



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

Tema:

**SISTEMA DE DOSIFICACIÓN INTELIGENTE BASADO EN EL INTERNET
INDUSTRIAL DE LAS COSAS (IIOT) PARA LOS TANQUES DE MEZCLA
EN LA EMPRESA LICOVAL S.A.S.**

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Electrónica y Comunicaciones

ÁREA: Física y Electrónica

LÍNEA DE INVESTIGACIÓN: Sistemas electrónicos

AUTOR: Marco Javier Pico Aldas

TUTOR: Ing. Santiago Mauricio Altamirano Meléndez

Ambato – Ecuador

marzo -2023

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA DE DOSIFICACIÓN INTELIGENTE BASADO EN EL INTERNET INDUSTRIAL DE LAS COSAS (IIOT) PARA LOS TANQUES DE MEZCLA EN LA EMPRESA LICOVAL S.A.S. desarrollado bajo la modalidad Proyecto de Investigación por el señor Marco Javier Pico Aldas, estudiante de la Carrera de Ingeniería Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, marzo 2023

Ing. Santiago Mauricio Altamirano Meléndez

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA DE DOSIFICACIÓN INTELIGENTE BASADO EN EL INTERNET INDUSTRIAL DE LAS COSAS (IIOT) PARA LOS TANQUES DE MEZCLA EN LA EMPRESA LICOVAL S.A.S. 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, marzo 2023



Marco Javier Pico Aldas

C.C. 1804418109

AUTOR

DERECHOS 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 a favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, marzo 2023



Marco Javier Pico Aldas

C.C. 1804418109

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Marco Javier Pico Aldas, estudiante de la Carrera de Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA DE DOSIFICACIÓN INTELIGENTE BASADO EN EL INTERNET INDUSTRIAL DE LAS COSAS (IIOT) PARA LOS TANQUES DE MEZCLA EN LA EMPRESA LICOVAL S.A.S., nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, marzo 2023

Ing. Pilar Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Pamela Castro, Mg.

PROFESOR CALIFICADOR

Ing. Andrea Sánchez, Mg.

PROFESOR CALIFICADOR

DEDICATORIA

El presente trabajo está dedicado a Jehová Dios, quien es mi fortaleza, y siempre me ha bendecido para cumplir mis objetivos. A mi madre, Doris Aldas que nunca me ha dejado caer, siempre esta cuando la necesito, a mi padre, Marco Pico, por el apoyo incondicional que me ha enseñado, a que todo lo que me proponga siempre requiere fuerza y dedicación, a mis hermanos Carolina, Christian, Richard, que con el amor de familia siempre nos hemos apoyado unos a otros.

Marco Javier Pico Aldas

AGRADECIMIENTO

Agradezco a mis padres, por todo el apoyo que me han brindado, por la paciencia y confianza que tuvieron durante esta carrera universitaria, porque nunca dejaron de confiar en mí, porque nunca me hicieron faltar nada, gracias por todo.

A mi hermana Carolina que siempre estuvo ahí, siempre preocupada por cómo me va en mis estudios, dándome un empujoncito cuando más lo necesitaba, en todo este proceso de estudios universitarios.

A mi familia, por todo el apoyo y ayuda que me brindaron.

A mi amigo Israel, el cual ha sido una persona que se puede confiar fuera de mi familia, siempre apoyándome y dando sus consejos para seguir estudiando.

A la Universidad Técnica de Ambato en especial a mi querida Facultad de Ingeniería en Sistemas, Electrónica e Industrial que con sus docentes fueron parte de mi crecimiento profesional y personal, gracias por todo el tiempo y experiencias compartidas.

A mi tutor el Ing. Santiago Altamirano, por la confianza y todo el apoyo que se necesita para que este trabajo de titulación sea posible.

A la Empresa Licoval S.A.S que me permitió demostrar mi conocimiento y poder realizar este trabajo de titulación.

¡¡¡Muchas Gracias!!!

Pico Aldas Marco Javier

ÍNDICE GENERAL

APROBACIÓN DEL TUTOR.....	II
AUTORÍA.....	III
DERECHOS DE AUTOR.....	IV
APROBACIÓN TRIBUNAL DE GRADO	V
DEDICATORIA	VI
AGRADECIMIENTO.....	VII
ÍNDICE GENERAL.....	VIII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIV
RESUMEN EJECUTIVO	XV
ABSTRACT.....	XVI
CAPÍTULO I.....	1
MARCO TEÓRICO.....	1
1.1. Tema de Investigación	1
1.2. Antecedentes Investigativos.....	1
1.2.1. Contextualización del problema	4
1.2.2. Fundamentación Teórica	6
1.2.2.1. Automatización Industrial.....	6
1.2.2.2. Internet de las Cosas	6
1.2.2.3. Industria 4.0	7
1.2.2.4. Tecnologías de la Industria 4.0.....	7
1.2.2.5. El Internet Industrial	10
1.2.2.6. El Internet Industrial De Las Cosas (IIoT)	10
1.2.2.7. Proceso de Dosificación.....	11
1.2.2.8. Medición de Nivel	12
1.2.2.9. Sensor de Nivel.....	12

1.2.2.10. Caudalímetro.....	13
1.2.2.11. Bomba de Agua	13
1.2.2.12. PLC	14
1.3. Objetivos	14
1.3.1. Objetivo general	14
1.3.2. Objetivos específicos.....	14
CAPÍTULO II	15
METODOLOGÍA	15
2.1. Materiales.....	15
2.2. Métodos.....	15
2.2.1. Modalidad de la Investigación	15
2.2.1.1. Investigación bibliográfica.....	15
2.2.1.2. Investigación de Campo.....	15
2.2.1.3. Investigación Experimental.....	15
2.2.2. Recolección de Información	16
2.2.3. Procesamiento y Análisis de Datos	16
2.2.4. Desarrollo del Proyecto.....	16
CAPÍTULO III.....	18
RESULTADOS Y DISCUSIÓN.....	18
3.1. Análisis y discusión de los resultados.....	18
3.1.2. Desarrollo de la propuesta.....	18
3.1.2.1. Análisis del actual proceso de dosificación	18
3.1.2.2. Determinación de la problemática en el proceso de dosificación.....	23
3.1.2.3. Análisis de la Tecnología a utilizar	24
3.1.2.4. Análisis del sistema de comunicación en Industria 4.0.....	25
3.1.2.5. Selección del Controlador	26
3.1.2.6. Selección de tarjeta de desarrollo para comunicación	27

3.1.2.7. Selección de elementos que actualmente posee la empresa.	28
3.1.2.8. Selección de los elementos a utilizar para el control y la automatización. ..	36
3.1.2.9. Selección software de diseño del control y la automatización.....	40
3.1.2.10. Diseño del control según los componentes seleccionados.....	40
3.1.2.11. Desarrollo de la programación para el procesamiento de las variables	42
3.1.2.12. Elaboración de la interfaz gráfica, visualización de datos, y monitoreo remoto	47
3.1.2.13. Implementación del sistema de dosificación inteligente.....	61
3.1.2.14. Desarrollo de las pruebas de funcionamiento.	67
3.2.1.1 Resultados	69
3.2.1.2 Presupuesto del Proyecto	70
CAPÍTULO IV.....	71
CONCLUSIONES Y RECOMENDACIONES.....	71
4.1. Conclusiones	71
4.3. Recomendaciones.....	72
Referencias Bibliográficas	72
Anexos.....	77
Anexo1 – DataSheet PLC	77
Anexo 2 – Sensor Ultrasónico JSN-SR04T-2.0.....	79
Anexo 3 – Modulo TM1638	80
Anexo 4 – Plano general de la empresa	81
Anexo 5 – Diagrama P&ID de la dosificación.....	82
Anexo 6 – Programación del ESP 32.....	83
Anexo 7 – Programación del PLC	94
Anexo 8 – Programación del ESP6266 Tanque 5000 L	109
Anexo 9 – Programación del ESP6266 Tanque 10000 L	111

ÍNDICE DE FIGURAS

Figura - 1- Bases tecnológicas de la industria 4.0[13].....	7
Figura - 2 Sensor de nivel [20].....	12
Figura - 3- Caudalímetro[21]	13
Figura - 4- Bomba de Agua[22].....	13
Figura - 5- Tanque de Almacenamiento de Alcohol - Fábrica Licoval S.A.S.....	18
Figura - 6- Tanque de Almacenamiento de Agua - Fábrica Licoval S.A.S.....	19
Figura - 7 - Tanque se mezcla con capacidad de 5000 L.....	20
Figura - 8- Tanque se mezcla con capacidad de 10000 L.....	20
Figura - 9- Accionamiento de la Bomba de Alcohol	21
Figura - 10 - Funcionamiento de la Bomba de Agua.....	21
Figura - 11- Accionamiento de los tanques de mezcla	22
Figura - 12 - Panel de Control.....	22
Figura - 13- Líneas marcadas en los tanques de mezcla	23
Figura - 14 - Piso mojado, en los tanques me mezcla.....	24
Figura - 15- Arquitectura IoT de Tres Capas [25]	25
Figura - 16- PLC AMSAMOTION AMX-FX3U-26MR[26].....	27
Figura - 17- Tarjeta De Desarrollo Esp-32 [27].....	28
Figura - 18- Bomba para el Caudal de Alcohol	29
Figura - 19- Placa Técnica de la Bomba de Alcohol	29
Figura - 20- Bomba para el Caudal de Agua	30
Figura - 21- Placa técnica de la Bomba de Agua.....	30
Figura - 22- Motor del Tanque de Mezcla	31
Figura - 23- Circuito de Fuerza del Panel de Control Principal.....	33
Figura - 24- Circuito de Control del Panel de Control Principal	34
Figura - 25- Conexiones Internas del panel de control	34
Figura - 26- Diagrama de Control - Bomba de Alcohol	35
Figura - 27- Sensor Ultrasónico Sensor HCSR04 [28].....	37
Figura - 28- Sensor Ultrasónico SUP-MP[29].....	38
Figura - 29- Sensor Ultrasónico Anti agua[30].....	38
Figura - 30- Modulo de Relé.....	38
Figura - 31- Modulo Esp8266[31]	39
Figura - 32- Módulo Tm1638 para Arduino[32]	40

Figura - 33- Entorno de Desarrollo Arduino[33]	40
Figura - 34- Diseño de Control de la Dosificación	41
Figura - 35- Diagrama de flujo General de la ESP32	42
Figura - 36- Subrutina de la comunicacion con HMI	43
Figura - 37- Subrutina de la comunicación con TCP/IP	44
Figura - 38- Subrutina de la comunicación con PLC	45
Figura - 39 - Diagrama de Flujo de la programación del PLC	46
Figura - 40- Diagrama de Flujo de la programación de ESP8266	47
Figura - 41- Creación de la Cuenta en Blynk.....	48
Figura - 42- Ingreso a la cuenta Blynk.....	48
Figura - 43- Creación de una plantilla	49
Figura - 44- Creación de los Pines Virtuales	49
Figura - 45- Lista de Datastreams	50
Figura - 46- Configuración del Widget.....	50
Figura - 47- Web Dashboard.....	51
Figura - 48- Pestaña Mobile Dashboard	51
Figura - 49- Creación de nuevo dispositivo	52
Figura - 50- Creación de Dispositivo.....	52
Figura - 51 - Token de Seguridad para colocar en ESP32	53
Figura - 52- Programación en la ESP32 para la comunicación entre la plataforma Blynk y la ESP32	53
Figura - 53- App Blynk IoT	54
Figura - 54- Configuración de Dashboard de la App Blynk IoT	54
Figura - 55- Dashboard General de la App	55
Figura - 56- Planes de la Plataforma Blynk	56
Figura - 57- Descargar Reportes Históricos.....	56
Figura - 58 Carpeta comprimida de datos históricos	57
Figura - 59- Archivos con Datos Históricos de la Dosificadora	57
Figura - 60- Pestaña de Eventos.....	57
Figura - 61- Creación del nuevo evento - Notificaciones	58
Figura - 62- Configuración de Notificaciones Vía Correo Electrónico	59
Figura - 63- Programación del enlace de la comunicación con Blynk IoT.....	59
Figura - 64- Reporte de Notificaciones en la App	60

Figura - 65- Notificación al encender la bomba de Agua	60
Figura - 66- Orificio para la colocación del sensor Ultrasónico - Tanque de 5000 Litros	61
Figura - 67- Caja de Distribución Eléctrica para los sensores	61
Figura - 68- Ubicación de la caja que contiene el módulo Esp8266 y el módulo de sensor ultrasónico.....	62
Figura - 69- Caja del módulo del sensor 10000 L.....	62
Figura - 70- Colocación del PLC en el panel de Control.....	63
Figura - 71- Panel de Control con PLC y Reles.....	63
Figura - 72- Conexiones de las protecciones a los módulos de relé	64
Figura - 73- Conexiones del PLC	64
Figura - 74- Corte con láser de la caja para el módulo Tm1638.....	65
Figura - 75- HMI en nuestro tablero de control	66
Figura - 76- Conmutación de control de la Bomba de Alcohol cerca del panel principal.....	66
Figura - 77- Comunicación del sistema con nuestra Plataforma Blynk.....	67
Figura - 78- Cantidad de litros en el tanque de 10000 L.....	68
Figura - 79- Configuración del Parámetro 21 - cantidad de litros a ingresar en el tanque de 10000 L.....	68
Figura - 80- Etiquetas de las funciones del HMI	68
Figura - 81- App en tiempo real de la Dosificación.....	69

ÍNDICE DE TABLAS

Tabla 1- Características de PLCs	26
Tabla 2- Comparación de Tarjetas de desarrollo	27
Tabla 3- Características Bomba de Alcohol.....	28
Tabla 4- Características Bomba de Alcohol.....	29
Tabla 5- Características Motor de Mezcla	31
Tabla 6- Funciones Panel del principal de Control Eléctrico.....	32
Tabla 7- Sensores Ultrasónicos	37
Tabla 8- Presupuesto Económico	70

RESUMEN EJECUTIVO

Este proyecto de investigación, describe un sistema de dosificación inteligente, que se basa en la Industria 4.0 para la empresa Licoval S.A.S. El problema existía al tener un control manual de encendido y apagado de las bombas de agua y alcohol, que llenan los tanques de mezcla y mediante un control visual de la cantidad de líquido que ingresan a los tanques se apagaba las bombas. Este sistema inteligente consta de tres capas. La primera capa es la de dispositivos, que está conformado por sensores ultrasónicos que están conectados a una ESP8266, para transmitir de manera inalámbrica los datos adquiridos y actuadores como las bombas y motores que van conectados al PLC Amsamotion. La segunda capa es de procesamiento, que establece la comunicación entre el PLC y el ESP32 esta es una comunicación serial TTL para el procesamiento de los datos adquiridos y a su vez establece las funciones a realizar. En esta etapa se elabora un HMI, con el módulo de display TM6138 para la interacción directa con el operador. La tercera capa es la nube, que a través de una infraestructura tipo SaaS como es la plataforma para aplicaciones Blynk IoT, se realizó un dashboard, para visualizar los datos en tiempo real del sistema, además esta plataforma brinda el almacenamiento de datos históricos con reportes y notificaciones de las acciones que se realiza en el sistema inteligente de dosificación. El proyecto fue implementado y puesto a prueba durante tres semanas en las cuales se realizaron dosificaciones dependiendo la demanda de pedidos de la empresa y se verificó la dosificación inteligente en los tanques de mezcla, suministrando de manera exacta la materia prima y controlando los tiempos de mezcla de los tanques.

Palabras clave: IIOT, Industria 4.0, automatización, dosificación, blynk IoT.

ABSTRACT

This research project describes an intelligent dosing system, which is based on Industry 4.0 for the company Licoval S.A.S. The problem existed by having a manual control on and off the water and alcohol pumps, which fill the mixing tanks and through a visual control of the amount of liquid entering the tanks the pumps were turned off. This intelligent system consists of three layers. The first layer is the device layer, which is made up of ultrasonic sensors that are connected to an ESP8266 to wirelessly transmit the acquired data and actuators such as pumps and motors that are connected to the Amsamotion PLC. The second layer is processing, which establishes the communication between the PLC and the ESP32, this is a TTL serial communication for processing the acquired data and in turn establishes the functions to be performed. At this stage an HMI is elaborated, with the TM6138 display module for direct interaction with the operator. The third layer is the cloud, which through a SaaS infrastructure such as the Blynk IoT application platform, a dashboard was created to visualize the system data in real time, and this platform also provides historical data storage with reports and notifications of the actions performed in the intelligent dosing system. The project was implemented and tested for three weeks in which dosing was performed depending on the demand of the company's orders and the intelligent dosing was verified in the mixing tanks, supplying the raw material accurately and controlling the mixing times of the tanks.

Keywords: IIOT, Industry 4.0, automation, dosing, blynk IoT.

CAPÍTULO I

MARCO TEÓRICO

1.1.Tema de Investigación

SISTEMA DE DOSIFICACIÓN INTELIGENTE BASADO EN EL INTERNET INDUSTRIAL DE LAS COSAS (IIOT) PARA LOS TANQUES DE MEZCLA EN LA EMPRESA LICOVAL S.A.S.

1.2.Antecedentes Investigativos

La investigación fue realizada en varios repositorios de las Universidades Nacionales e internacionales, a su vez se recolectó información de papers y artículos científicos de diferentes revistas con la información que relaciona la automatización de la dosificación de los tanques de mezcla.

En junio de 2022, Israel Ponce presenta su artículo científico con el título de “Integración de tecnologías asociadas con el internet de las cosas y la industria 4.0”, esta investigación se enmarca en tres tecnologías diferentes en el campo de la ingeniería, como son: redes inalámbricas de sensores, almacenamiento en la nube y visualización de datos mediante aplicaciones móviles. Estas integraciones son posibles al aumentar la probabilidad de su uso en entornos cotidianos debido a los desarrollos tecnológicos relacionados con Internet de las cosas (IoT) y la cuarta revolución industrial. La red de sensores inalámbricos se implementa utilizando el módulo Digi XBee S2C del proveedor del protocolo Digimesh junto con un microcontrolador Arduino Uno y un módulo WiFi ESP-01. La base de datos se implementó en Firebase y se desarrolló una aplicación para el sistema operativo Android para visualizar los datos. La implementación e integración de varias tecnologías puede monitorear varias variables deseadas en tiempo real y brindar a los usuarios soluciones prácticas y económicas.[1]

En Quito, en el año 2021, Álvarez Oswaldo y Arroyo Flavio en su trabajo de titulación “Análisis de la Industria 4.0 como Factor Diferenciador del Sector Industrial del Ecuador” indica que la migración a la industria inteligente se realiza con las restricciones de la infraestructura existente y capacidad tecnológica instalada, de aquí parte la brecha que complica la adopción de la industria 4.0. este trabajo plantea una metodología que permita, en primera instancia evaluar de manera cualitativa y rápida

cual es el estado actual del nivel de madurez de la implementación de la industria 4.0 en función de factores de éxito que son resultado de la experiencia de países que ya han generado lecciones aprendidas en dicha implementación. La evaluación cualitativa propuesta brinda una visión rápida del nivel de madurez de los factores de éxito: estrategia, tecnología, capacidad de innovación, ecosistemas de ayuda para la innovación y habilidades y manejo del cambio, en Ecuador todavía no es visible una política pública o el interés privado para dar ese salto a la nueva revolución industrial. [2]

En agosto de 2021, Gustavo Barona López y Luis Efraín Velastegui realizaron el artículo científico, “Automatización de procesos industriales mediante Industria 4.0” El artículo tiene por objetivo presentar un modelo automatizado de procesos industriales mediante la Industria 4.0, para establecer los niveles jerárquicos a partir de la pirámide de automatización. Para esto, se estructuró una teoría acerca de Industria 4.0. Posteriormente, se presenta la mecanización de procesos industriales relacionados a envasado de bebidas, medición de flujo, tecnología de red Wireless, integración total de niveles y refinación de petróleo, en cada uno se muestra su arquitectura para realizar la evaluación de sus niveles en referencia al concepto de industrialización. En la evaluación, se describieron los instrumentos de campo y sistema SCADA. Se concluye que la red de comunicación Ethernet industrial por su alta capacidad de transmitir datos entre los niveles de empresa, cableado de gran extensión, facilidad de comunicación entre controladores e ideal para redes de área local (LAN), es utilizada en el Nivel 2, Nivel 3 y Nivel 4, mientras que las redes de comunicación PROFIBUS, PROFINET, HART, IO-Link y AS-Interface, por su transferencia de datos en tiempo real entre instrumentos de campo y controladores, trabajo en ambientes explosivos, manejo de tráfico de datos, recuperación de datos en la red, es utilizada en los niveles 0 y 1 en la automatización de procesos industriales. [3]

En Costa Rica, en el año 2021, Sergio Solorzano Alfaro desarrolló un prototipo de sistema de control y monitoreo hídrico, que se basa en la tecnología inalámbrica LoRaWAN, para el acueducto principal de la Asociación Administradora del Acueducto Rural de Playa Sámara de Nicoya, usando tecnología IoT. El proyecto consistía en el desarrollo de tres dispositivos denominados ASADA V1.0, los cuales son administrados por un microcontrolador Heltec LoRa, este recibe los datos de un

sensor ultrasónico Octave, adquiriendo las unidades de volumen y caudal, además cuenta con un sistema de alimentación energética autónomo basado en paneles solares. En caso que se presente una falla en la toma de mediciones, se incorporó luces piloto que permiten alertar el problema, una vez realizadas las pruebas de funcionamiento, se obtuvo que el margen de error del prototipo varía entre 0% y 0,73%. Finalmente, el sistema generó un ahorro económico, con un margen de 5 a 10 meses en recuperar la inversión. [4]

En Madrid, en el año 2021, Ignacio Benito Urquijo presenta su proyecto de fin de grado que se centra en la automatización y control de stock en tiempo real de diferentes tipos de materiales en el final de su cadena de producción. Habiendo tomado la idea principal de la problemática del almacenaje de aluminio al finalizar su proceso de producción, en un patio, y su posterior salida al cliente. Por todo ello, el principal objetivo del proyecto es desarrollar el software necesario para poder realizar la comunicación entre el dispositivo encargado de leer el producto y una aplicación gráfica de usuario que controla en tiempo real, las diferentes etapas que atraviesa el producto antes de su envío al cliente. Se uso de una red Wifi definida para IoT, que será la encargada de enviar los datos a un Cloud que hará de intermediario, para finalmente volcar la información en la aplicación gráfica que hará lo propio en una base de datos local MySQL. A través del caso práctico se mostro la viabilidad de este proyecto, siempre de una manera que sea a escala, como posible inclusión en una industria actual. Permitiendo el ahorro de coste y una mayor facilidad a la hora de trabajar con los diferentes productos que pudieran salir.[5]

En la ciudad de Riobamba, en el año 2019, Erick Cristhian Soliz Oliveros implementó un sistema IoT, para el módulo de nivel y caudal, en el laboratorio de control y manipulación automática de la Escuela de Ingeniería de Mantenimiento. El sistema adquiere los datos por medio de un caudalímetro y un sensor ultrasónico, una vez obtenidos eran transmitidos mediante tecnología Node-RED e IoT hacia el controlador PLC S7-1200 encargado de la administración del sistema. La visualización de las variables se realiza mediante una pantalla HMI de manera local y a través de dispositivos móviles de forma remota. Al implementar este Sistema se logra introducir una nueva funcionalidad en la Pasarela Simatic, con bases en la Industria 4.0. [6]

Estos antecedentes investigativos, ayudan como fundamento de las tecnologías que utilizamos en el presente trabajo de investigación con la implementación de una arquitectura IoT para la dosificación inteligente.

1.2.1. Contextualización del problema

El mundo globalizado de hoy tiene un alto nivel de competitividad motivo por el cual se ha definido que la automatización en las empresas se constituya en uno de los pilares de mucha importancia para el sector productivo en cualquier país, implicando el uso de diversas tecnologías a fin de controlar un sinnúmero de procesos industriales.[7]

El término IoT Industrial se centra en la aplicación de las ventajas que ofrece el IoT a los ámbitos industriales y empresariales. Dicho de otra manera, toda la maquinaria de una industria conectada a internet gracias a la captación de datos que proviene de sensores y dispositivos, esta nueva etapa se le conoce como Industria 4.0; que considera la transparencia y conectividad de la línea de producción como factores fundamentales para la competitividad de una empresa, es decir, la detección de máquinas y herramientas permitirá una mejor visualización de toda la cadena de producción y, por lo tanto, las decisiones de producción, logística y suministro se llevarán a cabo de una manera más segura, minimizando los costos y aumentando la eficiencia. [8]

En Ecuador, la Industria 4.0 comenzó a tener más fuerza a partir del año 2018 con la presentación del Libro Blanco de la Sociedad de la Información y el Conocimiento por parte del Ministerio de Telecomunicaciones de Ecuador, con el objetivo “de dar a conocer la estrategia que ayudará al desarrollo de la sociedad y el conocimiento para impulsar el crecimiento económico, la equidad e inclusión y la eficiencia en la administración pública”. De acuerdo al estudio realizado por el World Economic Forum de “Readiness for the Future of Production“, que tiene como objetivo apoyar los esfuerzos de las economías e industrias 4.0, con el fin de liderar una transición inclusiva al panorama de producción con miras al futuro, el Ecuador se encuentra en el puesto 89/100 (2.85 puntos) en la estructura de producción y en componentes en el puesto 90/100 (3.66 puntos). Esta calificación se debe a la baja puntuación en cuanto a Tecnología e Innovación. La situación actual para la industria ecuatoriana, es que las tecnologías se encuentran de una forma aislada y no se encuentra un orden o una conexión entre ellas que permita a las empresas (grandes, medianas o pymes). [9]

LICOVAL S.A.S. es una empresa ambateña dedicada a la elaboración, producción y comercialización de bebidas alcohólicas. Esta empresa tiene varias fases para elaborar sus productos. Sin embargo, no todos sus procesos se encuentran completamente automatizados, como es el proceso de dosificación. Para esto la fábrica posee tanques de almacenamiento, uno de agua y otro de alcohol que suministran mediante tuberías a los dos tanques de mezcla. Actualmente, este proceso se lo realiza de forma manual con los siguientes detalles:

- En los tanques de mezcla existen líneas marcadas por los operarios para que realicen un control visual de la cantidad de líquido que existe.
- Para un control más exacto sumergen reglas graduadas con indicador de nivel para establecer la cantidad de litros de los tanques.
- Para la dosificación de alcohol y agua en los tanques de mezcla, los operadores activan o desactivan bombas por medio de pulsadores.

Esto genera un retraso de tiempo en la mezcla de agua con alcohol, además de un deficiente control del proceso, la cantidad de líquidos no son exactos en las mezclas lo que ocasiona el desperdicio de líquidos (materias primas) por goteos y la contaminación de los líquidos por partículas ambientales en el entorno.

Este proyecto es de suma importancia porque controla con precisión la cantidad de líquidos que van a pasar al tanque de mezcla. La inmersión del proceso de dosificación en el IIoT, permite que se obtenga con exactitud el mezclado de líquidos, esto genera el ahorro de costes y de tiempo, mejora de la eficiencia y del consumo energético, menos incidencias y mayor capacidad para concentrar los recursos, además contará con un sistema de monitoreo y control remoto mejorando el rendimiento de la empresa.

Los beneficiarios de este nuevo sistema inteligente de dosificación, basado en la Industria 4.0 es la empresa, los dueños y los trabajadores de la misma, porque mejora los tiempos de elaboración de los productos, aumentando la producción diaria del producto. A la vez, los distribuidores del producto pondrán tener en stock y a tiempo los pedidos que requieran, beneficiándose indirectamente del proyecto, junto a los consumidores finales.

La implementación del proyecto es factible para la empresa porque tendrá un sistema de control inteligente al mezclar las proporciones adecuadas y exactas de la materia

prima, reduciendo desperdicios y mejorando la economía de la empresa. No dependerá de la mano humana, al no utilizar un operario para el apagado de las bombas y motores, lo que optimiza el desempeño de la empresa.

1.2.2. Fundamentación Teórica

1.2.2.1. Automatización Industrial

La automatización de los procesos industriales constituye uno de los objetivos más importantes de las empresas en la siempre incesante tarea de la búsqueda de la competitividad en un entorno cambiante y agresivo. La automatización de un proceso industrial, (máquina, conjunto o equipo industrial) consiste en la incorporación al mismo, de un conjunto de elementos y dispositivos tecnológicos que aseguren su control y buen comportamiento. Dicho automatismo, en general ha de ser capaz de reaccionar frente a las situaciones previstas de antemano y además frente a imponderables, tener como objetivo situar al proceso y a los recursos humanos que lo asisten en la situación más favorable.[10]

1.2.2.2. Internet de las Cosas

El término Internet de las Cosas se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana. Esta tecnología se encarna en una amplia gama de productos, sistemas y sensores en red, que aprovechan los avances en la potencia de cálculo, la miniaturización de los componentes electrónicos y las interconexiones de red para ofrecer nuevas capacidades que antes no eran posibles.[11]

Las implementaciones del IoT utilizan diferentes modelos de conectividad, cada uno de los cuales tiene sus propias características. Los cuatro de los modelos de conectividad descritos por la Junta de Arquitectura de Internet incluyen: Device-to-Device (dispositivo a dispositivo), Device-to-Cloud (dispositivo a la nube), Device-to-Gateway (dispositivo a puerta de enlace) y Back-End Data-Sharing (intercambio de datos a través del back-end). Estos modelos destacan la flexibilidad en las formas en que los dispositivos de la IoT pueden conectarse y proporcionar un valor para el usuario[11]

1.2.2.3. Industria 4.0

Las tres primeras revoluciones industriales se caracterizan por estar impulsadas por la producción mecánica basada en la energía del agua y el vapor, el uso de mano de obra masiva, energía eléctrica, y el uso de la producción electrónica automatizada, respectivamente. Mientras que la supuesta cuarta revolución industrial ('Industria 4.0') se propuso por primera vez en 2011 en el contexto del objetivo de desarrollar la economía alemana. Esta revolución se caracteriza por su apuesta por el uso de Sistema Ciber Físico(CPS) capaces de comunicarse entre sí y de tomar decisiones autónomas y descentralizadas, con el objetivo de aumentar la eficiencia industrial, la productividad, la seguridad y la transparencia. [12]

1.2.2.4. Tecnologías de la Industria 4.0

Las tecnologías que respaldan la Industria 4.0 y la fabricación inteligente incluyen la simulación, fabricación aditiva, sistemas integrados horizontal y verticalmente, ciberseguridad, realidad aumentada, computación en la nube, robótica automatizada, internet industrial de las cosas y big data, así como análisis avanzados. [13]

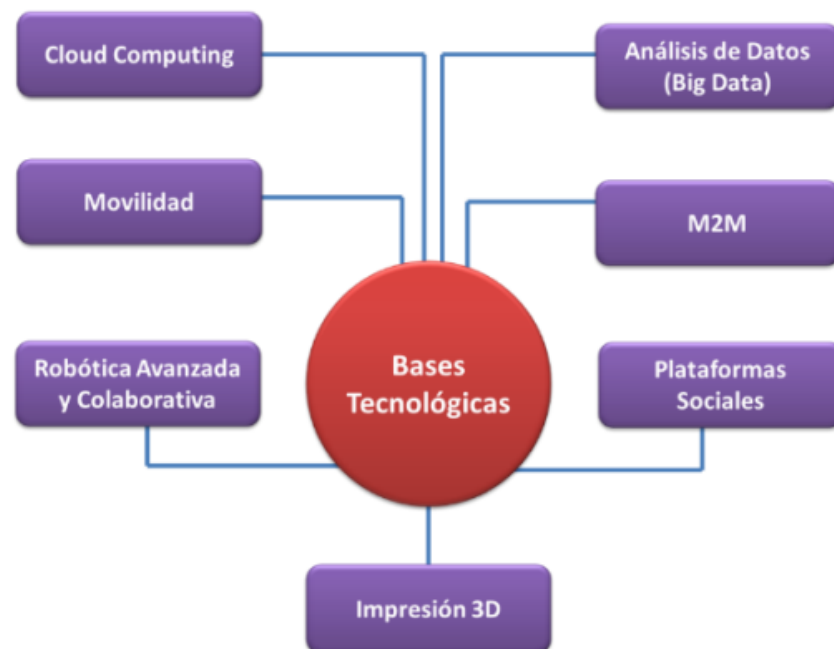


Figura - 1- Bases tecnológicas de la industria 4.0[13]

Además, sistemas de realidad virtual, dispositivos móviles (tabletas, teléfonos), plataformas y aplicaciones tecnológicas, así como inteligencia artificial y tecnologías de la información. [13]

Comunicación Móvil

El internet móvil es la base de IoT. El etiquetado de objetos y la comunicación M2M permiten entornos de producción conectados donde interactúan los sistemas y los productos. Esto permite la recopilación de datos, la coordinación de los Sistemas de Producción CyberFísicos (CPPS) y la implementación de servicios remotos. Y todo esto en tiempo real y en todas partes. [13]

Cloud Computing

La nube consta de aplicaciones e infraestructura entregadas como un servicio a través de una red pública o privada, a menudo sobre la base de pago por uso. Los productos y sistemas inteligentes generarán enormes cantidades de datos para almacenar y procesar a los que se debe acceder en línea desde cualquier lugar. La nube permite un tráfico de datos ilimitado y elimina la necesidad de invertir en infraestructura para aumentar la capacidad, proporcionando una flexibilidad sin precedentes. [14]

Análisis De Datos (Big Data)

A medida que aumenta la cantidad de productos inteligentes en las fábricas y en el mercado, la cantidad de datos disponibles para los fabricantes seguirá creciendo. Su análisis le permitirá identificar patrones e interdependencias, analizar procesos y detectar ineficiencias, e incluso predecir eventos futuros. Esto abrirá nuevas oportunidades no solo para aumentar la eficiencia, sino también para abrir los servicios a los clientes, que serán más conocidos.[15]

Comunicación Máquina a Máquina (M2M)

La comunicación M2M es la tecnología subyacente de Internet de las cosas (IoT). Se refiere a tecnologías que permiten el intercambio de información entre productos y sistemas inteligentes creando un entorno de Industria 4.0. Además, utilizando esta información, puede crear una copia virtual de la fábrica física, lo que le permite modelar no solo los productos sino los procesos de fabricación en general. La aplicación más obvia de la comunicación M2M sería la interconexión de los sistemas internos de la empresa, así como un elemento clave de la comunicación entre empresas. [15]

Plataforma Social

Las plataformas sociales han cambiado nuestra vida diaria con su capacidad de comunicarse de manera directa, global e instantánea. La comunicación en entornos industriales mejorará drásticamente con interacciones ricas en contenido más dinámicas que fomentan la colaboración y la innovación. Por otro lado, las redes sociales "clásicas" facilitarán la fabricación personalizada y proporcionarán una gran cantidad de información sobre el cliente.

Impresión 3D (fabricación aditiva).

Esto se aplica a la producción de objetos 3D a partir de modelos virtuales. Aunque su aplicación industrial es pequeña, la fabricación aditiva abordará las brechas de eficiencia en la producción de productos personalizados. Esto garantizará la creación rápida de prototipos y un alto grado de producción descentralizada: las muestras de productos se pueden enviar a la ubicación de "impresión" más cercana al cliente, sin la participación de intermediarios. [13]

Robots Avanzados Y Colaborativos

Las mejoras en inteligencia artificial, junto con un nuevo sistema sensorial, han permitido la creación de robots cada vez más autónomos, flexibles y colaborativos. Con el tiempo, interactuarán entre sí y trabajarán de manera segura con humanos, aprendiendo de ellos y brindando muchas capacidades mucho más allá de las que se usan en la fabricación actual. [13]

Realidad Aumentada

Los sistemas basados en realidad aumentada están actualmente en pañales, pero en el futuro permitirán proporcionar información en tiempo real a los empleados para mejorar la toma de decisiones y el flujo de trabajo. Por ejemplo, los trabajadores podrán recibir instrucciones de reparación y visualizarla mediante dispositivos como gafas de realidad aumentada, aumentando así la eficiencia y la seguridad. [14]

1.2.2.5.El Internet Industrial

El concepto de Internet industrial fue articulado por primera vez por General Electric (GE) [16], y se describe como: “La definición de internet industrial incluye dos componentes clave: la conexión de sensores y actuadores de máquinas industriales al procesamiento local y al internet. La conexión posterior con otras redes industriales importantes que pueden generar valor de forma independiente. La principal diferencia entre el internet social de consumo y el internet industrial radica en cómo y cuánto valor se crea. Para los consumidores de internet social la mayor parte del valor se crea a partir de anuncios”. [17]

Esta descripción separa claramente internet social con respecto al internet industrial, aunque en ambos casos la función de internet es proporcionar redes de área amplia. Más recientemente, Internet industrial se ha definido como: “Una fuente tanto de eficiencia operativa como de innovación, que es el resultado de una receta convincente de desarrollos tecnológicos”.

La suma resultante de esas partes le brinda Internet industrial: la estrecha integración de los mundos físico y digital. El Internet industrial permite a las empresas usar sensores, software, aprendizaje de máquina a máquina y otras tecnologías para recopilar y analizar datos de objetos físicos u otros grandes flujos de datos, y luego usar esos análisis para administrar operaciones y, en algunos casos, para ofrecer nuevos, servicios de valor añadido"[18]

1.2.2.6.El Internet Industrial De Las Cosas (IIoT)

Una definición inicial de IIoT podría ser: el uso de ciertas tecnologías de IoT (ciertos tipos de objetos inteligentes dentro de sistemas físicos cibernéticos) en un entorno industrial, para la promoción de objetivos distintivos de la industria. [18]

Varios investigadores que escriben en alemán ofrecen un grupo de definiciones de IIoT que comparten un enfoque en los tipos de tecnologías que se ponen en funcionamiento en los entornos de IIoT y las formas en que se utilizan en esos entornos. Se sugiere que un elemento central de IIoT es su dependencia [32], en un entorno industrial, de objetos, sistemas y maquinaria que se ha actualizado al estado de un sistema ciber físico (CPS), para que los productos y servicios puedan ser guiados a través del suministro y cadenas de valor de manera autónoma.

“Internet industrial de las cosas, es un sistema que comprende objetos inteligentes en red, activos ciber físicos, tecnologías de información genéricas asociadas y plataformas informáticas en la nube o de borde opcionales, que permitir el acceso, la recopilación, el análisis, las comunicaciones y el intercambio de información de procesos, productos y/o servicios en tiempo real, inteligentes y autónomos, dentro del entorno industrial, a fin de optimizar el valor general de la producción. Este valor puede incluir; mejorar la entrega de productos o servicios, aumentar la productividad, reducir los costos de mano de obra, reducir el consumo de energía y reducir el ciclo de fabricación bajo pedido”. [12]

1.2.2.7. Proceso de Dosificación

Se realiza mediante una solución tecnológica o sistema, que distribuye un material en cantidades y secuencias temporales exactas. Para cubrir esta necesidad tecnológica se han diseñado una serie de sistemas que permite llegar al objetivo de medir y pesar las cantidades a dosificar

Tipos de dosificación industrial

La puesta en marcha de este proceso puede realizarse de dos formas:

- **Proceso de dosificación volumétrica:** Sistema que funciona basado en el control del volumen del producto a dosificar.
- **Proceso de dosificación gravimétrica:** Este sistema funciona en base a la pérdida de peso del material durante un espacio de tiempo. Este sistema incluye un sistema de pesaje que permite mantener el control de la cantidad del material dosificado.[19]

Beneficios del proceso de dosificación

A continuación, se detalla algunos de los beneficios de utilizar soluciones tecnológicas para el proceso de dosificación.

- Mantener un control automatizado sobre el producto dosificado.
- Permite mejorar la homogeneidad de la mezcla final.
- Ahorro en costes del producto dosificado.
- Alta precisión y mayor productividad.[19]

1.2.2.8. Medición de Nivel

Los sensores de medición de nivel son una parte integral del control de procesos en muchas industrias y se dividen en dos tipos principales. Los sensores de medición de nivel de punto se utilizan para marcar una única altura de líquido separada, una condición de nivel predefinida. En general, este tipo de sensor opera con una alarma alta, lo que indica una condición de desbordamiento o un marcador de condición de alarma baja. Los sensores de nivel continuo son más sofisticados y pueden proporcionar monitoreo de nivel para un sistema completo. Miden el nivel de líquido dentro de un rango en lugar de un punto y producen una salida analógica que se correlaciona directamente con el nivel en el recipiente. Para crear un sistema de gestión de nivel, la señal de salida está vinculada a un circuito de control de proceso y un indicador visual.[20]

1.2.2.9. Sensor de Nivel



Figura - 2 Sensor de nivel [20]

El sensor de nivel tiene la función de detectar el nivel puntual y evitar la falta o el desbordamiento de líquido dentro del tanque u otro contenedor, a través del control de nivel de silos, tanques y controles de activación de la bomba. El sensor de nivel de agua es un dispositivo con un mecanismo activado por presión, que automatiza el llenado y vaciado de tanques con la ayuda de un flotador, un tipo de diafragma, mediante un interruptor, el sensor de nivel de agua limita el nivel de agua en el recipiente. Es muy útil para los sectores industriales que lidian con el agua en la producción de sus productos. [20]

1.2.2.10. Caudalímetro



Figura - 3- Caudalímetro[21]

Un caudalímetro es un instrumento de medida para la medición de caudal o gasto volumétrico de un fluido o para la medición del gasto másico. Estos aparatos suelen colocarse en línea con la tubería que transporta el fluido. También suelen llamarse medidores de caudal, medidores de flujo o flujómetros. Existen versiones mecánicas y eléctricas. Un ejemplo de caudalímetro eléctrico se encuentra en los calentadores de agua de paso que lo utilizan para determinar el caudal que está circulando o en las lavadoras para llenar su tanque a diferentes niveles.[21]

1.2.2.11. Bomba de Agua



Figura - 4- Bomba de Agua[22]

Es la máquina que transforma energía, aplicándola para mover el agua. Este movimiento, normalmente es ascendente. Las bombas pueden ser de dos tipos “volumétricas” y “turbo-bombas”. Todas constan de un orificio de entrada (de aspiración) y otro de salida (de impulsión). Las volumétricas mueven el agua mediante la variación periódica de un volumen. Es el caso de la bomba de émbolo. Las turbobombas poseen un elemento que gira, produciendo así el arrastre del agua. [22]

1.2.2.12. PLC

Un controlador lógico programable, más conocido por sus siglas en inglés PLC (Programmable Logic Controller), se trata de una computadora, utilizada en la ingeniería automática o automatización industrial, para automatizar procesos electromecánicos, tales como el control de la maquinaria de la fábrica en líneas de montaje o atracciones mecánicas. Sin embargo, la definición más precisa de estos dispositivos es la dada por la NEMA (Asociación Nacional de Fabricantes Eléctricos) que dice que un PLC es: “Instrumento electrónico, que utiliza memoria programable para guardar instrucciones sobre la implementación de determinadas funciones, como operaciones lógicas, secuencias de acciones, especificaciones temporales, contadores y cálculos para el control mediante módulos de E/S analógicos o digitales sobre diferentes tipos de máquinas y de procesos”. [23]

1.3. Objetivos

1.3.1. Objetivo general

- Implementar un sistema de dosificación inteligente basado en Internet Industrial de las Cosas (IIoT) para los tanques de mezcla en la Empresa LICOVAL S.A.S.

1.3.2. Objetivos específicos

- Analizar el sistema de control actual que tiene el proceso de dosificación.
- Determinar los elementos y dispositivos necesarios para el sistema de dosificación, basado en el Internet Industrial de las Cosas (IIoT)
- Diseñar el sistema inteligente de control para la automatización de proceso de la dosificación.
- Desarrollar el sistema electrónico y la interfaz de visualización con notificaciones de estado automáticas para la dosificación de los tanques.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

Este proyecto de investigación se desarrolló e implementó con una selección de materiales de acuerdo a artículos científicos, proyectos de investigación con características similares, para la implementación del tema propuesto se utilizó fichas técnicas de los componentes electrónicos necesarios para el sistema. Los recursos tecnológicos que se utilizaron para el desarrollo del proyecto son sensores, actuadores, microcontroladores, un PLC y elementos de comunicación inalámbrica

2.2. Métodos

2.2.1. Modalidad de la Investigación

El presente proyecto consta con una investigación aplicada, porque se utilizó los conocimientos adquiridos durante la formación académica para aplicarlos en el diseño y control del proceso de dosificación.

2.2.1.1. Investigación bibliográfica

La revisión bibliográfica se obtuvo de libros, revistas, artículos y bases de datos científicas de diversos repositorios de las Universidades Nacionales e Internacionales que tengan alguna relación con el tema de investigación, principalmente las bases de datos científicas.

2.2.1.2. Investigación de Campo

La investigación de campo se realizó directamente en la empresa con la finalidad de recolectar información y analizar el estado actual del proceso de dosificación de materia prima.

2.2.1.3. Investigación Experimental

Finalmente, la investigación experimental se realizó con un conjunto de pruebas y actividades con el fin de recolectar información necesaria, mediante diversas pruebas en los diseños de automatización y control de variables.

2.2.2. Recolección de Información

La recolección de la información del presente proyecto de investigación se analizó de diferentes fuentes de información como libros, artículos, proyectos de investigación, disponibles en los diferentes repositorios de las Universidades Nacionales e Internacionales

2.2.3. Procesamiento y Análisis de Datos

En el presente proyecto el procesamiento y análisis de datos se realizó mediante la revisión de la información recabada, el estudio de las opciones para dar solución a la problemática de la dosificación de materia prima a los tanques de mezcla, extraer la información más relevante que ayude a la ejecución de una propuesta de solución del problema.

- Interpretación de la información adquirida.
- Estudio de los dispositivos que se utilizaron.
- Análisis del diseño que previamente se realizó.
- Resultados de acuerdo con los objetivos planteados

2.2.4. Desarrollo del Proyecto

La presente investigación se desarrolló, en base a las siguientes actividades:

1. Análisis de la información que tiene actualmente el sistema de control de la empresa.
2. Evaluación de la situación actual del proceso de dosificación a los tanques de mezcla en la empresa.
3. Determinación de la problemática que representa el proceso para la empresa.
4. Verificación de las diferentes tecnologías que se puede utilizar para dar solución a la problemática.
5. Análisis del funcionamiento de un sistema de comunicación en Industria 4.0.
6. Selección de tarjeta de desarrollo idónea para comunicación, enfocados en la Industria 4.0.
7. Selección los elementos que actualmente posee la empresa para utilizar en el sistema de automatización inteligente.
8. Análisis de los elementos que se va a utilizar en las diferentes etapas del proyecto.

9. Selección de los elementos a utilizar para dar el control y la automatización del proyecto.
10. Selección del software de diseño del control y la automatización.
11. Diseño del control según los componentes seleccionados.
12. Desarrollo de la programación para el procesamiento de las variables.
13. Elaboración de una interfaz gráfica que permita la correcta visualización de los datos, y el monitoreo remoto, obtenidos de todos los sensores.
14. Implementación del sistema de dosificación inteligente.
15. Desarrollo de las pruebas de funcionamiento.
16. Elaboración del informe final.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

En este capítulo se presenta el desarrollo de la propuesta donde se describen los segmentos que conforman el sistema, los dispositivos utilizados, la programación y por ende el análisis de los resultados obtenidos en las pruebas de funcionamiento.

3.1. Análisis y discusión de los resultados.

La implementación del sistema inteligente que controle el flujo de líquidos para los tanques de mezcla, se lo realizó por medio de un PLC y microcontroladores, estableciendo un protocolo de comunicación basado en IoT Industrial y la elaboración de una interfaz gráfica, con notificaciones de estado, monitoreo remoto y reportes históricos, permitiendo una medición más exacta y funcional, con la visualización en tiempo real de la cantidad de líquido.

3.1.2. Desarrollo de la propuesta

3.1.2.1. Análisis del actual proceso de dosificación

El proceso de dosificación se da a partir de del almacenamiento de la materia prima con dos tanques reservorios de alcohol y un tanque reservorio de agua, que mediante tuberías se traslada estos líquidos a dos tanques de mezcla que existe actualmente en la empresa. Para el almacenamiento del alcohol se utiliza 2 tanques de similares características, como observamos en la figura 4, un tanque de color azul y otro de color blanco que están ubicados en la parte externa de la empresa, esta ubicación se da por medidas de seguridad puesto que el alcohol es un producto inflamable.



Figura - 5- Tanque de Almacenamiento de Alcohol - Fábrica Licoval S.A.S

Fuente: Investigador

El agua que se utiliza para la mezcla se conserva en un tanque de primario de 15000 litros (Figura 6), para luego pasar por el proceso de desinfección, a continuación, se da el proceso de osmosis y finalmente por proceso de purificación, una vez concluidos estos pasos el agua llega al tanque de almacenamiento. Este tanque está ubicado justo en la mitad de los dos tanques de mezcla en el interior de la fábrica en el área de producción.



Figura - 6- Tanque de Almacenamiento de Agua - Fábrica Licoval S.A.S

Fuente: Investigador

Actualmente, la empresa cuenta con dos tanques de mezcla, un tanque tiene una capacidad de 5000 litros (Figura 7) y el otro tanque una capacidad de 10000 litros (Figura 8), en el interior de estos tanques se encuentra unas hélices que al accionar un motor estas giran para que se integren los ingredientes del producto final.



Figura - 7 - Tanque se mezcla con capacidad de 5000 L

Fuente: Investigador



Figura - 8- Tanque se mezcla con capacidad de 10000 L

Fuente: Investigador

El traslado de alcohol, hacia los tanques de mezcla se lo realiza mediante un pequeño panel de control (Figura 9), el cual contiene dos pulsadores ON-OFF y dos indicadores luminosos, las botoneras realizan la función de encender y apagar una bomba, cuando la luz piloto de color verde está encendida, indica que la bomba esta activa, y cuando la luz piloto de color rojo se activa indica que el relé térmico entro en funcionamiento por una sobrecarga a la bomba.



Figura - 9- Accionamiento de la Bomba de Alcohol

Fuente: Investigador

El traslado de agua hacia los tanques de mezcla se da desde un panel de control que realiza la función de encender y apagar una bomba de agua, mediante un selector de dos posiciones, y se verifica que la bomba esté encendida atreves de una luz piloto de color verde, una luz piloto color roja encendida indica que el relé térmico se acciono, por una sobrecarga en la bomba, como podemos observar en la figura 10.



Figura - 10 - Funcionamiento de la Bomba de Agua

Fuente: Investigador

El funcionamiento de los tanques de mezcla, se realiza mediante el accionamiento de dos motores, un motor bifásico para cada tanque, estos son controlados por un selector de dos posiciones ON-OFF que se encuentran en el panel del control (Figura 11), además de una luz piloto color verde, que indica si el motor este encendido o apagado, y una luz piloto color roja que indica alguna sobrecarga.



Figura - 11- Accionamiento de los tanques de mezcla

Fuente: Investigador

La dosificación es de manera manual, porque se requiere de un operario para encender las bombas de agua y de alcohol y mediante un control visual se estima que la cantidad de líquido requerido llega a los tanques de mezcla se apaga las bombas.

Los motores de los tanques de mezcla se accionan cuando todos los ingredientes han sido colocados en dichos tanques y el operario se encarga de encender, apagar mediante un selector de manera manual.

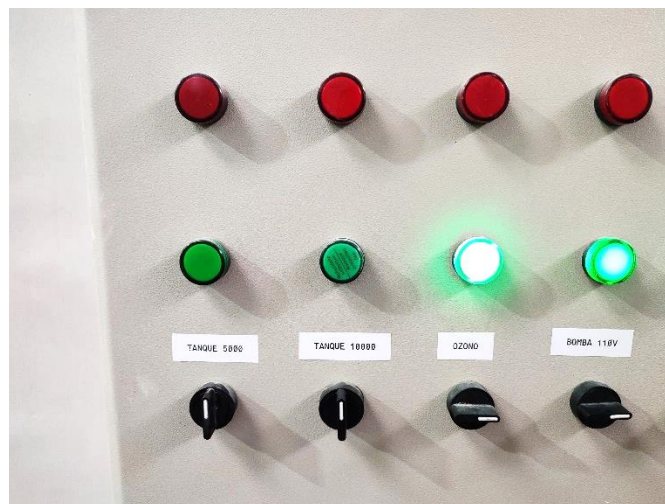


Figura - 12 - Panel de Control

Fuente: Investigador

El plano general de la ubicación de los tanques de mezcla, tanques de almacenamiento de agua y alcohol se puede ver en el Anexo 4.

La conexión general del proceso de dosificación se puede ver en el Anexo 5, para esto se utiliza el diagrama P&ID que demuestra las conexiones sistemáticas de las tuberías, la instrumentación y las partes del equipo del sistema.

3.1.2.2. Determinación de la problemática en el proceso de dosificación.

Al requerir una cierta cantidad de agua o alcohol, para los tanques de mezcla se enciende la respectiva bomba, la cual, el operario para medir la cantidad de líquido que necesita, debe estar pendiente de unas líneas marcadas manualmente en los tanques de mezcla, que indican la cantidad de líquido que existe en el tanque, como se observan en la Figura 13. En la investigación, los operarios manifestaron que en algunas ocasiones se les olvida apagar las bombas de agua o de alcohol, lo que produce que se llene el tanque y sobrepase su capacidad, produciendo que se derrame la materia prima en el piso, ocasionando pérdida del producto y a la vez se crea un problema para los operarios que caminan por ese sector de la planta, porque se puede producir algún tipo de accidente con el piso mojado (Figura 14).



Figura - 13- Líneas marcadas en los tanques de mezcla

Fuente: Investigador



Figura - 14 - Piso mojado, en los tanques me mezcla

Fuente: Investigador

Con estos inconvenientes, se produce un retraso en la producción diaria, que resulta en una pérdida economía para la empresa, además de un deficiente control del proceso, porque la cantidad de líquido no es exacta para la formulación de los diferentes productos de la empresa.

3.1.2.3. Análisis de la Tecnología a utilizar

Un artículo científico, publicado en 2019, hace la comparativa de la comunicación entre las redes inalámbricas de Industria 4.0, propone una solución basada en la nube compatible con la Calidad de Servicio (QoS) mediante la adaptación de una arquitectura de uso compartido de recursos que transparente la propuesta recientemente al escenario de IoT.

Los resultados de las simulaciones muestran que las métricas de QoS tanto de retardo como de fluctuación se mantienen por debajo de sus umbrales específicos en el contexto de las aplicaciones de VoIP utilizadas para el control de ajuste fino de manipuladores distribuidos. Al analizar los resultados, es posible darse cuenta de que la mayoría del tráfico se dirige a las redes Wi-Fi. Esto ocurre porque una de las reglas del algoritmo del controlador de recursos es priorizar los recursos de red baratos. [24]

Dado que las redes Wi-Fi pertenecen al régimen de uso compartido, cuando conduce a un mejor costo-beneficio, considerando que la red de destino es capaz de garantizar

los requisitos de QoS de la aplicación IIoT. Por tanto, una media del 50 al 60% del tráfico generado por los dispositivos IIoT se dirige a esta red. El tráfico restante lo alojan proveedores de recursos más caros, es decir , 4G y 5G. [24]

3.1.2.4. Análisis del sistema de comunicación en Industria 4.0.

El desarrollo de las Tecnologías de la información y la comunicación (TICs) ayuda a conectar el mundo físico y el mundo digital a través de dispositivos, materiales, productos, equipos, medios y medios de comunicación con el mundo digital expresado a través de sistemas colaborativos y software conectado con multitud de dispositivos para acelerar el desarrollo. En las fábricas del futuro, estos estándares estarán claramente definidos y utilizarán interfaces comúnmente aceptadas. La comunicación conjunta será la clave del éxito. El uso de estas tecnologías permitirá el reemplazo flexible de máquinas reparadas o modernizadas a lo largo de la cadena de valor. Adaptarse a los cambios del mercado y la productividad traerá enormes beneficios. [25]

La arquitectura IoT de tres capas que se referencia en el modelo IBM (ver Figura 32), que separa las aplicaciones en tres niveles informáticos lógicos y físicos, es la arquitectura de software predominante para las aplicaciones cliente-servidor tradicionales. Está conformada por la capa de dispositivos, la capa de procesamiento Fog y la capa Nube.

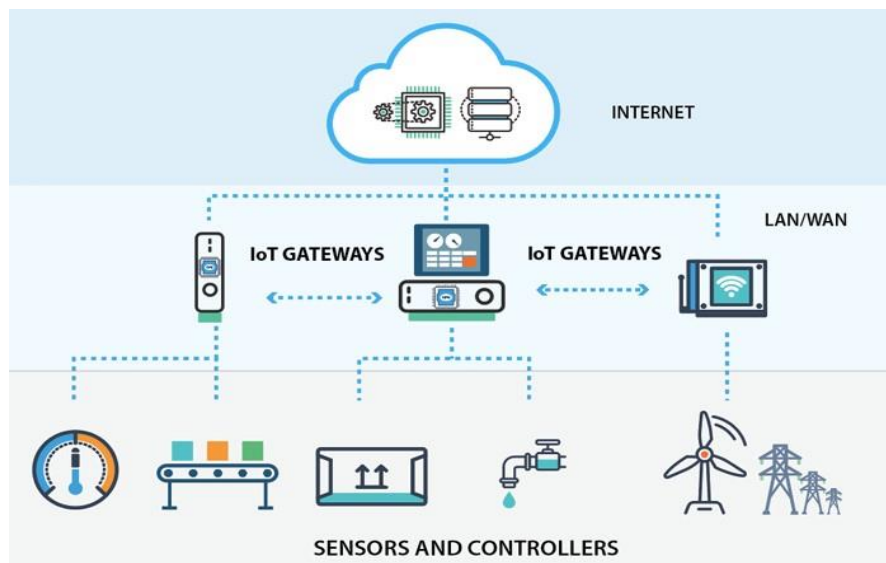


Figura - 15- Arquitectura IoT de Tres Capas [25]

3.1.2.5. Selección del Controlador

Los controles industriales, y especialmente los controladores lógicos programables (PLC), son ahora una base tecnológica importante para la automatización de procesos industriales. Incluso en la era de la Industria 4.0 y el Internet Industrial, se puede decir que estos controladores seguirán siendo necesarios en gran medida en la producción.

El controlador será el dispositivo encargado de ajustar los componentes eléctricos y electrónicos para asegurar las condiciones de dosificación en base a los datos recibidos de los sensores, en la tabla 1, se presentan las características de diferentes controladores.

Tabla 1- Características de PLCs

Marca	AMSAMOTION	SIEMENS	Controllino
Modelo	AMX-FX3U-26MR	LOGO 8	mini
Microcontrolador	Cortex M3		ATmega328P
V de Operación	12 - 24 VDC	115-230V AC/DC	12-24 VDC
Entradas Digitales	16	8	16
Entradas Analógicas	2	2	2
Salidas de Relay	10	4	8
Salidas Analógicas	2	0	4
Certificaciones	CE, UL	cULus y FM.	CE,
Precio	\$ 200	\$ 265	\$ 130
Disponibilidad en el País	Si	Si	No

Elaborado por: El Investigador

Según las características analizadas en la Tabla 1, el más apto para el proyecto es el PLC marca AMSAMOTION modelo AMX-FX3U-26MR , porque está disponible en el país y posee el número de salidas de Relay que son necesarias para controlar las bombas del proyecto , demás que tiene las certificaciones necesarias para trabajar en un área industrial.



Figura - 16- PLC AMSAMOTION AMX-FX3U-26MR[26]

3.1.2.6. Selección de tarjeta de desarrollo para comunicación

Una "placa de desarrollo" es un circuito electrónico que contiene un microcontrolador o dispositivo lógico y componentes como puertos, conectores y controladores que permiten a los usuarios acceder rápida y fácilmente a dispositivos periféricos para aplicaciones y pruebas.

Tabla 2- Comparación de Tarjetas de desarrollo

Nombre	Raspberry Pi 4BCM2711	ESP8266 NODEMCU V2	ESP32
CPU	BCM2711, Cortex-A72 (ARM v8) SoC de 64 bits 1,5 GHz	ESP8266 Tensilica Xtensa LX3 (32 bit)	Xtensa Dual-Core 32-bit LX6 con 600 DMIPS
Disponibilidad en Ecuador	SI	SI	SI
Voltaje de Alimentación	5 V CC - 3 A	5 V - 12 V	5V DC
Frecuencia de Reloj	1,5 GHz	52 MHz	hasta 240Mhz
Wifi	SI	SI	SI
Bluetooth	SI	NO	SI
Pines Digitales E/S	40	11	24 (Algunos pines solo como entrada)
Pines Analógicos E/S	-	1	2 ADC de 12bits
SRAM	8 GB DDR3	n. A	520 KByte
Precio	\$ 260	\$9,50	\$ 29

Elaborado por: El Investigador

Al realizar la comparación entre los microcontroladores se aprecia la similitud entre los mismos permitiendo desarrollar las mismas funciones, sin embargo, se selecciona el microcontrolador ESP32, debido al costo - beneficio porque permitirá la comunicación entre todos los demás dispositivos que son necesarios en el proyecto y a su vez permitirá la comunicación con la nube y el HMI.



Figura - 17- Tarjeta De Desarrollo Esp-32 [27]

3.1.2.7. Selección de elementos que actualmente posee la empresa.

A continuación, detallamos los elementos que actualmente tiene la empresa para poder realizar el proceso de dosificación, cabe recalcar que, en el mes de diciembre del año 2022, se hicieron algunos cambios en la empresa, entre ellos se destaca, que se cambió el lugar físico donde se realizaba el proceso, lo que llevo a empresa a contratar una nueva infraestructura de tuberías y cableado de control eléctrico. Esta nueva infraestructura cuenta con una garantía de 1 año. Los equipos para el funcionamiento del proceso, son los siguientes:

Bomba de Alcohol. - Para el movimiento del caudal del alcohol, se utiliza una bomba de agua de las siguientes características:

Tabla 3- Características Bomba de Alcohol

Modelo: SCM-26ST/HSE116-1HF	Comercializador: SARNES
Tensión - nominal: 110V~	Corriente nominal: 12A
Potencia nominal hasta 1000msnm: 1200W	Frecuencia nomina: 60Hzlfc
Velocidad nominal: 3450RPM	Número de fases: 1
Factor de potencia: 0.90	Eficiencia Energetica:75%
Grado de protección IP44	Potencia motor 1.6HP
Caudal Máximo: 125 L/min	Elevación Máximo: 26m
Clase de aislamiento :B/130°C	Referencia:E0777

Elaborado por: El Investigador



Figura - 18- Bomba para el Caudal de Alcohol

Fuente: El Investigador



Figura - 19- Placa Técnica de la Bomba de Alcohol

Fuente: El Investigador

Bomba de Agua.- Para el movimiento del caudal del agua, se utiliza una bomba de agua de las siguientes características:

Tabla 4- Características Bomba de Alcohol

Modelo : 65WUA200-I	
Tensión-nominal: 220 - 240 voltios	Corriente nominal: 8A
Potencia nominal hasta 1000msnm: 1200W	Frecuencia nomina: 60Hz
Velocidad nominal: 3450RPM	Potencia motor 1HP
Factor de potencia: 0.90	Elevación Máximo: 26m
Caudal Máximo: 635 L/min	AMB 50°C

Elaborado por: El Investigador

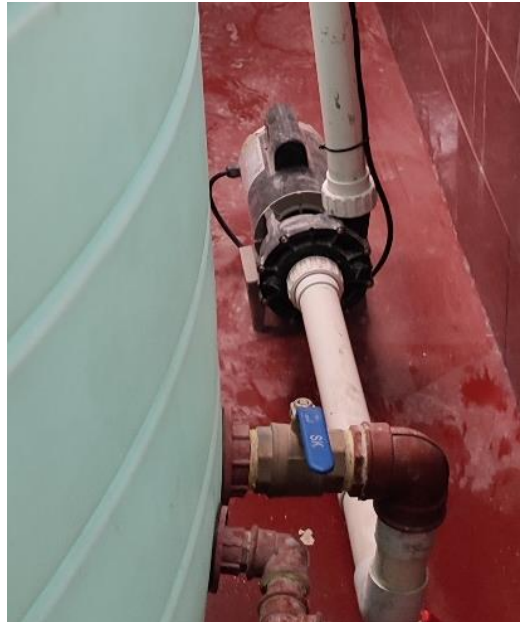


Figura - 20- Bomba para el Caudal de Agua

Fuente: El Investigador



Figura - 21- Placa técnica de la Bomba de Agua

Fuente: El Investigador

Motor de Mezcla tanque 5000 L. Una vez que todos los ingredientes del producto se encuentran en el tanque de mezcla, se procede a encender un motor el cual mueve unas aspas en el interior del tanque por un aproximado de 30 minutos, esto se hace para que se integren los ingredientes, este motor tiene las siguientes características:

Tabla 5- Características Motor de Mezcla

Marca	WEG
Modelo	Abierto
Potencia HP	2
Potencia KW	1,49142431
Tensión	110/220
Tipo Carcasa	G56H
Medida Eje	15.87
RPM	1800
Polos	4
Tolerancia %	15
Temperatura	130

Elaborado por: El Investigador



Figura - 22- Motor del Tanque de Mezcla

Fuente: El Investigador

Motor de Mezcla Tanque 10000 L. Cuando todos los ingredientes del producto están en el tanque de mezcla, el motor se activa y mueve las espas en el tanque durante unos 40 minutos, esto se hace para combinar los ingredientes, el motor tiene las mismas características de la Tabla 5.

Panel de Control Eléctrico. Para el control de cada una de las bombas y motores existe 2 paneles de control, uno que es exclusivamente para la bomba de alcohol,

(Figura 9), y otro panel principal que consta de seis selectores de dos posiciones (Figura 10), y cada selector cumple con la función de accionar un motor o bomba de manera independiente.

El panel principal acciona las siguientes funciones (de izquierda a derecha):

Tabla 6- Funciones Panel del principal de Control Eléctrico

SELECTOR	ACCIÓN	DETALLE
Selector 1	ON-OFF	Tanque de Mezcla con capacidad de 5000 L
Selector 2	ON-OFF	Tanque de Mezcla con capacidad de 10000 L
Selector 3	ON-OFF	Ozono
Selector 4	ON-OFF	Bomba de 110V
Selector 5	ON-OFF	Lámparas
Selector 6	ON-OFF	Bomba de Agua

Elaborado por: El Investigador

El esquema eléctrico de cada selector es independiente, pero cada uno de ellos tiene la misma estructura eléctrica, es decir el circuito de fuerza está conformado por el Breaker, seguido de un contactor y un relé térmico y finalmente de un motor, (Figura 23) y tiene la siguiente nomenclatura:

- **L1**= Línea 1
- **L2**= Línea 2
- **L3**= Línea 3
- **PE**= Línea a Tierra
- **K1**= Contactor 1
- **RT**= Relé Térmico
- **M1**= Motor 1

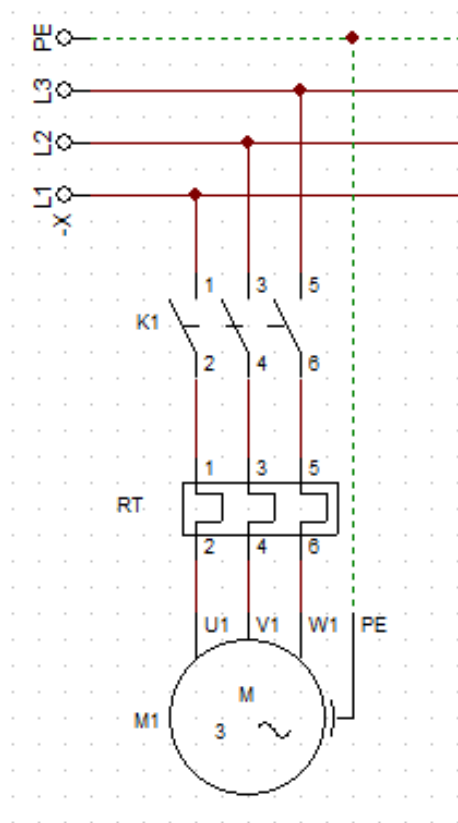


Figura - 23- Circuito de Fuerza del Panel de Control Principal

Elaborado por: El Investigador

El circuito de control está conformado de manera independiente para cada selector, pero todos tienen la misma estructura (Figura 24) con la siguiente nomenclatura:

- **L1**= Línea 1
- **SEL**= Selector de dos Posiciones
- **K1**= Contactor 1
- **LPV**= Luz Piloto Color Verde
- **RT**= Relé Térmico
- **LPR**= Luz Piloto Color Roja
- **L2** = Línea 2

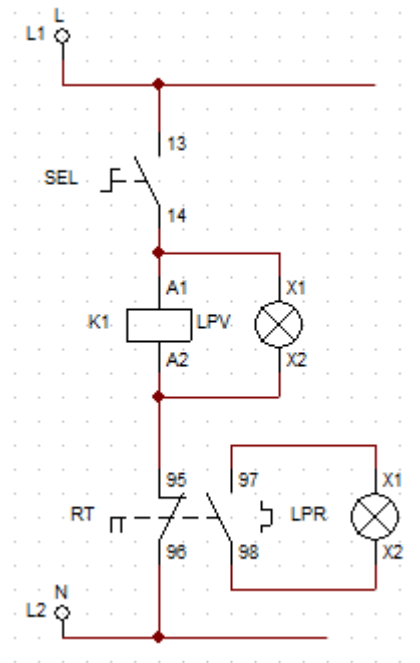


Figura - 24- Circuito de Control del Panel de Control Principal

Elaborado por: El Investigador

En la Figura 25, observamos las conexiones internas del panel de control, donde claramente se puede visualizar un espacio para implementar algún otro tipo de circuito que se podría necesitar o acoplar a un futuro.



Figura - 25- Conexiones Internas del panel de control

Fuente: El Investigador

Panel de Control de la Bomba de Alcohol

Este panel es independiente al panel de control principal, porque el almacenamiento del alcohol se encuentra en la parte exterior de la fábrica, tiene la misma estructura eléctrica del panel de control principal, con la diferencia de su accionamiento, que se lo realiza por medio de pulsadores. El circuito de fuerza está formado por la misma estructura eléctrica del panel principal (ver Figura 23). Y el circuito de control contiene dos pulsadores (Figura 26), con la siguiente nomenclatura.

- **L1**= Línea 1
- **P_ON**= Pulsador NA
- **P_OFF**= Pulsador NO
- **K1**= Contactor 1
- **LPV**= Luz Piloto Color Verde
- **RT**= Relé Térmico
- **LPR**= Luz Piloto Color Roja
- **N** = Neutro

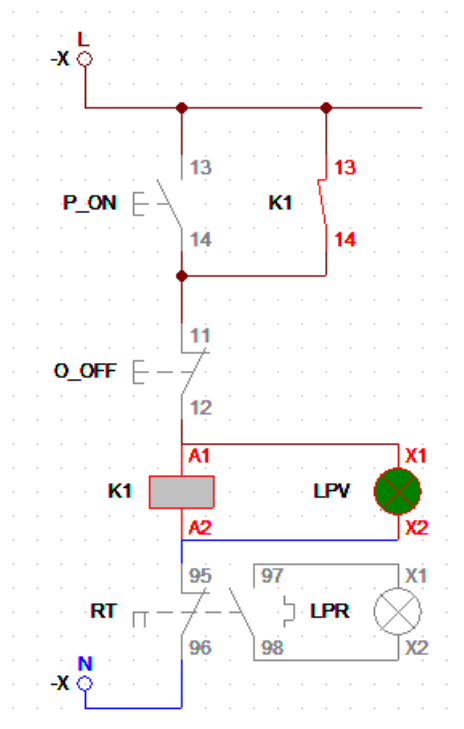


Figura - 26- Diagrama de Control - Bomba de Alcohol

Elaborado por: El Investigador

Todos los elementos antes vistos intervienen en el proceso de dosificación, por lo que utilizaremos estos mismos elementos para realizar un control inteligente y menos invasivo para no afectar la garantía de las instalaciones y sobre todo la producción diaria de la empresa, este control es basado en el internet industrial de las cosas.

3.1.2.8. Selección de los elementos a utilizar para el control y la automatización.

Análisis de Sensores. - A continuación, realizamos la comparación de diferentes tipos de sensores que se puede utilizar para el control de la automatización, para esto se toma en cuenta que se va a controlar la cantidad de líquido que va a pasar desde los tanques de agua y alcohol hacia los tanques de mezcla.

Inicialmente se estudió la posibilidad de utilizar caudalímetros electromagnéticos, para medir la cantidad de líquido que pasa por las tuberías hacia los tanques de dosificación, pero como se mencionó anteriormente en la planta se hicieron algunos cambios en el mes de diciembre, la estructura de la tubería tiene una garantía, y abrir para colocar los caudalímetros, se pierde la garantía, por esta razón esta opción fue descartada.

Otra opción para saber la cantidad de líquido que está ingresando a los tanques de mezcla, son Electrodo de Nivel de Agua, estos contienen hasta 5 electrodos de varilla/suspensión, una vez que el electrodo se sumerge en un líquido conductivo, fluye una corriente alterna baja y el amplificador de conmutación (relé de electrodo) evalúa esta corriente alterna si se supera el umbral de desencadenamiento ajustable, se activará el relé. Esta opción también fue descartada porque el sensor contiene máximo 5 electrodos, es decir 5 medidas fijas lo que no ayuda a un control exacto del flujo del agua.

Se estudio la mejor opción para no intervenir en la estructura de la tubería y tampoco la colocación de varios sensores dentro del tanque porque al ingresar a colocar los sensores se puede ocasionar una contaminación al producto final de la empresa. Al analizar toda la situación se determinó utilizar un sensor que sea menos invasivo para el proceso de dosificación actual, pero que sea efectivo para el control de flujo, el sensor que cumplía con estos requisitos fue el sensor ultrasónico, que se podía colocar en la parte superior del tanque sin interrumpir la producción diaria de la empresa.

A continuación, en la tabla 7, vemos los tipos de sensores ultrasónicos que se puede utilizar para nuestro proyecto.

Tabla 7- Sensores Ultrasónicos

Sensores Ultrasonidos	Sensor HCSR04	Anti-Agua Jsn-sr04t V3	SUP-MP
Imagen	Figura 27	Figura 29	Figura 28
Rango de medición	2 cm a 400 cm	10 cm a 500 cm	0 ~ 30 m
Precisión	+/- 3 mm	± 0.5 cm	0.3%F.S
Voltaje de Operación	5V DC	5V DC	12-24 V CC
Ángulo de apertura	15°	15°	<0.4-1.8m
Corriente de Trabajo	15 mA	30 mA	4 ~ 20mA RL < 500Ω
Frecuencia	40KHz	40KHz	20~350KHz
A prueba de Agua	NO	SI	SI
Disponibilidad	SI	SI	NO
Costo	\$4,50	\$22	\$ 167

Elaborado por: El Investigador



Figura - 27- Sensor Ultrasónico Sensor HCSR04 [28]



Figura - 28- Sensor Ultrasónico SUP-MP[29]

Al evaluar la Tabla 7 se selecciona para el proyecto el sensor Anti-Agua Jsn-sr04t V3 (Figura 29), porque este sensor soporta un ambiente expuesto a la humedad y líquidos, además de tenerlo disponible en el país.



Figura - 29- Sensor Ultrasónico Anti agua[30]

Selección de Componentes

Módulo de Relé

Es un dispositivo electromagnético que actúa como un interruptor, controlado por un circuito eléctrico en el que un conjunto de uno o más contactos es activado por una bobina y un electroimán, permitiendo abrir o cerrar el circuito eléctrico independiente.



Figura - 30- Modulo de Relé

Módulo NodeMCU ESP8266

Es una plataforma completa de desarrollo, módulo SoC (system on chip), basado en el microcontrolador Tensílica L106 de 32 bits, funcionando a 80 Mhz, con una memoria Flash de 4MB, y 32 KB de SRAM. Tiene conectividad Wifi, estándar 802.11 b/g/n, para la implementación de dispositivos operando como servidores de internet (web server). El NodeMCU integra el protocolo TCP-IP y tiene conectividad Wifi, con antena de radio frecuencia integrada, lo cual permite su conexión a internet a través de un router cercano al módulo, con la posibilidad de control remoto del tipo "Anywhere", desde cualquier parte. Esto significa que, ya sea a través de un teléfono celular ó una computadora conectados a internet, se pueden enviar comandos al NodeMCU para activar dispositivos y recibir información de sensores o alarmas instaladas en casas y oficinas. [31] Este dispositivo se utiliza para comunicación inalámbrica entre los sensores ultrasónicos y la ESP32.

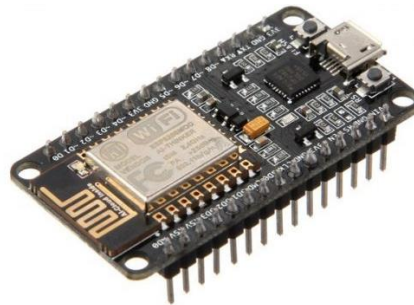


Figura - 31- Modulo Esp8266[31]

Módulo De Interfaz De Visualización Tm1638 Para Arduino

Este módulo de pantalla de 8 dígitos y siete segmentos utiliza un controlador TM6138 que permite un control total de la pantalla con solo 3 pines digitales en un microcontrolador. Además de la pantalla de siete segmentos, hay 8 LED de 3 mm controlables individualmente y un teclado con 8 botones dispuestos en una sola fila. Estos también se pueden controlar a través del IC TM6138 y, por lo tanto, no requieren pines digitales adicionales. [32] Este módulo se utiliza para que sea el HMI, de todo el Sistema de Dosificación Inteligente, que va ir conectado a la ESP32.

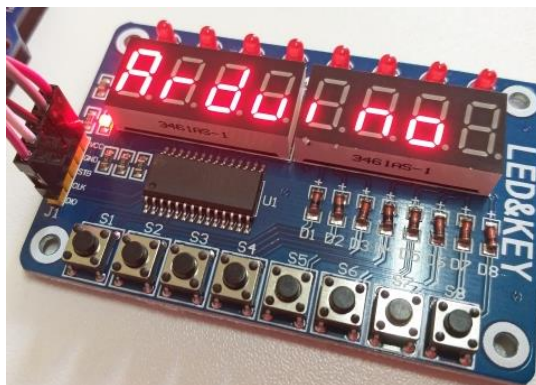


Figura - 32- Módulo Tm1638 para Arduino[32]

3.1.2.9. Selección software de diseño del control y la automatización

Con los elementos y dispositivos seleccionados, se decide escoger como software de programación a Arduino, porque es una plataforma de desarrollo libre que integra un microcontrolador programable y es compatible su entorno de desarrollo, esto facilita el establecimiento de comunicación entre todos los microcontroladores, para el control y automatización de varios sensores y actuadores. El PLC seleccionado al poseer un microcontrolador tipo Cortex M3, es compatible con el entorno de Arduino, por lo que el proyecto va a tener la programación del software de Arduino IDE (Entorno de desarrollo integrado).



Figura - 33- Entorno de Desarrollo Arduino[33]

3.1.2.10. Diseño del control según los componentes seleccionados.

Para el diseño del control se basa en la arquitectura IoT de tres capas que se referencia en el modelo IBM (ver Figura 33), que separa las aplicaciones en tres niveles informáticos lógicos y físicos, es la arquitectura de software predominante para las aplicaciones cliente-servidor tradicionales. Está conformada por la capa de dispositivos, la capa de procesamiento Fog y la capa Nube.

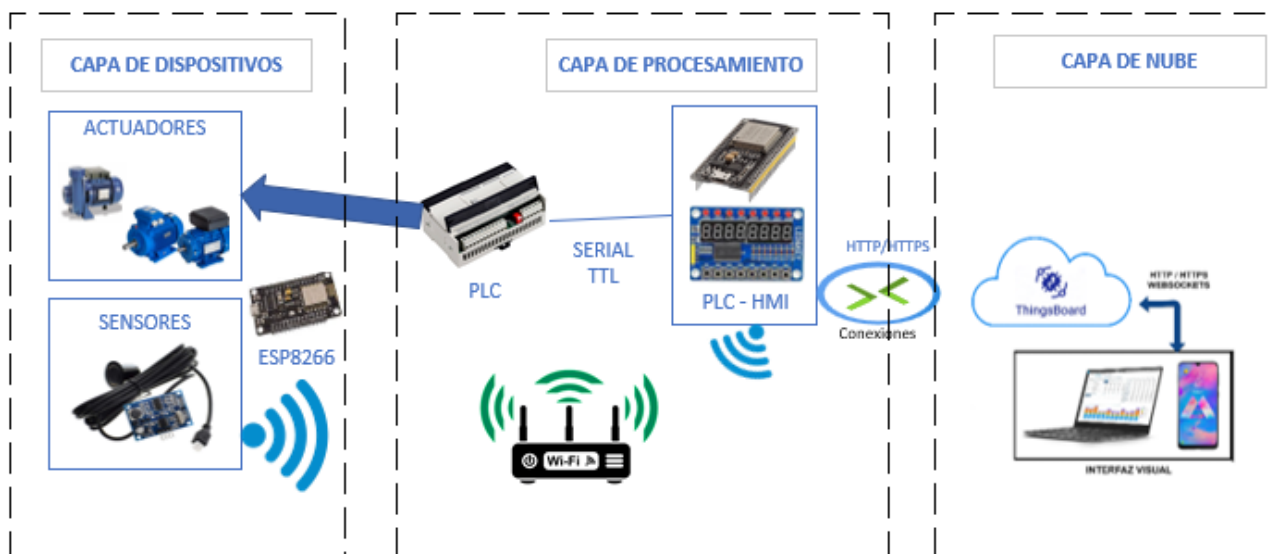


Figura - 34-Diseño de Control de la Dosificación

Elaborado por: El Investigador

Capa De Dispositivos. Esta capa está conformada por los sensores ultrasónicos, y las bombas de agua y alcohol conjuntamente con los motores de los tanques de mezcla. Los dos sensores ultrasónicos se van a comunicar de manera inalámbrica por medio del protocolo HTTP, gracias a los módulos ESP8266 que se colocan en cada tanque.

Capa de Procesamiento. - Esta capa está conformada el PLC, el microcontrolador ESP32, el HMI y la comunicación con el ESP8266. Se da con la comunicación entre el PLC y el ESP32 esta es una comunicación serial TTL para el procesamiento de los todos datos adquiridos y se estable la comunicación con la Nube.

Capa de la Nube. - Es donde se adquiere una Infraestructura tipo SaaS es un tipo de cloud computing, este es un modelo de distribución de software donde el soporte lógico y los respectivos datos que maneja se alojan en los servidores de un proveedor, cuyo acceso es a través de internet, donde se puede interactuar, y controlar a través de una interfaz gráfica.

Para una mayor referencia del diseño de control del proyecto, se visualiza la Figura 34, que ya incluye todos los elementos seleccionados en el proyecto.

3.1.2.11. Desarrollo de la programación para el procesamiento de las variables

Para la descripción del funcionamiento de cada uno de los dispositivos se utiliza diagramas de flujo.

Diagrama de Flujo del ESP32

En la Figura 35 se visualiza un diagrama de flujo general, para posteriormente detallar las subrutinas que contiene cada una de ellas.

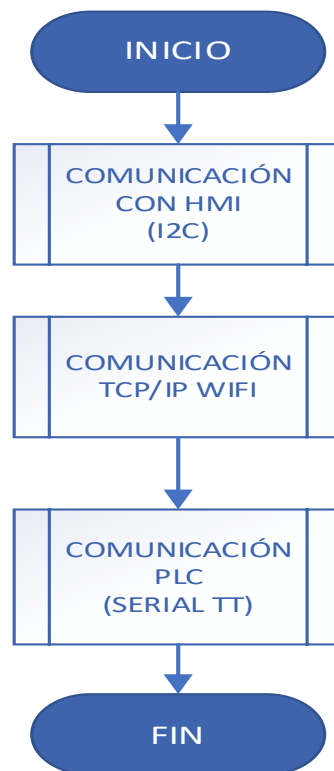


Figura - 35- Diagrama de flujo General de la ESP32

Elaborado por: El Investigador

Subrutina de la comunicación con el HMI

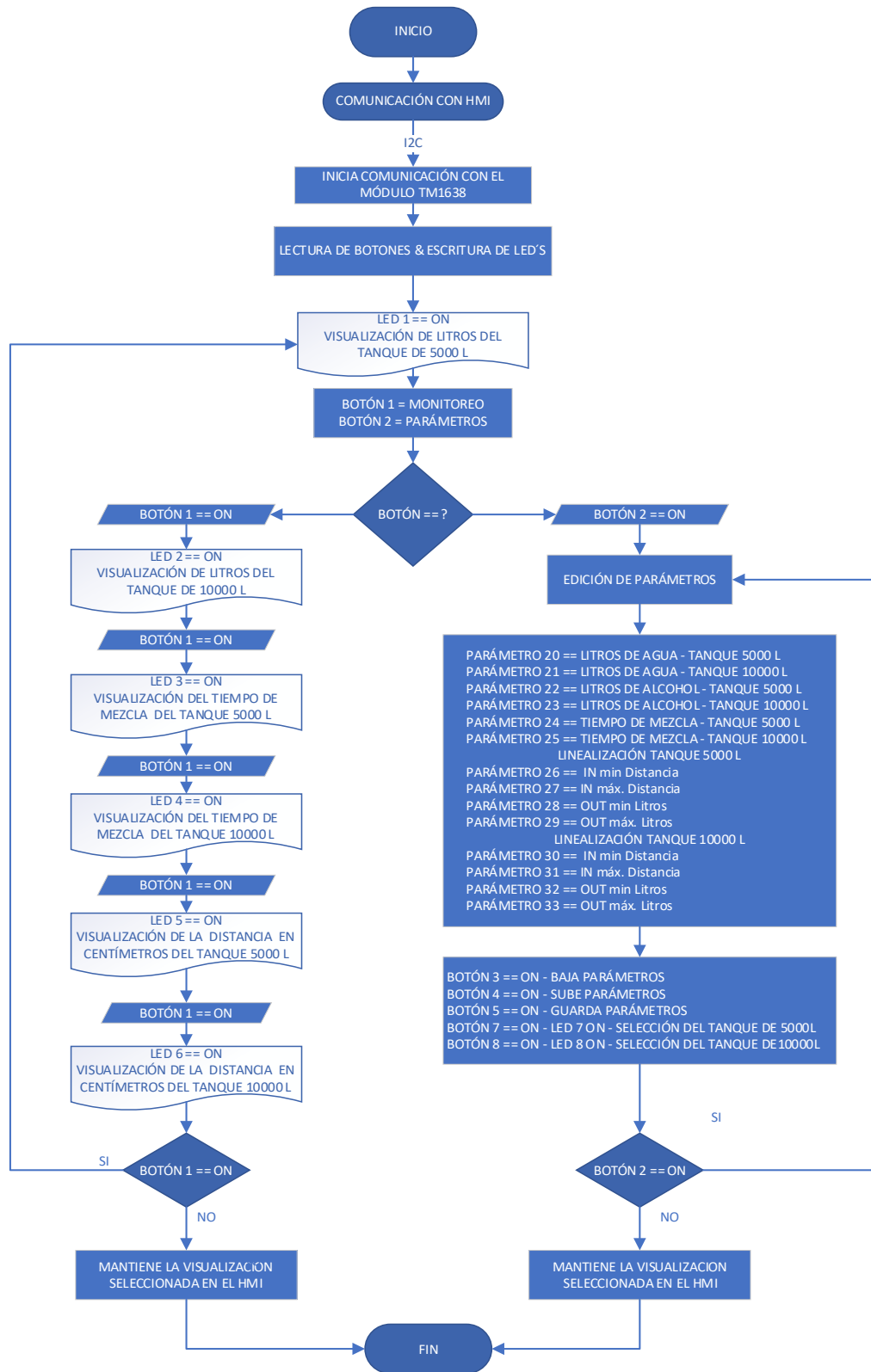


Figura - 36- Subrutina de la comunicacion con HMI
Elaborado por: El Investigador

Subrutina de la comunicación TCP/IP

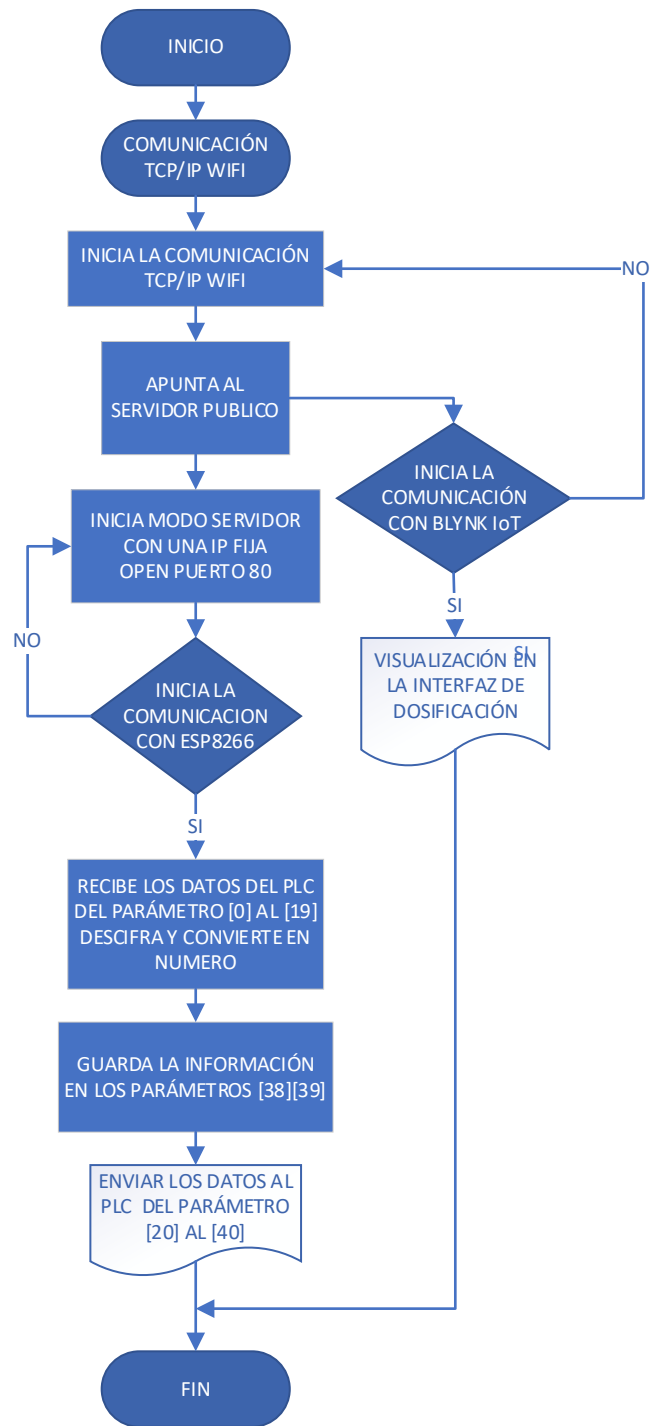


Figura - 37- Subrutina de la comunicación con TCP/IP

Elaborado por: El Investigador

Subrutina de la comunicación con el PLC

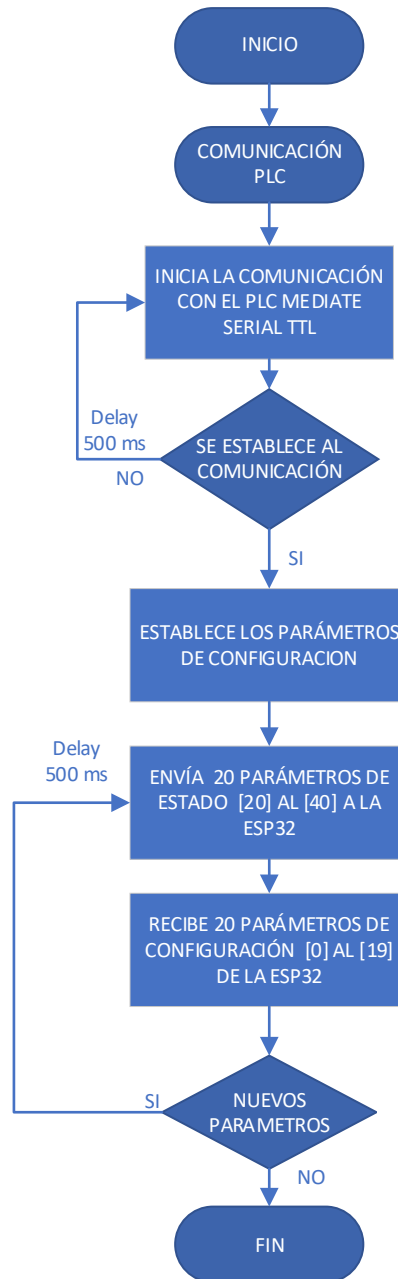


Figura - 38- Subrutina de la comunicación con PLC

Elaborado por: El Investigador

Diagrama de flujo de la programación del PLC

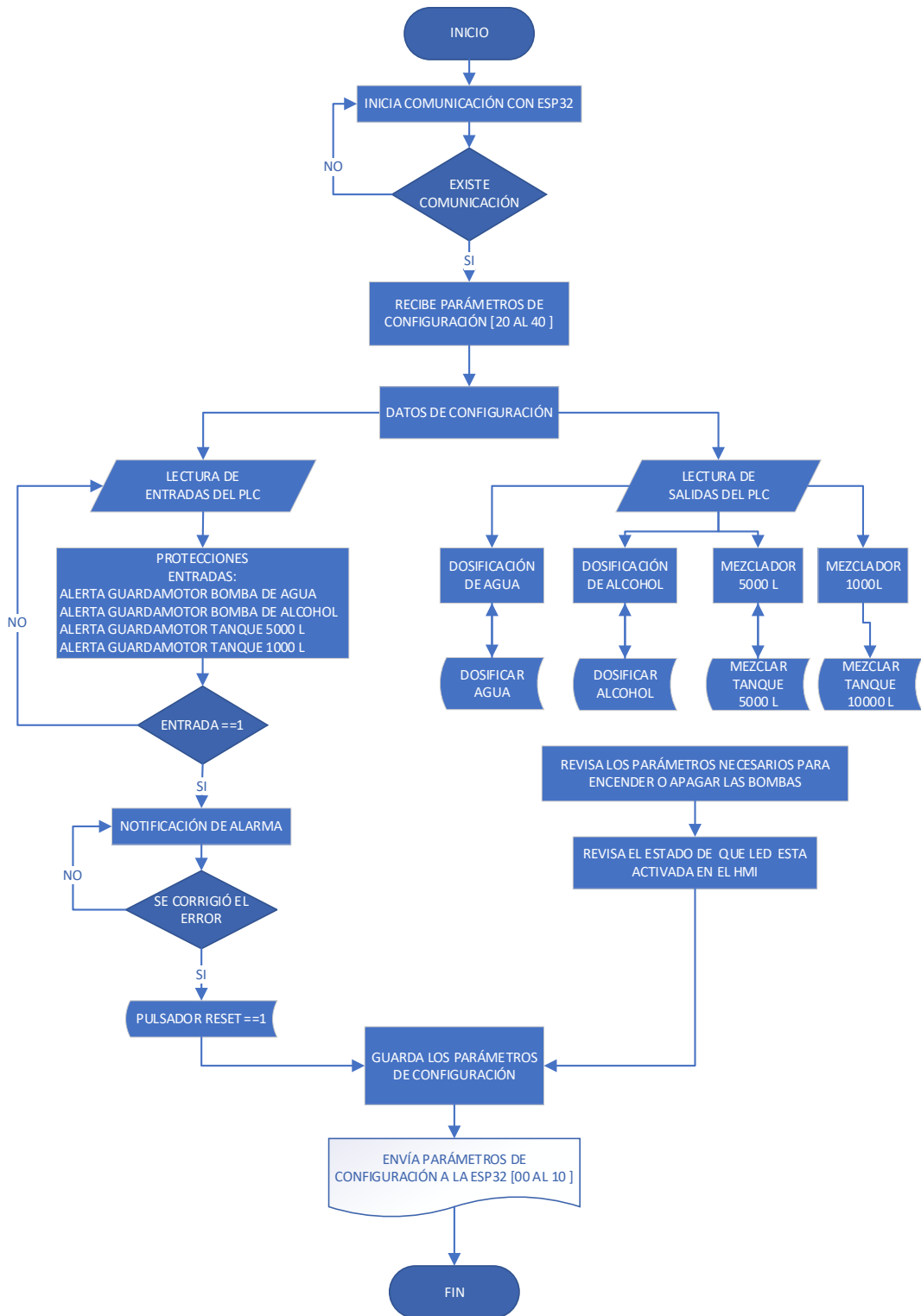


Figura - 39 - Diagrama de Flujo de la programación del PLC

Elaborado por: El Investigador

Diagrama de Flujo de ESP8266

Los dos sensores ultrasónicos en la programación, para diferenciar que medida corresponde a cada sensor, la etiqueta del tanque de 5000 L es “Dist1” y la etiqueta para el sensor del tanque de 10000 L es “Dist2”, a continuación, en la Figura 40 a la estructura de la programación de la Esp8266, del tanque de 5000 L puesto que, para el tanque del 10000 L, es la misma con la diferencia de la etiqueta final.

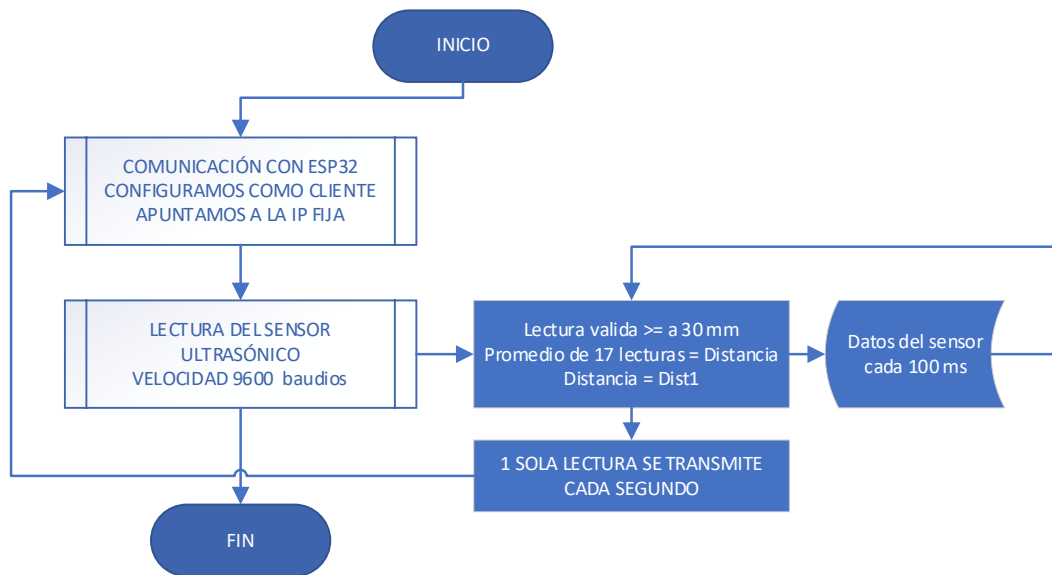


Figura - 40- Diagrama de Flujo de la programación de ESP8266

Elaborado por: El Investigador

3.1.2.12. Elaboración de la interfaz gráfica, visualización de datos, y monitoreo remoto

Para la interfaz gráfica se utiliza la plataforma Blynk que es un servicio en la nube, esta plataforma con acceso a iOS y Android se utiliza para el control de Arduino, Raspberry Pi, ESP8266 y las apps que funcionan a través de Internet, donde el usuario puede crear proyectos IoT desde el celular o computadora con una conexión de red.

Como el proyecto es para una empresa, se creó un nuevo correo electrónico para poder administrar y brindar esos datos a los operarios de la fábrica. A continuación, se muestra la creación de una cuenta en la plataforma Blynk IoT.

1. Se ingresa a la página web: <https://blynk.io/>
2. Se crea una cuenta ingresando el correo electrónico

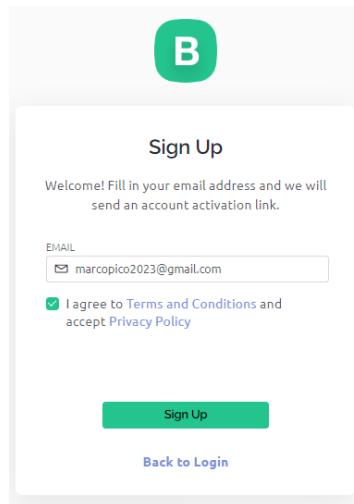


Figura - 41- Creación de la Cuenta en Blynk

Elaborado por: El Investigador

3. Se envía un enlace al correo registrado donde se procede a registrar la contraseña.
4. Ingresamos a la cuenta, con el correo y contraseña registrado

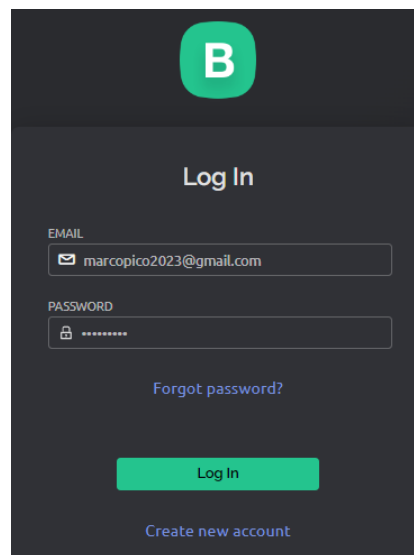
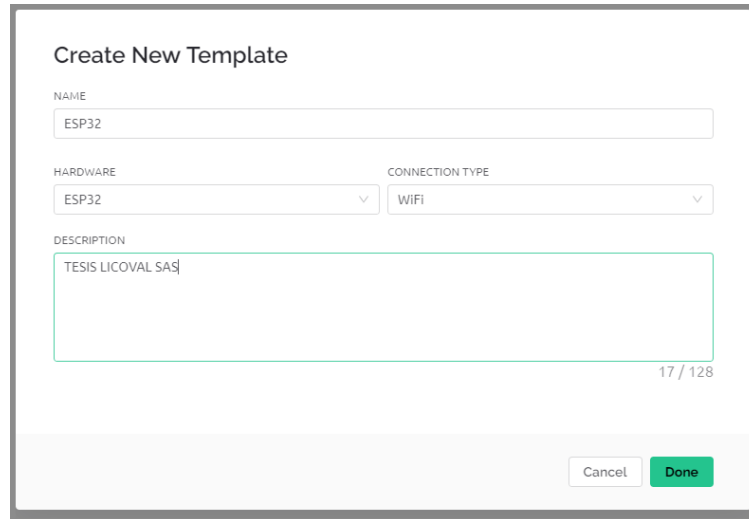


Figura - 42-Ingreso a la cuenta Blynk

Elaborado por: El Investigador

5. Ya en cuenta, se procede a crear una plantilla, donde se le asigna un nombre a la plantilla, se selecciona que tipo de hardware a utilizar, como es la ESP32, y se selecciona el tipo de comunicación que se enlaza con la nube, como es el WIFI.



The screenshot shows a 'Create New Template' form with the following fields and values:

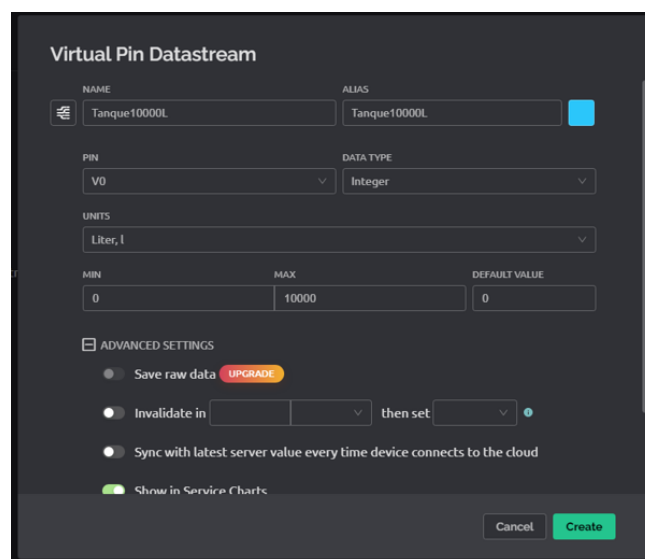
- NAME: ESP32
- HARDWARE: ESP32
- CONNECTION TYPE: WIFI
- DESCRIPTION: TESIS LICOVAL SAS

At the bottom right, there are 'Cancel' and 'Done' buttons. A character count '17 / 128' is visible at the bottom right of the description field.

Figura - 43- Creación de una plantilla

Elaborado por: El Investigador

6. Se ingresa a la pestaña de datastreams y se crea los dispositivos que conectamos, donde se define el nombre, el alias, cual es el pin a cuál va a conectar, el tipo de dato que se va a ingresar y el valor mínimo y máximo para la representación gráfica ver Figura 44, creamos 12 dispositivos que se van a manejar de manera virtual.



The screenshot shows a 'Virtual Pin Datastream' configuration form with the following fields and values:

- NAME: Tanque10000L
- ALIAS: Tanque10000L
- PIN: V0
- DATA TYPE: Integer
- UNITS: Liter, l
- MIN: 0
- MAX: 10000
- DEFAULT VALUE: 0

Below the main fields, there are 'ADVANCED SETTINGS' with the following options:

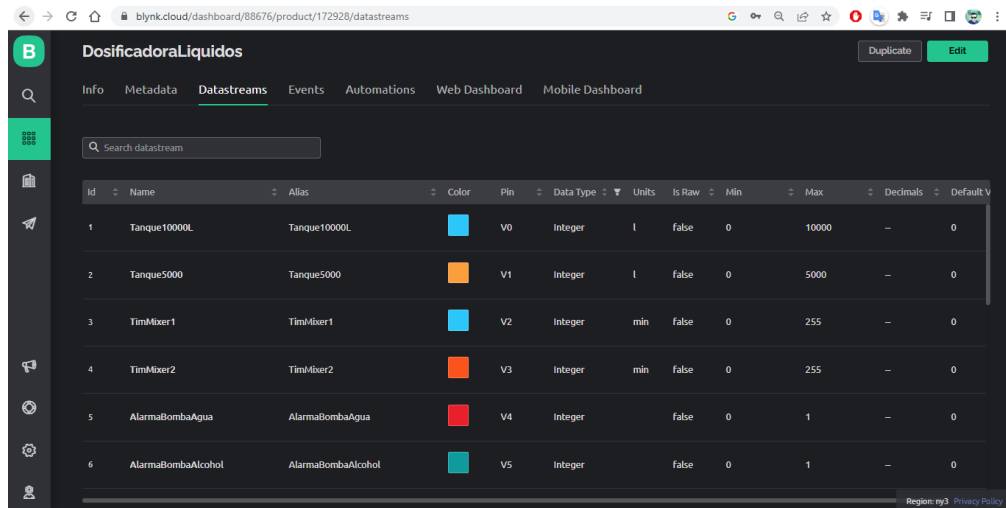
- Save raw data (UPGRADE)
- Invalidate in [] then set []
- Sync with latest server value every time device connects to the cloud
- Show in Service Charts

At the bottom right, there are 'Cancel' and 'Create' buttons.

Figura - 44- Creación de los Pines Virtuales

Elaborado por: El Investigador

Creación de los Datastreams que se van a representar, Figura 45



The screenshot shows the Blynk Cloud dashboard for a project named 'DosificadoraLiquidos'. The 'Datastreams' tab is active, displaying a table with the following data:

Id	Name	Alias	Color	Pin	Data Type	Units	Is Raw	Min	Max	Decimals	Default V
1	Tanque10000L	Tanque10000L	Blue	V0	Integer	l	false	0	10000	-	0
2	Tanque5000	Tanque5000	Orange	V1	Integer	l	false	0	5000	-	0
3	TimMixer1	TimMixer1	Blue	V2	Integer	min	false	0	255	-	0
4	TimMixer2	TimMixer2	Orange	V3	Integer	min	false	0	255	-	0
5	AlarmaBombaAgua	AlarmaBombaAgua	Red	V4	Integer		false	0	1	-	0
6	AlarmaBombaAlcohol	AlarmaBombaAlcohol	Teal	V5	Integer		false	0	1	-	0

Figura - 45- Lista de Datastreams

Elaborado por: El Investigador

7. Luego en la pestaña “Web Dashboard” se configura la interfaz de visualización de los dispositivos finales, para esto se añade Widget Box, que previamente tiene la plataforma, se crea y configura cada uno de los widget, con el nombre que la vamos a dar, enlazamos con el Datastream, y se configura la forma visual del widget como se observa en figura 46.

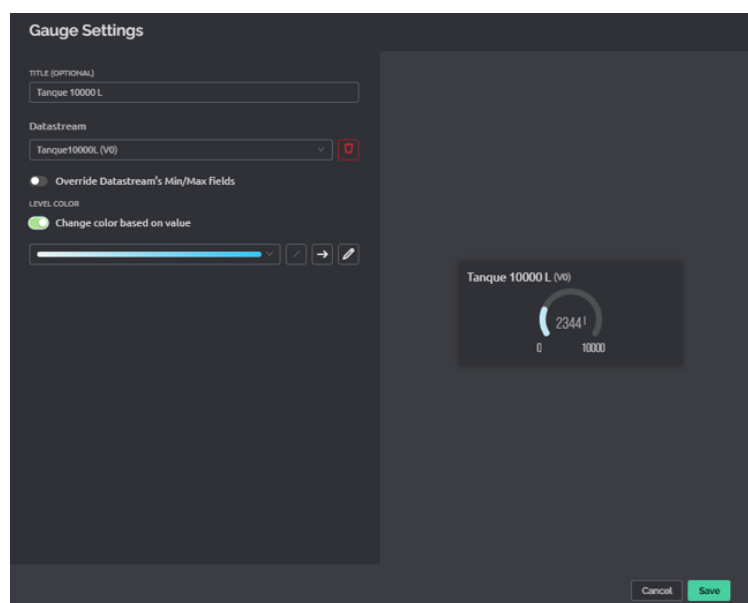


Figura - 46- Configuración del Widget

Elaborado por: El Investigador

8. Luego se procede a colocar todos los widgets en el Dashboard para visualizar y observar de manera ordenada la información.

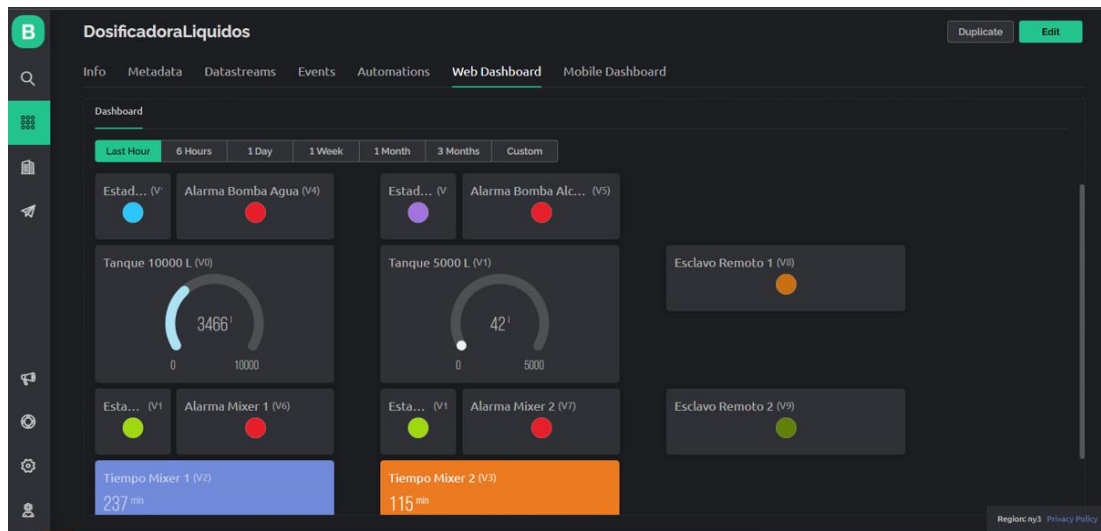


Figura - 47- Web Dashboard

Elaborado por: El Investigador

9. En la pestaña de Mobile Dashboard, se establece que descarguemos la aplicación Blynk IoT.App como se observa en la Figura 48.

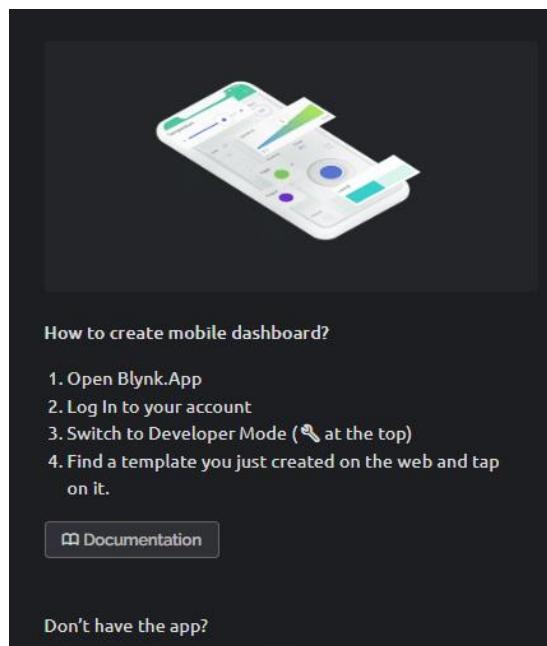


Figura - 48- Pestaña Mobile Dashboard

Elaborado por: El Investigador

10. Para descargar la aplicación se selecciona la pestaña de Búsqueda, donde se crea o añade un nuevo dispositivo, y se agrega el dispositivo por plantilla, puesto que anteriormente ya creamos una plantilla ver en la Figura 49.

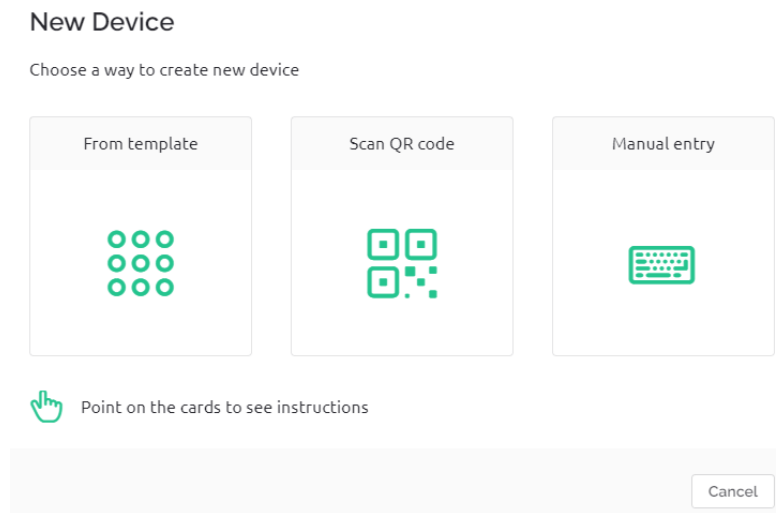


Figura - 49- Creación de nuevo dispositivo

Elaborado por: El Investigador

11. Creación del dispositivo

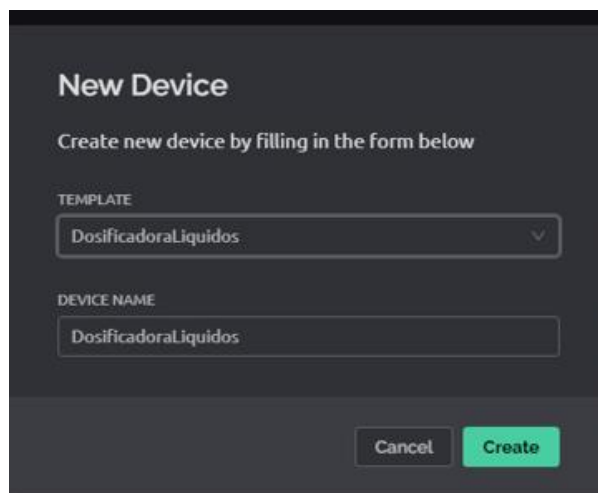


Figura - 50- Creación de Dispositivo

Elaborado por: El Investigador

12. Luego de crear el dispositivo se despliega El ID de plantilla, el nombre del dispositivo y el token de autenticación deben declararse en la parte superior del código de firmware.

```
FIRMWARE CONFIGURATION

#define BLYNK_TEMPLATE_ID "TMPLGDR1"
#define BLYNK_DEVICE_NAME "DosificadoraLiquidos"
#define BLYNK_AUTH_TOKEN "wmHowfnTgIhZoIq83Y92oQPci"

Template ID, Device Name, and AuthToken should be declared at the
very top of the firmware code.
```

Figura - 51 - Token de Seguridad para colocar en ESP32

Elaborado por: El Investigador

13. Este código desplegado se procede a colocar en la programación de nuestra ESP32 para lograr la comunicación entre la plataforma Blynk y el microcontrolador.

```
#define BLYNK_PRINT Serial

/* Complete su ID de plantilla (solo si usa Blynk.Cloud) */
#define BLYNK_TEMPLATE_ID "TMPLGDR1[REDACTED]"
#define BLYNK_DEVICE_NAME "DosificadoraLiquidos"
#define BLYNK_AUTH_TOKEN "wmHowfnTgIhZoIq83Y92oQPci[REDACTED]"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
```

Figura - 52- Programación en la ESP32 para la comunicación entre la plataforma Blynk y la ESP32

Elaborado por: El Investigador

14. Para la comprobación del programa, se descarga la aplicación que antes se mencionó Blynk IoT. Se descargo la aplicación y se inicia sesión con las mismas credenciales que anteriormente creadas, de forma automática encontramos la plantilla como se observa en la Figura 53.

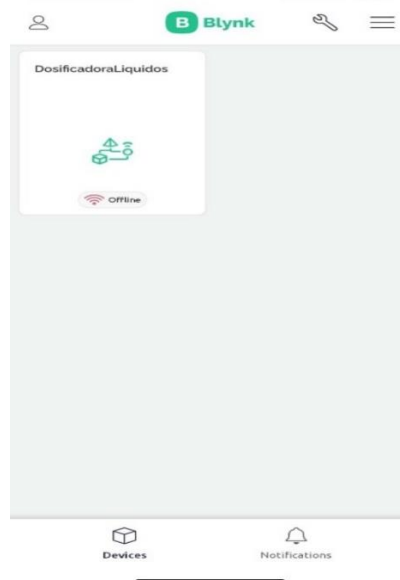


Figura - 53- App Blynk IoT

Elaborado por: El Investigador

Al igual que el dashboard de la Web, se procede a configurar el dashboard del celular, para tener una mejor visualización del terminar móvil.

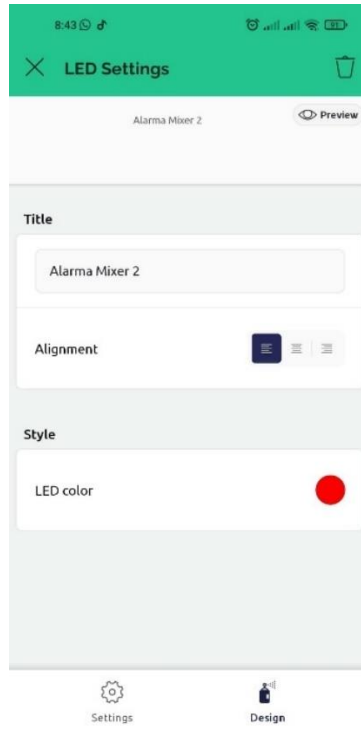


Figura - 54- Configuración de Dashboard de la App Blynk IoT

Elaborado por: El Investigador

Una vez realizadas todas las configuraciones y adecuaciones del interfaz del dashboard general de la App se visualiza en la Figura 55.

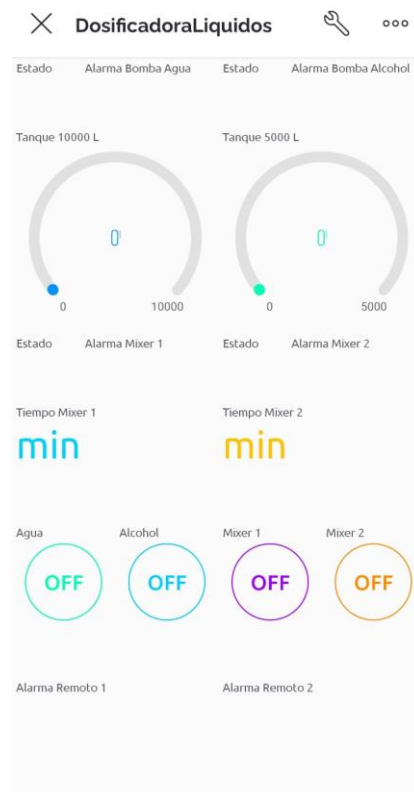


Figura - 55- Dashboard General de la App

Elaborado por: El Investigador

Reportes de Datos Históricos

Esta plataforma tiene varios planes de pago, (ver Figura 56), la versión gratuita ofrece el almacenamiento histórico de datos del dispositivo por una semana, la base datos de la plataforma da reportes de las acciones y eventos que se han realizado. Para poder ver esos reportes, en la cuenta web se descarga en una carpeta comprimida que contiene un archivo .csv que abrir con el programa Microsoft Excel y ver el historial, Se procedió a contratar el Plan Plus, con un costo final de \$6,99 al mes, este plan brinda la facilidad de sacar reportes de forma mensual, almacena los datos por un periodo de 3 meses, con estos reportes históricos la empresa podrá tomar decisiones para mejorar su producción y corregir posibles errores en este proceso.

Billing

You are using Free plan



Plan Comparison

Monthly Yearly (Save ~ 20%)

Free	Plus	PRO	Business
\$0/month	\$6.99/month	\$49/month	\$599/month
	Devices 10 20	Devices 40 100 500	Web Only One App Two Apps

Figura - 56- Planes de la Plataforma Blynk

Elaborado por: El Investigador

Una vez realizado el pago, en nuestro dashboard Web y se procede a descargar los reportes de los eventos que ha tenido nuestra plataforma en un periodo de tres meses Figura 57. Se descarga es una carpeta comprimida, (Figura 59), este archivo contiene dos archivos uno con extensión .txt y otro archivo con la extensión .csv que podemos abrir y verificar los reportes mediante el programa Microsoft Excel, ver Figura 60.

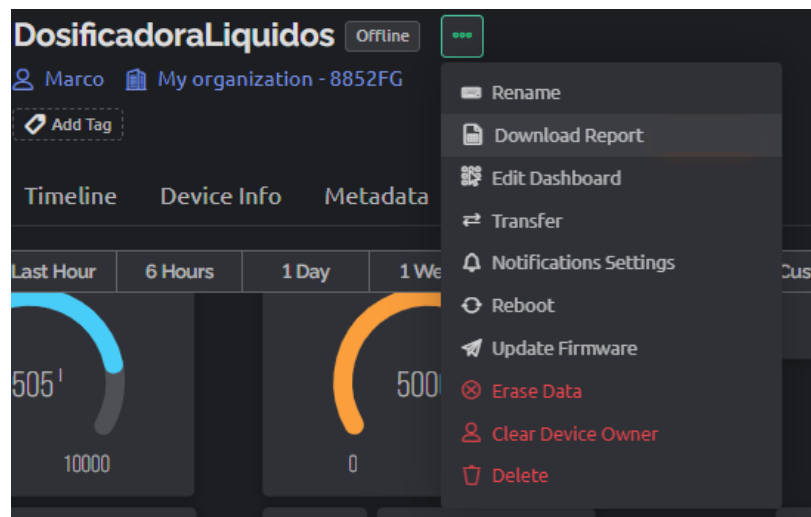


Figura - 57- Descargar Reportes Históricos

Elaborado por: El Investigador

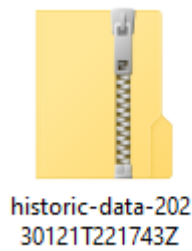


Figura - 58 Carpeta comprimida de datos históricos

Elaborado por: El Investigador

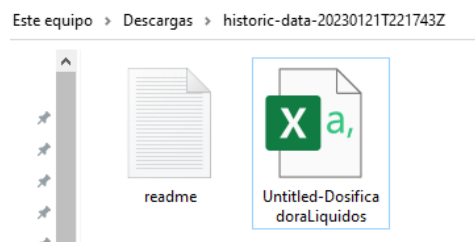


Figura - 59- Archivos con Datos Históricos de la Dosificadora

Elaborado por: El Investigador

Notificaciones de Estado Automáticas

Otra función que ofrece esta plataforma es alertar sobre los eventos que pasa en el proceso a nivel de hardware, para esto se realiza click en la pestaña de “Templates” que es donde se encuentra la plantilla y se selecciona en “Events” ver Figura 61, para poder configurar el tipo de alertas que se necesita para el proyecto.

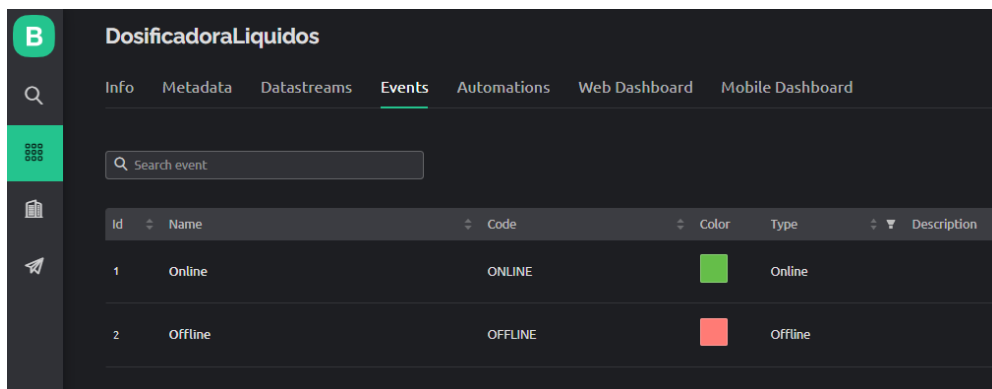


Figura - 60- Pestaña de Eventos

Elaborado por: El Investigador

Se procede a crear un nuevo evento, donde se designa el nombre del evento y automáticamente se crea un código para ese evento, y se elige qué tipo de alerta es ya sea de información, advertencia o un evento crítico, se puede dar los límites del evento como: cada que tiempo se activará el mensaje el evento, cuantas veces se envía el evento al usuario. Se puede activar que el evento pasé a la pestaña de notificaciones en la aplicación del terminal móvil, Activar para que el evento se envíe a la línea de tiempo y también se puede aplicar una etiqueta al evento, ver Figura 62.

The image shows a mobile application interface for creating a new event. The main screen is titled 'DosificadoraLiquidos' and has a sidebar with various icons. A modal window titled 'Add New Event' is open, showing a form with two tabs: 'General' and 'Notifications'. The 'General' tab is active and contains the following fields: 'EVENT NAME' with the value 'Mixer 1 ON', 'EVENT CODE' with the value 'mixer_1_on', a 'TYPE' section with radio buttons for 'Info' (selected), 'Warning', 'Critical', and 'Content', a 'DESCRIPTION (OPTIONAL)' text area with the value 'Activación del tanque de Mezcla de 5000 L', and a 'Limit' section with a frequency input set to '1' and a dropdown menu set to '1 second'. There is also a checkbox labeled 'Send event to Notifications tab' which is checked. At the bottom right of the modal, there are 'Cancel' and 'Create' buttons.

Figura - 61- Creación del nuevo evento - Notificaciones

Elaborado por: El Investigador

Adicional, se selecciona la pestaña donde la plataforma permite enviar notificaciones vía correo electrónico y configurar con una alerta de sonido, si la alerta es crítica, esta se configuro al tener activa la alarma del relé térmico Figura 63.

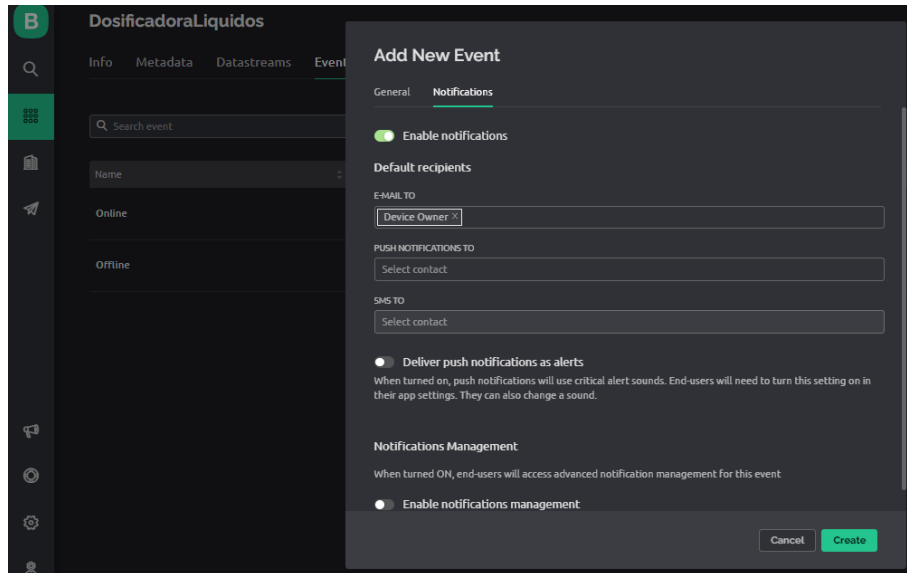


Figura - 62- Configuración de Notificaciones Vía Correo Electrónico

Elaborado por: El Investigador

La configuración del servidor, en este caso la Esp32 se agrega las líneas de código para que cuando el evento suceda, la plataforma blynk responda con la notificación ver Figura 64.

```
Serial.println("Notificaciones send");
SETBIT(PARAMETROS[9], 0, !READBIT(PARAMETROS[9], 0));
if (READBIT(PARAMETROS[9], 0) == True)
  Blynk.logEvent("bomba_agua_on", "Encendido Bomba de Agua");
if (READBIT(PARAMETROS[7], 0) == True)
  Blynk.logEvent("PROT_BOMBA_AGUA", "!ALERTA; Bomba de Agua");
if (READBIT(PARAMETROS[9], 1) == True)
  Blynk.logEvent("bomba_alcohol_on", "Encendido Bomba de Alcohol");
if (READBIT(PARAMETROS[7], 1) == True)
  Blynk.logEvent("PROT_BOMBA_ALCOHOL", "!ALERTA; Bomba de Alcohol");
if (READBIT(PARAMETROS[9], 2) == True)
  Blynk.logEvent("mixer_1_on", "Iniciado mexer 1");
if (READBIT(PARAMETROS[7], 2) == True)
  Blynk.logEvent("PROT_MIXER_1", "!ALERTA; mixer 1");
if (READBIT(PARAMETROS[9], 3) == True)
  Blynk.logEvent("mixer_2_on", "Iniciado mexer 2");
if (READBIT(PARAMETROS[7], 3) == True)
  Blynk.logEvent("PROT_MIXER_2", "!ALERTA; mixer 2");
if (READBIT(PARAMETROS[7], 4) == True)
  Blynk.logEvent("esclavo_1_offline", "!ALERTA; esclavo 1 OFFLINE");
if (READBIT(PARAMETROS[7], 5) == True)
  Blynk.logEvent("esclavo_2_offline", "!ALERTA; esclavo 2 OFFLINE");
```

Figura - 63- Programación del enlace de la comunicación con Blynk IoT

Elaborado por: El Investigador

Las notificaciones llegan de forma automática al celular y se guardan con una línea de tiempo donde se comprueba, los cambios que surgieron en el sistema (Figura 65).

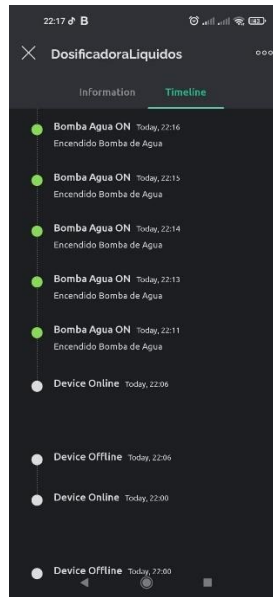


Figura - 64- Reporte de Notificaciones en la App

Elaborado por: El Investigador

En la siguiente Figura 66 se observar el encendido la bomba de agua, para ello se enciende el led verde y emite la notificación al celular.

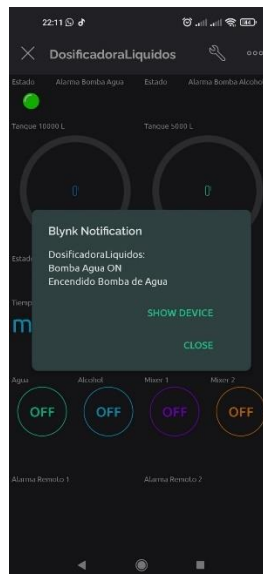


Figura - 65- Notificación al encender la bomba de Agua

Elaborado por: El Investigador

3.1.2.13. Implementación del sistema de dosificación inteligente.

Una vez que se comprobó el correcto funcionamiento del sistema de control inteligente de manera independiente, se procede a realizar la implementación en la fábrica, como primer paso fue instalar los sensores ultrasónicos en los tanques de mezcla.

Instalación de los Sensores Ultrasónicos

Estos sensores se proceden a instalar en la parte superior de los tanques de mezcla, con ayuda de unos orificios que previamente los dos tanques tenían (ver Figura 67), por donde salen los cables para conectar a su módulo de funcionamiento.



Figura - 66- Orificio para la colocación del sensor Ultrasónico - Tanque de 5000 Litros

Elaborado por : El Investigador

Para la colocación del módulo en el tanque disponemos de una caja apropiada para ambientes industriales, esta caja es ubicada y asegurada en la parte superior del tanque, como la comunicación de estos sensores es inalámbrica se realizó la instalación de una caja principal de control de energía eléctrica 110 V, que lleve energía hasta la ubicación de los dos tanques de mezcla, esta instalación se realizó con un interruptor que controle el encendido de los módulos (Figura 68).

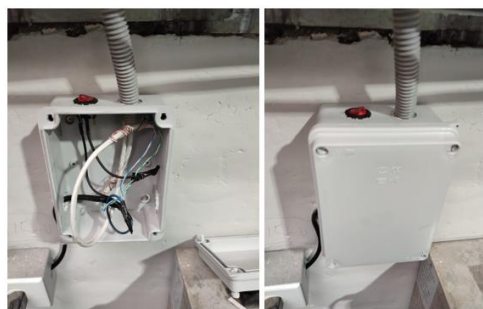


Figura - 67- Caja de Distribución Eléctrica para los sensores

Elaborado por: El Investigador

Como el módulo de la Esp8266 requiere un voltaje de alimentación de 5 Vcd, se coloca el módulo de relé con 5 pines y un convertidor de voltaje necesario para energizar el módulo, a continuación, todos estos dispositivos y con el módulo del sensor ultrasónico se colocó en la caja ubicada en el pate superior del tanque.



Figura - 68- Ubicación de la caja que contiene el módulo Esp8266 y el módulo de sensor ultrasónico

Elaborado por: El Investigador

La Figura 69, pertenece a la colocación del módulo en el tanque de 5000 Litros, para la colocación de los dispositivos en el tanque de 10000 Litros, observamos la siguiente Figura 70.



Figura - 69- Caja del módulo del sensor 10000 L

Elaborado por: El Investigador

Instalación del PLC,

El PLC se colocó en el interior de la caja principal del control eléctrico (Figura 12), puesto que disponía de espacio suficiente, también en su interior se situó los implementos necesarios para el control, como son los módulos de relé ver la Figura 71.

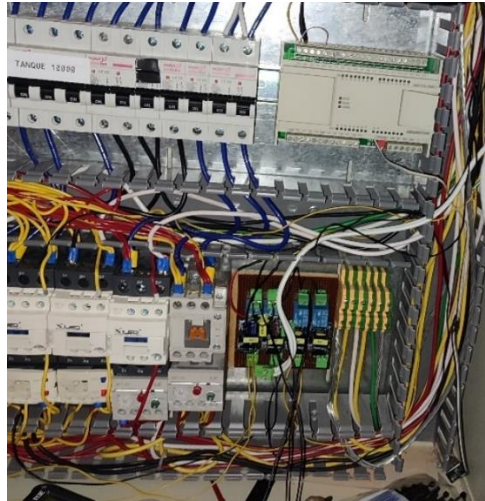


Figura - 70- Colocación del PLC en el panel de Control

Elaborado por: El Investigador

Se procede a conectar el módulo de relés con las entradas del PLC, y de la misma manera se coloca las salidas del PLC van conectados a la bobina del contactor para activar o desactivar los motores, hay que mencionar que el PLC contiene relés internos para este tipo de conexión, como se observa en la Figura 72.

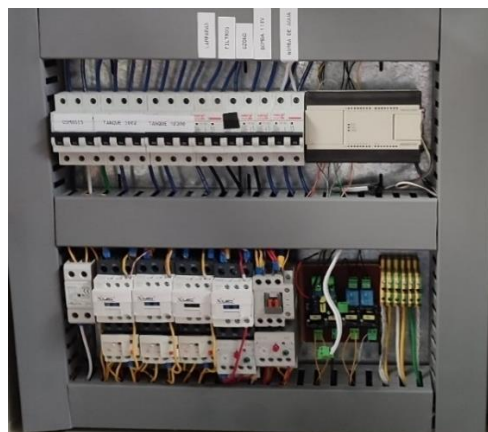


Figura - 71- Panel de Control con PLC y Relés

Elaborado por: El Investigador

La conexión física del PLC es de la siguiente manera, las protecciones de los relés térmicos, son conectados en paralelo en cada una de las alarma del relé térmico, de esta manera cuando se produzca una sobrecarga en alguna bomba o motor del mezclador la alarma se dispara, y activa al relé para que este mediante esa señal el PLC compara todas las salidas y determina si alguna esta activa para enviar el mensaje de sobrecarga, esta conexión observamos en la Figura 73 donde en la parte inferior de la figura se observa las entradas de cada uno de los cuatro relés y por la parte superior van conectados a las salidas del PLC.

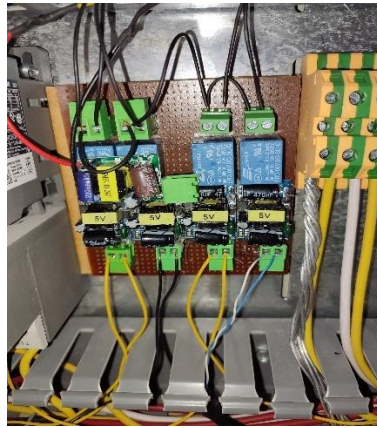


Figura - 72- Conexiones de las protecciones a los módulos de relé

Elaborado por: El Investigador

En la parte inferior del PLC se observa las salidas del PLC que están conectadas a las salidas de los módulos de relés, y por la parte superior las entradas del PLC donde se conecta directamente a la entrada de la bobina del contactor, de esa manera cuando se envía una señal, esta bobina se activa porque el PLC posee relés internos, como se observa en la Figura 74.

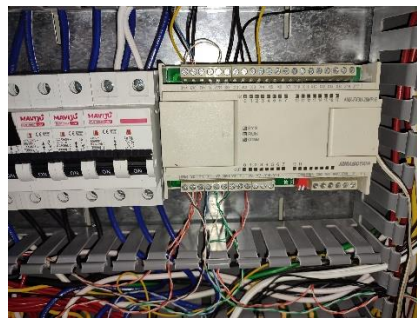


Figura - 73- Conexiones del PLC

Elaborado por: El Investigador

Instalación del HMI

Como se menciona anteriormente, para la interface física, se utiliza el módulo TM1638 (Figura 30), para esto se adquirió una caja para poder fijar este dispositivo en el tablero principal de control. A esta caja se realiza unas adecuaciones para visualizar solo los leds, los displays y poder accionar los botones. Para esto se realizó un corte mediante láser como se observa en la figura 75, adecuando de mejor manera el módulo con la placa.



Figura - 74- Corte con láser de la caja para el módulo Tm1638

Elaborado por: El Investigador

Una vez cortado se obtiene una mejor visualización estética de la caja y del dispositivo, esta caja va a ser fijada en la parte lateral derecha del panel principal de control, para poder interactuar de mejor manera con el usuario, ver en la Figura 76.

En la parte superior de la caja para el HMI, se colocó un selector de dos posiciones que va a determinar los modos a trabajar del PLC, ya sea en modo manual o automático. Si el selector está en modo manual, el control de activar y desactivar las bombas y motores es de los selectores y si está el modo automático entrará en funcionamiento el sistema de dosificación inteligente.



Figura - 75- HMI en nuestro tablero de control

Elaborado por: El Investigador

Bomba de Alcohol

Para el control de la bomba de alcohol, de manera física, se instaló un sistema de conmutación alado de la caja principal, esto es porque inicialmente se contempló realizar la comunicación con esta bomba de manera inalámbrica pero después de las pruebas, la comunicación no era buena por lo que se llevó las señal de forma física como podemos observar en la Figura 77.

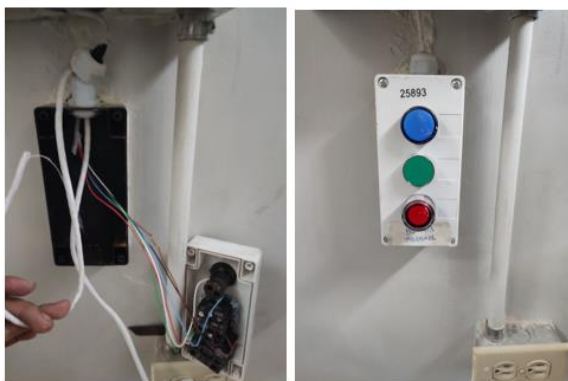


Figura - 76- Conmutación de control de la Bomba de Alcohol cerca del panel principal

Elaborado por: El Investigador

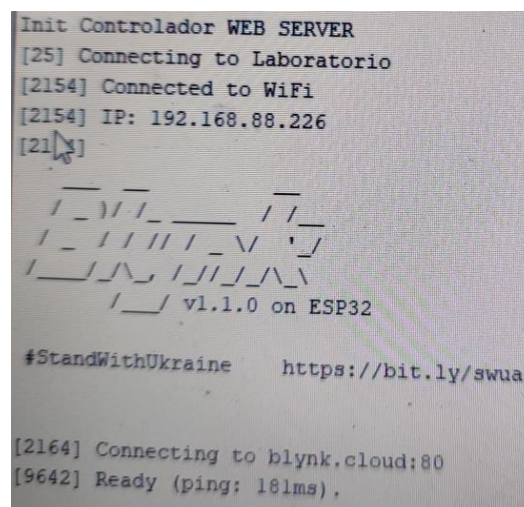
Una vez ubicado este sistema de conmutación cerca del panel de control, se procede a sacar las señales del circuito, la señal del relé térmico, la señal para activar y desactivar la bobina de encendido y apagado del contactor.

3.1.2.14. Desarrollo de las pruebas de funcionamiento.

Las pruebas de funcionamiento se realizaron a lo largo de todo el proyecto para que funcione de la mejor manera el sistema de dosificación inteligente, gracias a esto se procedió a realizar los cambios en el control que inicialmente no se contempló, uno de ellos son los sensores ultrasónicos, que no transmiten los datos a más de 6 metros por su cable, inicialmente se plateó que esta señal llegara a una caja ubicada en la parte central de los dos tanques de mezcla, por esta razón se modificó el proyecto a una comunicación inalámbrica.

Otra prueba que se dio mientras se desarrollaba el proyecto, es la comunicación inalámbrica hacia la caja de control de la bomba de alcohol, por la distancia que se encontraba la comunicación era inestable, por lo que se decidió realizar una comunicación física para no tener inconvenientes.

Una vez que se corrigieron todos los problemas antes mencionados que se dieron al instalar el sistema, se puso en funcionamiento. Se realizaron dosificaciones durante tres semanas de manera continua con el promedio de una preparación diaria es decir una sola vez en el día, dependiendo la demanda de producción de la empresa, dependiendo las cantidades de alcohol y de agua que ingresaron al tanque de mezcla fueron los programados inicialmente y se comprobó el accionamiento de los motores de mezcla por el tiempo programado.



```
Init Controlador WEB SERVER
[25] Connecting to Laboratorio
[2154] Connected to WiFi
[2154] IP: 192.168.88.226
[21] 
  _ _ _ _ _ 
 / _ \ / _ \ / _ \ / _ \
/_ _ \/_ _ \/_ _ \/_ _ \
  _ _ _ _ _ 
    / _ \ v1.1.0 on ESP32

#StandWithUkraine https://bit.ly/swua

[2164] Connecting to blynk.cloud:80
[9642] Ready (ping: 181ms).
```

Figura - 77- Comunicación del sistema con nuestra Plataforma Blynk

Elaborado por: El Investigador



Figura - 78- Cantidad de litros en el tanque de 10000 L

Elaborado por: El Investigador



Figura - 79- Configuración del Parámetro 21 - cantidad de litros a ingresar en el tanque de 10000 L

Elaborado por: El Investigador

Para una mejor visualización e interacción del HMI, se realizó etiquetas de información y se colocó cerca de cada botón, además de la información que corresponde cuando algún led este encendido, como se observa en la Figura 80.



Figura - 80- Etiquetas de las funciones del HMI

Elaborado por: El Investigador

En la App, se muestra la cantidad de litros de líquido que se encuentran en el tanque y el estado de las bombas y motores (Figura 81), que es una visualización en tiempo real.



Figura - 81- App en tiempo real de la Dosificación

Elaborado por: El Investigador

3.2.1.1 Resultados

Con las pruebas realizadas, se puede decir que el sistema inteligente tiene una efectividad del 100%, al dosificar la cantidad de agua y alcohol en proporciones exactas, evitando desperdicios que solían aparecer cuando se realizaba la medición mediante un control visual a los tanques, esto resulta un mejor manejo de la materia prima.

Se apaga las bombas de manera automática al momento que el sensor detecte que la cantidad de agua o alcohol es la programada, evitando la presencia de un operador que antes del sistema inteligente realizaba esta función, mejorando la productividad de la empresa al ocuparse de otras situaciones del operario.

Una vez incluidos todos los ingredientes del producto en los tanques de mezcla, se programa el tiempo de mezclado, esto mejora la eficiencia de la empresa al tener

encendido los motores el tiempo exacto y necesario para la mezcla, dando así, un ahorro de energía eléctrica, una vez cumplido el tiempo se apaga los motores de forma automática, además, de no requerir un operario para que realice esta función, como era antes de implementar el sistema.

3.2.1.2 Presupuesto del Proyecto

Para el análisis del presupuesto económico del proyecto se describe de manera individual todos los elementos y cantidades utilizadas para realizar este sistema de dosificación inteligente como se observa en la tabla 8.

Tabla 8- Presupuesto Económico

PRESUPUESTO			
Cantidad	Detalle	Valor Unitario	Valor Total
1	Caja para 10*10 para Control eléctrico de sensores	\$ 6,00	\$ 6,00
1	Interruptor para control Eléctrico	\$ 1,00	\$ 1,00
2	Sensor Ultrasónico	\$ 22,00	\$ 44,00
2	Relé 5 Pines SPDT JQC-3FF 5VDC-1ZS	\$ 3,00	\$ 6,00
2	Cajas para módulos del Ultrasónico	\$ 3,80	\$ 7,60
4	Convertidores de voltaje a 5 Vcd	\$ 4,00	\$ 16,00
2	ESP 8266	\$ 12,30	\$ 24,60
1	metro de manguera corrugada	\$ 1,00	\$ 1,00
1	PLC	\$ 200,00	\$ 200,00
1	ESP32 + Base para conexión	\$ 35,00	\$ 35,00
1	Modulo TM1638	\$ 9,50	\$ 9,50
1	Caja para HMI + Selector	\$ 20,00	\$ 20,00
30	metros de Cable UTP CAT 6	\$ 0,66	\$ 19,80
5	Pulsadores	\$ 0,50	\$ 2,50
1	Botones y luces Piloto + caja	\$ 11,00	\$ 11,00
1	Módulos de Relé	\$ 15,00	\$ 15,00
1	Material Eléctrico	\$ 20,00	\$ 20,00
1	Herramientas	\$ 60,00	\$ 60,00
1	Alimentación	\$ 35,00	\$ 35,00
1	Movilización	\$ 60,00	\$ 60,00
		TOTAL	\$ 594,00

Elaborado por: El investigador

El costo total de la implementación de sistema de dosificación inteligente es de \$594,00 dólares.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Al analizar el sistema actual de dosificación se puede decir que el sistema de control es completamente manual, porque requiere de un operador que acciona un selector, el cual enciende las bombas de agua y alcohol, y mediante un control visual de la cantidad de líquido, que se va a procesar en los tanques, se procede a apaga las bombas; y de igual manera los motores que producen la mezcla de ingredientes, el operario enciende y apaga los motores después de cierto tiempo.

- La infraestructura que tiene la empresa fue la base para poder determinar los elementos y dispositivos necesarios para el control IIoT. En los circuitos de control y de fuerza que tiene la empresa, no se realizaron cambios, se intervino en el panel principal de control, siendo menos invasivo en la estructura eléctrica. Se utilizó los sensores ultrasónicos anti-agua Jsn-sr04t V3 para medir la cantidad de líquido en los tanques de mezcla, un PLC Amsamotion con certificaciones industriales que contiene un microcontrolador que permite una fácil y rápida comunicación con una ESP32 que funciona como servidor y se encargada de procesar los datos, determinando las acciones que va a realizar dependiendo la lectura de los sensores.

- Se diseña un sistema inteligente que se basa en la arquitectura IoT de IBM que describe 3 capas, la primera capa de dispositivos, que comprende los sensores ultrasónicos y actuadores como las bombas de agua y alcohol, además de los motores de mezcla, la segunda capa es la de procesamiento, donde se adquiere los datos emitidos por los sensores y transmitidos de forma inalámbrica por la ESP8266, que procesa y determina que acción debe realizar dependiendo la lectura de los sensores, se establece la comunicación entre el PLC y el ESP32, esta es una comunicación serial TTL para el procesamiento de los datos. En esta capa se elabora un HMI, con el módulo de display TM6138 para la interacción directa con el operador. La última capa es de la Nube cuya función es ver el estado en tiempo real del proceso, ayudándonos de una interfaz gráfica que interactúa directamente en el proceso.

- Se desarrollo la interfaz de visualización en la plataforma Blynk,IoT donde se representa el estado de las bombas de agua y alcohol, además de los tiempos de mezcla de los tanques, esta visualización es en tiempo real mediante unos gadgets que se colocan en el dashboard, esta plataforma mediante su aplicación móvil permite alertar sobre los eventos que pasa en el proceso a nivel de hardware, creando notificaciones de estado de manera automática al terminal móvil, este evento puede ser de información, advertencia o una emergencia, el historial de datos se guardan durante tres meses, el cual se puede descargar como una carpeta comprimida que contiene archivos de tipo .texto y un archivo de tipo .csv, que se visualizan en el software Microsoft Excel.

4.3. Recomendaciones

- Para un mejor control de la cantidad de líquido de dosificación se deber adquirir caudalímetros y ubicarlos en las tuberías de esa manera controlar el flujo de agua y alcohol que llegan a los tanques de mezcla.
- Implementar un sistema de electroválvulas para la apertura y cierre de las tuberías en el caudal, de esa manera se podrá direccionar de mejor manera a que tanque de mezcla se va a realizar la dosificación.
- Para futuras automatizaciones se puede implementar de manera paulatina sistemas inteligentes de control en otras etapas de la producción que posee la fábrica, y a su vez realizar un sistema de control central para toda empresa.

Referencias Bibliográficas

- [1] R. Ramón, P. Villalón, I. U. Ponce Monarez, y A. S. Avila, “Integración de tecnologías asociadas con el internet de las cosas y la industria 4.0”, *THEOREMA*, vol. 18, pp. 1–8, 2022, Consultado: el 1 de marzo de 2023. [En línea]. Disponible en: <http://cathi.uacj.mx/handle/20.500.11961/24470>
- [2] O. Álvarez, “Análisis de la industria 4.0 como factor diferenciador del sector industrial del Ecuador”, 2021, UNIVERSIDAD CENTRAL DEL ECUADOR, Quito. [En línea]. Disponible en: <https://ciencialatina.org/index.php/cienciala/article/view/533>

- [3] G. Barona López y L. E. Velasteguí, “Automatización de procesos industriales mediante Industria 4.0”, *AlfaPublicaciones*, vol. 3, núm. 3.1, pp. 84–101, ago. 2021, doi: 10.33262/ap.v3i3.1.80.
- [4] A. Sergio, “Sistema de control y monitoreo hídrico, basado enLoRaWAN™, para el acueducto principal de la Asociación Administradora del Acueducto Rural de Playa Sámara de Nicoya”, Nicoya, jun. 2021. Consultado: el 7 de diciembre de 2022. [En línea]. Disponible en: <https://repositoriotec.tec.ac.cr/handle/2238/13234>
- [5] I. Benito, “SISTEMA DE LOCALIZACIÓN EN TIEMPO REAL APLICADO A LA INDUSTRIA 4.0”, 2021.
- [6] E. C. Soliz y E. D. Gomez, “IMPLEMENTACIÓN DEL SISTEMA INTERNET DE LAS COSAS, PARA EL MÓDULO DE CONTROL DE NIVEL Y CAUDAL, EN EL LABORATORIO DE CONTROL Y MANIPULACIÓN AUTOMÁTICA DE LA ESCUELA DE INGENIERÍA DE MANTENIMIENTO”, Riobamba, mar. 2019. Consultado: el 7 de diciembre de 2022. [En línea]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/11572/1/25T00356.pdf>
- [7] P. A. Quinteros, M. Calero Zurita, N. C. Zambrano, y E. Lapo Manchay, “Automatización de los procesos industriales * Automation of industrial processes”, *Journal of Business and entrepreneurial Julio-Diciembre*, vol. 4, doi: 10.37956/jbes.v4i2.82.
- [8] Secmotic, “¿Qué es el IoT Industrial?”, el 14 de febrero de 2020. <https://secmotic.com/que-es-el-iot-industrial/#gref> (consultado el 11 de diciembre de 2022).
- [9] A. Ing. Carrera, “INDUSTRIA 4.0 EN EL ECUADOR”, *FUNDACIONESFG*, el 11 de marzo de 2022. <https://sanfranciscoglobal.org/industria-4-0-en-el-ecuador/> (consultado el 11 de diciembre de 2022).
- [10] Emilio. García Moreno y Universidad Politécnica de Valencia., *Automatización de procesos industriales : robótica y automática*. Universidad Politécnica de Valencia, 1999.

- [11] K. Rose, S. Eldridge, y L. Chapin, “LA INTERNET DE LAS COSAS”, 2015.
- [12] H. Boyes, B. Hallaq, J. Cunningham, y T. Watson, “The industrial internet of things (IIoT): An analysis framework”, *Comput Ind*, vol. 101, pp. 1–12, oct. 2018, doi: 10.1016/j.compind.2018.04.015.
- [13] J. Luis y D. V. Román, “CONFERENCIA DE DIRECTORES Y DECANOS DE INGENIERÍA INFORMÁTICA Industria 4.0: la transformación digital de la industria”, 2016.
- [14] F. Rozo-García, “Revisión de las tecnologías presentes en la industria 4.0”, *Revista UIS Ingenierías*, vol. 19, núm. 2, pp. 177–191, may 2020, doi: 10.18273/revuin.v19n2-2020019.
- [15] C. Tecnológica *et al.*, “El Entorno de la Industria 4.0: Implicaciones y Perspectivas Futuras”, 2017, [En línea]. Disponible en: <https://www.redalyc.org/articulo.oa?id=94454631006>
- [16] J. Leber, “General Electric’s San Ramon Software Center Takes Shape MIT Technology”, 2012. : <http://www.technologyreview.com/news/507831/general-electric-pitches-an-industrial-internet/> (consultado el 9 de enero de 2023).
- [17] D. Floyer, “Defining and Sizing the Industrial Internet”, el 27 de junio de 2013. http://wikibon.org/wiki/v/Defining_and_Sizing_the_Industrial_Internet. (consultado el 9 de enero de 2023).
- [18] ge.com, “Industrial Internet Insights Report for 2015”, 2015. <https://www.ge.com/digital/sites/default/files/industrial-internet-insights-report.pdf> (consultado el 9 de enero de 2023).
- [19] mcr.es, “La dosificación industrial”, el 17 de septiembre de 2019. <https://mcr.es/la-dosificacion-industrial/> (consultado el 9 de enero de 2023).
- [20] mx.omega.com, “Introducción a la medición de nivel”, 2017. <https://mx.omega.com/prodinfo/medicion-de-nivel.html> (consultado el 10 de enero de 2023).
- [21] Creative Commons Attribution-Share Alike 3.0, “CAUDALIMETRO”.

- [22] Instituto Geologico y Minero de España, “Bomba de Agua”, 2018.
- [23] Ingeniería de Sistemas Industriales, *Controladores Lógicos Programables (PLCs)*.
- [24] R. Kunst, L. Avila, A. Binotto, E. Pignaton, S. Bampi, y J. Rochol, “Improving devices communication in Industry 4.0 wireless networks”, *Eng Appl Artif Intell*, vol. 83, pp. 1–12, ago. 2019, doi: 10.1016/j.engappai.2019.04.014.
- [25] A. Mon, H. René, y D. Giorgio, “Modelo de evaluación de Tecnologías de la Información y la Comunicación para la industria 4.0”, 2019. Consultado: el 23 de enero de 2023. [En línea]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/90359>
- [26] MITSUBISHI ELECTRIC, “Controladores lógicos programables”, el 21 de enero de 2009. <https://www.melcsa.com/public/frontend/images/products/AUMIPLC004/Manual%20de%20Introduccion%20FX3U.pdf> (consultado el 23 de enero de 2023).
- [27] CLAUDIA AGUIRRE, “TARJETAS DE DESARROLLO Y MCU”, el 30 de abril de 2020. <https://uelectronics.com/cual-tarjeta-de-desarrollo-elegir/> (consultado el 23 de enero de 2023).
- [28] “Elecbee - Ultrasonico”, 2023. https://www.elecbee.com/en-25082-5Pcs-HY-SRF05-Ultrasonic-Distance-Sensor-Module-Measuring-Sensor-Module?utm_term=&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=9958698819&hsa_cam=18719854073&hsa_grp=140153146062&hsa_ad=630917925115&hsa_src=g&hsa_tgt=pla-1186173532455&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQiAw8OeBhCeARIsAGxWtUwb7TXr3CneJM1P5HDfwmXXtfI4PYvMq_ybRQYeUWSwTQ6xTtY01qAaAmTaEALw_wcB (consultado el 24 de enero de 2023).
- [29] Supmea, “Medidor de nivel ultrasónico SUP-MP”, 2023. <https://es.supmeaauto.com/ultrasonic-level-meter/mp-a-ultrasonic-level-meter?ga=x1z14g77&keyword=sensor%20nivel%20ultrasonico&gclid=Cj0K>

CQiAw8OeBhCeARIsAGxWtUzwGu7zw-
MmnVhoBVNRqtnFyloLmNoM6A7krpGkTtHzyaaG7CpNrwEaAlHqEALw
_wcB (consultado el 24 de enero de 2023).

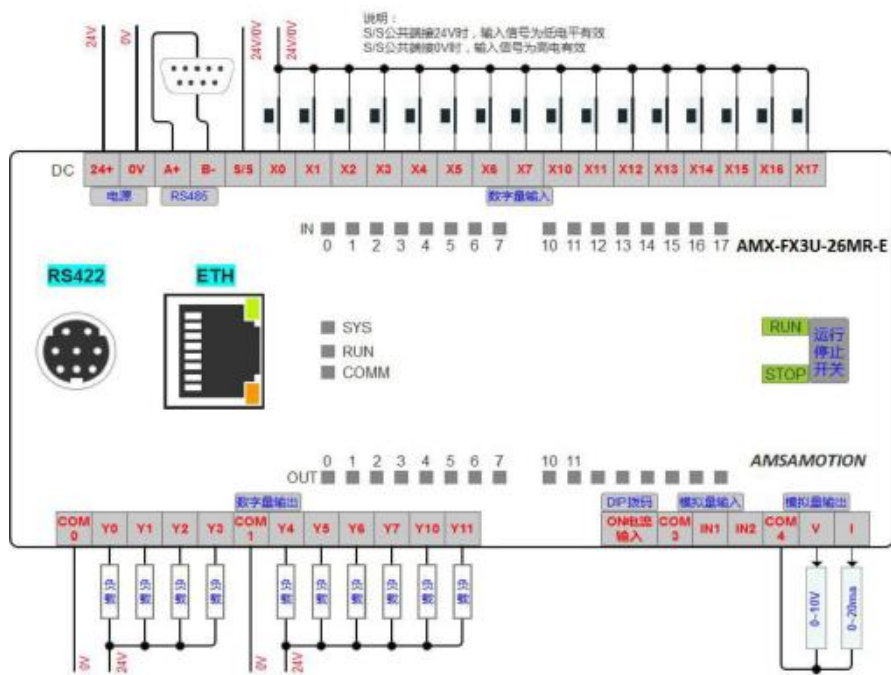
- [30] “Sensor Ultrasonico Contra El Agua”, 2023. <https://uelectronics.com/producto/sensor-ultrasonico-jns-sr04t/> (consultado el 24 de enero de 2023).
- [31] Punto Flotante S.A, “MANUAL-BASICO-NODEMCU-ESP-12E-ARDUINO”, 2022, Consultado: el 24 de enero de 2023. [En línea]. Disponible en: <https://www.puntofloante.net/MANUAL-BASICO-NODEMCU-ESP-12E-ARDUINO.pdf>
- [32] “Handson Technology User Guide TM1638 7 Segment Display Keypad & LED Module”. [En línea]. Disponible en: www.handsontec.com
- [33] Arduino.cc, “Arduino Mega 2560”, 2023. <https://store.arduino.cc/products/arduino-mega-2560-rev3> (consultado el 24 de enero de 2023).

Anexos

Anexo1 – DataSheet PLC

Model number	AMX-FX3U-26MR-E	AMX-FX3U-26MT-E
Digital input		
Number of input points	16	
Input signal type	Switch contact signal or level signal, support positive and negative trigger	
Input signal voltage	DC 18 ~ 28V	
Insulation Loop	Optocoupler isolation	
Digital output		
Number of output points	10	
Output type	Relay	NPN transistor
Output Capacity	2A/point; 8A/4 point; 12A/6 point	0.5 A/point; 2A/4 point; 3A/6 point
Insulation Loop	Mechanical insulation	Optocoupler isolation
Analog input		
Number of input points	2	
Input type	Voltage/current, DIP switch switch switch input type	
Input Range	0 ~ 10V/0 ~ 20ma	
Conversion accuracy	12 bits	
Analog output		
Number of output points	1	
Output type	Voltage, current	
Output Range	0 ~ 10V/0 ~ 20ma	
Conversion accuracy	12 bits	
High-speed counter		
Number of input points	Six channels (x 0 ~ x 5) single phase, two channels AB phase	
Pulse frequency	Single-phase 4-way 100K (x 0 ~ x 3) , 2-way 40K (x 4 ~ x 5)	
Input signal voltage	DC 18 ~ 28V	
High-speed pulse output		

Power supply	
Power Supply	DC24V, terminal access; with anti-reverse protection
Power consumption	2W ~ 4W
Surge protection	\$6 million
Structure and environment	
Size (mm)	129 * 86 * 50
Installation Mode	35mm DIN rail
Operating temperature	Working temperature 0 °C ~ + 50 °C (no freezing)
Working humidity	10 ~ 80% Rh (no condensation)



Terminal label	Function description
24 +	DC 24V Power Positive Pole
0V	DC 24V power supply negative pole
A-plus	485A+ (see Chapter 6.2 for features)
B minus	485B- (see Chapter 6.2 for features)
S/s	Common end of 1-16 digital input channel
X0	Channel 1 digital input
X1	Channel 2 digital input
X2	Channel 3 digital input
X3	Channel 4 digital input
X4	Channel 5 digital input
X5	Channel 6 digital input
X6	Number seven input
X7	Route 8 digital input
X10	Route 9 digital input
X11	10th digital input
X-12	Route 11 digital input
X13	Channel 12 digital input
X14	Route 13 digital input
X15	Route 14 digital input
X16	Route 15 digital input
X17	Route 16 digital input

Terminal label	Function description
COM0	Common terminal of 1 ~ 4 digital output channels
Y zero	Number One - way digital output
Y 1	2 - way digital output
Y 2	3 - way digital output
Y 3	4 - way digital output
COM1	Digital output common terminal of 5 ~ 10 channels
Y 4	5 - way digital output
Y-5	6 - way digital output
Y 6	7 - way digital output
Y7	No. 8 - way digital output
Y10	9 - way digital output
Y11	10th digital output
SW1	1 voltage/current Analog input switching code
SW2	2 voltage/current Analog input switching code
COM3	Analog input
In1	The first analog voltage is 0 ~ 10V/0 ~ 20ma Current input
In2	The second analog voltage is 0 ~ 10V/0 ~ 20ma Current input
COM4	Analog output
V	Analog 0 ~ 10V voltage output channel
I	Analog current output channel of 0 ~ 20ma
RS422	RS422 programming port, S terminal
Eth	10/100Mbps network port, RJ45 bus block
Sys.	Error indicator, PLC error when the Red Light
Run	Run Indicator: The PLC runs with a green light and stops with a red light
COMM	RS485 communication light, flashing

Anexo 2 – Sensor Ultrasónico JSN-SR04T-2.0

Specifications:

	Pulse width output / Serial Output
Operating Voltage	DC 3.0-5.5V
Working current	Less than 8mA
Probe frequency	40KHz
Farthest range	600cm
Recent range	20cm
Distance accuracy	+/- 1cm
Resolution	1mm
Measuring angle	75 degree
Enter the trigger signal	1,10uS above the TTL pulse 2, the serial port to send instructions 0X55
Output the echo signal	Output pulse width level signal, or TTL
Wiring	3-5.5V (power positive) Trig (RX) RX Echo (output) TX GND (power supply negative)
Product Size	L42 * W29 * H12 mm
Operating temperature	-20 ° C to + 70 ° C
Product color	PCB board is blue

Mode 1: R27 = open that is not welding. The pattern is described below

1. the basic working principle:

- (1) using IO port TRIG trigger range, to a minimum of 10us high letter.
- (2) module automatically send 8 40khz square wave, automatically detect whether there is a signal to return;
- (3) a signal to return, through the IO port ECHO output a high level, high time is the duration of ultrasound from the launch to the return time. Test distance = (high time * speed of sound (340M / s)) / 2;
- (4) module is triggered after the distance measurement, if you cannot receive the echo (the reason exceeds the measured range or the probe is not on the measured object), ECHO port will automatically become low after 60MS, marking the End of measurement, whether successful or not.
- (5) LED indicator, LED non-power indicator, it will receive the signal after the module will be lit, then the module is working.

Mode 2: R27 = 47K is the welding 47K resistance. The pattern is described below

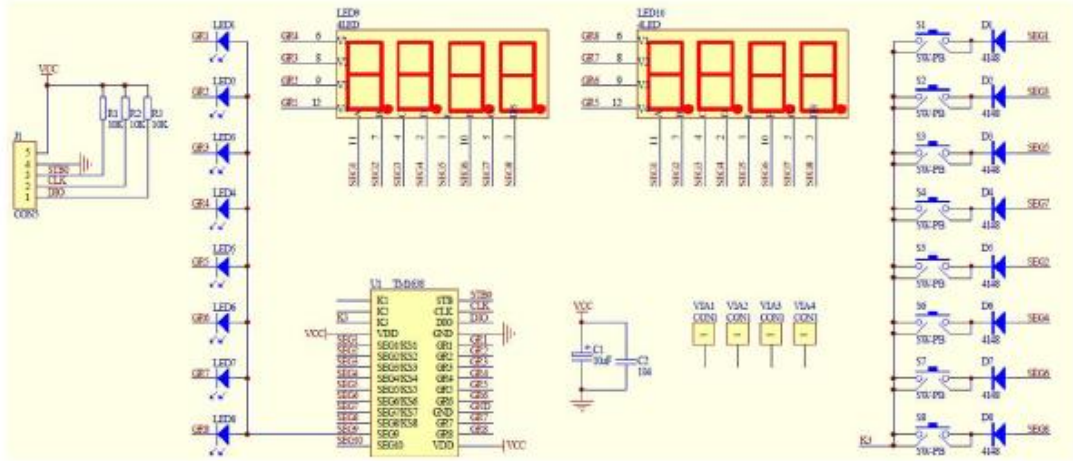
Serial **output format for the TTL level**, that: 100MS module for the cycle of automatic output distance value, the unit is mm. Serial baud rate: 9600, n, 8,1.

After the module is powered on, it will enter the working mode directly, and the range will be carried out every 100ms within the module and one frame will be output from the pin TX, including four 8-bit data. The frame format is: 0XFF + H_DATA + L_DATA + SUM

- 1.0XFF: for a frame to start the data, used to judge;
- 2.H_DATA: the upper 8 bits of the distance data;
- 3.L_DATA: the lower 8 bits of the distance data;
- 4.SUM: data and, for the effect of **its 0XFF + H_DATA + L_DATA = SUM (only low 8)**

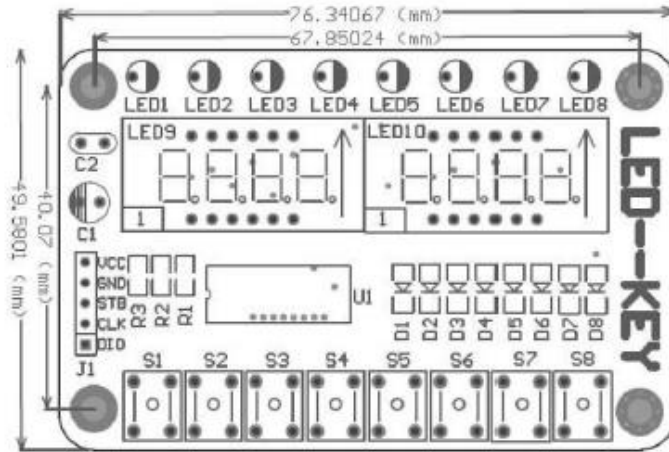
Anexo 3 – Modulo TM1638

Schematic Diagram:



Mechanical Dimension:

Unit: mm



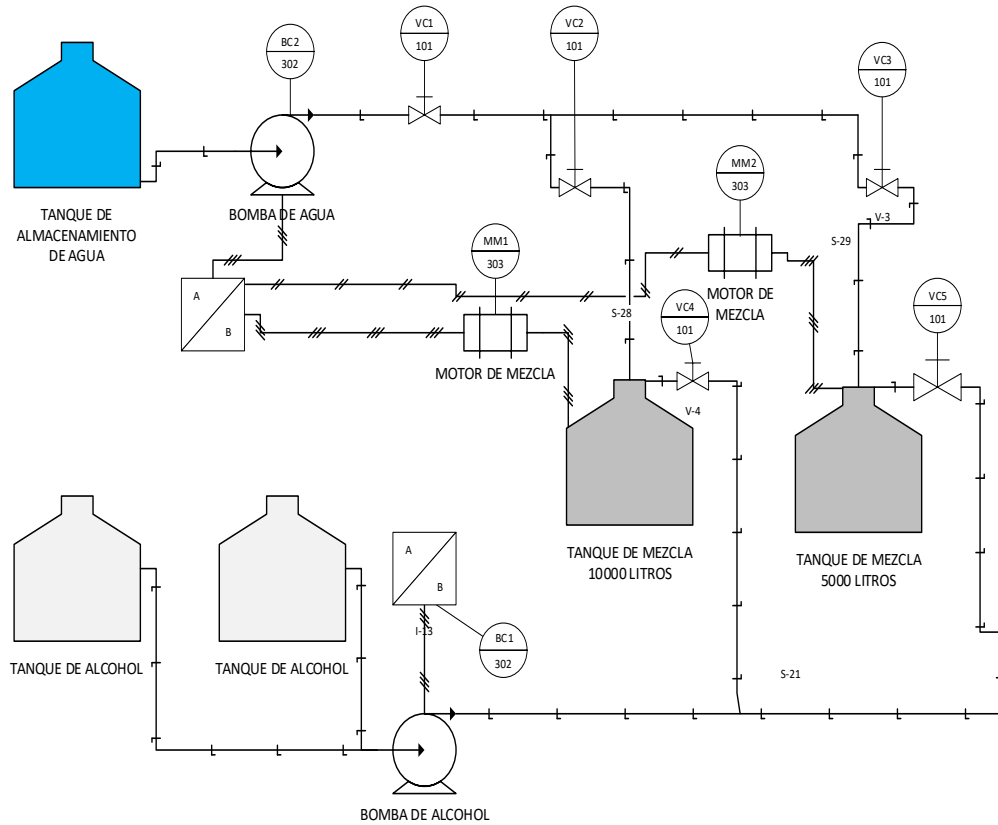
Pin Assignment:

Pin Name	Function
VCC	5-Vdc
GND	Ground
STB	Chip Select Input.
CLK	Clock Input
DIO	Data Input/output

Anexo 4 – Plano general de la empresa



Anexo 5 – Diagrama P&ID de la dosificación



Lazos		Etapas de Dosificación	
Lazo 100: Hidráulico		VC1, VC2, VC3, VC4, VC5 (101):	Control de válvulas
Lazo 200: Neumático		BC1(301):	Control de la Bomba de Alcohol
Lazo 300: Eléctrico		BC2(302):	Control de la Bomba de Agua
		MM1, MM2(303):	Motor de Mezcla

Anexo 6 – Programación del ESP 32

La programación esta realizado por subrutinas

```
#include "Macros.h"
#include "01APP.h"
#include "02Server.h"
#include "03HMI.h"
#include "04COM.h"

uint32_t TimerBlynk, TimerHMI, TimerPlcCom, TimerCom,
TimerServerInit;
uint8_t Iniciado=0;
void setup()
{
    // Debug console

    AppInit();
    Server_Init();
    HMI_Init();
    PlcComInit();
    TimerBlynk=millis()+10;
    TimerHMI=millis()+100;
    TimerPlcCom=millis()+500;
    TimerCom=millis()+100;

    TimerServerInit = millis() + 10000;
}

void loop()
{
    if (TimerBlynk < millis()){
        TimerBlynk =millis()+10;
        Server_TareaPeriodica();
    }
    if (TimerHMI < millis()){
        TimerHMI =millis()+100;
        HMI_TareaPeriodica();
    }
    if (TimerPlcCom < millis()){
        TimerPlcCom =millis()+500;
        PlcCom();
    }
    if (TimerServerInit < millis() && Iniciado ==0){
        Iniciado=1;
        COM_Init();
        Serial.print("Init COM");
    }
    if (Iniciado == 1){
        if (TimerCom < millis()){
            TimerCom =millis()+100;
            COM_TareaPeriodica();
        }
    }
}
```

Sub Rutina "Definiciones "

```
/*
    BITS Lecturas Habilitadas:
    */
#define MONITOR_LITROS_TANQUE1          0
#define MONITOR_LITROS_TANQUE2          1
#define MONITOR_TIEMPO_TANQUE1          2
#define MONITOR_TIEMPO_TANQUE2          3
#define MODO_                            4
#define MODO_SETUP_LED                   5
#define MODO_ENABLE_TANQUE1              6
#define MODO_ENABLE_TANQUE2              7

//ESTADO
#define SYS_ENABLE_TANQUE1                0
#define SYS_ENABLE_TANQUE2                1

#define READBIT(A, B) ((A >> (B & 31)) & 1)
#define SETBIT(T, B, V) (T = V ? T | (1<<B) : T & ~(1<<B))
#define True 1
#define False 0
```

Sub Rutina #include "01APP.h"

```
#include "Macros.h"
#include "01APP.h"
#include "02Server.h"
#include "03HMI.h"
#include "04COM.h"

/*#include <SoftwareSerial.h>
#define rxPin1 32
#define txPin1 33
#define BAUD_RATE1 9600
SoftwareSerial Sensor1;
*/
#include <HardwareSerial.h>
HardwareSerial PLC(2);

//APP
uint16_t Estado;
uint16_t PARAMETROS[64];

void AppInit (){
    //Para Depuracion
    Serial.begin(115200);
    Serial.println("Init Controlador WEB SERVER");

    EEPROM.begin(128);
    //LEER PARAMETROS
    uint8_t BufferRead[128];
```

```

    for (uint8_t i=0;i<128;i++){
        BufferRead[i]=EEPROM.read(i);
        //delayMicroseconds(10);
    }
    memcpy((uint8_t*)&PARAMETROS[20],(uint8_t*)&BufferRead[0], 40);
}
void PlcComInit (void){
    //Sensor1.begin(BAUD_RATE1, SWSERIAL_8N1, rxPin1, txPin1);
    PLC.begin (38400, SERIAL_8N1, 16, 17);
}

uint32_t TimPrint;
void PlcCom(void) {
    //ENVIO AL PLC ENABLE TANQUE 1 O 2

    PARAMETROS[36]=Estado;

    if (millis()>TimPrint){
        //Serial.println("ParametrosTX:");
        for (uint8_t i=20;i<40;i++){
            //Serial.print("[");Serial.print(i);Serial.print("]=");Serial.
print(PARAMETROS[i]);
        }
        //Serial.println("FIN");
    }
    uint8_t TxBuff[40];
    memcpy((uint8_t*)&TxBuff[0], (uint8_t*)&PARAMETROS[20], 40);
    for (uint8_t i=0;i<40;i++)
        PLC.write(TxBuff[i]);

    //Serial.println("Info PLC Sent");
    //TIRMP0 ESPERA RESPUESTA desde el PLC
    delay(30);
    if (PLC.available ()){
        uint8_t RxBuff[64], pBff=0;
        uint32_t TimeOut=millis()+15;
        while (PLC.available() && millis()<TimeOut){
            RxBuff[pBff]=PLC.read();
            if (pBff!=63)
                pBff++;
        }
        //Serial.print("Num:");
        //Serial.println(pBff);
        if (pBff == 40){
            memcpy((uint8_t*)&PARAMETROS[0], (uint8_t*)&RxBuff[0], 40);
        }
    }

    if (millis()>TimPrint){
        TimPrint=millis()+3000;
        //Serial.println("ParametrosRX:");
        for (uint8_t i=0;i<20;i++){
            //Serial.print("[");Serial.print(i);Serial.print("]=");Seria
l.print(PARAMETROS[i]);
        }
        //Serial.println("FIN");
    }
}

```

Sub Rutina #include "02Server.h"

```
#include "02Server.h"

//digitales
#define B_Agua_STT      V10
#define B_Agua_ALM      V4
#define B_Alcohol_STT  V11
#define B_Alcohol_ALM  V5
#define B_Mixer1_STT   V12
#define B_Mixer1_ALM   V6
#define B_Mixer2_STT   V13
#define B_Mixer2_ALM   V7
#define Remoto1_ALM    V8
#define Remoto2_ALM    V9
//analogos
#define Tanque10000L_   V0
#define Tanque5000L_   V1
#define TimMix1         V2
#define TimMix2         V3

#define BLYNK_PRINT Serial

/* Fill-in your Template ID (only if using Blynk.Cloud) */
#define BLYNK_TEMPLATE_ID "TMPLGDRITNdU"
#define BLYNK_DEVICE_NAME "DosificadoraLiquidos"
#define BLYNK_AUTH_TOKEN "wmHowfnTgIhZoIg83Y92oQPciOgFIFn1"

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "Laboratorio";
char pass[] = "Cartago2020";
//char ssid[] = "RED FRANCISCO";
//char pass[] = "0002427491";
// Set your Static IP address
IPAddress local_IP(192, 168, 88, 180);
//IPAddress local_IP(192, 168, 100, 180);
// Set your Gateway IP address
IPAddress gateway(192, 168, 88, 1);
//IPAddress gateway(192, 168, 100, 1);

IPAddress subnet(255, 255, 255, 0);
IPAddress primaryDNS(8, 8, 8, 8); //optional
IPAddress secondaryDNS(8, 8, 4, 4); //optional

BlynkTimer timer;
uint8_t StepSendInfo=0;
void myTimerEvent(){
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  if (StepSendInfo == 0){
    Serial.println("Dig send");
  }
}
```

```

    Blynk.virtualWrite(B_Agua_STT, READBIT(PARAMETROS[9], 0)); //
    Blynk.virtualWrite(B_Agua_ALM, READBIT(PARAMETROS[7], 0)); //
    Blynk.virtualWrite(B_Alcohol_STT, READBIT(PARAMETROS[9], 1)); //
    Blynk.virtualWrite(B_Alcohol_ALM, READBIT(PARAMETROS[7], 1)); //
    Blynk.virtualWrite(B_Mixer1_STT, READBIT(PARAMETROS[9], 2)); //
    Blynk.virtualWrite(B_Mixer1_ALM, READBIT(PARAMETROS[7], 2)); //
    Blynk.virtualWrite(B_Mixer2_STT, READBIT(PARAMETROS[9], 3)); //
    Blynk.virtualWrite(B_Mixer2_ALM, READBIT(PARAMETROS[7], 3)); //
    Blynk.virtualWrite(Remoto1_ALM, READBIT(PARAMETROS[7], 4)); //
    Blynk.virtualWrite(Remoto2_ALM, READBIT(PARAMETROS[7], 5)); //
    StepSendInfo++;
}
else if (StepSendInfo == 1){ //GAUGES
    Serial.println("Analog send");
    Blynk.virtualWrite(Tanque10000L_, PARAMETROS[0]); //
    Blynk.virtualWrite(Tanque5000L_, PARAMETROS[1]); //
    Blynk.virtualWrite(TimMix1, PARAMETROS[2]); //
    Blynk.virtualWrite(TimMix2, PARAMETROS[3]); //
    StepSendInfo++;
}
else if (StepSendInfo == 2){ //NOTIFICACIONES
    Serial.println("Notificaciones send");
    //SETBIT(PARAMETROS[9], 0, !READBIT(PARAMETROS[9], 0)); // TEST
    if (READBIT(PARAMETROS[9], 0) == True)
        Blynk.logEvent("bomba_agua_on", "Encendido Bomba de Agua");
    if (READBIT(PARAMETROS[7], 0) == True)
        Blynk.logEvent("PROT_BOMBA_AGUA", "!ALERTA; Bomba de Agua");
    if (READBIT(PARAMETROS[9], 1) == True)
        Blynk.logEvent("bomba_alcohol_on", "Encendido Bomba de
Alcohol");
    if (READBIT(PARAMETROS[7], 1) == True)
        Blynk.logEvent("PROT_BOMBA_ALCOHOL", "!ALERTA; Bomba de
Alcohol");
    if (READBIT(PARAMETROS[9], 2) == True)
        Blynk.logEvent("mixer_1_on", "Iniciado mexer 1");
    if (READBIT(PARAMETROS[7], 2) == True)
        Blynk.logEvent("PROT_MIXER_1", "!ALERTA; mixer 1");
    if (READBIT(PARAMETROS[9], 3) == True)
        Blynk.logEvent("mixer_2_on", "Iniciado mexer 2");
    if (READBIT(PARAMETROS[7], 3) == True)
        Blynk.logEvent("PROT_MIXER_2", "!ALERTA; mixer 2");
    if (READBIT(PARAMETROS[7], 4) == True)
        Blynk.logEvent("esclavo_1_offline", "!ALERTA; esclavo 1
OFFLINE");
    if (READBIT(PARAMETROS[7], 5) == True)
        Blynk.logEvent("esclavo_2_offline", "!ALERTA; esclavo 2
OFFLINE");

    StepSendInfo=0;
}
}
}

//FUNCION DE CONEXIÓN A TU RED WIFI
void setup_wifi(void){
    if (!WiFi.config(local_IP, gateway, subnet, primaryDNS,
secondaryDNS)) {
        Serial.println("STA Failed to configure");
    }
    Blynk.begin(auth, ssid, pass);
}

```

```

    timer.setInterval(2000L, myTimerEvent);
}
void Server_Init (void){
    setup_wifi();
}
void Server_TareaPeriodica (void){
    //CONTROL DE CONEXION Y RECONEXION
    Blynk.run();
    timer.run(); // Initiates BlynkTimer
}

```

Sub Rutina #include "03HMI.h"

```

#include "03HMI.h"

#include "04COM.h"

#define STB_PIN          26
#define SCL_PIN          4
#define SDA_PIN          23

#include <TM1638plus.h>
bool high_freq = true; //default false, If using a high freq CPU >
~100 MHZ set to true.
TM1638plus tm(STB_PIN, SCL_PIN , SDA_PIN, high_freq);
uint8_t Botones, Pilotos, Flancos;
//INTERFAZ
#define BOTON_DATOS          0
#define BOTON_PARAMETROS    1
#define BOTON_BAJAR         2
#define BOTON_SUBIR         3
#define BOTON_GUARDAR       4
#define BOTON                5
#define BOTON_ENABLE_TANQUE1 6
#define BOTON_ENABLE_TANQUE2 7

uint8_t Modo;
uint8_t Sumando=1, Restando=1, timerSumando, timerRestando;
uint8_t TimerGuardar, ToggleLed;

uint16_t TimReset;

void HMI_Init (void){
    tm.displayBegin();
    //Serial.println("Init HMI");
    if (PARAMETROS[20]>7)
        tm.brightness(7);//0-7, 0 no apaga
    else
        tm.brightness(PARAMETROS[20]);//0-7, 0 no apaga
    tm.brightness(0);//0-7, 0 no apaga
    //Modo=0;
}

void HMI_TareaPeriodica (void){
    InterfazTask();//CADA 100 mS
    if (TimerGuardar)
        TimerGuardar--;
    TimReset++;
    if (TimReset == 30){
        TimReset=0;
    }
}

```



```

        tm.reset();
        HMI_Init();
    }
}
/*
L0  L1  L2  L3  L4  L5  L6  L7
D0  D1  D2  D3  D4  D5  D6  D7
B0  B1  B2  B3  B4  B5  B6  B7
//ENTRADAS
L0
L1
L2
L3
L4
L5
L6
L7
GUARDA AUTOMATICAMENTE CON TIMER DESDE EL ULTIMA PULSACION PARA
SALIR MODO SET PARAMETROS
*/
bool LeerAccionBoton (uint8_t NumBoton){
    if (READBIT (Botones, NumBoton) == true && READBIT(Flancos,
NumBoton) == false)//PRESIONA BOTON
        SETBIT(Flancos, NumBoton, true);
    if (READBIT (Botones, NumBoton) == false && READBIT(Flancos,
NumBoton) == true){//SUELTA BOTON
        SETBIT(Flancos, NumBoton, false);
        return true;
    }
    return false;
}
void ImprimirNumTxt(uint16_t Num, char* Txt, uint8_t Toggle){
    String Buffer, Buffer2;
    uint8_t NumEspacios, PosicionEspacios;
    if (Toggle == 0){
        Buffer = String(Num);
        Buffer2 = String(Txt);
        NumEspacios = 8 - Buffer.length() - Buffer2.length();
        for (uint8_t i=0;i<NumEspacios;i++)
            Buffer+=' ';
    }
    else{
        Buffer = String(Txt);
        Buffer2 = String(Num);
        NumEspacios = 8 - Buffer.length() - Buffer2.length();
        for (uint8_t i=0;i<NumEspacios;i++)
            Buffer+=' ';
    }
    Buffer+=Buffer2;
    tm.displayText((char*)&Buffer);
}
void InterfazTask( void ){
    //LEER BOTONES
    Botones = tm.readButtons();
    //INTERFAZ
    if (LeerAccionBoton(BOTON_DATOS) == true ){ // 0 - 5
        if(Modo>=9){
            Modo=0;
        }
        else{

```

```

        Modo++;
    }
    Serial.print("Modo:");Serial.println(Modo);
}
if (LeerAccionBoton(BOTON_PARAMETROS) == true ){ // 20 - 40
    if(Modo <= 19 || Modo>=40){
        Modo=20;
    }
    else{
        Modo++;
    }
    Serial.print("Modo:");Serial.println(Modo);
}
if (READBIT (Botones, BOTON_SUBIR) == true){//PRESIONA BOTON
PARAMETROS[Modo]+=Sumando;
timerSumando++;
if (timerSumando>30)
    Sumando=10;
if (timerSumando>60)
    Sumando=100;
if (timerSumando>90)
    Sumando=1000;
//if (PARAMETROS[Modo]>9999)
//PARAMETROS[Modo]=0;
}
else{
    Sumando=1;
    timerSumando=0;
}
if (READBIT (Botones, BOTON_BAJAR) == true){//PRESIONA BOTON
PARAMETROS[Modo]-=Restando;
timerRestando++;
if (timerRestando>30)
    Restando=10;
if (timerRestando>60)
    Restando=100;
if (timerRestando>90)
    Restando=1000;
//if (PARAMETROS[Modo]>9999)
//PARAMETROS[Modo]=0;
}
else{
    Restando=1;
    timerRestando=0;
}
if (LeerAccionBoton(BOTON_GUARDAR) == true){//PRESIONA BOTON
if (TimerGuardar == 0){
    uint8_t BufferByte[128];
    memcpy((uint8_t*)&BufferByte[0], (uint8_t*)&PARAMETROS[20],
40);
//portMUX_TYPE myMutex = portMUX_INITIALIZER_UNLOCKED;
//taskENTER_CRITICAL(&myMutex);
for (uint8_t i=0;i<128;i++){
    EEPROM.write(i, BufferByte[i]);
    delay(1/portTICK_PERIOD_MS);
}
EEPROM.commit();
//taskEXIT_CRITICAL(&myMutex);
TimerGuardar=40;
}
}

```

```

    }
    if (LeerAccionBoton(BOTON_ENABLE_TANQUE1) == true){//PRESIONA
BOTON BOTON_ENABLE_TANQUE1
        SETBIT(Estado, SYS_ENABLE_TANQUE1, true); //CAMBIO DE ESTADO
        SETBIT(Estado, SYS_ENABLE_TANQUE2, false); //CAMBIO DE ESTADO
    }
    if (LeerAccionBoton(BOTON_ENABLE_TANQUE2) == true){//PRESIONA
BOTON BOTON_ENABLE_TANQUE2
        SETBIT(Estado, SYS_ENABLE_TANQUE2, true); //CAMBIO DE ESTADO
        SETBIT(Estado, SYS_ENABLE_TANQUE1, false); //CAMBIO DE ESTADO
    }

    if (Modo == 0){//LITROS 1
        ImprimirNumTxt(PARAMETROS[Modo], "Lt", 0);
    }
    else if (Modo == 1){//LITROS 2
        ImprimirNumTxt(PARAMETROS[Modo], "Lt", 0);
    }
    else if (Modo == 2){//TIEMPO 1
        ImprimirNumTxt(PARAMETROS[Modo], "sg", 0);
    }
    else if (Modo == 3){//TIEMPO 2
        ImprimirNumTxt(PARAMETROS[Modo], "sg", 0);
    }
    else if (Modo == 4){//DISTANCIA 1
        ImprimirNumTxt(PARAMETROS[Modo], "d1", 0);
    }
    else if (Modo == 5){//DISTANCIA 2
        ImprimirNumTxt(PARAMETROS[Modo], "d2", 0);
    }
    else if (Modo == 6){//ESTADO
        ImprimirNumTxt(PARAMETROS[Modo], "Est", 0);
    }
    else if (Modo == 7){//ALARMAS
        ImprimirNumTxt(PARAMETROS[Modo], "ALA", 0);
    }
    else if (Modo == 8){//ENTRADAS
        ImprimirNumTxt(PARAMETROS[Modo], "In ", 0);
    }
    else if (Modo == 9){//SALIDAS
        ImprimirNumTxt(PARAMETROS[Modo], "Out", 0);
    }
    else{//PARAMETRO
        //if (PARAMETROS[Modo] > 9999)
            //PARAMETROS[Modo]=9999;
        uint8_t NewInd=Modo;
        String BuffString= String(NewInd);
        BuffString.concat("=");
        char BuffAll[3];
        BuffString.toCharArray(BuffAll, 3);
        ImprimirNumTxt(PARAMETROS[Modo], BuffAll, 1);
    }
    //PILOTOS
    Pilotos=0;
    if (Modo == 0)//LITROS 1
        SETBIT (Pilotos, MONITOR_LITROS_TANQUE1, true);
    if (Modo == 1)//LITROS 2
        SETBIT (Pilotos, MONITOR_LITROS_TANQUE2, true);
    if (Modo == 2)//TIEMPO 1
        SETBIT (Pilotos, MONITOR_TIEMPO_TANQUE1, true);

```

```

if (Modo == 3)//TIEMPO 2
    SETBIT (Pilotos, MONITOR_TIEMPO_TANQUE2, true);
//if (Modo == 4)
    //SETBIT (Pilotos, MODO_, true);
if (Modo >= 4 && Modo <= 64)//PARAMETROS
    SETBIT (Pilotos, MODO_SETUP_LED, true);
if (TimerGuardar){
    if (ToggleLed){
        ToggleLed =0;
        SETBIT (Pilotos, MODO_SETUP_LED, true);
    }
    else{
        ToggleLed =1;
        SETBIT (Pilotos, MODO_SETUP_LED, false);
    }
}

//LED MODO_ENABLE_TANQUE1
SETBIT(Pilotos, MODO_ENABLE_TANQUE1, READBIT(Estado,
SYS_ENABLE_TANQUE1));
//LED MODO_ENABLE_TANQUE2
SETBIT(Pilotos, MODO_ENABLE_TANQUE2, READBIT(Estado,
SYS_ENABLE_TANQUE2));

tm.setLED(0, READBIT(Pilotos, 0));
tm.setLED(1, READBIT(Pilotos, 1));
tm.setLED(2, READBIT(Pilotos, 2));
tm.setLED(3, READBIT(Pilotos, 3));
tm.setLED(4, READBIT(Pilotos, 4));
tm.setLED(5, READBIT(Pilotos, 5));
tm.setLED(6, READBIT(Pilotos, 6));
tm.setLED(7, READBIT(Pilotos, 7));

Botones=0;
}

```

Sub Rutina #include "04COM.h"

```

#include "04COM.h"
#include "03HMI.h"
#include "02Server.h"

//Parameters
String nombre = "Master";
WiFiServer server(80);

uint16_t TimeAlarmaSLV1, TimeAlarmaSLV2;
void COM_Init (void){
    while (WiFi.status() != WL_CONNECTED) {
        delay(100);
    }
    server.begin();
    Serial.println("Server Distancias Init");
}
void COM_TareaPeriodica (void){
    TimeAlarmaSLV1++;
}

```

```

if (TimeAlarmaSLV1>=50){
    PARAMETROS[4]= 999;
    PARAMETROS[38]= 999;
}
TimeAlarmaSLV2++;
if (TimeAlarmaSLV2>=50){
    PARAMETROS[5]= 999;
    PARAMETROS[39]= 999;
}
WiFiClient client = server.available();
client.setTimeout(50);
if (client) {
    if (client.connected()) {
        String answerA = client.readStringUntil('\r'); //receives
the message from the client
        //Serial.println(answerA);
        if (answerA.indexOf("Tanque A") >= 0) {
            if (answerA.indexOf("Dist1:") >= 0){
                String
ReadStr=answerA.substring(answerA.indexOf("Dist1:") +
sizeof("Dist1:") -1,
                                                                    answerA.indexOf("EN
D"));
                Serial.println("Dist1: ");Serial.println(ReadStr);
                PARAMETROS[4]= ReadStr.toFloat();
                PARAMETROS[38]= PARAMETROS[4];
                TimeAlarmaSLV1=0;
            }
            // RESPUESTA
            //client.println(nombre + ":Out:" + String(0) +
"END\r");
            client.stop();
        }
        if (answerA.indexOf("Tanque B") >= 0) {
            if (answerA.indexOf("Dist2:") >= 0){
                String
ReadStr=answerA.substring(answerA.indexOf("Dist2:") +
sizeof("Dist2:") -1,
                                                                    answerA.indexOf("EN
D"));
                Serial.println("___Dist2:
");Serial.println(ReadStr);
                PARAMETROS[5]= ReadStr.toFloat();
                PARAMETROS[39]= PARAMETROS[5];
                TimeAlarmaSLV2=0;
            }
            // RESPUESTA
            //client.println(nombre + ":Out:" + String(0) +
"END\r");
            client.stop();
        }
    }
}
}
}
}

```

Anexo 7 – Programación del PLC

```
#include "00Definiciones.h"
#include "01APP.h"
HardwareSerial Serial1(UART4); //RS485 DEBUG
HardwareSerial Serial2(USART3); //ESP32

extern uint16_t ParametrosPLC[64];
extern uint16_t Entradas, Salidas;

void setup() {
    // put your setup code here, to run once:
    AppInit();
}

void loop() {
    // put your main code here, to run repeatedly:
    ComHMI();
    ProcesoTask();
}

uint32_t TimPrint;

void ComHMI (void){
    if (Serial2.available()){
        delay(12);
        //Serial1.println("data:");
        uint8_t RxBuff[64], pBff=0;
        uint32_t TimeOut=millis()+20;
        while (Serial2.available() && millis()<TimeOut){
            RxBuff[pBff]=Serial2.read();
            if (pBff!=63)
                pBff++;
        }
        Serial2.flush();
        //Serial1.print("N;");Serial1.println(pBff);
        if (pBff == 40){
            memcpy((uint8_t*)&ParametrosPLC[20], (uint8_t*)&RxBuff[0],
40);

            if (millis()>TimPrint){
                Serial1.println("ParametrosRX:");
                for (uint8_t i=20;i<40;i++){
                    Serial1.print("[");Serial1.print(i);Serial1.print("]=");Serial1.print(ParametrosPLC[i]);
                }
                Serial1.println("FIN");
            }

            //DESCARGA DE VARIABLES
            //SIMULACION DISTANCIA

            uint16_t Compensacion1 = map(analogRead(11), 1024, 0, 40,
300); // 40 a 300 cm
            uint16_t Compensacion2 = map(analogRead(12), 1024, 0, 40,
300); // 40 a 300 cm
            //ParametrosPLC[38] = map(analogRead(11), 1024, 0, 40, 300); //
40 a 300 cm
            //ParametrosPLC[39] = map(analogRead(12), 1024, 0, 40, 300); //
40 a 300 cm
```

```

//Serial1.print("Pot1:");Serial1.println(analogRead(11));
//Serial1.print("Pot2:");Serial1.println(analogRead(12));
//Serial1.print("Simul:");Serial1.println(ParametrosPLC[38]);
//Serial1.print("Simu2:");Serial1.println(ParametrosPLC[39]);
if (ParametrosPLC[38] != 999){
    LitrosRead1 = map(ParametrosPLC[38] - Compensacion1,
ParametrosPLC[26], ParametrosPLC[27], ParametrosPLC[28],
ParametrosPLC[29]);
    ParametrosPLC[0] = LitrosRead1;
    SETBIT(Alarmas, FALLA_Slave1, false);
}
else
    SETBIT(Alarmas, FALLA_Slave1, True);
if (ParametrosPLC[39] != 999){
    LitrosRead2 = map(ParametrosPLC[39] - Compensacion2,
ParametrosPLC[30], ParametrosPLC[31], ParametrosPLC[32],
ParametrosPLC[33]);
    ParametrosPLC[1] = LitrosRead2;
    SETBIT(Alarmas, FALLA_Slave2, false);
}
else
    SETBIT(Alarmas, FALLA_Slave2, True);
//RESPONDER
uint8_t TxBuff[40];
memcpy((uint8_t*)&TxBuff[0], (uint8_t*)&ParametrosPLC[0], 40);
for (uint8_t i=0;i<40;i++){
    Serial2.write(TxBuff[i]);

if (millis()>TimPrint){
    TimPrint=millis()+10000;
    Serial1.println("ParametrosTX:");
    for (uint8_t i=0;i<20;i++){
        Serial1.print("[");Serial1.print(i);Serial1.print("]=");Serial1.print(ParametrosPLC[i]);
    }
    Serial1.println("FIN");
}
}
Serial2.flush();
}
ParametrosPLC[2] = TIMERS[TIMER_TimerMix1]/100;
ParametrosPLC[3] = TIMERS[TIMER_TimerMix2]/100;

ParametrosPLC[6] = Estado;
ParametrosPLC[7] = Alarmas;
ParametrosPLC[8] = Entradas;
ParametrosPLC[9] = Salidas;
}

```

Sub Rutina Definiciones

```

#include "Arduino.h"

#define READBIT(A, B) ((A >> (B & 31)) & 1)
#define SETBIT(T, B, V) (T = V ? T | (1<<B) : T & ~(1<<B))
#define True 1
#define False 0

```

```

//SALIDAS PLC - MICROCONTROLADOR
#define RelayY0    PE8
#define RelayY1    PE7
#define RelayY2    PB0
#define RelayY3    PE6
#define RelayY4    PE5
#define RelayY5    PE4
#define RelayY6    PE3
#define RelayY7    PE2
#define RelayY10   PE15
#define RelayY11   PE14
#define _LED_SYS   PE1
#define _LED_RUN   PE0
//ENTRADAS PLC - MICROCONTROLADOR
#define EntradaX00  PA5
#define EntradaX01  PD9
#define EntradaX02  PD10
#define EntradaX03  PD11
#define EntradaX04  PD12
#define EntradaX05  PD13
#define EntradaX06  PD14
#define EntradaX07  PD15
#define EntradaX10  PD0
#define EntradaX11  PD1
#define EntradaX12  PD2
#define EntradaX13  PD3
#define EntradaX14  PD4
#define EntradaX15  PD5
#define EntradaX16  PD6
#define EntradaX17  PD7
//IN ADC
#define ADC_Pot2    PC1//ADC1-channel 11  POT2
#define ADC_Pot1    PC2//ADC1-channel 12  POT1
#define ADC_In2     PC3//ADC1-channel 13  IN1
#define ADC_In1     PC4//ADC1-channel 14  IN2
#define ADC_CHNL_Pot2  11
#define ADC_CHNL_Pot1  12
#define ADC_CHNL_In2   13
#define ADC_CHNL_In1   14
//OUT DAC
#define DAC1_PLC    PA4// DAC de 8 bits 0 - 255

/*
   BITS Entradas 1:
*/
#define Input1      0//X00
#define Input2      1//X01
#define Input3      2//X02
#define Input4      3//X03
#define Input5      4//X04
#define Input6      5//X05
#define Input7      6//X06
#define Input8      7//X07
#define Input9      8//X10
#define Input10     9//X11
#define Input11     10//X12
#define Input12     11//X13
#define Input13     12//X14
#define Input14     13//X15
#define Input15     14//X16

```



```

#define Input16      15//X17
/*
    BITS Salidas 1:
*/
#define Output1      0//Y00
#define Output2      1//Y01
#define Output3      2//Y02
#define Output4      3//Y03
#define Output5      4//Y04
#define Output6      5//Y05
#define Output7      6//Y06
#define Output8      7//Y07
#define Output9      8//Y10
#define Output10     9//Y11

//SALIDAS SYS PLC
#define LED_SYS(x)    GPIOE->BSRR = (x)? GPIO_PIN_1 :
(uint32_t)GPIO_PIN_1 << 16u;
#define LED_RUN(x)    GPIOE->BSRR = (x)? GPIO_PIN_0 :
(uint32_t)GPIO_PIN_0 << 16u;

//PLC
//ENTRADAS PLC
#define LEER_IN1      ((GPIOA->IDR) & GPIO_PIN_5)
#define LEER_IN2      ((GPIOD->IDR) & GPIO_PIN_9)
#define LEER_IN3      ((GPIOD->IDR) & GPIO_PIN_10)
#define LEER_IN4      ((GPIOD->IDR) & GPIO_PIN_11)
#define LEER_IN5      ((GPIOD->IDR) & GPIO_PIN_12)
#define LEER_IN6      ((GPIOD->IDR) & GPIO_PIN_13)
#define LEER_IN7      ((GPIOD->IDR) & GPIO_PIN_14)
#define LEER_IN8      ((GPIOD->IDR) & GPIO_PIN_15)
#define LEER_IN9      ((GPIOD->IDR) & GPIO_PIN_0)
#define LEER_IN10     ((GPIOD->IDR) & GPIO_PIN_1)
#define LEER_IN11     ((GPIOD->IDR) & GPIO_PIN_2)
#define LEER_IN12     ((GPIOD->IDR) & GPIO_PIN_3)
#define LEER_IN13     ((GPIOD->IDR) & GPIO_PIN_4)
#define LEER_IN14     ((GPIOD->IDR) & GPIO_PIN_5)
#define LEER_IN15     ((GPIOD->IDR) & GPIO_PIN_6)
#define LEER_IN16     ((GPIOD->IDR) & GPIO_PIN_7)

//SALIDAS PLC
#define SET_OUT1(x)    GPIOE->BSRR = (x)? GPIO_PIN_8 :
(uint32_t)GPIO_PIN_8 << 16u;
#define SET_OUT2(x)    GPIOE->BSRR = (x)? GPIO_PIN_7 :
(uint32_t)GPIO_PIN_7 << 16u;
#define SET_OUT3(x)    GPIOB->BSRR = (x)? GPIO_PIN_0 :
(uint32_t)GPIO_PIN_0 << 16u;
#define SET_OUT4(x)    GPIOE->BSRR = (x)? GPIO_PIN_6 :
(uint32_t)GPIO_PIN_6 << 16u;
#define SET_OUT5(x)    GPIOE->BSRR = (x)? GPIO_PIN_5 :
(uint32_t)GPIO_PIN_5 << 16u;
#define SET_OUT6(x)    GPIOE->BSRR = (x)? GPIO_PIN_4 :
(uint32_t)GPIO_PIN_4 << 16u;
#define SET_OUT7(x)    GPIOE->BSRR = (x)? GPIO_PIN_3 :
(uint32_t)GPIO_PIN_3 << 16u;
#define SET_OUT8(x)    GPIOE->BSRR = (x)? GPIO_PIN_2 :
(uint32_t)GPIO_PIN_2 << 16u;

```

```

#define SET_OUT9(x)      GPIOE->BSRR = (x)? GPIO_PIN_15 :
(uint32_t)GPIO_PIN_15 << 16u;
#define SET_OUT10(x)    GPIOE->BSRR = (x)? GPIO_PIN_14 :
(uint32_t)GPIO_PIN_14 << 16u;

/*
    BITS Salidas:
*/

#define MIXER1           Output1
#define MIXER2           Output2
#define BOMBA_AGUA      Output4
#define RUN_BOMBA_ALCOHOL      Output5
#define STOP_BOMBA_ALCOHOL      Output7

#define SIRENA_ALARMA   Output10

/*
    BITS Entradas:
*/

#define PROTECCION_MIXER1_IN      Input1
#define PROTECCION_MIXER2_IN      Input2
#define PROTECCION_B_AGUA_IN      Input3
#define PROTECCION_B_ALCOHOL_IN   Input4
#define AUTOMATICO_IN             Input5

#define DOSIFICAR_AGUA_IN         Input12
#define DOSIFICAR_ALCOHOL_IN      Input13
#define MIXER1_IN                 Input14
#define MIXER2_IN                 Input15
#define RESET_ALARMAS_IN          Input16

/*
    BITS Estado:
*/
#define DOSIFICAR_AGUA1_ST        00
#define DOSIFICAR_AGUA2_ST        01
#define DOSIFICAR_ALCOHOL1_ST     02
#define DOSIFICAR_ALCOHOL2_ST     03
#define MEZCLAR1_ST               04
#define MEZCLAR2_ST               05
#define GUARDAR_DATA              06
#define RESET_ALARMAS             07

//TIMERS SOFTWARE
#define TIMER_TimerMix1           0
#define TIMER_TimerMix2           1

#define TIMER_TimSirenaFinDosif   3

//FALLAS
#define FALLA_ProteccionAgua      0
#define FALLA_ProteccionAlcohol   1
#define FALLA_ProteccionMixer1    2
#define FALLA_ProteccionMixer2    3

```

```

#define FALLA_Slave1          4
#define FALLA_Slave2          5

//ESTADO
#define SYS_ENABLE_TANQUE1    0
#define SYS_ENABLE_TANQUE2    1

```

Sub Rutina APP

```

#include "00Definiciones.h"
#include "01APP.h"

//SYS
uint16_t ParametrosPLC[64];
uint16_t Entradas, Salidas, Estado, Alarmas;
uint8_t ANTIREBOTE[16];

uint32_t Ticks10ms;
uint16_t FlancosBoton, FlancosSelector;
uint64_t EstadoTimers;
uint32_t TIMERS[64];

uint16_t TimerProteccionesOn[16];

//APP
int16_t LitrosRead1, LitrosRead2;
int16_t LitrosActualAguar, LitrosActualAgua2, LitrosActualAlcohol1,
LitrosActualAlcohol2;
uint8_t StepDosificarAguar, StepDosificarAgua2,
StepDosificarAlcohol1, StepDosificarAlcohol2, StepMix1, StepMix2;

void AppInit (){
    //PERIFERICOS SESSORES
    Serial1.setRx(PC11);
    Serial1.setTx(PC10);
    Serial1.begin(115200);

    Serial1.println("PLC INIT");

    Serial2.setRx(PB11);
    Serial2.setTx(PB10);
    Serial2.begin(38400);

    //SALIDAS
    pinMode(RelayY0, OUTPUT);
    pinMode(RelayY1, OUTPUT);
    pinMode(RelayY2, OUTPUT);
    pinMode(RelayY3, OUTPUT);
    pinMode(RelayY4, OUTPUT);
    pinMode(RelayY5, OUTPUT);
    pinMode(RelayY6, OUTPUT);
    pinMode(RelayY7, OUTPUT);
    pinMode(RelayY10, OUTPUT);
    pinMode(RelayY11, OUTPUT);
    pinMode(_LED_SYS, OUTPUT);
    pinMode(_LED_RUN, OUTPUT);
}

```

```

//ENTRADAS
pinMode(EntradaX00, INPUT);
pinMode(EntradaX01, INPUT);
pinMode(EntradaX02, INPUT);
pinMode(EntradaX03, INPUT);
pinMode(EntradaX04, INPUT);
pinMode(EntradaX05, INPUT);
pinMode(EntradaX06, INPUT);
pinMode(EntradaX07, INPUT);
pinMode(EntradaX10, INPUT);
pinMode(EntradaX11, INPUT);
pinMode(EntradaX12, INPUT);
pinMode(EntradaX13, INPUT);
pinMode(EntradaX14, INPUT);
pinMode(EntradaX15, INPUT);
pinMode(EntradaX16, INPUT);
pinMode(EntradaX17, INPUT);
//ENTRADAS ADC
pinMode(ADC_Pot2, INPUT); //ADC1-channel 11 POT2
pinMode(ADC_Pot1, INPUT); //ADC1-channel 12 POT1
pinMode(ADC_In2, INPUT); //ADC1-channel 13 In2
pinMode(ADC_In1, INPUT); //ADC1-channel 14 In1
//DAC
pinMode(PA4, OUTPUT);

TIM6_Config();
}
void IOTask(void) {
//ENTRADAS
(!LEER_IN1) ? (ANTIREBOTE[0]<5) ? ANTIREBOTE[0]++ :
SETBIT(Entradas, Input1, True) : (ANTIREBOTE[0]>0) ? ANTIREBOTE[0]--
: SETBIT(Entradas, Input1, False);
(!LEER_IN2) ? (ANTIREBOTE[1]<5) ? ANTIREBOTE[1]++ :
SETBIT(Entradas, Input2, True) : (ANTIREBOTE[1]>0) ? ANTIREBOTE[1]--
: SETBIT(Entradas, Input2, False);
(!LEER_IN3) ? (ANTIREBOTE[2]<5) ? ANTIREBOTE[2]++ :
SETBIT(Entradas, Input3, True) : (ANTIREBOTE[2]>0) ? ANTIREBOTE[2]--
: SETBIT(Entradas, Input3, False);
(!LEER_IN4) ? (ANTIREBOTE[3]<5) ? ANTIREBOTE[3]++ :
SETBIT(Entradas, Input4, True) : (ANTIREBOTE[3]>0) ? ANTIREBOTE[3]--
: SETBIT(Entradas, Input4, False);
(!LEER_IN5) ? (ANTIREBOTE[4]<5) ? ANTIREBOTE[4]++ :
SETBIT(Entradas, Input5, True) : (ANTIREBOTE[4]>0) ? ANTIREBOTE[4]--
: SETBIT(Entradas, Input5, False);
(!LEER_IN6) ? (ANTIREBOTE[5]<5) ? ANTIREBOTE[5]++ :
SETBIT(Entradas, Input6, True) : (ANTIREBOTE[5]>0) ? ANTIREBOTE[5]--
: SETBIT(Entradas, Input6, False);
(!LEER_IN7) ? (ANTIREBOTE[6]<5) ? ANTIREBOTE[6]++ :
SETBIT(Entradas, Input7, True) : (ANTIREBOTE[6]>0) ? ANTIREBOTE[6]--
: SETBIT(Entradas, Input7, False);
(!LEER_IN8) ? (ANTIREBOTE[7]<5) ? ANTIREBOTE[7]++ :
SETBIT(Entradas, Input8, True) : (ANTIREBOTE[7]>0) ? ANTIREBOTE[7]--
: SETBIT(Entradas, Input8, False);
(!LEER_IN9) ? (ANTIREBOTE[8]<5) ? ANTIREBOTE[8]++ :
SETBIT(Entradas, Input9, True) : (ANTIREBOTE[8]>0) ? ANTIREBOTE[8]--
: SETBIT(Entradas, Input9, False);
(!LEER_IN10) ? (ANTIREBOTE[9]<5) ? ANTIREBOTE[9]++ :
SETBIT(Entradas, Input10, True) : (ANTIREBOTE[9]>0) ? ANTIREBOTE[9]--
- : SETBIT(Entradas, Input10, False);

```

```

        (!LEER_IN11) ? (ANTIREBOTE[10]<5) ? ANTIREBOTE[10]++ :
SETBIT(Entradas, Input11, True) : (ANTIREBOTE[10]>0) ?
ANTIREBOTE[10]-- : SETBIT(Entradas, Input11, False);
        (!LEER_IN12) ? (ANTIREBOTE[11]<5) ? ANTIREBOTE[11]++ :
SETBIT(Entradas, Input12, True) : (ANTIREBOTE[11]>0) ?
ANTIREBOTE[11]-- : SETBIT(Entradas, Input12, False);
        (!LEER_IN13) ? (ANTIREBOTE[12]<5) ? ANTIREBOTE[12]++ :
SETBIT(Entradas, Input13, True) : (ANTIREBOTE[12]>0) ?
ANTIREBOTE[12]-- : SETBIT(Entradas, Input13, False);
        (!LEER_IN14) ? (ANTIREBOTE[13]<5) ? ANTIREBOTE[13]++ :
SETBIT(Entradas, Input14, True) : (ANTIREBOTE[13]>0) ?
ANTIREBOTE[13]-- : SETBIT(Entradas, Input14, False);
        (!LEER_IN15) ? (ANTIREBOTE[14]<5) ? ANTIREBOTE[14]++ :
SETBIT(Entradas, Input15, True) : (ANTIREBOTE[14]>0) ?
ANTIREBOTE[14]-- : SETBIT(Entradas, Input15, False);
        (!LEER_IN16) ? (ANTIREBOTE[15]<5) ? ANTIREBOTE[15]++ :
SETBIT(Entradas, Input16, True) : (ANTIREBOTE[15]>0) ?
ANTIREBOTE[15]-- : SETBIT(Entradas, Input16, False);

//SALIDAS
SET_OUT1(READBIT(Salidas, Output1));
SET_OUT2(READBIT(Salidas, Output2));
SET_OUT3(READBIT(Salidas, Output3));
SET_OUT4(READBIT(Salidas, Output4));
SET_OUT5(READBIT(Salidas, Output5));
SET_OUT6(READBIT(Salidas, Output6));
SET_OUT7(!READBIT(Salidas, Output7));
SET_OUT8(READBIT(Salidas, Output8));
SET_OUT9(READBIT(Salidas, Output9));
SET_OUT10(READBIT(Salidas, Output10));

//ENTRADAS ANALOGAS
//In1 = analogRead(ADC_CHNL_Pot2)
//In2 = analogRead(ADC_CHNL_Pot1)
//In3 = analogRead(ADC_CHNL_In1)
//In4 = analogRead(ADC_CHNL_In2)

//SALIDAS ANALOGAS
//analogWrite(DAC1_PLC, DAC_VAR);
}
void ManejadorTimers (void){
    for (uint8_t i=0;i<64;i++){
        if (TIMERS[i] == 1){
            TIMERS[i]=0;
            SETBIT(EstadoTimers, i, True);
        }
        else{
            if (TIMERS[i])
                TIMERS[i]--;
        }
    }
}
void ResetAlarmas (void){
    if (READBIT (Estado, RESET_ALARMAS) == True){
        SETBIT(Estado, RESET_ALARMAS, False);
        Alarmas=0;
        SETBIT(Salidas, SIRENA_ALARMA, False);
    }
}
void DosificarAgua1 (void){

```

```

    if (READBIT (Estado, DOSIFICAR_AGUA1_ST) == True){
        if (StepDosificarAgua1 == 0){
            SETBIT(Salidas, BOMBA_AGUA, True);
            LitrosActualAgua1 = LitrosRead1;
            Serial1.print("Dos Agua 1
INIT:");Serial1.println(LitrosActualAgua1);
            StepDosificarAgua1++;
        }
        if (StepDosificarAgua1 == 1){
            //COMPARO CON DISTANCIA DE TANQUE
            if (abs(LitrosRead1 - LitrosActualAgua1) >=
ParametrosPLC[20]){
                Serial1.print("Dos Agua 1
FIN:");Serial1.println(abs(LitrosRead1 - LitrosActualAgua1));
                SETBIT(Salidas, BOMBA_AGUA, False);
                SETBIT(Estado, DOSIFICAR_AGUA1_ST, False);
                //AVISO FIN
                SETBIT(Salidas, SIRENA_ALARMA, True);
                TIMERS[TIMER_TimSirenaFinDosif]=10 * 10; // EN DECIMAS
SEGUNDOS Y ENTRA CADA 10 mS
                TIMERS[TIMER_TimSirenaFinDosif]++;
                StepDosificarAgua1=255;
            }
        }
    }
}
else{
    StepDosificarAgua1=0;
}
}
void DosificarAgua2 (void){
    if (READBIT (Estado, DOSIFICAR_AGUA2_ST) == True){
        if (StepDosificarAgua2 == 0){
            SETBIT(Salidas, BOMBA_AGUA, True);
            LitrosActualAgua2 = LitrosRead2;
            Serial1.print("Dos Agua 2
INIT:");Serial1.println(LitrosActualAgua2);
            StepDosificarAgua2++;
        }
        if (StepDosificarAgua2 == 1){
            //COMPARO CON DISTANCIA DE TANQUE
            if (abs(LitrosRead2 - LitrosActualAgua2) >=
ParametrosPLC[21]){
                Serial1.print("Dos Agua 2
FIN:");Serial1.println(abs(LitrosRead2 - LitrosActualAgua2));
                SETBIT(Salidas, BOMBA_AGUA, False);
                //AVISO FIN
                SETBIT(Salidas, SIRENA_ALARMA, True);
                TIMERS[TIMER_TimSirenaFinDosif]=10 * 10; // EN DECIMAS
SEGUNDOS Y ENTRA CADA 10 mS
                TIMERS[TIMER_TimSirenaFinDosif]++;

                SETBIT(Estado, DOSIFICAR_AGUA2_ST, False);
                StepDosificarAgua2=255;
            }
        }
    }
}
else{
    StepDosificarAgua2=0;
}
}
}

```

```

void DosificarAlcohol1 (void){
  if (READBIT (Estado, DOSIFICAR_ALCOHOL1_ST) == True){
    if (StepDosificarAlcohol1 == 0){
      SETBIT(Salidas, RUN_BOMBA_ALCOHOL, True);
      LitrosActualAlcohol1 = LitrosRead1;
      Serial1.print("Dos Alcohol 1
INIT:");Serial1.println(LitrosActualAlcohol1);
      StepDosificarAlcohol1++;
    }
    if (StepDosificarAlcohol1 == 1){
      //COMPARO CON DISTANCIA DE TANQUE
      if (abs(LitrosRead1 - LitrosActualAlcohol1) >=
ParametrosPLC[22]){
        Serial1.print("Dos Alcohol 1
FIN:");Serial1.println(abs(LitrosRead1 - LitrosActualAlcohol1));
        SETBIT(Salidas, RUN_BOMBA_ALCOHOL, False);
        SETBIT(Salidas, STOP_BOMBA_ALCOHOL, True);
        delay(500);
        SETBIT(Salidas, STOP_BOMBA_ALCOHOL, False);
        //AVISO FIN
        SETBIT(Salidas, SIRENA_ALARMA, True);
        TIMERS[TIMER_TimSirenaFinDosif]=10 * 10; // EN DECIMAS
SEGUNDOS Y ENTRA CADA 10 mS
        TIMERS[TIMER_TimSirenaFinDosif]++;
        SETBIT(Estado, DOSIFICAR_ALCOHOL1_ST, False);
        StepDosificarAlcohol1=255;
      }
    }
  }
  else{
    StepDosificarAlcohol1=0;
  }
}

void DosificarAlcohol2 (void){
  if (READBIT (Estado, DOSIFICAR_ALCOHOL2_ST) == True){
    if (StepDosificarAlcohol2 == 0){
      SETBIT(Salidas, RUN_BOMBA_ALCOHOL, True);
      LitrosActualAlcohol2 = LitrosRead2;
      Serial1.print("Dos Alcohol 2
INIT:");Serial1.println(LitrosActualAlcohol2);
      StepDosificarAlcohol2++;
    }
    if (StepDosificarAlcohol2 == 1){
      //COMPARO CON DISTANCIA DE TANQUE
      if (abs(LitrosRead2 - LitrosActualAlcohol2) >=
ParametrosPLC[23]){
        Serial1.print("Dos Alcohol 2
FIN:");Serial1.println(abs(LitrosRead2 - LitrosActualAlcohol2));
        SETBIT(Salidas, RUN_BOMBA_ALCOHOL, False);
        SETBIT(Salidas, STOP_BOMBA_ALCOHOL, True);
        delay(500);
        SETBIT(Salidas, STOP_BOMBA_ALCOHOL, False);
        //AVISO FIN
        SETBIT(Salidas, SIRENA_ALARMA, True);
        TIMERS[TIMER_TimSirenaFinDosif]=10 * 10; // EN DECIMAS
SEGUNDOS Y ENTRA CADA 10 mS
        TIMERS[TIMER_TimSirenaFinDosif]++;
        SETBIT(Estado, DOSIFICAR_ALCOHOL2_ST, False);
        StepDosificarAlcohol2=255;
      }
    }
  }
}

```

```

    }
  }
  else{
    StepDosificarAlcohol2=0;
  }
}
void Mezclar1 (void){
  if (READBIT (Estado, MEZCLAR1_ST) == True){
    if (StepMix1 == 0){
      SETBIT(Salidas, MIXER1, True);
      TIMERS[TIMER_TimerMix1]=ParametrosPLC[24] * 100;// * 60; // EN
DECIMAS SEGUNDOS Y ENTRA CADA 10 mS
      TIMERS[TIMER_TimerMix1]++;
      Serial1.print("Mezclar 1
INIT:");Serial1.println(TIMERS[TIMER_TimerMix1]);
      StepMix1++;
    }
    if (StepMix1 == 1){
      //TIMER
      if (READBIT(EstadoTimers, TIMER_TimerMix1) == True){
        SETBIT(EstadoTimers, TIMER_TimerMix1, False);
        Serial1.print("Mezclar 1
FIN:");Serial1.println(TIMERS[TIMER_TimerMix1]);
        SETBIT(Salidas, MIXER1, False);
        SETBIT(Estado, MEZCLAR1_ST, False);
        //AVISO FIN
        SETBIT(Salidas, SIRENA_ALARMA, True);
        TIMERS[TIMER_TimSirenaFinDosif]=10 * 10; // EN DECIMAS
SEGUNDOS Y ENTRA CADA 10 mS
        TIMERS[TIMER_TimSirenaFinDosif]++;

        StepMix1=255;
      }
    }
  }
  else{
    StepMix1=0;
    TIMERS[TIMER_TimerMix1]=0; //STOP TIMER
    SETBIT(EstadoTimers, TIMER_TimerMix1, False);
  }
}
void Mezclar2 (void){
  if (READBIT (Estado, MEZCLAR2_ST) == True){
    if (StepMix2 == 0){
      SETBIT(Salidas, MIXER2, True);
      TIMERS[TIMER_TimerMix2]=ParametrosPLC[25] * 100;// * 60; // EN
DECIMAS SEGUNDOS Y ENTRA CADA 10 mS
      TIMERS[TIMER_TimerMix2]++;
      Serial1.print("Mezclar 2
INIT:");Serial1.println(TIMERS[TIMER_TimerMix2]);
      StepMix2++;
    }
    if (StepMix2 == 1){
      //TIMER
      if (READBIT(EstadoTimers, TIMER_TimerMix2) == True){
        SETBIT(EstadoTimers, TIMER_TimerMix2, False);
        Serial1.print("Mezclar 2
FIN:");Serial1.println(TIMERS[TIMER_TimerMix2]);
        SETBIT(Salidas, MIXER2, False);
        SETBIT(Estado, MEZCLAR2_ST, False);

```



```

        //AVISO FIN
        SETBIT(Salidas, SIRENA_ALARMA, True);
        TIMERS[TIMER_TimSirenaFinDosif]=10 * 10; // EN DECIMAS
SEGUNDOS Y ENTRA CADA 10 mS
        TIMERS[TIMER_TimSirenaFinDosif]++;

        StepMix2=255;
    }
}
}
else{
    StepMix2=0;
    TIMERS[TIMER_TimerMix2]=0; //STOP TIMER
    SETBIT(EstadoTimers, TIMER_TimerMix2, False);
}
}
bool LeerAccionBoton (uint8_t NumBoton){
    if (READBIT (Entradas, NumBoton) == true && READBIT(FlancosBoton,
NumBoton) == false)//PRESIONA BOTON
        SETBIT(FlancosBoton, NumBoton, true);
    if (READBIT (Entradas, NumBoton) == false
&& READBIT(FlancosBoton, NumBoton) == true){//SUELTA BOTON
        SETBIT(FlancosBoton, NumBoton, false);
        return true;
    }
    return false;
}
bool LeerAccionSelector (uint8_t NumSelector){
    if (READBIT (Entradas, NumSelector) == true &&
READBIT(FlancosSelector, NumSelector) == false){//ACTIVA SELECTOR
        SETBIT(FlancosSelector, NumSelector, true);
        return true;
    }
    if (READBIT (Entradas, NumSelector) == false
&& READBIT(FlancosSelector, NumSelector) == true)//REGRESA SELECTOR
A OFF
        SETBIT(FlancosSelector, NumSelector, false);
    return false;
}
void Condicionantes (void){
    //BOTON
    if (LeerAccionBoton(RESET_ALARMAS_IN) == true ){// CUANDO SUELTA
EL BOTON
        SETBIT(Estado, RESET_ALARMAS, true);
    }
    //SOLO SI ESTA EN AUTOMATICO CON EL SELECTOR
    if (READBIT (Entradas, AUTOMATICO_IN) == true){
        if (READBIT (ParametrosPLC[36], SYS_ENABLE_TANQUE1) == true &&
READBIT (ParametrosPLC[36], SYS_ENABLE_TANQUE2) == false){
            //TANQUE 1 AGUA
            if (LeerAccionBoton(DOSIFICAR_AGUA_IN) == true ){
                if (READBIT(Estado, DOSIFICAR_AGUA1_ST) == false){
                    SETBIT(Estado, DOSIFICAR_AGUA1_ST, true);
                    SETBIT(Estado, DOSIFICAR_AGUA2_ST, false);
                    Serial1.println("Init T1 Agua");
                }
            }
            else{
                SETBIT(Estado, DOSIFICAR_AGUA1_ST, false);
                SETBIT(Estado, DOSIFICAR_AGUA2_ST, false);
                SETBIT(Salidas, BOMBA_AGUA, false);
            }
        }
    }
}

```

```

        Serial1.println("Fin T1 Agua");
    }
}
//TANQUE 1 ALCOHOL
if (LeerAccionBoton(DOSIFICAR_ALCOHOL_IN) == true ){
    if (READBIT(Estado, DOSIFICAR_ALCOHOL1_ST) == false){
        SETBIT(Estado, DOSIFICAR_ALCOHOL1_ST, true);
        SETBIT(Estado, DOSIFICAR_ALCOHOL2_ST, false);
        Serial1.println("Init T1 Alcohol");
    }
    else{
        SETBIT(Estado, DOSIFICAR_ALCOHOL1_ST, false);
        SETBIT(Estado, DOSIFICAR_ALCOHOL2_ST, false);
        SETBIT(Salidas, RUN_BOMBA_ALCOHOL, false);
        SETBIT(Salidas, STOP_BOMBA_ALCOHOL, false);
        Serial1.println("Fin T1 Alcohol");
    }
}
}
if (READBIT (ParametrosPLC[36], SYS_ENABLE_TANQUE1) == false &&
READBIT (ParametrosPLC[36], SYS_ENABLE_TANQUE2) == true){
    //TANQUE 2 AGUA
    if (LeerAccionBoton(DOSIFICAR_AGUA_IN) == true ){
        if (READBIT(Estado, DOSIFICAR_AGUA2_ST) == false){
            SETBIT(Estado, DOSIFICAR_AGUA1_ST, false);
            SETBIT(Estado, DOSIFICAR_AGUA2_ST, true);
            Serial1.println("Init T2 Agua");
        }
        else{
            SETBIT(Estado, DOSIFICAR_AGUA1_ST, false);
            SETBIT(Estado, DOSIFICAR_AGUA2_ST, false);
            SETBIT(Salidas, BOMBA_AGUA, false);
            Serial1.println("Fin T2 Agua");
        }
    }
}

//TANQUE 2 ALCOHOL
if (LeerAccionBoton(DOSIFICAR_ALCOHOL_IN) == true ){
    if (READBIT(Estado, DOSIFICAR_ALCOHOL2_ST) == false){
        SETBIT(Estado, DOSIFICAR_ALCOHOL1_ST, false);
        SETBIT(Estado, DOSIFICAR_ALCOHOL2_ST, true);
        Serial1.println("Init T2 Alcohol");
    }
    else{
        SETBIT(Estado, DOSIFICAR_ALCOHOL1_ST, false);
        SETBIT(Estado, DOSIFICAR_ALCOHOL2_ST, false);
        SETBIT(Salidas, RUN_BOMBA_ALCOHOL, false);
        SETBIT(Salidas, STOP_BOMBA_ALCOHOL, false);
        Serial1.println("Fin T2 Alcohol");
    }
}
}
//MIXER1
//SELECTOR INICIO MIXER TANQUE 1
if (LeerAccionBoton(MIXER1_IN) == true ){// CUANDO SELECTOR VA A
ON
    if (READBIT(Estado, MEZCLAR1_ST) == false){
        SETBIT(Estado, MEZCLAR1_ST, true);
    }
    else{

```

```

        SETBIT(Estado, MEZCLAR1_ST, false);
        SETBIT(Salidas, MIXER1, false);
    }
}
//MIXER2
//SELECTOR INICIO MIXER TANQUE 2
if (LeerAccionBoton(MIXER2_IN) == true ){
    if (READBIT(Estado, MEZCLAR2_ST) == false){
        SETBIT(Estado, MEZCLAR2_ST, true);
    }
    else{
        SETBIT(Estado, MEZCLAR2_ST, false);
        SETBIT(Salidas, MIXER2, false);
    }
}
}
else{
    SETBIT(Estado, DOSIFICAR_ALCOHOL1_ST, false);
    SETBIT(Estado, DOSIFICAR_ALCOHOL2_ST, false);
    SETBIT(Estado, DOSIFICAR_AGUA1_ST, false);
    SETBIT(Estado, DOSIFICAR_AGUA2_ST, false);
    SETBIT(Estado, MEZCLAR1_ST, false);
    SETBIT(Estado, MEZCLAR2_ST, false);

    SETBIT(Salidas, BOMBA_AGUA, false);
    SETBIT(Salidas, RUN_BOMBA_ALCOHOL, false);
    SETBIT(Salidas, STOP_BOMBA_ALCOHOL, false);
    SETBIT(Salidas, MIXER1, false);
    SETBIT(Salidas, MIXER2, false);
}
if (READBIT(EstadoTimers, TIMER_TimSirenaFinDosif) == True){
    SETBIT(EstadoTimers, TIMER_TimSirenaFinDosif, False);
    SETBIT(Salidas, SIRENA_ALARMA, False);
}
}
}

void Protecciones (void){
    //PROTECCION BOMBA AGUA
    if (READBIT(Salidas, BOMBA_AGUA) == True && READBIT(Entradas,
PROTECCION_B_AGUA_IN) == True){
        if (TimerProteccionesOn[BOMBA_AGUA] != 20)
            TimerProteccionesOn[BOMBA_AGUA]++;
        }
        else{
            TimerProteccionesOn[BOMBA_AGUA]=0;
        }
        if (TimerProteccionesOn[BOMBA_AGUA] >= 20){
            SETBIT(Alarmas, FALLA_ProteccionAgua, True);
            SETBIT(Salidas, SIRENA_ALARMA, True);
            SETBIT(Salidas, BOMBA_AGUA, false);
        }
    }
    //PROTECCION MIXER 1
    if (READBIT(Salidas, MIXER1) == True && READBIT(Entradas,
PROTECCION_MIXER1_IN) == True){
        if (TimerProteccionesOn[MIXER1] != 20)
            TimerProteccionesOn[MIXER1]++;
        }
        else{
            TimerProteccionesOn[MIXER1]=0;
        }
    }
}

```

```

    if (TimerProteccionesOn[MIXER1] >= 20){
        SETBIT(Alarmas, FALLA_ProteccionMixer1, True);
        SETBIT(Salidas, SIRENA_ALARMA, True);
        SETBIT(Salidas, MIXER1, false);
    }
    //PROTECCION MIXER 2
    if (READBIT(Salidas, MIXER2) == True && READBIT(Entradas,
PROTECCION_MIXER2_IN) == True){
        if (TimerProteccionesOn[MIXER2]!=20)
            TimerProteccionesOn[MIXER2]++;
        }
    else{
        TimerProteccionesOn[MIXER2]=0;
    }
    if (TimerProteccionesOn[MIXER2] >= 20){
        SETBIT(Alarmas, FALLA_ProteccionMixer2, True);
        SETBIT(Salidas, SIRENA_ALARMA, True);
        SETBIT(Salidas, MIXER2, false);
    }
    //REMOTO
    //PROTECCION ALCOHOL
    if (READBIT(Salidas, RUN_BOMBA_ALCOHOL) == True &&
READBIT(Entradas, PROTECCION_B_ALCOHOL_IN) == False){
        if (TimerProteccionesOn[RUN_BOMBA_ALCOHOL]!=30)
            TimerProteccionesOn[RUN_BOMBA_ALCOHOL]++;
        }
    else{
        TimerProteccionesOn[RUN_BOMBA_ALCOHOL]=0;
    }
    if (TimerProteccionesOn[RUN_BOMBA_ALCOHOL] >= 30){
        SETBIT(Alarmas, FALLA_ProteccionAlcohol, True);
        SETBIT(Salidas, SIRENA_ALARMA, True);
        SETBIT(Salidas, RUN_BOMBA_ALCOHOL, false);
    }
}
}
uint32_t TimerCheckProteccion=0;
void ProcesoTask (void){
    ResetAlarmas();
    DosificarAgua1();
    DosificarAgua2();
    DosificarAlcohol1();
    DosificarAlcohol2();
    Mezclar1();
    Mezclar2();
    Condicionantes();
    if (millis()>TimerCheckProteccion){
        TimerCheckProteccion = millis()+100;
        Protecciones();
    }
}

void TIM6_Config(void){
    TIM_TypeDef *Instance = TIM6;
    HardwareTimer *MyTim = new HardwareTimer(Instance);
    MyTim->setOverflow(100, HERTZ_FORMAT); // 10 mS
    MyTim->attachInterrupt(TIM6_IT_callback);
    MyTim->resume();
}
void TIM6_IT_callback(void){

```

```

    IOTask();
    Ticks10ms++;
    ManejadorTimers();
}

```

Anexo 8 – Programación del ESP6266 Tanque 5000 L

```

#include <SoftwareSerial.h>
#define BAUD_RATE 38400
SoftwareSerial testSerial;

#include <ESP8266WiFi.h>

float DistanciaAnterior1[20];
float Distancia[1];
#define UPDATE_TIME 1000
String nombre = "Tanque A";
char ssid[] = "Laboratorio";
char pass[] = "Cartago2020";
//char ssid[] = "RED FRANCISCO";
//char pass[] = "0002427491";

unsigned long previousRequest = 0;
uint32_t previousRequest2 = 0;
//Objects
WiFiClient master;
IPAddress server(192, 168, 88, 180);
//IPAddress server(192, 168, 100, 180);

void setup_wifi (void){
    WiFi.begin(ssid, pass);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        //Serial.print(F("."));
    }
}

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);
    pinMode(4, OUTPUT);
    //testSerial.begin(BAUD_RATE, SWSERIAL_8N1, 12, 14);
    //testSerial.println("init Serial Rx Test");

    //Init ESP8266 Wifi
    setup_wifi();
    //Serial.print(nombre);
    //Serial.print(F(" connected to Wifi! IP address : "));
    //Serial.println(WiFi.localIP()); // Print the IP address

}

void loop() {
    if (WiFi.status() != WL_CONNECTED) {
        setup_wifi(); //Revisa que este coinectado a tu servidor
    }
    requestMaster();
    LeerSensor();
}

```

```

}
void requestMaster( ) { /* function requestMaster */
  ///Request to master
  if ((millis() - previousRequest) > UPDATE_TIME) { // client
connect to server every 1000ms
  //Serial.print("Promedio: ");Serial.println(Distancia[0]);
  previousRequest = millis();
  if (master.connect(server, 80)) { // Connection to the server
  //ENVIAR INFORMACION DE ENTRADA
  master.print("millis: " + String(millis()/1000) + " ");
  /*master.print(RxBuff[0]);master.print(",");
  master.print(RxBuff[1]);master.print(",");
  master.print(RxBuff[2]);master.print(",");
  master.print(RxBuff[3]);master.println();*/
  float cmDist=Distancia[0]/10;
  master.println(nombre + ":Dist1:" + String(cmDist) +
"END\r");
  // RESPUESTA
  //String answerA = master.readStringUntil('\r'); //
receives the answer from the sever
  //master.flush();
  //Serial.println(answerA);
  }
}
}
uint32_t Timm32;
void LeerSensor(void) {
  if (Serial.available()) {
    delay(5);
    if (millis()>Timm32) {
      Timm32=millis()+4000;
      digitalWrite(4, !digitalRead(4));
    }
    //testSerial.println("Data");
    uint8_t RxBuff[4];
    uint8_t pBff=0, TimeOut=5;
    while (Serial.available() && TimeOut>0) {
      RxBuff[pBff]=Serial.read();
      if (pBff<3)
        pBff++;
      if (TimeOut)
        TimeOut--;
      //delay(1);
    }
    //testSerial.print(RxBuff[0]);testSerial.print(",");
    //testSerial.print(RxBuff[1]);testSerial.print(",");
    //testSerial.print(RxBuff[2]);testSerial.print(",");
    //testSerial.print(RxBuff[3]);testSerial.println();
    //testSerial.print(pBff);
    if (RxBuff[0] == 0xFF && pBff == 3){
      uint16_t Suma=RxBuff[0]+RxBuff[1]+RxBuff[2], Data =
RxBuff[1]<<8|RxBuff[2];
      uint8_t MyCRC8=Suma;
      //Serial.println(Data);
      //if (MyCRC8 == RxBuff[3]){
      if (Data > 400) {
        //PROMEDIO ESPECIAL
        DistanciaAnterior1[15]= DistanciaAnterior1[14];
        DistanciaAnterior1[14]= DistanciaAnterior1[13];
        DistanciaAnterior1[13]= DistanciaAnterior1[12];

```

```

        DistanciaAnterior1[12]= DistanciaAnterior1[11];
        DistanciaAnterior1[11]= DistanciaAnterior1[10];
        DistanciaAnterior1[10]= DistanciaAnterior1[9];
        DistanciaAnterior1[9]= DistanciaAnterior1[8];
        DistanciaAnterior1[8]= DistanciaAnterior1[7];
        DistanciaAnterior1[7]= DistanciaAnterior1[6];
        DistanciaAnterior1[6]= DistanciaAnterior1[5];
        DistanciaAnterior1[5]= DistanciaAnterior1[4];
        DistanciaAnterior1[4]= DistanciaAnterior1[3];
        DistanciaAnterior1[3]= DistanciaAnterior1[2];
        DistanciaAnterior1[2]= DistanciaAnterior1[1];
        DistanciaAnterior1[1]= DistanciaAnterior1[0];
        DistanciaAnterior1[0]= Data;

        Distancia[0] += DistanciaAnterior1[0];
        Distancia[0] += DistanciaAnterior1[1];
        Distancia[0] += DistanciaAnterior1[2];
        Distancia[0] += DistanciaAnterior1[3];
        Distancia[0] += DistanciaAnterior1[4];
        Distancia[0] += DistanciaAnterior1[5];
        Distancia[0] += DistanciaAnterior1[6];
        Distancia[0] += DistanciaAnterior1[7];
        Distancia[0] += DistanciaAnterior1[8];
        Distancia[0] += DistanciaAnterior1[9];
        Distancia[0] += DistanciaAnterior1[10];
        Distancia[0] += DistanciaAnterior1[11];
        Distancia[0] += DistanciaAnterior1[12];
        Distancia[0] += DistanciaAnterior1[13];
        Distancia[0] += DistanciaAnterior1[14];
        Distancia[0] += DistanciaAnterior1[15];

        Distancia[0]/=17;
        //testSerial.println(Distancia[0]);
    } //DATA > 400
    //}
}
}

```

Anexo 9 – Programación del ESP6266 Tanque 10000 L

```

#include < ESP8266WiFi . h >

float DistanciaAnterior1 [ 20 ] ;
float Distancia [ 1 ] ;
#define UPDATE_TIME 1000
String nombre = "Tanque B" ;
char ssid [ ] = "Laboratorio" ;
char pase [ ] = "Cartago2020" ;
//char ssid[] = "RED FRANCISCO";
// pase de caracteres [ ] = "0002427491";

solicitud anterior larga sin firmar = 0 ; uint32_t solicitud
anterior2 = 0 ; //Objetos WiFiClient master ; servidor de dirección

```

```

IP ( 192 , 168 , 88 , 180 ) ; //servidor IPAddress(192, 168, 100,
180);

vacío setup_wifi ( vacío ) {
  WiFi . empezar ( ssid , pase ) ;
  while ( WiFi . estado ( ) != WL_CONNECTED ) {
    retraso ( 500 ) ;
    //Serial.print(F("."));
  }
}

void setup ( ) {
  // pon tu código de configuración aquí, para ejecutar una vez:
  Serial . comenzar ( 9600 ) ;
  pinMode ( 4 , SALIDA ) ;
  //testSerial.begin(BAUD_RATE, SWSERIAL_8N1, 12, 14);
  //testSerial.println("init Serial Rx Test");

  //Iniciar ESP8266 Wifi
  setup_wifi ( ) ;
  //Serial.print(nombre);
  //Serial.print(F(" ;conectado a Wifi! Dirección IP: "));
  //Serial.println(WiFi.localIP()); // Imprimir la dirección IP
}

void loop ( ) {
  if ( WiFi . estado ( ) != WL_CONNECTED ) {
    setup_wifi ( ) ; //Revisa que este coinectado a tu servidor
  }
  requestMaster ( ) ;
  Leer Sensor ( ) ;
}

void requestMaster ( ) { /* function requestMaster */
  ////Solicitud de maestro
  if ( ( millis ( ) - solicitud anterior ) > UPDATE_TIME ) {
  // el cliente se conecta al servidor cada 1000ms
  //Serial.print("Promedio: ");Serial.println(Distancia[0]);
  solicitud anterior = milisegundos ( ) ;
  if ( master . connect ( server , 80 ) ) { // Conexión al
servidor
  //ENVIAR INFORMACION DE ENTRADA
  master . print ( "millis: " + String ( millis ( ) / 1000
) + " " ) ;
  /*maestro.imprimir(RxBuff[0]);maestro.imprimir(",");
maestro.imprimir(RxBuff[1]);maestro.imprimir(",");
maestro.imprimir(RxBuff[2]);maestro.imprimir(",");
master.print(RxBuff[3]);master.println();*/
float cmDist = Distancia [ 0 ] / 10 ;
maestro _ println ( nombre + ":Dist2:" + String (
cmDist ) + "FIN\r" ) ;
  // RESPUESTA
  //String respuestaA = master.readStringHasta('\r'); //
recibe la respuesta del servidor
  //master.flush();
  //Serial.println(respuestaA);
}
}

```



```

    }
}
uint32_t Timm32 ;
void LeerSensor ( void ) {
    if ( Serial . available ( ) ) {
        delay ( 5 ) ;
        if ( millis ( ) > Timm32 ) {
            Timm32 = millis ( ) + 4000 ;
            escritura digital ( 4 , ! lectura digital ( 4 ) ) ;
        }
        //testSerial.println("Datos");
        uint8_t RxBuff [ 4 ] ; _ uint8_t pBff = 0 , Tiempo de espera = 5
; while ( Serial . disponible ( ) && TimeOut > 0 ) { RxBuff [ pBff ]
= Serial . leer ( ) ; si ( pBff < 3 ) pBff ++ ; si (

        Tiempo de espera )
        Tiempo de espera -- ;
        //retraso(1);
    }
    //testSerial.print(RxBuff[0]);testSerial.print(",");
    //testSerial.print(RxBuff[1]);testSerial.print(",");
    //testSerial.print(RxBuff[2]);testSerial.print(",");
    //testSerial.print(RxBuff[3]);testSerial.println();
    //testSerial.print(pBff);
    if ( RxBuff [ 0 ] == 0xFF && pBff == 3 ) {
        uint16_t Suma = RxBuff [ 0 ] + RxBuff [ 1 ] + RxBuff [ 2 ] ,
Datos = RxBuff [ 1 ] << 8 | RxBuff [ 2 ] ;
        uint8_t MiCRC8 = Suma ;
        //Serial.println(Datos);
        //if (MyCRC8 == RxBuff[3]){
            if ( Data > 400 ) {
                //PROMEDIO ESPECIAL
                DistanciaAnterior1 [ 15 ] = DistanciaAnterior1 [ 14 ] ;
                DistanciaAnterior1 [ 14 ] = DistanciaAnterior1 [ 13 ] ;
                DistanciaAnterior1 [ 13 ] = DistanciaAnterior1 [ 12 ] ;
                DistanciaAnterior1 [ 12 ] = DistanciaAnterior1 [ 11 ] ;
                DistanciaAnterior1 [ 11 ] = DistanciaAnterior1 [ 10 ] ;
                DistanciaAnterior1 [ 10 ] = DistanciaAnterior1 [ 9 ] ;
                DistanciaAnterior1 [ 9 ] = DistanciaAnterior1 [ 8 ] ;
                DistanciaAnterior1 [ 8 ] = DistanciaAnterior1 [ 7 ] ;
                DistanciaAnterior1 [ 7 ] = DistanciaAnterior1 [ 6 ] ;
                DistanciaAnterior1 [ 6 ] = DistanciaAnterior1 [ 5 ] ;
                DistanciaAnterior1 [ 5 ] = DistanciaAnterior1 [ 4 ] ;
                DistanciaAnterior1 [ 4 ] = DistanciaAnterior1 [ 3 ] ;
                DistanciaAnterior1 [ 3 ] = DistanciaAnterior1 [ 2 ] ;
                DistanciaAnterior1 [ 2 ] = DistanciaAnterior1 [ 1 ] ;
                DistanciaAnterior1 [ 1 ] = DistanciaAnterior1 [ 0 ] ;
                DistanciaAnterior1 [ 0 ] = Datos ;

                Distancia [ 0 ] += DistanciaAnterior1 [ 0 ] ;
                Distancia [ 0 ] += DistanciaAnterior1 [ 1 ] ;
                Distancia [ 0 ] += DistanciaAnterior1 [ 2 ] ;
                Distancia [ 0 ] += DistanciaAnterior1 [ 3 ] ;
                Distancia [ 0 ] += DistanciaAnterior1 [ 4 ] ;
                Distancia [ 0 ] += DistanciaAnterior1 [ 5 ] ;
            }
        }
    }
}

```

```
Distancia [ 0 ] += DistanciaAnterior1 [ 6 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 7 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 8 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 9 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 10 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 11 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 12 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 13 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 14 ] ;
Distancia [ 0 ] += DistanciaAnterior1 [ 15 ] ;

Distancia [ 0 ] /= 17 ;
//testSerial.println(Distancia[0]);
} //DATOS > 400
// }
}
}
```