



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA
E INDUSTRIAL**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
COMUNICACIONES**

TEMA:

**“SISTEMA ELECTRÓNICO PARA EL MONITOREO DEL
ALMACENAMIENTO DE DIESEL DEL HOSPITAL GENERAL AMBATO”**

Trabajo de graduación, modalidad: Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Electrónica y Comunicaciones.

SUBLÍNEA DE INVESTIGACIÓN: Comunicaciones Inalámbricas

AUTOR: Oscar Iván Analuisa Criollo

TUTOR: Ing. Mg. Geovanni D. Brito Moncayo

AMBATO – ECUADOR


2018

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: "SISTEMA ELECTRÓNICO PARA EL MONITOREO DEL ALMACENAMIENTO DE DIESEL DEL HOSPITAL GENERAL AMBATO", del ANALUISA CRIOLLO OSCAR IVAN, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los tramites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato Febrero, 2018

TUTOR

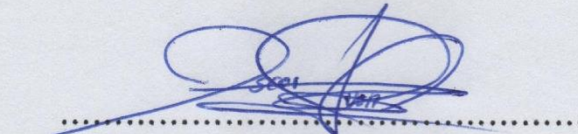


Ing. Mg. Giovanni Brito

AUTORÍA

El presente Proyecto de Investigación titulado “SISTEMA ELECTRÓNICO PARA EL MONITOREO DEL ALMACENAMIENTO DE DIESEL DEL HOSPITAL GENERAL AMBATO”, es absolutamente original, autentico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Febrero, 2018



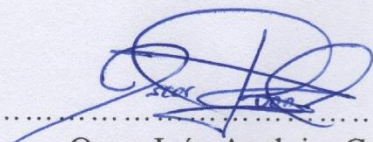
Oscar Iván Analuisa Criollo
C.C. 1804750113

DERECHO DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

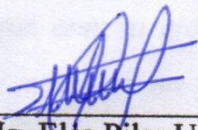
Ambato Febrero, 2018




.....
Oscar Iván Analuisa Criollo
C.C. 1804750113

APROBACIÓN DE LA COMISIÓN CALIFICADORA


La Comisión Calificadora del presente trabajo conformada por los señores docentes Ingeniero Marco Antonio Jurado Lozada y el Ingeniero Carlos Alberto Serra Jiménez, revisó y aprobó el Informe Final del Proyecto de Investigación titulado "SISTEMA ELECTRÓNICO PARA EL MONITOREO DEL ALMACENAMIENTO DE DIESEL DEL HOSPITAL GENERAL AMBATO", presentado por el señor Oscar Analuisa de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Mg. Elsa Pilar Urrutia Urrutia
PRESIDENTA DEL TRIBUNAL



Ing. Marco Antonio Jurado Lozada
DOCENTE CALIFICADOR



Ing. Carlos Alberto Serra Jiménez
DOCENTE CALIFICADOR

Oscar Iván Analuisa Criollo

DEDICATORIA

A mi madre por brindarme siempre su apoyo incondicional y por ofrecerme la posibilidad de estudiar, la que siempre me ha dado palabras de aliento, la que con su ejemplo de lucha y superación se ha convertido en un pilar fundamental en mi desarrollo académico y personal.

Oscar Iván Analuisa Criollo

AGRADECIMIENTO

A Dios, por haberme dado la vida y en los momentos difíciles de mi vida bendecirme para no rendirme y seguir adelante.

Al Ing. Geovanni Brito que en calidad de tutor guió este proyecto de investigación y por haber compartido sus conocimientos en mi etapa universitaria en la FISEI.

Al Hospital General Ambato por la apertura y apoyo brindado para la culminación de este proyecto.

Finalmente, a mi madre y a todas las personas que me brindaron su apoyo, quiero expresarles un infinito agradecimiento y recordarles que ocupan un lugar especial en mi corazón.

Oscar Iván Analuisa Criollo

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR	I
AUTORÍA	II
DERECHO DE AUTOR	III
APROBACIÓN DE LA COMISIÓN CALIFICADORA	IV
DEDICATORIA.....	V
AGRADECIMIENTO	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS.....	XIV
RESUMEN	XV
ABSTRACT	XVI
INTRODUCCIÓN.....	XVII
CAPÍTULO 1	1
EL PROBLEMA.....	1
1. Tema	1
1.2. Planteamiento del Problema	1
1.3. Delimitación.....	2
1.4 Justificación	3
1.5 Objetivos	4
1.5.1 General	4
1.5.2 Específicos:	4
CAPÍTULO 2	5

Marco Teórico	5
2.1 Antecedentes Investigativos.....	5
2.2 Fundamentación Teórica.....	7
2.2.1 Hospital General Ambato.....	7
2.2.2 El Hospital como Empresa.....	8
2.2.3 Medición de nivel.....	9
2.2.4 Tanque de almacenamiento de diésel.....	20
2.2.5 Pantallas HMI.....	20
2.2.6 Sistemas electrónicos de comunicación	21
2.2.7 Ethernet	22
2.2.8 Controlador Lógico Programable.....	23
2.2.9 El Internet de las cosas (IoT).....	23
2.2.10 Plataforma de desarrollo móvil	24
2.2.11 Lenguaje de programación C++.....	24
2.3 Propuesta de Solución.....	25
CAPÍTULO 3	26
METODOLOGÍA.....	26
3.1 Modalidad de investigación	26
3.2 Recolección de información	26
3.3 Procesamiento y análisis de datos.....	27
3.4 Desarrollo del proyecto.....	27
CAPÍTULO 4	28
DESARROLLO DE LA PROPUESTA	28
4.1 Introducción	28
4.2 Factibilidad del proyecto.....	28

4.2.1 Factibilidad económica	29
4.2.2 Factibilidad bibliográfica	29
4.2.3 Factibilidad técnica	29
4.3 Condiciones actuales y consumo de los tanques de almacenamiento.....	29
4.3.1 Datos técnicos de los tanques almacenamiento de diesel	30
4.4 Analizar el control actual de nivel de diesel en el sistema de Almacenamiento.	33
4.5 Requerimientos técnicos para diseño por parte del personal de mantenimiento.	34
4.6 Diagrama de Bloques del Sistema	35
4.7 Selección de elementos para el diseño y construcción del Sistema de Monitoreo. ...	35
4.7.1 Sensor de Nivel	36
4.7.2 Selección de la tarjeta de adquisición de datos	42
4.7.3 Visualización Local.....	45
4.7.4 Comunicación.....	46
4.7.5 Plataforma en la nube.....	47
4.8 Selección del entorno de desarrollo	52
4.9 Diseño de los circuitos electrónicos del sistema de monitoreo.....	53
4.10 Requerimientos Técnicos para el Diseño del Sistema	56
4.11 Visualización en Tiempo Real	57
4.12 Diseño Electrónico del Sistema de Monitoreo.....	59
4.13 Procesamiento de datos del sensor de nivel.....	61
4.14 Configuración de plataforma Thinger.io.....	63
4.15 Configuración Firebase	72
4.16 Diseño de aplicación para Android.....	78
4.17 Implementación Interface HMI.....	82
4.18 Implementación del Sistema	89
4.19 Ajustes del Sistema y Pruebas de Funcionamiento.....	92

4.19.1 Calibración y Ajustes del Sistema de Medición.....	93
4.19.2 Pruebas de funcionamiento	94
4.20 Funcionamiento del prototipo	98
4.21 Presupuesto.	100
CAPÍTULO 5	102
Conclusiones y Recomendaciones.....	102
5.1. Conclusiones	102
5.2 Recomendaciones	102
Bibliografía.....	104
ANEXOS	111

ÍNDICE DE FIGURAS

Figura 2.1: Mapa del Hospital General Ambato [13].....	7
Figura 2.2: Hospital General Ambato.....	8
Figura 2.3: Entrada principal del Hospital General Ambato	8
Figura 2.4: Clasificación de transductores de nivel.....	10
Figura 2.5: Medidor de cinta graduada con plomada [16]	11
Figura 2.6: Indicador de cristal [15]	12
Figura 2.7: Instrumentos de Flotador [15].....	12
Figura 2.8: Medidor de desplazamiento [15].....	13
Figura 2.9: Medidor manométrico [17]	14
Figura 2.10: Medidor de membrana [15].....	14
Figura 2.11: Medidor de trampa de aire [15].....	15
Figura 2.12: Medidor tipo burbujeo [17].....	15
Figura 2.13: Medidor de Presión Diferencial [15] [17].....	16
Figura 2.14: Medidor Resistivo [15] [16].....	16
Figura 2.15: Medidor de nivel capacitivo [19]	17
Figura 2.16: Medidor materiales conductivos [15]	18
Figura 2.17: Pantalla HMI (Interfaz Humano-Máquina) [23] [24]	21
Figura 2.18: Sistema Electrónico de Comunicaciones [25]	21
Figura 2.19: Controlador Lógico Programable [28].....	23
Figura 4.1: Tanques de almacenamiento de diesel	31
Figura 4.2: Generador Eléctrico marca Caterpillar modelo 3412	31
Figura 4.3: Caldero marca Superior Boiler Works.....	32
Figura 4.4: Visualización del nivel de diesel de los tanques de almacenamiento	34
Figura 4.5: Diagrama de Bloques del Sistema	35
Figura 4.6: Servicios de la plataforma thinger.io de la versión libre [46].....	49
Figura 4.7: Servicios que proporciona la plataforma firebase de la versión libre [33]	51
Figura 4.8: Diagrama de bloques de la fuente de alimentación DR-45 [59].....	53
Figura 4.9: Circuito que genera una señal de salida tipo lineal [51]	54
Figura 4.10: Diagrama esquemático de la placa MAX3232 [57]	55

Figura 4.11: Esquema de un conversor Análogo-digital	55
Figura 4.12: Diagrama de una señal aplicada un valor de offset [52]	57
Figura 4.13 : Diagrama de visualización del Nivel de Diesel	58
Figura 4.14: Diagrama de conexión del conector DB9 [36].....	59
Figura 4.15: Diagrama de conexión de los sensores de nivel.....	60
Figura 4.16: Diagrama de conexión de luces indicadoras	61
Figura 4.17: Diagrama de flujo del sistema de monitoreo	62
Figura 4.18: Icono Devices para agregar dispositivos.....	64
Figura 4.19: Detalles del dispositivo agregado	64
Figura 4.20: Lista de los dispositivos agregados	65
Figura 4.21: Tablero de información del dispositivo desconectado	65
Figura 4.22: Tablero de información del dispositivo conectado	66
Figura 4.23: Icono Data Bucketss para agregar contenedor de almacenamiento.....	66
Figura 4.24: Detalles requeridos para agregar un Bucket	67
Figura 4.25: Selección de “Samping Interval” para un muestreo de datos	68
Figura 4.26: Tabla de registro de información recibida del sensor de nivel	68
Figura 4.27: Opciones de importación de los datos del Bucket	69
Figura 4.28: Icono para crear interfaz de datos recibidos del transductor.....	69
Figura 4.29: Detalles requeridos para agregar un Dashboard	69
Figura 4.30: Selección de “Tme Sample Chart” para trabajar en función del tiempo	70
Figura 4.31: Selección de “Tme Sample Chart” para trabajar en función del tiempo	71
Figura 4.32: Icono para crear un destino para entrega de información	72
Figura 4.33: Detalles del endpoint creado para las notificaciones	72
Figura 4.34: Selección de añadir un proyecto	73
Figura 4.35: Detalles del proyecto.....	73
Figura 4.36: Opciones de sistema operativo de la aplicación móvil	74
Figura 4.37: Detalles para el registro de la app	74
Figura 4.38: Detalles de la descarga del archivo de configuración	75
Figura 4.39: Códigos para añadir en el código de la app	76
Figura 4.40: Códigos classpath para llamar a los servicios de google	77
Figura 4.41: Código final del gradle nos va a quedar de la siguiente manera.....	77
Figura 4.42: Visualización del proyecto creado en firebase.....	78

Figura 4.43: Pantalla del sistema de monitoreo desarrollada en Android Studio.....	79
Figura 4.44: Interfaz para el usuario.....	81
Figura 4.45: Esquema de comunicación Pantalla HMI y el Controllino.....	83
Figura 4.46: Seleccionar el modelo de la OP320-A	84
Figura 4.47: Seleccionar el PLC de la OP320-A.....	84
Figura 4.48: Añadir o Borrar Screen	85
Figura 4.49: Atributos del Screen.....	85
Figura 4.50: Barra de Herramientas	86
Figura 4.51: Barra de funciones	86
Figura 4.52: Área de trabajo	87
Figura 4.53: Selección del registro 6 para el valor de galones del tanque 1	87
Figura 4.54: Selección del registro 7 para el valor del sensor del tanque 1	88
Figura 4.55: Sección de código de programación de la selección de los registros 6 y 7.....	88
Figura 4.56: Diseño del sistema de conexión de los elementos	90
Figura 4.57: Colocación los elementos en el riel din	90
Figura 4.58: Conexión MAX3232 y alimentación a la pantalla OP320-A	91
Figura 4.59: Distribución ordenada de los elementos en el riel din	91
Figura 4.60: Instalación de tres las luces indicadoras para las alertas.....	92
Figura 4.61: Panel frontal instaladas las tres luces indicadoras	92
Figura 4.62: Ingreso de parámetros y de alarmas para el tanque 1	93
Figura 4.63: Ingreso de los valores de offset y escala del sensor	93
Figura 4.64: Tanque de almacenamiento de diesel de 300 Gal.	94
Figura 4.65: Datos de transmisión.....	95
Figura 4.66: Panel de monitoreo del tanque 1	96
Figura 4.67: Recepción de alertas al móvil	96
Figura 4.68: Recepción de alertas en la pantalla HMI	97
Figura 4.69: Encendido de luz roja indicando nivel bajo del tanque	97
Figura 4.70: Prototipo completo del sistema de monitoreo de níveles de diesel.....	97
Figura 4.71: Estructura del case del prototipo	98
Figura 4.72: Pantalla HMI OP 320-S	99
Figura 4.73: Pantalla de configuración.....	99

ÍNDICE DE TABLAS

Tabla 4.1: Datos técnicos de los Tanques de Almacenamiento.	30
Tabla 4.2: Datos técnicos del generador eléctrico marca Caterpillar	32
Tabla 4.3: Datos técnicos del caldero marca Superior Boiler Works.....	33
Tabla 4.4: Características de los Transductores de Nivel [35]	36
Tabla 4.5: Selección de los transductores de nivel	38
Tabla 4.6: Parámetros en la selección de un transductor de nivel	40
Tabla 4.7: Transductores de nivel que cumplen las características	41
Tabla 4.8: Alternativas para selección de las tarjetas [39]	43
Tabla 4.9: Características técnicas del Controllino	44
Tabla 4.10: Matriz de ponderación de la pantalla	46
Tabla 4.11: Tecnologías de la Comunicación	46
Tabla 4.12: Plataformas en la nube	48
Tabla 4.13: Cuadro comparativo de los servidores de mensajería de notificaciones	50
Tabla 4.14: Herramientas utilizadas para el desarrollo de la aplicación	52
Tabla 4.15: Conexión de Pines del controllino con conversor	58
Tabla 4.16: Pines de entrada de los sensores de nivel	60
Tabla 4.17: Características de la pantalla OP 320-S [54].....	82
Tabla 4.18: Tabla de registros para establecer la comunicación Modbus	88
Tabla 4.19: Datos recibidos del sensor	95
Tabla 4.20: Presupuesto del prototipo del sistema de monitoreo	100
Tabla 4.21: Presupuesto del Sistema Completo	101

RESUMEN

En la presente investigación se desarrolló un sistema encaminado a mejorar el monitoreo del nivel de diesel en el Hospital General Ambato. El diseño y funcionamiento del sistema ofrece un monitoreo en tiempo real y cuenta con alertas a un dispositivo móvil cuando algún tanque se encuentre en un nivel medio o bajo. De esta manera mantener un nivel apropiado en cada uno de los tanques de almacenamiento, ya que por ningún motivo estos pueden quedar vacíos porque podrían inhabilitar diferentes áreas del hospital.

La metodología del diseño está basado en el monitoreo local a través de una pantalla HMI y remotamente por medio de la plataforma en la nube. Las alertas de escasez de diesel se visualizan por medio de luces indicadoras, sonoras y de forma remota con notificaciones a una aplicación desarrollada en Android Studio. El sistema de monitoreo está diseñado para aplicarlo a los cuatro tanques que conforma el almacenamiento total de diesel y el prototipo está instalado en un tanque, el cual alimenta al generador eléctrico.

Palabras claves

Almacenamiento de Diesel, Controllino, HMI, Sensores de nivel.

ABSTRACT

In the present investigation, a system was developed to improve the monitoring of the diesel level in the Ambato General Hospital. The design and operation of the system offers a real-time monitoring, It has alerts to a mobile device when a tank is at a medium or low level. In this way maintain an appropriate level in each of the storage tanks, because by no means these can be empty because they could disable different areas of the hospital.

The design methodology is based on local monitoring through an HMI screen and remotely through the cloud platform. The diesel shortage alerts are displayed by means of indicator lights, sound and remotely with notifications to an application developed in Android Studio. The monitoring system is designed to apply to the four tanks that make up the total storage of diesel and the prototype is installed in a tank, which feeds the electric generator.

Key words

Controllino, Diesel storage, HMI, Level sensors.

INTRODUCCIÓN

Actualmente, la alta demanda de los sistemas de monitoreo tiene como propósito tomar acciones preventivas de acuerdo con los datos proporcionados por estos sistemas y proyectar el resultado de los mismos.

Se desarrolló en el presente proyecto de investigación un sistema electrónico de monitoreo para el almacenamiento de diesel, con la finalidad de monitorear los niveles de cada tanque y recibir alertas ya sea de niveles medio y bajos, con dos facilidades de visualización una en forma local y la otra en forma remota.

El presente proyecto muestra los parámetros técnicos de desarrollo y construcción del prototipo, así como las pruebas realizadas de su funcionamiento e instalación en campo.

El trabajo de investigación consta de cinco capítulos estructurados sistemáticamente de la siguiente manera:

Capítulo I

Se describe información sobre las características, problemas y la situación actual en que se realiza el monitoreo del nivel de diesel en el Hospital General Ambato, para así conocer los principales requerimientos.

Capítulo II

En este capítulo se detalla los antecedentes investigativos de temas relacionados con el sistema de monitoreo, además la fundamentación teórica para sustentar el desarrollo del mismo y la propuesta con la cual se dará solución al problema.

Capítulo III

Dentro de este capítulo tenemos la metodología utilizada para la diseño, ejecución e instalación del proyecto.

Capítulo IV

Se presenta el diseño, implementación e instalación en campo del prototipo de monitoreo de nivel del almacenamiento de diesel.

Capítulo V

Finalmente, en esta sección se presentan las conclusiones y recomendaciones del prototipo realizado para el Hospital General Ambato.

CAPÍTULO 1

EL PROBLEMA

1. Tema

SISTEMA ELECTRÓNICO PARA EL MONITOREO DEL ALMACENAMIENTO DE DIESEL DEL HOSPITAL GENERAL AMBATO

1.2. Planteamiento del Problema

La Organización Mundial de la Salud (OMS) señala en sus documentos técnicos que es indispensable en un programa de mantenimiento adoptar precauciones en la gestión de presupuesto y control de costos. Los rubros y documentos de órdenes de servicio a proveedores externos deben estar controlados con precisión de tiempo. A pesar de que la gestión de presupuesto es similar a cualquier otra institución es un punto de referencia del programa de gestión. Lo cual es importante ajustar el presupuesto porque los costos pueden ser impredecibles y así evitar un desequilibrio presupuestario considerable. [1]

En el documento técnico “Evolución del Mantenimiento Industrial”, es considerado un éxito el abastecimiento de suficiente combustible, esta actividad está inmersa en la reingeniería del mantenimiento preventivo. El departamento encargado de la adquisición de estos recursos debe tener una buena coordinación con el objetivo de evitar paradas totales de procesos y generar sobrecostos derivados de las compras urgentes. [2]

La publicación del Centro Ingeniería Clínica y Gestión de Tecnología en Salud (CENGETS) menciona que el desarrollo tecnológico es una estrategia que permitirá crear nuevas alternativas para mejorar la infraestructura hospitalaria. La ingeniería clínica tiene interrelación con los servicios y recursos ofrecida por terceros, a fin de garantizar una mejor

relación costo/beneficio y seguridad, lo que orienta a encontrar la tecnología apropiada para regulación de dispositivos, programas de mantenimiento preventivo y correctivo según el área. [3]

El Plan Nacional del Buen Vivir menciona en el objetivo número 3 el “Mejoramiento de la calidad de vida de la población”, por esta razón se ha dado un énfasis al mejoramiento de la atención y cuidado de los pacientes y como resultado ha dado una demanda creciente en el uso de equipos y la implementación de sistemas de automatización en los hospitales. [4]

El Ministerio de Salud Pública del Ecuador según el acuerdo ministerial en el estatuto de orgánico de gestión organizacional por procesos de los hospitales, confiere el subproceso de mantenimiento, asignando un plan de mantenimiento preventivo y correctivo de los bienes muebles, inmuebles, equipos de electro medicina a cargo del hospital. Otra actividad mencionada son los proyectos de infraestructura del Hospital de acuerdo con las directrices del Ministerio de Salud Pública (MSP). [5]

Una problemática surge debido a que el personal del área de mantenimiento realiza múltiples tareas y resuelve inconvenientes que se presentan inesperadamente. El monitoreo y los diferentes requerimientos hacen que el mantenimiento se convierta en un proceso desordenado e incluso ciertas actividades tienden a ser olvidadas. [6]

El Hospital General Ambato gestiona la adquisición de diesel con Petroecuador por medio del área de mantenimiento y la dirección administrativa, por lo cual una de las actividades del personal del área es revisar constantemente el nivel de los tanques que conforman el sistema de almacenamiento de diesel. La carencia de diésel dejaría inoperativo los calderos y el generador de energía eléctrica que es base de funcionamiento de múltiples áreas dentro del hospital. El consumo de este combustible se incrementa debido al alto índice de pacientes registrado actualmente. El financiamiento en la adquisición de recursos requiere de una alta demanda en el tiempo y si no se lleva un control de los mismos evitará tomar acciones inmediatas que puede genera gastos inesperados, provocando contratiempos para la gerencia del hospital. [7]

1.3. Delimitación

Área académica de la carrera: Comunicaciones.

Línea de investigación: Tecnologías de la Información y Comunicación

Sublínea de investigación: Comunicaciones Inalámbricas

Delimitación Espacial: El presente proyecto de investigación se realizó para el sistema de diésel en el área de infraestructura del Hospital General Ambato del Instituto Ecuatoriano de Seguridad Social.

Delimitación Temporal: El proyecto de investigación se desarrolló desde Marzo 2017– Febrero 2018 de acuerdo a lo establecido en el Reglamento de Graduación para la obtención de título de tercer nivel de la Universidad Técnica de Ambato

1.4 Justificación

El Hospital General Ambato del IESS es una institución que protege al afiliado y su familia en las contingencias de enfermedad y maternidad, por tal motivo es necesario y primordial el funcionamiento de toda la infraestructura que dan soporte a esta actividad, no siendo aceptable que éstos se encuentren inhabilitados o fuera de servicio por falta de una programación o la falta de recursos.

La importancia del proyecto de investigación radica en mantener los tanques de almacenamiento de diésel en niveles apropiados, debido a que los calderos y el generador de energía eléctrica trabajan con este combustible y por ningún motivo estos equipos pueden dejar de funcionar porque esto puede provocar un paro total del hospital, generando diversos problemas como resultado de este imprevisto. La necesidad de un monitoreo, así como también las alertas juega un papel muy importante debido a que es necesario brindar un mantenimiento a tiempo sin dejar de lado ningún tanque vacío. Todo esto satisface los requerimientos de la Unidad Médica tomando acciones oportunas, agilizando las rutinas del personal de mantenimiento, corrigiendo los errores y reduciendo el tiempo estimado de provisión de combustible.

Un sistema electrónico de monitoreo utiliza tecnología de vanguardia, de tal manera que los beneficiarios directos del proyecto de investigación son principalmente los asegurados y/o afiliados al Instituto Ecuatoriano de Seguridad Social Ambato debido a la atención permanente que reciben, también de manera indirecta el personal que labora en el área de

mantenimiento, agilizando las rutinas de trabajo y finalmente el prestigio al mantener el hospital con un continuo mejoramiento en su infraestructura.

El proyecto de investigación fue factible por el respaldo de diversas fuentes de información primaria y secundaria respectivamente, documentos técnicos de la OMS, los estatutos del ministerio de salud pública, Reglamento Orgánico del Instituto Ecuatoriano de Seguridad Social, artículos científicos con reglamentos y normas de mantenimiento, guías, tutoriales, etc.

Se contó también con el respaldo del coordinador y asistentes del área de mantenimiento, al analizar el beneficio de un monitoreo que es indispensable en una de sus principales actividades da la pauta para la realización de este proyecto; primeramente, dando la apertura en sus instalaciones, facilitando todos los documentos necesarios para la realización del mismo, además brindan el apoyo y colaboración en las actividades dentro del hospital y de igual manera facilitando con toda la tecnología que el hospital cuenta, todo esto con el fin de para lograr la meta propuesta.

1.5 Objetivos

1.5.1 General

Construir un prototipo del sistema de monitoreo en uno de los tanques de almacenamiento de diésel en el Hospital General Ambato.

1.5.2 Específicos:

1. Analizar la situación actual de control y monitoreo de los tanques de diésel del Hospital General Ambato
2. Seleccionar los sensores y dispositivos electrónicos adecuados para la medición y monitoreo en los tanques de diésel.
3. Diseñar un sistema electrónico para el monitoreo del almacenamiento de diésel del Hospital General Ambato.

CAPÍTULO 2

Marco Teórico

2.1 Antecedentes Investigativos

En la presente sección se va a dar a conocer los diversos trabajos de investigación que se han realizado con respecto al monitoreo de sistemas de almacenamiento de diésel y a su respectivo mantenimiento; información que ofreció una guía para el desarrollo del proyecto de investigación. En pesquisas realizadas en papers, documentos técnicos, reglamentos hospitalarios, monografías, tesis; o trabajos de investigación del Ecuador y del mundo se encontró:

Marlene Cecilia Tonato en su tesis de pregrado, de la Universidad Técnica de Ambato, realiza un “Diseño e implementación de un sistema de monitoreo del nivel de combustible para los tanques de la central de generación en el Campo Cuyabeno Petroproducción”, donde se muestra información del monitoreo a través de una pantalla HMI diseñada en software in touch. Los resultados obtenidos son reflejados en la obtención de datos en tiempo real. La descripción, diagramas, datos técnicos, desarrollo de software fueron el pilar fundamental para la puesta en marcha del proyecto. Este proyecto fue referencia fundamental para la resolución de la propuesta. [8]

Luis Taqueño en su proyecto de pregrado, de la Escuela Politécnica de Chimborazo, bajo el tema: “Diseño de un plan de mantenimiento preventivo planificado para los equipos de la casa de máquinas del hospital general docente Riobamba”, propone primeramente definir las tareas de mantenimiento de los equipos y componentes dentro de un Hospital. Este proceso de gestión, monitoreo y planificación se lo realiza mediante una aplicación Web (software de desarrollo de mantenimiento), la licencia es pagada lo que sería una desventaja de trabajar con este software, pero la eficiencia de utilizar una interfaz web permite agilizar el trabajo y

un ahorro de tiempo muy importante para personal de mantenimiento de las instalaciones. [9]

Cesar Alberto Valencia en su proyecto de pregrado, de la Universidad Autónoma de Occidente en el año 2013, realiza el “Diseño de un sistema de monitoreo de nivel de los tanques de emergencia de Emcali Telecomunicaciones”, donde realiza el control de nivel utilizando un sensor de nivel tipo burbujeo conectándolo a través de un bus de campo hacia el PLC, también realiza el control de presión de tanques de gasolina y todo lo controla por medio de un sistema SCADA (Supervisión, Control y Adquisición de Datos), el resultado fue favorable en cuanto el sistema de comunicación fue capaz de visualizar los niveles de gasolina de los tanques utilizando la metodología expuesta, facilitando el control de combustible. [10]

En el proyecto de grado de Diego Danilo Vargas, de la Universidad Politécnica Nacional, propone el “Diseño e implementación de un sistema de control de combustión y monitoreo de gases generados en el caldero 3 del IESS hospital de Ambato”, lo realiza utilizando una pantalla HMI donde el operador puede observar el estado de las variables controladas, La monitorización ayuda al administrador a tomar acción de acuerdo el estado visualizado como lo recomienda el manual en lo referente a componentes y sistemas interconectados. Los resultados de este proyecto garantizan la visualización de las variables monitorizadas evitando riesgos para el personal. El monitoreo a realizarse tendrá el mismo principio de funcionamiento, con el beneficio de la pantalla que facilitará el chequeo de los niveles de cada uno de los tanques del sistema de almacenamiento. [11]

Claudio Danilo Acuña O. en su trabajo de titulación, de la Universidad de Magallanes, presenta la “Implementación de un Sistema de Control de Nivel para el Suministro de Petróleo de grupos Electrógenos”, monitoriza y automatiza consumo diario de petróleo utilizando un PLC que cuente con lenguaje Grafcet o SFC debido a su sencillez de programación, un sensor de nivel on-off debido a que utilizada un control PID. El autor concluye que la utilización de un PLC programado mediante SFC hace que el sistema sea robusto y facilita la traducción de forma directa el diagrama tipo escalera. [12]

2.2 Fundamentación Teórica

El siguiente apartado describe la fundamentación teórica inmersa en la tecnología de las comunicaciones, conjuntamente con los elementos destinados al desarrollo del sistema de monitoreo del sistema de almacenamiento de diésel.

2.2.1 Hospital General Ambato

El Hospital General Ambato es una unidad pública de salud tipificada como nivel II de acuerdo con la complejidad y prestaciones que se efectúan, forma parte de la red de servicios de salud de acuerdo al Modelo de Salud del Ecuador. Se rige por el Ordenamiento Jurídico Nacional y normativa propia a través de las Resoluciones del Consejo Directivo del IESS y de las disposiciones de la Dirección General. La figura 2.1, muestra la ubicación geográfica del hospital.

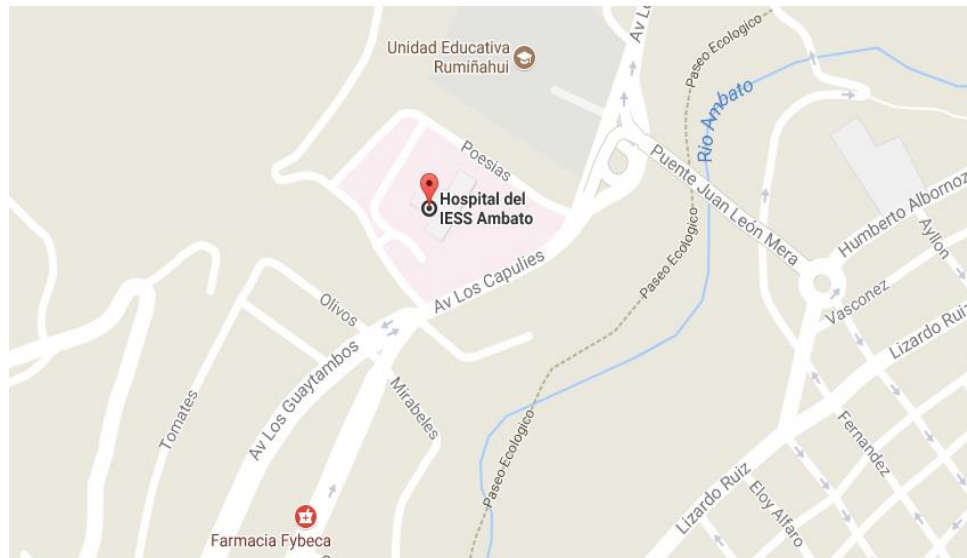


Figura 2.1: Mapa del Hospital General Ambato [13]

Su misión “El Hospital General Ambato es una entidad de atención médica moderna organizada e innovadora que se fundamenta en los principios de solidaridad, universalidad, equidad, eficiencia, ética que se encarga de prevención, promoción y recuperación de la salud de los usuarios, cuenta con un equipo multidisciplinario con gran capacidad y experiencia”. Esta casa de salud cuenta con ocho pisos, fue construida en el año de 1991, en la figura 2.2 se observa una fotografía panorámica del hospital y en la figura 2.3 muestra la entrada principal al Hospital General Ambato.



*Figura 2.2: Hospital General Ambato
Elaborado por: Oscar Analuisa*



*Figura 2.3: Entrada principal del Hospital General Ambato
Elaborado por: Oscar Analuisa*

2.2.2 El Hospital como Empresa.

Hospital es una Empresa compleja cuya dedicación es la Asistencia Médica o Atención de Servicios de Salud, enmarcado dentro de un Sistema retroalimentado por el afiliado.

La Empresa es una entidad social que coordina sus actividades para conseguir los objetivos de participación, e innovación en el mercado, de responsabilidad social, de rentabilidad, de productividad, de gestión y de formación, mediante la utilización de recursos humanos y materiales, con el fin de satisfacer una necesidad, produciendo un bien u ofreciendo un servicio. [7]

2.2.3 Medición de nivel

La medición de nivel es el conocimiento del nivel de un líquido dentro de un recipiente, con el objetivo de comprobar la cantidad de combustible en existencia, para determinar el volumen de un líquido que se suministra a un proceso, o bien puede ser la medición primaria en un sistema de regulación destinado a mantener el nivel en un recipiente que forma parte de un proceso continuo.

Sensores de nivel

Un sensor de nivel es un dispositivo diseñado para recibir información de una magnitud del exterior y transformarla en otra magnitud, normalmente eléctrica, que sea capaz de cuantificar y manipular. A partir de la medida de nivel del líquido acumulado en un tanque y conociendo la densidad de almacenamiento y geometría del mismo (dimensiones del tanque), puede determinarse el volumen y la masa del combustible. [14]

2.2.4 Métodos de Medición de Nivel

Los métodos más utilizados en la industria para la medición de nivel de líquidos pueden ser clasificados en métodos de medición directa e indirecta.

En el mapa conceptual de la figura 2.4, se observa de manera resumida los tipos de transductores de nivel de líquido.

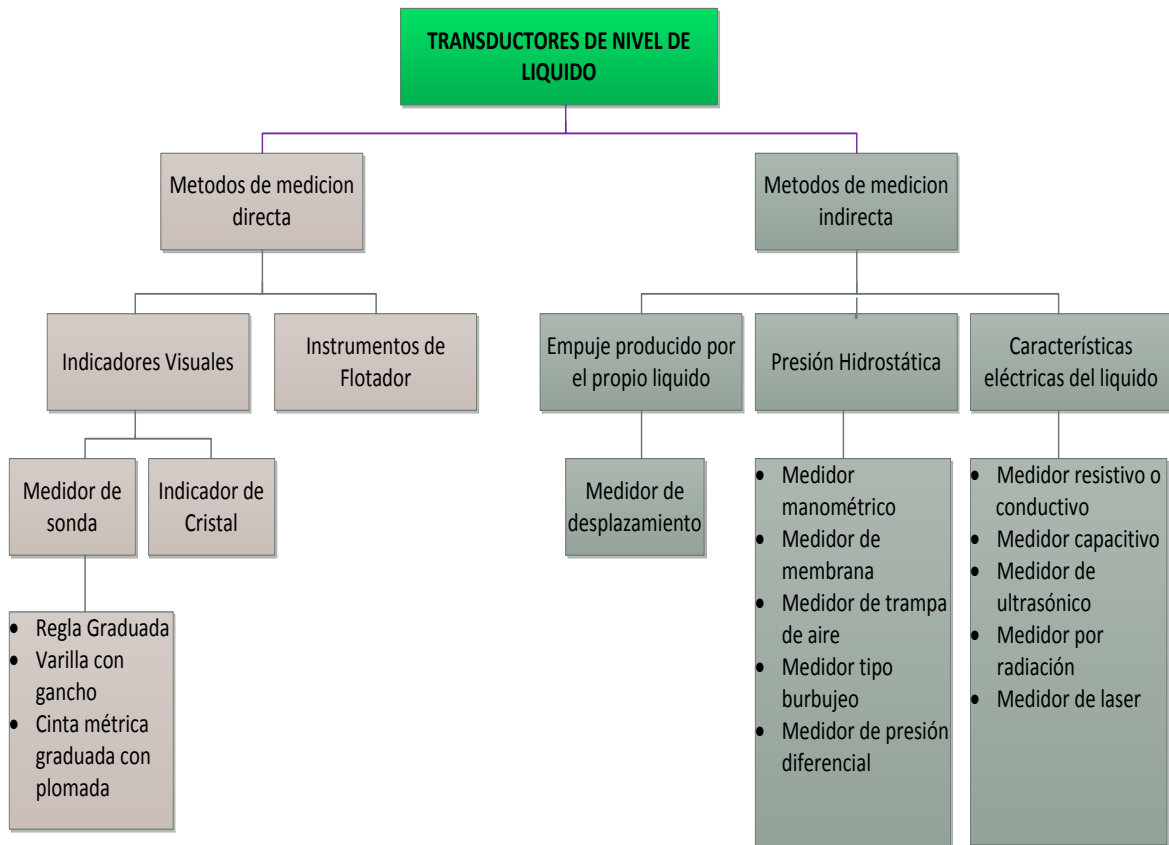


Figura 2.4: Clasificación de transductores de nivel

Elaborado por: Oscar Analuisa basado en [15]

Medición Directa

Los métodos de medición directa son aquellos que utilizan para la medición de nivel, la altura del líquido sobre una línea de referencia, entre los métodos de medición directa se tienen los indicadores visuales y los instrumentos de flotador.

Indicadores Visuales. - Los indicadores visuales permiten la observación directa de la altura de la superficie libre del líquido sobre una escala graduada apropiadamente.

Medidor de Sonda. - Los medidores de sonda se refieren a la utilización de aparatos mecánicos, generalmente operados de forma manual, los más utilizados son: Regla graduada, Varilla con gancho y cinta métrica metálica graduada con plomada.

La regla graduada se inserta verticalmente hacia el fondo del tanque. El nivel se determina leyendo directamente la longitud de la regla mojada por el líquido. En el momento de realizar

la medición, el tanque debe estar a presión atmosférica (el aire de la atmosfera debe entrar y salir del tanque libremente).

La varilla con gancho se sumerge en el seno del líquido y se levanta después, hasta que el gancho rompe la superficie del líquido. La distancia desde esta superficie hasta la parte superior del tanque representa indirectamente el nivel.

La cinta métrica metálica graduada con plomada se emplea cuando la regla graduada no tiene acceso al fondo del tanque. El instrumento está compuesto por tres partes principales: el carrete, la cinta graduada y un peso o plomada. La plomada sirve para que se mantenga la cinta tensa al penetrar en el líquido. En la figura 2.5 se observa el medidor de cinta graduada con plomada.

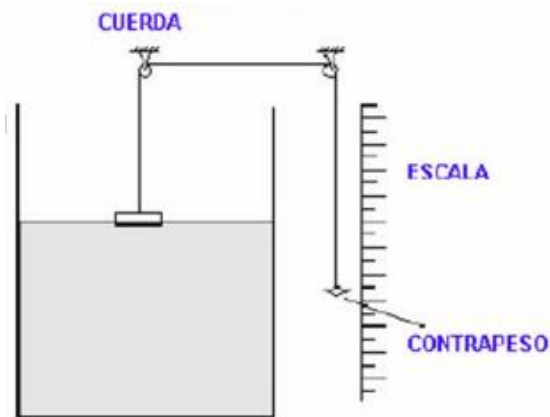


Figura 2.5: Medidor de cinta graduada con plomada [16]

Para medir el nivel, se deja que la cinta baje lentamente hasta que la plomada toque el fondo del recipiente. Una vez que la plomada toca el fondo se empieza a recoger la cinta con el carrete, hasta que aparezca la parte donde el líquido ha dejado la marca que indica su nivel. Se usan cuando la regla graduada no tenga acceso al fondo del tanque.

Indicador de cristal. – El indicador de cristal permite la visualización directa del nivel de tanques a través de un tubo indicador de vidrio, el funcionamiento del indicador de cristal se basa en el principio de los vasos comunicantes: con igual presión, el líquido del tanque sube en el tubo de vidrio hasta que ambos niveles sean iguales. Cuando el nivel varía en el tanque, varía también en el tubo de vidrio obteniéndose así una indicación real de nivel del proceso.

En la figura 2.6 se representa este método de medición y se puede observar que sirve tanto para tanques abiertos como cerrados.

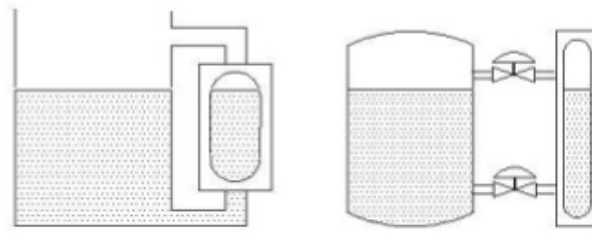


Figura 2.6: Indicador de cristal [15]

Instrumentos de flotador

Un instrumento de flotador es aquel transductor de nivel que se debe sumergir en el seno del líquido, el cual se mueve hacia arriba y hacia abajo con los cambios en el nivel, este movimiento del flotador puede ser transformado por diversos medios, en el exterior del tanque, en una acción de indicación, registro o control. El flotador se debe construir de tal manera que flote dentro del material como se observa en la figura 2.7 [15]

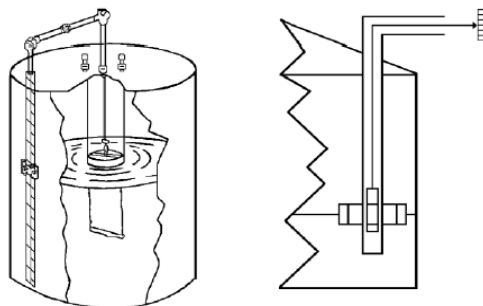


Figura 2.7: Instrumentos de Flotador [15]

Métodos de Medición Indirecta

Entre los métodos de medición indirecta para la medición de nivel de líquidos se tienen los que aprovechan el empuje producido por el propio líquido, la presión hidrostática y los que aprovechan las características del líquido.

Medidores que aprovechan el empuje producido por el propio líquido

Los medidores que aprovechan el empuje producido por el propio líquido son aquellos que utilizan el Principio de Arquímedes, el cual establece que un cuerpo (flotador) sumergido en

un líquido es empujado hacia arriba por una fuerza que es igual al peso del líquido desplazado, por ejemplo, el medidor de desplazamiento.

La ecuación utilizada para determinar la fuerza de flotación disponible es:

$$F = V \cdot s$$

donde:

F = Fuerza de flotación

V = Volumen del flotador

s = Gravedad específica del líquido

Medidor de desplazamiento. – En el medidor de desplazamiento la fuerza que actúa sobre el área (presión) del cuerpo sumergido crea la fuerza llamada flotabilidad, la cual permite a un cuerpo cuya densidad media sea inferior a la de un líquido, flotar parcialmente sumergido en la superficie libre del líquido. Un cuerpo al flotar pierde un peso equivalente al peso del volumen de líquido desplazado. En la figura 2.8 se muestra el funcionamiento del medidor de desplazamiento.

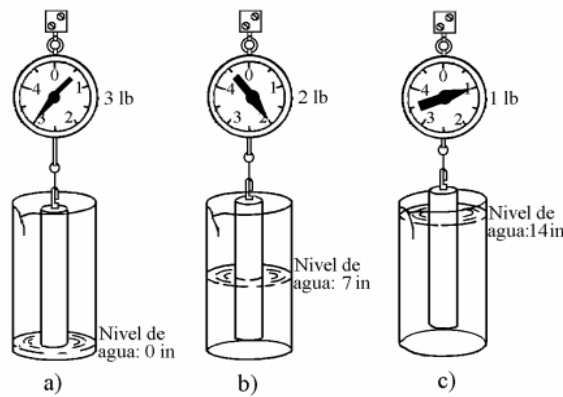


Figura 2.8: Medidor de desplazamiento [15]

Medidores que aprovechan la presión hidrostática

Los medidores que aprovechan la presión hidrostática son aquellos transductores que el nivel de líquido se puede medir por medio de la presión en un punto del fondo del tanque, ya que ésta es proporcional a la altura del líquido en ese punto y su peso específico, presión hidrostática (la presión de un fluido en reposo) ó cabezal estático.

Medidor manométrico. -El medidor manométrico como se conecta un manómetro en la línea de descarga de un tanque de almacenamiento, en la cual se pueden observar varios accesorios como son una válvula reguladora de flujo para el mantenimiento, y un pote de decantación con una válvula de purga, como se muestra en la figura 2.9.

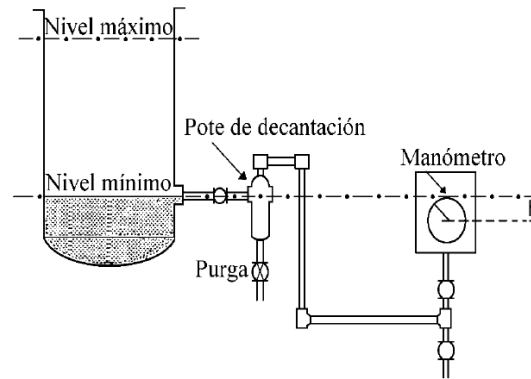


Figura 2.9: Medidor manométrico [17]

Medidor de membrana. - El medidor de membrana también es conocido como medidor de caja de diafragma, como se muestra en la figura 2.10. El sistema utiliza una membrana conectada con un tubo al instrumento receptor. El sistema se llena de aire a presión atmosférica antes de ser instalado. El aire atrapado dentro del sistema cerrado se comprime cuando el nivel sube debido a que la presión hidrostática aumenta en el fondo del recipiente, trayendo como consecuencia un empuje sobre el diafragma. Este aumento de presión es detectado por el indicador, el cual indicará, sobre una escala previamente calibrada, el nivel existente en el recipiente.

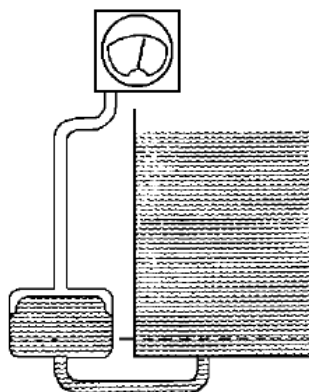


Figura 2.10: Medidor de membrana [15]

Medidor de trampa de aire. - El medidor de trampa de aire es muy similar al de la caja de diafragma, excepto que no se usa diafragma, figura 2.11. El medidor de presión opera por la presión de aire atrapado. Existe la posibilidad de pérdidas pequeñas de este aire, por lo que se debe proveer una toma auxiliar para recuperar el aire perdido cuando sea necesario.

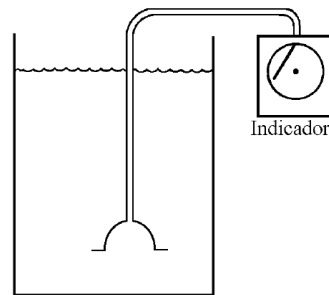


Figura 2.11: Medidor de trampa de aire [15]

Medidor tipo burbujeo. – El medidor de tipo burbujeo trabaja mediante un regulador de caudal se hace pasar por un tubo un pequeño caudal de aire o gas inerte hasta producir una corriente continua de burbujas. La presión requerida para producir el flujo continuo de burbujas es una medida de la columna de líquido. Este sistema es muy ventajoso en aplicaciones con líquidos corrosivos, ya que el fluido no penetra en el medidor ni en la línea de conexión. En la figura 2.12 se muestra el medidor de tipo burbujeo.

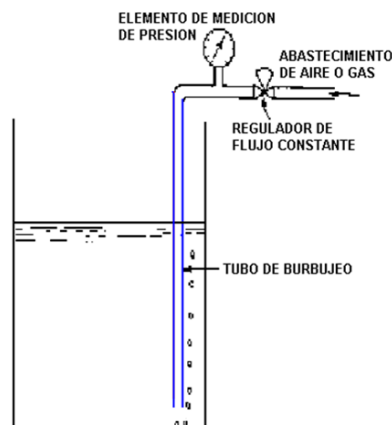


Figura 2.12: Medidor tipo burbujeo [17]

Medidor de Presión Diferencial. – El medidor de presión diferencial consiste en un diafragma en contacto con el líquido del tanque, que permite medir la presión hidrostática en un punto del fondo.

La presión es proporcional a la altura del líquido en ese punto y a su peso específico. El diafragma forma parte de un transmisor neumático o electrónico de presión diferencial.

La precisión de los instrumentos de presión diferencial es bastante buena tanto en tanques abiertos como en tanques presurizados. El medidor de presión diferencial se muestra en la figura 2.13.

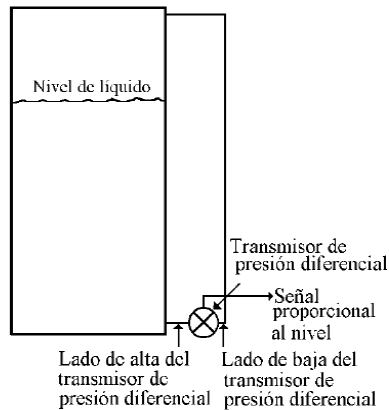


Figura 2.13: Medidor de Presión Diferencial [15] [17]

Medidores que aprovechan las características del líquido para la medición de nivel

Los medidores que aprovechan las características eléctricas para llevar a cabo su función, los medidores en esta categoría son los siguientes:

Medidor Resistivo. – El medidor resistivo consiste en uno o varios electrodos y un circuito electrónico que excita un relé, al ser los electrodos mojados por el líquido, cuando el líquido moja los electrodos se cierra el circuito electrónico y circula una corriente, el medidor resistivo abarca a los medidores conductivos por que los líquidos son conductores, el medidor resistivo se muestra en la figura 2.14.

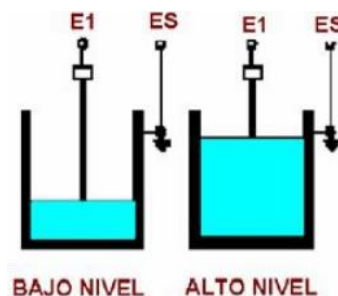


Figura 2.14: Medidor Resistivo [15] [16]

Medidor Capacitivo. - El medidor capacitivo tiene una de las placas del condensador está formada por un electrodo sumergido en el fluido y la otra está comprendida por las paredes del tanque, el dieléctrico es el del fluido dentro del tanque. El principio básico de funcionamiento del medidor de nivel capacitivo se basa en las propiedades que tiene un capacitor de almacenar cargas eléctricas y así, oponerse a cambios en el voltaje de un circuito. [18]

Un ejemplo básico de un medidor de nivel capacitivo se observa en la figura 2.15, donde el dieléctrico está sumergido dentro del tanque y la otra placa se conecta afuera, para que el líquido del contenedor trabaje como dieléctrico.

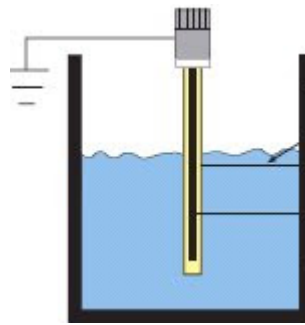


Figura 2.15: Medidor de nivel capacitivo [19]

La ecuación que define la capacitancia del medidor capacitivo es:

$$C = (K \cdot A) / D \quad \text{EC. 1}$$

donde:

C = Capacitancia

K = Constante dieléctrica

A = Área común entre las placas

D = Distancia entre las placas

Medidor ultrasónico

En el medidor de nivel por ultrasónico, las ondas sonoras viajan en un medio que absorbe el sonido y golpean a otro medio tal como una pared, una partícula en el líquido, o la superficie del líquido, solamente una pequeña porción de la energía de la onda sonora penetra la barrera

y el resto de la energía se refleja. La onda sonora reflejada es un eco. El medidor de ultrasonido utiliza el principio del eco para su funcionamiento.

La cantidad de energía reflejada depende del coeficiente de absorción de los materiales el cual se define como:

$$d = (\text{Energía absorbida})/(\text{Energía reflejada por el material})$$

Este valor depende de la frecuencia y propiedades del líquido tales como: porosidad, grosor del material y rigidez.

El diagrama de bloques que ilustra el principio de funcionamiento del medidor de ultrasonido se muestra en la figura 2.16.

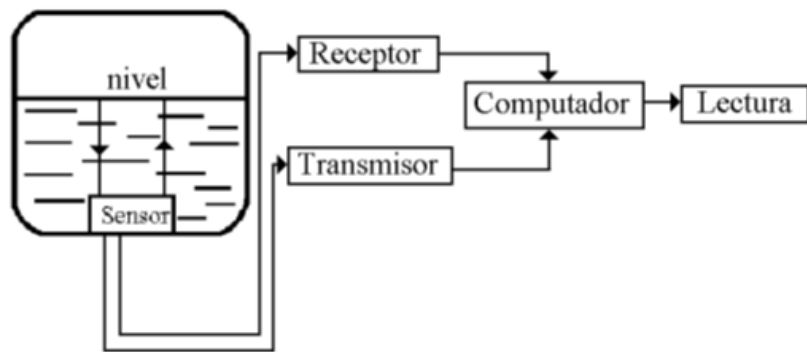


Figura 2.16: Medidor materiales conductivos [15]

El emisor dispone de un oscilador excitador para enviar un impulso ultrasónico a la superficie del fluido y el receptor, recibe esta señal reflejada enviando una señal función del tiempo transcurrido, y por lo tanto del nivel, a un indicador. En otras palabras, el nivel se mide en función del tiempo necesario para que la señal se desplace del transmisor a la superficie del líquido y retorne al receptor.

Ventajas

- Adecuados para todos los tipos de tanques y de líquidos.
- Muy exactos.
- Más sofisticados que los medidores convencionales.
- No poseen partes móviles.
- No requieren mantenimiento.
- No es intrusivo. [15]

Medidor por radar

El medidor por radar utiliza una onda electromagnética que es emitida por medio de una antena parabólica, esta señal rebota en el medio que se desea medir, de esta forma se puede obtener la variación de tiempo del rebote de la señal, determinando de esta manera el nivel.

Con el aumento del valor de los combustibles, en el último tiempo ha tomado especial importancia la medición con alta precisión del stock de estos productos. En general el volumen almacenado en estanques de combustible es obtenido indirectamente por medio de la medición de nivel. Por lo tanto, debido al gran diámetro de estos estanques (usualmente son de alrededor de 30 m) un error de algunos milímetros implica errores del orden de los miles de litros.

Para medir estos niveles se utiliza hace años radares de alta precisión con tecnología de modulación de frecuencia continua. Por otra parte, la tecnología de radar de pulsos se utiliza en la industria de procesos obteniendo precisiones del orden de 3mm.

Ventajas:

- Se pueden utilizar gabinetes tradicionales para los equipos sin requerir de certificaciones a prueba de explosión.
- Los técnicos pueden abrir, reparar, configurar o comprobar el estado del instrumento sin desconectar la energía debido a que utiliza sólo 24V y además la corriente es limitada por una barrera intrínseca.
- No se requieren canalizaciones o sellos a prueba de explosión para los conductos.
- Todo el cableado desde la barrera intrínseca hasta el radar puede ser instalado en bandejas o escalerillas lo cual implica ahorros importantes.
- La barrera puede ser instalada fuera del área clasificada en un gabinete tradicional. [11]
[15] [20]

Medidor de Láser

El medidor de laser envía un rayo a través de un tubo de acero y dirigido por reflexión en un espejo sobre la superficie del metal fundido. El aparato mide el tiempo que transcurre entre

el impulso emitido y el impulso de retorno que es registrado en un fotodetector de alta resolución, y este tiempo es directamente proporcional a la distancia del aparato emisor a la distancia a la superficie del metal en fusión, es decir, da la lectura del nivel. Se utiliza en aplicaciones donde las condiciones son muy duras, y donde los instrumentos de nivel convencionales fallan; tal es el caso de la medición de metal fundido, donde la medida del nivel debe realizarse sin contacto con el líquido y a la mayor distancia posible por existir unas condiciones de calor extremas. [21]

2.2.4 Tanque de almacenamiento de diésel

El tanque de diésel es un contenedor generalmente fabricado con acero al carbono o acero inoxidable que almacena combustible, son diseñados para evitar la evaporación del combustible, prevenir la entrada de humedad manteniendo el combustible libre de suciedad y evitando la degradación del diésel, su mantenimiento se lo realiza tomando en cuenta varios puntos como: limpieza interior (dos veces por año), pintura exterior, verificación del respiradero y la limpieza de las válvulas de nivel. [22]

En lo que respecta al cálculo del volumen en un cilindro vertical se calcula con la fórmula: $V = \pi r^2 h$. Pero para un cilindro horizontal se debe calcular primeramente el área (A) en función del nivel "H" del líquido, del radio "R" del recipiente y de la longitud "L" del mismo.

$$A = \frac{\pi R^2}{2} + (h - R)\sqrt{2Rh - h^2} + R^2 \arcsen \frac{-R + h}{R} \quad \text{EC. 2}$$

Esta es el área de la base, para encontrar el volumen tenemos que multiplicar por la altura o longitud "L" del cilindro; por lo tanto, la ecuación del volumen será:

$$V = L \left[\frac{\pi R^2}{2} + (h - R)\sqrt{2Rh - h^2} + R^2 \arcsen \frac{-R + h}{R} \right] \quad \text{EC. 3}$$

2.2.5 Pantallas HMI

Las pantallas de Interfaz Humano-Máquina denominado por sus siglas (HMI), son sistemas que muestran un proceso de forma visual por medio de una pantalla que el operador puede

acceder en cualquier momento para controlarlo. Esta ventana puede estar alojada en dispositivos especiales como paneles de operador o en un computador.

Dentro de las computadoras a los sistemas HMI se les conoce como software HMI o de monitoreo y control de supervisión. Las señales tomadas del proceso son enviadas al computador mediante dispositivos como tarjetas de entrada/salida, PLC (Controladores Lógicos Programables). La pantalla de una pantalla se muestra en la figura 2.17.



Figura 2.17: Pantalla HMI (Interfaz Humano-Máquina) [23] [24]

2.2.6 Sistemas electrónicos de comunicación

Un sistema electrónico de comunicaciones está constituido por un transmisor, un medio de transmisión y un receptor, como se observa en la figura 2.18. Un transmisor es un conjunto de uno o más dispositivos o circuitos electrónicos que convierte la información de la fuente original en una señal que se presta más a su transmisión a través de un determinado medio de transmisión. El medio de transmisión transporta las señales desde el transmisor hasta el receptor, y puede ser tan sencillo como un par de conductores de cobre que propaguen las señales en forma de flujo de corriente eléctrica.

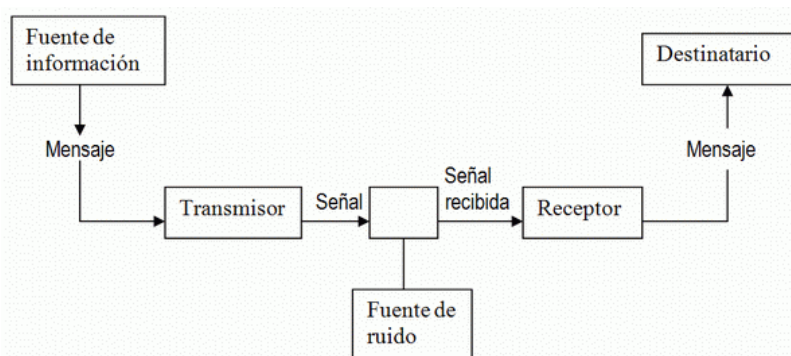


Figura 2.18: Sistema Electrónico de Comunicaciones [25]

El modelo que se muestra en la figura 2.18 ha sido, gracias a su carácter estadístico, principalmente utilizado en el lenguaje entre máquinas y en cuanto a la comunicación humana es visto de manera imperfecta. El punto central es que el mensaje emitido sea exactamente el mismo que recibe el destinatario, sin importar por cuantos elementos de ruido haya pasado.

La información también se puede convertir a ondas electromagnéticas luminosas, propagarlas a través de cables de fibra óptica hechas de vidrio o de plástico, o bien se puede usar el espacio libre para transmitir ondas electromagnéticas de radio, a grandes distancias o sobre terreno donde sea difícil o costoso instalar un cable físico. Un receptor es un conjunto de dispositivos y circuitos electrónicos que acepta del medio de transmisión las señales transmitidas y las reconvierte a su forma original. [25] [26]

2.2.7 Ethernet

Ethernet, al que también se conoce como IEEE 802.3, es el estándar más popular para las redes de área local (LAN), usa el método de transmisión de datos llamado Acceso múltiple con detección de portadora y detección de colisiones (CSMA/CD). Antes de que un nodo envíe algún dato a través de una red Ethernet, primero escucha y se da cuenta si algún otro nodo está transfiriendo información; de no ser así, el nodo transferirá la información a través de la red. Todos los otros nodos escucharán y el nodo seleccionado recibirá la información. En caso de que dos nodos traten de enviar datos por la red al mismo tiempo, cada nodo se dará cuenta de la colisión y esperará una cantidad de tiempo aleatoria antes de volver a hacer el envío. Cada paquete enviado contiene la dirección de la estación destino, la dirección de la estación de envío y una secuencia variable de bits que representa el mensaje transmitido. El dato transmitido tiene una velocidad de 10 millones de bits por segundo y el paquete varía en una longitud de 64 a 1518 bytes, así el tiempo de transmisión de un paquete en la Ethernet está en un rango de 50 a 1200 microsegundos dependiendo de su longitud. La dirección de la estación de destino normalmente es referida por una única interfaz de red. Cada estación recibe una copia de cada paquete, pero ignora los paquetes que son dirigidos a otras computadoras y procesa solamente los que son dirigidos a ella. [27] [28]

2.2.8 Controlador Lógico Programable

El Controlador Lógico Programable (PLC), es un dispositivo electrónico muy usado en Automatización Industrial. Un PLC controla la lógica de funcionamiento de máquinas, plantas y procesos industriales, procesan y reciben señales digitales y analógicas y pueden aplicar estrategias de control. En la figura 2.19 se muestra la imagen de un PLC.



Figura 2.19: Controlador Lógico Programable [28]

Los controladores lógicos pueden tener varios tipos de memorias, por ejemplo:

- RAM: Memoria de lectura y escritura.
- ROM: Memoria de solo lectura, no reprogramable.
- EPROM: Memoria de solo lectura, reprogramables con borrado por ultravioletas.
- EEPROM: Memoria de solo lectura, alterables por medios eléctricos.

La unidad de programación de un controlador es el conjunto de medios, hardware y software mediante los cuales el programador introduce y depura sobre las secuencias de instrucciones (en uno u otro lenguaje) que constituyen el programa a ejecutar. [29] [28]

2.2.9 El Internet de las cosas (IoT)

El internet de las cosas (IoT, por sus siglas en inglés) consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. El hecho de que Internet esté presente al mismo tiempo en todas partes permite que la adopción masiva de esta tecnología sea más factible, los sensores son fácilmente integrables en hogares, entornos de trabajo y lugares públicos, de esta manera, cualquier objeto es susceptible de ser conectado y manifestarse en la Red.

El internet de las cosas plantea que todos los objetos que nos rodean estén conectados a la Red. De esta forma, nuestro frigorífico podría informarnos de que alimentos necesitamos comprar o cuáles están a punto de caducar sin ni siquiera abrir la puerta; podríamos saber si tenemos caries porque nuestro cepillo de dientes las detectaría o que el inodoro haga un análisis de nuestra orina y nos recomiende una dieta adecuada.

El IoT implica que todo objeto puede ser una fuente de datos. Esto está empezando a transformar la forma de hacer negocios, la organización del sector público y el día a día de millones de personas. [30] [31]

2.2.10 Plataforma de desarrollo móvil

Una plataforma es el conjunto de dispositivos móviles, herramientas y aplicaciones que suelen ser definidas por su sistema operativo (SO). Los proveedores de plataformas son grandes empresas como Google, Apple, Microsoft, etc. Cada uno ha desarrollado un sistema operativo que licencia a los fabricantes de dispositivos. A veces, también pueden ser fabricantes de dispositivos. Los fabricantes de dispositivos diseñan y construyen los dispositivos, en su mayoría teléfonos inteligentes y tabletas, con un SO relevante preinstalado. Estos dispositivos se venden luego a los consumidores (usuarios).

Actualmente, el sistema operativo Android de Google tiene la cuota de mercado global más grande, con un enorme 86,1%. iOS de Apple tiene un 13,7% y mantiene la segunda posición. La porción restante de alrededor de 0.2% es compartida por todos los otros vendedores combinados. Esto incluye Windows Mobile de Microsoft, BlackBerry OS, Tizen OS, OS de Sailfish y Ubuntu Touch. [6] [32]

2.2.11 Lenguaje de programación C++

C++ es un lenguaje de programación con el objetivo de extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

El lenguaje de programación de Arduino está basado en C o C++, aunque tiene su propio lenguaje, pero también es posible usar comandos estándar de C++ en la programación de Arduino.

Características de C:

- Es el lenguaje de programación de propósito general asociado al sistema operativo UNIX.
- Es un lenguaje de medio nivel, trata con objetos básicos como caracteres, números y direcciones de memoria.
- Posee una gran portabilidad
- Se utiliza para la programación de sistemas: construcción de intérpretes, compiladores, editores de texto, etc. [33]

2.3 Propuesta de Solución

Por medio de un sistema electrónico de monitoreo se facilita el control de niveles de combustible en los tanques que conforman el sistema de almacenamiento de diésel de manera inalámbrica, para agilizar las rutinas de mantenimiento del sistema y gestionar la adquisición de este combustible de manera oportuna.

CAPÍTULO 3

METODOLOGÍA

3.1 Modalidad de investigación

El proyecto de investigación se ejecutó con modalidad aplicada debido a que se utilizó los conocimientos científicos y técnicos para brindar una solución a un problema o necesidad y proponiendo un prototipo de solución.

Se empleo la investigación bibliográfica, gracias a que es una herramienta indispensable para el desarrollo de todo proceso de investigación porque se obtuvo información de libros revistas, artículos digitales y reglamentos hospitalarios, para determinar los materiales adecuados para utilizar en el desarrollo del proyecto de investigación, de igual manera a través de este tipo de investigación se pudo analizar las posibles soluciones ante un determinado problema.

Investigación de Campo, porque los datos fueron recopilados en el lugar donde se va a desarrollar el sistema, es decir se realizó un análisis del ambiente de trabajo de los equipos a ser instalados y la incidencia de uso. Este tipo de investigación ayudó a tener un mejor enfoque del lugar de implementación del prototipo.

La investigación experimental permitió la implementación del prototipo, para verificar el la efectividad del sistema de monitoreo conforme a las pruebas de funcionamiento.

3.2 Recolección de información

El presente proyecto recopiló información por medio de los reportes de provisión de diésel, informes periódicos de consumo y datos estadísticos de documentos de rendición de cuentas;

todo esto como instrumento metodológico de investigación en el proceso de desarrollo del proyecto. Conforme el número de asistencias que está realizando el Hospital General, se analizó el consumo que se está llevando actualmente.

3.3 Procesamiento y análisis de datos

Se realizó un procesamiento y análisis de datos obtenidos, para determinar el consumo actual de diésel con relación al aumento de pacientes que están siendo atendidos en Hospital.

Posteriormente se procedió a la organización de la información, para ir desarrollando de manera sistemática paso a paso las actividades correspondientes perteneciente al sistema de monitoreo.

La presentación de los resultados está reflejada en el proyecto de investigación, brindando al área de mantenimiento un sistema fiable, en donde todo el personal tiene acceso sistema de monitoreo.

3.4 Desarrollo del proyecto

1. Determinar el estado actual de los tanques de almacenamiento.
2. Identificar las características del consumo de diésel en el Hospital General Ambato.
3. Analizar el control actual del nivel de diésel en el sistema de almacenamiento.
4. Determinar los requerimientos técnicos por parte del personal del área mantenimiento.
5. Seleccionar sensores de nivel, dispositivos de control, tarjetas electrónicas y tecnología inalámbrica que conformaran el sistema.
6. Analizar las tecnologías para el monitoreo y control en los tanques de diésel.
7. Diseñar los circuitos electrónicos del sistema de monitoreo
8. Generar las alertas a un dispositivo móvil de niveles medios y bajos en el sistema de almacenamiento de diésel.
9. Implementar el prototipo
10. Elaborar un presupuesto de equipos y materiales que se utilizan.
11. Realizar pruebas de funcionamiento del sistema.
12. Elaborar un informe final del proyecto de investigación.

CAPÍTULO 4

DESARROLLO DE LA PROPUESTA

4.1 Introducción

El Hospital General Ambato dispone de un sistema de almacenamiento de diesel conformado por cuatro tanques, la supervisión y provisión de este combustible está bajo el mando del personal del área de mantenimiento del hospital; la carencia de diesel podría dejar a esta Unidad Médica fuera de servicio, por lo cual es de mucha importancia la supervisión diaria del número de galones consumidos y los sobrantes en cada tanque. El personal del área de mantenimiento realiza muchas actividades inesperadas por lo que ciertas tareas tienden a ser olvidadas y esto podría generar contratiempos y gastos inesperados para la gerencia del hospital. Estas contrariedades han motivado al personal técnico y administrativo del hospital coincidan en la implementación del sistema de monitoreo de los niveles de diesel para una agilizar las rutinas de trabajo y mantener el prestigio del hospital. Por tanto, se propone el diseño del Sistema Electrónico de Monitoreo del Almacenamiento de Diesel, con la premisa de facilitar el registro y la supervisión del diesel e innovar tecnológicamente el monitoreo actual.

A continuación, se presenta el desarrollo de la propuesta según lo anterior expuesto.

4.2 Factibilidad del proyecto

A continuación, se detallan los factores que influyen en la factibilidad del proyecto.

4.2.1 Factibilidad económica

El presente proyecto de investigación es económicamente factible puesto que los recursos serán financiados por el investigador.

4.2.2 Factibilidad bibliográfica

El presente proyecto se fundamenta también en la disponibilidad de fuentes bibliográficas como libros, revistas técnicas publicaciones y artículos científicos afines al tema, por tanto, es bibliográficamente factible.

4.2.3 Factibilidad técnica

El presente proyecto investigativo es técnicamente factible dado que los equipos y materiales necesarios para su realización se encuentran disponibles en el mercado nacional.

4.3 Condiciones actuales y consumo de los tanques de almacenamiento

El Hospital General Ambato cuenta con un sistema de almacenamiento de diesel conformado por cuatro tanques:

- Dos Tanques de almacenamiento de diesel que son los máximos reservorios que posee el Hospital General Ambato, los cuales son de aproximadamente 5000 galones cada uno, el llenado de cada uno de los tanques se lo hace mediante una tubería por gravedad, aprovechando la topografía del terreno.
- Un Tanque de Combustible para el uso Diario, este tanque tiene una capacidad de 500 galones, y provee diesel para el funcionamiento de los calderos, el llenado se lo realiza mediante una bomba instalada a pie del mismo.
- Un Tanque de Diesel exclusivo para el Generador, este tanque posee una capacidad de 300 galones, y es de uso exclusivo para el consumo del generador eléctrico marca Caterpillar modelo 3412.

El consumo aproximado de diesel en el hospital por cada día es de 200 galones diarios, lo que mensualmente es un total de 6000 galones al mes, el costo por galón de diesel es 1.48 dólares americanos. Lo que da un total de 8880.00 mensuales. La empresa proveedora de diesel es Petroecuador.

La información detallada se obtuvo a través de una entrevista al personal de mantenimiento, más información detallada de la compra y otros detalles de abastecimiento son confidenciales.

4.3.1 Datos técnicos de los tanques almacenamiento de diesel

En la tabla 4.1, se muestra los datos técnicos de los tanques de almacenamiento de diesel.

Tabla 4.1: Datos técnicos de los Tanques de Almacenamiento.

Equipo	Tanque de combustible 1	Tanque de combustible 2	Tanques de combustible 3 y 4
Capacidad	300 gal.	500 gal.	5000 gal.
Longitud	2.50 m	2.5 m	4.48 m
Diámetro	0.76 m	1 m	2.32 m
Posición	Horizontal	Vertical	Horizontal
Codificación	AD t- 23 C	AD t- 23 C	AD t- 23 C
Color	Negro	Gris	Rojo
Tipo	Reservorio	Reservorio	Reservorio
Función	Consumo generador	Consumo diario	Almacenamiento
País Origen	Ecuador	Ecuador	Ecuador
Material	Acero	Acero	Acero
Llenado	Gravedad	Gravedad	Gravedad
Numero de tanques	1	1	2

Elaborado por: Oscar Analuisa basado en los documentos técnicos del hospital.

En la Figura 4.1 se muestra fotografías de cada tanque de almacenamiento, estos tanques se encuentran en la casa de máquinas, la primera fotografía es el tanque de 300 galones que alimenta al generador eléctrico, la segunda es el tanque de 500 galones que es el de consumo diario y la tercera muestra los dos tanques de 5000 cada uno que son los máximos reservorios.

Tanque de combustible 1	Tanque de combustible 2	Tanque de combustible 3 y 4
		

Figura 4.1: Tanques de almacenamiento de diesel

Elaborado por: Oscar Analuisa

El consumo principal de diesel es realizado por los calderos que son utilizados para generar el vapor seco para la producción de calefacción y agua caliente, otro consumo de diesel es realizado por parte del generador eléctrico. Los equipos que trabajan a diesel son: el generador eléctrico marca Caterpillar que se observa en la figura 4.2 y los tres calderos marca Superior Boiler Works que se muestran en la figura 4.3.



Figura 4.2: Generador Eléctrico marca Caterpillar modelo 3412

Elaborado por: Oscar Analuisa

En la tabla 4.2 se muestra los datos técnicos del caldero del sistema de generación de vapor.

Tabla 4.2: Datos técnicos del generador eléctrico marca Caterpillar

Datos Técnicos del Generador Eléctrico	
Máquina	Generador
Marca	Caterpillar
Calor	Amarillo
Potencia	900 KVA
Consumo de combustible	191.7 l/h (50.6 gal/h)
Modelo de motor	3412
Rendimiento	95.5 %
Frecuencia	50 Hz
Baterías	24 V
Protección	IP23

Elaborado por: Oscar Analuisa basado en [34]



Figura 4.3: Caldero marca Superior Boiler Works

Elaborado por: Oscar Analuisa basado

En la tabla 4.3 se muestra los datos técnicos del caldero del sistema de generación de vapor.

Tabla 4.3: Datos técnicos del caldero marca Superior Boiler Works

Datos Técnicos del Caldero	
Máquina	Caldero
Marca	Superior Boiler Works
Calor	Azul
Número de Equipos	3
Tipo	Horizontal
Modelo	46751
Año de Fabricación	1991
Presión de trabajo	150 psi
País de origen	USA
Voltaje	230 V
Amperaje	16 A
Frecuencia	60 Hz
Capacidad de vapor	4312 lib/hr

Elaborado por: Oscar Analuisa basado en datos técnicos del hospital

4.4 Analizar el control actual de nivel de diesel en el sistema de Almacenamiento.

Los cuatro tanques de almacenamiento de diesel utilizan para la visualización del nivel una manguera conectada junto a cada tanque, la manguera por presión atmosférica indica en qué posición se encuentra el diesel dentro del tanque, adherida a cada una de las mangueras se tiene pegada cinta graduada para visualizar el nivel de diesel. Los datos son recopilados diariamente en una hoja de control.

La figura 4.4 muestra la forma de visualización del nivel de diesel en cada uno de los tanques de almacenamiento, se observa que las etiquetas en escala de galones están pegadas con cinta adhesiva en cada tanque y paralelos a cada manguera.




Tanque de combustible 300 Gal.	Tanque de combustible 500 Gal.	Tanque de combustible 5000 Gal.
		

Figura 4.4: Visualización del nivel de diesel de los tanques de almacenamiento

Elaborado por: Oscar Analuisa

4.5 Requerimientos técnicos para diseño por parte del personal de mantenimiento.

El sistema desarrollado en la Casa de Máquinas del Hospital General Ambato tuvo los siguientes requerimientos técnicos que se obtuvo en base a reuniones con el personal de mantenimiento y entrevistas con el jefe de mantenimiento.

- Monitorear localmente los niveles de diesel a través de una pantalla HMI
- Monitorear remotamente los niveles de diesel a través de un dispositivo móvil
- Guardar información del consumo de diesel en un servidor en la nube
- Ofrecer escalabilidad para la instalación de sensores de nivel adicionales
- Generar notificaciones para alertar niveles medios (1/2 de tanque)
- Generar notificaciones para alertar niveles bajos (1/4 de tanque)
- Presentar los niveles de diesel en tiempo real
- Generar alertas sonoras

4.6 Diagrama de Bloques del Sistema

El diagrama de bloques del prototipo se lo puede observar en la figura 4.5. El sistema cuenta con dos etapas con el objetivo de separar el monitoreo local con el monitoreo remoto de esta manera mantener la visualización de los datos de nivel en la pantalla HMI, sin la necesidad de tener una conexión a internet. La segunda etapa se encarga de subir la información adquirida por el sensor a la nube, los datos se pueden visualizar ingresando a la página del servidor, seguidamente se crea un enlace al servidor de notificaciones para el envío de alertas al dispositivo móvil.

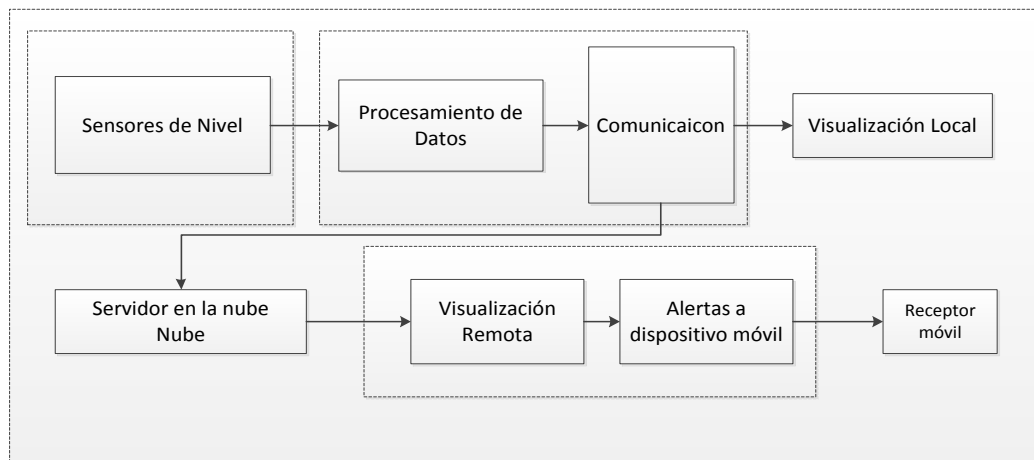


Figura 4.5: Diagrama de Bloques del Sistema

Elaborado por: Oscar Analuisa

4.7 Selección de elementos para el diseño y construcción del Sistema de Monitoreo.

La selección de los elementos se basa en el diagrama de bloques de la figura 4.5, se procede a elegir el mejor elemento para el sistema, empezando de forma ordenada de la siguiente manera:

- Sensor de nivel
- Tarjeta de adquisición de datos
- Pantalla para la visualización local
- Tipo de comunicación
- Plataforma en la nube
- Servidor de notificaciones
- Entorno de desarrollo para la aplicación

A continuación, se presenta detalladamente la selección de cada elemento.

4.7.1 Sensor de Nivel

Actualmente existe una gran variedad de medidores de nivel de líquido. Cada medidor existente en el mercado tiene ventajas y desventajas individuales, y cada uno incorpora distintas características que el otro no las posee, por esta razón se realiza un estudio de los transductores para la selección adecuada según la aplicación que se vaya a realizar.

Para elegir un tipo de medidor y determinar los requisitos que deberá tener es necesario conocer los principios físicos de la variable a medir, los dispositivos de medición y sus principios de operación. En la tabla 4.4 se detalla las características de los traductores de nivel.

Tabla 4.4: Características de los Transductores de Nivel [35]

Transductores de Nivel	Rango de Medición (m)	Temperatura en °C.	Presión Máxima del Líquido (PSI)	Exactitud (%FE)	Aplicaciones
Regla Graduada	Limitado por la longitud del dispositivo	60	14,696	±0,5 mm/FE	Tanques Abiertos Medición de Aceite y Gasolina. Procesos Granulosos
Varilla con Gancho	Limitado por la longitud del dispositivo	60	14,696	±0,5 mm/FE	Tanques Abiertos Medición de Aceite y Gasolina. Procesos Granulosos
Cinta Métrica Metálica con Plomada	Limitado por la longitud del dispositivo	60	14,696	±0,5 mm/FE	Tanques Abiertos Medición de Aceite y Gasolina.

					Procesos Granulosos
Indicador de Cristal	1,78 (Baja Presión) 2,5 (Alta Presión)	200	> 101,5	±0,5 mm/FE	Lecturas Periódicas en Tanques Abiertos y Cerrados
Instrumentos de Flotador	0 - 10	530	5000	±6,349mm/F E	Tanques Abiertos y Cerrados, Presurizados o al Vacío
Medidor de Desplazamiento	0 - 39,37	540	5000	±6,349mm/F E	Tanques Abiertos y Cerrados, Presurizados o al Vacío. Control, Alarmas y Medición de Interfaz Líquido a Líquido
Manométrico	Limitado (altura del Tanque)	60	14,696	±1	Tanques Abiertos y Cerrados a presión Atmosférica
Caja de Diafragma	Limitado (altura del Tanque)	60	14,696	±1	Tanques Abiertos
Burbujeo	Limitado por la altura del Tanque	200	5693	±1	Tanques Abiertos. Líquidos Corrosivos o con sólidos en suspensión
Presión Diferencial	Limitado por los intervalos	200	2135	±0,5	Medición de nivel de Interfaces

	del manómetro o diferencial				Indicación Registro y Control
Capacitivo	Amplio	400	3560	±1	Medición de Interfaz Alarmas Control
Ultrasonido	0,152 - 30,48	200	5700	±1 - ±3	Mediciones Continuas y Alarmas

Una vez analizado los diferentes transductores de nivel de combustibles se decidió utilizar un transductor de nivel ultrasónico, este tipo de transductor trabaja basándose en las características eléctricas del líquido, lo que es muy utilizado en lo que es medición de combustibles. Adicionalmente en la tabla 4.4 se observa las características técnicas del transductor ultrasónico, detallando una distancia máxima de 30,48 metros, lo que garantiza el monitoreo total de los tanques de almacenamiento ya que la distancia del tanque de mayor capacidad es de 2,50 metros, que se ajusta a las necesidades del sistema sin inconvenientes. Las aplicaciones del sensor ultrasónico están acorde a las especificaciones de funcionamiento del sistema, facilitando la adquisición de datos de forma continua para presentar datos actuales y en lo referente a la generación de alarmas para la creación de notificaciones al tener un nivel inapropiado.

A continuación, en la tabla 4.5 se detalla la clasificación de los tipos de transductores según las características del tanque y según la exactitud:

Tabla 4.5: Selección de los transductores de nivel

SELECCIÓN DE LOS TRANSDUCTORES DE NIVEL		
Según las Características del Tanque:	Tanques Abiertos	<ul style="list-style-type: none"> • Se pueden utilizar todos los dispositivos mencionados en la tabla 4.4
	Tanques no Presurizados	Se deben utilizar: <ul style="list-style-type: none"> • Indicador de Cristal • Medidor de Ultrasonido • Medidor de Desplazamiento

		<ul style="list-style-type: none"> • Instrumentos de flotador • Manométrico • Capacitivo
	Tanques Presurizados	Se deben utilizar: <ul style="list-style-type: none"> • Indicador de Cristal • Capacitivo • Medidor de Ultrasonido • Medidor de Desplazamiento • Instrumentos de flotador
Según la Exactitud:	Alta	<ul style="list-style-type: none"> • Medidor de Sonda • Medidor de Presión Diferencial • Indicador de Cristal (0,5%)
	Media	<ul style="list-style-type: none"> • Manométrico • Caja de Diafragma • Burbujeo • Capacitivo • Ultrasonido (1 %)
	Baja	<ul style="list-style-type: none"> • Instrumentos de Flotador • Medidor de desplazamiento (6,349%)

Elaborado por: Oscar Analuisa basado en [15]

Con estas características se puede realizar una selección adecuada del medidor. En el análisis de los transductores es indiscutible que un medidor de nivel realiza la misma tarea, pero la importancia radica en identificar los requerimientos de medición y cumplir las condiciones de operación. Analizando la tabla 4.5, se puede apreciar la relación de las características del tanque con la exactitud, por lo que se opta por la combinación de los dos, dando una mejor perspectiva de la selección de un sensor ultrasónico.

Esta acción se realiza en base a exactitud, repetibilidad, costo, confiabilidad, instalación, mantenimiento y visualización requerida. En la tabla 4.6 se muestra la selección de los parámetros necesarios para el transductor de nivel del prototipo y del sistema completo.

Tabla 4.6: Parámetros en la selección de un transductor de nivel

Parámetros en la selección de un transductor de nivel de líquidos	Parámetros para selección del transductor de nivel para el prototipo	Parámetros para selección del transductor de nivel para el sistema completo
Exactitud	✓	✓
Temperatura de trabajo.	X	X
El tipo de tanque en el cual va a ser instalado	✓	✓
Presión máxima.	X	X
Características del fluido.	✓	✓
Repetibilidad	✓	✓
Robustez	X	✓
Intercambiable	✓	✓
Compatibilidad	✓	✓
Tamaño apropiado.	X	X
Costo	X	✓
Sensibilidad	X	✓
Requerimientos de mantenimiento	X	X
Confiabilidad	✓	✓
Resistente al ruido eléctrico	✓	✓
Calibración en tiempo real	X	✓
Bajo requerimiento de potencia eléctrica	X	X
Inmune al ruido	✓	✓
Certificaciones	✓	✓
Tensión de Alimentación (ahorro de energía)	X	X
Fácil acoplamiento del sensor	X	X
Adecuados para las características eléctricas del líquido	✓	✓
Sencillez del diseño (uso por personal sin experiencia)	X	X

Elaborado por: Oscar Analuisa basado en [15]

Los requerimientos necesarios del sensor de nivel tanto para el prototipo como para el sistema completo están marcados con un visto y los que no son necesarios para el sistema se pone con una equis para facilitar la selección del transductor para que trabaje de manera eficiente.

Teniendo en cuenta los parámetros de selección de un transductor, ver tabla 4.7. Se procede a categorizar los transductores de las tres gamas existentes teniendo en cuenta el costo beneficio del transductor.

Tabla 4.7: Transductores de nivel que cumplen las características

Transductor	Gama Baja	Gama Media	Gama Alta
Características			
Tranductor	Sensor Ultrasónico HC-SR04	MB7138 XL-TrashSonar-WR	IRU-5000 Ultrasonic Level Sensor
Voltaje de Funcionamiento	5V (DC)	3 a 5 V (DC)	12-28 VDC
Corriente Stand By	<2 mA	50mA peak	75 mA
Consumo en Funcionamiento	15mA	100mA peak	75 mA a 24 VDC
Frecuencia de Funcionamiento	40KHz	42KHz	143 kHz
Señal de salida	Tren de Pulsos, nivel alto a 5V, nivel bajo a 0V	RS232 Serial (0 V a 5 V), I2C Bus	4-20 mA (0-2.5 V / 0-5 V)
Resolución	0,3 cm	1-cm	0.1 in. (2.5 mm)
zona muerta (Tipo de tanque)	2cm	0-20 cm	9° off axis
costo	bajo	medio	medio/alto
Distancia de Detección:	2 - 450 cm	0-cm to 350-cm	2700 mm
Inmune al ruido	No	Si	Si
Certificación	ninguna	CE (ISO 17050)	Ex nA IIB T6

Elaborado por: Oscar Analuisa basado en [36] [37] [38]

Para la selección del transductor de nivel para el prototipo según las características técnicas del tanque de almacenamiento, el tanque no es presurizado lo que la presión atmosférica entra y sale libremente del tanque. Las tres opciones de sensores se instalarían sin ningún problema, pero el sensor de la opción 1 solamente es utilizado para aplicaciones básicas sin certificaciones e inexacto al momento de monitorear, la mejor opción sería la número tres porque estaría acorde a las características del tanque y las certificaciones, pero el costo de este sensor se eleva debido a todo lo mencionado Una exactitud de nivel medio será adecuada para observar el correcto funcionamiento del prototipo. El transductor de ultrasónico MB7138 XL-TrashSonar-WR cumple las características mencionadas es robusto por su

certificación CE, el costo accesible y satisface las características del tanque de almacenamiento.

Por otra parte, para el sistema completo que está conformado por los cuatro tanques y se procede a una selección de tres sensores adicionales. Una característica requerida por el personal que trabaja en el hospital es el cuidado en la adquisición de sensores con certificaciones para trabajar con este tipo de combustible.

Después de ver los diferentes tipos de transductores en función tanto de sus capacidades y características, se ha decidido usar para los tanques de 5000 galones el sensor IRU-5000, apto para trabajar según la capacidad, longitud, diámetro y demás parámetros técnicos. Para el tanque 500 galones de alimentación diaria de los calderos se decide usar el mismo sensor debido a que la detección es de 2700 milímetros. El sensor seleccionado satisface los parámetros técnicos de los tanques, en distancia de detección, longitud, diámetro y las certificaciones de protección. Además, el transductor seleccionado se encuentra actualmente en el mercado y por la certificación EX (grado de protección en zonas con peligro de explosión), es el más beneficioso y seguro para el desarrollo del proyecto.

4.7.2 Selección de la tarjeta de adquisición de datos

La tarjeta de adquisición de datos es de relevante importancia para el proyecto debido a que es la encargada del procesamiento de los datos obtenidos por los sensores. En ella se ejecutan los cálculos, procesos e instrucciones que deberá realizar el sistema, por este motivo es adecuado analizar diferentes propuestas tomando en cuenta los requerimientos necesarios para el prototipo que se enlistan a continuación:

- Detección y lectura de las señales que envía el transductor de nivel instalado en un tanque de almacenamiento.
- Desarrollo del funcionamiento del algoritmo de control y envío de las acciones de mando al sistema.
- Comunicación hacia la nube para establecer para posterior establecer una transición desde el controlador al dispositivo móvil.
- Capacidad de ser reprogramado con un nuevo algoritmo de supervisión y control de alertas.

- Escalabilidad en el número de entradas para las señales de todos los transductores de nivel y las salidas para las luces indicadoras

Las alternativas se detallan en la tabla 4.8.

Tabla 4.8: Alternativas para selección de las tarjetas [39]

Tarjetas	PLC	PAC	Controllino	PIC
Conectividad a Ethernet vía WEB	No	Si	Si	No
Procesador	Microprocesadores	Core 2 Dou	ATMega2560	Microprocesadores de núcleo de 32 bits
Tecnología de comunicación de red	Ethernet	Ethernet, OPC y SQL	Ethernet	RS232
Licencias	Pagada	Pagada	Libre	Pagada
Memoria	No volátil	No volátil	No volátil	No volátil

A continuación, se detalla un análisis de cada una de las alternativas

Autómata Programable Industrial

Las funciones del autómata programable industrial (API) o controlador lógico programable (PLC) el cual forma parte del sistema de monitoreo, debe cumplir con las siguientes características básicas:

Se descarta la utilización de los Controladores de Automatización Programables (PAC) a pesar de que son equipos más recientes, que tiene las funciones de un PLC y otras funciones adicionales, por ejemplo: procesamiento avanzado para señales analógicas, comunicaciones avanzadas de gateways entre redes. conjunto de instrucciones para manejo de variadores de frecuencia (VDF). Por tal motivo es innecesario trabajar con un controlador con estas características que no serán utilizadas y solo elevara el costo del sistema.

También se descarta las soluciones basados en diseños con microcontroladores a pesar de que económicamente fuera reducido pero el sistema de monitoreo debe garantizar la robustez

y confiabilidad para su desempeño en un ambiente hostil de trabajo, considerando las condiciones ambientales y físicas. También se descarta debido al tiempo que requiere para probar y acondicionar el diseño para que cumpla todas las características de funcionamiento del sistema.

Con estos criterios de funcionamiento, en el mercado actual existe una gran gama de dispositivos que cumplen con las características especificadas. La selección del dispositivo se la realiza a través del análisis de los factores requeridos, las características del proceso en su ambiente de desarrollo, funciones requeridas y el costo.

El PLC de Arduino conocido con el nombre CONTROLLINO es el elegido para ser parte del sistema de monitoreo, el modelo maxi es muy utilizado de la gama Controllinos en procesos de automatización. Se lo ha seleccionado por su flexibilidad para establecer una conexión a la web a través de la conexión Ethernet, la programación basada en Arduino es de código abierto por esta razón se evita el pago de licencias, el costo se modera al evitar seleccionar solo los requerimientos necesarios para el sistema. En la tabla 4.9, se detalla las principales características técnicas del Controllino y el datasheet de esta tarjeta se presenta en el Anexo F.

Tabla 4.9: Características técnicas del Controllino

Características	Detalles
Procesador	ATmega2560
Oscilación:	16 MHz
Clock	RTC
Corriente de entrada	Máx. 20A
Salidas de Rele	10x Relays Outputs – 230V / 6A
Entradas	12x Analog/Digital Inputs
Salidas Digitales	12x Digital Outputs – 2A a 12V o 24V
Puertos de conexión	<ul style="list-style-type: none"> ▪ Conector Ethernet ▪ Interfaz serie 2x ▪ Interfaz 1x RS485 ▪ 1x interfaz I2C ▪ 1x interfaz SPI

Elaborado por: Oscar Analuisa basado en [40]

La selección de este tipo de controlador de una gran familia de controladores se debe a que cumple las características mencionadas cumpliendo los requerimientos del sistema y evitando adquirir un equipo con características innecesarias que solo aumenta el costo del equipo. Además, la característica fundamental de un análisis a futuro es la escalabilidad del sistema utilizando el mismo controlador, el prototipo solo usará una entrada de un transductor de nivel y tres salidas digitales para las luces indicadoras de estado de nivel, por lo que el controlador debe ser capaz de soportar entradas de los transductores que serán colocados en los demás tanques de diesel, de una forma eficiente y manteniendo la robustez del sistema.

4.7.3 Visualización Local

La visualización local tiene como objetivo presentar la información procesada por el Controllino. Existen una infinidad de dispositivos que permiten la visualización de información y para la selección de este elemento se toma en cuenta el siguiente listado de criterios:

- Acceso a la información para programar la pantalla HMI
- Conectividad para que el transporte de datos en la etapa de programación
- Interfaz amigable con el usuario
- Visualización de las cantidades

Las alternativas son las siguientes:

- A. Pantalla THINGET OP 320-S (Operator Panel 320-S)
- B. Pantalla SIMATIC IPC477D
- C. Pantalla de PC

Tabla 4.10: Matriz de ponderación de la pantalla

ASPECTOS	PONDERACIÓN	ALTERNATIVAS		
		A	B	C
Acceso a la información para programar la pantalla HMI	10	10	10	9
Conectividad	10	10	10	8
Costo	10	10	9	10
Interfaz amigable con el usuario	10	10	9	10
Visualización de las cantidades	10	10	9	8
PUNTUACIÓN TOTAL		50	47	45

Elaborado por: Oscar Analuisa basado en [41]

Después de analizar las posibles opciones de la tabla 4.10. La alternativa que presenta mejores beneficios es la opción A (Pantalla THINGET OP 320-S), por su acceso a la información para programar la pantalla, compatibilidad con el Controllino y la facilidad de manipulación que llega a ser amigable para el usuario, hay que tener en cuenta que el costo también es accesible. Se ha optado también por este tipo de pantalla por seguir una homologación de equipos de la misma marca ubicados en otras cajas de monitoreo instaladas en la casa de máquinas del hospital.

4.7.4 Comunicación

Una vez procesada toda la información adquirida por el controllino, es necesario transmitirla hacia un servidor en la nube para que pueda ser almacenada y poderla visualizar, existen diferentes tecnologías de comunicación, se detallan los más sobresalientes en la tabla 4.11.

Tabla 4.11: Tecnologías de la Comunicación

Tecnología	Velocidad de TX	Distancia Máxima	Medio de TX	Número Máximo de Dispositivos	Topología	Costo
Ethernet	10Mbit/s – 1Gbit/s	100m	Par Trenzado	100	Estrella	Bajo

WIFI	11Mbit/s – 248Mbit/s	30m – 100m	Radio Frecuencia	255 A la vez	Estrella	Medio - Alto
Bluetooth	1Mbit/s – 32Mbit/s	10m – 100m	Radio Frecuencia	1 a la vez.	Maestro - Esclavo	Bajo
USB	480Mbit/s	5m	Guiado	127	Estrella	Bajo
IRDA	4 Mbit/s	2m	Infrarrojo	1 a la vez	Maestro - Esclavo	Bajo
ZigBee	250 kbit/s	75m	Radio Frecuencia	2 (Transmisor y Receptor)	Maestro - Esclavo	Medio
GPRS	60 kbit/s	-	Radio	Depende de la estación base	Estrella	Alto

Elaborado por: Oscar Analuisa basado en [42]

Todas las tecnologías tienen cualidades relevantes con respecto a la otra, pero en un ambiente totalmente industrial la tecnología ethernet trabaja eficientemente sin perder conexión por ruido o por diferentes interferencias. La tecnología de comunicación seleccionada brinda una transmisión de datos rápida de 1 Gbit por segundo, la distancia de operación de 100 metros es óptima por lo que cubre la distancia del Controllino que se encuentra a una distancia de 30 metros del switch, sin la necesidad de utilizar repetidores.

4.7.5 Plataforma en la nube

Una plataforma permite tratar datos recogidos para luego mostrarlos, responder a ciertos eventos, realizar acciones, entre otras opciones, pero para el proyecto de investigación se tiene prioridad en el almacenamiento datos para luego visualizarlos.

El Servidor de una plataforma es una aplicación donde se puede gestionar recursos, el principio de operación es brindar respuestas de manera instantánea.

Existe una gran variedad de plataformas que poseen características similares, tanto en versión libre como pagada, entre estas tenemos a Temboo, ThingSpeak, Nimbits, Plot.ly, Thinger.io, entre otras. En la tabla 4.12 se observa las características de las alternativas para seleccionar el servidor.

Tabla 4.12: Plataformas en la nube

Servidores	Temboo	ThingSpeak	Nimbits	Thinger.io
Licencias	14 días libres	Libre Restringida	Libre y Pagada	Libre y Pagada
Categoría	Internet de las cosas	Internet de las cosas	Artistas	Internet de las cosas
Software	Editable	No editable	Editable	No editable
Aplicaciones	Una visión del IoT aplicado a la industria	Domótica con Matlab	Control sobre su música	Monitoreo
Conexión	Ethernet, Wifi, Gsm	Ethernet, Wifi, Gsm	Ethernet, Wifi, Gsm	Ethernet, Wifi, Gsm
Integración	Arduino, Raspberry Pi, Servidor publico	Arduino Raspberry Pi	Arduino Raspberry Pi	Arduino Raspberry Pi
Limitaciones	El alcance de 100 APIs (interfaz de programación de aplicaciones)	Para tres canales	500 MB	2 dispositivos y 4 Paneles

Elaborado por: Oscar Analuisa basado en [43] [44] [45]

Después de analizar las alternativas, Thinger.io por la aplicabilidad en monitoreo, es compatible con Arduino está acorde con los requerimientos de nuestro sistema. La versión libre ofrece la conexión de dos dispositivos, pero el sistema utiliza únicamente un elemento, el Controllino. En lo que se refiere a paneles si utiliza los cuatro debido a que el sistema se conecta a cuatro sensores de nivel. Por otra parte, la popularidad de la plataforma Thinger.io está aumentando, debido a que tiene una versión libre muy completa y para el sistema de monitoreo de tanques de diesel es idónea.

La versión libre de Thinger.io ofrece los siguientes servicios que se observa en la figura 4.6.

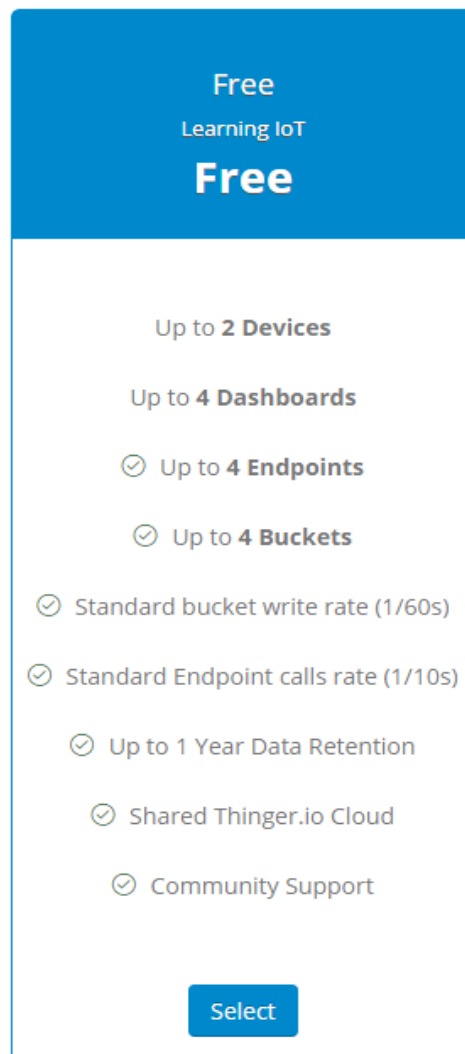


Figura 4.6: Servicios de la plataforma thinger.io de la versión libre [46]

4.7.6 Servidor para notificaciones

El servidor para notificaciones envía solicitudes de mensajes a las apps cliente que se ejecutan en los dispositivos de los usuarios.

Los servidores de mensajería para notificaciones utilizan Tecnología Push. Esta tecnología describe un estilo de comunicación entre cliente y servidor, en la cual el servidor es quien desea enviar un mensaje al cliente. Existen otras tecnologías como Pull, la cual define que el inicio de la comunicación la inicie el cliente. En la actualidad hay una variedad de

servidores para este tipo de servicios con características similares. Las posibles opciones para la selección del servidor se presentan en la tabla 4.13.

Tabla 4.13: Cuadro comparativo de los servidores de mensajería de notificaciones

	Firestore Cloud Messaging (FCM)	Meteor	Mean Stack
Plataforma	Base de datos en tiempo real	Web Framework	Web framework
Base de datos (DB)	Real-tiempo sobre Mongo (DB basado en código abierto)	Interfaz de programación de aplicaciones (API) de DB de Mongo	Mongo DB
Precio	Empaquetamiento simplificado	Escala basada	Libre
Interfaz	Firestore, React.js y otras	Blaze, Angular, React.js	Mongo, Express, Angular
Servidor	Programación limitada	Programación Libre	Libre con compleja programación
Nivel de complejidad	Fácil programación en la Web, iOS, Android apps.	Fácil Programación para Web Apps	Complejo en el lenguaje de programación

Elaborado por: Oscar Analuisa basado en [47] [48]

Luego de analizar las opciones se opta por utilizar Firestore Cloud Messaging (FCM). El servidor seleccionado se destaca en sus servicios principales: Base de datos en tiempo real, Sistema de autenticación y Hosting de archivos estático. [49]

Firestore ofrece un servicio gratuito que ayuda a la aplicación Android a enviar datos desde las apps de usuario. El servicio gestiona las colas de mensajes en todos los aspectos y nos permite enviar mensajes de hasta 4 KB a los dispositivos móviles.

Google Cloud Messaging para lograr una comunicación efectiva entre el dispositivo móvil y el servidor gracias a su baja complejidad al momento de la programación.

En la figura 4.7 se muestra los servicios que proporciona la plataforma firebase de la versión libre.

Plan Spark Sin cargo	
Incluido sin cargo Authentication (except Phone Auth), Analytics, Predictions, App Indexing, Dynamic Links, Invites, Remote Config, Cloud Messaging (FCM), Performance Monitoring, Crash Reporting, and Crashlytics.	Incluidos
Realtime Database	
Conexiones simultáneas	100
GB almacenados	1GB
GB descargados	10 GB/mes
Cloud Firestore	
Stored data	1 GB total
Bandwidth	10GB/month
Document writes	20,000/día
Document reads	50,000/día
Document deletes	20,000/día
Storage	
GB almacenados	5 GB
GB descargados	1 GB/día
Operaciones de carga	20,000/día
Operaciones de descarga	50,000/día
Cloud Functions	
Invocaciones	125,000/mes
GB-segundo	40,000/mes
CPU-segundo	40,000/mes
Redes de salida	Solo para Google
Phone Auth	
US, Canada, India	10k/month
All other countries	
Hosting	
GB almacenados	1 GB
GB transferred	10 GB/mes
Dominio personalizado y SSL	
Test Lab	
Virtual Device Tests	10 tests/day
Physical Device Tests	5 tests/day

Figura 4.7: Servicios que proporciona la plataforma firebase de la versión libre [33]

4.8 Selección del entorno de desarrollo

Para el desarrollo de la aplicación que permita monitorear los niveles de diesel, aplicado en la casa de máquinas del Hospital General Ambato, se usaran herramientas oficiales para el desarrollo de aplicaciones Android, y el entorno de desarrollo integrado, llamado también IDE que usaremos en nuestro proyecto será Android Studio que se detalla en la tabla 4.14.

Tabla 4.14: Herramientas utilizadas para el desarrollo de la aplicación

APLICACIÓN ANDROID		
Herramienta	Descripción	Enlace Descarga o Instalación
Android Studio	Android Studio es un nuevo entorno de desarrollo integrado para el sistema operativo Android lanzado por Google	https://android-studio.uptodown.com/windows
Java Platform (JDK) 8	Java Development Kit o (JDK), es un software que provee herramientas de desarrollo para la creación de programas en java.	http://www.oracle.com/technetwork/es/java/javase/downloads/index.htm
ADT(Android Development Tools) Plugin	(ADT) es un plugin para el IDE de Eclipse que está diseñado para darle un ambiente de gran alcance.	Se lo hace desde Eclipse: agregando repositorio: https://dl-ssl.google.com/android/eclipse/
Android SDK(Software Development Kit)	En el SDK podemos encontrar todas las librerías necesarias para poder trabajar con Android. Además de un Emulador.	http://developer.android.com/intl/es/sdk/index.html
Requerimientos para conexión a la web		
API Google Firebase Cloud Messaging	El API de Firebase Cloud Messaging es un servicio gratuito que te permite enviar notificaciones en tus aplicaciones o en tus páginas web.	Se necesita tener una clave para usar el API de Firebase Cloud Messaging se puede obtener: https://developers.google.com/console/help/?hl=es#generatingdevkeys
IIS (Internet Information Server 7)	Servidor web, conjunto de servicios para el S.O. M.Windows, convierte a una PC en un servidor web para Internet.	Viene integrado en M. Windows, Hay activar características y escoger Internet Information Services en Panel de Control

Elaborado por: Oscar Analuisa basado en [50]

4.9 Diseño de los circuitos electrónicos del sistema de monitoreo.

a) Alimentación permitida por el controllino

El controllino es un controlador lógico programable que trabaja con el microcontrolador Atmega2560, puede trabajar con 12 o 24 voltios de corriente directa. El voltaje de referencia tomado para el diseño del acondicionador es de 24V de entrada.

Para la alimentar al controllino se utiliza una fuente de alimentación en carril DIN modelo DR-45. EL diagrama de bloques de la fuente se observa en la figura 4.8.

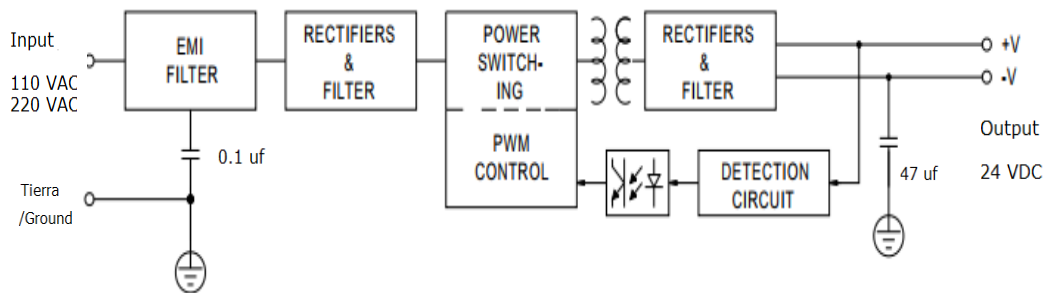


Figura 4.8: Diagrama de bloques de la fuente de alimentación DR-45 [59]

b) Acondicionamiento del sensor

Hay ocasiones en las que se necesita generar una ecuación de línea recta en función del voltaje de entrada.

La ecuación de salida es del tipo lineal

$$Y=mx+b.$$

Donde

- x es el voltaje de entrada
- Y el voltaje de salida
- b el nivel de offset
- m el factor escala

El Circuito Acondicionador de Señales se usa para acoplar los niveles de voltaje que envía un sensor.

El circuito acondicionador de señales (CAS) se comporta como una línea recta $Y=mx+b$. generando la siguiente ecuación.

$$m = \frac{y_{max} - y_{min}}{x_{max} - x_{min}}$$

En la figura 4.9, se observa el circuito general de una señal de salida tipo lineal, donde la señal de entrada (Y) va a un amplificador operacional y opera con el factor escala (m), el resultado proveniente del amplificador pasa a sumar con el valor de offset (b) y finalmente el resultado se muestra en el display.

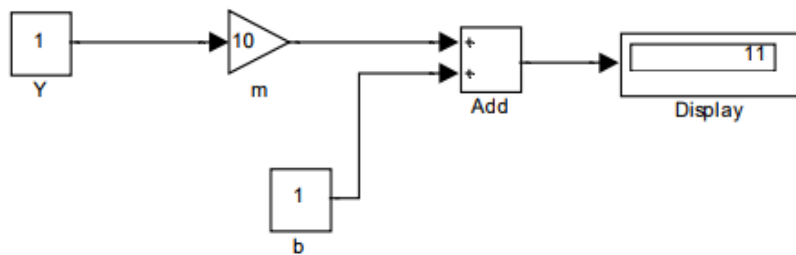


Figura 4.9: Circuito que genera una señal de salida tipo lineal [51]

c) Convertidor RS232 a TTL

Este conversor basado en el circuito integrado MAX232 permite convertir series de nivel TTL a serie RS-232 estándar. El conversor toma el nombre de MAX3232. Esto permitirá que un microcontrolador se comuniquen de forma serial con una computadora usando el conector DB9

Comuniquen microcontroladores utilizando el puerto serial. Es aplicable a una amplia variedad de plataformas, incluyendo Arduino, AVR, ARM, etc.

En la figura 4.10 se indica el diagrama esquemático de la placa MAX3232, para más detalles, observar el datasheet de la placa en el anexo I.

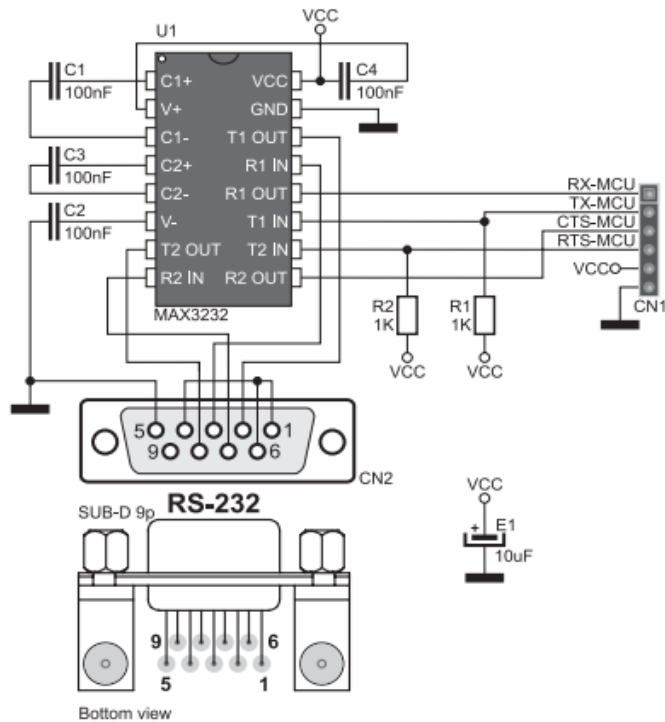


Figura 4.10: Diagrama esquemático de la placa MAX3232 [57]

d) Conversor Análogo Digital

El conversor Análogo Digital ADC está formado por tres etapas: muestreo, cuantificación y codificación como se indica en la figura 4.11. En la primera etapa se toma muestras de la señal, luego se aproxima el valor muestreado a un determinado nivel de cuantificación; este valor cuantificado se codifica en una palabra digital. [26]

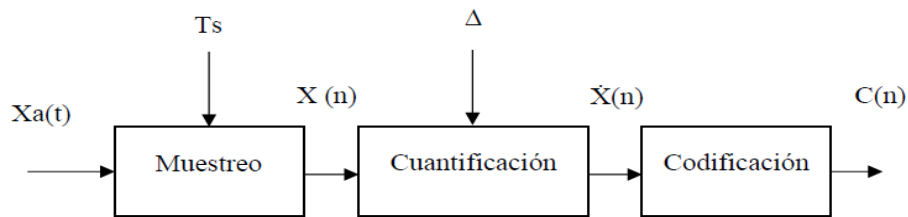


Figura 4.11: Esquema de un conversor Análogo-digital

Desarrollado por: Oscar Analuisa basado en [26]

e) Tiempo de Conversión y Frecuencia de Muestreo

Es necesario analizar el tiempo en que tarda el proceso de conversión, porque con eso se limita la velocidad máxima de la entrada. El número de conversiones que se pueden realizar en un determinado tiempo se lo conoce como frecuencia de muestreo.

La frecuencia máxima de muestreo se la calcula con la siguiente formula basado en el teorema del muestreo:

$$F_{max} = \frac{1}{t_c} \quad \text{EC.4}$$

Donde

F_{max} es la frecuencia máxima.

t_c es la frecuencia de conversión.

El tiempo de conversión del microcontrolador ATMEGA 2560 es de 13us, teniendo así:

$$F_{max} = \frac{1}{t_c}$$
$$F_{max} = \frac{1}{0.000013}$$
$$F_{max} = 76.9Khz$$

Entonces la frecuencia de muestreo máxima de 76.9KHz. [40]

f) Rango de voltaje de Trabajo

Es el rango de voltaje que acepta Arduino en sus entradas analógicas siendo de 0v a 5v.

4.10 Requerimientos Técnicos para el Diseño del Sistema

a) Detector de Cruce por Cero

El sistema de monitoreo utiliza un detector de cruce por cero, con objetivo de detectar cuando la señal registra el cambio de positivo a negativo, es decir cuando la señal realiza un cambio de polaridad.

b) Ganancia / Compensación

La ganancia también conocida como offset multiplica una señal con un factor de ganancia constante y agrega una compensación constante. Están disponibles tanto un offset de entrada como un offset de salida. Un desplazamiento de entrada puede eliminarse automáticamente utilizando la acción de compensación de entrada.

Una aplicación típica de la E/S de ganancia/compensación es la conversión de las señales de salida del sensor. Supongamos que el sensor de nivel tiene la siguiente característica:

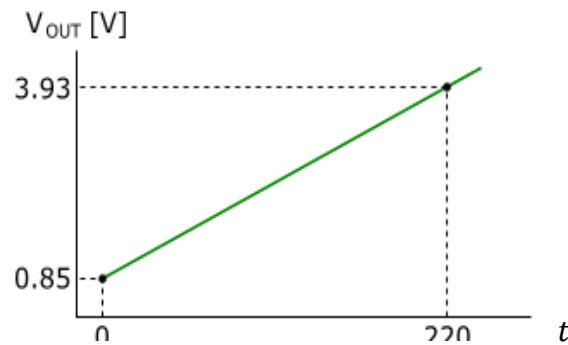


Figura 4.12: Diagrama de una señal aplicada un valor de offset [52]

En la figura 4.12 se observa que el voltaje de salida tiene un desplazamiento de 0.85, entonces para eliminar este valor se compensa con el mismo valor. En conclusión, en valor de salida siempre debe ser restado por el valor de offset (0.85) para obtener el valor correcto.

c) Calibración del Sensor de nivel

La calibración de los sensores de nivel se debe analizar voltaje analógico con un factor de escala de $(V_{cc} / 1024)$ por cm. Un suministro de 5 V rinde $\sim 4.9\text{mV} / \text{cm.}$, y 3.3V rinde $\sim 3.2\text{mV} / \text{cm.}$ La salida está almacenada en el búfer y corresponde a los datos de rango más recientes.

En este caso la entrada de voltaje al sensor es de 5 voltios por lo que se obtiene lo siguiente:

$$\text{Escala del Sensor} = \frac{5\text{ V}}{1024}$$

$$\text{Escala del Sensor} = 4.8828125\text{ mV/cm}$$

Este factor de escala se debe ingresar para linealizar el sensor de nivel, este valor se puede ingresar directamente en el código del programa o por medio de la pantalla HMI.

4.11 Visualización en Tiempo Real

La visualización en tiempo real se realiza utilizando una Pantalla HMI OP320 la cual se encuentra conectada al controllino utilizando en conversor RS232 a TTL que cuenta con características que facilitan la instalación y programación de la pantalla. En la figura 4.13 se muestran las conexiones realizadas al controllino. Se realiza el diseño de usando una placa

de Arduino mega debido a que el controllino tiene como base la placa de este Arduino igual que su procesador.

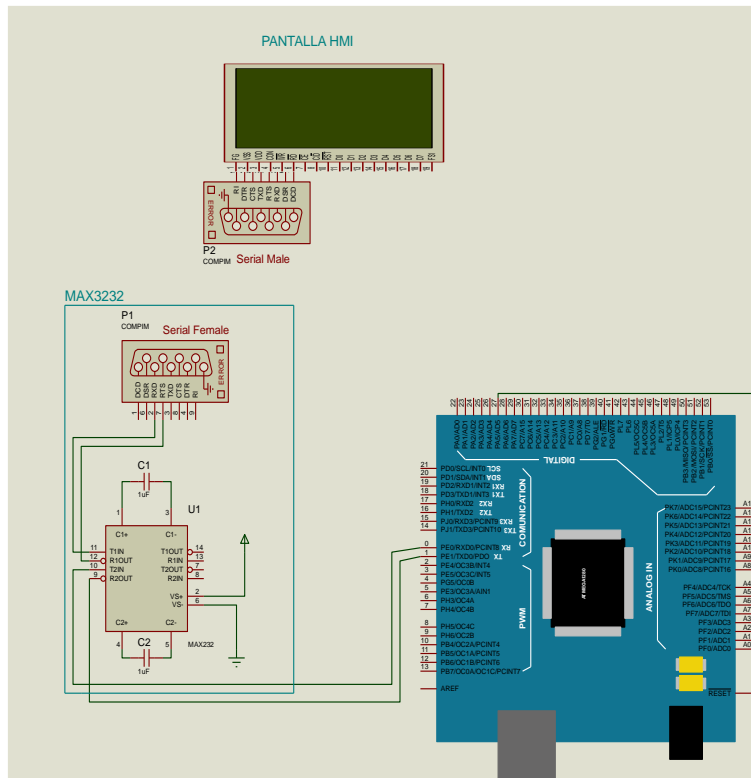


Figura 4.13 : Diagrama de visualización del Nivel de Diesel
Elaborado por: Oscar Analuisa

Los pines de conexión del controllino con convertor MAX3232 se los detalla en la tabla 4.15.

Tabla 4.15: Conexión de Pines del controllino con convertor

Controllino Pines (puerto X2)	Pines del Convertor MAX3232
Pin 1	5 V
Pin 2	GND
RXD2	TX
TXD2	RX

Elaborado por: Oscar Analuisa

Diagrama de conexión del conector DB9

El conector analógico DB9 tiene clavijas, se utiliza principalmente para conexiones en serie, ya que permite una transmisión asíncrona de datos según lo establecido en la norma RS-232.

El diagrama de conexión del conector DB9 se observa en la figura 4.14.

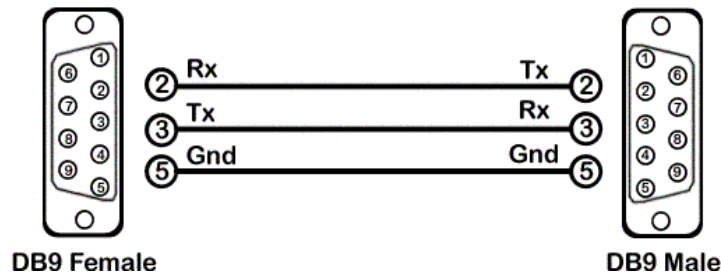


Figura 4.14: Diagrama de conexión del conector DB9 [36]

Este tipo de conexión se establecerá entre el convertidor Max3232 con la Pantalla OP320

4.12 Diseño Electrónico del Sistema de Monitoreo

El Sistema de monitoreo está conformado por un PLC basado en Arduino conocido denominado Controllino, el sensor de nivel y la pantalla HMI para la visualización local. La función realizada por el Controllino es establecer comunicación entre el servidor de thinger.io con los sensores de nivel y la pantalla HMI. El diseño se lo realizó para todo el sistema de almacenamiento de diesel y el prototipo se instaló en un tanque, teniendo en cuenta que el sistema es escalable. Se utilizó hardware y software libre para el proyecto por esta razón se ha trabajado en la plataforma de Arduino. El Controllino funciona a base de un Arduino MEGA, este detalle influye en el diseño de los circuitos esquemáticos, pero es suficiente para detallar las conexiones del sistema.

El diagrama de conexión de los sensores de nivel se visualiza en la figura 4.15 y los pines de entrada se detallan en la tabla 4.16.

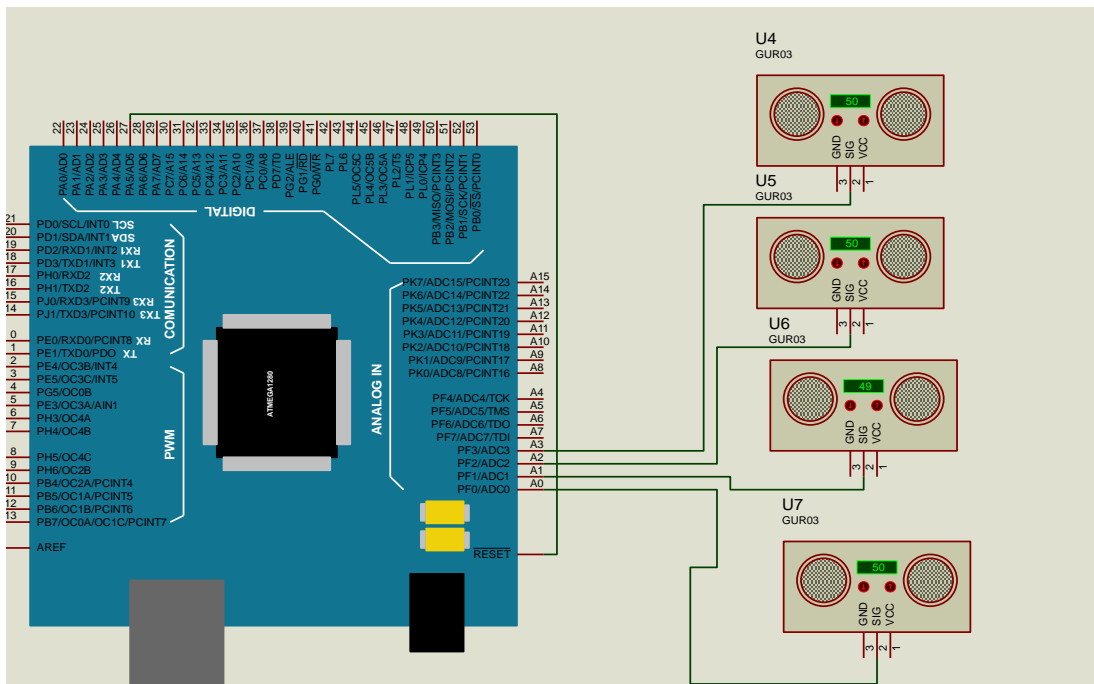


Figura 4.15: Diagrama de conexión de los sensores de nivel

Elaborado por: Oscar Analuisa

El sistema de monitoreo está diseñado para adquirir la información de cuatro sensores de nivel que es el número de tanques que conforma el sistema de almacenamiento de diesel, la señal analógica de cada sensor se conecta a las entradas analógicas del controlador, la alimentación de cada sensor es de 5 voltios de corriente directa que de igual manera salen de VCC y GND del Controlador.

Tabla 4.16: Pines de entrada de los sensores de nivel

Sensor	Pin de entrada
Tanque 1-Generador	Puerto X2 -A6
Tanque 2-Consumo Diario	A0
Tanque 3-Maxima reserva 1	A1
Tanque 4- Máxima reserva 2	A2

Elaborado por: Oscar Analuisa

A continuación, en la figura 4.16 se observa la conexión de las luces indicadoras para cada tanque.

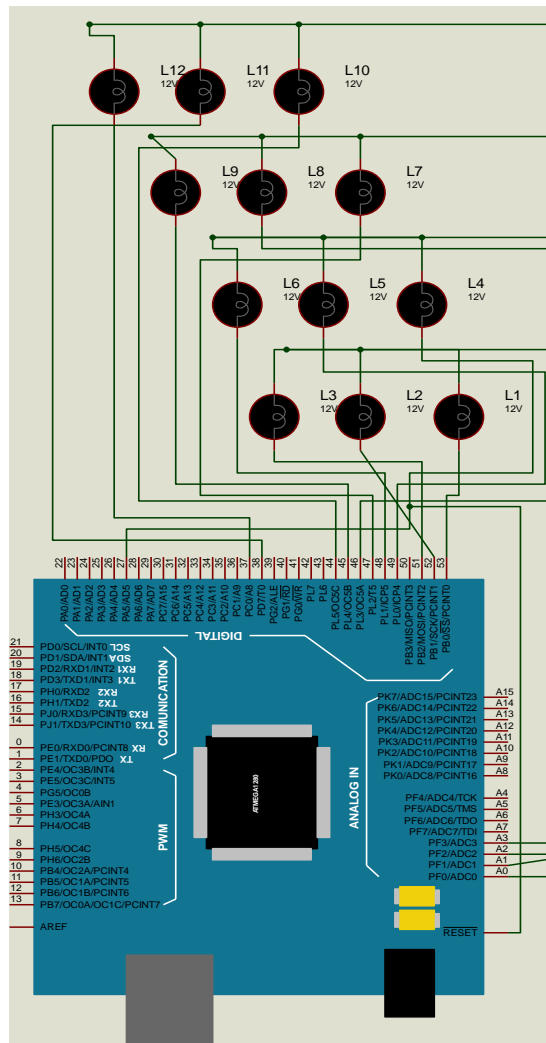


Figura 4.16: Diagrama de conexión de luces indicadoras

Elaborado por: Oscar Analuisa

El sistema total se conforma de 12 luces indicadoras conectadas en las salidas digitales D0 hasta las salidas D11. Se le asigna tres luces a cada sensor, una de color verde para indicar que el nivel de diesel está en nivel alto, una de color amarilla para indicar nivel medio y una luz de color rojo para indicar nivel bajo.

4.13 Procesamiento de datos del sensor de nivel

Este proceso se realizó utilizando el IDE de Arduino y se encuentra desarrollado en base al diagrama de flujo de la figura 4.17 para el Sistema Electrónico de Monitoreo de Niveles de Diesel.

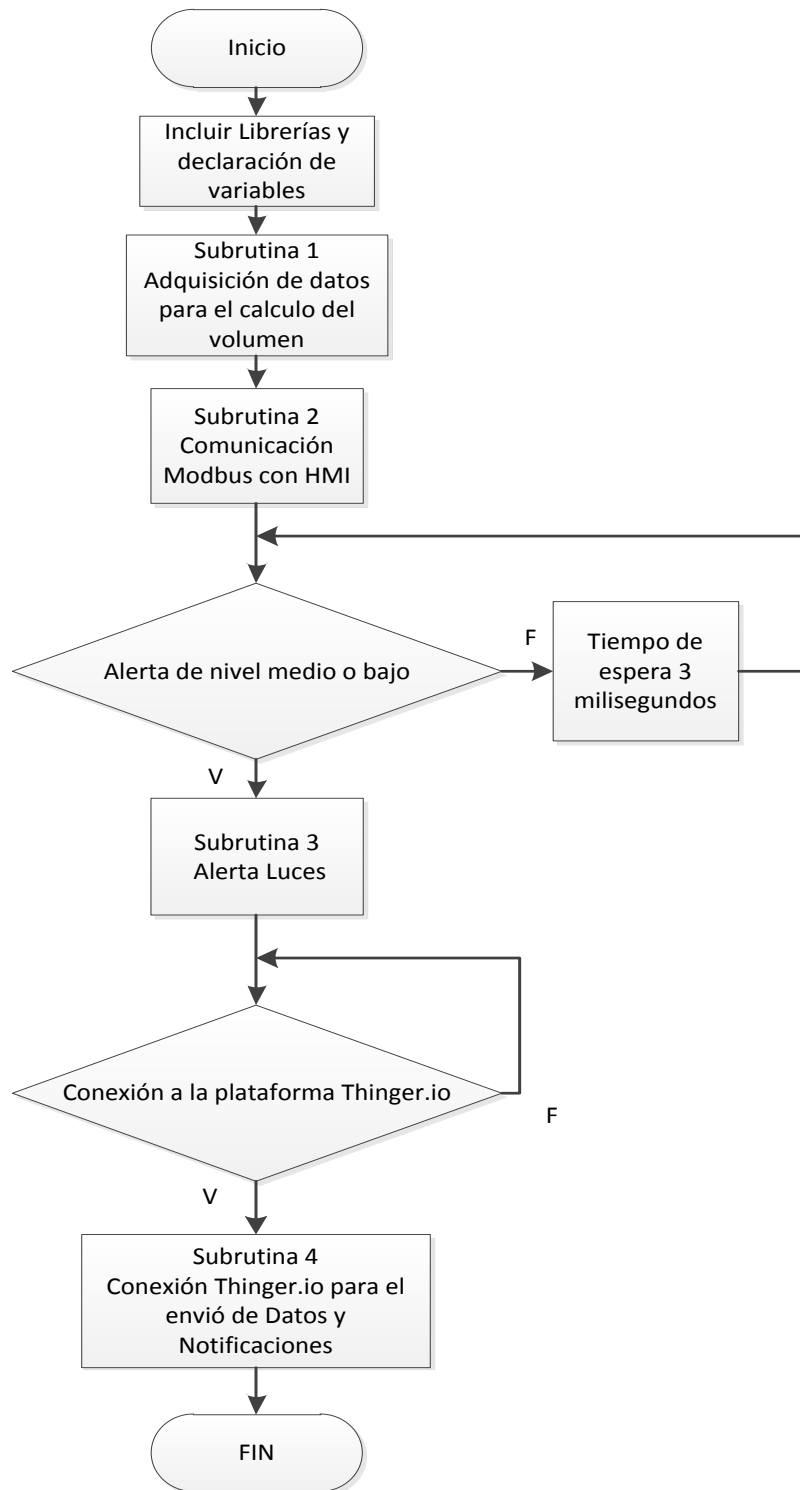


Figura 4.17: Diagrama de flujo del sistema de monitoreo

Elaborado por: Oscar Analuisa

El proceso inicia con la declaración de las variables de entrada y salida. En el Anexo J se indica todas las subrutinas que conforman el sistema electrónico de monitoreo. El proceso de adquisición de datos del transductor de nivel se indica en la subrutina uno, luego el proceso pasa a la subrutina dos que es la comunicación Modbus entre el Controllino y la pantalla HMI, a continuación, se tiene una estructura condicional (If), que en caso de tener niveles apropiados de diesel existe un bucle de tres milisegundos hasta cuando transductor monitoree un nivel medio o bajo y envíe una alerta, para la visualización de la alerta se envía un mensaje a la pantalla HMI y se activa una salida del Controllino para encender una luz, detallado en la subrutina tres. Para el monitoreo remoto se requiere una conexión a la plataforma Thingier.io que se encuentra en la nube, en caso de no tener conexión a internet se mantiene en un bucle hasta que la condición sea verdadera y establecer la conexión a la plataforma como se indica en la subrutina cuatro y finalmente en la subrutina cinco se presenta el proceso del envío de notificaciones con el valor del sensor y el número de galones de diesel.

4.14 Configuración de plataforma Thingier.io

Thingier.io es una plataforma de código abierto que podemos utilizarla gracias a sus propios servidores como en GitHub para instalarla en una máquina propia y ofrecen una cuenta gratuita para utilizar su infraestructura en la nube.

Esta plataforma responde al concepto de Internet de las cosas (en inglés, Internet of Things, abreviado IoT) que es el concepto que se refiere a la interconexión digital de objetos cotidianos con Internet.

Para la programación el Entorno de Desarrollo Integrado (IDE) de Arduino posee una librería en el repositorio oficial. [53]

Los pasos por seguir para la configuración de la plataforma Thingier.io son los siguientes:

- Crear un dispositivo
- Crear un Buckets de datos
- Crear Dashboards
- Crear un Endpoint

A continuación, se describe cada paso de configuración de forma detallada.

Crear un dispositivo

El primer paso para comenzar un proyecto de IoT en Thinger.io, es crear dispositivos que le otorguen acceso para conectar los transductores a la cuenta de la plataforma. Cualquier sensor en Thinger.io debe estar registrado para tener acceso a la nube. Cada dispositivo tiene su propio identificador y una credencial, estos datos están relacionados con la cuenta del usuario.

Para registrar el PLC controllino se debe iniciar sesión en el panel de la consola, luego se debe dar clic en el icono “Devices” que se encuentra el menú de la página de Thinger.io. La imagen del icono se muestra en la figura 4.18.

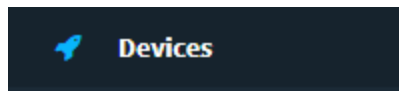


Figura 4.18: Icono Devices para agregar dispositivos

Elaborado por: Oscar Analuisa

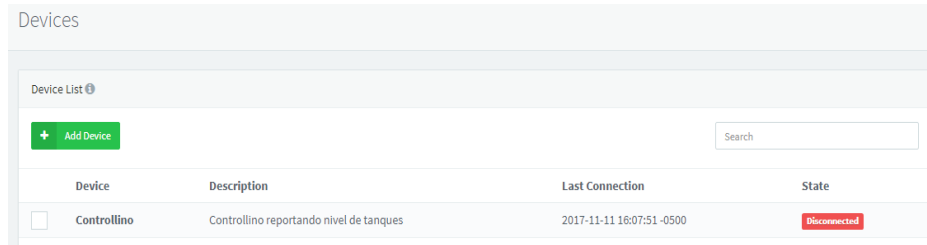
Posteriormente hacer clic en Agregar dispositivo para abrir un formulario e introducir detalles del Controllino.

El formulario 'Add Device' tiene un encabezado con el título 'Add Device'. Debajo hay una sección 'Device details' con tres campos de texto: 'Device Id' con el valor 'Controllino', 'Device description' con el valor 'Controllino reportando niveles de Tanques', y 'Device credentials' con el valor 'Enter device credentials'. Hay un botón 'Generate Random Credential' y un botón verde 'Add Device' con un ícono de checkmark.

Figura 4.19: Detalles del dispositivo agregado

Elaborado por: Oscar Analuisa

Una vez puesto el nombre y detalles del dispositivo como se muestra en la figura 4.19, las credenciales se generan automáticamente. Esta credencial sirve para identificar al dispositivo. En la siguiente imagen se observa que el controllino está en la lista de dispositivos registrados.



Device	Description	Last Connection	State	
<input type="checkbox"/>	Controllino	Controllino reportando nivel de tanques	2017-11-11 16:07:51 -0500	Disconnected

Figura 4.20: Lista de los dispositivos agregados

Elaborado por: Oscar Analuisa

Mientras tanto, se puede abrir el dispositivo en la consola de la nube, simplemente haciendo clic en su identificador (Controllino) en la lista de dispositivos. En la pantalla del dispositivo, podrá ver información sobre el dispositivo, como su dirección IP, estado de conexión o información enviada/recibida en tiempo real. De manera predeterminada, nuestro dispositivo aparece como desconectado como observa en la figura 4.20.

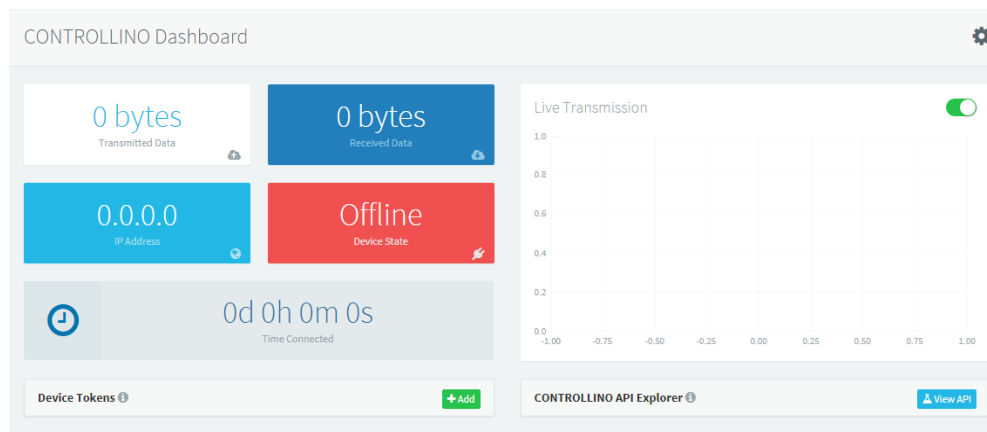


Figura 4.21: Tablero de información del dispositivo desconectado

Elaborado por: Oscar Analuisa

Una vez que el dispositivo se conecta a la cuenta, la interfaz cambia su estado, mostrando que está conectado y está transmitiendo información, observar la figura 4.22.

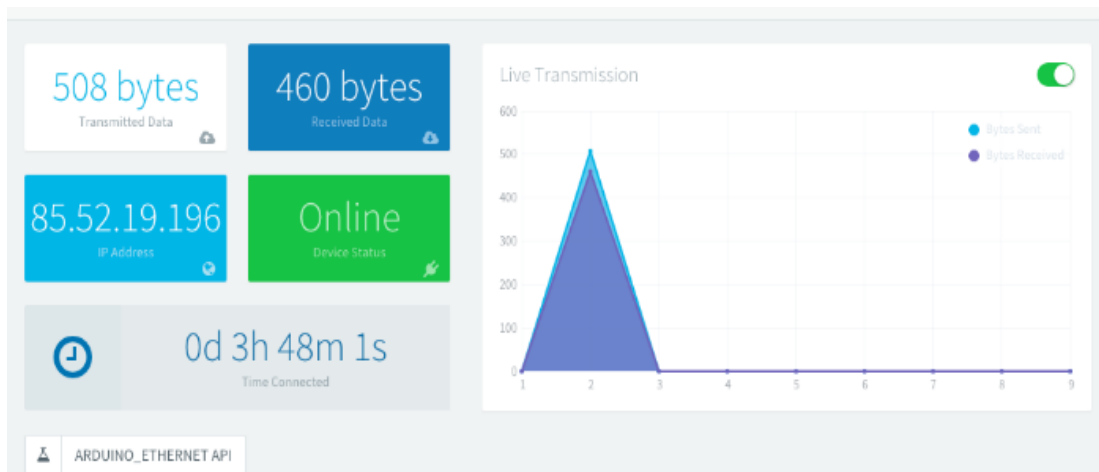


Figura 4.22: Tablero de información del dispositivo conectado

Elaborado por: Oscar Analuisa

Buckets de datos

Un Bucket o contenedor de datos es un tipo de almacenamiento virtual en el que puede guardar información en función del tiempo, como temperatura o humedad y en el caso del proyecto los niveles de diesel.

Esta información se puede usar para graficar información, o se puede exportar en formatos CSV, ARFF y Json.

Para crear el depósito se debe acceder a la función de Data Buckets, haciendo clic en el icono que dice Data Bucket, la figura del icono se muestra en la figura 4.23.

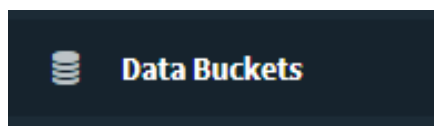


Figura 4.23: Icono Data Bucketss para agregar contenedor de almacenamiento

Elaborado por: Oscar Analuisa

Para crear el depósito, solo presione el botón Agregar depósito, se abre una ventana idéntica a la figura 4.24.

The image shows a web form titled "Add Bucket". It is organized into a "Bucket details" section. This section contains five main components: 1. "Bucket id": A text input field with a placeholder "Enter a bucket id" and an information icon. 2. "Bucket name": A text input field with a placeholder "Enter bucket description" and an information icon. 3. "Bucket description": A text input field with a placeholder "Enter bucket description" and an information icon. 4. "Enabled": A toggle switch currently in the "off" position, with an information icon. 5. "Data Source": A dropdown menu with the placeholder text "Select data source" and an information icon. At the bottom of the form is a green button with a white checkmark icon and the text "Add Bucket".

Figura 4.24: Detalles requeridos para agregar un Bucket

Elaborado por: Oscar Analuisa

En la ventana que aparece es necesario configurar diferentes parámetros tales como:

Bucket id: El bucket id es el identificador único para el bucket, el ID del sistema es: tanque bucket

Bucket name: El bucket name es el nombre del bucket, por ejemplo, en el sistema se denomina: Tanque1.

Bucket description: El bucket description es la descripción del contenedor, hay que llenar una breve descripción del bucket.

Enabled: El enabled activa la grabación del depósito de datos, esta opción puede habilitarse o deshabilitarse solo manualmente.

Fuente de datos: La fuente de datos toma información de un dispositivo específico. En esta opción, el dispositivo debe mantener una conexión permanente con el servidor. Esto agrega algunos beneficios ya que se puede cambiar la frecuencia de muestreo a pedido, sin tener que cambiar el código del dispositivo, Seleccionando la opción Intervalo de muestreo en inglés "Sampling Interval" como se indica en la figura 4.25.

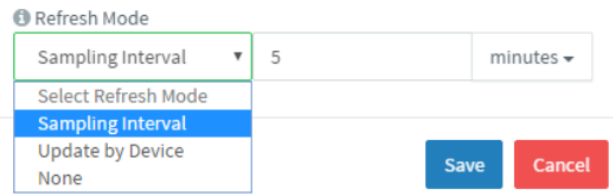


Figura 4.25: Selección de “Samping Interval” para un muestreo de datos

Elaborado por: Oscar Analuisa

Revisar los datos del bucket. - Una vez que el bucket de datos ha sido configurado, y comenzará a registrar datos desde un dispositivo controllino, mostrará la información dentro de una tabla. Cada registro contiene la marca de tiempo del servidor y el valor del registro. Los valores del consumo de diesel almacenados se muestran en formato tabular, La tabla de registro del consumo de diesel se muestra en la figura 4.26.

Bucket Explorer	
Date	Value
2017-11-11T16:02:46.301-0500	260.702
2017-11-11T15:13:09.135-0500	260.702
2017-11-11T15:08:08.410-0500	260.702
2017-11-11T15:06:52.967-0500	260.702
2017-11-11T14:58:36.866-0500	269.646
2017-11-10T12:39:14.890-0500	260.702
2017-11-10T12:29:31.476-0500	251.826
2017-11-10T12:24:09.307-0500	114.159
2017-11-10T12:18:49.293-0500	200.155
2017-11-10T09:17:59.513-0500	33.6454

Figura 4.26: Tabla de registro de información recibida del sensor de nivel

Elaborado por: Oscar Analuisa

Es posible exportar toda su información almacenada en diferentes formatos de archivo, para que pueda procesar los datos sin conexión. De esta manera, se puede acceder a su depósito y configurar el proceso de exportación, seleccionar el tipo de archivo o el rango de exportación, la ventana de opciones se muestra en la figura 4.27. Después de unos minutos recibe un correo electrónico con el archivo de la información.

Figura 4.27: Opciones de importación de los datos del Bucket

Elaborado por: Oscar Analuisa

Crear Dashboards

Un dashboard es una interfaz gráfica de usuario que permite mostrar su información en diferentes figuras y cuadros. Puede configurar los paneles con diferentes widgets, configurar su diseño, dimensión, color y fuentes de datos para generar información valiosa para su negocio o procesos.

Para crear el dashboard, es necesario acceder a la sección paneles, haciendo clic en el icono que se muestra en figura 4.28.

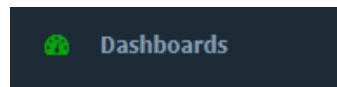


Figura 4.28: Icono para crear interfaz de datos recibidos del transductor

Elaborado por: Oscar Analuisa

A continuación, haga clic en el botón Agregar panel que abrirá una nueva interfaz para ingresar los detalles del panel, la ventana que aparece se muestra en la figura 4.29.

Figura 4.29: Detalles requeridos para agregar un Dashboard

Elaborado por: Oscar Analuisa

Aquí es necesario configurar los siguientes parámetros:

- Dashboard Id: Controllinodashboard
- Dashboard name: Controllino Dashborad
- Dashboard description: Panel de control de controllino

Añadir una pantalla Widget.- Para agregar un Widget o tablero de presentación de datos, el modo de edición debe estar habilitado, aparece un nuevo botón llamado Agregar Widget. Al hacer clic en él se muestra una ventana emergente en la que es posible seleccionar el tipo de widget para agregar en el tablero. Existen diferentes widgets para mostrar información o controlar dispositivos conectados, para el sistema de monitoreo de niveles de diésel usaremos la opción Time Series Chart como se muestra en la figura 4.30 para trabajar en función de los galones censados con respecto del tiempo.

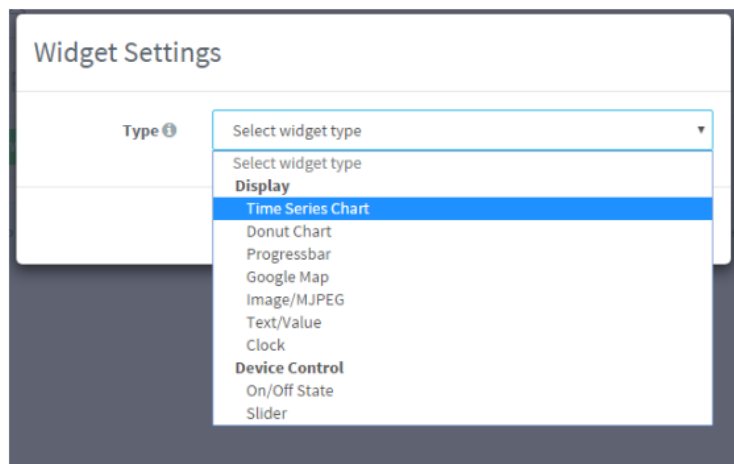


Figura 4.30: Selección de “Tme Sample Chart” para trabajar en función del tiempo

Elaborado por: Oscar Analuisa

Los parámetros configurables son los siguientes:

Title: título opcional para el widget.

Subtitle: subtítulo opcional para el widget.

Background: color opcional para el fondo del widget (predeterminado en blanco).

Chart Input: configure cómo alimentar los valores al gráfico de series temporales. Es posible alimentar la información desde un dispositivo conectado o desde un contenedor de datos.

La figura 4.31 se muestra la selección del dispositivo Controllino. Tener en cuenta que cuando un widget de serie temporal se alimenta desde un dispositivo, no guardará la información si el panel está cerrado o actualizado, ya que solo se trata de datos en tiempo real desde su dispositivo a su tablero.

The image shows a 'Widget Settings' dialog box with the following configuration:

- Type: Time Series Chart
- Title: Volumen Tanque 1
- Subtitle: galones
- Background: #ffffff
- Chart Input: From Bucket
- Select Bucket: ControllinoTanques
- Timeframe: Relative
- Time Period: latest | 10 | hours
- Options: Axis, Legend, Fill Splines, Multiple Axes
- Chart Color: #1E313E

Figura 4.31: Selección de “Tme Sample Chart” para trabajar en función del tiempo

Elaborado por: Oscar Analuisa

Crear un Endpoint (Punto Final)

Un punto final, es el punto de entrada a un servicio, un proceso o cualquier otro destino. Por lo tanto, en Thingiverse.io un punto final define el destino al que los dispositivos pueden llamar para realizar cualquier acción, como enviar un correo electrónico, enviar un SMS, llamar a una API, interactuar con IFTTT, llamar a un dispositivo desde otra cuenta, o llame a cualquier otro punto final HTTP.

Para administrar todos sus puntos finales, es necesario acceder a la sección Endpoints, haciendo clic en el icono Endpoints del menú, el icono se muestra en la figura 4.32.

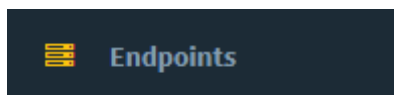


Figura 4.32: Icono para crear un destino para entrega de información

Elaborado por: Oscar Analuisa

A continuación, haga clic en el botón Agregar endpoint, que abre una nueva interfaz para ingresar los detalles del endpoint como indica la figura 4.33.

- Endpoint Id: identificador único para su punto final (el dispositivo debe usar este identificador para activar el punto final).
- Endpoint Description: se completa cualquier descripción o información detallada.
- Endpoint Type: define el tipo de punto final, dependiendo del tipo seleccionado, el punto final presentará diferentes campos. En las siguientes secciones se describen algunos de estos tipos.

A screenshot of a web interface for configuring an endpoint. It features several sections: 'Endpoint Identifier' with a text field containing 'NotificacionTanque'; 'Endpoint Description' with a text field containing 'notificacion de alarma de controllino'; 'Endpoint Type' with a dropdown menu set to 'HTTP Request'; 'Request URL' with a dropdown set to 'POST' and a text field containing 'https://fcm.googleapis.com/fcm/send'; and 'Request Headers' with two entries: 'Content-Type' set to 'application/json' and 'Authorization' set to 'key=AlzaSyAll3KmKP9', each with a red minus button to remove it. At the bottom, there is a plus button and an 'Add Header' button.

Figura 4.33: Detalles del endpoint creado para las notificaciones

Elaborado por: Oscar Analuisa

4.15 Configuración Firebase

Los dos pasos para configurar el servidor Firebase son: crear el proyecto y añadir Firebase a la app.

Crear el proyecto

El primer paso es crear el proyecto en la página <https://firebase.google.com>. Hay que dar clic en Añadir proyecto. La página inicial de fregase se muestra en la figura 4.34.

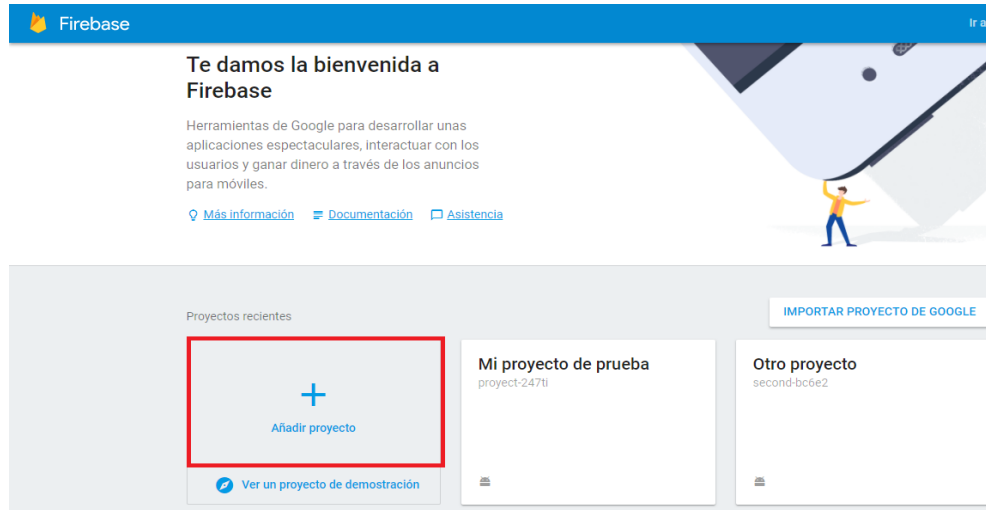


Figura 4.34: Selección de añadir un proyecto

Elaborado por: Oscar Analuisa

Los datos del proyecto llenar con la siguiente información como se ve en la figura 4.35:

Figura 4.35: Detalles del proyecto

Elaborado por: Oscar Analuisa

Nombre del proyecto: NotificacionTanque

ID del proyecto: se crea automáticamente

País/región: Ecuador

Elegir el sistema operativo del móvil, las opciones se muestran en la figura 4.36.



Figura 4.36: Opciones de sistema operativo de la aplicación móvil
Elaborado por: Oscar Analuisa

Se trabaja con el sistema operativo Android y se selecciona la opción haciendo clic en “Agrega Firebase a tu app para Android”.

Añadir Firebase a la aplicación

Para registrar la app se debe llenar con información que se obtiene de la creación de la app realizada en Android Studio. La información requerida se muestra en la figura 4.37.

ID de la app ⓘ	1:352334976030:android:33f569a2446d8d9a
Sobrenombre de la app	notificaión alarma ✎
Nombre del paquete	com.oscaranaluisa.notificacionalarma
Moneda de los informes ⓘ	(USD \$) Dólar estadounidense ▾
Zona horaria ⓘ	(GMT-05:00) GMT-05:00 ▾
Huellas digitales del certificado SHA ⓘ	Tipo ⓘ
CE:96:D7:E6:2C:B4:7A:C7:EA:40:68:3A:E6:05:A3:3C:13:25:BB:0A	SHA-1

Figura 4.37: Detalles para el registro de la app
Elaborado por: Oscar Analuisa

Nombre del paquete: com.oscaranaluisa.notificacionalarma

El nombre del paquete se obtiene

Moneda: Dólar estadounidense

Zona Horaria: GMT-05:00

Huellas digitales del certificado SHA

0000: CE 96 D7 E6 2C B4 7A C7 EA 40 68 3A E6 05 A3 3C

Para obtener la huella digital se debe ingresar al terminal de Android Studio donde se desarrolla la aplicación e ingresar el siguiente comando:

```
keytool -list -v -keystore "%USERPROFILE%\.android\debug.keystore" -alias  
androiddebugkey -storepass android -keypass android
```

Una vez completado toda la información, se pulsa en “Registrar la Aplicación” para continuar.

En el siguiente paso aparece un enlace de descarga de un fichero como se muestra en la figura 4.38.

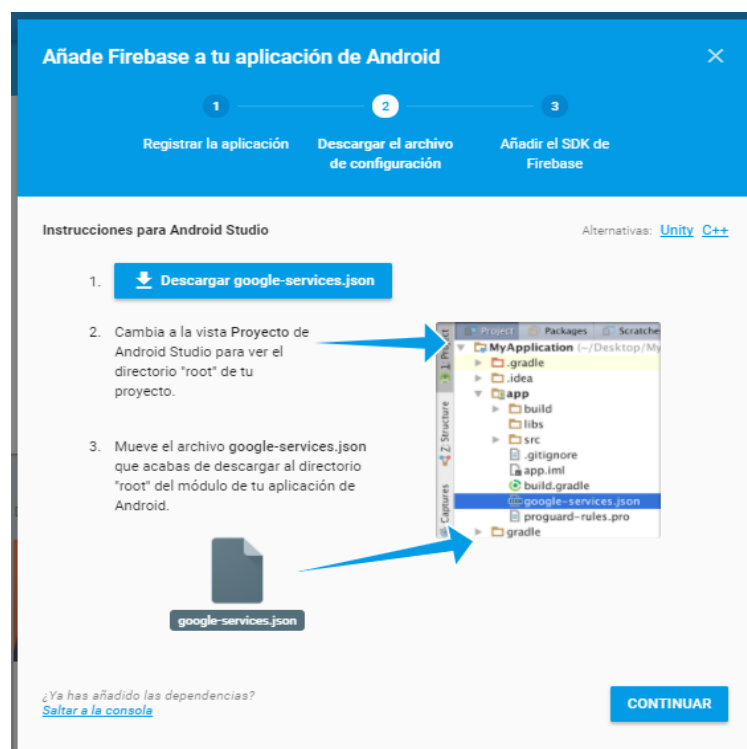


Figura 4.38: Detalles de la descarga del archivo de configuración

Elaborado por: Oscar Analuisa

El fichero a descargar es “google-services.json”. Luego se debe copiar a la carpeta “app” (la carpeta se puede ver en la vista “Project” desde Android Studio).

En caso de trabajar con windows, lo normal es que esté la carpeta en: C:\Users\NombreDeUsuario\AndroidStudioProjects\NombreDelProyecto\app. Este fichero contiene información del proyecto de Firebase que nuestra APP debe conocer.

Después hacer un clic el botón continuar y se llega al último paso.



Figura 4.39: Códigos para añadir en el código de la app

Elaborado por: Oscar Analuisa

Se debe añadir el código classpath de google-services al fichero gradle de proyecto, el código se observa en la figura 4.39. Se lo debe hacer dentro de buildscript -> dependencies.

```

buildscript {
    dependencies {
        //Necesario para firebase
        classpath 'com.google.gms:google-services:3.1.0'
        ....
    }
    ...
}

...

//Necesario para firebase
apply plugin: 'com.google.gms.google-services'

```

Figura 4.40: Códigos classpath para llamar a los servicios de google

Elaborado por: Oscar Analuisa

Al hacer un cambio en gradle, hay que sincronizar pulsando el enlace “Sync now” que se encuentra en la barra de herramientas del programa Android Studio o al momento de compilar el programa este proceso se realiza automáticamente.

El código final del gradle va a quedar como observa en la figura 4.41.

```

1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 24
5      buildToolsVersion '26.0.2'
6      defaultConfig {...}
14     buildTypes {
15         release {
16             minifyEnabled false
17             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
25         exclude group: 'com.android.support', module: 'support-annotations'
26     })
27     compile 'com.android.support:appcompat-v7:24.2.1'
28     compile 'com.android.support.constraint:constraint-layout:1.0.2'
29     testCompile 'junit:junit:4.12'
30     compile 'com.google.firebase:firebase-messaging:11.0.2'
31 }
32
33 apply plugin: 'com.google.gms.google-services'

```

Figura 4.41: Código final del gradle nos va a quedar de la siguiente manera

Elaborado por: Oscar Analuisa

Después de que la sincronización termine con éxito, hay que volver al asistente de Firebase y pulsar “Finalizar”.

En pocos segundos se ve cómo la app se ha añadido a las aplicaciones del proyecto de Firebase en el panel de control con el nombre “NotificacionTanque” como se muestra en la figura 4.42.

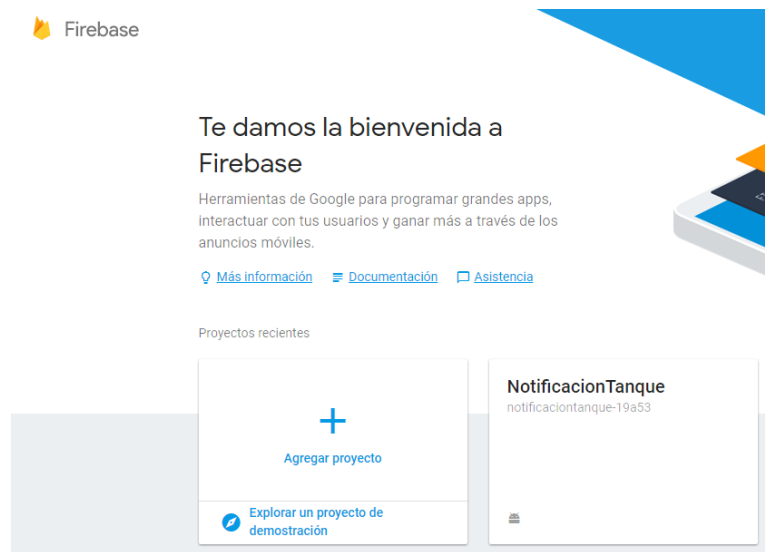


Figura 4.42: Visualización del proyecto creado en firebase

Elaborado por: Oscar Analuisa

4.16 Diseño de aplicación para Android

La programación para el Sistema de Monitoreo de Niveles de Diesel se ha utilizado el IDE oficial de Android, llamado Android Studio y se encuentra desarrollado en base a los siguientes pasos:

Interfaz de la aplicación

Para el diseño de la interfaz se utiliza la ayuda de Android Studio que permite visualizar la interfaz antes de instar la app en el dispositivo móvil. Esta opción de ayuda muestra una vista previa del diseño de la aplicación que crea el código HTML, adicionalmente genera unas ventanas con información denominadas capturas de la aplicación donde presenta la relación entre el código y el elemento creado, esto ayuda a verificar la programación correcta de cada elemento de la aplicación.

La vista previa de la pantalla programada para el sistema de monitoreo se muestra en la figura 4.43.



Figura 4.43: Pantalla del sistema de monitoreo desarrollada en Android Studio

Elaborado por: Oscar Analuisa

Fase de Codificación

En esta fase primeramente se analiza los requerimientos de desarrollo que se necesitan para crear un proyecto y segundo empezar a programar en Android Studio.

Una breve descripción de los códigos de las clases más relevantes dentro de la aplicación se muestra a continuación:

- Java MainActivity (Clase Principal, llamada diferentes objetos)
- Java miFirebaseInstanceService (Conexión de los servicios de Firebase)
- Java miFirebaseMessagingService (Conexión con los servicios de mensajería de Firebase)
- El Archivo XML Activity_Menu (Crear interfaz para el usuario)

A continuación, se detallan los dos pasos de la fase de codificación.

a) Requerimientos de desarrollo

Para el desarrollo se detalla las herramientas y servicios o aplicaciones que se necesita para empezar a desarrollar un proyecto en Android.

- Entorno de Desarrollo para Aplicación.
- JDK Java.
- Android Studio (IDE)
- ADT (Plugins)
- Android SDK (Librerías y Paquetes)
- ADV (Emulador Android)
- Webservice (JSON)
- Api Firebase Cloud Messaging

b) Desarrollo del Sistema de monitoreo.

En este apartado se muestra el código de las “llamadas” o líneas importantes que son necesarias acotar de las clases creadas en el proyecto:

Descripción la clase Java MainActivity

Esta clase se encarga de:

- Llamar al Activity o Ventana principal si es diferente de nulo para crear alerta
- Establecer la comunicación con firebase.
- Crear intents, un intent es el enlace de unas o más aplicaciones, con datos de firebase
- Obtener la suscripción.
- Botón Navegar, se redirecciona a la consola de thinger.io

El código completo de la clase MainActivity con la descripción se muestra en el Anexo B.

Descripción de la Clase Java miFirebaseInstanceService

Esta clase se encarga de:

- Para que trabaje esta clase se realiza el llamado a la instancia creada por firebase del archivo json.
- Enlaza la aplicación con la plataforma firebase.

- La etiqueta NOTICIAS es el Id string para la conexión

El código completo de la clase Java miFirebaseInstanceService con la descripción se muestra en el Anexo C.

Descripción de la Clase Java miFirebaseMessagingService

Esta clase se encarga de:

- Conexión a la plataforma Firebase usando la etiqueta NOTICIAS
- Recepción de la notificación si en la base de datos tiene información de alerta
- Alerta de notificación usando FLAG_ONE_SHOT
- Creación de la Notificación

El código completo de la clase Java miFirebaseMessagingService con la descripción se muestra en el Anexo D.

Descripción del Archivo XML Activity_Menu.xml (Menú Principal)

Este código genera la interfaz para el usuario como se muestra en la figura 4.44:



Figura 4.44: Interfaz para el usuario

Elaborado por: Oscar Analuisa

El código completo del Archivo XML Activity_Menu.xml se muestra en el Anexo E.

4.17 Implementación Interface HMI.

La implementación de la interface se realiza utilizando el software de programación OP20 Series Edit Tool V8.0n, este software presenta todas las características de los programas diseñados bajo el sistema operativo Windows, es decir, barras de menú y ventanas. La pantalla OP320-S, es compatible con distintas marcas de controladores programables. Las características y especificaciones de la pantalla se detallan en la tabla 4.17.

Tabla 4.17: Características de la pantalla OP 320-S [54]

CARACTERISTICAS	ESPECIFICACIONES
Área de Visualización	192 × 64
Tipo	3. 7" LCD
Memoria	1MB Flash, 1KB SRAM
Vida útil	Más de 20,000 horas, una temperatura ambiente de 25,24- horas de operación
Color	STN LCD
Tamaño del personaje	Fuentes de mapa de bits, las fuentes de vectores
Ventana	60 nits
Comunicación	RS 232 / RS 485
Tiempo Real	Si
Voltaje de Entrada	24VDC
Consumo de Corriente	4 W
Temperatura de Operación	0°C ~ 50 °C
Temperatura de Almacenamiento	10°C ~ 60 °C
Tension Admissible	10ms(real de la pérdida de potencia es de menos de un segundo)
Resistencia de Aislamiento	1000Vp-p voltage
Vibración Maxima Soportada	10-25Hz (X, Y, Z cada dirección 2G cada 30 min)
Datos	sram 1kb
Humedad de Funcionamiento	10% rh a 90% rh(no- de condensación)

A continuación, en la figura 4.45 se muestra el diagrama para la comunicación entre la pantalla HMI y el Controllino. La conexión se realiza a través del conversor MAX3232, el conector DB9 se conecta con la pantalla y el puerto TTL va al controllino.

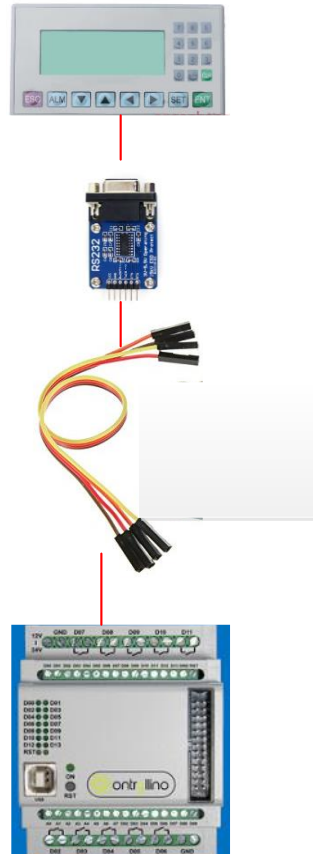


Figura 4.45: Esquema de comunicación Pantalla HMI y el Controllino

Elaborado por: Oscar Analuisa

Programación de la pantalla THINGET OP 320-S (Operator Panel 320-S)

Antes de iniciar la programación hay que verificar que la pantalla este alimentada a 24V, tener conectado el cable de comunicación serial RS232 entre el PC y la pantalla y tener instalado el software OP20 Edit Tool, una vez verificado lo mencionado seguir los pasos que se detallan a continuación.

a) Selección del modelo de la Operate Panel

Se debe seleccionar el tipo de panel view, que se debe utilizar para el diseño del HMI, como se indica en la figura 4.46.

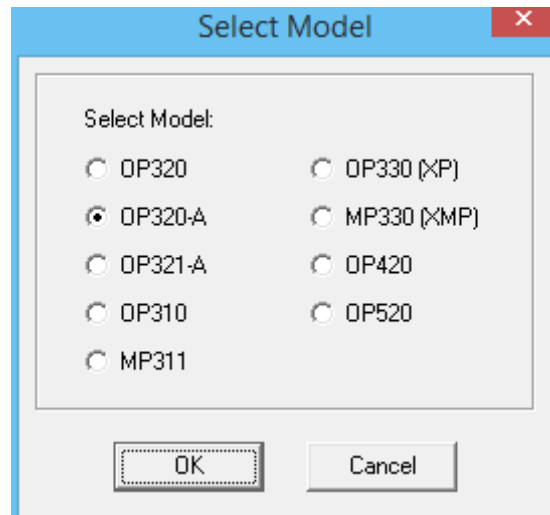


Figura 4.46: Selección del modelo de la OP320-A
Elaborado por: Oscar Analuisa

b) Selección del PLC

Cuando se realice la configuración del software OP20, se debe seleccionar la marca del PLC, seleccionando Modbus Slave como se observa en la figura 4.47:

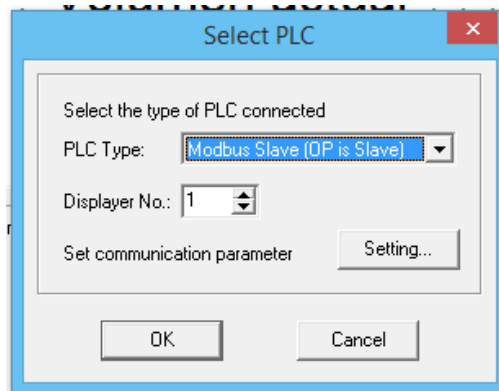


Figura 4.47: Selección del PLC de la OP320-A
Elaborado por: Oscar Analuisa

c) Añadir Screen (Pantalla)

En figura 4.48, se puede observar la cantidad de screens realizadas o construidas en el proyecto, en caso de ser necesario existen los botones NEW para crear un nuevo screen y DELETE sirve para borrar screen.

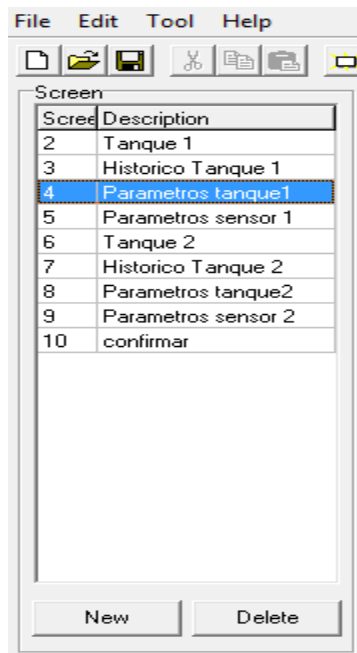


Figura 4.48: Añadir o Borrar Screen

Elaborado por: Oscar Analuisa

d) Atributos de screen

Luego de agregar un nuevo screen aparece una ventana de atributos, se debe ingresar un nombre o descripción y cual screen va antes y cual a continuación. En la figura 4.49 se muestra la información ingresada para el screen.

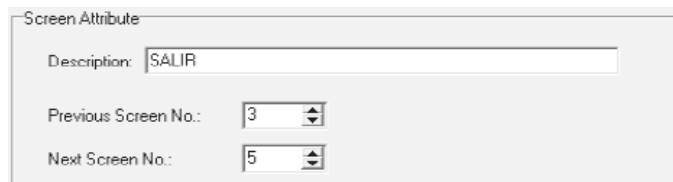


Figura 4.49: Atributos del Screen

Elaborado por: Oscar Analuisa

e) Barra de herramientas

En la barra de herramientas se encuentra distintas opciones, la más relevante es la opción “Subir Programa” que sirve para cargar el programa a la memoria de la pantalla. La barra de herramientas se observa en la figura 4.50:

Subir el programa



Figura 4.50: Barra de Herramientas

Elaborado por: Oscar Analuisa

f) Barra de funciones

En la barra de funciones se encuentra varias opciones, pero en el proyecto se utiliza la opción de agregar texto y la de agregar tanque, para añadir estos elementos al proyecto se debe arrastrar el elemento requerido hacia el área de trabajo. En la figura 4.51 se muestra la barra de funciones.

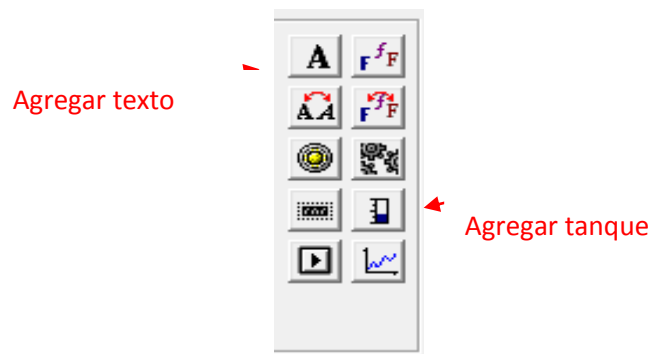


Figura 4.51: Barra de funciones

Elaborado por: Oscar Analuisa

g) Área de trabajo

El área de trabajo es aquella que sirve de visualización al momento de programar y poner en práctica el HMI como se observa en la figura 4.52.

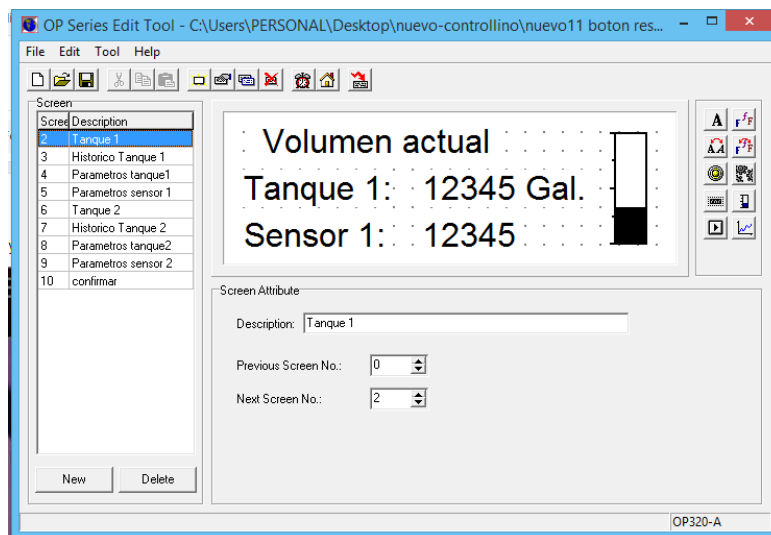


Figura 4.52: Área de trabajo

Elaborado por: Oscar Analuisa

h) Conexión de variables

Los datos que se visualizan en la pantalla HMI se realiza a través de comunicación Modbus desde y hacia el Controllino.

Modbus es un protocolo industrial que fue desarrollado en 1979 para hacer posible la comunicación entre dispositivos de automatización. Originalmente implementado como un protocolo al nivel de la aplicación con la finalidad de transferir datos por una capa serial. [55]

Para configurar la conexión de variables se realiza seleccionando el número de registro de memoria en el cual el protocolo va a transmitir ese valor.

En la figura 4.53, se visualiza como el valor en galones del tanque 1 tiene configurado el registro número 6.

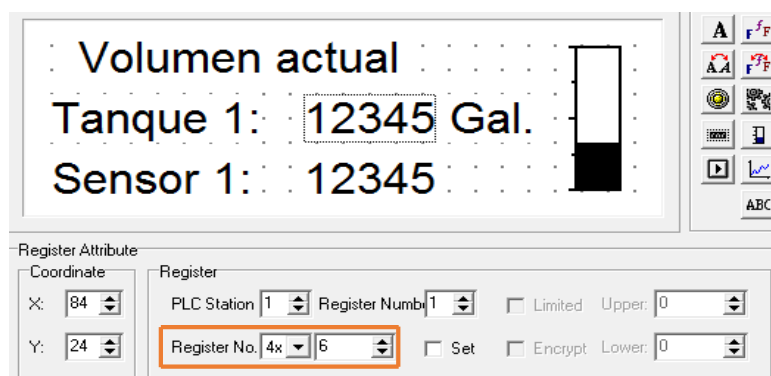


Figura 4.53: Selección del registro 6 para el valor de galones del tanque 1

Elaborado por: Oscar Analuisa

Para el valor del sensor se selecciona el registro número 7 como se indica en la figura 4.54.

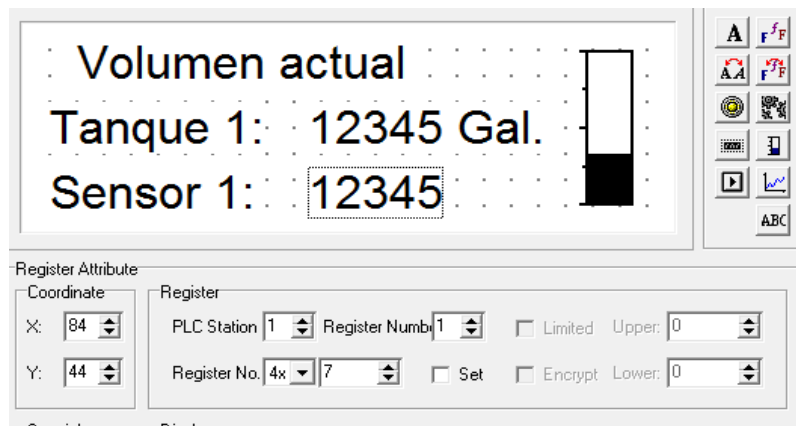


Figura 4.54: Selección del registro 7 para el valor del sensor del tanque 1

Elaborado por: Oscar Analuisa

Hay que tener en cuenta que en la programación debe ser configurado con el mismo número de registro Modbus.

```
ModbusSlaveRegisters[6]=(int)VolActualTanque1;
ModbusSlaveRegisters[7]=(int)sensorValue1;
```

Figura 4.55: Sección de código de programación de la selección de los registros 6 y 7

Elaborado por: Oscar Analuisa

En la figura 4.55 se observa al final del código que el volumen actual (VolActualTanque1) tiene el registro 6, y el valor del sensor (sensorValue1) tiene el registro 7. Lo cuales deben coincidir con todos los valores de registro seleccionados programados en la pantalla.

A continuación, en la tabla 4.18, se presenta la tabla de todos los registros para cada tanque para establecer la comunicación.

Tabla 4.18: Tabla de registros para establecer la comunicación Modbus

	Variable	Nº de Registro
TANQUE 1 GENERADOR	Radio	0
	Longitud	1
	Escala	2
	Alarma Bajo	3
	Alarma Medio	4

	Offset	5
	Volumen	6
	Valor de sensor	7
TANQUE 2 CONSUMO DIARIO	Radio	9
	Longitud	10
	Escala	11
	Alarma Bajo	12
	Alarma Medio	13
	Offset	14
	Volumen	15
	Valor de sensor	16
TANQUE 3 MÁXIMO RESERVORIO 1	Radio	17
	Longitud	18
	Escala	19
	Alarma Bajo	20
	Alarma Medio	21
	Offset	22
	Volumen	23
	Valor de sensor	24
TANQUE 4 MÁXIMO RESERVORIO 2	Radio	25
	Longitud	26
	Escala	27
	Offset	28
	Alarma Bajo	29
	Alarma Medio	30
	Volumen	31
	Valor de sensor	32

Elaborado por: Oscar Analuisa

4.18 Implementación del Sistema

La implementación del Sistema sigue la estructura del diagrama mostrado en la figura 4.56.

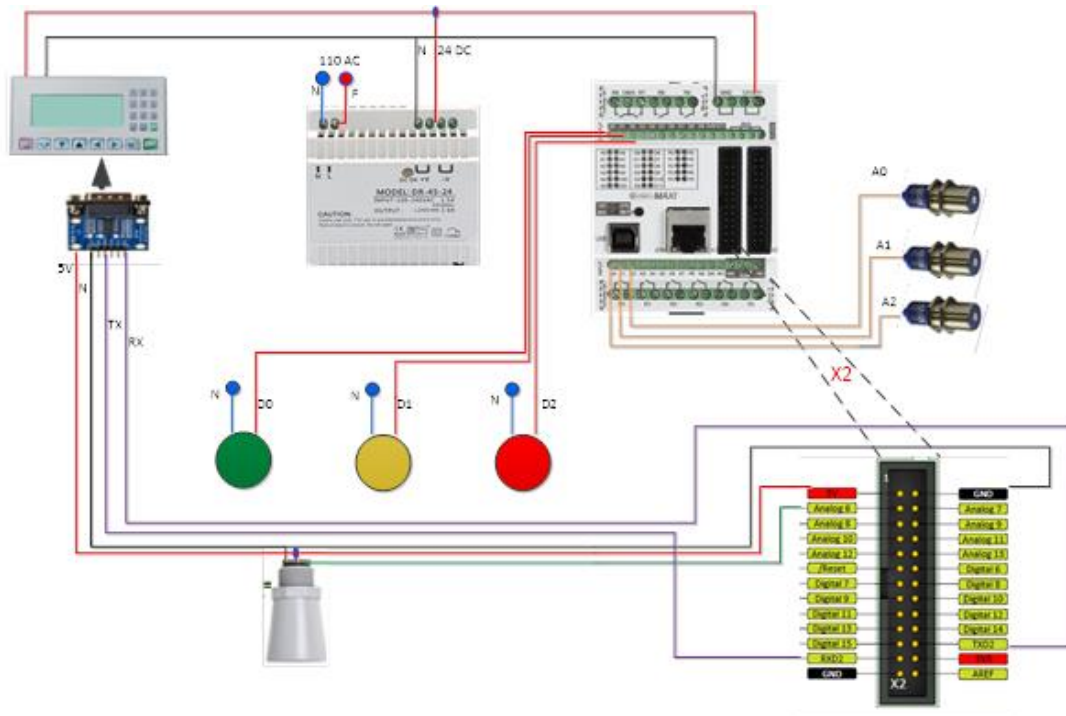


Figura 4.56: Diseño del sistema de conexión de los elementos

Elaborado por: Oscar Analuisa

En el esquema se puede visualizar la conexión de cada elemento utilizado en el desarrollo del sistema la única diferencia es que para el prototipo en lugar de ir los cuatro sensores de nivel solo va la conexión de un sensor.

El proceso de implementación inicia con la colocación del riel din para posteriormente situar los componentes dentro del case como se muestra en la figura 4.57.



Figura 4.57: Colocación los elementos en el riel din

Elaborado por: Oscar Analuisa

Posteriormente se procede a conectar la pantalla HMI con el convertidor MAX3232, juntamente con la alimentación de la pantalla de 24 voltios, todo el cableado se debe utilizar cable flexible número 20, como se muestra en la figura 4.58.



*Figura 4.58: Conexión MAX3232 y alimentación a la pantalla OP320-A
Elaborado por: Oscar Analuisa*

En la figura 4.59 se muestra una ubicación ordenada de los elementos que conforman el sistema de monitoreo.



*Figura 4.59: Distribución ordenada de los elementos en el riel din
Elaborado por: Oscar Analuisa*

Una vez realizado la conexión de los elementos según el diagrama de la figura 64, se procede a realizar las pruebas de funcionamiento para posteriormente realizar la instalación en campo.

El manual de instalación del Sistema Electrónico para el Monitoreo del Almacenamiento de Diesel del Hospital General Ambato se observa en el Anexo J.

4.19 Ajustes del Sistema y Pruebas de Funcionamiento.

La verificación de la visualización local es importante porque en caso de perder conectividad con la nube no se podrá visualizar el nivel de diesel remotamente para lo cual se instala tres luces indicadoras, verde para identificar un nivel alto de diesel, amarillo para un nivel medio y rojo para un nivel bajo. Las luces van conectadas a 24 voltios, de igual manera utilizando cable flexible número 20. Las luces instaladas se muestran en la figura 4.60.



*Figura 4.60: Instalación de tres las luces indicadoras para las alertas
Elaborado por: Oscar Analuisa*

A continuación, en la figura 4.61, se muestra en el panel frontal las tres luces indicadoras para las alertas instaladas y de igual manera se prueba el funcionamiento.



*Figura 4.61: Panel frontal instaladas las tres luces indicadoras
Elaborado por: Oscar Analuisa*

4.19.1 Calibración y Ajustes del Sistema de Medición.

Para la calibración del sensor se requiere insertar datos a través de la pantalla HMI de forma manual. El ingreso de las dimensiones del tanque es realizado por teclado al igual que los valores de alerta de nivel medio y bajo, hay que tener en cuenta que en el código de programación cada valor esta dividido para 100 con el objetivo de facilitar el ingreso de las dimensiones sin utilizar decimales, los datos ingresados se muestran en la figura 4.62.

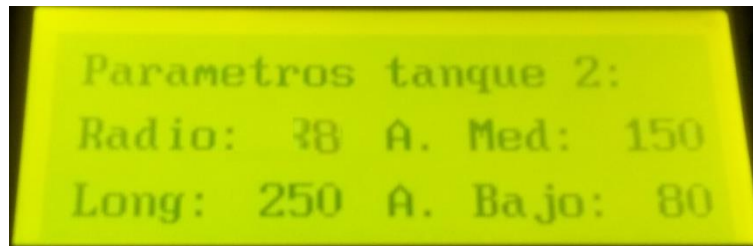


Figura 4.62: Ingreso de parámetros y de alarmas para el tanque 1

Elaborado por: Oscar Analuisa

La alerta de nivel medio es el valor de la mitad de la capacidad del tanque de 300 galones, el nivel bajo es un promedio del 1/4 de la capacidad total del tanque.

El valor de compensación (offset) y de la escala del sensor se configura por teclado, los valores ingresados se muestran en la figura 4.63.

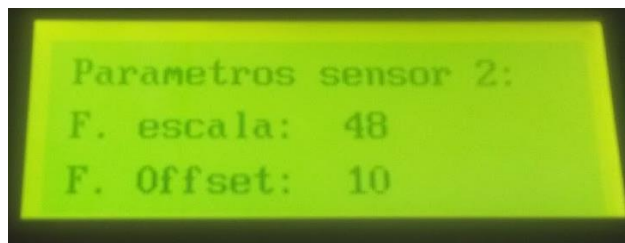


Figura 4.63: Ingreso de los valores de offset y escala del sensor

Elaborado por: Oscar Analuisa

El valor de la escala es proporcionado por el documento técnico del sensor, el datasheet del sensor XL-TrashSonar se muestra en el Anexo G. El valor de offset es de 10 cm, este valor es la altura de la tapa del tanque en donde se encuentra colocado el sensor. En la figura 4.64 se muestra el tanque de 300 galones con la tapa cerrada.

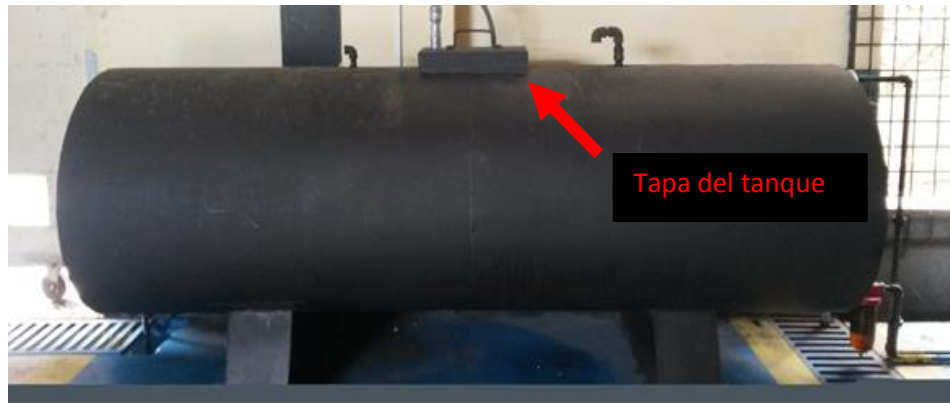


Figura 4.64: Tanque de almacenamiento de diesel de 300 Gal.

Elaborado por: Oscar Analuisa

Una vez calibrado las características del tanque se procede a realizar las pruebas de funcionamiento.

4.19.2 Pruebas de funcionamiento

Verificando el correcto funcionamiento del sistema se procede a llenar el tanque y a realizar una tabla de los datos recibidos por el sensor y el valor en galones, este proceso se realiza utilizando la siguiente fórmula del cálculo de volumen de un tanque horizontal:

$$V = L \left[\frac{\pi R^2}{2} + (h - R)\sqrt{2Rh - h^2} + R^2 \text{arc sen} \frac{-R + h}{R} \right]$$

Donde:

V = Es el volumen.

H = Es la altura del líquido

R = Es el radio del tanque

L = Es la longitud del tanque

En la tabla 4.19 se puede observar una variación lineal ascendente del número de galones en el proceso de llenado, el tanque empezó el llenado en 67 galones.

Tabla 4.19: Datos recibidos del sensor

DATOS RECIBIDOS DEL SENSOR		
Galones	Unidades Crudas del sensor	Alertas Generada
67	137	Alarma Bajo
100	132	Alarma Medio
128	128	Alarma Medio
233	29	Sin Alerta
240	26	Sin Alerta
271	11	Sin Alerta
277	7	Sin Alerta

Elaborado por: Oscar Analuisa

Cuando empieza la trasmisión de los datos del controllino, los datos leídos por el sensor están siendo subidos a la nube para ser mostrados remotamente, los datos se puede vizualizar entrando a la plataforma de thinger.io. Los bytes transmitidos y recibidos, la dirección Ip, y el tiempo de conexión se muestran en la figura 4.65 que son datos aparecen al establecer una comunicación.

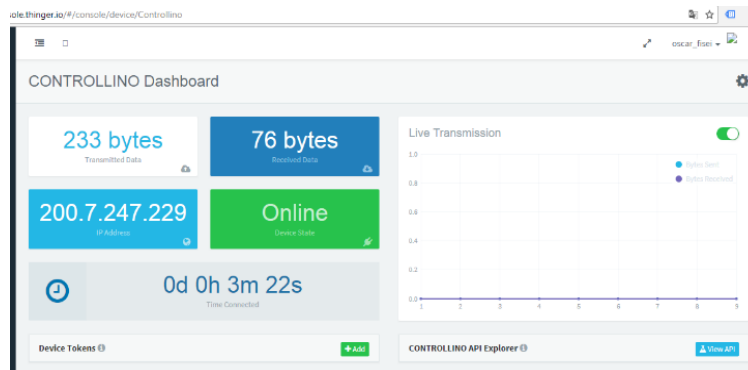


Figura 4.65: Datos de trasmisión

Elaborado por: Oscar Analuisa

El valor del sensor y el número de los galones se puede observar entrando en la pestaña dashboard de la página de thinger.io, en la figura 4.66 se observa la recepción correcta de los valores del tanque uno.

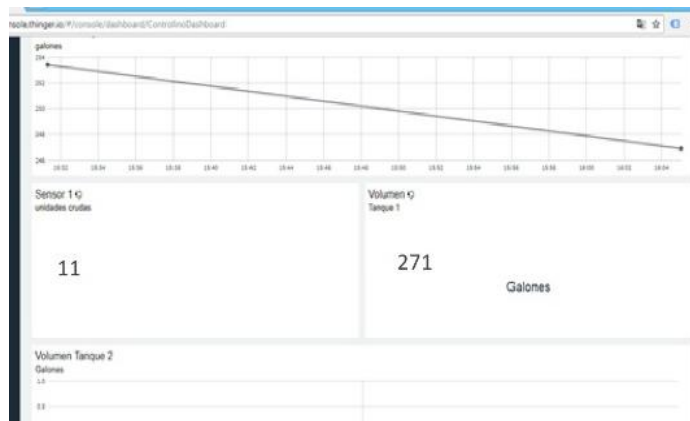


Figura 4.66: Panel de monitoreo del tanque 1

Elaborado por: Oscar Analuisa

Para la verificación de recepción de alertas al dispositivo móvil de un segundo tanque se utilizó un potenciómetro para variar la entrada al Controllino, en la figura 4.67 la alerta recibida en el móvil.



Figura 4.67: Recepción de alertas al móvil

Elaborado por: Oscar Analuisa

Por otra parte, la alerta de forma local se visualiza en la pantalla HMI como se muestra en la figura 4.68, adicionalmente en la figura 4.69 se observa la luz indicadora roja que se enciende al tener un nivel bajo.

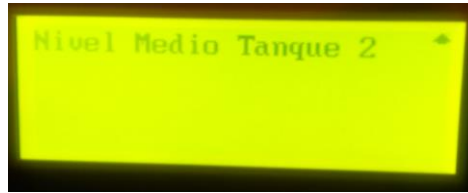


Figura 4.68: Recepción de alertas en la pantalla HMI

Elaborado por: Oscar Analuisa



Figura 4.69: Encendido de luz roja indicando nivel bajo del tanque

Elaborado por: Oscar Analuisa

Una vez implementado el sistema de monitoreo en campo y terminadas las pruebas para verificar el correcto funcionamiento, se procede a colocar una etiqueta para la señalización del equipo y finalmente dar la capacitación al personal del área de mantenimiento. Concluido todo lo mencionado se hace la entrega el sistema funcional satisfaciendo todos los requeridos y obteniendo los resultados esperados. El prototipo instalado se muestra en la figura 4.70.



Figura 4.70: Prototipo completo del sistema de monitoreo de níveles de diesel

Elaborado por: Oscar Analuisa

4.20 Funcionamiento del prototipo

El prototipo se encuentra instalado en la casa de máquinas para el monitoreo del tanque de 300 galones el cual alimenta al generador, desde la pantalla es posible acceder a los valores del sensor y a los galones actuales del tanque. El personal del área de mantenimiento puede monitorear el número de galones de diesel remotamente o localmente.

La estructura del case del sistema de monitoreo se observa en la figura 4.71, la caja es de metal doble fondo y sus medidas son: 45cm de alto por 30cm de ancho y 15cm de profundidad.

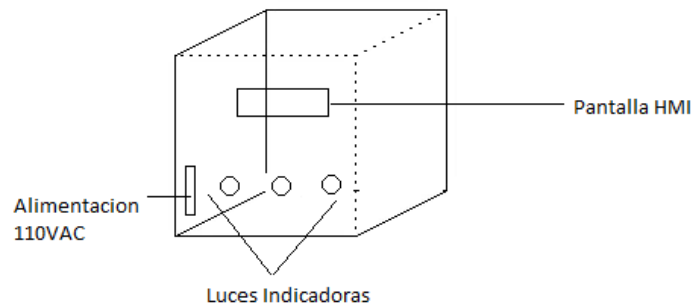


Figura 4.71: Estructura del case del prototipo

Elaborado por: Oscar Analuisa

Funcionamiento

El prototipo trabaja a 110 voltios, en la pantalla HMI se puede observar el valor del sensor y el valor de los galones actuales, Además en el panel frontal tiene tres luces una verde que marca un nivel alto de diesel, una luz amarilla que marca un nivel medio y una luz roja que marca un nivel bajo.

Los datos de alertas son los siguientes:

Nivel alto entre: 151 a 300 Gal.

Nivel medio entre: 80 a 150 Gal.

Nivel bajo entre: 0 a 79 Gal.

Configuración

El ingreso de algún dato de calibración se realiza a través de la pantalla OP 320-s. La pantalla es sencilla, pero posee los botones necesarios como se muestra en la figura 4.72.



Figura 4.72: Pantalla HMI OP 320-S

Elaborado por: Oscar Analuisa

- a) Para realizar cualquier configuración del sistema se debe ir a la pantalla (screen) donde muestra el modo configuración, la pantalla de configuración se muestra en la figura 4.73.

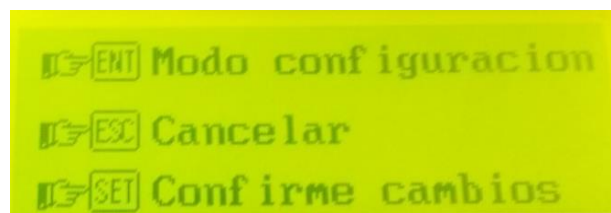


Figura 4.73: Pantalla de configuración

Elaborado por: Oscar Analuisa

- b) Presionar Enter (Botón ENT) para empezar
- c) Ubicar en el screen el cual se desee editar y presionar (SET).
- d) Digitar el valor de los datos que se desee cambiar.
- e) Regresar a la pantalla de inicio (Pantalla de modo configuración).
- f) Presionar SET.
- g) En caso de que no desee realizar ningún cambio presionar ESC.

4.21 Presupuesto.

Para la elaboración del proyecto se necesitó los materiales que se detallan con su respectivo costo en la tabla número 4.20.

Tabla 4.20: Presupuesto del prototipo del sistema de monitoreo

Presupuesto del Sistema					
Ítem	Descripción	Unidad	Cantidad	Precio Unitario	Valor Total
1	Sensor Ultrasónico	c/u	1	99	99
2	Controllino Maxi Arduino PLC Industrial Controlador	c/u	1	250	250
3	Caja Doble Fondo 400x300x150 mm	c/u	1	18.75	18.75
4	Pantalla HMI OP230	c/u	1	95	95
5	Fuente de alimentación 24 Voltios	c/u	1	30	30
6	Luces indicadoras 24 VDC	c/u	3	0.80	2.40
7	Convertor RS232 a TTL	c/u	1	10	10
8	Cable flexible #20	Metro	5	0.5	2.5
9	Tornillos	c/u	20	0.05	1
10	Cable UTP	metro	20	0.75	15
11	Cinta Adhesiva	c/u	2	0.5	1
12	Adaptador PVC 3/4	c/u	1	1	1
13	Breaker	c/u	1	3	3
14	Conector RJ45	c/u	2	0.5	1
15	Señalización Sticker	c/u	1	2	2
16	Conector IDC-26 con bornera	c/u	1	3.5	3.5
TOTAL					535.15
Imprevistos (5 %)					26.76
VALOR TOTAL					561.91

Elaborado por: Oscar Analuisa

En base al presupuesto detallado en la tabla 4.20 se obtiene un costo del prototipo de \$ 561.91 dólares; un valor moderado y con equipos acorde a los requerimientos del hospital. Hay que tener en cuenta que existen sistemas similares que están diseñados solo para ser conectados a un servidor propio del equipo y ofrecen el mismo servicio, con la diferencia del costo que se eleva a 1500.00 dólares solo el PLC como por ejemplo el PLC Productivity 2000, otro ejemplo que se puede mencionar es la pantalla BOP-B07 HMI que tiene un valor de 913,86 dólares sin contar el costo de envío hacia el país. Estos dos ejemplos requieren un pago de licencias mensuales por el servicio de alojamiento en la nube y mensajería o notificaciones.

El prototipo tiene como ventaja importante, que es un sistema escalable, por esta razón se presenta la tabla 4.21 con el costo de los sensores restantes para la implementación total del sistema.

Tabla 4.21: Presupuesto del Sistema Completo

Presupuesto del Sistema Completo					
Ítem	Descripción	Unidad	Cantidad	Precio Unitario	Valor Total
1	Costo del prototipo	c/u	1	535.15	535.15
2	Sensores de nivel	c/u	3	600	1800
3	Balizas Led	c/u	3	25	75
4	Canaletas	c/u	25	2	50
5	Cable flexible #20	metro	10	0.5	5
6	Tornillos	c/u	20	0.05	1
7	Cable UTP	metro	20	0.75	15
8	Señalización Sticker	c/u	2	2	4
9	Cinta Adhesiva	c/u	2	0.5	1
10	Adaptador PVC 3/4	c/u	4	1	4
TOTAL					2490.15
Imprevistos (5 %)					124.51
VALOR TOTAL					2614.66

Elaborado por: Oscar Analuisa

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- El Hospital General Ambato cuenta con cuatro tanques de almacenamiento de diesel los mismo que alimentan a tres calderos y un generador eléctrico, los que por ningún motivo pueden dejar de trabajar ya que son elementos fundamentales para la mayoría de las áreas del hospital y requieren un monitoreo continuo.
- El prototipo elaborado es de diseño abierto, presenta la libertad a los desarrolladores de generar nuevos códigos para mejorar la funcionalidad del sistema electrónico de monitoreo de niveles de diesel, y aprovechar la escalabilidad del sistema.
- El Hospital General Ambato cuenta con un sistema que agiliza las rutinas diarias de monitoreo de diesel y el proceso de gestión para la adquisición de este combustible de manera oportuna.
- Los equipos utilizados para el diseño del sistema electrónico del monitoreo del almacenamiento de diesel están certificados para trabajar en un ambiente industrial brindando confianza al Hospital General Ambato.

5.2 Recomendaciones

- Se recomienda al Hospital General Ambato la instalación de un tanque extra para el almacenamiento de diesel para extender el tiempo de consumo y reducir los procesos de adquisición de este combustible.
- Se recomienda al área de mantenimiento la instalación de los transductores de nivel en los tanques de almacenamiento restantes, aprovechando así la escalabilidad del sistema.

- Se recomienda al personal encargado del registro de los niveles de diesel, descargar periódicamente los datos de monitoreo que se encuentran en la nube ya que podrían perderse en un año.
- Se recomienda al Hospital General Ambato automatizar el llenado de los tanques de almacenamiento de diesel incluyendo el monitoreo del nivel para un control completo de este combustible.

BIBLIOGRAFÍA

- [1] Organización Mundial de la Salud, «Introducción al programa de mantenimiento de equipos médicos,» 2012. [En línea]. Available: http://apps.who.int/iris/bitstream/10665/44830/1/9789243501536_spa.pdf.
- [2] V. S. J. ALFARO, «Academia.edu,» 2017. [En línea]. Available: http://www.academia.edu/31571810/Evolucion_del_mantenimiento_industrial. [Último acceso: 7 10 2017].
- [3] R. R. Luis V., «Ingeniería Clínica y Gestión de Tecnología en Salud: Avances y Propuestas,» Noviembre 2006. [En línea]. Available: http://its.uvm.edu/PUCP_CENGETS/LIBRO-CENGETS-NOV2006.pdf. [Último acceso: 23 08 2017].
- [4] Secretaría Nacional de Planificación y Desarrollo, Objetivo 3. Mejorar la calidad de vida de la población, «buenvivir.gob.ec,» 2013-2017. [En línea]. Available: <http://www.buenvivir.gob.ec/versiones-plannacional;>. [Último acceso: 18 6 2017].
- [5] Ministerial en el estatuto de orgánico de gestión organizacional por procesos de, «nstituciones.msp.gob.ec,» 2012. [En línea]. Available: http://instituciones.msp.gob.ec/somossalud/images/guia/documentos/estatuto_de_hosp_acuerdo.pdf. [Último acceso: 23 06 2017].
- [6] R. C. M. Cristina, «Sistema Informático para el control del mantenimiento técnico de equipos médicos del Hospital IESS - Ambato,» Tesis de la carrera de Ingeniería en Sistemas Computacionales e Informáticos, 07 2015. [En línea]. Available: http://repositorio.uta.edu.ec/bitstream/123456789/13082/1/Tesis_1046si.pdf.
- [7] A. S. R. Cadena, «Estudio de un Programa y Sistema Informático de Mantenimiento para optimizar recursos en el área energética en el Hospital del Seguro Social en la ciudad de Ambato,» Tesis de la Facultad Ingeniería Civil y Mecánica, 2011. [En línea]. Available: <http://repo.uta.edu.ec/bitstream/123456789/1426/1/Tesis%20I.%20M.%20121%20->.

- [8] M. C. Tonato, «Diseño e Implementación de un Sistema de Monitoreo del Nivel de Combustible para los Tanques de la Central de Generación en el Campo Cuyabeno Petroproducción,» Tesis de Ingeniería en Electrónica y Comunicaciones, 2009. [En línea]. Available: <http://repositorio.uta.edu.ec/bitstream/123456789/171/1/t490ec.pdf>.
- [9] T. M. Jorge, «Diseño de un plan de mantenimiento preventivo planificado para los equipos de la casa de máquinas del Hospital General Docente Riobamba,» Tesis ESPOCH, 2012. [En línea]. Available: <http://dspace.espoch.edu.ec/bitstream/123456789/2399/1/25T00177.pdf>.
- [10] C. A. V. Aguilar, «Diseño de un Sistema de Monitoreo de Nivel de los Tanques de Emergencia de Emcali Elecomunicaciones,» Tesis de la Facultad de Ingeniería Departamento de Automática y Electrónica, 9 Diciembre 2013. [En línea]. Available: <https://red.uao.edu.co/bitstream/10614/5683/1/T03722.pdf>.
- [11] D. D. Vargas Culqui, «Diseño e implementación de un sistema de control de combustión y monitoreo de gases generados en el caldero 3 del IESS Hospital de Ambato,» Tesis Electrónica y Control (IEC), 13 Febrero 2014. [En línea]. Available: <http://bibdigital.epn.edu.ec/handle/15000/7167>.
- [12] C. D. A. Oyazún, «Implementación de un Sistema de Control de Nivel para el Suministro de Petróleo de grupos Electrógenos,» Universidad de Magallanes, 2006. [En línea]. Available: http://www.umag.cl/biblioteca/tesis/acuna_oyarzun_2006.pdf.
- [13] Google Maps, [En línea]. Available: <https://www.google.es/maps/place/Hospital+del+IESS+Ambato/@-1.2336363,-78.6331341,1095m/data=!3m1!1e3!4m2!1m6!3m5!1s0x0:0x46a98c02d9a72b78!2sHospital+del+IESS+Ambato!8m2!3d-1.2335867!4d-78.6311848!3m4!1s0x0:0x46a98c02d9a72b78!8m2>.
- [14] C. A. V. Aguilar, «Diseño de un Sistema de Monitoreo de Nivel de los Tanques de Emergencia de Emcali Telecomunicaciones,» Red Educativo Digital, [En línea]. Available: <https://red.uao.edu.co/bitstream/10614/5683/1/T03722.pdf>.

- [15] W. I. Industrial, «Cursos Dictados,» Transductores Para Procesos Industriales, [En línea]. Available: <http://webdelprofesor.ula.ve/ingenieria/oscaror/CursosDictados/web%20instrumentacion%20industrial/1%20transductores%20para%20procesos%20industriales/libro%20pdf/CAP%201%20INTRODUCCION%202009.pdf>.
- [16] M. Vallejos, «slideplayer,» 2016. [En línea]. Available: <http://slideplayer.es/slide/3186996/11/images/4/Medidor+de+cinta.jpg>.
- [17] G. Kwaii, «Blogspot,» Agosto 2011. [En línea]. Available: <http://planespecifico.blogspot.com/2011/08/medicion-de-nivel.html>.
- [18] blogspot.com, Medidor Capacitivo, [En línea]. Available: http://4.bp.blogspot.com/_SxqX8ha_1Qo/SVekfxkj9bI/AAAAAAAAAM8/GCEXJbf rxnw/w1200-h630-p-k-no-nu/nivel+capacitivo.bmp.
- [19] L. C., «Sensor Capacitivo para la Medicion de Nivel (ParteII),» 18 05 2007. [En línea]. Available: http://4.bp.blogspot.com/_j7TgkZBMg_w/Rk48cOwD7xI/AAAAAAAAADk/8Bil0pDXMVU/s320/sen_cap.bmp.
- [20] T. Godoy, «Comité de Automatización AIE,» Endress + Hauser, [En línea]. Available: <http://www.aie.cl/files/file/comites/ca/articulos/enero-06.pdf>. [Último acceso: 1 09 2017].
- [21] C. B. Provoste, «Automatización Industrial, Robótica e Industria 4.0,» Medición de Nivel, [En línea]. Available: http://www.infoplcn.net/files/documentacion/instrumentacion_deteccion/infoPLC_net_MEDICION_DE%20_NIVEL.pdf. [Último acceso: 5 09 2017].
- [22] Pressure Vessels & Engineering, Sateña, S.A., «Sateña, S.A.,» Tanque de Amacenamiento de Diesel, [En línea]. Available: <http://www.satena.com.mx/tanque-diesel.html>.

- [23] A. O. y. H. Poveda, «Tesis ESPE,» Diseño e Implementación de un Sistema HMI para el Monitoreo y Control de las Variables Presión, Temperatura y Nivel en la Casa de Máquinas del Hospital del IESS de la Ciudad de Latacunga, [En línea]. Available: <http://repositorio.espe.edu.ec/bitstream/21000/10593/1/T-ESPEL-ENI-0370-P.pdf>.
- [24] «Technologies larssystem,» WebCorp.gt, 2017. [En línea]. Available: <http://larssystem.com/tienda/automatizacion/hmi/hmi-elc-md204/>.
- [25] O. A. J. García, «blogspot,» Marzo 2013. [En línea]. Available: <http://loquemepidiomiprofesora.blogspot.com/2013/03/modelo-de-shannon-y-weaver.html>.
- [26] T.Wayne, Sistemas de Comunicaciones Electrónicas, Mexico: Pearson Educación, 2003.
- [27] G. Q. L. Adrián, «Repositorio Institucional de la Escuela Superior Politécnica de Chimborazo,» Diseño e Implementación de un Banco, 2016. [En línea]. Available: <http://dspace.esPOCH.edu.ec/bitstream/123456789/5178/1/25T00271.pdf>.
- [28] logicbus, «logicbus.com,» 2018. [En línea]. Available: <http://www.logicbus.com.mx/plc.php>.
- [29] F. y. N, «Facultad de Ciencias Exactas,» Elementos y Equipos Electricos, [En línea]. Available: http://www.efn.uncor.edu/departamentos/electro/cat/eye_archivos/apuntes/a_practico/CAP%209%20Pco.pdf.
- [30] Fundación de la Innovación Bankinter, «El Internet de las Cosas,» 2011. [En línea]. Available: http://www.belt.es/expertos/imagenes/XV_FTF_El_internet_de_las_cosas.pdf.
- [31] L. F. D. Agua, «fundacionaquae,» 2018. [En línea]. Available: <https://www.fundacionaquae.org/wiki-aquae/innovacion/internet-las-cosas-iot/>.

- [32] B. D. S. Siripathi, «Envato Elements,» Plataformas de desarrollo móvil, 2017. [En línea]. Available: <https://code.tutsplus.com/es/articles/mobile-development-platforms-cms-28944>. [Último acceso: 17 08 2017].
- [33] J. E. Crespo, «Aprendiendo Arduino,» Lenguaje de programación C++, [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>. [Último acceso: 30 08 2017].
- [34] Caterpillar, «CAT,» 2018. [En línea]. Available: https://www.cat.com/es_MX/products/new/power-systems/electric-power-generation/diesel-generator-sets/18321546.html.
- [35] s. O. C. S., «Web del Prof. Oscar Cárdenas,» [En línea]. Available: <http://webdelprofesor.ula.ve/ingenieria/oscaror/CursosDictados/>. [Último acceso: 12 8 2017].
- [36] E. Freaks, «micropik,» 2018. [En línea]. Available: <http://www.micropik.com/PDF/HCSR04.pdf>.
- [37] maxbotix, «maxbotix,» 2018. [En línea]. Available: https://www.maxbotix.com/documents/XL-TrashSonar-WR_Datasheet.pdf.
- [38] Telematic, «Telematic Controls,» 2018. [En línea]. Available: <https://www.telematic.com/Product/GI31/IRU-5000-Series-Ultrasonic-Level-Sensor>.
- [39] G. Crippen, «techlandia,» Diferencia entre el PLC y el microprocesador, 2018. [En línea]. Available: https://techlandia.com/diferencia-plc-microprocesador-hechos_547880/.
- [40] C. GmbH, «Controllino,» 2018. [En línea]. Available: <https://controllino.biz/>.
- [41] E. C. E. Mauricio, «Repositorio de Universidad Técnica de Cotopaxi,» 2013. [En línea]. Available: www.bibliotecasdelecuador.com/.../oai:oai:repositorio.utc.edu.ec:27000:27000-1875.

- [42] S. V. A. Sebastián, «Repositorio Universidad Técnica de Ambato,» Febrero 2017. [En línea]. Available: positorio.uta.edu.ec/handle/123456789/24812.
- [43] Temboo, «Temboo, Inc,» 2018. [En línea]. Available: <https://temboo.com/>.
- [44] ThingSpeak, TheMathWorks, [En línea]. Available: <https://thingspeak.com/>.
- [45] Nimbits, Nimbits Inc, 2018. [En línea]. Available: <https://www.nimbits.com/>.
- [46] T. B. Labs, «[https://thinger.io,](https://thinger.io/)» [En línea]. Available: <https://thinger.io/pricing/>. [Último acceso: 30 09 2017].
- [47] M. D. G. Inc., «Meteor,» 2018. [En línea]. Available: <https://www.meteor.com/>.
- [48] M. Stack, «Mean io,» 2018. [En línea]. Available: <http://mean.io/>.
- [49] M. A. Alvarez, «Firebase 3,» 31 05 2016. [En línea]. Available: <https://desarrolloweb.com/articulos/nuevo-firebase.html>.
- [50] J. R. C. Alvarado, «Repositorio Universidad Politecnica Salesiana,» 2015. [En línea]. Available: <https://dspace.ups.edu.ec/bitstream/123456789/7951/1/UPS-CT004811.pdf>.
- [51] F. H. R. Leyva, « Universidad Tecnológica de la Mixteca,» 10 2012. [En línea]. Available: <http://www.utm.mx/~hugo/metrologia/Metro6>.
- [52] I. c. h. TiePie, «TiePie engineering,» 2017. [En línea]. Available: https://www.tiepie.com/en/classroom/MultiChannel_measurement_software/IOs/Gain_offset.
- [53] Dani, «Esploradores,» Agosto 2016. [En línea]. Available: <http://www.esploradores.com/thinger-io-configuracion-almacenamiento-y-monitorizacion-de-datos-primera-parte/>.
- [54] Enigmind, «Electricas Bogota,» 2018. [En línea]. Available: http://www.electricasbogota.com/pdf/68000_68002.pdf.

- [55] N. Instruments, «Información Detallada sobre el Protocolo Modbus,» 16 10 2016. [En línea]. Available: <http://www.ni.com/white-paper/52134/es/>.
- [56] G. Developers, «Firebase,» Developers, 2017. [En línea]. Available: <https://firebase.google.com/pricing/?hl=es-419>.
- [57] E. Components, «Transfer Multisort Elektronik S.L.U.,» [En línea]. Available: <https://www.tme.eu/es/Document/8bf85f4d1c3d5c36f1632f12035133d2/mikroe602.pdf>. [Último acceso: 10 11 2017].
- [58] «Electricas Bogota Ltda» 2010. [En línea]. Available: http://www.electricasbogota.com/pdf/68000_68002.pdf.
- [59] B. SmartWorks, «B+B SmartWorks,» 2016. [En línea]. Available: <http://www.meanwell.com/Upload/PDF/DR-45/DR-45-SPEC.PDF>.

ANEXOS

ANEXO A

CÓDIGO DE PROGRAMACIÓN SISTEMA DE MONITOREO

```
#include <SPI.h>
#include <Ethernet.h>
#include <Controllino.h>
#include <ThingierEthernet.h>
#include <math.h>
#include "ModbusRtu.h"
#include <TimerOne.h> //libreria para manejo de interrupciones
#include <EEPROM.h>

#define VERDE CONTROLLINO_D0
#define AMARILLO CONTROLLINO_D1
#define ROJO CONTROLLINO_D2

#define VERDE CONTROLLINO_D3
#define AMARILLO CONTROLLINO_D4
#define ROJO CONTROLLINO_D5

#define VERDE CONTROLLINO_D6
#define AMARILLO CONTROLLINO_D7
#define ROJO CONTROLLINO_D8

#define VERDE CONTROLLINO_D9
#define AMARILLO CONTROLLINO_D10
#define ROJO CONTROLLINO_D11

//definicion de estructura para tanque
struct Tanque {
    int radio;
    int longitud;
    int FactorEscala; // y=mx+b donde m es la pendiente y b el cruce por cero
    int OffsetEscala; // factorescala=m y offsetescala=b
    int alarmaBajo;
    int alarmaMedio;
};

//Definir Variables
#define SlaveModbusAdd 1
#define RS485Serial 2
Modbus ControllinoModbusSlave(SlaveModbusAdd, RS485Serial, 0);
uint16_t ModbusSlaveRegisters[17];
unsigned long timer1, timer2, timer3, timerAlarm;
float HTanque1, VolActualTanque1;
int sensorValue1, NivelAlBajo1, NivelAlmedio1;
```

```

float HTanque2, VolActualTanque2;
int sensorValue2, NivelAlBajo2, NivelAlmedio2;
bool conexionNube;

// Credenciales para la nube
#define USERNAME "oscar_fisei"
#define DEVICE_ID "Controllino"
#define DEVICE_CREDENTIAL "g34RbHXRM@L#"

//Crea un instancia de la librería Thingier.IO usando las credenciales usadas en el registro y del dispositivo
creado en la plataforma IoT
ThingierEthernet thing(USERNAME, DEVICE_ID, DEVICE_CREDENTIAL);

Tanque tanque1, tanque2, tanque3, tanque4; // estructura que contiene la informacion de un tanque
pjson data; // estructura json que contiene la informacion a ser enviada a la nube

void setup() {

// Envío de Datos de cada sensor
thing["VolumenTanque1"] >> outputValue(VolActualTanque1);
thing["Sensor1"] >> outputValue(sensorValue1);

thing["VolumenTanque2"] >> outputValue(VolActualTanque2);
thing["Sensor2"] >> outputValue(sensorValue2);

thing["VolumenTanque3"] >> outputValue(VolActualTanque2);
thing["Sensor3"] >> outputValue(sensorValue2);

thing["VolumenTanque4"] >> outputValue(VolActualTanque2);
thing["Sensor4"] >> outputValue(sensorValue2);

ControllinoModbusSlave.begin(19200); // Start the communication over the ModbusRTU protocol. Baud
rate is set at 19200.
Serial.begin(9600);

//Obtener la información del tanque 1 de EEPROM
EEPROM.get(0, tanque1);
CopiarEepromModbusT1();
EEPROM.get(0 + sizeof(Tanque), tanque2);
CopiarEepromModbusT2();
EEPROM.get(0 + sizeof(Tanque) + Tanque3);
CopiarEepromModbusT3();
EEPROM.get(0 + sizeof(Tanque) + sizeof(Tanque3) + Tanque4);
CopiarEepromModbusT4();

//inicializa los temporizadores utilizados en el lazo principal de control
timer1=millis();
timer2=millis();
timer3=millis();
timerAlarm=0;

pinMode(VERDE, OUTPUT); // luz verde
pinMode(AMARILLO, OUTPUT); // luz amarilla
pinMode(ROJO, OUTPUT); // luz roja

Timer1.initialize(50000);

```

```

Timer1.attachInterrupt(modbusFunction);

  Serial.println("Inicializado");
}

void loop() {

//modbusFunction();

thing.handle();

//Notificaciones tanque 1
if(millis()>=timer1+timerAlarm+3000) //timer para verificar alarmas cada 3 segundos
{
  if(VolActualTanque1<tanque1.alarmaBajo || VolActualTanque1<tanque1.alarmaMedio)
  {
    // sea cuidadoso en enviar datos a un ritmo adecuado!
    // ejecuta el FirebaseNotification para notificar sobre la alarma a los dispositivos móviles
    // se adiciona los datos requeridos que serán enviados
    timerAlarm=57000; // si la alarma persiste se incrementa el delay a 60 saturar l nube
    data["VolumenTanque1"] = VolActualTanque1;
    data["Sensor1"] = sensorValue1;

    if( VolActualTanque1<tanque1.alarmaBajo)
    {
      data["TanqueNo"] = "Nivel bajo en Tanque 1";

    }
    else if (VolActualTanque1<tanque1.alarmaMedio )
    {
      data["TanqueNo"] = "Nivel medio en Tanque 1";

    }

    thing.call_endpoint("NotificacionTanque", data);
  }
  else
  {
    timerAlarm=0;
    bitClear(ModbusSlaveRegisters[8],0);
    bitClear(ModbusSlaveRegisters[8],1);
  }
}

//Notificaciones tanque 2

if(VolActualTanque2<tanque2.alarmaBajo || VolActualTanque2<tanque2.alarmaMedio)

  timerAlarm=57000; // si la alarma persiste se incrementa el delay a 60
  data["VolumenTanque2"] = VolActualTanque2;
  data["Sensor2"] = sensorValue2;

  if( VolActualTanque2<tanque2.alarmaBajo)
  {
    data["TanqueNo"] = "Nivel bajo en Tanque 2";

  }
  else if (VolActualTanque2<tanque2.alarmaMedio )
  {

```

```

        data["TanqueNo"] = "Nivel medio en Tanque 2";

    }

    thing.call_endpoint("NotificacionTanque", data);
}
else
{
    timerAlarm=0;
    bitClear(ModbusSlaveRegisters[8],2);
    bitClear(ModbusSlaveRegisters[8],3);
}
}
/// //Notificaciones tanque 3

if(VolActualTanque3<tanque3.alarmaBajo || VolActualTanque3<tanque3.alarmaMedio)
{
    // sea cuidadoso en enviar datos a un ritmo adecuado!
    // ejecuta el FirebaseNotification para notificar sobre la alarma a los dispositivos móviles
    // se adiciona los datos requeridos que serán enviados
    timerAlarm=57000; // si la alarma persiste se incrementa el delay a 30 minutos
    data["VolumenTanque3"] = VolActualTanque1;
    data["Sensor3"] = sensorValue1;

    if( VolActualTanque3<tanque3.alarmaBajo)
    {
        data["TanqueNo"] = "Nivel bajo en Tanque 3";

    }
    else if (VolActualTanque3<tanque3.alarmaMedio )
    {
        data["TanqueNo"] = "Nivel medio en Tanque 3";

    }

    thing.call_endpoint("NotificacionTanque", data);
}
else
{
    timerAlarm=0;
    bitClear(ModbusSlaveRegisters[8],4);
    bitClear(ModbusSlaveRegisters[8],5);
}
}
///// //Notificaciones tanque 4

if(VolActualTanque4<tanque4.alarmaBajo || VolActualTanque4<tanque4.alarmaMedio)
{
    // sea cuidadoso en enviar datos a un ritmo adecuado!
    // ejecuta el FirebaseNotification para notificar sobre la alarma a los dispositivos móviles
    // se adiciona los datos requeridos que serán enviados
    timerAlarm=57000; // si la alarma persiste se incrementa el delay a 60
    data["VolumenTanque4"] = VolActualTanque4;
    data["Sensor4"] = sensorValue4;

    if( VolActualTanque4<tanque4.alarmaBajo)
    {
        data["TanqueNo"] = "Nivel bajo en Tanque 4";

    }
    else if (VolActualTanque4<tanque4.alarmaMedio )

```

```

    {
        data["TanqueNo"] = "Nivel medio en Tanque 4";
    }

    thing.call_endpoint("NotificacionTanque", data);
}
else
{
    timerAlarm=0;
    bitClear(ModbusSlaveRegisters[8],6);
    bitClear(ModbusSlaveRegisters[8],7);
}
//////////Adquisición de información y envió atreves de modbus
timer1=millis();

}

}

void modbusFunction(void)
{

    ControllinoModbusSlave.poll(ModbusSlaveRegisters, 17);

    if(tanque1.radio!=ModbusSlaveRegisters[0] || tanque1.longitud!=ModbusSlaveRegisters[1] ||
tanque1.FactorEscala!=ModbusSlaveRegisters[2]
    || tanque1.OffsetEscala!=ModbusSlaveRegisters[3] || tanque1.alarmaBajo!=ModbusSlaveRegisters[4] ||
tanque1.alarmaMedio!= ModbusSlaveRegisters[5])
    {
        tanque1.radio=ModbusSlaveRegisters[0];
        tanque1.longitud=ModbusSlaveRegisters[1];
        tanque1.FactorEscala=ModbusSlaveRegisters[2];
        tanque1.OffsetEscala=ModbusSlaveRegisters[3];
        tanque1.alarmaBajo=ModbusSlaveRegisters[4];
        tanque1.alarmaMedio=ModbusSlaveRegisters[5];
        EEPROM.put(0,tanque1);
        EEPROM.get(0,tanque1);
        CopiarEepromModbusT1();
        // Serial.println("EEPROM actualizada");
    }
    ModbusSlaveRegisters[6]=(int)VolActualTanque1;
    ModbusSlaveRegisters[7]=(int)sensorValue1;

    if(tanque2.radio!=ModbusSlaveRegisters[9] || tanque2.longitud!=ModbusSlaveRegisters[10] ||
tanque2.FactorEscala!=ModbusSlaveRegisters[11]
    || tanque2.OffsetEscala!=ModbusSlaveRegisters[12] || tanque2.alarmaBajo!=ModbusSlaveRegisters[13] ||
tanque2.alarmaMedio!= ModbusSlaveRegisters[14])
    {
        tanque2.radio=ModbusSlaveRegisters[9];
        tanque2.longitud=ModbusSlaveRegisters[10];
        tanque2.FactorEscala=ModbusSlaveRegisters[11];
        tanque2.OffsetEscala=ModbusSlaveRegisters[12];
        tanque2.alarmaBajo=ModbusSlaveRegisters[13];
        tanque2.alarmaMedio=ModbusSlaveRegisters[14];
        EEPROM.put(0+sizeof(Tanque),tanque2);
    }
}

```

```

EEPROM.get(0+sizeof(Tanque),tanque2);
CopiarEepromModbusT2();
// Serial.println("EEPROM actualizada");
}

ModbusSlaveRegisters[15]=(int)VolActualTanque2;
ModbusSlaveRegisters[16]=(int)sensorValue2;
////////////////////////////////////
if(tanque3.radio!=ModbusSlaveRegisters[17] || tanque3.longitud!=ModbusSlaveRegisters[18] ||
tanque3.FactorEscala!=ModbusSlaveRegisters[19]
|| tanque3.OffsetEscala!=ModbusSlaveRegisters[20] || tanque3.alarmaBajo!=ModbusSlaveRegisters[21] ||
tanque3.alarmaMedio!= ModbusSlaveRegisters[22])
{
tanque3.radio=ModbusSlaveRegisters[17];
tanque3.longitud=ModbusSlaveRegisters[18];
tanque3.FactorEscala=ModbusSlaveRegisters[19];
tanque3.OffsetEscala=ModbusSlaveRegisters[20];
tanque3.alarmaBajo=ModbusSlaveRegisters[21];
tanque3.alarmaMedio=ModbusSlaveRegisters[22];
EEPROM.put(0+sizeof(Tanque),tanque3);
EEPROM.get(0+sizeof(Tanque),tanque3);
CopiarEepromModbusT2();
// Serial.println("EEPROM actualizada");
}

ModbusSlaveRegisters[23]=(int)VolActualTanque2;
ModbusSlaveRegisters[24]=(int)sensorValue2;
////////////////////////////////////
if(tanque4.radio!=ModbusSlaveRegisters[25] || tanque4.longitud!=ModbusSlaveRegisters[26] ||
tanque4.FactorEscala!=ModbusSlaveRegisters[27]
|| tanque4.OffsetEscala!=ModbusSlaveRegisters[28] || tanque4.alarmaBajo!=ModbusSlaveRegisters[29] ||
tanque4.alarmaMedio!= ModbusSlaveRegisters[30])
{
tanque4.radio=ModbusSlaveRegisters[25];
tanque4.longitud=ModbusSlaveRegisters[26];
tanque4.FactorEscala=ModbusSlaveRegisters[27];
tanque4.OffsetEscala=ModbusSlaveRegisters[28];
tanque4.alarmaBajo=ModbusSlaveRegisters[29];
tanque4.alarmaMedio=ModbusSlaveRegisters[30];
EEPROM.put(0+sizeof(Tanque),tanque2);
EEPROM.get(0+sizeof(Tanque),tanque2);
CopiarEepromModbusT2();
// Serial.println("EEPROM actualizada");
}

ModbusSlaveRegisters[31]=(int)VolActualTanque2;
ModbusSlaveRegisters[32]=(int)sensorValue2;
////////// Calculo de niveles de Diesel

if(millis()-250 >=timer2)// calculo del volumen se repite cada 250 ms
{
sensorValue1=analogRead(A6);
HTanque1=2.0*tanque1.radio/100.0-
(float)sensorValue1*tanque1.FactorEscala/100.0+tanque1.OffsetEscala/100.0;

```

```

    VolActualTanque1=calculoVolumen(tanque1.radio/100.0,tanque1.longitud/100.0,HTanque1)*264.172; //
    calculo ajustado a galones

    sensorValue2=analogRead(A0);
    HTanque2=2.0*tanque2.radio/100.0-
(float)sensorValue2*tanque2.FactorEscala/100.0+tanque2.OffsetEscala/100.0;

    VolActualTanque2=(3.1416* tanque2.radio/100.0* tanque2.radio/100.0*(float)sensorValue2)*264.172; //
    calculo ajustado a galones

    sensorValue3=analogRead(A1);
    HTanque3=2.0*tanque2.radio/100.0-
(float)sensorValue3*tanque3.FactorEscala/100.0+tanque3.OffsetEscala/100.0;

    VolActualTanque3=calculoVolumen(tanque3.radio/100.0,tanque3.longitud/100.0,HTanque3)*264.172; //
    calculo ajustado a galones

    sensorValue4=analogRead(A2);
    HTanque4=2.0*tanque2.radio/100.0-
(float)sensorValue4*tanque4.FactorEscala/100.0+tanque4.OffsetEscala/100.0;

    VolActualTanque4=calculoVolumen(tanque4.radio/100.0,tanque4.longitud/100.0,HTanque4)*264.172; //
    calculo ajustado a galones

    timer2=millis();

    // Serial.println("vol cal");
}

//Notificaciones locales en la pantalla
if( VolActualTanque1<tanque1.alarmaBajo)
{

    // ModbusSlaveRegisters[8]=1; // 0000000000000001
    bitSet(ModbusSlaveRegisters[8],0);
    digitalWrite( VERDE, LOW);
    digitalWrite( AMARILLO, LOW);
    digitalWrite( ROJO, HIGH);

}
else if (VolActualTanque1<tanque1.alarmaMedio )
{

    //ModbusSlaveRegisters[8]=2; // 0000000000000010
    bitSet(ModbusSlaveRegisters[8],1);
    digitalWrite( VERDE, LOW);
    digitalWrite( AMARILLO, HIGH);
    digitalWrite( ROJO, LOW);

}
else
{

    bitClear(ModbusSlaveRegisters[8],0);
    bitClear(ModbusSlaveRegisters[8],1);
    digitalWrite( VERDE, HIGH);
    digitalWrite( AMARILLO, LOW);
}

```



```

        digitalWrite( ROJO, LOW);

    }

if( VolActualTanque2<tanque2.alarmaBajo)
{

    bitSet(ModbusSlaveRegisters[8],2);
}
else if (VolActualTanque2<tanque2.alarmaMedio )
{

    bitSet(ModbusSlaveRegisters[8],3);
}
else
{

    bitClear(ModbusSlaveRegisters[8],2);
    bitClear(ModbusSlaveRegisters[8],3);

}
////////////////////////////////////
if( VolActualTanque3<tanque3.alarmaBajo)
{

    bitSet(ModbusSlaveRegisters[8],4);
}
else if (VolActualTanque3<tanque3.alarmaMedio )
{

    bitSet(ModbusSlaveRegisters[8],5);
}
else
{

    bitClear(ModbusSlaveRegisters[8],6);
    bitClear(ModbusSlaveRegisters[8],7);

}
////////////////////////////////////
if( VolActualTanque4<tanque4.alarmaBajo)
{

    bitSet(ModbusSlaveRegisters[8],8);
}
else if (VolActualTanque4<tanque4.alarmaMedio )
{

    bitSet(ModbusSlaveRegisters[8],9);
}
else
{

    bitClear(ModbusSlaveRegisters[8],10);
    bitClear(ModbusSlaveRegisters[8],11);
}

```

```

    }
    //////////////////////////////////////
}

//función para calcular el volumen de un tanque cilíndrico horizontal
float calculoVolumen(float R, float L, float h)
{
    float R2, h2;
    R2=R*R;
    h2=h*h;
    //M_PI_2 = M_PI/2
    // referencia: http://www.nongnu.org/avr-libc/user-manual/group__avr__math.html
    return L*(M_PI_2*R2+(h-R)*sqrt(2*R*h-h2) + R2*asin((-R+h)/R));
//Actualización de datos de cada tanque
}
void CopiarEepromModbusT1()
{
    ModbusSlaveRegisters[0]=tanque1.radio;
    ModbusSlaveRegisters[1]=tanque1.longitud;
    ModbusSlaveRegisters[2]=tanque1.FactorEscala;
    ModbusSlaveRegisters[3]=tanque1.OffsetEscala;
    ModbusSlaveRegisters[4]=tanque1.alarmaBajo;
    ModbusSlaveRegisters[5]=tanque1.alarmaMedio;

}
void CopiarEepromModbusT2()
{
    ModbusSlaveRegisters[9]=tanque2.radio;
    ModbusSlaveRegisters[10]=tanque2.longitud;
    ModbusSlaveRegisters[11]=tanque2.FactorEscala;
    ModbusSlaveRegisters[12]=tanque2.OffsetEscala;
    ModbusSlaveRegisters[13]=tanque2.alarmaBajo;
    ModbusSlaveRegisters[14]=tanque2.alarmaMedio;

}
////////////////////////////////////
void CopiarEepromModbusT3()
{
    ModbusSlaveRegisters[17]=tanque3.radio;
    ModbusSlaveRegisters[18]=tanque3.longitud;
    ModbusSlaveRegisters[19]=tanque3.FactorEscala;
    ModbusSlaveRegisters[20]=tanque3.OffsetEscala;
    ModbusSlaveRegisters[21]=tanque3.alarmaBajo;
    ModbusSlaveRegisters[22]=tanque3.alarmaMedio;
}
////////////////////////////////////
void CopiarEepromModbusT4()
{
    ModbusSlaveRegisters[25]=tanque4.radio;
    ModbusSlaveRegisters[26]=tanque4.longitud;
    ModbusSlaveRegisters[27]=tanque4.FactorEscala;
    ModbusSlaveRegisters[28]=tanque4.OffsetEscala;
    ModbusSlaveRegisters[29]=tanque4.alarmaBajo;
    ModbusSlaveRegisters[30]=tanque4.alarmaMedio;
}

```

ANEXO B

DESCRIPCIÓN LA CLASE JAVA MAINACTIVITY

Clase Java MainActivity

```
public class MainActivity extends AppCompatActivity {
    private TextView infoTextView;
    private TextView bodyTextView;
    private ImageView imageViewAlarma;
    private Button buttonNavegar;
    private static final String TAG = "MainActivity";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        infoTextView = (TextView) findViewById(R.id.infoTextView);
        bodyTextView = (TextView) findViewById(R.id.textViewBody);
        buttonNavegar=(Button) findViewById(R.id.buttonNavegar);

        if (getIntent().getExtras()!=null){
            imageViewAlarma=(ImageView)
findViewById(R.id.imageViewAlarma);
            imageViewAlarma.setVisibility(VISIBLE);
            buttonNavegar.setVisibility(VISIBLE);
            infoTextView.setText("Alarma: ");
            String s1, s2;
            s1="";
            s2="";

            for (String key:getIntent().getExtras().keySet()){
                if (key.equals("volumen1")){
                    String value =
getIntent().getExtras().getString(key);
                    Float numero= Float.parseFloat(value);
                    String s ;
                    s=String.format("%.2f",numero );
                    s1= "\n" +"Volumen tanque 1: " +s + " Gal.";
                }
                if (key.equals("sensor1")){
                    String value = getIntent().getExtras().getString(key);
                    s1= s1+ "\n" +"Unidades crudas sensor 1: " +value;
                }

                if (key.equals("volumen2")){
                    String value = getIntent().getExtras().getString(key);
                    Float numero= Float.parseFloat(value);
                    String s ;
                    s=String.format("%.2f",numero );
                    s2= "\n" +"Volumen tanque 2: " +s + " Gal.";
                }
                if (key.equals("sensor2")){
                    String value = getIntent().getExtras().getString(key);
                    s2= s2+ "\n" +"Unidades crudas sensor 2: " +value;
                }

                if (key.equals("tanqueNo"))
                {
                    String value = getIntent().getExtras().getString(key);
                    infoTextView.append(value);
                }
            }
        }
    }
}
```

```

    }
    bodyTextView.append(s1 + s2);
}

final Button button = (Button)findViewById(R.id.buttonSuscribe);
button.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
FirebaseMessaging.getInstance().subscribeToTopic("alarmas");
        // Log and toast
        String msg = getString(R.string.msg_subscribed);
        //Log.d(TAG, msg);
        Toast.makeText(MainActivity.this, msg,
Toast.LENGTH_SHORT).show();
    }
});

final Button button2 = (Button)findViewById(R.id.buttonCancela);
button2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
FirebaseMessaging.getInstance().unsubscribeFromTopic("alarmas");
        // Log and toast
        String msg = getString(R.string.msg_cancela);
        //Log.d(TAG, msg);
        Toast.makeText(MainActivity.this, msg,
Toast.LENGTH_SHORT).show();
    }
});

final Button button3 = (Button)findViewById(R.id.buttonNavegar);
button3.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent browserIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse("https://console.thinger.io/#/login"));
        startActivity(browserIntent);
    }
});

final Button button4 = (Button)findViewById(R.id.buttonSalir);
button4.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {

android.os.Process.killProcess(android.os.Process.myPid());
        System.exit(1);
    }
});
}
}

```

Descripción: Esta clase se encarga:

- Llamar al Activity o Ventana principal si es diferente de nulo para crear alerta
- Establecer la comunicación con firebase.
- Crear intents, con datos de firebase
- Obtener la suscripción.
- Botón Navegar, se redirecciona a la consola de thinger.io

ANEXO C

DESCRIPCIÓN DE LA CLASE JAVA MIFIREBASEINSTANCESERVICE

Clase Java miFirebaseInstanceService

```
package com.oscaranaluisa.notificacionalarma;
import android.util.Log;
import com.google.firebase.iid.FirebaseInstanceId;
import com.google.firebase.iid.FirebaseInstanceIdService;

/**
 * 7/19/2017.
 */

public class miFirebaseInstanceIdService extends
FirebaseInstanceIdService {
    public static final String TAG="NOTICIAS";

    @Override
    public void onTokenRefresh() {
        super.onTokenRefresh();

        String token =
FirebaseInstanceId.getInstance().getToken();
        // Log.d(TAG, "Token: " + token );

    }
}
```

Descripción: esta clase se encarga:

- Para que trabaje esta clase se realiza el llamado a la instancia creada por firebase del archivo json.
- Enlaza la aplicación con la plataforma firebase.
- La etiqueta NOTICIAS es el Id string para la conexión

ANEXO D

DESCRIPCIÓN DE LA CLASE JAVA MIFIREBASEMESSAGINGSERVICE

Clase Java miFirebaseMessagingService

```
package com.oscaranaluisa.notificacionalarma;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.app.NotificationCompat;
import android.util.Log;
import android.content.Intent;
import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

/**
 * 7/19/2017.
 */

public class miFirebaseMessagingService extends
    FirebaseMessagingService {
    public static final String TAG="NOTICIAS";
    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        super.onMessageReceived(remoteMessage);
        String from=remoteMessage.getFrom();
        // Log.d(TAG,"Mensaje recibido de:" + from);
        if(remoteMessage.getNotification() != null){
            //Log.d(TAG,"Notificación: " +
            remoteMessage.getNotification().getBody());

            mostrarNotificacion(remoteMessage.getNotification().getTitle(),rem
            oteMessage.getNotification().getBody());
        }

        if(remoteMessage.getData().size()>0)
        {

            // Log.d(TAG,"Data: "+ remoteMessage.getData());
        }
    }

    private void mostrarNotificacion(String title, String body){
        Intent intent =new Intent(this,MainActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

        // El pendingIntent define que se hace cuando se hace tap
        sobre la notificacion
        // Este pendingIntent a su vez requiere un Intent que se
        ha declarado previamente
        PendingIntent pendingIntent =
        PendingIntent.getActivity(this,0,intent,PendingIntent.FLAG_ONE_SHO
        T);
    }
}
```

```

        //Obtenemos la Uri del ringtone pro defecto usado para
notificaiones
        Uri soundUri =
RingtoneManager.getDefaultUri (RingtoneManager.TYPE_NOTIFICATION) ;

        // Creamos una notificacion con las caracter[isticas que
nosotros requerimos
        NotificationCompat.Builder notificationBuilder = new
NotificationCompat.Builder (this)

        .setSmallIcon (R.drawable.ic_notifications_active_black_24dp)
                .setContentTitle (title)
                .setContentText (body)
                .setAutoCancel (true)
                .setSound (soundUri)
                .setContentIntent (pendingIntent) ;

        NotificationManager notificationManager
= (NotificationManager)
getSystemService (Context.NOTIFICATION_SERVICE) ;
        notificationManager.notify (0,notificationBuilder.build()) ;
    }
}

```

Descripción: esta clase se encarga:

- Conexión a la plataforma Firebase usando la etiqueta NOTICIAS
- Recepción de la notificación si en la base de datos tiene información de alerta
- Alerta de notificación usando FLAG_ONE_SHOT
- Creación de la Notificación

ANEXO E

ARCHIVO XML ACTIVITY_MENU.XML (MENÚ PRINCIPAL)

Activiy main

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"

tools:context="com.oscaranaluisa.notificacionalarma.MainActivity">

    <TextView
        android:id="@+id/infoTextView"
        android:layout_width="283dp"
        android:layout_height="wrap_content"
        android:layout_gravity="start"
        android:layout_marginLeft="8dp"
        android:layout_marginRight="8dp"
        android:layout_marginTop="76dp"
        android:textAlignment="viewStart"

        android:textAppearance="@style/TextAppearance.AppCompat.Headline"
        android:textColor="@color/colorAccent"
        android:textSize="22sp"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintLeft_toRightOf="@+id/imageViewAlarma"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/buttonSuscribe"
        android:layout_width="99dp"
        android:layout_height="53dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="44dp"
        android:text="@string/stringButtonSuscribe"
        android:textColor="#009688"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonNavegar"
    />

    <Button
        android:id="@+id/buttonCancela"
        android:layout_width="99dp"
        android:layout_height="53dp"
        android:layout_marginBottom="68dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:text="@string/stringButtonCancela"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/buttonSuscribe"
```



```

        app:layout_constraintVertical_bias="1.0" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="350dp"
    android:layout_height="35dp"
    android:layout_marginRight="24dp"
    android:layout_marginTop="32dp"
    android:text="@string/stringMessage"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="@+id/buttonSuscribe"
    app:layout_constraintTop_toBottomOf="@+id/buttonCancela"
/>

<Button
    android:id="@+id/buttonNavegar"
    android:layout_width="172dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="228dp"
    android:layout_marginStart="28dp"
    android:text="@string/stringButtonDashboard"
    android:visibility="visible"
    android:background="#00E5FF"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<Button
    android:id="@+id/buttonSalir"
    android:layout_width="154dp"
    android:layout_height="40dp"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginTop="8dp"
    android:text="@string/stringButtonSalir"
    android:textColor="#F50057"
    app:layout_constraintBottom_toTopOf="@+id/textView2"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/buttonNavegar"
    app:layout_constraintVertical_bias="1.0" />

<TextView
    android:id="@+id/textViewBody"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginTop="8dp"

android:textAppearance="@style/TextAppearance.AppCompat.Body2"
android:textSize="18sp"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toBottomOf="@+id/infoTextView" />

<ImageView
    android:id="@+id/imageViewAlarma"

```

```

        android:layout_width="54dp"
        android:layout_height="47dp"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="8dp"
        android:background="@android:color/holo_red_light"
        android:contentDescription="@string/app_name"
        android:visibility="invisible"
        app:layout_constraintBottom_toTopOf="@+id/textViewBody"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.788"
        app:srcCompat="@drawable/ic_stat_name" />

<ImageView
    android:id="@+id/imageView7"
    android:layout_width="81dp"
    android:layout_height="44dp"
    app:layout_constraintEnd_toStartOf="@+id/textViewBody"
    app:layout_constraintHorizontal_bias="0.148"
    app:layout_constraintStart_toEndOf="@+id/textViewBody"
    app:srcCompat="@drawable/logo"
    tools:layout_editor_absoluteY="3dp" />

<ImageView
    android:id="@+id/imageView9"
    android:layout_width="85dp"
    android:layout_height="44dp"
    app:srcCompat="@drawable/maquinas"
    tools:layout_editor_absoluteX="0dp"
    tools:layout_editor_absoluteY="3dp" />

<TextView
    android:id="@+id/textView"
    android:layout_width="175dp"
    android:layout_height="19dp"
    android:layout_gravity="center_horizontal"
    android:text="MONITOREO DE NIVELES "

    android:textColor="#AA00FF"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.559"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.032" />

<android.support.constraint.Guideline
    android:id="@+id/guideline4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    app:layout_constraintGuide_begin="20dp" />

<android.support.constraint.Guideline
    android:id="@+id/guideline5"

```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:orientation="vertical"  
app:layout_constraintGuide_begin="20dp" />
```

```
<TextView
```

```
android:id="@+id/textView3"  
android:layout_width="wrap_content"  
android:layout_height="28dp"  
android:layout_gravity="center_horizontal"  
android:text="DE DIESEL"  
android:textColor="#AA00FF"  
android:textStyle="bold"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.528"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.072" />
```

```
<ImageView
```

```
android:id="@+id/imageView2"  
android:layout_width="150dp"  
android:layout_height="200dp"  
  
android:layout_marginBottom="8dp"  
android:layout_marginTop="8dp"  
app:layout_constraintBottom_toTopOf="@+id/buttonSalir"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/textViewBody"  
app:srcCompat="@drawable/iess" />
```

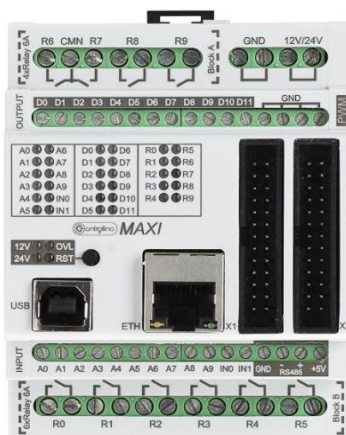
```
</android.support.constraint.ConstraintLayout>
```

Descripción: Este código genera la interfaz para el usuario:



ANEXO F

DATASHEET CONTROLLINO MAXI



Controllino is an Arduino standard and Arduino software compatible PLC.

General:

Standard	EN61010-1 EN61010-2-201 EN61131-2
Dimensions (W x H x D)	72x90x62mm
Weight	240g
Mounting	Top hat rail EN50022, 35mm

Environmental conditions, Indoor use only:

Operating ambient temperature	0°C – 55°C
Relative humidity – non-condensing	80 % for temp. up to 31 °C, decreasing linearly to 50 % relative humidity at 55 °C
Pollution Degree	PD2
Altitude	up to 2000m AMSL
Vibration (5 ≤ f ≤ 9 Hz)	1,75 mm amplitude sinus 3,5 mm amplitude random
Vibration (9 ≤ f ≤ 150 Hz)	0,5 g acceleration sinus 1,0 g acceleration random
Transport and Storage	-20°C – +70°C 10 to 90% no condensation Altitude 3000m AMSL
Shock response	15g, 11ms half sinus all 3 axes

I/O:

Supply voltage	12V or 24V
USB (Power for programming only)	USB-B, 2.0
Ethernet	RJ45, 10/100Mbps
RS485 (no termination inside)	250kb
Inputs, no galvanic insulation	12
Common analog/digital	10
Fixed digital, ext. Interrupt usable	2
Digital Outputs, no galvanic insulation	12
Relay output	10
PIN Header, no Galvanic insulation	
Logic level Input	12 parallel to terminal Inputs
Logic level Output	12 parallel to terminal Outputs
Communication	SPI, 2xUART, I2C, Reset
Internal Power	+3,3V, +5V, ARef, GND

Terminal capacities:

Relay Output, Power Input	2,5mm ² (24-12AWG)
Strip length	6-7mm
Max. tightening torque	0,5Nm
Digital, Analog Input Output	1,5mm ² (30-16AWG)
Strip length	5-6mm
Max. tightening torque	0,2Nm
Pin header connector	2x 26 Pin, Dual row, 2.54 pitch

Protection:

ESD HBM Class 0	Contact discharge: ±4kV Air discharge: ±8kV
Supply input over current protection	Internal Fuse 20A (fast)
Relay Output	External Fuse required
Digital Output	Overload, short circuit, ESD
Signal Input	Overvoltage, ESD
Pin header connector Current +5V, +3,3V	ESD total 200mA, resettable fuse

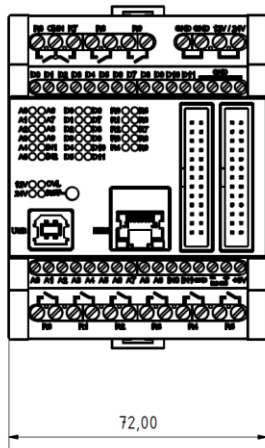
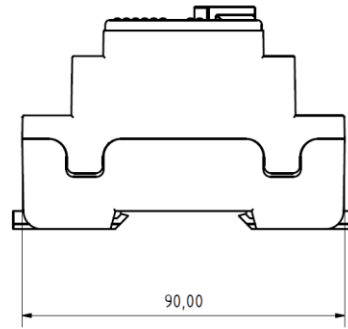
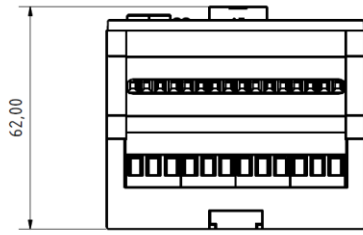
Electrical characteristics:

	Condition	Value
Supply voltage (Absolute Maximum)	12V range 24V range	10,2V – 15,0V 20,4V – 30,0V
Signal input low level	12V range 24V range	0V – 3,6V 0V – 7,2V
Signal input high level	12V range 24V range	9V – 13,2V 18V – 26,4V
Analog signal input	12V range 24V range	0 – 13,2V 0 – 26,4V
Signal input current	max. current	< 3mA
Logic “0” level	@ pin header	0V – 1,5V
Logic “1” level	@ pin header	3V – 5,5V
Signal output low level	12V range 24V range	0V – 2,4V 0V – 4,8V
Signal output high level		V _{in} – 10%
Signal output – PWM functionality	Duty cycle	15% - 85%
Relay output, Contact rating	Resistive Load	6A 250V AC / 30V DC
Common Relay terminal	max. current	6A
Galvanic insulation	coil to contact	3000VAC 1min
Relay ON in case of PWM functionality	Duty cycle	> 30%

LED signalization:

Power LEDs coding input voltage out of range e.g. only USB powered input voltage 10.2V – 15,0V input voltage 20.4V – 30,0V Input voltage < 5V	Color of power LED 12V orange, 24V orange 12V green, 24V orange 12V orange, 24V green both LEDs off
Device in reset state	Reset LED yellow
Device in run state	Reset LED off
Signal input at high (logic 1) level	Corresponding LED green
Signal input at low (logic 0) level	Corresponding LED off
Signal input in use as analog input	Corresponding LED green on when input level reach high (logic 1) state
Signal/Relay output set to active	Corresponding LED green
Signal/Relay output set to inactive	Corresponding LED off

Physical Dimensions:



ANEXO G

DATASHEET MB7138

XL-TrashSonar™ -WR™ Series

XL-TrashSonar™ -WR™ Series



IP67 Weather Resistant, Ultrasonic Trash Sensor

MB7137, MB7138, MB7139³

The XL-TrashSonar-WR sensor series provide users with robust range information in air. These sensors also feature high-power acoustic output along with real-time auto calibration for changing conditions (supply voltage sag, acoustic noise, or electrical noise), operation with supply voltage from 3.0V to 5.5V, object detection from 0-cm to 350-cm, and sonar range information from 20-cm out to 350-cm with 1-cm resolution. Objects from 0-cm to 20-cm typically range as 20-cm. (25cm for the MB7138²) The sensor is housed in a robust PVC housing, designed to meet the IP67 water intrusion standard. This sensor line is available in the following package types — 3/4" PVC threading, 1" NPS pipe threading, 1" BSPP pipe threading, and 30mm 1.5 pipe threading.



The user interface formats included are pulse-width, analog voltage output, RS232 serial output, and I2C (MB7137).

Features	Benefits	Applications & Uses
<ul style="list-style-type: none"> • Real-time auto calibration and noise rejection • High acoustic power output • Precise narrow beam • Object detection includes zero range objects • 3V to 5.5V supply with very low average current draw • Free run operation can continually measure and output range information • Triggered operation provides the range reading as desired • All interfaces are active simultaneously • RS232 Serial, 0 to Vcc, 9600 Baud, 81N (MB7138, MB7139) • I2C Bus operates up to 400KHz with clock stretching (MB7137) • Analog, (Vcc/1024) / cm (MB7138, MB7139) • Sensor operates at 42KHz 	<ul style="list-style-type: none"> • Acoustic and electric noise resistance • Reliable and stable range data • Robust, low cost IP67 standard sensor • Narrow beam characteristics • Very low power excellent for battery based systems • Ranging can be triggered externally or internally • Sensor reports the range reading directly, frees up user processor • Easy hole mounting or mating with standard electrical fittings • Filtering allows very reliable operation in most environments 	<ul style="list-style-type: none"> • Tank level measurement • Bin level measurement • Trash level measurement • Environments with acoustic and electrical noise • Distance measuring • Long range object detection • Industrial sensor • -40°C to +65°C operation¹ <p>Notes: ¹Please reference page 15 for minimum operating voltage verses temperature information. ²Minimum distance is 25cm for the MB7138 ³Please reference page 20 for part number key.</p>

MB7138 & MB7139 Pinout

Pin 1 - BW: Leave open (or high) for serial output on the Pin 5 output. When Pin 1 is held low the Pin 5 output sends a pulse (instead of serial data), suitable for low noise chaining.

Pin 2 - PW: This pin outputs a pulse-width representation of range. To calculate the distance, use a scale factor of 58uS per cm.

Pin 3 - AN: This pin outputs analog voltage with a scaling factor of (Vcc/1024) per cm. A supply of 5V yields ~4.9mV/cm., and 3.3V yields ~3.2mV/cm. The output is buffered and corresponds to the most recent range data.

Pin 4 - RX: This pin is internally pulled high. If Pin-4 is left unconnected or held high, the sensor will continually measure the range. If Pin-4 is held low the sensor will stop ranging. Bring high 20uS or more to command a range reading.

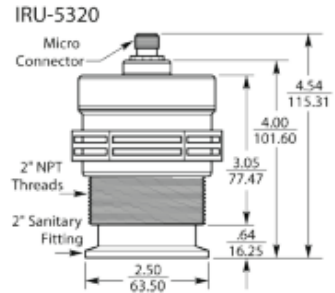
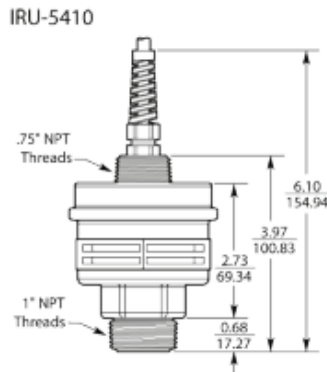
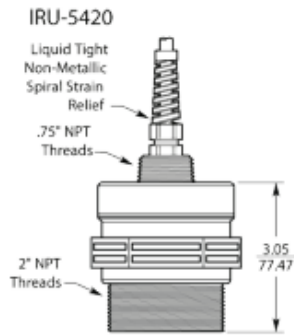
Pin 5 - TX: When Pin 1 is open or held high, the Pin 5 output delivers asynchronous serial data in an RS232 format, except the voltages are 0-Vcc. The output is an ASCII capital “R”, followed by ASCII character digits representing the range in centimeters up to a maximum of 350, followed by a carriage return (ASCII 13). The baud rate is 9600, 8 bits, no parity, with one stop bit. Although the voltages of 0V to Vcc are outside the RS232 standard, most RS232 devices have sufficient margin to read the 0V to Vcc serial data. If standard voltage level RS232 is desired, invert, and connect an RS232 converter such as a MAX232. When Pin 1 is held low, the Pin 5 output sends a single pulse, suitable for low noise chaining (no serial data).

Device Comparison Chart

Part Number	AN Voltage	Serial Data (0 to Vcc level)	Pulse Width	I2C Bus	Stability Filter	Most Likely Filter
MB7137				Yes		
MB7138	Yes	RS232	Yes		Yes	Yes
MB7139	Yes	RS232	Yes			

ANEXO H
DATASHEET IRU-5000

IRU-5000 Specifications



Performance

- Operating Range: 4 - 79 in. (102 - 2007 mm)
- Beam Pattern: 9° off axis
- Internal Temperature Compensation
- Frequency: 143 kHz
- Accuracy: ±0.25% of detected range
- Resolution: 0.1 in. (2.5 mm)
- Response Time: Programmable (20 ms minimum)

Connectivity

- Output:
- 4-20 mA
 - 4-20 mA with (2) NPN outputs
 - 0-2.5 V/0-5 V

Environmental

- Ratings: IP65
- Operating Temp: -40° - 140°F (-40° - 60°C)

Certification

- Class 1, Div. 2, Groups C, D, T6; Ex nA IIB T6
 - Class 1, Zone 2, Aex nA IIB, T6
 - IP65
- (*Models IRU-5413, -5415, -5423 & -5425 only)

Electrical

- Total Current Draw: 75 mA @ 24 VDC
- Supply Voltage: 12-28 VDC
- Wiring Connection: 4 or 5 conductor shielded cable

Programming

- Interface with RST-3001 Programmer (purchase separately - powered by USB)
 - The RST-3001 is 32 bit only
 - The RST-3002 is 32 and 64 bit compatible
- User selected units of measure

Physical

- PC/PET upper housing
- PC/PET Blend Transducer

IRU-5000 Accessories

Please order separately, by part number.

Description	Part Number
Over Molded Extension Cables with Foil Shield	
4 Conductor	
Straight, 2 m	135407-0002
Right Angle, 2 m	135407-1002
Straight, 5 m	135407-0005
Right Angle, 5 m	135407-1005
5 Conductor	
Straight, 2 m	135415-0002
Right Angle, 2 m	135415-1002
Straight, 5 m	135415-0005
Right Angle, 5 m	135415-1005
Programming Module	
RST-3001 (USB - 32 bit only)	125389
RST-3002 (USB - 32 and 64 bit compatible)	125389-1000

Wiring

IRU-5xx3-C	
Red	+24 VDC
Black	Ground
White	mA+

IRU-5xx3-M	
1 Brown	+24 VDC
2 White	Not used
3 Blue	Ground
4 Black	mA+
5 Grey	Not used



IRU-5xx9-C	
Red	+24 VDC
Black	Ground
White	mA+

IRU-5xx9-M	
1 Brown	+24 VDC
2 White	Not used
3 Blue	Ground
4 Black	Voltage Out
5 Grey	Not used



IRU-5xx5-C	
Red	+24 VDC
Black	Ground
Orange	mA+
White	NPN 1
Green	NPN 2

IRU-5xx5-M	
1 Brown	+24 VDC
2 White	NPN 1
3 Blue	Ground
4 Black	mA+
5 Grey	NPN 2



ANEXO I

DATASHEET MAX3232

RS232 to TTL Converter Module MAX3232. (Traducido por Rambal Ltda.)



Descripción

El módulo conversor RS232 a TTL MAX3232, puede ser utilizado para convertir el voltaje RS232 de un PC a TTL (lógica transistor a transistor) que utiliza el microcontrolador. Conecte el módulo directamente con el conector DB9 de su PC.

Comuniquen microcontroladores utilizando el puerto serial. Es aplicable a una amplia variedad de plataformas, incluyendo Arduino, AVR, ARM, etc.

Para Arduino tenga en cuenta que ya existe una conexión UART2USB, sólo utilice este módulo si desea una conexión serial extra para su equipo.

Características:

- Chip: MAX3232.
- Voltaje: 3.3V/5V.

Lista de partes:

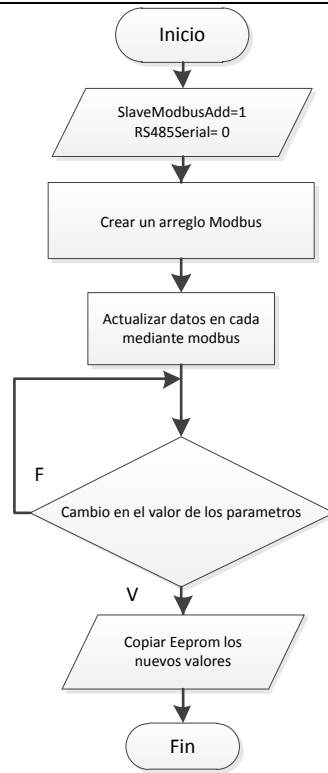
- RS232 a TTL Module x1.
- 4 Pin DuPont Line x1.

ANEXO J
SUBRUTINA 1

SUBRUTINA 1: Adquisición de datos para el cálculo del volumen	
Diagrama de Flujo	<pre> graph TD Inicio([Inicio]) --> Lectura[Lectura de Datos Puertos Análogos] Lectura --> Decision{millis() >= timer + 250} Decision -- F --> Lectura Decision -- V --> Sensor[Valor del sensor Volumen del tanque] Sensor --> Calculo[Calculo del Volumen] Calculo --> Fin([Fin]) </pre>
Proceso	<p>El proceso inicia con la lectura de los puertos análogos de las señales que envían los transductores, el muestreo del volumen del tanque se realiza cuatro veces por segundo (250 milisegundos) y para obtener el número de galones se utiliza la fórmula del volumen para cilindros.</p>

SUBROUTINA 2: Comunicación Modbus con la pantalla HMI

Diagrama de Flujo

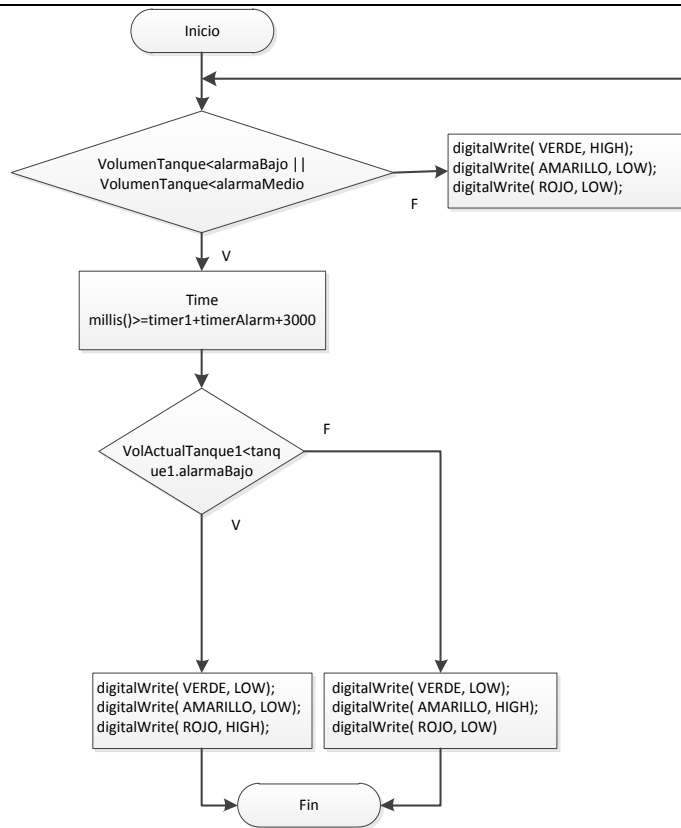


Proceso

En este proceso se utiliza el protocolo Modbus, esto se realiza a través de la comunicación serial entre el Controllino y la pantalla HMI, se enlaza cada dato ingresando con el número registro del arreglo de Modbus. Los datos que se enlazan son: las dimensiones de los tanques, valor de offset, escala del sensor y número de galones. El número de galones de diesel que generan una alerta son ingresados a través de la pantalla y en caso de existir algún cambio en alguno de los datos enlazados se envía a guardar el cambio en la memoria EEPROM del Controllino.

SUBRUTINA 3: Alertas luces

Diagrama de Flujo

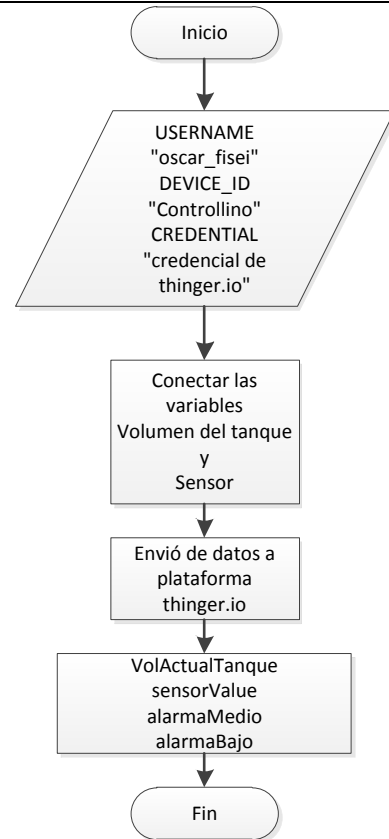


Proceso

Este proceso inicia con un condicional y se procede a comparar el valor del volumen actual con el valor configurado para las alertas. Para la alerta nivel medio (volumen entre la mitad y un cuarto del tanque) se activa una salida para encender una luz amarilla y si se tiene un nivel bajo (volumen menor a un cuarto de tanque) se activa una salida para encender una luz roja, en caso de ser falso la condición siempre se mantiene activa una salida para encender una luz verde

SUBROUTINA 4: Conexión a thinger.io para el envío de datos y notificaciones

Diagrama de Flujo



Proceso

El flujograma de este proceso inicia configurando las credenciales para la identificación para establecer la conexión con la plataforma de Thinger.io. Las variables de volumen del tanque y valor del sensor son enviados a la plataforma para mostrarlos remotamente. Al interconectar las variables recibe constantemente la información de las alertas de niveles inapropiados para enviar las notificaciones al dispositivo móvil.

Anexo K

Manual de Instalación

Inicialmente se observa la forma actual de medición del tanque en el cual se va a instalar el sensor, para el caso del proyecto de investigación, el prototipo será instalado en el tanque de la foto que se muestra a continuación:

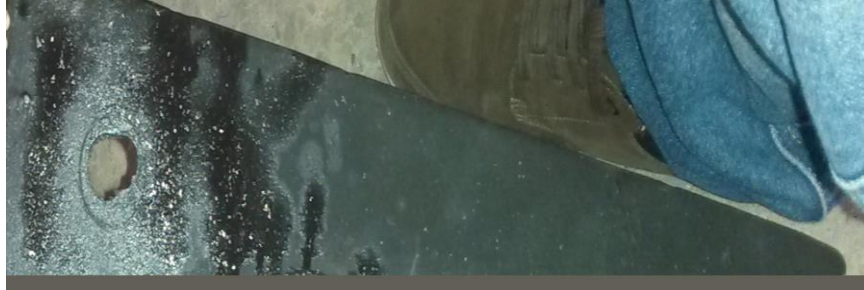


Como se puede observar en la siguiente imagen se utiliza una manguera de nivel, y se para registrar el valor se debe observar aproximadamente donde se encuentra el nivel y calcular según unas etiquetas sin numeración.



A continuación, se presenta los pasos a seguir para realizar la instalación del sistema electrónico de monitoreo del almacenamiento de diesel.

- a) Perforación de tanque para ubicar el sensor



- b) Instalación del sensor y conexión según el datasheet del sensor



- c) Fijar el sensor al tanque de almacenamiento



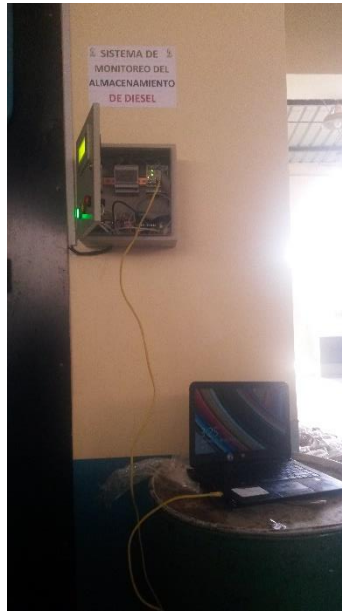
d) Tendido de canaletas



e) Fijar el prototipo



f) Pruebas de funcionamiento.



g) Revisión del sistema por parte de los tutores del Hospital General Ambato.



h) Finalización del proyecto.

