	ANEXO 3: CÓDIGO DEL DISEÑO WEB EN HTML	114
	ANEXO 4. PROCESO DE ELABORACION DEL HARDWARE.....	124

INDICE DE TABLAS

Tabla 1 Clasificación y tipos de generadores.	7
Tabla 2 Tipo de arranque de generadores de energía [7].....	8
Tabla 3 Clasificación de las topologías básicas de SAI [11]	10
Tabla 4 Comparativas de las distintas SAI [11].	14
Tabla 5 Categorías principales de Instrumentación y Control [21].....	23
Tabla 6 Análisis de funcionalidad de ciertas plataformas IoT. Fuente: Autor.....	39
Tabla 7 Características del módulo Controlino [41].....	43
Tabla 8 Características del módulo Siemens [42].	44
Tabla 9 Características del módulo Industruino [43].	45
Tabla 10 Características del módulo M-DUINO PLC Arduino [44].	46
Tabla 11 Características del módulo Arduino Uno R3 [45].	46
Tabla 12 Análisis de las capacidades que poseen lo diferentes hardware mencionados.	47
Tabla 13 Análisis de los aspectos relevantes de los microcontroladores.	48
Tabla 14 Comparación económica de equipos basados en Arduino.	49
Tabla 15 Comparación de facilidad de implementación del hardware.....	49
Tabla 16 Métodos aplicados para el desarrollo del proyecto.	51
Tabla 17 Características técnicas del Arduino Mega 2560 [38].....	53
Tabla 18 Características técnicas de la tarjeta Ethernet [48].....	54
Tabla 19. Características Técnicas [49].....	55
Tabla 20. Especificaciones del sensor de combustible [50].	56
Tabla 21. Especificaciones del Relé Arduino [51].	57
Tabla 22 Especificaciones del módulo Transformador [52].....	57
Tabla 23 Especificaciones del módulo Reloj de tiempo real RTC 3231 [53].	58

INDICES DE FIGURAS

Figura 1. Conversión de energía utilizando un generador de electricidad [5].....	7
Figura 2. Forma de trabajo del generador de corriente alterna [8].	9
Figura 3. Sistema de alimentación ininterrumpida off line [12].	12
Figura 4. Sistema de alimentación ininterrumpida Line-Interactive (VI) [12].	13
Figura 5. Sistema de alimentación ininterrumpido On-line conversión (VFI) [12].	13
Figura 6. Batería de acumuladores de la firma NIFE [13].	15
Figura 7. Curvas de tensión e intensidad del Método de carga a intensidad constante [13].	15
Figura 8. Curvas de tensión e intensidad del Método de carga a intensidad decreciente [13].	16
Figura 9. Sistema de flotación [13].	16
Figura 10. Clasificación de sensores por los principios de traducción [15].....	17
Figura 11. Clasificación de sensores por el tipo de variable medida [15].....	18
Figura 12. Clasificación de los actuadores dependiendo del tipo de energía utilizada [15].	19
Figura 13. Diversos sensores de nivel [9].	20
Figura 14. Relé electromagnético [17].	21
Figura 15. Símbolos para relés con diferentes tipos de contactos [18].	22
Figura 16. Esquema de un proceso de monitoreo de sistema [23].	25
Figura 17. Áreas de conocimiento para el desarrollo web [25].....	26
Figura 18. Tecnologías para el desarrollo web [25].	27
Figura 19. Plataformas orientadas al IoT [30].	34
Figura 20. Funciones de AWS IoT [31].	35
Figura 21. Logo de Ubidots [36].	38
Figura 22. PIC16F84A [39].	40
Figura 23. Modelo de una PLC. Fuente: RIEVTECH.....	41
Figura 24. Periféricos de Arduino [43].	42
Figura 25. Tipos de placas orientadas al internet de las cosas [43].	42
Figura 26. Controlador Controlino [45].	44
Figura 27. Módulo Siemens basado en Arduino [46].	44
Figura 28. Industruino [47].	45
Figura 29 M-DUINO PLC 19R I/Os [48].	46

Figura 30. Arduino uno R3 basado en ATmega328P [49].....	47
Figura 31. Diagrama de trabajo para el desarrollo del sistema de monitoreo.	52
Figura 32. Partes de la placa electrónica Arduino Mega 2560 [51].	53
Figura 33. Arduino Ethernet Shield [52].	54
Figura 34. Imagen del Generador eléctrico HISA LT3990E.	55
Figura 35. Sensor de nivel de combustible [54].	56
Figura 36. Relé Arduino de 1 canal [55].	56
Figura 37. Transformador reductor de AC a DC [56].	57
Figura 38. Divisor de voltaje para sensar nivel de batería con Arduino.	58
Figura 39. Reloj de Tiempo Real RTC 3231 [57].	59
Figura 40. Diagrama isométrico de la Infraestructura actual de la distribución de energía eléctrica.	59
Figura 41. Diagrama isométrico del sistema de monitoreo y respaldo de energía eléctrica.	60
Figura 42. Diagrama isométrico de la distribución eléctrica y del sistema de monitoreo del lugar a implementar.	61
Figura 43. Interfaz del software IDE [58].	62
Figura 44. Interfaz de Geany [59].	62
Figura 45. Software Proteus [60].	63
Figura 46. Diagrama de Conexiones del controlador ATMEGA 2560.	64
Figura 47. Diagrama en Proteus del circuito sensor de combustible.	66
Figura 48. Instrucciones de lectura del pin análogo A2 para analizar el nivel de combustible.	66
Figura 49. GUI del monitoreo del nivel de combustible.	67
Figura 50. Divisor de voltaje para poder medir el voltaje de la batería con el Arduino.	67
Figura 51. Instrucciones en el software IDE para la lectura del pin análogo A3 que es interpretado como nivel de energía del SAI.	68
Figura 52. GUI del monitoreo del nivel de energía en el SAI.	68
Figura 53. Transformador reductor de AC a DC [56].	69
Figura 54. Circuito de Proteus para analizar el suministro de energía.	69
Figura 55. Relé Arduino de 1 canal [55].	70
Figura 56. Diagrama en Proteus del circuito de control de arranque del generador.	70

Figura 57. Instrucciones en el software IDE para la lectura del pin digital D3, la cual analiza si existe voltaje en la red eléctrica.....	71
Figura 58. Instrucciones en el software IDE para la lectura del pin digital D5, la cual analiza si el generador está suministrando voltaje.....	71
Figura 59. Instrucciones para crear un botón de control de encendido / apagado del generador en la GUI de la web.	71
Figura 60. GUI que demuestra de donde se está suministrando energía a la red.	72
Figura 61. GUI del control de arranque.....	72
Figura 62. Diagrama en Proteus del circuito de control de arranque del generador.	72
Figura 63. Instrucciones para crear un botón de control de encendió / apagado del Sistema de alarma.....	73
Figura 64. GUI del control de la alarma.....	73
Figura 65. Interfaz gráfica del servidor WEB del monitoreo y control de las variables físicas de un generador de combustión.....	74
Figura 66. Registro generado por variaciones del sistema de alimentación de energía eléctrica.....	74
Figura 67. Ingreso a la plataforma de Ubidots.	78
Figura 68. Configuración de establecimiento de la conexión del controlador con la plataforma Ubidots.	78
Figura 69. Configuración de establecimiento de la conexión del controlador con la plataforma Ubidots.	79
Figura 70. Configuración de establecimiento de la conexión del controlador con la plataforma Ubidots.	79
Figura 71. Verificación del establecimiento de la conexión del controlador con la plataforma Ubidots.	79
Figura 72. Verificación del establecimiento de la conexión del controlador con la plataforma Ubidots.	80
Figura 73. Configuración de un indicador de variable Indicador combustible	80
Figura 74 . Configuración de un indicador de variable Indicador combustible	80
Figura 75. Configuración de un indicador de variable Indicador combustible	81
Figura 76. Configuración de un indicador de variable Indicador combustible	81
Figura 77 . Configuración de un indicador de variable N_ combustible	82
Figura 78. GUI del sistema de monitoreo del suministro de energía de energía eléctrica.	82

Figura 79. Proceso de configuración de eventos en Ubidots.....	83
Figura 80. Proceso de configuración de <<Si condición(es)>> en Ubidots.....	83
Figura 81. Proceso de configuración de <<Entonces acciones>> en Ubidots	83
Figura 82. Proceso de configuración de <<EN LA ACTIVACIÓN>> en Ubidots	84
Figura 83. Proceso de configuración de <<VUELTA A LA NORMALIDAD>> en Ubidots	84
Figura 84. Diseño del circuito de todos los módulos a implementar del proyecto, en la función de Schematic Capture. Software: Proteus.	99
Figura 85. Conexión de los módulos a implementar al controlador Arduino ATmega 2560. Software: Proteus.....	100
Figura 86. Diseño de la PCB mediante la función de ARES. Software: Proteus.....	100
Figura 87. Visualización en 3D de la placa PCB del proyecto. Vista del tablero de la parte superior. Software: Proteus.	101
Figura 88. Proceso químico para quitar residuos de cobre de la placa PCB.....	124
Figura 89. Proceso de soldado de soldado de los componentes externos a la PCB. ...	124
Figura 90. Placa de circuito impreso que permite adquirir información de variables físicas de un sistema de respaldo de energía eléctrica.	125
Figura 91. Ensamblado del equipo de monitoreo y control de un generador eléctrico y del SAI.....	125

1. TITULO

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y CONTROL DE VARIABLES FÍSICAS DE UN GENERADOR ELÉCTRICO Y SISTEMA DE BACKUP DE ALIMENTACIÓN A TRAVÉS DE UNA APLICACIÓN WEB”

2. RESUMEN

El presente documento muestra el desarrollo de un sistema de monitoreo y control que permite adquirir información de variables físicas de un generador de combustión y analizarlas a través de una interfaz web.

El proyecto de investigación se estructura en dos partes. La parte de hardware que lo integran controlador programable y sensores, y la parte software donde se configuran las instrucciones al controlador, interfaz gráfica web y generación de datos históricos del sistema.

En la parte de hardware se ha seleccionado los sensores adecuados que permiten adquirir información de los parámetros de interés como el nivel de combustible, el nivel de energía en el Sistema de Alimentación Ininterrumpida (SAI) y el tiempo de funcionamiento del generador. Por otro lado, en la parte de software, se ha seleccionado el programa ARDUINO IDE que permite programar y cargar las instrucciones necesarias al controlador ATMEGA 2560.

Como resultado, se obtiene la implementación de una interfaz de usuario gráfica (GUI) creada con ayuda de una plataforma de internet de las cosas (IoT) la cual permite monitorizar las variables físicas de un generador eléctrico de combustión y un sistema de alimentación ininterrumpida (SAI) de manera remota a través de internet y a la vez crear un registro del funcionamiento del sistema.

Palabras claves: Sistema de backup, Generadores de energía; Sistema de alimentación ininterrumpido; Sistemas de monitoreo; Aplicación web; Controladores; Sensores, Plataformas IoT; Ubidots.

ABSTRACT

This paper shows the development of a monitoring and control system that allows acquiring information of physical variables of a combustion generator and analyzing them through a web interface.

The research project is structured in two parts. The hardware part is composed of a programmable controller and sensors, and the software part where the instructions to the controller, web graphic interface and generation of historical data of the system are configured.

In the hardware part, the appropriate sensors have been selected to acquire information on the parameters of interest such as the fuel level, the energy level in the Uninterruptible Power Supply (UPS) and the running time of the generator. On the other hand, in the software part, the ARDUINO IDE program has been selected, which allows programming and loading the necessary instructions to the ATMEGA 2560 controller.

As a result, the implementation of a graphical user interface (GUI) created with the help of an Internet of Things (IoT) platform is obtained, which allows monitoring the physical variables of a combustion electric generator and an uninterruptible power supply (UPS) remotely through the Internet and at the same time creating a record of the system operation.

Keywords: Backup system; Power generators; Uninterruptible power supply; Monitoring systems; Web application; Controllers; Sensors; IoT platforms; Ubidots.

3. INTRODUCCIÓN

Los centros industriales que requieren un uso constante del suministro de energía eléctrica para sus instalaciones no pueden confiarse del correcto funcionamiento de la red eléctrica pública con un servicio 24/7 constante, por lo que surge la necesidad de poseer un sistema de respaldo de energía eléctrica. Existe centros de negocios, empresas públicas y privadas que utilizan generadores de energía eléctrica de combustión que no poseen el sistema adecuado de monitoreo del nivel de combustible en el tanque, el momento de encendido y apagado del generador, ni tampoco el sensado del nivel de batería del SAI.

Actualmente, en el mercado se encuentra una gran variedad de sensores y sistemas de control que permiten una adecuada automatización y monitoreo. Por lo que surge la necesidad de analizar cuál es el sensor óptimo para la función a ejecutar. En el caso de estudio, al tratarse de un generador de combustión, se incrementa el consumo de combustible fósil del sistema, por lo que es necesario buscar mecanismos de optimización de este recurso. El problema radica en que no se cuenta con un registro de encendido y apagado del generador eléctrico que indique el estado de las baterías, el funcionamiento del SAI y generador.

En el caso de los generadores eléctricos que funcionan con combustible fósil es necesario llevar un monitoreo constante del nivel del tanque de combustible del generador, así como el tiempo de encendido y apagado de este equipo. Además es importante programar un apagado automático del generador cuando exista el suministro de energía de la red pública. También es necesario sensar el estado y el nivel de baterías del SAI con el fin de que se encuentren cargados en el caso de fallo de alimentación de energía de la red eléctrica pública.

Por tal razón el presente trabajo supone la implementación hardware y software basado en Arduino y el lenguaje de marcado HTML que permita monitorizar las variables de nivel de combustible e indicar el momento de encendido y apagado de un generador. Así también, que permita sensar del nivel de la batería del SAI. Además, se plantea la automatización del apagado del generador al momento del retorno de abastecimiento de la energía de la red eléctrica pública.

Con la finalidad de tener acceso remoto en tiempo real del estado del sistema de respaldo y para disponer de la información inmediata acerca de cuándo ocurre un cambio de estado

en las variables a sensar, se busca tener un monitoreo del sistema de respaldo de la energía eléctrica vía web.

En base a lo planteado, se definen los siguientes objetivos:

Como objetivo general, diseñar e implementar un hardware y un software que permita sensar variables físicas de un generador eléctrico de combustión para un sistema de backup y monitorear los datos recolectados a través de una plataforma web.

Y como objetivos específicos:

- Implementar sensores que permitan medir variables como: nivel de combustible del generador de energía, tiempo de funcionamiento del generador y el estado de carga de la batería del SAI.
- Establecer un control de apagado automático del generador que evite la necesidad de la interacción humana.
- Implementar una plataforma de hardware para el control automático de los sensores.
- Montar una plataforma de gestión y supervisión remota mediante una aplicación web.

4. REVISIÓN LITERARIA

4.1 Estado del arte

En el siguiente apartado se ha realizado una recopilación de trabajos elaborados y afines con el monitoreo de variables físicas obtenidas por sensores y del uso de aplicaciones WEB para su tratamiento y gestión.

El trabajo realizado en [1] cuya finalidad es el monitoreo de un generador eléctrico de la terminal Portuaria de Guayaquil, consiguieron por medio de un controlador lógico programable (programmable logic controller, *PLC*¹), sensores y el software LabVIEW monitorizar el nivel de combustible, nivel de refrigerante y nivel de voltaje de las baterías de arranque para el generador. Tuvo como finalidad la creación de un sistema que permitió supervisar el *backup*² eléctrico.

En el proyecto [2] se monitorea el voltaje de la red eléctrica, voltaje del generador, voltaje del SAI, voltaje del banco de baterías del SAI y el nivel de combustible a través del microcontrolador PIC 16F871 y los recursos informáticos del software Visual Basic. El acceso remoto se logra a través del software TeamViewer³.

Así también en el trabajo de [3] se implementa un sistema de medición, monitoreo y control de carga. Con ayuda del Software Python, el hardware de Arduino y Raspberry Pi se logra una supervisión remota a través de la WEB.

En el artículo [4] se implementa un prototipo de control de conmutación eléctrica, el cual utiliza la plataforma electrónica de Arduino para el control y el software LabVIEW que muestra de forma gráfica los datos y señales del sistema a inspeccionar.

¹ Controlador lógico programable, más conocido por sus siglas en inglés PLC (programmable logic controller) o por autómatas programables, es una computadora utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas

² Un Back Up eléctrico es una máquina que mueve un generador de electricidad a través de un motor de combustión interna. Son comúnmente utilizados cuando hay déficit en la generación de energía eléctrica de algún lugar, o cuando son frecuentes los cortes en el suministro eléctrico.

³ TeamViewer. Es un software informático privado de fácil acceso, que permite conectarse remotamente a otro equipo. Entre sus funciones están: compartir y controlar escritorios, reuniones en línea, videoconferencias y transferencia de archivos entre ordenadores.

4.2 Generador eléctrico

Es una maquina capaz de transforma recursos renovables o no renovables que nos ofrece la naturaleza en energía eléctrica. Según Donatte *“El generador de electricidad es la fuente de energía de los sistemas eléctricos y electrónicos, estos proporcionan potencia eléctrica (tensión y corriente). En todos los generadores se realiza una transformación de energía a partir de una que puede ser mecánica, térmica, química, o de cualquier otro tipo, proporcionando energía eléctrica”* [5].

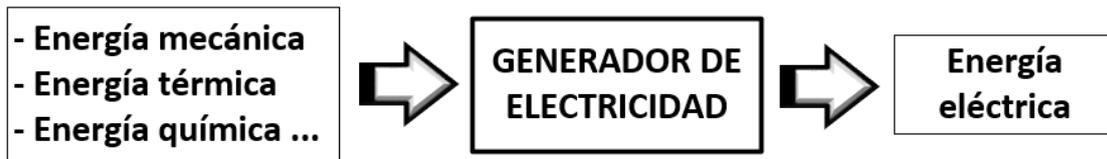


Figura 1. Conversión de energía utilizando un generador de electricidad [5].

4.2.1 Tipos de generadores

Existe una gran variedad de generadores eléctricos. En la Tabla 1 se describe los siguientes en base a [5], [6] y [7]:

Tabla 1 Clasificación y tipos de generadores.

Fuente: Autor

Generadores eléctricos según	Clasificación	Descripción
Aplicación y el tipo de funcionamiento	Generadores mecánicos	- Recurren a la energía mecánica para colocar el generador en funcionamiento.
	Generadores solares	- Captan la energía solar, para transformarla en energía eléctrica.
	Generadores térmicos	- Transforman la energía proveniente del calor en energía eléctrica.
Función de la clase de corriente que generan	Alternador	- Maquina eléctrica que transforma energía mecánica en energía eléctrica de corriente alterna (AC).
	Dínamo	- Máquina que transforma la energía mecánica en energía eléctrica de corriente continua (DC) .
Tipo de combustible	Gasoil (equipos diésel):	- Ideales para un uso continuo con una entrega de potencia superior a los 5 kW. - Son adecuados para trabajos en talleres y sitios remotos, para la energización de distintas máquinas eléctricas, incluso equipos de soldadura que requieren grandes corrientes.
	Nafta	- Electrógenos más asequibles.

		<ul style="list-style-type: none"> - Se emplean para un uso frecuente en potencias menores o iguales a 2 kW y para potencias de hasta 5 kW si se utilizan de manera ocasional. - Se utilizan en pequeños comercios, pequeños talleres entre otros.
	Gas	<ul style="list-style-type: none"> - Generadores de pequeña o mediana potencia de hasta 5 kW, ideales para uso periódico. - Su uso está indicado para pequeños comercios.
Tipo de tensión del generador	Monofásicos	<ul style="list-style-type: none"> - Entregan una tensión máxima de 220 V con frecuencia de 50 Hz. - Son los generadores a gas o nafta que suministran hasta una potencia máxima de 5kW
	Trifásicos	<ul style="list-style-type: none"> - Entregan hasta una tensión de 380 v a 50 Hz con una potencia superior a los 5 kW. - Son aquellos generadores de ciclo a diésel.

En base a requerimientos concretos de empresas locales y nacionales de monitorear los niveles de combustible en generadores eléctricos de combustión a Gasoil (equipo diésel), se propone la creación de un mecanismo de monitoreo del nivel de este fluido.

4.2.2 Clasificación de los generadores según el tipo de arranque

Es importante clasificar los generadores según el tipo de arranque ya que de esto dependerá en gran parte si es posible automatizar el sistema de respaldo de la energía eléctrica. Por lo que en la Tabla 2 se describen en dos grupos: los de arranque manual y los de arranque eléctrico.

Tabla 2 Tipo de arranque de generadores de energía [7].

Tipos	Características.
Arranque manual sin conmutador de carga	<ul style="list-style-type: none"> - Es el sistema de arranque más económico. - Es el menos funcional. - Un individuo debe estar encargado de conectar y arrancar el equipo.
Arranque manual con conmutador de carga	<ul style="list-style-type: none"> - El conmutador es quien opera toda la corriente de carga. - Es independiente de la capacidad del generador. - Se deben desconectar algunas cargas para evitar sobrecargar al generador. - Es recomendado para generadores de hasta 5 o 6 kW de potencia.
Arranque eléctrico con conmutador manual de carga	<ul style="list-style-type: none"> - Se debe presionar el botón de encendido para el funcionamiento del generador. - Se debe poner el conmutador en modo “generador”. - Se utiliza para generadores con potencia mayor a 6 kW.
Arranque eléctrico con	<ul style="list-style-type: none"> - Se emplea para generadores de reducida potencia. - Provee protección. - Impide la posibilidad de sobrecarga del generador.

control automático	- Aísla los aparatos con mayor consumo automáticamente, en caso de que se manifieste una alteración en la potencia.
---------------------------	---

Este proyecto consiste en automatizar el sistema de arranque del generador eléctrico en caso de falta de suministro de la red pública de energía por lo que el generador debe poseer un sistema de arranque eléctrico.

4.2.3 Principio de funcionamiento de un generador A.C.

El funcionamiento del generador de corriente alterna está basado en la primicia general de inducción de voltaje en un conductor en movimiento, cuando atraviesa un campo magnético.

La forma en cómo convierten los generadores la energía mecánica en energía eléctrica, se puede analizar la **Figura 2**, que muestra un generador elemental, en donde el campo magnético principal viene de un par de imanes permanentes. La bobina de la armadura está devanada sobre el rotor, cada extremo de esta bobina está fijo a su propia banda metálica. Estas bandas se llaman anillos rozantes y es en donde aparece el voltaje generado.

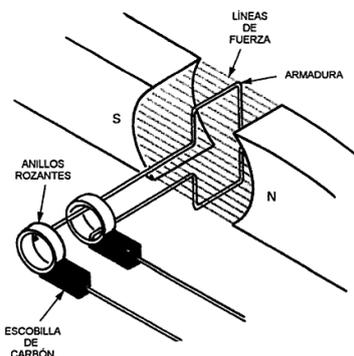


Figura 2. Forma de trabajo del generador de corriente alterna [8].

Para recaudar el voltaje producido, se debe tener un recorrido eléctrico de los anillos rozantes a las terminales del generador, esto se consigue con pequeñas piezas metálicas o de carbón llamadas escobillas, que se encuentran adheridas a los anillos rozantes por medio de resortes. En la medida que la bobina rota, los conductores cortan el campo magnético, lo que origina un voltaje inducido en la bobina [8].

4.3 Sistema de Alimentación Ininterrumpida (SAI)

Es un equipo encargado de proporcionar energía eléctrica a la carga en caso de que el suministro de la energía pública sea interrumpido por diversas circunstancias. Esta

función se realiza de manera automática, por lo tanto, al detectar una condición anormal en el abastecimiento (falla del suministro de energía eléctrica de la red pública) instantáneamente se reemplaza por medio de la energía almacenada en el banco de baterías [9]. Al momento que se restablezca el servicio se deshabilita el SAI y pasa a la carga de su almacenamiento.

El SAI abastece de energía por un lapso determinado, el cual dependerá de la capacidad de las baterías y de la potencia de las cargas que tenga conectada para dar tiempo suficiente a apagar adecuadamente los equipos o en el caso de que se posea un generador eléctrico ponerlo en operación. Algunos SAI también ofrecen capacidad de conectarse a los servidores encargándose de apagarlos de manera controlada.

Por lo general, los centros que albergan normalmente un SAI también poseen un generador de energía propio, por lo que en el momento de un corte del suministro eléctrico el SAI cumplirá la función de abastecer la energía durante el lapso comprendido entre el corte y la entrada en régimen del generador [10].

4.3.1 Topologías básicas de SAI

Existen cuatro tipos de topologías cada una con sus particularidades, ya que varían en costos y disponibilidad. En la **Tabla 3** se muestra una clasificación de estas [11].

Tabla 3 Clasificación de las topologías básicas de SAI [11].

Tipos de topología	Características
Capacidad normal (N)	<ul style="list-style-type: none"> - Económica. - Suministra electricidad por un tiempo limitado. - No es capaz de responder satisfactoriamente a ningún tipo de fallo de la SAI. - Se utiliza en ambientes pequeños.
Redundancia paralela aislada (N)	<ul style="list-style-type: none"> - Semejante a la capacidad normal. - Posee características en que los módulos son independientes del fabricante. - Mayor flexibilidad al momento de adquisición de equipos, aunque dificulta y encarece la instalación, el monitoreo y el mantenimiento.
Redundancia paralela (N+1)	<ul style="list-style-type: none"> - Posee un módulo adicional al de capacidad normal que le permite tolerar el fallo de solo un módulo de la SAI. - Soporta la carga total sin complicaciones. - Es posible realizar mantenimientos sin impactar al servicio. - Son los más utilizados en Centros de datos, porque funcionan 8 x 5 (8 horas por día hábil de la semana). - Se utiliza en entornos medianos.

Redundancia doble (2N)	- Suministran el doble de capacidad total de la carga. Por tal razón, si un sistema completo de SAI falla existe otro capaz de soportar toda la carga.
-------------------------------	--

Según el análisis de los tipos de topologías de los SAI, el mejor es el de redundancia doble (2N), pero con el inconveniente que es el más costoso. Este proyecto consiste en monitorizar el nivel de energía que el SAI posee para garantizar que se encuentre cargado ante una posible falla de la energía eléctrica convencional. Por lo que, en el caso de este proyecto la topología del SAI puede ser cualquiera de las mencionadas en la Tabla 3.

4.3.2 Componentes comunes a todos los tipos de SAI

Batería: se encarga de almacenar energía y suministrarla en caso de interrupción de la corriente eléctrica. Su capacidad se mide en Amperes por hora (A/h), y según el tamaño varía la autonomía (habitualmente, el SAI se calcula para tener una durabilidad que oscile entre los 5 y los 15 minutos).

Inversor: convierte la corriente continua de las baterías en corriente alterna, la cual sustenta los terminales conectados a la salida del SAI. Asimismo, desempeña el proceso inverso, transformando corriente alterna de la red en corriente continua para almacenarla en las baterías.

Para el adecuado funcionamiento de las baterías es recomendable que la temperatura en el lugar de operación oscile alrededor de los 20°C. En caso que la temperatura sea mayor no se tendrán los resultados deseables de funcionamiento, durabilidad y eficiencia [11].

La duración útil de la batería es de 5 años (es el promedio estándar en condiciones óptimas). Si funciona a otra temperatura como puede ser de 30°C, su durabilidad se reduce en un 50% o aún peor si opera a 40°C, donde la vida útil se reduce al 20% [11]. Por tal razón se recomienda ubicar las baterías en un ambiente de temperatura controlada, con el propósito de conservar la durabilidad y no aumentar los costos.

4.3.3 Clasificación de SAI

Existen variedad de SAIs. Su utilización dependerá de su aplicación, por lo que se distinguen tres sistemas de operación:

Off line, Stand-by (espera o pasivos)

Monitorea la entrada de energía en forma permanente, cambiando la fuente de alimentación a batería por medio del interruptor, como se muestra en la Figura 3. Es capaz de detectar problemas en el suministro eléctrico de red en cuestión de milisegundos. El SAI interviene solo en ausencia de tensión de entrada, alimentando la carga con el inversor que a su vez está alimentado por las baterías. Se utiliza en instalaciones pequeñas (monofásica) y brinda una autonomía reducida [12].

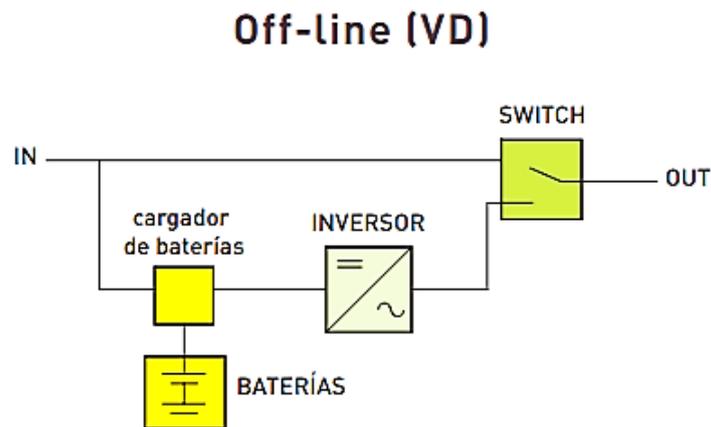


Figura 3. Sistema de alimentación ininterrumpida off line [12].

On-line, línea interactiva o activos

En presencia de una red de alimentación, la entrada y la salida están separadas por un circuito de filtración y estabilización, pero parte de las interferencias o variaciones de la forma de onda, posibles en la entrada pueden encontrarse en la salida. Como en off line, durante el momento de ausencia de red, la salida es conectada al inversor, alimentado a su vez por las baterías [12].

Como se observa en la Figura 4, el inversor se encuentra conectado a la red principal de energía a causa que constantemente se realiza una doble conversión en tiempo real y de manera constante. El conmutador (switch) encargado de detectar la interrupción de energía, se encuentra conectado a la salida del inversor y del regulador automático de voltaje (AVR). La corriente pasa por el rectificador permanentemente para cargar la batería y continuar con el inversor que transforma la energía corriente continua a corriente alterna para la salida del SAI. Cuando se interrumpe el flujo eléctrico de la red, el conmutador se alimenta del inversor, por esta razón la salida no sufre ninguna interrupción por el corte (ni siquiera milisegundos) [11].

Line-Interactive (VI)

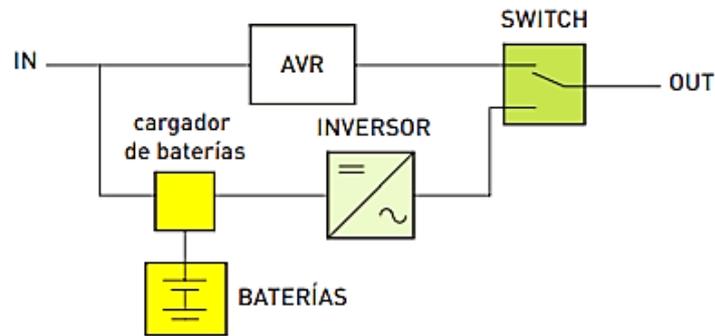


Figura 4. Sistema de alimentación ininterrumpida Line-Interactive (VI) [12].

On-line de doble conversión

La entrada es primero rectificada y después reconvertida en alterna con un inversor como se observa en la Figura 5. De este modo, la forma de onda de la tensión de salida es completamente independiente de la entrada. Todas las posibles interferencias de red son eliminadas y no hay tiempo de transitorio en el paso de red a batería porque la salida está siempre alimentada por el inversor. En caso de sobrecargas y eventuales problemas internos, este tipo de SAI dispone de bypass automático, que garantiza la alimentación de la carga conmutándola directamente en la entrada [12].

On line doble conversión (VFI)

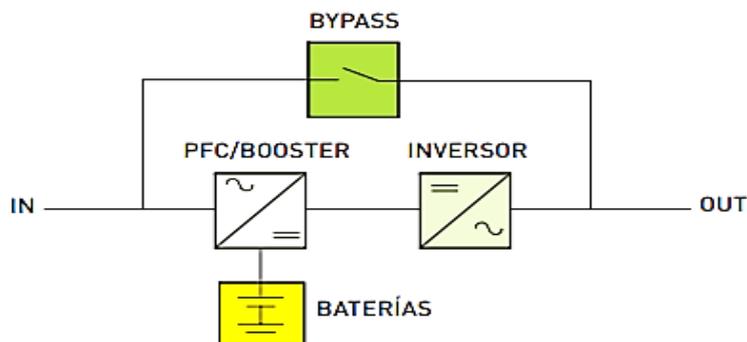


Figura 5. Sistema de alimentación ininterrumpido On-line conversión (VFI) [12].

En la Tabla 4 se puede analizar las características principales que definen a cada clasificación de un SAI.

Tabla 4 Comparativas de las distintas SAI [11].

Características	Stand-by	Línea Interactiva	Conversión Doble
Eficiencia (con carga al 100 %)	No aplica	98 %	86 % a 95 %
Rango de potencia aplicación práctica KVA	< 0.5	0.5 a 5	5 a 5000
Costo por KVA	Bajo	Medio	Medio
Uso de tensión	Baja	Baja o Alta	Alta
Beneficios	Bajo costo y tamaño reducido	Alta confiabilidad y eficiencia	Excelente confiabilidad
Destinos	Hogareño (N)	Centros de datos medianos (N+1)	Centros de datos grandes (2N)

4.3.4 Banco de Baterías

Dispositivo capaz de almacenar energía eléctrica para su uso posterior. Es un equipo electroquímico capaz de transforma energía química en eléctrica por medio de una celda galvánica. En base a [13] se describe lo siguiente:

Métodos de carga de la batería.

Una batería de acumuladores está formada por un cierto número de elementos que depende de la tensión deseada en la batería, tanto durante la carga como al final de la descarga, la intensidad exigida en la descarga y el tiempo de descarga. En la Figura 6 se muestra una aglomeración de elementos de la firma NIFE. En estas partes se pueden ver las placas con las que está compuesta.

Como ejemplo, para producir una batería con tensión de 110 VDC se ensambla 86 partes en serie. La tensión normalmente es aproximadamente 10 % mayor durante el periodo de carga y alcanzar 10 % menos de tensión al final de la descarga. Por lo cual, para la batería de 110 VDC serán necesarios durante la carga:

$$110 \times 1.10 = 121 \text{ voltios}$$

siendo la tensión aplicada a cada uno de los elementos.

$$121/86 = 1.41 \text{ voltios}$$

Y al final del ciclo de descarga se tendrá:

$$110 \times 0.90 = 99 \text{ voltios}$$

Quedando cada uno de los elementos con:

99/86 =1.15 voltios

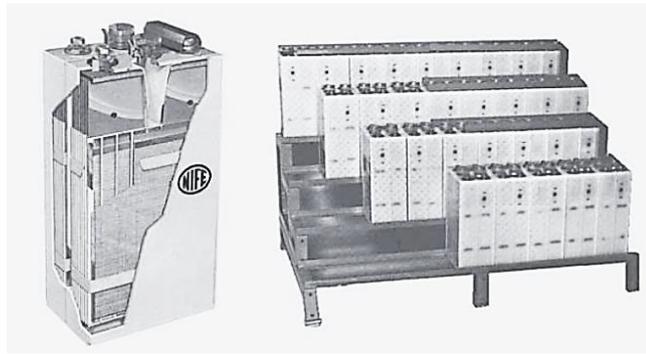


Figura 6. Batería de acumuladores de la firma NIFE [13].

En el periodo de descarga, dependiendo de la intensidad solicitada, la tensión va descargando gradualmente manteniéndose relativamente constante entre 112 y 103 VDC, o lo que es igual, entre 1.30 y 1.20 VDC por elemento al inicio y al final respectivamente. Después cae rápidamente hasta los 99 VDC considerados en este ejemplo.

Existen variantes de los métodos de carga de baterías, a continuación, se describen los tres siguientes:

- **Método de carga a intensidad constante.**

Esta se mantiene fija durante todo el tiempo. Cuando la batería se encuentra cargada aproximadamente a 80 % de su capacidad se produce una elevación en la tensión de los elementos desde 1.40 a 1.70 V aproximadamente. A partir de este momento se suele temporizar el sistema de carga para que al cabo de un tiempo se interrumpa la misma. En realidad, se interrumpe la carga rápida y se pasa a carga en flotación. La Figura 7 muestra las curvas de tensión e intensidad de este tipo de carga [13].

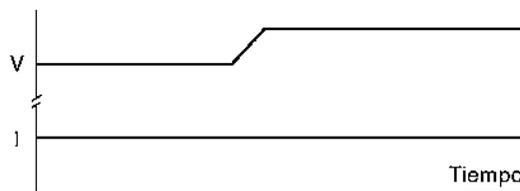


Figura 7. Curvas de tensión e intensidad del Método de carga a intensidad constante [13].

- **Método de carga a Intensidad decreciente.**

Tiene la particularidad de empezar con intensidad alta cuando la batería se encuentra descargada y va disminuyendo a medida que se realiza la carga. Es típico este sistema en

los cargadores para baterías de automóviles, donde la intensidad de carga depende de la tensión de salida del cargador y del estado de agotamiento de la batería. Este sistema no tiene regulación de carga [13].

Cuando la batería se carga a tensión constante se produce un aumento en la intensidad al principio de la carga, debiendo limitarse a un valor máximo para proteger el cargador. Durante la primera fase se mantendrá limitada la carga a intensidad máxima reduciendo la tensión, pero a medida que se va cargando disminuye la intensidad, momento en que se mantiene constante la tensión. Este sistema es el que se suele utilizar cuando se encuentran conectados en paralelo la batería y los consumidores del sistema. En operación normal el cargador suministra una tensión equivalente a la de flotación, por ejemplo, 1.40 V por elemento. La Figura 8 muestra las curvas de tensión e intensidad de este tipo de carga [13].

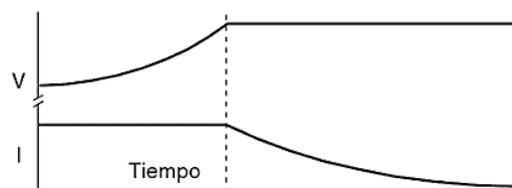


Figura 8. Curvas de tensión e intensidad del Método de carga a intensidad decreciente [13].

- **Método de carga de flotación.**

El sistema de flotación es aquel en el que la batería y la carga se encuentran conectadas en paralelo, como muestra la Figura 9. La tensión de suministro se suele conocer como tensión de flotación, siendo de valor constante y aproximadamente igual a 1,40 VDC por elemento. Con este sistema se mantiene la batería cargada constantemente [13].

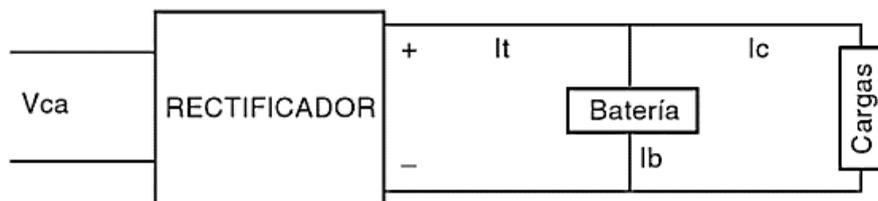


Figura 9. Sistema de flotación [13].

4.4 Sensores y Actuadores

Son los elementos que conectan el entorno físico a un sistema. La función de los sensores es adquirir señales eléctricas en respuesta a magnitudes de entrada de fenómenos físicos o químicos, mientras que la función de los actuadores es efectuar una acción mecánica en respuesta a una señal de entrada, que suele ser eléctrica, pero también puede ser neumática, hidráulica o mecánica [14].

Sensores

Los sensores son dispositivos electrónicos que permiten medir un parámetro físico con el objetivo de realizar un análisis a su comportamiento y efectuar una u otra acción. *“La utilización de sensores es indispensable en la automatización de industrias de proceso y manufacturados, incluida la robótica, en ingeniería experimental, en sectores no productivos como son el ahorro energético y el control ambiental (aire, ruido, calidad del agua), en automóviles y electrodomésticos, en la agricultura y medicina, etc. Incluso los equipos de gestión de datos, alejados de las aplicaciones industriales incorporan internamente para su correcto funcionamiento varios sensores”* [9].

El sensor es un equipo de entrada, por lo tanto, es un intermediario entre la variable física y el sistema de medida. El sensor además debe tener un tipo de dominio requerido. Los sensores pueden entregar tanto señales eléctricas analógicas como digitales dependiendo del tipo de sensor, debido a que este tipo de dominio físico es el más utilizado en los sistemas de medida actuales.

Los sensores se clasifican de algunas formas, pero las más habituales son por el principio de traducción utilizado o por el tipo de variable a medir como se muestra en la Figura 10 y Figura 11 respectivamente [15].

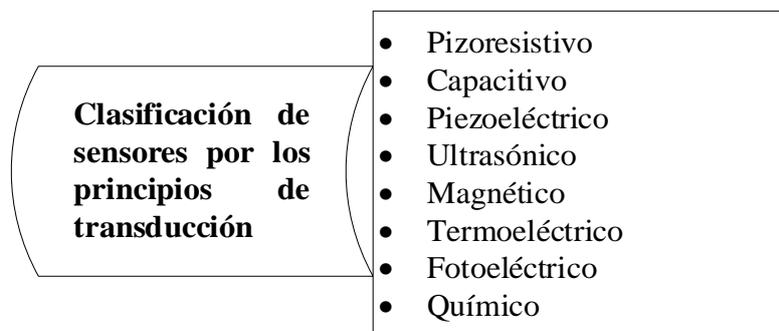


Figura 10. Clasificación de sensores por los principios de traducción [15].

Fuente: Autor

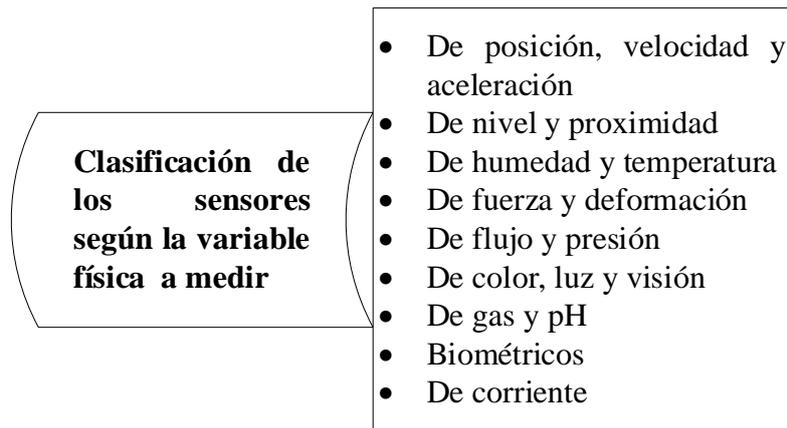


Figura 11. Clasificación de sensores por el tipo de variable medida [15].

Fuente: Autor

Actuadores

Nombrados también dispositivos de salida, son piezas que se encargan de controlar de manera aislada los receptores y cargas eléctricas, modificando su funcionamiento y comportamiento en la instalación [16].

Un actuador es un dispositivo con la capacidad de crear una fuerza que ejerce una variación en la posición, velocidad o estado de algún tipo, sobre un mecanicismo, a partir de la transformación de energía.

Por lo regular, los actuadores se clasifican en dos grandes grupos [15]:

1. Por el tipo de energía utilizada: actuador neumático, hidráulico y eléctrico como se observa en la Figura 12.
2. Por el tipo de movimiento que generan: actuador lineal y rotatorio.

4.4.1 Sensores de Nivel

Existen variedad de mecanismos para medir la cantidad de un fluido que posee un recipiente, dependiendo del lugar de aplicación, un método u otro será más adecuado.

Una técnica sencilla para precisar el nivel de un fluido es introducir una regla fraccionada y observar la longitud que queda empapada al extraerla, pero este método no se precisa exactamente la automatización.

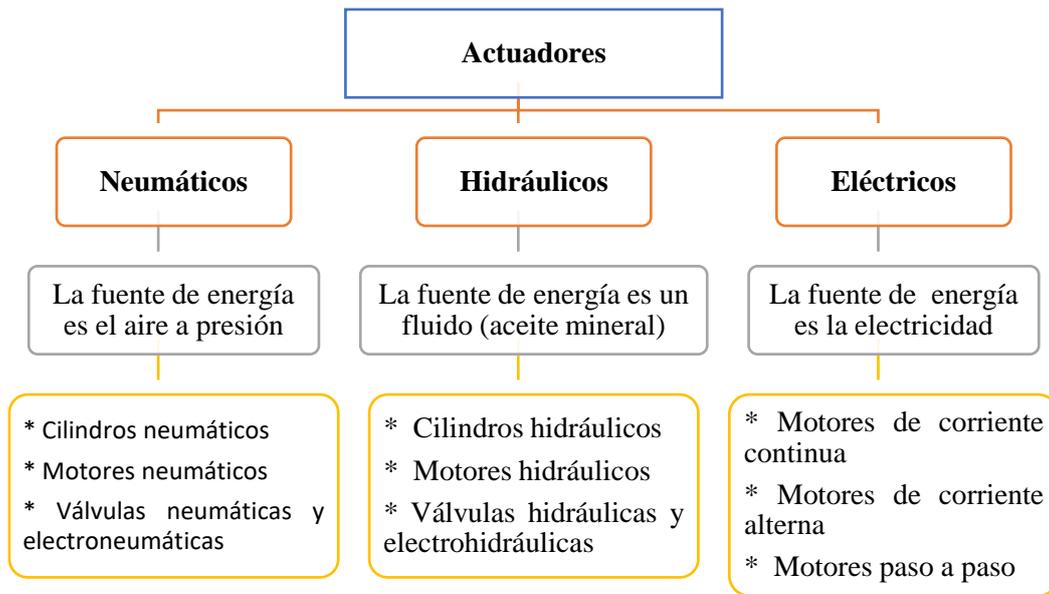


Figura 12. Clasificación de los actuadores dependiendo del tipo de energía utilizada [15].

A continuación, se describen formas de medir el nivel de un fluido [9]:

- **Boya con acoplamiento mecánico:** es una manera para obtener una señal eléctrica, que en base al Principio de Arquímedes, transforma el desplazamiento de la superficie libre en una fuerza θ par, y éstos en un ángulo de giro como se muestra en la Figura 13.
- **Sondeo de peso:** Otra opción es utilizar una boya con una polea y un contrapeso como indica la Figura 13, donde el ángulo girado por la polea es ajustado al nivel de líquido. El flotador se contiene en un tubo-guía para evitar que se esté moviendo lateralmente.
- **Presión Hidrostática, Conductancia y Capacitancia:** medir la diferencia de presiones entre el fondo del depósito y la superficie del líquido es otro mecanismo como se presencia en la Figura 13. Este método se puede aplicar tanto en depósitos abiertos como depósitos cerrados a presión.
- **Ultrasónico:** la altura del nivel se calcula por el tiempo que tardan los impulsos ultrasónicos en viajar desde el sensor hasta la superficie del medio y volver. Las propiedades químicas y físicas del medio no influyen en el resultado de la medición, como se presencia en la Figura 13. Por lo tanto se pueden medir sin problemas, medios agresivos y abrasivos, viscosos y pegajosos.

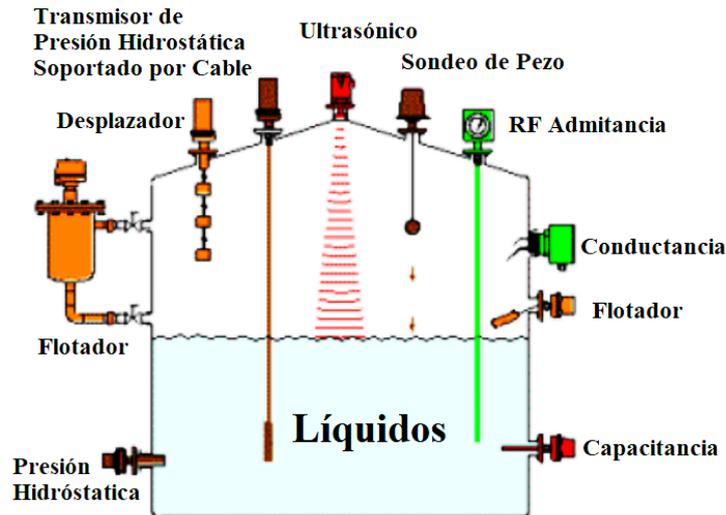


Figura 13. Diversos sensores de nivel [9].

Para el desarrollo del proyecto se consideró el sensor boya con acoplamiento mecánico, debido a sus excelentes características, la cual al desplazarse la boya nos genera una variación resistiva que se traduce como variación de voltaje, que será interpretada por el controlador como nivel de fluido.

4.4.2 Relés electromagnéticos

Es un interruptor mecánico accionado eléctricamente. Funciona como un interruptor controlado por un circuito eléctrico en el que, por medio de una bobina y un electroimán, se acciona un juego de uno o varios contactos que permiten abrir o cerrar otros circuitos eléctricos independientes.

Una característica importante de los relés es que la parte que emite la señal para activar el relé está aislada de la parte del relé que pone en marcha o enciende el receptor.

Funciona mediante la fuerza magnética que ejercen entre si dos piezas imantadas. La **Figura 14** se muestra que una pieza es móvil y la otra fija. La corriente o tensión que se mide es aplicada a la bobina (E). El elemento móvil (M), que lleva un resorte opuesto (R), accionara los contactos (C) del circuito de protección de alarma, según la corriente o tensión a la que esté sometido el bobinado [17] [18].

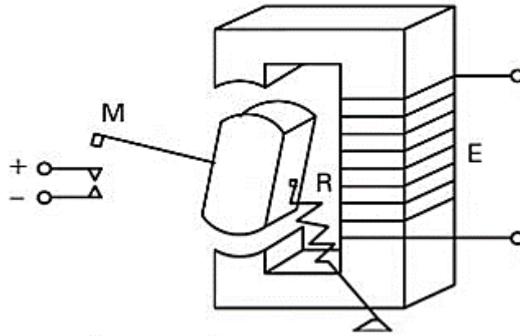


Figura 14. Relé electromagnético [17].

Tipos de contactos y acciones.

El contacto móvil de un relé o interruptor se llama polo. Un relé sencillo tiene un único polo. En un relé con dos polos hay dos contactos móviles accionados paralelamente. Al activar el dispositivo, el polo se mueve hasta que toca un contacto cuya posición es fija. Para cada polo puede haber uno o dos contactos fijos [18].

Se habla en consecuencia de relés de:

- Solo un polo solo un tiro (Single Pole Single Throw, *SPST*)
- Doble tiro de un solo polo (Single Pole Double Throw, *SPDT*)
- Polo doble tiro doble (Dual Pole Dual Throw, *DPDT*)

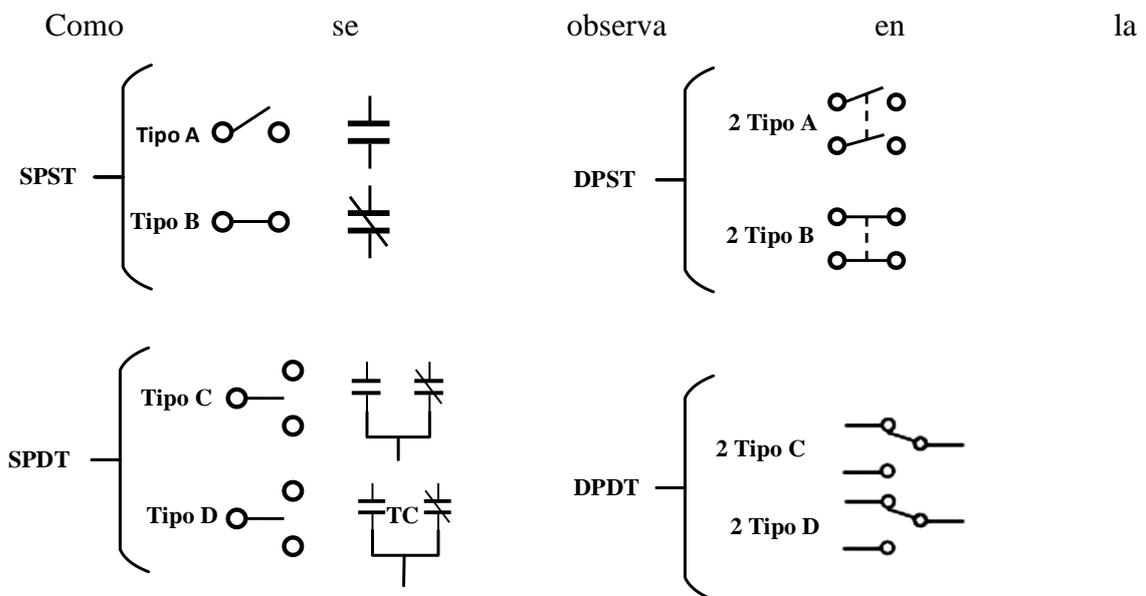


Figura 15. Los relés SPST poseen dos posiciones que son:

- Contactos normalmente abiertos que se denominan de Tipo A.

- Y los contactos normalmente cerrados, se denominan de Tipo B.

También existen relés con dos posiciones o contactos fijos, cuya acción de apertura (break⁴) y de cierre (make⁵) es tal que queda garantizado que se abrirá un circuito antes de cerrar el otro (bbm, break before make), mientras que en otros sucede a la inversa. Se cierra un circuito antes de abrir el otro (mbb, make before break). Los relés SPDT se denominan entonces de:

- Tipo C, si son bbm.
- Tipo D o de transferencia continua, si son mbb.

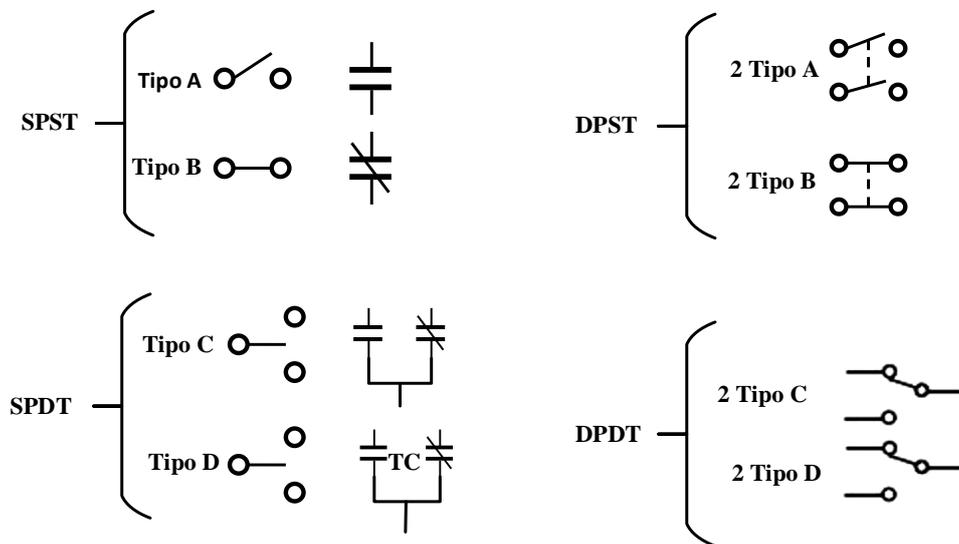


Figura 15. Símbolos para relés con diferentes tipos de contactos [18].

Para el desarrollo de este proyecto se utilizará el relé con el tipo de contacto SPST, utilizando el contacto normalmente abierto Tipo A, la cual permitirá activar un circuito en las condiciones requeridas por las instrucciones de programación previamente configuradas en el controlador.

4.5 Sistemas de Monitoreo

El sistema de monitorización o subsistema de monitorización es el encargado de hacer un seguimiento del estado del sistema completo, tanto de la infraestructura como del resto

⁴ Break término del inglés, su traducción es romper. En el texto hace referencia a abrir un circuito.

⁵ Make Término del inglés, su traducción es hacer o fabricar. En el texto hace referencia a cerrar un circuito.

de subsistemas, con el fin de asegurar la fiabilidad y estabilidad de los servicios que provee el conjunto [19].

Este tipo de sistemas se basan en la recogida de métricas, procesamiento y visualización de los datos junto con la generación de alertas cuando sucede algún fenómeno que puede ser un síntoma de un riesgo o mal funcionamiento [20].

El monitoreo puede efectuarse con el personal y sus herramientas en mano, o bien con la intervención automática de los equipos de instrumentación completados con dispositivos de transmisión a través de redes locales o remotas, que envían información a un monitor o a un panel de administración de las variables.

Los instrumentos que permiten al operador del sistema de distribución monitorizar y controlar las cantidades de flujo, presiones, niveles y otras variables a sensor se observan en la **Tabla 5** [21].

Tabla 5 Categorías principales de Instrumentación y Control [21].

Categorías	Características
Instrumentos primarios	Miden flujo, presión, niveles y temperatura.
Instrumentos secundarios	Despliegan la información que proviene de los instrumentos primarios y responde a ellos.
Sistemas de control	Operan los equipos como bombas y válvulas, en forma manual o automática.

Los sistemas de monitoreo son primordiales en las industrias en donde se puede implementar con la finalidad de optimizar recursos. Se puede diferenciar entre sistemas de monitoreo local y remoto [22].

Los sistemas de control de dispositivos definen a nivel de programación la información que reciben. Puede haber un controlador solo o varios distribuidos por el sistema. Es utilizada en ingeniería para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas.

4.5.1 Rendimiento del sistema de monitoreo

El rendimiento del monitoreo de aplicaciones (application performance monitoring , *APM*) es un área de la tecnología de la información (TI) que se centra en asegurarse de que los programas de aplicaciones de software se desempeñan tal y como se espera. El

objetivo de la supervisión del rendimiento es proporcionar a los usuarios finales una experiencia de calidad [19].

Las herramientas de monitoreo de aplicaciones proporcionan a los administradores la información que necesitan para descubrir rápidamente, aislar y resolver problemas que impactan negativamente en el rendimiento de una aplicación. Tales herramientas pueden ser específicas para una aplicación en particular como en el caso del monitoreo y control del generador eléctrico.

Para la eficiencia del monitoreo es recomendable agrupar en una plataforma todo el sistema de control que visualice el estado de cada componente que forma la estructura que en el caso de estudio se hará mediante indicadores gráficos, a través de la web.

4.5.2 Arquitectura de un sistema de monitoreo

La arquitectura en un sistema de monitoreo es la estructuración del hardware necesario (controlador, sensores y actuadores) de tal manera que con la lógica programable adecuada permita supervisar el comportamiento del sistema.

En el caso del proyecto se utilizará un controlador basado en la plataforma IDE Arduino, así mismo para la adquisición de información de los parámetros de interés, se requiere de un sensor del nivel de combustible y un sensor de nivel de energía.

En la parte del software se analiza la información adquirida de los sensores mediante el controlador y se actúa a consecuencia de la lógica programable para que cumpla con las funciones requeridas.

En este proyecto al tratarse de variables que fluctúan lentamente en el tiempo se realiza un monitoreo asíncrono.

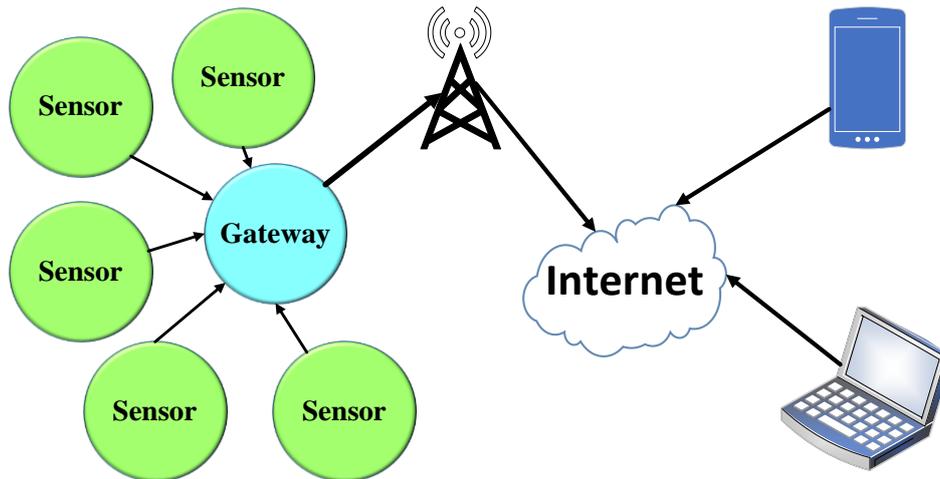


Figura 16. Esquema de un proceso de monitoreo de sistema [23].

4.6 Aplicación WEB

Es una herramienta informática a la que se puede acceder desde un navegador usando la red internet o bien por medio de una red local por la que el usuario pueda ingresar a un servidor web a través de un navegador. Se programa en un lenguaje interpretable por los navegadores web en la que se confía la ejecución al navegador.

Las aplicaciones web son utilizadas debido a la capacidad del navegador web como cliente ligero, autonomía del sistema operativo, facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar *software*⁶ a miles de usuarios potenciales. Existen una gran variedad de aplicaciones web como son: los correos web, wikis, blogs, tiendas en línea entre otras bastante conocidas.

Además, algunas páginas web pueden contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite el acceso interactivo del usuario con los datos, como puede ser el acceder a gestores de base de datos de todo tipo, rellenar y enviar formularios, entre otras [24] .

Un desarrollo web comprende múltiples áreas de conocimiento como se destacan en el diagrama de la Figura 17.

⁶ Software: término del inglés que hace referencia al soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas

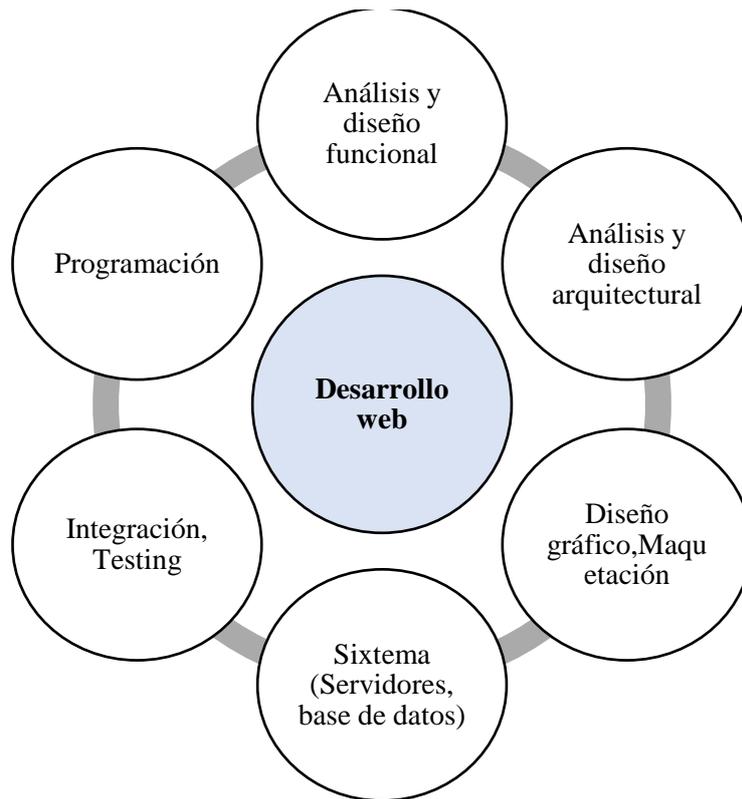


Figura 17. Áreas de conocimiento para el desarrollo web [25].

El desarrollo web tiene dimensiones muy variables. Podemos hablar desde un pequeño sitio web para una empresa local hasta un gran portal para una empresa internacional. De cualquier manera, puede decir que la programación web y el diseño web están relacionados. Sin embargo, un pequeño desarrollo puede ser realizado por una persona, mientras que un gran desarrollo requiere de un equipo de trabajo especializados en el análisis, diseño, programación, sistemas, integración, pruebas, entre otras funciones [25].

Si observamos a los lenguajes o tecnologías necesarias para el desarrollo web, podríamos hacer una clasificación tal como se observa en el diagrama de la Figura 18.

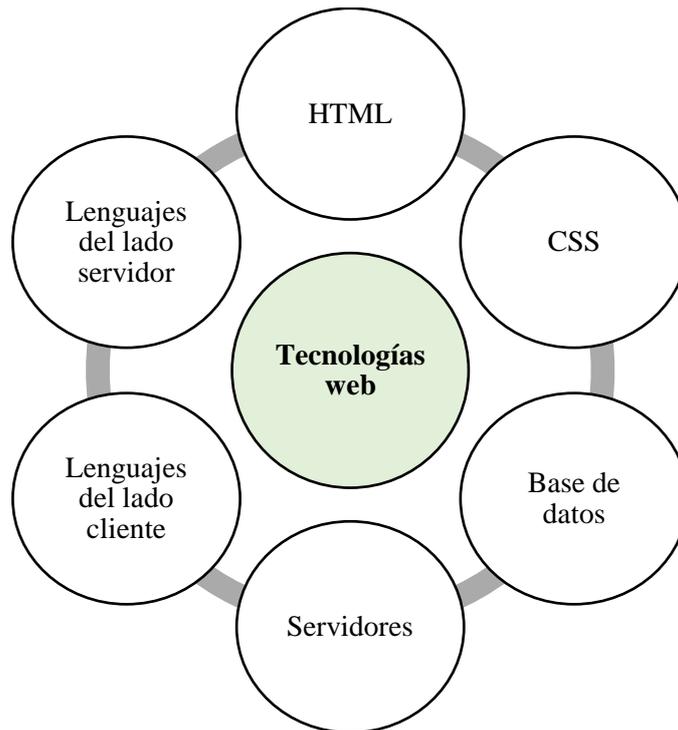


Figura 18. Tecnologías para el desarrollo web [25].

HTML y CSS son tecnologías que intervienen prácticamente en cualquier desarrollo, grande o pequeño. Son responsables de proporcionar una estructura y un diseño agradables para la visualización de los sitios web.

Los lenguajes de programación del lado del servidor son los encargados de ejecutar procesos en el servidor, por ejemplo la computadora remota responsable de enviar páginas web a través de Internet. Se puede citar estos lenguajes entre los principales: Java (JSP), ASP.NET, PHP o Perl.

Los lenguajes de programación del lado del cliente son los encargados de realizar los procesamientos en el ordenador personal del usuario, estos incluyen Javascript, Java o VBScript, entre ellos los principales. En cuanto a las bases de datos, podemos citar MySQL, SQLServer y Oracle [25].

4.6.1 Lenguaje de Marcas de Hipertexto (HTML)

Lenguaje de Marcas de Hipertexto (HyperText Markup Language, *HTML*) es el componente más básico de la Web. Define el significado y la estructura del contenido web [26]. HTML no es un lenguaje de programación que consiste en una serie de elementos que se usan para encerrar diferentes partes del contenido para que se vean o comporten de

una determinada manera. Las etiquetas de encierre pueden hacer de una palabra o una imagen un hipervínculo a otro sitio; se pueden cambiar palabras a cursiva, agrandar o achicar la letra, etc.

Las partes principales del lenguaje son [26]:

- Etiqueta de apertura: consiste en el nombre del elemento, encerrado por paréntesis angulares (< >) de apertura y cierre. Establece dónde comienza o empieza a tener efecto el elemento.
- Etiqueta de cierre: es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (/) antes del nombre de la etiqueta. Establece dónde termina el elemento.
- Contenido: este es el contenido del elemento, alberga la información
- El elemento: está formado por la etiqueta de apertura, la etiqueta de cierre y el contenido.

Anatomía de un documento HTML [26]:

- `<!DOCTYPE html>` — el tipo de documento. Es un preámbulo requerido.
- `<html></html>` . Este elemento encierra todo el contenido de la página entera y, a veces, se le conoce como el elemento raíz (*root element*).
- `<head></head>` . Este elemento actúa como un contenedor de todo aquello que quieres incluir en la página HTML que *no* es contenido visible por los visitantes de la página. Incluye cosas como palabras clave (keywords), una descripción de la página que quieres que aparezca en resultados de búsquedas, código CSS para dar estilo al contenido, declaraciones del juego de caracteres, etc.
- `<meta charset="utf-8">`. Elemento que establece el juego de caracteres que un documento usará en `utf-8`, incluye casi todos los caracteres de todos los idiomas humanos.
- `<title></title>` establece el título de tu página, que es el título que aparece en la pestaña o en la barra de título del navegador cuando la página es cargada, y se usa para describir la página cuando es añadida a los marcadores o como favorita.
- `<body></body>` . Encierra *todo* el contenido que deseas mostrar a los usuarios web que visiten tu página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás.

Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia y presentación de una página web (Lenguaje CSS) o la funcionalidad y comportamiento (JavaScript).

4.6.2 Lenguajes de programación

Un lenguaje de programación es un sistema de comunicación estructurado, compuesto por conjuntos de símbolos, palabras clave, reglas semánticas y sintácticas que permiten el entendimiento entre programadores y máquinas. Se puede listar diversos lenguajes de programación tal como se indica a continuación [27].

Lenguaje de programación PHP

Es de código abierto muy usado en el desarrollo web, ya que puede incrustarse en HTML. El código se ejecuta en el servidor, genera HTML y se lo envía al cliente.

Se puede mencionar las siguientes ventajas:

- Muy fácil de usar.
- Muy completo para los programadores profesionales.
- Se orienta al desarrollo de aplicaciones web dinámicas.
- Es un código seguro y confiable, ya que el código fuente está oculto para el navegador y cliente.
- Licencia abierta de uso.

Entre sus desventajas se encuentran:

- Necesitas un servidor web para ejecutar una aplicación, aunque las últimas versiones traen un servidor incorporado.
- Tener un conocimiento genérico de HTML
- Las últimas versiones hacen que PHP sea un lenguaje orientado a objetos.

Lenguaje de programación Perl

Se lo utiliza en la construcción de aplicaciones CGI para la web. Se trata de un lenguaje orientado a la extracción de información de archivos de texto y generar informes a partir de su contenido.

Se puede mencionar las siguientes ventajas:

- Licencia abierta
- Lenguajes de programación muy sencillo.
- Muy eficiente y de elevado rendimiento a la hora de tratar un gran volumen de datos.
- Puede usarse tanto en desarrollo web como en otros entornos.
- No necesitas hacer instalaciones a posteriori, ya que el intérprete viene por defecto.
- Un mismo código corre en multitud de distribuciones.

Entre sus desventajas se encuentran:

- Lento comparado con otras herramientas, ya que se compila al principio de su ejecución.
- Su código es difícil de leer.
- No dispone de control de excepciones
- Es complicado encontrar algunos errores.

Lenguaje de programación Java

Es uno de los lenguajes de programación de código abierto más populares. Java fue comercializado por primera vez 1995 por Sun Microsystems, este lenguaje de programación se encuentra detrás de muchas webs y aplicaciones, consolas y súper ordenadores, de móviles y de hasta Internet.

Se puede mencionar las siguientes ventajas:

- Lenguaje multiplataforma, ejecutable en la mayoría de los sistemas operativos.
- Software de distribución libre.
- Lenguaje completo y dispone de una librería y utilidades amplias.

Entre sus desventajas se encuentran:

- Lenguaje de ejecución lenta al ser interpretado.
- Difícil de aprender debido a su compleja sintaxis.

Lenguaje de programación C++

C++ permite la manipulación de objetos y tiene como base el lenguaje C. Se le reconoce como uno de los lenguajes de programación multiparadigma (programación

orientada a objetos y estructurada). Es una de las herramientas más populares por su versatilidad y potencia.

Se puede mencionar las siguientes ventajas:

- Lenguaje muy didáctico, y a través del mismo puedes aprender otros como C#, Java, PHP.
- Muy utilizado en la creación de sistemas operativos, utilidades, bibliotecas...
- Permite programar con múltiples estilos, como los estructurados.
- Admite la compilación de nuestro código en múltiples plataformas.

Entre sus desventajas se encuentran:

- El uso de librerías dinámicas se hace muy complicado, ya que la carga y la liberación de memoria de la librería corre a cargo de los programadores.
- No es muy usable para desarrollar páginas web.
- Complejo programar bases de datos.
- Más pesado que otros programas similares.

Lenguaje de programación Python.

Se crea a finales de los 80 y principios de los 90. Su programación se orienta a objetos y se emplea para hacer aplicaciones genéricas (juegos, web o de escritorio), en la informática científica y big data. Su código es de distribución libre.

Se puede mencionar las siguientes ventajas:

- Es escalable y flexible.
- Funciona en múltiples plataformas (Windows, Mac, Linux).
- Soporte de programación de interfaces gráficas de usuario.
- Conexión con las bases de datos más usadas.
- Se integra fácilmente con lenguajes como C, C++, y Java entre otros.

Lenguaje Arduino (Wiring)

El equipo que desarrolló la tarjeta Arduino creó un lenguaje propio (Wiring), para programar estas tarjetas. Este es un framework⁷ de código abierto para la programación de microcontroladores [28].

Wiring está escrito en los lenguajes de programación C y C++. De allí que la sintaxis sea parecida a la usada en estos lenguajes, además Wiring cuenta con un IDE basado en Processing donde se puede realizar la programación de los microcontroladores

Se puede mencionar las siguientes ventajas:

- Al usar framework permite soportar una gran variedad de microcontroladores.
- Soporta microcontroladores atmega que son los usados por las familias de tarjetas de Arduino.
- Se encuentra disponible para Windows, Linux y MAC

Existe gran cantidad de lenguajes de programación que pueden servir para crear un servidor web. En este caso se utilizará el hardware de Arduino para albergar el servidor web, la cual se diseñará una página web con ayuda de un programa para desarrollo web como es Geany, que nos permite configurar el lenguaje de escritura de texto HTML y las funcionalidades del lenguaje CSS para la apariencia y presentación. A la vez el lenguaje de programación que se maneja en Arduino IDE es Wiring que está escrito en los lenguajes de programación C y C++ la cual permitirá programar las instrucciones necesarias al controlador en base al comportamiento de la información adquirida por los sensores.

4.7 Plataforma de IoT

El Internet de las Cosas (IoT) consiste en dotar de inteligencia a las cosas, mediante sensores, actuadores, microcontroladores entre otros. Gracias a esto se obtiene una gran cantidad de información en forma de datos que es posible representar y trabajar con ellos. En sí, se define como el software que permite conectar todo en un sistema IoT permitiendo monitorizar, supervisar y controlar los datos [29].

⁷ Framework: es una especie de plantilla, esquema o estructura conceptual de base tecnológica que nos permite trabajar de una manera mucho más sencilla.

Las plataformas IoT están alojadas en grandes clusters con gran cantidad de espacio para poder acumular grandes volúmenes de datos, de donde surge la denominación Big Data. Gracias a una gestión multiusuario y a su interfaz, permite trabajar con los datos almacenados, pudiendo entre otras cosas poder representarlos gráficamente o aplicar minería de datos.

Las plataformas IoT pueden adaptarse a las necesidades del cliente pero también las podemos encontrar genéricas. La mayoría incluyen las siguientes características [29]:

- Conectividad: Adaptación de varios protocolos de comunicaciones. MQTT, CoAp, etc.
- Gestión de dispositivos: Permiten la utilización de distintos dispositivos independientemente de sus marcas, gracias a los protocolos de comunicación.
- Base de datos: Para un almacenamiento masivo de datos, que permita análisis de estos.
- Gestión de los datos: Para interactuar con el cliente.
- Representación gráfica: Representación de uno o más datos en función del tiempo.
- Análisis de datos: Hacer un estudio exhaustivo de los datos.

En el diagrama de la Figura 19 se puede apreciar algunas plataformas destacables orientadas al IoT que permiten monitorizar sistemas a distancia

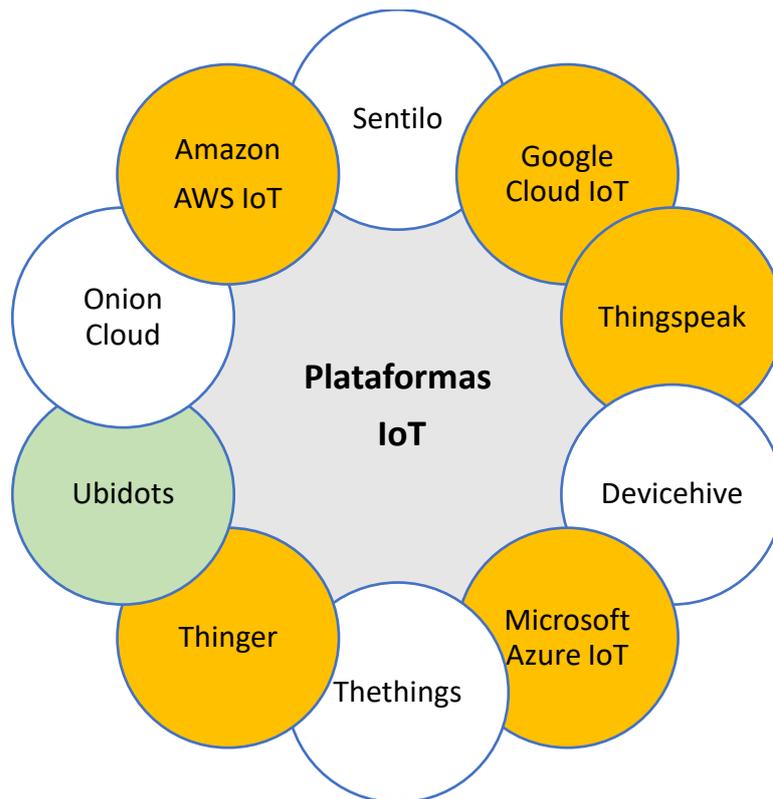


Figura 19. Plataformas orientadas al IoT [30].

4.7.1 Amazon AWS IoT

AWS ofrece servicios y soluciones del Internet de las cosas (IoT) para conectar y administrar miles de millones de dispositivos. Recopila, almacena y analiza los datos de IoT de las cargas de trabajo industriales, de consumo, comerciales y automotrices [31].

Los dispositivos conectados, como sensores, accionadores, dispositivos integrados, electrodomésticos inteligentes y dispositivos portátiles, se conectan a AWS IoT Core mediante HTTPS, WebSockets, LoRaWAN o MQTT seguro.

AWS IoT Core incluye una gateway para dispositivos que permite la comunicación bidireccional, segura, de baja latencia y baja sobrecarga entre dispositivos conectados, la nube y las aplicaciones móviles.

AWS IoT Core incluye también un motor de reglas que permite el procesamiento continuo de los datos que envían los dispositivos conectados. Puede configurar reglas

para filtrar y transformar los datos. En la Figura 20 se puede observar el proceso de las funciones de AWS IoT.

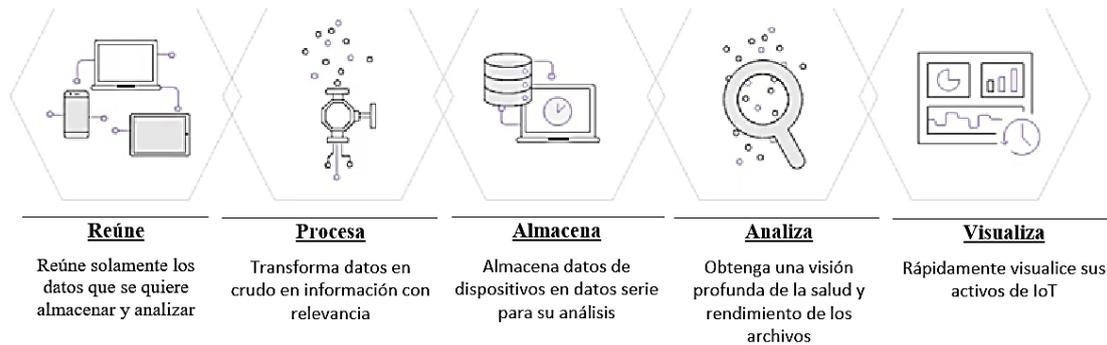


Figura 20. Funciones de AWS IoT [31].

4.7.2 Google Cloud IoT

Google Cloud Platform se trata de la suite de infraestructuras y servicios que Google utiliza a nivel interno y, ahora, disponible para cualquier empresa, de tal forma que sea aplicable a multitud de procesos empresariales.

Este conjunto de servicios ofrece prestaciones muy dispares; desde machine learning hasta Inteligencia artificial pasando por el big data, todo englobado bajo el paraguas del cloud computing.

Google Cloud IoT Core, está diseñado para ayudar a las empresas, incluidas las compañías de servicios públicos y las agencias de transporte, a conectar de forma segura dispositivos distribuidos a nivel mundial a Google Cloud Platform, en donde los datos se pueden gestionar de forma centralizada e integrarse con los servicios de análisis de datos de Google, entre sus principales prestaciones esta [32]:

- Permite utilizar su propio certificado. Los usuarios que ya han utilizado Cloud IoT Core han solicitado la posibilidad de verificar la propiedad de las claves del dispositivo. Además de la autenticación basada en una clave asimétrica para cada dispositivo.
- Conecta dispositivos existentes mediante http. Además del protocolo estándar MQTT, ahora es posible conectar, con más seguridad, dispositivos IoT

existentes y puertas de enlace a Cloud IoT Core mediante HTTP, para registrar de forma sencilla datos en el GCP a gran escala.

- Representación lógica de dispositivos. Ciertos casos de uso requieren una aplicación de IoT para recuperar el último estado y las propiedades de un dispositivo, incluso cuando el dispositivo no esté conectado.

4.7.3 Microsoft Azure IoT

Es una plataforma de aplicaciones IoT la cual se caracteriza por ser altamente seguro, permite escalar a medida que crece su negocio, garantiza que sus inversiones sean repetibles y se integra con sus aplicaciones empresariales existentes, sus precios son predecibles y está enfocado en la industria [33].

La plataforma ayuda a simplificar las interacciones con los dispositivos. El lenguaje de modelado abierto de la aplicación permite una integración perfecta del dispositivo a la nube y ayuda a reducir el tiempo de desarrollo, el costo y la complejidad, la cual permite acelerar las soluciones sin escribir código para insertar.

4.7.4 ThingSpeak

ThingSpeak es un servicio de plataforma de análisis de IoT que le permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Puede enviar datos a ThingSpeak desde sus dispositivos, crear una visualización instantánea de datos en vivo y enviar alertas [34].

Características de ThingSpeak:

- Recolectar datos en canales privados
- Compartir datos con canales públicos
- API RESTful y MQTT
- Análisis y visualizaciones de MATLAB
- Programación de eventos
- Alertas
- Integraciones de aplicaciones

Funciona con:

- MATLAB y Simulink
- Arduino
- Dispositivos de partículas
- Módulos ESP8266 y ESP32
- Frambuesa Pi
- LoRaWAN
- Red de cosas
- Senet
- Libelio
- Beckhoff

4.7.5 Thinger.io

Es una plataforma IoT en la nube que proporciona todas las herramientas necesarias para crear prototipos, escalar y administrar productos conectados de una manera muy simple. El objetivo de Thinger es democratizar el uso de IoT haciéndolo accesible a todo el mundo y agilizar el desarrollo de grandes proyectos de IoT [35].

Características de Thinger.io

- Plataforma IoT gratuita : proporciona una cuenta gratuita de por vida con solo algunas limitaciones para comenzar a aprender y crear prototipos cuando su producto esté listo para escalar, puede implementar un servidor Premium con capacidades completas en minutos.
- Simple pero poderoso : solo un par de líneas de código para conectar un dispositivo y comenzar a recuperar datos o controlar sus funcionalidades con nuestra Consola basada en la web, capaz de conectar y administrar miles de dispositivos de una manera simple.
- Independiente del hardware: cualquier dispositivo de cualquier fabricante se puede integrar fácilmente con la infraestructura de Thinger.io.
- Infraestructura extremadamente escalable y eficiente: gracias al paradigma de comunicación único, en el que el servidor IoT suscribe los recursos del dispositivo para recuperar datos solo cuando es necesario, una sola instancia

de Thinger.io puede administrar miles de dispositivos IoT con poca carga computacional, ancho de banda y latencias.

- Open-Source : la mayoría de los módulos de la plataforma, librerías y código fuente de la APP están disponibles en nuestro repositorio de Github para ser descargados y modificados con licencia MIT

4.7.6 Ubidots

Ubidots es una plataforma orientada al IoT que facilita el proceso de configuración de la interfaz gráfica de usuario (GUI). Además, sirve para monitorizar y/o controlar ambientes. La plataforma Ubidots recepta información desde cualquier dispositivo con acceso a Internet, configura acciones y alertas basadas en sus datos en tiempo real y desbloquea el valor de sus datos a través de herramientas visuales [36].

Ubidots ofrece una API REST que le permite leer y escribir datos en los recursos disponibles: fuentes de datos, variables, valores, eventos y conocimientos. La API admite HTTP y HTTPS y se requiere una clave API.



Figura 21. Logo de Ubidots [36].

La desventaja de usar una plataforma orientada al IoT es que se genera un costo mensual de utilización de la plataforma, pero con la ventaja de que se puede tener acceso al sistema a través de internet.

4.7.7 Analices de plataforma IoT a utilizar

Existen una gran cantidad de plataformas orientadas al internet de las cosas que permiten monitorear sensores a través de una GUI web, pero no todas esas plataformas poseen las librerías necesarias para establecer comunicación con los equipos requeridos.

En este proyecto se requiere de las librerías para la tarjeta de red Ethernet Shield W5100, la cual como se puede observar en la Tabla 6 solo tres plataformas IoT posee esta funcionalidad. De las tres se ha seleccionado la plataforma Ubidots debido a su simplicidad de uso y su capacidad de funcionalidad con una versión gratuita.

Tabla 6 Análisis de funcionalidad de ciertas plataformas IoT.

Fuente: Autor

Plataformas IoT	Compatible con Arduino	Posee librería Ethernet Shield	Seguridad	sistema de alertas	Plan Gratuito	Capacidad de implementación
Amazon AWS IoT	SI	NO	SI	SI	NO	No aplica
Google Cloud IoT	SI	NO	SI	No	Limitado	No aplica
Microsoft Azure IoT	SI	NO	SI	SI	Limitado	No aplica
ThingSpeak	SI	SI	SI	SI	Limitado	Si aplica
Thinger.io	SI	SI	SI	SI	Limitado	Si aplica
Ubidots	SI	SI	SI	SI	Limitado	Si aplica

4.8 Hardware de control para monitoreo de variables

Existen gran cantidad de dispositivos orientados a controlar la información que reciben de diversas fuentes para actuar a consecuencia, en las que se puede mencionar: los microcontroladores de Arduino, circuito integrado programable (Programmable Integrated Circuit, *PIC*) y el controlador lógico programable (programmable logic controller, *PLC*) [37].

Estos están diseñados para múltiples señales de entrada y salida, rangos de temperatura amplios, inmunidad al ruido electrónico y resistencia a la vibración y al impacto [38].

En los siguientes apartados se analizan las características de los principales tipos de controladores existentes en la industria.

4.8.1 Circuito Integrado Programable PIC

Es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S. Estos funcionan dependiendo las instrucciones que se le graben a su memoria y su capacidad de almacenamiento es acorde al tipo y o modelo de microcontrolador.

Los microcontroladores PIC ofrecen una amplia gama de dispositivos desde 6 hasta 100 pines, desde los más sencillos de 8 bits hasta los más poderosos de 32 bits.

La gama de microcontroladores de 8 bits consta de más de 400 modelos diferentes para elegir. Microchip es actualmente el mayor fabricante de microcontroladores de 8 bits en todo el mundo. Sus familias de 8 bits más importantes son: PIC12, PIC16 y PIC18. En la Figura 22 se muestra el PIC16F84A, que es de gama media, muy versátil, económico y fácil de usar. Este PIC es ideal para aprender las bases tanto del uso como de la programación de microcontroladores PIC. Además, una vez que conozcamos bien a fondo este dispositivo, la migración hacia cualquier otro microcontrolador PIC será bastante fácil. Incluso la compatibilidad entre los microcontroladores PIC hace que los programas puedan migrar de uno hacia otro con muy pocos cambios en ellos [39].



Figura 22. PIC16F84A [39].

El PIC viene a ser un chip que permite obtener un circuito integrado a la medida que actuaría según las instrucciones programadas que se le graba en la memoria ROM interna [40].

El programa que se cargue debe estar en un lenguaje comprensible por el chip, que viene a ser el lenguaje de maquina en código binario, aunque al PIC se le introduce el programa en código hexadecimal, internamente lo convierte en binario para realizar su interpretación [40].

4.8.2 Controlador lógico programable PLC

Dispositivo desarrollado para sustituir los circuitos secuenciales de relevadores para el control de dispositivos. Dependiendo de las señales de entrada este conecta o desconecta sus salidas [41].

PLC básico está formado por entradas y salidas digitales. En sus entradas se pueden conectar elementos como:

- Pulsadores.
- Swiches y micro-swiches.
- Sensores ópticos, magnéticos y capacitivos.
- Sensores digitales para variables como temperatura, presión, nivel, humedad, entre otros.

Las salidas de los PLC son digitales⁸ por la que se puede conectar actuadores digitales que controlan la potencia de motores, válvulas, lámparas y resistencias eléctricas. Debido a que son implementados en gran variedad de aplicaciones reales, cualquier aplicación que necesite algún tipo de control eléctrico se lo puede realizar con un PLC, ya que este es una computadora especializada para controlar máquinas y procesos en ambientes industriales para maniobrar en tiempo real.

El PLC puede operar la lógica y secuencia de los procesos haciendo arreglos de las combinaciones que se puedan dar entre entradas y salidas y el uso de herramientas como temporizadores y contadores. El usuario introduce las instrucciones, normalmente vía software que proporciona los resultados deseados.

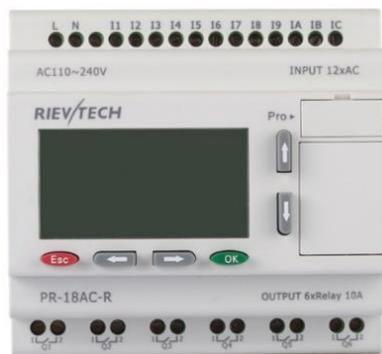


Figura 23. Modelo de una PLC. Fuente: RIEVTECH

4.8.3 Hardware Arduino

Es una plataforma de código abierto para prototipos de electrónica apoyada en una integración de hardware y software simple, fácil de usar y flexible. Su diseño está enfocado hacia artistas, diseñadores o cualquier interesado en crear ambientes y objetos interactivos. Arduino permite medir y percibir ambientes al acoger como entrada una variedad de sensores e interruptores. También sirve para controlar lugares al manejar

⁸ Salidas digitales. Se les llama digitales porque solo pueden tomar dos estados; prendidos o apagados, activos o no activos

dispositivos como luces, motores y varios otros actuadores. Además, posee las tres "B" Bueno, Bonito y Barato. Su placa más popular el Arduino UNO, se basa en el microcontrolador Atmega328 de Atmel [42].

Este microcontrolador 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 o altavoces en los que reproducir los datos procesados, he incluso pueden ser otras placas o controladores [43].



Figura 24. Periféricos de Arduino [43].

Arduino es un proyecto y no un modelo concreto de placa, lo que significa que compartiendo su diseño básico te puedes encontrar con diferentes tipos de placas. Las hay de varias formas, tamaños y colores para las diversas necesidades de un proyecto. Existen placas para funcionalidades sencillas o con características mejoradas. Dependiendo de los requisitos de funcionalidad su costo variara [44].

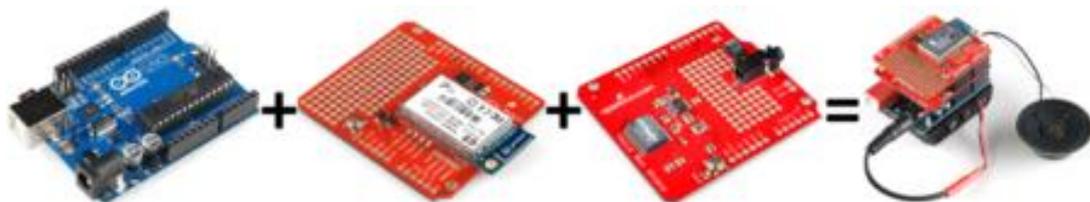


Figura 25. Tipos de placas orientadas al internet de las cosas [43].

Además, las placas Arduino también cuentan con otro tipo de componentes llamados Escudos (Shields) o mochilas. Se trata de una especie de placas que se conectan a la placa principal para añadirle una infinidad de funciones, como GPS, relojes en tiempo real, conectividad por radio, pantallas táctiles LCD, placas de desarrollo, y un larguísimo etcétera de elementos.

“La mayoría de las placas Arduino constan de un microcontrolador AVR Atmel-8 bits (ATmega8, ATmega168, ATmega328, ATmega1280, ATmega2560), cada microcontrolador consta de diversas cantidades de memoria flash, pines y funciones. Las placas utilizan pines/cabezales hembra de una o dos hileras que facilitan las conexiones e incorporación en otros circuitos [43].”

4.8.4 Arduino en la Industria 4.0

Arduino cada vez se está ganando terreno en la industria y no solo como una herramienta de prototipo, sino que los sistemas de control se están haciendo cada vez más funcionales. Existen algunas empresas que se aprovechan de la plataforma libre de Arduino para crear sus propias versiones de hardware totalmente funcionales para la industria entre los que se puede mencionar están:

Controlino

Es un PLC de grado industrial con compatibilidad 100% Arduino. Se puede utilizar para casi cualquier propósito de control y automatización. Además, combina la flexibilidad y la naturaleza de código abierto del ecosistema Arduino con la seguridad y confiabilidad de los PLC de grado industrial [45].

Tabla 7 Características del módulo Controlino [45]

Categorías	Características
Pines de entrada digital	21 con protección ESD.
Pines de salida digital	24 12x High Side Switch, 12x Half Bridge 2A @ 12V o 24V).
Salidas de relé	16 (230 V a 6 A máx.).
Interfaz CONTROLLINO	a través de serie, RS485, I2C o SPI.
Conectividad	La comunicación se habilita a través de USB o Ethernet.
temporizador	Reloj en tiempo real integrado (RTC).

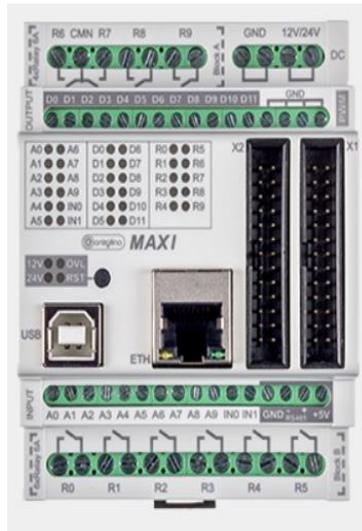


Figura 26. Controlador Controlino [45].

Módulo de E/S PLC Siemens, 6ES7647, 5 x Entrada/Salida, 300 mA, 28,8 V DC.

El módulo de expansión de IoT posee interfaces digitales y analógicas diseñadas para conectar puerta de enlace IOT directamente a sensores locales y otros dispositivos. Este módulo es robusto y está diseñado para funcionar las 24 horas del día, los 7 días de la semana en entornos de prueba. Además, permite conectividad de E/S local para gateway IOT. Sus E/S analógicas y digitales se ha diseñado para conectar directamente sensores locales y activar procesos no críticos [46].



Figura 27. Módulo Siemens basado en Arduino [46].

Tabla 8 Características del módulo Siemens [46].

Categorías	Características
Nivel de funcionalidad	Industrial (forma ininterrumpida).
condiciones ambientales	IOT2040 (por ejemplo, para 0 - 50 °C).
Certificación	CE y UL.
Direccionamiento de módulo	a través de GPIO (permite la selección arbitraria del lenguaje de programación y entorno de desarrollo).
Conectividad	La comunicación se habilita a través de USB o Ethernet.
CPU	Atmega 32u4 MCU.
Tipo de tensión de la alimentación	24 V DC.

Pines de entradas digitales	5 (5-12V DC).
Pines de salidas digitales	2 (0-28.8 V DC).
Pines de entrada analógica	2 (0-10 V).
RTC	Si.

Industruino

Es una placa compatible con Arduino con todas las funciones alojada en una caja montable, por la que permite instalar una aplicación Arduino en muy poco tiempo. Se usa para proyectos de automatización, registradores de datos o una instalación artística interactiva. Además, ofrece robustez industrial. Existen gran variedad de modelos ofreciendo cada modelo cierta funcionalidad, a continuación, se describirá las características del Industruino PROTO D21G [47].

Tabla 9 Características del módulo Industruino [47].

Categorías	Características
CPU	D21G equivalente a Arduino Zero / MKR.
Tipo de tensión de la alimentación	8 - 28 V DC.
Velocidad del reloj	16 MHz.
Flash Memory	256 KB (de los cuales 4 KB utilizados por el gestor de arranque).
SRAM	32 KB.
Pines de E/S digitales	17 (3,3 - 5V DC).
Canales PWM	12 .
Pines de entrada analógicos	12.
RTC	No.



Figura 28. Industruino [47].

M-DUINO PLC Arduino

Es una tecnología Arduino diseñado para un uso profesional. Contiene varios puertos de comunicación que proporcionan más flexibilidad y control. Ofrece la posibilidad de expandir hasta 127 módulos a través de I2C, lo que significa que puede tener hasta 7100 entradas / salidas en conexiones maestro-esclavo, además de sensores, etc. Existen

diversos modelos a continuación se describirá las características principales del M-DUINO PLC Arduino Ethernet 19R I/Os [48].

Tabla 10 Características del módulo M-DUINO PLC Arduino [48].

Categorías	Características
Pines de entrada digitales	2 (5-24Vdc).
Salidas Digitales y PWM	3 (5-24Vdc) Optoaisladas.
Pines de entrada analógicas	4 (5-24Vdc).
Salidas Relé	8 (220Vac - 5A).
Comunicaciones	RTC, μ SD, Ethernet, USB, Full/Half duplex, RS485, RS232, I2C, Modbus.
Flash Memory	256 KB
SRAM	8 KB
CPU	Arduino Mega



Figura 29 M-DUINO PLC 19R I/Os [48].

Arduino Uno R3

Arduino Uno r3 es la placa más robusta y la más documentada con respecto a Arduino que se puede encontrar, ya que es la tarjeta más utilizada de todas para realizar pruebas [49].

Tabla 11 Características del módulo Arduino Uno R3 [49].

Categorías	Características
CPU	ATmega328P
Voltaje de funcionamiento	5V
Voltaje de entrada	7-12V
Pines de E/S digitales	14 (6 con salida PWM).
Pines de entrada analógica	6.
Pines de salida	11.
Comunicaciones	RTC, μ SD, Ethernet, USB, Full/Half duplex, RS485, RS232, I2C, Modbus.
Programación	Arduino IDE.

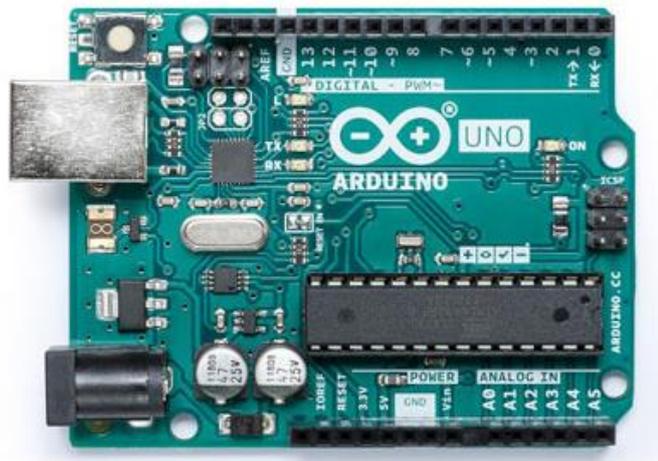


Figura 30. Arduino uno R3 basado en ATmega328P [49]

A continuación, en la Tabla 12 se hace un análisis de los factores de interés que posee los diferentes equipos basados en la plataforma Arduino que ayuden a establecer el controlador a utilizar para el sistema que se desea crear.

Tabla 12 Análisis de las capacidades que poseen lo diferentes hardware mencionados.
Fuente: Autor.

Equipo	Controlino	Módulo de E/S PLC Siemens, 6ES7647	Industruino	M-DUINO PLC	Arduino
Posibilidad de expandir	X	X	X	X	X
PLC de grado industrial	X	X	X	X	En prueba
Propósito de control y automatización.	X	X	X	X	X
Posee relés	X	X	X	X	Añadiendo módulos de Arduino relé
Reloj en tiempo real integrado	X	X	X	X	X
E/S analógicas y digitales	X	X	X	X	X
Funcionamiento 24/7	X	X	X	X	Añadiendo sistema de enfriamiento
La comunicación se habilita a través de USB o Ethernet	X	X	X	X	Añadiendo una Ethernet Shield

4.8.5 Aspectos relevantes costo / beneficio

Dependiendo de la tecnología del sistema de monitoreo a utilizar si es PIC, ARDUINO o PLC el costo variara y sus beneficios de igual manera. Si se utiliza el microcontrolador PIC el sistema de monitoreo se creará a la medida de los requerimientos actuales sin posibilidad de un a dimensionamiento futuro, este vendría hacer el más económico.

En el caso del microcontrolador Arduino es una placa funcional con diversas entradas y salidas de control lista para implementación, por la parte del hardware y con facilidad de programación. Además, posee la posibilidad de a dimensionamiento por la facilidad de implementar más módulos.

Finalmente, el Controlador PLC es una placa lista para la implementación del monitoreo del sistema con un hardware robusto capaz de cumplir a la perfección sus requerimientos. Su programación es medianamente sencilla, pero con el inconveniente de que su costo es elevado con respecto al PIC y a ARDUINO.

En la Tabla 13 se muestra un análisis de los factores a favor y en contra para seleccionar un microcontrolador para un sistema de control. La puntuación se asigna de tal manera que bajo equivale a 1, medio a 5 y alto a 10. Si el factor a medir es perjudicial se resta y si es beneficioso se suma. El valor más alto indica que es el más favorable para la ejecución del proyecto por tal razón se opta por el microcontrolador Arduino.

Tabla 13 Análisis de los aspectos relevantes de los microcontroladores.

Fuente: Autor

Características	Microcontrolador de Circuito Integrado Programable PIC		ARDUINO		Controlador lógico programable PLC	
	Valoración	Puntuación	Valoración	Puntuación	Valoración	Puntuación
Coste económico	bajo	-1	medio	-5	alto	-10
Coste de implementación	medio	-5	bajo	-1	bajo	-1
Facilidad de programación	medio	5	alto	10	medio	5
Funcionalidad	medio	5	alto	10	alto	10
Escalabilidad	bajo	1	alto	10	medio	5
	Total	5	Total	24	Total	9

Ya establecida la plataforma de Arduino ATmega con la que se va a trabajar para la ejecución del proyecto, existen gran variedad de modelos en esta marca que son útiles para automatización como se muestran en el apartado 4.8.4 , por la que es necesario analizar detenidamente cada una y determinar cuál es el hardware de monitoreo más adecuado, ajustándose a crear un sistema de lo más económico y funcional posible.

En la Tabla 14 se realiza una valoración económica del costo a implementar el hardware de monitoreo del generador eléctrico y SAI, usando diferentes marcas de controladores basados en Arduino. La puntuación más baja es la que requiere un menor coste de implementación y la más alta un mayor coste. Por tal razón se decide aplicar como controlador del sistema el hardware Arduino Mega.

Tabla 14 Comparación económica de equipos basados en Arduino.

Fuente: Autor

Equipo	Coste económico de hardware	Coste de implementación	Valoración
Controlino	-10	-1	-11
Módulo de E/S PLC Siemens, 6ES7647, 5 x Entrada/Salida	-5	-5	-10
Industruino	-10	-1	-11
M-DUINO PLC Arduino	-10	-1	-11
Arduino ATmega2560	-1	-5	-6

Un análisis de enfoque distinto se indica en la Tabla 15 en el cual el que tiene mayor puntaje es el más adecuado para la implementación del proyecto, pero a causa de lo establecido de crear el hardware más económico posible que cumpla con los requerimientos planteados en este proyecto se opta por el hardware Arduino Mega 2560.

Tabla 15 Comparación de facilidad de implementación del hardware.

Fuente: Autor

Equipo	Facilidad de programación	Funcionalidad	Escalabilidad	Valoración
Controlino	1	10	10	21
Módulo de E/S PLC Siemens	1	10	10	21
Industruino	1	10	10	21
M-DUINO PLC Arduino	1	10	10	21
Arduino Mega	1	5	10	16

Hasta este punto se ha recopilado información acerca de los equipos, materiales y componentes que forman un sistema de monitoreo y control de un generador eléctrico. La incorporación de estos equipos como son los sensores, actuadores y microcontroladores adecuados llevarán a crear un hardware que junto con la programación adecuada permitirá construir un sistema de monitoreo a través de la web para un generador eléctrico de combustión.

En la siguiente sección, se procederá a definir las características de los equipos, materiales y sensores que se van a utilizar para poder crear el sistema de monitoreo. De igual manera, se describen los métodos que se utilizarán para poder cumplir con los objetivos planteados.

5. MATERIALES Y MÉTODOS

5.1 Métodos

Para el desarrollo de este proyecto, se basa en la aplicación de algunos métodos como se indica en la Tabla 16.

Tabla 16 Métodos aplicados para el desarrollo del proyecto.

Fuente: Autor

MÉTODOS	
Cuantitativo	Cualitativo
Este método se aplica para: <ul style="list-style-type: none"> - Determinar el volumen de combustible a través de la medición de un sensor de nivel de combustible. - Determinar la carga de energía en el SAI. - Determinar el tiempo de funcionamiento del generador eléctrico. 	Este método se aplica para: <ul style="list-style-type: none"> - Determinar el estado de suministro de energía de la red pública. - Determinar el momento de encendido y apagado del generador de energía eléctrica.
Investigativo	Observación
<ul style="list-style-type: none"> - Se utiliza para obtención de información referente al tema de investigación. - Para adquirir los conocimientos previos para el desarrollo y posteriormente la implementación del sistema de monitoreo del generador eléctrico y el sistema de respaldo de energía. 	Se aplica para: <ul style="list-style-type: none"> - El levantamiento de la información (trabajo de campo). - Verificar la infraestructura actual.
Deductivo	Analítico
Ya conocida la infraestructura actual se utiliza este método para: <ul style="list-style-type: none"> - Determinar el problema actual del sistema. - Determinar los mecanismos a implementar para dar solución al problema. 	<ul style="list-style-type: none"> - Sirve este método para analizar los diversos recursos de hardware y software que permitan dar solución a la problemática. - Determinar los equipos a utilizar tomando en cuenta la relación costo beneficio.

Así mismo para poder llevar un orden de trabajo por etapas, se ha considerado conveniente crear un diagrama de trabajo que permita cumplir con los objetivos planteados tal como indica la Figura 31.

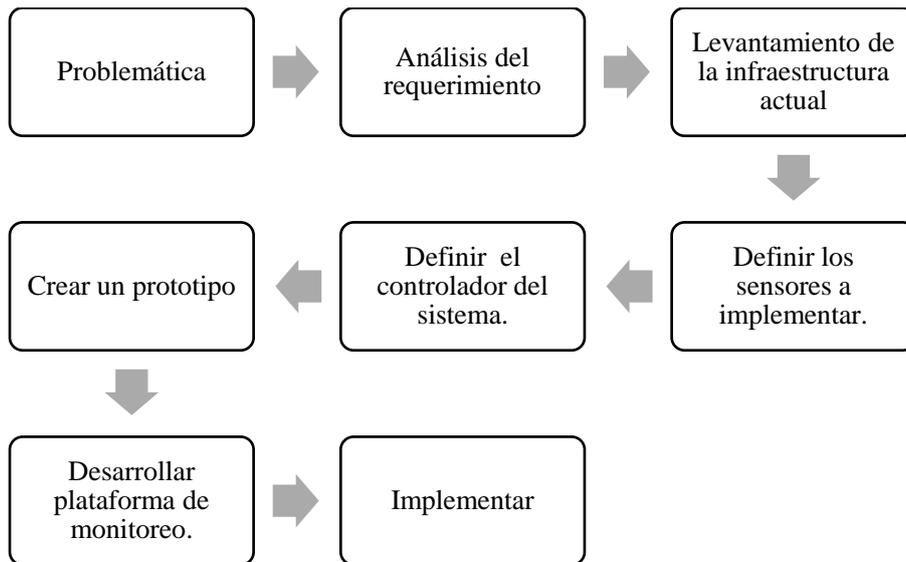


Figura 31. Diagrama de trabajo para el desarrollo del sistema de monitoreo.
Fuente: Autor

5.2 Materiales

5.2.1 Arduino Mega 2560

Es una placa electrónica con una plataforma de código abierto (open-source) lo que permite realizar proyectos y modificaciones tanto de hardware como de software a cualquier persona sin ningún problema. En la **Figura 32.** Partes de la placa electrónica Arduino Mega 2560 .Figura 32 se puede observar las partes de las que está compuesta la Arduino Mega 2560 [50].

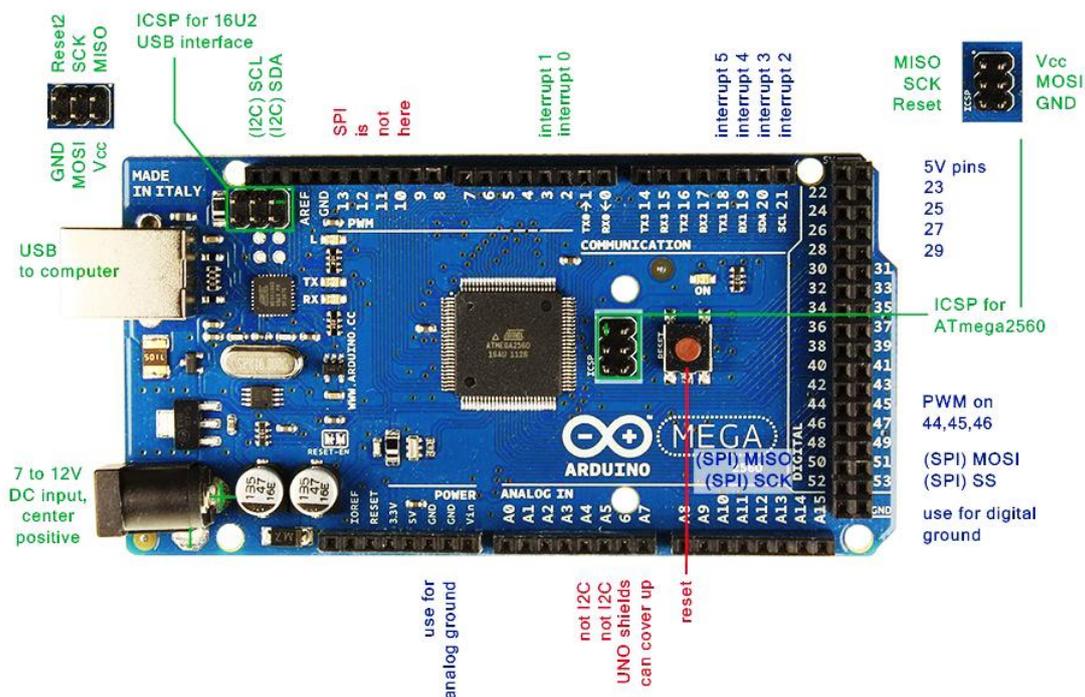


Figura 32. Partes de la placa electrónica Arduino Mega 2560 [51].

Tabla 17 Características técnicas del Arduino Mega 2560 [38].

Microcontrolador	ATmega 2560
Tensión de funcionamiento	5 V
Voltaje de entrada (recomendado)	7-12 V
Voltaje de entrada (límite)	6-20 V
Pines de E / S digitales	54
Pines de E / S digitales PWM	15
Pines de entrada analógica	16
Corriente CC por pin de E / S	20 mA.
Corriente CC para Pin de 3.3 V	50 mA.
Memoria flash	256 KB (8KB usados por el bootloader).
SRAM	8 KB
EEPROM	8 KB
Velocidad de reloj	16 MHz
LED_BUILTIN	13
Longitud	101,52 mm
Anchura	53,3 mm
Peso	37 g

Arduino puede ser alimentada de varias formas, con un cable USB conectado al ordenador o con una fuente externa.

- A través de fuente externa cuenta con un zócalo donde se ubica un jack de 2,1mm para conectar un adaptador que se encuentre entre los rangos de 7 – 12 V que es la tensión recomendada.
- Con un conector USB tipo-B para conectarlo al ordenador con el cual podemos programarlo y a su vez alimentarlo.

5.2.2 Arduino Ethernet Shield W5100 + Lector De SD

El Arduino Ethernet Shield es una tarjeta de red que permite a la placa de Arduino ATmega 2560 conectarse a internet. Se compone del chip Wiznet W5100 ethernet que proporciona la capacidad de utilizar los protocolos TCP y UDP. Además, soporta hasta cuatro conexiones de sockets simultáneas [52].

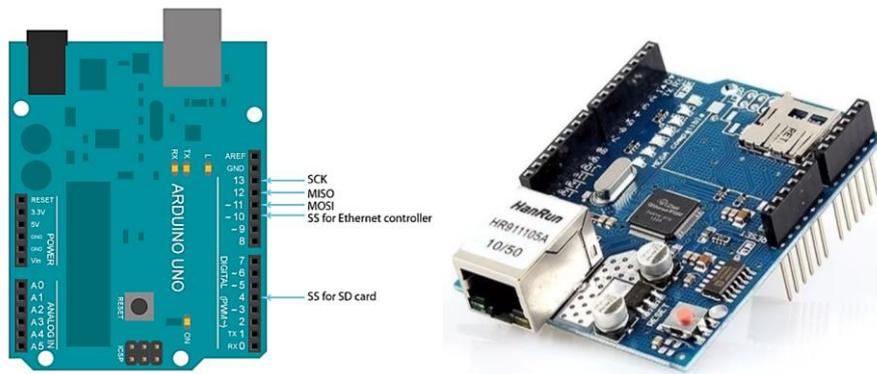


Figura 33. Arduino Ethernet Shield [52].

Tabla 18 Características técnicas de la tarjeta Ethernet [52].

Conector de la tarjeta	Ethernet estándar RJ45.
Pines digitales que usa	10, 11, 12, y 13 (SPI). Estos pines no pueden ser usados para e/s genéricas.
Botón de reset	Resetea la W5100 y la placa Arduino.
Opera	5 V suministrados desde la placa de Arduino.
Buffer interno	16 KB. No consume memoria.
Se comunica con el microcontrolador	Bus SPI. , para usarlo siempre debemos incluir la librería SPI.h
Tarjetas micro-SD	Puede ser usado para guardar ficheros y servirlos sobre la red. Para ello es necesaria la librería SD.

Para la utilización de la tarjeta de red Ethernet basta con acoplarla sobre el Arduino Mega. Para cargar los archivos de líneas de código (sketches) a la placa con el shield, se debe conectar al ordenador mediante el cable USB como se hace normalmente. Luego conectar la ethernet a un ordenador, a un switch o a un router utilizando un cable ethernet standard (CAT5 o CAT6 con conectores RJ45) [52].

5.2.3 Generador de energía HISA LT3990E

En el lugar a implementar el sistema de monitoreo, disponen de un generador eléctrico de combustión a Gasolina Marca HISE modelo LT3990E, que posee las siguientes especificaciones técnicas.



Figura 34. Imagen del Generador eléctrico HISA LT3990E.
Fuente: HISA

Tabla 19. Características Técnicas [53].

Generador	Bifásico
Transporte	De fácil transporte con Kit de 2 llantas.
Voltímetro	Sí.
Tipo de planta de energía	Gasolina.
Salida eléctrica	AC con voltaje 127 / 220 V.
Capacidad de combustible	15 litros.
Capacidad de aceite	0.6 litros.
Nivel de ruido	-66db
Potencia nominal	2500 W.
Potencia máxima	2800 W.
Motor	4 tiempos, Mono cilíndrico.
Sistema de arranque	Eléctrico y Retráctil
Horas de operación	13 horas al 50% de carga.
Dimensiones (l * w * h)	597 * 465 * 487 mm.
Peso	55 kg.
Batería	Si.

Para el proyecto los parámetros de interés del generador para el sistema de monitoreo son: el depósito de combustible, la capacidad de combustible de 80 litros (21,13 gal), el sistema de arranque eléctrico y las horas de operación: 17 horas.

5.2.4 Sensor nivel de combustible Marine Boat RV

Sensor de medida de nivel de combustible y agua para diversas aplicaciones como son los barcos, camiones o generadores de combustible. Su funcionamiento consiste en un cilindro flotante al fluido, la cual al variar la posición de la bolla tiene a efecto una variación en la resistencia del sensor que puede ser interpretado por un controlador como nivel de fluido [54].



Figura 35. Sensor de nivel de combustible [54].

Tabla 20. Especificaciones del sensor de combustible [54].

Medidor	Agua y combustible.
Voltaje	12-24 V.
Resistencia	240-33 Ω .
Material	Acero inoxidable SUS316.
Rango de longitud	9.843 in o 7.874 in.
Orificio de resolución	2.047 in.
Alambre	Material de PVC.
Montaje	Rosca BSP de 1 1/4 pulgadas o rosca NPT.
Accesorios:	Junta tórica.
Rango de precisión	0.394-0.827 in.
Temperatura de funcionamiento	-40.0 °F a 185.0 °F.

5.2.5 Relé Arduino

Módulo relé de 5 V para Arduino ARM PIC AVR MCU 5 V, luz indicadora LED, módulo de relé de 1 canal, funciona con placas Arduino oficiales marca Tolako [55].

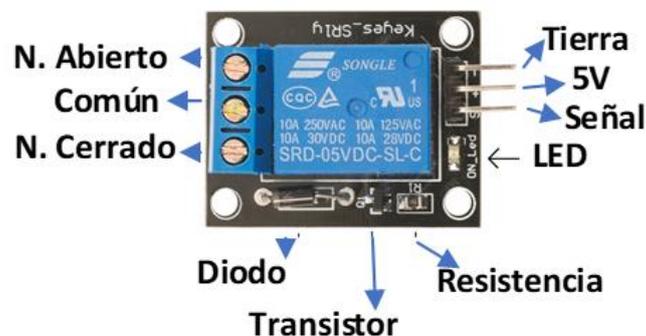


Figura 36. Relé Arduino de 1 canal [55].

Tabla 21. Especificaciones del Relé Arduino [55].

Utilización	Control de electrodomésticos, motores entre otros.
Voltaje de señal de control	5 V – 12 V TTL.
Voltaje de alimentación	5 V.
Control de carga de señales	10 A a 250 V AC.
Control de carga de señales	15 A a 125 V AC.
Contactos	Un contacto normalmente abierto y un contacto normalmente cerrado.

5.2.6 Módulo de fuente de alimentación CA, 110 V, 220 V a CC, 5 V

Es un módulo que transforma el voltaje alterno de 110 V o 220 V a un voltaje continuo de 5 V que puede ser leído por un controlador Arduino, posee protección de temperatura, protección contra sobre corriente y protección contra cortocircuitos [56].



Figura 37. Transformador reductor de AC a DC [56].

Tabla 22 Especificaciones del módulo Transformador [56].

Voltaje de alimentación	3,3 V – 5 VDC.
Voltaje alterno de entrada	CA 85 ~ 265 V 50/60 Hz o DC 100 ~ 370 V.
Corriente de entrada	0,0273 A (AC110V) 0,014 A (AC220V).
Voltaje de salida	DC 5 V (+ / -0,2 V).
Corriente de salida	700 mA.
Potencia	3,5 W.
Temperatura de funcionamiento	-20 ~ 60 °C.
Interruptor de sobretensión de la máquina	Máx 10%.
Tiempo de subida del voltaje de salida	Máx. 100 ms.
Protección contra sobrealimentación de salida	Sí.
Protección contra cortocircuitos	Sí.
Tamaño	(3 x 2 x 1.8) cm.

5.2.7 Sensor de nivel de energía

Para poder controlar el nivel de energía que posee un SAI, es necesario crear un circuito sencillo como es un divisor de tensión en el cual se adapte a las características de entrada de los pines analógicos del Arduino. De tal forma a la variación de tensión que

existirá al cargar o descargarse la batería se interpretará por el Arduino como un nivel de energía. En la Figura 38 se muestra un divisor de voltaje diseñado en Proteus con ayuda de un potenciómetro, donde el pin de resistencia variable se encuentra conectado a una entrada analógica del controlador.

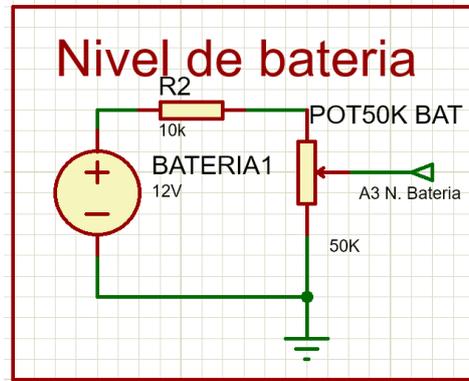


Figura 38. Divisor de voltaje para sensar nivel de batería con Arduino.
Fuente: Autor.

5.2.8 Reloj de tiempo real RTC 3231.

Dispositivo electrónico que permite obtener mediciones de tiempo en las unidades temporales. Normalmente están formados por un resonador de cristal integrado con la electrónica necesaria para contabilizar de forma correcta el paso del tiempo.

Los RTC son parecidos a los relojes y calendarios que usamos, la cual funcionan con segundos, minutos, horas, días, semanas, meses, años e incluso diferenciar los meses con diferentes días, o los años bisiestos. Además, incorpora una batería que permite mantener el valor del tiempo en caso de pérdida de alimentación. [57].

Tabla 23 Especificaciones del módulo Reloj de tiempo real RTC 3231 [57].

Voltaje de alimentación	2,3 V – 5,5 VDC.
Batería	CR2032 5 V.
Voltaje alterno de salida	Onda senoidal 5VAC máx.
Señal de salida	Analógica senoidal.
Dimensiones	5 cm x 2 cm x 2.4 cm.
Corriente nominal de entrada	2 mA.
Corriente nominal de salida	2 mA.

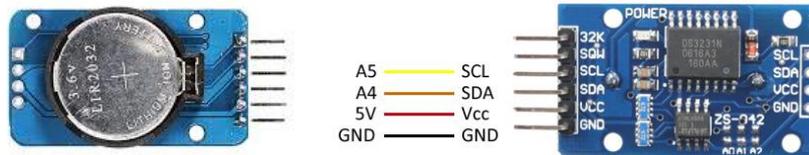


Figura 39. Reloj de Tiempo Real RTC 3231 [57].

5.3 Arquitectura del sistema

Previo a realizar la estructuración del sistema a crear, se hace un levantamiento de la infraestructura actual. En la Figura 40 se puede observar la distribución de la alimentación de los equipos en diversos tableros dependiendo la prioridad que estos posean. Así mismo se puede observar que cada tablero de distribución es alimentado por cierto equipo de suministro de energía ante una posible falta del suministro de la red pública.

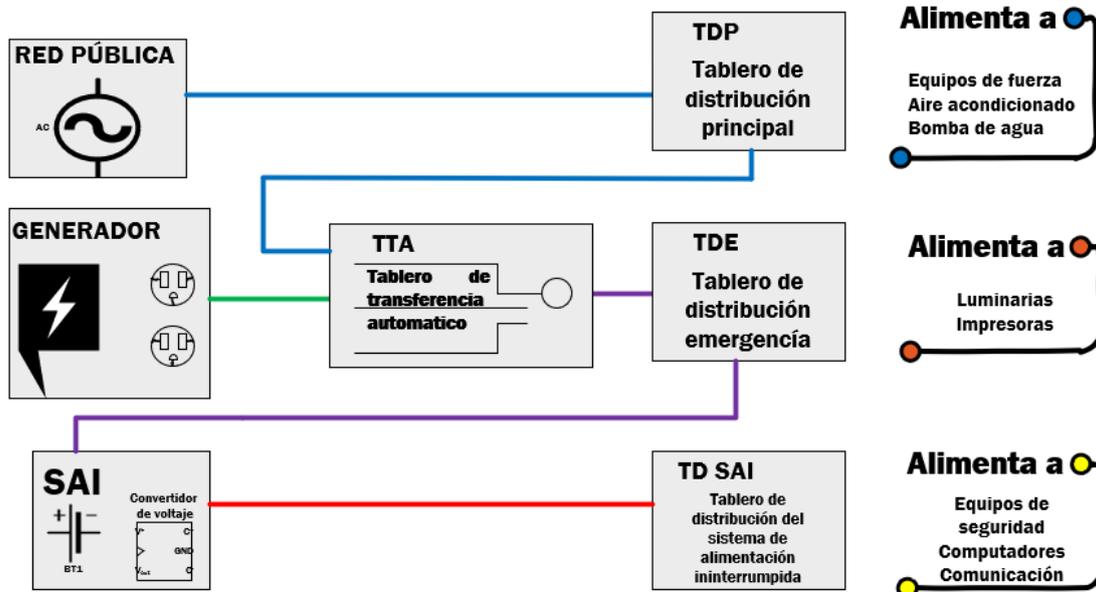


Figura 40. Diagrama isométrico de la Infraestructura actual de la distribución de energía eléctrica.

Fuente: Autor

A continuación, se puede observar en la Figura 41, el sistema de control de monitoreo que se va a implantar. Este está conformado por un conjunto de mecanismos e instrucciones que se irán describiendo a lo largo de este capítulo como son:

- **Arduino Mega:** es el micro controlador encargado de adquirir información de los sensores, almacenar, analizar y ejecutar las instrucciones configuradas previamente. Así mismo al Arduino se configura la programación adecuada para que sirva como un servidor web con una interfaz interactiva con el usuario

- **Sensor de combustible:** este sensor permite saber el nivel del combustible existente en el tanque del generador eléctrico.
- **Sensor de nivel de nivel de energía:** encargado de medir el nivel de energía que posee el banco de baterías del SAI.
- **Actuador:** sirve para realizar un cambio en el sistema tras cumplirse ciertas condiciones externas como en el caso nuestro, la falta de suministro de energía eléctrica de la red pública llevaría a que un relé o contactor a ejecutar un cambio para encender el generador de energía.
- **Fuente de alimentación:** al tratarse el proyecto de un mecanismo de automatización de fuente de energía de emergencia. La energía utilizada para el funcionamiento del sistema de monitoreo es suministrada por el SAI.
- **Acoplador de tensión:** es un modulo que permite reducir la tensión de 110 V 0 220 V AC a una tensión de 5 V DC.
- **RTC 3231:** es un Reloj de tiempo real que permite llevar un registro de fechas del funcionamiento del sistema.
- **Shield Ethernet:** tarjeta de red que permite a la placa de Arduino ATmega 2560 conectarse a internet y a la vez cuenta con una ranura micro SD que permite guardar el registro del funcionamiento del sistema.
- La Fuente de alimentación que se utiliza para energizar sistema de monitoreo y control del respaldo de la energía eléctrica es suministrada por el SAI.

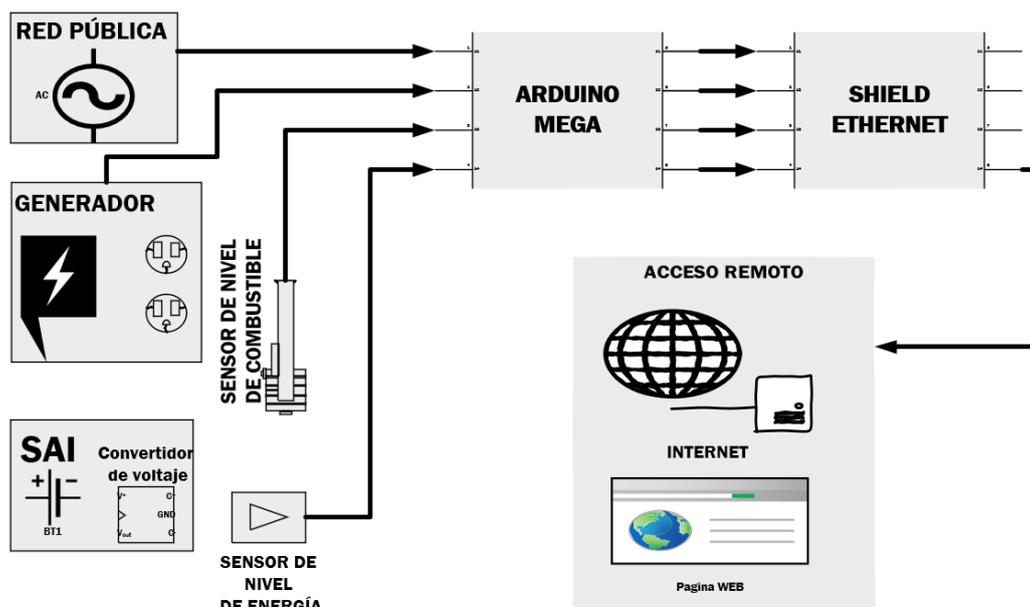


Figura 41. Diagrama isométrico del sistema de monitoreo y respaldo de energía eléctrica.

Fuente: Autor

Ya sabiendo el sistema de monitoreo a implementar, se procederá a crear un diagrama isométrico incorporando el sistema propuesto con la infraestructura actual quedando de tal manera como se indica en la Figura 42.

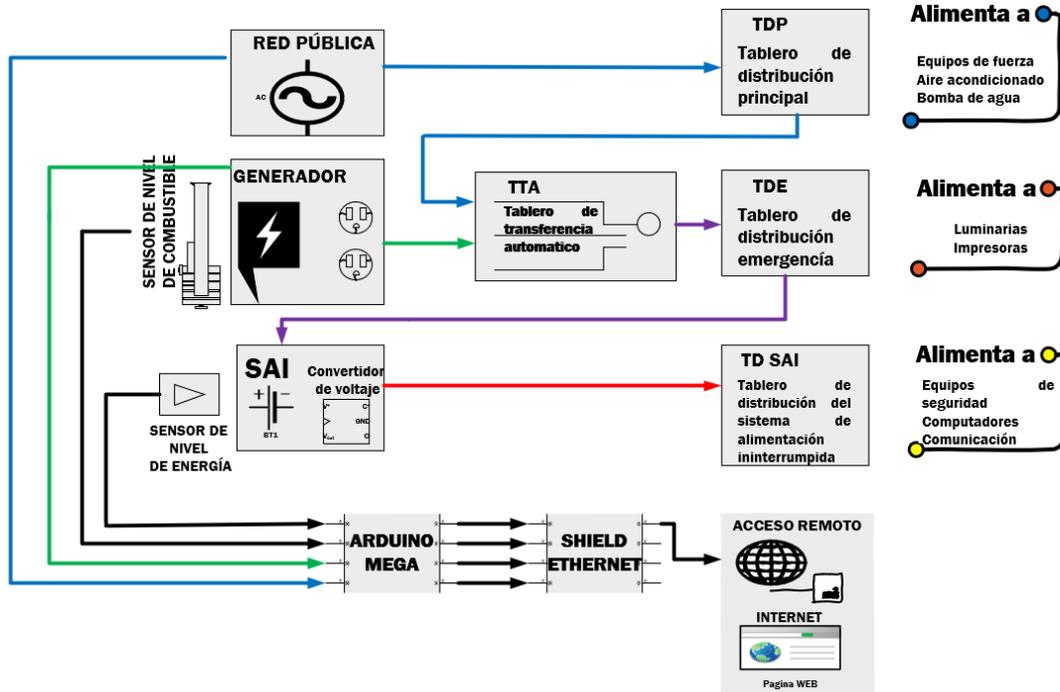


Figura 42. Diagrama isométrico de la distribución eléctrica y del sistema de monitoreo del lugar a implementar.

Fuente: Autor

5.3.1 Software Seleccionado

Arduino IDE

El software Arduino de código abierto (IDE) hace que sea fácil escribir código y cargarlo en su placa, se puede ejecutar en Windows, Mac OS X y Linux. El entorno está escrito en Java y se basa en el procesamiento y software de código abierto. Este software se puede usar con cualquier placa Arduino [58].

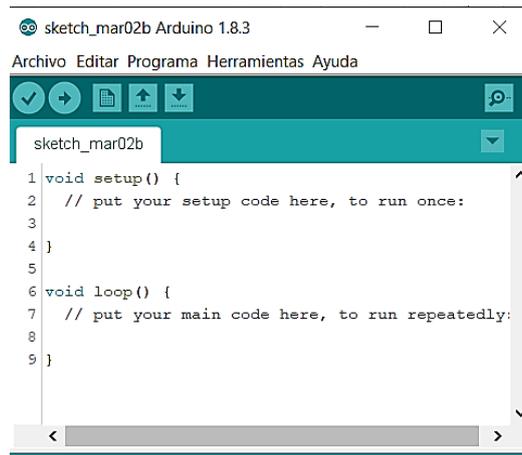


Figura 43. Interfaz del software IDE [58].

Geany

Editor de texto pequeño y ligero de interfaz gráfica de usuario (GUI) basado en Scintilla y bibliotecas gráficas GTK⁹, incluye características básicas de IDE (Entorno de desarrollo integrado) entre sus principales funciones esta: resaltado de sintaxis, plegado de código, nombre del símbolo autocompletado, cierre automático de etiquetas XML y HTML y consejos de llamadas. Además es compatible con muchos archivos incluyendo C, Java, PHP, HTML, Python, Perl, Pascal.

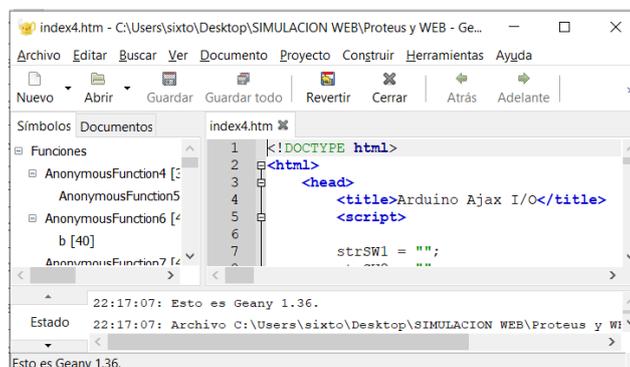


Figura 44. Interfaz de Geany [59].

Proteus

Es un programa que posee un conjunto de funciones para permitir el diseño, prueba y disposición rápida de placas de circuito impreso profesionales que tiene como funciones el diseño de PCB, simulación de circuitos y diseñador visual [60].

⁹ GTK: es una biblioteca de componentes gráficos multiplataforma para desarrollar interfaces gráficas de usuario (GUI).

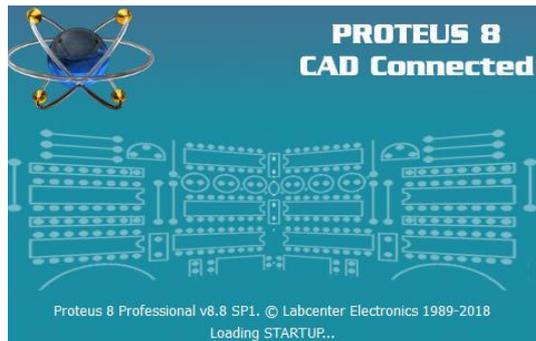


Figura 45. Software Proteus [60].

5.3.2 Diagrama de conexiones del sistema

En la Figura 46 se observa la manera de realizar las diversas conexiones. En el pin analógico A2 del Arduino MEGA se conecta el sensor de nivel de combustible. En el pin A3 se conecta el circuito que permite medir la variación de voltaje del SAI. Mientras que los pines digitales 10, 11, 12 y 13 son para la conexión SPI y SD card de la Shield ethernet.

El pin digital D9 sirve para el control de alarma en caso de algún tipo perjuicio del sistema, este pin también es posible maniobrarlo a través de una GUI en una plataforma WEB.

Así también en el pin digital D8 configurado como salida, se conecta un relé Arduino que permite controlar el encendido y apagado del generador. Este pin se activará o se desactivará dependiendo de la entrada del pin digital D3 que indica si existe suministro de energía en la red pública o no. Mientras que el pin digital D5 indica si el generador este encendido o no.

DIAGRAMA DE CONEXIONES DEL SISTEMA DE CONTROL Y MONITOREO

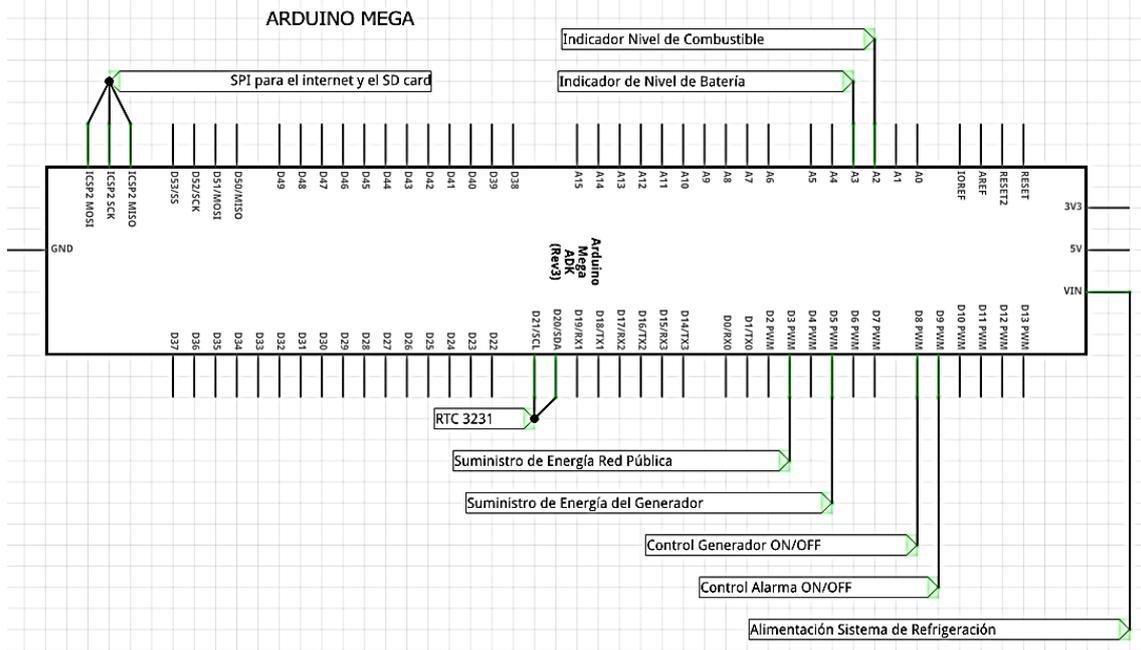


Figura 46. Diagrama de Conexiones del controlador ATMEGA 2560.

Fuente: Autor

6. RESULTADOS

Para el desarrollo del proyecto se ha planteado ofrecer dos plataformas para el monitoreo y control del sistema de respaldo de alimentación eléctrica. La primera plataforma permite ejecutar el proyecto con acceso vía red LAN y la segunda permite el acceso al sistema vía WAN, pero este tiene el inconveniente de que se requiere estar conectado a una plataforma IoT externa que requiere de una suscripción de pago mensual.

6.1 Plataforma de monitoreo y control del sistema con acceso vía LAN.

Para el desarrollo del proyecto se ha planteado desarrollarlo en dos partes principales que son el desarrollo del hardware y el desarrollo del software.

El desarrollo a nivel de hardware se ha realizado con ayuda del software Proteus tanto para el diseño del Schematic Capture como para el PCB (Placa de Circuito Impreso), mientras que para el desarrollo de las instrucciones de programación se usó el software de Arduino IDE. Para la visualización amigable de la información generada por los sensores se ha diseñado una página web basada en html con ayuda del software GEANY.

A continuación, se explica el funcionamiento a nivel de hardware y software de las partes del sistema de monitoreo y control de variables físicas de un generador eléctrico y sistema de backup de Alimentación.

6.1.1 Nivel de combustible

Tras haber determinado el sensor de nivel a utilizar en el apartado 5.2.4. se procede a crear un diseño en un circuito PCB (Placa de Circuito Impreso) en Proteus tal como se indica en la Figura 47. Debido a que las características del sensor es una resistencia variable de $240-33 \Omega$ a 12 V, es necesario conectar una resistencia para acoplar la tensión del sensor con el voltaje de entrada del Arduino. En la cual se puede observar que el diferencial de tensión del sensor de nivel es leído a través del puerto analógico A2 e interpretado como nivel de combustible mediante la lógica de programación.

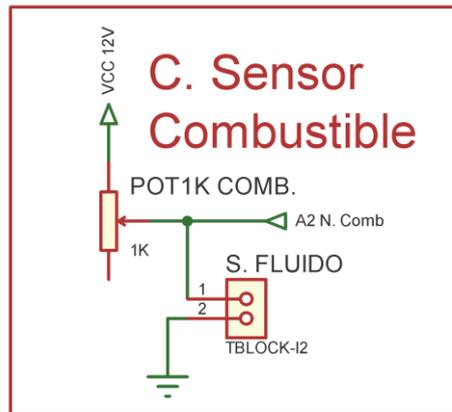


Figura 47. Diagrama en Proteus del circuito sensor de combustible.
Fuente: Autor

La parte de instrucciones de programación se lo realizo con ayuda de dos softwares como es el Arduino IDE y el software GEANY.

El software Arduino IDE se utilizó para configurar las instrucciones de la Figura 48 que se cargarán al Arduino Mega para la lectura e interpretación de la información obtenida de los sensores. En este caso, la variación del nivel de la bolla del sensor es interpretada por una variación de voltaje que se leerá en el pin análogo A2.

La lectura del pin análogo A2 es enviada a un archivo web html, que permite visualizar a través de un GUI la información adquirida por el sensor de nivel de combustible por parte del usuario, tal como se observa en la Figura 49.

```
// Lee el pin analogico A2
// Indicador del nivel de combustible
analog_val_2 += analogRead(2);
cl.print("<analog>");
cl.print(analog_val_2/10.23-23.07);
cl.println("</analog>");
```

Figura 48. Instrucciones de lectura del pin análogo A2 para analizar el nivel de combustible.
Fuente: Autor



Figura 49. GUI del monitoreo del nivel de combustible.
Fuente: Autor

6.1.2 Nivel de batería

Para el sensado del nivel de energía almacenado en el SAI es necesario adaptar el voltaje de 12 V a 5 V requeridos por el Arduino con un divisor de voltaje tal como se indica en la Figura 50. La variación de tensión generada al cargarse o descargarse la batería es detectada por el pin analógico A3.

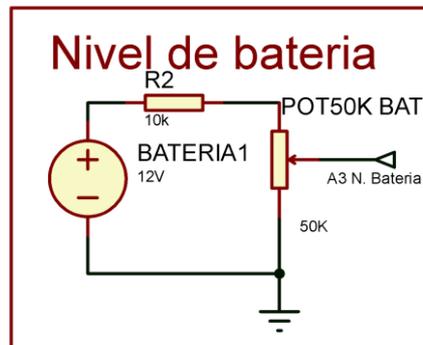


Figura 50. Divisor de voltaje para poder medir el voltaje de la batería con el Arduino.
Fuente: Autor

Para la lectura del pin analógico A3 es necesario configurar las instrucciones en el software IDE como indica la Figura 51. La información obtenida es enviada por la red para ser reflejada en un visualizador web como se muestra en la Figura 52.

```
//Lee el pin analogico A3
// Indicador de nivel de batería
analog_val_3 += analogRead(3);
cl.print("<analog>");
cl.print(analog_val_3/10.23);
cl.println("</analog>");
```

Figura 51. Instrucciones en el software IDE para la lectura del pin análogo A3 que es interpretado como nivel de energía del SAI.

Fuente: Autor



Figura 52. GUI del monitoreo del nivel de energía en el SAI.

Fuente: Autor

6.1.3 Funcionamiento de la red y del generador

Para analizar la fuente de suministro de energía eléctrica del sistema se requiere de un módulo que permite adquirir el voltaje de la red alterna de 110 V o 220 V y acoplarla a la tensión de voltaje de corriente continua del Arduino. Esto se consigue con transformadores de energía. En la Figura 53 se puede observar el transformador reductor que se utilizó en este proyecto, mientras que en la Figura 54 se observa el diseño de dos conectores que van a incorporarse en la PCB a implementar. El primer conector es para acoplar el sensor de tensión, para así determinar si se encuentra encendido o apagado el generador. El segundo conector sirve para analizar si existe suministro de energía de la red pública.



Figura 53. Transformador reductor de AC a DC [56].

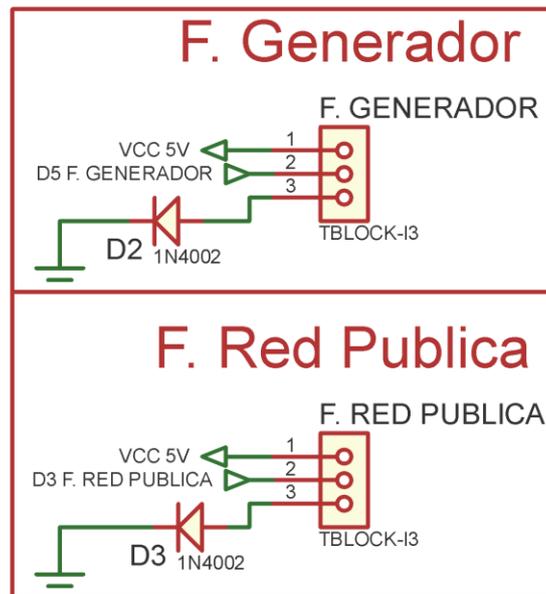


Figura 54. Circuito de Proteus para analizar el suministro de energía.

Fuente: Autor

6.1.4 Control de encendido y apagado

Para el control de encendido y apagado del generador es necesario colocar al circuito un actuador. Para ello se acopla a la PCB el relé Arduino, para que el contacto N.O. se cierre al momento de que el suministro de energía publica falle dando lugar a que el circuito del generador de energía entre en funcionamiento. Por lo cual, es de vital importancia la información adquirida del módulo de tensión analizado en el apartado anterior. En la Figura 56 se observa el diseño del conector en la PCB para implementar este actuador al circuito.

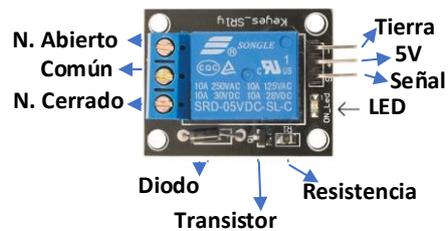


Figura 55. Relé Arduino de 1 canal [55].

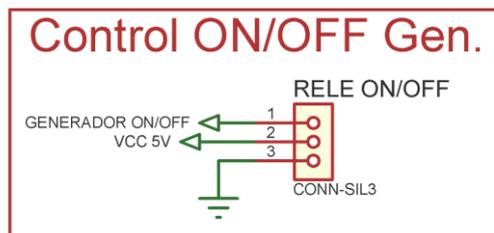


Figura 56. Diagrama en Proteus del circuito de control de arranque del generador.

Fuente: Autor

La lógica de programación en el software Arduino IDE se configurará de tal manera que:

- En caso de existir energía en la red pública el botón de control no podrá encender el generador.
- En caso de falla de suministro de energía de la red pública el programa hará que se encienda automáticamente el generador.
- En caso de falla de que el generador no se encienda automáticamente, se configurará un botón en la web que permite encenderlo manualmente.
- Toda la información generada en el Arduino es enviada a través de la red en paquetes como se indican en la Figura 57, Figura 58 y Figura 59.
- La información enviada del Arduino Mega a la web es descifrada por el archivo HTML e interpretado en un GUI como se indica en la Figura 60 y Figura 61 .

```

// Lee el pin digital D3
// analiza si existe energia red electrica
digital_val_3 = digitalRead(3);
cl.print("<switch>");
if (digital_val_3) {
    digitalWrite(8, LOW);
    cl.print("ENCENDIDO");
}
else {
    digitalWrite(8, HIGH);
    cl.print("APAGADO");
}
cl.println("</switch>");

```

Figura 57. Instrucciones en el software IDE para la lectura del pin digital D3, la cual analiza si existe voltaje en la red eléctrica.

Fuente: Autor

```

// Lee el pin digital D5
// analiza si existe energia generador
digital_val_5 = digitalRead(5);
cl.print("<switch>");
if (digital_val_5) {
    cl.print("ENCENDIDO");
}
else {
    cl.print("APAGADO");
}
cl.println("</switch>");

```

Figura 58. Instrucciones en el software IDE para la lectura del pin digital D5, la cual analiza si el generador está suministrando voltaje.

Fuente: Autor

```

// button de Control de estados
// Control del Generador
cl.print("<INTER>");
if (SW_state[0] || !digital_val_3) {
    cl.print("on");
}
else {
    cl.print("off");
}
cl.println("</INTER>");

```

Figura 59. Instrucciones para crear un botón de control de encendido / apagado del generador en la GUI de la web.

Fuente: Autor



Figura 60. GUI que demuestra de donde se está suministrando energía a la red.

Fuente: Autor

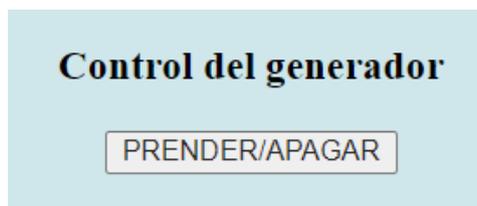


Figura 61. GUI del control de arranque.

Fuente: Autor

6.1.5 Sistema de advertencia

Para el sistema de alerta se deja a consideración del usuario ya que se podría colocar un sistema visual como un sistema sonoro.

Dependiendo de los estados del pin D3 y D5 del micro controlador Arduino ATMEGA 2560 se activará la salida del pin D9, la cual accionará un relé que cerrará el circuito indicador de advertencia. En la Figura 62 se puede observar el diagrama en Proteus que permitirá crear el circuito del PCB para poder adaptarlo al controlador.

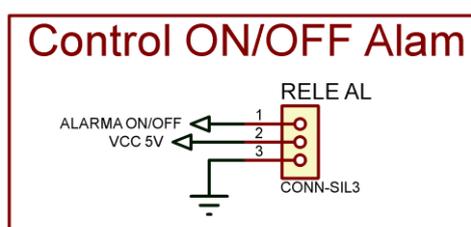


Figura 62. Diagrama en Proteus del circuito de control de arranque del generador.

Fuente: Autor

El sistema de advertencia está configurado en el software Arduino IDE, de tal manera, que al existir un nivel bajo de combustible o un nivel bajo de energía en el SAI, éste active el pin digital D9, dando a efecto la conmutación de un relé que hará que el circuito se cierre, encendiendo el sistema de alarma. También está configurado para que se pueda encender manualmente a través de la página web.

```

// Control de alarma
cl.print("<INTER>");
if (SW_state[1] || analog_val_2/10.23-23.07 <=40 || analog_val_3/10.23 <=40 ) {
    digitalWrite(9, HIGH);
    cl.print("on");
}
else {
    digitalWrite(9, LOW);
    cl.print("off");
}
cl.println("</INTER>");
cl.print("</inputs>");

```

Figura 63. Instrucciones para crear un botón de control de encendió / apagado del Sistema de alarma.

Fuente: Autor

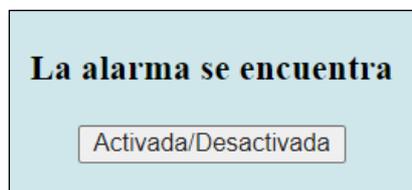


Figura 64. GUI del control de la alarma.

Fuente: Autor

6.1.6 Visualización y registro de datos en aplicación web

Para que el usuario pueda entender de una manera amigable la información que genera los sensores es necesario crear una GUI. En este caso se optó por generar una página web basado en HTML. Aquí se puede observar un indicador de nivel de combustible, un indicador de nivel del SAI, un indicador para saber si existe energía de la red pública o si el generador está suministrando energía. Así mismo posee botones para controlar el funcionamiento del generador como el sistema de alarma, como indica en la Figura 65.

A través de la aplicación web también se puede obtener un registro del funcionamiento del sistema, en el que se indica cuando el sistema haya tenido una alteración en su funcionamiento. Esto se lo realiza accediendo a la dirección 192.168.1.5:81, como se indica en la Figura 66. En la cual se puede observar fecha y hora del suceso, la fuente de alimentación suministrada, el nivel de combustible en generador, el nivel de energía que posee el SAI y la temperatura ambiental.

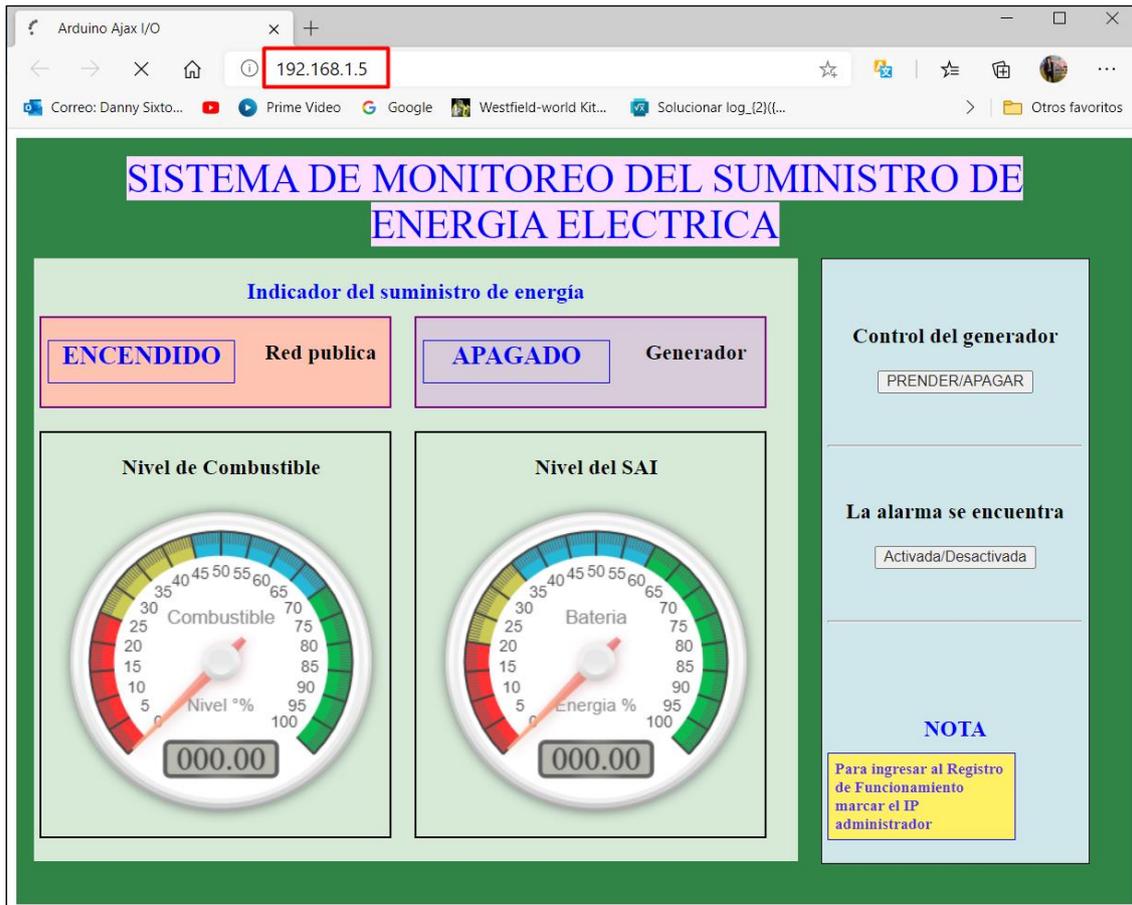


Figura 65. Interfaz gráfica del servidor WEB del monitoreo y control de las variables físicas de un generador de combustión.

Fuente: Autor

The screenshot shows a text-based log file with the following data:

Fecha y Hora	Estado	Nivel de combustible	Nivel de energía	Temperatura
Monday/2020/12/21 14:20:5	EQ. funcionando GENERADOR	66.00	91.20	19.25 C
Monday/2020/12/21 14:20:34	EQ. funcionando RED PÚBLICA	65.64	91.50	19.50 C
Monday/2020/12/21 14:20:34	EQ. funcionando GENERADOR	65.64	91.30	19.50 C
Monday/2020/12/21 14:20:35	EQ. funcionando RED PÚBLICA	66.36	90.71	19.50 C

Figura 66. Registro generado por variaciones del sistema de alimentación de energía eléctrica.

Fuente: Autor.

6.2 Plataforma de monitoreo y control del sistema a través de un servidor IoT el cual permite el acceso vía WAN .

Para la configuración de este sistema se requiere la programación del controlador ATmega 2560 con las librerías que establezcan conexión con la plataforma Ubidots.

6.2.1 Programación en el ATmega2560

En la primera parte de la programación es donde se cargan las librerías que permita utilizar los puertos SPI y establecer conexión con la plataforma remota de Ubidots. Así también se asigna un nombre de etiqueta de dispositivo, nombres de variables y el TOKEN de Ubidots.

```
#include <Ethernet.h> // Libreria para conexión a internet
#include <SPI.h> // Libreria para usar puerto SPI
#include <UbidotsEthernet.h> // Librerias de ubidots para establecer conexion con la
plataforma
/* Constantes y objetos */ /* Asigna los parámetros de Ubidots */
char const * TOKEN = "BBFF-NiwVjGad2Lfg5TRTK8mspBi770ctoZ"; // Asigna
tu TOKEN de Ubidots
char const * DEVICE_LABEL = "arduino-ethernet"; // Asignar la etiqueta de
dispositivo única
char const * VARIABLE_LABEL_1 = "Control"; // Asignar la etiqueta de variable
única para obtener el último valor
char const * VARIABLE_LABEL_2 = "N_combustible"; // Asignar la etiqueta de
variable única para enviar los datos
char const * VARIABLE_LABEL_3 = "N_bateria"; // Asignar la etiqueta de
variable única para enviar los datos
char const * VARIABLE_LABEL_4 = "S_EnergiaRed"; // Asignar la etiqueta de
variable única para enviar los datos
char const * VARIABLE_LABEL_5 = "S_Generador"; // Asignar la etiqueta de
variable única para enviar los datos
/* Ingrese una dirección MAC para su controlador a continuación */
/* Los protectores de Ethernet más nuevos tienen una dirección MAC impresa en
una etiqueta en el protector */
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

/* Configure la dirección IP estática que se utilizará si el DHCP no puede asignar */
IPAddress ip(192,168,0,5);
int arranque= LOW;
Ubidots client(TOKEN); /* inicializa la instancia */
```

Seguidamente se configura en la función void setup () para el acceso a internet , además se configuran los pines del Arduino a conveniencia como se muestra en el código.

```
/**Funciones principales***/
void setup() {
  Serial.begin(9600);
  //client.setDebug(true);// descomente esta línea para visualizar el mensaje de
  depuración
  /* iniciar la conexión Ethernet*/
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    /* Try to configure using IP address instead of DHCP */
    Ethernet.begin(mac, ip);
  }
  /* Asigna un tiempo a la tarjeta de red para que inicialice*/
  delay(1000);

  /****Se declaran los pines del controlador como entrada o salida***/
  pinMode(3, INPUT); // Analizar si existe suministro de energía de la red
  pinMode(5, INPUT); // Analizar si el generador este prendido
  /****Para saber el control de RELE ARDUINO:***/
  pinMode(8, OUTPUT); //Sirve para apagar el generador
  pinMode(9, OUTPUT); //Sirve para arrancar el generador
}
```

Finalmente en el Void loop se carga la información de la lectura de los sensores y controlador en diversas variables que van a ser enviadas he interpretadas por la plataforma de Ubidots y reflejadas en tableros gráficos. Así mismo dentro de esta función se dará las instrucciones de funcionamiento. Como por ejemplo cuando la lectura del pin digital (3) este en HIGH significa que existe suministro de energía de la red eléctrica por lo tanto el generador debe estar apagado por la que la salida del pin digital (8) se configura a LOW.

```
void loop() {
  /* Obtener el último valor de una variable */
  float value_1 = client.getValue(DEVICE_LABEL, VARIABLE_LABEL_1);
  /* Imprime el valor obtenido*/
  //Serial.print("the value received is: ");
  // Serial.println(value_1);
}
```

```

//Lógica de funcionamiento del Sistema
//CUANDO HAY SUMINISTRO DE LA RED PUBLICA
if (digitalRead(3)==HIGH||value_1==10) { // Si existe energia en red
pública entonces
    digitalWrite(8, HIGH); // Generador apágado los cables se conectan a
COM y NO al rele arduino
    digitalWrite(9, LOW); // Abre el circuito de arranque para que no se
encienda el generador
    arranque=LOW; //variable utilizada para ejecutar una solo una vez el
arranque
}
// cuando no hay suministro de energía publica
if (digitalRead(3)==LOW && value_1==50){ // Si no existe energia en red
pública y el control generador esta encendido
    digitalWrite(8, LOW); // Abre el interruptor pasando de OFF a ON
    if (arranque==LOW){
        digitalWrite(9, HIGH); // sirbe para da un pulso de arranque al
generador
        delay(2500); // determina la duración del pulso
        digitalWrite(9, LOW); // cierra el interruptor de arranque
        arranque=HIGH; // permite negar el valor de arranque con el
objetico que no se ejecute vuelta la condición
    }
}
Ethernet.maintain();
/* Se lee la informacion de los sensores y las acopla a las variables que se envian
a ubidots */
float value_2 = 15349.81/analogRead(A2)+analogRead(A2)/(-39.8392); //
Sensor Combustible
float value_3 = analogRead(A3)/10.23; //Sensor Bateria
boolean value_4 = digitalRead(3); // Analiza si hay suministro de energia de la
RED ELCECTRICA
boolean value_5 = digitalRead(5); // Analiza si hay suministro de energia del
GENERADOR
/* Envío de valores a Ubidots */
client.add(VARIABLE_LABEL_2, value_2);
client.add(VARIABLE_LABEL_3, value_3);
client.add(VARIABLE_LABEL_4, value_4);
client.add(VARIABLE_LABEL_5, value_5);
client.sendAll();
delay(1000);
}

```

6.2.2 Configuración en la plataforma Ubidots

Primeramente se ingresa a la cuenta Ubidots a través del link <https://industrial.ubidots.com/accounts/signin/> con el usuario y contraseña como se muestra en la Figura 67.

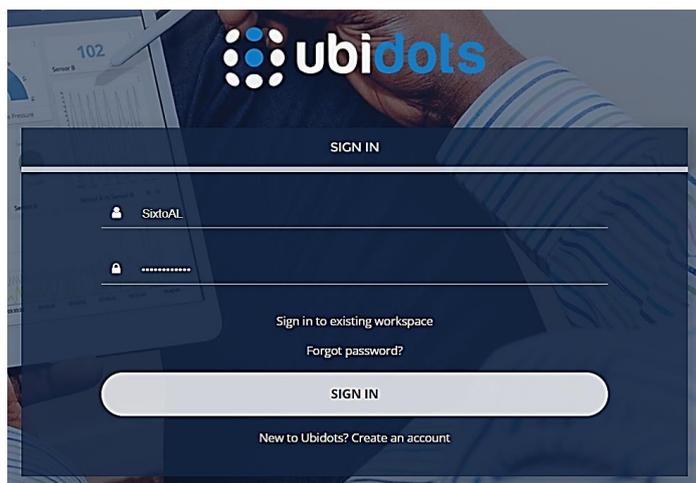


Figura 67. Ingreso a la plataforma de Ubidots.

Fuente: Autor

Seguidamente para establecer la conexión entre el hardware Arduino con la plataforma de Ubidots es necesario seleccionar el icono de <<usuario >>, <<Credenciales>>, la cual abrirá una ventana emergente, en la que se acude a la parte de <<Tokens>> y se selecciona el icono de copiar tal como se indica en la Figura 68 y Figura 69.

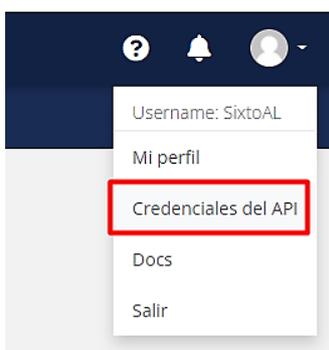


Figura 68. Configuración de establecimiento de la conexión del controlador con la plataforma Ubidots.

Fuente: Autor

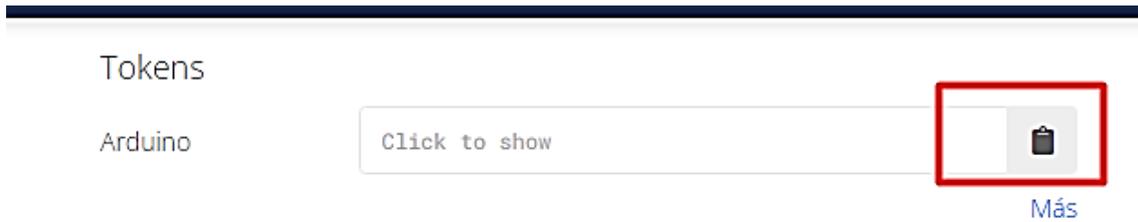


Figura 69. Configuración de establecimiento de la conexión del controlador con la plataforma Ubidots.

Fuente: Autor

Con el código copiado se procede a pegarlo en las instrucciones del Arduino IDE en la parte (`char const * TOKEN = "" PEGAR CODIGO AQUI`), tal como se puede observar en la Figura 70.

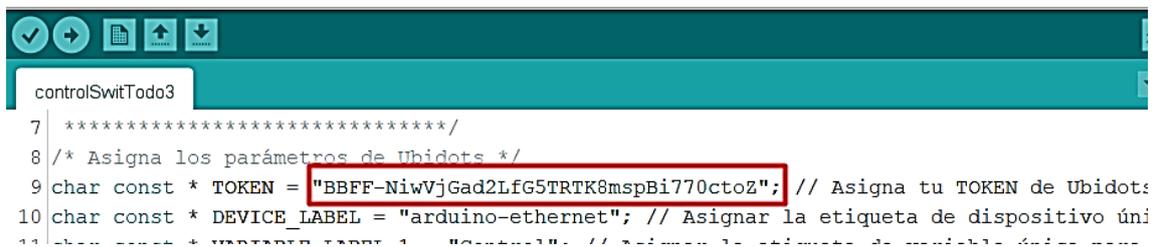


Figura 70. Configuración de establecimiento de la conexión del controlador con la plataforma Ubidots.

Fuente: Autor

Una vez terminada la programación del Arduino se procede a cargar el código al hardware y posteriormente realizar todas las conexiones del sistema. Luego se habla Ubidots para verificar el establecimiento de la conexión con la plataforma.

Si la conexión se realizó exitosamente en la parte de Dispositivos debe aparecer el icono de la Figura 72.



Figura 71. Verificación del establecimiento de la conexión del controlador con la plataforma Ubidots.

Fuente: Autor

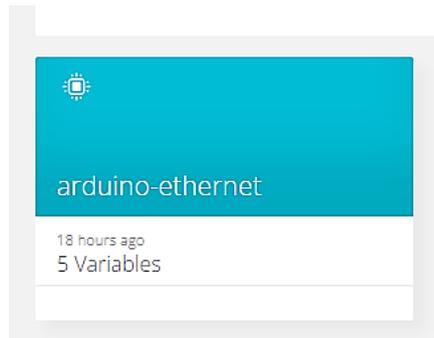


Figura 72. Verificación del establecimiento de la conexión del controlador con la plataforma Ubidots.

Fuente: Autor

Ya comprobado la creación del dispositivo se procede a crear los indicadores para crear la GUI por la que se procede a seleccionar en <<Datos >>, <<Tableros>>, seguidamente se selecciona en << + >>, luego se busca el indicador a conveniencia, en este caso es el <<TAQUE>>.



Figura 73. Configuración de un indicador de variable Indicador combustible

Fuente: Autor

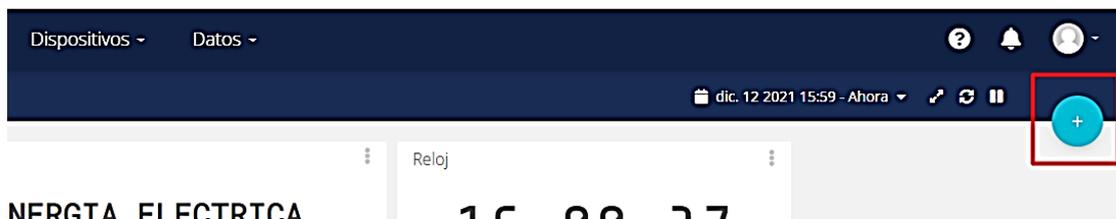


Figura 74 . Configuración de un indicador de variable Indicador combustible

Fuente: Autor

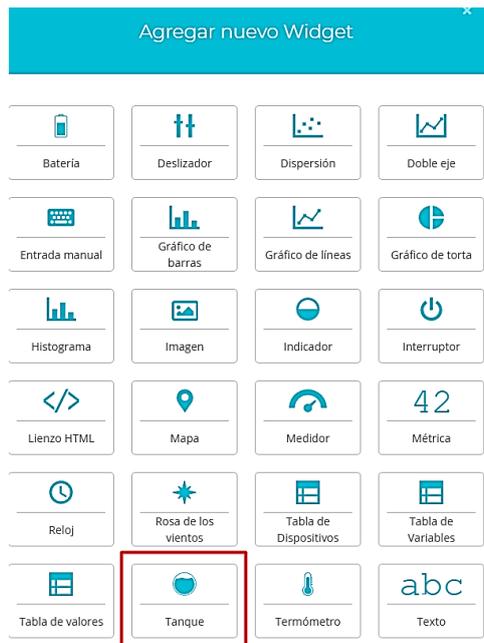


Figura 75. Configuración de un indicador de variable Indicador combustible
Fuente: Autor

Ya seleccionado el icono del <<TANQUE>> nos aparecerá una pantalla que pedirá que se agregue una variable al indicador. Para este proyecto se tienen cinco variables por la que hace falta configurar un indicador para cada variable. El indicador de Tanque se lo va usar para determinar el nivel de combustible en el depósito del generador por la que se lo enlaza con la variable <<N_Combustible>>. El proceso de enlace de variable con indicador se lo debe repetir con cada variable.



Figura 76. Configuración de un indicador de variable Indicador combustible
Fuente: Autor

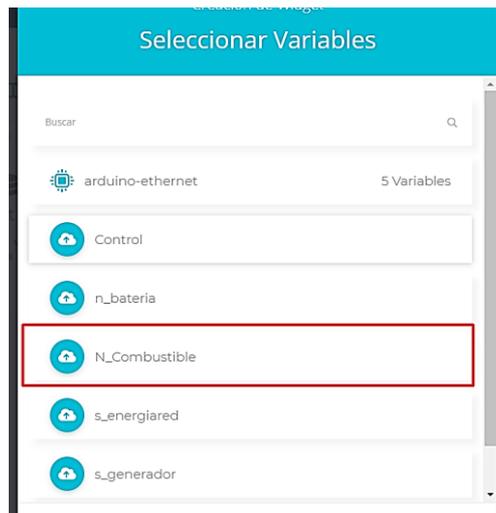


Figura 77 . Configuración de un indicador de variable N_ combustible
Fuente: Autor

Una vez realizado este procedo con cada variable, la GUI de la plataforma quedara tal como se observa en la Figura 78.

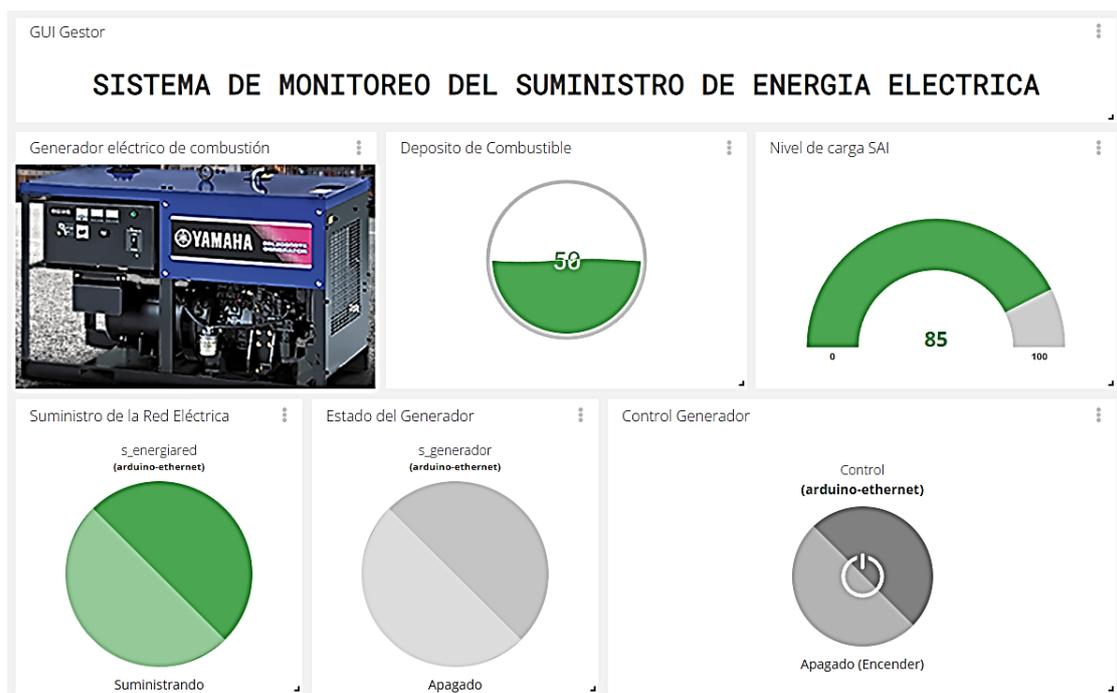


Figura 78. GUI del sistema de monitoreo del suministro de energía de energía eléctrica.
Fuente: Autor

6.2.3 Configuración de eventos en Ubidots.

La configuración de eventos de Ubidots es una función de la plataforma que nos permite configurar avisos mediante correos electrónicos o mensajes SMS, para así determinar sucesos del sistema. En este proyecto se configurará un evento que indique el

momento en que se fue la energía eléctrica, por la que en el caso de que falle el suministro eléctrico se enviara un correo electrónico y un SMS para avisar del suceso. En las siguientes imágenes se puede observar el proceso de configuración de este evento.

En el primer paso se selecciona en <<Datos>> posteriormente en <<Eventos>>. Seguidamente se abrirá un icono que indicará <<Crear un evento>>.



Figura 79. Proceso de configuración de eventos en Ubidots
Fuente: Autor

Una vez seleccionado el icono Crear un evento se procederá a configurar <<Si condición(es)>> y <<Entonces acciones>> como se indica en la **Figura 80** y **Figura 81.**



Figura 80. Proceso de configuración de <<Si condición(es)>> en Ubidots
Fuente: Autor



Figura 81. Proceso de configuración de <<Entonces acciones>> en Ubidots
Fuente: Autor

En el icono de <<Entonces acciones>> se configura dos secciones <<EN LA ACTIVACIÓN>> y <<VUELTA A LA NORMALIDAD>>, tal como se muestra en la Figura 82 y Figura 83.

Si condición(es)

Dirección de correo electrónico
sixtodanny@gmail.com

Agregue correos electrónicos separados por comas

Entonces acciones

EN LA ACTIVACIÓN VUELTA A LA NORMALIDAD

Asunto
[Nombre de la variable] [Marca de tiempo de activación] alerta

Mensaje
Hola, [Nombre de la variable] fue [Valor de activación] a las [Marca de tiempo de activación].

Repetir acción cada 60 Minutos(s) hasta 10 veces

Figura 82. Proceso de configuración de <<EN LA ACTIVACIÓN>> en Ubidots
Fuente: Autor

Si condición(es)

Dirección de correo electrónico
sixtodanny@gmail.com

Agregue correos electrónicos separados por comas

Entonces acciones

VUELTA A LA NORMALIDAD

Activa esta acción cuando la condición del evento vuelva a su estado normal

Asunto
[Marca de tiempo de activación] [Variable]

Mensaje
[Marca de tiempo del último valor] [Nombre de la variable]

Figura 83. Proceso de configuración de <<VUELTA A LA NORMALIDAD>> en Ubidots
Fuente: Autor

7. DISCUSIÓN

Tras haber analizado, diseñado, probado y elaborado el servidor web y el datalogger que permite monitorizar y registrar las variables físicas de un generador eléctrico de combustión, se determinó que es posible desarrollar equipos y sistemas de bajo costo orientado al internet de las cosas, ejemplo de ello es la plataforma de Arduino, cuyas características permitieron cumplir los requerimientos planteados en este proyecto, diferenciando con los PLC, cuya implementación tendría un alto costo.

El estudio y análisis de las tecnologías orientadas a la automatización de sistemas de control permitió establecer una tabla comparativa respecto al costo / beneficio, logrando determinar entre otras, que la opción más económica y fiable, sin dejar de lado su funcionabilidad y escalabilidad, es la tecnología de open source ARDUINO, cuyas características permiten implementar el sistema.

Seguidamente, se realizó un análisis para determinar el modelo de Arduino a utilizar que cumpla con los requerimientos como memoria, puertos analógicos y digitales, controladores Ethernet, voltaje de entrada y costo de adquisición. Cabe indicar que a primera instancia se optó por el microcontrolador ATMEGA 328P, cuyas características cumplen con los puertos necesarios para la conexión de los diversos sensores y actuadores del sistema implementado. Sin embargo surgió el inconveniente de que su memoria no fue suficiente para albergar el servidor web, el registro de datos y las instrucciones del sistema, por lo que se optó por un módulo de mayor memoria; en este caso el microcontrolador ATMEGA 2560 que cumple con todos los requerimientos para el sistema implementado.

Posteriormente, se estableció el tipo de sensor a utilizar para medir la variación del nivel de fluido que existe en el depósito del generador, esto se lo realizó a través del Sensor nivel de combustible Marine Boat RV, cuyo funcionamiento consiste en el movimiento vertical de una boya flotante que permite medir la variación del nivel de combustible. Este sensor es capaz de variar su resistencia, que al acoplarla a un divisor de tensión, le permite al controlador interpretar esas variaciones como nivel de combustible disponible en el depósito.

Por otra parte, para medir la cantidad de energía que dispone el SAI, se optó por acoplar al circuito un divisor de tensión paralelo a este sistema, reduciendo su voltaje y

permitiendo acoplarlo al controlador, cuya interpretación se presenta como porcentaje de energía en la interfaz gráfica.

Asimismo, para adquirir la información del suministro de energía eléctrica fue necesario acoplar módulos transformadores de voltaje al circuito, que ajustan la tensión de 110 V a 5 V que permite el puerto digital del Arduino. En este caso, la lógica de programación que se siguió está diseñada de tal forma que, cuando falle el suministro de energía eléctrica pública, automáticamente encienda el sistema de respaldo eléctrico. Para ejecutar este cambio fue necesario implementar un actuador que permite controlar de manera automatizada o manual el encendido o apagado del generador eléctrico. De igual forma, se implementó otro actuador que permite activar una alerta visual en caso de que exista un percance en el sistema.

Por otro lado, para crear un registro que indique el momento en el que el suministro de energía eléctrica pública ha fallado y ha entrado en funcionamiento el generador eléctrico, se ha incorporado al sistema un controlador de tiempo real modelo RTC3231, cuya función es llevar la base de registro y a través del controlador poder guardar esta información en una tarjeta de memoria micro SD.

Para que la información generada por los sensores pueda ser interpretada de manera amigable por el usuario, se creó una interfaz gráfica de usuario (GUI), cuyo acceso se lo realiza a través de una página web. Para la elaboración de esta interfaz, se planteó dos maneras de realizarlo. La primera interfaz es necesario configurar al ATMEGA 2560 como servidor web que permite visualizar y adquirir la información del sistema a través de una red LAN. La segunda interfaz requiere establecer conexión con una plataforma externa orientada al IoT como es Ubidots por la cual se tendrá acceso al sistema a través de una red WAN.

Finalmente, para acoplar el hardware del sistema se elaboró la placa de circuito impresa PCB, en el software de Proteus. Para el desarrollo e interpretación del software del sistema implementado se utilizó el programa Arduino IDE y Geany. La implementación del software y hardware resultó en el correcto funcionamiento del sistema.

8. CONCLUSIONES

- En la presente tesis se diseñó e implementó un sistema de monitoreo y control de variables físicas como el nivel de combustible y nivel de energía de un generador eléctrico y sistema de alimentación de respaldo, para visualizarlas a través de una página web.
- En el análisis de la revisión literaria se determinó la mejor opción a utilizar de hardware de control mediante la comparación de especificaciones técnicas de los diferentes controladores disponibles en el mercado local haciendo una relación del costo/beneficio como se describe en la Tabla 12. Esto nos lleva a la conclusión de que el microcontrolador Arduino MEGA 2560 es el más óptimo para el desarrollo de este sistema, debido a que es una placa multifuncional con diversas entradas y salidas, con facilidad de programación y escalable.
- Se analizó que el sensor de nivel Marine Boat RV permite adquirir el nivel de combustible que se encuentra en el depósito del generador, de manera eficaz y con error despreciable, de tal forma que la adquisición de este parámetro sea fácil de interpretar y requiera menos cálculos para su visualización en la interfaz gráfica.
- Se identificó que el módulo RTC3231 es un controlador de tiempo real bastante preciso que permitió llevar un registro del momento de encendido/apagado del generador de energía eléctrica y del momento de retorno de energía de la red eléctrica pública.
- Se determinó que el microcontrolador ATMEGA 2560 tiene suficiente capacidad de memoria para configurar las instrucciones necesarias. Además, otorga la facilidad de escalabilidad para trabajos futuros que permitan medir otros parámetros de interés que existen en un sistema de respaldo eléctrico como es la presión de aceite y las revoluciones por minuto del motor generador.

- Mediante el software Proteus, se diseñó el circuito del sistema de monitoreo y control de variables físicas del generador eléctrico y sistema de respaldo de alimentación, implementando todos los módulos, sensores y actuadores que controlan y monitorizan el sistema de respaldo de energía eléctrica.
- Se diseñó dos interfaces de monitoreo y control para el sistema de respaldo eléctrico. En la primera se configuró un servidor web en el controlador Arduino Mega, con la cual se tiene acceso a la GUI a través de una red LAN. En la segunda GUI se utilizó una plataforma de IoT que ofrece mayor funcionalidad, la cual permitió cumplir con todos los requisitos solicitados en los objetivos de este proyecto.
- Actualmente nos encontramos en un mundo que se dirige hacia la automatización de los sistemas que nos rodean, por la cual es necesario mantenerse a la vanguardia de las plataformas del internet de las cosas y la industria 4.0.

9. RECOMENDACIONES

En base a los planteamientos y observaciones que permiten la culminación de este proyecto, se recomienda:

- Realizar una investigación previa de sistemas de automatización orientada al IoT que se pretende realizar, organizar las ideas en diagramas de donde se pretende comenzar y a donde se pretende llegar. Posteriormente comenzar a recolectar el estado del arte con una amplia revisión bibliográfica la cual facilitará los procesos y acortará el tiempo de trabajo.
- Para la compra e instalación de los componentes eléctricos y electrónicos, se recomienda, que los equipos a adquirir cumplan con las especificaciones requeridas, así como realizar una correcta colocación e instalación de cada uno de estos. Además, tener siempre en cuenta una posible escalabilidad del sistema.
- Se recomienda comprobar que los elementos usados para hacer este sistema funcionen correctamente, de esta manera, las pruebas futuras que se realicen podrán arrojar resultados veraces, que demuestren sin interferencias el comportamiento real del sistema.
- Previamente a elaborar la placa PCB del sistema, se recomienda cerciorarse del correcto funcionamiento del circuito a implementar a través de un banco de pruebas en una Protoboard, teniendo siempre el respectivo cuidado al momento de manipular los equipos, ya que pueden ser sensibles a la energía cinética.
- Para realizar la plataforma de monitoreo y control del sistema con acceso vía LAN se recomienda adquirir conocimiento en el lenguaje de escritura HTML, para poder así crear un diseño web eficiente y amigable con el usuario.
- Para la plataforma de monitoreo y control del sistema a través de un servidor IoT se recomienda analizar que estas plataformas ofrezcan compatibilidad con los equipos que se está utilizando, además que existan las librerías necesarias para poder implementar el sistema.

- Se recomienda tener cuidado al momento de la programación del ATMEGA 2560 ya que la utilización incorrecta de puertos genera problemas de interpretación en el microcontrolador. Es el caso que si se utiliza los puertos SPI para la lectura y escritura de la tarjeta SD y la conexión a Ethernet, estas funciones no trabajan simultáneamente por lo que se requiere de librerías adicionales para trabajar en diferentes intervalos.
- Para completar el desarrollo de este proyecto, se requiere de la inclusión de una plataforma IoT en la red, de tal forma que la visualización de datos y las acciones de control se puedan realizar a través de Internet desde cualquier parte del mundo.
- Importante no compartir las credenciales de ingreso a la plataforma de monitoreo y control del sistema eléctrico debido a que la plataforma no posee una cantidad máxima de usuarios que pueden ingresar a la vez al sistema.
- Para mejora futura del proyecto se recomienda implementar mayor cantidad de sensores que permitan monitorear: las revoluciones por minuto del motor generador, el voltaje de línea que emite el generador, la corriente de línea y la presión de aceite.

10. REFERENCIAS

- [1] R. VIVERO, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO PARA UN GENERADOR EÉCTRICO EN EL TERMINAL PORTUARIO DE GUAYAQUIL UTILIZANDO EL SOFTWARE LABVIEW, Guayaquil: UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL, 2016.
- [2] D. A. Baculima, Diseño de un sistema de monitoreo y alarma para el sistema de respaldo eléctrico de la empresa Puntonet en la ciudad de Cuenca aplicado al nodo ubicado en Barabon, Cuenca: Universidad Politécnica Saleciana Sede Cuenca, 2012.
- [3] M. Contreras, Desarrollo e implementación de un sistema de medición, monitoreo y control de carga eléctrica para aplicaciones domésticas, Cuenca: Universidad Politecnica Saleciana Sede Cuenca, 2015.
- [4] H. Pedro, N. Baena y F. Lopez, Integración de un control automático de conmutación eléctrica., Juárez: Universidad Autónoma de Ciudad Juárez, 2012.
- [5] A. Hermosa, «Generadores mecánicos,» de *Principios de Electricidad y Electronica II*, Barcelona, MARCOMBO, 1999, p. 2.
- [6] GRUPEL, «Diferentes tipos de generadores de corriente eléctrica,» [En línea]. Available: <https://grupel.eu/es/grupel-es/tipos-generadores-de-corriente-electrica/>. [Último acceso: 5 Noviembre 2019].
- [7] «DE MAQUINAS Y HERRAMIENTAS,» 23 Diciembre 2015. [En línea]. Available: <https://www.demaquinasyherramientas.com/maquinas/generadores-o-grupos-electrogenos-tipos>. [Último acceso: 7 Noviembre 2019].
- [8] G. Enríquez, «El Generador Electrico,» de *El Libro Practico De Los Generadores, Transformadores Y Motores Electricos*, Mexico, LIMUSA, 2004, pp. 3-5.
- [9] R. Pallás, SENSORES Y ACONDICIONADORES DE SEÑAL, Barcelona: MARCOMBO, 2003.

- [10] E. Viola, *La calidad de una obra*, Buenos Aires: Nobuku, 2007.
- [11] G. Pacio, *Data Centers hoy*, Buenos Aires: Alfaomega, 2014.
- [12] «SAI, Sistema de alimentación ininterrumpida,» LEGRAND, Madrid, 2013.
- [13] A. S. José, «Sistemas de corriente continua,» de *Instrumentación y control básico de procesos*, Madrid, Ediciones Díaz de Santos, 2013, pp. 119-121.
- [14] R. Pallas, «Sensores y Actuadores,» de *Adquisición y distribución de señales*, Barcelona, marcombo, 2005, pp. 27-84.
- [15] L. Corona Ramírez, S. Abarca Jiménez y J. Mares Carreño, «Actuadores,» de *Sensores y actuadores*, Mexico, PATRIA, S.A, 2014, pp. 16-35.
- [16] M. Juan Carlos, «Actuadores y sus aplicaciones (Inslaciones domóticas): Ciclos Formativos,» de *Instalaciones domóticas*, 2011, pp. 61-62.
- [17] F. Cembranos Nistal, «Relés electromagnéticos,» de *Automatismos eléctricos, neumáticos e hidráulicos*, Paraninfo, 2008, p. 35.
- [18] R. Pallás Areny, «Relés,» de *Adquisición y distribución de señales*, Barcelona, Marcombo, 1993, pp. 36-37.
- [19] M. Rouse, «TechTarget,» WhatIs.com, Enero 2014. [En línea]. Available: <https://searchdatacenter.techtarget.com/es/definicion/Monitoreo-del-rendimiento-de-aplicaciones-APM>. [Último acceso: 30 Noviembre 2019].
- [20] L. Guarín, L. Morantes Guzmán y E. Delgado Trejos, *Transferencia óptima de datos para el monitoreo y control remoto de sistemas en tiempo real*, Medillín: ITM, 2015.
- [21] American Water Works Association, de *MAnual de entrenamiento para operadores de sistemas de distribución de agua*, Estados Unidos de América, Carol Stearns, 2012, p. 279.

- [22] SUHISSA, «SHISSA,» SuHISSA, 2017. [En línea]. Available: <https://suhissa.com.mx/sistemas-de-monitoreo/>. [Último acceso: 11 Noviembre 2019].
- [23] «Hogar,» PROJECT Abstracts, [En línea]. Available: <https://projectabstracts.com/21102/an-indoor-monitoring-system-for-ambient-assisted-living-based-on-internet-of-things-architecture-project.html>. [Último acceso: 3 Enero 2020].
- [24] A. Ramos Martín y R. M. Maria, «Instalación de aplicaciones de ofimática WEB,» de *Aplicaciones Web*, Madrid, Paraninfo, 2014, p. 300.
- [25] C. Krall, «aprendeaprogramar.com,» apr, [En línea]. Available: https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=707:ies-css-un-lenguaje-de-programacion-ipara-que-sirve-diferencias-entre-html-css-php-asp-cu01004d&catid=75&Itemid=203. [Último acceso: 11 Diciembre 2021].
- [26] MDN Web Docs, «Conceptos básicos de HTML,» Moz://a, 10 Diciembre 2020. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics#anatom%C3%ADa_de_un_documento_html. [Último acceso: 18 Enero 2021].
- [27] L. González, «dinahosting,» NEWSLETTER, 16 Julio 2020. [En línea]. Available: <https://dinahosting.com/blog/los-10-lenguajes-de-programacion-mas-usados/>. [Último acceso: 11 Diciembre 2021].
- [28] R. Concepción, «rjconcepcion,» 16 Julio 2020. [En línea]. Available: <https://www.rjconcepcion.com/podcast/que-lenguaje-de-programacion-usa-arduino/>. [Último acceso: 13 Diciembre 2021].

- [29] kiconex, «¿QUÉ ES Y PARA QUÉ SIRVE UNA PLATAFORMA IOT?,» kiconex, 7 Mayo 2019. [En línea]. Available: <https://www.kiconex.com/que-es-y-para-que-sirve-una-plataforma-iot/>. [Último acceso: 13 Diciembre 2021].
- [30] Gravatar, «Plataformas IoT,» Aprendiendo Arduino, 31 Octubre 2019. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2017/03/31/plataformas-iot/>. [Último acceso: 13 Diciembre 2021].
- [31] Amazon Web Services, «AWS IoT Core,» AWS, 2022. [En línea]. Available: https://docs.aws.amazon.com/es_es/iot/latest/developerguide/aws-iot-how-it-works.html. [Último acceso: 11 Enero 2022].
- [32] Business Publications Spain S.L., «Redes & Telecom,» 28 Septiembre 2017. [En línea]. Available: <https://www.redestelecom.es/infraestructuras/noticias/1101052001803/ya-disponible-google-cloud-iot-core-conectar-dispositivos-iot.1.html>. [Último acceso: 12 Enero 2022].
- [33] Microsoft Azure, «Azure IoT Central,» [En línea]. Available: <https://azure.microsoft.com/es-mx/services/iot-central/#features>. [Último acceso: 12 Enero 2022].
- [34] MathWorks, «información sobre ThingSpeak,» [En línea]. Available: https://thingspeak.com/pages/learn_more. [Último acceso: 12 Enero 2022].
- [35] GitBook, «Documentacion de Thinger.io,» [En línea]. Available: <https://docs.thinger.io/>. [Último acceso: 12 Enero 2022].
- [36] Ubidots, «Ubidots,» Ubidots, [En línea]. Available: <https://ubidots.com/>. [Último acceso: 20 Octubre 2020].
- [37] Domotizados, «DOMOTIZADOS,» Media Incubator, 12 Enero 2018. [En línea]. Available: <https://domotizados.co/conoce-los-tipos-de-instalaciones-domoticas-y-sus-componentes/>. [Último acceso: 28 Noviembre 2019].

- [38] M. Álvarez Pulido, *Controladores lógicos*, Barcelona (España): Marcombo, 2007.
- [39] V. Rossano, de *Electrónica & microcontroladores PIC*, USERSHOP, 2009, pp. 18-20.
- [40] M. Morandi, «¿Qué es un PIC?,» de *Arquitectura de los PICs*, Argentina, Quark SRL, 2002, pp. 1-6.
- [41] J. Álvarez Salazar y J. G. Mejía, «Control lógico y secuencial,» de *TIA Portal Aplicaciones de PLC*, Medillín, Instituto Tecnológico Metropolitano, 2017, pp. 11-12.
- [42] J. Reyes, «CCRMA,» Stanford University, 18 Agosto 2018. [En línea]. Available: https://www.maginvent.org/articles/pidht/pidtoot/Ejemplos_Microcontroladores.html. [Último acceso: 28 Noviembre 2019].
- [43] Y. FM, «Xataka Basics,» WEBEDIA, 3 Agosto 2018. [En línea]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. [Último acceso: 28 Noviembre 2019].
- [44] E. M. Pérez, «Fundamentos de los Microcontroladores,» de *Microcontroladores PIC. Sistema Integrado para el Auto Aprendizaje*, Barcelona (España), MARCOMBO, 2007, pp. 11-12.
- [45] CONELCOM GmbH, «Controlino,» [En línea]. Available: <https://www.controllino.biz/>. [Último acceso: 2020 Mayo 18].
- [46] RS Components Ltd, «RS,» [En línea]. Available: <https://cl.rsdelivers.com/product/siemens/6es7647-0ka01-0aa2/m%3%b3dulo-de-e-s-plc-siemens-6es7647-5-x-entrada/1354133>. [Último acceso: 2020 Mayo 18].
- [47] ES Gear Ltd., «Industruino,» 2019. [En línea]. Available: <https://industruino.com/>. [Último acceso: 18 Mayo 2020].
- [48] Industrial Shield, «M-DUINO,» INNOVATIVE SME, [En línea]. Available: https://www.industrialshields.com/es_ES/shop/product/is-mdduino-19r-m-duino-

plc-arduino-ethernet-19r-i-os-rele-analog-digital-plus-8?category=1. [Último acceso: 18 Mayo 2020].

- [49] ARDUINO, «Arduino,» 2020. [En línea]. Available: <https://store.arduino.cc/usa/arduino-uno-rev3>. [Último acceso: 18 Mayo 2020].
- [50] Arduino, «Arduino,» Arduino store, 18 Agosto 2020. [En línea]. Available: <https://store.arduino.cc/usa/mega-2560-r3>. [Último acceso: 28 Noviembre 2019].
- [51] A. G. González, «PanamaHITEK,» 23 Enero 2013. [En línea]. Available: <http://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>. [Último acceso: 28 Noviembre 2019].
- [52] JECRESPOM, «Aprendiendo Arduino,» WordPress.com., [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2016/07/04/ethernet-shield/>. [Último acceso: 6 Febrero 2020].
- [53] M. i. CHINA, «LUTIAN,» [En línea]. Available: https://es.made-in-china.com/co_lutian/product_Lutian-Type-Gasoline-Generator-2-5kw-2-8kw_rshushsig.html. [Último acceso: 6 Febrero 2020].
- [54] Amazon, «amazon,» [En línea]. Available: https://www.amazon.com/-/es/gp/product/B079L7YL2W/ref=ox_sc_act_title_1?smid=AJ9XYIB0X303E&th=1. [Último acceso: 2019 Febrero 15].
- [55] Amazon, «amazon,» [En línea]. Available: https://www.amazon.com/-/es/Tolako-Arduino-indicadora-funciona-oficiales/dp/B00VRUAHLE/ref=sr_1_6?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=arduino+relay&qid=1581131272&sr=8-6. [Último acceso: 2019 Febrero 15].
- [56] «AC-DC convertidor Buck de reducción para 110V 220V AC a DC 5V suministro para transformador de energía convertidor de voltaje regulador para Arduino,» AliExpress, [En línea]. Available: <https://es.aliexpress.com/item/32798625952.html>. [Último acceso: 21 Julio 2020].

- [57] L. LLamas, «RELOJ Y CALENDARIO EN ARDUINO CON LOS RTC DS1307 Y DS3231,» Tutoriales Android Intermedios, 2016 Octubre 2016. [En línea]. Available: <https://www.luisllamas.es/reloj-y- calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/>. [Último acceso: 18 Junio 2020].
- [58] Arduino, [En línea]. Available: <https://www.arduino.cc/en/Main/Software>. [Último acceso: 28 Febreo 2020].
- [59] CC BY-SA, «Geany,» 2020. [En línea]. Available: <https://geany.org/about/geany/>. [Último acceso: 15 Julio 2020].
- [60] Labcenter Electronics, «Proteo,» Labcenter Electronics, 2020. [En línea]. Available: <https://www.labcenter.com/>. [Último acceso: 2020 Junio 10].
- [61] D. Baculima, Diseño de un sistema de monitoreo y alarma para el sistema de respaldo electrico de la empresa Puntonet en la ciudad de Cuenca aplicado al nodo ubicado en Barabon, Cuenca: Universidad Politécnica Saieciana Sede Cuenca, 2012.
- [62] A. H. Donatte, Principios de la electricidad y electrónica II, Barcelona: MARCOMBO, 1999, p. 1.
- [63] «Paterson y A. Yau (2006). «Cryptography in theory and practice: The case of encryption in IPsec». Eurocrypt 2006, Lecture Notes in Computer Science Vol. 4004. pp. 12-29. Consultado el 13 de noviembre de 2007.Texto « Berlin » ignorado (ayuda)».
- [64] D. N. Acosta, «01-IntroduccionRT053,» [En línea]. Available: <http://www.exa.unicen.edu.ar/catedras/rtlinux/material/filminas/01-IntroduccionRT053.pdf>. [Último acceso: 7 Diciembre 2019].
- [65] jecrespon, «Aprendiendo Arduino,» WordPress, 26 Junio 2016. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/tag/aplicaciones-industriales/>. [Último acceso: 11 Enero 2020].
- [66] Amazon, «amazon,» [En línea]. Available: <https://www.amazon.com/-/es/jbtek-canales-Module-Arduino->

Raspberry/dp/B00KTEN3TM/ref=sr_1_7?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=arduino+relay&qid=1581131272&sr=8-7. [Último acceso: 2019 Febrero 15].

- [67] electric-generators.ru, «Generadores Yamaha EDL20000TE.,» 2014. [En línea]. Available: <http://electric-generators.ru/language/spanish/index372.html>. [Último acceso: 08 Febreo 2021].

11. ANEXOS

ANEXO 1. DISEÑO DEL CIRCUITO DE LA PCB EN EL SOFTWARE PROTEUS

Para el diseño del circuito se lo realizo con ayuda del Software de Proteus versión 8.8. Para añadir los módulos externos se optó por dejar conectores en la placa PCB tal como se observa la Figura 84.

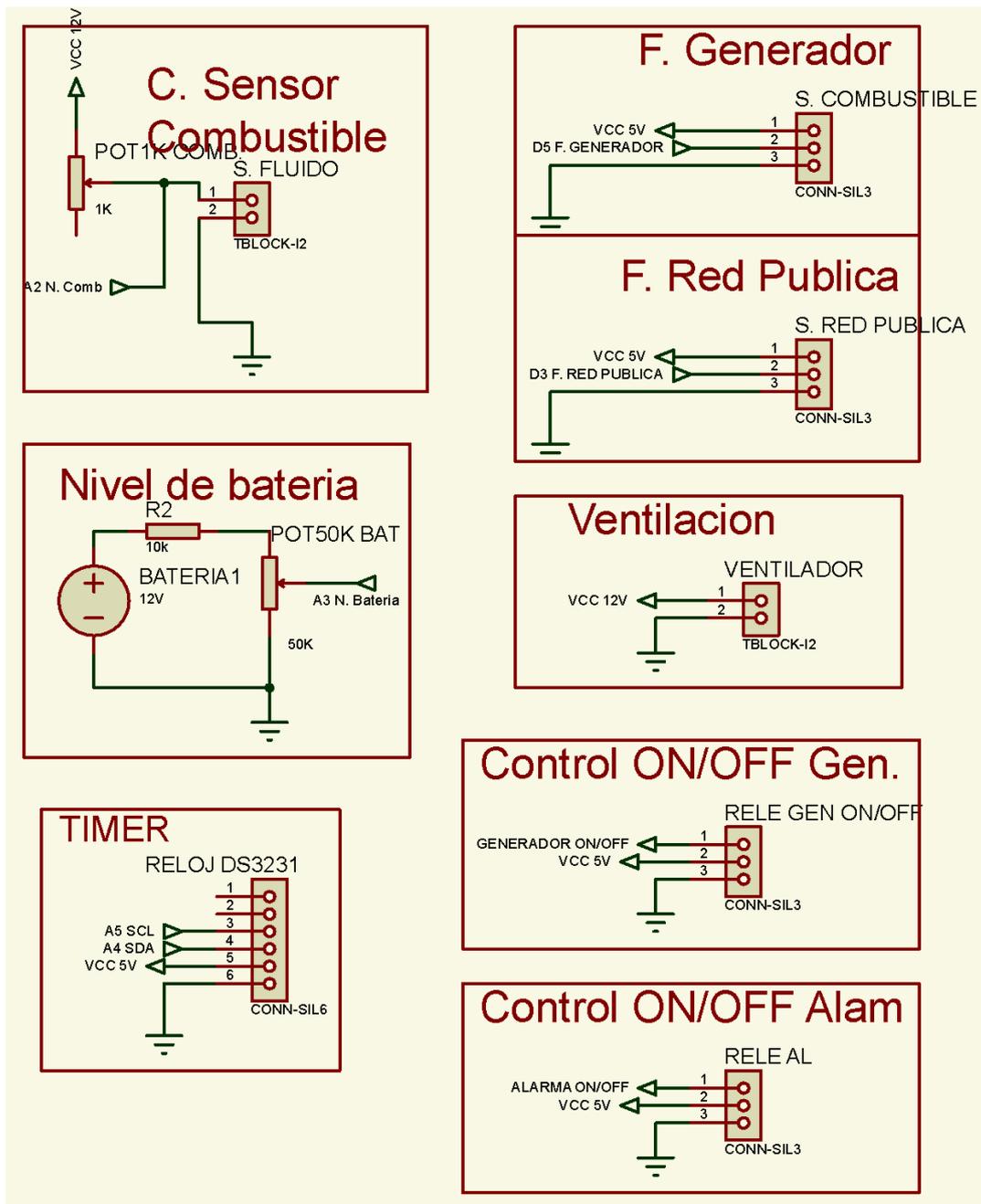


Figura 84. Diseño del circuito de todos los módulos a implementar del proyecto, en la función de Schematic Capture. Software: Proteus.

Fuente: Autor

Para la conexión de los módulos al controlador se lo realizo a través de terminales de nombre tanto de entrada como de salida  para evitar la confusión de cruce de cables.

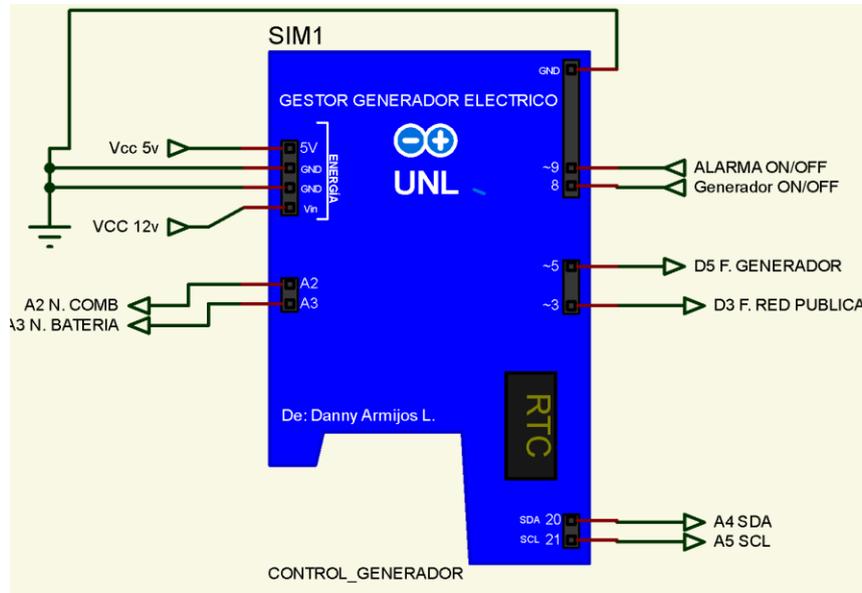


Figura 85. Conexión de los módulos a implementar al controlador Arduino ATmega 2560. Software: Proteus.

Fuente: Autor

La conexión de las pistas para crear la PCB se la realizo de manera manual, para así poder colocar los equipos en la posición más conveniente posible. Es conveniente tener a disposición los módulos externos a completar, pasa así tener en consideración sus dimensiones a ocupar en la placa de circuito impreso. También es importante etiquetar cada componente para así poder interpretar la posición de cada elemento de manera correcta en el circuito tal como se puede observar en la Figura 86 y Figura 87.

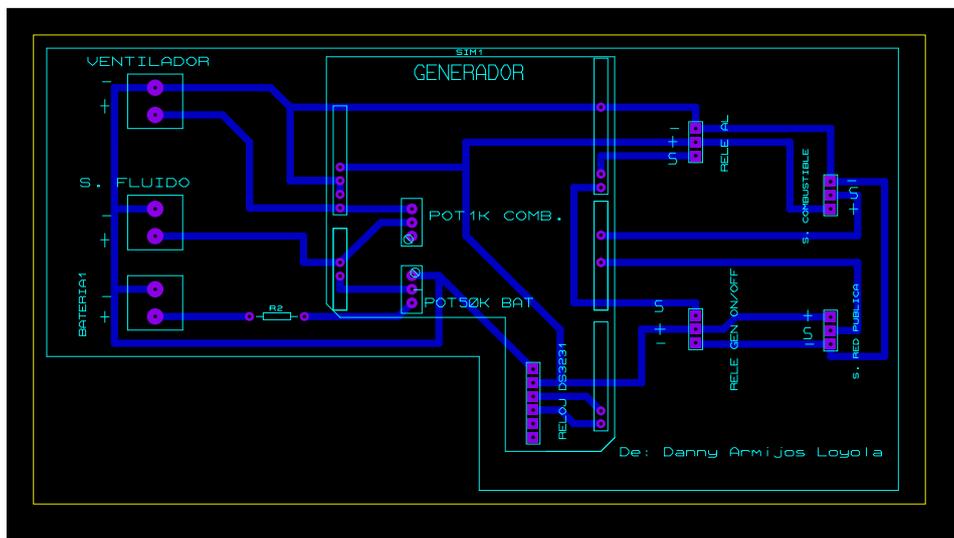


Figura 86. Diseño de la PCB mediante la función de ARES. Software: Proteus.

Fuente: Autor

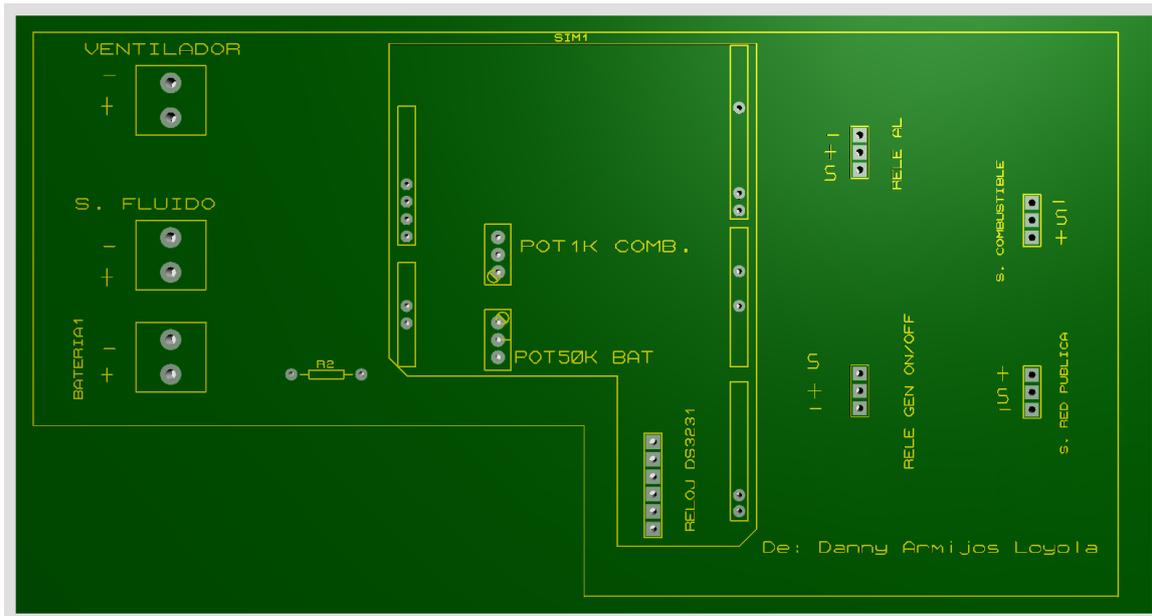


Figura 87. Visualización en 3D de la placa PCB del proyecto. Vista del tablero de la parte superior. Software: Proteus.

Fuente: Autor

ANEXO 2. INSTRUCCIONES PROGRAMADAS EN ARDUINO IDE PARA EL SERVIDOR WEB LOCAL.

Las siguientes instrucciones forman parte del código de programación que se carga al controlador ATmega 2560, en la cual se podrá observar librerías, funciones y comandos necesarios para poder interpretar la información obtenida por los módulos externos y reflejarla a través de una GUI en una página web.

```
❖ #include <SPI.h>
❖ #include <SD.h>
❖ #include <Ethernet.h>
❖ #include <GSM.h> //Permite realizar varias funciones a la vez
❖ #include "RTCLib.h"
❖ #include <Wire.h>
❖ #define REQ_BUF_SZ 60 // tamaño del búfer utilizado para capturar solicitudes HTTP
❖ RTC_DS3231 rtc; // módulo temporizador
❖ char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday", "Wednesday",
    "Thursday", "Friday", "Saturday"};
❖ const byte mac[] PROGMEM = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; // MAC del
    dispositivo
❖ const byte ip[] = { 192, 168, 1, 5 }; // ip de acceso
❖ const byte gateway[] PROGMEM = { 192, 168, 1, 1 }; // Puerta de enlace
❖ const byte subnet[] PROGMEM = { 255, 255, 255, 0 };
❖ String readString;
❖ EthernetServer server(80); //Definimos que el puerto HTTP será el 80
❖ EthernetServer server2(81);
❖ File webFile;
❖ File logFile; // El archivo de la página web en la tarjeta SD
❖ char HTTP_req[REQ_BUF_SZ] = {0}; // solicitud HTTP almacenada en búfer como una
    cadena terminada en nulo
❖ char req_index = 0; // index into HTTP_req solicitud de búfer
❖ boolean SW_state[2] = {0}; // Almacena estados de entrada digitales On/off
❖ int REGISTROg = LOW;

❖ /**** INSTRUCCIONES DE LA TARJETA SD ****/
❖ #define SDCARD_CS 4
❖ File root;
❖ #if defined(ESP8266)
❖ // default for ESPressif
❖ #define WIZ_CS 15
❖ #elif defined(ESP32)
❖ #define WIZ_CS 33
❖ #elif defined(ARDUINO_STM32_FEATHER)
❖ // default for WICED
❖ #define WIZ_CS PB4
❖ #elif defined(TEENSYDUINO)
```

```

❖ #define WIZ_CS 10
❖ #elif defined(ARDUINO_FEATHER52)
❖ #define WIZ_CS 11
❖ #else // default for 328p, 32u4 and m0
❖ #define WIZ_CS 10
❖ #endif
❖ #define error(s) error_P(PSTR(s)) // almacenas cadenas de error en flash para ahorrar RAM
❖ void error_P(const char* str) {
❖ while(1);
❖ }
❖ /**** FIN INSTRUCCIONES DE LA TARJETA SD ****/

❖ void setup(){
❖ Serial.begin(115200);
❖ if (!SD.begin(SDCARD_CS)) {
❖ error("card.init failed!");
❖ }
❖ root = SD.open("/");
❖ printDirectory(root, 0);
❖ Ethernet.init(WIZ_CS);
❖ delay(1000);

❖ Ethernet.begin(mac, ip); //Iniciar la conexión de red y serie
❖ server.begin();
❖ server2.begin();

❖ //****Para analizar el suministro de energía****//
❖ pinMode(3, INPUT); // red publica
❖ pinMode(5, INPUT); // Generador
❖ //****Para saber el control de RELE ARDUINO:****//
❖ pinMode(8, OUTPUT); //Generador ON/OFF
❖ pinMode(9, OUTPUT); // Alarma ON/OFF

❖ //**** Inicio Fecha****//
❖ #ifndef ESP8266
❖ // while (!Serial); // Espere a que se conecte el puerto serie. Necesario para USB nativo
❖ #endif
❖ if (!rtc.begin()) {
❖ Serial.println("Couldn't find RTC");
❖ Serial.flush();
❖ abort();
❖ }
❖ if (rtc.lostPower()) {
❖ Serial.println("RTC perdió energía, ajustemos la hora!");
❖ // rtc.adjust(DateTime(F(__DATE__), F(__TIME__))); // solo usar una vez y luego desactivar

```

```

❖ }
❖ //**** Fin Fecha ****//
❖ }

❖ void registro(){
❖ File logFile;
❖ logFile = SD.open("datalog.txt", FILE_WRITE);
❖ if (logFile) {

❖ //**** INICIO CONTROL DE FUNCIONAMIENTO ****//
❖ if (digitalRead(3)==HIGH ) { // Si existe energia en red pública entonces
❖     digitalWrite(8, HIGH); // Generador apagado los cables se conectan a COM y
    NO al rele arduino
❖     digitalWrite(9, LOW); // Abre el circuito de arranque para que no se encienda el
    generador
❖     arranque=LOW; //variable utilizada para ejecutar una solo una vez el arranque
❖     }
❖
❖     if (digitalRead(3)==LOW && SW_state[0]== 1) { // Si no existe energia en red
    pública y el control generador esta encendido
❖         digitalWrite(8, LOW); // Abre el interruptor pasando de OFF a ON
❖         while(arranque==LOW){ //nos indica que mientras que mientras arranque
    este en bajo se ejecutara las intrucciones
❖             digitalWrite(9, HIGH); // sirbe para da un pulso de arranque al generador
❖             delay(2500); // determina la duración del pulso
❖             digitalWrite(9, LOW); // cierra el interruptor de arranque
❖             arranque=!arranque; // permite negar el valor de arranque con el objetico
    que no se ejecute vuelta la condición
❖         }
❖     }
❖
❖     if (digitalRead(3)==LOW && SW_state[0]== 0) { // Si no existe energia en red
    pública y el control generador esta apagado
❖         digitalWrite(8, HIGH); // Generador apagado los cables se conectan a COM y
    NO al rele arduino
❖         digitalWrite(9, LOW); // Abre el circuito de arranque para que no se encienda el
    generador
❖         arranque=LOW; // variable utilizada para ejecutar una solo una vez el
    arranque
❖     }

❖ //**** Inicio de registro cuando retorna energia Electrica de la red pública ****//
❖ if(digitalRead(3)==HIGH && REGISTROg==LOW){
❖     DateTime now = rtc.now(); // MODULO RTC PARA EL REGISTRO DE LA HORA//
❖     //logFile.println(" Fecha          | Hora          | EQ. funcionando | Nivel de
    combustible | Nivel de energia | Temperatura  ");
❖     logFile.print(daysOfTheWeek[now.dayOfTheWeek()]); logFile.print("/");
❖     logFile.print(now.year(), DEC); logFile.print("/"); logFile.print(now.month(), DEC);
    logFile.print("/"); logFile.print(now.day(), DEC); logFile.print(" ");
❖     logFile.print(" Hora  :"); logFile.print(now.hour(), DEC); logFile.print(':');
    logFile.print(now.minute(), DEC); logFile.print(':'); logFile.println(now.second(), DEC);

```

```

❖ logFile.println(" EQ. funcionando | Nivel de combustible | Nivel de
energia | Temperatura ");
❖ logFile.print(" RED PÚBLICA "); logFile.print(" | ");
❖ logFile.print(15349.81/analogRead(2)+analogRead(2)/(-39.8392)); // Lectura del sensor
de Combustible
❖ logFile.print(" | ");
❖ logFile.print(analogRead(3)/10.23); // Lectura del sensor de nivel de energía
❖ logFile.print(" | ");
❖ logFile.print(rtc.getTemperature());
❖ logFile.println(" C");
❖ logFile.println();
❖ REGISTROg = !REGISTROg;
❖ } //**** FIN de registro cuando retorna energia Electrica de la red pública ****//

❖ //**** Inicio de registro cuando se enciende el generador de combustible ****//
❖ if(digitalRead(5)==HIGH && REGISTROg==HIGH){
❖ DateTime now = rtc.now(); // MODULO RTC PARA EL REGISTRO DE LA HORA//
❖ //logFile.println(" Fecha | Hora | EQ. funcionando | Nivel de
combustible | Nivel de energia | Temperature ");
❖ logFile.print(daysOfTheWeek[now.dayOfTheWeek()]); logFile.print("/");
❖ logFile.print(now.year(), DEC); logFile.print("/"); logFile.print(now.month(), DEC);
logFile.print("/"); logFile.print(now.day(), DEC); logFile.print(" ");
❖ logFile.print(" Hora :"); logFile.print(now.hour(), DEC); logFile.print(':');
logFile.print(now.minute(), DEC); logFile.print(':'); logFile.println(now.second(), DEC);
❖ logFile.println(" EQ. funcionando | Nivel de combustible | Nivel de
energia | Temperatura ");
❖ logFile.print(" GENERADOR | "); // log the analog pin value
❖ logFile.print(15349.81/analogRead(2)+analogRead(2)/(-39.8392)); // Lectura del sensor
de Combustible
❖ logFile.print(" | ");
❖ logFile.print(analogRead(3)/10.23); // Lectura del sensor de nivel de energía
❖ logFile.print(" | ");
❖ logFile.print(rtc.getTemperature());
❖ logFile.println(" C");
❖ logFile.println(); //**** FIN de registro cuando se apaga el generador de combustible
****//
❖ REGISTROg = !REGISTROg;
❖ }
❖ logFile.close();
❖ delay(1000);
❖ }
❖ else {
❖ Serial.println(F("Error al abrir el archivo"));
❖ }
❖ }

❖ //
❖ void ListFiles(EthernetClient cliente, uint8_t flags, File dir) {

```

```

❖ cliente.println("<ul>");
❖ while (true) {
❖ File entry = dir.openNextFile(); // hecho si pasó la última entrada utilizada
❖ if (! entry) { // no más archivos
❖ break;
❖ }
❖ cliente.print("<li><a href=\""); //imprime los espacios de sangría
❖ cliente.print(entry.name());
❖ if (entry.isDirectory()) {
❖ cliente.println("/");
❖ }
❖ cliente.print("\>");
❖ cliente.print(entry.name()); // imprimir nombre de archivo con posible relleno en blanco
❖ if (entry.isDirectory()) {
❖ cliente.println("/");
❖ }
❖ cliente.print("</a>");
❖ cliente.println("</li>");
❖ entry.close();
❖ }
❖ cliente.println("</ul>");
❖ }
❖ // Qué tan grande debe ser nuestro búfer de línea
❖ // 100 es suficiente!
❖ #define BUFSIZ 100

❖ void loop()
❖ {
❖ EthernetClient client = server.available(); // intenta conseguir cliente
❖ if (client) { // tienes cliente?
❖ boolean currentLineIsBlank = true;
❖ while (client.connected()) {
❖ if (client.available()) { // datos del cliente disponibles para leer
○ char c = client.read(); // lee 1 byte (carácter) del cliente
○ // limitar el tamaño de la solicitud HTTP recibida almacenada
○ // almacena en búfer la primera parte de la solicitud HTTP en la matriz HTTP_req
(cadena)
○ // dejar el último elemento en la matriz como 0 para terminar la cadena nula
(REQ_BUF_SZ - 1)
○ if (req_index < (REQ_BUF_SZ - 1)) {
▪ HTTP_req[req_index] = c; // guardar el carácter de solicitud HTTP
▪ req_index++;
○ }
○ // la última línea de la solicitud del cliente está en blanco y termina con \ n
○ // responder al cliente solo después de recibir la última línea
○ if (c == '\n' && currentLineIsBlank) {
▪ // enviar un encabezado de respuesta http estándar
▪ client.println("HTTP/1.1 200 OK");

```

- // el resto del encabezado sigue a continuación, dependiendo de si
 - // se solicita una página web o XML
 - // Solicitud Ajax - enviar archivo XML
 - `if (StrContains(HTTP_req, "ajax_inputs")) {`
 - // enviar el resto del encabezado HTTP
 - `client.println("Content-Type: text/xml");`
 - `client.println("Connection: keep-alive");`
 - `client.println();`
 - `SetLEDs();`
 - // enviar archivo XML que contiene estados de entrada
 - `XML_response(client);`
 - `}`
 - `else {` // solicitud de página web
 - // enviar el resto del encabezado HTTP
 - `client.println("Content-Type: text/html");`
 - `client.println("Connection: keep-alive");`
 - `client.println();`
 - // enviar página web
- // abre el archivo de página web solicitado
 - `if (StrContains(HTTP_req, "GET / ") || StrContains(HTTP_req, "GET /index.htm")) {`
 - `webFile = SD.open("index.htm");` //abre el archivo de página web
 - `}`
 - `else if (StrContains(HTTP_req, "GET /page2.htm")) {`
 - `webFile = SD.open("page2.htm");` // abre el archivo de página web
 - `}`
 - `if (webFile) {`
 - `while(webFile.available()) {`
 - `client.write(webFile.read());` // enviar página web al cliente
 - `}`
 - `webFile.close();`
 - `}`
 - `}`
 - // muestra la solicitud HTTP recibida en el puerto serie
 - `//Serial.print(HTTP_req);`
 - // restablece el índice del búfer y todos los elementos del búfer a 0
 - `req_index = 0;`
 - `StrClear(HTTP_req, REQ_BUF_SZ);`
 - `break;`
 - `}`
 - // cada línea de texto recibida del cliente termina con `\r \n`
 - `if (c == '\n') {`
 - // último carácter en la línea del texto recibido
 - // comenzando una nueva línea con el siguiente carácter leído
 - `currentLineIsBlank = true;`
 - `}`

```

    ○ else if (c != '\r') {
    ○ // se recibió un carácter de texto del cliente
      ▪ currentLineIsBlank = false;
    ○ }
❖ } // finaliza si (cliente disponible ())
❖ } // finaliza mientras (cliente.conectado ())
❖ delay(1); // darle tiempo al navegador web para recibir los datos
❖ client.stop(); // cerrar la conexión
❖ } // finaliza if (cliente)
❖ //Cerrar sesión en puerto 80

❖ // Esperamos que venga algún cliente web al puerto 81
❖ char clientline[BUFSIZ];
❖ char name[17];
❖ int index = 0;
❖ EthernetClient cliente = server2.available();
❖ if (cliente) {
❖ // una solicitud http termina con una línea en blanco
❖ boolean current_line_is_blank = true;
❖ // restablecer el búfer de entrada
❖ index = 0;
❖ while (cliente.connected()) {
❖ if (cliente.available()) {
❖ char c = cliente.read();
❖ // Si no es una línea nueva, agregue el carácter al búfer
❖ if (c != '\n' && c != '\r') {
❖ clientline[index] = c;
❖ index++;
❖ // ¿Son demasiado grandes para el búfer? empezar a tirar datos
❖ if (index >= BUFSIZ)
❖ index = BUFSIZ - 1;
❖ // ¡sigue leyendo más datos!
❖ continue;
❖ }
❖ // obtuvo una nueva línea \n o \r, lo que significa que la cadena está lista
❖ clientline[index] = 0;
❖ // Imprimirlo para depurarlo
❖ //Serial.println(clientline);
❖ // Busque una subcadena como una solicitud para obtener el archivo
❖ if (strstr(clientline, "GET /") != 0) {
❖ // esta vez no hay espacio después de /, ¡así que un subarchivo!
❖ char *filename;
❖ filename = clientline + 5; // cuida el "GET /" (5 caracteres) *****
❖ // un pequeño truco, busque la cadena "HTTP / 1.1" y
❖ // convierte el primer carácter de la subcadena en un 0 para borrarlo.
❖ (strstr(clientline, " HTTP"))[0] = 0;

❖ if(filename[strlen(filename)-1] == '/') { // Recortar un nombre de archivo de directorio

```

```

❖ filename[strlen(filename)-1] = 0;    // como abierto lanza error con final /
❖ }

❖ // Serial.print (F ("Solicitud web para:")); Serial.println (nombre de archivo); //
  imprimimos el archivo que queremos
❖ File file = SD.open(filename, O_READ);
❖ if ( file == 0 ) { // Abrir el archivo con el código de retorno 0 es un error en SDFile.open
❖ cliente.println("HTTP/1.1 404 Not Found");
❖ cliente.println("Content-Type: text/html");
❖ cliente.println();
❖ cliente.println("<h2>File Not Found!</h2>");
❖ cliente.println("<br><h3>Couldn't open the File!</h3>");
❖ break;
❖ }

❖ // Serial.println ("¡Archivo abierto!");
❖ cliente.println("HTTP/1.1 200 OK");
❖ if (file.isDirectory()) {
    o // Serial.println ("es un directorio");
    o //file.close ();
❖ cliente.println("Content-Type: text/html");
❖ cliente.println();
❖ cliente.print("<h1>Base de datos del funcionamiento del Sistema </h1>");
    o cliente.print("<h2>Bienvenido tecnico encargado REALIZE SUS FUNCIONES
      CON RESPONSABILIDAD atentamente Gerencia ");
❖ cliente.print(filename);
❖ cliente.println(":</h2>");
❖ ListFiles(cliente,LS_SIZE,file);
❖ file.close();
❖ } else { // Cualquier clic que no sea de directorio, server2 enviará el archivo al cliente
  para su descarga
❖ cliente.println("Content-Type: application/octet-stream");
❖ cliente.println();

❖ char file_buffer[16];
❖ int avail;
❖ while (avail = file.available()) {
    o int to_read = min(avail, 16);
    o if (to_read != file.read(file_buffer, to_read)) {
    o break;
    o }
    o // descomenta el serial para depurar (lento!)
    o //Serial.write(((char)c);
    o cliente.write(file_buffer, to_read);
❖ }
❖ file.close();
❖ }
❖ } else {

```

```

❖ // todo lo demás es un 404
❖ cliente.println("HTTP/1.1 404 Not Found");
❖ cliente.println("Content-Type: text/html");
❖ cliente.println();
❖ cliente.println("<h2>File Not Found!</h2>");
❖ }
❖ break;
❖ }
❖ }
❖ // darle tiempo al navegador web para recibir los datos
❖ delay(1);
❖ cliente.stop();
❖ }
❖ registro();
❖ }

❖ void printDirectory(File dir, int numTabs) {
❖ while(true) {
❖ File entry = dir.openNextFile();
❖ if (! entry) {
❖ // no más archivos
❖ break;
❖ }
❖ for (uint8_t i=0; i<numTabs; i++) {
❖ //serial.print('\t');
❖ }
❖ //serial.print(entry.name());
❖ if (entry.isDirectory()) {
❖ //serial.println("");
❖ printDirectory(entry, numTabs+1);
❖ } else {
❖ // files have sizes, directories do not
❖ //serial.print("\t\t");
❖ //serial.println(entry.size(), DEC);
❖ }
❖ entry.close();
❖ }
❖ }

❖ // comprueba si la solicitud HTTP recibida está encendiendo / apagando los RELES
❖ // también guarda el estado de los RELES
❖ void SetLEDs(void)
❖ {
❖ // Control ON/OFF del generador mediante la web
❖ if (StrContains(HTTP_req, "SW1=1")) {
❖ SW_state[0] = 1; // guarda estado del Generador
❖ digitalWrite(8, HIGH); // Prende el generador
❖ }

```

```

❖ else if (StrContains(HTTP_req, "SW1=0")) {
❖ SW_state[0] = 0; // guarda estado del Generador
❖ digitalWrite(8, LOW); // apaga el generador
❖ }
❖ // Control ON/OFF de la Alarma mediante la web
❖ if (StrContains(HTTP_req, "SW2=1")) {
❖ SW_state[1] = 1; // guarda estado de la Alarma
❖ digitalWrite(20, HIGH); // prende la alarma
❖ }
❖ else if (StrContains(HTTP_req, "SW2=0")) {
❖ SW_state[1] = 0; // guarda estado del Alarma
❖ digitalWrite(20, LOW); //Apaga la alarma
❖ }
❖ }

❖ // envía el archivo XML con valores analógicos, estado del interruptor
❖ // y estado del Generador
❖ void XML_response(EthernetClient cl)
❖ {
❖ int analog_val_2; // almacena el valor leído de las entradas analógicas
❖ int analog_val_3;
❖ int digital_val_3; // almacena el valor leído de las entradas digitales
❖ int digital_val_5;

❖ cl.print("<?xml version = \"1.0\" ?>");
❖ cl.print("<inputs>");

❖ // Lee el pin analogico A2
❖ // Indicador del nivel de combustible
❖ analog_val_2 += analogRead(2);
❖ cl.print("<analog>");
❖ //cl.print(18254.88959/analog_val_2-23.3438);
❖ cl.print(((15349.81/analog_val_2)+(analog_val_2/(-39.8392))));
❖ cl.println("</analog>");

❖ //Lee el pin analogico A3
❖ // Indicador de nivel de batería
❖ analog_val_3 += analogRead(3);
❖ cl.print("<analog>");
❖ cl.print(analog_val_3/10.23);
❖ cl.println("</analog>");

❖ // Lee el pin digital D3
❖ // analiza si existe energia red electrica
❖ digital_val_3 = digitalRead(3);
❖ cl.print("<switch>");
❖ if (digital_val_3) {
❖ //digitalWrite(8, LOW); // si hay energia electrica obliga apagar el generador

```

```

    ○ cl.print("ENCENDIDO"); // señal que el suministro de red electrica esta
      funcionando
    ○ }
    ○ else {
    ○ // digitalWrite(8, HIGH); // si no hay energia electrica prende el generador
    ○ cl.print("APAGADO"); // señal que el suministro de red electrica NO esta
      funcionando
    ○ }
❖ cl.println("</switch>");

❖ // Lee el pin digital D5
❖ // analiza si existe energia generador
❖ digital_val_5 = digitalRead(5);
❖ cl.print("<switch>");
❖ if (digital_val_5) {
    ○ cl.print("ENCENDIDO"); // señal que el generador esta funcionando
    ○ }
    ○ else {
    ○ cl.print("APAGADO");// señal que el generador NO funciona
    ○ }
❖ cl.println("</switch>");

❖ // button de Control de estados
❖ // Control del Generador
❖ cl.print("<INTER>");
❖ if (SW_state[0] || !digital_val_3) {
❖ cl.print("on");
❖ }
❖ else {
❖ cl.print("off");
❖ }
❖ cl.println("</INTER>");
❖ // Control de alarma
❖ cl.print("<INTER>");
❖ if (SW_state[1] || ((15349.81/analog_val_2)+(analog_val_2/(-39.8392))) <=40 ||
  analog_val_3/10.23 <=40 ) {
❖ digitalWrite(9, HIGH);
❖ cl.print("on");
❖ }
❖ else {
❖ digitalWrite(9, LOW);
❖ cl.print("off");
❖ }
❖ cl.println("</INTER>");

❖ cl.print("</inputs>");
❖ }
❖ // establece cada elemento de str en 0 (limpia la matriz)

```

```

❖ void StrClear(char *str, char length)
❖ {
❖ for (int i = 0; i < length; i++) {
❖ str[i] = 0;
❖ }
❖ }
❖ // busca la cadena find en la cadena str
❖ // devuelve 1 si se encuentra una cadena
❖ // devuelve 0 si no se encuentra la cadena
❖ char StrContains(char *str, char *sfind)
❖ {
❖ char found = 0;
❖ char index = 0;
❖ char len;
❖ len = strlen(str);
❖ if (strlen(sfind) > len) {
❖ return 0;
❖ }
❖ while (index < len) {
❖ if (str[index] == sfind[found]) {
❖ found++;
❖ if (strlen(sfind) == found) {
❖     return 1;
❖ }
❖ }
❖ else {
❖ found = 0;
❖ }
❖ index++;
❖ }
❖ return 0;
❖ }

```

ANEXO 3: CÓDIGO DEL DISEÑO WEB EN HTML

Para la creación de la GUI es necesario crear un archivo .htm basado el lenguaje de etiquetas de hipertexto (página web) que permita recibir tramas de información del controlador Arduino. En las instrucciones siguiente se describe las funciones necesarias para poder crear la GUI del sistema.

- ❖ `<!DOCTYPE html>` `<!-- indica el tipo de documento. Es un preámbulo requerido. -->`
- ❖ `<html>` `<!--elemento encierra todo el contenido de la página entera -->`
- ❖ `<head>` `<!--Este elemento actúa como un contenedor de todo aquello que quieres incluir en la página HTML que no es contenido visible por los visitantes de la página -->`
- ❖ `<title>Arduino Ajax I/O</title>` `<!-- Sirve para asignar un título a la página web-->`
- ❖ `<script>` `<!-- contiene las instrucciones, funciones y variables que interactúa cliente servidor. -->`
 - `strSW1 = "";` `<!-- representa la cadena de un numero -->`
 - `strSW2 = "";`
 - `var SW1_state = 0;` `<!-- se asigna la variable -->`
 - `var SW2_state = 0;`
 - `var data_val2 = 0;`
 - `var data_val3 = 0;`
 - `var data_val4 = 0;`
 - `var data_val5 = 0;`
- ❖ `<!-- Comienzos de código de calibre / se lo utiliza para los indicadores de nivel de combustible y nivel de batería-->`
- ❖ `var Gauge=function(b){function l(a,b){for(var c in b)"object"===typeof b[c]&&"[object Array]"!==Object.prototype.toString.call(b[c])&&"renderTo"!==c?("object"!==typeof a[c]&&(a[c]={}),l(a[c],b[c])):a[c]=b[c]}function q(){z.width=b.width;z.height=b.height;A=z.cloneNode(!0);B=A.getContext("2d");C=z.width;D=z.height;t=C/2;u=D/2;f=t<u?t:u;A.i8d=!1;B.translate(t,u);B.save();a.translate(t,u);a.save()}function v(a){var b=new Date;G=setInterval(function(){var c=(new Date-b)/a.duration;1<c&&(c=1);var f=("function"===`
- ❖ `typeof`
`a.delta?a.delta:M[a.delta])(c);a.step(f);1===c&&clearInterval(G)},a.delay||10)}function k(){G&&clearInterval(G);var a=In,h=n,c=b.animation;v({delay:c.delay,duration:c.duration,delta:c.fn,step:function(b){n=parseFloat(h)+a*b;E.draw()}})}function e(a){return a*Math.PI/180}function g(b,h,c){c=a.createLinearGradient(0,0,0,c);c.addColorStop(0,b);c.addColorStop(1,h);return c}function p(){var m=93*(f/100),h=f-m,c=91*(f/100),e=88*(f/100),d=85*(f/100);a.save();b.glow&&(a.shadowBlur=h,a.shadowColor=`
- ❖ `"rgba(0, 0, 0,`
`0.5)");a.beginPath();a.arc(0,0,m,0,2*Math.PI,!0);a.fillStyle=g("#ddd", "#aaa",m);a.fill();a.restore();a.beginPath();a.arc(0,0,c,0,2*Math.PI,!0);a.fillStyle=g("#fafafa", "#ccc",c);a.fill();a.beginPath();a.arc(0,0,e,0,2*Math.PI,!0);a.fillStyle=g("#eee", "#f0f0f0",e);a.fill();a.beginPath();a.arc(0,0,d,0,2*Math.PI,!0);a.fillStyle=b.colors.plate;a.fill();a.save()}function w(a){var`

```

h=!1;a=0===b.majorTicksFormat.dec?Math.round(a).toString():a.toFixed(b.majorTicksFormat.dec);return 1<b.majorTicksFormat["int"]?
❖ (h=-1<a.indexOf("."),-1<a.indexOf("-")?"-
"+(b.majorTicksFormat["int"]+b.majorTicksFormat.dec+2+(h?1:0)-a.length)+a.replace("-
",""):""+(b.majorTicksFormat["int"]+b.majorTicksFormat.dec+1+(h?1:0)-
a.length)+a)}function
d(){var
m=81*(f/100);a.lineWidth=2;a.strokeStyle=b.colors.majorTicks;a.save();if(0===b.majorTi
cks.length){for(var
h=(b.maxValue-
b.minValue)/5,c=0;5>c;++c)b.majorTicks.push(w(b.minValue+h*c));b.majorTicks.push(w
(b.maxValue))}for(c=0;c<b.majorTicks.length;++c)a.rotate(e(45+c*(270/(b.majorTicks.len
gth-
❖ 1))))a.beginPath(),a.moveTo(0,m),a.lineTo(0,m-
15*(f/100)),a.stroke(),a.restore(),a.save();b.strokeTicks&&(a.rotate(e(90)),a.beginPath(),a.a
rc(0,0,m,e(45),e(315),!1),a.stroke(),a.restore(),a.save())}function
J(){var
m=81*(f/100);a.lineWidth=1;a.strokeStyle=b.colors.minorTicks;a.save();for(var
h=b.minorTicks*(b.majorTicks.length-
1),c=0;c<h;++c)a.rotate(e(45+c*(270/h))),a.beginPath(),a.moveTo(0,m),a.lineTo(0,m-
7.5*(f/100)),a.stroke(),a.restore(),a.save())}function
s(){for(var
m=55*(f/100),h=0;h<b.majorTicks.length;++h){var c=
❖ F(m,e(45+h*(270/(b.majorTicks.length-1)))));a.font=20*(f/200)+"px
Arial";a.fillStyle=b.colors.numbers;a.lineWidth=0;a.textAlign="center";a.fillText(b.majorT
icks[h],c.x,c.y+3)}function
x(a){var
h=b.valueFormat.dec,c=b.valueFormat["int"];a=parseFloat(a);var
f=0>a;a=Math.abs(a);if(0<h){a=a.toFixed(h).toString().split(".");h=0;for(c-
=a[0].length;h<c;++h)a[0]="0"+a[0];a=(f?"-
":"")+a[0]+"."+a[1]}else{a=Math.round(a).toString();h=0;for(c-
=a.length;h<c;++h)a="0"+a;a=(f?"-":"")+a}return a}function F(a,b){var c=
❖ Math.sin(b),f=Math.cos(b);return{x:0*f-a*c,y:0*c+a*f}}function
N(){a.save();for(var
m=81*(f/100),h=m-15*(f/100),c=0,g=b.highlights.length;c<g;c++){var
d=b.highlights[c],r=(b.maxValue-b.minValue)/270,k=e(45+(d.from-
b.minValue)/r),r=e(45+(d.to-
b.minValue)/r);a.beginPath();a.rotate(e(90));a.arc(0,0,m,k,r,!1);a.restore();a.save();var
l=F(h,k),p=F(m,k);a.moveTo(l.x,l.y);a.lineTo(p.x,p.y);var
p=F(m,r),n=F(h,r);a.lineTo(p.x,p.y);a.lineTo(n.x,n.y);a.lineTo(l.x,l.y);a.closePath();a.fillSt
yle=d.color;a.fill();
❖ a.beginPath();a.rotate(e(90));a.arc(0,0,h,k-
0.2,r+0.2,!1);a.restore();a.closePath();a.fillStyle=b.colors.plate;a.fill();a.save()}}function
K(){var
m=12*(f/100),h=8*(f/100),c=77*(f/100),d=20*(f/100),k=4*(f/100),r=2*(f/100),l=function(
){a.shadowOffsetX=2;a.shadowOffsetY=2;a.shadowBlur=10;a.shadowColor="rgba(188,
143,
143,
0.45)"};l();a.save();n=0>n?Math.abs(b.minValue-n):0<b.minValue?n-
b.minValue:Math.abs(b.minValue)+n;a.rotate(e(45+n/((b.maxValue-
b.minValue)/270)));a.beginPath();a.moveTo(-r,-d);a.lineTo(-k,
❖ 0);a.lineTo(-1,c);a.lineTo(1,c);a.lineTo(k,0);a.lineTo(r,-
d);a.closePath();a.fillStyle=g(b.colors.needle.start,b.colors.needle.end,c-
d);a.fill();a.beginPath();a.lineTo(-0.5,c);a.lineTo(-1,c);a.lineTo(-k,0);a.lineTo(-r,-
d);a.lineTo(r/2-2,-d);a.closePath();a.fillStyle="rgba(255,
255,
255,
0.2)";a.fill();a.restore();l();a.beginPath();a.arc(0,0,m,0,2*Math.PI,!0);a.fillStyle=g("#f0f0f0

```

```

", "#ccc", m); a.fill(); a.restore(); a.beginPath(); a.arc(0, 0, h, 0, 2 * Math.PI, !0); a.fillStyle = g("#e8e8e8", "#f5f5f5", h); a.fill()
❖ function L() { a.save(); a.font = 40 * (f / 200) + "px Led"; var b = x(y), h = a.measureText("-" + x(0)).width, c = f - 33 * (f / 100), g = 0.12 * f; a.save(); var d = -h / 2 - 0.025 * f, e = c - g - 0.04 * f, h = h + 0.05 * f, g = g + 0.07 * f, k = 0.025 * f; a.beginPath(); a.moveTo(d + k, e); a.lineTo(d + h - k, e); a.quadraticCurveTo(d + h, e, d + h, e + k); a.lineTo(d + h, e + g - k); a.quadraticCurveTo(d + h, e + g, d + h - k, e + g); a.lineTo(d + k, e + g); a.quadraticCurveTo(d, e + g, d, e + g - k); a.lineTo(d, e + k); a.quadraticCurveTo(d, e, d + k, e); a.closePath(); d = a.createRadialGradient(0, c - 0.12 * f - 0.025 * f + (0.12 * f + 0.045 * f) / 2, f / 10, 0, c - 0.12 * f - 0.025 * f + (0.12 * f + 0.045 * f) / 2, f / 5); d.addColorStop(0, "#888"); d.addColorStop(1, "#666"); a.strokeStyle = d; a.lineWidth = 0.05 * f; a.stroke(); a.shadowBlur = 0.012 * f; a.shadowColor = "rgba(0, 0, 1)"; a.fillStyle = "#babab2"; a.fill(); a.restore(); a.shadowOffsetX = 0.004 * f; a.shadowOffsetY = 0.004 * f; a.shadowBlur = 0.012 * f; a.shadowColor = "rgba(0, 0, 0.3)"; a.fillStyle = "#444"; a.textAlign = "center"; a.fillText(b, 0, c); a.restore() } Gauge.Collection.push(this); this.config = { renderTo: null, width: 200, height: 200, title: !1,
❖ max: 100, min: 0, major: [], minor: 10, stroke: !0, units: !1, valueFormat: { "int": 3, dec: 2 }, majorFormat: { "int": 1, dec: 0 }, glow: !0, animation: { delay: 10, duration: 250, fn: "cycle" }, colors: { plate: "#fff", major: "#444", minor: "#666", title: "#888", units: "#888", numbers: "#444", needle: { start: "rgba(240, 128, 128, 1)", end: "rgba(255, 160, 122, .9)" } }, highlights: [ { from: 20, to: 60, color: "#eee" }, { from: 60, to: 80, color: "#ccc" }, { from: 80, to: 100, color: "#999" } ]; var y = 0, E = this, n = 0, I = 0, H = !1; this.setValue =
❖ function(a) { n = b.animation ? y : a; var d = (b.max - b.min) / 100; I = a > b.max ? b.max + d : a < b.min ? b.min - d : a; y = a; b.animation ? k() : this.draw(); return this }; this.setRawValue = function(a) { n = y = a; this.draw(); return this }; this.clear = function() { y = n = I = this.config.min; this.draw(); return this }; this.getValue = function() { return y }; this.onready = function() { l(this.config, b); this.config.min = parseFloat(this.config.min); this.config.max = parseFloat(this.config.max); b = this.config; n =
❖ y = b.min; if (!b.renderTo) throw Error("Canvas element was not specified when creating the Gauge object!"); var z = b.renderTo.tagName ? b.renderTo : document.getElementById(b.renderTo), a = z.getContext("2d"), A, C, D, t, u, f, B; q(); this.updateConfig = function(a) { l(this.config, a); q(); this.draw(); return this }; var M = { linear: function(a) { return a }, quad: function(a) { return Math.pow(a, 2) }, quint: function(a) { return Math.pow(a, 5) }, cycle: function(a) { return 1 - Math.sin(Math.acos(a)) }, bounce: function(a) { a: { a = 1 - a; for (var b = 0,
❖ c = 1; b += c, c /= 2) if (a >= (7 - 4 * b) / 11) { a = -Math.pow((11 - 6 * b - 11 * a) / 4, 2) + Math.pow(c, 2); break } a = void 0 } return 1 - a }, elastic: function(a) { a = 1 - a; return 1 - Math.pow(2, 10 * (a - 1)) * Math.cos(30 * Math.PI / 3 * a) }, G = null; a.lineCap = "round"; this.draw = function() { if (!A.i8 d) { B.clearRect(-t, -u, C, D); B.save(); var g = { ctx: a }; a = B.p(); N(); J(); d(); s(); b.title && (a.save(), a.font = 24 * (f / 200) + "px Arial", a.fillStyle = b.colors.title, a.textAlign = "center", a.fillText(b.title, 0, -f / 4.25), a.restore()); b.units && (a.save(), a.font = 22 * (f / 200) + "px Arial",

```

- ❖ `a.fillStyle=b.colors.units,a.textAlign="center",a.fillText(b.units,0,f/3.25),a.restore();A.i8d=!`
`0;a=g.ctx;delete g.ctx}a.clearRect(-t,-u,C,D);a.save();a.drawImage(A,-t,-u,C,D);if(Gauge.initialized)L(),K(),H||(E.onready&&E.onready(),H=!0);else var e=setInterval(function(){Gauge.initialized&&(clearInterval(e),L(),K(),H||(E.onready&&E.onready(),H=!0))},10);return this}};Gauge.initialized=!1;`
- ❖ `(function(){var b=document,l=b.getElementsByTagName("head")[0],q=-1!=navigator.userAgent.toLocaleLowerCase().indexOf("msie"),v="@font-face {font-family: 'Led';src: url('fonts/digital-7-mono."+q?"eot":"ttf")+";}";k=b.createElement("style");k.type="text/css";if(q)l.appendChild(k),l=k.styleSheet,l.cssText=v}else{try{k.appendChild(b.createTextNode(v))}catch(e){k.cssText=v}l.appendChild(k);l=k.styleSheet?k.styleSheet:k.sheet||b.styleSheets[b.styleSheets.length-1]}var g=setInterval(function(){if(b.body){clearInterval(g);`
- ❖ `var e=b.createElement("div");e.style.fontFamily="Led";e.style.position="absolute";e.style.height=e.style.width=0;e.style.overflow="hidden";e.innerHTML=".";b.body.appendChild(e);set`
`Timeout(function(){Gauge.initialized=!0;e.parentNode.removeChild(e)},250)},1)}();Gauge.Collection=[];`
- ❖ `Gauge.Collection.get=function(b){if("string"==typeof b)for(var l=0,q=this.length;l<q;l++){if((this[l].config.renderTo.tagName?this[l].config.renderTo:document.getElementById(this[l].config.renderTo)).getAttribute("id")==b)return this[l]}else return"number"==typeof b?this[b]:null};function domReady(b){window.addEventListener?window.addEventListener("DOMContentLoaded",b,!1):window.attachEvent("onload",b)}`
- ❖ `domReady(function(){function b(b){for(var e=b[0],d=1,g=b.length;d<g;d++)e+=b[d].substr(0,1).toUpperCase()+b[d].substr(1,b[d].length-1);return e}for(var l=document.getElementsByTagName("canvas"),q=0,v=l.length;q<v;q++)`
- ❖ `if("canv-gauge"==l[q].getAttribute("data-type")){var k=l[q],e={g,p=parseInt(k.getAttribute("width"),10),w=parseInt(k.getAttribute("height"),10);e.renderTo=k;p&&(e.width=p);w&&(e.height=w);p=0;for(w=k.attributes.length;p<w;p++)if(g=k.attributes.item(p).nodeName,"data-type"!=g&&"data-"==`
- ❖ `g.substr(0,5)){var d=g.substr(5,g.length-5).toLowerCase().split("-");if(g=k.getAttribute(g))switch(d[0]){case "colors":d[1]&&(e.colors||(e.colors={}),"needle"==d[1]?(d=g.split(/\s+/),e.colors.needle=d[0]&&d[1]?{start:d[0],end:d[1]}:g):(d.shift(),e.colors[b(d)]=g));break;case "highlights":e.highlights||(e.highlights=[]);g=g.match(/(?:-?\d*\.)?(-?\d+){1,2}){2}(?:#(0x)?(?:[0-9A-F|a-f]){3,8}|rgba?(.*?))/g);for(var d=0,J=g.length;d<J;d++){var s=g[d].replace(/^\s+|\s+$/g,"").split(/\s+/),x={};`
- ❖ `s[0]&&"!"=s[0]&&(x.from=s[0]);s[1]&&"!"=s[1]&&(x.to=s[1]);s[2]&&"!"=s[2]&&(x.color=s[2]);e.highlights.push(x)}break;case "animation":d[1]&&(e.animation||(e.animation={}),"fn"==d[1]&&/\s*function\s*\(/.test(g))&&(g=eval("("+g+"")),e.animation[d[1]]=g);break;default:d=b(d);if("onready"==d)continue;if("majorTicks"==d)g=g.split(/\s+/);else if("strokeTicks"==d||"glow"==d)g="true"==g?!0:!1;else if("valueFormat"==d)if(g=g.split("."),2==g.length)g={"int":parseInt(g[0],10),dec:parseInt(g[1],10)};else continue;`
- ❖ `e[d]=g}e=new Gauge(e);k.getAttribute("data-value")&&e.setRawValue(parseFloat(k.getAttribute("data-value")));k.getAttribute("data-`

```
onready")&&(e.onready=function(){eval(this.config.renderTo.getAttribute("data-
onready"))});e.draw()});window.Gauge=Gauge;
```

❖ <!-- Extremos del código de calibre -->

- function GetArduinoIO() // declara una función que devuelve los datos de GetArduinoIO
- {
 - nocache = "&nocache=" + Math.random() * 1000000; //especifica directivas (instrucciones) para almacenar temporalmente (caching) tanto en peticiones como en respuestas.
 - var request = new XMLHttpRequest(); // intercambiar datos entre el sitio web y el servidor
 - request.onreadystatechange = function() //es invocado cada vez que cambia el atributo readyState
 - {
 - if (this.readyState == 4) {
 - if (this.status == 200) {
 - if (this.responseXML != null) {
 - // Archivo XML recibido: contiene valores analógicos, valores de cambio y estados del Generador
 - var count; // es una variable contadora
 - // obtener entradas analógicas
 - data_val2 = this.responseXML.getElementsByTagName('analog')[0].childNodes[0].nodeValue; // asigna el dato enviado de arduino con analog0 a la variable data_val2
 - data_val3 = this.responseXML.getElementsByTagName('analog')[1].childNodes[0].nodeValue; // asigna el dato enviado de arduino con analog1 a la variable data_val3
 - //obtener entradas de interruptor
 - var num_an = this.responseXML.getElementsByTagName('switch').length;
 - for (count = 0; count < num_an; count++) {
 - ❖ document.getElementsByClassName("switches")[count].innerHTML =
 - ❖ this.responseXML.getElementsByTagName('switch')[count].childNodes[0].nodeValue;
 - }
 - // ENCEDER/ APAGAR GENERADOR
 - if (this.responseXML.getElementsByTagName('INTER')[0].childNodes[0].nodeValue == "on") {
 - ❖ document.getElementById("SW1").innerHTML = "Encendido (APAGAR)";
 - SW1_state = 1;

- SW2_state = 0;
 - strSW2 = "&SW2=0";
 - }
 - else {
 - SW2_state = 1;
 - strSW2 = "&SW2=1";
 - }
 - }
- ❖ </script> <!-- cierra las instrucciones del script -->
- ❖ <!--contiene declaraciones de estilo para presentar la página WEB -->
- ❖ <style>
- .txt_block {
 - border: 2px solid black;
 - width: 285px;
 - height: 350px;
 - padding-left: 10px;
 - float: left;
 - margin: 0 20px 20px 0;}

 - .IO_box0 {
 - border: 0px solid black;
 - width: 935px;
 - padding: 0 5px 0 5px;
 - margin: 0 20px 20px 0;
 - text-align: center;
 - float: left;}

 - .IO_box {
 - border: 0px solid black;
 - width: 650px;
 - padding: 0 5px 0 5px;
 - margin: 0 20px 20px 0;
 - text-align: center;
 - float: left;}

 - .IO_box1 {
 - border: 1px solid blue;
 - width: 150px;
 - padding: 0 5px 0 5px;
 - margin: 0 20px 20px 0;
 - text-align: center;
 - float: left;}

 - .IO_box2 {

- border: 2px solid purple;
- width: 290px;
- padding: 0 5px 0 5px;
- margin: 0 20px 20px 0;
- float: left;}

- .IO_box3 {
- border: 1px solid black;
- width: 220px;
- padding: 0 5px 0 5px;
- margin: 0 20px 20px 0;
- text-align: center;
- float: left;}

- h1 {
- font-size: 120%;
- color: blue;
- margin: 0 0 10px 0;}

- h2 {
- font-size: 85%;
- color: #5734E6;
- margin: 5px 0 5px 0;}

- p, form, button {
- font-size: 80%;
- color: #252525;
- }
- .small_text {
- font-size: 70%;
- color: #737373;
- }

- ❖ </style> <!--cierra el formato de los estilos de se usan en la pagina -->
- ❖ </head> <!--cierra el cuerpo de la página, informacion no visible por el usuario -->

- ❖ <!-- Encierra todo el contenido que deseas mostrar a los usuarios web que visiten la página -->
- ❖ <body onload="GetArduinoIO()"> <!-- toma los datos de la funcion GetArduinoIO--> <!-- -->
- ❖ <!-- muestra en la web el titulo de la pagina con el estilo especificado -->
- ❖ <div class="IO_box0" style="background:#308446;padding:15px 15px;">
- ❖ <h1 style="text-align: center;"><font style="vertical-align:

inherit;">SISTEMA DE MONITOREO DEL SUMINISTRO DE ENERGIA ELECTRICA</h1>

- <div class="IO_box" style= "center; background:#d7ead8 ">
 <h1>Indicador del suministro de energia</h1>
- <div class="IO_box2" style= "center; background:#fec4b1 "> <h3>Red publica
- <div class="IO_box1"> <h1 style="text-align: center"> </h1>
- </div> </h3>
- </div>
- <div class="IO_box2" style= "center; background:#d9ccda "> <h3 >Generador
- <div class="IO_box1"> <h1 style="text-align: center;"> </h1>
- </div> </h3>
- </div>
- ❖ <!-- indica el nivel de combustible en el gauge -->
- ❖ <div class="txt_block" > <h3 style="text-align: center">Nivel de Combustible</h3>
- ❖ <canvas id="an_gauge_1" data-title="Combustible" data-units="Nivel °%" width="280" height="280" data-major-ticks="0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100" data-type="canv-gauge" data-min-value="0" data-max-value="100" data-highlights="0 25 #f33, 25 45 #cc5, 45 70 #25B8D9, 70 100 #0BB950" data-onready="setInterval(function() { Gauge.Collection.get('an_gauge_1').setValue(data_val2);}, 200);"></canvas> </div>
- ❖ <!-- indica el nivel de energía en el gauge -->
- ❖ <div class="txt_block"> <h3 style="text-align: center;">Nivel del SAI </h3>
- ❖ <canvas id="an_gauge_2" data-title="Bateria" data-units="Energia %" width="280" height="280" data-major-ticks="0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100" data-type="canv-gauge" data-min-value="0" data-max-value="100" data-highlights="0 20 #f33, 20 35 #cc5, 35 60 #25B8D9, 60 100 #0BB950" data-onready="setInterval(function() { Gauge.Collection.get('an_gauge_2').setValue(data_val3);}, 200);"></canvas> </div>
- </div>
- ❖ <!-- Crea el botón de control del generador -->
- <div class="IO_box3" style="text-align: center;background:#cfe7eb">
-

- <h3> Control del generador</h3>
- <button type="button" id="SW1" onclick="GetButton1()">PRENDER/APAGAR</button>

<hr>
-

- <!-- Crea el botón de control del sistema de alerta -->
- <h3>Sistema de Alerta </h3>

- `<button type="button" id="SW2" onclick="GetButton2()">Activar/Desactivar</button>`
- `

<hr>`
- `

`
- `<h1>NOTA</h1>`
- `<div class="IO_box1" style="text-align: left;background:#fff163">`
- `<h2>Para ingresar al Registro de Funcionamiento marcar el IP administrador </h2>`
- `</div>`

- `</div>`

- `</div>`

- ❖ `</body>` `<!-- Cierra la función del cuerpo de la pagina que se va a mostrar en la pagina web -->`
- ❖ `</html>`

ANEXO 4. PROCESO DE ELABORACION DEL HARDWARE



Figura 88. Proceso químico para quitar residuos de cobre de la placa PCB.
Fuente: Autor

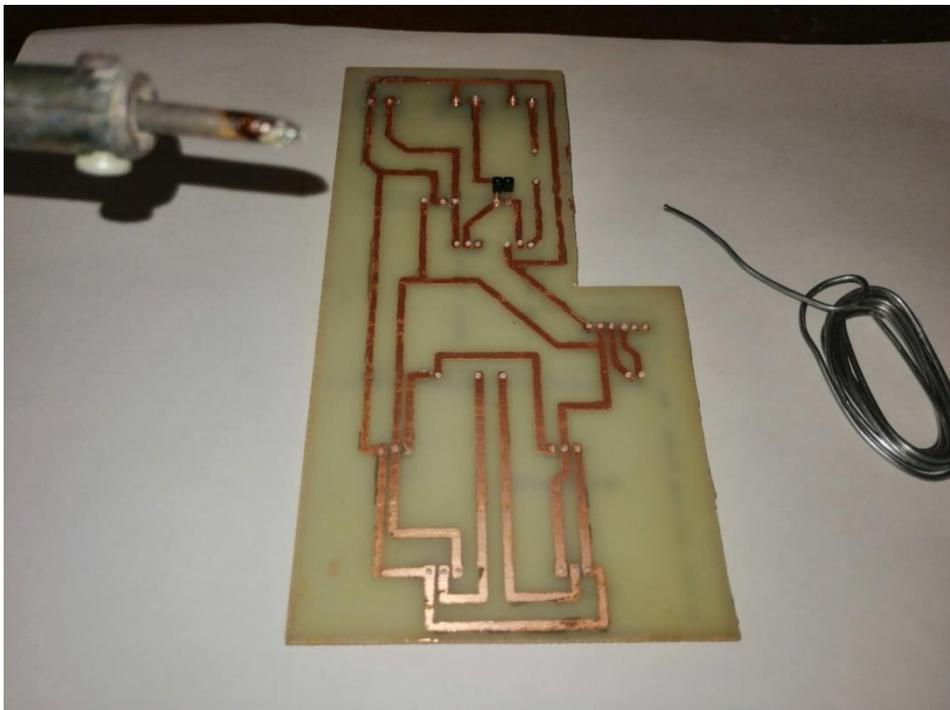


Figura 89. Proceso de soldado de los componentes externos a la PCB.
Fuente: Autor

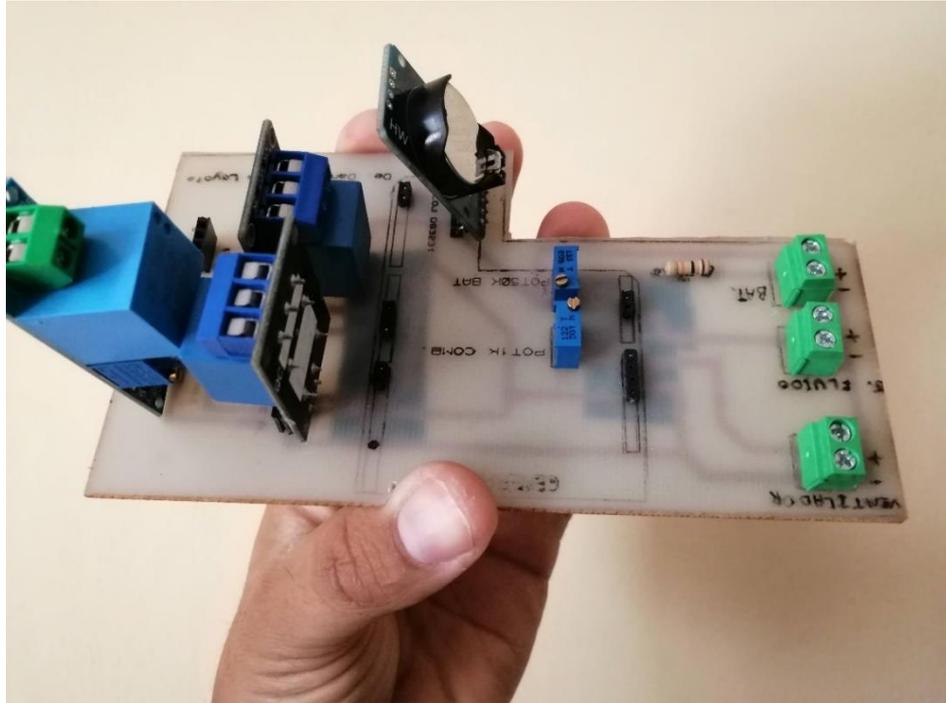


Figura 90. Placa de circuito impreso que permite adquirir información de variables físicas de un sistema de respaldo de energía eléctrica.

Fuente: Autor



Figura 91. Ensamblado del equipo de monitoreo y control de un generador eléctrico y del SAI.

Fuente: Autor