



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE INGENIERÍA EN ELECTRÓNICA Y COMUNICACIONES

TEMA:

SISTEMA DE CONTROL Y ALERTA PARA EL TANQUE PURIFICADOR DE
AGUA EN LA PLANTA PURIFICADORA ECOAGUA

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la
obtención del Título de Ingeniero en Electrónica y Comunicaciones.

Sublínea de Investigación: Sistemas Embebidos.

Autor: Jessica Fabiola Moposita Moposita.

Tutor: Ing. MSc. Carlos Alberto Serra Jiménez.

**Ambato – Ecuador
[Septiembre, 2018]**

APROBACION DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el tema: **SISTEMA DE CONTROL Y ALERTA PARA EL TANQUE PURIFICADOR DE AGUA EN LA PLANTA PURIFICADORA ECOAGUA**, de la señorita Jessica Fabiola Moposita Moposita, estudiante de la Carrera de Ingeniería en Electrónica y Comunicaciones, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato septiembre, 2018

EL TUTOR



Ing. MSc. Carlos Alberto Serra Jiménez

AUTORÍA

El presente Proyecto de Investigación titulado: **SISTEMA DE CONTROL Y ALERTA PARA EL TANQUE PURIFICADOR DE AGUA EN LA PLANTA PURIFICADORA ECOAGUA**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato septiembre, 2018



Moposita Moposita Jessica Fabiola

CC: 1804899365

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, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato septiembre, 2018



Moposita Moposita Jessica Fabiola

CC: 1804899365

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Patricio Encalada e Ing. Patricio Córdova, revisó y aprobó el Informe Final del Proyecto de Investigación titulado: SISTEMA DE CONTROL Y ALERTA PARA EL TANQUE PURIFICADOR DE AGUA EN LA PLANTA PURIFICADORA ECOAGUA, presentado por el señorita Jessica Fabiola Moposita Moposita de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



Ing. Mg. Elsa Pilar Urrutia Urrutia

PRESIDENTA DEL TRIBUNAL



Ing. Mg. Patricio Encalada.

DOCENTE CALIFICADOR



Ing. Mg. Patricio Córdova.

DOCENTE CALIFICADOR

DEDICATORIA

A Dios, por haberme permitido llegar hasta este momento, bendiciéndome con salud y vida para alcanzar mis objetivos.

Al amor de mi vida, mi mami María quien día a día ha sabido brindarme su cariño, apoyo y comprensión; siendo el pilar fundamental en mi vida con sus consejos y valores inculcados, la misma quien me ha demostrado que se debe luchar por lo que se quiere.

A mi papi, Ignacio que sin su ayuda durante este largo camino no hubiese podido lograr esta gran meta.

A mi hermosa hija Aimee, quién fue, es y será un gran regalo en mi vida, con su cariño y alegría ilumina mi existencia, constituyéndose el motor que me impulsa a seguir adelante.

A mis queridos hermanos Jimmy y Marisol, quienes con sus consejos y palabras de aliento me ha enrumbado alcanzar mi sueño.

A Misael, quien me apoyado en los buenos y malos momentos, brindándome su amor y animándome a luchar por mi meta.

A mis sobrinos, Britney, Lessder y Dreyk quienes con su apoyo moral han conseguido que alcance mi ideal.

Moposita Moposita Jessica Fabiola

AGRADECIMIENTO

Agradezco enormemente a mi Dios y a mi Jesús del Gran Poder, quienes con su inmensa bondad han bendecido y guiado mi camino.

Agradezco a mis padres, hermanos, esposo, sobrinos y a mi querida hija, gracias por estar siempre pendiente y por el apoyo incondicional a alcanzar mis sueños.

A mi Tutor de Tesis el Ing. Carlos Serra, por la paciencia, apoyo y tiempo empleado para guiarme en el proceso de desarrollo de este proyecto de investigación.

A la Facultad de Ingeniería en Sistemas, Electrónica e Industrial y sus docentes por haber ayudado a formarme como persona y profesional.

Un agradecimiento especial a la Planta purificadora de agua ECOAGUA por abrirme sus puertas y permitir que desarrolle mi tema de investigación.

Finalmente, agradezco a toda mi familia y amigos cercanos por el apoyo y el granito de arena que aportaron para que sea una buena persona y alcance esta meta en mi vida

Moposita Moposita Jessica Fabiola

ÍNDICE

APROBACION DEL TUTOR	i
AUTORÍA	ii
DERECHOS DE AUTOR	iii
APROBACIÓN DE LA COMISIÓN CALIFICADORA	iv
DEDICATORIA	v
AGRADECIMIENTO	vi
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xiv
RESUMEN	xv
ABSTRACT	xvi
GLOSARIO DE TÉRMINOS Y ACRÓNIMOS	xvii
INTRODUCCIÓN	xix
CAPÍTULO I	1
EL PROBLEMA	1
1.1 TEMA DE INVESTIGACIÓN	1
1.2 PLANTEAMIENTO DEL PROBLEMA	1
1.3 DELIMITACIÓN	2
1.4 JUSTIFICACIÓN	3
1.5 OBJETIVOS	4
1.5.1 General	4
1.5.2 Específicos	4
CAPÍTULO II	5

MARCO TEÓRICO	5
2.1 ANTECEDENTES INVESTIGATIVOS.....	5
2.2 FUNDAMENTACIÓN TEÓRICA	7
2.2.1. La composición del Agua.	7
2.2.2. Sistemas de Automatización Industrial.....	10
2.2.3. Tipos de Control de Nivel.	13
2.2.4. Dispositivos en los Sistemas de Control Industrial.....	14
2.2.5. Monitorización en la Industria.	16
2.2.6. Interfaz Hombre-Máquina (HMI).	18
2.2.7. Hardware Libre, modelos de negocios para América Latina y El Caribe.....	20
2.3 PROPUESTA DE SOLUCIÓN	25
CAPÍTULO III.....	26
METODOLOGÍA	26
3.1 MODALIDAD DE LA INVESTIGACIÓN.....	26
3.2 RECOLECCIÓN DE INFORMACIÓN	26
3.3 PROCESAMIENTO Y ANÁLISIS DE DATOS	27
3.4 DESARROLLO DEL PROYECTO	27
CAPÍTULO IV	28
DESARROLLO DE LA PROPUESTA	28
4.1 INTRODUCCIÓN.....	28
4.2 ANÁLISIS DE FACTIBILIDAD	29
4.2.1. Factibilidad Técnica.....	29
4.2.2. Factibilidad Institucional	29

4.2.3.	Factibilidad Bibliográfica	29
4.2.4.	Factibilidad Económica	30
4.3	SITUACIÓN ACTUAL DE LA PLANTA PURIFICADORA ECOAGUA	30
4.4	REQUERIMIENTOS DEL SISTEMA	31
4.4.1	Hardware.....	32
4.4.2	Software para el diseño de la interfaz gráfica.	51
4.5	DISEÑO DEL SISTEMA PROPUESTO	54
4.6	DIAGRAMA GENERAL DEL SISTEMA PROGRAMADO	59
4.7	DESARROLLO DEL SISTEMA	64
4.7.1	Implementación del sistema de tuberías.	64
4.7.2	Implementación del sistema de Sensorización.....	69
4.7.3	Construcción del Tablero eléctrico industrial.	72
4.7.4	Construcción y configuración del sistema de control.....	81
4.7.5	Desarrollo del HMI del sistema de control y alerta para el tanque purificador de agua.....	85
4.8	PRUEBAS DE FUNCIONAMIENTO.....	88
4.9	ANÁLISIS DE RESULTADOS	98
4.10	ANÁLISIS ECONÓMICO DEL SISTEMA IMPLEMENTADO	106
4.10.1	Costos de Hardware y Software.	106
4.10.2	Costo inicial del Diseño.....	111
4.10.3	Costo total del sistema Implementado.....	111
CAPÍTULO V.....		112
CONCLUSIONES Y RECOMENDACIONES		112
5.1	CONCLUSIONES	112

5.2 RECOMENDACIONES	114
BIBLIOGRAFÍA.....	115
ANEXOS	123

ÍNDICE DE FIGURAS

Figura 2. 1: Composición del agua.	8
Figura 2. 2: Purificación de agua.	9
Figura 2. 3: Fases para la puesta en marcha de un proyecto de automatización.	12
Figura 2. 4: Control de nivel alto y bajo.	13
Figura 2.5: Arduino Mega 2560.	24
Figura 4. 1: Planta de tratamiento y envasado de agua para consumo humano ECOAGUA.	30
Figura 4. 2: Tanques de agua cruda y agua filtrada de ECOAGUA.	32
Figura 4. 3: Diagrama de conexión del sensor ultrasónico MB7369 para obtener una salida análoga, ancho de pulso y serial.	35
Figura 4. 4: Conexión del Sensor Ultrasónico MB7369 a la tarjeta de control.	36
Figura 4. 5: Electroválvulas de ½” y de ¾”	36
Figura 4. 6: Electroválvula de Membrana 2w-200-20.	38
Figura 4. 7: Transistor tip122.	38
Figura 4. 8: ZG3NC Relé de estado sólido.	40
Figura 4. 9: Diagrama electrónico del Circuito MAX232.	41
Figura 4. 10: Circuito estabilizador de voltaje.	41
Figura 4. 11: Arduino Mega 2560.	44
Figura 4. 12: Pines del conector DB9 Macho y Hembra.	45
Figura 4. 13: Pines del conector USB.	46
Figura 4. 14: Cable Convertidor USB a Serial RS232.	47
Figura 4. 15: Trama serial de 28 registros.	47
Figura 4. 16: Ventana IDLE de Python.	52
Figura 4. 17: Ventanas Shell y Editor del IDLE de Python.	54
Figura 4. 18: Plano del Sistema de control y alerta para el tanque purificador de agua.	55
Figura 4. 19: Diagrama secuencial Temporizado del Tanque 1.	56
Figura 4. 20: Diagrama secuencial Temporizado del Tanque 2.	58
Figura 4. 21: Diagrama general del sistema de control basado en Arduino.	59

Figura 4. 22: Diagrama de flujo control del sistema de llenado.....	63
Figura 4. 23: Secciones de tuberías a modificarse en ECOAGUA.	64
Figura 4. 24: Diagrama Hidráulico del sistema de control y alerta según ISA 5.1.	65
Figura 4. 25: Tubería de ingreso de agua de la red pública hacia los tanques de agua cruda, sección 1.....	66
Figura 4. 26: Conexión de tuberías de ingreso de líquido para los tanques de agua cruda, sección 1.....	66
Figura 4. 27: Tubería deteriorada sección 2.	67
Figura 4. 28: Nueva Tubería sección 2.....	67
Figura 4. 29: Tubería deteriorada sección 3.	68
Figura 4. 30: Nueva Tubería sección 3.....	68
Figura 4. 31: Lugares para la implementación de Sensores.	69
Figura 4. 32: Circuito para evitar averías en los Sensores.....	69
Figura 4. 33: Conexión del sensor ultrasónico.	70
Figura 4. 34: Diagrama de calibración del sensor ultrasónico.....	71
Figura 4. 35: Sensores MB7369 HRXL-MaxSonar-WRM instalados.	71
Figura 4. 36: Gabinete Metálico.	72
Figura 4. 37: Tablero principal de distribución.	73
Figura 4. 38: Tablero eléctrico interno.	74
Figura 4. 39: Borneras y fuentes de paso superiores.	76
Figura 4. 40: Borneras inferiores.	76
Figura 4. 41: Indicadores del tablero eléctrico frontal.....	77
Figura 4. 42: Diagrama de conexión e instalación de la interfaz HMI.....	79
Figura 4. 43: Trama serial de 26 registros.	80
Figura 4. 44: Diagrama Electrónico de la tarjeta de control.....	82
Figura 4. 45: Instalación de la Tarjeta de Control.	83
Figura 4. 46: Ventana de inicio, selección de puerto de comunicación y pantalla principal del sistema de control y alerta para el tanque purificador de agua en la Planta Purificadora ECOAGUA.	87
Figura 4. 47: Información visualizada en la pantalla LCD.....	88
Figura 4. 48: Pantalla principal de la Interfaz Hombre-Máquina.....	89

Figura 4. 49: Ventanas de los botones Gráfica Tanque 1 y Tanque 2.	90
Figura 4. 50: Ventanas de gráfica anterior tanque 2.	91
Figura 4. 51: Ventanas de los botones de Eventos y Alarmas	91
Figura 4. 52: Proceso de vaciado de tanques.	92
Figura 4. 53: Producción Automática.	93
Figura 4. 54: Tiempo de Reposo del tanque de agua Cruda.	94
Figura 4. 55: Proceso de retrolavado de los filtros de arena-antracita y carbón activo. .	94
Figura 4. 56: Sistema Detenido.....	95
Figura 4. 57: Sistema Detenido-Falla Sensor Uno.	96
Figura 4. 58: Sistema Detenido-Falla Sensor Dos.....	96
Figura 4. 59: Nivel crítico tanque uno y dos.....	97
Figura 4. 60: Paro de Emergencia.....	98
Figura 4. 61: Llenado de tanques de agua cruda y filtrada anteriormente.....	99
Figura 4. 62: Implementación del sistema de control y alerta para el tanque purificador de agua en la planta purificadora ECOAGUA.	100
Figura 4. 63: Monitoreo del Nivel de líquido del tanque de agua cruda.	101
Figura 4. 64: Monitoreo del Nivel de líquido del tanque agua purificada.	102

ÍNDICE DE TABLAS

Tabla 4. 1: Sensor de presencia o proximidad.	33
Tabla 4. 2: Selección del Sensor Ultrasónico.	34
Tabla 4. 3: Selección de actuador.	37
Tabla 4. 4: Selección de actuador para el accionamiento de las Electroválvulas.	39
Tabla 4. 5: Selección Dispositivo que funcionará como PLC.	42
Tabla 4. 6: Descripción de los bytes que conforman la Trama que recibe la PC.	48
Tabla 4. 7: Selección del dispositivo de visualización.....	49
Tabla 4. 8: Motor trifásico P55CS-1247.....	50
Tabla 4. 9: Características de la computadora.	51
Tabla 4. 10: Selección del software.	51
Tabla 4. 11: Variables del Sistema.....	60
Tabla 4. 12: Parámetros de Configuración.....	60
Tabla 4. 13: Conexión de sensor MB7369 HRXL-MaxSonar con el cable UTP.	70
Tabla 4. 14: Conexión de cables.	73
Tabla 4. 15: Conexión de las Borneras Superiores.	75
Tabla 4. 16: Conexión de los sensores en las Borneras inferiores.....	76
Tabla 4. 17: Funciones que desempeñan los indicadores del tablero eléctrico frontal ...	78
Tabla 4. 18: Descripción de los bytes que conforman la Trama que llega a la LCD.....	81
Tabla 4. 19: Librerías empleadas en la programación de Python.	86
Tabla 4. 20: Indicadores de la Guía GEDIS.....	87
Tabla 4. 21: Tiempo de llenado y vaciado del tanque 1.	102
Tabla 4. 22: Tabla de Resultado Llenado del Tanque 1.....	103
Tabla 4. 23: Análisis de fallas de los principales componentes del sistema.	105
Tabla 4. 24: Costos de hardware y software de dispositivos industriales.....	107
Tabla 4. 25: Costos de hardware y software de componentes Electrónicos.	109
Tabla 4. 26: Costos total de hardware y software.	110
Tabla 4. 27: Costos total de implementación del sistema.	111

RESUMEN

El presente trabajo de investigación se realizó para solventar una falencia en la planta ECOAGUA, misma que carecía de un adecuado sistema de control de llenado en los tanques de agua, lo que provocaba que se ponga en riesgo sus instalaciones y los equipos como bombas por desbordes en los tanques, todo esto provocaba grandes retrasos en la producción y elevados costos de operación.

Se implementó un sistema de control automático armado y colocado en un tablero de control montado cerca del sistema de tanques a controlar, ésta tarjeta electrónica fue diseñada a partir de componentes electrónicos de uso local, debidamente programados y adecuados eléctricamente con el propósito de la automatización, la misma que está diseñada para que soporte la mayor cantidad de ruido eléctrico y perturbaciones de voltaje presentes en el entorno industrial, aumentado la fiabilidad del sistema considerablemente, como medida de protección la tarjeta cuenta con un sistema de reinicio automático en el caso de falla salvaguardando el proceso.

El resultado que se obtuvo es el manejo totalmente automático del sistema de llenado con una gestión de programación basada en la experiencia y el manejo de los operarios, mismo que permiten que el sistema trabaje de forma autónoma en función de la selección del modo de funcionamiento por parte del usuario, con lo cual abre y cierra válvulas en el momento adecuado, protegiendo principalmente las motobombas que son las que se colocan en riesgo por falla al trabajar sin agua, además de permitir registrar el consumo y presentarlos en un sistema Scada, con lo que los operarios pueden planificar con mayor eficiencia la producción.

Palabras claves: Arduino, Sensor Ultrasónico, Control, HMI, Comunicación serial, Sistema de Llenado, Scada.

ABSTRACT

This research work was performed to resolve a shortcoming in the plant ECOAGUA, same that lacked an adequate system of control of filling in water tanks, which caused that put at risk their facilities and equipment such as pumps by excesses in the tanks, all this was causing major delays in production and high costs of operation.

Implemented an automatic control system armed and placed in a control board mounted near the tank system to control, this electronic card was designed on the basis of electronic components for local use, duly programmed and suitable electrically with the purpose of automation, The same one that is designed to support the largest amount of electrical noise and voltage disturbances present in the industrial environment, increased the reliability of the system considerably, as a measure of protection the card has a system of automatic restart in the event of a failure while safeguarding the process.

The result obtained is handling the fully automatic filling system with a scheduling management based on experience and the management of the operators, which enable the system to operate autonomously in function of the operating mode selection by the user, With which opens and closes the valves in the right time, protecting the motor pumps that are mainly those that are placed at risk by fails to work without water, in addition to allow recording the consumption and present them in a Scada system, with what operators can plan more effectively the production.

Keywords: Arduino, Ultrasonic Sensor, Control, HMI, Serial Communication, Filling System, Scada.

GLOSARIO DE TÉRMINOS Y ACRÓNIMOS

- **BSD:** Berkeley Software Distribution - Distribution de software Berkeley.
- **CCM:** Control Centers Low Voltage Motors - Centros de Control de Motores de Bajo Voltaje.
- **Controlador PID:** Proportional Integral Derivative Controller - Controlador Proporcional-Integral-Derivativo.
- **DCS:** Distributed Control System - Sistemas Distribuidos de Control.
- **DDE:** Dynamic Data Exchange - Tecnología de Comunicación entre aplicaciones Windows y OS/2
- **EEPROM:** Electrically Erasable Programmable Read-Only Memory – Memoria de solo lectura programable y borrrable eléctricamente.
- **GSM:** Global System for Mobile Communications - Sistema global para las comunicaciones móviles
- **HMI:** Human Machine Interface - Interfaz Hombre Máquina.
- **ICSP:** In-Circuit Serial Programming - Programación serial en circuito.
- **IEEE:** Institute of Electrical and Electronics Engineers - Instituto de Ingeniería Eléctrica y Electrónica.
- **IED:** Intelligent Electronic Device - Dispositivo Electrónico Inteligente.
- **IPC:** Industrial Personal Computer - Ordenadores Personales Industriales.
- **LCD:** Liquid Crystal Display - Pantalla de cristal líquido.
- **LGPL:** GNU Lesser General Public License - Licencia Pública General Reducida de GNU.
- **OPC:** Ole for Process Control - Estándar de comunicación en el campo del control y supervisión de procesos industriales.
- **OSHWA:** Open Source Hardware Association - Asociación de hardware de código abierto.
- **OSI:** Open System Interconnection - Modelo de interconexión de sistemas abiertos.
- **PAC:** Programmable Automation Controller - Controlador Automático Programable.
- **PLC:** Programmable Logic Controller - Controlador Lógico Programable.

- **PVC:** Polyvinyl chloride - Policloruro de Vinilo.
- **PWM:** Pulse Width Modulation - Modulación por ancho de pulso.
- **RAM:** Random Access Memory - Memoria de acceso aleatorio.
- **RTU:** Remote Terminal Unit - Unidad Terminal Remota.
- **SCADA:** Supervisory Control and Data Acquisition - Supervisión, Control y Adquisición de Datos.
- **SRAM:** Static Random Access Memory - Memoria estática de acceso aleatorio.
- **UART:** Universal Asynchronous Receiver-Transmitter - Transmisor-Receptor Asíncrono Universal.
- **USB:** Universal Serial Bus - Bus Universal en Serie.
- **UTP:** Unshielded Twisted Pair - Par trenzado no blindado.

INTRODUCCIÓN

El sistema de control y alerta para el tanque purificador de agua en la planta purificadora ECOAGUA, permite la optimización de tiempo y recursos al contar con procesos automatizados que evitan el desperdicio del agua. El proyecto consiste en la construcción de una tarjeta electrónica basada en un microcontrolador el mismo que es bajamente inmune al ruido electromagnético, este permite adquirir datos de señales analógicas procedentes de los sensores y niveles de líquido, adicional genera una trama de información que sirve para el monitoreo del sistema mediante una HMI de computador y control por medio del tablero eléctrico del funcionamiento de los sensores y actuadores involucrados en este sistema.

El proyecto de investigación se estructura en cinco capítulos que se detallan brevemente a continuación:

En el capítulo uno, se detalla la problemática existente en la Planta ECOAGUA, al no contar con un sistema automatizado de control del llenado de sus contenedores, se establece la solución al problema presente, la justificación que sustenta el desarrollo del proyecto, los objetivos que al finalizar debe ser metas cumplidas, el período y lugar donde se lleva a cabo la investigación.

En el capítulo dos, se recopila información sobre antecedentes investigativos similares al proyecto, así también se desarrolla la fundamentación teórica referentes al sistema.

El capítulo tres presenta la modalidad de investigación, los tipos de investigación utilizados, la técnica, procesamiento y análisis de la información recopilada, las actividades que se efectúan para el desarrollo del proyecto.

El capítulo cuatro describe de forma detallada el diseño e implementación del sistema de control y alerta para el tanque purificador de agua.

Finalmente, el capítulo cinco señala las conclusiones y recomendaciones adquiridas mediante la implementación del proyecto.

CAPÍTULO I

EL PROBLEMA

1.1 TEMA DE INVESTIGACIÓN

SISTEMA DE CONTROL Y ALERTA PARA EL TANQUE PURIFICADOR DE AGUA EN LA PLANTA PURIFICADORA ECOAGUA.

1.2 PLANTEAMIENTO DEL PROBLEMA

En Ecuador el desperdicio del agua es alto en comparación a otros países. La media está en el 70% que debe ser totalmente utilizada, es la media latinoamericana. En países europeos es del 90% o 95%. En el Ecuador normalmente se da a la inversa esta ecuación se usa el 30% y se desperdicia el 70% de agua. Es un derecho humano el agua, pero lo que no es derecho es desperdiciar el agua para futuras generaciones. El artículo 140 de la Ley de Recursos Hídricos establece que todo exceso será cobrado al prestador [1]. El LORHUAA (Ley Orgánica de Recursos Hídricos, Usos y Aprovechamiento del Agua) prevé la introducción de un sistema tarifario para el agua cruda que sirva para incentivar uso eficiente del agua, desincentivando consumos excesivos, desperdicios, pérdidas operativas y otras y favoreciendo el ahorro y conservación del agua. En el caso del uso del agua para abastecimiento doméstico, la ley prevé el establecimiento de un consumo mínimo vital, con el objeto de definir los volúmenes a ser entregados a los prestadores de servicio libre de coste para poder hacer efectivo el derecho humano al agua. [2]

En Tungurahua es necesario racionalizar el consumo de agua. No basta con mejorar la oferta de los servicios de agua potable y saneamiento. Los consumos exagerados de agua, desperdicio, mal uso y la contaminación atentan contra el ejercicio pleno del derecho al agua que tienen todas las personas. La propuesta actualmente se está consensuando con la Asociación de Municipalidades del Ecuador (AME). Explicó que el problema del despilfarro del agua cruda es ambiental, ya que se estaría afectando a las fuentes de agua del país. Por ende, el estado cobrará a los Municipios el despilfarro del líquido. El titular de Senagua Carlos Bernal, manifestó que “mientras más se emplea el agua de manera ineficiente, también estamos contaminando más nuestras fuentes y ríos y se vuelve más caro el tratamiento del agua”. [3]

Actualmente la Planta Purificadora ECOAGUA presenta inconvenientes en el control del llenado y vaciado de sus tanques almacenadores, así que desperdician en gran cantidad el líquido o lo que es peor paralizan sus actividades, todo esto representa pérdidas para la empresa. De esta manera y en conversación con los directivos de empresa surge la necesidad de crear un sistema de control y alerta que garantice el correcto funcionamiento del tanque purificador de agua.

1.3 DELIMITACIÓN

Delimitación de Contenidos.

Área Académica: Física y Electrónica.

Línea de Investigación: Sistemas de Control.

Sublínea de Investigación: Sistemas Embebidos.

Delimitación Espacial.

La presente investigación se realizó en la Planta de tratamiento y envasado de agua para el consumo humano ECOAGUA ubicado en el caserío San Vicente de la parroquia de Quinchicoto del cantón Tisaleo.

Delimitación Temporal.

El proyecto de investigación se desarrolló durante el periodo Marzo-Agosto 2018, de acuerdo a lo establecido en el Reglamento de Graduación para Obtener el Título Terminal de Tercer nivel en la Universidad Técnica de Ambato.

1.4 JUSTIFICACIÓN

El agua es un elemento indispensable en la vida del ser humano; por esa razón, debe ser tratada y desarrollada con alta tecnología, siguiendo rigurosos estándares de calidad. [4] La implementación del sistema de control y alerta en el tanque purificador tiene gran importancia ya que protege la seguridad e infraestructura de la planta ECOAGUA, evitando la pérdida de agua, tiempo o recursos de la empresa. La empresa inició con esta actividad comercial en el año 2008, brindando este producto a todas las familias que conforman el cantón Tisaleo y sus alrededores. Por lo tanto, deben cubrir las necesidades que la población requiere, incorporando automatización dentro de los procesos para obtener un producto de manera rápida.

Los beneficiarios de forma directa constituyen el conjunto de personas (gerente, empleados, etc.) que forman parte de la planta ECOAGUA, ya que con la automatización del tanque se evitó errores por parte del operario destinado a este proceso, se pudo optimizar el tiempo de trabajo y realizar acciones preventivas gracias a la alerta que se incorpora. Adicionalmente se beneficiarán los clientes de manera indirecta ya que contarán con un producto garantizado y que cumple con los parámetros de calidad para el consumo humano.

Su desarrollo fue factible puesto que contaba con recursos bibliográficos actualizados, la tecnología necesaria, además de personal docente de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, concedor del tema, y principalmente con el aval del dueño de la Planta Purificadora. Adicionalmente los equipos y dispositivos electrónicos estuvieron disponibles en gran variedad dentro del

mercado ecuatoriano, se trabajó con software libre por las facilidades y ventajas que contiene, por tanto, el costo de la implementación fue relativamente bajo.

Mediante la automatización del tanque y el empleo de los sistemas embebidos se obtuvo un funcionamiento más robusto y a su vez confiable, de esta manera se pudo actuar de forma apropiada y aplicar soluciones preventivas a cualquier desperfecto.

1.5 OBJETIVOS

1.5.1 General

Implementar el sistema de control y alerta para el tanque purificador de agua en la planta purificadora ECOAGUA.

1.5.2 Específicos

- Analizar la situación actual del funcionamiento del tanque purificador de agua en la Planta ECOAGUA.
- Desarrollar una interfaz Hombre-Máquina (HMI) para el control del proceso en el tanque purificador de agua.
- Diseñar el sistema electrónico de control y alerta del tanque purificador de agua.

CAPÍTULO II

MARCO TEÓRICO

2.1 ANTECEDENTES INVESTIGATIVOS

Mediante la búsqueda de trabajos de titulación, investigaciones y artículos técnicos, se encontraron temas similares que sirvieron como apoyo para la presente investigación.

Ramiro Andrés Astudillo Bastidas, en su Tesis de Grado de la Universidad Tecnológica Israel, en el año 2016 desarrolló el “Diseño e implementación de un prototipo de medidor de nivel de agua a través de un sensor ultrasónico para pasos deprimidos”, tema que involucra el diseño e implementación de un prototipo de medidor de nivel de agua a través de un sensor ultrasónico para pasos deprimidos, el dispositivo cuenta con un módulo GSM que sirve para el envío de mensajes de texto a un número de emergencia cuando alcanza los tres niveles de agua previamente programados. Este sistema realiza un monitoreo diario de los niveles de agua en los pasos deprimidos, teniendo como solución un sistema de adquisición que permita recolectar las muestras del sensor de nivel a través de una memoria micro SD incorporada en su respectivo adaptador. [5]

En el año 2014, Manuel A. Méndez Robles y Pablo J. Erazo Silva de la Escuela Superior Politécnica de Chimborazo en su Tesis de Grado, presentan la “Implementación de un banco de pruebas para el control de nivel de líquidos en el laboratorio de instrumentación industrial de la Facultad de Mecánica”, este trabajo partió de una investigación minuciosa en el ámbito de los líquidos, sus propiedades, métodos e instrumentos de medida de nivel de líquido, válvulas, sistemas de control para su monitorización, se seleccionó los sensores

ultrasónicos por sus características y propiedades que se ajustan al proceso, se utilizó el software LabView aplicando una lógica de control fuzzy que facilita el monitoreo de las variables de control de nivel mediante el set-point de modo automático; además, el encendido y apagado se lo realiza de forma manual. [6]

Claudia Y. Peñaranda, Wilson R. Silva y Eulices Gómez, en su Tesis de Grado de la Universidad Industrial de Santander año 2014, realizaron la “Instrumentación y Control de nivel para un sistema de Tanques acoplados en el Laboratorio de Control e Instrumentación de la E3T-UIS”, allí se menciona la instrumentación y montaje de un sistema de tanques acoplados que permite regular el nivel del líquido en el tanque de salida empleando un controlador industrial. Se realizó el cálculo para los parámetros del sistema representado por resistencias y capacitancias hidráulicas, obteniendo un modelo para la función de transferencia del sistema, relacionando el caudal de entrada con respecto al nivel de tanque de salida; este modelo fue validado experimentalmente a partir de una respuesta al escalón. Con este modelo se sintonizaron los parámetros para un control PID empleado el primer método de Ziegler&Nichols. [7]

Joseph Vergara, Robinson Molina y César Machado, 2011, de la Revista de Ciencia y Tecnología INGENIUS en su artículo titulado “Prototipo para control automático de nivel y caudal de líquidos”, describen un prototipo que fue desarrollado para observar y controlar las variables de nivel y de flujo mediante una interfaz hombre-máquina (HMI). Se compone principalmente de una estructura diseñada para contener los elementos necesarios y del software de visualización y control. El objetivo del proyecto es encontrar lazos de control apropiados para el control efectivo de dos de las variables ampliamente utilizadas en la industria, nivel y flujo, con equipos que fácilmente pueden ser escalados hacia una operación industrial. [8]

En el año 2015, Luis A. Restrepo Álvarez y John de Jesús Cardona Salazar de la Universidad Tecnológica de Pereira, en su Tesis de Grado presentan el “Diseño de un sistema de control de nivel para la preparación de líquidos dializantes basado en señales ultrasónicas”, la propuesta de este diseño permite automatizar el llenado del tanque garantizando así la calidad de los líquidos dializantes y la optimización del servicio que

se presta en la entidad, se optó por utilizar un sensor ultrasónico ya que los líquidos dializantes no pueden entrar en contacto con agentes externos al tanque, para evitar pirógenos y bacterias que podrían afectar a los pacientes. Luego se procede a diseñar el control PID y simular la dinámica del sistema utilizando el software Matlab para definir los parámetros del sistema y la simulación. [9]

La información recopilada de los repositorios de las Universidades, artículos científicos y páginas destacadas del Internet sobre el control y alertas del vaciado o llenado de un tanque, son de gran importancia que servirán de pauta para continuar en el desarrollo del proyecto.

2.2 FUNDAMENTACIÓN TEÓRICA

2.2.1. La composición del Agua.

El agua, un compuesto extraordinariamente simple, es sin embargo una sustancia de características tan excepcionales y únicas que sin ella sería imposible la vida. El hombre tiene necesidad de agua para realizar sus funciones vitales, para preparar y cocinar los alimentos, para la higiene y los usos domésticos, para regar los campos, para la industria, para las centrales de energía: en una palabra, para vivir. Es una molécula sencilla formada por tres pequeños átomos, uno de oxígeno y dos de hidrógeno, con enlaces polares que permiten establecer puentes de hidrógeno entre moléculas adyacentes. [10]

El agua es una molécula sencilla formada por átomos pequeños, dos de hidrógeno y uno de oxígeno, unidos por enlaces covalentes muy fuertes que hacen que la molécula sea muy estable. Tiene una distribución irregular de la densidad electrónica, pues el oxígeno, uno de los elementos más electronegativos, atrae hacia sí los electrones de ambos enlaces covalentes, de manera que alrededor del átomo de oxígeno se concentra la mayor densidad electrónica (carga negativa) y cerca de los hidrógenos la menor (carga positiva). [10]

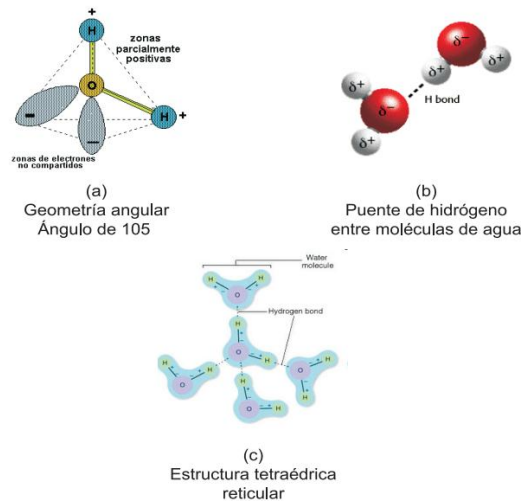


Figura 2. 1: Composición del agua. [10]

Proceso de purificación del Agua.

La pureza del agua es extremadamente importante en la industria farmacéutica y bioquímica, Las partículas disueltas o en suspensión, los compuestos orgánicos, las impurezas y otros contaminantes impiden utilizar agua corriente en aplicaciones de laboratorio e investigación científica. Se utilizan parámetros como la resistividad, la conductividad, el tamaño de partículas y la concentración de microorganismos para determinar la calidad del agua y, en consecuencia, especificar los usos previstos para el agua. Algunas aplicaciones toleran la presencia de determinadas impurezas en el agua, pero otras, como la cromatografía líquida de alto rendimiento (HPLC) exigen eliminar la mayoría de los contaminantes. [11]

Existen distintos métodos que se emplean habitualmente en la purificación de agua. Su efectividad depende del tipo de contaminante tratado y del tipo de aplicación en la que se va a utilizar el agua: [11]

a. Filtrado: este proceso puede consistir en cualquiera de los siguientes:

- **Filtrado grueso:** también llamado filtrado de partículas, puede utilizar desde un filtro de arena de 1 mm hasta un filtro de cartucho de 1 micra.

- **Microfiltrado:** utiliza dispositivos de 1 a 0,1 micras para filtrar las bacterias. Una implementación habitual de esta técnica se encuentra en el proceso de elaboración de cerveza.
- **Ultrafiltrado:** elimina pirógenos, endotoxinas y fragmentos de ADN y ADR.

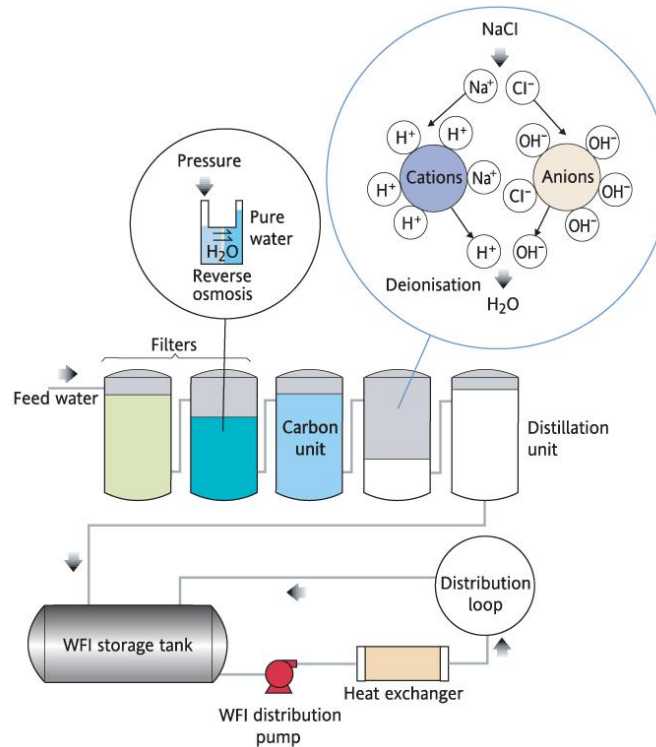


Figura 2. 2: Purificación de agua. [11]

- Ósmosis inversa (OI):** la ósmosis inversa es el grado más refinado de filtrado de líquidos. En lugar de un filtro, se utiliza un material poroso que actúa como tamiz unidireccional para separar partículas de tamaño molecular.
- Destilación:** el método de purificación más antiguo. Es un método asequible, pero no puede utilizarse para un proceso bajo demanda. El agua debe destilarse y, a continuación, almacenarse para usarse después, con lo que puede volver a contaminarse si no se almacena correctamente.
- Adsorción de carbono activado:** funciona como un imán para el cloro y los compuestos orgánicos.

- e. **Radiación ultravioleta:** con una longitud de onda determinada, permite esterilizar las bacterias y descomponer otros microorganismos.
- f. **Desionización:** también llamada intercambio de iones, se utiliza para producir agua purificada bajo demanda haciéndola pasar a través de capas de resina. La resina con carga negativa (catiónica) elimina los iones positivos, mientras que la resina con carga positiva (aniónica) elimina los iones negativos. La supervisión y el mantenimiento continuo de los cartuchos produce el agua más pura.

2.2.2. Sistemas de Automatización Industrial.

La automatización implica un grado de control mecánico que se inicia, comprueba y detiene por sí mismo. Siempre que se produce una desviación de un patrón establecido y proyectado (o planificado) de antemano, la máquina corrige esta desviación o impide que continúe el proceso, realizando la comprobación o detención con la máquina sin necesidad alguna de intervención humana. El termostato de uso doméstico es un ejemplo de dispositivo automático; sin precisar atención, inicia o interrumpe el flujo de calor para mantener una temperatura previamente determinada. [11]

La Real Academia de las Ciencias Físicas y Exactas define la automática como el conjunto de métodos y procedimientos para la sustitución del operario en tareas físicas y mentales previamente programadas. De esta definición original se desprende la definición de la automatización como la aplicación de la automática al control de procesos industriales. Por proceso, se entiende aquella parte del sistema en que, a partir de la entrada de material, energía e información, se genera una transformación sujeta a perturbaciones del entorno, que da lugar a la salida de material en forma de producto. Los procesos industriales se conocen como procesos continuos, procesos discretos y procesos batch. [12]

Las industrias relacionadas con la automatización son básicamente la industria manufacturera y la industria de procesos. La industria manufacturera (discrete parts manufacturing) se caracteriza por la presencia de máquinas herramienta de control numérico por ordenador como núcleo de sistemas de fabricación flexible. En esta

industria, destaca el uso de estaciones robotizadas en tareas de soldadura al arco o por puntos, pintura, montaje, etc., uno de los temas principales a resolver en este tipo de industria es la planificación y gestión de la producción: asignación de tareas a máquinas, diseño del layout de la planta, sistemas flexibles que fabriquen diversos productos, políticas de planificación cercanas a la optimización, etc. En cuanto a la industria de procesos (continuous manufacturing), existen fábricas de productos de naturaleza más o menos continua, como la industria petroquímica, cementera, de la alimentación, farmacéutica, etc. Dentro del proceso de fabricación de estas industrias, se investiga en nuevas tecnologías, para la obtención de nuevos catalizadores, bioprocesos, membranas para la separación de productos, microrreactores, etc. Es necesario la aplicación de algoritmos de control avanzado como, por ejemplo, el control predictivo, o la formación experta de operarios de salas de control mediante simuladores. [12]

Control de Procesos Industriales.

Esta abarca, desde un punto de vista académico, la teoría de control básica de realimentación y acción PID, la instrumentación de control (sensores, actuadores, dispositivos electrónicos, etc.), la aplicación a procesos industriales (como, por ejemplo, la mezcla de componentes en un reactor químico), las diversas arquitecturas de control (centralizado, distribuido), las estructuras de control (feedback, feedforward, cascada, etc.) y la teoría de control avanzada (control predictivo, control multivariable, etc.). [12]

Los algoritmos de control presentes en las industrias citadas, son el control secuencial y la regulación continua. El control secuencial propone estados (operaciones a realizar para la transformación de la materia prima en producto) y transiciones (información relativa a sensores o elementos lógicos como temporizadores o contadores) en una secuencia ordenada que identifica la evolución dinámica del proceso controlado. En la regulación continua, mediante la estructura de control clásica feedback, se aborda la acción de control proporcional, la acción de control derivativo o la acción de control integral, respecto al error (diferencia entre la consigna y la medida de la variable de salida del proceso) para conseguir así una regulación adecuada de la variable (temperatura, caudal, nivel, etc.). Respecto a instrumentación de control, los tres elementos básicos capaces de llevar a cabo

el control secuencial o la regulación continua dentro del control de procesos industriales son el llamado autómeta programable PLC. [12]

Fases para la puesta en marcha de un proyecto de Automatización.

Existen complejos procesos de automatización que requieren de la colaboración entre los diversos departamentos de una empresa (gestión, logística, automatización, distribución, etc.). En esta sección se enfoca el problema en concreto en la parte de automatización, desde el punto de vista del trabajo que debe realizar el ingeniero/ingeniera técnica. El marco metodológico consta de las fases siguientes, que el operario debe realizar: [12]

- Automatización
- Supervisión
- Interacción
- Implementación
- Pruebas

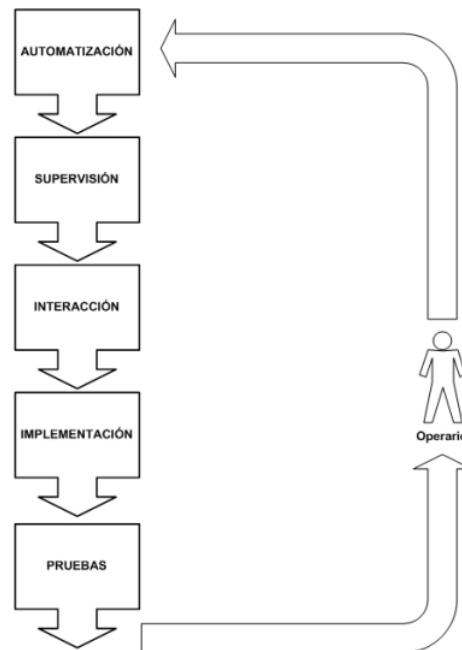


Figura 2. 3: Fases para la puesta en marcha de un proyecto de automatización. [12]

2.2.3. Tipos de Control de Nivel.

Debido a las diferentes características del proceso donde se tiene que efectuar control de nivel, esto puede llevarse a cabo de las siguientes maneras: [13]

- **Entre Límites alto y bajo:** Este sistema de control es suficientemente válido cuando el nivel puede oscilar entre ambos límites sin afectar al proceso. Se conoce como ON-OFF control y utiliza como elementos detectores de nivel dos levostatos situados en los límites superior e inferior del rango de nivel a controlar.
- **Nivel Medio:** Se usa cuando del recipiente sobre el cual se efectúa control de nivel puede absorber perturbaciones y, de esta forma, evitar que se trasladen aguas abajo del proceso. Se suelen conocer como averaging control y se lleva a cabo sintonizando el controlador de nivel con ganancia proporcional baja y tiempo integral alto.
- **Nivel Ajustado:** Se usa cuando la capacidad del recipiente es pequeña para el caudal de entrada y salida. En otras palabras, cuando el tiempo de residencia del producto en el recipiente es pequeño. Se suele conocer como tight control y se lleva a cabo sintonizando el controlador de nivel con ganancia relativamente alta y tiempo integral relativamente bajo.

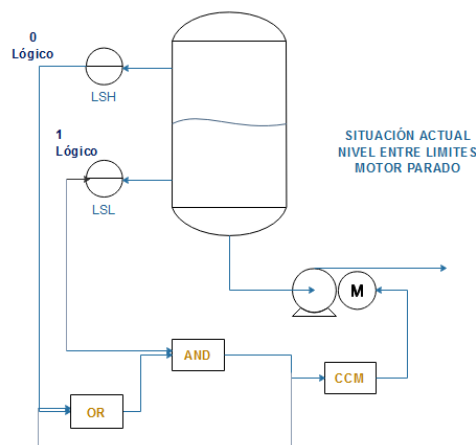


Figura 2. 4: Control de nivel alto y bajo. [13]

En la Figura 2.4 se muestra un sistema de control entre límites (ON-OFF) para desalojar el líquido que se va acumulando en un recipiente, partiendo de la situación actual, descrita

en la propia figura, cuando se alcance el nivel alto se cerrará el contacto LSH, pasando al nivel lógico 1, que junto con el otro de la puerta AND pone la bomba en marcha, La cual permanecerá marchando, retenida por la puerta OR, hasta que se alcance el nivel bajo detectado por el LSL. Pasando este a nivel 0 lógico. Ambas maniobras envían señal al centro control de motores (CCM), desde donde se envía la corriente eléctrica a la bomba. [13]

2.2.4. Dispositivos en los Sistemas de Control Industrial.

Los sistemas de control industrial se componen de multitud de dispositivos, un sistema pequeño cuenta con unas pocas variables y necesita pocos sensores y elementos de control, sin embargo, un gran sistema que tenga que procesar miles de variables debe incorporar un gran número de sensores, actuadores y multitud de dispositivos de control. [14]

Sensores y actuadores.

Los componentes del nivel más bajo de los sistemas de control también han ido evolucionando a lo largo de los años. En un principio los sensores y actuadores únicamente eran capaces de intercambiar su valor de estado (la propia variable), para después evolucionar y permitir comunicaciones, ya sea por un cable dedicado o bien por el mismo cable de alimentación. En la actualidad, estos equipos han evolucionado mucho más y los sensores y los actuadores se han convertido en dispositivos con grandes capacidades de comunicación a través de protocolos de nueva generación inalámbricos como ZigBee o WirelessHART, han incrementado sensiblemente la precisión necesaria para realizar su función. [14]

- **Sensor:** Bajo esta palabra de sensores se entiende tanto las unidades que emite una señal analógica, como las unidades que emite una señal binaria (encendido o apagado). En todos aquellos lugares donde no sea posible detectar magnitudes eléctricas se requiere los sensores. Convierte una magnitud física en una magnitud eléctrica. [15]

- **Actuador:** Es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o “actuar” otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). Dependiendo de el origen de la fuerza el actuador se denomina “neumático”, “hidráulico” o “eléctrico”. Los actuadores, en función de la energía que transformen se clasifican en: [16]
 - Eléctricos.
 - Neumáticos.
 - Hidráulicos.
 - Térmicos.
 - Elásticos.
 - Gravitatorios.
 - De origen animal.

Electroválvula.

Una electroválvula está compuesta por dos partes: Una cabeza magnética constituida principalmente por una bobina, tubo, culata, anillo de desfasado, resorte(s). Un cuerpo, con orificios de racordaje, obturados por clapet, membrana, pistón, etc. según el tipo de tecnología empleada. La apertura y el cierre de la electroválvula está unida a la posición del núcleo móvil que se desplaza bajo el efecto del campo magnético provocado por la puesta contención de la bobina. [17]

Pulsadores y Luces Piloto.

Son los dispositivos más sencillos del mundo, empleados para crear y romper un circuito, transmitir un mensaje o controlar una operación: en resumen, constituyen un nexo vital entre el hombre y la máquina. Un pulsador tiene que ser robusto y fiable, apto para cualquier entorno y adaptable a las más variadas exigencias. Además, tiene que ser fácil de montar y configurar. [18]

Controladores.

Tradicionalmente han existido dos tipos de dispositivos principales muy utilizados en los sistemas de control industrial: los PLCs (Programmable Logic Controller, Controlador Lógico Programable) y las RTUs (Remote Terminal Unit, Unidad Terminal Remota). Así mismo, equipos como los PACs (Programmable Automation Controller, Controlador Automático Programable) o los IEDs (Intelligent Electronic Device, Dispositivo Electrónico Inteligente) también son habituales en estos entornos. [14]

- **PLC:** Es un dispositivo desarrollado principalmente para recoger los datos de los sensores y mandar respuestas a los actuadores. Su trabajo se centra a nivel de campo y una de sus características más importantes es la velocidad a la que puede realizar su ciclo de operación (lectura de variables de entrada, ejecución del programa, escritura de las variables de salida). [14]
- **RTU:** Es un equipo con múltiples capacidades de comunicación. Su velocidad no es excesivamente alta, por tanto, no es útil para el control de procesos que requieran tomar decisiones rápidas; pero es capaz de trabajar con varios protocolos a la vez, lo que le permite comunicarse con muchos dispositivos, incluso de múltiples fabricantes, para intercambiar información entre ellos. [14]

2.2.5. Monitorización en la Industria.

La función de monitoreo de procesos dentro de la industria se realiza sobre un computador, ofreciendo una visión de los parámetros de control sobre la pantalla de ordenador, lo que se denomina un HMI (Human Machine Interface), como en SCADA, pero solo ofrecen una función complementaria de monitorización: observar mediante aparatos especiales el curso de uno o varios parámetros fisiológicos o de otra naturaleza para detectar posibles anomalías. Es decir, los sistemas de automatización de interfaz gráfica tipo HMI básicos ofrecen una gestión de alarmas básica, mediante las cuales la única opción que le queda al operario es realizar una parada de emergencia, reparar o compensar la anomalía y hacer un reset. Los sistemas SCADA utilizan un HMI interactivo que permite detectar alarmas y a través de la pantalla solucionar el problema mediante las acciones adecuadas en tiempo

real. Esto les otorga una gran flexibilidad. En definitiva, el modo supervisor del HMI de un SCADA no solo señala los problemas, sino que, lo más importante, orienta en cuanto a los procedimientos para solucionarlos. [19]

El seguimiento del proceso y de la evolución de forma continua del producto son los que permitirán alertar del buen o mal funcionamiento a la vez establecer criterios de ajuste y cambio dentro del proceso. Se entiende como monitorización a la automatización de este proceso de vigilancia dotando al operario de los mecanismos necesarios para su alerta, así como la interacción amigable con el proceso y el registro de su evolución. Su propósito es facilitar la detección de situaciones anómalas y su diagnóstico a través de un seguimiento continuo de las variables del proceso. Etapas de la monitorización. [20]

Adquisición y registro de Datos: La evolución de los ordenadores hacia sistemas más abiertos y modulares a base de expansiones de bus y tarjetas funcionales que permite el acceso a representaciones eléctricas de las magnitudes de proceso a propiciado la interconectividad del proceso con los sistemas de monitorización. Los autómatas programables (PLC, Programable Logic Computer), los sistemas distribuidos de control (DCS, Distributed Control System) y los ordenadores industriales (IPC, Industrial Personal Computer) han irrumpido en el control de procesos no solo como elementos de control, sino como verdaderos interfaces de acceso al proceso. Junto a ellos las comunicaciones industriales han evolucionado para facilitar la su interconexión a la vez que sensores y actuadores han evolucionado para su conexión directa. [20]

Representación del Proceso: La representación del proceso es una etapa básica de la monitorización. La identificación inmediata de los elementos del proceso con una representación gráfica de estos es importante para lograr los objetivos de la monitorización. La representación visual de información y su interactividad es lo que se conoce como Interface Gráfico de Operador o Interface Máquina Hombre (HMI o MMI). Los paquetes SCADA actuales incorporan estas facilidades a través de ventanas en que se representa el proceso o parte de este. La navegación por estas pantallas se hace a través de menús desplegados y botones activados por el cursor gráfico asociado a un ratón, teclado o interfaz similar. [20]

Alarmas: Las desviaciones en la magnitud de una variable superior a unos límites específicos, son concebidas en los sistemas de monitorización como alarmas. Su objetivo es poner al operario sobre aviso. De esta forma, una vez generada una alarma se fecha, visualiza y registra, la automatización de la tarea de vigilancia del proceso se logra en los entornos de monitorización mediante estas alarmas usadas para detectar situaciones de comportamiento anómalo. para su análisis posterior, estas se registran juntamente con los instantes en los que se han producido. Una clasificación simple de los diferentes tipos de alarmas se obtiene de considerar las variables: [20]

Las alarmas discretas son indicadores de un cambio binario en el estado de la variable que representan. Una variable discreta puede ser utilizada para definir el estado de un sistema (paro/marcha, manual/automático), la presencia o ausencia de material, un finde carrera, la detección de paso, etc. Esta información es de gran utilidad para la secuenciación de tareas en el desarrollo de sistemas de automatización. Pero a su vez puede alertar al operario, en caso de ser necesario, utilizando para ello las señales directas del sensor tipo todo/nada u otros de propósito específico. [20]

En caso de las variables continuas se utilizan umbrales numéricos para designar los límites de operación normal de dicha variable. Su elección se hará de acuerdo con el significado físico de la variable y los límites permitidos para el funcionamiento correcto del proceso. Los sistemas de monitorización actuales incorporan diferentes formas de definir los umbrales de operación. Es habitual dichos umbrales como un porcentaje respecto a un valor y especificar diferentes límites para el cruce de umbral ascendente (situación de alarma) o descendente (restablecimiento de la situación normal) en forma de banda muerta. [20]

2.2.6. Interfaz Hombre-Máquina (HMI).

La interfaz Hombre-Máquina parte de un sistema interactivo (software y hardware) que proporciona la información y permite el control necesario para que el usuario lleve a cabo una tarea de dicho sistema. Sigla HMI. [22] Tradicionalmente estos sistemas consistían

en paneles compuestos por indicadores y comandos, tales como luces pilotos, indicadores digitales y análogos, registradores, pulsadores, selectores y otros que se interconectaban con la máquina o proceso. En la actualidad, dado que las máquinas y procesos en general están implementadas con controladores y otros dispositivos electrónicos que dejan disponibles puertas de comunicación, es posible contar con sistemas de HMI bastantes más poderosos y eficaces, además de permitir una conexión más sencilla y económica con el proceso o máquinas, como mostraremos a continuación. [22]

Tipos de HMI.

Descontando el método tradicional, podemos distinguir básicamente dos tipos de HMIs: [22]

- **Terminal de Operador:** consistente en un dispositivo, generalmente construido para ser instalado en ambientes agresivos, donde pueden ser solamente desde pliegues numéricos, o alfanuméricos o gráficos. Pueden ser además con pantalla sensible al tacto (touch screen).
- **PC + Software:** esto constituye otra alternativa basada en un PC en donde se carga un software apropiado para la aplicación. Como PC se puede utilizar cualquiera según lo exija el proyecto, en donde existen los llamados Industriales (para ambientes agresivos), los de panel (Panel PC) que se instalan en gabinetes dando una apariencia de terminal de operador, y en general veremos muchas formas de hacer un PC, pasando por el tradicional PC de escritorio. Respecto a los softwares a instalar en el PC de modo de cumplir la función de HMI hablamos a continuación.

Software HMI.

Estos softwares permiten entre otras cosas las siguientes funciones: Interfase gráfica de modo de poder ver el proceso e interactuar con él, registro en tiempo real e histórico de datos, manejo de alarmas. Si bien es cierto sólo con la primera función enunciada es la propiamente HMI, casi todos los proveedores incluyen las otras dos ya sea en el mismo paquete o bien como opcionales. También es normal que dispongan de muchas más herramientas. Al igual que en los terminales de operador, se requiere de una herramienta de diseño o desarrollo, la cual se usa para configurar la aplicación deseada, y luego debe

quedar corriendo en el PC un software de ejecución (Run Time). Por otro lado, este software puede comunicarse directamente con los dispositivos externos (proceso) o bien hacerlo a través de un software especializado en la comunicación, cual es la tendencia actual. [22]

Comunicación.

La comunicación con los dispositivos de las máquinas o proceso se realiza mediante comunicación de datos empleando las puertas disponibles para ello, tanto en los dispositivos como en los PCs. Actualmente para la comunicación se usa un software denominado servidor de comunicaciones, el que se encarga de establecer el enlace entre los dispositivos y el software de aplicación (HMI u otros) los cuales son sus clientes. La técnica estandarizada en estos momentos para esto se llama OPC (Ole for Process Control), por lo que contamos entonces con Servidores y Clientes OPC, sin embargo, aún quedan algunas instalaciones donde se usaba DDE (Dynamic Data Exchange) para este propósito, como también muchos softwares de aplicación sólo son clientes DDE por lo que lo usual es que los servidores sean OPC y DDE. [22]

2.2.7. Hardware Libre, modelos de negocios para América Latina y El Caribe.

Hardware Libre es una materialización particular del conocimiento libre en el área de hardware. En otras palabras, se podrá considerar que un hardware es libre, cuando el conocimiento asociado al mismo es de acceso público, es decir, que el autor o los autores están de acuerdo en facilitar el acceso al conocimiento generado en torno al desarrollo a través de distintos medios (impresos, digitales, entre otros). [23] Es un hardware cuyo diseño se pone a disposición del público de modo que cualquiera puede estudiar, modificar, distribuir, hacer y vender el diseño o el hardware que se sustente en dicho diseño. Diseño a partir del cual se construye está disponible en el formato que se prefiera para que se hagan modificaciones en él. Idealmente, el hardware libre utiliza elementos y materiales inmediatamente disponibles, procesos estandarizados, infraestructura abierta, contenido no restringido y herramientas de diseño libres para aumentar al máximo la

capacidad de los individuos de hacer y utilizar el hardware. El hardware libre brinda a los individuos la libertad de controlar su tecnología mientras comparten el conocimiento y alientan el comercio a través del intercambio abierto de diseños. [24]

Hardware Abierto los principios que definen el carácter abierto de un hardware fueron establecidos por la organización OSHWA (Open Source Hardware Association) con la finalidad de definir un marco básico de trabajo, de manera resumida los siguientes: [23]

- **Documentación:** El hardware debe ser puesto en libertad con su documentación completa y debe permitir la modificación.
- **Alcance:** La documentación debe especificar claramente qué parte del diseño se publica bajo la licencia.
- **Software necesario:** Si el diseño requiere de licencia de software, este debe cumplir unos parámetros de documentación suficiente y ser publicada bajo una licencia de código abierto aprobada por OSI.
- **Obras derivadas:** La licencia debe permitir modificaciones y trabajos derivados, así como la fabricación, venta, distribución y uso de productos creados a partir de los archivos de diseño.
- **Redistribución libre:** La licencia no debe restringir a un tercero el vender o entregar la documentación del proyecto. No puede ejercerse ningún derecho sobre obras derivadas tampoco.
- **Atribución:** La licencia puede requerir documentos derivados y avisos de copyright asociados a los dispositivos. Asimismo, debe hacer mención al diseñador.
- **No discriminatoria:** La licencia no debe discriminar a ningún grupo o persona.
- **No discriminación en función de la finalidad perseguida:** La licencia no debe restringir a ningún campo o actividad el uso de la obra.
- **Distribución de la licencia:** La licencia se da por distribuida sin necesidad de ir solicitando permisos adicionales.
- **La licencia no debe ser específica de un producto:** Los derechos de productos derivados hacen extensiva esta licencia.

- **La licencia no debe restringir otro hardware o software:** No se ponen objeciones a la naturaleza de lo que pueda implementarse a esta tecnología de forma externa o añadida.
- La licencia debe ser tecnológicamente neutral: Ninguna disposición de la misma debe de basarse en una tecnología específica, parte o componente, material o interfaz para su uso.

También hay algunos proyectos de HL que utilizan las licencias de software libre: [24]

- **Opencores:** Esta comunidad de hardware libre utiliza las licencias LGPL y licencias BSD modificadas para distribuir sus diseños y productos. Esta comunidad cuenta con más de 200,000 usuarios registrados al día de hoy.
- **FreeCores:** Esta es una comunidad derivada de Opencores que utiliza licencias GPL y Apache.
- **La Open Hardware Foundation:** Promueve la licencia copyleft (originaria del software libre) y otras licencias permisivas en la creación de hardware.
- **Raspberry Pi:** Computadora incrustada sobre una placa de hardware reducida que utiliza la licencia BSD para distribuir diferentes partes del diseño.
- **Arduino:** Placa de hardware diseñada principalmente para crear prototipos electrónicos y uno de los ejemplos más relevantes del HL. Arduino utiliza la licencia Creative Commons para distribuir y compartir su diseño. La licencia Creative Commons comprenden un conjunto de licencias publicadas desde 2002 por un equipo dirigido por el abogado Lawrence Lessig. Su uso abarca prácticamente cualquier producto creativo o cultural, desde películas hasta libros y sitios web. En particular, Arduino utiliza la versión Creative Commons Share Alike (para “compartir de la misma manera” los trabajos derivados).

Algunos modelos de negocios desde la perspectiva del acceso libre.

La publicación de planos esquemáticos e impresos bajo la modalidad open-source hardware manteniendo la posibilidad a los usuarios y desarrolladores de adquirir las tarjetas o bien descargar las fuentes y fabricarlas, también es común ver en este tipo de iniciativas el uso de la Internet, las redes sociales para darse a conocer. La participación

de la comunidad de usuarios y desarrolladores también es importante, la generación de foros de discusiones de forma libre sobre temas en particular o sobre los productos que se ofrecen desde estas empresas. [23]

- **Adafruit Industries:** Es una empresa open-source hardware fundada por Limor Fried en 2005. La compañía diseña y manufactura un número de productos electrónicos, vende una variedad de componentes electrónicos, herramientas, y accesorios vía tienda online, también produce una buena cantidad de tutoriales para la enseñanza de la electrónica y uso de sus productos, videos introductorios tanto para niños como para profesionales todo vía Internet. Todos los productos AdaFruit son fabricados en Manhattan, Ciudad de New York. En 2013, la compañía reportó ganancias en el orden de los \$15 millones de dólares, ha realizado más de un millón de envíos a nivel mundial, nada mal para una iniciativa que inicia en el dormitorio de Limor Fried en el MIT.
- **Sparkfun:** Es una tienda electrónica fundada en 2003 con sede en la Ciudad de Boulder, Colorado USA, que ahora cuenta con más de 3500 partes distribuidos entre componentes a detal y kits electrónicos, los planos de los kits electrónicos están disponibles así que es una empresa apoyada en la tecnología open-source hardware. Desde este portal se ofrecen foros, videos tutoriales, planos, programas, enlaces de interés entre otros. Nathan Seidle es el Fundador y CEO de Sparkfun Electronics, tienden a ser unos de los pioneros en el mundo de la electrónica DIY (Do it yourself - Hágalo usted mismo).

Arduino Mega

En la figura 2.5, se muestra un Arduino Mega 2560 es un producto popular entre las tarjetas Arduino. Se puede alimentar a través del puerto USB cuando se conecta al ordenador o a través de la toma de alimentación con el puerto Jack o la entrada de alimentación, es recomendable utilizar una alimentación de entre 7V y 12V de corriente continua para alimentar la tarjeta. Arduino Mega 2560, tiene como Arduino uno, una entrada de alimentación (VIN), un conector 5v, un conector 3,3v y tomas de tierra. [25]

Los conectores de entrada y salida de la tarjeta Arduino Mega 2560 se pueden describir en dos partes. Los 54 conectores de entrada y salida digital se utilizan como su nombre indica, en modo entrada o salida, utilizando las funciones pin Mode, Digital Read y/o Digital Write. Entre estos 54 conectores, varios tienen funciones extendidas: [25]

- Conectores acoplados de unión en serie: 0(RX) y 1(TX); 19(RX) y 18(TX); 17(RX) y 16(TX); 15(RX) y 14(TX). Estos conectores son particulares porque su función se puede extender a una entrada en serie (RX) y una salida en serie (TX).
- Conectores interruptores: 2 (interruptor 0), 3(interruptor 1), 18 (interruptor 5), 19 (interruptor 4), 20 (interruptor 3), y 21 (interruptor 2). Estos conectores pueden jugar el papel de interruptor en diferentes componentes, utilizando la función attachInterrupt ().
- Conectores PWM: de 2 a 13 y de 44 a 46. Estos conectores se utilizan como salidas PWM, utilizando la función analogWrite().
- Conectores SPI: 50(MISO), 51(MOSI), 52(SCK) y 53 (SS). Estos conectores se pueden utilizar para gestionar una comunicación SPI.
- Conector Led-13: en la tarjeta Arduino Mega, existe un led controlable directamente a través del conector 13, utilizando los valores HIGH para encender y LOW para apagar el Led.
- Conectores TWI: 20 (SDA), y 21 (SCL): administración de la comunicación I²C.



Figura 2.5: Arduino Mega 2560. [26]

2.3 PROPUESTA DE SOLUCIÓN

El sistema de control y alerta del tanque purificador de agua, contribuye a la disminución del tiempo de respuesta en caso de presencia de anomalías en el tanque, el sistema electrónico precautela el bienestar de la planta purificadora ECOAGUA y de las personas que la conforman, brindando la facilidad de monitorear mediante una alarma emitida desde tanque a través de una estación remota con una interfaz hombre-máquina.

CAPÍTULO III

METODOLOGÍA

3.1 MODALIDAD DE LA INVESTIGACIÓN

En el presente proyecto de investigación, se desarrolló una investigación aplicada debido a que se trabajó en la solución a un problema técnico y se propone un grupo de ideas que contribuyen a generar mayor eficiencia y seguridad en el proceso de control y alertas del tanque purificador de agua en la planta purificadora ECOAGUA, con la finalidad de evitar pérdidas del líquido y paradas inesperadas del proceso de embotellamiento de agua.

Se enmarcó dentro de la investigación bibliográfica ya que la obtención de información sobre el tema fue referenciada y recolectada de proyectos de titulación de repositorios públicos, revistas y artículos científicos, libros electrónicos para tener diferentes enfoques con respecto al tema de investigación.

La investigación de campo se realizó en la planta ECOAGUA, donde se determinó el problema de desperdicio del agua y paradas inesperadas del tanque.

3.2 RECOLECCIÓN DE INFORMACIÓN

La información se obtuvo principalmente de referencias y fuentes bibliográficas, revistas y artículos científicos, libros impresos y proyectos de titulación que se han desarrollado en los últimos años, cada una de estas vinculadas con el control y alertas del nivel de líquido en los tanques.

3.3 PROCESAMIENTO Y ANÁLISIS DE DATOS

Una vez obtenida la información necesaria para la investigación planteada, esta información formó parte de un proceso metódico, el cual consistió en:

- Revisión de la información recopilada
- Manejo de información
- Análisis de información adquirida, con la finalidad de diseñar estrategias de solución al problema.
- Interpretación de los resultados obtenidos

3.4 DESARROLLO DEL PROYECTO

A continuación, se presenta los pasos que se efectuaron en el desarrollo del proyecto de investigación:

- Reconocimiento de las instalaciones de la planta purificadora ECOAGUA del Cantón Tisaleo.
- Determinación de las características del tanque purificador de agua.
- Análisis de las variables físicas del proceso.
- Desarrollo de la interfaz para el control y alerta del tanque purificador de Agua.
- Análisis comparativo de características de los equipos, costos y materiales necesarios para el sistema de control y alerta del tanque purificador de Agua.
- Selección del Hardware a emplearse en el sistema electrónico.
- Adquisición de los equipos para la implementación del sistema de control y alerta del tanque purificador de Agua.
- Implementación del sistema de control y alerta.
- Pruebas de funcionamiento del sistema y depuración de errores.
- Análisis de Resultados y Conclusiones.
- Elaboración Informe Final.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4.1 INTRODUCCIÓN

El presente proyecto de investigación se desarrolló en la Planta ECOAGUA, se centra en el almacenamiento de líquido en el sistema de tanques en donde se identifica el problema el deficiente control en la etapa de llenado. Este inconveniente surge debido al funcionamiento manual de sus equipos como son válvulas, electro-bombas y el inadecuado control del llenado de líquido en los de tanques de agua cruda, todo esto produce malestar a los procesos siguientes.

El proceso de llenado de líquido en los tanques es operado por una sola persona y lo realiza de forma manual teniendo que esperar que los tanques posean los niveles adecuados del líquido para continuar al proceso de purificado de agua, en el caso de que los tanques no tengan los niveles apropiados de agua las bombas succionan impurezas o aparecen espacio huecos provocando un avance en el deterioro de las electroválvulas, con grandes posibilidades de que se quemen en el acto y detengan la producción al no contar con un sistema de relevos 2:1 en las electrobombas.

Otro aspecto es que los filtros de arena y de carbón activado, deben presentar un caudal constante para evitar taponamientos en sus membranas al activarse el proceso de retro-lavado, si se suscita este inconveniente la electrobomba debe ser detenida y el flujo de líquido de agua cruda debe impedirse hasta que termine este ciclo, por lo que el control debe ser una prioridad.

El proyecto de investigación consiste en elaborar una tarjeta electrónica basada en un microcontrolador con arquitectura de hardware capaz de soportar picos de corriente, que sea bajamente inmune al ruido electromagnético producido por contaminación eléctrica presente en las líneas de tensión al activarse y desactivarse los equipos en el entorno industrial. Debe ser capaz de trabajar en ciclos continuos 24:7 sin interrupción, permitiendo adquirir señales analógicas procedentes de los niveles de líquido y otros sensores que se utilizan, con la capacidad de poder controlar activamente cargas de gran potencia como electrobombas, electroválvulas entre otras y que genere una trama de información que servirá para poder monitorear, supervisar, registrar y controlar el sistema en un HMI por medio de una PC ubicada estratégicamente en la planta ECOAGUA.

4.2 ANÁLISIS DE FACTIBILIDAD

4.2.1. Factibilidad Técnica

El presente proyecto de investigación, es factible ya que los elementos y dispositivos electrónicos para el desarrollo se los puede adquirir en el mercado nacional.

4.2.2. Factibilidad Institucional

Para la empresa ECO-AGUA la propuesta presentada le es muy beneficiosa, debido a que le permite salvaguardar sus equipos, controlar de manera eficiente y automática parte de su proceso con una baja intervención humana.

4.2.3. Factibilidad Bibliográfica

Desde el punto de vista bibliográfico el proyecto de investigación es factible ya que se cuenta con la información necesaria de la Planta, una búsqueda complementada y robusta gracias a la posibilidad de consulta en libros, artículos científicos, publicaciones en páginas web, revistas científicas y tesis de grado relacionadas al tema de investigación.

4.2.4. Factibilidad Económica

El desarrollo del proyecto es económicamente factible, ya que los gastos que conllevan el presente proyecto de investigación serán asumidos por parte del propietario de la Planta Purificadora ECOAGUA.

4.3 SITUACIÓN ACTUAL DE LA PLANTA PURIFICADORA ECOAGUA

ECOAGUA es una microempresa dedicada al tratamiento y envasado de agua para consumo humano, inicio sus actividades comerciales el 19 de agosto del 2008, registrado en el SRI con RUC número 1804264933001 y como tipo de contribuyente "PERSONAS NATURALES". El señor José Neptalí Almendariz López es propietario de la planta mencionada, se encuentra ubicada en el Caserío San Vicente de la Parroquia Quinchicoto (Tisaleo-Ecuador).



Figura 4. 1: Planta de tratamiento y envasado de agua para consumo humano ECOAGUA.

Elaborado por: Investigador.

ECOAGUA cuenta con una producción de 100 botellones de agua en el día, para la obtención del producto se realiza varios procesos previos como: lavado, esterilizado, envasado, sellado y etiquetado.

El proceso químico para el pre-tratamiento del agua inicia con el almacenamiento del agua potable en los tanques de agua cruda, posteriormente esta agua pasa por el filtro de arena y antracita con el propósito de la retención de las grandes impurezas que contiene el agua,

inmediatamente por el filtro de carbón activado el mismo que se encarga de eliminar cloro, mal olor y retener plaguicidas, aceites.

Ósmosis inversa, es el proceso que ayuda a la retención de las sales y minerales restantes, concentrando los sólidos en una corriente de rechazo, el agua continua y el siguiente proceso es el Ozonificador este destruye las bacterias, virus, y toda clase de organismos patógenos. Finalmente, el paso por la Unidad Ultravioleta eliminando los microorganismos y bacterias inmunes al cloro, todos estos procesos ayudan a la obtención de agua purificada, cumpliendo con las normas de calidad.

Todo este trabajo es realizado de forma manual, hasta el momento la planta ECOAGUA no cuenta con un sistema automatizado en el proceso del llenado de los tanques de agua cruda y filtrada. Por lo tanto, surge gran cantidad de desperdicio de agua, constituyendo una pérdida para la planta tanto en recursos como en tiempo.

4.4 REQUERIMIENTOS DEL SISTEMA

Para la implementación del sistema de control es necesario contar con una buena acometida eléctrica, equipos en buen estado, puesta a tierra, dispositivos libres de polvo ya que el sistema se encuentra dentro de un ambiente industrial el mismo que esta expenso a ruido electromagnético. Además, la placa electrónica a construir debe contar con un grado de protección contra ruido, sobretensión, etc.

Para evitar desperdicio de agua por desbordamiento, es importante contar con un control en el llenado de los tanques, en este caso el tipo de control que se utiliza es un control On/Off ya que el operario debe ser cuidadoso con el trato del agua al momento de la purificación, puesto que es inapropiado que el agua se mantenga estancada ya que debe ser de buena calidad para el consumo humano.

Los equipos instalados deben tener su respectivo grado de protección, cableado adecuado para los equipos. De igual forma mantener normativas para el desarrollo de la interfaz hombre-máquina.

A continuación, se detalla el hardware y software a utilizar:

- Sensor de nivel
- Electroválvulas
- Relé de Estado Sólido
- Microcontrolador
- Tipo de Comunicación
- Computador
- Visualización local de la información
- Entorno de Desarrollo

4.4.1 Hardware

MEDICIÓN DE NIVEL DE LIQUIDO

Tenemos dos tanques de capacidad 1200 litros. Cada uno de ellos posee un sensor de nivel, este sensor puede ser ultrasónico, laser o por radar. Como se observa en la figura 4.2.

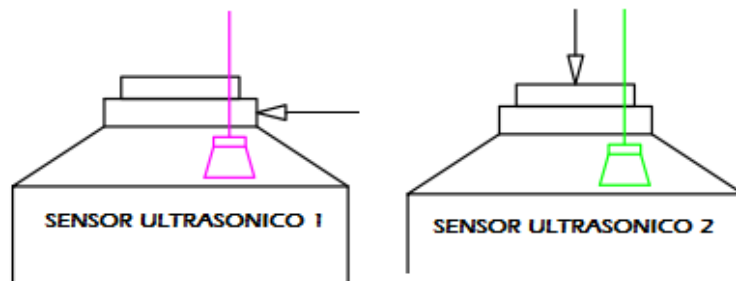


Figura 4. 2: Tanques de agua cruda y agua filtrada de ECOAGUA.

Elaborado por: Investigador.

Selección del Sensor de Nivel.

Los sensores tienen características, ventajas, desventajas y aplicaciones que se muestran en la tabla 4.1.

Tabla 4. 1: Sensor de presencia o proximidad. [27] [28] [29] [30] [31]

Sensor de proximidad	Características	Ventajas	Desventajas	Aplicación
Magnéticos o Reed	Reaccionan frente a campo magnéticos generados por imanes permanentes o electroimanes.	Controla el nivel de llenado de sólidos en un contenedor	Debe reducirse al máximo flujo de corriente de lo contrario esto puede provocar un arco de descarga durante la conexión/desconexión.	Detectores de posición en los cilindros neumáticos.
Inductivos	Consiste en una bobina situada junto a un imán permanente.	Robustez mecánica, resistencia a ambientes agresivos y altas temperatura	Solo detecta la presencia de objetos metálicos, el margen de operación es corto.	Detector de posición de objetos.
Capacitivos	Es un condensador constituido por un electrodo sensible y un electrodo de referencia separados por un dieléctrico.	Permiten detectar materiales metálicos y no metálicos (Polvos, granulados, líquidos, sólidos)	Alcance de detección corto,	Medida de nivel de tanques. Detección de objetos no metálicos
Ultrasónicos	Usan un transductor para enviar y recibir señales de sonido de alta frecuencia.	Detección del objeto independiente del color y del material, insensible a la suciedad y el polvo.	Objetos con posiciones inclinadas, el sonido se desvía. Frecuencias de conmutación máxima entre 1 y 125 Hz.	Medición de nivel. Detección de objetos en movilidad.
Ópticos	Utilizan un haz luminoso modulado que puede ser interrumpido o reflejado por un objeto.	Incremento de la fiabilidad debido a la presencia permanente de luz durante el estado de reposo.	No para objetos completamente transparentes, un fallo en el emisor es evaluado como objeto.	Verificando objetos en envases transparentes.

Luego de realizar una comparación entre los diferentes sensores de nivel o proximidad se elige a los sensores ultrasónicos por su capacidad de detectar objetos pequeños y en movilidad, detección de nivel y lo más importante poseen un precio económico y accesible, para la realización de este proyecto de investigación.

Para seleccionar el sensor ultrasónico más idóneo para el sistema de control y alerta, se realizó la valoración de distintos sensores ultrasónicos como se visualiza en la tabla 4.2.

Tabla 4. 2: Selección del Sensor Ultrasónico. [32] [33] [34] [35]

Sensor Ultrasónico	Tensión	Corriente	Resolución	Rango	Temperatura de Operación	Precisión	Protección IP
HC-SR04	5 V	15mA	0.003m	0.02m-4m	20°	3mm o 0.3cm	—
MB7369 HRXL-MaxSonar	2.7V a 5.5V	2.3mA	0.001 m	0.3m-5m	-40 ° C a + 65 ° C	1mm	IP67 conforme a IEC 60529
SRF04	5V	30mA	0.01m	0.3m-3m	20°C	3cm a 3m	—
xx930a3a2m12	15V-24V	60 mA	---	0.2m-8m	-20°C a 60°C	2.54%	IP67 conforme a IEC 60529

Debido a las características se seleccionó el sensor MB7369 HRXL-MaxSonar-WRM, ya que es un sensor resistente a la intemperie diseñado para detección de objetos y su alcance al aire libre o en tanques y contenedores.

La línea de sensores HRXL-MaxSonar-WR proporciona alta precisión y alta resolución de proximidad ultrasónica detección y variación en el aire. Esta línea de sensores cuenta con una resolución de 1 mm, tamaño objetivo y compensación de voltaje de operación para una mayor precisión, rechazo superior de fuentes de ruido externas, compensación de temperatura de velocidad de sonido interna y compensación de temperatura de

velocidad de sonido externa opcional. Los formatos de salida de la interfaz son de pulso ancho, voltaje analógico y serie digital en RS232 (serie MB7360) o TTL (serie MB7380). [32]

Características Generales del sensor ultrasónico MB7369 HRXL-MaxSonar-WRM: [32]

- Detección a 5 metros o 10 metros.
- Resolución de 1 mm.
- Sensor de distancia de 30 cm a 5 metros o 50 cm a 10 metros basado en el modelo.
- Operación de ejecución libre con superior rechazo de ruido 3.
- Rango de temperatura de funcionamiento de -40°C a $+65^{\circ}\text{C}$.
- Voltaje de funcionamiento de 2.7V a 5.5V.
- Consumo de corriente nominal de 2.3 mA.
- Firmware filtrado de ruido excelente tolerancia y rechazo de parásitos.
- Resistente a la intemperie (IP67), opcional resistente a productos químicos F- opción.
- Real-Time calibración automática (tensión, humedad, y Ruido ambiente).

En el Anexo A, se detallan todas las característica y especificaciones. En la figura 4.3 se muestra la imagen del Sensor Ultrasónico MB7369 HRXL-MaxSonar.

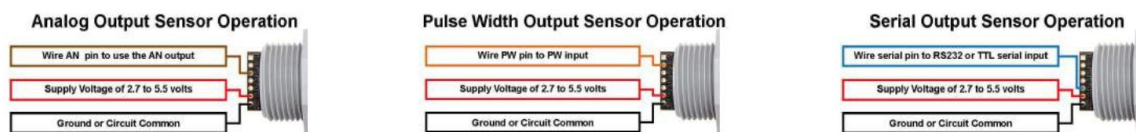


Figura 4. 3: Diagrama de conexión del sensor ultrasónico MB7369 para obtener una salida analógica, ancho de pulso y serial. [32]

El siguiente diagrama electrónico presenta el modo de conexión que tendrá lugar en el proyecto de investigación, el tipo de sensor que se utilizó es de canal analógico, con un sistema de filtro pasa bajos para disminuir el ruido producido por interferencias del sistema.

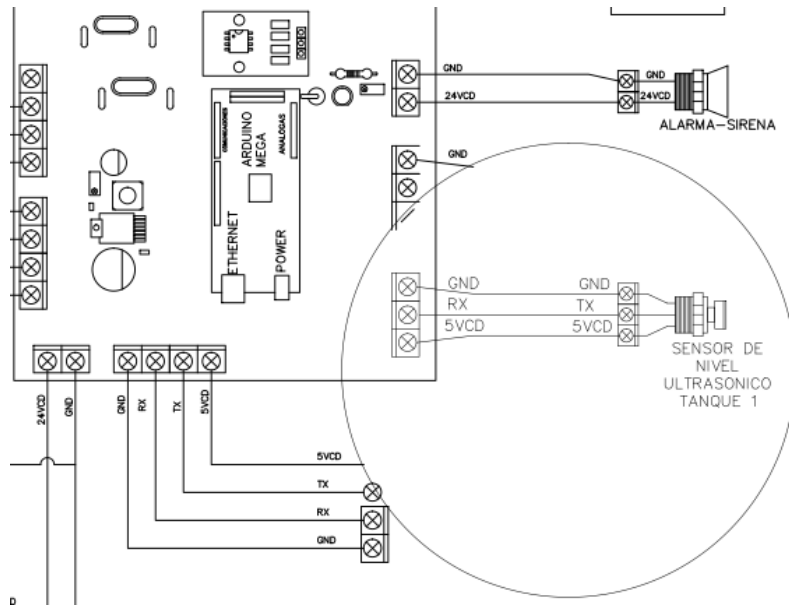


Figura 4. 4: Conexión del Sensor Ultrasónico MB7369 a la tarjeta de control.

Elaborado por: Investigador.

CONTROL DE ELECTROVÁLVULAS

Para el control de flujo de agua hacia y desde los tanques de reserva se utiliza electroválvulas de $\frac{1}{2}$ " y de $\frac{3}{4}$ ", en este caso las electroválvulas funcionan a 110vca.

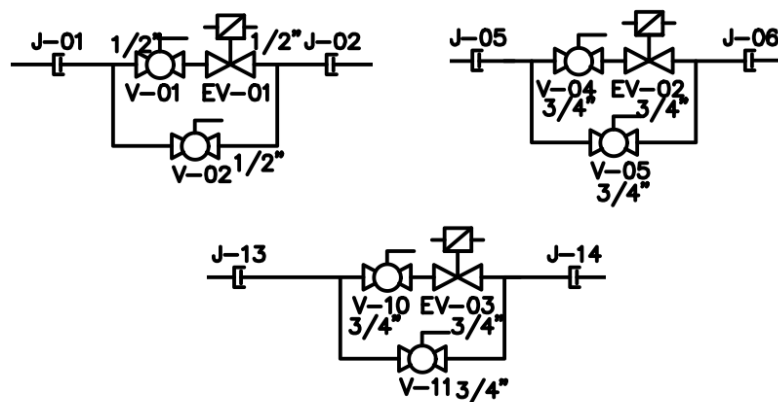


Figura 4. 5: Electroválvulas de $\frac{1}{2}$ " y de $\frac{3}{4}$ ".

Elaborado por: Investigador.

Para la elección se las electroválvulas adecuadas, se realizó una comparación entre las características más sobresalientes, como se muestra en la tabla 4.3.

Tabla 4. 3: Selección de actuador. [36][37][38]

Actuador	Voltaje de Operación	Presión	Frecuencia	Temperatura	Grado de protección	Material
Electroválvula NF – latón – hembra BSP	12-24-48-110-230-400 vac / 12-24-48-110 vcc	DN ≤ 2" : 0,1 bar DN > 2" : 0,5 bar	50/60 HZ	+ 90 °C (NBR) + 140 °C (EPDM)	Estándar ISO 228-1.	Latón
Electroválvula modelo 2w-200-20	AC110V	0 ~ 1 MPa (10 bar)	50/60 HZ	-5 ~ + 80 °C	IP65	Latón
Electroválvula de control asistido de 2/2 vías de acero inoxidable	---	Mín.: 0.3 bar (4.35 psi) Máx.: 10 bar (145.04 psi)	50/60 HZ	0 °C (32 °F) a 60 °C (140 °F)	IP67	Acero Inoxidable

Según la comparación realizada entre los diferentes actuadores, se optó por la electroválvula de membrana modelo 2w-200-20, ya que es una electroválvula de membrana de dos vías una posición, normalmente cerrada y funcionan a 110 VCA.

Características de la Electroválvula de Membrana 2w-200-20.

Modelo Electroválvula de 2w-200-20, voltaje AC110V, material latón, abertura de flujo 20mm, presión máxima 10 kg / cm², temperatura de trabajo: -5 ~ + 80 °C, temperatura del ambiente: -10 ~ + 50 °C, presión de trabajo 0 ~ 1 MPa, medio de trabajo aire, agua y aceite [36], como se muestra en la figura 4.6.



Figura 4. 6: Electroválvula de Membrana 2w-200-20. [36]

La tarjeta electrónica controla este tipo de válvula con el accionamiento por medio de Relay de estado sólido, el cual va a ser activado electrónicamente por un transistor tip122. Ver figura 4.7.

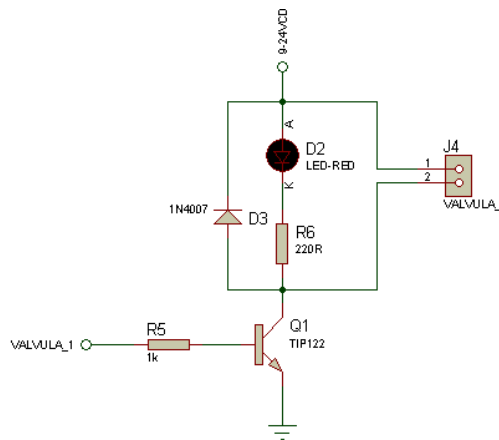


Figura 4. 7: Transistor tip122.

Elaborado por: Investigador.

Para elegir el actuador correspondiente para el accionamiento de las electroválvulas se realizó una comparativa que se detalla en la tabla 4.4.

Tabla 4. 4: Selección de actuador para el accionamiento de las Electroválvulas. [39][40][41][42]

Actuador	Voltaje de Control	Corriente de Control	Tiempo de conexión e interrupción	Dimensión
Control Parada Emergencia CPA-D	24 VCA/CC	---	10 ms	97 x 22,5 x 115,5 [mm] (H x W x D)
Relay SRD-12VCD-SL-C	12 V	10A	10ms	19 x 15 x 15 mm
ZG3NC-370B relé de estado sólido	3-32VDC	3-25mA	≤10ms	63L*45W*26H
IP Relé	12-15VDC	150 mA	10ms	25 x 82 x 90 [mm] (H x W x D)

Se ha escogido el relay de estado sólido ZG3NC-370B dadas sus características, ya que está orientado en mayor proporción a aplicaciones industriales, adicionalmente son dispositivos semiconductores que usan luz en lugar de magnetismo para activar interruptores. Este relay permite controlar cargas sin presencia de contactor mecánicos disminuyendo considerablemente ruido electromagnético producido por el accionamiento de cargas inductivas, cuentan con baja interferencia electromagnética e interferencia de radio frecuencia.

Características de ZG3NC Relé de estado sólido: [39]

- Modelo: ZG3NC-370B, como se muestra en la figura 4.8.
- Nombre: Relé de estado sólido monofásico
- Voltaje de control: 3-32VDC
- Corriente de control: 3-25mA
- Corriente de carga: 70 A
- Voltaje de carga: 90-480 VAC
- Tiempo de conexión e interrupción: ≤10ms
- Resistencia de aislamiento: 100mΩ 500VDC
- Temperatura ambiente: -25~75°C
- Dimensión: 63L*45W*26H

- Resistencia eléctrica 2500VAC
- Fuga de corriente en estado interrumpido: $\leq 4\text{mA}$
- La norma IEC
- Peso: 135g
- Método de montaje: tornillo.



Figura 4. 8: ZG3NC Relé de estado sólido. [39]

SALIDA DE DATOS DIGITAL

La tarjeta se diseñó para que entregue datos de vía serial hacia el sistema Scada, para lo cual se utiliza un circuito bastante práctico y sencillo con el MAX232 para acoplar voltajes entre equipos. Como se muestra en la figura 4.9.

En el caso de que por algún motivo la tarjeta se cuelgue o no trabaje adecuadamente, la tarjeta de control será provista de un circuito externo de Reset automático que trabaja en forma paralela al proceso de control, el cual supervisará al Arduino reiniciándolo cuando estime conveniente y no deje de trabajar.

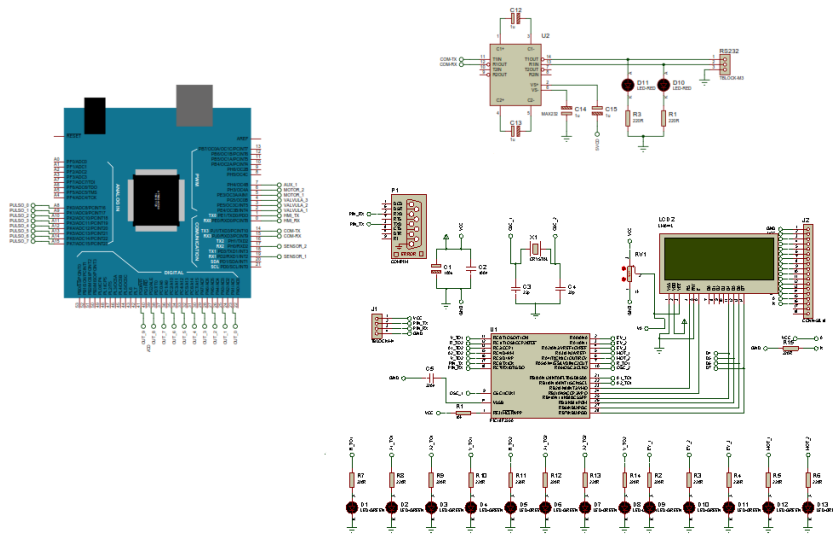


Figura 4. 9: Diagrama electrónico del Circuito MAX232.

Elaborado por: Investigador.

Sistema de energizado

Para que la tarjeta trabaje con los niveles de voltaje adecuados se dispondrá de un sistema estabilidad de voltaje que permita protecciones de sobre voltaje y polarización inversa, con un sistema filtro de armónicos que reducirán en la mayor parte le ruido procedente del sistema eléctrico de la planta. Como se observa en la figura 4.10.

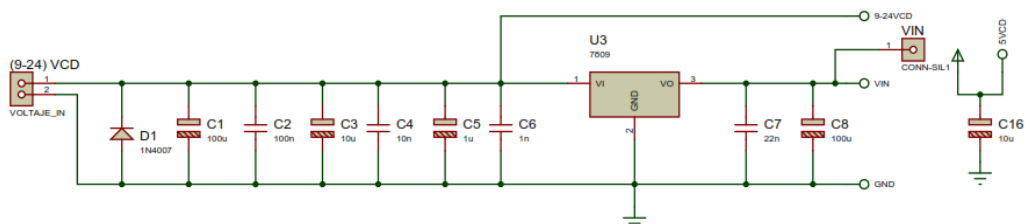


Figura 4. 10: Circuito estabilizador de voltaje.

Elaborado por: Investigador.

SELECCIÓN DEL DISPOSITIVO QUE FUNCIONARÁ COMO UN CONTROLADOR LÓGICO PROGRAMABLE.

Se realizó una tabla comparativa entre las tarjetas electrónicas más usados detallando ventajas y desventajas, como se muestra en la tabla 4.5.

Tabla 4. 5: Selección Dispositivo que funcionará como PLC. [43][44][45]

Tarjetas	Ventaja	Desventaja	Procesador	Entradas /Salidas	Velocidad	Bus
M- DUINO PLC Arduino	Este autómata basado en Arduino (Open Source Hardware) está especialmente diseñado para su uso en un entorno profesional. Ofrece la posibilidad de expandirse con 127 módulos mediante el sistema I2C, lo que significa que puede gobernar hasta 6604 E/S en modo maestro esclavo	Elevado costo. Se debe adquirir módulos de expansión.	Celeron 2.0Ghz	13 entrada y 8 salidas	16 MH z	SPI, I2C, TCP IP, Modbus TCP, Modbus RTU
Arduino Mega 2560	Arduino Mega 2560 posee un bajo costo de adquisición, Las plataformas de Arduino pueden ser adquiridas en versiones de montaje superficial que ocupan muy poco espacio y consumen muy poca potencia. Tanto el software como el hardware son libres.	El hecho de que la plataforma venga ya ensamblada les quita flexibilidad a los proyectos, obligados a usar un espacio y forma acorde con el PCB del Arduino.	Atmega 2560.	54 pines de entradas/ salidas	16 MH z	I2C, SPI
BeagleB one Black	Es la combinación de un Arduino y Raspberry Pi. Tiene el poder de procesamiento de la Raspberry Pi, y la opción de manejar interfaces externas como Arduino.	BeagleBone no dispone de tantos puertos USB. Tampoco tiene la codificación de vídeo incorporado	AM33x	69 digitales, 7 entradas analógicas, 4 Times, Serial Ports.	1 GH z	I2C, CEC

Una vez realizada la comparación se opta por elegir la tarjeta Arduino Mega 2560 ya que cuenta con variedad entradas analógicas y entradas/salidas digitales, además su precio y adquisición son accesible.

Sin embargo, el M-Duino también posee características sobresalientes, pero las causas porque no se elige el M-Duino Plc Arduino son: este dispositivo industrial cuenta con un solo puerto RS232 y se necesitan dos ya que los elementos que controlan el nivel de los tanque son dos, en el proyecto se necesitan 8 entradas y 16 salidas el inconveniente del M-Duino es que para obtener las salidas requerida se debe adquirir un módulo expansor, y todo esto representa un costo alto de adquisición.

Arduino Mega 2560

Arduino Mega 2560 es una versión ampliada de la tarjeta original de Arduino y está basada en el microcontrolador Atmega2560. Dispone de 54 entradas/salidas digitales, 14 de las cuales se pueden utilizar como salidas PWM (modulación de anchura de pulso). Además, dispone de 16 entradas analógicas, 4 UARTs (puertas series), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un conector ICSP y un pulsador para el reset. Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o alimentarla con un adaptador de corriente AC/DC. También, para empezar, puede alimentarse mediante una batería. Esta placa debido a su gran poder es utilizada para grandes proyectos, entre los más importantes se encuentran los de domótica e impresoras 3D. El Arduino MEGA2560 es compatible con la mayoría de los shield o tarjetas de aplicación/ampliaciones disponibles para las tarjetas Arduino UNO original. A continuación, se muestran las Especificaciones Técnicas: [46]

- Microprocesador ATmega2560
- Tensión de alimentación (recomendado) 7-12V
- Integra regulación y estabilización de +5Vcc
- 54 líneas de Entradas/Salidas Digitales (14 de ellas se pueden utilizar como salidas PWM)
- 16 entradas Analógicas
- Máxima corriente continua para las entradas: 40 mA

- Salida de alimentación a 3.3V con 50 mA
- Memoria de programa de 256Kb (el bootloader ocupa 8Kb)
- Memoria SRAM de 8Kb para datos y variables del programa
- Memoria EEPROM para datos y variables no volátiles
- Velocidad del reloj de trabajo de 16MHz
- Reducidas dimensiones de 100 x 50 mm

En el Anexo B se detallan las características técnicas.

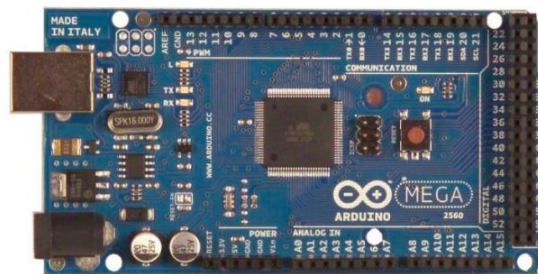


Figura 4. 11: Arduino Mega 2560. [46]

Elaborado por: Investigador.

COMUNICACIÓN ENTRE LA TARJETA DE CONTROL Y LA COMPUTADORA

Luego de contar con el gabinete instalado, fue necesario la elección de un medio de comunicación para el monitoreo de sistema de control y alerta del tanque purificador de agua, la elección se inclinó por la comunicación RS232 ya que este brinda una estabilidad en la comunicación entre la PC y sus periféricos usando una conexión local de los mismos. Adicional a esto, los sensores y el Arduino Mega cuentan con una salida serial RS232 que directamente ayudaron a la fácil manipulación de las señales. No obstante, se usó un conversor USB a serial macho DB9 para la conexión de la tarjeta de control con la PC.

Estándar RS-232: El estándar se ha desarrollado por más de cuarenta (40) años durante los cuales la EIA ha publicado tres (3) modificaciones, la más reciente llamada EIA-232F introducida en 1997. El nombre del estándar paso de RS-232 a ANSI/EIA-232 al igual que otros elementos de la norma original han cambiado su denominación. Las características Eléctricas: La norma define un margen de tensión de +3 V a +15 V para el

“0” lógico y -3 V a -15 V para el “1” lógico; por lo que, se hace necesario que tanto en la transmisión como en la recepción se utilice un circuito de adaptación que transforme los niveles de tensión utilizados en los circuitos digitales en los que se acaban de indicar y viceversa, ésta característica está descrita en mayor detalle en la norma CCITT V.28 definida para las características eléctricas del estándar RS-232C. [47]

La longitud de la línea de transmisión tiene influencia en las señales eléctricas que viajan por ella. El valor de la resistencia, que depende sólo de la longitud de la línea, no es el que limita su longitud máxima, sino que es la capacidad y la inductancia, que dependen también de la velocidad de transmisión, las que determinan esta longitud. Las longitudes seguras para los cables de transmisión serie según EIA (Asociación de Industrias Electrónicas) recomienda que la capacidad total del cable no exceda los 2500 picofaradios. Como los cables de transmisión de datos tienen un valor de entre 120 a 150 picofaradios por metro, entonces 15 metros sería la longitud máxima que podría tener, con velocidad de transmisión de 20 Kbps. [48] En la figura 4.12, se muestra los pines del conector DB9.

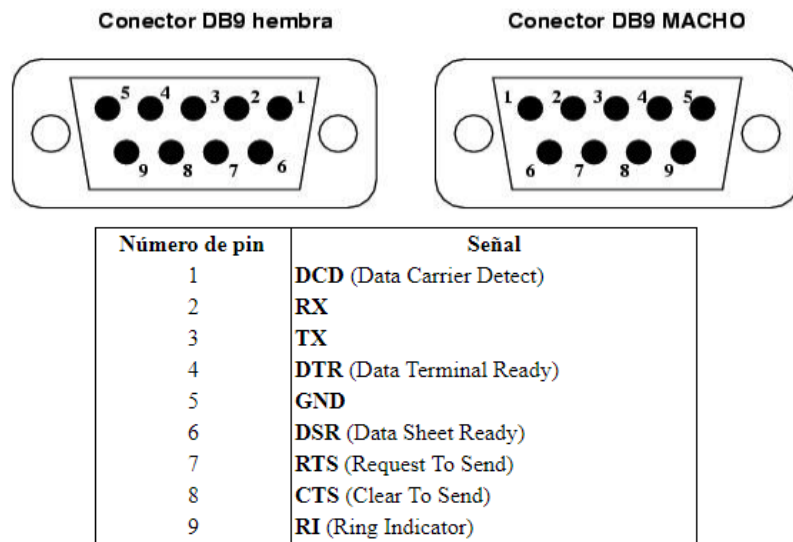


Figura 4. 12: Pines del conector DB9 Macho y Hembra. [48]

Elaborado por: Investigador.

Tecnología USB: Las siglas USB corresponden a Universal Serial Bus, Bus Serie Universal, por lo que como su nombre indica, se trata de un sistema de comunicación entre

dispositivos electrónicos informáticos que sólo transmite una unidad de información a la vez. El bus USB puede trabajar en dos modos, a baja velocidad (1,5 Mbps, para dispositivos como teclados, ratones, que no manejan grandes cantidades de información) y a alta velocidad (12 Mbps, para dispositivos como unidades de CDROM, altavoces, módems RTC e ISDN, etc.). En cuanto a la comodidad, el bus USB se compacta en un cable de cuatro hilos, dos para datos, dos para alimentación. Esto supone un gran ahorro, tanto de espacio como de material. [49] En la figura 4.13 se muestra los pines del conector USB.

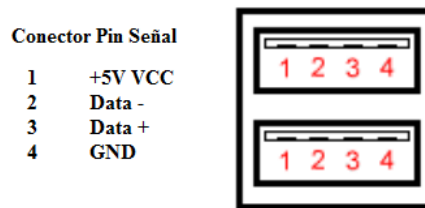


Figura 4. 13: Pines del conector USB. [49]

Elaborado por: Investigador.

Cable Convertidor USB a Serial RS232: Sirve como adaptador convertidor de un puerto USB a puerto serial RS232 DB9 para conectar, monitorear y controlar un dispositivo serial. Con soporte para tasas de transferencia de datos de hasta 921.6 Kbps, el cable USB a serial constituye una solución ideal para proporcionar conectividad serial a dispositivos como terminales de punto de venta, módems seriales, equipos industriales, así como una amplia gama de otros periféricos que manejen el estándar RS232. El conversor USB a RS232 es fácil de instalar y ofrece amplia compatibilidad con sistemas operativos, lo cual permite su fácil integración en entornos mixtos. Especificaciones del Cable Convertidor USB a Serial RS232: [50]

- Adaptador USB a serial RS232 DB9 directo
- Alta velocidad de baudios de 921,6 Kbps
- Compatible con sistemas Windows, Mac OS X y Linux
- Compatible con puertos USB 1.1 o superior
- Alimentación por USB – no requiere adaptador de alimentación externo

- Diseño de cable único que ocupa poco espacio, para permitir su portabilidad
- Portátil y de peso liviano, su diseño con alimentación USB no requiere ningún adaptador de alimentación adicional
- Color Azul metálico con plateado. Como se muestra en la figura 4.14.



Figura 4. 14: Cable Convertidor USB a Serial RS232. [50]

Elaborado por: Investigador.

Trama que se envía desde la tarjeta de control hacia la PC: La trama que se maneja es una trama serial con 28 registros cada uno equivalente a un byte, y cada byte equivale a una operación, sabiendo que un registro es un número que va desde 0-255. Como se muestra en la figura 4.15.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
I N T R A M A O	*	1/0	1/0	1/0	1/0	1/0	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	#	
		E	EV	EV	MT	MT	lt	%	m	lt	%	m	H	Min	O	P	E	R	R	R	SM	MA	LS	MIN	ht2	SM	MA	LS	MI	F
	V1	2	3	1	2	TANQUE 1			TANQUE 2			A C T U A L		E R R O R	E R R O R	R E P O S O	VALORES DE REFERENCIA TANQUE 1				VALORES DE REFERENCIA TANQUE 2				N					
																														I N T R A M A

Figura 4. 15: Trama serial de 28 registros.

Elaborado por: Investigador.

Entonces el byte 0 es el de inicio de trama; el byte 1 es la información procedente de la EV1 si escribo el valor “1” equivale al encendido del actuador y si escribo el valor “0” es el equivalente al apagado del actuador; el byte 2 es la información procedente de la EV2; el byte 3 es la información originaria de la EV3; los byte 4 y 5 son los datos procedente del motor 1 y motor 2; los bytes 6, 7, 8 equivalen al valor de medida del tanque1 tanto en unidades de litros, porcentaje y metros; los bytes 9, 10, 11 son información procedente del tanque2 correspondiente a la altura del tanque2 en litros, porcentaje y metros; los bytes 12, 13 son la representación de la hora y minuto actual del sistema; el byte 14 equivale a una operación del 0-4; el byte 15 son errores del 0-7; los bytes 16, 17 son la representación de la hora y minuto de reposo que sucede en el tanque1; los bytes 18, 19, 20, 21, 22 son la representación de los valores referenciales de los límites del tanque1; los bytes 23, 24, 25, 26, 27 son la representación de los valores de referencia de los límites de seguridad del tanque2 y finalmente el byte 28 es el fin de trama. En la tabla 4.6 se especifica el contenido de los bytes contenidos de la trama. Por otra parte, el código de programación me muestra en el Anexo H.

Tabla 4. 6: Descripción de los bytes que conforman la Trama que recibe la PC.

DESCRIPCIÓN DE LA TRAMA			
Byte 14		Byte 15	
Operación 0	“Vaciado Tanques”	Error 0	“No hace Nada”
Operación 1	“Producción Automática”	Error 1	“Nivel crítico tanque1”
Operación 2	“Tiempo de Reposo”	Error 2	“Nivel crítico tanque2”
Operación 3	“Retrolavado”	Error 3	“Nivel crítico T1 y T2”
Operación 4	“Sistema Detenido”	Error 4	“Falla Sensor 1”
		Error 5	“Falla Sensor 2”
		Error 6	“Falla T1 y T2”
		Error 7	“Pausa”

Elaborado por: Investigador.

VISUALIZACIÓN LOCAL DE LA INFORMACIÓN

Existe una variedad de dispositivos que permiten la visualización de la información. En la tabla 4.7, se detallan la información de los principales dispositivos de visualización:

Tabla 4. 7: Selección del dispositivo de visualización. [51] [52] [53]

Dispositivos de visualización	Voltaje de alimentación	Corriente	Temperatura de funcionamiento	Número de caracteres
Display 7 segmentos ánodo común	5vdc	30 mA	-40°C a +85°C	1 carácter
Pantalla LCD 20x4	2.7 vdc -5.5 vdc	0.3 mA	-30°C a +75°C	20 caracteres de ancho * 4 caracteres de fila
TPD-433F TouchPAD	+10~+30 vdc	104mA	-20°C~+50°C	Imagen flexible y definida por el usuario

De los tres dispositivos antes mencionados se elige la pantalla LCD 20x4, debido a que la información que visualizo es sencilla y solo maneja información alfanumérica.


Pantalla LCD 20x4

El controlador de pantalla de cristal líquido de matriz de puntos HD44780U muestran información alfanumérica. Caracteres japoneses y símbolos. Se puede configurar para conducir una pantalla de cristal líquido de matriz de puntos bajo el control de un microprocesador de 4 u 8 bits. Especificaciones técnicas: Amplia gama de potencia del controlador de pantalla de cristal líquido 3.0 a 11V, RAM de visualización de 80 × 8 bits (80 caracteres máx.), ROM generador de caracteres de 9,920 bits para un total de 240 fuentes de caracteres, Circuito de reinicio automático que inicializa el controlador, Oscilador interno con resistencias externas, Bajo consumo de energía. [52]

HARDWARE QUE CONTIENE LA PLANTA

La planta ECOAGUA cuenta con este actuador que sirve para el control del paso de agua desde el tanque de agua cruda hacia los filtros de arena-antracita y de carbón activo.

Tabla 4. 8: Motor trifásico P55CS-1247.

	
CARACTERÍSTICAS	
Actuador	Motor eléctrico trifásico
Modelo	P55CS-1247
Frecuencia	60/50 Hz
Potencia mecánica	3 / 4HP
Tensión nominal de operación	208-230 / 460 VOLTS
Régimen de servicio	DUTY CONT
Grado de protección	IP22
Clase de aislamiento	B
Velocidad nominal del motor	3450 RPM
Corriente nominal de operación	2.7 -3.0 / 1.5
Temperatura ambiente máxima	40°C
Factor de servicio	1.5 / 1.0

Elaborado por: Investigador.

Adicional, cuenta con una computadora que sirvió para el monitoreo del sistema de control y alerta del tanque purificador de agua, las principales características se detallan en la tabla 4.9. El motivo de la elección de una computadora de escritorio es que se creó el software de monitoreo, el mismo que es ejecutable para plataforma Windows.

Tabla 4. 9: Características de la computadora.

	CARACTERÍSTICA	
	Sistema operativo	Windows 10 Pro
	Procesador	Pentium Dual-Core CPU
	Memoria RAM	2.00 GB
	Tipo de Sistema	Sistema operativo de 64 bits
	Marca	LG

Elaborado por: Investigador.

4.4.2 Software para el diseño de la interfaz gráfica.

En la tabla 4.10, se realizó una comparación de los principales lenguajes de programación, para el desarrollo de la interfaz gráfica detallando sus características sobresalientes.

Tabla 4. 10: Selección del software. [54][55]

Características	Intouch	Labview	Python
Lenguaje de programación	Por líneas de texto (Ladder)	Gráficamente	Orientada a objetos
Permite aplicaciones de medición, control y automatización	Si	No	Si
Compilador grafico para la ejecución	Si	Si	No
Incorpora otras aplicaciones de otros lenguajes	No	Si	Si
Permite integración con Hardware	Si	Si	Si
Tiene librerías para la adquisición de datos	No	Si	Si

Luego de comparar los softwares de programación se inclinó por trabajar con Python ya que es un software libre. Por lo tanto, mediante el uso de librerías se pudo realizar una interfaz amigable hacia el usuario y lo que es importante no tiene costo ya que no maneja licencia para su instalación y ejecución.

PYTHON

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. Python permite separar tu programa en módulos que pueden reusarse en otros programas en Python. Viene con una gran colección de módulos estándar que puedes usar como base de tus programas, o como ejemplos para empezar a aprender a programar en Python. Algunos de estos módulos proveen cosas como entrada/salida a archivos, llamadas al sistema, sockets, e incluso interfaces a sistemas de interfaz gráfica de usuario como Tk. [55]

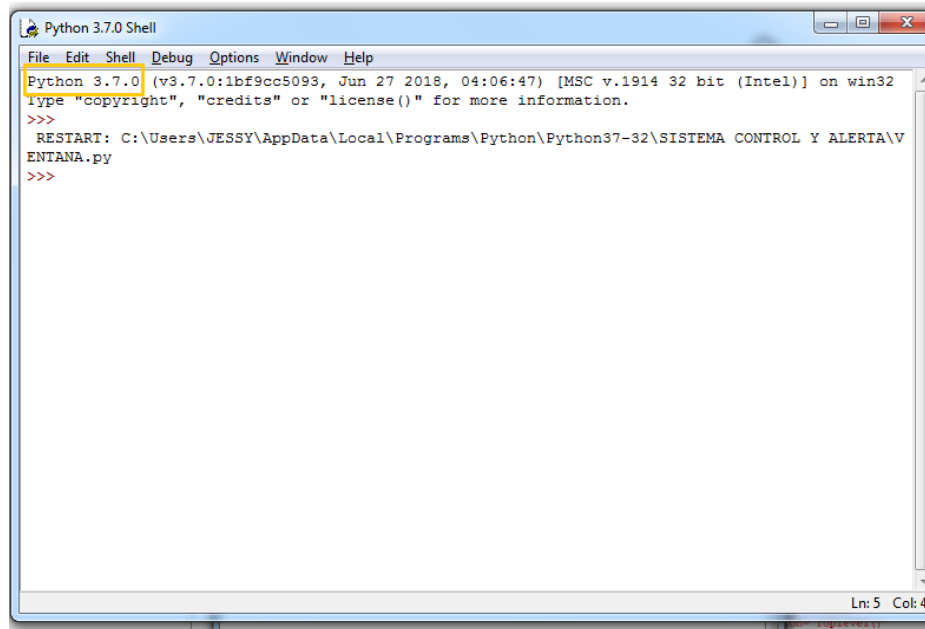


Figura 4. 16: Ventana IDLE de Python.

Elaborado por: Investigador.

IDLE, el entorno de desarrollo de Python

Python es el nombre del lenguaje de programación. IDLE (Integrated Development Environment For Python) es el entorno de desarrollo que permite editar y ejecutar los programas. se pueden crear y ejecutar programas en Python sin utilizar IDLE. Al abrir IDLE con el acceso directo Inicio > Python 3.7.1 > IDLE (Python 3.7.1), se abrirá la ventana principal de IDLE, como muestra la figura 4.16, en él se muestra la versión de Python instalada. [56]

IDLE tiene las siguientes características:

- Codificado en Python 100% puro, utilizando el tkinter kit de herramientas GUI
- Multiplataforma: funciona prácticamente igual en Windows, Unix y Mac OSX
- Ventana de shell de Python (intérprete interactivo) con colores de entrada de código, salida y mensajes de error
- Editor de texto de múltiples ventanas con múltiples opciones para deshacer, colorear Python, sangría inteligente, sugerencias de llamadas, finalización automática y otras características.
- Busque dentro de cualquier ventana, reemplace dentro de las ventanas del editor y busque a través de múltiples archivos (grep)
- Depurador con puntos de interrupción persistentes, pasos y visualización de espacios de nombres globales y locales
- Configuración, navegadores, y otros diálogos. [57]

IDLE tiene dos tipos de ventana principal, la ventana de Shell y la ventana del Editor. Es posible tener múltiples ventanas de editor simultáneamente. Las ventanas de salida, como las utilizadas para Editar/Buscar en archivos, son un subtipo de ventana de edición. Actualmente tienen el mismo menú superior que las ventanas del Editor, pero un título y un menú de contexto predeterminados diferentes. Los menús de IDLE cambian dinámicamente según la ventana seleccionada actualmente. Cada menú documentado a continuación indica con qué tipo de ventana está asociado, [57] como se muestra en la figura 4.17.

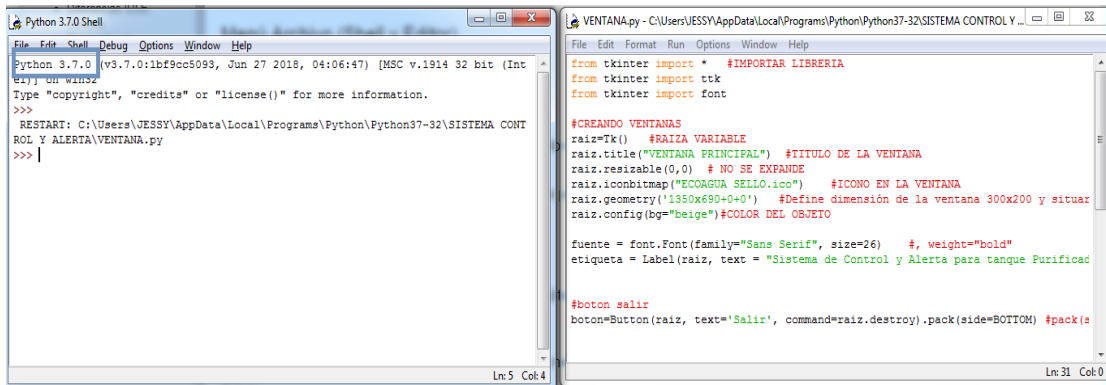


Figura 4. 17: Ventanas Shell y Editor del IDLE de Python.

Elaborado por: Investigador.

Actualmente, los lenguajes de programación se rigen a la norma IEC 61499 fue creado para sistemas de control distribuido, incluyendo su arquitectura y los requisitos de herramientas de software. Uno de los principales objetivos es promover el desarrollo de sistemas heterogéneos compuestos de dispositivos de control de diferentes fabricantes permitiendo la reconfiguración dinámica. IEC 61499 es visto como la siguiente generación de estándares en sistemas de automatización y está diseñado para cubrir interoperabilidad, portabilidad y reconfigurabilidad. La unidad central de esta norma es el Bloque de Función (FB), el cual permite encapsular los algoritmos de control o comunicación y pueden ser escritos en lenguajes propios de IEC-61131 o en otros de alto nivel como JAVA, C++, etc. [58]

4.5 DISEÑO DEL SISTEMA PROPUESTO

El sistema automático fue desarrollado para controlar los niveles adecuados de líquido tanto del tanque de agua cruda como del tanque de agua filtrada, por medio de la activación y desactivación controlada de las electroválvulas y motores conforme a la secuencia de monitoreo y programación del sistema de control y alerta. Como se muestra en la figura 4.18.

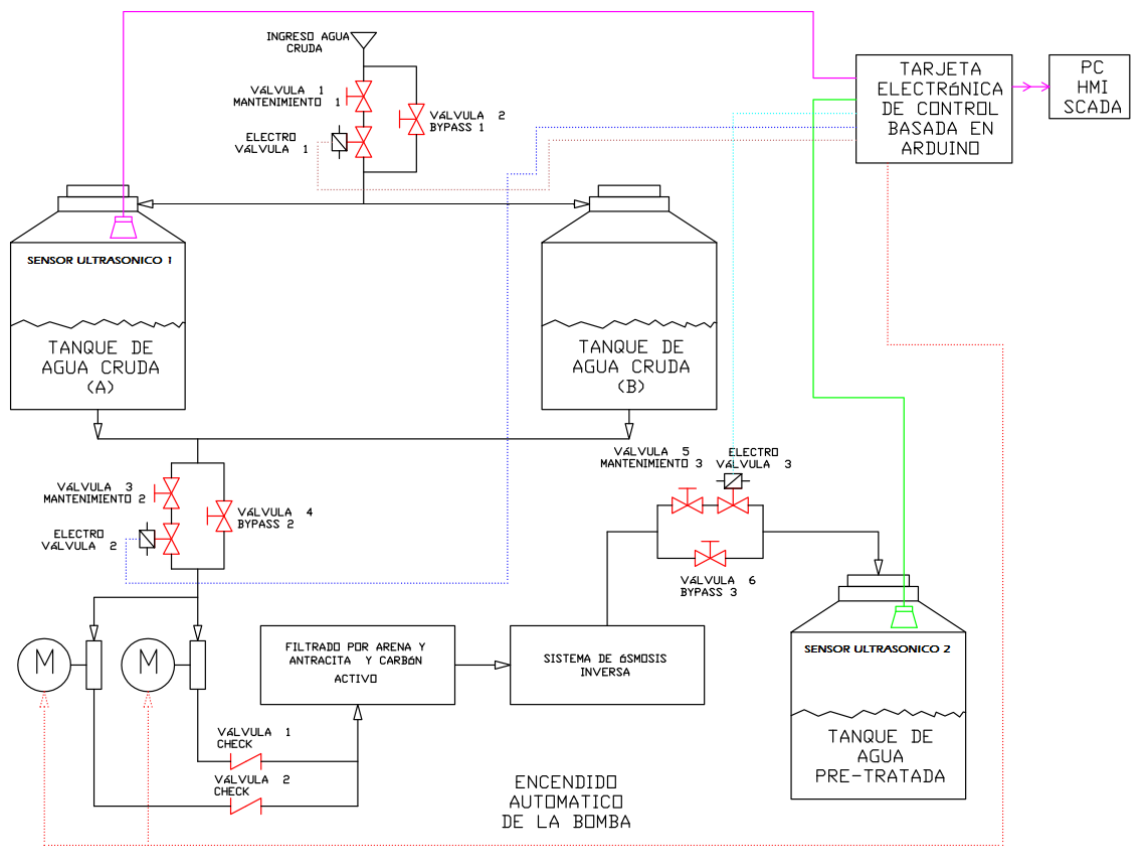


Figura 4. 18: Plano del Sistema de control y alerta para el tanque purificador de agua.

Elaborado por: Investigador.

El proyecto se ejecuta cuando el tanque de agua cruda está vacía el sensor ultrasónico 1 envía una señal a la tarjeta de control informando que hay un nivel bajo de líquido, por tanto la tarjeta de control envía una señal de respuesta a la electroválvula (EV1) para que se active permitiendo que el agua del sistema de red pública ingrese a llenar los tanques, en este caso la electroválvula (EV2) y el motor (MT1 ó MT2) tienen que estar apagados para que no pase el agua al siguiente proceso que es el de filtrado. Una vez que el nivel de agua cruda haya llegado al nivel adecuado el sensor ultrasónico 1 va a detectar este nivel máximo y envía esta señal a la tarjeta de control la misma que responde con una señal a la electroválvula (EV1) para que se apague, adicionalmente las electroválvulas (EV2 y EV3) y el motor (MT1 ó MT2) tienen que estar apagados debido a que el agua cruda debe reposar por dos horas para permitir que los sólidos suspendidos se asienten y quede el agua en un estado de reposo, culminado este tiempo automáticamente se activa la

electroválvula (EV2) y el motor que se encuentre en servicio (MT1 ó MT2) envía el agua cruda al sistema de filtrado conjuntamente se activa la electroválvula (EV3) para que por medio de este empiece el llenado del tanque de agua pre-tratada, como se observa en la figura 4.19.

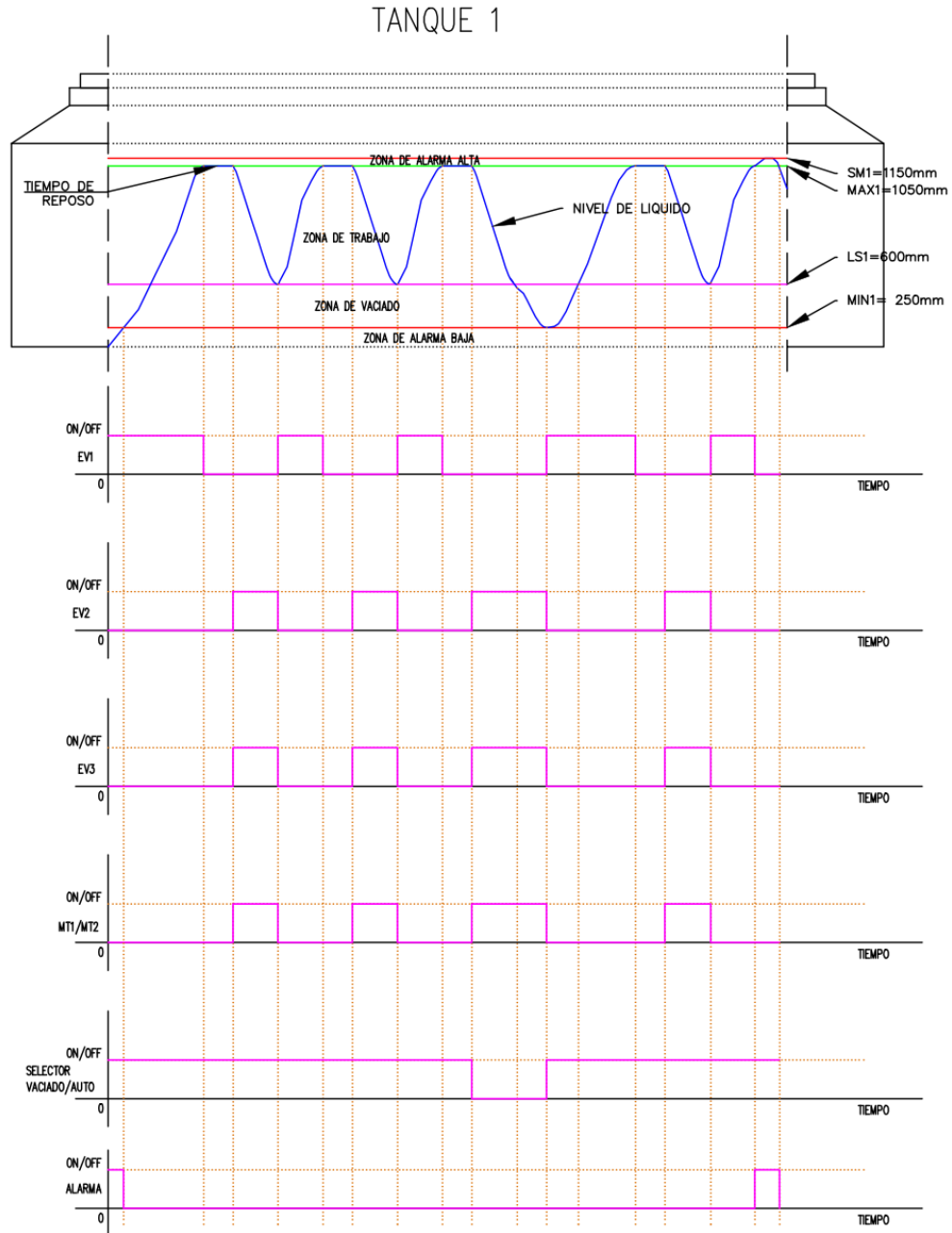


Figura 4. 19: Diagrama secuencial Temporizado del Tanque 1.

Elaborado por: Investigador.

En el tanque de agua pre-tratada se encuentra ubicada el sensor ultrasónico 2, el mismo que se encarga de la detección del nivel de líquido en el tanque, si el nivel de líquido está por debajo del límite superior sigue activo la electroválvula (EV2 y EV3) y el motor que se encuentre en servicio (MT1 ó MT2) para que continúe con la etapa de llenado del tanque, una vez que el sensor ultrasónico 2 detecte el nivel máximo automáticamente va a pagar la electroválvula (EV3) para evitar el paso del agua hacia el tanque de agua pre-tratada; siendo así, queda habilitado la electroválvula (EV2) y el motor (MT1 ó MT2) hasta que la tubería tenga la presión adecuada; una vez que la presión se encuentre en los 40 psi automáticamente el motor (MT1 ó MT2) se apaga debido al presostato con que cuenta el sistema, pero la electroválvula (EV2) seguirá activa siempre y cuando se encuentre el nivel de líquido adecuado en el tanque de agua cruda, surge esta condición de encendido de la electroválvula (EV2) ya que no se controla el sistema de retrolavado de las bombas; entonces siempre debe existir agua y el motor (MT1 ó MT2) debe estar en servicio para que haga el sistema de retrolavado y únicamente se apagan el motor (MT1 ó MT2) y las electroválvulas cuando en el tanque de agua cruda presente un nivel crítico de agua. Por lo tanto, el tanque 1 regresa al ciclo inicial que es activar la electroválvula 1 porque el sensor ultrasónico ya detecta el nivel bajo del tanque. Como se muestra en la figura 4.20.

Adicionalmente, se realizó un plano según la normativa ISA 5.1 que recomienda utilizar ciertas clases de líneas para representar flujos de procesos y diagramas de instrumentos. Además, define símbolos para válvulas, actuadores y otros, etc. Adicional, ISA (INTERNATIONAL SOCIETY OF AUTOMATION) Es una organización internacional que se encarga del desarrollo de estándares relacionados con el mundo de la instrumentación, el control y la automatización en general. Asimismo, proporciona información y publica numerosos libros, revistas y artículos técnicos para divulgar el conocimiento en esta área, con el objetivo de destacar las últimas novedades tecnológicas, tendencias y soluciones reales a los problemas de actualidad en materia de producción e ingeniería. [59]

En el Anexo C se muestra el Plano P&ID del sistema general de control y alerta para el tanque purificador de agua según la Norma ISA 5.1.

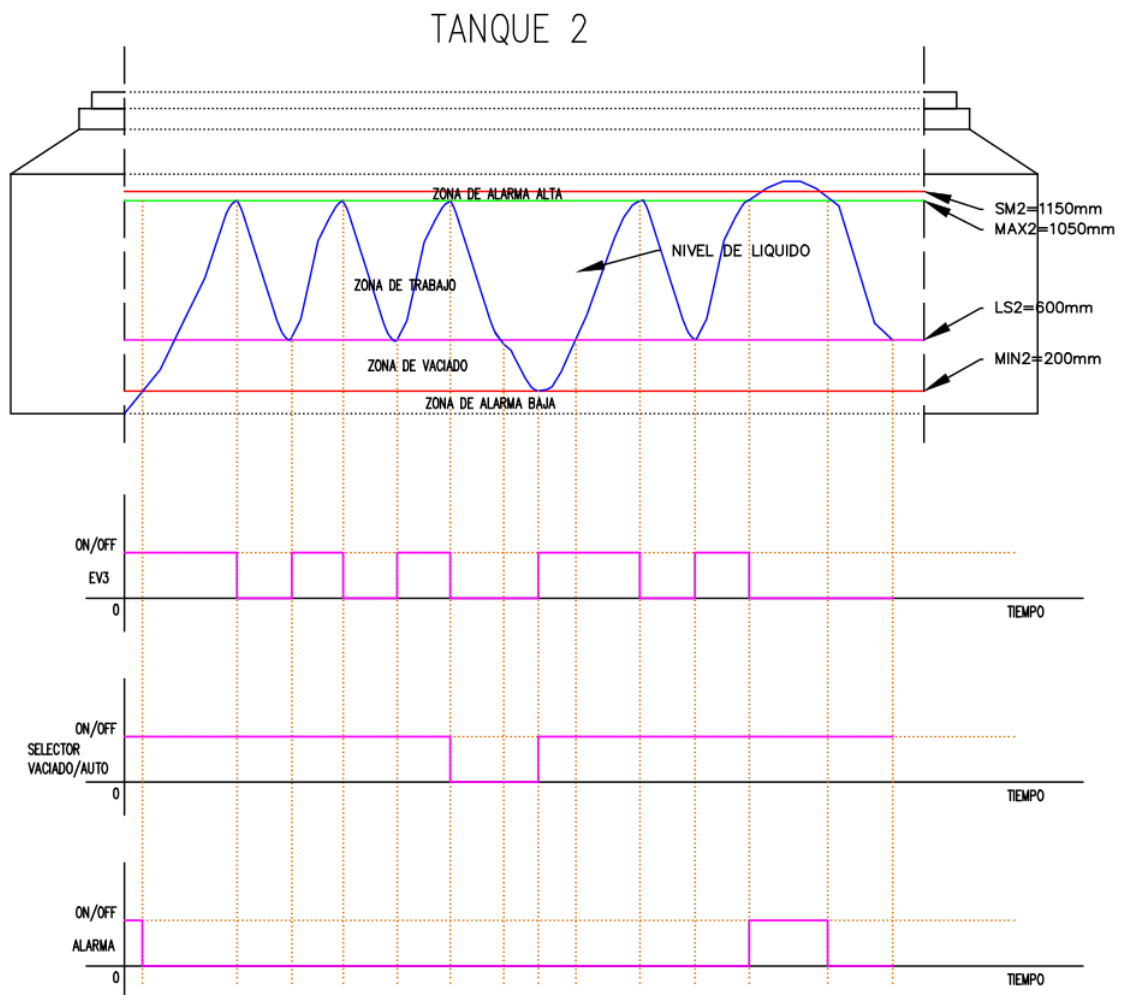


Figura 4. 20: Diagrama secuencial Temporizado del Tanque 2.

Elaborado por: Investigador.

En el diagrama general del sistema se puede observar los dispositivos electrónicos (como: sensor ultrasónico 1, sensor ultrasónico 2, selector 2p, selector 3p, pulsador de emergencia) son señales que ingresan al sistema de control basado en Arduino los mismos que controlan el comportamiento de motor (1,2), electroválvulas (1,2,3) y el sistema de alarma. Como se visualiza en la figura 4.21.

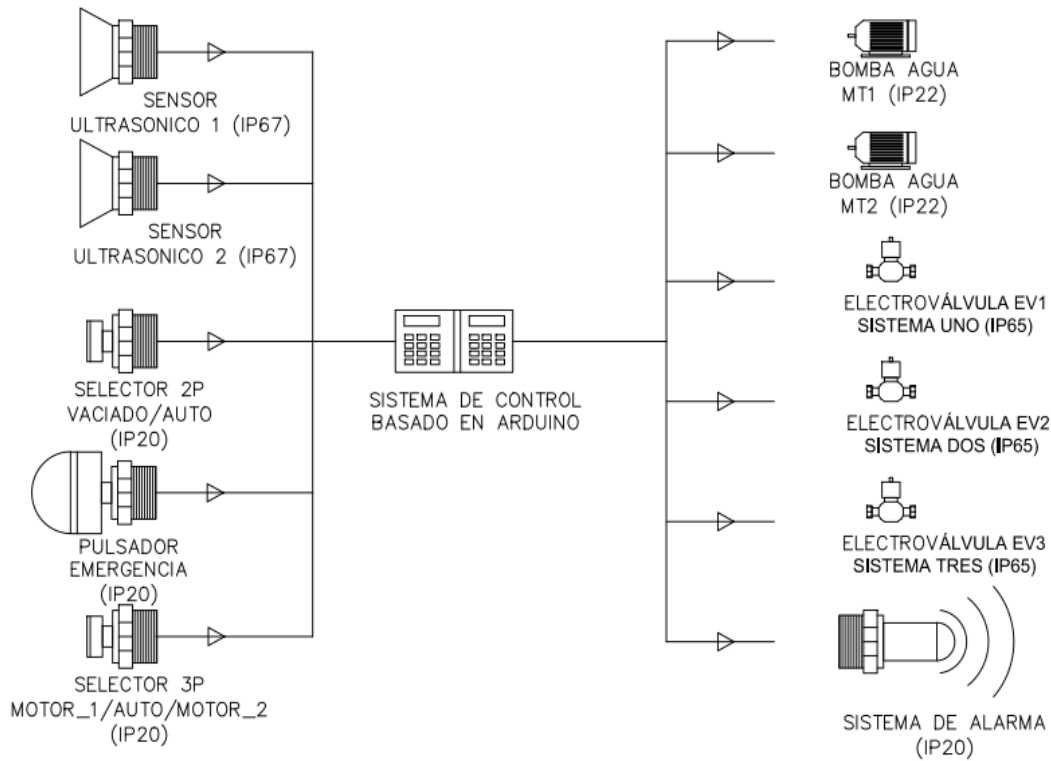


Figura 4. 21: Diagrama general del sistema de control basado en Arduino.

Elaborado por: Investigador.

4.6 DIAGRAMA GENERAL DEL SISTEMA PROGRAMADO

El sistema anuncia con parámetros de inició. Las variables electroválvula (EV1), electroválvula (EV2), electroválvula (EV3) y el motor (MT1 y MT2) deben estar apagados, como se muestra en la tabla 4.11 y los parámetros de configuración como se visualiza en la tabla 4.12. El sistema arranca cuando la perilla del interruptor cambia de apagado a encendido, con esta acción seleccionamos el modo de operación.

Tabla 4. 11: Variables del Sistema.

MARCA	VALOR INICIAL	DESCRIPCIÓN
EV1	Apagado	Electroválvula uno
EV2	Apagado	Electroválvula dos
EV3	Apagado	Electroválvula tres
MT1	Apagado	Relay motor impulsor uno
MT2	Apagado	Relay motor impulsor dos
CICLO_1	0	Secuencia de llenado tanque 1
CICLO_2	0	Secuencia de llenado tanque 2
TIEMPO_1	0	Tiempo de procesos (estado de reposo)
TIEMPO_2	0	Tiempo de procesos (relevo de motores)

Elaborado por: Investigador.

Tabla 4. 12: Parámetros de Configuración.

MARCA	VALOR INICIAL	DESCRIPCIÓN
MAX1	5 (mts) prueba	Límite máximo en tanque 1
MIN1	1 (mts) prueba	Límite mínimo de alarma en tanque 1
LS1	0.90 (mts) prueba	Límite de seguridad bajo tanque 1
SM1	5.10 (mts) prueba	Límite máximo de alarma superior tanque 1
TIEMPO_1	2(horas) prueba	Tiempo para hacer que el agua repose tanque 1
MAX2	5 (mts) prueba	Límite máximo en tanque 2
MIN2	1 (mts) prueba	Límite mínimo de alarma en tanque 2
LS2	0.90 (mts) prueba	Límite de seguridad bajo tanque 2
TIEMPO_2	1 (horas) prueba	Tiempo para relevar la operación de las bombas
SM2	5.1 (mts) prueba	Límite máximo de alarma superior tanque 2

Elaborado por: Investigador.

Modo Vaciado

En esta operación el sistema de purificación de agua debe descargar los tanques tanto de agua cruda como de agua purificada, esta acción permite que los dos tanques se queden sin contenido permitiendo al operador aprovechar para lavar los tanques, o dar mantenimiento a cualquiera de los sistemas involucrados.

Sistema tanque 1.- El procedimiento consiste en apagar la electroválvula (EV1) impidiendo que ingrese agua cruda a los tanques de almacenamiento, si existe el nivel de líquido adecuado por encima de la cota de seguridad mínima la electroválvula (EV2) se activa permitiendo el paso de líquido hacia el motor de impulsión ya sea el (MT1 ó MT2) esto dependerá de la selección de motor de trabajo o de la selección automática que el sistema de control elija, con esto transportamos agua a los tanques de filtrado (filtrado por carbón activado y arenas-antracita y purificado por osmosis inversa), si el nivel de líquido se encuentra por debajo de la zona mínima de seguridad el motor que se encuentre funcionando se apaga automáticamente.

Sistema tanque 2.- El procedimiento para el tanque dos dependerá del nivel de líquido presente en el mismo, si el nivel de líquido está por debajo del nivel superior de agua configurado, automáticamente activa la electroválvula (EV3), hasta que el tanque uno se encuentre vacío o el tanque dos se haya llenado completamente.

Sistema de alarmas. - el sistema de monitoreo de alarmas y control de actuadores está por encima de cualquier orden automática, en este caso el sistema distingue si los niveles de líquido se encuentran por debajo o supera los parámetros de configuración; activando o desactivando las electroválvulas y motores en servicio, precautelando la integridad de los equipos y evitando pérdidas de fluido por derrames.

Modo automático

El sistema automático controla los niveles adecuados de líquido tanto de agua cruda como de agua purificada, por medio de la activación y desactivación controlada de las electroválvulas y motores conforme a la secuencia de monitoreo y programación del sistema de control.

Sistema tanque 1.- El sistema de control del tanque 1 permite llenar el nivel de líquido hasta su cota máxima, una vez alcanzado el nivel óptimo la electroválvula (EV1) se apaga impidiendo que más agua cruda ingrese al tanque 1, ya estando en este punto cierra la electroválvula (EV2) y el motor de impulsión en servicio ya sea el (MT1 ó MT2), el agua antes de ingresar al sistema de filtrado (filtrado por carbón activado y arenas-antracita purificado por osmosis inversa), debe reposar por dos horas permitiendo que los sólidos

suspendidos se asienten y quede el agua en un estado de reposo, terminado este tiempo automáticamente se activa la electroválvula (EV2) y el motor de impulsión en servicio (MT1 ó MT2) enviando el agua cruda a través del sistema de filtrado.

Sistema tanque 2.- Para el tanque dos, se establece el nivel de agua actual a través del sensor de nivel y los compara con los parámetros de configuración, si el nivel está por debajo del límite superior se activa la electroválvula (EV3) siempre y cuando el nivel de líquido presente en el tanque 1 sea el adecuado, una vez lleno el tanque 2 se apaga la electroválvula de servicio (EV3) y a medida del uso de agua en producción se ira descargando el tanque hasta que el nivel llegue al límite de seguridad mínimo del tanque 2, con lo que nuevamente se repita el ciclo de llenado.

Sistema de control de relevo motores. - El sistema de control de motores de impulsión (MT1 y MT2), está controlado tanto para el sistema de “Vaciado de tanques” como para el modo “Automático de llenado”, este sistema consiste en activar cada dos horas de manera alternada el accionamiento de los motores.

Sistema de alarmas

Para el Tanque 1, la alarma se activa cuando el nivel de líquido este por debajo de la cota mínima de seguridad con lo que automáticamente inhabilita el accionamiento de la bomba en servicio; así también cuando el nivel del líquido supere el límite máximo superior activa la alarma por posible desborde y bloquea el sistema de llenado.

Para el Tanque 2, la alarma se activa cuando el nivel de líquido este por debajo de la cota mínima de seguridad del tanque 2 en este caso no se toma acciones de control sobre ningún actuador puesto que no lo requiere, pero para cuando el sistema supere el límite de seguridad superior este automáticamente apaga la electroválvula de llenado (EV3). Como se muestra en la figura 4.22.

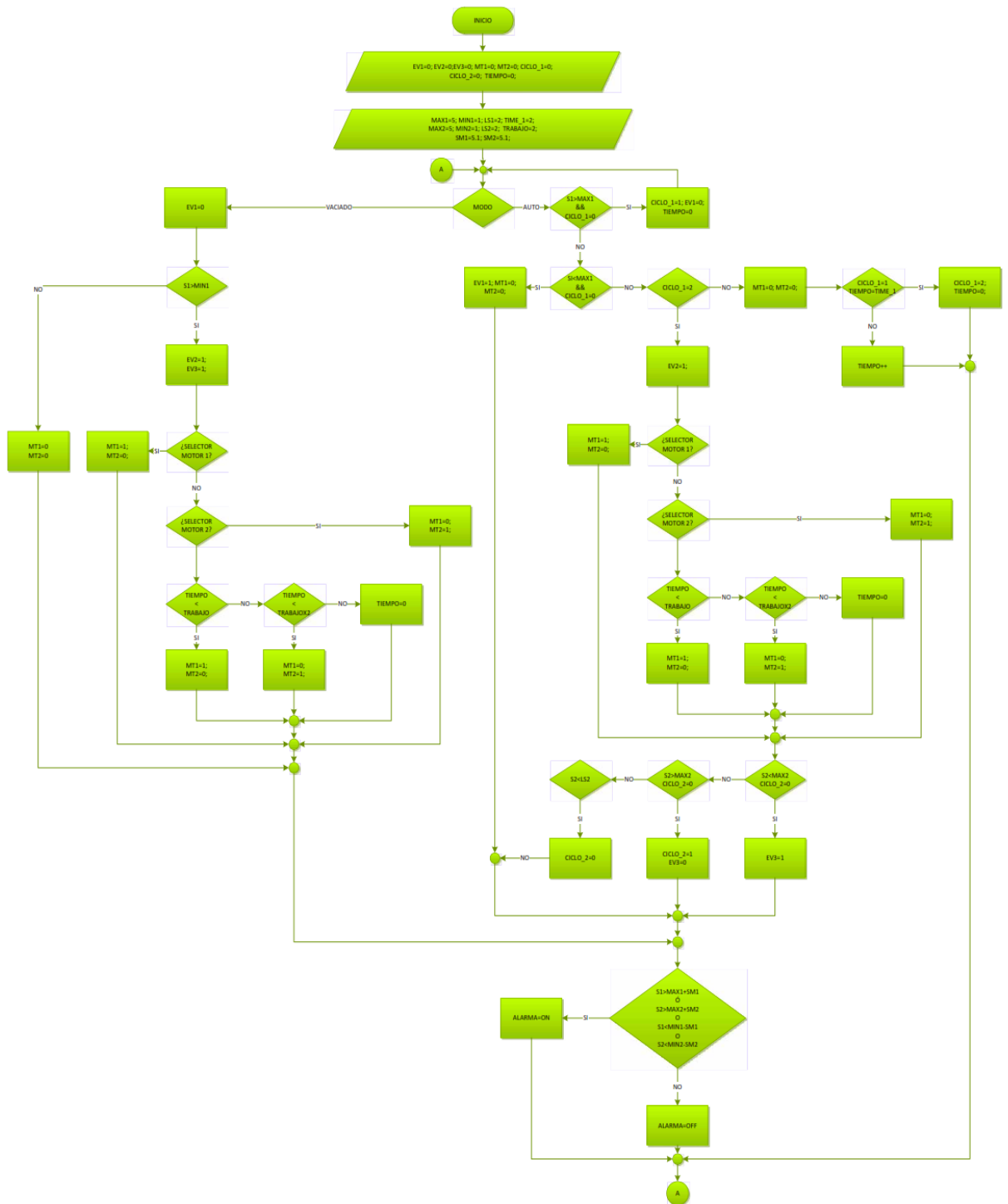


Figura 4. 22: Diagrama de flujo control del sistema de llenado.

Elaborado por: Investigador.

4.7 DESARROLLO DEL SISTEMA

Para el desarrollo del sistema de control y alerta para el tanque purificador de agua, es necesario la instalación de varios componentes como tubería, software las mismas que se detallan a continuación. En el Anexo D, se especifica documentación adicional según la norma IEC 61131.

4.7.1 Implementación del sistema de tuberías.

El cambio de tuberías se realizó en tres secciones que conforman el tanque de purificación de agua de la Planta Purificadora ECOAGUA. Como se muestra en la figura 4.23.



Figura 4. 23: Secciones de tuberías a modificarse en ECOAGUA.

Elaborado por: Investigador.

Como guía se tomó a la normativa ISA 5.1, para el diagrama de tuberías y elementos que conforman las tres secciones que se modificaron. Como se muestra en la Figura 4.24.

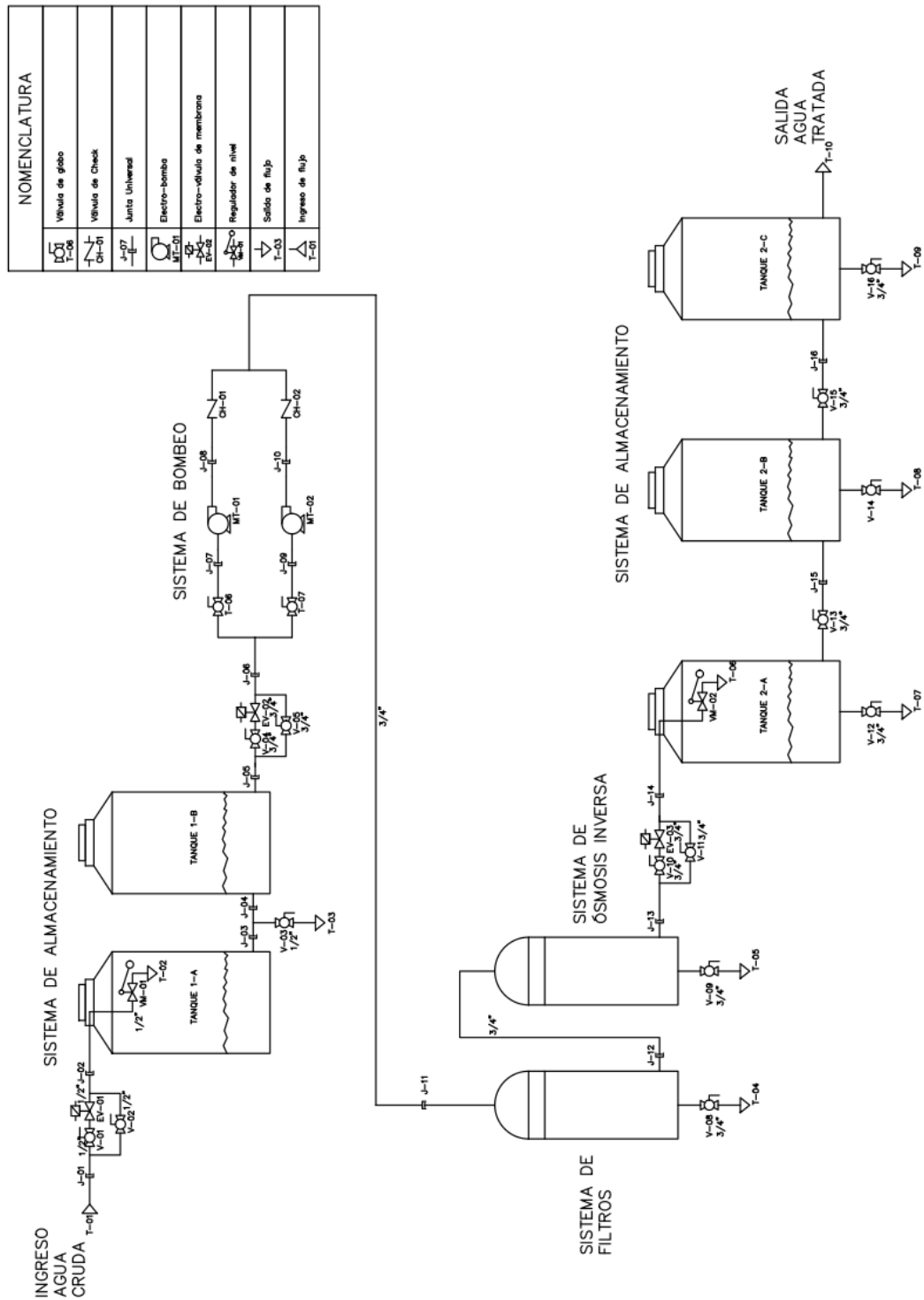


Figura 4. 24: Diagrama Hidráulico del sistema de control y alerta según ISA 5.1.

Elaborado por: Investigador.

Según se observa en la figura 4.25, las tuberías con que contaba la planta ECOAGUA estaban deterioradas y, por ende, necesitaban de un cambio para obtener un sistema automatizado.



Figura 4. 25: Tubería de ingreso de agua de la red pública hacia los tanques de agua cruda, sección 1.

Elaborado por: Investigador.

Entonces, este primer cambio de tubería se realizó usando tubo de agua PVC de 1/2", una electroválvula de 1/2" normalmente cerrada y dos válvulas de paso, como se muestra en la Figura 4.26. Se controló el ingreso de agua de la red pública, para el llenado de los tanques A y B de agua cruda realizando un acondicionamiento de dispositivos en paralelo, para que funcionen independientemente tanto la rama que contiene la válvula de paso 1 y la electroválvula, de la rama que está compuesta por la válvula de paso 2.

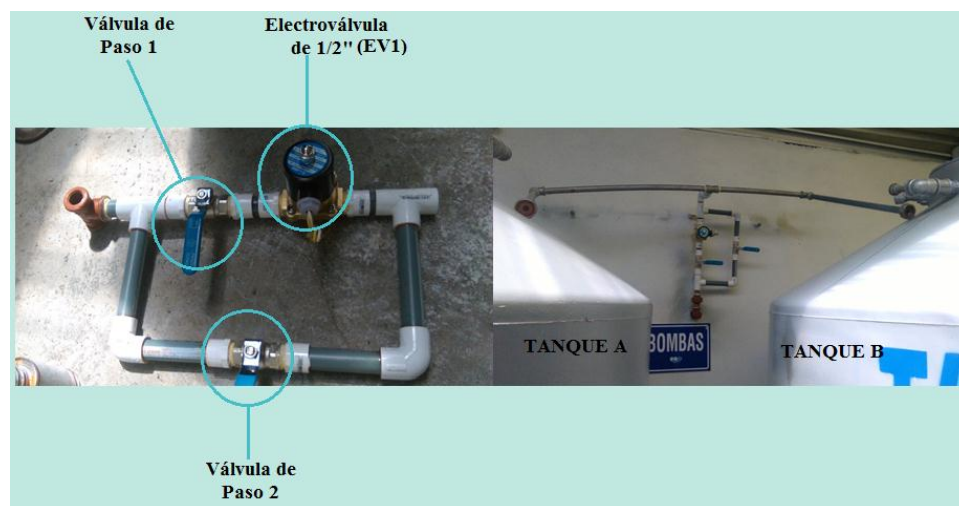


Figura 4. 26: Conexión de tuberías de ingreso de líquido para los tanques de agua cruda, sección 1.

Elaborado por: Investigador.

En la segunda sección se puede ver claramente una fuga de agua como se muestra en la Figura 4.27.



Figura 4. 27: Tubería deteriorada sección 2.

Elaborado por: Investigador.

De la misma manera se implementó un sistema de tubería en paralelo para la sección 2 misma que provee agua hacia los filtros de arena-antracita y carbón activo, esta tubería cuenta con los mismos elementos que se detallaron en la primera parte, la única diferencia es que su medida es de $\frac{3}{4}$ " y adicional cuenta con una válvula de paso auxiliar ubicada en la parte inferior como se muestra en la figura 4.28, que sirve para el desfogue del agua cuando se realiza la limpieza de los tanques A y B de agua cruda.

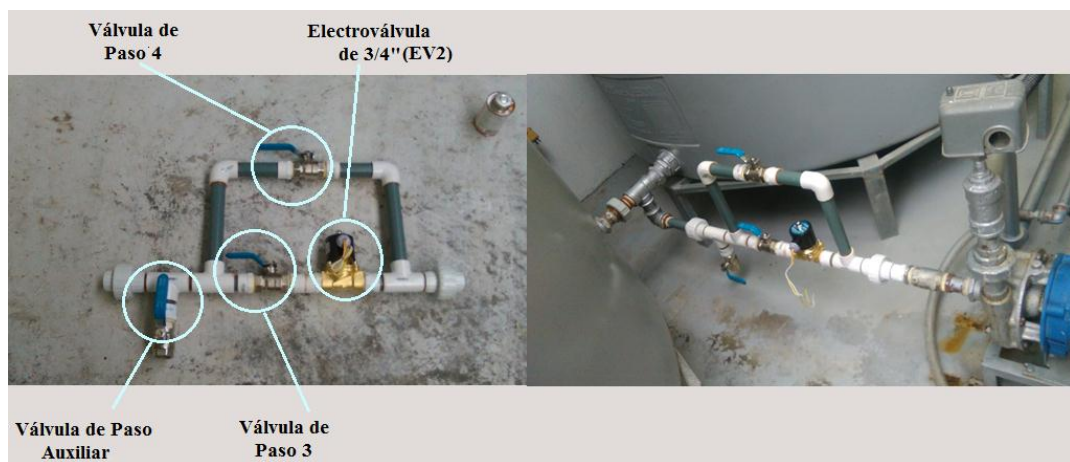


Figura 4. 28: Nueva Tubería sección 2.

Elaborado por: Investigador.

En la sección 3, la tubería se encontró en desperfecto como se muestra en la figura 4.29.



Figura 4. 29: Tubería deteriorada sección 3.

Elaborado por: Investigador.

Del mismo modo se implementó un sistema de tubería en paralelo para la sección 3, la que esta constituida por una tubería PVC de $\frac{3}{4}$ " , y dos válvulas de paso (5 y 6) como se muestra en la figura 4.30, este sistema controla el paso de agua que viene desde los filtros de arena-antracita y carbón activo hacia el Tanque de Agua Filtrada (Osmosis Inversa).



Figura 4. 30: Nueva Tubería sección 3.

Elaborado por: Investigador.

4.7.2 Implementación del sistema de Sensorización.

La implementación del sistema de sensores se ubicó en la parte superior de los tanques de agua cruda y agua filtrada, como se visualiza en la figura 4.31.



Figura 4. 31: Lugares para la implementación de Sensores.

Elaborado por: Investigador.

Se procedió a instalar el sensor ultrasónico MB7369 HRXL-MaxSonar-WRM, el mismo que está compuesto de 7 Pines, pero para nuestro proyecto solo se consideraron 3 Pines (5-6-7) siendo ellos de salida serial, alimentación y GND respectivamente. Se realizó un pequeño circuito para evitar averías en los pines del sensor como se muestra en la figura 4.32. Este sensor es resistente a la intemperie diseñado para la detección de objetos y su alcance en exteriores o en aplicaciones de tanques. La robusta carcasa de PVC está diseñada para cumplir con los estándares IP67. El cono del sensor en este modelo le da un ancho de haz muy estrecho, ideal para detectar objetos directamente en frente del sensor, es de fácil montaje ya que su terminado en rosca coincide con los accesorios de tubería PVC de $\frac{3}{4}$ ".



Figura 4. 32: Circuito para evitar averías en los Sensores.

Elaborado por: Investigador.

Para el acoplamiento en los tanques se usó un codo de ¾”, una prensa stop, cable UTP categoría 6E y un tubo PVC de 6 cm. Como se muestra en la figura 4.33.



Figura 4. 33: Conexión del sensor ultrasónico.

Elaborado por: Investigador.

El cable UTP consta de 4 pares de cables (café, azul, verde y naranja) los mismos que se conectan con el sensor como se muestra en la tabla 4.13.

Tabla 4. 13: Conexión de sensor MB7369 HRXL-MaxSonar con el cable UTP.

Conexión del cable UTP con el Sensor MB7369 HRXL-MaxSonar-WRM	
Colores del cable UTP	PINES del Sensor MB7369
-----	Pin 1: Conexión del sensor de temperatura
-----	Pin 2: Salida de ancho de pulso
-----	Pin 3: Salida de voltaje analógico.
-----	Pin 4: Rango de inicio / parada
Blanco verde - Verde	Pin 5: Salida serial.
Blanco café – Café	Pin 6: Alimentación
Blanco azul - Azul	Pin 7: GND. Conexión a tierra.
Blanco naranja - Naranja	-----

Elaborado por: Investigador.

La calibración del sensor ultrasónico se realizó con respecto a la altura del tanque, en la figura 4.34 se puede apreciar que el sensor ultrasónico 1 se encuentra instalado a 1450 mm desde la base del tanque, entonces para hallar el valor de la altura del líquido la

formula será: $h_{agua} = \text{altura del tanque} - a$. Siendo “a” la dimensión entre el sensor y el líquido

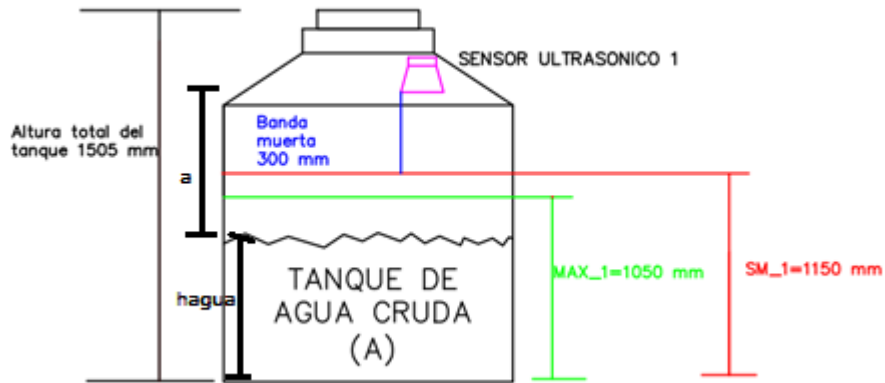


Figura 4. 34: Diagrama de calibración del sensor ultrasónico.

Elaborado por: Investigador.

Se procedió a la ubicación de los sensores ultrasónicos en el tanque de Agua Cruda y tanque de Agua Filtrada (Osmosis Inversa), considerando que por motivos de mantenimiento de los tanques se dejó una reserva de 2m de cable, como se observa en la figura 4.35.



Figura 4. 35: Sensores MB7369 HRXL-MaxSonar-WRM instalados.

Elaborado por: Investigador.

4.7.3 Construcción del Tablero eléctrico industrial.

Se encuentra ubicado junto al tanque de agua cruda A, el gabinete metálico tiene un tamaño de (60x40) cm, dentro de este se colocan los dispositivos de conexión, control, maniobra, señalización, etc., todos estos dispositivos permiten que la instalación eléctrica funcione de forma correcta. Para el acople del gabinete en la pared fue necesario observar los planos de la planta para evitar perforar por error alguna tubería que se encuentre en ella. Este gabinete fue aislado de la pared para evitar algún contacto con la humedad o algún agente nocivo. Como se muestra en la figura 4.36.

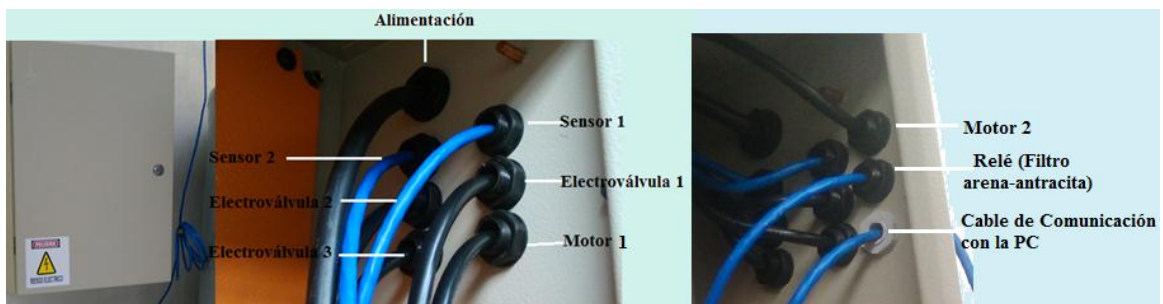


Figura 4. 36: Gabinete Metálico.

Elaborado por: Investigador.

Las características eléctricas con que trabaja el cajetín son 110v con puesta a tierra aislada (tierra aislada es la conexión que se realiza entre el chasis, todo lo que es tuberías y lo que tenga que ver en contacto con el usuario la misma que se dirige hacia la tierra, la misma que se encarga de proteger al usuario y también a los equipos de la interferencia electromagnética que existe en ambientes industriales), el gabinete metálico se encuentra alimentado con cable concéntrico 3x12, que se conecta directamente al gabinete principal, como se muestra en la figura 4.37, la conexión obedece a los siguientes colores: verde (tierra exclusivo), rojo (fase) y blanco (neutro) los mismos que se incorporan al tablero principal de distribución a través de un breaker de 20 A.



Figura 4. 37: Tablero principal de distribución.

Elaborado por: Investigador.

Al gabinete ingresar 10 cables, que sirven para realizar el acondicionamiento, control y protección del sistema como se observa en la tabla 4.14.

Tabla 4. 14: Conexión de cables.

Conexión de Cables	
Cable	Conexión
Concéntrico 3x12	Tablero principal (Alimentación)
UTP	Conexión sensor 1
UTP	Conexión sensor 2
UTP	Relé 110vca
UTP	Comunicación a la PC
Concéntricos 2x16	Electroválvula 1
Concéntricos 2x16	Electroválvula 2
Concéntricos 2x16	Electroválvula 3
Concéntricos 2x16	Motor 1
Concéntricos 2x16	Motor 2

Elaborado por: Investigador.

Este tablero eléctrico secundario está protegido de la electricidad por un breaker diferencial 32 A (encargado de la protección a usuarios), dentro del gabinete se encuentran instalados, canaletas ranuradas, breaker diferencial, borneras, porta fusibles, breaker 32 A, 2 relay, borneras de paso 20 pines, fuente de paso 6 pines, fuente de 24 VCD, relay para electroválvulas, toma de corriente 110VCD. Como se visualiza en la figura 4.38.

El acoplamiento de la fuente beneficia en la estabilización del circuito para evitar algún daño, el voltaje de entrada a la fuente es de 110 vcd y la que genera hacia el circuito es de 24 V a 4.7 A.

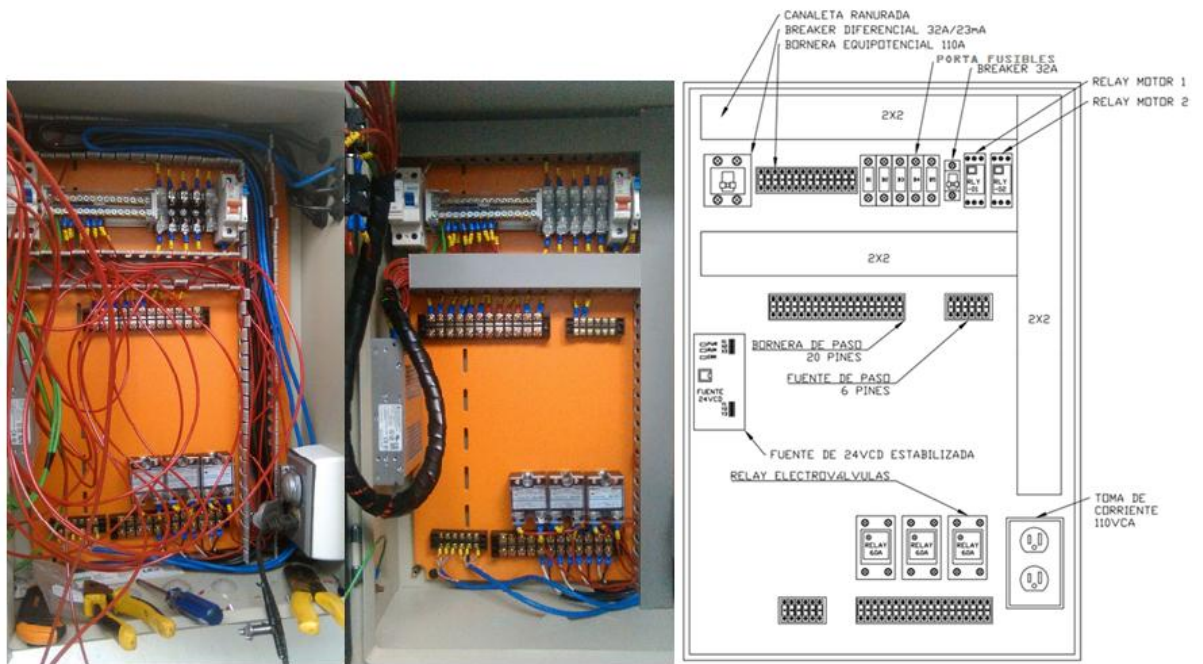


Figura 4. 38: Tablero eléctrico interno.

Elaborado por: Investigador.

Norma ANSI/TIA-606-B es una norma voluntaria que proporciona directrices para etiquetado uniforme y mantenimiento de registros para sistemas de telecomunicaciones y redes en todo tipo de plantas. Algunas de las principales directrices para producir etiquetas incluyen: El tamaño, color y contraste de las etiquetas debe permitir que los identificadores se puedan leer fácilmente y que estén visibles durante el mantenimiento normal, las etiquetas deben ser resistentes a las condiciones ambientales en el punto de

instalación y deben durar lo mismo o más que el componente etiquetado, el texto en las etiquetas debe ser generado por una máquina. [60]

Requisitos para las etiquetas

- Todas las etiquetas deben usar un identificador permanente que se pueda rastrear fácilmente, este esquema debe ser lógico en la organización, utilizando caracteres alfanuméricos para facilitar la referencia.
- Cada cable y cada vía deben estar etiquetados en cada extremo, y cada etiqueta debe identificar los puntos de terminación de ambos extremos del cable.
- Todas las etiquetas deben cumplir con los requisitos de legibilidad, desfiguración y adherencia. [61]

En la figura 4.39 se ilustra las conexiones que se efectúan en las borneras superiores del tablero eléctrico, adicional la simbología se especifica en la tabla 4.15.

Tabla 4. 15: Conexión de las Borneras Superiores.

SIMBOLOGÍA	CONEXIONES
E1	Paro de emergencia
E2	Selector amarillo (VACÍO)
E3	Selector verde (AUTO)
F4	Luz piloto de EV1
F5	Luz piloto de EV2
G6	Luz piloto de EV3
G7	Luz piloto de VACIO
G8	Luz piloto de AUTOMATICO
H9	Luz piloto de Paro de Emergencia
T9	Selector MT1
T8	Selector MT2
S9	Luz piloto MT1
M3	Alimentación 24v
M4	Negativo
O8	Luz piloto MT2

Elaborado por: Investigador.

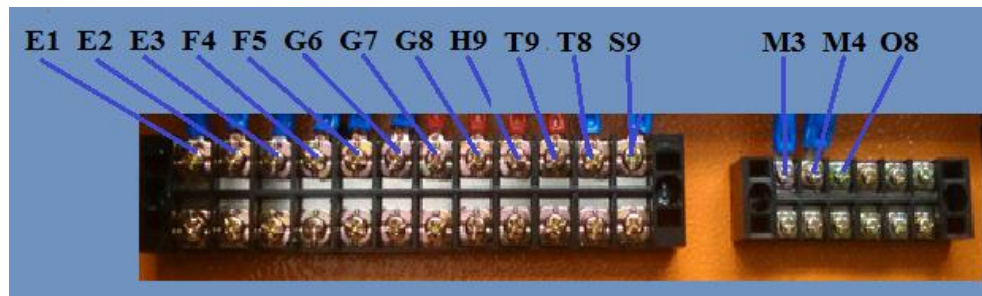


Figura 4. 39: Borneras y fuentes de paso superiores.

Elaborado por: Investigador.

A las borneras inferiores se conectan los pines del sensor 1 y sensor 2, relé para el motor 1 y 2, los cables de fase y neutro de las electroválvulas 1, 2 y 3. Como se muestra en la figura 4.40. adicional se detalla la tabla 4.16, en la que se especifica la conexión de los cables de los sensores de nivel.

Tabla 4. 16: Conexión de los sensores en las Borneras inferiores.

Sensor 1	Pin 1	Café	Alimentación	Sensor 2	Pin 4	Café	Alimentación
	Pin 2	Verde	Señal		Pin 5	Verde	Señal
	Pin 3	Azul	GND		Pin 6	Azul	GND

Elaborado por: Investigador.

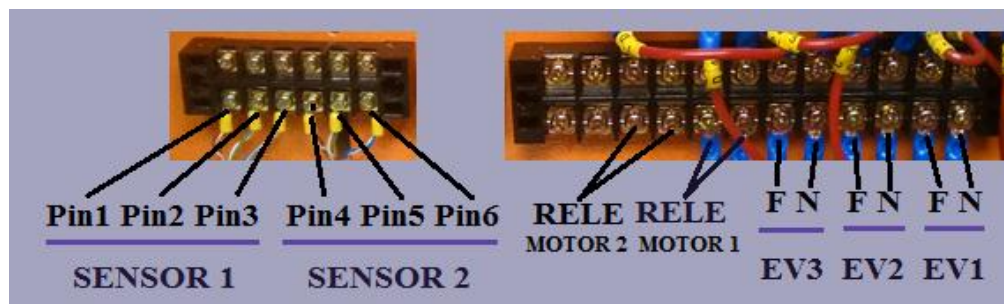


Figura 4. 40: Borneras inferiores.

Elaborado por: Investigador.

En el Anexo E, se puede observar el diagrama de potencia del tablero eléctrico.

En el tablero eléctrico frontal se colocaron los indicadores para la visualización de los procesos que ocurren en el tanque purificador de agua. Luz piloto energizado, luz piloto de electroválvula 1, 2 y 3, luz piloto audible (alarma), selector on/off de alarma, selector on/off del sistema general, luz piloto de vaciado, selector de operaciones, luz piloto automático, pulsador paso de emergencia, luz piloto de pausa, luz piloto bomba (1 y 2), selector de bombas y pantalla LCD. Como se muestra en la figura 4.41. adicional en la tabla 4.17, se describe la funcionalidad que cumple cada indicador.

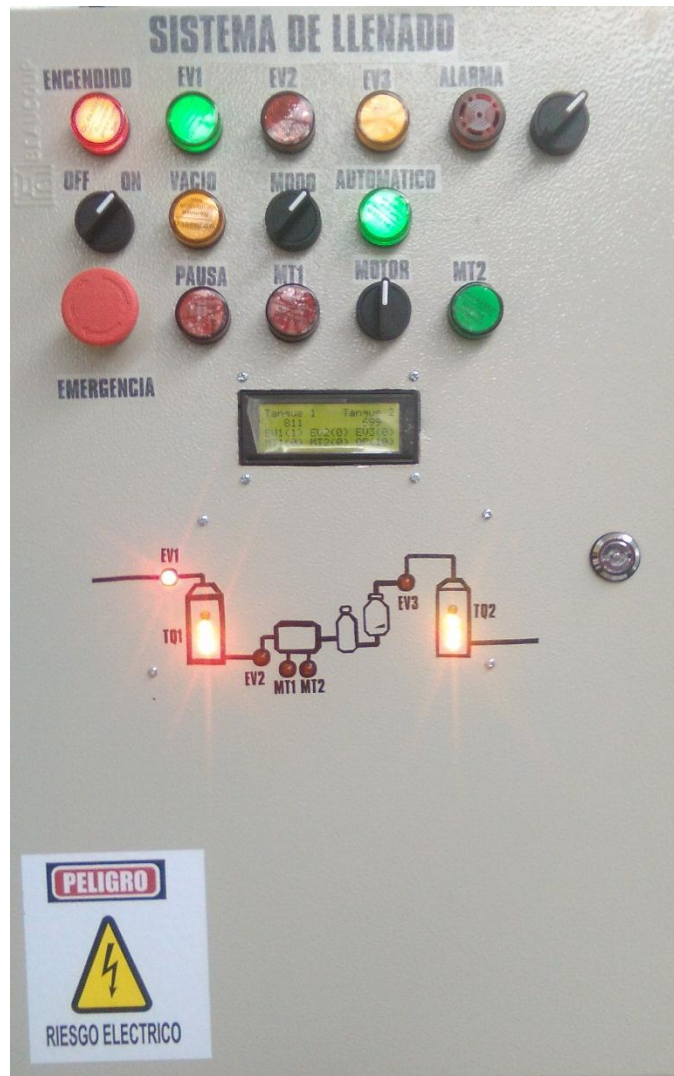


Figura 4. 41: Indicadores del tablero eléctrico frontal.

Elaborado por: Investigador.

Tabla 4. 17: Funciones que desempeñan los indicadores del tablero eléctrico frontal

Indicador	Función
Luz piloto energizado	Se enciende cuando el tablero eléctrico esta energizado, y solo se apagará cuando exista un corto circuito, se dispare el relé que lo controla o por falta de energía eléctrica.
Luz piloto EV1	Este indicador se activa cuando la electroválvula uno está encendido.
Luz piloto EV2	Este indicador se activa cuando la electroválvula dos está encendidas.
Luz piloto EV3	Este indicador se activa cuando la electroválvula tres está encendidas.
Luz piloto audible-alarma	Este se activa cuando en el modo automático el nivel de líquido está por debajo del límite de seguridad baja, cuando el líquido supera el límite máximo de agua en el tanque 1 y tanque 2 y cuando existe demora en el cumplimiento del tiempo de llenado de los tanques.
Selector dos posiciones	Permite activar o desactivar la alarma audible (girando a la derecha el selector se activa y calla la alarma)
Selector dos posiciones	Es el interruptor de encendido y apagado del tablero eléctrico, es el que no se encuentra conectado a la tarjeta de control.
Luz piloto Vaciado	Este trabaja cuando se encuentra seleccionado la posición de vaciado del selector, y se activa cuando los tanques 1 y 2 están vacíos.
Selector de operación	Este se puede manipular para elegir la forma del trabajo del sistema de control y alerta: cuenta con tres posiciones vacío, detenido y automático.
Luz piloto Automático	Cuando se enciende muestra que el sistema de control y alerta está trabajando en modo automático.
Pause/Paro de Emergencia	Al presionar este pulsador se detiene todo el sistema no importa qué modo de operación se encuentre.
Luz piloto Bomba 1	Se activa cuando el selector este ubicado en el modo de operación para que trabaje la bomba uno. Alterna el trabajo cada 10 minutos.
Selector Bombas	Este es el que controla la alternabilidad del trabajo que realizará la bomba uno o dos.
Luz piloto Bomba 2	Se activa cuando el selector este ubicado en el modo de operación para que trabaje la bomba dos. Alterna su trabajo cada 10 minutos.

Elaborado por: Investigador.

En el tablero eléctrico frontal se conecta una LCD de 20x4, la misma que recibe información de la tarjeta de control, despliega contenido acerca de los niveles del tanque de agua cruda y agua filtrada, muestra el estado de las electroválvulas, motores, operaciones y errores dentro del proceso de llenado, esta LCD recibe instrucciones del PIC 18F2550 que es el encargado de realizar la decodificación de la trama serial que recibe y la envía a visualizar información del proceso en la LCD y en los indicadores luces Led presentando así una interfaz HMI (terminal de operador) local. Como se muestra en la figura 4.42. Para esta implementación se realizó perforaciones en el tablero eléctrico frontal. Para saber información detallada de la pantalla LCD 20x4, observar [51].

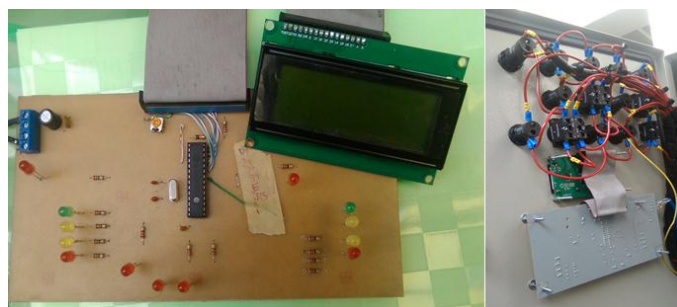
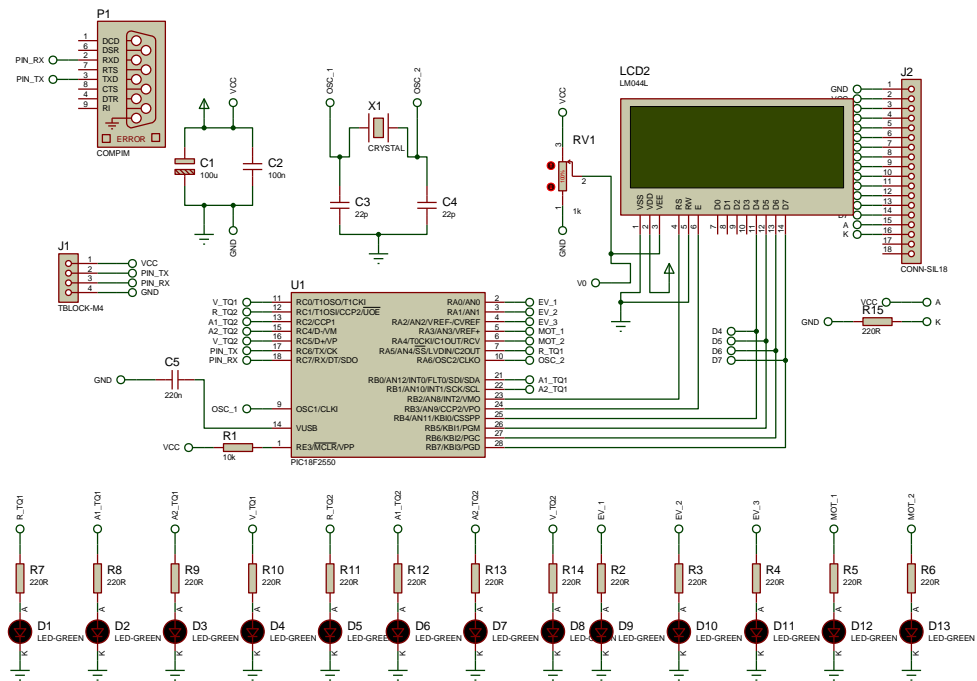


Figura 4. 42: Diagrama de conexión e instalación de la interfaz HMI

Elaborado por: Investigador.

Trama que se envía a la pantalla LCD: La trama que se maneja es una trama serial con 26 registros cada uno equivalente a un byte, y cada byte equivale a una operación, sabiendo que un registro es un número que va desde 0-255. Como se muestra en la figura 4.43.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	REGISTROS
*	1/0	1/0	1/0	1/0	1/0	1/0	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	0/9	#	TRAMA
INICIO TRAMA	EV 1	EV 2	EV 3	MT 1	MT 2	TANQUE 1				TANQUE 2				OPERACIÓN	ERROR	HORA ACTUAL		MINUTO	N1	N2	HORA REPOSO		MINUTO	FIN TRAMA				

Figura 4. 43: Trama serial de 26 registros.

Elaborado por: Investigador.

Entonces el byte 0 es el de inicio de trama, el byte 1 es la información procedente de la EV1 si escribo el decimal “1” equivale al encendido del actuador y si escribo el decimal “0” es el equivalente al apagado del actuador, el byte 2 es la información procedente de la EV2, el byte 3 es la información originaria de la EV3, el cuarto y quinto byte es el dato procedente del motor 1 y motor 2. Los bytes 6, 7 8, 9 equivalen al valor de medida del tanque1 cada uno de estos bytes corresponden a las unidades de miles, centenas, decenas y unidad, de la misma forma los bytes 10, 11, 12, 13 son información procedente del tanque2, el byte 14 equivale a una operación del 0-4. El byte 15 son errores del 0-7.

Los bytes 16 y 17 son la representación de la hora actual del sistema, los bytes 18 y 19 son la representación de los minutos actuales del sistema. El byte 20 es la información procedente del nivel del tanque1 y varía desde 0-8 los mismos que permiten la visualización en los indicadores Luces Led. El byte 21 es la información procedente del nivel del tanque2 y varía desde 0-6.

Los bytes 22, 23, 24 y 25 es el minuto/segundo de fin de ciclo, este almacena el tiempo de llenado del tanque1 y le acumula las dos horas que son las de reposo del tanque de agua cruda. y finalmente el byte 26 es el fin de trama. En la tabla 4.18 se especifica el contenido de los bytes. El código de programación del PIC 18F2550 se muestra en al Anexo F.

Tabla 4. 18: Descripción de los bytes que conforman la Trama que llega a la LCD.

Descripción de la Trama			
Byte 14			
Operación 0	"Vaciado Tanques"		
Operación 1	"Producción Automática"		
Operación 2	"Tiempo de Reposo"		
Operación 3	"Retrolavado"		
Operación 4	"Sistema Detenido"		
Byte 15			
Error 0	"No hace Nada"		
Error 1	"Nivel crítico tanque1"		
Error 2	"Nivel crítico tanque2"		
Error 3	"Nivel crítico T1 y T2"		
Error 4	"Falla Sensor 1"		
Error 5	"Falla Sensor 2"		
Error 6	"Falla T1 y T2"		
Error 7	"Pausa"		
Byte 20		Byte 21	
0	Rojo OFF	0	Rojo OFF
1	Rojo Pulsante	1	Rojo Pulsante
2	Rojo Estático	2	Rojo Estático
3	Amarillo1 Pulsante	3	Amarillo Pulsante
4	Amarillo1 Estático		
5	Amarillo1 Pulsante	4	Amarillo Estático
6	Amarillo2 Estático		
7	Verde Pulsante	5	Verde Pulsante
8	Verde Estático	6	Verde Estático

Elaborado por: Investigador.

4.7.4 Construcción y configuración del sistema de control.

La tarjeta de control simula a un PLC, ya que cuenta con una fuente de alimentación, memoria, módulo de entrada, módulo de salida, puertos de comunicación y una unidad central de proceso, todos ellos en conjunto son los encargados realizar la automatización de los procesos de control y alerta del tanque purificador de agua. En la figura 4.44 se muestra el diagrama electrónico de la tarjeta de control.

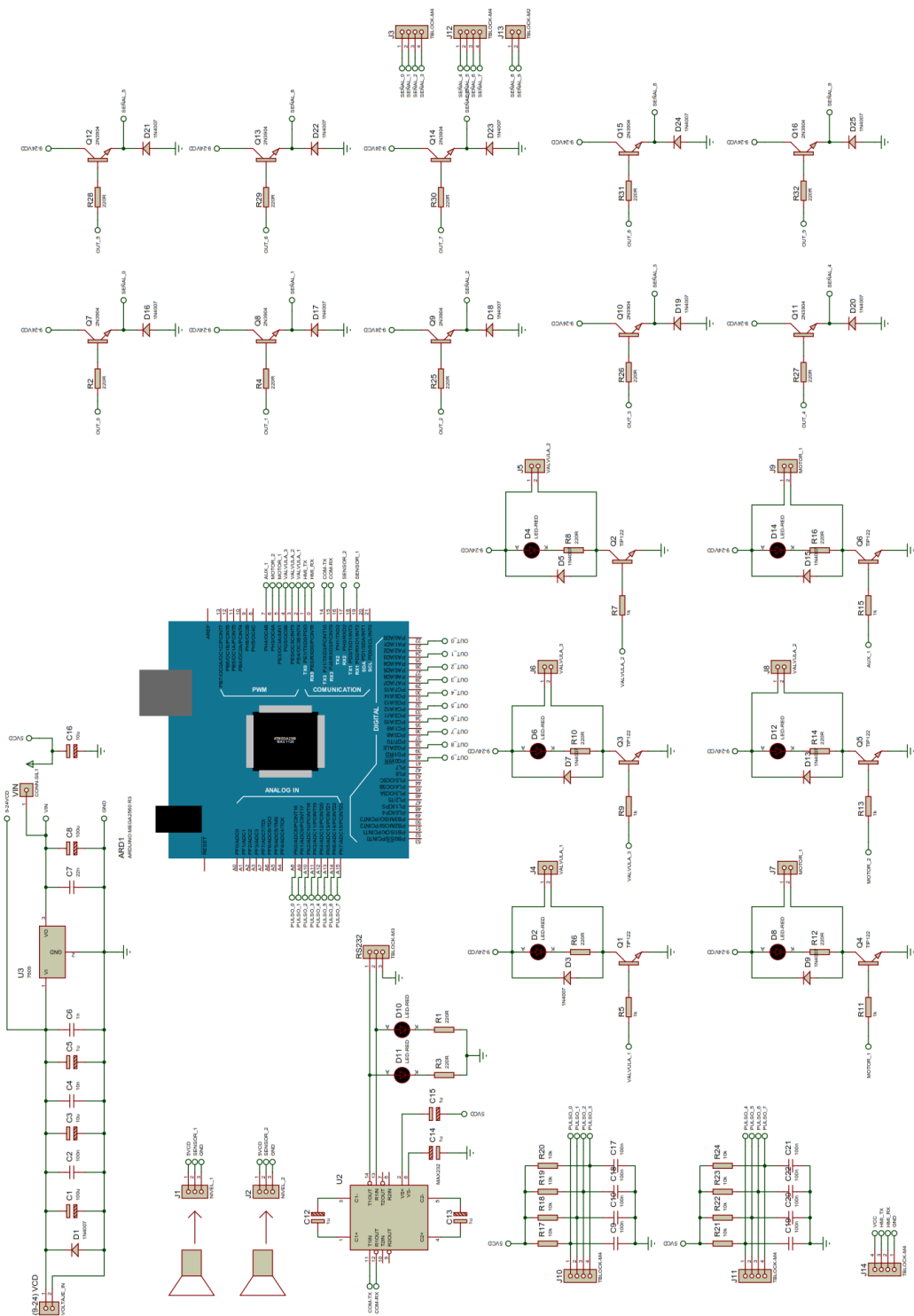


Figura 4. 44: Diagrama Electrónico de la tarjeta de control.

Elaborado por: Investigador.

Adicional en el ANEXO G se puede ilustrar las conexiones de la tarjeta de control. La tarjeta de control contiene pines de alimentación de (5-24)V; salida de comunicación serial para la segunda tarjeta que cumple la función de interfaz HMI local; puertos para el ingreso de señal desde los sensores (1 y 2), electroválvulas (EV1, EV2 y EV3), motores (MT1 y MT2) y un auxiliar que se conecta a la luz piloto audible-alarma; puertos para el ingreso de señal desde los botones de control del tablero eléctrico frontal como son: selector de modos (vacío y automático), pulsador paro de emergencia, selector de motores (MT1 y MT2), señal del relé que se encuentra en el filtro de arena-antracita y adicional dos 2 auxiliares; contiene también borneras para señales de salida dirigidas hacia las luces pilotos del tablero eléctrico frontal de: electroválvulas (EV1, EV2 y EV3), modos (vacío, automático y pausa), motores (MT1 y MT2), y dos auxiliares. También se encuentra incorporado un puerto para el envío de información hacia la Pc. Esta tarjeta está controlada por una Arduino Mega 2560 interactúa junto a él, un módulo de tiempo real (DS1307 Tiny RTC) el mismo que mantiene la fecha y hora actual en caso de la ausencia de luz eléctrica debido al ruido eléctrico presente, este módulo no se desconfigura ya que cuenta con una fuente auxiliar (pila) para su alimentación independiente. Como se muestra en la figura 4.45.

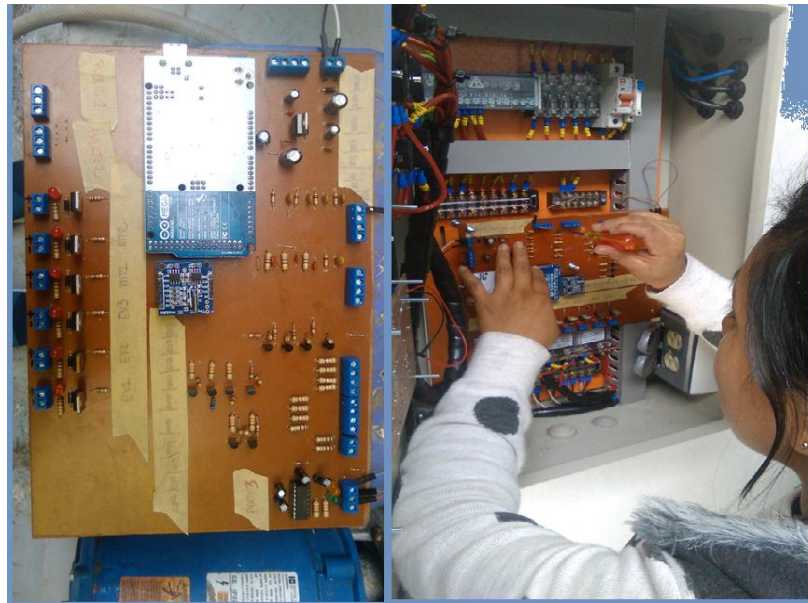


Figura 4. 45: Instalación de la Tarjeta de Control.

Elaborado por: Investigador.

Para efectuar los cálculos de Nivel del agua en litros se utilizó ecuación 1.

Datos:

$$ht_tanque_1 = 1505 \text{ mm}$$

$$diametro_1 = 1550 \text{ mm}$$

$$ht_tanque_2 = 1485 \text{ mm}$$

$$diametro_2 = 1550 \text{ mm}$$

$$tanque_1 = 1505 \text{ mm}$$

$$tanque_2 = 1485 \text{ mm}$$

Donde:

- **ht_tanque_1**= altura referencial del tanque 1
- **diametro_1**= diámetro referencial del tanque 1
- **tanque_1**= valor real de líquido en el tanque 1
- **ht_tanque_2**= altura referencial del tanque 2
- **diametro_2**= diámetro referencial del tanque 2
- **tanque_2**= valor real de líquido en el tanque 2

El volumen del líquido que contiene el tanque agua cruda y agua filtrada se halla con la fórmula del cilindro, se utiliza la ecuación 3.

$$V = \pi * r^2 * tanque_1 \tag{1}$$

$$V1 = \pi \left(\frac{diametro_1}{2} \right)^2 * tanque_1$$

$$V1 = \pi \left(\frac{1550 \text{ mm}}{2} \right)^2 * 1505 \text{ mm}$$

$$1 \text{ Milímetros cúbicos} = 0.000001 \text{ Litros}$$

$$V1 = 2839819868 \text{ mm}^3 * \frac{0.000001 \text{ litros}}{1 \text{ mm}^3} = 2839.81 \text{ litros}$$

Para el tanque de agua filtrada:

$$V2 = \pi \left(\frac{\text{diametro}_2}{2} \right)^2 * \text{tanque}_2$$

$$V2 = \left[\pi \left(\frac{1550 \text{ mm}}{2} \right)^2 * 1485 \text{ mm} \right] * \frac{0.000001 \text{ litros}}{1 \text{ mm}^3}$$

$$V2 = 2802.08 \text{ litros}$$

Para hallar el porcentaje de líquido que contienen los tanques se trabajó con la ecuación 2.

$$\text{ht_tanque}_1 \quad 100\%$$

$$\text{tanque}_1 \quad x$$

$$x = \frac{\text{tanque}_1 * 100\%}{\text{ht_tanque}_1} \quad (2)$$

Para la programación del volumen del tanque de agua cruda se toma en cuenta a los dos tanques de agua A y B. Y en el caso del tanque de agua filtrada solo existirá un volumen. La programación de la tarjeta de control se observa en el Anexo H.

4.7.5 Desarrollo del HMI del sistema de control y alerta para el tanque purificador de agua.

La interfaz hombre-máquina se construyó en el lenguaje de programación Python, llamando una serie de librerías que permitieron la comunicación con el Arduino, creación de ventanas, desarrollo de gráficas, etc. Las principales librerías instaladas se presentan en la tabla 4.19.

Tabla 4. 19: Librerías empleadas en la programación de Python.

Módulos	Descripción	Aplicación en el sistema
Tkinter	Esta librería permite la creación de interfaz graficas de usuarios, viene por defecto con la instalación para Microsoft Windows.	Permitió la creación de ventanas, botones, etiquetas, inserción de imágenes, etc.
Matplotlib	Es un módulo de dibujo 2D y 3D, es una biblioteca para la generación de gráficos.	Permitió la creación de gráfico de nivel de líquido para el tanque uno y tanque 2
PySerial	Es una librería que permite comunicarse a través de la comunicación serial (rs-232)	Permitió la recepción de datos desde la tarjeta de control que contiene Arduino.

Elaborado por: Investigador.

Al instalar el programa ejecutable se presenta una primera ventana en la que el operario puede elegir entre iniciar el monitoreo del sistema o a su vez salir de él, al presionar el botón “Ingresar” se visualiza una segunda ventana en la cual se elige el puerto de comunicación con el que trabaja la computadora, si la elección de la comunicación es correcta se despliega un mensaje en la pantalla mostrando que la comunicación fue exitosa, seguidamente se presenta la pantalla principal el sistema de control y alarma del tanque purificador de agua de forma dinámica hacia el operario que se encarga del monitoreo y supervisión del sistema, en esta ventana se incluye electroválvulas, motores, filtros de arena-antracita y carbón activo, sistema de osmosis inversa, y el tablero eléctrico, estado de los sensores, niveles de los tanques de agua cruda y agua filtrada, visualización de gráficas en tiempo real, eventos y alarmas. Como se muestra en la figura 4.46. Todo el diseño fue orientado según la Guía GEDIS es la guía de interfaz de supervisión que se maneja con respecto al uso de color, información textual, distribución de pantalla, alarmas, etc. [62]. Ante las recomendaciones para el desarrollo de la interfaz Hombre-Máquina que emite la Guía Gedis en la tabla 4.20, se detallan las especificaciones usados en el desarrollo de software.

En el Anexo I se puede apreciar la programación realizada en el Software libre Python.

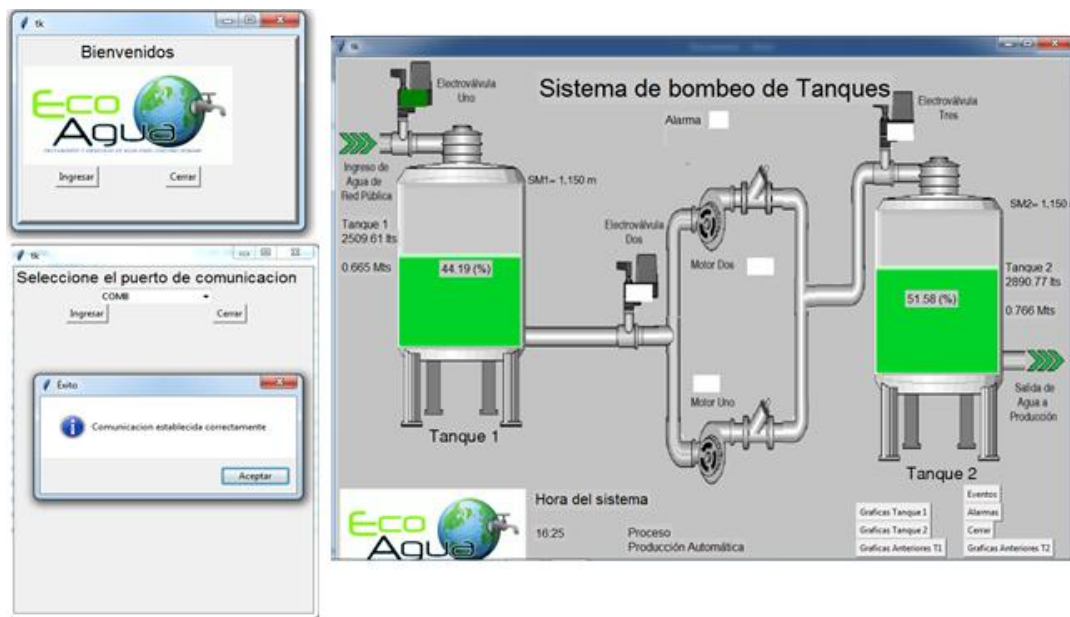


Figura 4. 46: Ventana de inicio, selección de puerto de comunicación y pantalla principal del sistema de control y alerta para el tanque purificador de agua en la Planta Purificadora ECOAGUA.

Elaborado por: Investigador.

Tabla 4. 20: Indicadores de la Guía GEDIS

Indicadores	Descripción	Indicadores	Descripción		
Distribución de pantallas	La pantalla principal cuenta con logo de la planta, hora, alarmas de procesos, botones de navegación.	Color para Alarmas			
		Alarma Crítica	Rojo		
		Color para material del proceso			
		Agua	Verde		
Colores para estatus de Equipos de Proceso		Colores de ítems adicionales			
Equipo parado	Blanco	Fondo de Pantalla	Gris		
Equipo trabajando	Verde	Fondo de Texto	Gris		
		Texto (títulos, subtítulos, texto normal)	Negro		
Tipo de Texto					
Título	Estilo: Sans Serif	Subtítulo	Estilo: Sans Serif	Texto Normal	Estilo: Sans Serif
	Tamaño: 26		Tamaño: 16		Tamaño: 14

Elaborado por: Investigador.

4.8 PRUEBAS DE FUNCIONAMIENTO

Luego de haber concluido con la instalación tanto de la parte física como lógica del sistema en la Planta Purificadora de Agua se procedió a realizar varias pruebas de funcionamiento las mismas que fueron evaluadas en varios procesos que realiza el sistema de control y alerta para el tanque purificador de Agua:

Visualización de información

El primer mensaje que se desplaza en la pantalla LCD es de inicio del sistema, el sistema contiene dos tipos de presentación de la información adquirida de los procesos que se llevan a cabo dentro del sistema de control y alerta para el tanque purificador de agua, la primera es los datos que se presentan en la pantalla de cristal líquida en los cuales se pueden diferenciar los datos de nivel de líquido en los tanques de agua cruda y agua filtrada, el estado activado/desactivado de las electroválvulas (EV1, EV2 y EV3), activación/desactivación de los motores (MT1 y MT2) y la leyenda OP (valor1 valor2) que equivale el “valor1” a la operación que se lleva a cabo dentro el sistema y el “valor2” un representa un error. Cuando surge alguna anomalía en el sistema en la parte de OP() se colocaran valores los mismos que se conocerán gracias a un mensaje presente en la pantalla. Como se muestra en la figura 4.47.

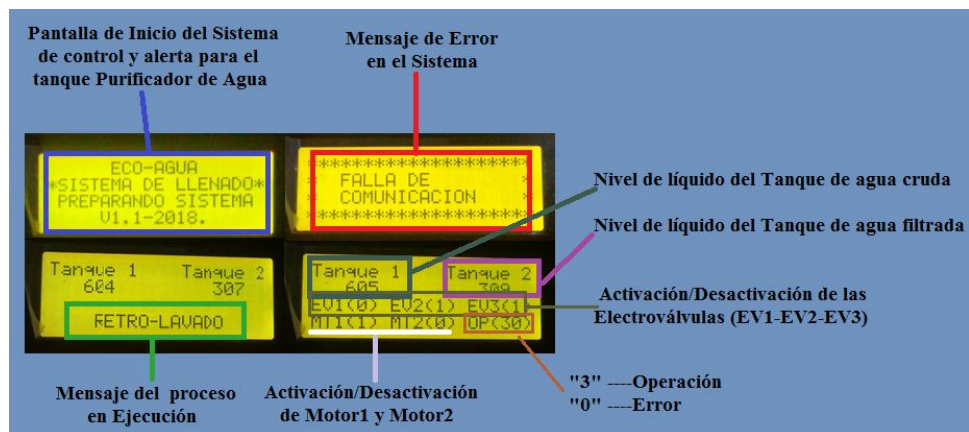


Figura 4. 47: Información visualizada en la pantalla LCD.

Elaborado por: Investigador.

La segunda pantalla que visualiza la información es el HMI del computador que monitorea el sistema de control y alerta para el tanque purificador de agua, en él se puede mostrar el estado de las electroválvulas (EV1, EV2 y EV3), los motores (MT1 y MT2), la hora actual, nivel de líquido del tanque de agua cruda y filtrada, adicional en la pantalla se encuentran botones como: graficas del tanque 1 y tanque 2, graficas anteriores del tanque 1 y tanque 2, eventos, alarmas y el botón cerrar. Como se muestra en la figura 4.48.

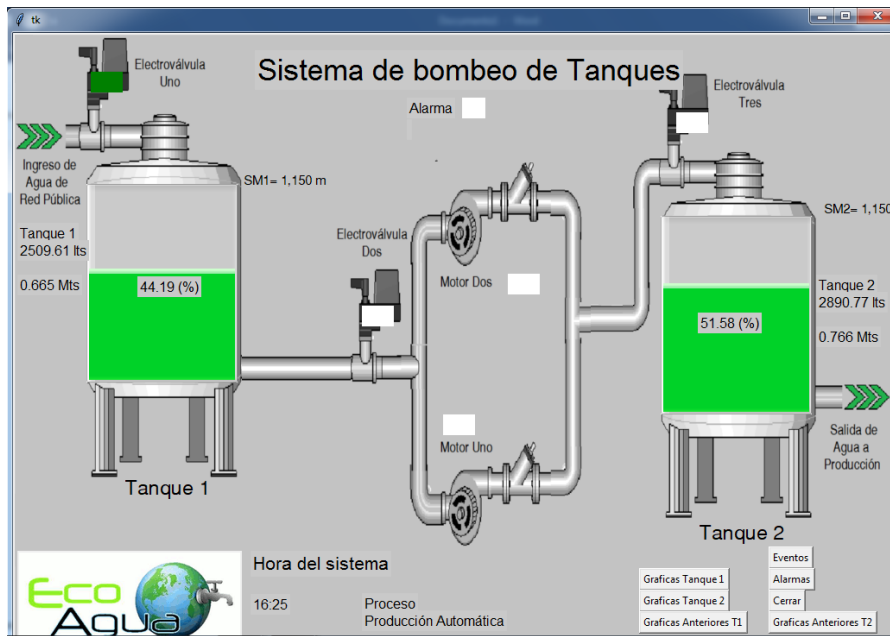


Figura 4. 48: Pantalla principal de la Interfaz Hombre-Máquina.

Elaborado por: Investigador.

Al elegir los botones de “Gráficas Tanque 1” o “Gráficas Tanque 2” estas muestran una gráfica de los niveles de líquido tanto del tanque de agua cruda como del tanque de agua filtrada en tiempo real, adicionalmente cuentan con unos botones extras que permiten detener la gráfica, guardar los datos de adquisición del nivel de agua los mismo que deben tener un nombre seguido de la extensión .txt, regresar y cerrar la ventana respectivamente. Como se muestra en la figura 4.49.

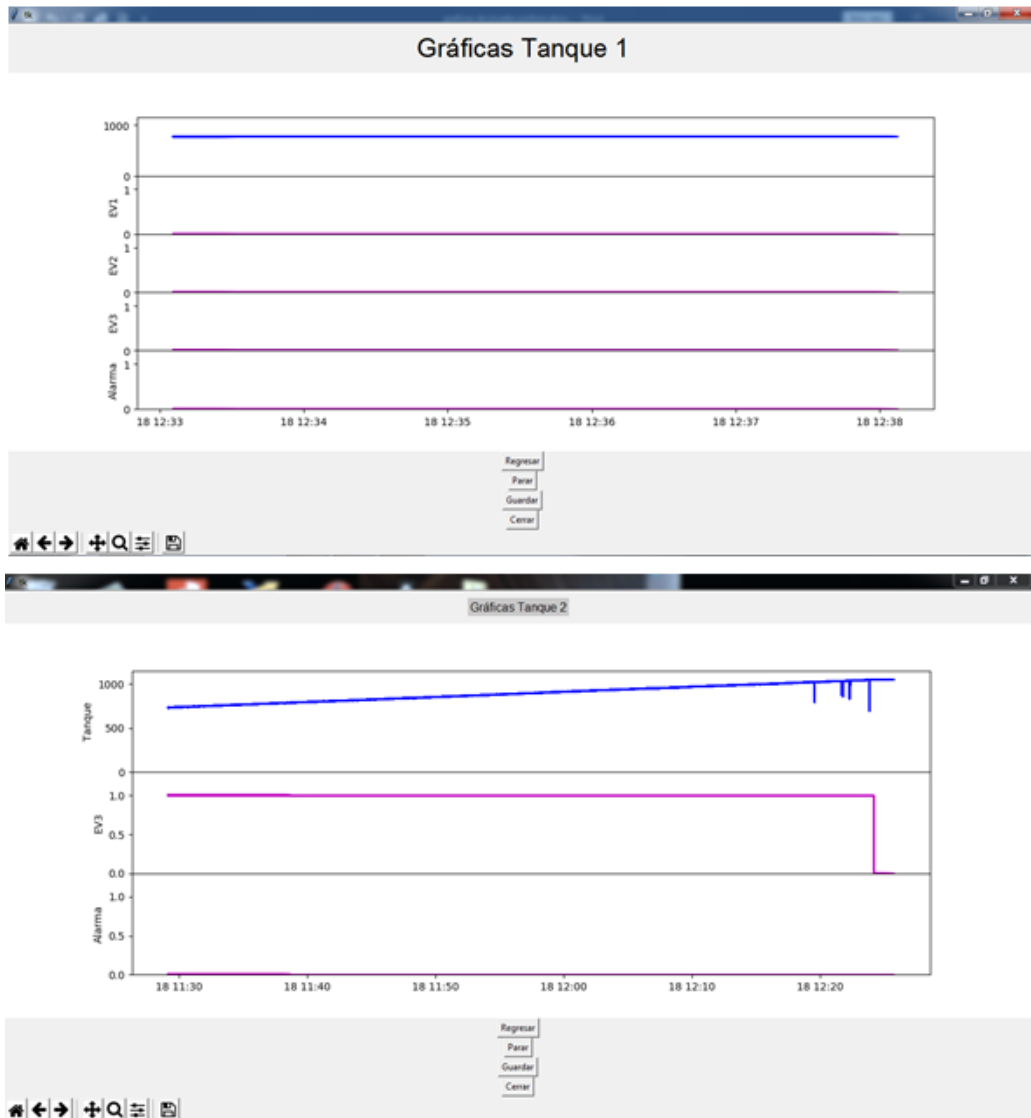


Figura 4. 49: Ventanas de los botones Gráfica Tanque 1 y Tanque 2.

Elaborado por: Investigador.

Por el contrario, si se presiona los botones “Gráficas Anteriores Tanque 1” o “Gráficas Anteriores Tanque 2” en estas ventanas se puede abrir las gráficas guardadas como archivos de extensión .txt, al abrir una gráfica se puede observar una barra de navegación ubicada en la parte inferior izquierda de la pantalla con las que puede regresar a la pantalla principal, ampliar la imagen, guardar la imagen con extensión .png, etc. Como se muestra en la figura 4.50.

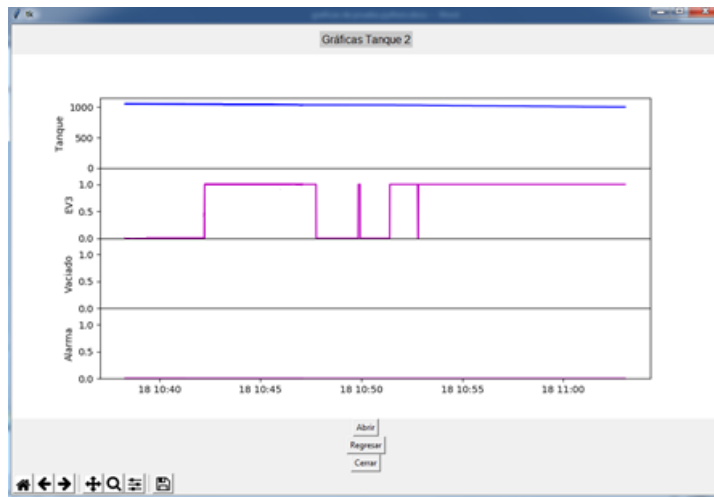


Figura 4. 50: Ventanas de gráfica anterior tanque 2.

Elaborado por: Investigador.

Al elegir el “botón eventos” se desplaza una lista de información referente a todos los procesos de operación sucedidos dentro del sistema como: inicio del programa, puerto serial abierto, vaciado de tanques, producción automática, tiempo de reposo, retrolavado y sistema detenido. Si se presiona el “botón alarmas” de la misma forma se despliega una lista de información con los errores sucedidos dentro del monitoreo del sistema de control, los datos de estos dos botones se guardan en un archivo denominado registros.log la misma que se crea dependiendo de la ubicación que el operario desee darle. En este documento se puede observar fecha, hora, y nombre de los eventos suscitados. Como se muestra en la figura 4.51, y finalmente el “botón cerrar” sale totalmente del sistema.

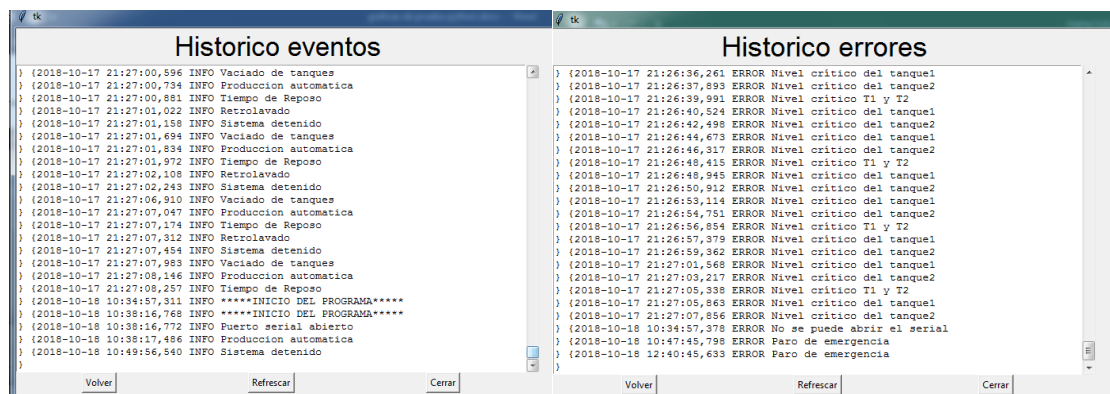


Figura 4. 51: Ventanas de los botones de Eventos y Alarmas

Elaborado por: Investigador.

Operaciones comunes dentro del Sistema de Control y Alerta para el tanque Purificador de agua.

▪ Vaciado de tanques (operación 0)

Para obtener este modo de operación se manipula en el tablero eléctrico el selector de modos de trabajo en este caso “modo vacío”, se puede visualizar en el HMI del computador el proceso que se ejecuta, adicional a esto en el tablero frontal se ve que componentes se encuentran activos o no mediante el comportamiento de los indicadores (luces piloto) y el HMI local.

La electroválvula (EV1) se desactiva, mientras las electroválvulas (EV2 y EV3) se encuentran activas hasta que los tanques de agua cruda y agua filtrada se vacíen por completo, la etiqueta que se muestra en el LCD OP(00) significa el primer “0” muestra la operación que es “vaciando tanques” y el segundo “0” identifica un error, pero en esta situación no representa ningún irregularidad en el sistema, este proceso se realiza para dar el respectivo mantenimiento a los tanques. Como se muestra en la figura 4.52.

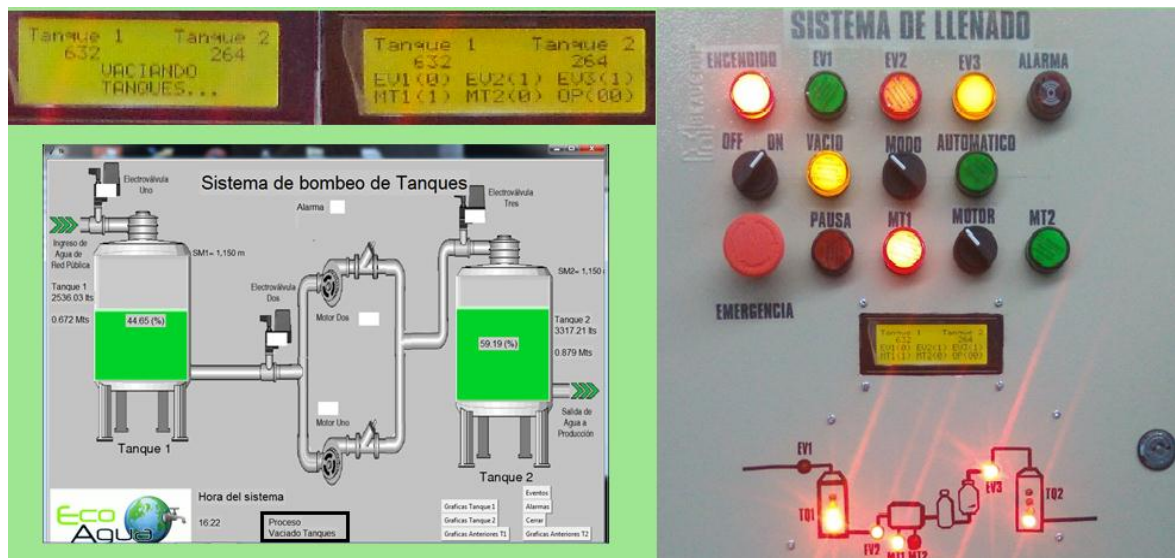


Figura 4. 52: Proceso de vaciado de tanques.

Elaborado por: Investigador.

- **Producción automática (operación 1)**

Este es el modo de operación principal de nuestro sistema de control y alerta del tanque purificador de agua, al elegir esta modo de operación el trabajador no se ve involucrado en ningún proceso ya que todo es controlado automáticamente por el sistema, específicamente en esta etapa las electroválvulas (EV2 y EV3) son activas para que se genere el paso de agua hacia el tanque de agua filtrada y por tanto se inicie la actividad de producción de la Planta, en el HMI local se puede observar el cambio de color (verde) del nivel de tanque de agua filtrada ya que está cerca de llegar a su nivel máximo de líquido del tanque. Como se muestra en la figura 4.53.



Figura 4. 53: Producción Automática.

Elaborado por: Investigador.

- **Tiempo de reposo (operación 2)**

El sensor ultrasónico 1 es el encargado de la detección del nivel de líquido equivalente a una altura de 1.05 m en el tanque de agua cruda, una vez que es detectada esa altura el sistema se detiene por dos horas aproximadamente mientras el agua reposa para que se asiente cualquier tipo de impurezas que se hayan filtrado del agua de la red pública hacia los tanques de agua cruda, mientras este proceso se lleva a cabo, se inhabilitan todas las electroválvulas y motores. Como se muestra en la figura 4.54.



Figura 4. 54: Tiempo de Reposo del tanque de agua Cruda.

Elaborado por: Investigador.

▪ **Retrolavado (operación 3)**

El retrolavado que realizan los filtros de arena-antracita y carbón activo, se efectúa dependiendo de la configuración realizada en dichos dispositivos, en este caso están programados para realizarse en la madrugada y los días lunes, miércoles, jueves y sábado, de forma independiente. Para su comprobación se puede observar en la figura 4.55, que las electroválvulas (EV1 y EV3) se encuentran desactivadas, la electroválvula (EV2) permanece activa para esta operación, el motor (MT1) se encuentra activo ya que en este período de evaluación él estaba operativo, la OP (30) es equivalente a: el número “3” es la operación de retrolavado y el número “0” es un error, pero en este caso no representa ninguna anomalía en el sistema. Es muy importante que para este proceso el tanque de agua cruda se encuentre con el nivel de líquido adecuado, caso contrario no se podrá efectuar este proceso.



Figura 4. 55: Proceso de retrolavado de los filtros de arena-antracita y carbón activo.

Elaborado por: Investigador.

- **Sistema detenido (operación 4)**

Esta operación entra en curso cuando el selector de modos de operación (3 posiciones) queda en el estado que corresponde a la posición “0” que es el sistema detenido, el mismo que como su nombre lo dice detiene el funcionamiento de todos los procesos que se encuentre habilitados como se muestra en la figura 4.56, el sistema se paraliza y envía a desactivarse las electroválvulas y motores como se indica en la pantalla LCD, adicional el texto que se señala en la LCD como OP(40) significa que el número “4” nos da a conocer “sistema detenido” y el número “0” representa un error, que usualmente no equivale a ninguna irregularidad dentro del sistema.



Figura 4. 56: Sistema Detenido.

Elaborado por: Investigador.

- **Sistema Detenido – Falla Sensor 1, OP(44)**

Esta prueba de funcionamiento del sistema, se efectúa cuando existe una anomalía en el sensor uno del tanque de agua cruda, el sistema envía activarse la alarma sonora que se encuentra ubicado en el tablero eléctrico, y conjuntamente se queda paralizado el proceso desactivando las electroválvulas y motores que en ese instante estuvieron trabajando hasta que el operario corrija el percance presentado en ese dispositivo, esta operación-falla corresponde la OP (44), como se indica en la figura 4.57.



Figura 4. 57: Sistema Detenido-Falla Sensor Uno.

Elaborado por: Investigador.

- **Sistema Detenido – Falla Sensor 2, OP(45)**

En la figura 4.58, se observa el fallo del sensor dos que corresponde al tanque de agua filtrada, este error es esquinante a la OP(45) como se observa en la imagen, de la misma manera que en la OP(44) el sistema envía habilitarse la alarma audible que se encuentra ubicado en el tablero eléctrico, y simultáneamente interrumpe el proceso inhabilitando las electroválvulas y motores, hasta que el operario intervenga y repare el dispositivo.



Figura 4. 58: Sistema Detenido-Falla Sensor Dos.

Elaborado por: Investigador.

- **Nivel crítico Tanque Uno y Tanque Dos, OP(13)**

Esta operación surge cuando los niveles de líquido de los tanques de agua cruda y filtrada llegan a cero, esto puede suceder por mantenimiento de los tanques ya que el operario debe vaciar los tanques por completo ó cuando la producción termina con toda la reserva de líquido de los contenedores, con base en esto el sistema sensa los dos tanques reportando un nivel cero de líquido en ellos y todos los actuadores se encuentran desactivados. Por lo tanto, el sistema envía activar la alarma sonora con un mensaje en la LCD de “nivel crítico tanque uno y dos” para que con esa orden se habilite la electroválvula uno quien inicia el llenado del tanque de agua cruda, como señala la figura 4.59.



Figura 4. 59: Nivel crítico tanque uno y dos

Elaborado por: Investigador.

- **Paro de Emergencia**

El paro de emergencia se realiza en los 3 modos de operación “vaciado de tanque” que es el OP(07), “producción automática” corresponde al OP(17) y en “sistema detenido” es equivalente al OP(47), esta acción sucede cuando el operario pulsa el paro de emergencia que se encuentra ubicado en el tablero eléctrico el mismo que inhabilita los motores y electroválvulas en cualquiera de los modos de trabajo en el que se encuentre el sistema de

control y alerta del tanque purificador de agua, adicional activa el alarma audible. Como se presenta en la figura 4.60.



Figura 4. 60: Paro de Emergencia.

Elaborado por: Investigador.

4.9 ANÁLISIS DE RESULTADOS

Anteriormente el llenado de agua en los tanques de agua cruda y agua filtrada en la Planta Purificadora de Agua ECOAGUA se realizaba de forma manual, dado así el operario que trabajaba tenía que estar pendiente de cuánta agua suministrar en los tanques y cerrar las válvulas de paso para evitar desbordamiento y desperdicio del líquido vital, adicional a eso el trabajador encargado para tener noción de la altura en la que se encontraba el agua debía subirse en una escalara para abrir el tanque y observa, o lo que es inadecuado debía golpear el depósito para saber cuánto de agua contenían los tanques. Por otra parte, la instalación de tuberías se encontraba en mal estado y deterioradas, como se muestra en la figura 4.51.



Figura 4. 61: Llenado de tanques de agua cruda y filtrada anteriormente.

Elaborado por: Investigador.

Con la implementación del sistema de control y alerta para el tanque purificador de agua en la planta purificadora ECOAGUA se cubrió esta necesidad, con la automatización de los procesos de llenado, el control mediante el tablero eléctrico y monitoreo por medio del HMI local y del computador en la cual se usó software libre, se obtuvo un proceso mejorado ya que con esta implementación el operario tuvo mayor tiempo para desempeñarse en otra actividad, pues con el movimiento de la perilla a “modo automático” el sistema se encarga del adecuado funcionamiento de todos los procesos de llenado. Como se observar en la figura 4.62. Las normativas usadas para esta implementación se basaron en IEC 61499 norma para sistemas industriales y la Guía GEDIS para el desarrollo de la interfaz hombre máquina.



Figura 4. 62: Implementación del sistema de control y alerta para el tanque purificador de agua en la planta purificadora ECOAGUA.

Elaborado por: Investigador.

Mediante este sistema de control automático se consigue condiciones adecuadas de trabajo en la parte de llenado de los tanques de agua cruda y filtrada, se encuentra orientado a todos los usuarios en especial al operario de turno que se encuentre en la supervisión del proceso.

El gabinete metálico es el encargado de brindar protección en este caso grado IP55, (protegido contra polvo-entrada limitada permitida, protegido contra chorro de agua a baja presión de todas las direcciones), ya que cuenta con una puerta metálica con un empaque de caucho que evita el ingreso de agentes externos que provoquen daño al sistema, para la entrada/salida de cables se colocó una presa estopa, adicional a esto las pistas de la tarjeta de control se encuentra protegidas por un barnizado para evitar corrosión de la misma, esta tarjeta de control se encuentra empotrada en el gabinete metálico de doble fondo con lo cual se evita que se presente vibraciones salvo el caso de desastres naturales (terremoto) o se incruste humedad desde la pared.

No se realizó un control PID, por dos sencillas razones: la primera el sistema trabaja con electroválvulas que no son proporcionales, esto quiere decir la electroválvula se habilita o

deshabilita de forma brusca, lo que provoca que el agua se detenga de forma abrupta produciendo el llamado golpe de arrite, el mismo que provoca daños a la tubería y afecta a la vida útil del dispositivo. La segunda razón es que la empresa es una planta que provee el líquido para el consumo humano, al contar con un control PID el agua debe mantenerse siempre en un punto de referencia (set point) esto conlleva a que los tanques no renueven el líquido manteniéndose agua estancada el cual es inadecuado para la producción. Por lo tanto, se eligió un control ON/OFF del sistema ya que de esta manera se cuida la integridad de los equipos ya que cumple con un tiempo de reposo y renovación del agua.

En las Figura 4.63 y 4.64, se muestra el comportamiento de las variables controladas tanto del tanque de agua cruda como del tanque de agua filtrada (EV1, EV2, EV3) y las alarmas que se activan cuando surge anomalías dentro del sistema estas se activan cuando la altura del tanque 1 está entre los límites < 250 mm y > 1050 mm para el tanque dos la alarma se activa < 200 mm y > 1050 mm. A demás, se manifiesta un control por histéresis por medio del control de los limites (si llega al límite máximo de altura del tanque 1050mm toma una acción y si llega al límite minino de altura del tanque 250mm determina otra acción en este caso el accionamiento de las alarmas). En el tanque 1 se muestra también el tiempo de reposo del líquido para que se asiente los sólidos presentes en el agua y esto sucede cuando la altura del líquido llegue a 1050mm.

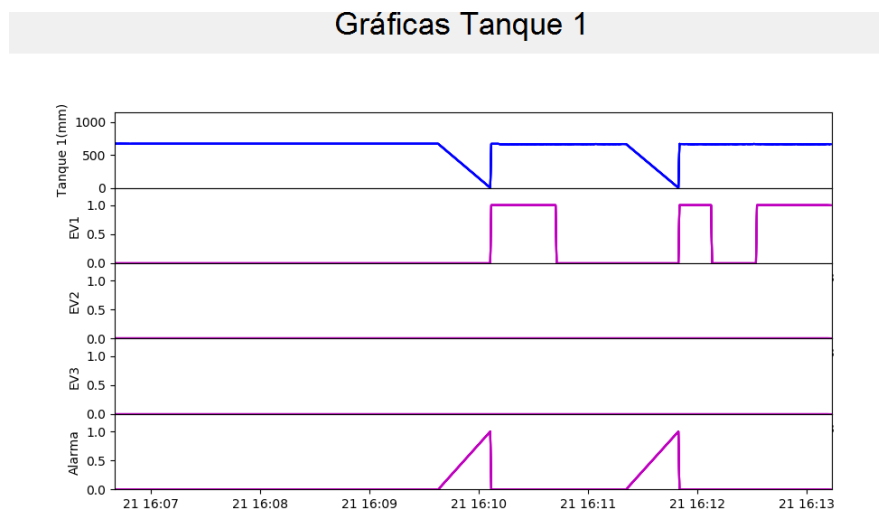


Figura 4. 63: Monitoreo del Nivel de líquido del tanque de agua cruda.

Elaborado por: Investigador.

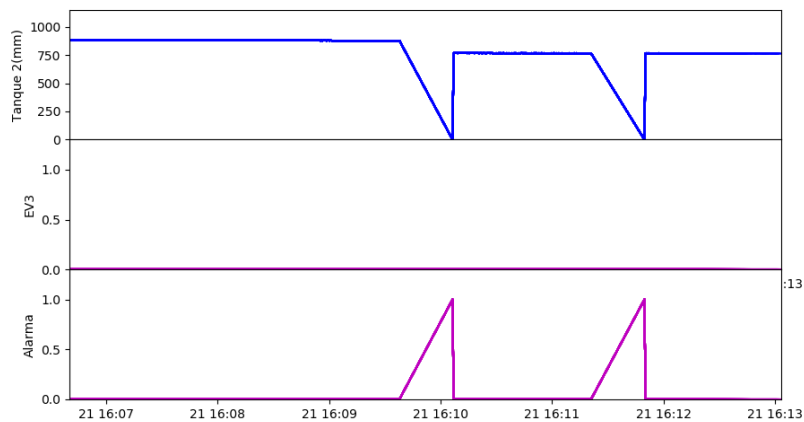


Figura 4. 64: Monitoreo del Nivel de líquido del tanque agua purificada.

Elaborado por: Investigador.

Comparación del llenado del tanque de agua cruda y vaciado del mismo

- Llenado del tanque 1: Siendo las 7:45am se tiene un nivel de tanque uno de 767mm. Luego de haber transcurrido aproximadamente 60 minutos (8:45 am), la nueva altura del tanque uno es 803mm. Por lo ende, con los datos de la tabla 4.21, se realiza una regla de tres simple:

$$\begin{array}{l} 60 \text{ min} \quad \nearrow \quad 36 \text{ mm} \\ 1 \text{ min} \quad \quad \quad x \end{array}$$

$x = \frac{1 \text{ min} * 36 \text{ mm}}{60 \text{ min}} = 0,6 \text{ mm}$ El tanque 1 se llena 0,6mm cada minuto ó 3mm cada 5 minutos.

Tabla 4. 21: Tiempo de llenado y vaciado del tanque 1.

Llenado tanque 1		Vaciado tanque 1	
Hora	Nivel tanque 1	Hora	Nivel tanque 1
7:45am	767mm	12:58am	971 mm
8:45am	803mm	13:34 pm	864 mm
Ha transcurrido 60min	36 mm	Ha transcurrido 36 min	107 mm

Elaborado por: Investigador

- Vaciado del tanque1: Siendo las 12:58am se tiene una altura de tanque uno de 971 mm, luego de haber transcurrido aproximadamente 36 minutos (13:34 pm), y tomando en consideración que la planta se encontraba produciendo la nueva altura del tanque es de 864mm. Por lo tanto, con los datos de la tabla 4.21, se realiza una regla de tres simple:

$$\begin{array}{ccc} 36 \text{ min} & \nearrow & 107 \text{ mm} \\ 1 \text{ min} & & x \end{array}$$

$$x = \frac{1 \text{ min} * 107 \text{ mm}}{36 \text{ min}} = 3 \text{ mm} \quad \text{El tanque 1 se vacía 3mm cada minuto.}$$

Siendo así se realiza una comparación en donde el valor de vaciado del tanque uno es de 3mm cada minuto transcurrido, esto se debe a que cuenta con un motor de impulsión y la tubería que transporta este líquido del tanque uno hacia los filtros es de ¾". Lo que no sucede en el llenado del tanque uno ya que el ingreso de agua de la red pública hacia el tanque se realiza mediante una tubería de ½" y a gravedad, un aproximado del llenado del tanque de agua cruda es de 3mm cada 5 minutos en condiciones de ingreso constante de agua.

Para interpretar el funcionamiento del sistema implementado se realizó 50 pruebas del llenado del tanque de agua cruda, en diferentes horario y días. Como se muestra en la tabla 4.22.

Tabla 4. 22: Tabla de Resultado Llenado del Tanque 1.

Ítem	Fecha (día/mes/año)	Hora (hora: minuto)	Altura de Tanque 1 (milímetros)
1	24/07/2018	17:39 (intervalo 15min)	2464
2	24/07/2018	17:54 (intervalo 15min)	2464
3	24/07/2018	18:09 (intervalo 15min)	2464
4	24/07/2018	18:24 (intervalo 15min)	2464
5	24/07/2018	18:39 (intervalo 15min)	440
6	24/07/2018	18:54 (intervalo 15min)	495
7	24/07/2018	19:09 (intervalo 15min)	440
8	24/07/2018	19:24 (intervalo 15min)	440
9	24/07/2018	19:39 (intervalo 15min)	440
10	24/07/2018	19:54 (intervalo 15min)	440

11	25/07/2018	13:00 (intervalo 30min)	328
12	25/07/2018	13:30 (intervalo 30min)	373
13	25/07/2018	14:00 (intervalo 30min)	418
14	25/07/2018	14:30 (intervalo 30min)	463
15	25/07/2018	15:00 (intervalo 30min)	508
16	25/07/2018	15:30 (intervalo 30min)	553
17	25/07/2018	16:00 (intervalo 30min)	598
18	25/07/2018	16:30 (intervalo 30min)	643
19	25/07/2018	17:00 (intervalo 30min)	688
20	25/07/2018	17:30 (intervalo 30min)	733
21	27/07/2018	13:45 (intervalo 30min)	821
22	27/07/2018	14:15 (intervalo 30min)	834
23	27/07/2018	14:45 (intervalo 30min)	847
24	27/07/2018	15:15 (intervalo 30min)	860
25	27/07/2018	15:45 (intervalo 30min)	873
26	27/07/2018	16:15 (intervalo 30min)	886
27	27/07/2018	16:45 (intervalo 30min)	899
28	27/07/2018	17:15 (intervalo 30min)	912
29	27/07/2018	17:45 (intervalo 30min)	925
30	27/07/2018	18:15 (intervalo 30min)	938
31	03/08/2018	19:00 (intervalo 30min)	641
32	03/08/2018	19:30 (intervalo 30min)	643
33	03/08/2018	20:00 (intervalo 30min)	645
34	03/08/2018	20:30 (intervalo 30min)	647
35	03/08/2018	21:00 (intervalo 30min)	649
36	03/08/2018	21:30 (intervalo 30min)	651
37	03/08/2018	22:00 (intervalo 30min)	653
38	03/08/2018	22:30 (intervalo 30min)	655
39	03/08/2018	23:00 (intervalo 30min)	657
40	03/08/2018	23:30 (intervalo 30min)	659
41	04/08/2018	06:45 (intervalo 60min)	731
42	04/08/2018	07:45 (intervalo 60min)	767
43	04/08/2018	08:45 (intervalo 60min)	803
44	04/08/2018	09:45 (intervalo 60min)	839
45	04/08/2018	10:45 (intervalo 60min)	875
46	04/08/2018	11:45 (intervalo 60min)	911
47	04/08/2018	12:45 (intervalo 60min)	947
48	04/08/2018	13:45 (intervalo 60min)	983
49	04/08/2018	14:45 (intervalo 60min)	1019
50	04/08/2018	15:45 (intervalo 60min)	1055

Elaborado por: Investigador.

Las pruebas realizadas con base en los datos adquiridos sobre el llenado del tanque de agua cruda demuestran que en los 10 primeros datos obtenidos existían mayor déficit de control de la altura del líquido, mientras que en los datos correspondientes de 41 hasta 50 el nivel se mantuvo permanente y se demuestra que el tanque se llena 3 milímetros cada 5 minutos transcurridos durante el día.

Para el análisis de fallas se obtuvo un total de 442 fallas de los componentes principales del sistema de control y alerta para el tanque purificador de agua. Eligiendo a dos dispositivos uno con mayor falla de funcionamiento y el otro con menor falla para este estudio. Como se presenta en la tabla 4.23.

Tabla 4. 23: Análisis de fallas de los principales componentes del sistema.

	NÚMERO DE PRUEBAS <i>[esta evaluación utiliza una escala de valorización de 10 puntos, siendo la ponderación de 0-9, mal funcionamiento (9) y buen funcionamiento (0)]</i>										$\sum x$	% FALLAS
	1	2	3	4	5	6	7	8	9	10		
FECHAS	24/07/2018	25/07/2018	26/07/2018	27/07/2018	30/07/2018	31/07/2018	01/08/2018	02/08/2018	03/08/2018	04/08/2018		
DISPOSITIVOS												
Electroválvula 1 (EV1)	6	6	6	5	5	5	4	2	0	0	39	8.8%
Electroválvula 2 (EV2)	6	6	5	5	4	5	4	2	0	0	37	8.4%
Electroválvula 3 (EV3)	6	6	5	4	5	4	4	2	0	0	36	8.1%
Motor 1 (MT1)	8	7	5	5	4	4	3	1	1	0	38	8.6%
Motor 2 (MT2)	8	8	7	6	4	4	3	2	1	0	43	9.7%
Sensor 1	8	8	8	7	6	4	3	2	1	0	47	10.6%
Sensor 2	8	8	7	5	5	3	3	1	2	0	42	9.5%
Tarjeta Electrónica	7	6	5	6	5	4	3	3	2	0	41	9.3%
Regulador de voltaje	9	8	6	6	6	6	6	2	0	0	49	11.1%
Pantalla LCD	6	5	5	5	4	4	2	1	0	0	32	7.2%
Comunicación	7	7	6	6	4	3	3	2	0	0	38	8.7%
$\sum y$	79	75	65	60	52	46	38	20	7	0		
$\sum y_{total}$	442				$\sum x_{total}$			442			TOTAL % FALLAS	100%

Elaborado por: Investigador.

Se halló el porcentaje de falla del “regulador de voltaje”:

$$442 \text{ fallas} \text{ --- } 100\%$$

$$49 \text{ fallas} \text{ --- } x1$$

Resolviendo esta regla de tres simple para hallar en valor de $x1$, para el cálculo se usa la ecuación 3, tenemos:

$$x1 * 442 \text{ fallas} = 49 \text{ fallas} * 100\%$$

$$x1 = \frac{49 \text{ fallas} * 100\%}{442 \text{ fallas}} = 11.1\% \quad (3)$$

Seguidamente se encontró el porcentaje de falla de la “pantalla LCD”:

$$442 \text{ fallas} \text{ --- } 100\%$$

$$32 \text{ fallas} \text{ --- } x2$$

Resolviendo esta regla de tres simple para hallar en valor de $x2$, tenemos:

$$x2 * 442 \text{ fallas} = 32 \text{ fallas} * 100\%$$

$$x2 = \frac{32 \text{ fallas} * 100\%}{442 \text{ fallas}} = 7.2\%$$

Se analizó el sistema de control y alerta para el tanque purificador de agua en la planta purificadora ECOAGUA, durante un periodo de 10 días, en los cuales se obtuvo al componente “regulador de voltaje” con mayor número de fallas siendo así su porcentaje del 11.1% con referencia al porcentaje principal, mientras que el dispositivo con menor porcentaje de error fue la “pantalla LCD” con un porcentaje del 7.2% en comparación al porcentaje referencial. Por lo tanto, este análisis permitió determinar la confiabilidad del sistema por cada uno de los componentes a partir de la información de los registros adquiridos de las fallas.

4.10 ANÁLISIS ECONÓMICO DEL SISTEMA IMPLEMENTADO

4.10.1 Costos de Hardware y Software.

A continuación, en la Tabla 4.24 y Tabla 4.25, se detalla el costo de los dispositivos y componentes electrónicos e industriales que fueron utilizados en el sistema de control y alerta para el tanque purificador de agua en el Planta Purificadora ECOAGUA. Adicional en la tabla 4.26 se presenta los costos totales de hardware y software.

Tabla 4. 24: Costos de hardware y software de dispositivos industriales.

CANTIDAD	DESCRIPCION	VALOR UNITARIO (\$)	VALOR TOTAL (\$)
30	Metros cable flexible#18	0,19	5,61
130	Terminales en u 18-20 aislado abierto azul	0,05	6,38
100	Terminales punteras amarillo 1.7mm	0,07	6,88
14	Tornillos dexion	0,02	0,31
16	Remaches 1/2"x1/8"	0,02	0,35
1	Breaker diferencial 30ma	11,04	11,04
1	Base riel din	0,00	0,00
1	Distribuidor diferencial legrand 125a	26,52	26,52
5	Porta fusibles riel din	1,67	8,33
5	Fusibles de 1a 20x5	0,20	0,98
1	Breaker monofásico riel din 32a	8,93	8,93
1	RSLM001024 +ZRLS1-2NA;rele slim 24v/6a +base	17,05	17,05
1	Cajetín dexion	1,93	1,93
1	Toma de corriente empotrable cooper	0,50	0,50
1	Tapa intemperie	3,85	3,85
2	Metros cable flexible#12	0,28	0,55
2	Borneras con aisladas 12 polos	3,34	6,68
2	Borneras con aisladas 6 polos	0,93	1,87
7	Prensa estopa pg9	0,73	5,08
20	Cable utp k6e	0,62	12,32
2	Sensores ultrasónicos blindados v/pwm/rs232 plc	215,04	430,08
2	Codos plastigama 3/4"	0,25	0,50
3	Teflones	2,00	6,00
2	Sellador ips	2,25	4,50
1	Fuente de poder estabilizada 4.2a 24v	40,90	40,90
1	Caja doble fondo metálica abatible 60x40	62,85	62,85
1	Luz piloto roja 120vca	1,67	1,67
2	Luz piloto roja 24vcd	1,62	3,24
2	Luz piloto verde 24vd	1,62	3,24

1	Selector dos posiciones con retención	1,72	1,72
1	Selector tres posiciones con retención	2,26	2,26
1	Zumbador buzzer rojo 120vca	2,89	2,89
1	Pulsador paro de emergencia rojo con retención	2,76	2,76
1	Luz verde 24vcd	1,67	1,67
1	Luz roja 24vcd	1,57	1,57
1	Luz amarilla 24vcd	1,67	1,67
2	Funda de amarras negras 20cm	2,95	5,89
1	Funda de canaleta espiral 1/2"	5,20	5,20
40	Metros cable sucre 2x16	0,68	27,28
12	Metros cable sucre 3x12	1,82	21,80
1	Breaker monofásico un polo para caja 22a	5,16	5,16
3	Relay de estado sólido 1p	25,30	75,90
2	Tapas db 9	0,30	0,59
1	Electroválvula bronce 3/4" 110vca	61,88	61,88
1	Electroválvula bronce 3/4" 110vca	61,88	61,88
1	Electroválvula bronce 1/2" 110vca	49,10	49,10
4	Universales lasco 3/4"	3,08	12,32
5	Tee 3/4" lasco	0,39	1,93
14	Terminales macho 3/4" lasco	0,50	6,93
5	Llaves esférica 3/4fv	7,26	36,30
5	Codo 3/4" lasco	0,29	1,43
1	2 metros de tubo plastigama 1/2"	1,38	1,38
1	2 metros de tubo plastigama 3/4"	1,38	1,38
2	Uniones lasco 3/4"	0,39	0,77
2	Bushin 3/4" x1/2 lasco	0,33	0,66
2	Universales lasco 1/2"	2,75	5,50
2	Tee 1/2" lasco	0,35	0,70
6	Terminales macho 1/2" lasco	0,22	1,32
2	Llave esferica 1/2" fv	5,06	10,12
2	Codo lasco 1/2"	0,25	0,51
1	Sellador ips 25cc	3,52	3,52
2	Teflones amarillos	0,44	0,88
2	Codo 3/4" incamep	0,77	1,54
		SUBTOTAL	\$ 1099,54
		IVA (12%)	\$ 131,94
		TOTAL	\$ 1231,49

Elaborado por: Investigador.

Tabla 4. 25: Costos de hardware y software de componentes Electrónicos.

CANTIDAD	UNIDAD	DESCRIPCION	VALOR UNITARIO (\$)	VALOR TOTAL (\$)
20	Unidad	Bornera doble para PCB	0,24	4,80
5	Unidad	Bornera triple para PCB	0,26	1,30
9	Unidad	Diodo 1n4007	0,04	0,36
4	Unidad	Condensador electrolítico 100Uf/50V (50<VCD<100)	0,08	0,32
2	Unidad	Condensador electrolítico 10Uf/50V (50<VCD<100)	0,08	0,16
5	Unidad	Condensador electrolítico 1Uf/50V (50<VCD<100)	0,08	0,40
10	Unidad	Condensador cerámico 104, (50<VCD<100)	0,08	0,80
1	Unidad	Condensador cerámico 103, (50<VCD<100)	0,08	0,08
1	Unidad	Condensador cerámico 102, (50<VCD<100)	0,08	0,08
1	Unidad	Condensador cerámico 22nf, (50<VCD<100)	0,08	0,08
1	Unidad	Condensador cerámico 220nf, (50<VCD<100)	0,08	0,08
1	Unidad	Condensador cerámico 22Pf, (50<VCD<100)	0,08	0,08
1	Unidad	Lm7809, regulador de voltaje	0,75	0,75
2	Unidad	Max232, circuito integrado	1,25	2,50
13	Unidad	Led bajo brillo (ROJO)	0,10	1,30
4	Unidad	Led bajo brillo (VERDE)	0,10	0,40
6	Unidad	Led bajo brillo (AMARILLO)	0,10	0,60
19	Unidad	Resistencia 220 ohmios, 1/4W	0,05	0,95
6	Unidad	Resistencia 10 kilo-ohmios, 1/4W	0,05	0,30
14	Unidad	Resistencia 330 ohmios, 1/4W	0,05	0,70
27	Unidad	Resistencia 1 kilo-ohmios, 1/4W	0,05	1,35
6	Unidad	Transistor TIP122	0,60	3,60
10	Unidad	Transistor 2n3904	0,10	1,00

3	Unidad	Regleta Conector sill (espadines macho 40 pines)	0,45	1,35
1	Unidad	Baquelita simple capa 200mmx300mm	3,75	3,75
1	Unidad	Baquelita simple capa 200mmx100mm	1,85	1,85
1	Unidad	LCD 4x20	14,35	14,35
2	Unidad	Zocalo 2x8 (16 pines)	0,20	0,40
3	Unidad	Papel termico (+-A4)	3,15	9,45
1	Unidad	PIC 18F2550	12,00	12,00
1	Unidad	ZOCALO 2X14 (28 pines)	0,25	0,25
1	Unidad	Cristal 10Mhz	0,60	0,60
1	Unidad	Trimmer (5K ó 10K)	0,85	0,85
1	Unidad	Arduino Mega	16,50	16,50
1	Unidad	Reloj realtime	5,15	5,15
1	Unidad	Integrado 7805	0,45	0,45
1	Unidad	Bus de datos	0,80	0,80
3	Unidad	Cloruro ferrico 90/35	0,80	2,40
8	Unidad	Condensador electrolitico 0.1Uf/50V (50<VCD<100)	0,08	0,64
2	Metros	Estaño	0,60	1,20
2	Unidad	Broca 1.2mm	0,80	1,60
1	Unidad	Impresión laser	0,30	0,30
11	Unidad	Tornillo milimétrico m4 y m6 armado	0,06	0,66
			SUBTOTAL	\$ 96,54
			IVA (12%)	\$ 11,58
			TOTAL	\$ 108,12

Elaborado por: Investigador.

Tabla 4. 26: Costos total de hardware y software.

DESCRIPCIÓN	VALOR
Costos de hardware y software de dispositivos industriales	\$ 1231.49
Costos de hardware y software de componentes Electrónicos	\$ 108,12
Costo total de hardware y software	\$ 1339,61

Elaborado por: Investigador.

4.10.2 Costo inicial del Diseño.

Para realizar la valoración del costo inicial del diseño e implementación se tomó en cuenta el total de horas trabajadas en el sistema, salario básico de un Ingeniero en Electrónica y Comunicaciones determinado por el Ministerio del Trabajo. [63]

Sueldo mensual: \$ 415

$$\text{Sueldo diario: } \frac{\$ 415}{21} = \$ 19.76$$

$$\text{Sueldo hora: } \frac{\$ 19.76}{8} = \$ 2.47$$

Tomando como tiempo estimado de trabajo total de 280 horas de investigación: distribuidas en la realización del diseño, programación, implementación y pruebas de funcionamiento del sistema de control y alerta. Para el cálculo del costo inicial del diseño se utiliza la ecuación 4.

$$\text{Costo inicial de diseño} = \text{total de horas trabajadas} * \text{sueldo hora} \quad (4)$$

$$\text{Costo inicial de diseño} = 280 * \$ 2.47$$

$$\text{Costo inicial de diseño} = \$ 691,6$$

4.10.3 Costo total del sistema Implementado

En la tabla 4.27 se presenta el costo total del sistema de control y alerta para el tanque purificador de agua.

Tabla 4. 27: Costos total de implementación del sistema.

DESCRIPCIÓN	VALOR
Costo total de hardware y software	\$ 1339,61
Costo inicial del diseño	\$ 691,6
Costo total de implementación	\$ 2031.21

Elaborado por: Investigador.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

La implementación del sistema de control y alerta automático permitió contar con un adecuado método de control para el llenado de los tanques, para ello se instaló un sistema de control basado en una tarjeta electrónica diseñada en Proteus y programada en el IDE de Arduino, la cual controla la activación de electroválvulas y motores conforme a la secuencia de control. Con esta implementación se evita eventos catastróficos como pérdida de líquidos por desborde, protegiendo además a los motores ya que se evita que trabajen cuando los tanques estén sin el nivel adecuado de líquido para producir (Altura adecuada de líquido para trabajar es $>250\text{mm} - < 1050$ en el tanque 1 y para el tanque 2 el líquido es adecuado cuando la altura se encuentra $>200\text{mm} - <1050\text{mm}$).

El sistema de purificación anteriormente se realizaba de forma manual y el operario debía supervisar que exista suficiente agua para producción. El sistema de control instalado se encarga de conservar los niveles de líquido adecuados realizando ciclos de renovación de agua bajo un sistema de control de histéresis (control de límites de llenado), para controlar el nivel de líquido se utiliza sensores ultrasónicos (MB7369 HRXL-MaxSonar) con comunicación RS-232 con este sistema la planta ECO-AGUA tiene un sistema totalmente automático en esta etapa de pre-producción.

La tarjeta de control fue diseñada para que el microcontrolador trabaje de forma segura, la misma permite visualizar la información del estado de eventos y monitoreo de las

variables de control, alertas de emergencias o alarmas y las presenta en las respectivas luces indicadores y en una pantalla LCD ubicado en el tablero de control en la planta de producción. La supervisión remota se realiza por medio de una interfaz HMI según la guía GEDIS ubicado en la oficina administrativa dentro de la planta, este HMI fue programada en software libre Python el cual presenta un entorno amigable hacia el usuario encargado de esta proceso, permitiendo la visualización del nivel de líquido presente en los tanques, sistema de alarmas y graficas de las variables controladas, además ofrece al usuario una representación básica del funcionamiento del sistema de control de llenado de tanques.

La tarjeta de control está conformada electrónicamente por un microcontrolador “Atmega 2560” la cual gestiona el control de actuadores eléctricos con la información procedentes de los sensores y de la selección del modo de operación por parte del usuario. Esta placa electrónica fue diseñada para soportar fallas eléctricas y reducir el ruido, además para la protección contra la humedad en la tarjeta se colocó un barnizado en las pistas para evitar corrosión de las mismas, adicionalmente ofrece un sistema de protección de la información con el uso del MAX232 al comunicarse con sus periféricos a distancia como son el HMI, la tarjeta de visualización y las comunicaciones con los sensores. Todo esto se encuentra dentro de un gabinete metálico que cuenta con grado de protección IP55.

En la actualidad el uso de software y hardware libre han incrementado las formas de solucionar problemas en todos los ámbitos, en este caso el uso de Arduino y Python solucionaron el problema de la planta ECOAGUA en el proceso de control de llenado del tanque purificador de agua. Reduciendo costos en la implementación ya que el software de programación no requirió de compra de licencia para su uso.

5.2 RECOMENDACIONES

Al momento de hacer el lavado de tanques de forma manual por parte de los operarios se debe tener cuidado con el sensor ultrasónico, en el caso de que se mueva o lo retiren por causas adversas deben colocarlo de la misma forma en la que estaba dispuesta su ubicación, es siempre perpendicular al nivel de agua.

El sistema se controla automáticamente, pero en el caso de requerir usar el sistema manual deben apagar o colocar la perilla de selección de la máquina en “Sistema detenido” y manipular manualmente las válvulas especificadas por el diseñador del sistema de tuberías, y una vez terminado ese proceso retomar nuevamente al sistema automático se debe cerrar las válvulas especificadas para que el sistema pueda operar con el máximo rendimiento.

Se recomienda no cambiar la posición de los sensores ultrasónicos debido a que están configurados a una determinada distancia y se usa esta medida para determinar la altura de líquido, con lo que si se mueve o lo reubican se puede tener medidas no precisas del nivel actual de líquido presente en los tanques.

Antes de operar el sistema de control y alerta el operador o el usuario debe leer el manual de operación, evitar que personal ajeno abra imprudentemente el tablero de control, mantener en buen estado los cables de comunicación, no rasgar o doblar los cables eléctricos, no forzar el sistema de llenado de forma extrema ya que se pierde control sobre el sistema.

BIBLIOGRAFÍA

[1] El Telégrafo. “En Ecuador utilizamos el 30% del agua y se desperdicia el 70%.” [En línea]. Disponible en: <http://www.eltelegrafo.com.ec/noticias/politica/2/en-ecuador-utilizamos-el-30-del-agua-y-se-desperdicia-el-70>.

[2] “Estrategia Nacional de Agua Potable y Saneamiento.” [En línea]. Disponible en: <http://www.agua.gob.ec/wp-content/uploads/downloads/2016/07/REVISTA-SENAGUA.compressed.pdf>.

[3] El Telégrafo. “El desperdicio de agua cruda será cobrado a los municipios.” [En línea]. Disponible en: <http://www.eltelegrafo.com.ec/noticias/politica/2/el-desperdicio-de-agua-cruda-sera-cobrado-a-los-municipios>.

[4] L. D. Almagor Farinango y D. A. Chalan Chagna. “Propuesta estratégica y estudio de prefactibilidad para la creación de una empresa dedicada a la producción y venta de botellones de agua purificada ubicada en el barrio el condado del Cantón Quito” Tesis de Grado, Universidad Central del Ecuador, Quito, 2013. [En línea]. Disponible en: <http://www.dspace.uce.edu.ec/bitstream/25000/2466/1/T-UCE-0005-396.pdf>.

[5] R. A. Astudillo Bastidas. “Diseño e Implementación de un prototipo de medidor de Nivel de agua a través de un sensor ultrasónico para pasos Deprimidos,” Tesis de Grado, Universidad Tecnológica Israel, Quito, 2016. [En línea]. Disponible en: <http://repositorio.uisrael.edu.ec/bitstream/47000/1181/1/UISRAEL-EC-ELDT-378.242-19.pdf>

[6] M. Á. Méndez Robles y P. J. Erazo Silva. “Implementación de un banco de pruebas para el control de nivel de líquidos en el laboratorio de instrumentación industrial de la Facultad de Mecánica” Tesis de Grado, Escuela Superior Politécnica de Chimborazo, Riobamba, 2014. [En línea]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/3338/1/25T00225.pdf>.

[7] C. Peñaranda, W. Silva y E. Gómez. “Instrumentación y Control de nivel para un sistema de Tanques acoplados en el Laboratorio de Control e Instrumentación de la E3T-

UIS” Tesis de Grado, Universidad Industrial de Santander, Bucaramanga, 2014. [En línea]. Disponible en: <http://tangara.uis.edu.co/biblioweb/tesis/2014/151339.pdf>.

[8] J. Vergara, R. Molina y C. Machado. “Prototipo para control automático de nivel y caudal de líquidos.” *INGENIUS*, vol. 6, pp. 9-18, Julio/diciembre 2011. [En línea]. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/5972815.pdf>.

[9] L. A. Restrepo Álvarez y J. D. J. Cardona Salazar. “Diseño de un sistema de control de nivel para la preparación de líquidos dializantes basado en señales ultrasónicas” Tesis de Grado, Universidad Tecnológica de Pereira, Pereira, Febrero 2015. [En línea]. Disponible en: <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/5111/6298R436.pdf;sequence=1>.

[10] Á. Carbajal Azcona y M. González Fernández. “Propiedades y funciones biológicas del agua,” 2012. [En línea]. Disponible en: <https://www.ucm.es/data/cont/docs/458-2013-07-24-Carbajal-Gonzalez-2012-ISBN-978-84-00-09572-7.pdf>.

[11] Eurotherm. “El proceso de purificación de agua.” [En línea]. Disponible en: <http://www.eurotherm.es/water-purification>.

[12] P. Ponsa y A. Granollers. “Diseño y Automatización Industrial.” [En línea]. Disponible en: <https://www.epsevg.upc.edu/hcd/material/lecturas/interfaz.pdf>.

[13] J. Acedo Sánchez, “Tipos de Control de Nivel,” de *Instrumentación y Control básico de procesos*, Madrid, Ediciones Díaz de Santos, S.A, 2013, pp. 443, 444.

[14] Instituto Nacional de Ciberseguridad de España S.A., “La evolución de los dispositivos en los sistemas de control industrial,” 09 septiembre 2015. [En línea]. Disponible en: <https://www.certsi.es/blog/evolucion-dispositivos-sistemas-control-industrial>.

[15] “Sensor,” [En línea]. Disponible en: <http://www.pce-iberica.es/instrumentos-de-medida/sistemas/sensores.htm>.

- [16] E. Vildósola, “Actuadores,” [En línea]. Disponible en: <http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>.
- [17] ASCONUMATICS, “Tecnología Electroválvulas y Válvulas,” [En línea]. Disponible en: https://www.asconumatics.eu/images/site/upload/_es/pdf1/00005es.pdf.
- [18] ABB Asea Brown Boveri, S.A, “Elemento de mando y señalización,” [En línea]. Disponible en: https://library.e.abb.com/public/26352578ada4d1d4c125793c003453da/Elementos%20mando%20y%20senalizacion%2022mm_1TXA151023C0701-0310.pdf.
- [19] E. Pérez. “Los sistemas SCADA en la automatización industrial,” Máster en Sistemas Modernos de Manufactura y Bach. Universidad de Costa Rica. Recinto Grecia-Costa Rica, 2015. [En línea]. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/5280242.pdf>
- [20] J. Colomer, J. Meléndez y J. Ayza. “Introducción a la monitorización y supervisión experta de Procesos” [En línea]. Disponible en: <http://intranet.ceautomatica.es/sites/default/files/upload/10/files/sistemas%20de%20supervision.pdf>
- [21] J. C. Roa, “Interfaz Hombre Maquina.” *Sociedad Nuclear Española SNE*, p. 7, 2015. [En línea]. Disponible en: <https://www.sne.es/images/stories/recursos/publicaciones/notas/NT%200115%20-%20Interfaz%20hombre%20m%C3%A1quina.pdf>
- [22] R. Cobo, “El ABC de la Automatización, HMI,” [En línea]. Disponible en: <http://www.aie.cl/files/file/comites/ca/abc/hmi.pdf>
- [23] A. J. Medrano Villalobos y L. M. Chourio Acevedo (2017, abril) “Hardware libre y abierto, modelos de Negocio para América Latina y El Caribe.” *LATIN AMERICAN JOURNAL OF BUSINESS MANAGEMENT*. Vol. 8, nº 1, pp. 61-83. [En línea]. Disponible en: <http://www.lajbm.net/index.php/journal/article/viewFile/407/191>
- [24] A. Lazalde, J. Torres y D. Vila, “Hardware Libre,” 2015. [En línea]. Disponible en: <http://flokssociety.org/docs/Espanol/4/4.1.pdf.g>

- [25] N. Goilav y G. Loi, “Arduino Mega,” ARDUINO Aprender a desarrollar para crear objetos inteligentes, Barcelona, Editions ENI, 2016, pp. 222-224.
- [26] Arduino, “Arduino Mega 2560 REV3,” [En línea]. Disponible en: <https://store.arduino.cc/usa/arduino-mega-2560-rev3>
- [27] G. Fernández. “Sensores Magnéticos e Inductivos” Tesis de Grado, Universidad Autónoma del Estado de Hidalgo, Pachuca, 2005. [En línea]. Disponible en: <https://www.uaeh.edu.mx/docencia/Tesis/icbi/licenciatura/documentos/Sensores%20magneticos.pdf>
- [28] “Sensores” [En línea]. Disponible en: <http://www.isa.cie.uva.es/~maria/sensores.pdf>. [Último acceso: 03 Marzo 2018].
- [29] “Sensores y transductores” [En línea]. Disponible en: http://www.eudim.uta.cl/files/5813/2069/8949/fm_Ch03_mfuentesm.pdf [Último acceso: 03 Marzo 2018].
- [30] F. Torres y C. Fernández. “Sensores y Detectores” [En línea]. Disponible en: https://rua.ua.es/dspace/bitstream/10045/18433/1/Tema%202_Sensores%20y%20Detectores.pdf. [Último acceso: 04 Marzo 2018].
- [31] A. Gil, “Sensores y Actuadores” [En línea]. Disponible en: <http://isa.umh.es/asignaturas/asc/temasautomatas/Tema6.pdf>. [Último acceso: 04 Marzo 2018].
- [32] MaxBotix. “HRXL-MaxSonar-WR Series” 2012. [En línea]. Disponible en: https://www.maxbotix.com/documents/HRXL-MaxSonar-WR_Datasheet.pdf. [Último acceso: 04 Marzo 2018].
- [33] R. Diosdado. “Sensor de Ultrasonidos HC-SR04” [En línea]. Disponible en: <https://www.zonamaker.com/arduino/modulos-sensores-y-shields/ultrasonido-hc-sr04>. [Último acceso: 01 Abril 2018].

- [34] SuperRobotica. “SRF04 sensor distancias por ultrasonidos SRF04 S320110” [En línea]. Disponible en: <http://www.superrobotica.com/s320110.htm>. [Último acceso: 01 Abril 2018].
- [35] Telemecanique Sensor. “xx930a3a2m12 DATASHEET” [En línea]. Disponible en: <https://docs-emea.rs-online.com/webdocs/129c/0900766b8129cba2.pdf>. [Último acceso: 01 Abril 2018].
- [36] BANG GOOD. “Electroválvula 2w-200-20 AC110V” [En línea]. Disponible en: https://www.banggood.com/es/2W-200-20-AC110V-34-Brass-Electric-Solenoid-Valve-Water-Air-Fuel-p-1058503.html?cur_warehouse=CN. [Último acceso: 07 Abril 2018].
- [37] Tecofi.fr. “Electroválvula NF – latón – hembra BSP” [En línea]. Disponible en: https://www.tecofi.fr/MEDIAS/passport/EV1140_EN.pdf. [Último acceso: 07 Abril 2018].
- [38] Direct Industry. “Electroválvula de control asistido de 2/2 vías NC de acero inoxidable”. [En línea]. Disponible en: http://www.directindustry.es/prod/danfoss-industrial-automation/product-40815-1945169.html#product-item_1945346. [Último acceso: 07 Abril 2018].
- [39] CEA Group International Co., Ltd. “ZG3nc Relé de estado sólido”. [En línea]. Disponible en: https://es.made-in-china.com/co_ceaelectric/product_SSR-Zg33-Zg3nc-Solid-State-Relay_hosyussog.html. [Último acceso: 09 Abril 2018].
- [40] WEG. “Control Parada Emergencia CPA-D” [En línea]. Disponible en: <https://www.weg.net/catalog/weg/BR/es/Seguridad-de-M%C3%A1quinas-y-Sensores-Industriales/Rel%C3%A9s-de-Seguridad/CONTROL-PARADA-EMERGENCIA-CPA-D/p/12381149>. [Último acceso: 09 Abril 2018].
- [41] Makerlab Electronics. “Relay SRD-12VCD-SL-C, Datasheet” [En línea]. Disponible en: https://www.makerlab-electronics.com/my_uploads/2017/11/relay-datasheet.pdf. [Último acceso: 09 Abril 2018].

- [42] HWgroup. “IP Relay HWg-ER02b” [En línea]. Disponible en: https://www.hw-group.com/products/ip_relay/index_es.html. [Último acceso: 09 Abril 2018].
- [43] Industrial Shield. “M-DUINO PLC Arduino Ethernet 21 I/Os Analog/Digital PLUS”. [En línea]. Disponible en: https://www.industrialshields.com/es_ES/shop/product/is-mduino-21-m-duino-plc-arduino-ethernet-21-i-os-analog-digital-plus-3?category=1 [Último acceso: 29 Septiembre 2018].
- [44] E-ELEKTRONIC. “¿Qué tarjeta es mejor? Arduino, Raspberry Pi o BeagleBone.” 28 Agosto 2013. [En línea]. Disponible en: <https://e-elektronic.com/tarjeta-arduino-vs-raspperri-pi-vs-beaglebone/>. [Último acceso: 15 Abril 2018].
- [45] “Espressif Smart Connectivity Platform: ESP8266.” 12 Octubre 2013. [En línea]. Disponible en: https://nurdspace.nl/images/e/e0/ESP8266_Specifications_English.pdf. [Último acceso: 15 Abril 2018].
- [46] “Arduino Mega 2560 DATASHEET” [En línea]. Disponible en: https://www.tiendaarduino.com/datasheet/arduino-mega_2560.pdf [Último acceso: 15 Abril 2018].
- [47] N. G. Forero Saboya. “Normas de Comunicación en Serie: RS-232, RS-422 y RS-485.” *INGENIO Libre*, vol. 1, pp. 1-4, Mayo/Junio 2012. [En línea]. Disponible en: <http://www.unilibre.edu.co/revistaingeniolibre/revista-11/art13.pdf>
- [48] J. Sirgo y R. Gonzáles. “Norma RS-232-C.” [En línea]. Disponible en: <http://www.isa.uniovi.es/docencia/redes/rs-232-c.pdf>. [Último acceso: 12 Septiembre 2018].
- [49] A. López. “Puertos USB - Bus Serie Universal y descripción de la norma IEEE 1394.” [En línea]. Disponible en: http://platea.pntic.mec.es/~alopez1/web_Eldad/_gs_srca/CI/Puertos%20USB.pdf. [Último acceso: 12 Septiembre 2018].

[50] Greek Factory. “Cable Convertidor USB a Serial RS232.” [En línea]. Disponible en: <https://www.geekfactory.mx/tienda/cables-y-conectores/cable-convertidor-usb-a-serial-rs232/>. [Último acceso: 12 Septiembre 2018].

[51] ELECTRONICS CO., LTD. “LED DIGIT DISPLAY DATASHEET” [En línea]. Disponible en: https://www.electronicoscaldas.com/datasheet/5611AX-5611BX-Series_NINGBO-FORYARD-OPTO.pdf [Último acceso: 14 Septiembre 2018].

[52] Datasheet Search Site. “HD44780U (LCD-II)” [En línea]. Disponible en: <https://pdf1.alldatasheet.com/datasheet-pdf/view/63673/HITACHI/HD44780.html>. [Último acceso: 14 de Septiembre 2018].

[53] Logicbus. “TPD-433F” [En línea]. Disponible en: <http://www.logicbus.com.mx/pdf/02/TPD-43xF.pdf> [Último acceso: 14 Septiembre 2018].

[54] C. Paspuezán y M. Gavilánez. “Monitoreo del nivel de agua en los tanques de la subestación azaya de la empresa EMAPA-I a través de una red inalámbrica basada en IEEE802.11a” Tesis de Grado, Universidad Técnica del Norte, Ibarra - Ecuador, 2013. [En línea]. Disponible en: <http://repositorio.utn.edu.ec/bitstream/123456789/1950/1/04%20RED%20026%20ARC%20HIVO%20T%C3%89CNICO%20EMAPA.pdf>

[55] G. V. Rossum. “Tutorial de Python.” Septiembre 2009. [En línea]. Disponible en: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>. [Último acceso: 14 Septiembre 2018].

[56] “IDLE, el entorno de desarrollo de Python.” [En línea]. Disponible en: <http://www.mclibre.org/consultar/python/otros/python-idle.html>. [Último acceso: 14 Septiembre 2018].

[57] Python Software Foundation (US). “IDLE.” [En línea]. Disponible en: <https://docs.python.org/3/library/idle.html>. [Último acceso: 14 Septiembre 2018].

[58] M. V. García Sánchez. “Metodologías para el Diseño de Sistemas de Control Distribuido bajo el estándar IEC 61499 aplicados al control de procesos” Tesis Doctoral, Departamento de Ingeniería de Sistemas y Automática, Universidad del País de Vasco, España, 2018. [En línea]. Disponible en: https://addi.ehu.es/bitstream/handle/10810/26649/TESIS_GARCIA_SANCHEZ_MARC_ELO%20VLADIMIR.pdf?sequence=1

[59] J. López. “Normas ANSI / ISA Para Instrumentación” [En línea]. Disponible en: <ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/Proyecto%20de%20Automatizacion/ANSI%20ISA%20-%20Espa%F1ol%20Incompleto.pdf> [Último acceso: 18 Septiembre 2018].

[60] BRADY. “Etiquetado de cables y alambre Estándares y aplicaciones” [En línea]. Disponible en: <http://www.bradylatinamerica.com/es-mx/normatividad/etiquetado-de-cables-y-alambre/cables-y-alambre> [Último acceso: 18 Septiembre 2018].

[61] “Normas de etiquetado de cables ANSI TIA 606-B.” [En línea]. Disponible en: <https://www.graphicproducts.com/articles/ansi-tia-606-b-cable-labeling-standards/> [Último acceso: 29 Septiembre 2018].

[62] P. Ponsa, M. Díaz y A. Catalá. “Creación de guía ergonómica para el diseño de interfaz” [En línea]. Disponible en: <https://aipo.es/articulos/4/4.pdf>. [Último acceso: 18 Septiembre 2018].

[63] Ecuador Legal Online. “Salarios Mínimos Sectoriales 2018” 15 Marzo 2018. [En línea]. Disponible en: <http://www.ecuadorlegalonline.com/laboral/tabla-salarios-minimos-sectoriales-2018/>. [Último acceso: 25 Julio 2018].

Datasheet MB7369 HRXL-MaxSONAR-WRM.

HRXL-MaxSonar® - WR/WRC™ Series

HRXL-MaxSonar® - WR™ Series

High Resolution, Precision, IP67 Weather Resistant, Ultrasonic Range Finder
 MB7360, MB7363, MB7366, MB7367, MB7369, MB7380, MB7383, MB7386, MB7387, MB7389



The HRXL-MaxSonar-WR sensor line is a cost-effective solution for applications requiring precision range-finding, low-voltage operation, space saving, low-cost, and IP67 rating for weather resistance.

The HRXL-MaxSonar-WR sensor line provides high accuracy and high resolution ultrasonic proximity detection and ranging in air. This sensor line features 1-mm resolution, target-size and operating-voltage compensation for improved accuracy, superior rejection of outside noise sources, internal speed-of-sound temperature compensation and optional external speed-of-sound temperature compensation. The HRXL-MaxSonar-WR/WRC models are available in 5-meter or 10-meter models. This ultrasonic sensor detects objects from 1-mm and ranges to objects from 30-cm* to maximum range. Objects closer than 30-cm* are typically reported as 30-cm*. The interface output formats are pulse width, analog voltage, and digital serial in either RS232 (MB7360 series) or TTL (MB7380 series). Factory calibration is standard. *For select sensors this distance is 50-cm, refer to pages 4 and 5.

<p>Precision Ultrasonic Range Sensing</p> <ul style="list-style-type: none"> • Range-finding at a fraction of the cost of other precision rangefinders • Reading-to-reading stability of 1-mm at 1-meter is typical¹ • Accuracy is factory-matched providing a typical accuracy of 1% or better^{1,2} • Internal temperature compensation is standard and optional external temperature compensation • Determines range to largest object (MB7369, MB7389) • Determines range to first detectable object (MB7360, MB7363, MB7366, MB7367, MB7380, MB7383, MB7386, MB7387) • Excellent clutter rejection • Additional chemical resistance available⁶ <p>Very Low Power Requirements</p> <ul style="list-style-type: none"> • Fast first reading after power-up eases battery requirements • Wide, low supply voltage of 2.7V to 	<p>5.5V requirements eases battery powered design</p> <ul style="list-style-type: none"> • Low current draw reduces current drain for battery operation • Very low-power rangefinder, excellent for multiple sensor or battery based systems <p>Easy to Use Component Module</p> <ul style="list-style-type: none"> • Gracefully handles other ultrasonic sensors⁵ • Stable and reliable range readings and excellent noise rejection make the sensor easy to use for most users • Easy to use interface with distance provided in a variety of outputs • Target size compensation provides greater consistency and accuracy • Sensor automatically handles acoustic noise^{2,4} • Calibrated sensor eliminates most sensor to sensor variations 	<p>Applications & Uses</p> <ul style="list-style-type: none"> • Tank level measurement (MB7369 or MB7389) • Weather station monitoring • Bin level measurement • Proximity zone detection • People detection • Robot ranging sensor • Long range object detection • Environments with acoustic and electrical noise • Height monitors • Auto sizing • Box dimensions • Automated factory systems <p>Notes:</p> <p>¹ Refer to section that compares WR to alternative housings on page 4</p> <p>² Users are encouraged to evaluate the sensor performance in their application</p> <p>³ Reference pages 8-9 for part specific timing information by design</p> <p>⁴ See page 6 for multi-sensor operation</p> <p>⁵ F-Option provides additional protection from hazardous chemical environments</p> <p>⁶ Please reference page 10 & 11 for minimum operating voltage verses temperature information</p> <p>⁸ Please reference page 18 for part number key</p>
--	--	---

Close Range Operation

Applications requiring 100% reading-to-reading reliability should not use MaxSonar sensors at a distance closer than 50cm. Although most users find MaxSonar sensors to work reliably from 0 to 50cm for detecting objects in many applications, MaxBotix® Inc. does not guarantee operational reliability for objects closer than the minimum reported distance. Because of ultrasonic physics, these sensors are unable to achieve 100% reliability at close distances.

Warning: Personal Safety Applications

We do not recommend or endorse this product be used as a component in any personal safety applications. This product is not designed, intended or authorized for such use. These sensors and controls do not include the self-checking redundant circuitry needed for such use. Such unauthorized use may create a failure of the MaxBotix® Inc. product which may result in personal injury or death. MaxBotix® Inc. will not be held liable for unauthorized use of this component.

General Characteristics

- Low cost ultrasonic rangefinder
- Detection out to 5-meters or 10-meters
- Resolution of 1-mm
- Distance sensor from 30-cm to 5-meters or 50-cm to 10-meters based on model
- Excellent ² Mean Time Between Failure (MTBF)

- Triggered operation yields real-time range data
- Free run operation with superior noise rejection ³
- Operating temperature range from -40°C to +65°C ⁷
- Operating voltage of 2.7V to 5.5V⁷
- Nominal current draw of 2.3mA (peak ~49mA) at 3.3V, and 3.1mA (peak ~98mA) at 5V
- IP67 Rated

Range Outputs

- Pulse width, 1uS/mm resolution
- Analog Voltage, 5-mm resolution (5-meter sensors)
- Analog Voltage, 10-mm resolution (10-meter sensors)
- Serial, 1-mm resolution
- Available in RS232 (MB7360 series) or TTL (MB7380 series)

HRXL-MaxSonar-WR Pin Out

Pin 1- Temperature Sensor Connection: Leave this pin unconnected if an external temperature sensor is not used. For best accuracy, this pin is optionally connected to the HR-MaxTemp temperature sensor. Some additional information for the temperature sensor can be found on page 10 of the datasheet.

Pin 2- Pulse Width Output: This pin outputs a pulse width representation of the distance with a scale factor of 1uS per mm. The pulse width output is sent with a value within 0.5% of the serial output.

Pin 3- Analog Voltage Output: This pin outputs a single ended analog voltage scaled representation of the distance. This output is referenced to the sensor ground and Vcc. After the ~50mS power up initialization, the voltage on this pin is set to a low voltage. Once the sensor has completed a range reading the voltage on this pin is set to the voltage corresponding to the latest measured distance.

The 5-meter sensors (MB7360, MB7369, MB7380, and MB7389) use a scale factor of (Vcc/5120) per 1-mm. The distance is output with a 5-mm resolution. The analog voltage output is typically within ±5-mm of the serial output.

The 10-meter sensors (MB7363, MB7366, MB7383, and MB7386) use a scale factor of (Vcc/10240) per 1-mm. The distance is output with a 10-mm resolution. The analog voltage output is typically within ±10-mm of the serial output.

Using a 10-bit analog to digital converter with the 5-meter sensors, one can read the analog voltage counts (i.e. 0 to 1023) directly and just multiply the number of counts in the value by 5 to yield the range in mm. For example, a converted value of 60 corresponds to 300-mm (where 60 x 5 = 300), and 1000 counts corresponds to 5,000-mm (where 1000 x 5 = 5,000-mm).

Using a 10-bit analog to digital converter with the 10-meter sensors, one can read the analog voltage counts (i.e. 0 to 1023) directly and just multiply the number of counts in the value by 10 to yield the range in mm. For example, 30 counts corresponds to 300-mm (where 30 x 10 = 300), and 1000 counts corresponds to 10,000-mm (where 1000 x 10 = 10,000-mm).

Pin 4- Ranging Start/Stop: This pin is internally pulled high. If this pin is left unconnected or held high, the sensor will continually measure and output the range data. If held low, the HRXL-MaxSonar-WR will stop ranging. Bring high for 20uS or longer to command a range reading.

Filtered Range Data: When pin 4 is left high on the sensors, the sensors will continue to range. The data that is output includes a filter for increased accuracy. The sensors will output the range based on recent range information. The filter does not affect the speed at which data is made available to the user but instead allows for more consistent range information to be presented. For sensor specific timing and filter information refer to pages 8 and 9.

Real-time Range Data: When pin 4 is low and then brought high, the sensor will operate in real time and the first reading output will be the range measured from this first commanded range reading. When the sensor tracks that the RX pin is low after each range reading, and then the RX pin is brought high, unfiltered real time range information can be obtained. For timing information please refer to pages 8 and 9.

Pin 5-Serial Output: The MB736X sensors have an RS232 data format (with 0V to Vcc levels) and the MB738X sensors have a TTL outputs. The output is an ASCII capital "R", followed by four ASCII character digits representing the range in millimeters, followed by a carriage return (ASCII 13). The maximum range reported is 4999 mm (5-meter models) or 9998 mm (10-meter models). A range value of 5000 or 9999 corresponds to no target being detected in the field of view.

The serial data format is 9600 baud, 8 data bits, no parity, with one stop bit (9600-8-N-1).

Because the data is presented in a binary data format, the serial output is most accurate .

V+ Pin 6 - Positive Power, Vcc: The sensor operates on voltages from 2.7V - 5.5V DC. For best operation, the sensor requires that the DC power be free from electrical noise. (For installations with known dirty electrical power, a 100uF capacitor placed at the sensor pins between V+ and GND will typically correct the electrical noise.) Please reference page 10 & 11 for minimum operating voltage verses temperature information.

GND Pin 7 – Sensor ground pin: DC return, and circuit common ground.

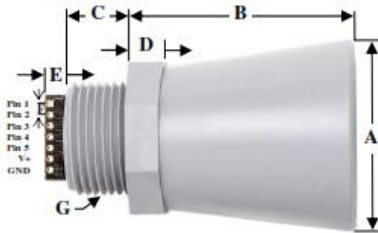
Performance Changes when Selecting a Non-Full Horn Package

When selecting a HRXL-MaxSonar-WR sensor without the full horn the sensor will experience the following performance changes:

- The sensor will have a wider beam shape for the first meter.
- The sensor may be less accurate by an additional +/- 0.5%.
- The sensor may have a dead zone from 0mm-500mm.
- The sensor may have worse performance to small or soft targets.
- The sensor may experience decreased noise immunity when ranging to small, soft, angled, or distant targets.

Mechanical Dimensions

Full Horn

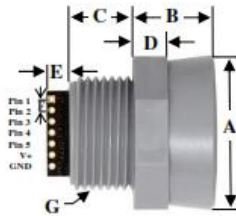


A	1.72" dia.	43.8 mm dia.
B	2.00"	50.7 mm
C	0.58"	14.4 mm
D	0.31"	7.9 mm
E	0.23"	5.8 mm
F	0.1"	2.54 mm
G	3/4"-14 NPS	
H	1.032" dia.	26.2 mm dia.
I	1.37"	34.8 mm
Weight, 1.76 oz., 50 grams		

Values Are Nominal



Compact Housing

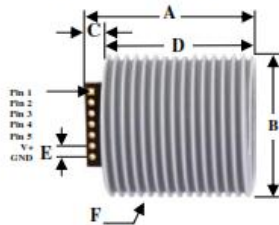


A	1.37" dia.	34.7 mm dia.
B	0.70"	17.9 mm
C	0.57"	14.4 mm
D	0.31"	7.9 mm
E	0.23"	5.8 mm
F	0.1"	2.54 mm
G	3/4"-14 NPS	
H	1.032" dia.	26.2 mm dia.
I	1.37"	34.8 mm
Weight, 1.23 oz., 32 grams		

Values Are Nominal

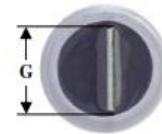


1" NPS Pipe Threading



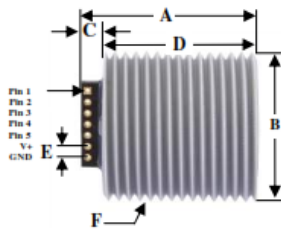
A	1.52"	38.5 mm
B	1.29" dia.	33.0 mm dia.
C	0.22"	5.5 mm
D	1.30"	33.1 mm
E	0.10"	2.54 mm
F	1" - NPS	
G	0.78"	19.81 mm
Weight, 1.23 oz., 35 grams		

Values Are Nominal



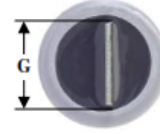
Mechanical Dimensions Continued

1" BSPP Pipe Threading

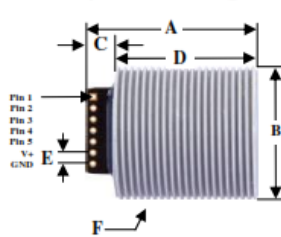


A	1.52"	38.5 mm
B	1.29" dia.	33.0 mm dia.
C	0.22"	5.5 mm
D	1.30"	33.1 mm
E	0.10"	2.54 mm
F	1" — BSPP	
G	0.78"	19.81 mm
Weight, 1.21 oz., 34.3 grams		

Values Are Nominal

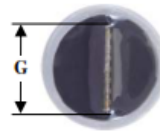


30mm1.5 Pipe Threading



A	1.52"	38.5 mm
B	1.17" dia.	29.7 mm dia.
C	0.22"	5.5 mm
D	1.30"	33.1 mm
E	0.10"	2.54 mm
F	30mm 1.5	
G	0.78"	19.81 mm
Weight, 1.10 oz., 31.1 grams		

Values Are Nominal



Operating Modes

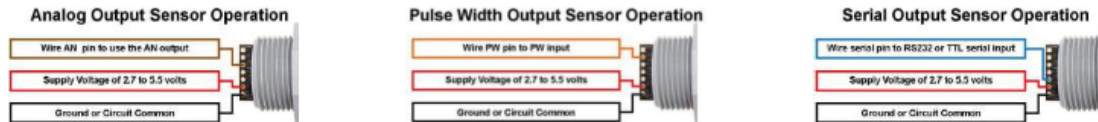
Free-Run Operation

When operating in free run mode, the HRXL-MaxSonar-WR sensors are designed to be used in a variety of outdoor, industrial, or indoor situations. Many acoustic noise sources will have little to no effect on the reported range of the HRXL-MaxSonar-WR sensors. Most range readings are accurately reported. If the range readings are affected, the effect is typically less than 5-mm¹. This allows users to employ real-time ultrasonic distance sensing without the need for additional supporting circuitry or complicated user software.

Multiple HRXL-MaxSonar-WR sensors can be operated in the same general locations. The internal noise filter is able to filter out the ultrasonic noise from other HRXL-MaxSonar-WR sensors with minimal interference. Typically, when operating with multiple sensors, the range readings will be within ±1 cm of the actual range to the intended target.

Independent Sensor Operation

The HRXL-MaxSonar-WR sensors have the capability to operate independently when the user desires. When using the HRXL-MaxSonar-WR sensors in single or independent sensor operation, it is easiest to allow the sensor to free-run. Free-run is the default mode of operation for all of the MaxBotix Inc., sensors. The HRXL-MaxSonar-WR sensors have three separate outputs that update the range data simultaneously: Analog Voltage, Pulse Width, and Serial Data. Below are diagrams on how to connect the sensor for each of the three outputs for single or independent sensor operation.



Using Multiple Sensors in a Single System

Multiple HRXL-MaxSonar-WR sensors can be used simultaneously in the same environment with little to no interference (cross-talk). Even so, some cross-talk may still occur for users wishing to use a large number of sensors in the same environment.

If interference is occurring in the sensor setup please visit www.maxbotix.com/chaining for diagrams on correcting cross-talk between sensors.

Notes: ¹ Refer to section that compares WR to WRC on page 4

ANEXO B

Datasheet Arduino Mega 2560.

1 Arduino Mega2560



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

The Mega 2560 is an update to the [Arduino Mega](#), which it replaces.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB (8 KB used by bootloader)
SRAM	8 KB
EEPROM	4 KB (ATmega328)
Clock Speed	16 MHz

2 Arduino Mega2560

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 5), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

3 Arduino Mega2560

- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available [in the Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labelled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

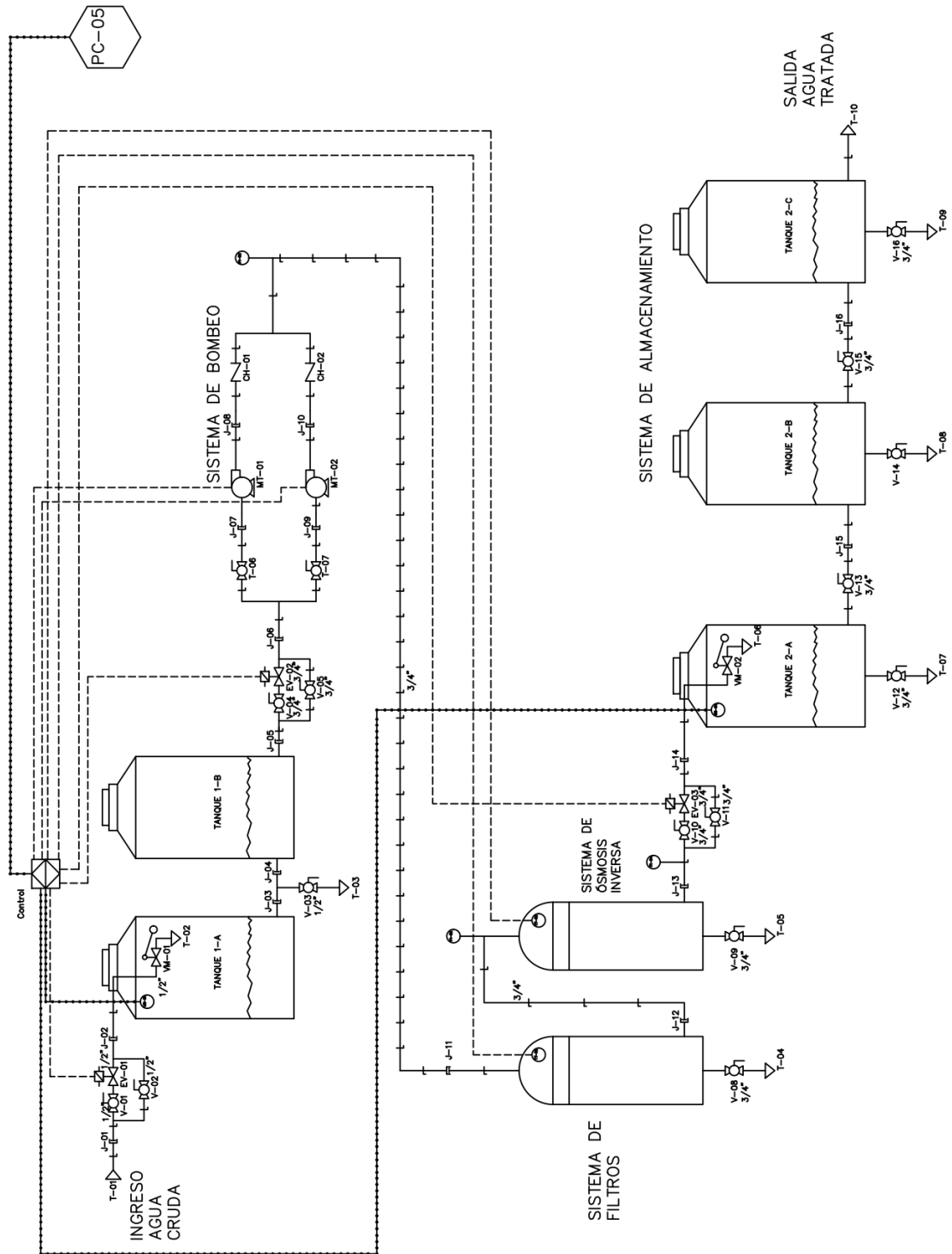
Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

ANEXO C

Plano P&ID del sistema general de control y alerta para el tanque purificador de agua.

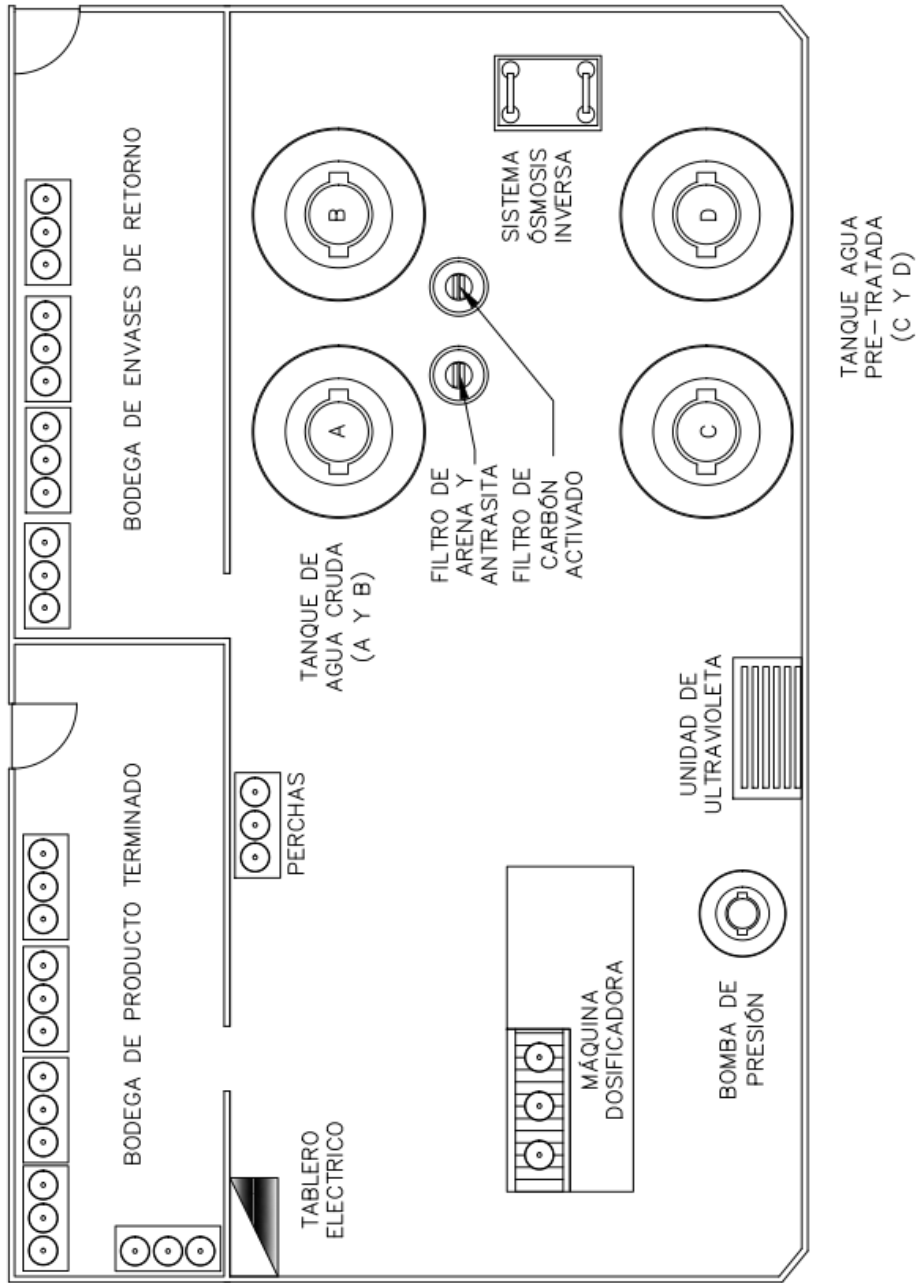


ANEXO D

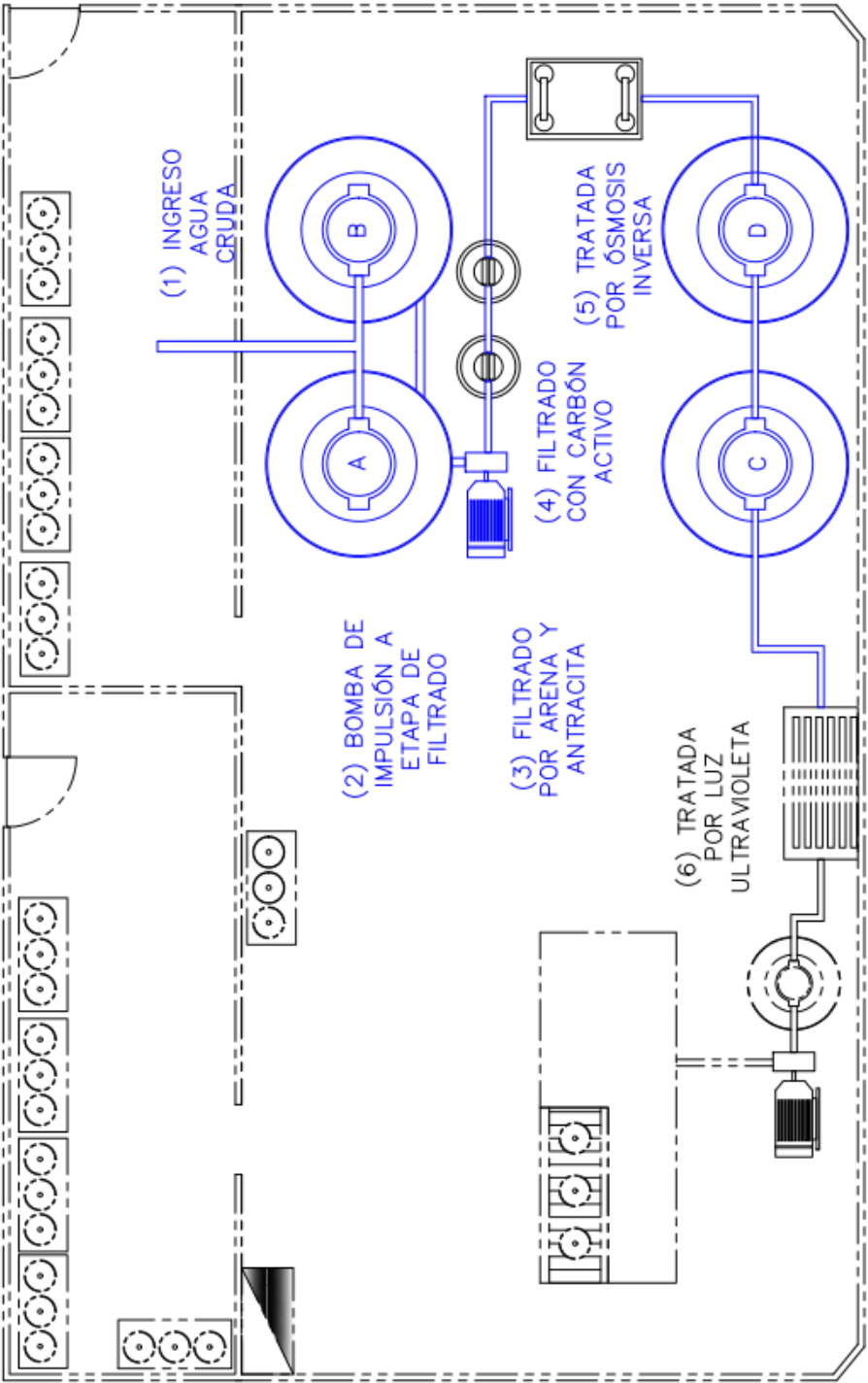
Documentación según IEC61131

Información General:

Distribución de la planta

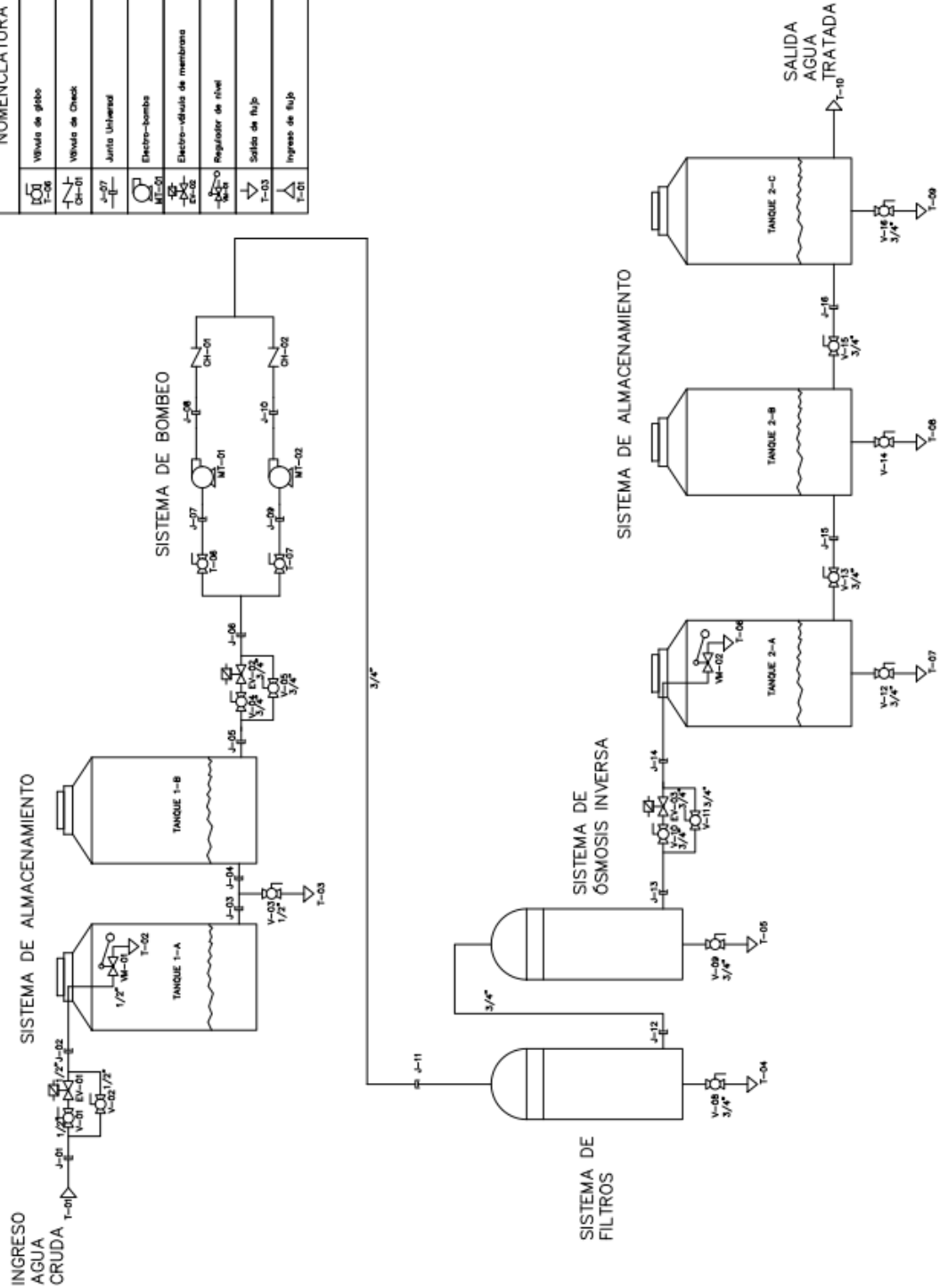


Sistema a Controlar



Simbología del Plano Hidráulico según norma ISA 5.1

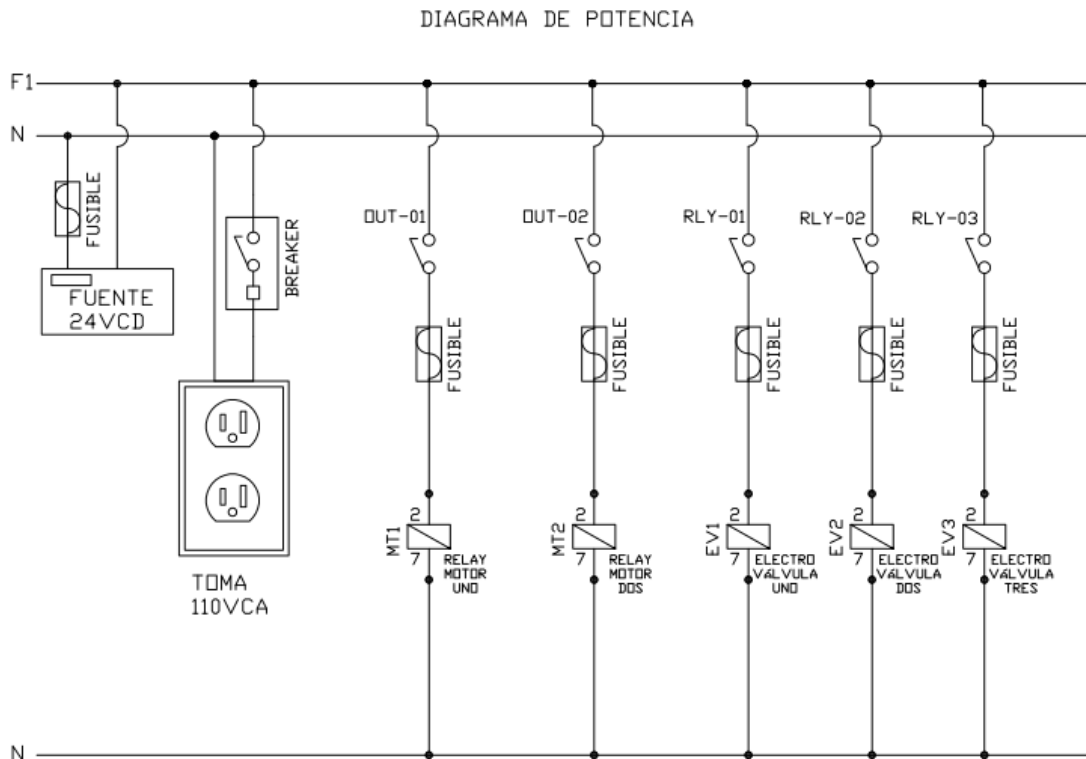
NOMENCLATURA	
	Válvula de globo
	Válvula de Check
	Junta Universal
	Electro-bomba
	Electro-válvula de membrana
	Regulador de nivel
	Salida de flujo
	Ingreso de flujo



ANEXO E

Diagrama de Potencia del tablero Eléctrico secundario

El circuito funciona a 110v, está constituida por una línea de fase y una de neutro, contiene un dispositivo de protección para actuadores denominado fusible que cumple el rol de interrumpir la corriente cuando esta es demasiada alta, haciendo que el fusible se funda en primer lugar y por lo tanto protege al resto del circuito de daños por presencia de sobrecarga o cortocircuito. Adicional a esto cuenta con un toma corriente de 110vca y una fuente estabilizada de 24 vcd con su respectiva protección.



ANEXO F

Código Proteus para la pantalla LCD

El código está desarrollado en PIC C COMPILER y PROTEUS, muestra información acerca de los niveles del tanque 1 y 2, estado en el que se encuentran los actuadores, las operaciones y errores que surgen en el proceso de llenado, las mismas que se visualizan en la pantalla de cristal liquida y luces led respectivamente mediante el uso del PIC18F2550.

```
#include <18F2550.h>
#FUSES WDT //No Watch Dog Timer
#FUSES WDT2048 //Watch Dog Timer uses 1:128 Postscale
#FUSES PLL1 //No PLL PreScaler
#FUSES CPUDIV1 //No System Clock Postscaler
#FUSES NOUSB DIV //USB clock source comes from primary oscillator
#FUSES INTRC_IO //Internal RC Osc, no CLKOUT
#FUSES FCMEN //Fail-safe clock monitor enabled
#FUSES IESO //Internal External Switch Over mode enabled
#FUSES NOPUT //No Power Up Timer
#FUSES NOBROWNOUT //No brownout reset
#FUSES BORV27 //Brownout reset at 2.0V
#FUSES VREGEN //USB voltage regulator enabled
#FUSES PBadEN //PORTB pins are configured as analog input channels on RESET
#FUSES LPT1OSC //Timer1 configured for low-power operation
#FUSES MCLR //Master Clear pin enabled
#FUSES STVREN //Stack full/underflow will cause reset
#FUSES NOLVP //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOXINST //Extended set extension and Indexed Addressing mode disabled (Legacy mode)
#FUSES NODEBUG //No Debug mode for ICD
#FUSES NOPROTECT //Code not protected from reading
#FUSES NOCPB //No Boot Block code protection
#FUSES NOCPD //No EE protection
#FUSES NOWRT //Program memory not write protected
#FUSES NOWRTC //configuration not registers write protected
#FUSES NOWRTB //Boot block not write protected
#FUSES NOWRTD //Data EEPROM not write protected
#FUSES NOEBTR //Memory not protected from table reads
#FUSES NOEBTRB //Boot block not protected from table reads
#use delay(int=8000000)
#use RS232(BAUD=9600, BITS=8, PARITY=N, XMIT=PIN_C6,RCV=PIN_C7)
#include <C:\Program Files (x86)\PICC\Drivers\stdlib.h>
#include <C:\Program Files (x86)\PICC\Drivers\math.h>
#include <C:\Program Files (x86)\PICC\Drivers\stdio.h>
#include <Flex_LCD420.c>
//DECLARACION DE VARIABLES
char cadena[30];
int i=0, operacion=0,horas=0,minutos=0, error=0, n1=0, n2=0, horas_fin=0, minuto_fin=0;
int i6 tanque_1=0, tanque_2=0,valor=0, trama[30];
int32 contador=0, duracion_1=0, duracion_2=0, comunicacion=0, error_comunicacion=0, pulso=0, parpadeo=0;

//CARACTERES ESPECIALES
//FIN CHARACTERS ESPECIAL

//RECEPCION DE DATOS SERIALES
```

```

#int_rda

void serial_isr()
{
    comunicacion=0;
    gets(cadena);
    if((cadena[0]!='*')&&(cadena[26]!='#'))
    {
        trama[0]=cadena[0];/*
        trama[1]=cadena[1];//ev1
        trama[2]=cadena[2];//ev2
        trama[3]=cadena[3];//ev3
        trama[4]=cadena[4];//mot_1
        trama[5]=cadena[5];//mot_2
        trama[6]=cadena[6];//dato1
        trama[7]=cadena[7];//dato2
        trama[8]=cadena[8];//dato3
        trama[9]=cadena[9];//dato4
        trama[10]=cadena[10];//dato1
        trama[11]=cadena[11];//dato2
        trama[12]=cadena[12];//dato3
        trama[13]=cadena[13];//dato4
        trama[14]=cadena[14];//operacion
        trama[15]=cadena[15];//error
        trama[16]=cadena[16];//hora
        trama[17]=cadena[17];//hora
        trama[18]=cadena[18];//minuto
        trama[19]=cadena[19];//minuto
        trama[20]=cadena[20];//nivel 1
        trama[21]=cadena[21];//nivel 2
        trama[22]=cadena[22];//min
        trama[23]=cadena[23];//min
        trama[24]=cadena[24];//seg
        trama[25]=cadena[25];//seg
        trama[26]=cadena[26];##
        printf("%s",cadena);
    }
    //printf("( %c ",cadena[0]);
    //printf("( %c)",cadena[20]);
}

//FIN RECEPCION DATOS SERIALES
void main(void)
{
    //setup_oscillator( OSC_8MHZ|OSC_INTRC,OSC_STATE_STABLE );
    enable_interrupts(int_rda); //Habilitación interrupción por recepción RS232
    enable_interrupts(global);
    //setup_wdt(wdt_1152ms);
    setup_wdt(wdt_on);

    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    SET_TRIS_A(0x00);
    SET_TRIS_B(0x00);
    SET_TRIS_C(0b10000000);
    i=0;
    comunicacion=0;
    duracion_1=240;//80 ES CASI UN SEGUNDO
    duracion_2=160;
    error_comunicacion=500; //100 ES CASI UN SEGUNDO
    parpadeo=80;
}

```

```

restart_wdt();
for(i=0;i<20;i++)
{ trama[i]=48;}
lcd_init();
printf(lcd_putc, "\f");

output_high(PIN_A0);// ELECTROVALVULA_UNO
lcd_gotoxy(1,1);
printf(lcd_putc, " ECO-AGUA \n");
printf(lcd_putc, "*SISTEMA DE LLENADO*\n");
printf(lcd_putc, " PREPARANDO SISTEMA \n");
printf(lcd_putc, " V1.1-2018 ");
restart_wdt();
delay_ms(1000);
output_low(PIN_A0);// ELECTROVALVULA_UNO
lcd_gotoxy(1,4);
printf(lcd_putc, " V1.1-2018. ");
restart_wdt();
delay_ms(1000);
output_high(PIN_A0);// ELECTROVALVULA_UNO
lcd_gotoxy(1,4);
printf(lcd_putc, " V1.1-2018.. ");
restart_wdt();
delay_ms(1000);
output_low(PIN_A0);// ELECTROVALVULA_UNO
lcd_gotoxy(1,4);
printf(lcd_putc, " V1.1-2018... ");
restart_wdt();
delay_ms(1000);
output_high(PIN_A0);// ELECTROVALVULA_UNO
printf(lcd_putc, "\f");

lcd_gotoxy(1,2);
printf(lcd_putc, " SISTEMA LISTO \n");
restart_wdt();
delay_ms(500);
output_low(PIN_A0);// ELECTROVALVULA_UNO
output_low(PIN_A1);// ELECTROVALVULA_DOS
output_low(PIN_A2);// ELECTROVALVULA_TRES
output_low(PIN_A3);// MOTOR UNO
output_low(PIN_A4);// MOTOR DOS
output_low(PIN_A5);//ROJO TANQUE UNO
output_low(PIN_B0);//AMARILLO UNO TANQUE UNO
output_low(PIN_B1);//AMARILLO DOS TANQUE UNO
output_low(PIN_C0);//VERDE TANQUE UNO
output_low(PIN_C1);//ROJO TANQUE DOS
output_low(PIN_C2);//AMARILLO UNO TANQUE DOS
// output_low(PIN_C4);//AMARILLO DOS TANQUE DOS
output_low(PIN_A6);//VERDE TANQUE DOS
printf(lcd_putc, "\f");
restart_wdt();
while(TRUE)
{
restart_wdt();
if(trama[1]=='1')
{output_high(PIN_A0);}// ELECTROVALVULA_UNO
else
{output_low(PIN_A0);}// ELECTROVALVULA_UNO

if(trama[2]=='1')
{output_high(PIN_A1);}// ELECTROVALVULA_DOS

```

```

else
{output_low(PIN_A1);}// ELECTROVALVULA_DOS

if(trama[3]=='1')
{output_high(PIN_A2);}// ELECTROVALVULA_TRES
else
{output_low(PIN_A2);}// ELECTROVALVULA_TRES

if(trama[4]=='1')
{output_high(PIN_A3);}// MOTOR_UNO
else
{output_low(PIN_A3);}// MOTOR_UNO

if(trama[5]=='1')
{output_high(PIN_A4);}// MOTOR_DOS
else
{output_low(PIN_A4);}// MOTOR_DOS
restart_wdt();
valor=0;
valor=valor+(trama[6]-48)*1000;
valor=valor+(trama[7]-48)*100;
valor=valor+(trama[8]-48)*10;
valor=valor+(trama[9]-48)*1;
tanque_1=valor;
restart_wdt();
valor=0;
valor=valor+(trama[10]-48)*1000;
valor=valor+(trama[11]-48)*100;
valor=valor+(trama[12]-48)*10;
valor=valor+(trama[13]-48)*1;
tanque_2=valor;
restart_wdt();
valor=0;
valor=valor+(trama[14]-48)*1;
operacion=valor;
restart_wdt();
valor=0;
valor=valor+(trama[15]-48)*1;
error=valor;
restart_wdt();
valor=0;
valor=valor+(trama[16]-48)*10;
valor=valor+(trama[17]-48)*1;
horas=valor;
restart_wdt();
valor=0;
valor=valor+(trama[18]-48)*10;
valor=valor+(trama[19]-48)*1;
minutos=valor;
restart_wdt();
valor=0;
valor=valor+(trama[20]-48)*1;
n1=valor;
restart_wdt();
valor=0;
valor=valor+(trama[21]-48)*1;
n2=valor;
restart_wdt();
valor=0;
valor=valor+(trama[22]-48)*10;
valor=valor+(trama[23]-48)*1;

```

```

horas_fin=valor;
restart_wdt();
valor=0;
valor=valor+(trama[24]-48)*10;
valor=valor+(trama[25]-48)*1;
minuto_fin=valor;

//INSTRUCCIONES DE VISUALIZACION
if((error==0)&&(comunicacion<error_comunicacion))
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"Tanque 1   Tanque 2");
lcd_gotoxy(1,2);
printf(lcd_putc," %4Lu   ",tanque_1);
lcd_gotoxy(15,2);
printf(lcd_putc,"%4Lu ",tanque_2);
restart_wdt();
if(contador<=duracion_1)
{
restart_wdt();
lcd_gotoxy(1,3);
printf(lcd_putc,"EV1(%c) EV2(%c) EV3(%c)",trama[1],trama[2],trama[3]);
lcd_gotoxy(1,4);
printf(lcd_putc,"MT1(%c) MT2(%c) OP(%c%c)",trama[4],trama[5],trama[14],trama[15]);
restart_wdt();
}
}
else
{
if(operacion==0)
{
restart_wdt();
lcd_gotoxy(1,3);
printf(lcd_putc," VACIANDO   ");
lcd_gotoxy(1,4);
printf(lcd_putc," TANQUES... ");
restart_wdt();
}
if(operacion==1)
{
restart_wdt();
lcd_gotoxy(1,3);
printf(lcd_putc," PRODUCCION ");
lcd_gotoxy(1,4);
printf(lcd_putc," AUTOMATICA ");
restart_wdt();
}
if(operacion==2)
{
restart_wdt();
lcd_gotoxy(1,3);
printf(lcd_putc," TIEMPO DE REPOSO ");
lcd_gotoxy(1,4);
printf(lcd_putc," %c%c:%c%c%c --> %c%c:%c%c",trama[16],trama[17],trama[18],trama[19],trama[22],trama[23],trama[24],trama[25]);
restart_wdt();
}
if(operacion==3)
{

```



```

restart_wdt();
lcd_gotoxy(1,3);
printf(lcd_putc,"          ");
lcd_gotoxy(1,4);
printf(lcd_putc,"  RETRO-LAVADO  ");
restart_wdt();
}
if(operacion==4)
{
restart_wdt();
lcd_gotoxy(1,3);
printf(lcd_putc,"  SISTEMA  ");
lcd_gotoxy(1,4);
printf(lcd_putc,"  DETENIDO  ");
restart_wdt();
}
}
}
restart_wdt();
if(contador>(duracion_1+duracion_2))
{contador=0;}
else
{contador++;}
//printf("%Lu#",contador);

//SECUENCIA DE ERRORES
restart_wdt();
if((error>0)&&(comunicacion<error_comunicacion))
{
if(contador<=duracion_1)
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"Tanque 1  Tanque 2");
lcd_gotoxy(1,2);
printf(lcd_putc," %4Lu  ",tanque_1);
lcd_gotoxy(15,2);
printf(lcd_putc,"%4Lu  ",tanque_2);
lcd_gotoxy(1,3);
printf(lcd_putc,"EV1(%c) EV2(%c) EV3(%c)",trama[1],trama[2],trama[3]);
lcd_gotoxy(1,4);
printf(lcd_putc,"MT1(%c) MT2(%c) OP(%c%c)",trama[4],trama[5],trama[14],trama[15]);
restart_wdt();
}
}
else
{
restart_wdt();
if(error==1)//error nivel crítico tanque 1
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"*****");
lcd_gotoxy(1,2);
printf(lcd_putc,"* NIVEL CRITICO *");
lcd_gotoxy(1,3);
printf(lcd_putc,"* TANQUE UNO *");
lcd_gotoxy(1,4);
printf(lcd_putc,"*****");
restart_wdt();
}
}
}
}
}

```

```

}

if(error==2) //error nivel crítico tanque 2
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"*****");
lcd_gotoxy(1,2);
printf(lcd_putc,"* NIVEL CRITICO *");
lcd_gotoxy(1,3);
printf(lcd_putc,"* TANQUE DOS *");
lcd_gotoxy(1,4);
printf(lcd_putc,"*****");
restart_wdt();
}

if(error==3) //error nivel crítico tanque 1 Y 2
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"*****");
lcd_gotoxy(1,2);
printf(lcd_putc,"* NIVEL CRITICO *");
lcd_gotoxy(1,3);
printf(lcd_putc,"* TANQUE UNO Y DOS *");
lcd_gotoxy(1,4);
printf(lcd_putc,"*****");
restart_wdt();
}

if(error==4) //error FALLA EN SENSOR TANQUE 1
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"*****");
lcd_gotoxy(1,2);
printf(lcd_putc,"* FALLA EN SENSOR *");
lcd_gotoxy(1,3);
printf(lcd_putc,"* TANQUE UNO *");
lcd_gotoxy(1,4);
printf(lcd_putc,"*****");
restart_wdt();
}

if(error==5) //error FALLA EN SENSOR TANQUE 2
{
restart_wdt();
lcd_gotoxy(1,1);
printf(lcd_putc,"*****");
lcd_gotoxy(1,2);
printf(lcd_putc,"* FALLA EN SENSOR *");
lcd_gotoxy(1,3);
printf(lcd_putc,"* TANQUE DOS *");
lcd_gotoxy(1,4);
printf(lcd_putc,"*****");
restart_wdt();
}

if(error==6) //error FALLA EN SENSOR TANQUE 1 Y 2
{
restart_wdt();

```

```

lcd_gotoxy(1,1);
printf lcd_putc, "*****");
lcd_gotoxy(1,2);
printf lcd_putc, "* FALLA EN SENSOR *");
lcd_gotoxy(1,3);
printf lcd_putc, "*TANQUE UNO Y DOS *");
lcd_gotoxy(1,4);
printf lcd_putc, "*****");
restart_wdt();
}

if(error==7) //PAUSE EN CICLO DE VACIADO
{
restart_wdt();
lcd_gotoxy(1,1);
printf lcd_putc, "*****");
lcd_gotoxy(1,2);
printf lcd_putc, "*PARO DE EMERGENCIA*");
restart_wdt();
if(operacion==0)
{
restart_wdt();
lcd_gotoxy(1,3);
printf lcd_putc, "* CICLO DE VACIADO *");
lcd_gotoxy(1,4);
printf lcd_putc, "*****");
restart_wdt();
}
}
if(operacion==1)
{
restart_wdt();
lcd_gotoxy(1,3);
printf lcd_putc, "*PRODUCCION AUTOMAT*");
lcd_gotoxy(1,4);
printf lcd_putc, "*****");
restart_wdt();
}
}
if(operacion==2)
{
restart_wdt();
lcd_gotoxy(1,3);
printf lcd_putc, "*SISTEMA DE REPOSO *");
lcd_gotoxy(1,4);
printf lcd_putc, "*****");
restart_wdt();
}
}
if(operacion==3)
{
restart_wdt();
lcd_gotoxy(1,3);
printf lcd_putc, "* RETRO LAVADO *");
lcd_gotoxy(1,4);
printf lcd_putc, "*****");
restart_wdt();
}
}
if(operacion==4)
{
restart_wdt();
lcd_gotoxy(1,3);
printf lcd_putc, "* SISTEMA DETENIDO *");
lcd_gotoxy(1,4);

```



```

    output_low(PIN_B1);//AMARILLO DOS TANQUE UNO
    output_low(PIN_C0);//VERDE TANQUE UNO
}
restart_wdt();
if(n1==3)
{
    output_high(PIN_A5);//ROJO TANQUE UNO
    if(pulso<=parpadeo)
    {
        output_low(PIN_B0);//AMARILLO UNO TANQUE UNO
    }
    else
    {
        output_high(PIN_B0);//AMARILLO UNO TANQUE UNO
    }
    output_low(PIN_B1);//AMARILLO DOS TANQUE UNO
    output_low(PIN_C0);//VERDE TANQUE UNO
    restart_wdt();
}
restart_wdt();
if(n1==4)
{
    output_high(PIN_A5);//ROJO TANQUE UNO
    output_high(PIN_B0);//AMARILLO UNO TANQUE UNO
    output_low(PIN_B1);//AMARILLO DOS TANQUE UNO
    output_low(PIN_C0);//VERDE TANQUE UNO
}
restart_wdt();
if(n1==5)
{
    output_high(PIN_A5);//ROJO TANQUE UNO
    output_high(PIN_B0);//AMARILLO UNO TANQUE UNO
    if(pulso<=parpadeo)
    {
        output_low(PIN_B1);//AMARILLO DOS TANQUE UNO
    }
    else
    {
        output_high(PIN_B1);//AMARILLO DOS TANQUE UNO
    }
    output_low(PIN_C0);//VERDE TANQUE UNO
}
restart_wdt();
if(n1==6)
{
    output_high(PIN_A5);//ROJO TANQUE UNO
    output_high(PIN_B0);//AMARILLO UNO TANQUE UNO
    output_high(PIN_B1);//AMARILLO DOS TANQUE UNO
    output_low(PIN_C0);//VERDE TANQUE UNO
}
restart_wdt();
if(n1==7)
{
    output_high(PIN_A5);//ROJO TANQUE UNO
    output_high(PIN_B0);//AMARILLO UNO TANQUE UNO
    output_high(PIN_B1);//AMARILLO DOS TANQUE UNO
    if(pulso<=parpadeo)
    {
        output_low(PIN_C0);//VERDE TANQUE UNO
    }
    else

```

```

    {
        output_high(PIN_C0);//VERDE TANQUE UNO
    }

}
restart_wdt();
if(n1==8)
{
    output_high(PIN_A5);//ROJO TANQUE UNO
    output_high(PIN_B0);//AMARILLO UNO TANQUE UNO
    output_high(PIN_B1);//AMARILLO DOS TANQUE UNO
    output_high(PIN_C0);//VERDE TANQUE UNO
}

restart_wdt();
//FIN SECUENCIA PARA LOS INDICADORSDE NIVEL DEL TANQUE UNO

//INICIO SECUENCIA INDICADORES TANQUE DOS

if(n2==0)
{
    output_low(PIN_C1);//ROJO TANQUE DOS
    output_low(PIN_C2);//AMARILLO UNO TANQUE DOS
    output_low(PIN_A6);//VERDE TANQUE DOS
}
restart_wdt();
if(n2==1)
{
    if(pulso<=parpadeo)
    {
        output_low(PIN_C1);//ROJO TANQUE DOS
    }
    else
    {
        output_high(PIN_C1);//ROJO TANQUE DOS
    }
    output_low(PIN_C2);//AMARILLO UNO TANQUE DOS
    output_low(PIN_A6);//VERDE TANQUE DOS
}
restart_wdt();
if(n2==2)
{
    output_high(PIN_C1);//ROJO TANQUE DOS
    output_low(PIN_C2);//AMARILLO UNO TANQUE DOS
    output_low(PIN_A6);//VERDE TANQUE DOS
}

if(n2==3)
{
    output_high(PIN_C1);//ROJO TANQUE DOS
    if(pulso<=parpadeo)
    {
        output_low(PIN_C2);//AMARILLO UNO TANQUE DOS
    }
    else
    {
        output_high(PIN_C2);//AMARILLO UNO TANQUE DOS
    }
    output_low(PIN_A6);//VERDE TANQUE DOS
}
}

```

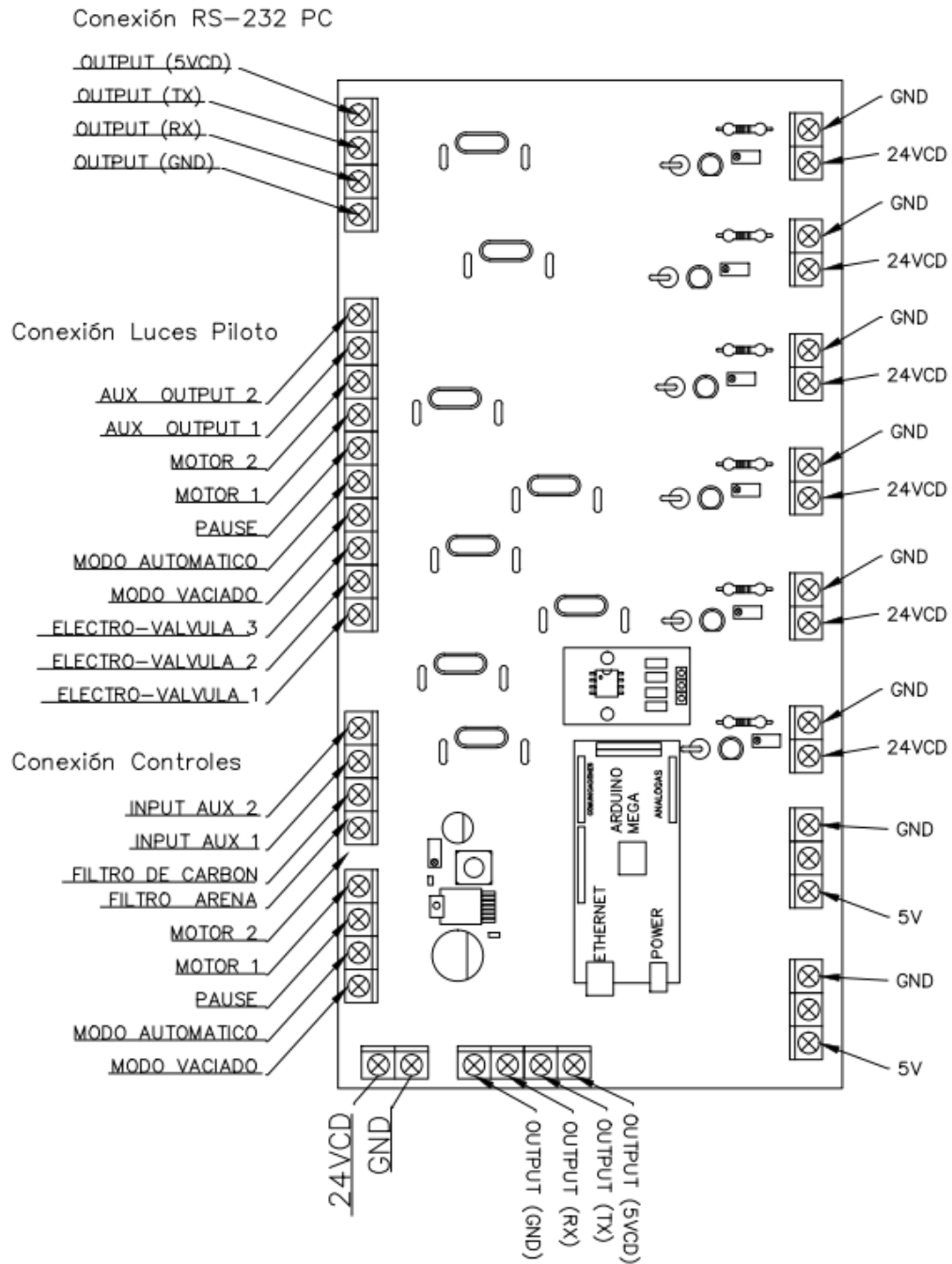
```

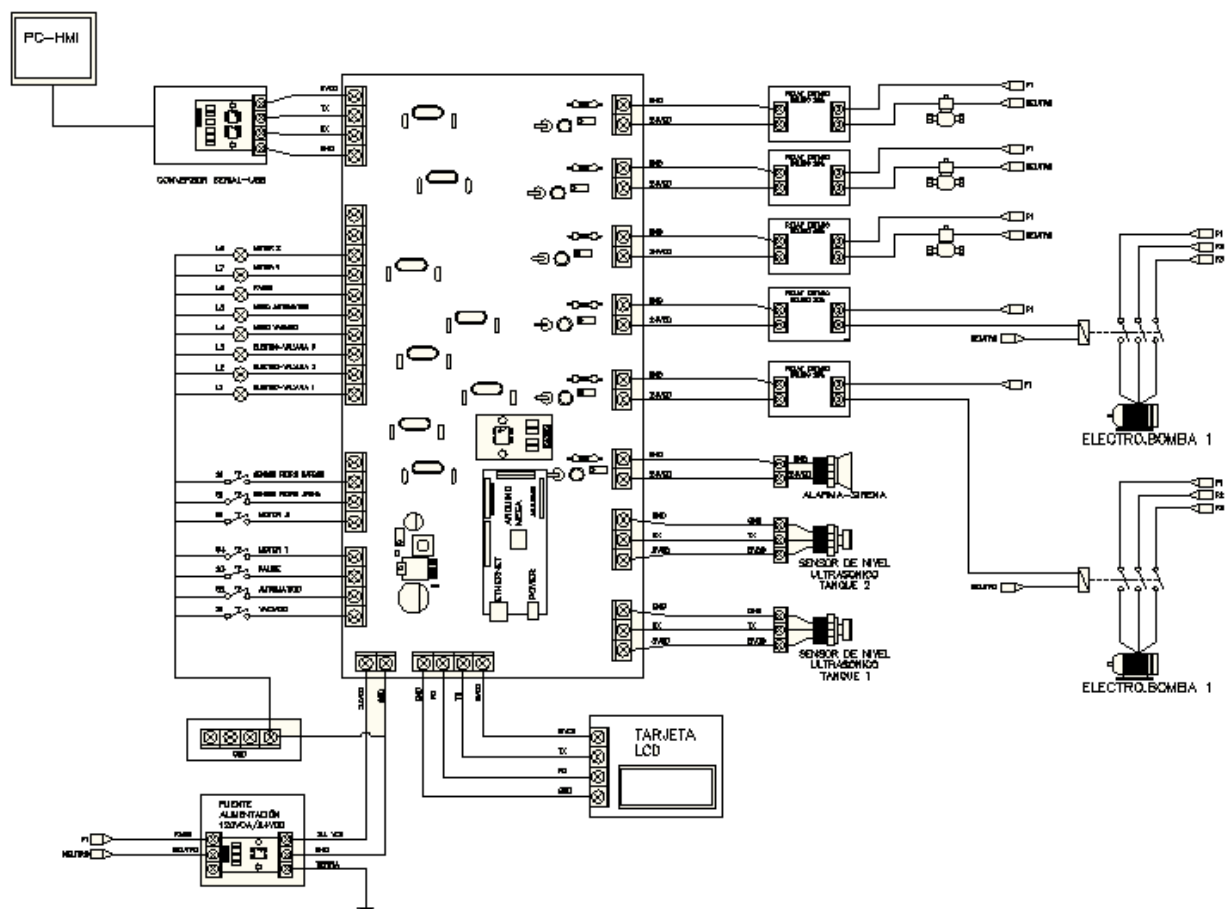
restart_wdt();
if(n2==4)
{
    output_high(PIN_C1);//ROJO TANQUE DOS
    output_high(PIN_C2);//AMARILLO UNO TANQUE DOS
    output_low(PIN_A6);//VERDE TANQUE DOS
}
restart_wdt();
if(n2==5)
{
    output_high(PIN_C1);//ROJO TANQUE DOS
    output_high(PIN_C2);//AMARILLO UNO TANQUE DOS
    if(pulso<=parpadeo)
    {
        output_low(PIN_A6);//VERDE TANQUE DOS
    }
    else
    {
        output_high(PIN_A6);//VERDE TANQUE DOS
    }
}
restart_wdt();
if(n2==4)
{
    output_high(PIN_C1);//ROJO TANQUE DOS
    output_high(PIN_C2);//AMARILLO UNO TANQUE DOS
    output_high(PIN_A6);//VERDE TANQUE DOS
}
//FIN INICIO SECUENCIA INDICADORES TANQUE DOS
restart_wdt();
if(pulso>(2*parpadeo))
{
    pulso=0;
}
else
{
    pulso++;
}
restart_wdt();
} // fin WHILE (true)
} // fin funcion VOID

```

ANEXO G

Conexiones de elementos en la tarjeta de control





ANEXO H

Código de Programación en la Tarjeta De Control

El siguiente código está desarrollado en el IDE de Arduino, para el control de las electroválvulas, motores, botones de mando e indicadores de visualización, modos de operación del sistema, relevo de motores y alarma mediante comunicación serial y utilizando el microcontrolador Arduino Mega 2560.

```
//LIBRERIA EXTERNAS
#include <Wire.h>
#include "RTclib.h"
//#include <stdlib.h>
RTC_DS1307 rtc;

//DEFINICION PINES DE ENTRADA
int modo_vaciado=62, modo_automatico=63, paro_emergencia=64;
int selector_motor_1=65, selector_motor_2=66;
int retro_lavado_1=67, retro_lavado_2=68, entrada_aux_3=69;

//DEFINICION PINES DE SALIDA INDICADORES
int piloto_electrovalvula_1=37, piloto_electrovalvula_2=39, piloto_electrovalvula_3=41;
int piloto_vaciado=35, piloto_automatico=33, piloto_pause=31, piloto_bomba_1=29, piloto_bomba_2=27;
int piloto_aux_1=25, piloto_aux_2=23;

//DEFINICION DE PINES SALIDA ACTUADORES
int aux_output=7, motor_2=5, motor_1=6;
int electrovalvula_3=4, electrovalvula_2=3, electrovalvula_1=2;

//VARIABLES DEL SISTEMA
String TRAMA_0 = "";
boolean TRAMA_COMPLETA_0 = false;
boolean TRAMA_COMPLETA_1 = false;
boolean TRAMA_COMPLETA_2 = false;
String TRAMA_3 = "";
boolean TRAMA_COMPLETA_3 = false;
char TX_HMI[20];
char TRAMA_LCD[20];
char trama_aux[9];
long max_1=0, min_1=0, ls_1=0, sm_1=0, max_2=0, min_2=0, ls_2=0, sm_2=0;
long ht_tanque_1=0, ht_tanque_2=0;
int enable_motor=0, nivel_1=0, nivel_2=0, operacion=0, error=0, ciclo=0, minuto_inicio=0, hora_inicio=0, evento=0;

char ultrasonico_1[5];
char ultrasonico_2[5];
float tanque_1=0, tanque_2=0, diametro_1=0, diametro_2=0, volumen_1=0, volumen_2=0;
int marca=0, secuencia=0, duracion_hora=0, duracion_minuto=0, retrolavado=0;
long monitoreo_1=0, monitoreo_2=0, delta_x1=0, delta_x2=0;
void setup()
{
  sm_1=1150;
  max_1=1050;
  ls_1=600;
  min_1=250;
```

```

sm_2=1150;
max_2=1050;
ls_2=600;
min_2=200;

ht_tanque_1=1505;
diametro_1=1550;
ht_tanque_2=1485;
diametro_2=1550;

duracion_hora=2;
duracion_minuto=0;

Serial.begin(9600);
Serial1.begin(9600);
Serial2.begin(9600);
Serial3.begin(9600);
TRAMA_0.reserve(200);
TRAMA_3.reserve(200);
pinMode(aux_output,OUTPUT);
pinMode(motor_2,OUTPUT);
pinMode(motor_1,OUTPUT);
pinMode(electrovalvula_3,OUTPUT);
pinMode(electrovalvula_2,OUTPUT);
pinMode(electrovalvula_1,OUTPUT);
pinMode(modos_vaciado,INPUT);
pinMode(modos_automatizado,INPUT);
pinMode(paro_emergencia,INPUT);
pinMode(selector_motor_1,INPUT);
pinMode(selector_motor_2,INPUT);
pinMode(retro_lavado_1,INPUT);
pinMode(retro_lavado_2,INPUT);
pinMode(entrada_aux_3,INPUT);
pinMode(piloto_electrovalvula_1,OUTPUT);
pinMode(piloto_electrovalvula_2,OUTPUT);
pinMode(piloto_electrovalvula_3,OUTPUT);
pinMode(piloto_vaciado,OUTPUT);
pinMode(piloto_automatizado,OUTPUT);
pinMode(piloto_pause,OUTPUT);
pinMode(piloto_bomba_1,OUTPUT);
pinMode(piloto_bomba_2,OUTPUT);
pinMode(piloto_aux_1,OUTPUT);
pinMode(piloto_aux_2,OUTPUT);
if (! rtc.begin())
{
  //Serial.println("Couldn't find RTC");
  while (1);
}

if (! rtc.isrunning())
{
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
}
}

void loop()
{
  entradas();
  puerto_serial_0();
  puerto_serial_1();
}

```

```

puerto_serial_2();
puerto_serial_3();
if(TRAMA_COMPLETA_0)
{
  TRAMA_0="";
  TRAMA_COMPLETA_0=false;
}

if(TRAMA_COMPLETA_1)
{

  TRAMA_COMPLETA_1=false;
  monitoreo_1=0;
}

if(TRAMA_COMPLETA_2)
{
  TRAMA_COMPLETA_2=false;
  monitoreo_2=0;
}

if(TRAMA_COMPLETA_3)
{
  TRAMA_3="";
  TRAMA_COMPLETA_3=false;
}
if((digitalRead(modos_vaciado)==1)&&(digitalRead(modos_automatico)==1)&&(retrolavado==0))
{
  operacion=4;
  digitalWrite(electrovalvula_1,LOW);
  digitalWrite(electrovalvula_2,LOW);
  digitalWrite(electrovalvula_3,LOW);
  enable_motor=0;
  ciclo=0;
  secuencia=0;
  marca=0;
  retrolavado=0;
}

//SECUENCIA PARA MODO DE VACIADO
if((digitalRead(modos_vaciado)==0)&&(digitalRead(modos_automatico)==1)&&(digitalRead(paros_emergencia)==0)
&&(retrolavado==0))
{
  operacion=0;
  digitalWrite(electrovalvula_1,LOW);
  if((tanque_1<=min_1)&&(tanque_2<=min_2))
  {
    enable_motor=0;
    digitalWrite(electrovalvula_2,LOW);
    digitalWrite(electrovalvula_3,LOW);
    marca=0;
    operacion=5;
  }
  else
  {
    if((tanque_1<=min_1)&&(tanque_2>min_2))
    {
      enable_motor=0;
      digitalWrite(electrovalvula_2,LOW);
    }
  }
}

```

```

digitalWrite(electrovalvula_3,LOW);
marca=0;
}
else
{
if((tanque_1>min_1)&&(tanque_2<ls_2)&&(marca==0))
{
marca=1;
}
else
{
if((tanque_1>min_1)&&(tanque_2>=ls_2)&&(tanque_2<max_2)&&(marca==0))
{
marca=0;
}
}
}
}
if(marca==0)
{
enable_motor=0;
digitalWrite(electrovalvula_2,LOW);
digitalWrite(electrovalvula_3,LOW);
marca=0;
}
if(marca==1)
{
enable_motor=1;
digitalWrite(electrovalvula_2,HIGH);
digitalWrite(electrovalvula_3,HIGH);
if(tanque_2>=max_2)
{
marca=0;
enable_motor=0;
digitalWrite(electrovalvula_2,LOW);
digitalWrite(electrovalvula_3,LOW);
}
}
}
//FIN MODO VACIADO
//-----
//INICIO MODO AUTOMATICO
if((digitalRead(modos_vaciado)==1)&&(digitalRead(modos_automatico)==0)&&(digitalRead(paros_emergencia)==0)
&&(retrolavado==0))
{
operacion=1;
if((tanque_1>ls_1)&&(ciclo==0))
{
ciclo=1;
}
else
{
if((tanque_1<=ls_1)&&(ciclo==0))
{
ciclo=2;
}
else
{
if(ciclo==1)
{
digitalWrite(electrovalvula_1,LOW);

```

```

if((tanque_2<=ls_2)&&(secuencia==0))
{
    secuencia=1;
}
else
{
    if(secuencia==1)
    {
        digitalWrite(electrovalvula_2,HIGH);
        digitalWrite(electrovalvula_3,HIGH);
        enable_motor=1;
        if(tanque_2>max_2)
        {
            secuencia=0;
            digitalWrite(electrovalvula_2,LOW);
            digitalWrite(electrovalvula_3,LOW);
            enable_motor=0;
        }
    }
    else
    {
        secuencia=0;
        digitalWrite(electrovalvula_2,LOW);
        digitalWrite(electrovalvula_3,LOW);
        enable_motor=0;
    }

    if(tanque_1<=ls_1)
    {
        digitalWrite(electrovalvula_2,LOW);
        digitalWrite(electrovalvula_3,LOW);
        enable_motor=0;
        ciclo=0;
    }
}
else
{
    if(ciclo==2)
    {
        digitalWrite(electrovalvula_1,HIGH);
        digitalWrite(electrovalvula_2,LOW);
        digitalWrite(electrovalvula_3,LOW);
        enable_motor=0;
        if(tanque_1>max_1)
        {
            ciclo=3;
            digitalWrite(electrovalvula_1,LOW);
            digitalWrite(electrovalvula_2,LOW);
            digitalWrite(electrovalvula_3,LOW);
            enable_motor=0;
            DateTime now = rtc.now();
            DateTime future (now+TimeSpan(0,duracion_hora,duracion_minuto,0));
            hora_inicio=future.hour();
            minuto_inicio=future.minute();
        }
    }
    else
    {
        if(ciclo==3)
        {

```

```

operacion=2;
digitalWrite(electrovalvula_1,LOW);
digitalWrite(electrovalvula_2,LOW);
digitalWrite(electrovalvula_3,LOW);
enable_motor=0;
DateTime now = rtc.now();
if((hora_inicio==now.hour())&&(minuto_inicio==now.minute()))
{
  ciclo=0;
}
}
}
}
}

} //FIN MODO AUTOMATICO

//INICIO SECUENCIA DE RETRO LAVADO FILTRO UNO
if(((digitalRead(retro_lavado_1)==0)||((digitalRead(retro_lavado_2)==0)&&(digitalRead(paro_emergencia)==0))
{
  if(tanque_1>min_1)
  {
    retrolavado==1;
    operacion=3;
    enable_motor=1;
    digitalWrite(electrovalvula_2,HIGH);
  }
}
else
{
  retrolavado=0;
}
} //FIN SECUENCIA DE RETRO LAVADO FILTRO UNO

//ACCIONES DE EMERGENCIA
if(tanque_1>sm_1)
{
  digitalWrite(electrovalvula_1,LOW);
}
if(tanque_1<(min_1-50))
{
  digitalWrite(electrovalvula_2,LOW);
  digitalWrite(electrovalvula_3,LOW);
  enable_motor=0;
}
if(tanque_2>sm_2)
{
  digitalWrite(electrovalvula_2,LOW);
  digitalWrite(electrovalvula_3,LOW);
  enable_motor=0;
}
} //FIN ACCIONES DE EMERGENCIA

//-----
//MODO RELEVOS MOTORES
if((enable_motor==1)&&(digitalRead(paro_emergencia)==0))
{

```

```

if((digitalRead(selector_motor_1)==0)&&(digitalRead(selector_motor_2)==1))
{
digitalWrite(motor_1,HIGH);
digitalWrite(motor_2,LOW);
}
else
{
if((digitalRead(selector_motor_1)==1)&&(digitalRead(selector_motor_2)==0))
{
digitalWrite(motor_1,LOW);
digitalWrite(motor_2,HIGH);
}
else
{
if((digitalRead(selector_motor_1)==1)&&(digitalRead(selector_motor_2)==1))
{
DateTime now = rtc.now();

if((now.minute()<=10)||((now.minute())>20)&&(now.minute()<=30))||((now.minute())>40)&&(now.minute()<=50))
{
digitalWrite(motor_1,HIGH);
digitalWrite(motor_2,LOW);
}
else
{
digitalWrite(motor_1,LOW);
digitalWrite(motor_2,HIGH);
}
}
}
}
else
{
digitalWrite(motor_1,LOW);
digitalWrite(motor_2,LOW);
}
}
//FIN RELEVOS MOTORES

//MONITOREO DE COMUNICACION SENSORES
if(monitoreo_1>10)
{
monitoreo_1=11;
}
else
{
if(monitoreo_2>10)
{
monitoreo_2=11;
}
}
monitoreo_1++;
monitoreo_2++;
//FIN MONITOREO DE COMUNICACION SENSORES

//INICO SECUENCIA DE ERRORES
if(digitalRead(paro_emergencia)==1)
{
error=7;
digitalWrite(electrovalvula_1,LOW);
digitalWrite(electrovalvula_2,LOW);
}

```



```

delay(500);
} //fin void loop
//*****
//*****
//*****
void reporte_hmi()
{
  Serial3.print('*');
  Serial3.print(int(digitalRead(electrovalvula_1))); Serial3.print(';');
  Serial3.print(int(digitalRead(electrovalvula_2))); Serial3.print(';');
  Serial3.print(int(digitalRead(electrovalvula_3))); Serial3.print(';');
  Serial3.print(int(digitalRead(motor_1))); Serial3.print(';');
  Serial3.print(int(digitalRead(motor_2))); Serial3.print(';');
  Serial3.print(int(tanque_1)); Serial3.print(';');
  Serial3.print(float((tanque_1*100)/ht_tanque_1)); Serial3.print(';');
  float volumen_1=(((3.1416/4)*diametro_1*diametro_1*tanque_1)*0.000001; //volumen en litros T1
  //Serial.println(' ');
  //Serial.print("Volumen_x= ");
  //Serial.println(float(volumen_1*2));
  Serial3.print(float(volumen_1*2)); Serial3.print(';');
  Serial3.print(int(tanque_2)); Serial3.print(';');
  Serial3.print(float((tanque_2*100)/ht_tanque_2)); Serial3.print(';');
  float volumen_2=(((3.1416/4)*diametro_2*diametro_2*tanque_2)*0.000001; //volumen en litros T2
  Serial3.print(float(volumen_2*2)); Serial3.print(';');
  DateTime now = rtc.now();
  Serial3.print(int(now.hour())); Serial3.print(';');
  Serial3.print(int(now.minute())); Serial3.print(';');
  Serial3.print(operacion); Serial3.print(';');
  Serial3.print(error); Serial3.print(';');
  Serial3.print(hora_inicio); Serial3.print(';');
  Serial3.print(minuto_inicio); Serial3.print(';');
  Serial3.print(ht_tanque_1); Serial3.print(';');
  Serial3.print(sm_1); Serial3.print(';');
  Serial3.print(max_1); Serial3.print(';');
  Serial3.print(ls_1); Serial3.print(';');
  Serial3.print(min_1); Serial3.print(';');
  Serial3.print(ht_tanque_2); Serial3.print(';');
  Serial3.print(sm_2); Serial3.print(';');
  Serial3.print(max_2); Serial3.print(';');
  Serial3.print(ls_2); Serial3.print(';');
  Serial3.print(min_2); Serial3.print(';');
  Serial3.print(int(now.second())); Serial3.print(';');
  Serial3.print(marca); Serial3.print(';');
  Serial3.print(ciclo); Serial3.print(';');
  Serial3.print(secuencia); Serial3.print(';');
  Serial3.println('#');
}

//FUNCIONES EXTRA POLARES
void reporte_lcd()
{
  //UTILIZANDO PUERTO 3 CON MAX-RS232
  Serial.print('*');
  Serial.print(digitalRead(electrovalvula_1));
  Serial.print(digitalRead(electrovalvula_2));
  Serial.print(digitalRead(electrovalvula_3));
  Serial.print(digitalRead(motor_1));
  Serial.print(digitalRead(motor_2));
  if(tanque_1<10)
  {
    Serial.print("000");//DATO 1 TANQUE 1
  }
}

```

```

Serial.print(int(tanque_1));//DATO 1 TANQUE 1
}
else
{
  if(tanque_1<100)
  {
    Serial.print("00");//DATO 1 TANQUE 1
    Serial.print(int(tanque_1));//DATO 1 TANQUE 1
  }
  else
  {
    if(tanque_1<1000)
    {
      Serial.print('0');//DATO 1 TANQUE 1
      Serial.print(int(tanque_1));//DATO 1 TANQUE 1
    }
    else
    {
      Serial.print(int(tanque_1));//DATO 1 TANQUE 1
    }
  }
}

if(tanque_2<10)
{
  Serial.print("000");//DATO 1 TANQUE 2
  Serial.print(int(tanque_2));//DATO 1 TANQUE 2
}
else
{
  if(tanque_2<100)
  {
    Serial.print("00");//DATO 1 TANQUE 2
    Serial.print(int(tanque_2));//DATO 1 TANQUE 2
  }
  else
  {
    if(tanque_2<1000)
    {
      Serial.print('0');//DATO 1 TANQUE 2
      Serial.print(int(tanque_2));//DATO 1 TANQUE 2
    }
    else
    {
      Serial.print(int(tanque_2));//DATO 1 TANQUE 2
    }
  }
}

Serial.print(operacion); //OPERACION
Serial.print(error); //ERROR

DateTime now = rtc.now();
if(int(now.hour())<10)
{
  Serial.print('0');
}
Serial.print(int(now.hour()));

if(now.minute()<10)
{

```

```

    Serial.print('0');
  }
  Serial.print(int(now.minute()));
  Serial.print(nivel_1);
  Serial.print(nivel_2);
  if(hora_inicio<10)
  {
    Serial.print('0');
    Serial.print(hora_inicio);
  }
  else
  {
    Serial.print(hora_inicio);
  }
  if(minuto_inicio<10)
  {
    Serial.print('0');
    Serial.print(minuto_inicio);
  }
  else
  {
    Serial.print(minuto_inicio);
  }
  Serial.print('#');
  Serial.print('\r');
}

```

```

void puerto_serial_0()
{
  while (Serial.available())
  {
    char inChar = (char)Serial.read();
    TRAMA_0 += inChar;
    if (inChar == '\n') {
      TRAMA_COMPLETA_0 = true;
    }
  }
}

```

```

void puerto_serial_1()
{
  while (Serial1.available(>0))
  {
    char inChar = (char)Serial1.read();
    if((inChar=='R')&&(TRAMA_COMPLETA_1==false))
    {
      long val_tq1=0;
      ultrasonico_1[0]=(char)Serial1.read();
      ultrasonico_1[1]=(char)Serial1.read();
      ultrasonico_1[2]=(char)Serial1.read();
      ultrasonico_1[3]=(char)Serial1.read();
      val_tq1 +=(ultrasonico_1[0]-48)*1000;
      val_tq1 +=(ultrasonico_1[1]-48)*100;
      val_tq1 +=(ultrasonico_1[2]-48)*10;
      val_tq1 +=(ultrasonico_1[3]-48)*1;
      if((val_tq1>0)&&(val_tq1<5000))
      {
        if(ht_tanque_1-val_tq1)>0)
        {
          tanque_1= ht_tanque_1-val_tq1;

```

```

        trama_aux[0]=ultrasonico_1[0];
        trama_aux[1]=ultrasonico_1[1];
        trama_aux[2]=ultrasonico_1[2];
        trama_aux[3]=ultrasonico_1[3];
        TRAMA_COMPLETA_1 = true;
    }
}
}
}

void puerto_serial_2()
{
    while (Serial2.available()>0)
    {
        char inChar = (char)Serial2.read();
        if((inChar=='R')&&(TRAMA_COMPLETA_2==false))
        {
            long val_tq2=0;
            ultrasonico_2[0]=(char)Serial2.read();
            ultrasonico_2[1]=(char)Serial2.read();
            ultrasonico_2[2]=(char)Serial2.read();
            ultrasonico_2[3]=(char)Serial2.read();
            val_tq2 +=(ultrasonico_2[0]-48)*1000;
            val_tq2 +=(ultrasonico_2[1]-48)*100;
            val_tq2 +=(ultrasonico_2[2]-48)*10;
            val_tq2 +=(ultrasonico_2[3]-48)*1;
            if((val_tq2>0)&&(val_tq2<2000))
            {
                if((ht_tanque_2-val_tq2)>0)
                {
                    tanque_2= ht_tanque_2-val_tq2;
                    trama_aux[4]=ultrasonico_2[0];
                    trama_aux[5]=ultrasonico_2[1];
                    trama_aux[6]=ultrasonico_2[2];
                    trama_aux[7]=ultrasonico_2[3];
                    TRAMA_COMPLETA_2 = true;
                }
            }
        }
    }
}

void puerto_serial_3()
{
    while (Serial3.available())
    {
        char inChar = (char)Serial3.read();
        TRAMA_3 += inChar;
        if (inChar == '\n')
        {
            TRAMA_COMPLETA_3 = true;
        }
    }
}

void puerto_de_salida()
{

```

```

digitalWrite(aux_output,HIGH);
delay(10);
digitalWrite(aux_output,LOW);
delay(10);
digitalWrite(motor_2,HIGH);
delay(10);
digitalWrite(motor_2,LOW);
delay(10);
digitalWrite(motor_1,HIGH);
delay(10);
digitalWrite(motor_1,LOW);
delay(10);
digitalWrite(electrovalvula_3,HIGH);
delay(10);
digitalWrite(electrovalvula_3,LOW);
delay(10);
digitalWrite(electrovalvula_2,HIGH);
delay(10);
digitalWrite(electrovalvula_2,LOW);
delay(10);
digitalWrite(electrovalvula_1,HIGH);
delay(10);
digitalWrite(electrovalvula_1,LOW);
delay(10);
}

```

```

void puerto_luz_piloto()
{
digitalWrite(piloto_electrovalvula_1,HIGH);
digitalWrite(piloto_electrovalvula_2,HIGH);
digitalWrite(piloto_electrovalvula_3,HIGH);
digitalWrite(piloto_vaciado,HIGH);
digitalWrite(piloto_automatico,HIGH);
digitalWrite(piloto_pause,HIGH);
digitalWrite(piloto_bomba_1,HIGH);
digitalWrite(piloto_bomba_2,HIGH);
digitalWrite(piloto_aux_1,HIGH);
digitalWrite(piloto_aux_2,HIGH);
delay(2000);
digitalWrite(piloto_electrovalvula_1,LOW);
digitalWrite(piloto_electrovalvula_2,LOW);
digitalWrite(piloto_electrovalvula_3,LOW);
digitalWrite(piloto_vaciado,LOW);
digitalWrite(piloto_automatico,LOW);
digitalWrite(piloto_pause,LOW);
digitalWrite(piloto_bomba_1,LOW);
digitalWrite(piloto_bomba_2,LOW);
digitalWrite(piloto_aux_1,LOW);
digitalWrite(piloto_aux_2,LOW);
delay(2000);
}

```

```

void entradas()
{
if(digitalRead(modos_vaciado)==1) {TX_HMI[0]='0';}else{TX_HMI[0]='1';}
if(digitalRead(modos_automatico)==1) {TX_HMI[1]='0';}else{TX_HMI[1]='1';}
if(digitalRead(modos_emergencia)==1) {TX_HMI[2]='0';}else{TX_HMI[2]='1';}
if(digitalRead(selector_motor_1)==1) {TX_HMI[3]='0';}else{TX_HMI[3]='1';}
if(digitalRead(selector_motor_2)==1) {TX_HMI[4]='0';}else{TX_HMI[4]='1';}
if(digitalRead(retro_lavado_1)==1) {TX_HMI[5]='0';}else{TX_HMI[5]='1';}
if(digitalRead(retro_lavado_2)==1) {TX_HMI[6]='0';}else{TX_HMI[6]='1';}
}

```

```

if(digitalRead(entrada_aux_3)==1) {TX_HMI[7]='0';}else{TX_HMI[7]='1';}
//Serial.println(TX_HMI);
}

void reloj()
{
  DateTime now = rtc.now();
  Serial.print(now.year(), DEC);
  Serial.print("/");
  Serial.print(now.month(), DEC);
  Serial.print("/");
  Serial.print(now.day(), DEC);
  Serial.print(" ");
  Serial.print(now.hour(), DEC);
  Serial.print(":");
  Serial.print(now.minute(), DEC);
  Serial.print(":");
  Serial.print(now.second(), DEC);
  Serial.println();
}

void niveles_tanque_1()
{
  delta_x1=(ht_tanque_1/8);
  if(tanque_1<delta_x1){nivel_1=1;}
  if((tanque_1>=delta_x1)&&(tanque_1<(2*delta_x1))){nivel_1=2;}
  if((tanque_1>=(2*delta_x1))&&(tanque_1<(3*delta_x1))){nivel_1=3;}
  if((tanque_1>=(3*delta_x1))&&(tanque_1<(4*delta_x1))){nivel_1=4;}
  if((tanque_1>=(4*delta_x1))&&(tanque_1<(5*delta_x1))){nivel_1=5;}
  if((tanque_1>=(5*delta_x1))&&(tanque_1<(6*delta_x1))){nivel_1=6;}
  if((tanque_1>=(6*delta_x1))&&(tanque_1<(7*delta_x1))){nivel_1=7;}
  if(tanque_1>=(7*delta_x1)){nivel_1=8;}
}

void niveles_tanque_2()
{
  delta_x2=(ht_tanque_2/8);
  if(tanque_2<delta_x2){nivel_2=1;}
  if((tanque_2>=delta_x2)&&(tanque_2<(2*delta_x2))){nivel_2=2;}
  if((tanque_2>=(2*delta_x2))&&(tanque_2<(4*delta_x2))){nivel_2=3;}
  if((tanque_2>=(4*delta_x2))&&(tanque_2<(6*delta_x2))){nivel_2=4;}
  if((tanque_2>=(6*delta_x2))&&(tanque_2<(7*delta_x2))){nivel_2=5;}
  if(tanque_2>=(7*delta_x2)){nivel_1=6;}
}

void luces_piloto()
{
  //LUZ PILOTO VACIADO
  if(digitalRead(modos_vaciado)==0)
  { digitalWrite(piloto_vaciado,HIGH); }
  else
  { digitalWrite(piloto_vaciado,LOW); }
  //LUZ PILOTO AUTOMATICO
  if(digitalRead(modos_automatico)==0)
  { digitalWrite(piloto_automatico,HIGH); }
  else
  { digitalWrite(piloto_automatico,LOW); }
  //LUZ PILOTO ELECTROVALVULA_1
  if(digitalRead(electrovalvula_1)==1)

```

```

{ digitalWrite(piloto_electrovalvula_1,HIGH); }
else
{ digitalWrite(piloto_electrovalvula_1,LOW); }
//LUZ PILOTO ELECTROVALVULA_2
if(digitalRead(electrovalvula_2)==1)
{ digitalWrite(piloto_electrovalvula_2,HIGH); }
else
{ digitalWrite(piloto_electrovalvula_2,LOW); }
//LUZ PILOTO ELECTROVALVULA_3
if(digitalRead(electrovalvula_3)==1)
{ digitalWrite(piloto_electrovalvula_3,HIGH); }
else
{ digitalWrite(piloto_electrovalvula_3,LOW); }
//LUZ PILOTO MOTOR_1
if(digitalRead(selector_motor_1)==0)
{ digitalWrite(piloto_bomba_1,HIGH); }
else
{ digitalWrite(piloto_bomba_1,LOW); }
//LUZ PILOTO MOTOR_2
if(digitalRead(selector_motor_2)==0)
{ digitalWrite(piloto_bomba_2,HIGH); }
else
{ digitalWrite(piloto_bomba_2,LOW); }
//LUZ PILOTO PARO EMERGENCA
if(digitalRead(paro_emergencia)==1)
{ digitalWrite(piloto_pause,HIGH); }
else
{ digitalWrite(piloto_pause,LOW); }

} //fin luces piloto

```


ANEXO I

Código de Programación de la Interfaz Hombre- Máquina.

La programación se realizó en Python que es un lenguaje abierto, mediante el uso de librería para establecer la comunicación, creación de gráfico, construcción de ventanas y botones, etc.

```
import tkinter as tk
from tkinter import * #interfaz grafica
from tkinter import ttk
from tkinter import messagebox
from tkinter import filedialog
import numpy as np # # Importamos numpy como el alias np.
import comserial
import matplotlib as mpl
import numpy as np
import matplotlib
import datetime
import os
import logging
import shutil
import matplotlib.pyplot as plt
matplotlib.use("TkAgg")
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg, NavigationToolbar2Tk
from matplotlib.figure import Figure
import sys

valortanque1 = 0
valortanque2 = 0
parar = False
TITULOS = ("Sans", 26)
SUBTITULOS = ("Sans", 16)
LARGE_FONT = ("Sans", 12)
pagina = "
def obtenerdatoserial():
    if comserial.estadocomunicacion():
        if comserial.recibirdatos():
            app.after(300, app.frames[pagina].dibujar)
            app.after(20, obtenerdatoserial)
class aplicacionTanques(tk.Tk):

    def __init__(self, *args, **kwargs):

        tk.Tk.__init__(self, *args, **kwargs)
        #self['screenName']="ECOAGUA SELLO.ico"
        container = tk.Frame(self)
        f = open("log.txt","w+b")
        f.close()
        container.pack(side="top", fill="both", expand = True)

        container.grid_rowconfigure(0, weight=1)
        container.grid_columnconfigure(0, weight=1)

        self.frames = { }

        for F in (bienvenidos, comunicacionSerial, HMI, eventos, graficasanterior_tanque2, graficasanterior, alarmas,
graficas, programacion, graficas_tanque2):

            frame = F(container, self)
```

```

        self.frames[F] = frame

        frame.grid(row=0, column=0, sticky="nsew")

    self.show_frame(bienvenidos)

def show_frame(self, cont):
    global pagina
    pagina = cont
    frame = self.frames[cont]
    frame.tkraise()

def dibujar(self):
    pass

class bienvenidos(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent, width=300, height=300, relief='raised', borderwidth=5)
        label = tk.Label(self, text="Bienvenidos", font=SUBTITULOS)
        label.grid(columnspan = 2)
        imagen = PhotoImage(file='ECOAGUA SELLO.gif')
        imagenbienvenidos = tk.Label(self, image=imagen)
        imagenbienvenidos.image = imagen
        imagenbienvenidos.grid(columnspan=2)
        button = tk.Button(self, text="Ingresar",
                           command=lambda: controller.show_frame(comunicacionSerial))
        button.grid(row=2, column=0)

        button2 = tk.Button(self, text="Cerrar",
                            command=lambda: self.destroy())
        button2.grid(row=2, column=1)
        global pagina
        pagina = bienvenidos
    def dibujar(self):
        pass

class comunicacionSerial(tk.Frame):
    combo = ""
    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent, width=300, height=300) #Paracambiar tamaño
        self.label = tk.Label(self, text="Seleccione el puerto de comunicacion", font=SUBTITULOS)
        self.label.grid(columnspan=2)
        so = ""
        if sys.platform == 'win32':
            print("Corriendo dispositivo en Windows")
            so = 'win32'
        elif sys.platform == 'linux':
            print("Corriendo dispositivo en Linux")
            so = 'linux'
        elif sys.platform == 'darwin':
            print("Corriendo en un dispositivo Mac")
            so = 'darwin'
        self.combo = ttk.Combobox(self)

        if so == 'linux':

```

```

self.combo["values"] = ['/dev/ttyACM0', '/dev/ttyACM1', '/dev/ttyUSB0', '/dev/ttyUSB1']

elif so == 'win32':
    self.combo["values"] = ["COM1", "COM2", "COM3", "COM4", "COM5", "COM6", "COM7", "COM8"]
self.combo.grid(columnspan=2)
self.combo.bind("<<ComboBoxSelected>>", self.selection_changed)

self.button1 = Button(self, text="Ingresar",
                      command=lambda: controller.show_frame(HMI))
self.button1.grid(row=2, column=0)
self.button1["state"] = 'disabled'
self.button2 = Button(self, text="Cerrar",
                      command=lambda: self.destroy())
self.button2.grid(row=2, column=1)
global pagina
pagina = comunicacionSerial

def selection_changed(self, event):
    if comserial.iniciarserial(self.combo.get(),9600):
        self.button1["state"] = 'active'
        messagebox.showinfo("Éxito","Comunicacion establecida correctamente")
    else:
        messagebox.showerror("Error comunicacion", "No se puede conectar en ese puerto, verifique que sea el puerto correcto")
def dibujar(self):
    pass

class HMI(tk.Frame):

def __init__(self, parent, controller):
    tk.Frame.__init__(self, parent, width=1024, height=768)
    imagen = PhotoImage(file='portada2.png')
    imagenbienvenidos = tk.Label(self, image=imagen)
    imagenbienvenidos.image = imagen
    imagenbienvenidos.place(x=0, y=0)
    label = tk.Label(self, text="Sistema de bombeo de Tanques", font=TITULOS, bg="gray79")
    label.place(x=280, y=20)
    self.tanque1 = ttk.Progressbar(self, orient=VERTICAL, length=100, mode='determinate', maximum=1151)
    self.tanque1.place(x=85, y=175, height=230, width=175)
    self.tanque2 = ttk.Progressbar(self, orient=VERTICAL, length=100, mode='determinate', maximum=1151)
    self.tanque2.place(x=750, y=212, height=230, width=175)
    label1 = tk.Label(self, text="Hora del sistema", font=SUBTITULOS, bg="gray79")
    label1.place(x=275, y=600)
    self.tiempoactual = tk.Label(self, text="", font=LARGE_FONT , bg="gray79")
    self.tiempoactual.place(x=275, y=650)
    #self.ev1indicador = tk.Label(self, text="EV1", font=LARGE_FONT , bg="gray79")
    #self.ev1indicador.place(x=40, y=40)
    self.ev1color = tk.Label(self, text="EV1", font=LARGE_FONT , bg="gray79", foreground='white')
    self.ev1color.place(x=90, y=45)
    #self.ev2indicador = tk.Label(self, text="EV2", font=LARGE_FONT , bg="gray79")
    #self.ev2indicador.place(x=495, y=600)
    #Tanque 1
    self.tanque_1indicador = tk.Label(self, text="Tanque 1", font=LARGE_FONT , bg="gray79")
    self.tanque_1indicador.place(x=5, y=220)
    self.tanque_1litros = tk.Label(self, text="0 Lts", font=LARGE_FONT , bg="gray79")
    self.tanque_1litros.place(x=5, y=240)
    self.tanque_1porcentaje = tk.Label(self, text="0 (%)", font=LARGE_FONT , bg="gray79")
    self.tanque_1porcentaje.place(x=144, y=282)
    self.tanque_1metros = tk.Label(self, text="0 Mts", font=LARGE_FONT , bg="gray79")

```

```

self.tanque_1metros.place(x=5, y=280)
self.imagen2 = PhotoImage(file='ECOAGUA SELLO.gif')
self.imagenbienvenidos2 = tk.Label(self, image=self.imagen2)
self.imagenbienvenidos2.place(x=5, y = 600)
self.razonsocial = tk.Label(self, text="TRATAMIENTO Y ENVASADO DE AGUA PARA CONSUMO
HUMANO ECOAGUA...", font=LARGE_FONT , bg="gray79")
self.razonsocial.place(x=5, y=700)
#Tanque 2
self.tanque_2indicador = tk.Label(self, text="Tanque 2", font=LARGE_FONT , bg="gray79")
self.tanque_2indicador.place(x=930, y=280)
self.tanque_2litros = tk.Label(self, text="=0 Lts", font=LARGE_FONT , bg="gray79")
self.tanque_2litros.place(x=930, y=300)
self.tanque_2porcentaje = tk.Label(self, text=" 0 (%)", font=LARGE_FONT , bg="gray79")
self.tanque_2porcentaje.place(x=793, y=324)
self.tanque_2metros = tk.Label(self, text="0 Mts", font=LARGE_FONT , bg="gray79")
self.tanque_2metros.place(x=930, y=340)
self.ev2color = tk.Label(self, text="EV2", font=LARGE_FONT , bg="gray79", foreground='white')
self.ev2color.place(x=404, y=316)
#self.ev3indicador = tk.Label(self, text="EV3", font=LARGE_FONT , bg="gray79")
#self.ev3indicador.place(x=540, y=600)
self.ev3color = tk.Label(self, text="EV3", font=LARGE_FONT , bg="gray79", foreground='white')
self.ev3color.place(x=768, y=92)
#self.mt1indicador = tk.Label(self, text="MT1", font=LARGE_FONT , bg="gray79")
#self.mt1indicador.place(x=450, y=650)
self.mt1color = tk.Label(self, text="MT1", font=LARGE_FONT , bg="gray79", foreground='white')
self.mt1color.place(x=498, y=442)
#self.mt2indicador = tk.Label(self, text="MT2", font=LARGE_FONT , bg="gray79")
#self.mt2indicador.place(x=495, y=650)
self.mt2color = tk.Label(self, text="MT2", font=LARGE_FONT , bg="gray79", foreground='white')
self.mt2color.place(x=573, y=279)
self.errorindicador = tk.Label(self, text="Alarma", font=LARGE_FONT , bg="gray79")
self.errorindicador.place(x=456, y=75)
self.errorcolor = tk.Label(self, text="Err", font=LARGE_FONT , bg="gray79", foreground='red')
self.errorcolor.place(x=520, y=75)
self.errorvalor = tk.Label(self, text="", font=LARGE_FONT , bg="gray79")
self.errorvalor.place(x=456, y=100)
self.procesoindicador = tk.Label(self, text="Proceso", font=LARGE_FONT , bg="gray79")
self.procesoindicador.place(x=404, y=649)
self.procesovalor = tk.Label(self, text="", font=LARGE_FONT , bg="gray79")
self.procesovalor.place(x=404, y=669)
botonGraficasT1 = tk.Button(self, text="Graficas Tanque 1",
command=lambda: controller.show_frame(graficas))
botonGraficasT1.place(x=725, y=620)
botonGraficasT2 = tk.Button(self, text="Graficas Tanque 2",
command=lambda: controller.show_frame(graficas_tanque2))
botonGraficasT2.place(x=725, y=645)
botonGraficasantT1 = tk.Button(self, text="Graficas Anteriores T1",
command=lambda: controller.show_frame(graficasanterior))
botonGraficasantT1.place(x=725, y=670)

botonEventos = tk.Button(self, text="Eventos",
command=lambda: controller.show_frame(eventos))
botonEventos.place(x=875, y=595)
botonAlarmas = tk.Button(self, text="Alarmas",
command=lambda: controller.show_frame(alarmas))
botonAlarmas.place(x=875, y=620)
botoncerrar = tk.Button(self, text="Cerrar",
command=lambda: self.destroy())
botoncerrar.place(x=875, y=645)
botonGraficasantT2 = tk.Button(self, text="Graficas Anteriores T2",
command=lambda: controller.show_frame(graficasanterior_tanque2))

```

```
botonGraficasantT2.place(x=875, y=670)
```

```
def dibujar(self):
    self.tanque1['value'] = comserial.tanque_1metros
    self.tanque2['value'] = comserial.tanque_2metros
    self.tiempoactual['text'] = str(comserial.horaactual) + ':' + str(comserial.minutoactual)
    self.tanque_1metros['text'] = str(comserial.tanque_1metros/1000) + " Mts"
    self.tanque_1litros['text'] = str(comserial.tanque_1litros) + " lts"
    self.tanque_1porcentaje['text'] = str(comserial.tanque_1porcentaje) + " (%)"
    self.tanque_2metros['text'] = str(comserial.tanque_2metros/1000) + " Mts"
    self.tanque_2litros['text'] = str(comserial.tanque_2litros) + " lts"
    self.tanque_2porcentaje['text'] = str(comserial.tanque_2porcentaje) + " (%)"
    if comserial.electrovalvula_1 == 0:
        self.ev1color['foreground'] = 'white'
        self.ev1color['bg'] = 'white'
    else:
        self.ev1color['foreground'] = 'green'
        self.ev1color['bg'] = 'green'
    if comserial.electrovalvula_2 == 0:
        self.ev2color['foreground'] = 'white'
        self.ev2color['bg'] = 'white'
    else:
        self.ev2color['foreground'] = 'green'
        self.ev2color['bg'] = 'green'
    if comserial.electrovalvula_3 == 0:
        self.ev3color['foreground'] = 'white'
        self.ev3color['bg'] = 'white'
    else:
        self.ev3color['foreground'] = 'green'
        self.ev3color['bg'] = 'green'
    if comserial.motor_1 == 0:
        self.mt1color['foreground'] = 'white'
        self.mt1color['bg'] = 'white'
    else:
        self.mt1color['foreground'] = 'green'
        self.mt1color['bg'] = 'green'
    if comserial.motor_2 == 0:
        self.mt2color['foreground'] = 'white'
        self.mt2color['bg'] = 'white'
    else:
        self.mt2color['foreground'] = 'green'
        self.mt2color['bg'] = 'green'
    if comserial.error == 0:
        self.errorcolor['foreground'] = 'white'
        self.errorcolor['bg'] = 'white'
        self.errorvalor['text'] = ""
    else:
        self.errorcolor['foreground'] = 'red'
        self.errorcolor['bg'] = 'red'
        if comserial.error == 1:
            self.errorvalor['text'] = "Nivel crítico Tanque 1"
        elif comserial.error == 2:
            self.errorvalor['text'] = "Nivel crítico Tanque 2"
        elif comserial.error == 3:
            self.errorvalor['text'] = "Nivel crítico Tanque 1 y 2"
        elif comserial.error == 4:
            self.errorvalor['text'] = "Falla Sensor 1"
        elif comserial.error == 5:
            self.errorvalor['text'] = "Falla Sensor 2"
        elif comserial.error == 6:
            self.errorvalor['text'] = "Falla Tanque 1 y Tanque 2"
```

```

        elif comserial.error == 7:
            self.errorvalor['text'] = "Presionado paro de emergencias"
    if comserial.operacion == 0:
        self.procesovalor['text'] = "Vaciado Tanques"
    elif comserial.operacion == 1:
        self.procesovalor['text'] = "Producción Automática"
    elif comserial.operacion == 2:
        self.procesovalor['text'] = "Tiempo de Reposo"
    elif comserial.operacion == 3:
        self.procesovalor['text'] = "Retrolavado"
    elif comserial.operacion == 4:
        self.procesovalor['text'] = "Sistema Detenido"

class eventos(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Historico eventos", font=TITULOS)
        label.grid(columnspan=4)
        infile = "registro.log"
        f = open(infile,"r")
        alarmasgeneradas = []
        g = f.readlines()
        for line in g:
            if "INFO" in line:
                alarmasgeneradas.append(line)
        f.close()
        self.t = tk.Text(self)
        self.t.insert(0.0,alarmasgeneradas)
        self.t.grid(columnspan=3)
        self.s = tk.Scrollbar(self, orient=VERTICAL, command=self.t.yview)
        self.t['yscrollcommand'] = self.s.set
        self.s.grid(column=3, row=1, sticky=(N,S))
        button1 = tk.Button(self, text="Volver",
            command=lambda: controller.show_frame(HMI))
        button1.grid(row=2, column=0)
        button3 = tk.Button(self, text="Refrescar",
            command=lambda: self.refrescar())
        button3.grid(row=2, column=1)
        button2 = tk.Button(self, text="Cerrar",
            command=lambda: self.destroy())
        button2.grid(row=2, column=2)
    def dibujar(self):
        pass
    def refrescar(self):
        infile = "registro.log"
        f = open(infile,"r")
        alarmasgeneradas = []
        g = f.readlines()
        for line in g:
            if "INFO" in line:
                alarmasgeneradas.append(line)
        f.close()
        self.t.delete('1.0',END)
        self.t.insert(0.0,alarmasgeneradas)

class alarmas(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Historico errores", font=TITULOS)

```

```

label.grid(columnspan=4)
infile = "registro.log"
alarmasgeneradas = []
f = open(infile,"r")
g = f.readlines()
f.close()
for line in g:
    if "ERROR " in line:
        alarmasgeneradas.append(line)

self.t = tk.Text(self)
self.t.delete('1.0',END)
self.t.insert(0.0,alarmasgeneradas)
self.t.grid(columnspan=3)
self.s = tk.Scrollbar(self, orient=VERTICAL, command=self.t.yview)
self.t['yscrollcommand'] = self.s.set
self.s.grid(column=3, row=1, sticky=(N,S))
button1 = tk.Button(self, text="Volver",
                    command=lambda: controller.show_frame(HMI))
button1.grid(row=2, column=0)
button3 = tk.Button(self, text="Refrescar",
                    command=lambda: self.refrescar())
button3.grid(row=2, column=1)
button2 = tk.Button(self, text="Cerrar",
                    command=lambda: self.destroy())
button2.grid(row=2, column=2)
def dibujar(self):
    pass
def refrescar(self):
    infile = "registro.log"
    alarmasgeneradas = []
    f = open(infile,"r")
    g = f.readlines()
    f.close()
    for line in g:
        if "ERROR " in line:
            alarmasgeneradas.append(line)
    self.t.delete('1.0',END)
    self.t.insert(0.0,alarmasgeneradas)

```

```

class graficas(tk.Frame):

```

```

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Gráficas Tanque 1", font=TITULOS)
        label.pack(pady=10,padx=10)
        self.f = Figure(figsize=(5,5), dpi=100)
        self.a = self.f.add_subplot(511)
        self.a.set_ylabel('Tanque 1(mm)')
        self.a.set_ylim(0,1150)
        self.b = self.f.add_subplot(512, sharex = self.a)
        self.b.set_ylabel('EV1')
        self.b.set_ylim(0,1.3)
        self.c = self.f.add_subplot(513, sharex = self.b)
        self.c.set_ylabel('EV2')
        self.c.set_ylim(0,1.3)
        self.d = self.f.add_subplot(514, sharex = self.c)
        self.d.set_ylabel('EV3')
        self.d.set_ylim(0,1.3)
        self.h = self.f.add_subplot(515, sharex = self.d, sharey=self.d)

```

```

self.h.set_ylabel('Alarma')
self.h.set_ylim(0,1.3)
self.f.subplots_adjust(hspace=0)
plt.setp([self.a.get_xticklabels() for a in self.f.axes[:-1]], visible=False)
self.canvas = FigureCanvasTkAgg(self.f, self)
self.canvas.draw()
self.canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)

toolbar = NavigationToolbar2Tk(self.canvas, self)
toolbar.update()
self.canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)

button1 = tk.Button(self, text="Regresar",
                    command=lambda: controller.show_frame(HMI))
button1.pack()
self.button3 = tk.Button(self, text="Parar",
                        command=self.paragrafica)
self.button3.pack()
self.button4 = tk.Button(self, text="Guardar",
                        command=self.guardardatos)
self.button4.pack()

button2 = tk.Button(self, text="Cerrar",
                    command=lambda: self.destroy())
button2.pack()
global pagina
pagina = graficas
def dibujar(self):
    global parar
    if parar == False:
        minimox = comserial.historicotiempo[0]
        maximox = comserial.historicotiempo[len(comserial.historicotiempo)-1]
        self.a.set_xlim(minimox, maximox)
        self.b.set_xlim(minimox, maximox)
        self.c.set_xlim(minimox, maximox)
        self.d.set_xlim(minimox, maximox)
        self.a.plot(comserial.historicotiempo, comserial.historiconiveltanque_1, 'b')
        self.b.plot(comserial.historicotiempo, comserial.historicoev1, 'm')
        self.c.plot(comserial.historicotiempo, comserial.historicoev2, 'm')
        self.d.plot(comserial.historicotiempo, comserial.historicoev3, 'm')
        self.h.plot(comserial.historicotiempo, comserial.historicoalarmatanque1, 'm')
        self.canvas.draw()
def paragrafica(self):
    global parar
    parar = not parar
    if parar:
        self.button3['text']="Iniciar"
    else:
        self.button3['text']="Parar"
def guardardatos(self):
    global parar
    parar = True
    f = filedialog.asksaveasfile(mode='w+', title = "Seleccione el Archivo para guardar", filetypes = (("Archivo
texto", "*.txt"),("all files", "*.*")))
    if f is None: # asksaveasfile return `None` if dialog closed with "cancel".
        return
    rutadirectorio = f.name
    f.close()
    shutil.copyfile("log.txt",rutadirectorio)
    f = open(rutadirectorio,"a")
    parar = False

```



```

self.vectortotal = []
for i in range(0,len(comserial.historicoev1)):
    vectorparcial = ""
    vectorparcial+=str(comserial.historicoev1[i])
    vectorparcial+=","
    vectorparcial+=str(comserial.historicoev2[i])
    vectorparcial+=","
    vectorparcial+=str(comserial.historicoev3[i])
    vectorparcial+=","
    vectorparcial+=str(comserial.historiconiveltanque_1[i])
    vectorparcial+=","
    vectorparcial+=str(comserial.historiconiveltanque_2[i])
    vectorparcial+=","
    vectorparcial+=str(comserial.historicoalarmatanque1[i])
    vectorparcial+=","
    vectorparcial+=str(comserial.historicoalarmatanque2[i])
    vectorparcial+=","
    vectorparcial+=comserial.historicotiempo[i].isoformat()
    f.write(vectorparcial)
    f.write("\n")
messagebox.showinfo("Éxito", "Archivo guardado en "+ f.name)
f.close()

```

```

class graficas_tanque2(tk.Frame):

```

```

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Gráficas Tanque 2", font=LARGE_FONT , bg="gray79")
        label.pack(pady=10,padx=10)
        self.f = Figure(figsize=(5,5), dpi=100)
        self.a = self.f.add_subplot(311)
        self.a.set_ylabel("Tanque 2(mm)")
        self.a.set_ylim(0,1150)
        self.b = self.f.add_subplot(312, sharex = self.a)
        self.b.set_ylabel('EV3')
        self.b.set_ylim(0,1.3)
        self.d = self.f.add_subplot(313, sharex = self.b, sharey = self.b)
        self.d.set_ylabel('Alarma')
        self.d.set_ylim(0,1.3)
        self.f.subplots_adjust(hspace=0)
        plt.setp([self.a.get_xticklabels() for a in self.f.axes[:-1]], visible=False)
        self.canvas = FigureCanvasTkAgg(self.f, self)
        self.canvas.draw()
        self.canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)

        toolbar = NavigationToolbar2Tk(self.canvas, self)
        toolbar.update()
        self.canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)

        button1 = tk.Button(self, text="Regresar",
            command=lambda: controller.show_frame(HMI))
        button1.pack()
        self.button3 = tk.Button(self, text="Parar",
            command=self.paragrafica)
        self.button3.pack()
        self.button4 = tk.Button(self, text="Guardar",
            command=self.guardardatos)
        self.button4.pack()

        button2 = tk.Button(self, text="Cerrar",

```

```

        command=lambda: self.destroy()
    boton2.pack()
def dibujar(self):
    global parar
    if parar == False:
        minimox = comserial.historicotiempo[0]
        maximox = comserial.historicotiempo[len(comserial.historicotiempo)-1]
        self.a.set_xlim(minimox, maximox)
        self.b.set_xlim(minimox, maximox)
        self.d.set_xlim(minimox, maximox)
        self.a.plot(comserial.historicotiempo, comserial.historiconiveltanque_2, 'b')
        self.b.plot(comserial.historicotiempo, comserial.historicoev3, 'm')
        self.d.plot(comserial.historicotiempo, comserial.historicoalarmatanque2, 'm')
        self.canvas.draw()
def paragrafica(self):
    global parar
    parar = not parar
    if parar:
        self.button3['text']="Iniciar"
    else:
        self.button3['text']="Parar"
def guardardatos(self):
    global parar
    parar = True
    f = filedialog.asksaveasfile(mode='w+', title = "Seleccione el Archivo para guardar", filetypes = (("Archivo
texto", "*.txt"),("all files", "*.*")))

    parar = False
    if f is None: # asksaveasfile return `None` if dialog closed with "cancel".
        return
    rutadirectorio = f.name
    f.close()
    shutil.copyfile("log.txt",rutadirectorio)
    f = open(rutadirectorio,"a")
    self.vectortotal = []
    for i in range(0,len(comserial.historicoev1)):
        vectorparcial = ""
        vectorparcial+=str(comserial.historicoev1[i])
        vectorparcial+=", "
        vectorparcial+=str(comserial.historicoev2[i])
        vectorparcial+=", "
        vectorparcial+=str(comserial.historicoev3[i])
        vectorparcial+=", "
        vectorparcial+=str(comserial.historiconiveltanque_1[i])
        vectorparcial+=", "
        vectorparcial+=str(comserial.historiconiveltanque_2[i])
        vectorparcial+=", "
        vectorparcial+=str(comserial.historicoalarmatanque1[i])
        vectorparcial+=", "
        vectorparcial+=str(comserial.historicoalarmatanque2[i])
        vectorparcial+=", "
        vectorparcial+=comserial.historicotiempo[i].isoformat()
        f.write(vectorparcial)
        f.write("\n")
    #np.savetxt(f.name,self.vectortotal,delimiter=",")
    messagebox.showinfo("Éxito", "Archivo guardado en "+ f.name)
    f.close()

class graficasanterior(tk.Frame):

```

```

def __init__(self, parent, controller):
    tk.Frame.__init__(self, parent)
    label = tk.Label(self, text="Gráficas Anterior Tanque 1", font=LARGE_FONT , bg="gray79")
    label.pack(pady=10,padx=10)
    self.f = Figure(figsize=(5,5), dpi=100)
    self.a = self.f.add_subplot(511)
    self.a.set_ylim(0,1150)
    self.b = self.f.add_subplot(512, sharex = self.a)
    self.b.set_ylabel('EV1')
    self.b.set_ylim(0,1.3)
    self.c = self.f.add_subplot(513, sharex = self.b)
    self.c.set_ylabel('EV2')
    self.c.set_ylim(0,1.3)
    self.d = self.f.add_subplot(514, sharex = self.c)
    self.d.set_ylabel('EV3')
    self.d.set_ylim(0,1.3)
    self.h = self.f.add_subplot(515, sharex = self.d, sharey=self.d)
    self.h.set_ylabel('Alarma')
    self.h.set_ylim(0,1.3)
    self.f.subplots_adjust(hspace=0)
    plt.setp([self.a.get_xticklabels() for a in self.f.axes[:-1]], visible=False)
    self.canvas = FigureCanvasTkAgg(self.f, self)
    self.canvas.draw()
    self.canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)

    toolbar = NavigationToolbar2Tk(self.canvas, self)
    toolbar.update()
    self.canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)

    button1 = tk.Button(self, text="Regresar",
                        command=lambda: controller.show_frame(HMI))
    button1.pack()

    button1 = tk.Button(self, text="Abrir",
                        command=lambda: self.dibujar2())
    button1.pack()
    button2 = tk.Button(self, text="Cerrar",
                        command=lambda: self.destroy())
    button2.pack()
    self.hev1=[]
    self.hev2=[]
    self.hev3=[]
    self.hnivektanque1=[]
    self.halarmast1=[]
    self.htiempo=[]
def dibujar(self):
    pass
def dibujar2(self):
    archivoentrada = filedialog.askopenfilename(initialdir = os.getcwd(),title = "Seleccione el archivo donde guardo
las graficas",filetypes = (("Archivos de texto","*.txt"),("all files","*.*")))
    if archivoentrada == "":
        messagebox.showerror("Error al abrir el archivo", "Seleccione un archivo correcto")
        return
    f = open(archivoentrada)
    g = f.readlines()
    for line in g:
        vectortemporal = []
        vectortemporal = line.strip("\n").split(",")

        self.hev1.append(float(vectortemporal[0]))
        self.hev2.append(float(vectortemporal[1]))

```

```

self.hev3.append(float(vectortemporal[2]))
self.hniveltanque1.append(float(vectortemporal[3]))
self.halarmast1.append(float(vectortemporal[5]))
self.htiempo.append(datetime.datetime.fromisoformat(vectortemporal[7]))
f.close()
self.a.plot(self.htiempo, self.hniveltanque1, 'b')
self.b.plot(self.htiempo, self.hev1, 'm')
self.c.plot(self.htiempo, self.hev2, 'm')
self.d.plot(self.htiempo, self.hev3, 'm')
self.h.plot(self.htiempo, self.halarmast1, 'm')
self.canvas.draw()
self.hev1 = []
self.hev2 = []
self.hev3 = []
self.hniveltanque1 = []
self.halarmast1 = []
self.htiempo = []

```

```
class graficasanterior_tanque2(tk.Frame):
```

```

def __init__(self, parent, controller):
    tk.Frame.__init__(self, parent)
    label = tk.Label(self, text="Gráficas Tanque 2", font=LARGE_FONT , bg="gray79")
    label.pack(pady=10,padx=10)
    self.f = Figure(figsize=(5,5), dpi=100)
    self.a = self.f.add_subplot(411)
    self.a.set_ylabel('Tanque')
    self.a.set_ylim(0,1150)
    self.b = self.f.add_subplot(412, sharex = self.a)
    self.b.set_ylabel('EV3')
    self.b.set_ylim(0,1.3)
    self.c = self.f.add_subplot(413, sharex = self.b)
    self.c.set_ylabel('Vaciado')
    self.c.set_ylim(0,1.3)
    self.d = self.f.add_subplot(414, sharex = self.c, sharey = self.c)
    self.d.set_ylabel('Alarma')
    self.d.set_ylim(0,1.3)
    self.f.subplots_adjust(hspace=0)
    plt.setp([self.a.get_xticklabels() for a in self.f.axes[:-1]], visible=False)
    self.canvas = FigureCanvasTkAgg(self.f, self)
    self.canvas.draw()
    self.canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)

    toolbar = NavigationToolbar2Tk(self.canvas, self)
    toolbar.update()
    self.canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
    botonAbrir = tk.Button(self, text="Abrir",
        command=lambda: self.dibujar2())
    botonAbrir.pack()
    boton1 = tk.Button(self, text="Regresar",
        command=lambda: controller.show_frame(HMI))
    boton1.pack()
    boton2 = tk.Button(self, text="Cerrar",
        command=lambda: self.destroy())
    boton2.pack()
    self.hev3=[]
    self.hniveltanque2=[]
    self.halarmast2=[]
    self.htiempo=[]

```

```

def dibujar2(self):
    archivoentrada = filedialog.askopenfilename(initialdir = os.getcwd(),title = "Seleccione el archivo donde guardo
las graficas",filetypes = (("Archivos de texto", "*.txt"),("all files", "*.*")))
    if archivoentrada == "":
        messagebox.showerror("Error al abrir el archivo", "Seleccione un archivo correcto")
        return
    f = open(archivoentrada)
    g = f.readlines()
    for line in g:
        vectortemporal = []
        vectortemporal = line.strip("\n").split(",")
        self.hev3.append(float(vectortemporal[2]))
        self.hniveltanque2.append(float(vectortemporal[3]))
        self.halarmast2.append(float(vectortemporal[6]))
        self.htiempo.append(datetime.datetime.fromisoformat(vectortemporal[7]))
    f.close()
    self.a.plot(self.htiempo, self.hniveltanque2, 'b')
    self.b.plot(self.htiempo, self.hev3, 'm')
    self.d.plot(self.htiempo, self.halarmast2, 'm')
    self.canvas.draw()
    self.hev1 = []
    self.hev2 = []
    self.hev3 = []
    self.hniveltanque1 = []
    self.halarmast1 = []
    self.htiempo = []

def dibujar(self):
    pass

class programacion(tk.Frame):

    def __init__(self, parent, controller):
        tk.Frame.__init__(self, parent)
        label = tk.Label(self, text="Programacion", font=LARGE_FONT , bg="gray79")
        label.grid(columnspan=2)
        botonModoAutomatico = tk.Button(self, text="Modo Automatico",
            command=lambda: controller.show_frame(HMI))
        botonModoAutomatico.grid(row=1, column=0)

        botonModoManual = tk.Button(self, text="Modo Manual",
            command=lambda: controller.show_frame(HMI))
        botonModoManual.grid(row=1, column=1)
        botonVolver = tk.Button(self, text="Volver",
            command=lambda: controller.show_frame(HMI))
        botonVolver.grid(row=2, column=0)

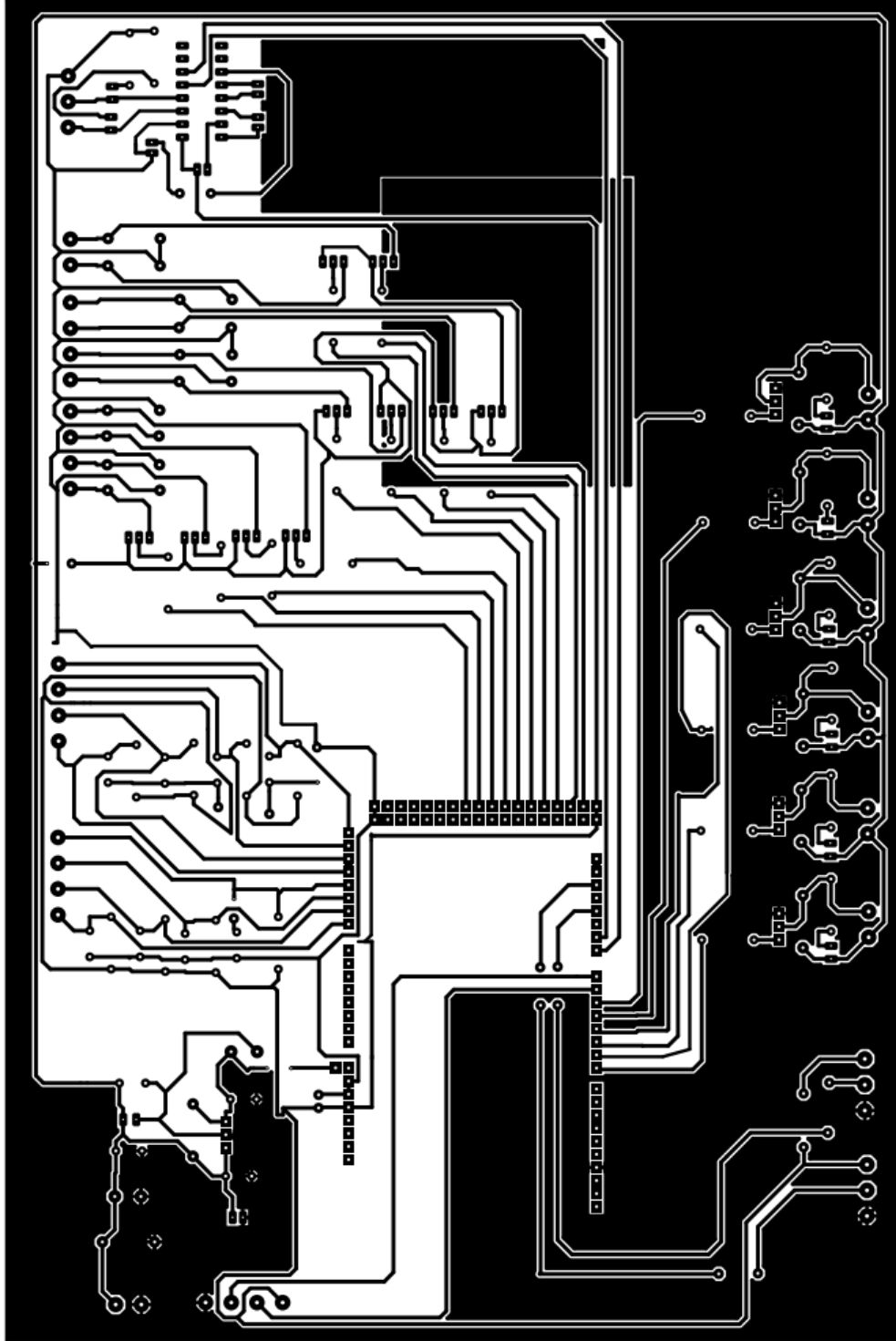
        botonCerrar = tk.Button(self, text="Cerrar",
            command=lambda: self.destroy())
        botonCerrar.grid(row=2, column=1)
    def dibujar(self):
        pass

pagina = comunicacionSerial
app = aplicacionTanques()
app.after(20, obtenerdatoserial)
app.mainloop()

```

ANEXOS ADICIONALES

PISTAS DE LA TARJETA DE CONTROL DEL SISTEMA DE CONTROL Y ALERTA DEL TANQUE PURIFICADOR DE AGUA



PISTAS DE LA TARJETA HMI LOCAL

