



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA
UDELAR

Sistema dosificador de reactivos líquidos para análisis de propiedades fisicoquímicas del agua: Automatización industrial aplicada en análisis de calidad de agua.

Ignacio Miranda
Emiliano Ferreira

Proyecto de grado presentado a la Facultad de Ingeniería de la Universidad de la Repùblica
en cumplimiento parcial de los requerimientos para la obtención del título de Ingeniero
electricista.

Tutores

Florencia Blasina
Nicolás Pérez

Tribunal

Javier Roman
Moisés Knochen
Mariana del Castillo

Montevideo, Uruguay
30 de septiembre de 2025

Agradecimientos

Agradezco profundamente los desafíos que brinda la Facultad de Ingeniería. Me acompaña un espíritu innovador y emprendedor, forjado desde mis inicios en un pequeño taller eléctrico perteneciente a una empresa familiar. Tengo el orgullo de haber culminado mis estudios en la principal institución de ingeniería del país y de haber iniciado mis primeras experiencias laborales en distintas empresas, procurando siempre ser un agente de cambio e innovación. Extiendo mi sincero agradecimiento a nuestros tutores, Florencia y Nicolás, por su orientación, paciencia y por haber estado presentes en cada etapa del proceso, guiando tanto en los aspectos técnicos como en los humanos.

Quiero expresar mi más sincero agradecimiento a todas las personas que, de una forma u otra, formaron parte de mi camino durante estos años de formación como ingeniero. A mi madre, por haber sembrado en mí el valor de la educación desde el hogar, por su ejemplo de constancia y por motivarme siempre a seguir adelante. A mi padre, por estar dispuesto a escucharme, brindarme consejos sinceros y enseñarme, con su ejemplo, cómo encarar con responsabilidad y entereza tanto la vida académica como la laboral.

En lo profesional, agradezco a Pablo Peluffo por haber sido una figura clave en mi primera experiencia laboral en Spymovil. Su forma de transmitir conocimiento y su enfoque práctico me ayudaron a comprender que los grandes desafíos se enfrentan con paciencia, organización y perseverancia.

También quiero agradecer a Stefano Oppido por confiar en mis capacidades y brindarme la oportunidad de asumir desafíos tecnológicos en INDUS. Su apoyo no solo me permitió crecer profesionalmente, sino que también fue clave en el acceso a tecnologías que resultaron esenciales para integrar en este proyecto.

A los docentes del Instituto de Ingeniería Eléctrica (IIE), gracias por su manera comprometida de enseñar, por su disposición a compartir experiencias más allá del aula y por su vocación de formar ingenieros con pensamiento crítico. En especial, al equipo de Electrónica de Potencia, cuyo enfoque práctico y exigente dejó una huella en mi formación.

Guardo con especial aprecio una enseñanza del profesor Luis García, que me acompañó durante toda mi etapa como Técnico y que mantendré presente como profesional: "Les van a pagar por pensar, y mientras más piensen, más les van a pagar". Palabras simples pero potentes, que me enseñaron a valorar el pensamiento como herramienta central del trabajo ingenieril.

También deseo agradecer a Pablo Monzón por haberme recibido personalmente y decirme su tiempo en una conversación que fue muy significativa. Me ayudó a entender

mejor el rol del ingeniero en Uruguay, a reconocer cuáles son mis fortalezas dentro del amplio campo de la ingeniería eléctrica, y me dejó un mensaje de motivación que guardo con gratitud para mis próximos pasos profesionales.

A todos ellos, gracias por acompañarme, por confiar en mí y por ser parte de este recorrido que culmina una etapa pero que, sin duda, abre muchas otras por venir.

Ignacio Miranda

Agradezco el acceso a la facultad de ingeniería de manera gratuita, a los docentes, personas cercanas y a mi familia por su acompañamiento y apoyo constante para alcanzar esta meta.

Emiliano Ferreira

Resumen

El presente proyecto tiene como objetivo principal la implementación de un sistema dosificador de reactivos líquidos, orientado al análisis automático de muestras de agua. Esta iniciativa surge ante la necesidad de automatizar procesos de medición en entornos donde se desea monitorear en tiempo real ciertas propiedades físico-químicas del agua, tales como la concentración de compuestos específicos. Dispositivos de estas características pueden utilizarse en plantas de tratamientos de agua y efluentes industriales.

El empleo de sensores y dosificación automática, representa una alternativa cada vez más utilizada frente a los métodos tradicionales de muestreo, que implican altos costos, necesidad de personal capacitado y transporte hacia laboratorios especializados. Los sistemas automáticos permiten obtener datos continuos con una frecuencia de muestreo mucho mayor, lo cual es especialmente útil para el análisis de series temporales y para la detección temprana de cambios en el entorno.

Por decisión de los tutores, el alcance del proyecto se ha centrado en el diseño e implementación del sistema físico, el desarrollo del sistema de control con su respectiva interfaz de usuario, y la realización de pruebas preliminares utilizando colorantes comerciales. Esta estrategia permite validar el concepto del sistema en condiciones controladas, facilitando una posterior transición hacia aplicaciones reales con muestras y reactivos específicos.

Desde el punto de vista técnico, se diseñó e integró un sistema completo. El sistema físico permite aislar una muestra de agua dentro de una cápsula de medición, utilizando bombas peristálticas calibradas para dosificar los volúmenes necesarios con precisión. Esta cápsula deja pasar la luz emitida por un diodo emisor a través de un cristal de cuarzo y la solución dosificada, cuya intensidad es captada por un fotorreceptor en el otro extremo. Esta configuración posibilita detectar los cambios de color resultantes de la reacción.

Se desarrolló una interfaz de usuario capaz de integrarse a una red LAN privada, permitiendo la comunicación bidireccional con los controladores del sistema en tiempo real. Esta interfaz no solo facilita la supervisión del proceso, sino que también permite la configuración de parámetros y el monitoreo de variables críticas desde un entorno gráfico amigable. Además, se implementó un sistema de control automatizado mediante un PLC y una tarjeta Raspberry Pi. Se integra la lógica de funcionamiento del proceso, coordinando la operación de los distintos actuadores. Para ello, se integraron drivers específicos que permiten el accionamiento preciso de válvulas, bombas y otros elementos del sistema, lo que hace posible la ejecución automática de ciclos de medición, garantizando precisión, repetibilidad y seguridad operativa.

En aplicaciones concretas, el análisis requiere la incorporación de uno o más reactivos que, al mezclarse con la muestra, generan una reacción química que modifica su color. Este cambio cromático es indicativo de la presencia o concentración de determinadas sustancias, y permite inferir propiedades relevantes de la calidad del agua.

A nivel práctico, se ha optado por una estrategia de pruebas que incluye el uso de colorantes domésticos como sustitutos de reactivos reales, con el objetivo de validar el funcionamiento del sistema sin necesidad de manejar compuestos costosos o peligrosos en etapas tempranas del desarrollo.

El equipo a diseñar dispone de dos bombas peristálticas, una de ellas tomará la muestra de agua que se desea estudiar y otra estará encargada de dispensar el reactivo para realizar la mezcla con el agua. La masa de los líquidos involucrados será medida con balanzas analíticas durante la calibración del equipo, para tener el factor de escala de los pasos de dosificación. Finalmente, la muestra será analizada con un sensor óptico que realizará una medida de colorimetría.

Los principales objetivos planteados se pueden resumir en la aislación de la muestra por medio del control de bombas peristálticas y la dosificación del químico reactivo a incluir en la solución. También registrar las medidas para poder comparar con una lectura sin reactivo el cambio de color y construir un dispositivo fácilmente controlable por un operario sin amplios conocimientos técnicos.

Prefacio

Consideramos que ser ingeniero eléctrico implica la combinación de varias competencias, el ingeniero eléctrico no solo maneja las temáticas teóricas incluidas en el programa de la carrera, sino que requiere de habilidades como la selección y adquisición de los equipos requeridos para realizar una tarea, el correcto manejo de un presupuesto, la habilidad de programar y ensamblar manualmente equipos o componentes y la solución de posibles inconvenientes que puedan surgir en el proceso de ejecución.

Este proyecto no solo cubre todas estas competencias, sino que también nos permitió ejercer un nivel alto de creatividad. Ya que como objetivo se planteó solamente la construcción de un dispositivo que cumpliera ciertas funciones, estuvo en nuestras manos el campo de aplicación de este dispositivo, los componentes que formaron al equipo, los programas y lenguajes de programación utilizados y las funciones particulares que realizaría el prototipo.

También identificamos que hay un campo esencial en la ingeniería eléctrica orientada a la potencia que no es ampliamente cubierto en la carrera, este es la conexión, programación y manejo de dispositivos de control. En redes de potencia, tanto interruptores, disyuntores, motores y gran parte de los equipos modernos, deben ser operados con dispositivos de control programables. Entre estos dispositivos podemos resaltar los relés de protección que establecen las funciones de protección utilizadas en la red, los IED (Intelligent Electronic Devices) controladores generalmente utilizados en alta tensión y los PLC. El PLC es un dispositivo de control aplicado en el campo industrial que proporciona un nivel de flexibilidad superior al de los dispositivos previamente mencionados. Equipos como relés de protección son programables mediante bloques o variables previamente definidos por el fabricante, mientras que el PLC permite programar funciones desde cero, permitiendo definir la función de entradas y salidas varias para cumplir con las funcionalidades requeridas.

La posibilidad de profundizar en este tipo de equipos esenciales en la industria, junto con la oportunidad de crear un equipo físico que nos permitió aplicar habilidades de diseño y gestión que serán utilizadas en un futuro en el ámbito profesional, fueron las principales razones que nos motivaron a seleccionar este proyecto en particular.

Índice general

Agradecimientos	1
Resumen	3
Prefacio	5
1. Introducción	9
1.1. Objetivo y descripción del proyecto	9
1.2. Revisión bibliográfica	10
1.2.1. Sistema de monitoreo de cloro Spymovil-UdelaR	10
1.2.2. Micromac C	13
1.2.3. Analizador automático para la determinación de nutrientes en aguas superficiales	15
1.3. Organización del texto	18
2. Diseño del dispositivo	19
2.1. Operativa del dispositivo	19
2.2. Manual de usuario	22
2.2.1. Modo manual	22
2.2.2. Modo automático	23
2.3. Diagrama del sistema	26
2.4. Alimentación y protecciones	29
2.4.1. Conversores	29
2.4.2. Llave termomagnética	30
2.4.3. Fusible	31
2.5. Dispositivos de control	32
2.5.1. PLC Crevis GL-9972	32
2.5.2. Raspberry PI	33
2.6. Hardware auxiliar y actuadores	35
2.6.1. Bombas peristálticas	35
2.6.2. Driver para motores de paso	37
2.6.3. Electroválvula	38
2.6.4. Relé	38

2.6.5. Módulo de salidas digitales CREVIS GT-226F	39
2.6.6. Sensor óptico	39
2.6.7. Switch de puertos ethernet	42
2.7. Componentes físicos	43
2.7.1. Montaje final	43
2.7.2. Cubeta de cuarzo	43
2.7.3. Cámara para test óptico	44
2.7.4. Recipientes de reactivo y Muestra	45
2.7.5. Placas	45
3. Software para controladores	46
3.1. Ambiente de programación	46
3.1.1. Asociación de variables globales y mapeo de I/O	48
3.1.2. CODESYS Runtime	48
3.2. Esquema de comunicación Modbus TCP	50
3.3. Software en PLC	51
3.4. Software en Raspberry PI	53
4. Operación y calibración del dispositivo	55
4.1. Ensayo de flujo de bombas	55
4.2. Ensayos de funcionamiento	58
4.2.1. Datos relevados	58
5. Conclusiones y trabajos a futuro	62
5.1. Conclusiones de proyecto	62
5.2. Posibles futuras mejoras	63
5.2.1. Modos de operación	63
5.2.2. Visualización	63
5.2.3. Expansión de la red de medida y muestreo	64
5.2.4. Control a distancia	64
5.2.5. Integración en industria o laboratorio	65
A. Apéndice A: Planos	66
B. Apéndice B:Normas y fundamentos	67
B.1. Motores paso a paso	67
B.2. Espectroscopía	68
B.2.1. Espectroscopía UV/Vis	69
B.2.2. Absorción de luz como herramienta analítica	71
B.2.3. Ley de Lambert-Beer	75
B.2.4. Aplicación en Longitud de onda fija	77
B.3. Protocolos de comunicación	81
B.3.1. Protocolo Modbus	81
B.3.2. Protocolo I2C	82

B.4. Resumen Norma IEC 61131-3	85
C. Apéndice C: Memoria de programación	86
C.1. Software en PLC	86
C.1.1. Programa Procesamiento Registros Modbus Drivers Bombas (ST) .	86
C.1.2. Programa Principal (CFC)	87
C.1.3. Programa Copia Salidas	99
C.1.4. Programa Modbus Tcp (ST)	100
C.1.5. Programa Calibración Bombas (ST)	104
C.2. Software en Raspberry Pi	105
C.2.1. Programa I2C COM (ST)	105
C.2.2. Programa Visu (ST)	108
C.2.3. Programa Modbus TCP Master (ST)	108
C.2.4. Programa Modbus TCP Slave (ST)	108
C.2.5. Programa Registro de medidas (ST)	109
Referencias	113

Capítulo 1

Introducción

1.1. Objetivo y descripción del proyecto

Varias industrias requieren de la verificación de propiedades en líquidos mediante la utilización de espectrofotómetros (Apéndice B.2). Generalmente, estos estudios requieren de un laboratorio y personal capacitado que retire muestras desde la fuente, mida las proporciones de reactivo a aplicar en la solución y realice la mezcla dentro de un recipiente adecuado, para luego introducir el mismo dentro del equipo encargado del ensayo óptico. Finalmente, la solución debe ser descartada en un recipiente adecuado y los datos del ensayo son registrados.

Cobra interés el desarrollo de sistemas automatizados que permitan que este proceso sea ejecutado desde una sala de control, agilizando los resultados y minimizando la carga sobre el personal encargado. Muchas de esas medidas se realizan por espectroscopía óptica (Se profundiza sobre este tema en el apéndice B.2) a partir de una reacción química conocida y para ello se necesitan sistemas que realicen una dosificación precisa de reactivos y muestras, generando una reacción de color acorde a la propiedad de la solución que se desee medir.

Las medidas realizadas deberán ser configuradas y visualizadas mediante una interfaz de usuario fácilmente manipulable por el personal de planta encargado de realizar los ensayos.

El principal criterio de éxito en este proyecto, es lograr las funciones básicas de aislar la muestra y dispensar el reactivo acorde a las instrucciones del usuario.

El dispositivo debe guardar las medidas realizadas por el sensor óptico, junto con las características del ensayo. Mediante un menú interactivo el usuario podrá también extraer datos de medidas realizadas por el equipo y configurar parámetros del ensayo.

1.2. Revisión bibliográfica

Presentaremos a continuación ejemplos de proyectos o dispositivos similares que utilizamos como referencia durante la ejecución del proyecto.

1.2.1. Sistema de monitoreo de cloro Spymovil-Udelar

Este sistema fue construido durante un proyecto apoyado por la ANII (Agencia Nacional de Investigación e Innovación).

En el desarrollo del sistema de monitoreo de cloro en línea Spymovil-Udelar, se definieron los siguientes componentes clave:

- Placa de control, que incluye drivers para bombas, y electroválvulas, entrada para caudalímetro para la correcta manipulación de la muestra y los reactivos.
- Módulo de Medida Fotométrica basado en un LED de 515 nm y un fotodetector OPT101, que permite la cuantificación de la reacción colorimétrica.
- Comunicación mediante protocolo RS-485, asegurando una transmisión de datos robusta y confiable.

Este fue el prototipo final, desarrollado mediante una electrónica diseñada a medida para la aplicación.

Sistema implementado por Spymovil-Udelar

Se muestran en la figura 1.1 los elementos del sistema físico y la placa de control donde se controlan (a través de drivers) las bombas peristálticas, se accionan las electro-válvulas y se toma la lectura del sensor de colorimetría.

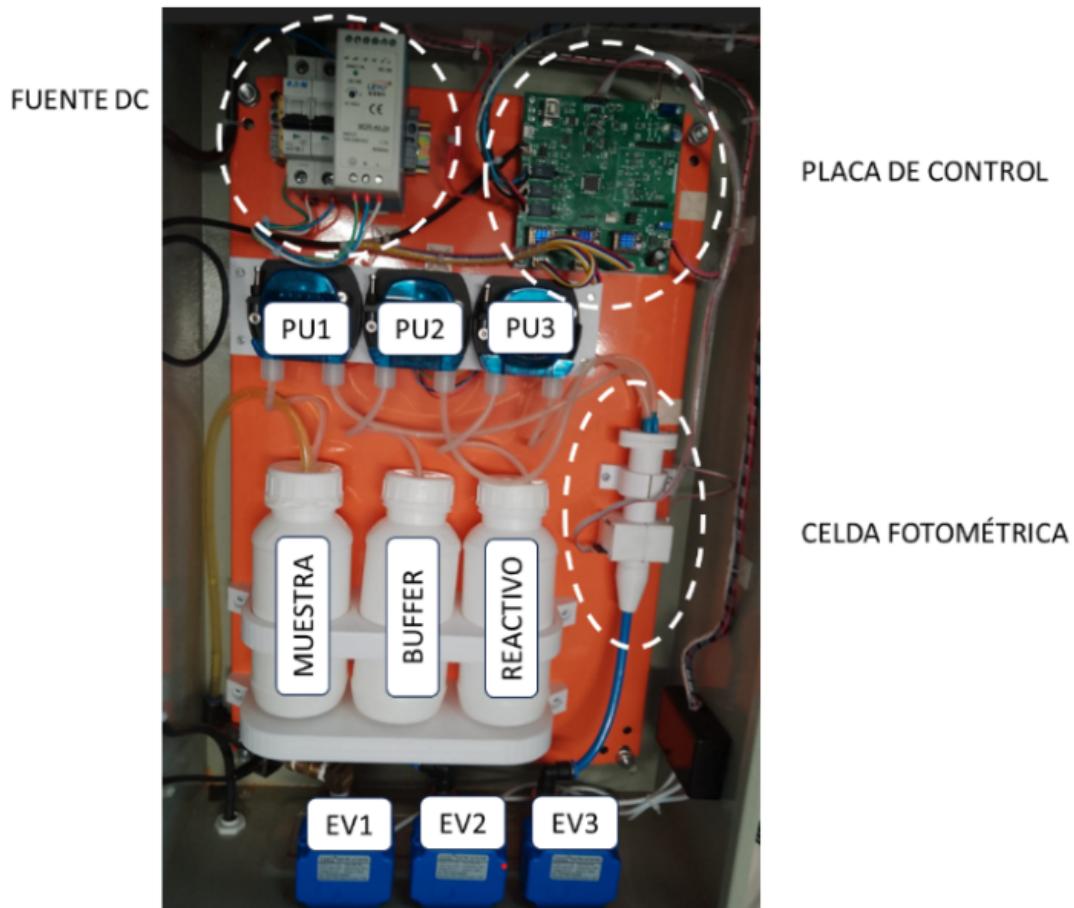


Figura 1.1: Sistema completo implementado por Spymovil-Udelar (EV - electrovalvula, PU - bomba peristáltica)

En esta implementación se utilizan tres electro-válvulas (EV1, EV2, EV3), tres bombas (PU1, PU2, PU3) y tres frascos de líquido (MUESTRA, BUFFER, REACTIVO).

La figura 1.2 muestra un diagrama de los circuitos hidráulicos.

- EV1 Electroválvula de admisión de muestra
- EV2 Electroválvula de descarga de muestra
- EV3 Electroválvula de descarga de celda de medida
- PU1 Dosificación de muestra
- PU2 Dosificación de buffer
- PU3 Dosificación de reactivo

Diagrama del sistema físico Spymovil-Udelar

Se involucra buffer y reactivo debido al objetivo de detección de cloro por DPD. El DPD es un reactivo químico que adquiere un tono rosado ante la presencia de cloro y el buffer es un tercer componente agregado para lograr condiciones óptimas de medición.

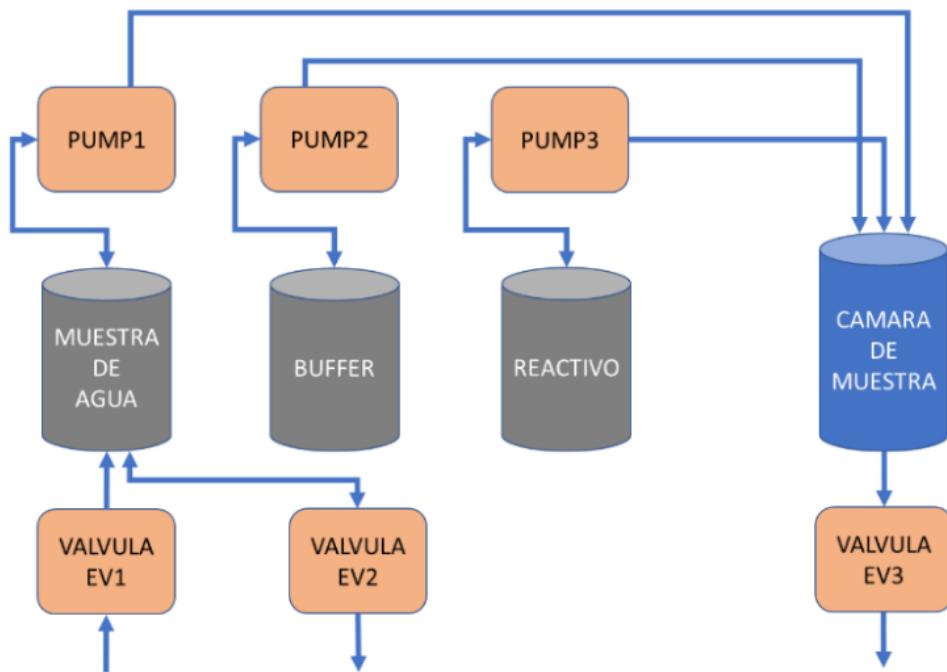


Figura 1.2: Sistema físico en detección de DPD - Spymovil-Udelar

1.2.2. Micromac C

Systea es una empresa italiana especializada en analizadores automáticos. Uno de los equipos que ofrecen en el mercado es el denominado Micromac C (hoja de datos [6], pagina oficial [7]), este equipo cumple con los requerimientos planteados dentro de este proyecto y prueba la aplicabilidad de este tipo de dispositivos en campos industriales.

La función del Micromac C es tomar una muestra del objeto de estudio mediante bombas peristálticas. Luego mediante un sistema de flujo constante patentado (Loop Flow Analysis), aplica reactivos a esta muestra y ejecuta los ensayos ópticos requeridos para verificar sus cualidades. Este dispositivo tiene la capacidad de conectarse a redes industriales con protocolos de comunicación compatibles.

La figura 1.3 muestra, un conjunto de equipos Micromac C instalados en una planta industrial, donde debajo de cada tablero se visualizan los tubos encargados de tomar la muestra y desecharla luego de realizado el ensayo.



Figura 1.3: Equipos Micromac C instalados [8]

La figura 1.4 muestra el interior de uno de los dispositivos. Se puede identificar en esta imagen una separación entre los componentes electrónicos (gabinete superior) y los hidráulicos (gabinete inferior). En nuestro proyecto también intentamos separar los dos procesos en placas individuales, tal como se muestra en el plano 07.

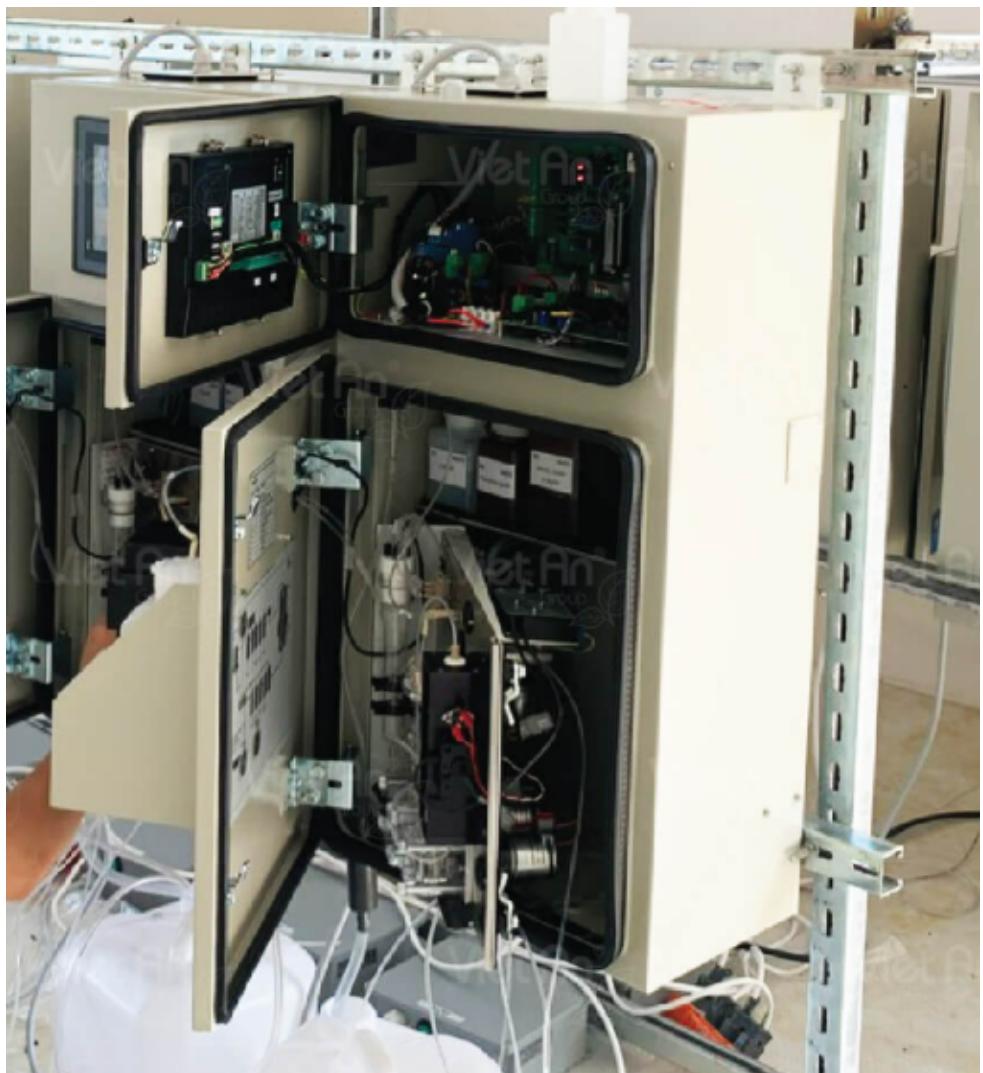


Figura 1.4: Interior de equipos Micromac C [8]

1.2.3. Analizador automático para la determinación de nutrientes en aguas superficiales

Este dispositivo fue desarrollado como proyecto final en maestría de química por Pablo Ismael González Cazzola en el año 2019. Consta del diseño y construcción de un analizador autónomo de campo para la determinación de fósforo y nitrógeno en aguas superficiales.

El analizador construido permite la determinación de fosfato, nitrito y nitrato, está basado en un sistema de flujo pulsado con detección fotométrica y cuenta con comunicación de los resultados mediante mensaje de texto al teléfono móvil del usuario.

Este proyecto se enfocó en el aspecto químico del análisis automático del agua mediante espectrofotometría. Sin embargo, el dispositivo construido cumple con gran cantidad de los objetivos que nos plantemos.

El armado del analizador se basó en un diseño “box in box”; esto es, los componentes están agrupados en cajas estancas (módulos) y estas cajas están dentro de una carcasa también estanca. La Figura 1.5 muestra una vista interior del gabinete principal.



Figura 1.5: Vista interior del gabinete principal [9]

Dentro del gabinete principal se colocaron diferentes módulos:

- Módulo de muestreo: Como se muestra en la figura 1.7, contiene a la bomba solenoide que impulsa al patrón o la muestra; contiene a la válvula solenoide que selecciona si se bombea patrón (solución de referencia con una concentración conocida de la sustancia que se quiere medir) o si se bombea muestra y una segunda válvula solenoide que selecciona si el líquido pasa por la columna de determinación de nitrato o determinación de fosfato.

- Módulo de bombas de reactivos y portador: Este módulo contiene en su interior 3 bombas solenoide (Figura 1.8); una para impulsar el portador (agua desionizada que transporta la muestra y los reactivos a través del sistema), otra para impulsar el reactivo para fosfato y otra para impulsar el reactivo para nitrito/nitrato.
- Módulo de detección: Contiene al sistema de detección (portaceldas, LEDs y fotodioidos, celdas de flujo) y una válvula solenoide que selecciona si el líquido pasa a través del detector de fosfato o a través del detector de nitrito/nitrato.
- Módulo de la electrónica: Contiene todo el resto de la electrónica; la placa Arduino UNO mostrada en la Figura 1.6, el circuito de acondicionamiento de las señales que salen de los detectores, la electrónica de potencia para los actuadores, las baterías, un interruptor y un LED que indica si el analizador está encendido.

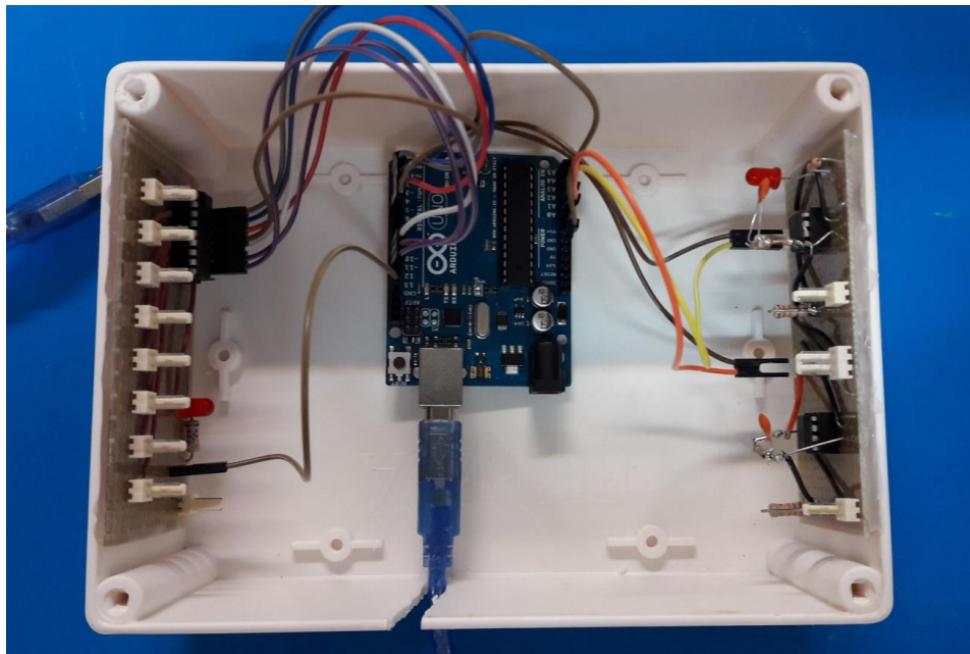


Figura 1.6: Módulo de la electrónica [9]

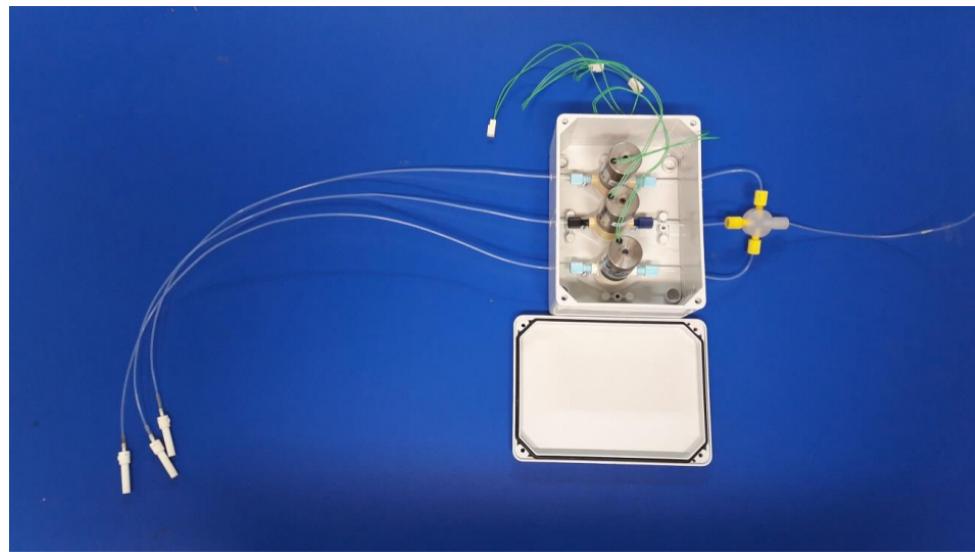


Figura 1.7: Módulo de muestreo [9]



Figura 1.8: Bomba peristáltica utilizada para la toma de agua a muestrear [9]

1.3. Organización del texto

Este documento está dividido en capítulos que cubren todas las temáticas pertinentes tanto a las características y operación del dispositivo final, como el proceso de desarrollo atravesado por los integrantes de este proyecto.

Daremos a continuación un breve resumen de los capítulos que integran a esta tesis y la temática incluida en cada una de estas secciones para introducir al lector a la estructura de este texto:

- Capítulo 1 - Introducción: Se utiliza este capítulo como introducción del objetivo de este proyecto y las razones por las que fue seleccionado.
- Capítulo 2 - Diseño del dispositivo: Presentaremos al prototipo en su versión definitiva, se especificará la estructura, componentes y operativa. También se incluye en este capítulo un manual de usuario que permitirá comprender como controlar cada una de sus funciones y visualizar las medidas tomadas.
- Capítulo 3 - Software para controladores: Se detalla una documentación técnica del software implementado en ambos controladores.
- Capítulo 4 - Calibración del dispositivo: En esta sección presentamos los resultados de los ensayos realizados para el ajuste y la verificación del correcto funcionamiento del equipo.
- Capítulo 5 - Conclusiones y trabajos a futuro: Analizaremos el prototipo obtenido como producto final y las posibles mejoras que podrían realizarse.
- Elementos de apoyo al lector - Luego del capítulo 5 se incluye un índice de cuadros, índice de figuras y la Bibliografía.
- Apéndice A - Contiene los planos que serán referenciados en el texto, estos incluirán todas las conexiones realizadas junto con la numeración de cables y una vista esquemática identificando a cada componente.
- Apéndice B - Normas y fundamentos: Este capítulo cubrirá conceptos teóricos que fueron estudiados y aplicados durante la ejecución del proyecto.
- Apéndice C - Memoria de programación: Contiene detalles técnicos sobre como se desarrolló la programación.

Capítulo 2

Diseño del dispositivo

2.1. Operativa del dispositivo

En esta sección se presentan las funciones que puede ejecutar el prototipo final. Asimismo, se describe su utilización desde la perspectiva del usuario, con el fin de ilustrar el modo de operación y las principales interacciones con el sistema.

El equipo construido cumple con las funciones de extraer de manera autónoma una muestra del líquido a evaluar, mezclarla con el reactivo requerido para la realización del ensayo espectroscópico y registrar automáticamente los resultados obtenidos en cada medición. Asimismo, el sistema permite en forma manual el control de los actuadores, ofreciendo una alternativa adicional para la operación y supervisión del proceso.

El apéndice B.2 explica a detalle el proceso de la espectroscopía y porqué es extremadamente útil en el campo químico, alimenticio, ecológico o industrial. En este proyecto se realizan ensayos con longitud de onda fija, ya que la fuente de luz es idealmente monocromática, por lo tanto, transmitirá una longitud de onda en particular. Sin embargo, el dispositivo permite un fácil intercambio del emisor de luz, siendo la ventana de recepción del receptor utilizado entre 300 nm y 700 nm. En el apéndice B.2.4 se especifican diferentes análisis particulares que utilizan la espectroscopía y la frecuencia de luz requeridas para obtener resultados satisfactorios. Como ejemplo, en la determinación de presencia de fosfatos se suele utilizar fuentes de luz con una longitud de onda de 880 nm, mientras que en la determinación de la pureza del aceite de oliva se suelen utilizar entre 230 y 270 nm. Cualquiera sea la longitud de onda aplicada, contará con un sensor que realizará una medida de la energía lumínica que logra pasar a través del líquido depositado dentro de una cubeta de cuarzo para espectrofotómetro. Realizando dos medidas, una sobre la muestra sin reactivo aplicado y otra sobre la mezcla, es posible registrar valores de absorbancia o transmitancia comúnmente utilizados en este tipo de ensayos.

El equipo se controla mediante una interfaz visual fácilmente manipulable por el usuario, esta interfaz es manejada mediante cualquier PC que se establezca en la misma LAN. Se verá en la pantalla, un menú donde se controlarán las acciones y se desplegarán las medidas tomadas. La correcta manipulación de este menú es explicada en la sección 2.2.

El equipo posee dos modos de funcionamiento, un modo manual y un modo automático.

El modo manual permite configurar la velocidad en RPM, de las bombas encargadas de suministrar el reactivo y la muestra de agua individualmente. Se podrá encender o apagar cada una de las bombas. También se permite en este modo el control de la electroválvula que desecha la muestra una vez finalizado el ensayo.

El modo automático, por otra parte, permite dosificar directamente la cantidad de líquido de muestra y la cantidad de reactivo que se agregará (ambos valores en ml) para formar la solución. Luego se tomará la cantidad de medidas requeridas con el sensor óptico integrado.

Este equipo fue diseñado para utilizarse en el ámbito industrial y separando los componentes eléctricos (figura 2.1) de los hidráulicos (figura 2.2). Todos los componentes que forman parte del proceso, fueron instalados sobre una bandeja perforada para fondo de tablero eléctrico. Se considera su instalación dentro de un tablero próximo al depósito de los líquidos que se deseen estudiar.

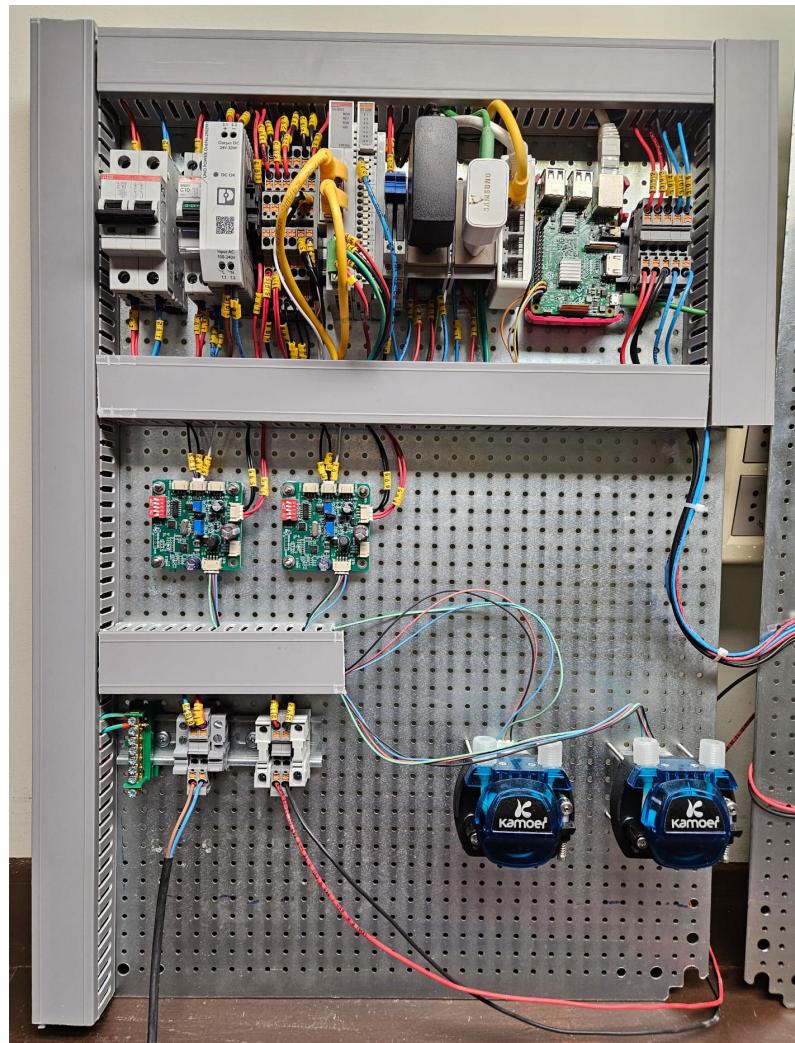


Figura 2.1: Vista frontal de placa eléctrica del Espectrofotómetro Industrial Automatizado

La arquitectura general del sistema consiste en dos bombas peristálticas (Sección 2.6.1) que dosifican a la capsula de medida, accionadas por drivers controlados de manera serial por el PLC. El PLC integra la lógica del proceso en sus distintos modos en su programación y establece comunicación vía ethernet con la tarjeta Raspberry Pi. La Raspberry Pi dispone de un servidor web donde se despliega la visualización en la misma LAN y también integra la comunicación con el sensor, tomando los valores a comparar de manera relativa.

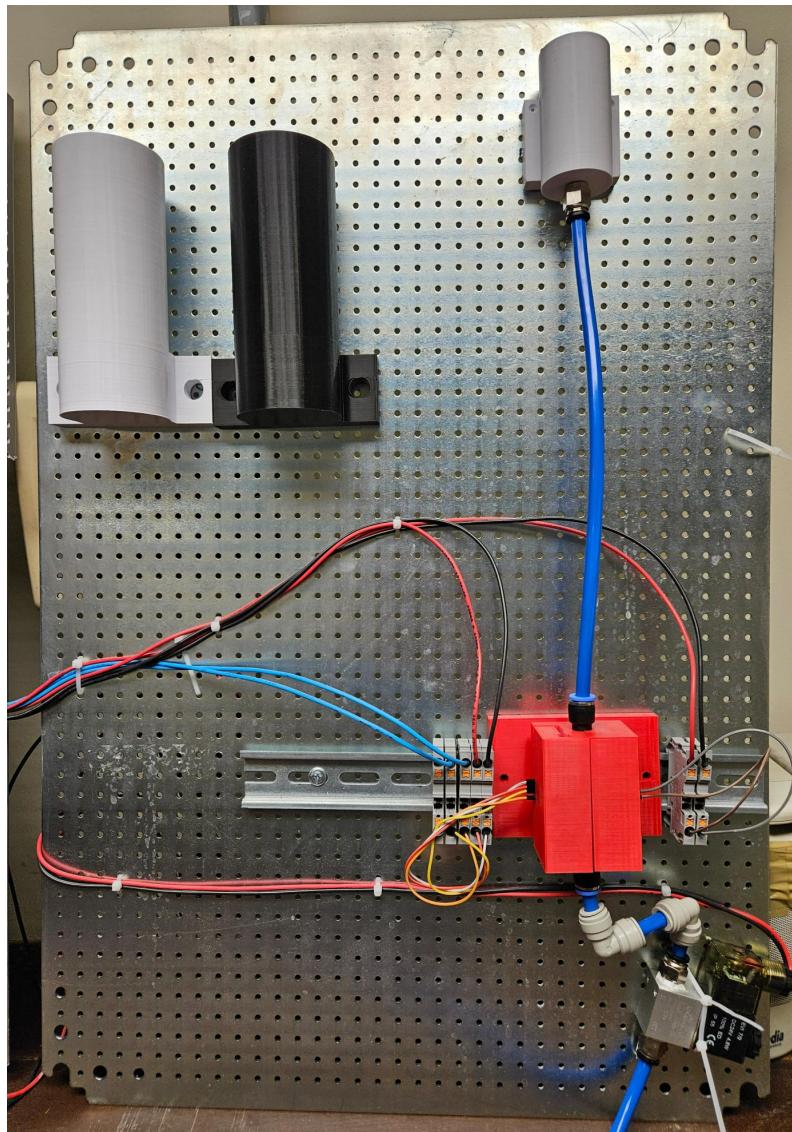


Figura 2.2: Vista frontal de placa hidráulica del Espectrofotómetro Industrial Automatizado

2.2. Manual de usuario

El equipo debe ser conectado a un tomacorriente tres en linea en 230 V, luego se debe integrar el switch a la LAN privada donde se haga uso (Sección 2.6.7). Finalmente, se abrirá un buscador donde se introducirá la dirección IP 192.168.100.10/webvisu:8080, este enlace nos permitirá visualizar el menú que se implementó para operar al equipo. Se puede acceder de forma rápida con un PC a través de Ethernet configurando una dirección IP en el rango 192.168.100.0/24 distinta de 192.168.100.10 (Tarjeta Raspberry Pi) y 192.168.100.100 (PLC).

Una vez se acceda al menú principal, se verán tres elementos interactivos, tal como se muestra en la figura 2.3, estos pulsadores permitirán al usuario acceder a los tres menús secundarios de modo manual, modo automático y menú de medidas, donde se desplegarán los resultados de las medidas tomadas con el sensor óptico.

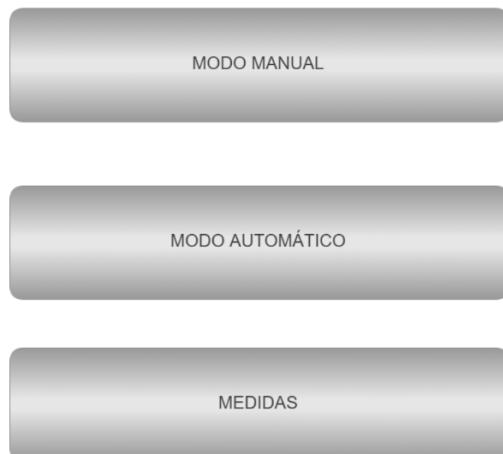


Figura 2.3: Menú principal

2.2.1. Modo manual

Para realizar cualquier acción dentro de este modo, se debe activar primero el interruptor etiquetado como “Modo Manual”.

El menú de modo manual (Figura 2.4) permite ingresar la velocidad requerida en cada una de las bombas en RPM y luego encenderlas o apagarlas según sea necesario, también se mostrará en este menú una lectura constante de la velocidad actual de cada componente de bombeo.

Este menú permite también el control de la válvula encargada del desecho de la solución desde la cubeta (Sección 2.7.2), para luego recibir una nueva mezcla. El control de esta válvula se realiza con el pulsador etiquetado “Control manual válvula desagüe” que cambia el estado del componente. Puede verse debajo del pulsador, una luz indicando si la válvula está cerrada o abierta.

El interruptor “Menú” permitirá al usuario regresar al menú principal para cambiar de modo.

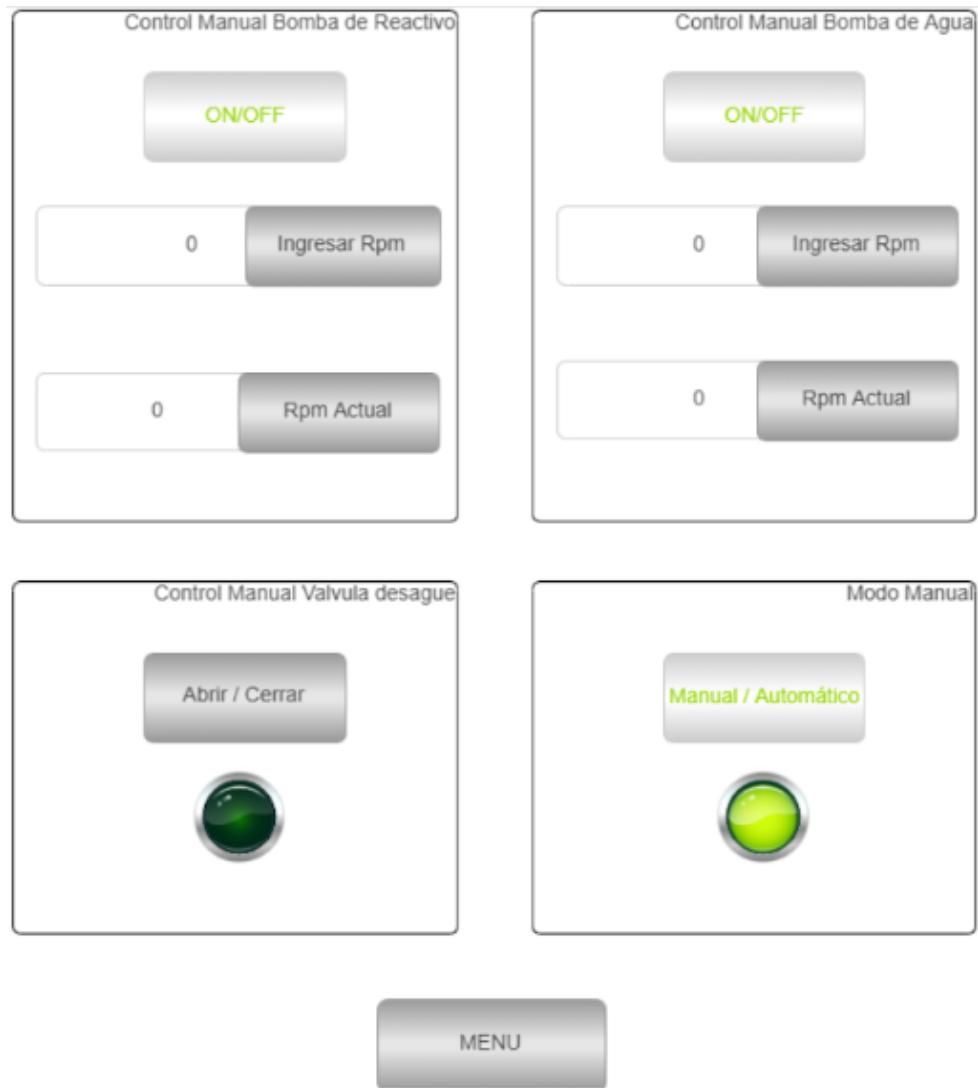


Figura 2.4: Modo manual

2.2.2. Modo automático

El menú de modo automático (Figura 2.5) proporciona una lectura en tiempo real de la velocidad de cada una de las bombas dosificadoras (Rpm Actual), se muestra también el estado de las electroválvulas.

Para realizar cualquier acción dentro de este modo, se debe activar primero el interruptor etiquetado como “Modo Automático”.

Este menú permite establecer parámetros particulares para realizar un ensayo específico. En caso de ejecutarse un ensayo en modo automático, se especifica la cantidad de agua

a dosificar en ml, la cantidad de reactivo que se aplicará a la mezcla también en ml y la cantidad de medidas que se desean tomar, para cada una de las medidas configuradas se realizará una nueva solución con las mismas proporciones. Una vez especificados los parámetros del ensayo, se debe presionar el pulsador “inicio” comenzando el proceso, en caso de presionarse “parar” se apagan las bombas y se cierra la electrovalvula. Las medidas se registran luego de realizar la dosificación con la solución en reposo.

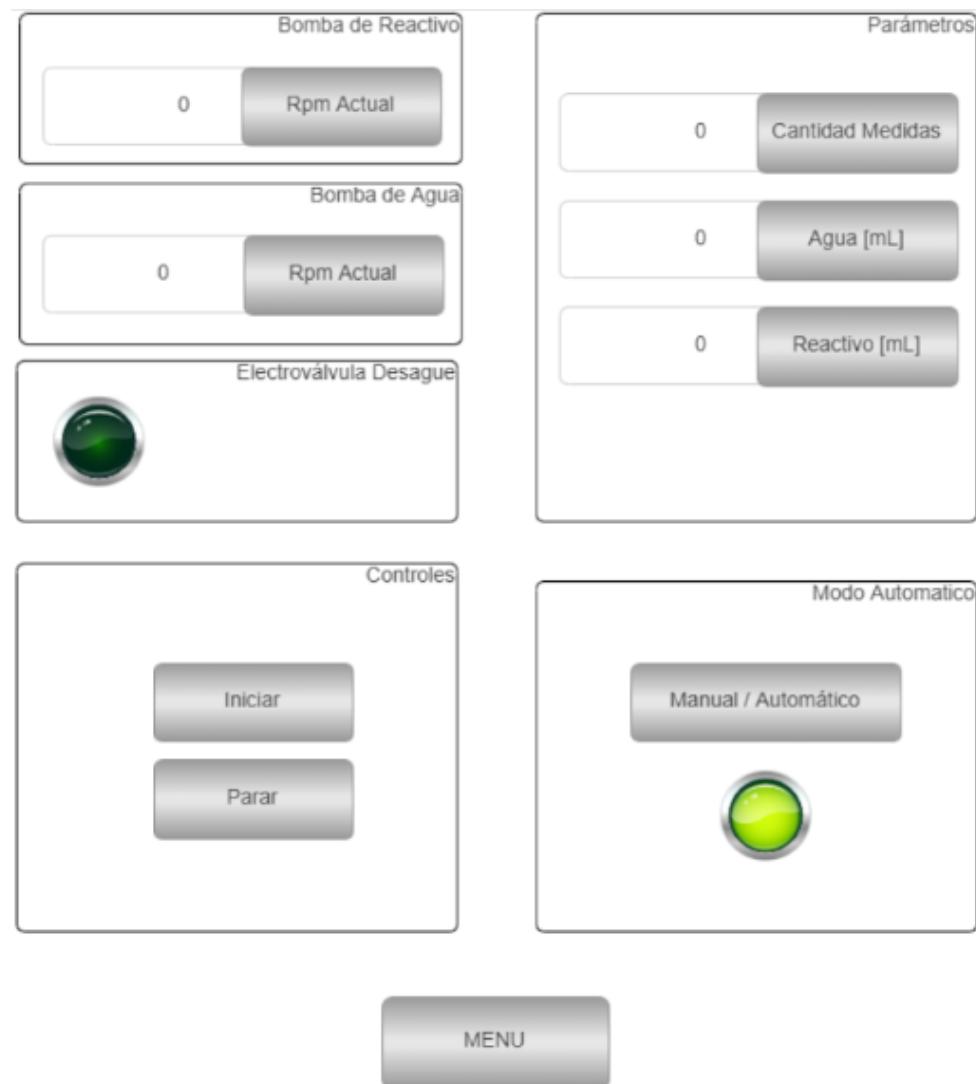


Figura 2.5: Modo automatico

El menú de medidas mostrará en una tabla el resultado de las medidas tomadas en los ensayos, estas medidas serán mostradas en nanovatios por centímetro cuadrado, unidad de densidad de potencia que indica cuanta energía se transmite a través de la cubeta que contiene la mezcla.

También se muestra en esta tabla, la cantidad de agua y la cantidad de reactivo por

cada valor que se registra.

La tabla de ensayos se irá llenando, desplazando hacia abajo los resultados de ensayos previos y colocando en la primera fila de la tabla al último ensayo ejecutado.

2.3. Diagrama del sistema

En la figura 2.6 se muestra un diagrama de bloques representando las diferentes categorías funcionales de los componentes utilizados en la fabricación del Espectrofotómetro Automatizado Industrial; las flechas rojas representan una conexión eléctrica y las azules una vinculación física o hidráulica.

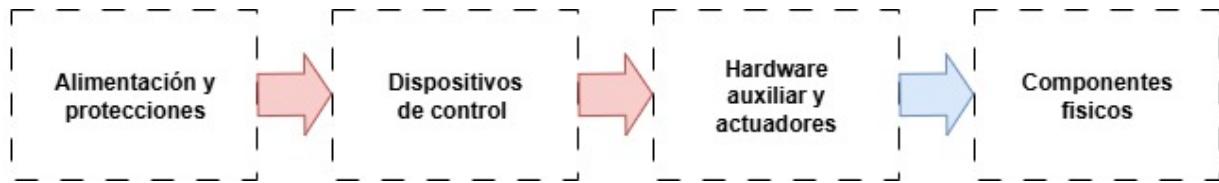


Figura 2.6: Diagrama funcional de componentes

En la figura 2.7, los componentes principales están diferenciados en sus correspondientes categorías de función y se distinguen con líneas rojas y azules, las conexiones eléctricas e hidráulicas respectivamente.

Distinguimos cuatro categorías funcionales, entre los elementos utilizados dentro de este proyecto:

- Alimentación y protecciones - Elementos aplicados para energizar y proteger de fallas eléctricas.
- Dispositivos de control - Controladores del sistema que interactúan con los drivers o preactuadores como relés. También establecen una comunicación entre ellos y con la interfaz de usuario.
- Actuadores, sensores y HW auxiliar - Drivers para las bombas peristálticas accionadas por motor de paso, circuito auxiliar para uso del sensor incluyendo led interno para la celda. Actuadores del sistema válvula de desagote y dos bombas peristálticas de motor de paso.
- Componentes físicos no electrónicos - Elementos no eléctricos implementados en la construcción del dispositivo, permiten el circuito hidráulico y realizar la mezcla en la celda.

A continuación explicaremos y asociaremos los componentes de este diagrama con los planos unifilares incluidos en el Anexo. Cada componente utilizado en el prototipo, incluidos los elementos que no figuran en el diagrama, serán presentados y explicados en detalle en la sección correspondiente a su categoría funcional.

El sistema se conecta a alimentación de UTE 230 VAC en monofásico (unifilar 00). Luego de la conexión a tomacorriente, se coloca una llave termomagnética (TM) implementada para detectar cortocircuitos y sobrecargas.

Esta llave termomagnética se conecta a una llave diferencial (DIF - unifilar 01) no montada en el prototipo entregado, para proteger al usuario en caso de tocar algún elemento bajo tensión o cortocircuitos a tierra. Tres conversores son alimentados a partir de este punto (Sección 2.4.1), estos equipos alimentarán al PLC (Sección 3.1), Raspberry (Sección 2.5.2) y un switch de puertos ethernet (Sección 2.6.7) utilizado para interconectar el PLC, la Raspberry y la red LAN privada que se utiliza para acceder a la interfaz de usuario (Unifilar 04).

El conversor AC/DC “UNO POWER” (CONV1 - unifilar 01) tendrá una salida de 24 VDC donde se implementó un fusible de respaldo (Sección 2.4.3) y será utilizado para alimentar múltiples equipos, entre los que se encuentran el PLC, la electroválvula (Sección 2.6.3), un relé de control (Sección 2.6.4) y los drivers (DV1, DV2) (Sección 2.6.2) que controlan a los motores de paso de las bombas peristálticas (Unifilares 02, 05 y 06).

El unifilar 03 muestra la Raspberry (RP), el sensor (PS) y detalla la conexión entre estos componentes.

Finalmente, los planos 07 y 08 presentan, respectivamente, una vista frontal del prototipo y el diagrama completo de todas las conexiones entre los elementos del proyecto.

Tal como se muestra en la figura 2.7, el PLC manejará tanto los drivers como la electroválvula. Cada driver controla una de las bombas (Sección 2.6.1) utilizadas para la dosificación del reactivo y la muestra.

El PLC y la Raspberry se comunicarán entre ellos facilitando el proceso automático de medición, la Raspberry controla al sensor y es programada con la interfaz de usuario.

En las próximas secciones se detallan los componentes utilizados en el proyecto dentro de cada una de las categorías funcionales representadas en la figura 2.7.

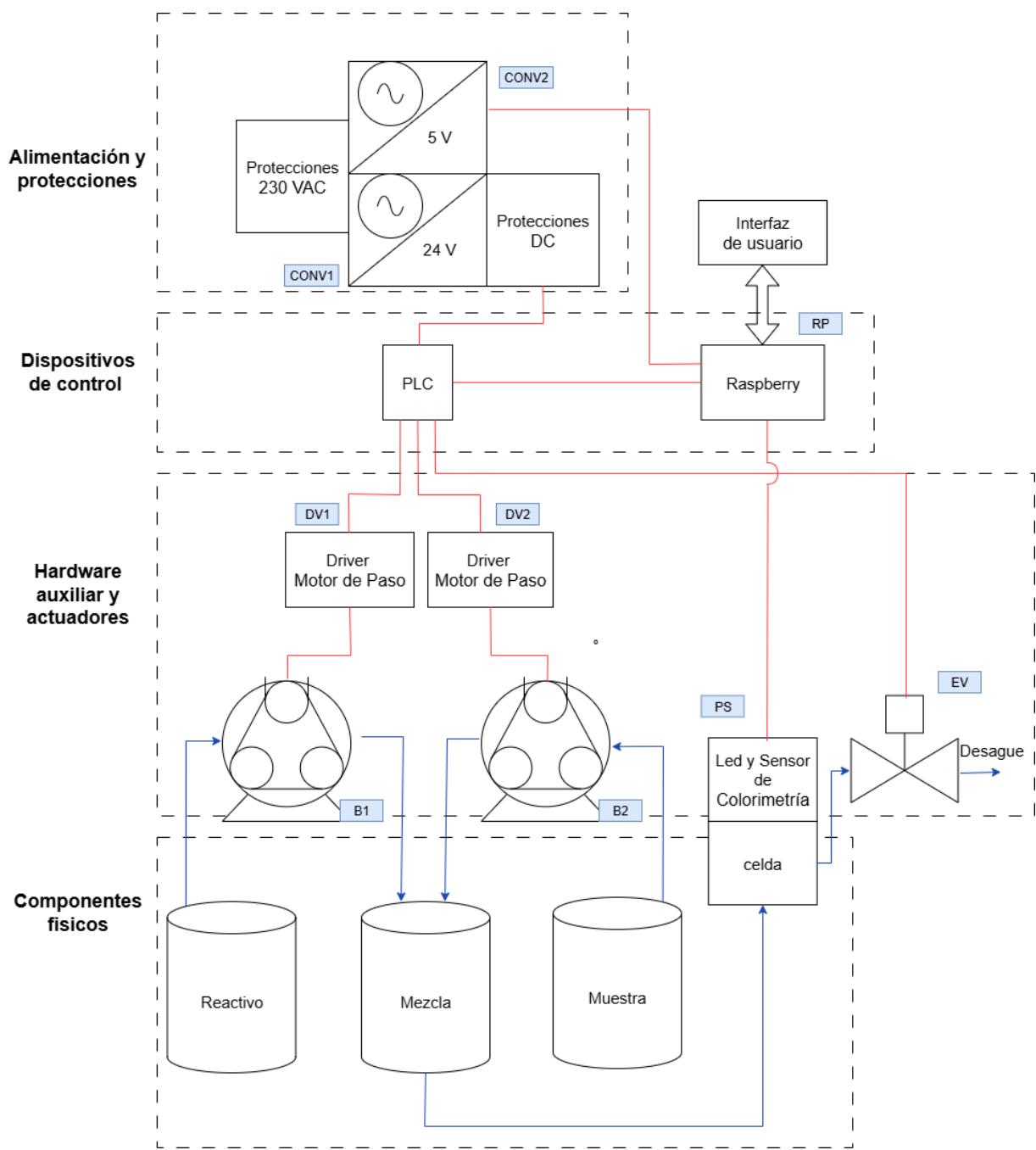


Figura 2.7: Diagrama de componentes principales

2.4. Alimentación y protecciones

En la alimentación del dispositivo se usaron dos cables de 0,75 mm² en 230 VAC, también se instalaron cables de 0,75 mm² en 24 VDC. Detallamos a continuación los elementos principales instalados en el circuito de suministro eléctrico.

2.4.1. Conversores

Requerimos de una fuente de 24 VDC para alimentar tanto el PLC, como los drivers que controlan las bombas y el relé que controla la electrovalvula. Utilizamos con este fin, el conversor UNO POWER 2902991 [10], que permite un rango de tensión de entrada de 100 VAC - 230 VAC y genera una tensión de salida en 24VDC/30W.

Para evaluar la cantidad de potencia que tiene que entregar el conversor de 24 V, consideramos que las bombas peristálticas tienen un consumo máximo de 36 W a 350 rpm. Sin embargo, en el proceso automático de medida utilizamos dos bombas funcionando simultáneamente a 20 rpm y en el lavado se activa solamente una de las bombas a 100 rpm. Estimamos entonces un consumo máximo de 18 W.

Por otro lado, la electroválvula tiene un consumo puntual de 5 W, el plc 1,2 W y el relé 0,22 W. Consideramos que los 30 W nominales de capacidad del conversor UNO POWER 2902991 son más que suficientes para asegurar el funcionamiento del sistema.

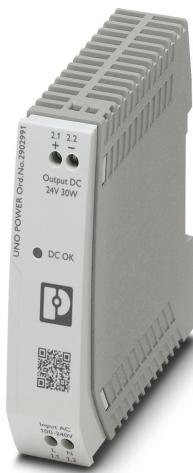


Figura 2.8: Conversor 230 VAC / 24 VDC - UNO POWER 2902991 [10]

El switch de puertos Ethernet y la Raspberry pi fueron alimentados mediante los cargadores originales suministrados por el fabricante, incluidos en el paquete al momento de la adquisición. Son dos cargadores estándar de 5 VDC / 3 A como el representado en

la figura 2.9. Estos dispositivos son referenciados como CONV2 (Raspberry) y CONV3 (Switch) dentro del unifilar 1.



Figura 2.9: Alimentación de Raspberry Pi - 5 VDC / 3 A

El conversor UNO POWER se conecta a 230 Vac. Los dispositivos CONV2 y CONV3 se conectan a tomacorrientes instalados dentro del prototipo.

2.4.2. Llave termomagnética

Utilizamos una llave termomagnética ABB modelo SN201-C10 para seccionar y proteger el equipo en nivel 230 VAC.



Figura 2.10: Llave termomagnética ABB modelo SN201-C10

2.4.3. Fusible

Se utilizan en la industria fusibles con valores estándar de disparo en 1, 2 y 5 amperes. Considerando que implementamos una fuente de 24 VDC y 30 W, en condiciones nominales de funcionamiento se entregaran 1,25 A. Para evitar que la fuente entregue más corriente de la que puede manejar y se dañe, dentro de un borne carril para fusible (figura 2.11), colocamos un fusible con corriente de corte equivalente a 2 amperes. Este fusible también protegerá a todos los elementos alimentados en 24 V y será conectado acorde al unifilar 1 (Posición E 5).

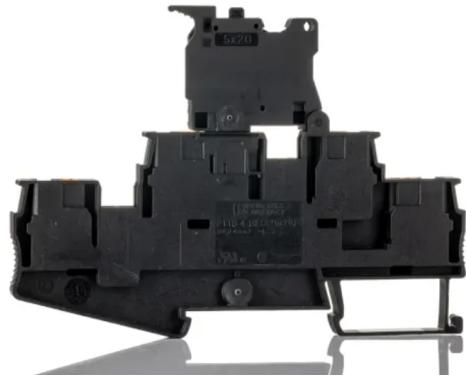


Figura 2.11: Borne carril para fusible

2.5. Dispositivos de control

2.5.1. PLC Crevis GL-9972

Este PLC modelo GL-9972 [11] programable, desarrollado por la empresa Crevis, está diseñado para aplicaciones industriales y permite, mediante módulos acoplables en el costado del dispositivo, agregar entradas y salidas analógicas o digitales.

La figura 2.12 muestra una vista frontal del PLC. Se pueden ver sus dos puertos ethernet, utilizamos el superior para su conexión ethernet con el switch y la LAN utilizando el protocolo Modbus TCP, y el inferior corresponde a su puerto serial, donde tiene una comunicación en RS485 mediante el protocolo Modbus RTU con los drivers para las bombas peristálticas.

En el controlador se definió resolver el manejo de actuadores, la lógica del proceso y la definición de variables que se esperan de la visualización por medio de la comunicación con la Raspberry Pi.

En la figura 2.13 se tiene el módulo de salidas digitales CREVIS GT-226F implementado en este proyecto. Las conexiones realizadas sobre el PLC se muestran en el unifilar 02.

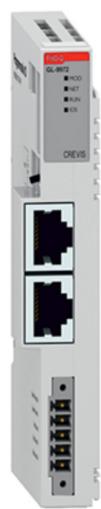


Figura 2.12: PLC Crevis GL-9972



Figura 2.13: Módulo de salidas digitales CREVIS GT-226F

Se definió la utilización de un PLC como controlador del proceso, dado que se optó por no desarrollar un hardware a medida. Esta elección fue considerada de mayor valor para el perfil académico y profesional de ambos estudiantes.

2.5.2. Raspberry PI

La tarjeta Raspberry Pi fue seleccionada debido a su compatibilidad con el entorno CODESYS (Sección 3.1) y a la disponibilidad de un puerto de comunicación I2C, característica poco frecuente en los controladores industriales. Otro aspecto valorado fue la posibilidad de contar con un servidor web integrado, al cual se puede acceder mediante el puerto Ethernet, así como la capacidad de programar visualizaciones directamente en el entorno CODESYS.



Figura 2.14: Raspberry pi 3b+

Los pines de conexión verticales mostrados en la figura 2.14 fueron utilizados para controlar y leer el sensor OPT3002 [4] con comunicación I2C. La figura 2.15 mapea la función que puede cumplir cada uno de estos pines. La conexión de este controlador con el sensor OTP3002 y el led utilizado como fuente de luz, es detallada dentro del unifilar 3. En el plano referenciado, se señala que tanto el sensor como el led fueron alimentados en 3,3 volts (pin 1) y la comunicación I2C es ejecutada por los pines 3 y 5 (pines reservados en el dispositivo para esta función), el pin 39 se implementa como tierra.

Figura 2.15: Numerado de pines en Raspberry pi 3b+

2.6. Hardware auxiliar y actuadores

Presentaremos en esta sección el hardware no controlador, utilizado en el dispositivo. En el caso de las bombas peristálticas, se incluye el driver correspondiente para su control. Se presentará la electroválvula, la cual requiere un circuito auxiliar con relé y el módulo de salidas digitales del PLC. Posteriormente, se presentará sensor OPT3002 junto con el circuito adicional necesario para establecer la comunicación mediante el protocolo I2C. Finalmente, se describe el switch de puertos ethernet utilizado, montado sobre riel DIN.

2.6.1. Bombas peristálticas

Se decidió debido a su precisión y confiabilidad utilizar dos bombas peristálticas kamoer KCM idénticas para dosificar tanto el reactivo como la muestra de agua a evaluar. Estas bombas funcionan utilizando 3 cilindros que presionaran un tubo de 1,6 mm de diámetro interno. La estructura interna de estos dispositivos se muestra en la figura 2.17. En el extremo derecho de esta figura se puede ver el motor de paso (sección B.1) y referenciado como “Disco sincrono” se muestra una pieza circular con tres rodillos que al girar el motor, presionan el tubo liberando líquido.



Figura 2.16: Bomba micro-peristáltica kamoer KMC [3]

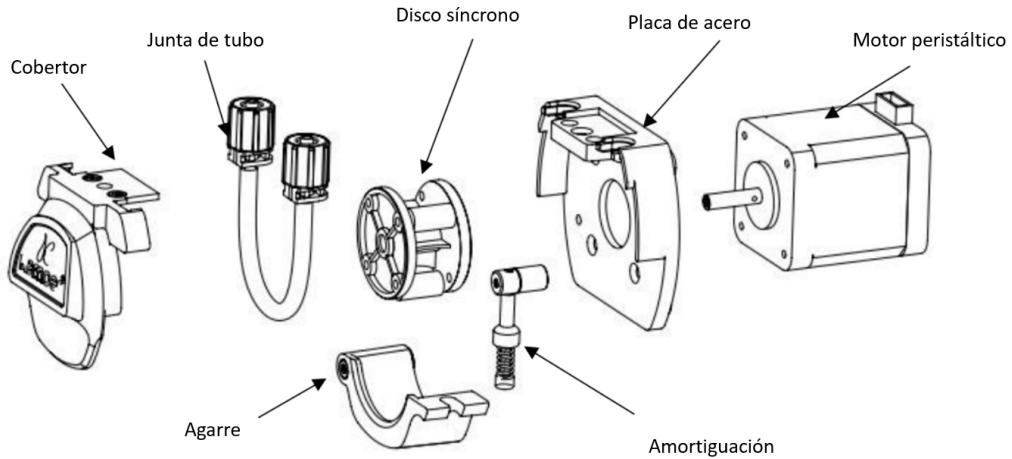


Figura 2.17: Elementos que forman a las bombas peristálticas kamoer [3]

En este caso se utiliza un motor de paso bipolar (Apendice B.1). Este motor tiene un ángulo de rotación igual a $1,8^\circ$ y como se muestra en la figura 2.16, es controlado mediante cuatro cables. Dos de estas entradas son usadas para alimentar una bobina y las dos restantes alimentan la segunda. El manual técnico de este equipo nos proporciona también la relación entre el flujo por minuto y el RPM impuesto en el motor, en nuestro caso se utilizaron caños de 1,6 mm de diámetro interno para lograr la mayor precisión posible, el flujo (ml/min) generado por el equipo cuando se aplica este diámetro de caño es representado por la curva azul en la figura 2.18.

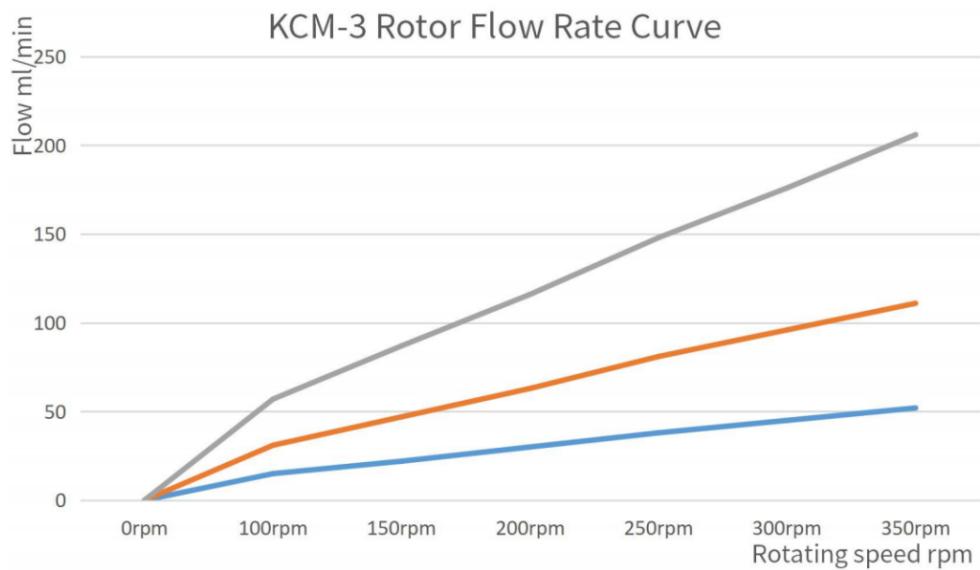


Figura 2.18: rpm vs caudal de salida (azul - caño 1.6 mm) [3]

2.6.2. Driver para motores de paso

Para controlar los motores paso a paso, se seleccionaron los drivers kamoer modelo 4460.5, que fueron diseñados específicamente para manejar las bombas kamoer KCM. Estos drivers son conectados a las bombas y pueden ser controlados mediante dos métodos. Aplicando una configuración específica sobre los interruptores S1 mostrados en la figura 2.19, es posible seleccionar entre un modo manual para controlar las bombas o un control mediante ModBusRTU. En el caso del control manual, los drivers son alimentados por una fuente DC de 24V y mediante un potenciómetro se regula el RPM del motor.

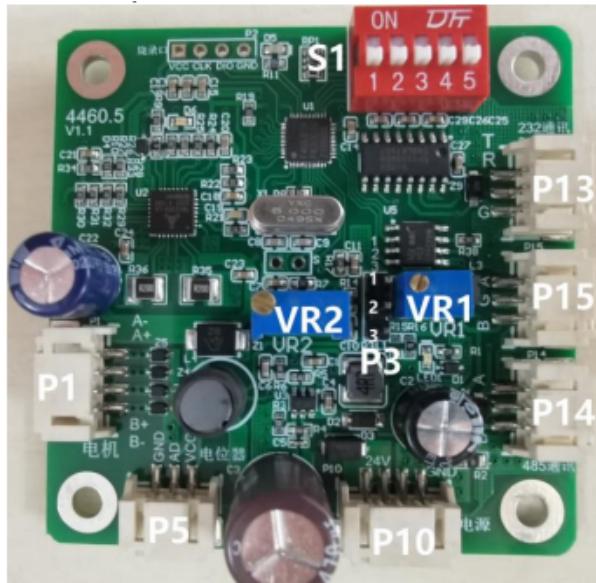


Figura 2.19: Placa de drivers 4460.5 [5]

En nuestro proyecto las bombas fueron controladas mediante el protocolo de comunicación ModBusRTU presentado en la sección B.3.1. Para controlar el driver con protocolo ModBusRTU se debe, en principio, colocar el switch 1 de S1 en posición OFF, luego se configura el baud rate que establece la velocidad con que se transmiten los caracteres en la comunicación. El baud rate es establecido mediante el swtch 2 de S1 que dependiendo de su posición establece una configuración estándar de 9600 baudios.

La entrada P1 que se muestra en la figura 2.19, se usa para conectar el driver con la bomba que controlará. P10 se implementa para alimentar a 24 V el driver y la bomba. P14 y P15 se utilizan para conectar el driver con el dispositivo controlador.

Utilizando ModBusRTU, es posible encender las bombas, apagar las bombas, leer la velocidad actual de funcionamiento en RPM y dar órdenes de tanto velocidad como sentido de giro.

2.6.3. Electroválvula

Para el desecheo de la mezcla una vez finalizado el ensayo, se implementó una electroválvula Heschen modelo 2V025-08. En su estado sin energizar se encuentra normalmente cerrada (NC), y al ser energizada modifica la posición de su obturador permitiendo el paso del líquido. La alimentación de la electroválvula se realiza a través de un relé de estado sólido, presentado a continuación. El componente opera a una tensión de 24 VDC y presenta un consumo de 6 W de potencia.



Figura 2.20: Electroválvula Heschen modelo 2V025-08

2.6.4. Relé

Para la conmutación de la alimentación de la electroválvula se empleó un relé de estado sólido. Es estándar en la industria de automatización implementar relés para el control de hardware secundario, evitando así que se dañe el PLC en caso de producirse una falla eléctrica en la válvula.

Este dispositivo presenta una tensión de control y de salida de 24 VDC, con una corriente máxima de salida de 3 A. Su montaje se realiza sobre carril DIN mediante un zócalo específico para relés, lo que facilita su integración y reemplazo en el sistema.



Figura 2.21: Relé de estado sólido



Figura 2.22: Zócalo para relé

2.6.5. Módulo de salidas digitales CREVIS GT-226F

En la Figura 2.13 se ilustra el módulo, diseñado con un formato de acople a la CPU del PLC y provisto de 16 salidas digitales de 24 V a 0,5 A. Para el control de la electroválvula se emplea un relé de estado sólido, evitando así su alimentación directa a través del módulo. Si bien la corriente admisible lo permitiría, esta práctica no se considera adecuada en términos de diseño y fiabilidad del sistema.

2.6.6. Sensor óptico

Para medir la energía de luz que atraviesa a la muestra luego de aplicar el reactivo, se utilizará el sensor OPT3002. Este sensor fue adquirido de la empresa Texas Instruments ya montado en una placa con su circuito de acondicionamiento.

Como se muestra en la figura 2.23 esta placa tiene 2 entradas disponibles. La entrada J1 fue utilizada para evaluar el sensor mediante un dispositivo USB y un software proporcionado por Texas Instruments. El accesorio con entrada USB del sensor, permite visualizar las medidas tomadas de intensidad de luz en una PC ya sea de forma gráfica o en un registro de datos que puede ser guardado en un archivo de texto.

Por otro lado, la entrada J2 permite la conexión de las salidas del sensor al dispositivo controlador (Raspberry). Esta placa incluye la capacitancia C1 requerida para el correcto funcionamiento del sensor OPT3002 y una conexión de la salida ADDR a tierra por medio de una resistencia, estableciendo así una dirección particular de destino en el protocolo de comunicación I2C que se utiliza para enviar comandos y leer las medidas tomadas, este protocolo y su aplicación para el control de este sensor en particular son explicados en la sección B.3.2 de este documento.

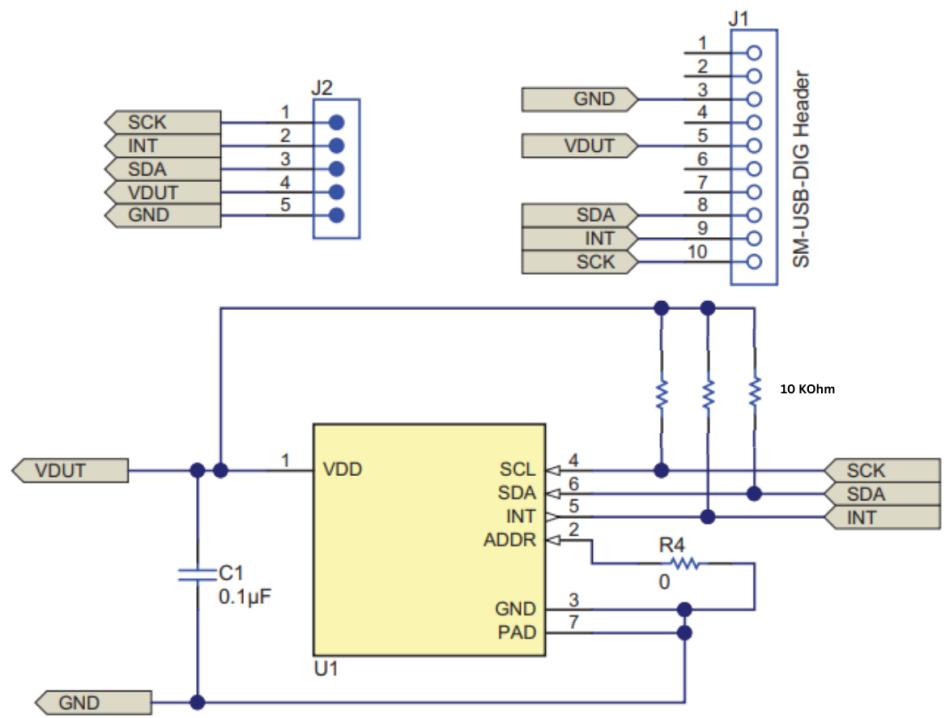


Figura 2.23: Esquemático la placa donde se monta el sensor - luz a digital - OPT3002 [2]

Para controlar este sensor óptico, fue necesario estudiar las entradas y las salidas originales del sensor OPT3002 y no solo la placa de Texas Instruments sobre la que se fue montado para facilitar la conexión de los cables de alimentación y control. En la figura 2.24 se pueden identificar los puertos del sensor OPT3002, se explica a continuación la función de cada uno:

1. VDD - Esta entrada se utiliza para alimentar al dispositivo con los 1,6 a 3,6 Volts requeridos para su funcionamiento.
2. ADDR - Esta entrada define la dirección del dispositivo en la comunicación I2C, dependiendo de con qué señal se cortocircuite (GND, VDD, SDA, SCL), la dirección sera diferente.
3. GND - Conexión a tierra del dispositivo.
4. SCL - Entrada de reloj, esencial para la comunicación I2C. Debe conectarse con una resistencia de Pullup (10 KΩ) a la fuente de alimentación.
5. INT - Salida de interrupciones programable para actuar en caso de alcanzarse cierto nivel de medidas u otras situaciones particulares.
6. SDA - Entrada de datos, esencial para la comunicación I2C. Debe conectarse con una resistencia de Pullup (10 KΩ) a la fuente de alimentación.

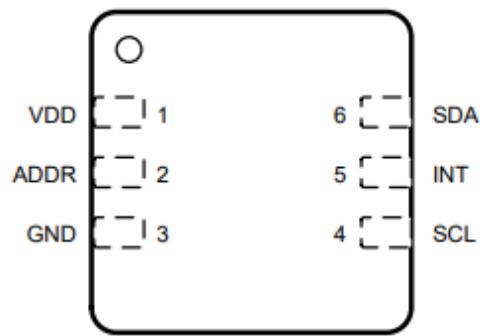


Figura 2.24: Esquemático de entradas del sensor OPT3002

2.6.7. Switch de puertos ethernet

Se implementó el switch de puertos ethernet MERCUSYS MS105 (Figura 2.25) para establecer una red LAN cableada entre el PLC, la tarjeta Raspberry Pi y la red privada del usuario. De esta forma se posibilitan tanto las modificaciones en la programación de ambos dispositivos como el acceso al servidor web de la Raspberry Pi. El switch se alimenta con una tensión de 5 V provista por el conversor *CONV3* y dispone de un total de cinco puertos de red.



Figura 2.25: Switch de puertos ethernet

2.7. Componentes físicos

2.7.1. Montaje final

La figura 2.26 muestra una vista frontal del equipo, donde decidimos separar los componentes eléctricos de los componentes hidráulicos, evitando así dañar la parte eléctrica del dispositivo ante posibles fugas de líquido.

El plano número 7 muestra la disposición e identificación de cada uno de los componentes en las dos placas metálicas.

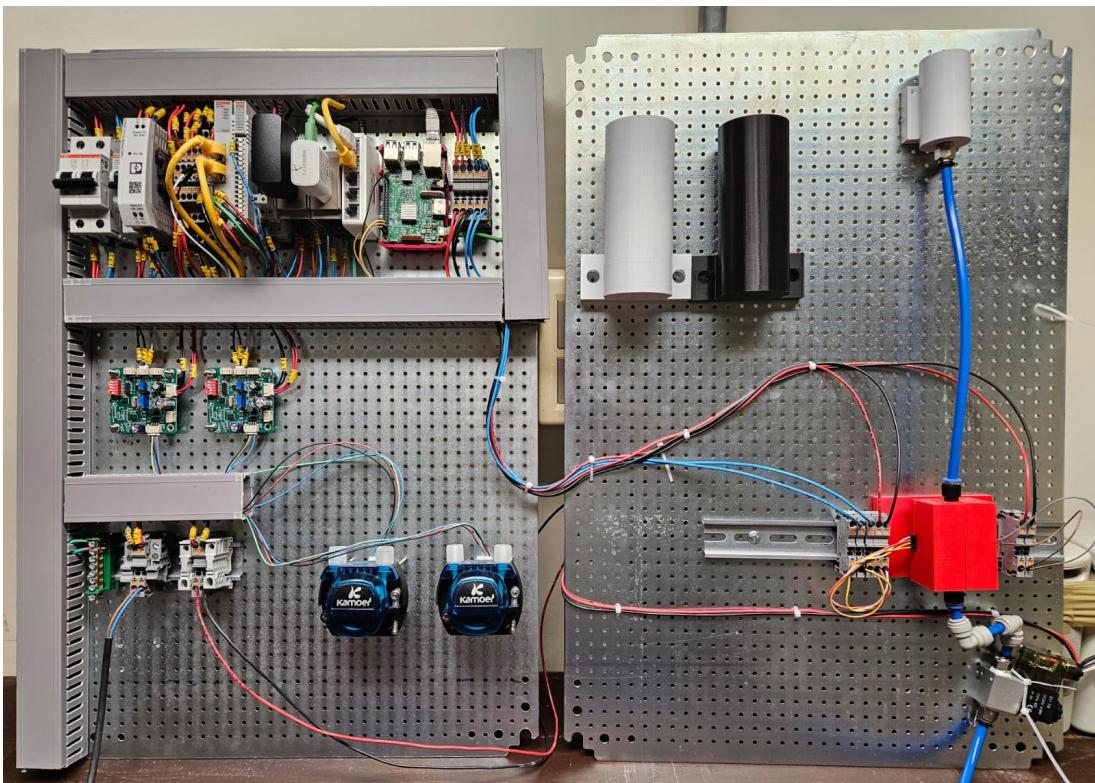


Figura 2.26: Vista frontal del prototipo

2.7.2. Cubeta de cuarzo

Utilizamos cubetas de cuarzo para tests con espectrofotómetros, como recipiente para contener la mezcla sobre la cual se realizará el ensayo óptico. Estas cubetas tienen 2 caras opacas enfrentadas y dos transparentes, tal como se muestra en la figura 2.27. El sensor estará enfrentado a la luz led con las caras transparentes de la cubeta en el medio.

La base inferior de la cubeta fue retirada para permitir acoplar la electroválvula de desagüe. En modo automático se controla la misma para, luego de dosificar tomar en reposo la medida fotométrica.



Figura 2.27: Cubeta de cuarzo - 4,5 x 1,2 x 1,2 cm

2.7.3. Cámara para test óptico

Diseñamos una cámara para realizar el test óptico sobre la mezcla del reactivo con la muestra, de esta manera se evitará que entre luz desde el exterior, lo que alteraría los resultados del test dependiendo del entorno donde se encuentre el dispositivo.

La cámara fue diseñada (figura 2.28) con dos elementos que serán encastrados encerrando en su interior la cubeta de cuarzo modificada y generando dos orificios circulares en los extremos de la cubeta, con el diámetro necesario para permitir el paso de los tubos que transportan a la mezcla.

Se fijó en cada extremo de la cubeta un acople rápido que permite conectar y desconectar la cámara del sistema, de manera simple.

Uno de los elementos fijará la luz led y el otro fijará el sensor. Estos dispositivos estarán enfrentados en el punto medio de la cubeta.

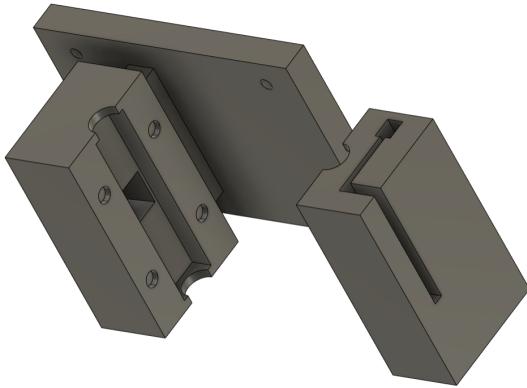


Figura 2.28: Diseño 3D de cámara para test óptico

2.7.4. Recipientes de reactivo y Muestra

Se diseñaron también tres recipientes, dos de ellos son idénticos y se utilizan para contener por separado el reactivo y la muestra previo a la mezcla. El tercer recipiente tiene un tamaño menor y es implementado para recibir la muestra y el reactivo desde las bombas, conduciendo la solución hacia la celda.

2.7.5. Placas

Los equipos serán montados sobre placas para tableros de tamaño 500x700 mm, esta decisión fue tomada ya que permite visualizar los procesos de una manera mucho más clara que en la implementación de una caja cerrada. Además, decidimos orientar la construcción para que el equipo sea utilizado en un entorno industrial, por lo tanto, la implementación de este tipo placas es conveniente al permitir el montaje dentro de un tablero con las dimensiones adecuadas.

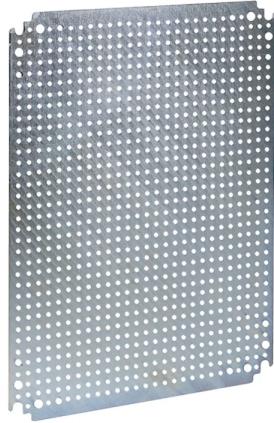


Figura 2.29: Placa Metálica Perforada Schneider - 500 x 700 mm

Capítulo 3

Software para controladores

3.1. Ambiente de programación

Se definió como estrategia la unificación del entorno de programación de los dispositivos involucrados, lo que aportó beneficios en términos de estandarización y mantenibilidad, aunque también supuso desafíos adicionales en determinados aspectos. El entorno de programación CODESYS fue seleccionado debido a su independencia respecto a fabricantes, su amplio ecosistema de paquetes y librerías, y su capacidad para integrar, en un mismo entorno, una amplia gama de protocolos de comunicación industriales que abarcan controladores, sensores, actuadores, variadores de frecuencia, entre otros.

El PLC contaba desde el inicio con licencia nativa de CODESYS, mientras que en la Raspberry Pi se instaló un paquete específico del mismo entorno, lo que permitió homogeneizar el desarrollo en ambos dispositivos.

El PLC, como controlador principal, tiene como objetivo la integración del sistema de dosificación y la ejecución automática de consignas previamente definidas, gestionando la lógica de control del proceso y sus distintos accionamientos. La Raspberry Pi, por su parte, cumple funciones complementarias al establecer la comunicación con el sensor de colorimetría (o una red de sensores), desplegar una interfaz de usuario en una LAN definida y convertir los parámetros ingresados en variables de proceso. Estas variables son transmitidas al PLC o a una red de controladores, cada uno con su propia lógica de operación o integrados con otros procesos de planta o laboratorio.

La arquitectura de programación se estructuró mediante la ejecución de múltiples programas en una tarea cíclica (TASK) con un tiempo de ciclo de 10 ms. Cada programa cumple objetivos funcionales específicos o simplifica tareas comunes requeridas por el resto. Finalmente, estos programas se apoyan en bloques de funciones (Function Blocks) que permiten modularizar las operaciones básicas necesarias para implementar la lógica de control completa, particularmente en el PLC.

Se utilizarán conceptos abordados en IEC 61131-3 y sección B.4. Se recuerda el abreviado de los siguientes lenguajes.

- **Structured Text (ST)**
- **Function Block Diagram (FBD)**
- **Continuous Function Chart (CFC)**

Detallaremos a continuación las diferentes secciones en el programa que definen el funcionamiento del equipo, el lenguaje utilizado y su función.

Se comentará tanto la programación de modos de funcionamiento de los componentes como los protocolos de comunicación y la visualización.

3.1.1. Asociación de variables globales y mapeo de I/O

En la sección 4.3 se explica cómo se organiza la programación, la cual trabaja con variables globales y, posteriormente, se asocia de manera efectiva a salidas a nivel de hardware o bus de campo. Esto se realiza mediante el *mapeo de I/O*, concepto que consiste en vincular variables internas del programa con direcciones físicas de entrada y salida del PLC.

En los módulos de E/S físicos directamente conectados al PLC, el mapeo es generalmente claro, ya que el hardware es totalmente compatible y la asignación de direcciones es fija o determinada por la posición física del módulo. Sin embargo, cuando se trata de dispositivos conectados por buses de campo, la situación es diferente: el mapeo depende de la configuración establecida en los *drivers* de comunicación.

En CODESYS, los *drivers* representan tanto dispositivos físicos como protocolos de comunicación, y suelen disponer de parámetros configurables que determinan el intercambio de datos. El programa en el PLC ejecuta las tareas definidas por el usuario y, posteriormente, establece la comunicación con los *drivers* asociados a las entradas y salidas físicas o a los dispositivos de bus de campo. Este proceso actualiza la memoria de la imagen de proceso, compuesta por áreas de entradas (%I) y salidas (%Q), pudiendo reflejar los cambios en el peor de los casos en dos ciclos de reloj.

3.1.2. CODESYS Runtime

El **CODESYS Runtime** es el componente de software encargado de ejecutar el programa de control en un dispositivo objetivo (PLC, panel HMI con control integrado, PC industrial, etc.). Este entorno de ejecución implementa la especificación de la norma IEC 61131-3, permitiendo interpretar y ejecutar programas desarrollados en el entorno de programación CODESYS.

El *runtime* cumple funciones clave:

1. **Gestión del ciclo de ejecución:** controla el bucle de lectura de entradas, procesamiento lógico y escritura de salidas (*scan cycle*).
2. **Gestión de la imagen de proceso:** mantiene en memoria las variables de entradas y salidas, actualizadas por los *drivers* de hardware y comunicación.
3. **Interacción con drivers:** actúa como intermediario entre el código de aplicación y los controladores de hardware o protocolos de comunicación (Modbus, Profinet, EtherCAT, etc.).
4. **Gestión de tareas:** permite la ejecución de múltiples tareas cíclicas o eventuales con diferentes prioridades y ciclos de repetición independientes.
5. **Servicios del sistema:** proporciona funciones de comunicación remota, diagnóstico, trazado de variables, actualización de firmware y depuración en línea.

El *runtime* se instala en el dispositivo objetivo mediante una imagen de firmware específica para el hardware. Una vez cargado, el PLC se comporta como un sistema programable compatible con CODESYS, capaz de recibir aplicaciones desde el entorno de desarrollo (*CODESYS Development System*) y ejecutarlas de forma autónoma, interactuando con módulos de E/S locales y dispositivos remotos a través de los buses de campo configurados.

En este proyecto, la programación emplea *Structured Text* para asignar valores de variables globales —utilizadas en el resto del sistema— a un mapeo de salidas correspondiente, ya sea a hardware físico o a palabras intercambiadas por protocolos de comunicación. Este mapeo de salidas se almacena en la memoria de tipo Q, tal como se aprecia en la figura 3.1.

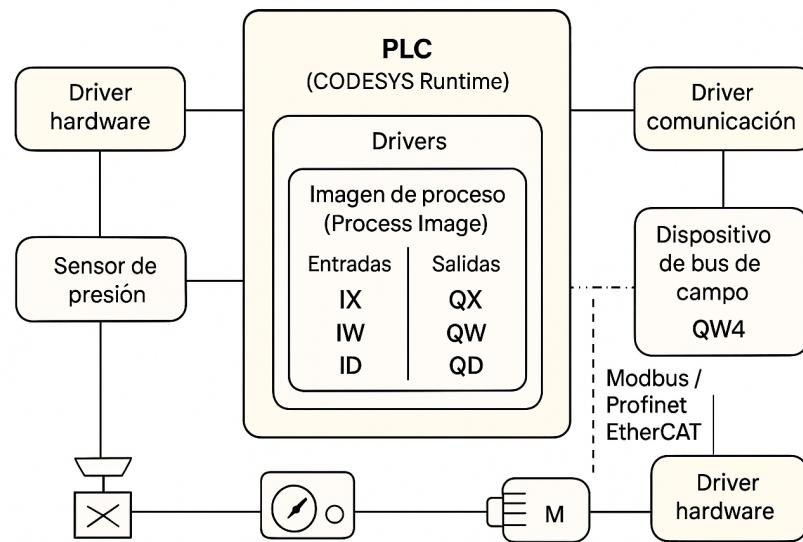


Figura 3.1: Esquema de mapeo de variables globales hacia salidas físicas y de bus de campo en CODESYS.

3.2. Esquema de comunicación Modbus TCP

En la Figura 3.2 se presenta el esquema de comunicación Modbus TCP implementado entre el PLC y la tarjeta Raspberry Pi. Ambos dispositivos ejecutan simultáneamente un **servidor Modbus TCP (Slave)** y un **cliente Modbus TCP (Master)**, lo cual permite una comunicación bidireccional y simétrica.

En el caso del **PLC**, las variables globales internas del sistema se asocian al programa *Modbus TCP Master*, que se encarga de realizar la lectura de registros en el servidor de la Raspberry Pi. De manera complementaria, el PLC también dispone de un *servidor Modbus TCP (Slave)* donde otros dispositivos pueden leer o escribir registros, permitiendo así exponer sus variables globales al exterior.

Por su parte, la **Raspberry Pi** integra un programa *Modbus TCP Master* que lee registros desde el servidor del PLC, además de un programa *Modbus TCP Slave* que enlaza sus variables globales al servidor Modbus TCP local. De esta forma, el PLC puede escribir o leer datos directamente en la Raspberry mediante la interfaz Modbus.

El flujo de datos ilustrado en la figura muestra que:

- El **PLC Master** realiza lecturas en el **servidor Modbus TCP de la Raspberry**.
- La **Raspberry Master** realiza lecturas en el **servidor Modbus TCP del PLC**.
- Ambos servidores permiten además la escritura de registros desde los clientes.

Este diseño garantiza un mecanismo de intercambio de información confiable, donde cada dispositivo expone sus variables internas y, al mismo tiempo, tiene la capacidad de acceder a las variables del otro.

En la Figura 3.2 se presenta un esquema ilustrativo de la interacción entre estos programas y el flujo de comunicación establecido.

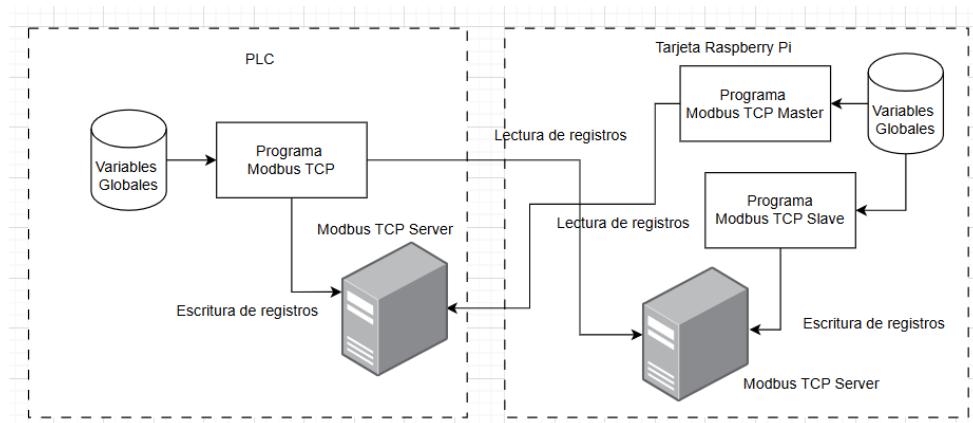


Figura 3.2: Interacción de programas y servidores en dispositivos

3.3. Software en PLC

En PLC PGM o la tarea principal, tenemos corriendo de manera cíclica cada 10 ms los siguientes programas representados en el diagrama de flujo. Se considera de mayor valor por parte del grupo de estudiantes, para el perfil de carrera de ambos, integrar un PLC como controlador del proceso.

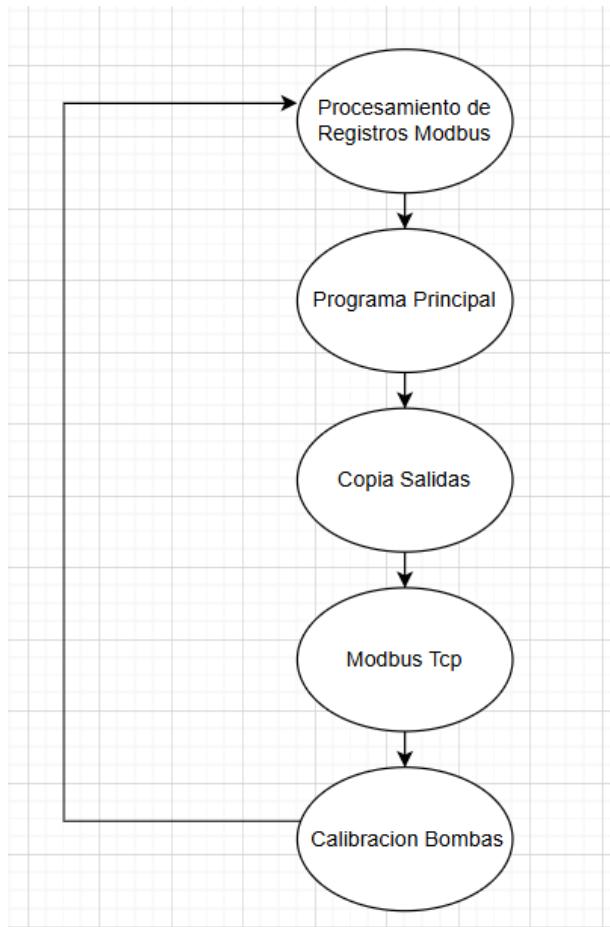


Figura 3.3: Diagrama de flujo en tarea principal

Como iniciativa y buena práctica, es importante contar con varios programas que permitan identificar y resolver problemas en diferentes niveles.

A continuación, se detallan los objetivos principales de cada uno, las soluciones que representan y otras utilidades que aportan al sistema:

- **Programa Procesamiento Registros Modbus (ST)** Procesa los registros Modbus de los drivers de bombas, gestionando palabras de 16/32 bits y aplicando funciones como SWAP. Permite configurar parámetros como esclavo, baudrate y RPM, y documenta los registros más relevantes (Start-Stop, Actual Speed, Operating Speed, Max Speed).
- **Programa Principal (CFC)** Coordina todo el sistema en modo automático y manual, organizando las etapas del proceso (dosificación, medida y lavado). Utiliza bloques SEL y banderas internas para controlar bombas y electroválvulas.
- **Programa Copia Salidas (ST)** Mapea variables globales a salidas físicas (digitales, analógicas o Modbus). Facilita la depuración y asegura la correcta transmisión de consignas hacia el hardware.
- **Programa Modbus TCP (ST)** (Apéndice B.3.1) Escribe variables globales del PLC en memoria del tipo % MW tambien llamadas marcas. Donde las mismas se corresponden a un registro que se puede consultar al servidor del PLC Modbus. También pasa de memoria tipo % IW correspondiente a registros holding leídos de la Raspberry Pi a variables globales en el PLC. En sus inicios Modbus fue el primer protocolo implementado por la empresa Modicon en formato serial. Con el paso del tiempo se paso maestro esclavo a un formato cliente servidor. En este protocolo, es equivalente dirigirse como cliente a maestro y servidor a esclavo. Con el avance de las telecomunicaciones se paso de serial al encapsulado TCP-IP bien conocido en redes de datos. Se obtienen mayores libertades se puede usar mas de un maestro y mas de un esclavo en una red, obteniendo la equivalencia cliente-servidor. Para la comodidad de la configuración de estos servidores Modbus en ambos controladores a nivel de Codesys, se tiene la implementación de estos programas.

3.4. Software en Raspberry PI

En la sección 3.1.2 se explica el concepto de *Runtime* para controladores como pueden ser el PLC o la tarjeta Raspberry Pi. Se llevó a cabo el proceso de instalación y puesta en marcha del *Runtime* en la Raspberry Pi, junto con la configuración adecuada del proyecto en *CODESYS*. Asimismo, se realizaron ajustes en el sistema de la Raspberry, tales como la habilitación del servidor web, la asignación de la dirección IP y la activación del puerto I2C, entre otros.

Para poner en marcha la tarjeta Raspberry Pi como dispositivo CODESYS. Se resumen los siguientes pasos.

1. Instalar el sistema operativo Raspberry en una micro sd que se inserta dentro del dispositivo.
2. Configurar la Raspberry pi 3b+, asignacion de IP, habilitar puerto I2C, etc.
3. Instalar paquete “CODESYS Control for Raspberry pi SL” en CODESYS.

En la sección 4.3 se presentan las configuraciones de diversos parámetros en los *drivers* asociados al uso del bus de campo Modbus, tanto en su variante RTU (serial) como en IP a través del puerto Ethernet. En este contexto, también se identifica un objeto (*driver*) de *CODESYS* asociado al puerto I2C, el cual dispone de un número limitado de parámetros de configuración y la capacidad de asociar un *proceso*. Dicho proceso consiste en invocar al *driver* mediante una extensión *read* o *write* en un punto específico de la programación, con el fin de ejecutar una operación a nivel de hardware.

Como se mencionó previamente, la tarjeta Raspberry Pi se encarga de la comunicación con el sensor óptico mediante el protocolo I2C, a través de los *drivers* de comunicación y eventos asociados. Asimismo, recibe órdenes desde el PLC a través de Modbus TCP, tales como la instrucción de registrar una toma de medida. Posteriormente, el valor obtenido se almacena en la estructura de datos presentada en la visualización y se genera una notificación también vía Modbus TCP. De esta forma, la Raspberry cumple la función de articulador entre el usuario y uno o varios sensores integrados en un sistema de este tipo, orientado a aplicaciones de control y monitoreo.

En el mismo ambiente de programación CODESYS se tiene asociada una tarea que se ejecuta cada 10 ms donde se organiza el código en los programas. Se presenta en la siguiente figura un diagrama de flujo con los programas.

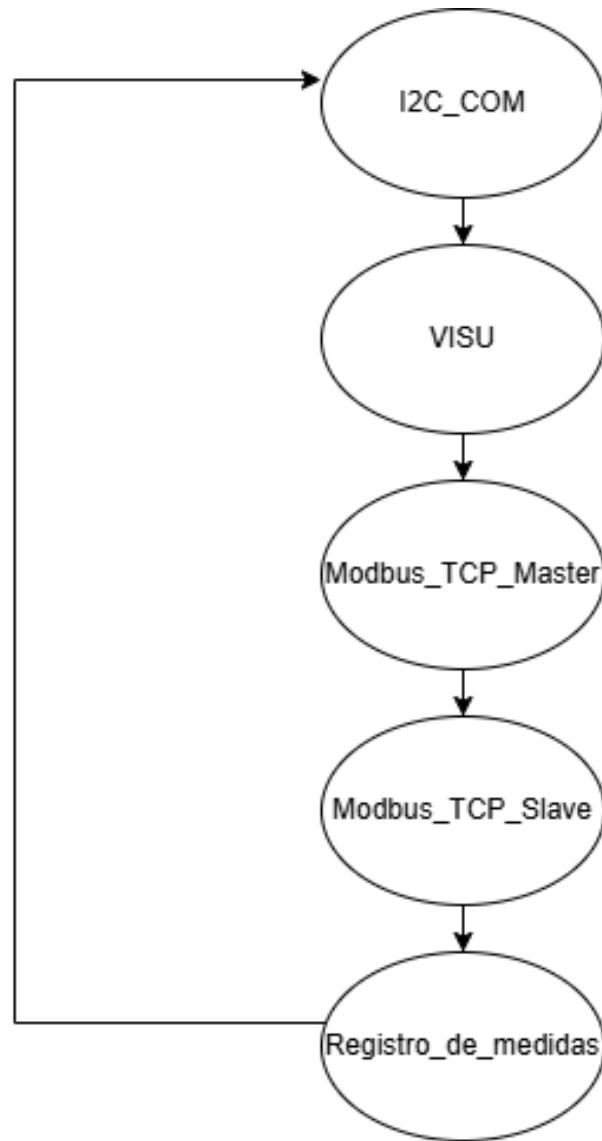


Figura 3.4: Diagrama de flujo en tarea principal

Capítulo 4

Operación y calibración del dispositivo

4.1. Ensayo de flujo de bombas

Para calibrar correctamente el flujo generado por las bombas no fue suficiente con seguir la gráfica de flujo proporcionada por el fabricante kamoer. Decidimos realizar un estudio en detalle de la cantidad de líquido entregado por las bombas en periodos particulares de tiempo y rpm específicas. Para ello solicitamos acceso al Instituto de Química dentro de la Facultad de Ingeniería, allí hicimos uso de balanzas de precisión que nos permitieron identificar el flujo de las bombas, aplicando que un litro de agua pesa 1 kg.

Cabe resaltar que debido a la tensión superficial del agua y el grosor de los caños que salen de las bombas, no es posible emitir cantidades de agua menores a una gota. Para verificar la menor cantidad de reactivo que se puede utilizar, se realizó un primer ensayo. El mismo se observa en la (tabla 4.1)

Num. de medida	RPM configurada	Tiempo configurado (s)	Peso medido (g)	Volumen equivalente (ml)
1	1	20	0,05	0,05
2	1	20	0,05	0,05
3	1	20	0,05	0,05

Cuadro 4.1: Datos relevados en el ensayo 1

Previamente, por el fabricante se conoce la curva de la figura 2.18. Se toma la primera hipótesis que forma parte de nuestro modelo. Para un valor de rpm menor a 100.

$$Q = K_1 \cdot \text{rpm}$$

Se quiere mantener un caudal constante y activar la bomba un tiempo programado en

función del volumen requerido. Despreciando el tiempo en llegar a régimen, se obtiene.

$$\frac{V_{mL}}{t_s} = K_1 \cdot \text{rpm}$$

Despejando:

$$V_{mL} = K_1 \cdot \text{rpm} \cdot t_s$$

A distintas consignas de velocidad para la bomba se realizaron los siguientes dos ensayos, con un ajuste por mínimos cuadrados en cada caso. De forma general se desarrolla el análisis de mínimos cuadrados bajo este modelo.

Sean los datos

$$\{(t_i, V_i)\}_{i=1}^N$$

Bajo el modelo

$$V = K_{\text{rpm}} \cdot t$$

Se minimiza la suma de errores al cuadrado:

$$S(K_{\text{rpm}}) = \sum_{i=1}^N (V_i - (K_{\text{rpm}} \cdot t_i))^2.$$

Derivando y anulando:

$$\frac{\partial S}{\partial K_{\text{rpm}}} = -2 \sum_{i=1}^N t_i (V_i - K_{\text{rpm}} \cdot t_i) = 0.$$

Se obtiene

$$K_{\text{rpm}} = \frac{\sum_{i=1}^N t_i V_i}{\sum_{i=1}^N t_i^2}.$$

En la tabla 4.2 se presentan los datos del segundo ensayo.

Num. de medida	RPM configurada	Tiempo configurado (s)	Peso medido (g)	Volumen equivalente (ml)
10	20	1	0,05	0,05
11	20	5	0,27	0,27
12	20	5	0,25	0,25
13	20	5	0,27	0,27
14	20	10	0,49	0,49
15	20	10	0,49	0,49
16	20	10	0,47	0,47
17	20	20	1,02	1,02
18	20	20	0,97	0,97
19	20	20	0,96	0,96

Cuadro 4.2: Datos relevados en el ensayo 2

En la tabla 4.3 se presentan los datos del tercer ensayo.

Num. de medida	RPM configurada	Tiempo configurado (s)	Peso medido (g)	Volumen equivalente (ml)
20	100	20	4,85	4,85
21	100	20	4,91	4,91
22	100	20	4,80	4,80
23	100	40	9,39	9,39
24	100	40	9,88	9,88
25	100	40	9,79	9,79
26	100	60	14,60	14,60
27	100	60	14,66	14,66
28	100	60	14,69	14,69

Cuadro 4.3: Datos relevados en el ensayo 3

Se presenta en la siguiente tabla el error cuadrático medio junto con el valor de la constante del modelo obtenida y el valor de las rpm de consigna para cada ensayo.

Num. de ensayo	RPM del ensayo	ECM obtenido	K_{rpm} (mL/(seg·rpm))
2	20	4,24e-4	0,0498
3	100	1,78e-2	0,243

Cuadro 4.4: Resultados obtenidos

Despejando del modelo ajustado, se obtiene de manera lineal la configuración de los temporizadores en el PLC bajo una consigna de rpm preestablecida. Se tomó como válido el modelo del ensayo dos debido a obtenerse un menor error cuadrático medio. En los ensayos realizados no se tomó o planificó de mejor manera recopilar datos o tomar criterios para trabajar con la incertidumbre de la balanza.

4.2. Ensayos de funcionamiento

En espectroscopia es habitual el término de absorbancia sobre una solución. Es un concepto usado para cuantificar la energía lumínica que absorbe una sustancia cuando un haz de luz pasa a través de la muestra (Apéndice B.2.2).

Se tiene bajo la ley de Beer-Lambert (Apéndice B.2.3) que la absorbancia aumenta linealmente frente a la concentración de un aditivo no translúcido en la solución.

Se espera que la absorbancia obtenida de los datos del sensor OPT3002 crezca linealmente conforme aumenta el porcentaje de colorante en la solución. Se preparó una solución de colorante con agua como reactivo, la cual fue utilizada para ambos ensayos. El primer ensayo se realizó administrando los porcentajes en forma manual e inyectando la solución en la celda de medida. El segundo ensayo se realizó de forma automática por el sistema. Para hallar el valor de las medidas relativas se presenta el siguiente modelo.

Sea de energía lumínica que pasa a través de la mezcla i_i , el porcentaje de solución con colorante c_i , y la medida tomada de agua sin colorante i_0 , se obtiene:

$$Ai = -\log\left(\frac{i_i}{i_0}\right) \quad (4.1)$$

4.2.1. Datos relevados

Para ensayar el funcionamiento del equipo comenzamos por ejecutar cada una de las funciones y los modos. Luego cargamos agua de la canilla en el depósito de agua y como reactivo utilizamos una mezcla de agua y colorante de comida azul.

Realizamos manualmente mezclas entre el colorante y agua, con porcentajes preestablecidos de colorante en la muestra, luego se introdujo manualmente cada una de las mezclas en la cubeta. La tabla 4.5 muestra los valores de absorbancia despejados a partir de la ecuación 4.1.

Porcentaje de colorante	Absorbancia
0	0
20	0.0440
40	0.0856
60	0.1645
100	0.2536

Cuadro 4.5: Relación entre porcentaje de colorante y absorbancia

Como se aprecia en la figura 4.3, los resultados de lecturas expresados en absorbancia en función del porcentaje de reactivo agregado son aproximadamente lineales, esto verifica el correcto funcionamiento del sensor óptico OPT3002.

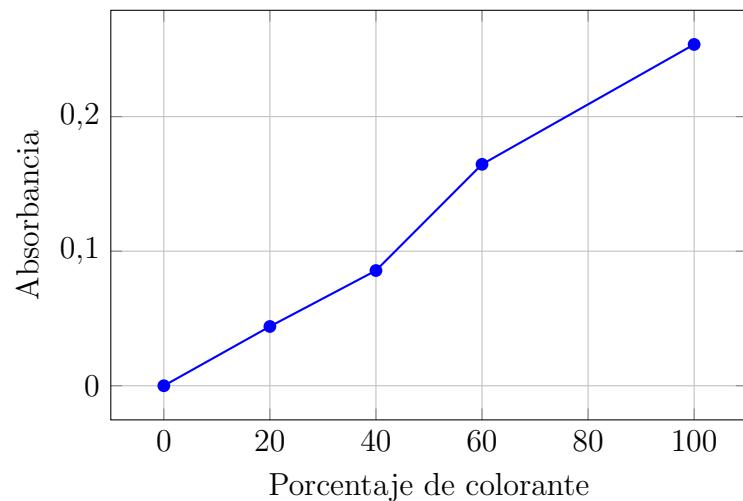


Figura 4.1: Curva de absorbancia en función de porcentaje de colorante en la solución -
Ensayo de referencia

Por otro lado, realizamos ensayos implementando todo el proceso de dosificación mediante bombas peristálticas y medida que caracteriza al modo automático. Los resultados adquiridos y la gráfica lineal de estos resultados se muestra en la tabla 4.6 y la figura 4.2. Consideraremos que también se mantiene una linealidad en estos casos, verificando el correcto funcionamiento del proceso.

Porcentaje de colorante	Absorbancia
0	0
10	0,0389
20	0,0547
30	0,0866
40	0,1108

Cuadro 4.6: Relación entre porcentaje de colorante y absorbancia

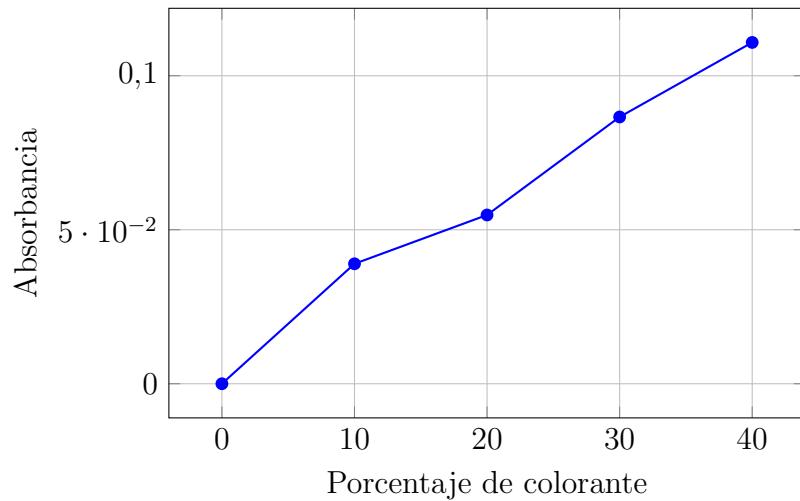


Figura 4.2: Curva de absorbancia en función de porcentaje de colorante en la solución - Prueba con prototipo

Obtenidos los valores de absorbancia A_i bajo un porcentaje de concentración de solución de agua con colorante dosificada c_i , se realiza un ajuste lineal por mínimos cuadrados. Bajo el siguiente modelo.

$$A_i = K c_i$$

Se obtiene el ajuste para cada ensayo.

$$K = \frac{\sum_{i=1}^5 c_i A_i}{\sum_{i=1}^5 c_i^2}.$$

También se calcula el error cuadrático medio en cada uno.

Num. de ensayo	ECM obtenido	K
1	0,195	2,53e-3
2	0,00532	2,84e-3

Cuadro 4.7: Resultados obtenidos

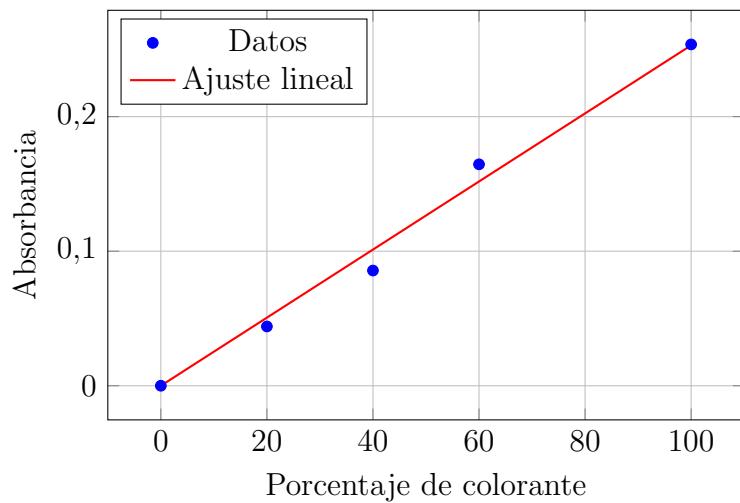


Figura 4.3: Curva de absorbancia en función de porcentaje de colorante en la solución con ajuste lineal - Ensayo de referencia

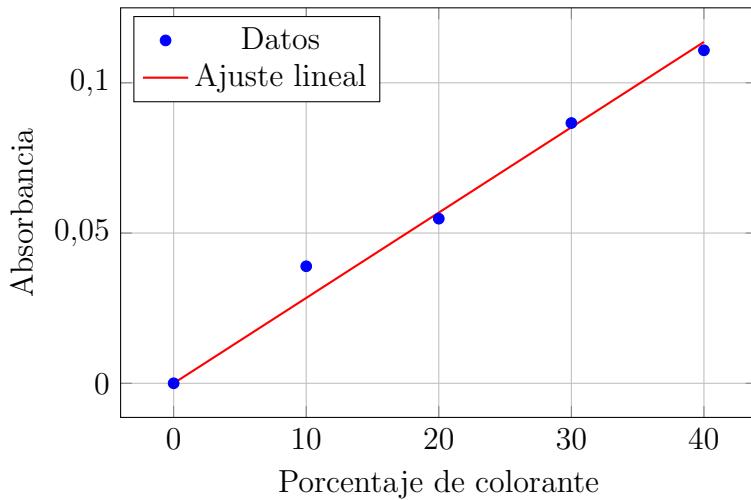


Figura 4.4: Curva de absorbancia en función de porcentaje de colorante en la solución con ajuste lineal - Prueba con prototipo

Como era de esperarse, el sistema en forma automática obtiene un error cuadrático medio menor frente al modelo establecido. También se debe tomar en cuenta que el ensayo en forma manual fue realizado por los estudiantes, sin disposición de instrumentación adecuada y poca formación en ensayos similares. Funcionando de forma automática con mayor confiabilidad que el ensayo realizado en forma doméstica, sin ambiente de laboratorio ni experiencia en tareas similares.

Capítulo 5

Conclusiones y trabajos a futuro

5.1. Conclusiones de proyecto

Se trabajó con actuadores y sensores previamente seleccionados en investigaciones realizadas por docentes del IIE, docentes de la Facultad de Química y la empresa Spymovil. A lo largo del proyecto se alcanzó una sólida experiencia en la integración de los actuadores al sistema diseñado, así como en el estudio de los procedimientos necesarios para llevar a cabo la calibración.

Construimos un equipo robusto, diseñado para instalarse dentro de tableros eléctricos en entornos industriales y adaptable a distintas instalaciones para aplicaciones de este tipo.

Al inicio de este proyecto, identificamos como principales criterios de éxito:

1. Lograr las funciones básicas de aislar la muestra y dispensar el reactivo acorde a las instrucciones del usuario.
2. El dispositivo debe guardar las medidas realizadas para luego ser visualizadas por el usuario.
3. El dispositivo se comunicará con el usuario, el cual podrá extraer datos de medidas realizadas por el equipo y configurar parámetros de medición si el equipo no está ejecutando una medida.

Consideramos que fueron logrados ampliamente los objetivos de proyecto, no solo construimos un dispositivo que cumple con las funciones requeridas por el proceso de ensayo automatizado. Sino que lo realizamos utilizando componentes aplicables a procesos industriales que cumplen con requerimientos de precisión a nivel de mercado.

El prototipo final puede ser controlado por un usuario sin preparación extensa ni conocimientos técnicos sobre el dispositivo, solo se requeriría el manual de usuario presentado en la sección 2.2.

También, como se presenta en la sección 5.2, los elementos, protocolos de comunicación y métodos de programación implementados permiten una continua evolución del prototipo en un futuro.

Concluimos que mediante la aplicación de conceptos como protocolos de comunicación, conexión e instalación de hardware auxiliar, métodos de instalación física de componentes en el ámbito industrial, programación de dispositivos de control, selección de componentes adecuados, registros técnicos y el estudio de las funciones requeridas por un producto que se desea presentar a un cliente en el mercado; logramos construir un dispositivo final completamente funcional y esperamos que en un futuro pueda ser utilizado profesionalmente.

5.2. Posibles futuras mejoras

Una de las principales ventajas del software y hardware implementado en nuestro proyecto, es la facilidad que proporciona para realizar mejoras y actualizaciones sobre el prototipo final. Presentaremos a continuación áreas específicas sobre las cuales consideramos que se podrían incorporar avances en un futuro.

5.2.1. Modos de operación

La programación de los controladores en CODESYS, nos permite modificar y ajustar los modos de funcionamiento y el proceso de toma de medida. En caso de requerirse un protocolo particular de ensayo, el código podría modificarse para implementarlo.

5.2.2. Visualización

Consideramos que el prototipo final proporciona una interfaz de usuario fácilmente accesible e intuitiva; sin embargo, en un futuro podría implementarse una pantalla HMI (Human Machine Interface). Estas pantallas táctiles permitirían el control del proceso de dosificación y medida, desde el exterior de la tapa del tablero donde se instaló el equipo. Evitaríamos así la necesidad de conectar a una LAN privada para el control y visualización del menú, el equipo quedaría operativo con solamente una conexión a tensión y sería controlable sin la necesidad de apertura de la cubierta del tablero (figura 5.1).



Figura 5.1: Ejemplo de tablero industrial con control mediante HMI

5.2.3. Expansión de la red de medida y muestreo

Una ventaja de los protocolos de comunicación ModBus e I2C, es la posibilidad de integrar nuevos dispositivos controlables a la red, mediante la configuración del ID y la dirección respectivamente. La conexión de estos dispositivos se puede realizar mediante los mismos puertos de comunicación tal como se visualiza en el Unifilar 02 (Sección E4), donde los dos drivers que manejan las bombas, se comunican con el PLC mediante las mismas entradas A y B. La misma configuración, se puede implementar a la comunicación I2C entre el sensor y la Raspberry pi, donde se comparten las entradas SDA (Serial Data Line) y SCL (Serial Clock Line).

Esto permitiría controlar varios procesos de dosificación y medida desde una única combinación de PLC y Raspberry, integrando juegos de dos bombas y un sensor.

También podríamos agregar diferentes opciones de reactivos, integrando una bomba por cada uno de los químicos que se desee agregar al ensayo.

5.2.4. Control a distancia

La *Visualization* en CODESYS permite controlar el prototipo mediante el acceso a una dirección específica desde el navegador de una computadora conectada a la misma red LAN privada. Mediante el uso de un router es posible extender la red para habilitar accesos remotos. Asimismo, el sistema puede integrarse con plataformas SCADA o con soluciones de visualización basadas en tecnologías 5.0, como es el caso de Corvina Cloud. Esta es una plataforma en la nube de Exor International que permite conectarse de forma remota a PLCs, HMIs y otros dispositivos industriales.

5.2.5. Integración en industria o laboratorio

La utilización de protocolos de comunicación normalizados en redes de control industriales brinda la posibilidad de integrar el prototipo en instalaciones que ya cuenten con automatización, tableros de control u otros procesos que no estén vinculados directamente con el monitoreo de la calidad del agua.

De esta manera, sería posible realizar ensayos cuyos resultados, en caso de alcanzar valores específicos, activen alarmas en redes previamente configuradas para el control de una planta en particular. Asimismo, desde estas mismas redes podría emitirse órdenes de control y registrarse los resultados en una base de datos unificada.

Para lograr integrarlo en una aplicación específica se necesitaría trabajar en modificaciones que acompañen a la misma.

Apéndice A

Apéndice A: Planos

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Referencias :	
-DIF	Llave diferencial
-PE	Conexión a tierra

A

A

B

B

C

C

D

D

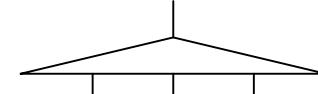
E

E

F

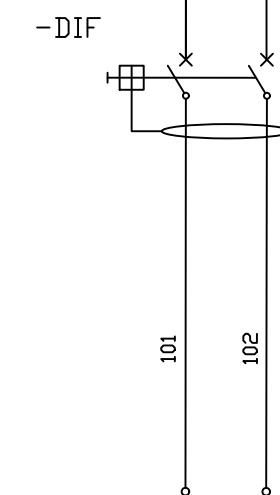
F

Fuente de tensión 230 V AC



001

002



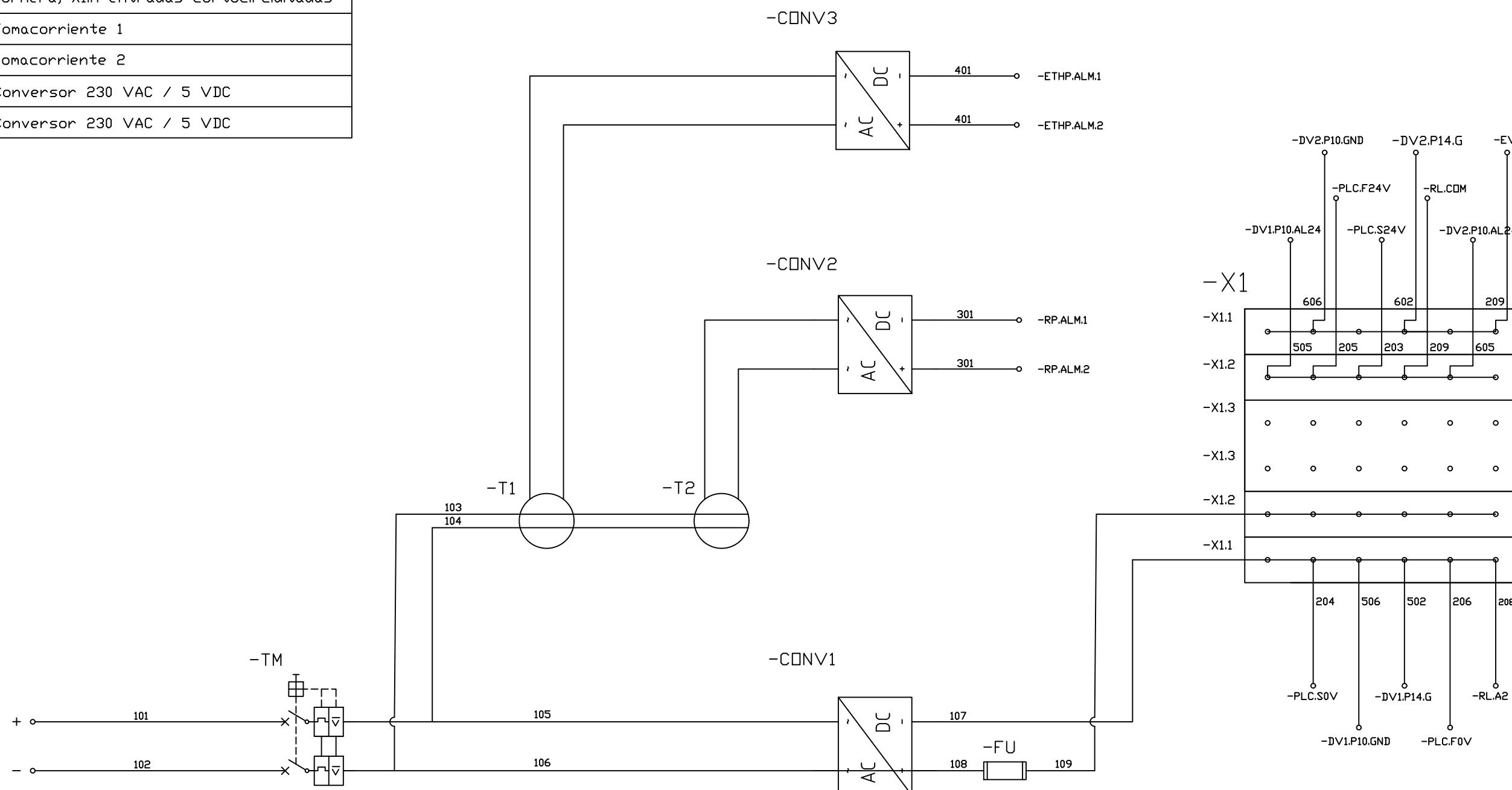
-PE

Titulo: Fuente de alimentación	Campo de aplicación: Este unifilar muestra la conexión a fuente de tensión de 230 V AC.			Fecha Elab. 14/8/25
	Escala NO APLICA	Voltaje de operación: 230V	Num. dibujo: 00	Hoja: A3

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

Referencias pertinentes:	
-TM	Llave termomagnética
-C ^{ONV} 1	Conversor 230 VAC / 24 VDC
-FU	Fusible
-X1	Bornera, X1.n entradas cortocircuitadas
-T1	Tomacorriente 1
-T2	Tomacorriente 2
-C ^{ONV} 2	Conversor 230 VAC / 5 VDC
-C ^{ONV} 3	Conversor 230 VAC / 5 VDC

A							
B							
C							
D							
E							
F							

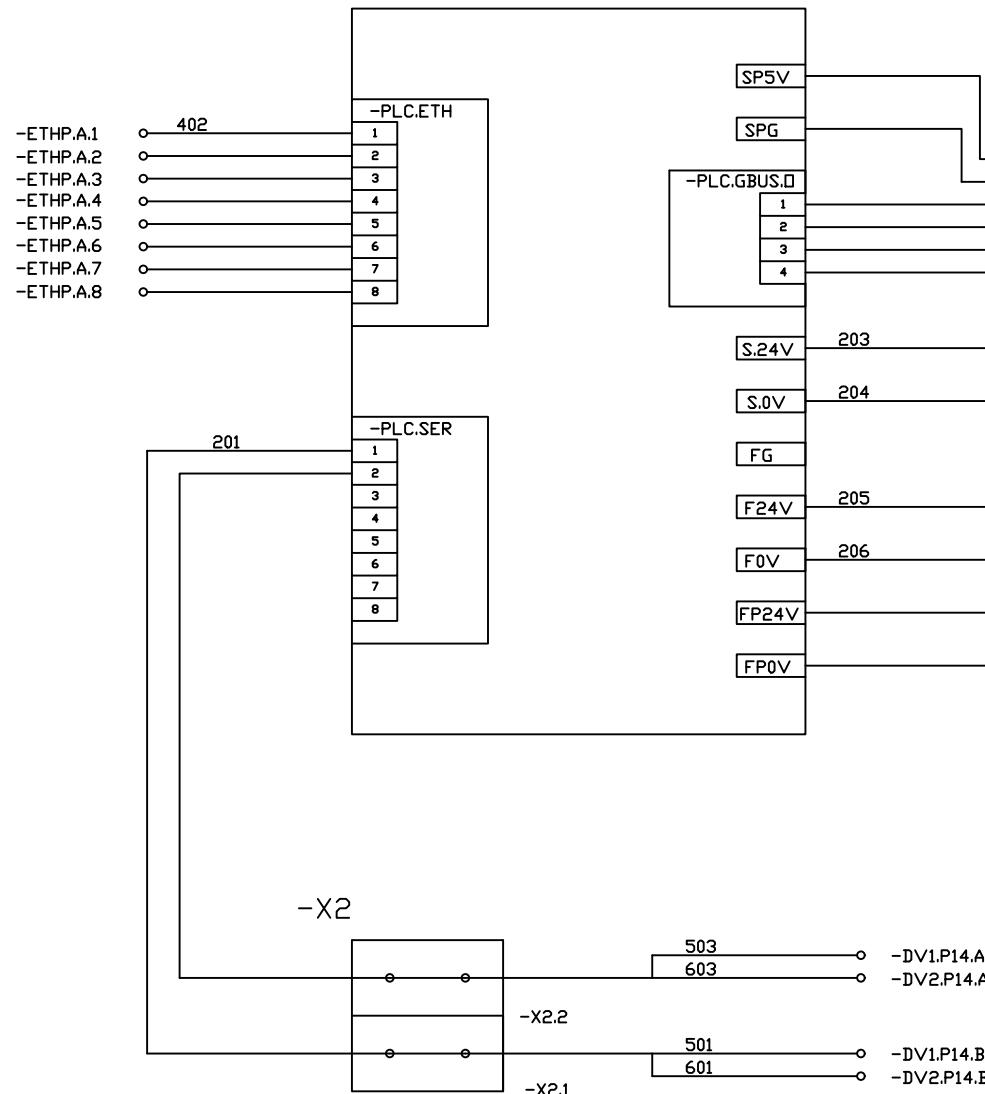


Título: Circuito de alimentación	Campo de aplicación: Este unifilar muestra los diferentes elementos y conexiones usadas para alimentar a los equipos desde la entrada de tensión 230 V AC.	Fecha Elab. 14/08/25
Escala NO APLICA	Voltaje de operación: 230V / 24V / 5V	Modif.
		Modif.
		Modif.
		Hoja: A3

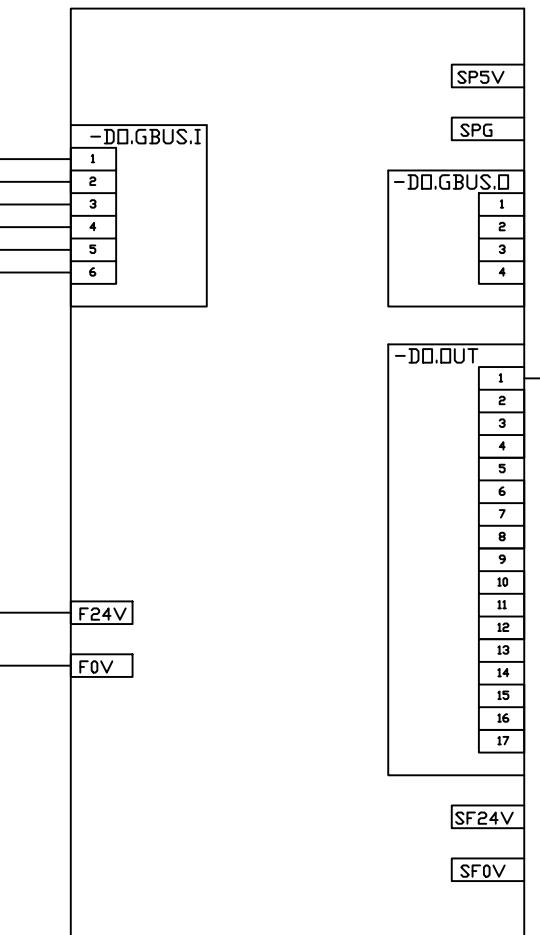
Referencias :	
-PLC	PLC CREVIS GL-9972
-DO	Modulo de salidas digitales
-RL	Rele para alimentar electrovalvula
-X2	Bornera
-EV	Electrovalvula

B

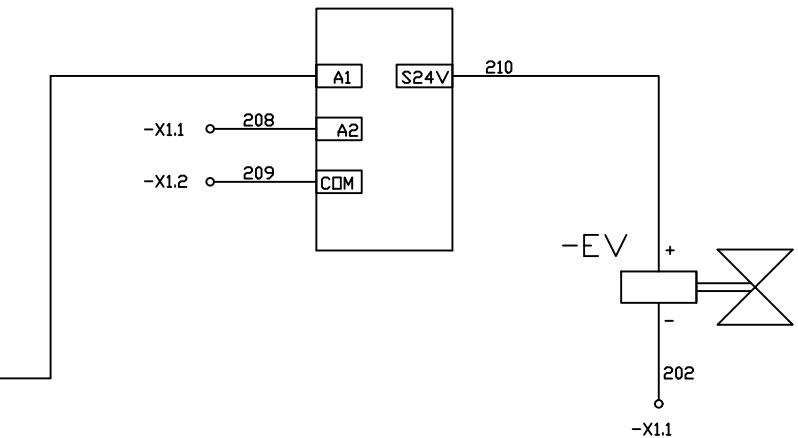
-PLC



-DO



-RL



-EV

Titulo: PLC y componentes inmediatos	Campo de aplicación: Conexiones de PLC con modulo de salidas digitales, electroválvula con su correspondiente relé de comando y bornera 2.	Fecha
		Elab. 14/08/25
		Modif.
		Modif.
		Modif.
Escala NO APLICA	Voltaje de operación: 24V / 5V	Num. dibujo: 02
		Hoja: A3

1	2	3	4	5	6	7	8	
Referencias pertinentes:								
A	-RP	Raspberry Pi 3B+						
	-LED	Luz led para test óptico						
	-PS	Placa para sensor óptico DPT3002EVM						
	-RP.OUT.1	Salida 3.3V de Raspberry						
	-RP.OUT.3	Salida SDA para comunicacion I2C						
B	-RP.OUT.5	Salida SCL para comunicacion I2C						
	-RP.OUT.39	Salida GND de Raspberry						
	-R1	Resistencia Pullup 10 KΩhm						
	-R2	Resistencia Pullup 10 KΩhm						
<p>Diagram illustrating the connections between a Raspberry Pi 3B+ (labeled -RP) and a DPT3002EVM optical sensor board (labeled 302).</p> <p>The connections are as follows:</p> <ul style="list-style-type: none"> Raspberry Pi 3B+ (RP): <ul style="list-style-type: none"> Pins 1-4 connect to -RP.USB. Pins 5-8 connect to -RP.ETH. Pins 1-2 connect to -RP.ALM. Pins 301 (GND) and 302 (VDD) connect to the DPT3002EVM board. Pins 303 (VOUT), 304 (GND), 305 (SDA), 306 (INT), and 307 (SCK) connect to the DPT3002EVM board. Pins 301 (GND) and 302 (VDD) also connect to resistors R1 and R2, which are connected to ground. Pins 302 (VDD) and 303 (VOUT) are connected to a LED. DPT3002EVM Board (302): <ul style="list-style-type: none"> Pin 1: VDD Pin 2: GND Pin 3: SDA Pin 4: INT Pin 5: SCK Pin 6: GND Pin 7: GND Pin 8: GND Pin 9: GND Pin 10: GND Pin 11: GND Pin 12: GND Pin 13: GND Pin 14: GND Pin 15: GND Pin 16: GND Pin 17: GND Pin 18: GND Pin 19: GND Pin 20: GND Pin 21: GND Pin 22: GND Pin 23: GND Pin 24: GND Pin 25: GND Pin 26: GND Pin 27: GND Pin 28: GND Pin 29: GND Pin 30: GND Pin 31: GND Pin 32: GND Pin 33: GND Pin 34: GND Pin 35: GND Pin 36: GND Pin 37: GND Pin 38: GND Pin 39: GND Pin 40: GND 								
<p>Titulo: Raspberry y sistema óptico</p> <p>Campo de aplicación: Conexiones de raspberry Pi con placa que contiene el sensor de intensidad luminica DPT3002, led generador de luz, alimentacion y PLC para comunicacion MODBUS.</p> <p>Escala N° APLICA</p> <p>Voltaje de operación: 3,3V / 5V</p> <p>Num. dibujo: 03</p> <p>Hoja: A3</p>								

Referencias pertinentes:	
-ETHP	Swtch interconecta 5 puertos ethernet

A

A

B

B

C

C

D

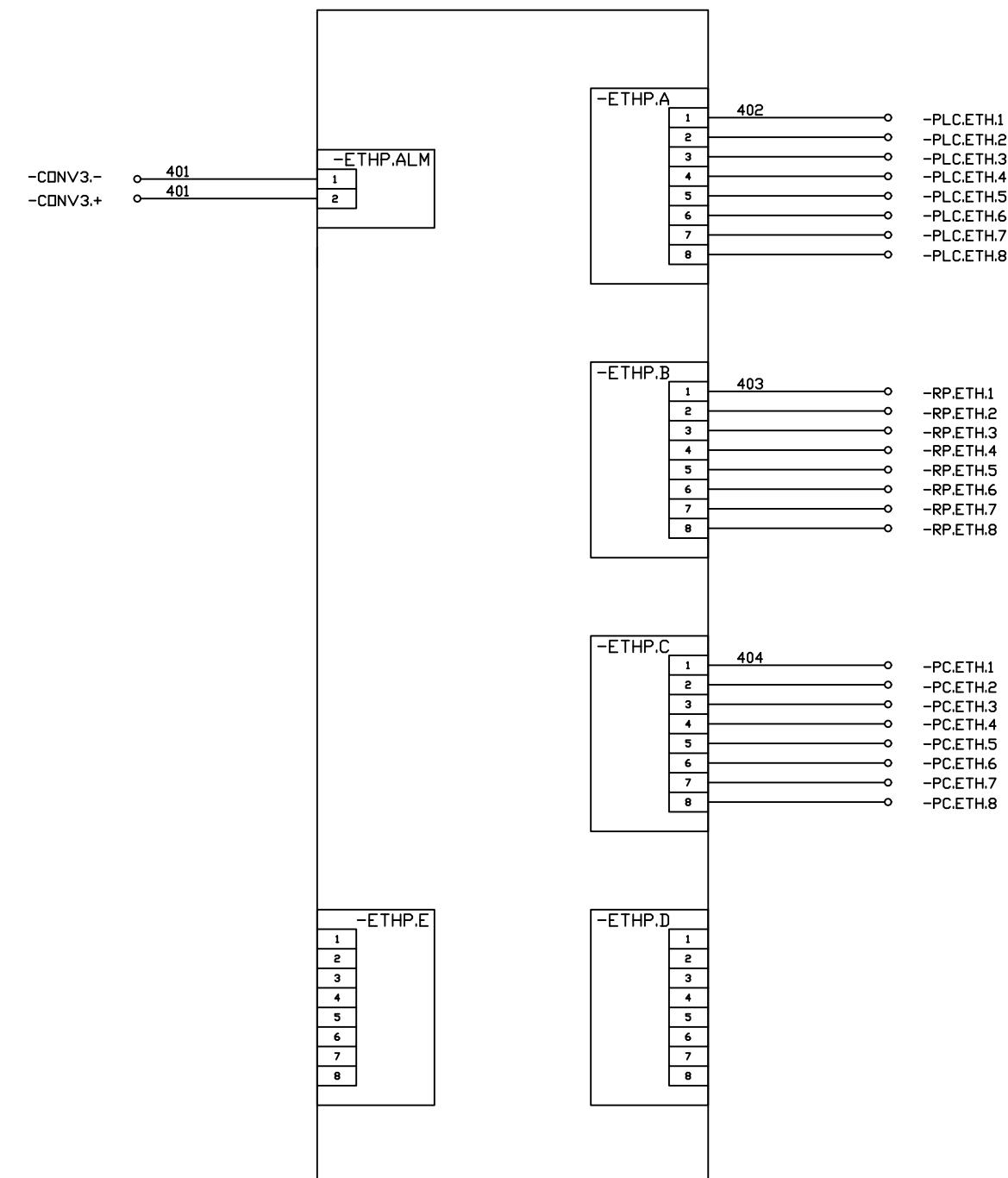
D

E

E

F

F

-ETHP

Titulo: Switch de puertos Ethernet	Campo de aplicación: Conexiones que relacionan al PLC con el equipo Raspberry permitiendo comunicación modbus y con el PC utilizado para el manejo y programación.	Fecha
		Elab. 04/21/25
		Modif.
		Modif.
		Modif.
Escala NO APLICA	Voltaje de operación: 5V	Num. dibujo: 04
		Hoja: A3

Referencias pertinentes:	
-DV1	Driver 1
-B1	Bomba 1

A

A

B

B

C

C

D

D

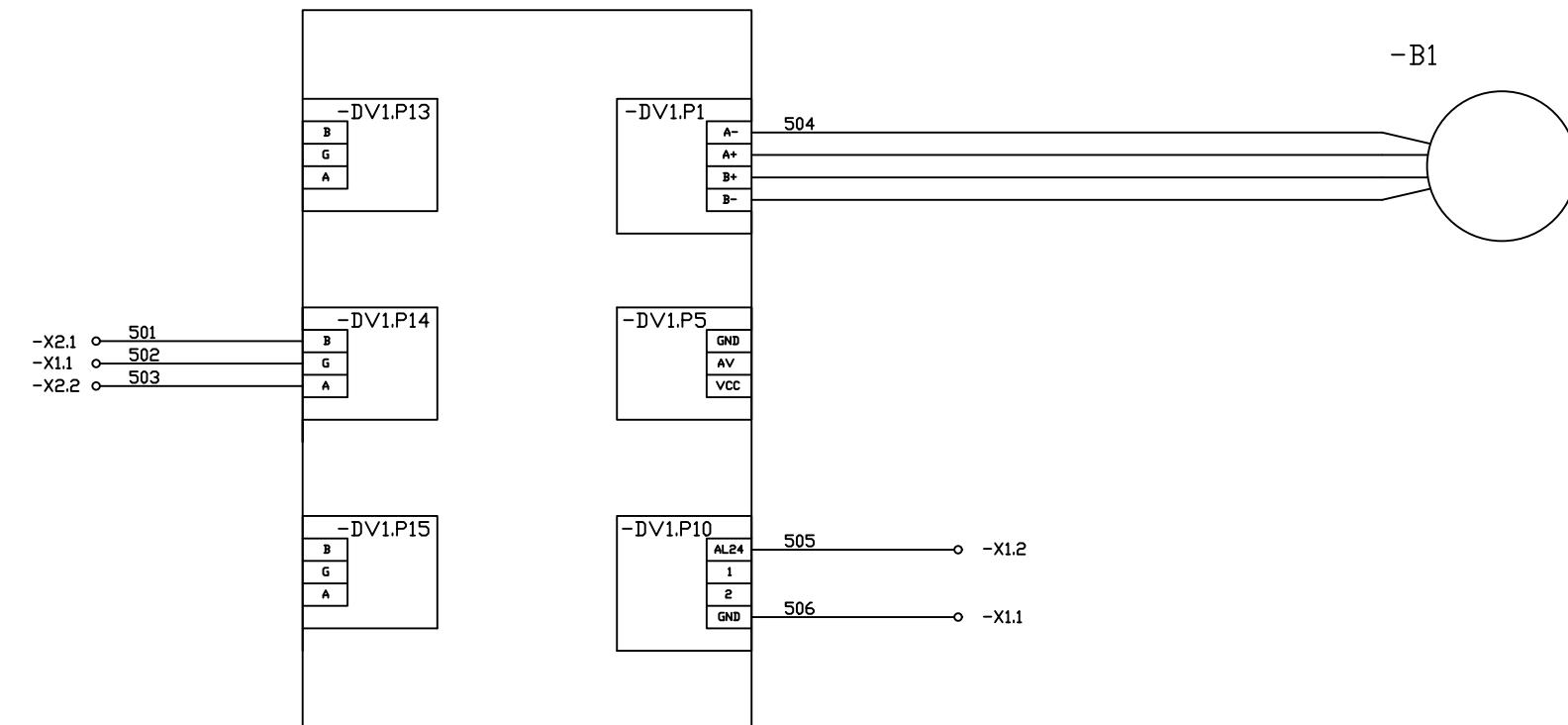
E

E

F

F

-DV1



Título:
Drivers y bombas

Campo de aplicación:
Conexiones entre los drivers utilizados para el control y las bombas peristálticas que dispensan el reactivo.

Escala NO APLICA

Voltaje de operación: 24 V

Num. dibujo: 05

Elab.	14/08/25
Modif.	
Modif.	
Modif.	

1 2 3 4 5 6 7 8

Referencias pertinentes:	
-DV2	Driver 2
-B2	Bomba 2

A

A

B

B

C

C

D

D

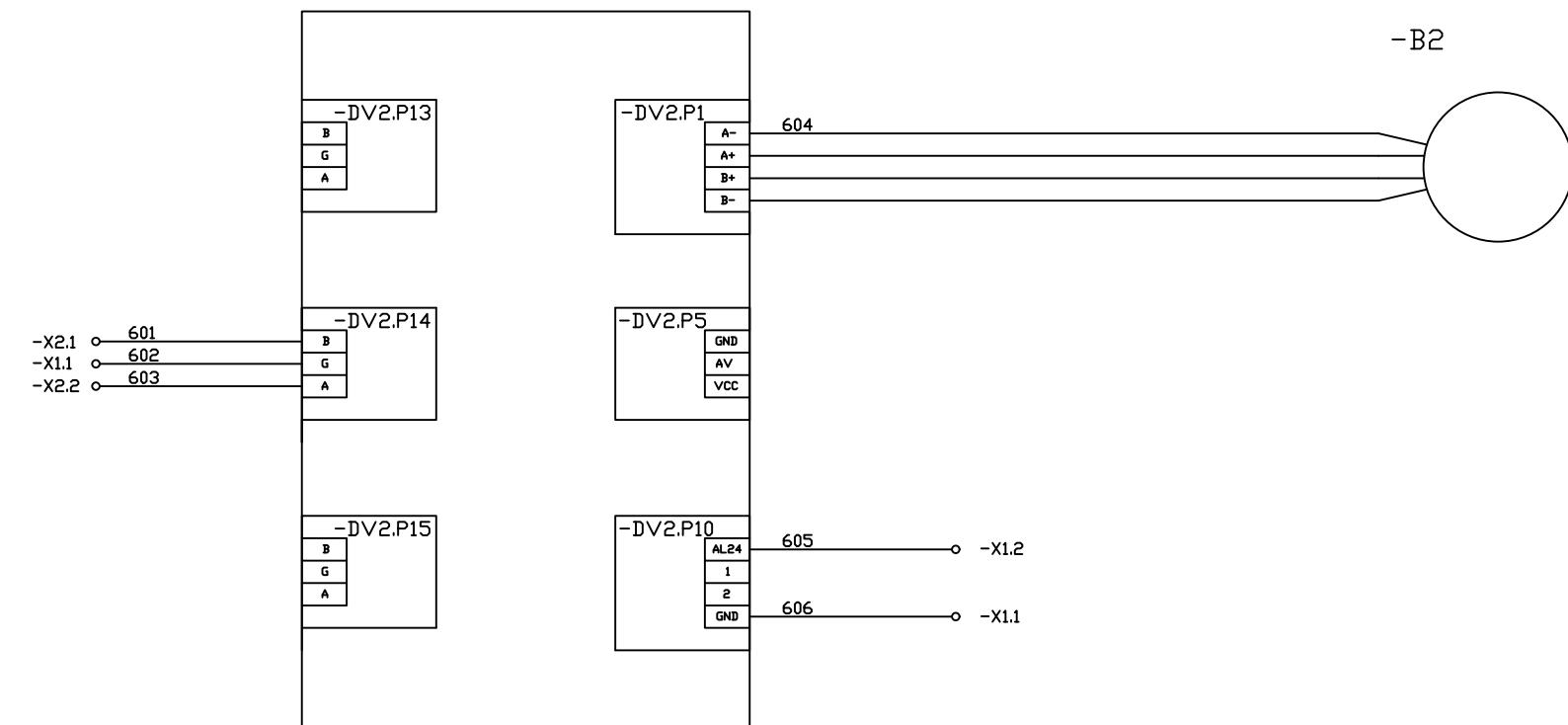
E

E

F

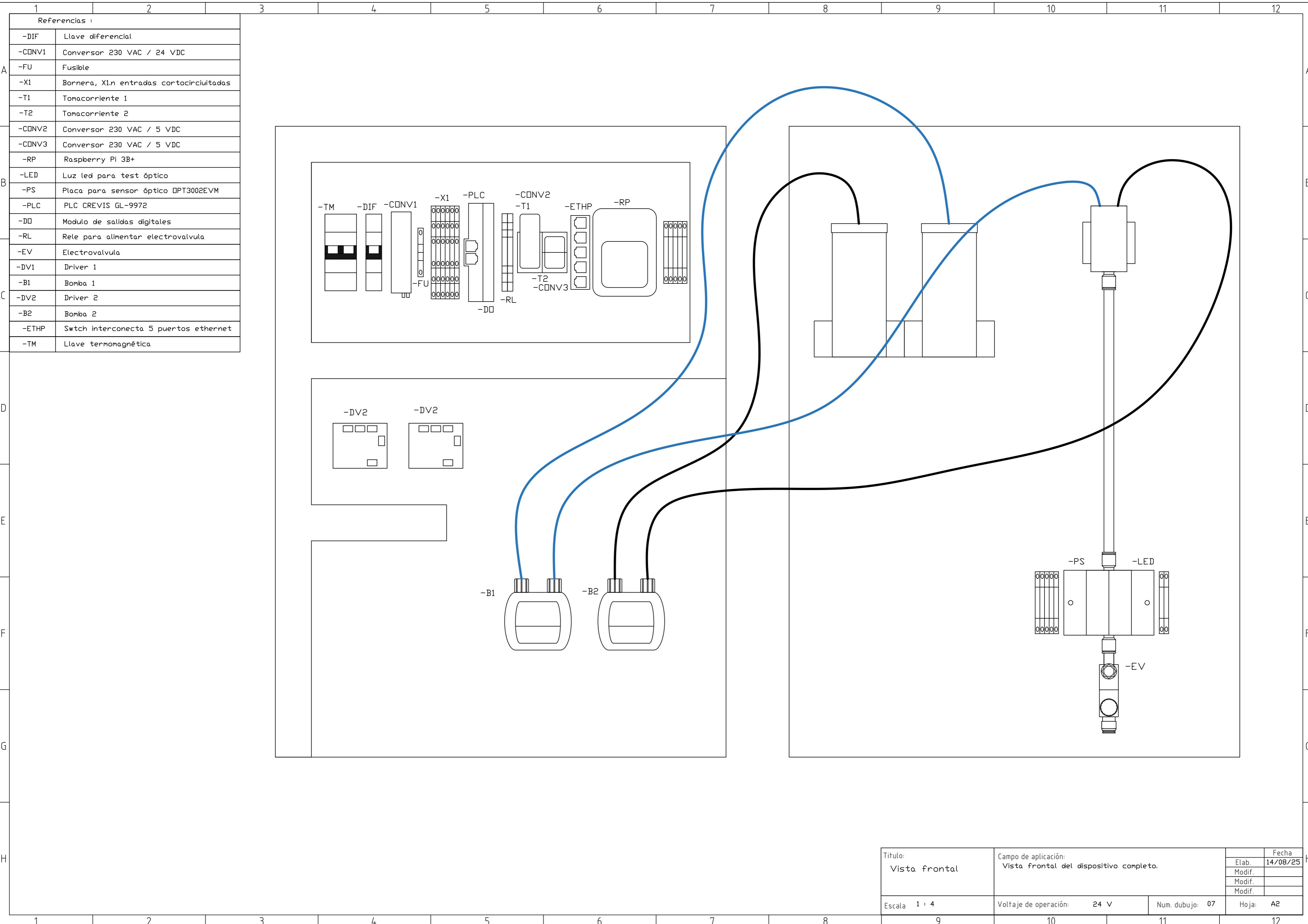
F

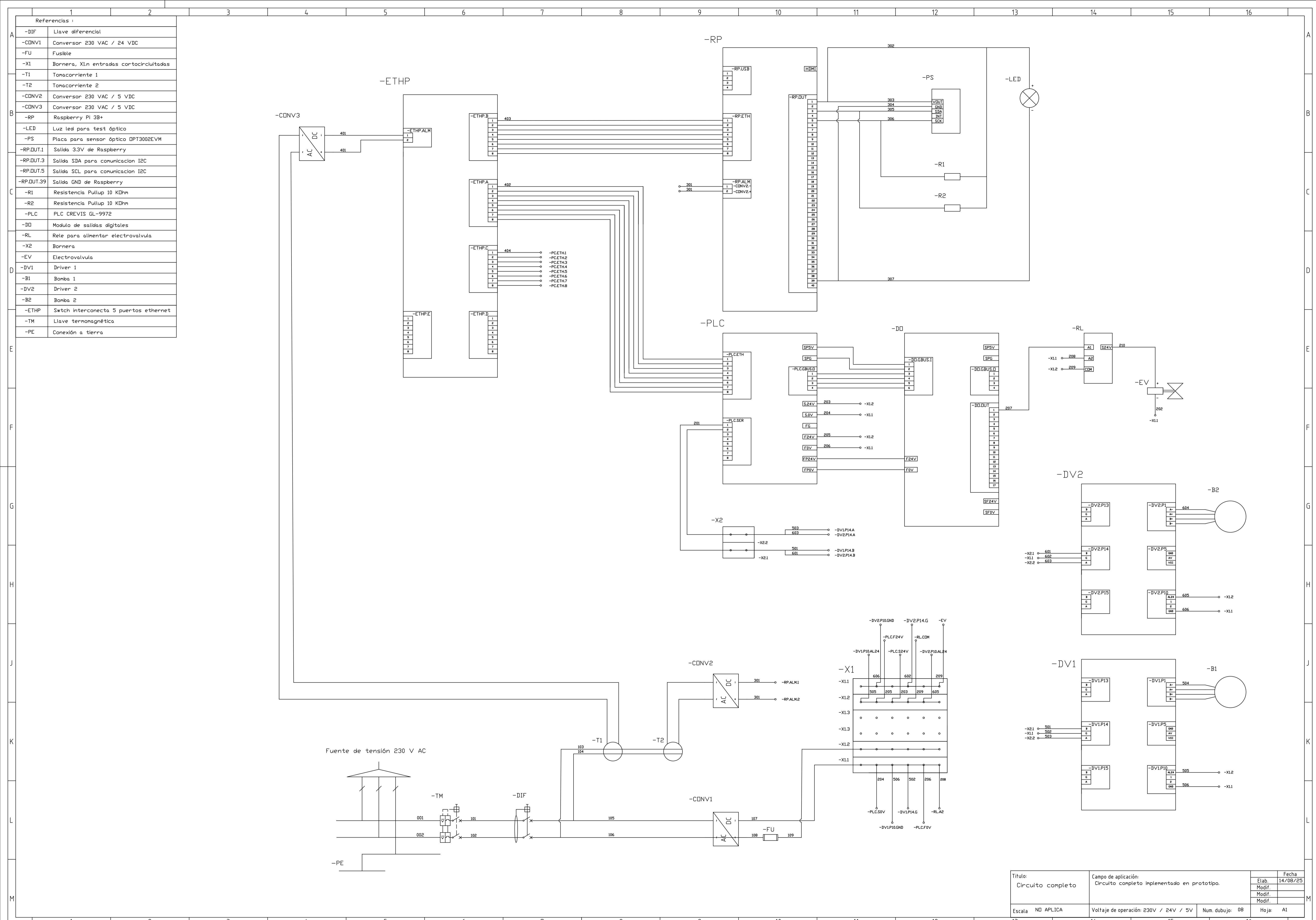
-DV2



1 2 3 4 5 6 7 8

Titulo: Drivers y bombas	Campo de aplicación: Conexiones entre los drivers utilizados para el control y las bombas peristálticas que dispensan la muestra.			Fecha Elab. 14/08/25
	Modif.	Modif.	Modif.	
Escala N° APLICA	Voltaje de operación: 24 V	Num. dibujo: 06	Hoja: A3	





Apéndice B

Apéndice B:Normas y fundamentos

B.1. Motores paso a paso

Un **motor de paso** es un tipo de motor eléctrico que se mueve en pasos discretos en lugar de girar continuamente. El principio de funcionamiento de estos motores, está basado en un estator construido por varios bobinados sobre un material ferromagnético y un rotor que puede girar libremente con respecto al estator.

Estos diferentes bobinados son alimentados uno a continuación del otro y causan un determinado desplazamiento angular que se denomina “paso angular” y es la principal característica del motor. Un paso en un motor paso a paso se define como la precisión que tiene el motor en un giro completo. Por ejemplo, un motor de 200 pasos cuenta con un ángulo de rotación de $1,8^\circ$, siendo este ángulo el paso del motor, producto de dividir 360° entre 200. Este Angulo corresponde a la resolución del motor.

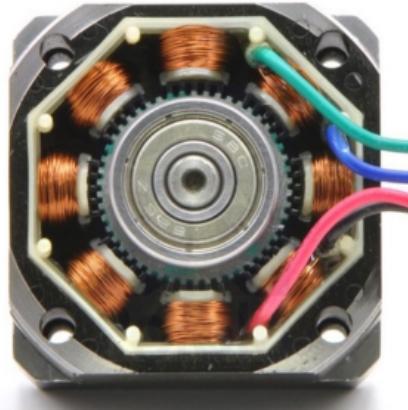


Figura B.1: Bobinado de motor paso a paso

En la figura B.1 se muestra la estructura clásica de un bobinado usado para este tipo de motores, mientras que en la figura B.2 se enseña un modelo de rotor con imán permanente.

Es común que los dientes de polaridad sur estén desfasados de los dientes de polaridad norte, como se muestra también en la figura B.2.

Los hierros de los bobinados estáticos, poseen dientes que se alinean con los dientes rotóricos y los cables de cobre están enhebrados de manera que cuatro de los bobinados generan un campo magnético sobre el rotor, inverso a los cuatro restantes. Cada uno de estos grupos de cuatro bobinas está alimentado por una fuente distinta, por lo que es posible generar el movimiento del rotor alterando la alimentación de los dos grupos de bobinas mediante pulsos y alineando los dientes del estator con los del rotor de polaridad norte y polaridad sur de manera alternada.

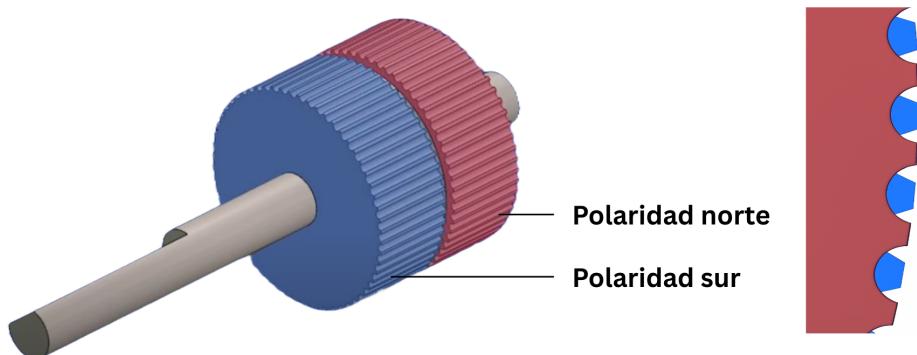


Figura B.2: Modelo del rotor en motor paso a paso

Los motores de paso se dividen principalmente en:

- **Bipolares:** Requieren inversión de corriente en las bobinas para cambiar de estado, poseen 5 o 6 entradas para su control.
- **Unipolares:** Utilizan un punto común en las bobinas para facilitar el control, poseen 4 entradas.

B.2. Espectroscopía

Además del análisis químico, la caracterización de sustancias puras y mezclas también se puede lograr mediante métodos físicos. Entre otras técnicas, tales como la determinación del punto de fusión, el índice de refracción y la densidad, la espectroscopía en el rango ultravioleta y visible (UV/Vis) se utiliza ampliamente en casi todos los segmentos de mercado y lugares de trabajo en investigación, producción y control de calidad para la

clasificación y el estudio de sustancias. La espectroscopía UV/Vis utilizada en este proyecto, se basa en la absorción de luz por la muestra. Dependiendo de la cantidad de luz absorbida por la muestra y su longitud de onda, se puede obtener información valiosa.

B.2.1. Espectroscopía UV/Vis

La espectroscopía se basa en la interacción de la luz con la materia. La figura B.3 ilustra lo que sucede cuando la luz incide en un objeto:

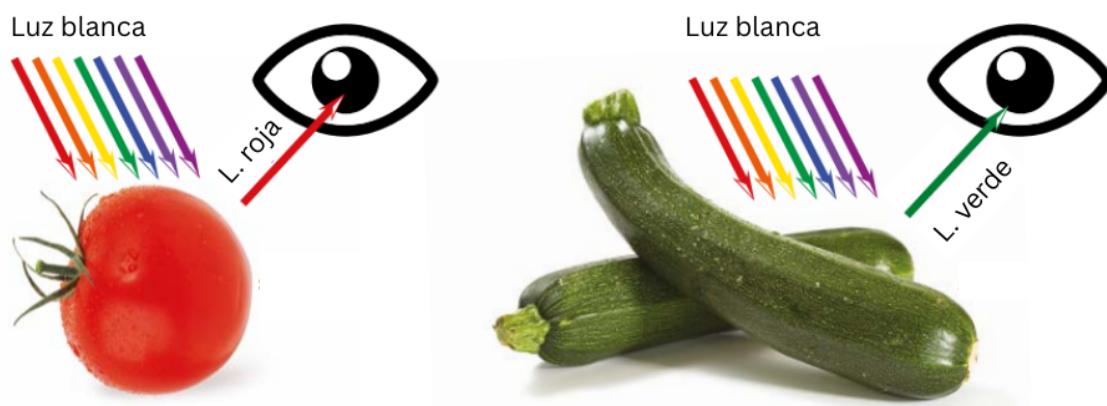


Figura B.3: La luz que no es absorbida por el objeto es reflejada y puede verse por el ojo.
[1]

Ambos objetos son iluminados por luz visible o blanca, los diferentes colores representan diferentes componentes de la luz visible. Cuando los rayos de luz inciden sobre un objeto, pueden ser absorbidos por el objeto, en particular, uno o más componentes de luz (es decir, sus colores) se absorben específicamente.

Los colores que no son absorbidos por el objeto son reflejados. En nuestro ejemplo, la luz roja se refleja por la cáscara roja del tomate (Figura B.3), mientras que la luz verde se refleja por la superficie verde del calabacín. Todos los otros colores son absorbidos por estos dos objetos. Entonces el ojo ve la luz reflejada: el tomate es rojo, mientras que el calabacín es verde.

En términos físicos, la luz es una onda electromagnética que viaja a través del espacio. La energía de la luz oscila periódicamente entre un mínimo y un máximo como una onda en función del tiempo. La distancia entre dos máximos o dos mínimos de una onda electromagnética se define como longitud de onda como se muestra en la figura B.4, este dato se suele medir en nanómetros (nm).

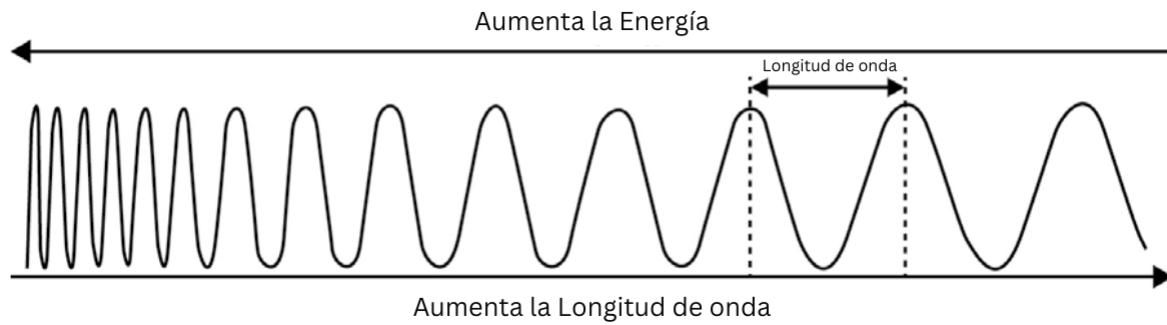


Figura B.4: La energía de las ondas electromagnéticas aumenta a medida que disminuye la longitud de onda y viceversa. [1]

Los diferentes componentes de la luz se caracterizan por longitudes de onda específicas. Se cumple que la luz roja tiene una longitud de onda de 660 nm, mientras que la luz verde tiene una longitud de onda de 520 nm.

La suma de todos los componentes (es decir, todas las longitudes de onda) se denomina espectro. Más específicamente, el espectro representa la distribución de la energía radiante. Por ejemplo, el espectro electromagnético de la luz visible varía desde aproximadamente 390 nm hasta aproximadamente 780 nm (Figura B.5).

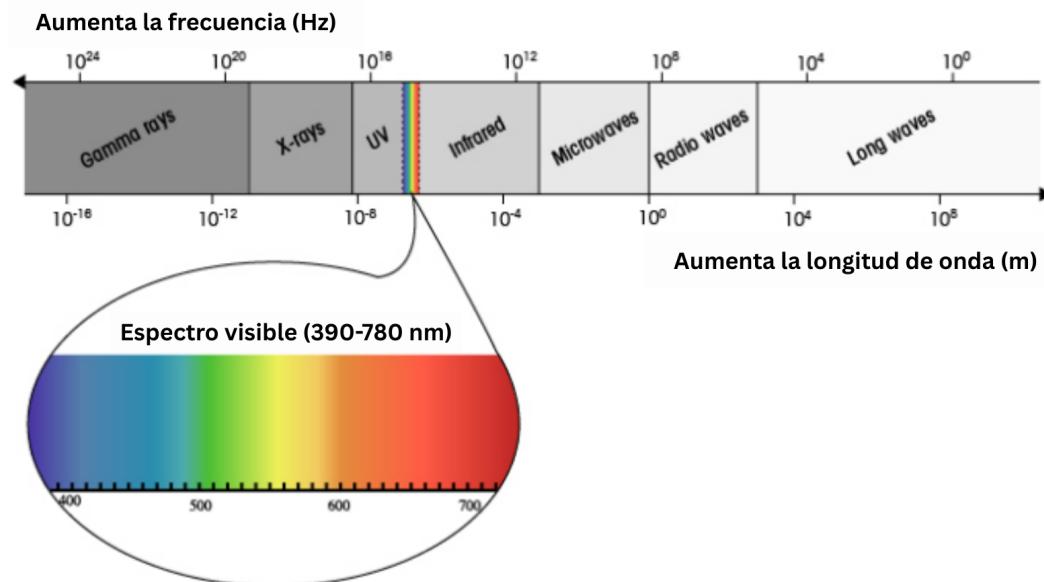


Figura B.5: El espectro visible (390-780 nm) representa solo una pequeña parte del espectro electromagnético completo. [1]

Cuanto más corta es la longitud de onda, mayor es la energía. Por ejemplo, la luz

violeta tiene una longitud de onda más corta que la luz roja y, por lo tanto, tiene un nivel de energía más alto, mientras que la luz infrarroja tiene menos energía que la luz visible debido a su longitud de onda más larga. Esta relación se expresa en la ecuación B.1, donde se distinguen los elementos:

- E: Energía del fotón.
- h: Constante de Planck.
- c: Velocidad de la luz en el vacío.
- λ : Longitud de onda.

$$E = \frac{hc}{\lambda} \quad (\text{B.1})$$

B.2.2. Absorción de luz como herramienta analítica

La absorción de luz se puede utilizar para la caracterización y determinación cuantitativa de sustancias en el campo de la química analítica. La espectroscopía UV/Vis es una técnica basada en la absorción de luz por una muestra. Aquí, la muestra se irradia con rayos electromagnéticos de diversas longitudes de onda en luz visible y en el rango adyacente (es decir, ultravioleta y parte de la región infrarroja). Dependiendo de la sustancia, la luz se absorbe parcialmente (Figura B.6). La luz restante, es decir, la luz transmitida, se registra mediante un detector adecuado como una función de la longitud de onda, proporcionando el espectro UV/Vis de la muestra. Como resultado, debido a que cada sustancia absorbe la luz de una manera diferente, existe una relación única y específica entre la sustancia y su espectro UV/Vis. Este espectro se puede utilizar entonces para identificar o cuantificar la sustancia.

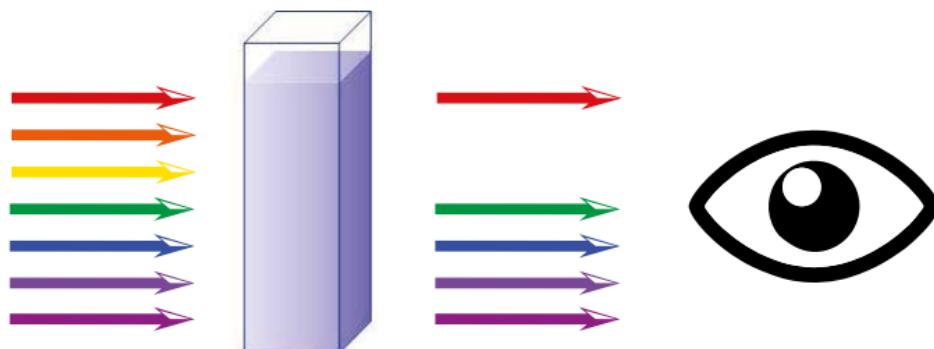


Figura B.6: La luz que pasa a través de la solución de muestra es parcialmente absorbida por los componentes. [1]

Un espectrofotómetro UV/Vis mide la intensidad de la luz que pasa a través de la solución de muestra en la cubeta y la compara con la intensidad de la luz antes de pasar a través de la muestra. Los componentes principales de un espectrofotómetro UV/Vis son una fuente de luz, un portamuestras, opcionalmente un dispositivo de dispersión (por ejemplo, un monocromador) que separa la luz de diferentes longitudes de onda y un detector adecuado (Figura B.7).



Figura B.7: Principio y componentes básicos de medición en espectroscopía UV/Vis [1]

Un detector en un espectrofotómetro UV/Vis mide la intensidad de la luz después de que ha pasado a través de la solución de muestra. Esta parte de la luz recogida por el detector se denomina intensidad de transmisión I . La intensidad de la luz transmitida es atenuada por la solución de muestra debido, por ejemplo, a la absorción de una longitud de onda específica. Por lo tanto, su valor es inferior a la intensidad original I_0 en la fuente de luz (Figura B.8).

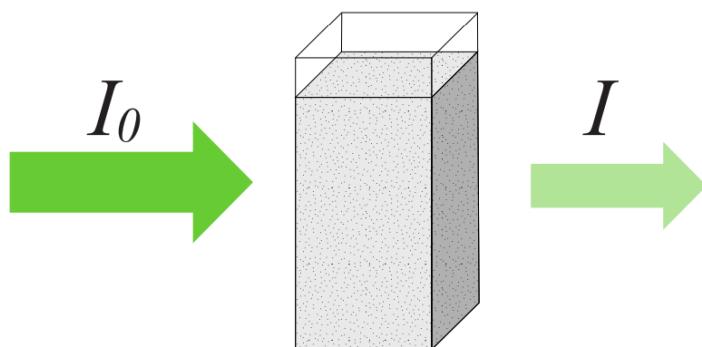


Figura B.8: Atenuación de la luz causada por la absorción de la solución. [1]

La relación I/I_0 de las dos intensidades se define como la transmitancia T, que se expresa en porcentaje.

$$T = \frac{I}{I_0}$$

La absorbancia A representa un resultado adicional ampliamente utilizado en el registro de espectros UV/Vis. Se define como el logaritmo negativo de la transmitancia, y tiene grandes ventajas.

$$A = -\log(T)$$

La absorbancia A no tiene ninguna unidad de medida. En otras palabras, es un valor adimensional. Sin embargo, a menudo se representa usando la letra “A” o “AU” como unidad de absorbancia. Por ejemplo, 0,3 A o 0,3 unidades de absorbancia, respectivamente.

Desde el punto de vista operativo, esas definiciones teóricas no se pueden emplear directamente. Para realizar una medición fotométrica se debe primero definir una escala fotométrica (rango 0 % - 100 % de transmitancia). Este rango lineal se define por la medida de dos puntos:

1. El 0 % T (S_0) correspondiente a la respuesta del detector en ausencia de iluminación, conocido a veces como “corriente oscura” o “señal oscura”. Se puede medir apagando la fuente de radiación (el LED en el caso de nuestro proyecto).
2. El 100 % T (S_{ref}), se obtiene con la fuente de radiación encendida y corresponde a la señal de referencia. En caso de estudiarse una muestra de agua, se obtiene tomando la medida sobre la muestra sin el agregado de reactivos. Esto proporcionaría la señal luminosa de mayor intensidad posible en el ensayo.

Se define entonces la transmitancia relativa como:

$$T_r = \frac{S_x - S_0}{S_{ref} - S_0}$$

Siendo S_x la señal que entrega el detector al medir una solución con reactivo agregado.

Puede ocurrir que algunos detectores hayan sido diseñados para generar una respuesta nula en condiciones de oscuridad, en cuyo caso no sería necesaria la medición de la “señal oscura”. A partir del valor de la transmitancia relativa se calcula la absorbancia relativa A_r aplicando:

$$A_r = -\log(T_r)$$

Desde el punto de vista experimental, lo que se mide es siempre una absorbancia relativa calculada a partir de una transmitancia relativa.

Es de gran importancia para este tipo de ensayo la selección de un led de frecuencia adecuada al tipo de medición que se desea realizar. En la sección B.2.4 se detallan ensayos particulares y la frecuencia de la fuente lumínica requerida para realizarlos. El diseño del prototipo final permite un fácil intercambio del led implementado para realizar la medida óptica, como ejemplo presentamos algunos leds disponibles en el mercado y la frecuencia que emiten:

- Los leds infrarrojos emiten una señal lumínica de alrededor de 850-940 nm y se pueden encargar para generar un valor particular y considerablemente preciso.
- Leds ultravioleta manejan una frecuencia de entre 365-405 nm.
- Leds rojos funcionan a una frecuencia en el entorno de 620–750 nm.
- Leds azules funcionan a una frecuencia en el entorno de 450–495 nm.

Es necesario también en este tipo de ensayos, verificar que la fuente lumínica es alimentada por una corriente suficientemente estable. Si no es el caso, no se alcanzará la precisión requerida en las medidas tomadas.

Como ejemplo se muestran los resultados de mediciones realizadas sobre una muestra utilizando un instrumento UV/Vis en un laboratorio químico por terceros:

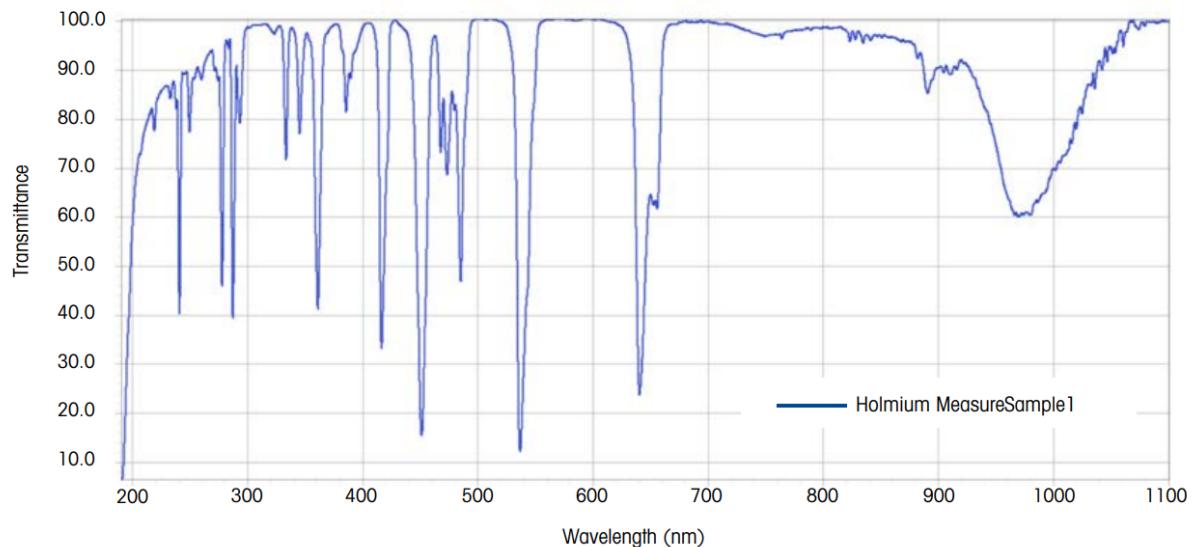


Figura B.9: Espectro de transmisión de una solución particular en función de la longitud de onda. [1]

El espectro de transmisión que se muestra en la figura B.9, se registró en función de la longitud de onda. En este ejemplo particular, la muestra absorbe principalmente luz a cuatro longitudes de onda diferentes, es decir, aprox. 370, 450, 480 y 540 nm. La absorción de luz se caracteriza por una reducción drástica de la transmitancia a estas longitudes de onda.

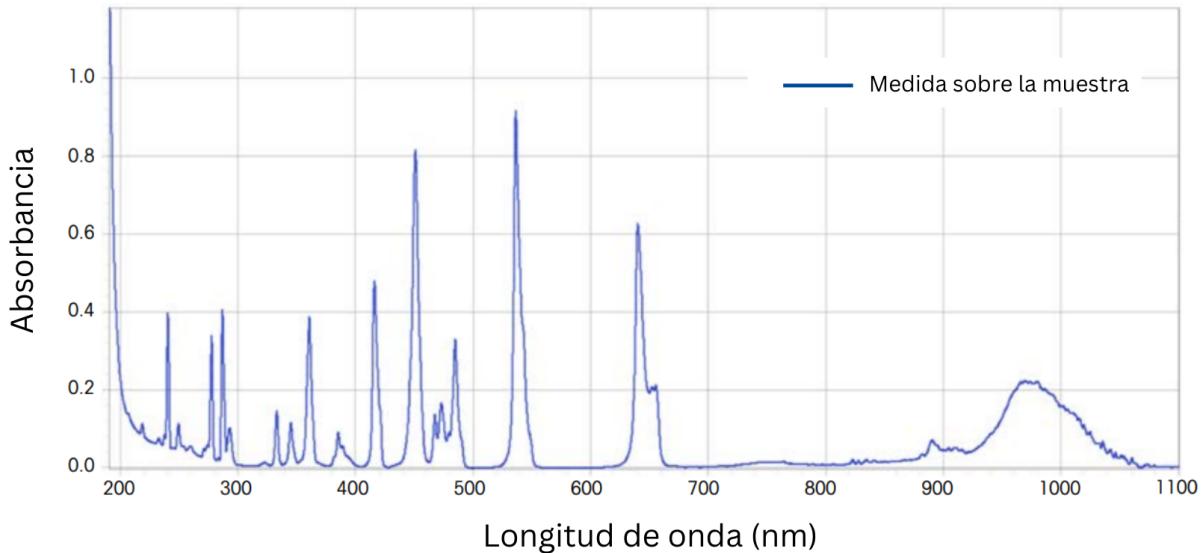


Figura B.10: Espectros de absorción de soluciones de holmio en función de la longitud de onda. [1]

En figura B.10, los espectros de absorción de la misma muestra, también se dan en función de la longitud de onda. Se puede identificar que los picos de absorción están situados en las mismas longitud de onda.

En este caso, el grado de absorción de luz se indica mediante un valor de absorbancia más alto. En general, los espectros UV/Vis se representan gráficamente como la absorbancia en función de la longitudes de onda. La altura del pico de absorción es proporcional a la concentración de la sustancia.

B.2.3. Ley de Lambert-Beer

Cuando se pasa a través de una cubeta transparente llena de solución de muestra, la intensidad de la luz se atenúa en proporción a la concentración de la muestra. En otras palabras, una solución de muestra de mayor concentración absorberá más luz. Además, la atenuación también es proporcional a la longitud de la cubeta; una cubeta más larga dará como resultado una mayor absorción de luz.

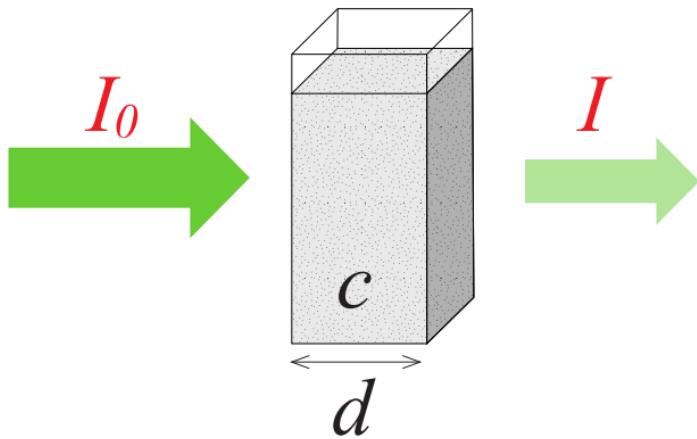


Figura B.11: La atenuación de la intensidad de la luz es proporcional a la concentración de la solución de muestra y a la longitud de la cubeta. [1]

Ambos factores se pueden generalizar expresando la absorbancia A en función de la concentración y la longitud de la cubeta. En particular, la absorbancia A es igual al producto del coeficiente de extinción ε , la concentración c (Figura B.11) y la longitud de trayectoria d (ecuación B.2).

$$A = \varepsilon \cdot c \cdot d \quad (\text{B.2})$$

A esta relación se la llama ley de Lambert-Beer, donde:

- La concentración de la muestra c está dada en mol / L or g / mL.
- El largo de la cubeta está dado en cm.
- El coeficiente de extinción ε (epsilon) es una constante que describe cuánto de una determinada longitud de onda absorbe la muestra (en mol / L or g / mL respectivamente).

Se muestra en la Figura B.12 un ejemplo de esta propiedad física.

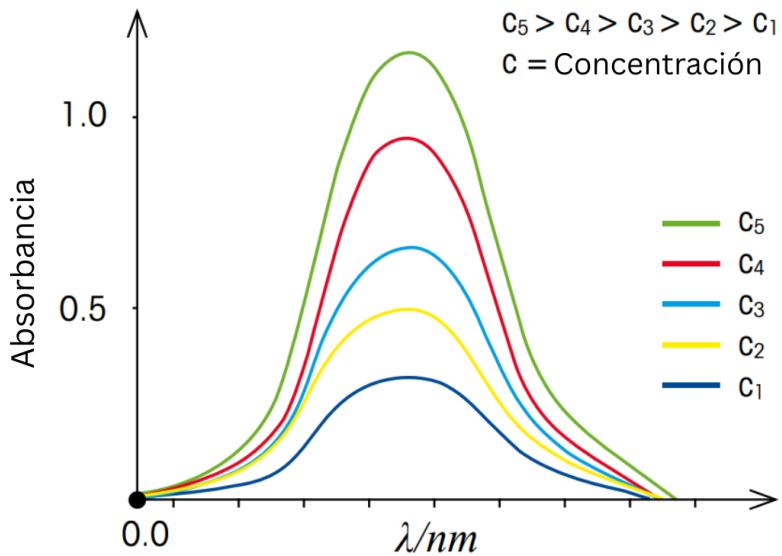


Figura B.12: Cuanto mayor es la concentración, mayor es el valor de absorbancia. [1]

B.2.4. Aplicación en Longitud de onda fija

La medición de longitud de onda fija es la aplicación más sencilla del espectrofotómetro. Es una medición de longitud de onda única o de longitud de onda múltiple, los resultados pueden reportarse en términos de densidad de energética, absorbancia o transmitancia. Se pueden realizar cálculos adicionales para obtener un resultado final, por ejemplo, la concentración de la sustancia.

Pureza del aceite de oliva (industria alimentaria)

Una aplicación típica de este segmento del mercado alimenticio es la verificación de la pureza del aceite de oliva, lo que permite clasificar el producto como “virgen extra”, “virgen” o simplemente “aceite de oliva”. Basándose en las propiedades de absorbancia de ciertas moléculas en el espectro UV/Vis.

El aceite de oliva contiene alrededor del 98 % de triglicéridos. Los ácidos grasos insaturados en el aceite son fácilmente descomponibles y oxidados. La oxidación de los ácidos grasos libres conduce a la formación de peróxidos. Con el tiempo, esto puede causar rancidez y degradación del aceite de oliva.

Las normas sobre el aceite de oliva de la Comisión Oleícola Internacional (COI) especifican con precisión los umbrales de medición que el aceite de oliva debe cumplir para ser clasificado como virgen extra, virgen, etc. La calidad del aceite se determina observando el comportamiento de absorbancia de una solución de alcohol isopropílico al 1% (99 % alcohol isopropílico, 1 % aceite) entre 200 y 400 nm. Los niveles elevados de absorbancia en el intervalopectral de 200 a 400 nm indican aceites oxidados (de peor calidad). La

absorbancia entre 232 nm y 270 nm se correlaciona con el estado de oxidación mediante la detección de compuestos oxidantes específicos.

En la figura B.13 podemos ver que el espectro UV del aceite de oliva virgen extra de alta calidad no muestra picos, en comparación con los picos evidentes del aceite de oliva de baja calidad.

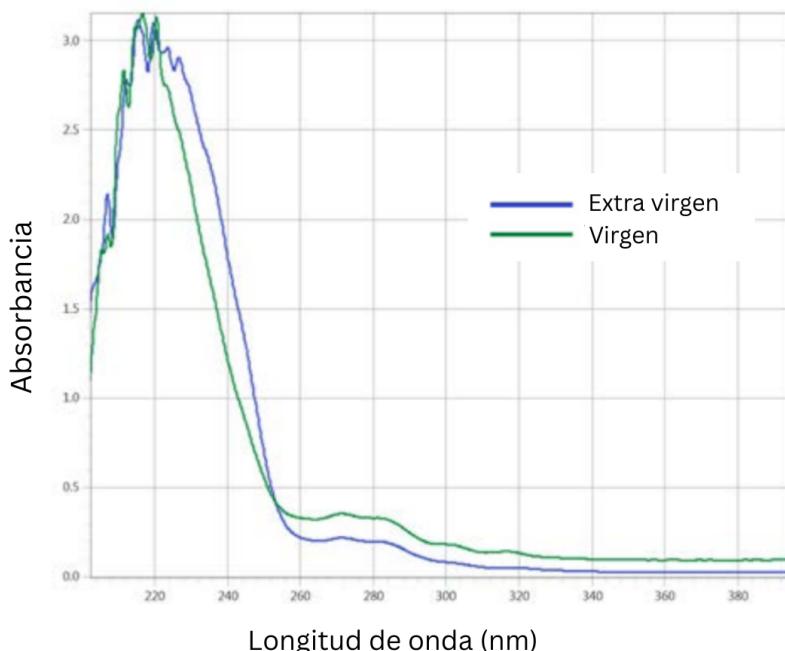


Figura B.13: Absorción medida en solución al 1 % de aceite de oliva virgen extra (azul) y aceite de oliva virgen (verde) [1]

Aplicación en la industria química

La espectroscopía de absorción ultravioleta es uno de los mejores métodos para determinar la pureza de soluciones orgánicas. Debido a las impurezas en la muestra, se pueden observar picos adicionales que aparecen a longitudes de onda específicas. Un ejemplo en la industria química es el control de la pureza del alcohol. El alcohol puede estar contaminado con benceno, que absorbe luz a 280 nm, mientras que el alcohol absorbe luz a 210 nm. Si existe un pico adicional a 280 nm, la medición del espectro UV/Vis puede determinar fácilmente si la muestra está contaminada o no.

Determinación cuantitativa de la concentración

La determinación cuantitativa o de concentración de una sustancia mediante espectroscopía UV/Vis se basa en la ley de Lambert-Beer descrita en el capítulo 2.4, que establece que la absorbancia de una solución es proporcional a la concentración de la sustancia ab-

sorbente en la solución y a la longitud de la trayectoria de la cubeta. Por lo tanto, para una longitud de trayectoria fija , la espectroscopía UV/Vis puede usarse para determinar la concentración de la sustancia absorbente en la solución. Sin embargo, es necesario conocer el grado en que la absorbancia varía con la concentración. Esto puede obtenerse a partir de referencias tales como tablas de coeficientes de extinción, o puede determinarse más fácilmente a partir de curvas de calibración.

Determinación colorimétrica de los fosfatos

El fósforo es el undécimo elemento más abundante en la superficie de la Tierra y su forma más común es el fosfato. Desempeña un papel importante en los procesos bioquímicos y es un factor clave en las aguas superficiales. El aumento de la concentración de fosfato se asocia con un aumento de la tasa de crecimiento de plantas. Por lo tanto, el análisis del fósforo es de gran importancia en muchos campos, como la ciencia médica y clínica, la agricultura, la metalurgia y las ciencias ambientales. Además, en los últimos años, se han utilizado grandes cantidades de fosfato en las bebidas, detergentes y fertilizantes. En este caso, la determinación colorimétrica de fosfato se lleva a cabo utilizando espectrofotometría UV/Vis. La base de la técnica colorimétrica es la relación directa entre la intensidad del color de la solución y la concentración del componente coloreado contenido.

En presencia de un agente reductor adecuado, el fosfato reaccionará con el molibdato de amonio (Reactivo) para generar un tono azul (Figura B.14) cuya intensidad es proporcional a la concentración de fosfato en la solución. El contenido de fosfato de una muestra desconocida puede obtenerse trazando primero la absorbancia de una serie de soluciones patrón frente a las concentraciones correspondientes, dando así una curva de calibración. La concentración desconocida de fosfato en la muestra se puede determinar entonces a partir de la curva (Figuras B.15 y B.16).



Figura B.14: Serie de soluciones patrón de fosfato preparadas [1]

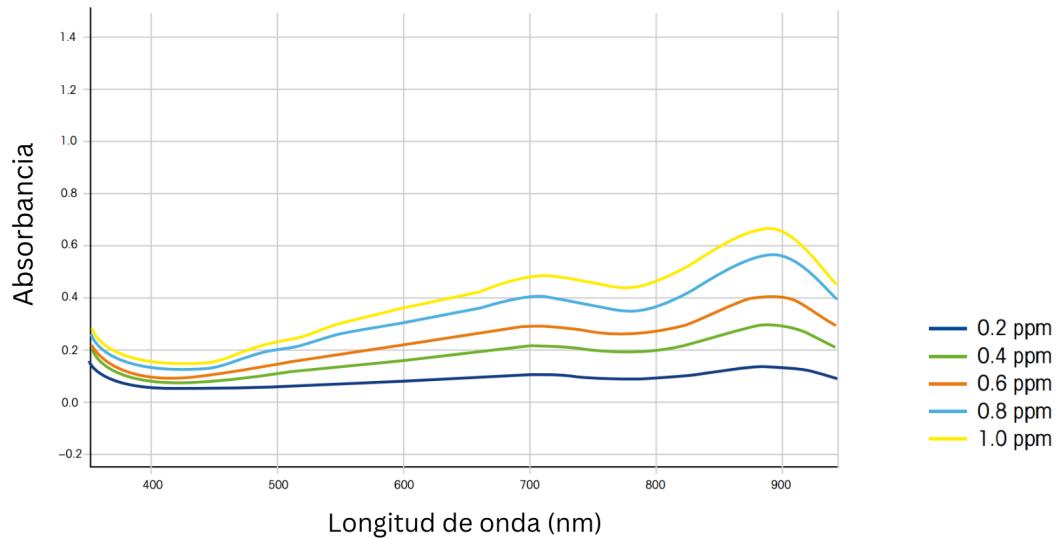


Figura B.15: Espectros de absorción trazados para las concentraciones correspondientes. [1]

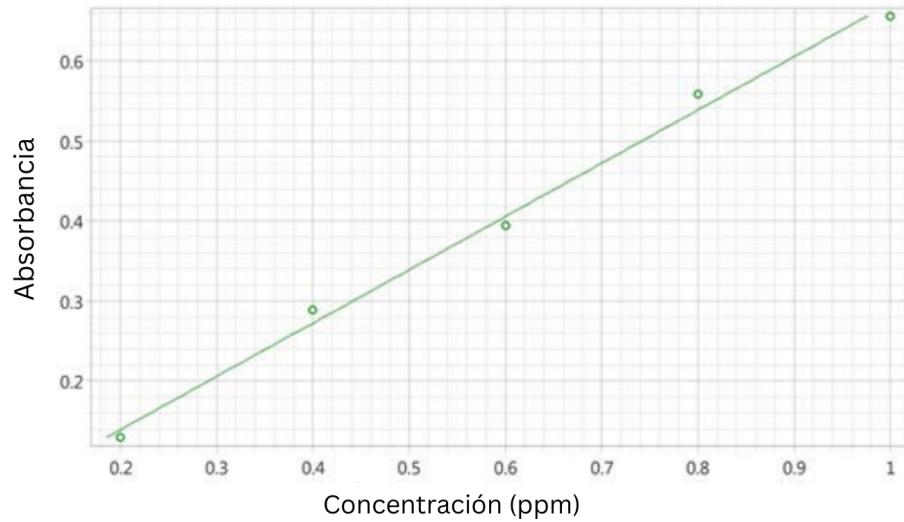


Figura B.16: Valores de absorbancia a 880 nm frente a la concentración de cada solución patrón para generar la curva de calibración. La concentración desconocida de fosfato en la muestra puede determinarse a partir de una curva de regresión lineal. [1]

B.3. Protocolos de comunicación

B.3.1. Protocolo Modbus

Modbus utiliza la modalidad maestro esclavo, donde el maestro envía y recibe tramas comunicándose con uno o más esclavos identificados mediante direcciones y tipos de registros.

Este protocolo tiene modalidades y funciones que proporcionan diferentes opciones. En nuestro proyecto se implementó el protocolo Modbus-rtu y la función “holding registers” para lograr la comunicación entre varios componentes distintos.

La figura B.17 muestra la estructura de un mensaje tipo “read holding registers” donde la imagen superior representa una solicitud de un número de registros (datos) particulares y la inferior es la estructura de la respuesta por parte del esclavo enviando la información solicitada.

Dirección	Función	Datos				CRC
		Dirección de inicio		Cantidad de registros		
add	3	MSB	LSB	MSB	LSB	crc16

Dirección	Función	Datos		CRC
		Num. de bytes datos	Información	
add	3	n (8 bit)	n/2 registros	crc16

Figura B.17: Estructura de las tramas de comunicación modbus rtu - holding registers

Para explicar de manera más clara la estructura de estos mensajes, exponemos un ejemplo:

Maestro solicita registros (HEX): 11 03 006B 0003 7687

- 11 - Dirección del esclavo.
- 03 - Código de función “read holding registers”.
- 006B - Dirección del primer registro a recibir.
- 0003 - Número de registros solicitados.
- 7687 - CRC para verificación de errores.

Esclavo responde (HEX): 11 03 06 AE41 5652 4340 49AD

- 11 - Dirección del esclavo.

- 03 - Código de función “read holding registers”.
- 06 - Número de bytes que se envían con datos (3 registros, 2 bytes cada uno).
- AE41 - Contenidos del registro 1 (datos).
- 5652 - Contenidos del registro 2 (datos).
- 4340 - Contenidos del registro 3 (datos).
- 49AD - CRC para verificación de errores.

B.3.2. Protocolo I2C

Este protocolo de comunicación es utilizado para comunicarse con el sensor de intensidad de luz OPT3002 en nuestro proyecto, consiste en el envío y recepción de tramas compuestas por bytes de valores específicos.

La figura B.18 es un ejemplo de la trama enviada al sensor por parte del controlador, donde el primer byte “frame 1” de la figura está compuesto por 7 bits correspondientes a la dirección del dispositivo a manejar, en nuestro caso la dirección es 0x44 HEX (Dirección correspondiente a la salida ADDR del sensor cortocircuitada con GND), el último bit se envía en baja, de esta manera se señalaiza que se están enviando datos.

El resto de los bytes mostrados (frame 2, 3, 4) corresponden a los datos que se envían al sensor, donde inicialmente se debe configurar al dispositivo para actuar de la manera requerida.

Una vez enviada de esta trama se escribe un nuevo mensaje indicando al dispositivo que se quiere comenzar a leer medidas, en este caso los datos enviados son 0x00 HEX.

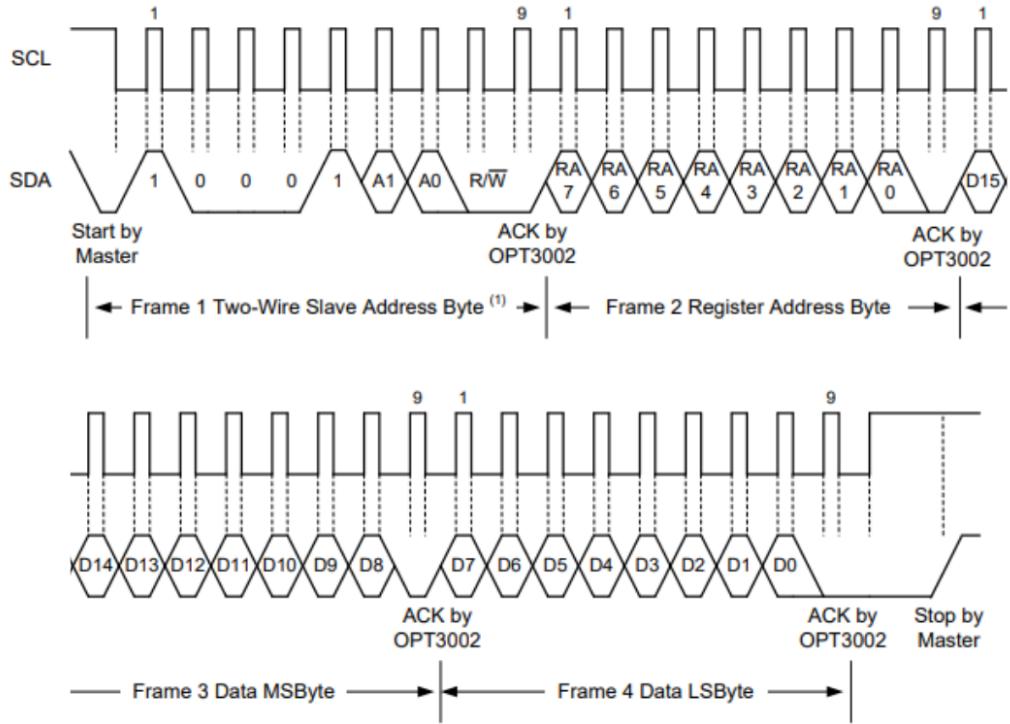


Figura B.18: Estructura de la trama enviada a sensor OPT3002 [4]

La figura B.19 es un ejemplo de la trama enviada hacia el controlador, por parte del sensor, donde el primer byte “frame 1” de la figura está compuesto por 7 bits correspondientes a la dirección del dispositivo a manejar 0x44 HEX, el último bit se envía en alta, ya que se están recibiendo datos.

El resto de los bytes mostrados (frame 2, 3) corresponden a los datos que envía al sensor. Al recibir medidas desde el OPT3002, estas están compuestas por 2 bytes, los primeros 4 bits recibidos contienen el valor de la exponencial que debe aplicarse al valor representado 12 bits restantes, el resultado final se despeja mediante la siguiente ecuación:

$$Medida = (2^{R[16:12]}) \times R[11 : 0] \times 1,2 \quad [nW/cm^2] \quad (B.3)$$

Donde $R[16:0]$ representa el mensaje recibido de medida.

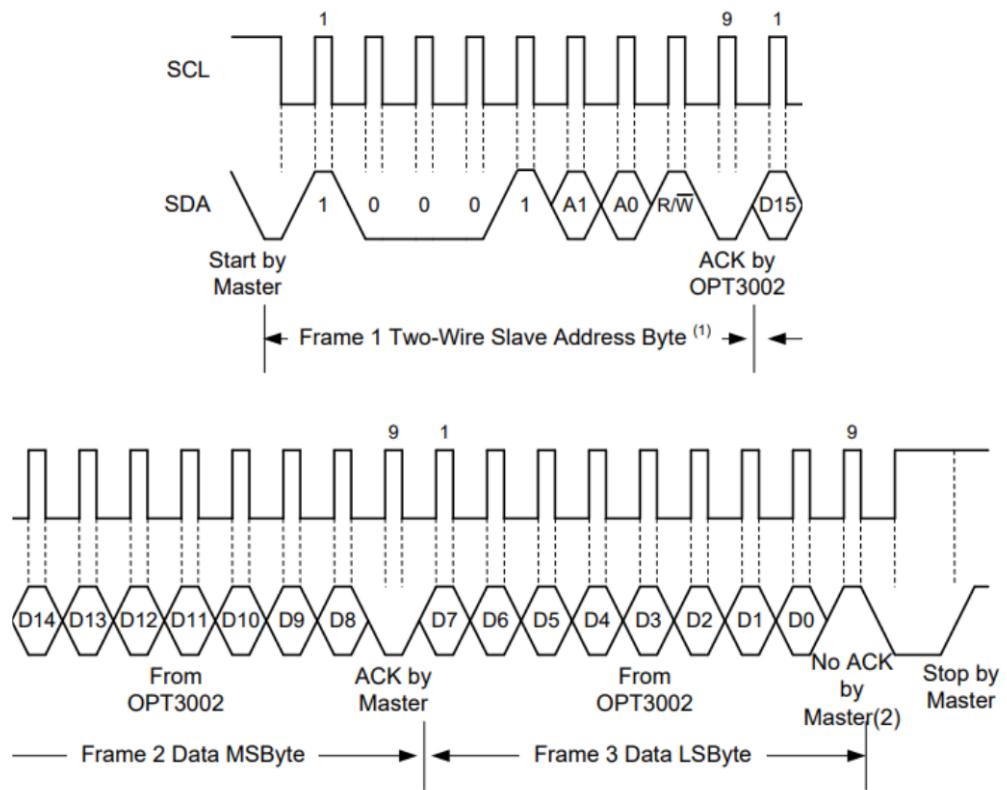


Figura B.19: Estructura de la trama recibida por el sensor OPT3002 [4]

B.4. Resumen Norma IEC 61131-3

La norma IEC 61131-3 constituye el estándar internacional para la programación de controladores lógicos programables (PLC). Esta norma, perteneciente a la familia IEC 61131, establece un marco estructurado que busca garantizar la portabilidad del software, la reutilización de componentes y la independencia del fabricante.

Uno de sus principales aportes es la estandarización de cinco lenguajes de programación utilizados en automatización industrial:

- **Ladder Diagram (LD):** basado en lógica de relés, con un enfoque gráfico que facilita su comprensión por parte de técnicos eléctricos.
- **Function Block Diagram (FBD):** permite representar funciones mediante bloques conectados entre sí, ideal para lógica combinacional y procesos repetitivos.
- **Structured Text (ST):** lenguaje textual de alto nivel, similar a Pascal, adecuado para operaciones matemáticas, estructuración condicional y algoritmos complejos.
- **Instruction List (IL):** lenguaje de bajo nivel similar a un ensamblador; ha sido declarado obsoleto en versiones recientes.
- **Sequential Function Chart (SFC):** orientado a la programación de procesos secuenciales mediante etapas (steps) y transiciones.

La norma también introduce el concepto de *Program Organization Units* (POU), que incluye programas, funciones y bloques funcionales, promoviendo una programación modular y jerárquica. Además, define tipos de datos estandarizados, estructuras de control, y reglas para la declaración de variables locales, globales y temporales.

En conjunto, la IEC 61131-3 promueve buenas prácticas de desarrollo en sistemas de automatización, facilitando el mantenimiento, la escalabilidad y la interoperabilidad entre plataformas. Su adopción en la industria ha permitido una mayor estandarización y calidad en los proyectos de control.

Apéndice C

Apéndice C: Memoria de programación

C.1. Software en PLC

C.1.1. Programa Procesamiento Registros Modbus Drivers Bombas (ST)

El objetivo principal es interpretar correctamente la información proveniente de los registros Modbus de los drivers de las bombas peristálticas.

Se pone como ejemplo ejemplo que se leer un dato Udim32. Al ser un dato de 32 bits corresponden dos registros o palabras Modbus cada registro tiene 16 bits y a priori no se sabe en que orden están incluso existen casos que una palabra se lea al revés.

En el siguiente código se exemplifica el uso de un array de registros Modbus consecutivos y como se crea a partir del mismo otro array en versión SWAP, esencial en las primeras instancias de interacción con el driver.

```
1 FOR i:=0 TO 6 BY 2 DO
2     Registros_Holding_Swap[i] := Registros_Holding[i+7];
3     Registros_Holding_Swap[i+1]:= Registros_Holding[i+6];
4 END_FOR
```

Aplicando funciones como SWAP para mostrar de forma adecuada parámetros como la velocidad. Se configuraron los parámetros de cada driver, incluyendo el número de esclavo Modbus RTU, el baudrate y las RPM máximas de cada bomba, con el fin de establecer la comunicación inicial en la red RS485. Resultó especialmente útil probar la lectura y escritura de registros de manera online, lo que permitió visualizar en tiempo real toda la información relevante en esta etapa y verificar el accionamiento esperado.

En CODESYS, los drivers se configuran como dispositivos asociados, brindando la posibilidad de utilizar librerías que permiten ejecutar lecturas y escrituras según la lógica programada por el maestro Modbus, generalmente vinculadas a banderas dentro del código. Asimismo, es posible crear canales dentro de un esclavo Modbus, asociándolos a

funciones específicas y a un mapeo de memoria de entradas y salidas. Tras un proceso de pruebas sobre los drivers, se procede a documentar los registros principales y sus funciones asociadas.

Los registros mas relevantes son:

- **Start-Stop-Control (Coil Register)**: Se configuro un canal con la funcion 5 de Modbus, en cada esclavo (driver) permitiendo establecer el encendido y apagado de cada bomba asociado a una memoria de salida.
- **Actual-Speed (Input Register)**: Se configuro un canal con la funcion 4 en cada esclavo (driver) Modbus, es un tipo de dato Uint32 en dos registros 0x3001-0x3002, el cual permite con una resolucion de 0.01 conocer las rpm actuales.
- **Operating-Speed (Holding Register)**: Se configuro un canal con la funcion 6 en cada esclavo (driver) Modbus, es un tipo de dato Uint32 en dos registros 0x4009-0x400A, el cual permite escribir las rpm de operacion en regimen.
- **Max-Speed (Holding Register)**: Se configuró un canal con la Función Modbus 6 (Write Single Register) en cada esclavo (driver) Modbus. Este canal corresponde a un dato de tipo UInt32 distribuido en dos registros consecutivos: 0x4004 – 0x4005.

Este registro permite establecer el valor máximo de RPM para la velocidad de operación, con una resolución de 0.01 RPM. Durante la etapa de pruebas, este registro presentó dificultades, ya que no se encontraba documentado en la tabla de registros Holding de la versión del manual utilizada para el testing.

La funcionalidad fue identificada al realizar pruebas de escritura masiva de registros: se observaba que, tras escribir todos los valores, el registro Operating-Speed siempre retornaba un valor cero, independientemente del valor escrito. Tras modificar el registro de Max-Speed, se comprobó lo siguiente:

Si se intenta escribir un valor de velocidad de operación superior al definido en Max-Speed, el driver rechaza la escritura.

Al modificar el valor de Max-Speed y volver a escribir el parámetro de velocidad de operación, se verificó el comportamiento anteriormente descrito.

Esta validación permitió confirmar el propósito del registro y su influencia directa sobre los límites de velocidad operativa de los drivers.

C.1.2. Programa Principal (CFC)

El presente programa tiene como objetivo permitir el funcionamiento de todos los actuadores del sistema tanto en modo automático —según parámetros predefinidos— como en modo manual.

Asimismo, se establece una secuencia de etapas a seguir para ejecutar el proceso de medición automática, la cual se detalla en el bloque correspondiente al proceso automático.

La modularización de esta estructura facilita la integración de múltiples procesos automáticos, la automatización basada en horarios, o bien en otras configuraciones específicas según las necesidades del sistema general. Por ejemplo, la incorporación de más de una bomba peristáltica podría ser necesaria en aplicaciones que requieran el uso de varios reactivos. Del mismo modo, la adición de electroválvulas permite modificar el circuito hidráulico del sistema para adaptarlo a distintos requerimientos operativos.

Se seleccionó el lenguaje CFC (Continuous Function Chart) con el objetivo de facilitar la visualización rápida y directa de las variables de entrada y salida de los bloques principales del programa. Este lenguaje permite representar de forma gráfica y continua el flujo de señales y datos, lo cual resulta especialmente útil en aplicaciones donde se requiere observar con claridad el comportamiento lógico del sistema.

Además, CFC proporciona una mayor flexibilidad al momento de interconectar bloques funcionales, permitiendo organizar el programa de forma más intuitiva y comprensible. Esto favorece tanto el desarrollo como el mantenimiento del software, particularmente en sistemas complejos o con múltiples condiciones de operación.

El prefijo "Visu" se utiliza para identificar aquellas variables que se modifican en función de la interacción con la interfaz de visualización, permitiendo distinguirlas fácilmente dentro del entorno de programación y depuración.

Se profundiza a continuación sobre el funcionamiento interno de cada bloque.

Los siguientes dos primeros bloques se basan en el diagrama de estados de la figura.

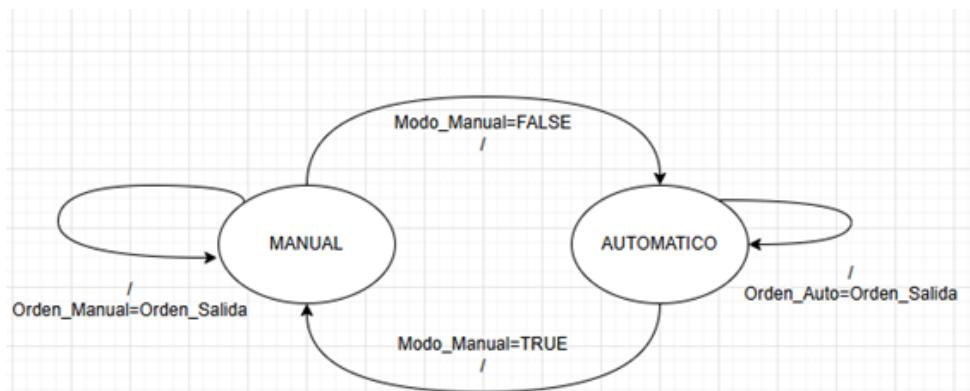


Figura C.1: Diagrama de estados bloques Electroválvula y Bomba

Bloque de bomba (FBD)

Se utiliza la función SEL (Selector) para operar la bomba en modo automático o manual, según la condición establecida. Como se muestra en la figura se tiene una entrada G booleana donde se elige entre dos entradas IN0, IN1 y da IN1 como salida si el valor de G es TRUE, de lo contrario da IN0. Se tiene como observación que IN0, IN1 y la salida tienen que ser del mismo tipo.

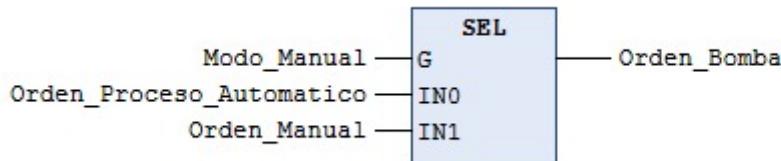


Figura C.2: Bloque SEL en FBD

El bloque cuenta con **cinco variables de entrada** y **dos variables de salida**, descritas a continuación:

VARIABLES DE ENTRADA

- **Modo Manual:** Se conecta a la entrada G de la función SEL. Indica si se ha seleccionado el modo manual desde la visualización.
- **Orden Proceso Automático:** Bit de control generado como salida del bloque de proceso automático. Solo es considerado cuando el modo manual no está activo.
- **RPM Proceso Automático:** Valor de velocidad asignado por el bloque de proceso automático, utilizado únicamente cuando el sistema se encuentra operando en modo automático. La lógica de selección de esta velocidad se detalla dentro del bloque correspondiente.
- **Orden Manual:** Variable que se modifica en función de la interacción con la visualización y el programa Modbus TCP Master. Representa la orden de marcha en modo manual.
- **RPM Manual:** Corresponde a las revoluciones por minuto ingresadas manualmente a través de la visualización, también mediante comunicación con el programa Modbus TCP Master.

Variables de Salida

Corresponden directamente a las salidas de los bloques SEL asociados a cada tipo.

- **Orden bomba:** Directamente cambia el registro Coil en los drivers asociados a las bombas.
- **Rpm Bomba:** Directamente escribe las RPM en el registro asociado a la velocidad de operación en régimen.

Bloque de electroválvula

Se utiliza una línea de código con el bloque SEL como se ilustra en la figura C.2 en FBD, igual forma que en el bloque bomba, para encender y apagar la bomba con variables asociadas al modo manual o modo automático. La salida del bloque en el programa principal está asignada directamente a una salida de Hardware en el módulo de salidas digitales.

Bloque de proceso automático

El bloque se encuentra programado en lenguaje FBD y define el funcionamiento del **modo automático** del proceso.

El proceso se divide en tres etapas secuenciales:

- **Etapa de dosificación**
- **Etapa de medida**
- **Etapa de lavado**

En **modo automático**, se habilitan las siguientes funcionalidades:

- Iniciar y detener el proceso.
- Cambiar al **modo manual** en cualquier momento, pausando la secuencia.
- Repetir el proceso las veces que sea necesario.
- Registrar automáticamente las medidas obtenidas durante la ejecución.

Conceptualmente se sigue el diagrama de estados ilustrado en la figura C.3.

El sistema se organiza en torno a cuatro estados principales, cada uno representando una etapa funcional del proceso automatizado:

- **Sin accionar:** Estado de reposo del sistema. No hay actuadores encendidos.
- **Dosificar:** Se activan bombas para dosificar reactivo y muestra.
- **Tomar Medida:** Se habilita el flujo de muestra hacia el sensor, se espera un tiempo definido y se registra el valor medido.
- **Lavado:** Se realiza una limpieza de las líneas y sensores para preparar el sistema para una nueva medición.

Las transiciones entre estos estados están gobernadas por condiciones lógicas internas y dependen tanto del modo de operación (**Modo_Manual**) como del cumplimiento de señales de avance definidas en el proceso.

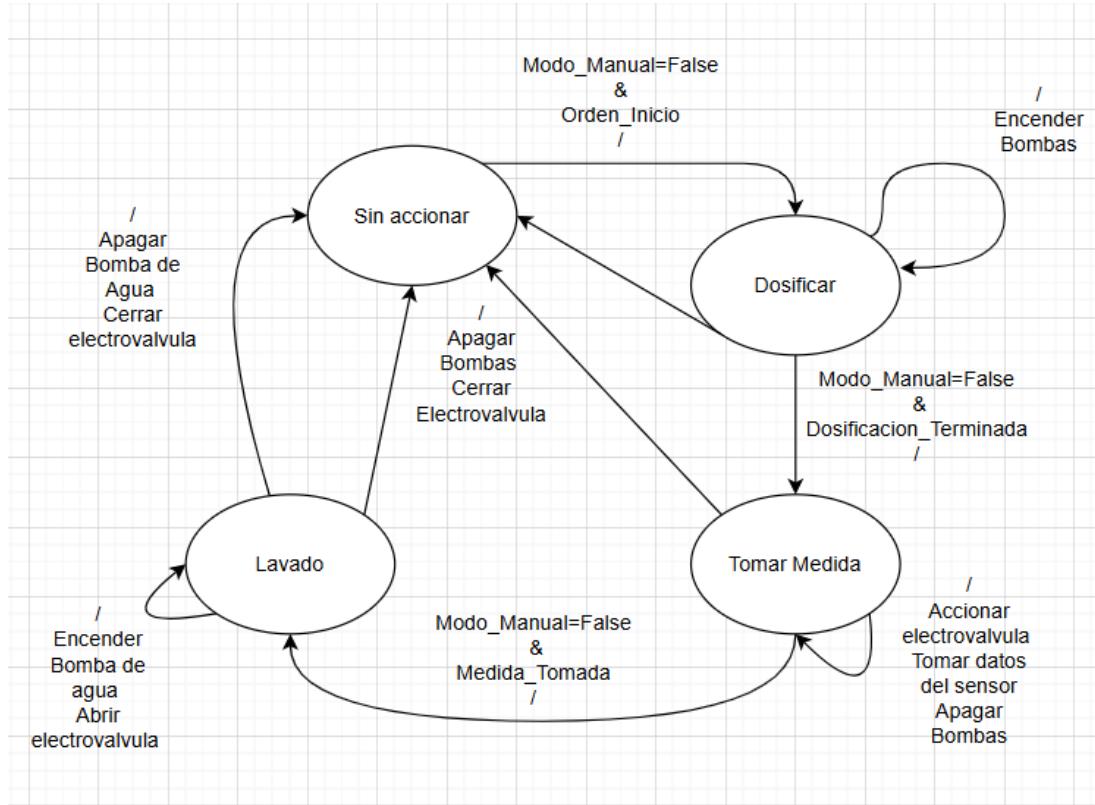


Figura C.3: Diagrama de estados proceso automático

Estado: Sin accionar

Este es el estado de reposo del sistema. Desde aquí se puede iniciar el proceso automático cuando se cumplan las condiciones:

- Modo_Manual = False
- Orden_Inicio = True

Durante este estado, se tiene como consigna:

- Se apagan todas las bombas.
- Se cierra la electroválvula.

El sistema puede volver a este estado desde cualquier etapa del proceso si se interrumpe el modo automático. En este estado se restablecen todos los temporizadores que involucran a los estados siguientes.

Estado: Dosificar

En este estado se activa la bomba correspondiente para iniciar la dosificación, permaneciendo en él hasta que se detecta la finalización del proceso de dosificación mediante la condición **Dosificación_Terminada = True**, siempre que el sistema se mantenga en modo automático (**Modo_Manual = False**).

Dado que el caudal de la bomba es proporcional a la consigna de velocidad (RPM), se implementa un modelo de calibración que traduce el volumen deseado en un tiempo de funcionamiento. Este modelo se encuentra encapsulado en el programa **Calibración_Bombas**, el cual determina el tiempo requerido para dosificar, en función de la cantidad ingresada por el usuario desde la interfaz de visualización.

Esta estrategia permite separar conceptualmente dos niveles del sistema:

- El modelo físico y calibración de la bomba, obtenido mediante ensayos experimentales.
- La lógica operativa del proceso automático, basada en ese modelo.

Una vez determinado el tiempo de operación, se utiliza un temporizador tipo TON para controlar la duración de la activación de cada bomba. Generalmente, las bombas se apagan de forma independiente, dado que los volúmenes de reactivo y agua de muestra suelen ser distintos, predominando el volumen de agua.

Como se ilustra en la figura C.4, el apagado de bombas está coordinado mediante banderas internas del programa, así como la bandera de estado activo **Dosificar**. También se consideran condiciones de interrupción manual, como la pulsación del botón **Parar** desde la visualización o la activación del **Modo_Manual**, que fuerzan la salida inmediata del estado.

En dosificación de agua y reactivo Apago las bombas por tiempo

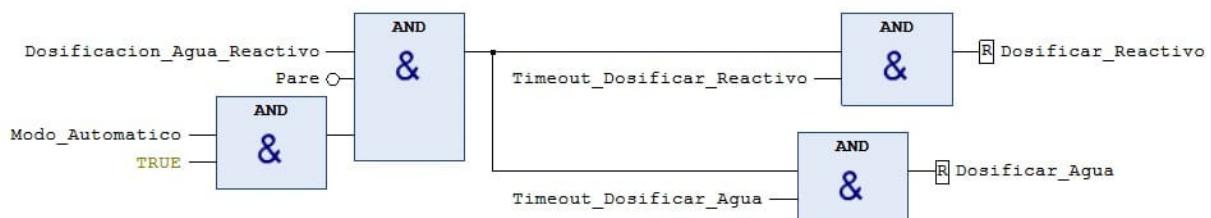


Figura C.4: Lógica de apagado de las bombas

Al final del programa principal se implementa el accionamiento de las salidas físicas del sistema (bombas y electroválvulas) en función del estado del proceso, utilizando una lógica basada en banderas internas. Un ejemplo representativo es la bandera **Dosificar_Reactivio**, la cual, al activarse, indica que el bit de salida correspondiente a la bomba de reactivo debe energizarse.

El accionamiento de cada salida se realiza al final del ciclo de programa, permitiendo una clara separación entre la lógica de control del proceso y la activación efectiva del

hardware. Esta estructura mejora la modularidad del código, facilita el mantenimiento del sistema y permite realizar modificaciones o ampliaciones de manera más sencilla y rápida.

Estado: Tomar Medida

Una vez finalizada la dosificación, el sistema avanza al estado de toma de mediciones. En esta etapa se toman las siguientes acciones:

- Acciona la electroválvula para permitir el flujo de muestra y se vuelve a cerrar.
- Se apagan las bombas.

La transición al estado lavado ocurre cuando:

- Modo_Manual = False
- Medida_Tomada = True

Durante este estado, se acciona la electroválvula con el objetivo de permitir el paso del fluido hacia el sensor. Posteriormente, se cierra nuevamente, permitiendo un tiempo de estabilización previamente definido en la lógica del programa.

Una vez transcurrido este tiempo, se procede a la adquisición de la muestra. Esta acción se acompaña de una orden generada dentro del mismo bloque funcional, que se transmite vía Modbus TCP hacia la Raspberry Pi. La Raspberry guarda el valor medido en su tabla de medidas que también se muestra en una sección dedicada a la misma en la visualización, avisa vía Modbus TCP al PLC que la medida fue registrada y en la transición de cambio de estado apaga la bandera **Medida_Tomada**. Se observa que la variable medida tomada es una variable IN OUT para el bloque proceso automático.

En la figura C.5 se ilustra el intercambio de banderas entre los estados que involucran al estado tomar medida.

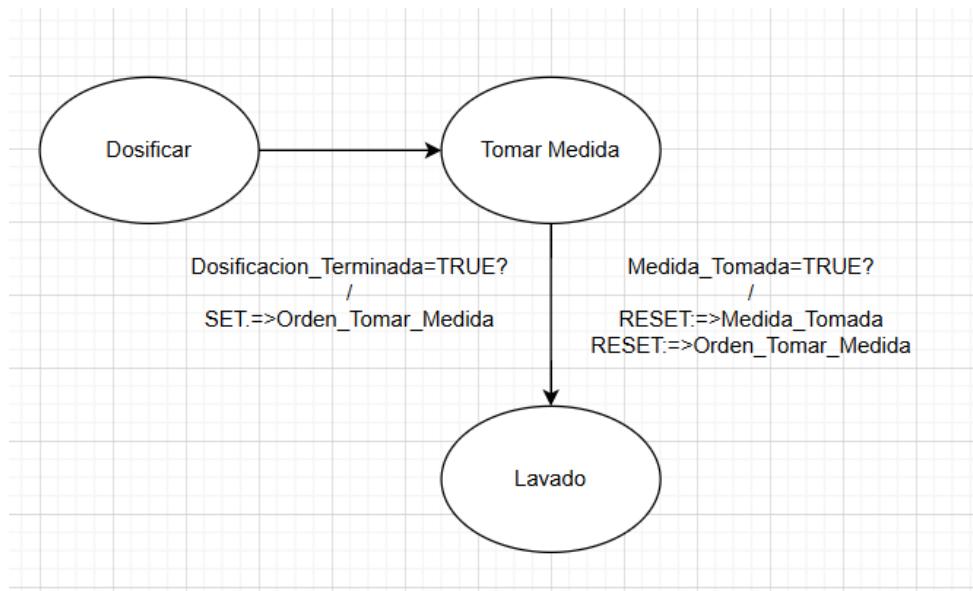


Figura C.5: Logica de banderas en estado tomar medida

Estos conceptos nos acercan más al código de programación implementado explicado de manera conceptual en un principio. Las medidas se guardan bajo una estructura creada específicamente para la aplicación, la cual se profundiza en la sección C.2.5.

Estado: Lavado

El sistema procede al lavado de las líneas y sensores para evitar contaminación cruzada en futuras mediciones. En este estado se:

- Enciende la bomba de lavado (agua).
- Se abre la electroválvula para permitir el flujo de agua.

Una vez finalizado el lavado el cual finaliza mediante un timer, el sistema regresa al estado **Sin accionar**.

Adicionalmente, si en cualquier momento el operador cambia el sistema a **modo manual**, el proceso se interrumpe y se retorna de inmediato al estado **Sin accionar**, apagando todos los actuadores.

Observaciones

- Las transiciones están claramente condicionadas por banderas internas del sistema (`Orden_Inicio`, `Dosificacion_Terminada`, `Medida_Tomada`) y por el modo de operación (`Modo_Manual`).
- La lógica asegura seguridad y consistencia en el proceso, garantizando que los actuadores solo se activen en el momento correspondiente.
- Este tipo de diseño facilita la implementación en lenguajes gráficos como **FBD** o **SFC**, cumpliendo con la norma IEC 61131-3. Se optó por utilizar **FBD** con el objetivo de reducir la extensión del código, ya que en **SFC** se generaban transiciones vacías que carecían de sentido dentro del proceso. Además, el uso de **SFC** incrementaba innecesariamente la cantidad de pasos requeridos para permitir la interrupción del proceso en modo manual en cualquier momento.
- El diagrama de la figura C.5 esta simplificado, se tiene un parámetro a ingresar correspondiente a la cantidad de veces que se quiere repetir el proceso automático.

Iteraciones del proceso automático

El proceso automático mantiene la estructura de **cuatro estados** previamente definidos:

1. **Sin accionar**
2. **Dosificar**
3. **Tomar medida**
4. **Lavado**

Parámetro de control:

- **Consigna de repeticiones:** valor asignado por el usuario desde la visualización en modo automático.
- **Contador de ciclos realizados:** variable interna que registra la cantidad de veces que el proceso ha sido completado.

Lógica de transición: Las transiciones entre los estados **Desagüe, Dosificar y Sin accionar** consideran ahora dos condiciones aparte de las descritas anteriormente:

1. Condiciones operativas establecidas para cada cambio de estado (según la lógica anterior).
2. Cumplimiento de la consigna de repeticiones, comparando el **contador de ciclos realizados** con el valor de consigna definido por el usuario.

Secuencia de finalización: Cuando se cumpla que:

$$\text{Contador de ciclos realizados} \geq \text{Consigna de repeticiones}$$

el sistema interrumpirá la secuencia automática y retornará al estado **Sin accionar**.

En el diagrama de estados de la figura C.6 se presenta la secuencia de funcionamiento considerando la nueva condición de repetición establecida por el usuario.

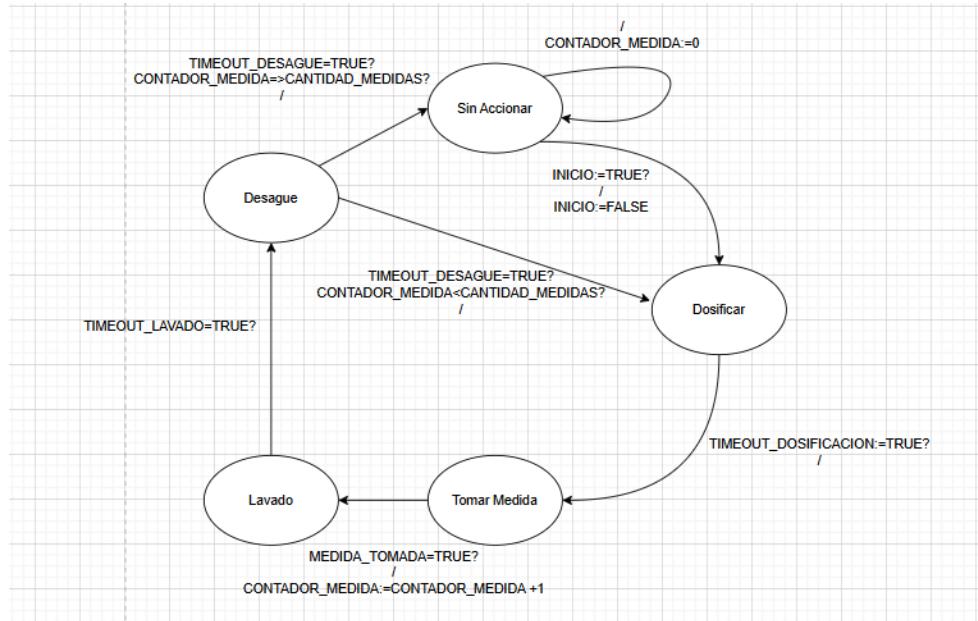


Figura C.6: Diagrama de estados con contador

Consideraciones de Funcionamiento

El bloque **Proceso_Automatico** posee variables vinculadas tanto a los parámetros configurados desde la interfaz de visualización como al estado interno del programa ejecutado en la **Raspberry Pi** coordinando la lógica del proceso con el registro de valores medidos de forma automática.

Por otro lado, los bloques correspondientes a los actuadores admiten consignas establecidas manualmente, las cuales se definen directamente a través de los parámetros disponibles en la visualización, permitiendo así la operación en **Modo Manual** de forma independiente al flujo automático.

C.1.3. Programa Copia Salidas

El siguiente programa tiene como objetivo **mapear variables globales a direcciones físicas de memoria del PLC**, lo que permite que dichas variables sean enviadas a salidas digitales, salidas analógicas o a dispositivos conectados por bus de campo. Este mapeo también podría realizarse mediante la declaración de variables con referencia de memoria utilizando la palabra clave AT, o bien asociando las variables directamente en la configuración de Hardware. En este caso, se ha optado por implementarlo en un programa escrito en *Texto Estructurado* (ST), lo cual permite, al trabajar en modo *online* con CODESYS conectado al PLC, visualizar de forma clara y práctica los valores transmitidos hacia el Hardware. También permite desacoplar la memoria de salida de las variables clave en la programación de manera rápida.

```
1 // INICIO PROGRAMA
2
3 // Mapear salidas digitales de bombas
4 SalidaDigital[%QX2.0] <- Run_Bomba_Agua
5 SalidaDigital[%QX12.0] <- Run_Bomba_Reactiva
6
7 // Mapear salidas digitales de electrovalvulas
8 SalidaDigital[%QX0.0] <- Open_Ev_Medida
9 SalidaDigital[%QX0.1] <- Open_Ev_Desague
10
11 // Mapear palabras de bus de campo (palabra) para RPM de bombas
12 SalidaWord[%QW2] <- Rpm_Bomba_Agua
13 SalidaWord[%QW7] <- Rpm_Bomba_Reactiva
14
15 // Configurar RPM maxima por medio de bus de campo
16 SalidaWord[%QW3] <- 50000
17 SalidaWord[%QW8] <- 50000
18
19 // FIN PROGRAMA
```

C.1.4. Programa Modbus Tcp (ST)

En la sección 3.1.1 se detalla el procedimiento de asociación del mapeo de variables dentro del programa. Cabe destacar que los controladores de comunicación *Modbus TCP* presentan diferencias significativas respecto a los controladores *Modbus RTU* utilizados en el puerto serial COM. El controlador para bus de campo *Modbus TCP* admite las mismas funciones *Modbus RTU*.

Es importante señalar que, a diferencia de la comunicación serial, donde únicamente puede existir un maestro, en la comunicación basada en Ethernet es posible disponer de dos maestros en funcionamiento simultáneo.

Se simplificó la comunicación usando escritura y lectura solamente de palabras completas bajo un valor predefinido. Se utilizó la función tres y se trabaja con Holding register en ambos extremos.

El siguiente pseudocódigo muestra cómo se utiliza una función IF para discriminar la variable asignada a un registro en el mapa de memoria.

Algorithm 1 Selección de modo de operación

```
1: if IW50 = 100 then
2:   Visu_Modo_Manual ← TRUE                                ▷ Modo Manual
3: else
4:   Visu_Modo_Manual ← FALSE                             ▷ Modo Automático
5: end if
```

El algoritmo de selección de modo de operación verifica el valor de la variable de entrada IW50 y, en función de este, asigna el estado de la variable global Visu_Modo_Manual. Cuando IW50 es igual a 100, el sistema interpreta que debe trabajar en **modo manual** y la variable se establece en TRUE; en cualquier otro caso, se asume el **modo automático** y la variable toma el valor FALSE. De esta forma, la lógica del programa simplifica directamente la interacción del usuario con la visualización.

El prefijo Visu indica que la variable es global, pero representa dentro de la programación que su origen proviene de la visualización.

Se detalla el pseudocódigo del programa separado en variables de escritura y lectura y se añaden comentarios generales.

Algorithm 2 Escritura de palabras Modbus desde PLC

```
1: if Orden_Tomar_Medida then
2:   QW9 ← 100
3: else
4:   QW9 ← 101
5: end if
6: if QX0.1 then                                ▷ Electroválvula Encendida
7:   QW12 ← 100
8: else
9:   QW12 ← 101                                ▷ Electroválvula Apagada
10: end if                                         ▷ rpm actual de bombas
11: QW10 ← IW2 / 100
12: QW11 ← IW26 / 100
```

Se observa que la memoria escrita es del tipo **Q**, la cual tiene una dirección asociada a un registro Modbus correspondiente dada por el fabricante de PLC.

A continuación, se describe la operativa caso por caso. La variable **Orden_Tomar_Medida** se activa desde el programa principal, específicamente en el bloque de proceso automático. Al ponerse en TRUE, tal como se muestra en el pseudocódigo, se escribe la palabra con el valor 100. Posteriormente, cuando se activa el bit de **Medida_Tomada**, el bloque de proceso automático apaga la variable **Orden_Tomar_Medida** y se escribe el valor 101. Estos valores son decodificados por el programa Modbus TCP Master en la Raspberry Pi, donde el valor 100 se interpreta como TRUE y el valor 101 como FALSE.

En cuanto al estado de la electroválvula, la salida de memoria **QX0.1** corresponde directamente a la salida física del PLC que acciona el dispositivo. Esta salida también se escribe en la palabra de comunicación con los valores ya explicados.

Finalmente, se escribe el valor actual de las RPM, leído desde los *drivers* vía Modbus RTU, asociados al *driver* de bus de campo en CODESYS en el puerto COM serial. Dicho valor se obtiene de las memorias **IW2** e **IW26**, las cuales representan las RPM actuales en formato entero con una precisión de 0.01. Por ejemplo, un valor de 10 RPM se registra como 1000. Para obtener el valor real, este registro se divide entre 100 y, posteriormente, se asigna a la palabra de comunicación asociada al canal de salida.

Algorithm 3 Lectura de palabras Modbus en PLC

```
1: if IW49 = 100 then
2:   Medida_Tomada ← TRUE
3: else
4:   Medida_Tomada ← FALSE
5: end if
6: if IW50 = 100 then
7:   Visu_Modo_Manual ← TRUE                                ▷ Modo Manual
8: else
9:   Visu_Modo_Manual ← FALSE                             ▷ Modo Automático
10: end if
11: Visu_Rpm_Manual_Bomba_Agua ← IW51
12: Visu_Rpm_Manual_Bomba_Reactiva ← IW52
13: if IW53 = 100 then
14:   Visu_Orden_Manual_Bomba_Agua ← TRUE                  ▷ Encendido
15: else
16:   Visu_Orden_Manual_Bomba_Agua ← FALSE                 ▷ Apagado
17: end if
18: if IW54 = 100 then
19:   Visu_Orden_Manual_Bomba_Reactiva ← TRUE
20: else
21:   Visu_Orden_Manual_Bomba_Reactiva ← FALSE
22: end if
23: if IW55 = 100 then
24:   Visu_Orden_Manual_Electrovalvula_Desague ← TRUE
25: else
26:   Visu_Orden_Manual_Electrovalvula_Desague ← FALSE
27: end if
28: Visu_Cantidad_Medidas ← IW56
29: if IW57 = 100 then
30:   Visu_Orden_Iniciar ← TRUE
31: else
32:   Visu_Orden_Iniciar ← FALSE
33: end if
34: if IW58 = 100 then
35:   Visu_Orden_Parar ← TRUE
36: else
37:   Visu_Orden_Parar ← FALSE
38: end if
39: Visu_mL_Reactiva ← IW59
40: Visu_mL_Agua ← IW60
```

Para el caso de lectura, se dispone de un registro de las variables obtenidas desde la visualización. Se observa que las condiciones de los IF se basan en memorias del tipo IW, correspondientes a entradas asociadas a registros de lectura en el *driver* de comunicación de bus de campo en CODESYS en el puerto ethernet. En función de dichos valores, se asignan las variables internas: en el caso de variables booleanas, el valor depende de la evaluación de la condición, mientras que en variables no booleanas la asignación se realiza de forma directa.

C.1.5. Programa Calibración Bombas (ST)

Este programa, en función de la cantidad de reactivo asignada en la visualización (expresada en mililitros), calcula el tiempo de dosificación requerido en segundos para una velocidad de agitación determinada en **rpm**. Dicha variable se toma como entrada en el programa principal, específicamente en el bloque de proceso automático, con el fin de activar los temporizadores en función de la consigna de mililitros definida por el usuario.

Se establece una misma velocidad en rpm para la dosificación en modo automático, aunque con tiempos distintos según la cantidad de reactivo. Para este caso, se eligió una velocidad de dosificación de 20 **rpm**.

Se ajusta bajo el resultado obtenido del ensayo de flujo de bombas. Bajo una consigna de mililitros determinada, se calcula el tiempo de dosificación.

Posteriormente, el resto de los valores se asigna de forma directa, dejando este espacio de código disponible para futuras modificaciones o configuraciones adicionales.

Algorithm 4 Calibracion de Bombas

```
1:                                                               ▷ Reactivo
2: Rpm_Dosificacion_Reactivo ← 2000
3: Tiempo_Dosificar_Reactivo ← (Visu_mL_Reactivo / 0,0498/ )
4:                                                               ▷ Agua
5: Rpm_Dosificacion_Agua ← 2000
6: Tiempo_Dosificacion_Agua ← (Visu_mL_Agua / 0,0498 )
7:                                                               ▷ Lavado
8: Rpm_Lavado ← 10000
9: Tiempo_Lavado ← 15
```

C.2. Software en Raspberry Pi

C.2.1. Programa I2C COM (ST)

En el puerto de hardware de la tarjeta, dentro de *CODESYS*, se encuentra configurado un *driver* I2C Master. Este *driver* es invocado desde el programa mediante las extensiones `.read` y `.write`, las cuales permiten ejecutar métodos que pueden interpretarse como funciones a nivel de hardware. A dichos métodos se les pasa como parámetros un puntero a un arreglo de bytes, la dirección del sensor OPT3002 bajo el protocolo I2C, y la cantidad de bytes que se desea leer o escribir a partir de la dirección señalada por el puntero.

El programa se organiza en ventanas de tiempo, de manera que solo se ejecuta una acción en cada instante. Para ello, se emplea un contador asociado a la variable `i`, que registra distintos ciclos del programa antes de habilitar la transición al siguiente estado. Estos ciclos están sincronizados con el ciclo de ejecución de la tarea principal en la que se corre el programa.

Se dispone de *buffers* de lectura y escritura, y el proceso se divide en dos etapas principales. La primera consiste en la configuración del sensor: en esta etapa se asignan los parámetros de inicialización a las variables del programa utilizadas por el método del *driver* I2C y se establecen los punteros a las direcciones de memoria correspondientes a los *buffers*. Se observa que, al pasar del estado *Configurar sensor* al estado *Leer configuración*, tanto el contador `i` como la variable `nBytes` se reinician. Esto último se debe a que la variable `nBytes` se comparte entre los procesos de escritura y lectura: en la fase inicial correspondiente a la configuración del sensor se escriben tres valores y se leen dos la siguiente corresponde al algoritmo cinco detallado a continuación.

Algorithm 5 Configuracion inicial

```
1: Inicio: Al encender la Raspberry, inicia el proceso automático.  
2: if not (Configuro_Sensor or Leo_Configuracion or Pedir_Datos or Leer_Datos) then  
3:   Configuro_Sensor  $\leftarrow$  true  
4: end if  
5: Configuración inicial  
6: Direccion_Esclavo  $\leftarrow$  0x44  
7: Adress_I2C_device  $\leftarrow$  0x44  
8: Pointer_In  $\leftarrow$  &Buffer_Read_I2C[0]  
9: Pointer_Out  $\leftarrow$  &Buffer_Write_I2C[0]  
10: Operacion  $\leftarrow$  I2C_master.Operational  
11: Configurar sensor  
12: if Configuro_Sensor then  
13:   nBytes  $\leftarrow$  3  
14:   Buffer_Write_I2C[0]  $\leftarrow$  0x01  
15:   Buffer_Write_I2C[1]  $\leftarrow$  0xCC  
16:   Buffer_Write_I2C[2]  $\leftarrow$  0x01  
17:   i  $\leftarrow$  i + 1  
18:   if (i > 2) and (i < 9) then  
19:     I2C_master.write(Adress_I2C_device, Pointer_Out, nBytes)  
20:   end if  
21:   if i > 10 then  
22:     Configuro_Sensor  $\leftarrow$  false  
23:     Leo_Configuracion  $\leftarrow$  true  
24:     i  $\leftarrow$  0, nBytes  $\leftarrow$  2  
25:   end if  
26: end if  
27: Leer configuración  
28: if Leo_Configuracion then  
29:   I2C_master.read(Adress_I2C_device, Pointer_In, nBytes)  
30:   i  $\leftarrow$  i + 1  
31:   if i > 5 then  
32:     Configuro_Sensor  $\leftarrow$  false  
33:     Leo_Configuracion  $\leftarrow$  false  
34:     Pedir_Datos  $\leftarrow$  true  
35:     Configuracion_Actual_1  $\leftarrow$  Buffer_Read_I2C[0]  
36:     Configuracion_Actual_2  $\leftarrow$  Buffer_Read_I2C[1]  
37:     i  $\leftarrow$  0, nBytes  $\leftarrow$  1  
38:     Buffer_Write_I2C[0]  $\leftarrow$  0x00  
39:   end if  
40: end if
```

Con el sensor configurado se reciben datos. Observar que luego de tener el sensor configurado no se vuelve a configurar ni leer la configuración luego de dar la orden de pedir datos. Para pedir datos se manda un valor de cero en la escritura. Se observa en el algoritmo anterior que el valor del primer byte del buffer es borrado y se asigna la cantidad de bytes en uno. El valor de la medida es un dato real, pero se nos da la información en dos bytes. Si ordenamos los bytes los primeros 4 bits más significativos representan el exponente en base dos de la medida a multiplicar por un factor 1.2 y por otro factor dado por los siguientes 12 bits como se explica en la sección B.3.2

Listing C.1: Lectura de datos del sensor OPT302

```

1 { Leer datos del sensor }
2 if Leer_datos then
3 begin
4     I2C_master.read(Adress_I2C_device, Pointer_In, nBytes);
5
6     Dato_Actual_1 := Buffer_Read_I2C[0];
7     Dato_Actual_2 := Buffer_Read_I2C[1];
8
9     Dato_Actual := (ByteToWord(Dato_Actual_1) shl 8) or Dato_Actual_2;
10
11    Medida_Factor := Dato_Actual and 2#0000111111111111;
12
13    Medida_Exponente_Desplazado := Dato_Actual and 2#1111000000000000;
14
15    Medida_Exponente := (Medida_Exponente_Desplazado shr 12);
16 end;
17
18 { Calculos }
19 Operacion_Exponente := 2#0000000000000001;
20 Dos_Elevado_Exponente := (Operacion_Exponente shl Medida_Exponente);
21
22 Valor_Medida := 1.2 * WordToReal(Dos_Elevado_Exponente) *
23                         WordToReal(Medida_Factor);

```

En la línea 20 se observa la operación de corrimiento del exponente en base dos, utilizada para calcular su valor exponencial. En la línea 9, los bits obtenidos se agrupan en una palabra WORD, la cual se separa mediante máscaras binarias: por un lado se extrae directamente el valor del factor de multiplicación y, por otro, el valor del exponente. Este último se obtiene tras aplicar la máscara correspondiente y realizar un desplazamiento de 12 bits hacia la izquierda. Finalmente, con ambos valores se lleva a cabo el cálculo de la medida.

C.2.2. Programa Visu (ST)

Este espacio de programa tiene como objetivo servir de apoyo a los objetos de la visualización. La visualización en el servidor web se programa de forma gráfica, y cada objeto utilizado corresponde a lo que se conoce como un *widget*, tal como también se denomina en las interfaces gráficas empleadas para programar HMIs. Un ejemplo claro de su uso se da cuando, al accionar un botón en la visualización, se genera un flanco de subida que dura unos dos ciclos de programa. Este tipo de funcionalidades se resuelven en este espacio de programa con el fin de que no interfieran con el resto de la lógica implementada, y además permitan realizar pruebas de la visualización de manera independiente del programa principal.

C.2.3. Programa Modbus TCP Master (ST)

En el PLC, el programa *Modbus TCP* gestiona de manera integrada tanto la asignación de escritura como de lectura en los registros *holding*. Sin embargo, en la tarjeta se optó por implementar esta lógica en dos programas separados. El presente programa tiene como función recibir las órdenes enviadas desde el PLC, correspondientes al algoritmo 2 descrito en la sección C.1.4, decodificando los valores clave de los registros Modbus en variables globales. Cabe señalar que el maestro Modbus se utiliza para realizar la lectura de los registros del esclavo, siguiendo la lógica de comunicación *cliente-servidor*.

C.2.4. Programa Modbus TCP Slave (ST)

Siguiendo la lógica expuesta en la sección anterior, se tiene presente que el espacio de este programa se destina a la escritura de las variables que posteriormente son leídas por el PLC en su rol de maestro. La tarea principal consiste en asignar los valores clave, ya sea 100 o 101, en los registros *holding* correspondientes, de manera que puedan ser considerados por el algoritmo 3 descrito en la sección C.1.4.

C.2.5. Programa Registro de medidas (ST)

En CODESYS se admiten datos definidos mediante estructuras denominadas *DUT* (Data Unit Type), las cuales serán utilizadas en este programa. En particular, se define una variable **medida** que contiene como parámetros la cantidad de mL de reactivo, la cantidad de mL de agua y el valor registrado por el sensor OPT3002 en un instante dado. Estas mediciones se organizan en otra estructura denominada **tabla_medidas**, con una longitud de diez elementos.

En el siguiente código se presenta el funcionamiento del programa para dos medidas a modo de ejemplo. En un desarrollo completo, el programa extendería los casos de la instrucción CASE para abarcar todas las medidas previstas.

```
1 IF (Orden_Registrar_Medida = 100) THEN
2     IF (NOT(Medida_Registrada)) THEN
3         CASE Indice OF
4             0: Indice := 1;
5                 Registro.Medida_1.mlReactivo := ValorActual_Reactiv0;
6                 Registro.Medida_1.mlAgua      := ValorActual_Agua;
7                 Registro.Medida_1.Valor       := Valor_Medido;
8             1: Indice := 0;
9                 Registro.Medida_2.mlReactivo := ValorActual_Reactiv0;
10                Registro.Medida_2.mlAgua      := ValorActual_Agua;
11                Registro.Medida_2.Valor       := Valor_Medido;
12        END_CASE
13        Medida_Registrada := TRUE;
14    END_IF
15 END_IF
16
17 IF (Orden_Registrar_Medida = 101) THEN
18     Medida_Registrada := FALSE;
19 END_IF
```

Índice de cuadros

4.1. Datos relevados en el ensayo 1	55
4.2. Datos relevados en el ensayo 2	56
4.3. Datos relevados en el ensayo 3	57
4.4. Resultados obtenidos	57
4.5. Relación entre porcentaje de colorante y absorbancia	58
4.6. Relación entre porcentaje de colorante y absorbancia	59
4.7. Resultados obtenidos	60

Índice de figuras

1.1.	Sistema completo implementado por Spymovil-Udelar (EV - electrovalvula, PU - bomba peristáltica)	11
1.2.	Sistema físico en detección de DPD - Spymovil-Udelar	12
1.3.	Equipos Micromac C instalados [8]	13
1.4.	Interior de equipos Micromac C [8]	14
1.5.	Vista interior del gabinete principal [9]	15
1.6.	Módulo de la electrónica [9]	16
1.7.	Módulo de muestreo [9]	17
1.8.	Bomba peristáltica utilizada para la toma de agua a muestrear [9]	17
2.1.	Vista frontal de placa eléctrica del Espectrofotómetro Industrial Automatizado	20
2.2.	Vista frontal de placa hidráulica del Espectrofotómetro Industrial Automatizado	21
2.3.	Menú principal	22
2.4.	Modo manual	23
2.5.	Modo automatico	24
2.6.	Diagrama funcional de componentes	26
2.7.	Diagrama de componentes principales	28
2.8.	Conversor 230 VAC / 24 VDC - UNO POWER 2902991 [10]	29
2.9.	Alimentación de Raspberry Pi - 5 VDC / 3 A	30
2.10.	Llave termomagnética ABB modelo SN201-C10	30
2.11.	Borne carril para fusible	31
2.12.	PLC Crevis GL-9972	32
2.13.	Módulo de salidas digitales CREVIS GT-226F	32
2.14.	Raspberry pi 3b+	33
2.15.	Numerado de pines en Raspberry pi 3b+	34
2.16.	Bomba micro-peristáltica kamoer KMC [3]	35
2.17.	Elementos que forman a las bombas peristálticas kamoer [3]	36
2.18.	rpm vs caudal de salida (azul - caño 1.6 mm) [3]	36
2.19.	Placa de drivers 4460.5 [5]	37
2.20.	Electroválvula Heschen modelo 2V025-08	38
2.21.	Relé de estado sólido	39
2.22.	Zócalo para relé	39

2.23. Esquemático la placa donde se monta el sensor - luz a digital - OPT3002 [2]	40
2.24. Esquemático de entradas del sensor OPT3002	41
2.25. Switch de puertos ethernet	42
2.26. Vista frontal del prototipo	43
2.27. Cubeta de cuarzo - 4,5 x 1,2 x 1,2 cm	44
2.28. Diseño 3D de cámara para test óptico	44
2.29. Placa Metálica Perforada Schneider - 500 x 700 mm	45
 3.1. Esquema de mapeo de variables globales hacia salidas físicas y de bus de campo en CODESYS.	49
3.2. Interacción de programas y servidores en dispositivos	50
3.3. Diagrama de flujo en tarea principal	51
3.4. Diagrama de flujo en tarea principal	54
 4.1. Curva de absorbancia en función de porcentaje de colorante en la solución - Ensayo de referencia	59
4.2. Curva de absorbancia en función de porcentaje de colorante en la solución - Prueba con prototipo	60
4.3. Curva de absorbancia en función de porcentaje de colorante en la solución con ajuste lineal - Ensayo de referencia	61
4.4. Curva de absorbancia en función de porcentaje de colorante en la solución con ajuste lineal - Prueba con prototipo	61
 5.1. Ejemplo de tablero industrial con control mediante HMI	64
 B.1. Bobinado de motor paso a paso	67
B.2. Modelo del rotor en motor paso a paso	68
B.3. La luz que no es absorbida por el objeto es reflejada y puede verse por el ojo. [1]	69
B.4. La energía de las ondas electromagnéticas aumenta a medida que disminuye la longitud de onda y viceversa. [1]	70
B.5. El espectro visible (390-780 nm) representa solo una pequeña parte del espectro electromagnético completo. [1]	70
B.6. La luz que pasa a través de la solución de muestra es parcialmente absorbida por los componentes. [1]	71
B.7. Principio y componentes básicos de medición en espectroscopía UV/Vis [1]	72
B.8. Atenuación de la luz causada por la absorción de la solución. [1]	72
B.9. Espectro de transmisión de una solución particular en función de la longitud de onda. [1]	74
B.10. Espectros de absorción de soluciones de holmio en función de la longitud de onda. [1]	75
B.11. La atenuación de la intensidad de la luz es proporcional a la concentración de la solución de muestra y a la longitud de la cubeta. [1]	76
B.12. Cuanto mayor es la concentración, mayor es el valor de absorbancia. [1]	77

B.13. Absorción medida en solución al 1% de aceite de oliva virgen extra (azul) y aceite de oliva virgen (verde) [1]	78
B.14. Serie de soluciones patrón de fosfato preparadas [1]	79
B.15. Espectros de absorción trazados para las concentraciones correspondientes. [1]	80
B.16. Valores de absorbancia a 880 nm frente a la concentración de cada solución patrón para generar la curva de calibración. La concentración desconocida de fosfato en la muestra puede determinarse a partir de una curva de regresión lineal. [1]	80
B.17. Estructura de las tramas de comunicación modbus rtu - holding registers	81
B.18. Estructura de la trama enviada a sensor OPT3002 [4]	83
B.19. Estructura de la trama recibida por el sensor OPT3002 [4]	84
C.1. Diagrama de estados bloques Electroválvula y Bomba	88
C.2. Bloque SEL en FBD	89
C.3. Diagrama de estados proceso automático	92
C.4. Lógica de apagado de las bombas	93
C.5. Logica de banderas en estado tomar medida	95
C.6. Diagrama de estados con contador	98

Bibliografía

- [1] METTLER TOLEDO Group, UV/Vis Spectrophotometry - Fundamentals and Applications, METTLER TOLEDO , 2025.
- [2] Texas Instruments , OPT3002EVM User's Guide (Hoja de datos).
- [3] kamoer , Micro peristaltic pump KMC (Manual).
- [4] Texas Instruments , OPT3002 Light-to-Digital Sensor (Manual).
- [5] kamoer , 4460.5 Stepper Control and Drive Board (Manual).
- [6] SYSTE A , Micromac C, hoja de datos: www.enotek.com.tr/indir.php?ID=225
- [7] SYSTE A: www.systea.it/en/our-products/online-analyzers/micromac
- [8] SYSTE A: vietan-enviro.com/en/water-quality-analyzer-micromac-systea
- [9] Lic. Pablo Ismael González Cazzola, Tesis de Maestría en Química - Desarrollo de un analizador automático para la determinación de nutrientes en aguas superficiales, 2019.
- [10] Phoenix contact, UNO POWER 2902991, hoja de datos:
www.phoenixcontact.com/en-pc/products/power-supply-unit-uno-ps-1ac-24dc-30w-2902991
- [11] CREVIS, PLC GL-9972, hoja de datos:
<https://www.crevis.co.kr/Product/productView?idx=737>