Sistema de entrenamiento modular didáctico para la introducción a la automatización y control con énfasis en la industria 4.0

ImporteQ



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Autor
Jean Carlo Leguizamón Martínez
Nicolas Sopo Bobadilla

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
Facultad Tecnológica
Ingeniería en (Control y automatización)

Bogotá D.C. 08, 2025

Sistema de entrenamiento modular didáctico para la introducción a la automatización y control con énfasis en la industria 4.0

ImporteQ



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Autor
Jean Carlo Leguizamón Martínez, 20231583010,
jcleguizamonm@udistrital.edu.co
Nicolás Sopo Bobadilla, 20231583009,
nsopob@udistrital.edu.co

INFORME FINAL DE PASANTÍA

Presentado para optar al título de: Ingeniero(a) en (Control y Automatización)

Director Interno
Frank Nixon Giraldo

Director Externo
Edward Hernando Bejarano Barreto

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
Facultad Tecnológica
Ingeniería en (Control/ Telecomunicaciones)
Bogotá D.C. 08, 2025

Dedicatoria

Este trabajo está dedicado a nuestra familia, los pilares fundamentales de nuestra vida, a las personas que estuvieron en todo el proceso y nos acompañaron incondicionalmente. A todas las personas y compañeros que nos cruzamos y nos enseñaron algo significativo en el día a día.

Agradecemos especialmente a ImporteQ SAS y todo su equipo de trabajo por abrirnos las puertas de su empresa y permitirnos explorar nuestras capacidades, por ser parte de nuestra familia todo este tiempo y contribuir a nuestro crecimiento profesional y personal. Al ingeniero Edward Hernando por dejarnos ser parte de esta aventura...

Índice

Índice ii
Índice de Figuras v
Índice de Tablas viii
Índice de Anexos ix
Glosario x
Lista de Abreviaturas y Siglas xi
Resumen xii
1. Introducción 1
2. Presentación de la Empresa 2
3. Planteamiento del problema en la Empresa 3
4. Justificación 4
Objetivos 5
4.1. Objetivo General 5
4.2. Objetivos específicos 5
5. Marco de referencia 6
5.1. Antecedentes 6
5.2. Marco teórico 11
5.2.1. IoT 11
5.2.2. Protocolo Modbus 14
5.2.3. MQTT 15
5.2.4. SCADA 16
5.2.5. Control PID 17
5.2.5.1. Control proporcional 17
- Ventajas 19
- Desventajas 19

5.2.5.2. Control integral 20
- Ventajas 21
- Desventajas 21

5.2.5.3. Control derivativo 22
- Ventajas 23
- Desventajas 23

5.3. Marco Legal 24
5.3.1. ley 842 de 2003 24
5.3.2. DECRETO 585 1991 24
5.3.3. Ley 1286 de 2009 24
5.3.4. La Resolución 2400 de 1979 25
5.3.5. Norma RETIE (Reglamento Técnico de Instalaciones Eléctricas): 25
5.3.6. Ley 1341 de 2009 25
5.3.7. DECRETO 3573 DE 2011: 26
6. Actividades realizadas durante la pasantía 27

6.1. Cronograma de actividades Nicolas Sopo Bobadilla 27
6.1.1. Configuración servidor HiveMQ 28
6.1.2. Configuración placa ESP32 32
6.1.3. Pruebas de envió de datos de ESP32 a HiveMQ 32
6.1.4. Instalación Node-Red 33
6.1.5. Enlace HiveMQ – Node-Red 36
6.1.6. Configuración PLC 40
6.1.7. Conexión ModBus PLC Unitronics y apertura de puertos ethernet 42
6.1.8. Prueba conexión PLC-ESP32 por ModBus 45
6.1.9. Diseño HMI SCADA 47
6.1.10. Diseño de placa comunicación PLC-ESP32 48
6.1.11. Prueba final comunicación PLC-SCADA 48
6.1.12. Conexión PLC con teléfono móvil 49
6.1.13. Prueba completa del sistema con el servidor 49
7. Resultados 51
7.1. Compra de elementos 51
7.1.1. Plano 3D 56
7.1.2. Diagrama P&ID y planos eléctricos 60
Tanques 61
Instrumentación de nivel 61
Instrumentación de temperatura 61
Calefactor 61
Instrumentación de caudal y presión 61
Válvulas de control 61
Válvula manual 61
Bomba 61
7.1.3. Diseño tablero 62
7.1.4. Montaje de tablero 62
7.1.5. Montaje estructura 64
7.1.6. Pruebas de funcionamiento sistema de circulacion 67
7.1.7. Placa de comunicación ESP32 68
7.1.8. Contenedor circuito de comunicación 71
7.1.9. Control On-Off de nivel 72
7.1.10. Control PID de nivel 76
7.1.11. Control On-Off de temperatura 83
7.1.12. Control PID de temperatura 85
7.1.13. Control On-Off flujo 88
7.1.14. Control PID flujo 94
7.1.15. Comunicación entre dispositivos 96
7.1.16. Diseño del SCADA 103
7.1.17. Diseño HMI SEDAI 109
7.1.18. Actividades 110
8. Conclusiones y Recomendaciones 113
Recomendaciones: 113
9. Referencias 114
10. Anexos 117
- Código ESP32 117

- Planos Eléctricos 121
- Códigos Visilogic 128
  - 2) Control ON/OFF temperatura 132
  - 3) Control PID Temperatura 133
  - 4) Control ON/OFF Nivel 136
  - 5) Control PID Nivel 138
  - 6) Control ON/OFF flujo 141
  - 7) Control PID Flujo 143

Índice de Figuras

Figura 1. Conexión Node-Red [23] 12
Figura 2. En la interfaz puede haber distintos tipos de graficas y visualizaciones de datos [24] 13
Figura 3. Forma de os paquetes de mensajes en Modbus RTU y ModBus ASCII tomado de [26] 15
Figura 4. Forma de los paquetes de mensajes en Modbus TCP [26]. 15
Figura 5. SCADA de circulación y control de temperatura de mezclas, tomado de [31] ... 16
Figura 6. Efecto de la ganancia del control proporcional tomado de [33] 17
Figura 7. Ubicación del Bias en el esquema de control [Fuente propia] 19
Figura 8. Incorporación del controlador integral al controlador proporcional [Fuente Propia] 20
Figura 9. Cambio de variable integral en respuesta a cambio en una variable [35] 21
Figura 10. Efecto de una acción derivativa grande [Fuente Propia] 22
Figura 11. Efecto de una acción derivativa pequeña [Fuente Propia] 22
Figura 12. Diagrama de Gantt Nicolas Sopo Bobadilla 27
Figura 13. Creación del cluster [Fuente propia] 28
Figura 14. Los planes del HiveMQ pueden cambiar dependiendo de la aplicación y el consumo 29
Figura 15. Panel del cluster [Fuente propia] 29
Figura 16. Gestión de accesos del cluster [Fuente propia] 30
Figura 17. Creación de usuario y contraseña para el acceso [Fuente propia] 31
Figura 18. En la conexión web se pueden ver mensajes de los tópicos declarados o de todos los tópicos recibidos [Fuente propia] 31
Figura 19. Función Setup y función Loop [Fuente propia] 33
Figura 20. Mensaje recibido del tópico "Valor1" [Fuente propia] 33
Figura 21. Versión de descarga de Node.Js para SEDA V22.109.0 LTS [Fuente propia] . 34
Figura 22. Instalación Node-Red [Fuente propia] 35
Figura 23. Ejecución de Node red desde la consola de comandos (CMD)[Fuente propia] 35
Figura 24. Zona de trabajo Node-red [Fuente propia] 36
Figura 25. Nodos para la conexión MQTT [Fuente propia] 36
Figura 26. Nodo "MQTT in" [Fuente propia] 37
Figura 27. Configuración Nodo "MQTT in" [Fuente propia] 37
Figura 28. Configuración seguridad Nodo "MQTT in" con parámetros de HiveMQ [Fuente propia] 38
Figura 29. Configuración tópico "Valor1" [Fuente propia] 39
Figura 30. Conexión de nodo "MQTT in" a la consola de Node-Red [Fuente propia] 39
Figura 31. Valor tópico "Valor1" [Fuente propia] 40
Figura 32. Descarga de VisiLogic [Fuente propia] 40
Figura 33. Configuración del modelo de PLC-HMI en VisiLogic [Fuente propia] 41
Figura 34. Configuración HMI [Fuente propia] 42
Figura 35. Creación del usuario y contraseña [Fuente propia] 43
Figura 36. Apertura del puerto ethernet [Fuente propia] 44
Figura 37. Bloque de COM y ModBus [Fuente propia] 44
Figura 38. Configuración del puerto [Fuente propia] 45

Figura 39. Configuración bloque ModBus [Fuente propia] 45
Figura 40. Configuración Modbus Poll [Fuente propia] 46
Figura 41. Datos recibidos [Fuente propia] 46
Figura 42. Visualización en HMI [Fuente propia] 46
Figura 43. Placa de comunicación ModBus, modelo en 3D [Fuente propia] 48
Figura 44. Remote Operator [Fuente propia] 49
Figura 45. Visualización guías 50
Figura 46. Diseño 3D prototipo SEDAI [Fuente propia] 56
Figura 48. SEDAI: Tanques [Fuente propia] 58
Figura 49. Diseño Tablero en Tinkercad [Fuente propia] 59
Figura 50. Vista superior [Fuente propia] 59
Figura 51. Diagrama P&ID [Fuente propia] 60
Figura 52. Plano tablero eléctrico 3D EPLAN Pro Panel 2022 [Fuente propia] 62
Figura 53. Montaje exterior y diseño interior 63
Figura 54. Cableado de tablero [Fuente propia] 63
Figura 55. Cableado PLC-HMI [Fuente propia] 64
Figura 56. Montaje tanques con tablero eléctrico en estructura [Fuente propia] 65
Figura 57.Cableado sensores y actuadores a tablero eléctrico [Fuente propia] 66
Figura 58. Vista lateral sistema de circulación [Fuente propia] 66
Figura 59. Construcción SEDAI [Fuente propia] 67
Figura 60. Ingeniero Jean Carlo Leguizamón realizando pruebas de circulación de caudal [Fuente propia] 68
Figura 61. Conexión modem nulo con protocolo de enlace de bucle hacia atrás de Lammert Bies [41] 69
Figura 62. Circuito de trazado de placas realizado en Kicad [Fuente propia] 70
Figura 63. Esquema final tarjeta de comunicación Modbus-MQTT [Fuente propia] 70
Figura 64. Circuito de comunicación soldado con conversos MAX3232 y MJ10-22-CS25 [Fuente propia] 71
Figura 65. Diseño contenedor circuito en Tinkercad [Fuente propia] 71
Figura 66. Instalación módulo de comunicación en tablero eléctrico [Fuente propia] 72
Figura 67. Grafica linealización sensor TL-136 [Fuente propia] 73
Figura 68. Configuración entrada análoga [Fuente propia] 73
Figura 69. Filtro digital sensor nivel [Fuente propia] 74
Figura 60. Grafica control nivel On-Off -1 [Fuente propia] 75
Figura 61. Grafica control nivel On-Off -2 [Fuente propia] 75
Figura 62. Grafica control nivela On-Off -3 [Fuente propia] 76
Figura 63. Configuración bloque PID control nivel [Fuente propia] 78
Figura 64. Control PID, primera prueba nivel-1 [Fuente propia] 78
Figura 65. Control PID, primera prueba nivel-2 [Fuente propia] 79
Figura 66. Configuración bloque PID, segunda prueba [Fuente propia] 79
Figura 67. Control PID nivel, segunda prueba-1 [Fuente propia] 80
..... 80
Figura 68. Control PID nivel, segunda prueba-2 [Fuente propia] 80
..... 81
Figura 69. Control PID nivel, segunda prueba-3 [Fuente propia] 81
Figura 70. Grafica linealización sensor TL-136 [Fuente propia] 81
Figura 71. Control PID nivel, tercera prueba-1 [Fuente propia] 82

Figura 72. Control PID nivel, tercera prueba-2 [Fuente propia] 82
Figura 73. Control PID nivel, tercera prueba-3 [Fuente propia] 83
Figura 74. Configuración punto decimal en campo de texto [Fuente propia] 84
Figura 75. Control On-Off temperatura-1 [Fuente propia] 84
Figura 76. Control On-Off temperature-2 [Fuente propia] 85
Figura 78. Control PID temperatura-1 [Fuente propia] 86
..... 86
Figura 79. Control PID temperatura-2 [Fuente propia] 86
..... 87
Figura 80. Control PID temperatura-3 [Fuente propia] 87
..... 87
Figura 81. Control PID temperatura-4 [Fuente propia] 87
Figura 82. Control PID temperatura-5 [Fuente propia] 88
Figura 83. Señal del sensor efecto hall YF-S201 [Fuente propia] 89
Figura 84. Programación de variables [Fuente propia] 89
Figura 85. Programación de la interrupción y ModBus [Fuente propia] 90
Figura 86. Envío de dato de flujo por vía Modbus [Fuente propia] 90
Figura 87. Programación control, On-Off, proceso flujo [Fuente propia] 91
Figura 88. Diseño HMI ventana inicial [Fuente propia] 92
Figura 89. Control On-Off de flujo-1 [Fuente propia] 92
Figura 90. Control On-Off de flujo-2 [Fuente propia] 93
..... 93
Figura 91. Control On-Off de flujo-3 [Fuente propia] 93
Figura 92. Configuración bloque PID, control caudal [Fuente propia] 94
Figura 93. Control PID de flujo-1 [Fuente propia] 94
Figura 94. Control PID de flujo-2 [Fuente propia] 95
Figura 95. Control PID de flujo-3 [Fuente propia] 95
Figura 96. Válvula proporcional y sensor de flujo YF-S201 [Fuente propia] 96
Figura 97. Diagrama de flujo SEDAI [Fuente propia] 98
Figura 98. Librerías y variables, módulo de comunicación [Fuente propia] 99
Figura 99. Conexión wifi [Fuente propia] 100
Figura 100. Conexión MQTT [Fuente propia] 100
Figura 101. Lectura registros comunicación ModBus [Fuente propia] 101
Figura 102. Recepción de temperatura en HiveMQ [Fuente propia] 102
Figura 103. Lectura temperatura de tópico "Valor1" 102
Figura 104. Efecto de una acción derivativa grande [Fuente propia] 103
Figura 105. Nodos de tópicos enlazados al SCADA [Fuente propia] 104
..... 104
Figura 106. Configuración archivo SVG [Fuente propia] 104
Figura 107. Configuración archivo SVG en Node-red [Fuente propia] 105
Figura 108. Programación tópico "Valor1" [Fuente propia] 105
Figura 109. Programación tópico "Valor2" [Fuente propia] 106
..... 106
Figura 110. Programación Variable booleanas [Fuente propia] 106
Figura 111. SCADA SEDAI [Fuente propia] 107
Figura 112. SCADA SEDAI [Fuente propia] 109
Figura 113. SCADA SEDAI [Fuente propia] 109

Índice de Tablas

Tabla 1. Learning factorys, adaptada de [18]. 10
Tabla 2. Elementos comprados [Fuente Propia]. 51
Tabla 3. Conexión de modem nulo, adaptado de [41]. 69
Tabla 4. Datos de linealización [Fuente propia]. 72

Índice de Anexos

10. Anexos 117
10.1. Código ESP32 117
10.2. Planos Eléctricos 121
10.3. Códigos Visilogic 128
10.3.2. 2) Control ON/OFF temperatura 132
10.3.3. 3) Control PID Temperatura 133
10.3.4. 4) Control ON/OFF Nivel 136
10.3.5. 5) Control PID Nivel 138
10.3.6. 6) Control ON/OFF flujo 141
10.3.7. 7) Control PID Flujo 143

Glosario

Concepto	Explicación breve y detallada del concepto
accesibilidad	La condición de ser accesible. Por ejemplo, una carretera transitable todo el año proporciona mejor accesibilidad que una vía transitable en tiempo seco.
Control	Acción de regular variables de un sistema para mantenerlas en valores deseados.
PID	Algoritmo de control que combina acciones proporcional, integral y derivativa para optimizar la respuesta de un sistema.
Modbus RTU	Protocolo de comunicación industrial basado en transmisión serial para intercambio de datos entre dispositivos.
Protocolo MQTT	Protocolo ligero de mensajería orientado a la comunicación máquina a máquina (M2M) y IoT.
Middleware	Capa intermedia de software que facilita la comunicación y gestión de datos entre aplicaciones o dispositivos.
SCADA	Sistema de supervisión y adquisición de datos que monitorea y controla procesos a distancia.
Control On-OFF	Método de control que activa o desactiva un actuador según la variable de proceso esté por debajo o por encima de un valor establecido.

Lista de Abreviaturas y Siglas

Sigla/Abreviatura	Significado
RETIE	Reglamento Técnico de Instalaciones Eléctricas
MQTT	Message Queuing Telemetry Transport
IoT	Internet of things
PLC	Controlador lógico programable
PCB	Placa de circuito impreso
HMI	Interfaz Humano-Maquina
P&ID	Diagrama de Tuberías e Instrumentación
RTU	Unidad Terminal Remota
Control PID	Control Proporcional-Integral-Dervativo
SCADA	Sistema de supervisión y adquisición de datos que monitorea y controla procesos a distancia.
UI	User Interface

Resumen

En el contexto colombiano la automatización es un elemento que desde hace un tiempo ha venido transformado la industria, los nuevos enfoques y tecnologías permiten al sector empresarial implementar soluciones que cubran la demanda producida por un país que está en camino al desarrollo. Las universidades tienen como deber orientar y anticipar sus programas en pro de brindar soluciones a los retos a los que se enfrenta la industria colombiana apropiándose de tecnologías emergentes, es por ello, que los contenidos académicos deben están enriquecidos con equipos que acerquen al estudiante y al docente a formar habilidades teóricas y prácticas. SEDAI es un proyecto realizado con el fin de cerrar brechas de habilidades prácticas en automatización industrial presente en la educación técnica actual. Realizar una unión entre la parte teórica y práctica resulta fundamental en la formación académica del estudiante y en la profundización del docente hacia nuevo conocimiento y formas de aprendizaje.

La industria moderna, impulsada por la demanda y la Industria 4.0, requiere profesionales capaces de manejar y optimizar sistemas automatizados mediante la programación y la integración tecnológica. Sin embargo, muchas instituciones educativas no han adaptado sus currículos para incluir prácticas sólidas en automatización, programación y tecnologías emergentes, lo que genera una brecha entre las habilidades con las que egresan los estudiantes y las que demanda el sector industrial. La enseñanza, especialmente en ingeniería, necesita experiencias didácticas que vinculen la teoría con la práctica, fomentando habilidades críticas como el razonamiento lógico y el método científico. No obstante, la falta de herramientas, simulaciones y equipos especializados limita el desarrollo de competencias en automatización.

Con el fin de brindar una solución integral, se desarrolla un prototipo de planta de control de nivel y temperatura, incorporando un módulo que actúa como middleware entre la planta y los datos enviados al sistema SCADA implementado en Node-RED. El proyecto abarca desde el diseño hasta la construcción completa, incluyendo sensores, actuadores y demás componentes que conformarán el sistema de entrenamiento. El objetivo principal del montaje es facilitar que estudiantes y docentes identifiquen de manera clara las partes del sistema, optimizando así el desarrollo de las prácticas para las cuales la planta ha sido diseñada. SEDAI contara con unas guías enfocadas a la pedagogía, sirviendo como material de apoyo para la correcta manipulación y cuidado del sistema.

Finalmente se obtiene un prototipo funcional en el cual se pueden aplicar diferentes tipos de control a las variables de temperatura y nivel, logrando la comunicación a tiempo real mediante el protocolo de comunicación MQTT y Modbus lo que permite supervisar todo el estado del sistema durante su operación. Con las guías y talleres se explica los procesos completos y el accionamiento de cada componente, explicando practicas aumentando el nivel de grado de complejidad en cada actividad. Entre los entregables se encuentran códigos del PLC y del ESP32 empleados en las guías que funcionaran como base para los ejercicios que se propone profundizar. El proyecto SEDAI ofrece una solución práctica para reducir la brecha de habilidades en automatización industrial en Colombia, mediante un prototipo funcional de planta de control con tecnologías de Industria 4.0

Palabras claves: Control, industria 4.0, sistema, PLC, protocolo, Programación

1. Introducción

La formación en automatización industrial requiere cada vez más de equipos pedagógicos especializados que permitan trasladar los conceptos teóricos al ámbito práctico, mediante equipos que integren tecnológicas que estén a la vanguardia de los avances en la industria. Estos dispositivos de entrenamiento constituyen una herramienta fundamental para el aprendizaje, ya que facilitan la comprensión de sistemas de control, instrumentación y comunicación aplicados en la industria Las necesidades prácticas de la automatización y el control surgen de la creciente complejidad de los sistemas de producción, que requieren mantener altos niveles de eficiencia, seguridad y adaptabilidad. En este sentido, el uso de kits didácticos basados en PLC, sensores, actuadores y módulos de comunicación proporciona un espacio idóneo para el desarrollo de competencias en la programación, integración y optimización de procesos de automatización y control [1].

Aunque existen equipos que sirven como instrumentos de capacitación en el manejo de PLC, HMI y procesos es importante realizar un énfasis hacia donde se encamina los retos que traen consigo una industria que crece, se transforma y evoluciona con la sociedad. En este marco, los avances de la Industria 4.0 han transformado los procesos industriales mediante la integración de tecnologías digitales, conectividad, análisis de datos y sistemas ciber físicos. Estas innovaciones han generado nuevas demandas en la formación académica, donde se hace imprescindible incluir prácticas con equipos que reflejen escenarios propios de la digitalización, el internet industrial de las cosas (IIoT) y la supervisión remota de procesos. Mantener una conexión no solo brinda una producción más eficaz, si no también establece nuevos parámetros de seguridad, donde prima el bienestar de la población y la integridad de los procesos [2]

Capacitar al estudiante en este tipo de prácticas mediante equipos que ofrezcan la integración de tecnologías permiten adquirir experiencia desde la formación, fomentando la resolución de problemas, conocimiento de protocolos industriales, IoT e integración de tecnología. Estas habilidades tienen una demanda alta dentro del sector empresarial. En el presente documento se presentará un prototipo de equipo que realiza una integración de tecnologías y pretenden ser un instrumento de capacitación en diferentes áreas del control y la automatización. Asimismo, se detallan las etapas de diseño, montaje, programación y pruebas realizadas, con el fin de que este documento sirva como guía de referencia para futuros proyectos académicos y como base para la introducción pedagógica de conceptos de automatización industrial.

2. Presentación de la Empresa

ImporteQ S.A.S es una compañía con más de 10 años de experiencia en la importación de diversos productos tecnológicos, brindando servicios como proveedores y distribuidores en múltiples empresas y universidades del país. Tienen experiencia en el mantenimiento y reparación de equipos eléctricos y electrónicos industrial y de laboratorio. El equipo de trabajo especializado brinda soluciones a diferentes problemas de ingeniería en el campo de la automatización y optimización de procesos industriales, ofreciendo asesoramiento oportuno, para encontrar la mejor alternativa dependiendo de las necesidades. La política tiene por objeto el bienestar del cliente y el cumplimiento del deber en las mejores condiciones posibles motivando la innovación tecnológica día tras día

Misión:
Somos una empresa que opera en la ciudad de Bogotá D.C., con cobertura a nivel nacional, nos especializamos en brindar soluciones efectivas de abastecimiento, diagnóstico, reparación y asesoría en equipos eléctricos y electrónicos industriales y de laboratorio, también ofrecemos el servicio de aprovisionamiento de diversos insumos y soluciones efectivas, para el sector educativo, empresarial e industrial.

Visión:
Para el 2030 seremos reconocidos como una compañía líder a nivel nacional en la distribución de productos y servicios relacionados con el comercio integral de equipos eléctricos y electrónicos, guiados por el desarrollo e innovación de nuestros procesos y el desarrollo integral de nuestros colaboradores.

Estructura Organizacional:

- Gerente General: Edward Hernando Bejarano Barreto
- Departamento Comercial : Zully Paola Ferreira Capera
- Departamento Técnico: Jhon Alexander Chávez
- Departamento Administrativo: Angie Lizeth Bautista

Área de trabajo:

- Se desarrolla el proyecto en el área de automatización y control

3. Planteamiento del problema en la Empresa

Como parte de ofrecer soluciones a diferentes universidades a nivel nacional y a empresas del sector, se evidencia que programas de ingeniería como electrónica, control y automatización necesitan equipos que se adecuen a los nuevos desafíos que enfrenta la industria moderna, una industria que está en continuo cambio y desarrollo. Se evidencia que los equipos actuales no están enfocados en tecnologías emergentes como la industria 4.0 o IoT, es un mercado que está surgiendo y diferentes empresas del sector como Siemens, Allen Bradley, Unitronics, entre otros están potenciando gracias a tecnologías como la nube. Las universidades son parte fundamental de esa transición es por ello que la adquisición de equipos que brinden a los estudiantes nuevas formas de aprendizaje resulta en una solución desde la academia.

En las universidades se evidencia que no enriquecer sus programas con equipos y metodologías de aprendizaje innovador provocan dos consecuencias principales: la falta de interés de la comunidad en cursar carreras profesionales y la falta de formación práctica, en este caso, del manejo de equipos de automatización y control. Estas problemáticas afectan directamente a las universidades y a la industria en la falta de ingenieros preparados.

En la enseñanza, especialmente en carreras como la ingeniería donde los conceptos y teorías aprendidas deben ser puestas en práctica es fundamental crear experiencias didácticas que permitan al estudiante y al docente abordar problemas de forma real y tangible. Este enfoque busca desarrollar una comprensión más profunda de los conceptos teóricos a través de la experiencia y la experimentación, lo que fomenta habilidades críticas como la observación, el razonamiento lógico y el método científico [3]. Uno de los retos que afrontan las instituciones educativas es la falta de herramientas y recursos que permitan a los estudiantes desarrollar prácticas que incentiven el conocimiento en diferentes áreas del control y automatización. La falta de simulaciones en el campo industrial priva al estudiante de adquirir experiencia, lo que puede crear una brecha significativa entre el conocimiento académico y las habilidades requeridas en el entorno laboral

Muchos de los equipos especializados en la enseñanza de control y automatización son excesivamente caros, limitando la adquisición para ciertas universidades e instintos donde se imparte esta área. Estas barreras dificultan la implementación de programas de capacitación que permitan a los estudiantes desarrollar competencias en automatización industrial de manera accesible. Basado en ese contexto surge como pregunta:

¿Cómo diseñar e implementar un sistema de entrenamiento didáctico que ofrezca a los estudiantes un aprendizaje práctico y efectivo en automatización industrial, integrando tecnologías de la Industria 4.0?

4. Justificación

La necesidad de profesionales capacitados en automatización industrial está en constante crecimiento, un sector que evoluciona rápidamente y demanda habilidades prácticas más avanzadas. A lo que anteriormente era el mundo de la automatización en donde el ingeniero estaba regido por computadoras análogas, lógica de relés, control neumático e hidráulico, ahora con avance de la tecnología e internet, la automatización es todo un mundo de competencias que día a día exigen el conocimiento de gran variedad de diciplinas, como son la programación, instrumentación, manejos de datos y protocolos de comunicación [4]. Con la introducción del IoT trajo nuevos paradigmas de la industria, la producción tomo un rumbo más automatizado e "inteligente" por lo que esta diciplina significo una revolución. Dando inicio a lo que se conoce como fabrica digital [4].

Muchas instituciones educativas enfocan su enseñanza en la teoría, dejando a los estudiantes sin la experiencia necesaria para enfrentar situaciones reales en entornos automatizados. Esta carencia crea una brecha significativa entre la formación académica y las competencias requeridas por la industria, lo que impacta negativamente en la eficiencia operativa de las empresas y limita las oportunidades al nivel laboral [5]. El proyecto SEDAI busca subsanar esta deficiencia mediante la creación de un kit modular que permite a los estudiantes interactuar con sistemas de control reales, como PLCs y HMIs, que son fundamentales en la automatización de procesos industriales. Al implementar este tipo de entrenamientos prácticos, se facilita la comprensión y el manejo de tecnologías industriales, lo que contribuye directamente a mejorar la capacidad de los estudiantes para resolver problemas complejos [6]. Esta formación práctica no solo mejora la empleabilidad de los graduados, sino que también impulsa la competitividad de las empresas que integran a estos profesionales [7].

En este sentido, SEDAI no solo beneficia a los estudiantes y a la institución educativa, sino que también tiene un impacto positivo en la industria local y en la sociedad al cerrar diferencias al nivel de competencias técnicas. Además, el kit puede replicarse en otras instituciones educativas, creando un estándar de formación técnica adaptado a las demandas actuales del mercado laboral.

Objetivos

4.1. Objetivo General

Diseñar e implementar un sistema de entrenamiento SEDAI para un proceso de control de nivel, temperatura y flujo, con monitoreo remoto de estas variables a través de una conexión a Node- RED.

4.2. Objetivos específicos

4.2.1. Desarrollar una guía de uso y materiales educativos para estudiantes y docentes, que incluya prácticas y experimentos orientados al control del sistema.

4.2.2. Evaluar el sistema de entrenamiento, validando la funcionalidad del control para cada variable, garantizando un dispositivo que cumpla con los estándares técnicos requeridos.

4.2.3. Desarrollar un modelo físico del sistema de control de nivel, temperatura y flujo utilizando dispositivos industriales como sensores, actuadores y controladores que permitan una aproximación a procesos de automatización.

4.2.4. Configurar la conexión en Node-red para recepción y procesamiento de datos en tiempo real, visualizando y registrando variables como activación actuadores y lectura de los sensores.

5. Marco de referencia

A continuación se presenta información sobre los antecedentes del proyecto, es decir, la información correspondiente a diferentes proyectos realizados en los últimos años sobre plantas-sistemas de entrenamiento industrial así como enfoques e innovaciones hechos para brindar soluciones técnicas a problemáticas y desafíos en el área del control y la automatización. Como segunda parte se presentará el marco teórico del proyecto, donde se presenta toda la información base de todo el proyecto, incluyendo las características y descripción de los componentes que conforman la planta, protocolos de comunicación y herramientas de software que soportan el sistema. Finalmente, se tiene el marco legal donde se especifican las leyes y normas que rigen la implementación del proyecto.

5.1. Antecedentes

La capacitación basada en la simulación en entornos controlados ha sido fundamental y pilar en diferentes áreas del conocimiento. El tener la posibilidad de observar el comportamiento de diversos sistemas de simulados permite al usuario ganar experiencia y habilidades que son requeridas en el campo laboral. Esta práctica permite brindar al usuario un espacio seguro, donde la posibilidad de daños se minimiza considerablemente. En áreas como la salud estas aplicaciones son fundamentales ya que no solamente aseguran una curva de aprendizaje óptima para los practicantes si no también aporta seguridad a los pacientes [8]. La inclusión de la simulación ha no solo ha permitido crear una eficiente forma de capacitación, si no también ha permitido extender el conocimiento, integrando nuevos proyectos formando bases para la innovación [9]

En la industria son diversos procesos que se pueden simular, como lo son: procesos de flujo, temperatura, presión, nivel entre otros. Estos procesos pueden trabajar de forma individual o conjunta. La finalidad de entender el comportamiento natural de estos procesos brinda al ingeniero la posibilidad de adquirir experiencias prácticas que son altamente requeridas en la industria [3]. Varias empresas dedicadas a la automatización han creado proyectos que ofrecen a las instituciones del sector educativo o industrial como los son Amatrol recrean proceso con el fin de simularlos en entornos totalmente controlados. Basado en el estándar ISO 9001:2015 certifica a los clientes un producto confiable y seguro para las practicas realizadas [10].

En primer lugar el proyecto realizado por Diego Fernando Molina Ortega es un proyecto de implementación realizado en la universidad Santo Tomas. En el proyecto se destaca la problemática en la universidad en cuanto a la existencia de laboratorios y equipos dedicados a la automatización y control industrial. El proyecto de Diego se enfoca en crear un prototipo portátil dedicado a la simulación del control de procesos. La solución propuesta consta de un maletín con un PLC integrado con HMI y su correspondiente módulo de entradas y salidas. El objetivo de este proyecto es crear un dispositivo que se adapte a cualquier espacio en la universidad debido a su versatilidad y portabilidad. Su aplicación es amplia ya que no tiene un proceso fijo, su conexión se puede realizar a proceso de temperatura, flujo, nivel y presión. El proyecto también busca ser un prototipo que se puedan replicar en diferentes universidades de Tunja [11].

Por otra parte en argentina se evidencia el mismo problema, Nicolas A. y Faccioli Bruno proponen en su documento *"Los trabajos prácticos como materiales didácticos para aplicaciones tecnológicas en carreras de ingeniería"* un kit didáctico para el aprendizaje de procesos industriales. El proyecto es una mejora al anterior proyecto de Diego Fernando, ya que el prototipo consta además del maletín una planta para el control de velocidad de un motor AC. Su objetivo es abordar la importancia de los trabajos didácticos para el aprendizaje óptimo de las aplicaciones de ingeniería. Su característica principal es su portabilidad, teniendo en cuenta que como objetivo principal es que se pueda adaptar en cualquier entorno dentro de la institución. Para su construcción se seleccionó un maletín industrial junto con un tablero adecuado para las condiciones de movilidad humana. Este diseño proporciona la resistencia mecánica, aislamiento ambiental y movilidad requeridos [12].

En la Universidad Distrital Francisco José de Caldas se han propuesto proyectos robustos que pretenden formar al estudiante en la construcción de procesos, el control y el mantenimiento de equipos. La planta de procesos análogos (PPA) es un proyecto académico llevado por estudiantes de la carrera de ingeniera en control y automatización liderado por el docente Jorge Eduardo Porras Bohada. Este proyecto surge como una necesidad de ampliar el acceso a los estudiantes a este tipo de plantas de forma libre e innovadora de manera que, el estudiante pueda realizar cambios y mejoras que fomenten el desarrollo de nuevo conocimiento que sirva a los mismos estudiantes para seguir fortaleciendo el aprendizaje mediante la práctica. Es así como PPA surge como una solución integral.

La planta está diseñada para el control de procesos de flujo, nivel y temperatura, mediante el uso de instrumentos de medición analógicos y digitales. El sistema se gestiona a través de un PLC, una HMI y un variador de frecuencia de Rockwell, apoyados en un software de amplia aplicación en la industria colombiana. Con el tiempo, la planta ha incorporado nuevas tecnologías que han optimizado el control y la supervisión de las variables de proceso, ampliando sus capacidades técnicas. El objetivo de esta planta es desarrollar habilidades prácticas en los estudiantes, reforzando áreas como la programación, control y mantenimiento. Por lo que las planta siempre será supervisada por estudiantes quienes evalúan el estado de la planta y hacen correcciones de las guías de manipulación y del funcionamiento óptimo de cada uno de los procesos [13].

En la misma facultad se encuentra las plantas de procesos de AMATROL. estas plantas son: Planta de nivel, planta de control de pH, planta de temperatura y planta de presión. Estas plantas han servido para la introducción de procesos que normalmente se encuentran en la industria. Sus manuales permiten a los estudiantes la correcta manipulación de todos los componentes que la conforman. Sus múltiples conexiones permiten su control tanto de forma manual como de forma automática con un PLC. Lo cual brinda todo un abanico de posibilidades en cuanto control se refiere. Por último conocer cómo funcionan sus componentes de forma individual y conjunta permite a los estudiantes reforzar sus conocimientos en materias como son procesos, control e instrumentación, pilares fundamentales de la ingeniería en control y automatización [14].

A diferencia de la PPA, estas plantas no pueden modificarse, ya que sus procesos y componentes están específicamente definidos y no es permitido realizar cambios que alteren los instrumentos originales con los que fueron construidas. La planta ha sido diseñada con el

propósito de que el estudiante identifique claramente cada uno de sus componentes, mientras que al docente se le facilita la explicación de cada etapa del proceso. Además, la planta prioriza la creación de entornos seguros, dado que está dirigida principalmente a personas que inician en el campo de la automatización y el control.

Introducirse en el manejo de equipos de instrumentación y control es también afrontar los nuevos retos que la industria tiene gracias a una demanda que está en constante crecimiento. Las nuevas tecnologías emergentes son un gran aliado cuando se trata de mejorar la producción, la innovación y la eficiencia. El aumento de digitalización y la hiperconectividad permiten mejorar y potenciar tareas con procesos altamente automatizados y controlados. Es por ello que es importante para las universidades proyectar sus carreras tecnológicas hacia el IoT integrando herramientas que potencien el conocimiento y el desarrollo de nuevas formas de aprendizaje en los estudiantes. En un sector como la industria es importante que desde la formación se adquiera experiencia en soluciones que faciliten la comunicación y la adquisición de datos en entornos automatizados.

A nivel nacional ya existen proyectos que abarcan el área de la industria 4.0 como por ejemplo *"Prototipo de un equipo didáctico para laboratorio de control y automatización en industria 4.0"* llevado a cabo por la empresa INGECAEM para Unidad Central del Valle del Cauca Tuluá. El proyecto es una planta para la medición de temperatura, nivel, flujo, presión y peso. Este laboratorio -como ellos definen- tiene un módulo IoT que permite publicar los datos de todos los procesos que se estén ejecutando en tiempo real. Esto lo realiza mediante un Gateway con Node-Red donde mediante el PLC Siemens S7-1200 CPU1214 DC/DC/DC se envían los datos por ethernet y su publican en el dashboard. La comunicación es bidireccional por lo que el proyecto busca no solo realizar una visualización de datos si no también controlar las entradas del PLC con el protocolo de comunicación [15].

Actualmente las universidades apuntan a implementar *"Learning Factory"* como su nombre lo indica en ingles son fábricas de aprendizaje en los cuales se busca desarrollar competencias prácticas en los estudiantes mediante entornos que se asemejan a cadenas de valor real, creando espacios donde se aproxima lo más cercano a la industria pero sin dejar conceptos de lado como educativo, didáctico y pedagógico. Empresas a nivel global potencian estos entornos, ya que, ven este tipo de escenarios como centros de investigación donde se pueden resolver problemas reales de las fábricas y el desarrollo de innovaciones que apliquen en las industrias actuales [16].

Un ejemplo de Learning Factory la *Process Learning Factory CiP* de la *Technische Universität Darmstadt*, que se estableció en 2007, es un espacio de 500 m² creado para fomentar habilidades en producción ajustada (lean) e Industria 4.0. Actúa como una planta piloto que simula los procesos de una pequeña y mediana empresa, abarcando desde la recepción de materiales hasta el mecanizado, control de calidad, montaje y logística. Los estudiantes ponen en práctica métodos lean, combinando teoría y práctica a través de 15 módulos de aprendizaje que van desde los conceptos básicos hasta el pensamiento lean. Con el tiempo, el concepto se ha enriquecido con tecnologías de Industria 4.0, como la trazabilidad, el mantenimiento predictivo, la gestión digital, la asistencia al trabajador y los vehículos autónomos (AGV). Además, este modelo ha servido como base para implementar fábricas de aprendizaje en instituciones académicas e industriales alrededor del mundo [17].

Por otra parte *LPS Learning Factory*, se fundó en 2009 gracias a la Ruhr Universität Bochum, esta fábrica se especializa en producción ajustada, Industria 4.0 y eficiencia en el uso de recursos. Con un área de 1.800 m², aquí se producen tapones, portatapas y otros artículos a medida, utilizando una variedad de herramientas, transportadores, estaciones de montaje manual y robots industriales. Cada año, alrededor de 900 estudiantes tienen la oportunidad de realizar prácticas con productos reales que incluso se utilizan en la industria. Además, la fábrica promueve proyectos de investigación en áreas como modelos de madurez de Industria 4.0, sistemas de asistencia y aprendizaje, producción ciber física y robótica industrial. Desde 2018, forma parte del centro de competencia SME 4.0 Siegen [17].

A continuación se presentan en una tabla las diferentes fabricas ubicadas alrededor del mundo con su nombre, operador, país, producto y fuertes de aprendizaje:

Nombre	Operador	País	Producto	Temas principales
Fábrica de aprendizaje de procesos CiP	Universidad Técnica de Darmstadt	Alemania	Cilindro neumático	Producción ajustada e Industria 4.0
Fábrica de demostración DFA	RWTH Augsburgo	Alemania	Vehículos de movilidad eléctrica	Industria 4.0, prototipos e industrialización
Die Lernfabrik	TU Braunschweig	Alemania	Diverso	Sistemas sostenibles, CPPS, producción ajustada
El Centro de conocimiento	Universidad Friedrich-Alexander Erlangen Nürnberg	Alemania	Motores eléctricos	Tecnología de producción
Werk150	Escuela de Negocios ESB Hochschule Reutlingen	Alemania	Scooter urbano y accesorios	Diseño, implementación, sistemas de ensamblaje y logística
Fábrica de aprendizaje ESB Logistics Learning Factory	ESB Hochschule Reutlingen	Alemania	Scooter urbano y accesorios	Diseño, implementación, sistemas de ensamblaje y logística

ETA-Fábrica	PTW, TU Darmstadt	Alemania	Placa de control para bomba hidráulica, combinada con engranajes de eficiencia energética	Eficiencia energética, flexibilidad energética
Fábrica de aprendizaje de Festo Scharnhausen	Festo AG	Alemania	Válvulas neumáticas y terminales de válvulas	Formaciones orientadas al lugar de trabajo, Industria 4.0 y producción ajustada
iFactory	Universidad de Windsor	Canadá	Productos integrados	Productos integrados: aprendizaje interdisciplinario
IFA-Fábrica	Universität Erlangen Nürnberg	Alemania	Helicópteros y componentes	Planificación de producción, procesos ajustados
Lernfabrik de la Universidad del Ruhr	Universidad del Ruhr Bochum	Alemania	Máquina perforadora de precisión	Colaboración en el desarrollo de productos
Fábrica LEAD	Universidad Tecnológica de Graz	Austria	Patinete	Lean Management, Eficiencia energética, Operaciones ágiles, Digitalización
Fábrica LEAN-Fábrica	Fraunhofer IPK, TU Berlin, ITCL GmbH	Alemania	Tabletas farmacéuticas	Gestión ajustada
Fábrica de Aprendizaje e Innovación (LIF)	Universidad Técnica de Viena	Austria	Coche de slot	Planificación integrada de productos electrónicos, optimización de

Tabla 1. Learning factorys, adaptada de [18].

Por otra parte los simuladores como factory.io permiten tener toda una fábrica virtual en el computador. Se trata de un simulador el cual permite crear entornos industriales virtuales, pudiendo así crear procesos con equipos como bandas transportadoras, sensores, actuadores, grúas, dispositivos, entre otros. El potencial de este simulador es que se puede integrar con PLC's Siemens, Allen-Bradley, Omron y Codesys e incluso lenguajes de programación como C, Python, Labview y demás lenguajes que permiten un control de las entradas y salidas del entorno industrial. Su principal objetivo es ser un software que funciona como una planta didáctica donde se promueve la formación de automatización, control y desarrollo de prácticas de Industria 4.0 **y** gemelos digitales**.** Al ser un entorno totalmente seguro Factory.io funciona como una herramienta que permite explorar la creatividad del estudiante [19].

5.2. Marco teórico

En esta sección se explicarán conceptos y tecnologías en las que están soportadas el desarrollo del proyecto, necesarias para el entendimiento y compresión del prototipo. De esta forma se detallan conceptos como IoT, su definición, aplicaciones y proyecciones de esta tecnología. Por otra parte, se detalla el protocolo de comunicación ModBus RTU el cual es fundamental para la comunicación con el Middleware de la planta. Finalmente se abordarán conceptos de gran complejidad como son el control y la automatización enfocados a sistema SCADA

5.2.1. IoT

El internet de las cosas o IoT por sus siglas en inglés, se refiere a objetos físicos que están equipados con sensores, software y otras tecnologías que les permiten conectarse y compartir datos con distintos dispositivos y sistemas a través de Internet. Estos equipos que ahora se definen como dispositivos inteligentes pueden integrarse en todo tipo de espacios, permitiendo automatizaciones de procesos de manera rápida desde cualquier parte del mundo. EL IoT se puede integrar en ambientes cotidianos como refrigeradores y termostatos hasta maquinaria industrial La idea del Internet de las Cosas fue presentada por primera vez en 1999 por Kevin Ashton, un tecnólogo británico que trabajaba con Procter & Gamble. Ashton utilizó este término para mostrar cómo la tecnología de Identificación por Radiofrecuencia (RFID) podía mejorar la gestión de las cadenas de suministro [20].

Uno de los primeros ejemplos de IoT se remonta a los años 80 en la Universidad Carnegie Mellon, donde una máquina expendedora de Coca-Cola podía informar sobre su inventario a través de ARPANET, que es el precursor de Internet. Sin embargo, el IoT tal como lo conocemos hoy comenzó a tomar forma a finales de la década de 2000 y principios de 2010, gracias al crecimiento de la computación en la nube que hace posible el almacenamiento de grandes volúmenes de datos y al avance de las tecnologías de comunicación inalámbrica que facilita la interacción entre dispositivos con el usuario final, fundamental para conectar el mundo físico con el mundo digital [21].

Según el Banco interamericano de desarrollo en su *reporte de: internet de las cosas* los componentes claves que conforman los sistemas IoT son [22] :

- **La conectividad**

la conectividad se refiere a los métodos y tecnologías que permiten la comunicación entre dispositivos y sistemas. Esto implica la transmisión y recepción de datos a través de diferentes tipos de redes, que pueden ir desde las redes celulares tradicionales hasta las más recientes redes de baja potencia y amplia cobertura (LPWA), diseñadas para optimizar el alcance y la duración de la batería. La elección de la conectividad es un factor clave, pues influye directamente en el rendimiento de los datos, los costos operativos y la eficacia de la implementación.

- **Procesamiento de datos**

En los sistemas IoT, el procesamiento de datos transforma la información de los sensores en datos que el usuario puede identificar fácilmente. Este proceso lleva un acondicionamiento, fundamental para traducir esas señales eléctricas en valores ya sea de temperatura, presión, nivel e incluso de tiempo. Para esta etapa interna se precisa de circuitos eléctricos y aplicaciones matemáticas, por ejemplo, linealizaciones de valores o incluso por software. Las plataformas IoT hoy en día, usan algoritmos sofisticados, manejando enormes cantidades de datos, asegurando análisis rápidos y también muy precisos. Un ejemplo de ello es la **computación en el borde (edge computing)**, un enfoque distribuido que procesa la información directamente en los dispositivos cercanos a la fuente. De este modo, se reduce la latencia y se optimiza el uso del ancho de banda antes de enviar los datos a la nube.

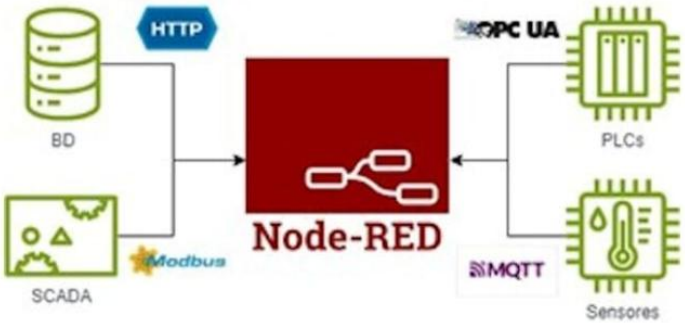


Figura 1. Conexión Node-Red [23]

- **Interfaz de usuario (IU)**

La IU permite la interacción entre el sistema y el usuario. Su objetivo es visualizar la información de clara concisa facilitando la interacción con los dispositivos IoT. Estas interfaces pueden estar diseñadas para dispositivos móviles, pantallas HMI, monitores y paneles de control. Por ejemplo, Unitronics mediante un aplicativo móvil se puede conectar al HMI del PLC y supervisar mediante protocolo ethernet. Es por ello que dependiendo de la aplicación y la complejidad de la implementación es una ventaja adaptar estas interfaces a

diferentes entornos de ejecución. El diseño de la interfaz de usuario (UI) en un sistema IoT debe contemplar aspectos como la usabilidad, la capacidad de respuesta y la accesibilidad, con el fin de asegurar una experiencia interactiva eficiente, intuitiva y satisfactoria para el usuario.



Figura 2. En la interfaz puede haber distintos tipos de graficas y visualizaciones de datos [24]

El Internet de las Cosas IoT ha revolucionado sectores como la salud, la industria, la agricultura y las ciudades inteligentes al habilitar la recolección y análisis de datos en tiempo real. En salud, los dispositivos portátiles permiten el monitoreo remoto de pacientes, mejorando la atención y la gestión de enfermedades crónicas; en la industria, los sensores facilitan el mantenimiento predictivo, optimizan la producción y fortalecen la cadena de suministro; en agricultura, drones y sensores impulsan la agricultura de precisión, maximizando recursos y reduciendo impactos ambientales; y en las ciudades, el IoT mejora la movilidad, la gestión de residuos y el consumo energético. Estos avances generan beneficios transversales como mayor eficiencia operativa, reducción de costos, personalización de servicios, mejor toma de decisiones y sostenibilidad, consolidando al IoT como un pilar clave en la innovación y el desarrollo tecnológico futuro [25].

Para pablo Lluch , líder de la Estrategia y el Desarrollo de Negocio de Smart Industries/IIoT para Orange Business en América:

> *"El Industrial IoT (IIoT) se está consolidando como uno de los habilitadores tecnológicos que mejor representa la convergencia entre los entornos IT (Information Technology) y OT (Operation Technology) en las plantas industriales. De esta forma y a nivel de tendencias cada vez vemos casos de uso más transversales y con mayor impacto en distintas líneas de negocio que tradicionalmente habían operado de forma aislada. Hablamos de tecnologías como 5G, Edge Computing, AI/ML o Digital Twins que, aplicadas a los centros de producción, habilitan nuevas funcionalidades que ayudan a las empresas a ganar competitividad estableciendo un entorno industrial más seguro y sostenible." Pablo Lluch*

En América latina hay que entender que el IoT no es una tecnología propia de países desarrollados si no como una tecnología que potencia el camino al desarrollo. Partiendo de esa idea según proyecciones de Orange Business se espera una transformación digital que alcance los 207.87 millones de dólares en 2032. Principalmente en empresas de Brasil, Argentina y Chile son empresas en las que se está trabajando para realizar implementaciones de Smart Industries y Operational Experience (OeX). El importante volumen de datos se traduce en una monetización, de manera que la información es un objetivo a la hora de ejecutar estos proyectos de gran envergadura [25]

5.2.2. Protocolo Modbus

Modbus es un protocolo de comunicación industrial originalmente desarrollado en 1979 por la empresa Modicon (hoy parte de Schneider Electric). Se utiliza para trasmitir información a través de redes entre dispositivos electrónicos. Estos dispositivos electrónicos no solamente se centran en la conexión de PLCs si no también variadores de frecuencia, medidores e incluso microcontroladores y computadoras. El protocolo ModBus está conformado por dos actores importantes, maestro ModBus y esclavo ModBus. En una red Modbus estándar se configura un único maestro y hasta 247 dispositivos esclavos, cada uno identificado mediante una dirección única en el rango de 1 a 247. Además de realizar consultas, el maestro tiene la capacidad de enviar y escribir información en los esclavos. Este sistema de comunicación industrial se implementa comúnmente sobre protocolos físicos como RS-232, RS-485 o RS-422 o Ethernet. Su sencillez y el hecho de ser un protocolo abierto, que los fabricantes pueden integrar sin coste de licencia, han favorecido su adopción como uno de los métodos más utilizados para la conexión de dispositivos electrónicos en entornos industriales [26] .

En la comunicación el dispositivo maestro controla el intercambio de datos implementando tareas de coordinación entre los diferentes esclavos que estén conectados. Mientras que el dispositivo esclavo conectado al bus de comunicación atiende a las peticiones del maestro proporcionado información o ejecutando tareas solicitadas por el maestro. Este intercambio de información puede ser el valor de sensores, valores de variables booleanas, mensajes, actualizaciones de usuario entre otros. La importancia de tener un registro único radica en que aunque el mensaje se está trasmitiendo por el mismo bus, si no corresponde a la dirección de envió el esclavo no lo escucha. Esta característica permite comunicarse con 247 dispositivos de manera individual mediante el id de registro único [26].

En la industria se maneja dos tipos de comunicación ModBus: Modbus RTU y ModBus TCP/IP. El ModBus RTU funciona sobre la comunicación serial (RS-232/RS-485) en donde los mensajes se trasmiten en formato binario compacto (RTU = *Remote Terminal Unit*). Su velocidad es limitada a la tasa del puerto serial, generalmente trasmitiendo a una tasa entre 1.2 kbps y 115 kbps. La aplicación de este protocolo es en sistemas industriales donde se conectan pocos dispositivos a una distancia no superior a 1200 m (si se maneja en RS 485). Por otra parte el protocolo Modbus TCP funciona sobre redes Ethernet (capa TCP/IP) en donde los mensajes se encapsulan dentro de tramas y la velocidad depende de la red ethernet, típicamente de 100 mbps o más. Puede comunicarse a través de redes locales (LAN) o incluso redes remotas (WAN/Internet) por lo que la hace ideal para integración de equipos industriales con sistemas SCADA servidores o aplicaciones en la nube [27]. .

A continuación se presenta la forma en que ambos protocolos envían los mensajes:

Mensaje ASCII					
Inicio	Dirección	Función	Datos	LRC	Final
:	2 Char	2 Char	0 a 252 Char	2 Char	CR, LF

Mensaje RTU			
Dirección	Función	Datos	CRC
1 Byte	1 Byte	0 a 252 Bytes	2 Byte

Figura 3. Forma de os paquetes de mensajes en Modbus RTU y ModBus ASCII tomado de [26].

Mensaje TCP					
ID transición	Protocolo	Longitud	Dirección	Función	Datos
2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	0 a 252 Bytes

Figura 4. Forma de los paquetes de mensajes en Modbus TCP [26].

5.2.3. MQTT

MQTT (Message Queuing Telemetry Transport) es un protocolo de comunicación ligero y abierto, diseñado para el intercambio eficiente de datos en aplicaciones de Internet de las Cosas (IoT) y sistemas con recursos limitados. Se basa en el modelo publicador/suscriptor, donde los dispositivos publican mensajes en un broker central, que a su vez los distribuye a los suscriptores interesados. Esta arquitectura desacopla a los emisores y receptores, lo que simplifica la comunicación y la hace más escalable. MQTT funciona sobre TCP/IP, garantiza sesiones confiables y soporta diferentes niveles de calidad de servicio (QoS 0, 1 y 2), que determinan si un mensaje se entrega una, al menos una o exactamente una vez. Su principal ventaja es la ligereza y eficiencia, lo que permite operar en redes con ancho de banda limitado o inestable, así como en dispositivos con baja capacidad de procesamiento o energía. Gracias a estas características, se ha convertido en un estándar (OASIS, ISO/IEC 20922) ampliamente usado en monitoreo industrial, sistemas SCADA, domótica, automatización de edificios y comunicaciones máquina a máquina [27] .

Este protocolo de comunicación permite una sesión persistente, entre el cliente y servidor, manteniendo las sesiones activas incluso si la red se desconecta. Cuando la red se restablece, la información necesaria para reconectar el cliente con el broker permanece disponible. Esta funcionalidad es una de las ventajas clave que hacen que el protocolo MQTT resulte más eficiente que HTTP en entornos con redes celulares inestables. Esta característica convierte a MQTT en un protocolo adecuado para comunicaciones prolongadas, lo que facilita su aplicación en procesos de operación continua o de larga duración, por ejemplo en controles de temperatura, monitorio de humedad e incluso aplicaciones de seguridad donde factores como el tiempo y la estabilidad de la comunicación son fundamentales para evitar perdida de información o errores de vulnerabilidad [28].

5.2.4. SCADA

SCADA (Supervisory Control And Data Acquisition, en español Supervisión, Control y Adquisición de Datos) es una arquitectura de sistema de control industrial diseñada para supervisar, controlar y optimizar procesos distribuidos, tanto de forma local como remota. Los sistemas SCADA permiten la adquisición de datos en tiempo real de dispositivos de campo, su procesamiento centralizado, visualización mediante pantallas, y la emisión de comandos para controlar los sensores/actuadores. Es muy usado en industrias como generación y distribución de energía, agua y aguas residuales, petróleo y gas, telecomunicaciones, transporte, manufactura, entre otras. Su capacidad de detectar fallos, generar alarmas, almacenar históricos y permitir operaciones remotas lo hace fundamental para mejorar la eficiencia, reducir tiempos muertos, aumentar la seguridad y facilitar la toma de decisiones [29]. .

Según [30] los sistemas SCADA están conformados por tres actores principales:

- **Elementos de hardware:** Los sistemas SCADA esta conformados por un PLC o un sistema de PLC que ejecutan la lógica de control y recolectan la información de los sensores y actuadores trasmitiendo estos datos mediante RTU, estos datos de presión, nivel, temperatura entre otros se pueden visualizar y controlar mediante un HMI

- **Elementos de Software:** La principal característica de un sistema SCADA es el control y la adquisición de los datos del proceso a tiempo real. Mediante el software se recopila procesa la información teniendo la posibilidad de ser gestionados a través de bases datos, con el fin de realizar el análisis de tendencias y generación reportes

- **Infraestructura de red:** Los sistemas requieren de redes de comunicación seguras y estables para lograr la trasmisión de datos. Los protocolos estándar para implementar SCADA son Modbus, DNP3 y TCP/IP, debido a que su gran uso extendido garantiza la interoperabilidad entre dispositivos que conforman el proceso.

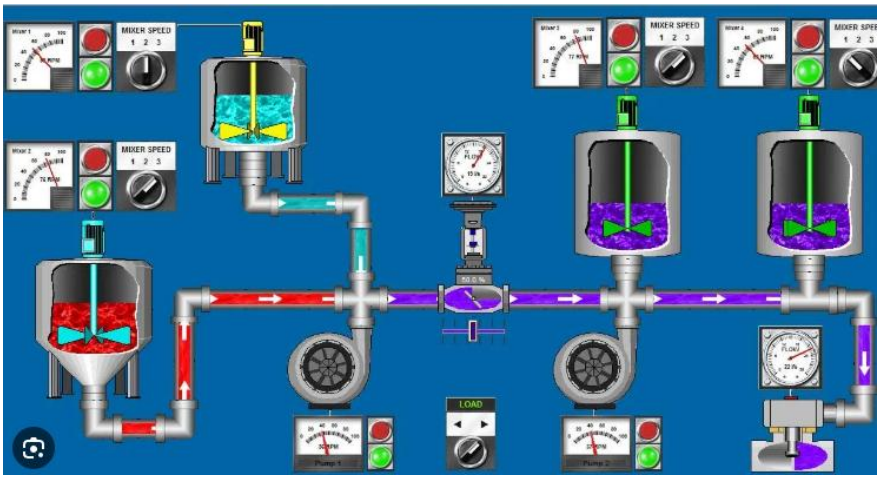


Figura 5. SCADA de circulación y control de temperatura de mezclas, tomado de [31]

5.2.5. Control PID

La implementación del control de retroalimentación PID (por sus siglas proporcional-integra-derivado) trajo grandes ventajas a nivel industrial puesto que incremento el número de procesos al que se le podía aplicar un monitoreo. Convirtiéndose en un estándar en los sistemas de control. El procedimiento utilizado para configurar los parámetros de ajuste de PID se conoce comúnmente como ajuste de bucle [32]. Cada parámetro del controlador PID tiene el propósito de regular la perturbación de entrada, actuar frente a la respuesta dinámica del sistema y es crítico para llevar el proceso al punto de ajuste (SP). Cada variable debe ajustarse de acuerdo con el proceso para que funcione, por lo que pueden tomar muchas combinaciones. Los parámetros de control de PID y su función en el controlador se describirán por separado a continuación.

5.2.5.1. Control proporcional

La acción proporcional tiene como objetivo enviar a la salida del controlador una señal de respuesta en proporción al error entre el setpoint (SP) y la variable de proceso (PV). Por lo que si se tiene un error muy grande de entrada la reacción de salida será grande y si el error es pequeño la reacción también lo será.

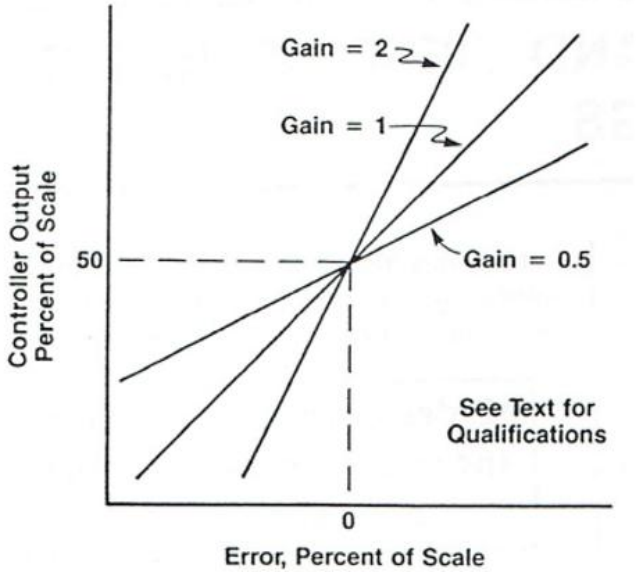


Figura 6. Efecto de la ganancia del control proporcional tomado de [33]

La Figura 3 muestra el error de respuesta de ganancia a la salida del controlador. Para una ganancia de 1, la salida cambia en la misma cantidad que la variable controlada (SP). Para el mismo cambio de error ganancias altas o bajas provocan cambios mayores o menores en la salida. Se denomina acción directa si la salida aumenta en proporción a la variable controlada. Si la salida disminuye a medida que la variable aumente se denomina acción inversa [33].

La pendiente de la rampa viene dada por el valor de Kp, cuanto mayor sea el valor, más inclinada será la rampa. Los grados de libertad del controlador puede ser observado en la pendiente y es conocido como la banda proporcional, cuanto mayor sea el valor de Kp, menor será la banda proporcional, lo que hace que hace que un proceso llega más rápido o más lento. Recuerde que realizar un control muy rápido podría llegar a ser perjudicial para los elementos del sistema

- *Ecuaciones de control*

La ecuación de control de un modo proporcional se describe de la siguiente forma.

$$M = K_c e + M_r$$

dónde

E = Error de control, %

$= SP - PV$ para un controlador de acción inversa

$= PV - SP$ para un controlador de acción directa

K_C = Ganancia del controlador, %/%

M = Salida del controlador o variable manipulada, %

M_r = salida del controlador (BIAS), %

PV = Variable de proceso, %

SP = Punto de ajuste, %

- *Compensación proporcional (offstet)*

El offset es la diferencia que existe entre el setpoint (SP) y la variable de proceso (PV). Para corregir este error se asigna un sesgo de salida (denominado BIAS) que disminuye esta diferencia y fija el controlador al SP. Para entender como el BIAS ayuda a corregir este problema supongamos un proceso controlado por un modo proporcional únicamente. El proceso consta del control de velocidad un automóvil siendo M la señal del acelerador que ajusta el flujo de combustible al motor y diferente de cero. Asignamos un SP de 65 km/h. Una vez dada la instrucción la velocidad comienza a aumentar hasta los 65 km/h. Como las dos variables PV y SP son exactamente iguales se tendría que:

$$M = K_c(0)$$

Luego de minimizar el error a cero la señal que estaría enviando el acelerador que ajusta el flujo seria cero, sin embargo, la salida del controlador es diferente de cero. La única forma de que la salida sea diferente de cero asignar un sesgo de salida (BIAS en algunas literaturas) [34]. Por lo que:

$$M = K_c(0) + M_r$$

En este ejemplo el valor de M_r corresponde el flujo de combustible que, en modo manual, hace que el automóvil viaje a la velocidad establecida de 65 km/h. A continuación, se mostrará el comportamiento de este valor (BIAS) en el proceso mediante un diagrama de bloques.

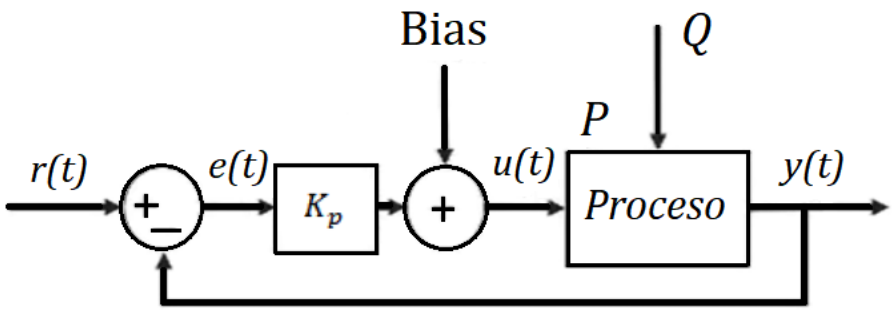


Figura 7. Ubicación del Bias en el esquema de control [Fuente propia]

Este parámetro es susceptible a cambios una vez establecido el proceso en automático. Debido a las características del proceso como variaciones el valor establecido de BIAS puede no ser preciso para el (SP) requerido

- *Ventajas*

1. Este tipo de controladores ayuda a reducir el error de estado estable. Lo que provoca una respuesta del sistema hacia un punto deseado (SP)
2. La respuesta lenta del sistema sobre amortiguado puede acelerarse con la ayuda de estos controladores.

- *Desventajas*

1. Debido a la presencia de estos controladores, se deben asignar compensaciones al sistema (offsets) lo cuales ayudan a fijar el sistema en el SP, pero son propensos a cambios debido a perturbaciones en el sistema.
   Los controladores proporcionales también aumentan el sobre impulso máximo del sistema. Lo cual puede ser perjudicial para los componentes que formen el sistema, por ejemplo, si se tiene una válvula mecánica que recibe la respuesta de abrir y cerrar de un controlador de una forma rápida puede causar un acortamiento del ciclo de vida e incluso un daño súbito en el componente.

5.2.5.2. *Control integral*

Para el anterior proceso de control de velocidad del automóvil asignemos la siguiente situación. Resulta que para conservar una velocidad de 65 km/h el BIAS tiene que mantener un valor para el cual el flujo de combustible al motor mantiene la velocidad una vez $SP = PV$. En un punto la carretera se empina por lo que el esfuerzo que debe aplicarse al motor debe ser mayor para mantener una velocidad constante. Para ello el controlador estará obligado a disminuir la variable de proceso (PV) para que la diferencia entre $SP - PV$ multiplicada por K_p más el BIAS actual pueda mantener la velocidad en esas condiciones constante.

El inconveniente con este procedimiento es el offset que se está generando nuevamente entre SP y PV. Provocando que nunca se pueda a llegar al setpoint de forma constante en la carretera empinada. Conociendo la naturaleza del controlador se podría parar el proceso, modificar el BIAS y de nuevo iniciar el proceso. Para cada variación de la carretera se tendría que modificar este valor manualmente aunado a ello se debe realizar los respectivos cálculos para no causar fallas en el sistema. A nivel practico este procedimiento resultara muy poco práctico puesto que a nivel industrial parar un proceso se traduce en pérdidas millonarias, además se estaría aumentando las probabilidades de error en cada paso del procedimiento.

El propósito principal del controlador integral K_i es eliminar la diferencia que puede haber entre la variable de proceso y el setpoint. Como se explicó anteriormente la función del BIAS en el proceso es eliminar el offset entre la variable del proceso y el setpoint. Cuando alguno de los dos valores cambia el dato de BIAS es ineficaz resultando en un nuevo offset que necesita ser ajustado para lograr el setpoint. El controlador integral proporciona elementos analíticos para establecer un valor de BIAS que se ajusta con respecto a las exigencias del sistema. En las primeras implementaciones a nivel industrial la función integral se denominaba “reset”. En algunas literaturas se sigue manteniendo el término [33].

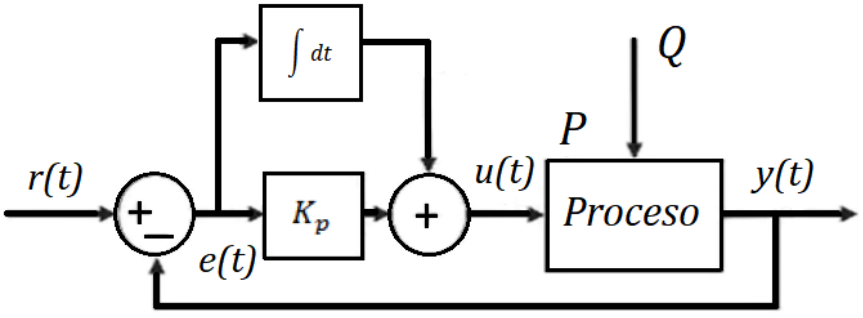


Figura 8. Incorporación del controlador integral al controlador proporcional [Fuente Propia]

El controlador integral recibe el error y ajusta el BIAS con respecto al setpoint y a la variable de proceso. Una vez establecido el setpoint con éxito las variables de entrada pueden cambiarse y el ajuste será automático.

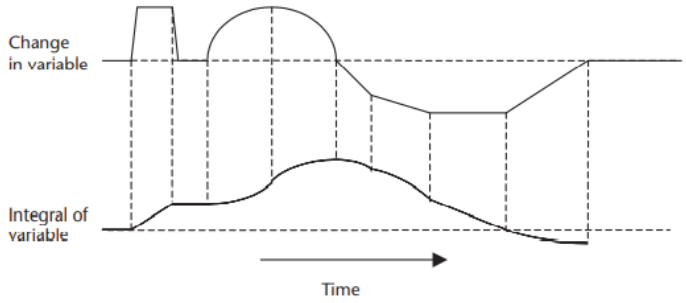


Figura 9. Cambio de variable integral en respuesta a cambio en una variable [35]

Si bien el modo integral es efectivo para eliminar la diferencia, es más lento que el modo proporcional porque tiene que funcionar durante un período de tiempo. Un modo más rápido que el proporcional es el modo diferencial

- *Ecuaciones de control*

La salida del controlador es proporcional a la integral del error de control, siendo la ecuación.

$$M = K_i \int_0^t E\, dt + M_0$$

Donde:

E = Error de control, %

$= SP - PV$ para controlador de acción inversa

$= PV - SP$ para controlador de acción directa

K_i = Ganancia integral, $(\%/\%)/min$

M = Salida del controlador, %

M_0 = Salida inicial del controlador, %

- *Ventajas*

1. Los controladores integrales tienen la capacidad de devolver la variable controlada al SP después de una variación en el sistema. De ahí su nombre de controladores de reinicio (reset).

- *Desventajas*

1. a hacer que el sistema sea inestable porque responde lentamente hacia el error producido. Por lo que para algunos procesos tiene que tener ajuste preciso para no causar errores en la salida

5.2.5.3. Control derivativo

El propósito del control derivativo es predecir el movimiento de una variable de proceso tomando una acción proporcional a su tasa de cambio. la respuesta adelantada del modo derivativo aumenta la estabilidad [36]. En resumen, la derivada es la expresión de la tasa de cambio de una variable con respecto a otra variable. La aplicación de este controlador acelera la respuesta de salida del sistema. Mediante aproximaciones se puede tener una visión del estado en el que se comportara el controlador. La respuesta nunca será perfecta aun así resulta practico para la mayoría de resultados cuando se usa la función derivada [33].
El aplicar una acción derivativa grande no significara la predicción total del estado del sistema, puesto que puede causar una diferencia grande entre la señal de error y la aproximación [32].

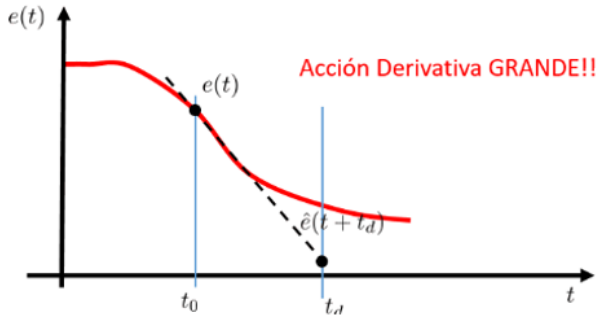


Figura 10. Efecto de una acción derivativa grande [Fuente Propia]

El adaptar una acción derivativa muy pequeña tampoco es factible para el proceso, puesto que el aporte que estaría haciendo al controlador la predicción del comportamiento del sistema seria nulo. En procesos demasiado rápidos, por ejemplo, flujo y presión los tiempos integrales no deben ser menores a segundos.

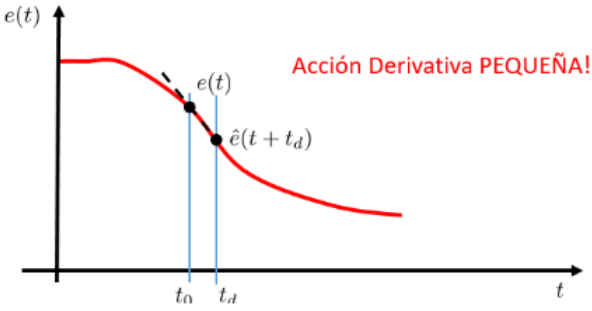


Figura 11. Efecto de una acción derivativa pequeña [Fuente Propia]

Este parámetro de ajuste del controlador es muy difícil de calibrar. Es por ello que la mayoría de controladores pasan de este modo a no ser que el proceso exija un controlador derivativo estrictamente [37].

- *Ecuaciones de control*

El controlador derivativo no se puede utilizar solo. Debe usarse mínimo con el controlador proporcional. Estos controladores PD son muy raros, por lo que al introducir el modo derivativo es casi siempre para complementar un control proporcional-integral (PI) resultando en un controlador PID [33].
Para las ecuaciones se trabajará la forma más simple. Por ello se explicará las ecuaciones en el contexto de un controlador PD.

$$M = K_c \left[E + T_d \frac{dE}{dt} \right] + M_r$$

- Donde:
- E = Error de control, %
- $= SP - PV$ para controlador de acción inversa
- $= PV - SP$ para controlador de acción directa
- K_c = Ganancia del controlador, (%/%)
- T_d = Tiempo derivativo, min
- M = Salida del controlador, %
- M_r = salida del controlador (BIAS), %

- *Ventajas*
    1. La principal ventaja de un controlador derivado es que mejora la respuesta hacia el punto de ajuste del sistema. Incrementando la rapidez y la estabilidad del controlador ante perturbaciones [35].
- *Desventajas*

    1. Mal sintonizada puede generar cambios muy bruscos en el proceso produciendo oscilaciones en el sistema. Por lo que introducir acciones derivativas muy grandes es perjudicial para la estabilización e introducir acciones derivativas pequeñas es insignificante para el control del proceso [33] .
    2. Puede amplificar las señales de ruido del proceso.
    3. No mejora el error de estado estacionario.

- *Ecuaciones de control*

El controlador derivativo no se puede utilizar solo. Debe usarse mínimo con el controlador proporcional. Estos controladores PD son muy raros, por lo que al introducir el modo derivativo es casi siempre para complementar un control proporcional-integral (PI) resultando en un controlador PID.
Para las ecuaciones se trabajará la forma más simple. Por ello se explicará las ecuaciones en el contexto de un controlador PD.

$$M = K_c \left[E + T_d \frac{dE}{dt} \right] + M_r$$

Donde:

E = Error de control, %
  $= SP - PV$ para controlador de acción inversa
  $= PV - SP$ para controlador de acción directa
K_c = Ganancia del controlador, (%/%)
T_d = Tiempo derivativo, min
M = Salida del controlador, %
M_r = salida del controlador (BIAS), %

5.3. Marco Legal

5.3.1. ley 842 de 2003

El código ético del ingeniero el cual es capaz de aplicar materias de ciencias físicas, químicas y matemáticas, deba regirse como lo estipula el ARTÍCULO 2: EJERCICIO DE LA INGENIERÍA el desempeño del ingeniero bajó el perfil inventivo en el contexto de proyectos mecánicos, eléctricos, electrónicos, de computación contiene un régimen del diseño funcional ante el fenómeno estudiado, así mismo el nivel de estudio profesional en la ejecución de proyectos [38].

5.3.2. DECRETO 585 1991

El consejo de ciencia y tecnología de nuestra república está formado por miembros de diferentes organizaciones encargados de supervisar el proceso tecnológico según el ARTÍCULO 4. El sistema no excluye los proyectos que se presenten, sin importar quién los presente o de qué institución educativa provengan. Esto es crucial para los estudiantes de electrónica, ya que les permite participar en la estructuración de un proyecto. El encargado de dirigirlo debe tener las capacidades descritas en el ARTÍCULO 11. Las normas de este decreto son reemplazadas por la ley 1286 de 2009 [38].

5.3.3. Ley 1286 de 2009

La ley constitucional promueve el desarrollo nacional de la ciencia, tecnología e innovación a través de Colciencias, con el objetivo de fortalecer la economía e industria del país. Esto incluye la participación del ingeniero electrónico en la implementación de nuevas estrategias, como la maquinaria que beneficia el mercado nacional. Aportando de forma significativa al desarrollo de diferentes áreas que promueven la investigación, la sana competencia y el mercado laboral [38].

5.3.4. La Resolución 2400 de 1979

La resolución reglamenta las condiciones de seguridad e higiene en los lugares de Colombia. Los empleadores tienen la responsabilidad de cumplir con las normativas de salud, higiene y seguridad laboral, estableciendo un servicio médico permanente en lugares con mayor riesgo de accidentes o enfermedades. Es fundamental recolectar los desperdicios en recipientes tapados y evitar su acumulación para proteger la salud de los trabajadores. Las instalaciones deben contar con paredes lisas, pintadas en tonos claros y en buen estado, así como mantener un espacio libre de 1,50 metros alrededor de equipos generadores de energía térmica, como hogares, hornos o calderas. Además, el suministro de agua, la gestión de aguas industriales y la disposición adecuada de residuos deben garantizar condiciones saludables para los empleados [39].

5.3.5. Norma RETIE (Reglamento Técnico de Instalaciones Eléctricas):

El Reglamento Técnico de Instalaciones Eléctricas (RETIE) es una normativa colombiana que establece los requisitos técnicos y de seguridad para las instalaciones eléctricas en el país. Su propósito principal es proteger la vida y la salud de las personas, así como preservar el medio ambiente y garantizar la seguridad de los bienes. El RETIE aplica tanto a instalaciones nuevas como a modificaciones de sistemas eléctricos existentes, incluyendo áreas residenciales, comerciales, e industriales Este reglamento detalla especificaciones sobre el diseño, construcción, operación y mantenimiento de las instalaciones, con énfasis en el cumplimiento de estándares internacionales. También contempla la certificación de productos eléctricos y la capacitación de los profesionales encargados de las instalaciones, buscando prevenir riesgos eléctricos, como choques eléctricos, incendios y explosiones [40].

5.3.6. Ley 1341 de 2009

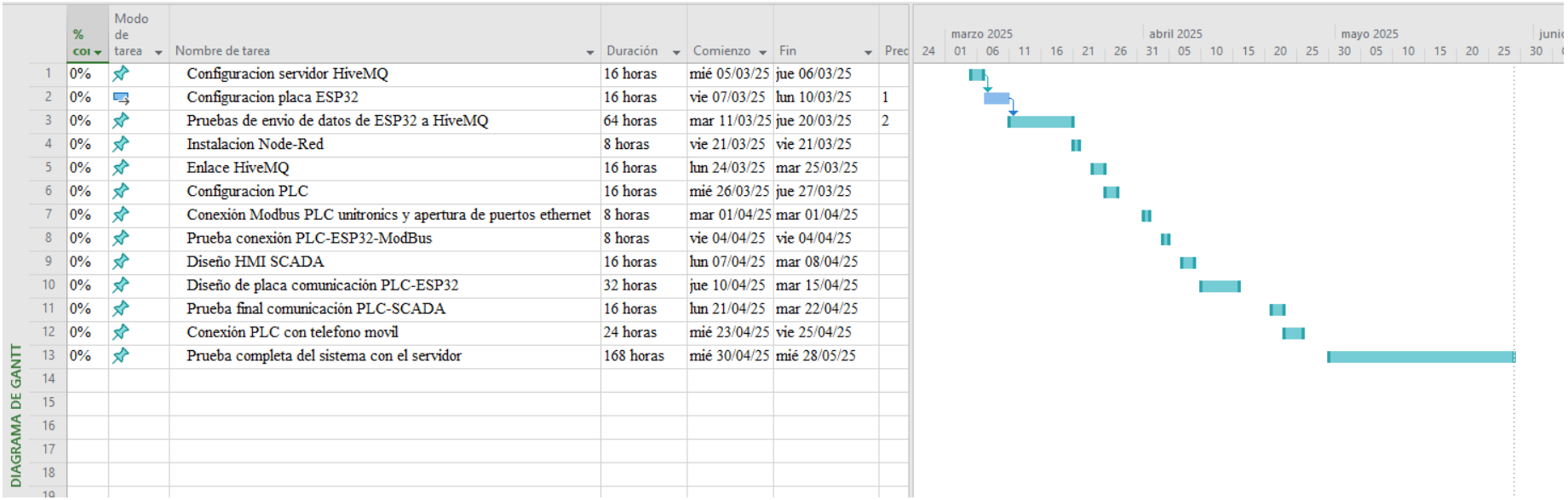
la Ley de Tecnologías de la Información y las Comunicaciones (TIC), es una normativa colombiana que regula el sector de las TIC en el país. Su objetivo principal es promover el acceso y el uso equitativo de las tecnologías de la información y las comunicaciones para avanzar en el desarrollo económico, social y cultural en Colombia. Esta ley establece los principios rectores del sector, como la masificación de las TIC, la promoción de la inversión, el acceso a internet, la libre competencia y la protección de los derechos de los usuarios. Define el marco institucional para la regulación, supervisión y control de las actividades relacionadas con las TIC, asignando responsabilidades al Ministerio de Tecnologías de la Información y las Comunicaciones (MinTIC). [38].

5.3.7. DECRETO 3573 DE 2011:

Dictaminan las licencias ambientales aprobadas por un organismo técnico que estudia el debido con autonomía administrativa y financiera para la eficacia de la gestión ambiental y desarrollo sostenible. ARTÍCULO 2: la autoridad nacional de licencias ambientales se encarga de que proyectos, obras o actividades estén sujetas a la licencia, que permitan cumplir la normativa y contribuir al desarrollo sostenible. ARTÍCULO 3: (expresado en ítems: 7,9,11) permite vigilar, controlar y suspender el desarrollo de cualquier proyecto que atente contra el medio ambiente bajo orden del de ministro de ambiente, además de adelantar los procesos de investigación, así mismo proteger proyectos de su competencia, si estos no poseen la licencia para el desarrollo del planteamiento [38].

6. Actividades realizadas durante la pasantía

6.1. Cronograma de actividades Nicolas Sopo Bobadilla



	% col	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Prec
1	0%		Configuracion servidor HiveMQ	16 horas	mié 05/03/25	jue 06/03/25	
2	0%		Configuracion placa ESP32	16 horas	vie 07/03/25	lun 10/03/25	1
3	0%		Pruebas de envio de datos de ESP32 a HiveMQ	64 horas	mar 11/03/25	jue 20/03/25	2
4	0%		Instalacion Node-Red	8 horas	vie 21/03/25	vie 21/03/25	
5	0%		Enlace HiveMQ	16 horas	lun 24/03/25	mar 25/03/25	
6	0%		Configuracion PLC	16 horas	mié 26/03/25	jue 27/03/25	
7	0%		Conexión Modbus PLC unitronics y apertura de puertos ethernet	8 horas	mar 01/04/25	mar 01/04/25	
8	0%		Prueba conexión PLC-ESP32-ModBus	8 horas	vie 04/04/25	vie 04/04/25	
9	0%		Diseño HMI SCADA	16 horas	lun 07/04/25	mar 08/04/25	
10	0%		Diseño de placa comunicación PLC-ESP32	32 horas	jue 10/04/25	mar 15/04/25	
11	0%		Prueba final comunicación PLC-SCADA	16 horas	lun 21/04/25	mar 22/04/25	
12	0%		Conexión PLC con telefono movil	24 horas	mié 23/04/25	vie 25/04/25	
13	0%		Prueba completa del sistema con el servidor	168 horas	mié 30/04/25	mié 28/05/25	

Figura 12. Diagrama de Gantt Nicolas Sopo Bobadilla

A continuación se hará una descripción de las actividades designadas por Nicolas Sopo Bobadilla. Para este plan de trabajo se centró en la comunicación del middleware y en el desarrollo de las prácticas de control de nivel y temperatura así como su documentación. El principal reto que se presento fue establecer una comunicación PLC-ESP32-Node Red de manera estable y continua por lo que la mayor parte de investigación fue dedicada a realizar la configuración de los dispositivos para realizar él envió de la información a tiempo real. Las practicas se realizan con el fin de probar el sistema y diseñar las guías que servirán como apoyo para los estudiantes.

6.1.1. Configuración servidor HiveMQ

Como bróker MQTT se eligió HiveMQ, ya que ofrece una implementación profesional ampliamente utilizada en entornos de IoT y automatización industrial, tanto en la nube como de manera local. Además, se trata de una plataforma de fácil acceso que garantiza una comunicación segura mediante TLS/SSL. Para iniciar con HiveMQ es importante crear una cuenta, esta es gratuita y en ella se podrá manejar los clusters, que corresponde a un conjunto de brókers MQTT. Este servidor proporcionara información para realizar la conexión a la ESP32 mediante la librería *PubSubClient.* Además de ello permitirá crear las credenciales y definir parámetros como suscripción y publicación de datos, con estas especificaciones se puede establecer si la comunicación es unidireccional o bidireccional

A continuación se presentará la configuración del cluster:

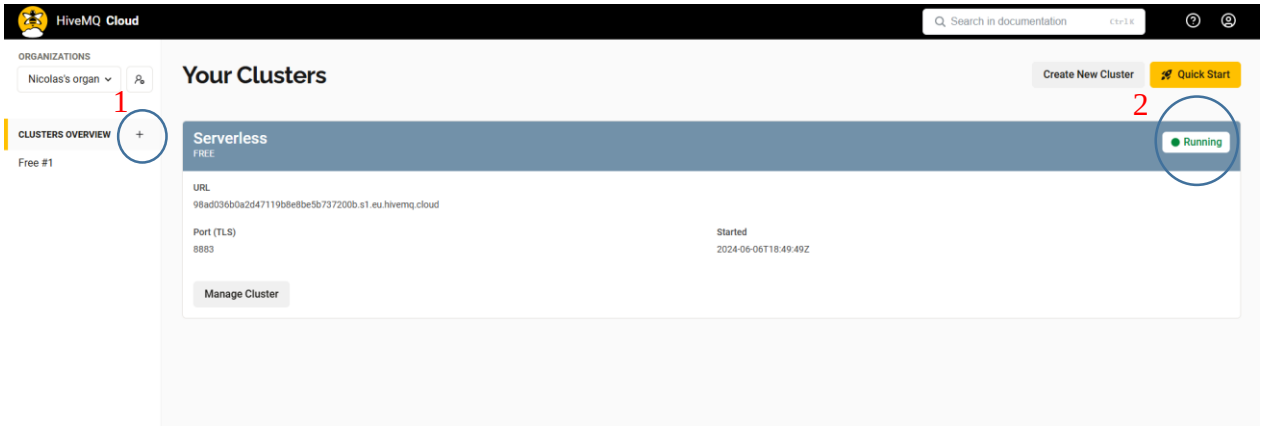


Figura 13. Creación del cluster [Fuente propia]

Cuando se crea una cuenta este será el panel principal que se presentara, aquí se podrá agregar los clusters necesarios para realizar el envío de datos MQTT organizando los clusters por servidores.

1. Cuando se pulse el botón de más (+) se podrán elegir dos planes para crear el cluster. A continuación se presentarán los planes

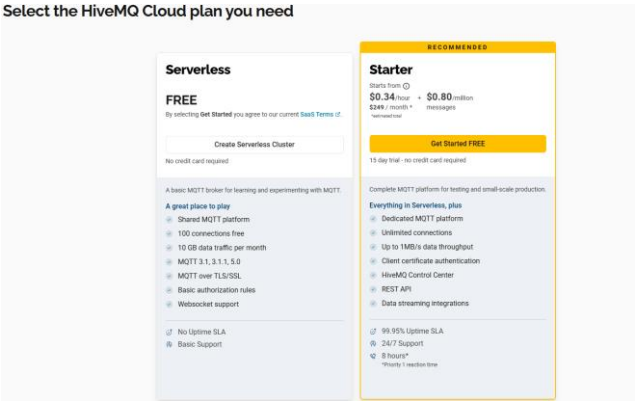


Figura 14. Los planes del HiveMQ pueden cambiar dependiendo de la aplicación y el consumo

En el plan gratuito cuenta con las características necesarias para la implementación del proyecto, debido a que la cantidad de datos enviados no superan las 10GB mensuales ya que el sistema estará pensado para ser un banco de prácticas y no una planta de funcionamiento permanente.

2. Una vez creado el cluster se podrá el estado del servidor, además que proporcionará el URL para la conexión y el número de puerto TLS.

Una vez abierto el cluster se presentará el siguiente panel:

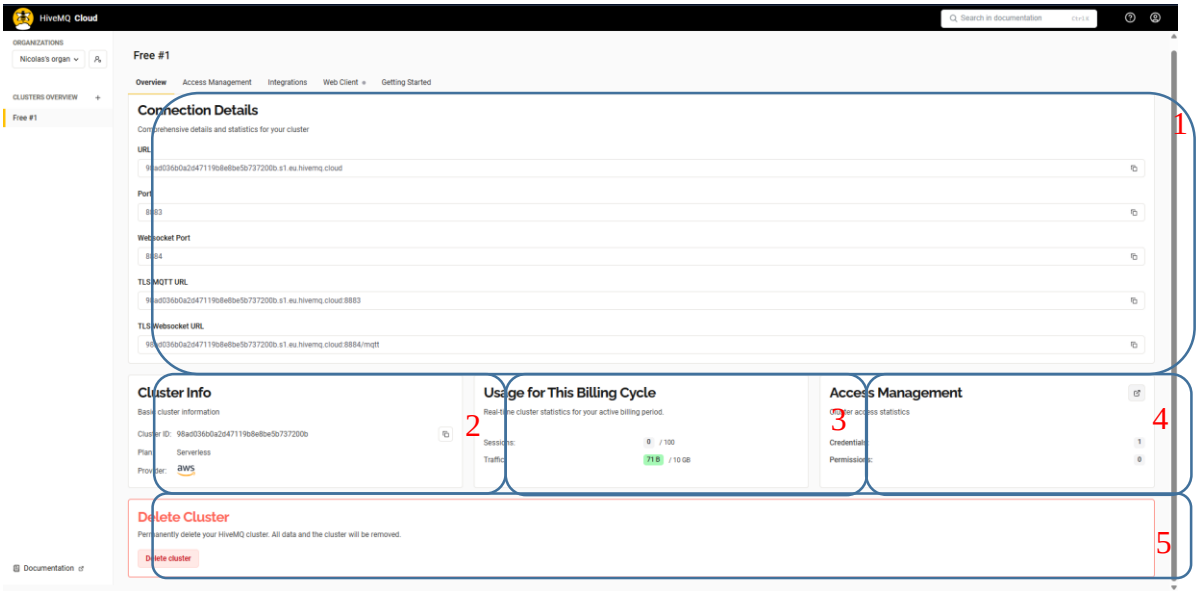


Figura 15. Panel del cluster [Fuente propia]

Las ventanas presentadas son:

1. **Detalles de conexión:** En esta ventana se presentarán los datos de:

- URL Servidor
- Puerto
- Puerto websocket
- URL MQTT TLS
- URL websocket TLS

2. **La información del cluster**: Información como el id, el plan y la nube que está soportando el servidor (en este caso AWS)

3. **Uso de ciclo de facturación (mes):** en este apartado se podrá visualizar el número de sesiones que se están conectando al cluster, para el plan gratuito será de máximo 100 sesiones al mismo tiempo. Cada sesión será un dispositivo, usuario o enlace que envíe y reciba datos del cluster. Además la cantidad de datos enviado, para el plan gratuito será de 10GB

4. **Gestión de accesos:** En este apartado se visualiza el número de credenciales y permisos. Los permisos los define el usuario en el cluster y denotan si un acceso puede publicar y suscribir datos

5. **Borrar cluster:** Es este recuadró estará la opción de eliminar el cluster. Con esta acción se podrá eliminar toda la información del cluster, accesos y credenciales. La información no se podrá recuperar

Para el proyecto es importante al menos crear una credencial, para ello en el panel central se accede al apartado *Access management*. En esta sección se gestionarán los datos de autenticación y autorización. Tendremos la opción de credencial (1), el certificado de cliente (2) y el token JSON web (3) siendo estas dos últimas opciones de pago. Por último está el recuadro de crear permisos y restricciones a los accesos del cluster (4).

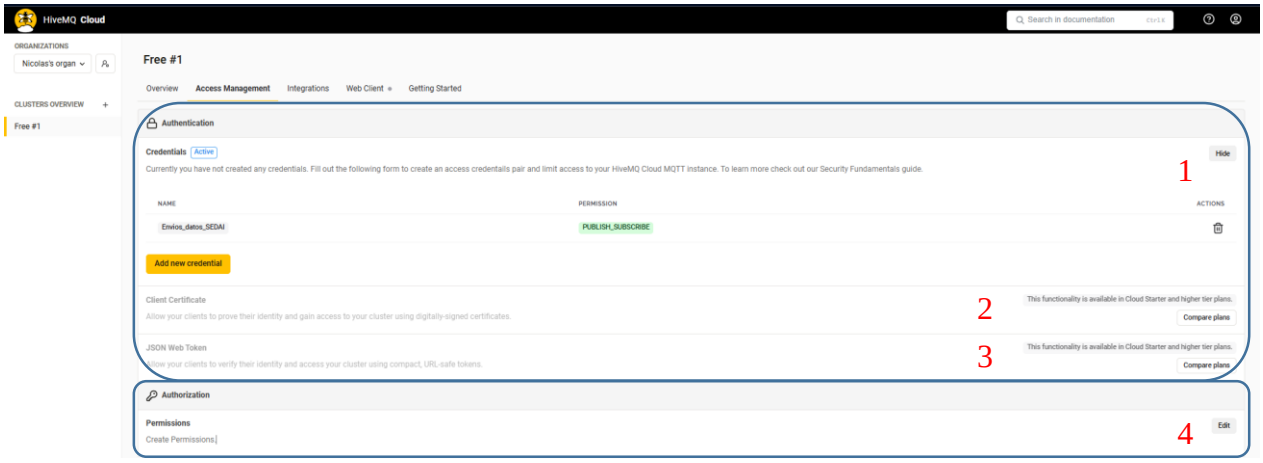


Figura 16. Gestión de accesos del cluster [Fuente propia]

Para crear una credencial se pulsa el botón *"add new credential"* el cual mostrara el siguiente formulario:

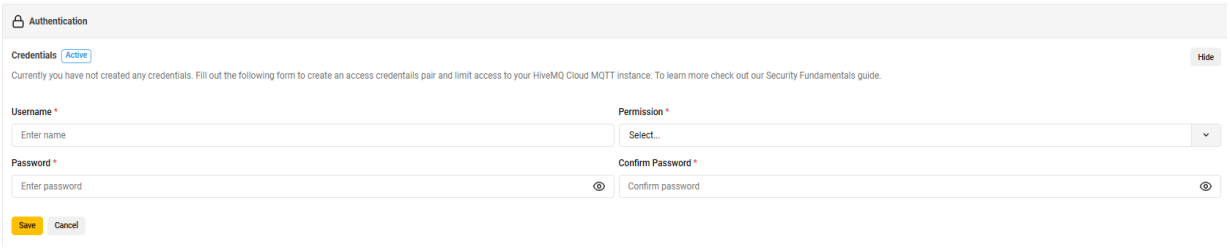


Figura 17. Creación de usuario y contraseña para el acceso [Fuente propia]

Para el proyecto se configura una credencial que se llama "Envios_datos_SEDAI" con los permisos para publicar y suscribir. Se define asimismo una contraseña que será importante para acceder a los datos con Node-Red. El acceso a la credencial creada se realizará desde el apartado *"Web Client"*. En esta ventana se encontrará la siguiente información:

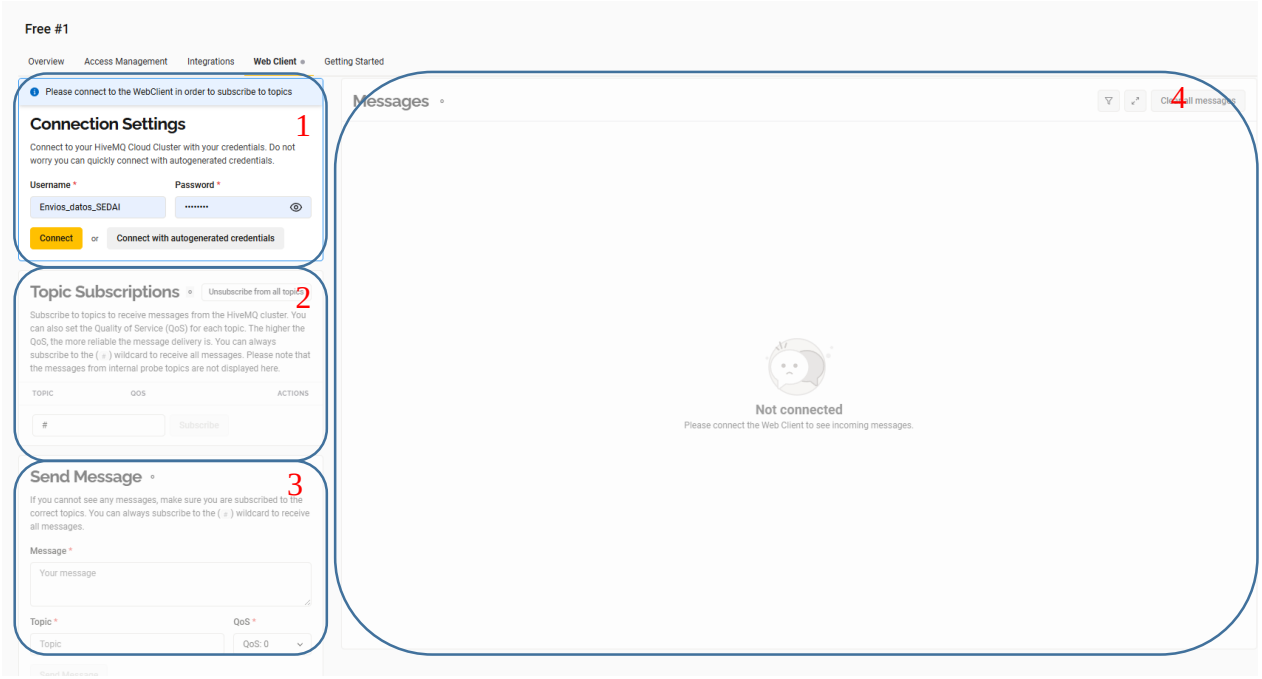


Figura 18. En la conexión web se pueden ver mensajes de los tópicos declarados o de todos los tópicos recibidos [Fuente propia]

1. **Configuración de conexión:** Este apartado se ingresará los datos de la credencial para acceder a ella. También HiveMQ tendrá una opción de generar una credencial autogenerada

2. **Suscripción tópicos:** Esta sección se habilita una vez se ingrese los datos de la credencial. Este apartado se definen los tópicos a los cuales HiveMQ se va a enlazar para enviar datos por ese canal, es un identificador y por este medio es donde por ejemplo, se podrá enviar valores análogos y digitales de los sensores, además de ello,

contará con la opción de definir el nivel de garantía en la entrega de los mensajes entre el publicador y el suscriptor (QoS:0,1,3)

3. **Envío mensaje:** Esta sección se habilita una vez se ingrese los datos de la credencial. Esta opción brinda la posibilidad de enviar mensajes a un tópico en específico. Este apartado también será posible definir el nivel de garantía

4. **Mensajes:** Esta sección se habilita una vez se ingrese los datos de la credencial. Este recuadro imprime los mensajes que están siendo enviados y recibidos de los tópicos que estén suscritos en la credencial. Se pueden filtrar los mensajes y borrar la consola. Esta opción será útil para implementar pruebas de recepción de datos del sistema.

6.1.2. Configuración placa ESP32

Para configurar la ESP32 se utilizó el IDE de Arduino, se eligió este software debido a su gran uso extendido en proyectos IoT, este software además brinda las librerías para poder utilizar el wifi de la placa de desarrollo, fundamental para la comunicación MQTT con Node red y el protocolo ModBus para la comunicación con PLC. Las librerías que se utilizaron son:

- ***ModBus master:*** Esta librería se utilizó para definir la ESP32 como maestro Modbus y realizar el pedido de los valores de los sensores y actuadores al PLC, actuando este como esclavo

- ***Wifi:*** Para enviar los datos por protocolo MQTT es necesario activar el wifi de la placa de desarrollo, sin la conexión a ethernet la conexión con el SCADA no sería posible.

- ***PubSubClient:*** Se utiliza para la conexión del bróker MQTT y realizar él envió de datos a HiveMQ

- ***WiFiClientSecure:*** es una clase de la librería de Arduino/ESP32/ESP8266 que se utiliza para establecer conexiones seguras a través de TLS/SSL (la misma tecnología que usan las páginas con https://) sirve para que el microcontrolador pueda conectarse a servidores de manera segura

6.1.3. Pruebas de envió de datos de ESP32 a HiveMQ

Para llevar a cabo la prueba, se desarrolla un código de verificación, este código implementara una conexión MQTT entre la ESP32 y HiveMQ. Con esta prueba se realizará él envió de un valor entero para ser publicado en la consola del cluster. En la siguiente sección del código se realiza un contador simple que incrementa de uno en uno y se publica en el servidor HiveMQ, así mismo el mensaje se publica en la consola de Arduino. El código completo se encontrará en la sección de anexos

```
  }
}//Conexion MQTT

void setup() {
  Serial.begin(9600);
  Serial1.begin(9600, SERIAL_8N1, 16, 17);  // UART2 en GPIO16 y 17
  Serial1.flush();
  setup_wifi();

  espClient.setCACert(root_ca);
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);

  Serial.println("Inicialización completa. Enviando datos...");
  delay(5000);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

    counter++;

    client.publish(outTopicTemp1, String(counter).c_str(), true);
    Serial.println(counter);

    delay(1000); // Publica cada 1 segundo

  }
```

Figura 19. Función Setup y función Loop [Fuente propia]

Una vez ejecutado el código en la placa se comprueba en HiveMQ la correcta recepción de los datos, para ello se pulsa en la opción suscribirse a todos los mensajes y los mensajes deberían aparecer en la consola

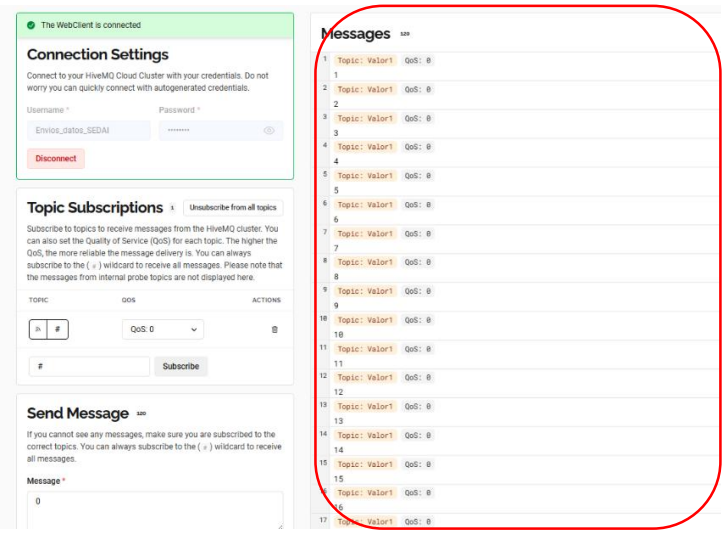


Figura 20. Mensaje recibido del tópico "Valor1" [Fuente propia]

6.1.4. Instalación Node-Red

Node-RED es una herramienta basada en flujo utilizada para la programación visual en sistemas de automatización y control. Permite conectar hardware, API y servicios en línea,

facilitando la integración y la creación de soluciones IoT. Principalmente Node RED se desarrolló para comunicar hardware por lo que se ha convertido en un estándar para procesar datos a tiempo real. La programación visual facilita y reduce los procesos de conexión. Debido a su alta conectividad entre equipos es perfecto para trabajos a nivel industrial y sirve como solución para equipos que necesiten intercomunicarse entre sí, como sensores, plantas, equipos y servidores. Su enfoque se basa en el IoT basándose en protocolo de comunicación como son MQTT. Para SEDA Node-RED se utilizará para monitorear los sensores de flujo, nivel y temperaturas. Una vez procesada la información por el microcontrolador ESP32.

Para iniciar con la integración de Node-Red el primer paso es realizar la instalación en el equipo. Para ello desde la página oficial de Node-Red en el link: Node.js — Descarga Node.js®.



Figura 21. Versión de descarga de Node.Js para SEDA V22.109.0 LTS [Fuente propia]

Para este proyecto se descargará la versión LTS (*long term support*). Esta versión brindara un soporte a largo plazo para corrección de error y de seguridad que se presenten. En cambio la versión current tiene las ultimas características nuevas, lo cual lo hace ideal para experimentar y probar funciones recientes. Para seguridad y robustez del proyecto es necesario utilizar una versión estable. Según Node js el periodo de las actualizaciones es de 30 meses. Para instalarlo seguimos los pasos que nos da el asistente:

Ahora para instalar el entorno de Node-Red se hará en npm con el siguiente enlace : node-red - npm. Posteriormente sigue el instructivo que se presenta a continuación.

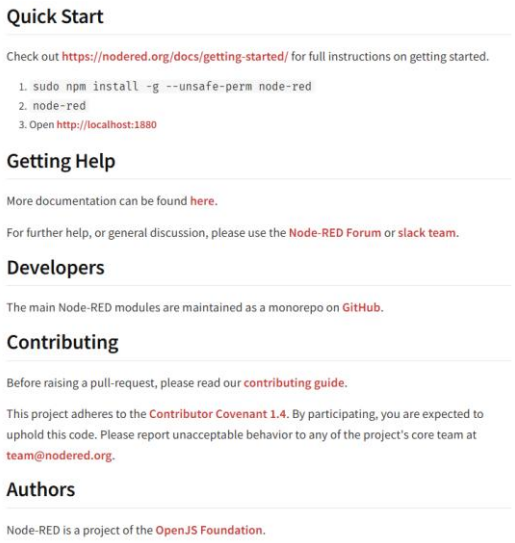


Figura 22. Instalación Node-Red [Fuente propia]

Una vez instalado Node-Red este se abrirá desde la consola de comandos de Windows , como se observa en la siguiente figura:

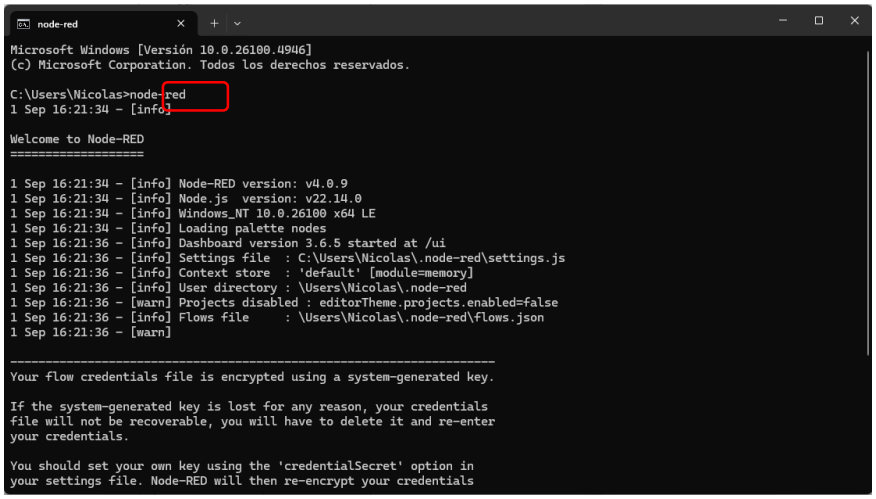


Figura 23. Ejecución de Node red desde la consola de comandos (CMD) [Fuente propia]

Después de ejecutar el comando, en el navegador deberá ingresarse la siguiente dirección en la barra de búsqueda: http://localhost:1880/. Con esto ya se tendrá el entorno de programación para la recepción de datos MQTT.

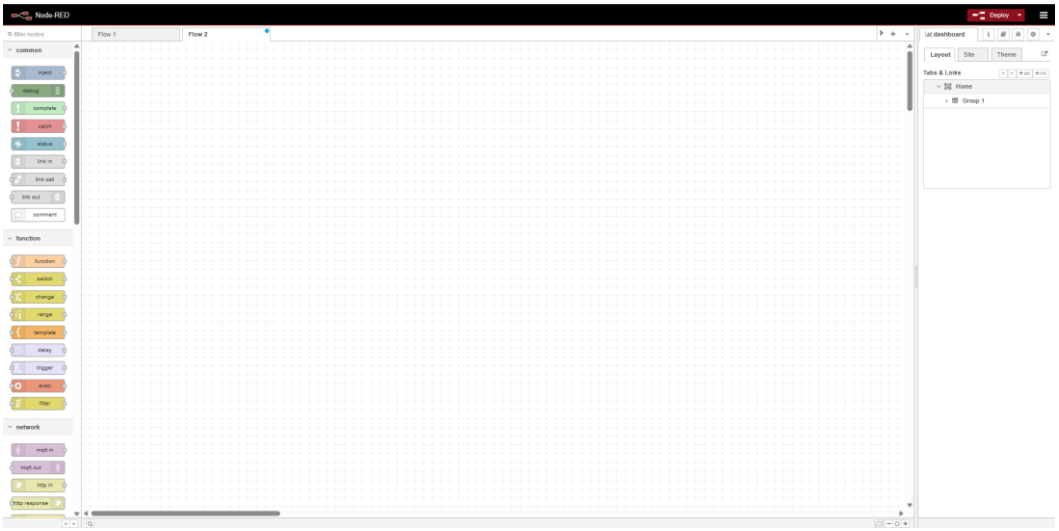


Figura 24. Zona de trabajo Node-red [Fuente propia]

6.1.5. Enlace HiveMQ – Node-Red

Para establecer la conexión entre HiveMQ y Node-Red, se requiere la información generada a partir del clúster previamente configurado en HiveMQ. En el entorno de Node-Red, la recepción de datos mediante el protocolo MQTT se realiza mediante la configuración de los nodos correspondientes, los cuales gestionan la información transmitida por el servidor. Dichos nodos se encuentran en la sección *"network"*, siendo el nodo *"mqtt in"* el encargado de la suscripción y recepción de los mensajes publicados.

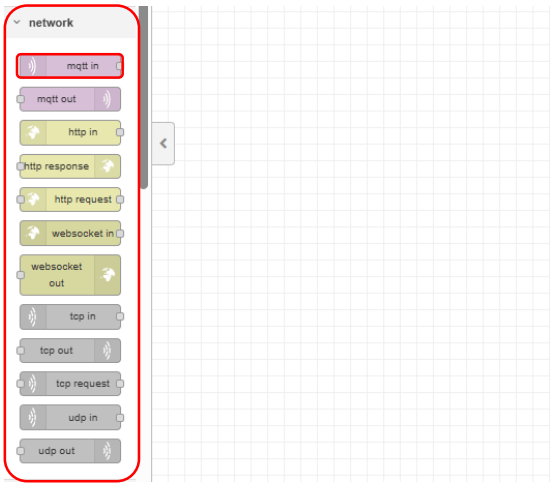


Figura 25. Nodos para la conexión MQTT [Fuente propia]

Para iniciar se arrastra un nodo al espacio de trabajo :

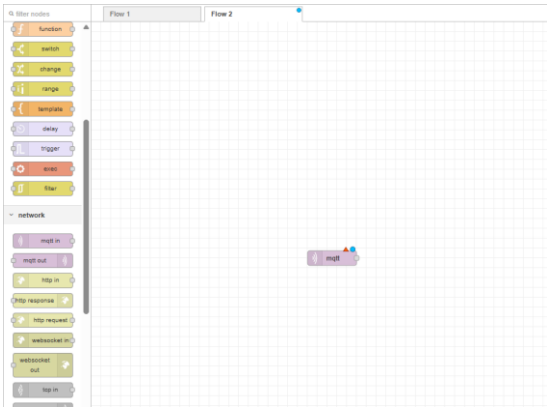


Figura 26. Nodo "MQTT in" [Fuente propia]

Al presionar doble clic al nodo se podrá acceder a la configuración para establecer la comunicación con el cluster creado en HiveMQ.

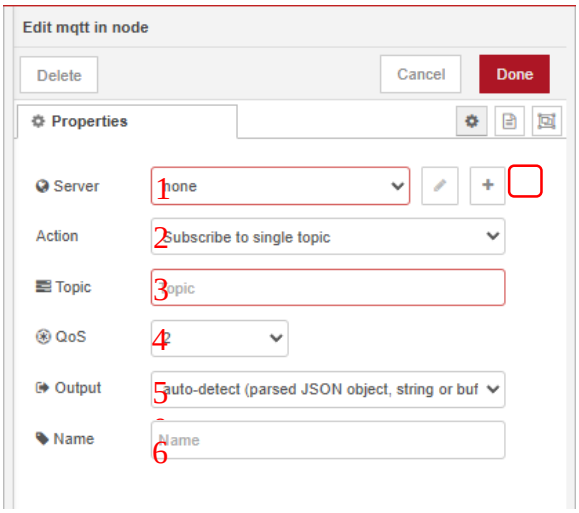


Figura 27. Configuración Nodo "MQTT in" [Fuente propia]

En el recuadro de configuración MQTT que brinda Node-Red se podrán establecer los siguientes parámetros:

1. **Servidor:** En esta sección se añadirá el servidor MQTT, en este caso será el del cluster creado en HiveMQ. En el botón "+" se configurarán los siguientes parámetros:

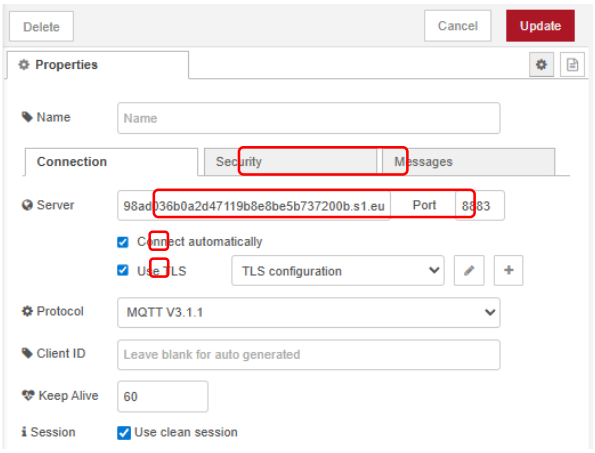


Figura 28. Configuración seguridad Nodo "MQTT in" con parámetros de HiveMQ [Fuente propia]

- Servidor: **TLS MQTT URL** (Información en Cluster)
- Puerto: **8883**
- Opción conectar automáticamente: **Si**
- Opción utilizar TLS: **Si**

En el apartado de seguridad se ingresarán los datos de la credencial, el usuario y la contraseña

2. **Acción:** Parámetro para suscribir a un solo tópico

3. **Tópico:** Nombre del tópico a suscribir

4. **QoS:** Nivel de seguridad

5. **Output:** Tipo de formato de salida

6. **Nombre:** Nombre del nodo

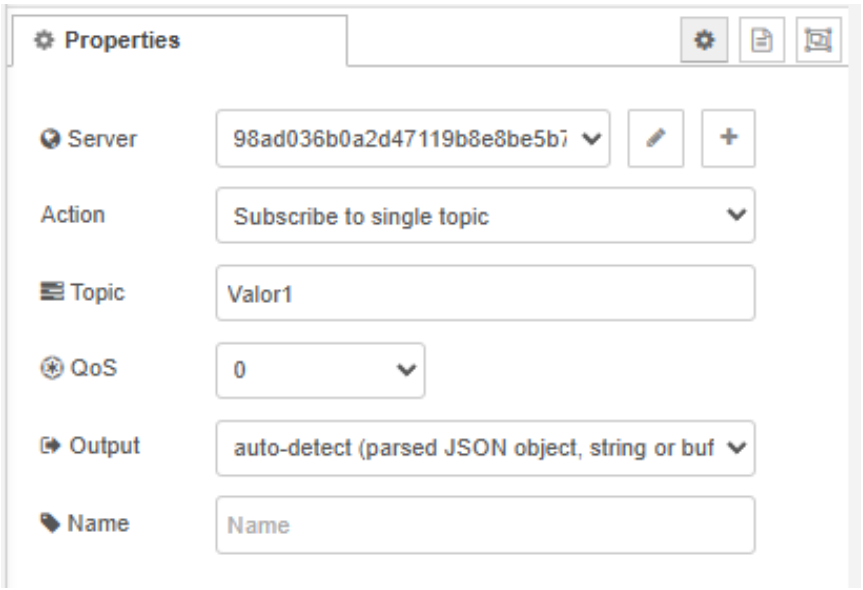


Figura 29. Configuración tópico "Valor1" [Fuente propia]

Una vez creado el nodo y configurado este se enlazará un nodo de *"debug"* , este nodo permitirá imprimir los datos en la consola de Node-Red. Para ejecutar la aplicación se presiona en el botón superior derecho en desplegar (*"Deploy"*)

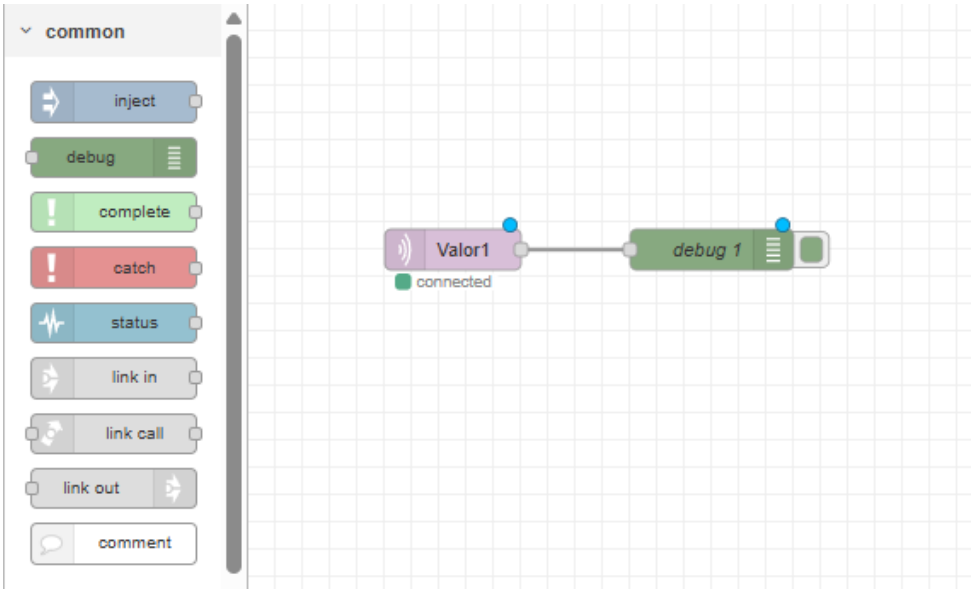


Figura 30. Conexión de nodo "MQTT in" a la consola de Node-Red [Fuente propia]

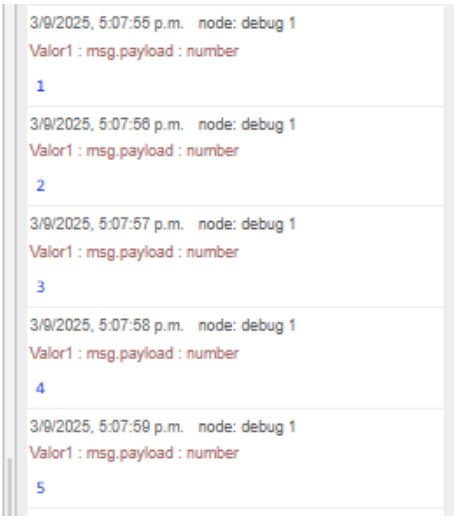


Figura 31. Valor tópico "Valor1" [Fuente propia]

6.1.6. Configuración PLC

Para programar el PLC Unitronics visión 700 es necesario descargar el VisiLogic que es el software proporcionado para programar la línea de PLC-HM Visión . Con ese objetico se descargará desde la propia página mediante el siguiente link : [VisiLogic Software: Vision & Samba Programmable Controllers]

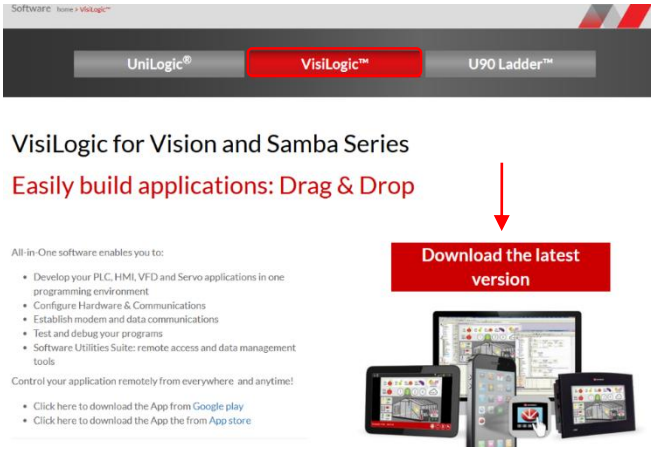


Figura 32. Descarga de VisiLogic [Fuente propia]

Se procede a la instalación de la versión más reciente de VisiLogic, un software que integra en un mismo entorno la configuración del módulo de entradas y salidas, así como la programación de la pantalla HMI incorporada en el controlador. Esta plataforma de desarrollo permite trabajar de manera centralizada en ambos elementos, optimizando el tiempo de implementación y facilitando la curva de aprendizaje. El lenguaje de programación que utiliza VisiLogic es el lenguaje de contactos (Ladder), ampliamente adoptado en el sector

industrial por su similitud con los diagramas eléctricos tradicionales. Gracias a esta característica, el estudiante y el docente pueden comprender y diseñar programas de control de una forma más intuitiva y práctica, aun sin contar con experiencia previa en programación estructurada.

La versión instalada incluye una biblioteca de bloques de funciones predefinidas, las cuales permiten programar de manera más ágil los distintos controles que se aplicarán en las prácticas de la planta didáctica. Estos bloques abarcan desde operaciones básicas de lógica hasta procesos más avanzados, lo que garantiza una amplia cobertura de aplicaciones industriales. Adicionalmente, cada bloque cuenta con una guía de uso que describe su funcionamiento y posibles aplicaciones. Este recurso constituye una herramienta pedagógica fundamental, facilitando la comprensión del software y del entorno de programación, garantizando así una introducción sólida a la plataforma del fabricante y a los principios de la automatización industrial.

Descargado VisiLogic se presentará el siguiente entorno de trabajo:

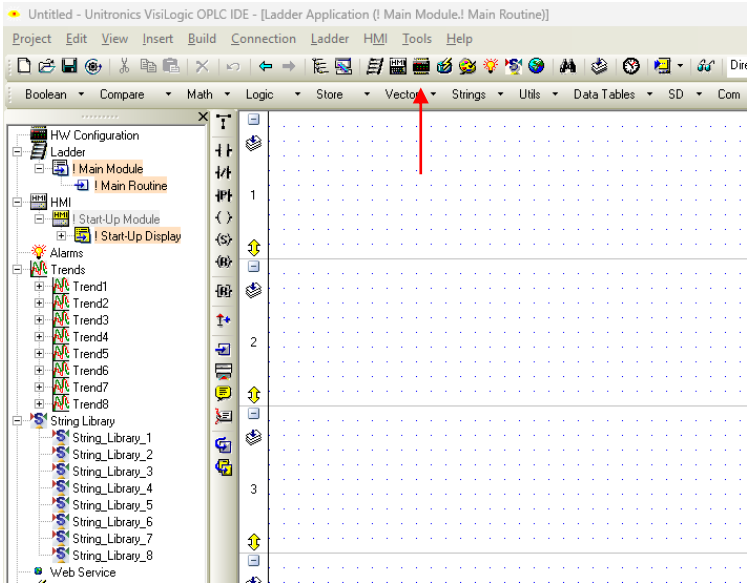


Figura 33. Configuración del modelo de PLC-HMI en VisiLogic [Fuente propia]

Para configurar la versión del PLC se pulsa en *"Hardware configuration"* esta ventana permitirá establecer la versión del HMI, que para este caso será V700 y en el apartado de *"Snap I/O"* se elegirá el módulo de entradas y salidas correspondiente a V200-18-E3B.

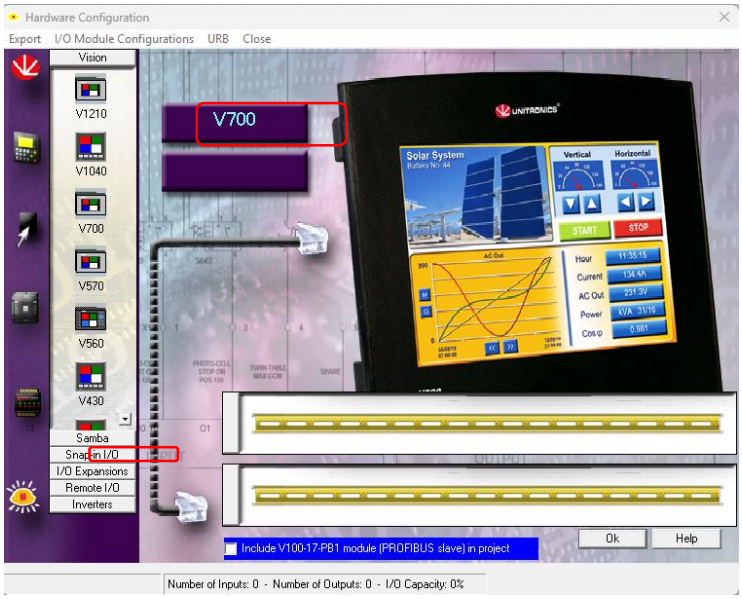


Figura 34. Configuración HMI [Fuente propia]

Con esta configuración el PLC estará listo para programar.

6.1.7. Conexión ModBus PLC Unitronics y apertura de puertos ethernet

Para realizar la conexión con ModBus RTU es necesario realizar el análisis de cómo funcionan los bloques de la conexión para efectuar la comunicación serial y el envío de los datos de los sensores y actuadores. En VisiLogic la configuración de ModBus se realiza mediante el bloque *"ModBus Config"* este bloque permite definir principalmente el puerto en el que estará el dispositivo, abriendo así la comunicación con los demás equipos que se conecten. Para el proyecto SEDAI se estableció que el PLC iba a funcionar como esclavo y la ESP32 como maestro, esta jerarquía permite acceder a los datos del PLC a través de las marcas de memoria y así extraer los valores análogos de los sensores y los valores booleanos de los actuadores. Una vez obtenidos se enviarán por MQTT al servidor de Node-RED . El protocolo Modbus y MQTT permite procesar todos los datos que el usuario defina, la única limitación es la capacidad de datos que permite HiveMQ con la cuenta gratuita que es de 10GB.

Para realizar la configuración ModBus primero hay que realizar dos pasos previos, se creara una contraseña por seguridad debido a unos ataques cibernéticos que sufrieron en unas instalaciones de sistemas agua en Estados Unidos. Es por ello que antes de realizar cualquier programación es importante crear este parámetro de seguridad. Para ello al inicio del programa se crean los siguientes bloques:

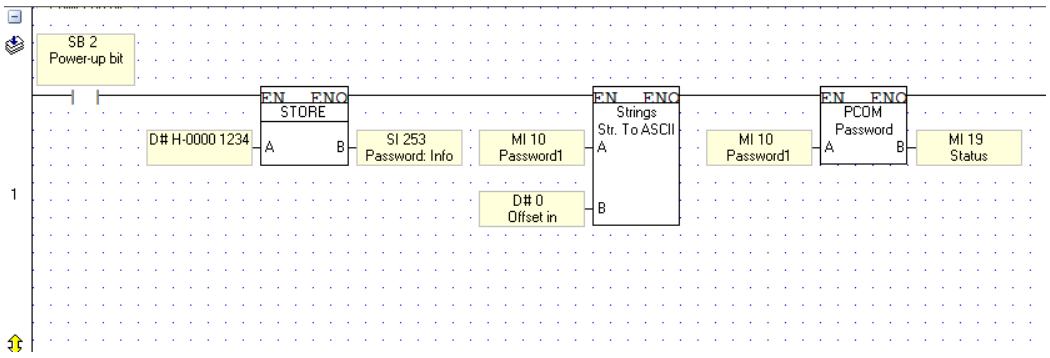


Figura 35. Creación del usuario y contraseña [Fuente propia]

Este programa almacenara una clave que en este caso es *"1234"* y se almacenara en un bloque *"store"* siendo SI 253 la marca de memoria asignada para información de la contraseña. Esta contraseña se ingresará en el bloque *"Pcom password"* quien una vez verificada cambiará el estatus para el acceso a la ejecución del programa. Esta sección se ejecuta una vez inicia el programa gracias a la acción del *"Power-ip bit"* que cambia a el estado de 0 a 1 al instante que se pone en marcha el PLC. Ya con esa sección el PLC estará en funcionamiento. Esto se debe repetir con cualquier programa que se vaya a realizar. El fabricante recomienda establecer una contraseña con mayor rigurosidad por seguridad, ya que estos equipos no están exentos de ataques cibernéticos que vulneren principios de funcionamiento y estabilidad.

Seguido a ello se abrirá el puerto ethernet del HMI Vision V700 este paso será útil más adelante para conectar el HMI a un dispositivo Android o iOS mediante una aplicación propia de Unitronics llamada *"Remote operator"* cuya acción ampliará la capacidad de control del proyecto con IoT. Para configurar el puerto ethernet se deben tener los siguientes bloques de función:

1. ***"TCP/IP Card Init":*** Este bloque permite abrir el puerto ethernet del HMI, para ello es necesario establecer la IP del equipo, la máscara de subred y la puerta de enlace. Estos puertos deberán fijarse de acuerdo a la red a la que vayamos conectar físicamente el HMI con el cable RJ45.

2. ***"TCP/IP Socket init":*** Este bloque permite definir el puerto con el que el dispositivo se conectara al HMI y la cantidad de bloques de esta función es proporcional a la cantidad de dispositivos que harán un enlace para el acceso al HMI, por ejemplo, si al HMI se desean acceder con cinco dispositivos móviles se crearan paralelamente cinco bloques de esta función. Para esta versión de HMI la tarjeta de red permite un máximo de 8 socket configurables (socket 0 – socket 7). En este bloque se configura por el protocolo TCP y el puerto. Se deja por defecto en servidor.

3. ***"PLC Name":*** Es fundamental asignar un nombre al dispositivo para establecer el enlace. Este bloque permite definir dicha identificación, la cual será visible al momento de realizar una conexión remota

La implementación de los bloques será de la siguiente forma:

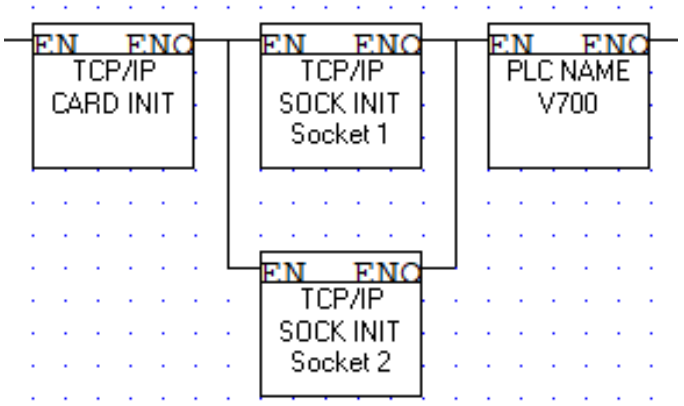


Figura 36. Apertura del puerto ethernet [Fuente propia]

Luego de esos bloques se agrega un bloque de función *"Com Init"* el cual permite abrir el puerto serial del PLC, al igual que un puerto normal se configurarán el id el número de bits , la velocidad del puerto, bits de variedad y en protocolo se manejará el estándar RS232. Esta configuración será indispensable para el funcionamiento del ModBus RTU. Unido con este bloque estará la configuración de ModBus

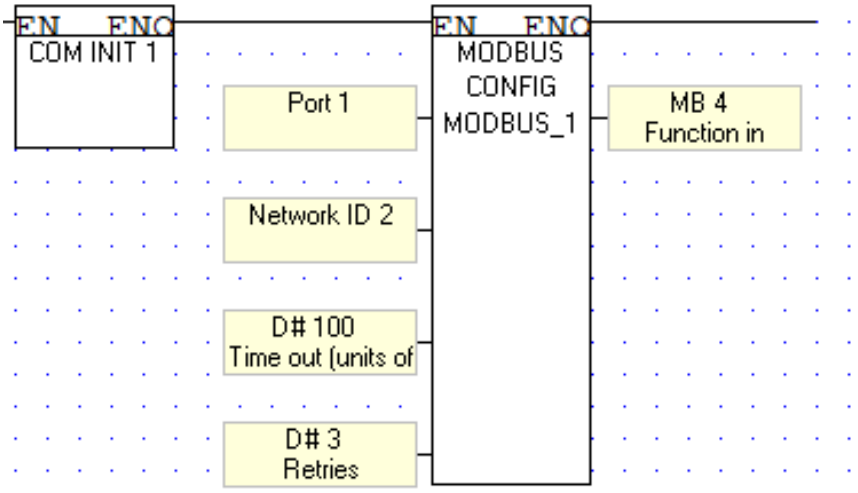


Figura 37. Bloque de COM y ModBus [Fuente propia]

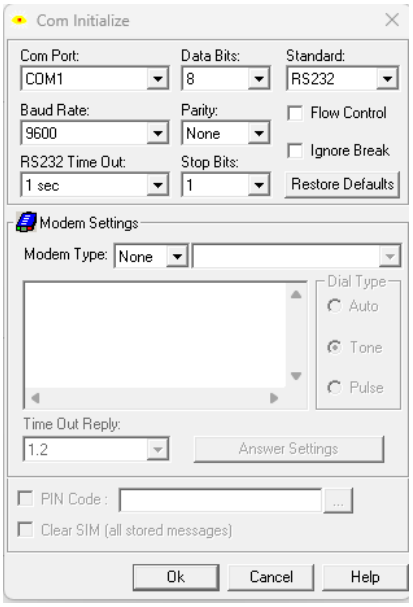


Figura 38. Configuración del puerto [Fuente propia]

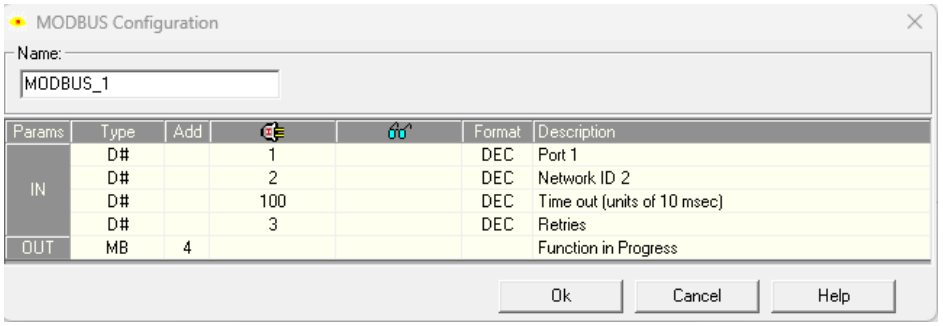


Figura 39. Configuración bloque ModBus [Fuente propia]

6.1.8. Prueba conexión PLC-ESP32 por ModBus

Para establecer la comunicación entre el PLC y el microcontrolador se conectó el conector RJ11 del PLC a un conversor propio de Unitronics denominado *"MJ10-22-CS25"*. A partir de este conversor se obtiene un conector DB9, el cual se enlaza con un módulo MAX3232 que habilita la comunicación serial entre dispositivos utilizando niveles de voltaje TTL, requeridos por la ESP32. La conexión directa de estas dos partes no permite la trasmisión de datos entre los dispositivos. Las pruebas realizadas con conexión directa no mostraron ningún tipo de comunicación bidireccional o unidireccional. Es por ello que la comunicación por ModBus del PLC se comprobó mediante la conexión con ***"ModBus poll"*** este programa permite crear una conexión con los dispositivos ModBus conectados al computador. Con esta prueba se verifica que el Modbus configurado en el PLC Unitronics está funcionando correctamente y la falla corresponde a la recepción y envío de datos del módulo RS3232.

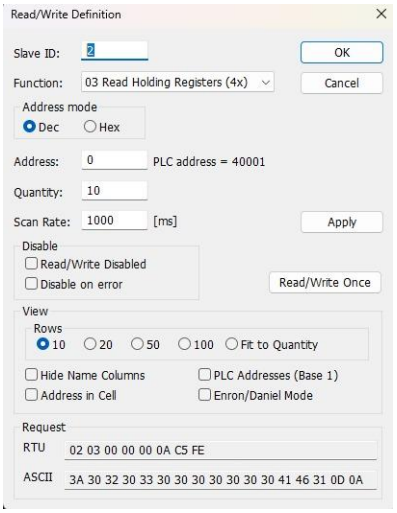


Figura 40. Configuración Modbus Poll [Fuente propia]

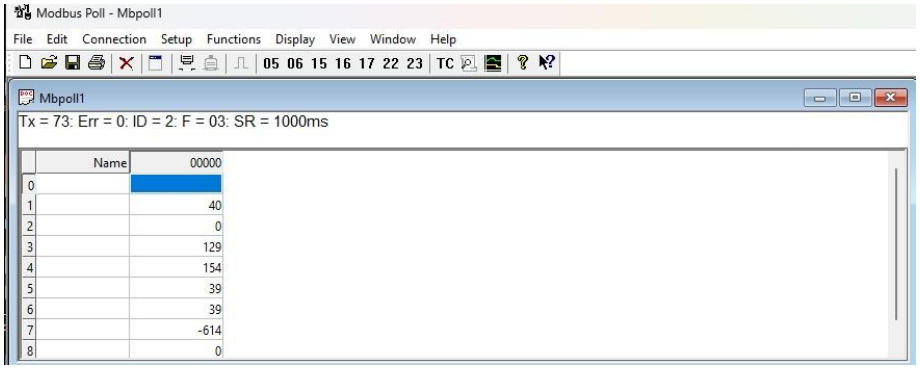


Figura 41. Datos recibidos [Fuente propia]

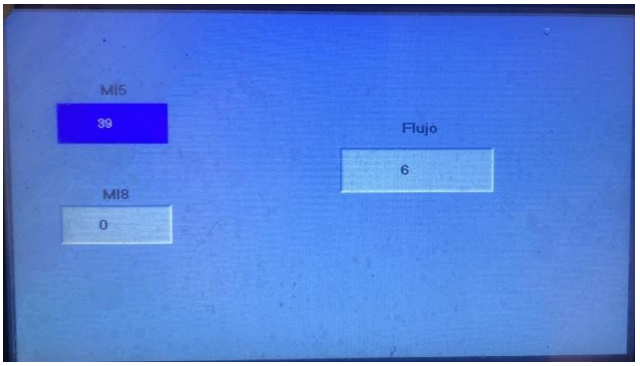


Figura 42. Visualización en HMI [Fuente propia]

El error identificado parte de realizar un análisis de cómo funciona normalmente el estándar RS232. Cuando dos equipos se comunican por el protocolo RS232 hay dos pines que hacen control de flujo de los datos que son RTS y CTS, estos pines se encargan de solicitar y comprobar el envío de datos respectivamente. Cuando un paquete de datos se trasmite de un equipo a otro equipo se envía una señal del segundo dispositivo denominada CTS. Antes de

trasmitir esta señal el pin RTS envía un permiso para empezar con el envío de paquetes, sin esta confirmación la comunicación no inicia. Ambos equipos deben tener esta capacidad para que la comunicación sea bidireccional. En el proyecto implementado el PLC cuenta con este sistema pero la ESP32 con el módulo MAX3232 no es capaz de generar estos permisos y confirmaciones.

Lammert Bies detalla en su bloc el funcionamiento de un ***"null modem"***, en él explica que normalmente el protocolo RS232 es un método de comunicación bien definido entre computadoras y equipos de comunicación de datos (módems). Este estándar funciona porque la recepción de datos está coordinada entre estos dos dispositivos, es decir, el envío de datos es exitoso debido a que el receptor confirma y el emisor envía. Lammert define modem como un equipo de comunicación de datos, es así que siguiendo la analogía de Lammert el PLC sería una computadora y el modem seria la ESP32 conjunto con el módulo MAX3232. Por las pruebas realizadas y siguiendo la hoja de datos la recepción de datos no es confirmada por el módulo debido a que no genera estas señales de confirmación al PLC.

Para solucionar este inconveniente Lammert propone cuatro soluciones que son conexiones que saltan estas confirmaciones y permiten la comunicación de los dispositivos conectados. Estas soluciones no son 100% efectivas ya que tienen problemas de compatibilidad y en muchos casos pueden resultar en pérdidas de datos. Estos modos de modem nulo son:

- Módem nulo sin protocolo de enlace
- Módem nulo con protocolo de enlace de bucle hacia atrás
- Módem nulo con protocolo de enlace parcial
- Módem nulo con protocolo de enlace completo

Para el proyecto SEDAI se utilizó un modem nulo con protocolo de enlace de bucle hacia atrás. Se realiza las conexiones propuestas por Lammert y se verifican pruebas de comunicación,

6.1.9. Diseño HMI SCADA

Para realizar el diseño del SCADA se utilizó Inskape este software brinda la posibilidad de crear archivos .SVG. que es un formato de gráfico vectorial, basado en XML, que se utiliza para representar imágenes bidimensionales en la web y en otras aplicaciones. Este tipo de archivos permite realizar la conexión con Node-RED, ya que cada elemento que está en el HMI se puede vincular a través de un ID. Asociar estos elementos a los valores recibidos de los sensores y actuadores permite publicar los datos, cambiar colores y crear animaciones en el HMI. Como diseño se tomó principalmente el modelo propuesto por Jean Carlo Leguizamón recreando así un HMI donde se identifiquen todos los componentes del sistema modular SEDAI.

6.1.10. Diseño de placa comunicación PLC-ESP32

El diseño de la placa PCB se realizó bajo las conexiones hechas del modem nulo con protocolo de enlace de bucle hacia atrás. En esta placa va a ir contenido todo el módulo que soporta el middleware del sistema y permitirá la comunicación entre el PLC y finalmente el SCADA. Los elementos que tendrá son:

- 1 ESP32
- 2 DB9 macho
- 1 MAX3232
- 1 conversor MJ10-22-CS25
- 1 regulador de voltaje LM7805

Con estos elementos se crea el siguiente diseño en 3D de la placa:

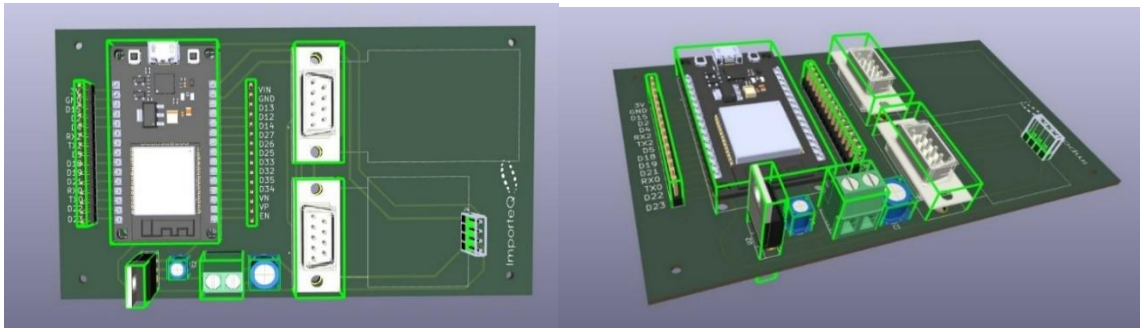


Figura 43. Placa de comunicación ModBus, modelo en 3D [Fuente propia]

6.1.11. Prueba final comunicación PLC-SCADA

Para comprobar la comunicación y el funcionamiento del circuito anteriormente realizado se hace la prueba de envíos de datos desde el PLC al SCADA con los protocolos anteriormente mencionados. Para llevar a cabo esta prueba se conecta los elementos a la placa de circuitos y se desarrolla un programa que envía un dato entero del PLC al Node-Red y se imprime en la consola, posteriormente se envían más datos al mismo tiempo y se grafican en el dashboard. Una vez se obtiene la correcta recepción de los datos las variables se enlazan con los ID de archivo .SVG del SCADA y se observan los datos en el SCADA. Esta prueba es fundamental porque con ella se verifica la corrección de "modem nulo con protocolo de enlace de bucle hacia atrás" de Lammert para la conexión ModBus entre el microcontrolador y la comunicación MQTT para el SCADA.

6.1.12. Conexión PLC con teléfono móvil

Una de las funciones que tiene las versiones más recientes de los HMI's Unitronics es la función de "*Remote operator"*. Esta función permite conectar y controlar el HMI desde un dispositivo móvil ya sea Android o iOS. El HMI V700 soporta esta función y para utilizar esta característica es necesario conectar el dispositivo a una red con un cable RJ45 y abrir el puerto ethernet como anteriormente se explicó en la sección 6.1.7. Esta caracteriza fue la principal razón del porque se utilizó ModBus RTU y no ModBus IP, ya que al aplicar este protocolo la función no se podría implementar debido a que el HMI solo cuenta con un puerto ethernet. Esta función amplia las funciones IoT de la planta y el control de forma remota no solamente con el SCADA. Este tipo de tecnologías enfocan al estudiante y al docente el énfasis actual de la industria.



Figura 44. Remote Operator [Fuente propia]

Para realizar la conexión se necesitan parámetros como el nombre asignado al PLC, la dirección IP, contraseña y tipo de control (vista o control completo). Para que funcione la conexión es necesario que el PLC este encendido y este en el mismo segmento de red que el dispositivo donde se va a realizar la conexión . Por socket se puede realizar máximo una conexión por lo que si se desea conectar más dispositivos móviles hay que configurar proporcionalmente los mismos sockets. V700 admite máximo siete conexiones simultaneas. Esta conexión no afecta la realizada por ModBus

6.1.13. Prueba completa del sistema con el servidor

Para comprobar el funcionamiento optimo del sistema con su módulo de IoT se realiza una prueba con las prácticas de control de nivel, temperatura y flujo en conjunto. Esta actividad amplia la capacidad de trabajo de la planta, aplicando este proceso directamente a la documentación de la planta, detallando y proponiendo al usuario a crear operaciones con los elementos de SEDAI con un fin específico. Con el módulo IoT se propone un SCADA el cual ve el estado de los sensores y actuadores. Plasmando datos a tiempo real mediante los protocolos Modbus y MQTT. Las actividades, soluciones, HMI's y SCADA es algo que se

anexará a las guías realizadas. Con este fin se propone brindar un material de apoyo a los estudiantes siendo base para seguir profundizando con el sistema

Las guías como material de apoyo están pensadas para el estudiante y como elemento de soporte para el docente. El fin de esta documentación pretender orientar al usuario a la correcta manipulación de SEDAI y a conocer los elementos de control que lo conforman además de teoría que hace posible el funcionamiento del sistema. Para la documentación se tiene en cuenta los siguientes parámetros:

- Plano con entradas y salidas del PLC
- Elementos de control (sensores y actuadores)
- Manipulación del módulo IoT
- Teoría de control
- Diagrama P&ID
- Rangos de seguridad
- Actividades



Figura 45. Visualización guías
[Fuente propi

7. Resultados

A continuación se presentan los resultados de las actividades desarrolladas por Nicolás Sopo Bobadilla y Jean Carlo Leguizamón Martínez. Ambos planes de trabajo se complementan entre sí, destacando la integración de la comunicación entre dispositivos y la construcción física de la planta. El propósito de estos resultados es responder a la pregunta de investigación que dio origen al proyecto: diseñar y materializar un prototipo que sirva como modelo de aprendizaje de conceptos prácticos y teóricos en el ámbito de la automatización industrial. Cabe resaltar que este proyecto no está dirigido exclusivamente a docentes, sino que ha sido concebido como una herramienta accesible para cualquier persona interesada en la automatización, ofreciendo una base sólida y un punto de partida para enfrentar y resolver problemas de mayor complejidad basado en la industria 4.0.

7.1. Compra de elementos

Para la adquisición de los componentes se estableció una división entre los elementos a comprar a nivel nacional e internacional. Esta estrategia permitió ajustarse al presupuesto asignado al proyecto y, al mismo tiempo, atender las limitaciones de disponibilidad, dado que algunos elementos no podían adquirirse fácilmente en el mercado local. A continuación se presenta una tabla con los elementos más relevantes que conforman el proyecto

Componente	Imagen	Características	Cantidad	Compra
Cofre metálico doble fondo		En este cofre estará instalado toda la parte eléctrica de tablero. En el estará incorporado el PLC, interruptores y pilotos además de soportar toda la parte de cableado y conexiones. Sus dimensiones están definidas por 500mm de alto, 500mm anchi y 250mm de profundo	1 U	Nacional
Pilotos		Los pilotos se eligen de 24V de acuerdo a la salida del PLC. Esto con el fin de evitar conexiones extras con relés a 120V AC. Los colores elegidos son: Verde, Rojo, Amarillo	3 U	Nacional

Tabla 2. Elementos comprados [Fuente Propia].

Selector Switch		Este selector permite la selección de modos, se incorpora con el fin de darle versatilidad al estudiante a la hora de programar un proceso	1 U	Nacional
Interruptores NO y NC		Mediante estos pulsadores se podrán activar las entradas digitales del PLC	3 U-NO 1U-NC	Nacional
Parada de emergencia		Con este switch se podrá simular una parada de emergencia. Es un interruptor NC	1 U	Nacional
HMI Vision700		La **HMI Vision700** es una interfaz hombre-máquina de Unitronics que integra pantalla táctil y PLC, permitiendo control y supervisión de procesos en un solo equipo.		
Módulo de entrada y salidas V200-18-E3B Unitronics		El V200-18-E3B de Unitronics es un módulo de entradas y salidas digitales para controladores PLC. Tiene 18 puntos de E/S (6 entradas y 12 salidas), se utiliza para conectar sensores, interruptores, relés y otros dispositivos industriales. Permite expandir las capacidades de un PLC, facilitando la interacción con el entorno de un sistema automatizado.	1 U	Internacional

Fuente Unitronics 24V		La fuente Unitronics será encarga de alimentar principalmente el PLC y las entradas digitales del controlador	1 U	Interacional
Bornes de conexión		En el barraje es el punto de conexión de todo el tablero: 24V, 0V y neutro	1 U	Nacional
Canaleta ranurada		Se utiliza para almacenar y organizar los cables que salen del barraje, componentes y PLC	4m	Nacional
Riel		Se utiliza para fijar la fuente de alimentación, el breaker y barraje de conexión	1 M	Nacional
Toma corriente		Está ubicada en un lateral de lateral del tablero y servirá como tomas corriente convencionales	1 U	Nacional
Braker		Dispositivo de protección eléctrica diseñado para interrumpir la corriente eléctrica en un circuito en caso de una sobrecarga o cortocircuito	1 U	Nacional

Cable calibre 16		Se utilizará para realizar todas las conexiones internas del tablero	20 m	Nacional
Terminales de Ojo		Se utilizar para acoplar los cables con los puntos de conexión	50U	Nacional
Terminales de punta		Se utilizar para acoplar los cables con los puntos de conexión	50U	Nacional
Terminales de u		Se utilizar para acoplar los cables con los puntos de conexión	50 U	Nacional
Sensor de nivel TL-136		Este sensor funciona por inmersión mediante la presión hidrostática. El tango de media es de 0 a 1 metro	1 U	Internacional
Sensor de temperatura termocupla tipo J		La **termocupla tipo J (hierro–constantán)** opera de **–40 °C a 750 °C**, con posibilidad de llegar hasta **1.200 °C** en condiciones especiales, aunque se recomienda no superar los **750 °C** por la rápida oxidación del hierro.	1U	Nacional

Sensor de flujo YF-S201		YF-S201 es un sensor de efecto hall de tamaño ½” y de 24V	1U	Nacional
Variador de frecuencia Optidrive		El Optidrive E3 es un variador de frecuencia que controla la velocidad de motores AC, ofreciendo configuración sencilla, ahorro de energía y versatilidad en aplicaciones industriales.	1 U	Nacional
Válvula proporcional		La válvula proporcional es un dispositivo que ajusta de forma continua el caudal o la presión de un fluido según una señal eléctrica de control.	1 U	Internacional
Válvulas solenoides		La **válvula solenoide** es un dispositivo electromecánico que abre o cierra el paso de un fluido mediante una bobina eléctrica.	3 U	Internacional
Relé de estado solido		El **relé de estado sólido** es un dispositivo electrónico que conmuta cargas eléctricas sin partes mecánicas, usando semiconductores para ofrecer mayor durabilidad y velocidad.	1 U	Internacional
Microcontrolador ESP32		El **ESP32** es un microcontrolador de bajo costo y alto rendimiento que integra WiFi y Bluetooth, ideal para proyectos de IoT y automatización.	1 U	Nacional

Electrobomba ½ Hp		La **electrobomba de ½ Hp** es un equipo electromecánico que impulsa fluidos utilizando un motor de media potencia, común en sistemas de agua y procesos industriales.	1 U	Nacional
Radiador		El **radiador** es un intercambiador de calor que disipa la energía térmica de un fluido hacia el ambiente, usado en sistemas de refrigeración y climatización.	1U	Nacional

7.1.1. Plano 3D

Para comenzar a realizar el montaje del tablero y la estructura es importante contar con los planos realizados a lo largo de las actividades. Para esta fase surgieron cambios principalmente por el tamaño y ubicación de entradas y salidas del radiador por lo que la versión definitiva tuvo que sufrir cambios en el sistema de circulación de agua. La idea inicial del montaje del proyecto nació en Tinkercad por lo que los planos surgieron de la construcción final de la planta en 3D, a continuación se presentara el diseño final de SEDAI.

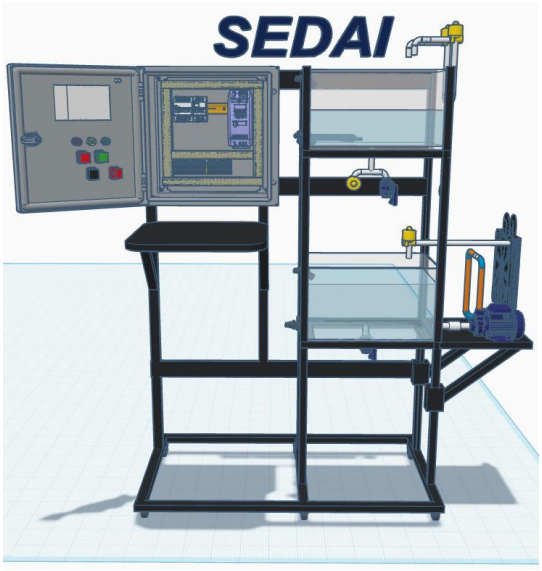
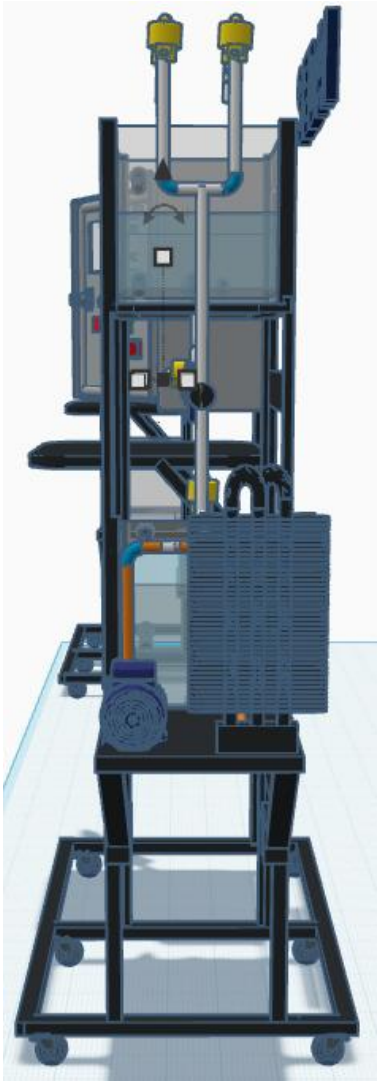


Figura 46. Diseño 3D prototipo SEDAI [Fuente propia]



En la parte lateral de la planta se encuentra el sistema de circulación de agua, este sistema es conformado por la electrobomba que realiza la extracción del agua del tanque inferior y la envía directamente al radiador, esto con el fin de enfriar el agua que baja del tanque superior donde se encuentra la resistencia. Posteriormente la salida del radiador hay una división, una para la recirculación de agua nuevamente al tanque inferior mediante una válvula solenoide. Y la segunda división dirige el caudal al tanque superior donde se encuentra el sensor de flujo, una válvula solenoide y la válvula proporcional.

El control de estos elementos se realiza por medio del PLC y son fundamentales para el control de nivel y flujo pudiendo así dirigir el caudal según la programación realizada de cada elemento. La medida para del tubo seleccionado es de ½", por lo que todas las válvulas se eligieron de esta medida. Para la conexión de la electrobomba, cuya salida es de 1", se instalaron acoples de reducción para adaptar la tubería al radiador, el cual cuenta con entrada y salida de ¾"

El principal objetivo de este diseño es disponer los elementos de modo que para el usuario final se facilite un reconocimiento de todos los sensores y actuadores que conforman el sistema, de esta forma se identifica el funcionamiento y la función de cada elemento dentro del sistema. Los elementos se modelaron a escala con el propósito de obtener medidas precisas de la estructura y servir como referencia para su construcción. El plano en 3D facilita un dimensionamiento más tangible de la estructura y proporciona un mejor control sobre los detalles constructivos.

Figura 47. SEDAI: Sistema de circulación [Fuente propia]

Dada la medida de diseño de los tanques se calcula el volumen máximo que puede soportar los tanques.

Dimisiones de los tanques:

- Largo = 50cm
- Alto = 30cm
- Ancho = 30cm

$$V = 50\ cm\ x\ 30\ cm\ x\ 30\ cm = 45.000\ cm^3 \quad (1)$$

$$45.000\ cm^3 = 45\ L$$

Para evitar desbordamientos de agua es importante limitar la medida de altura, por diseño y dada las indicaciones del ingeniero a cargo del proyecto se limitó a 25 cm de altura por lo que ahora el volumen ahora es

$$V = 50\ cm\ x\ 30\ cm\ x\ 25\ cm = 37.500\ cm^3 \quad (2)$$

$$37.500\ cm^3 = 37.5\ L$$

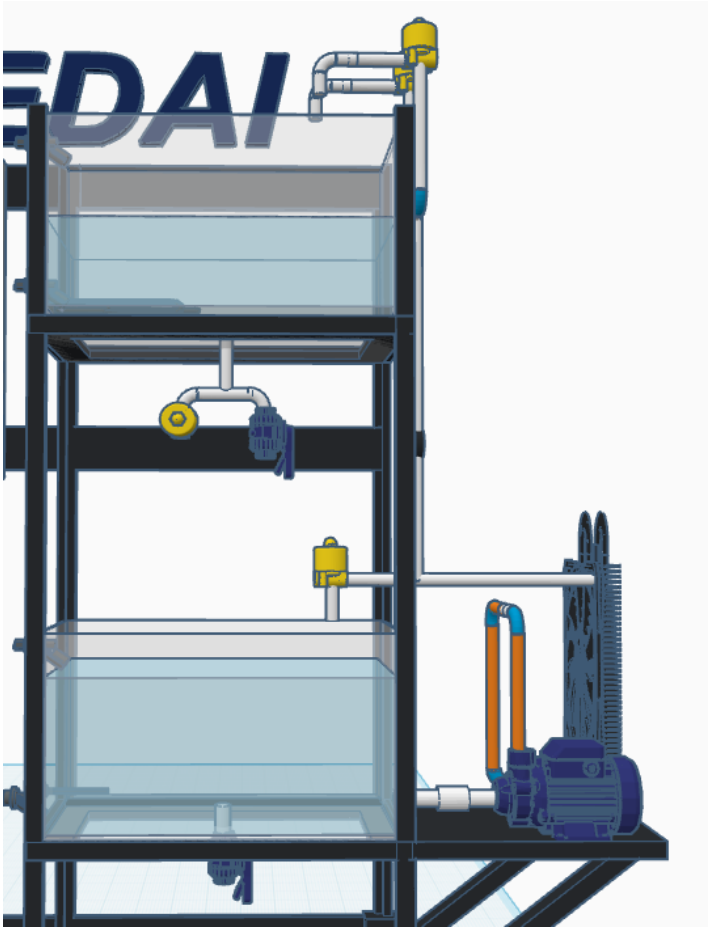


Figura 48. SEDAI: Tanques [Fuente propia]

Para descargar el agua del tanque superior al inferior se utilizó una válvula manual y una válvula solenoide sin presión, esto debido a que la presión que ejerce el agua al solenoide es mínima. Normalmente las válvulas solenoide necesitan una presión mínima de operación de alrededor de 0,2 – 0,3 MPa (≈ 2 a 3 bar, o 20–30 mca). Y una presión típica de trabajo de entre 0,3 y 0,8 MPa (≈ 3 a 8 bar). Por lo que era fundamental que este solenoide trabajara sin presión . Ambas válvulas de igual forma se adaptaron a la medida de la tubería de ½”. Para drenar el tanque inferior se utilizó una válvula manual de ½” igualmente



Figura 49. Diseño Tablero en Tinkercad [Fuente propia]

Desde el diseño 3D se empezó construir un diseño del tablero, y aunque no fue el prototipo definitivo ayudo definir la ubicación de los elementos de la tapa exterior y la ubicación del tablero dentro del propio sistema, así como su dimensión aproximada. En la parte inferior del cofre se diseñó un soporte el cual sirve para apoyar el computador y facilitar la programación del PLC y la ESP32.

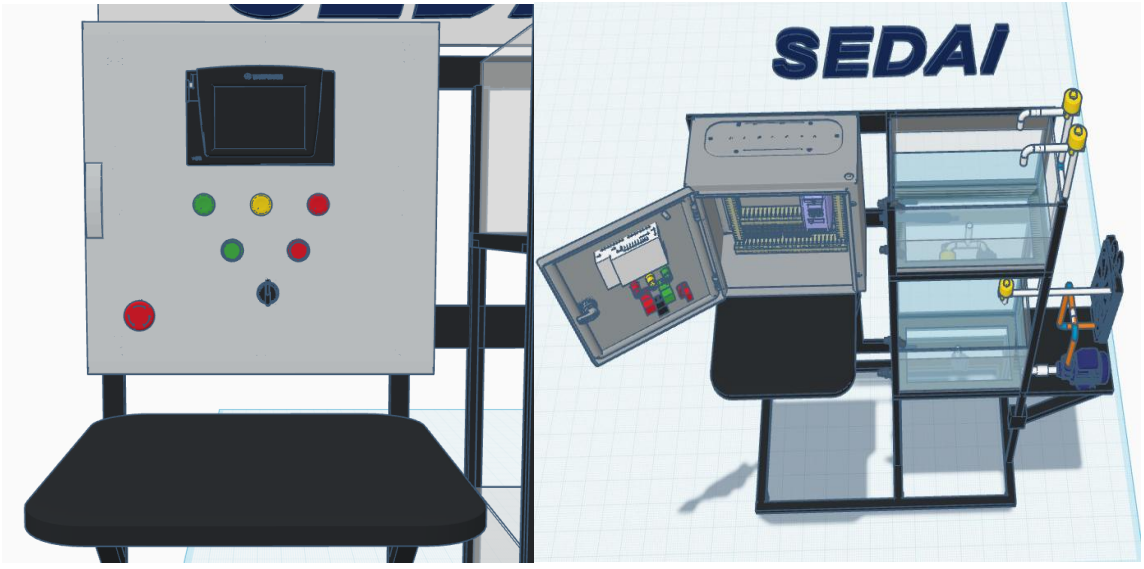


Figura 50. Vista superior [Fuente propia]

7.1.2. Diagrama P&ID y planos eléctricos

Para el diagrama P&ID y los planos eléctricos se utilizó el software Lucid Chart y QElectroTech respectivamente. Estos planos ayudan a tener un diseño más concreto del sistema, así como definir las conexiones de los sensores, actuadores, entradas y salidas del PLC. A continuación será presentará el diagrama P&ID con los elementos adquiridos para el proyecto. Los planos eléctricos se anexarán al final del documento

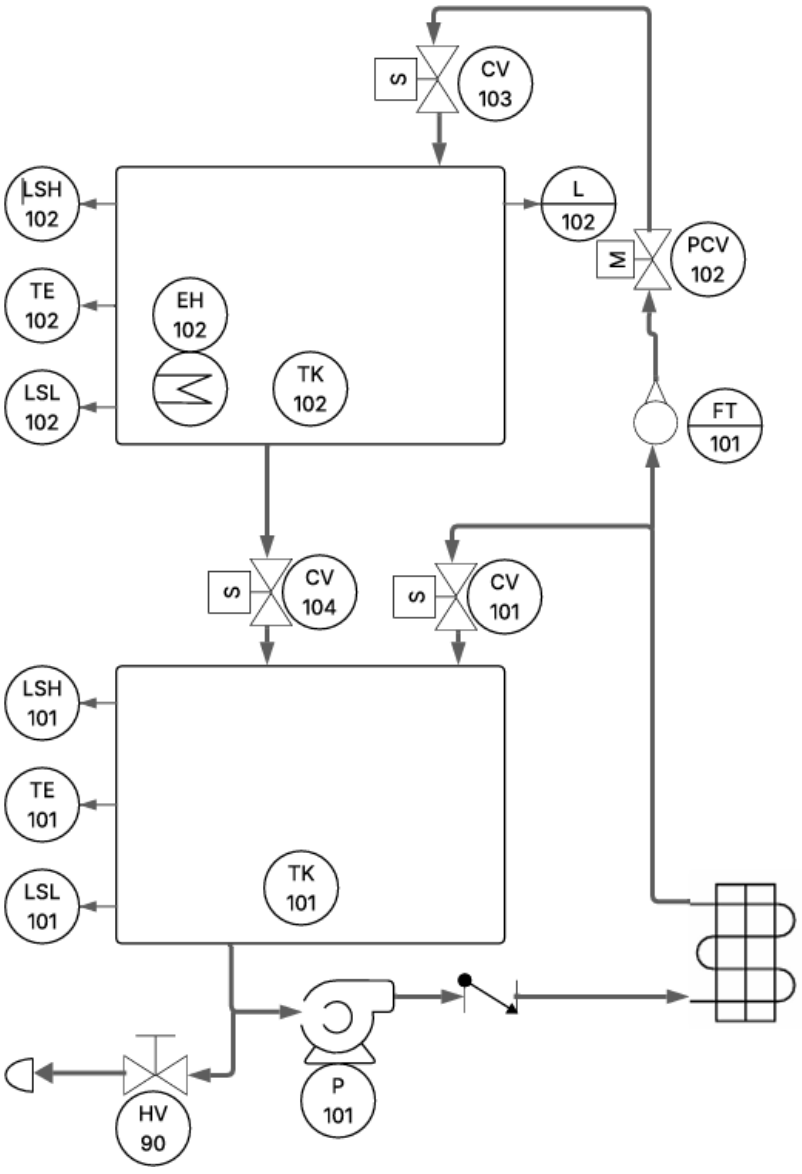


Figura 51. Diagrama P&ID [Fuente propia]

Tanques

- **TK 101** → Tanque 101.
- **TK 102** → Tanque 102.

Instrumentación de nivel

- **LSH 101** → Level Switch High (alarma de nivel alto en TK101).
- **LSL 101** → Level Switch Low (alarma de nivel bajo en TK101).
- **LSH 102** → Level Switch High (alarma de nivel alto en TK102).
- **LSL 102** → Level Switch Low (alarma de nivel bajo en TK102).

Instrumentación de temperatura

- **TE 101** → Temperature Element (sensor de temperatura en TK101).
- **TE 102** → Temperature Element (sensor de temperatura en TK102).

Calefactor

- **EH 102** → Electric Heater (resistencia de calentamiento en TK102).

Instrumentación de caudal y presión

- **FT 101** → Flow Transmitter (transmisor de caudal en la línea de salida).
- **PCV 102** → Pressure Control Valve (válvula de control de presión en la descarga de TK102).

Válvulas de control

- **CV 101** → Control Valve en la salida de TK101 hacia la línea principal.
- **CV 103** → Control Valve en la salida superior de TK102.
- **CV 104** → Control Valve en la descarga de TK102 hacia TK101.

Válvula manual

- **HV 90** → Hand Valve (válvula manual en la línea de entrada).

Bomba

- **P 101** → Pump (bomba centrífuga).

7.1.3. Diseño tablero

Antes de realizar cualquier montaje en el cofre es importante tener certeza de la ubicación de todos los elementos que conforman, esto debido a que el cofre no puede ser excesivamente grande por el peso ni lo suficientemente pequeño para que los elementos no se ajusten en el interior. Mediante la herramienta EPLAN Pro Panel 2022 se define el tamaño exacto del cofre, como resultado se tiene la distribución definitiva de todos los elementos. Esta herramienta permite exportar elementos a tamaño real y crear figuras que permiten una aproximación exacta de componentes que no cuenta el software. Este diseño parte desde la tapa interior del cofre. A continuación se evidencia los planos en 3D realizados

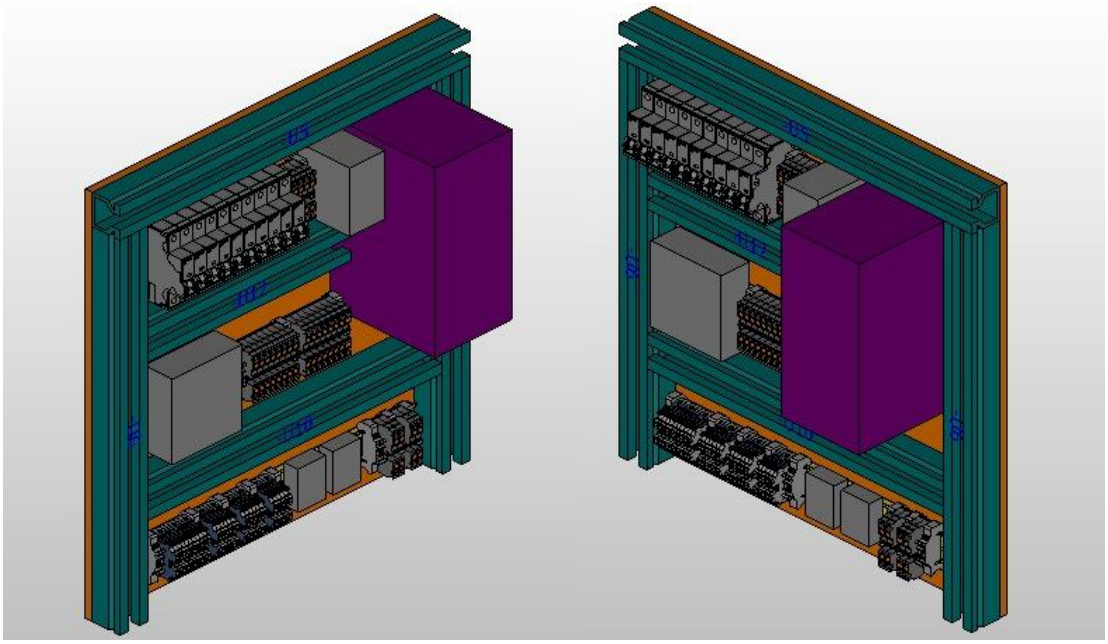


Figura 52. Plano tablero eléctrico 3D EPLAN Pro Panel 2022 [Fuente propia]

7.1.4. Montaje de tablero

Una vez definido los planos y los elementos adquiridos se proceden a realizar el montaje en el tablero, como resultado se tiene un cofre doble fondo de 50cm x 45cm x 25. Inicialmente se instala el PLC-HMI con los pilotos, indicadores, para de emergencia y selector en la parte exterior del tablero eléctrico. Para la parte interior se retira la tapa y se realiza el montaje, para esta sección se utiliza la canaleta ranurada con los raíles para insertar los elementos. De acuerdo con los planos elaborados, a medida que se realiza el cableado se colocan las marquillas correspondientes en cada componente. Esto permite mantener un orden en la conexión de los elementos externos del cofre y facilita su identificación posterior.

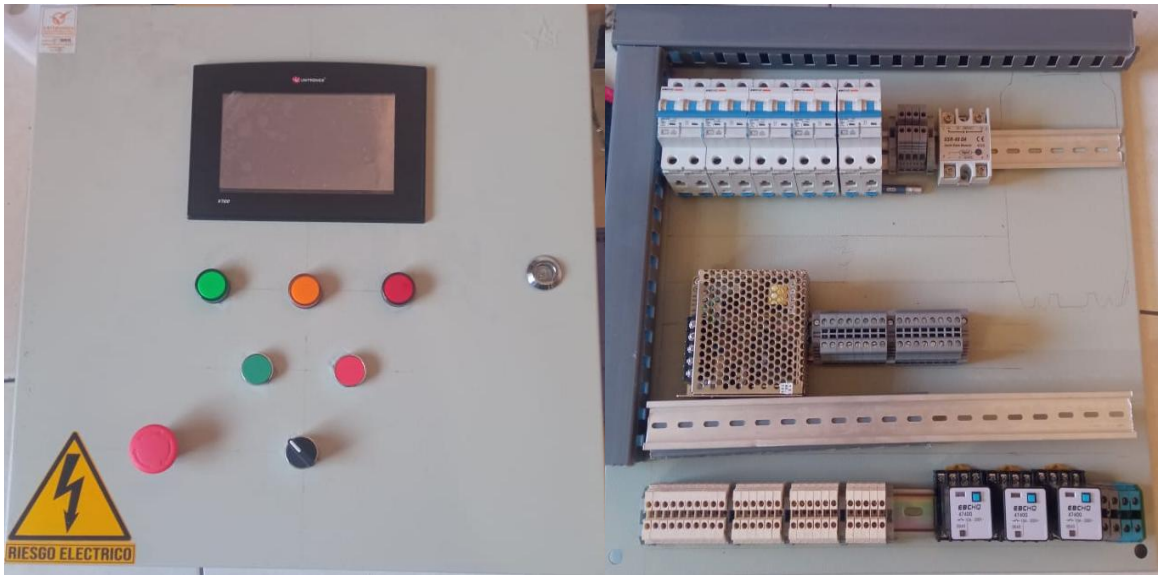


Figura 53. Montaje exterior y diseño interior

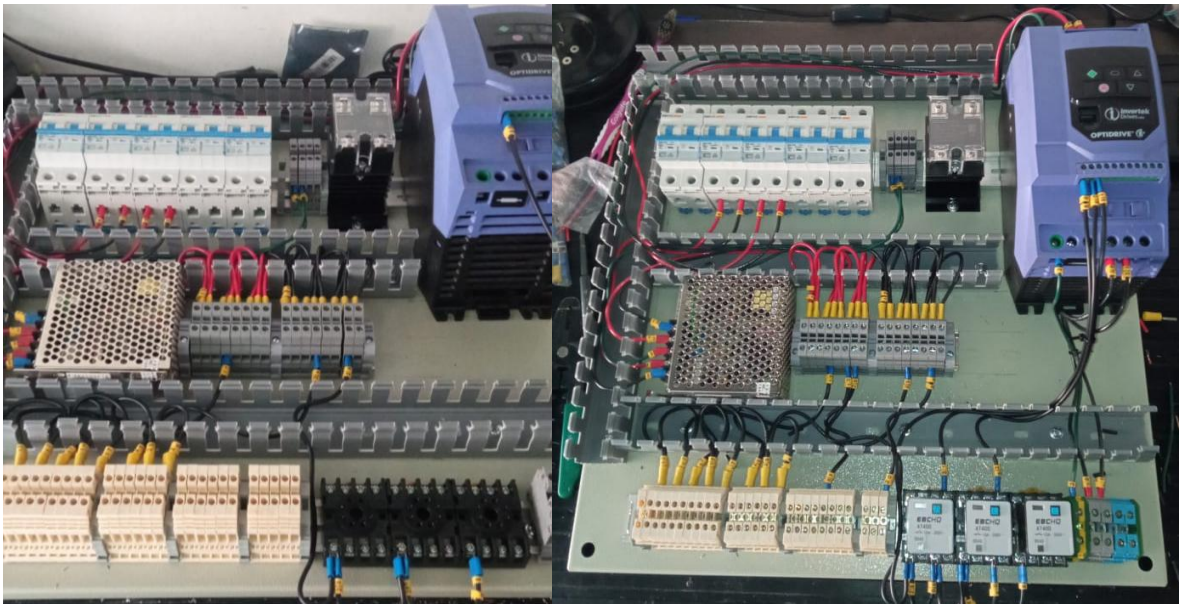


Figura 54. Cableado de tablero [Fuente propia]

Por seguridad de los elementos del sistema se decidio asilar con braker de forma individual los siguientes comoponentes:

- Sistema general
- Fuente de 24V DC
- Variador de frecuencia
- Resitencia terminica
- Tomacorriente (Estara ubicada en la parte lateral del cofre)

Una vez finalizada la instalación de los elementos internos del cofre, este se fija a la estructura para continuar con la conexión de los sensores y actuadores del sistema. El uso de marquillas y planos facilita la correcta identificación y conexión de los componentes. Posteriormente, se realizan las conexiones a los bornes y, de allí, a las entradas y salidas digitales y analógicas del módulo PLC, siguiendo las marquillas definidas en los planos previamente elaborados.

7.1.5. Montaje estructura

El montaje de la estructura está conformado por la instalación de los tanques, el cofre, sensores y actuadores. Para ello se fijó primero el tablero con anclaje metálico tipo mariposa M6, esto con el fin de asegurar de forma rigurosa el tablero y evitar accidente por su tamaño y peso. Posteriormente se comienza con la conexión de la parte exterior del tablero a los bornes de conexión interno. Una vez ubicado el tablero la conexión se comienza a realizar la distribución de los demás elementos de acuerdo a los planos y diagrama

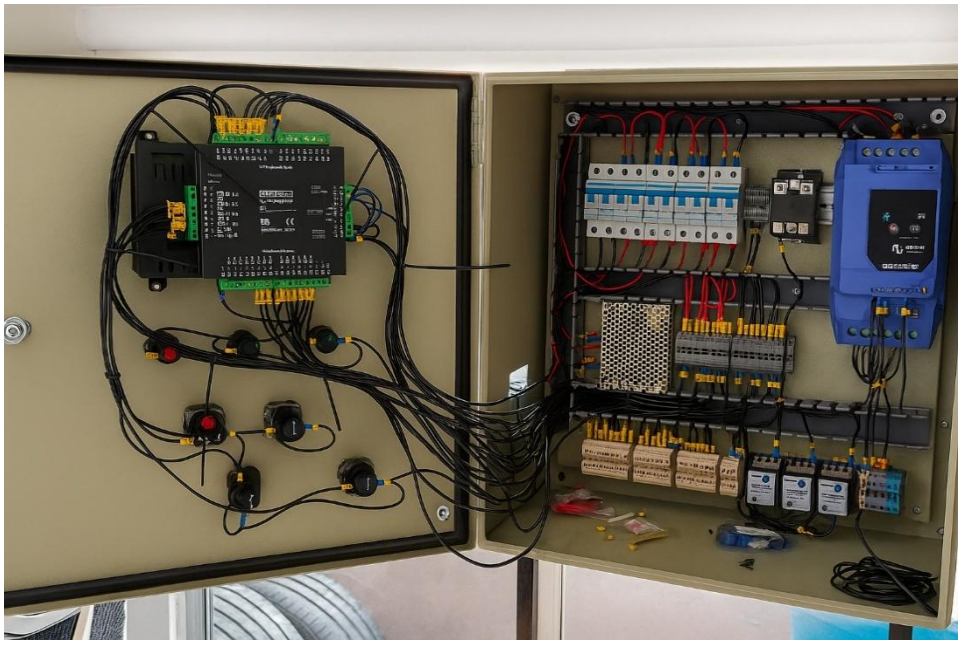


Figura 55. Cableado PLC-HMI [Fuente propia]

Para garantizar la correcta conexión de los elementos del sistema, se emplea de manera constante un multímetro en modo continuidad, verificando cada punto de conexión siguiendo las marquillas y planos previamente elaborados. Esta verificación resulta fundamental, ya que la gran cantidad de cables puede dificultar la identificación de posibles errores, incluso con la ayuda de las marquillas. Durante este proceso, es esencial asegurar que los terminales estén correctamente fijados, evitando falsos contactos que puedan generar fallas intermitentes o lecturas incorrectas. Este procedimiento no solo asegura la integridad de las conexiones, sino que también contribuye a la confiabilidad y seguridad del funcionamiento del sistema en etapas posteriores de la puesta en marcha.

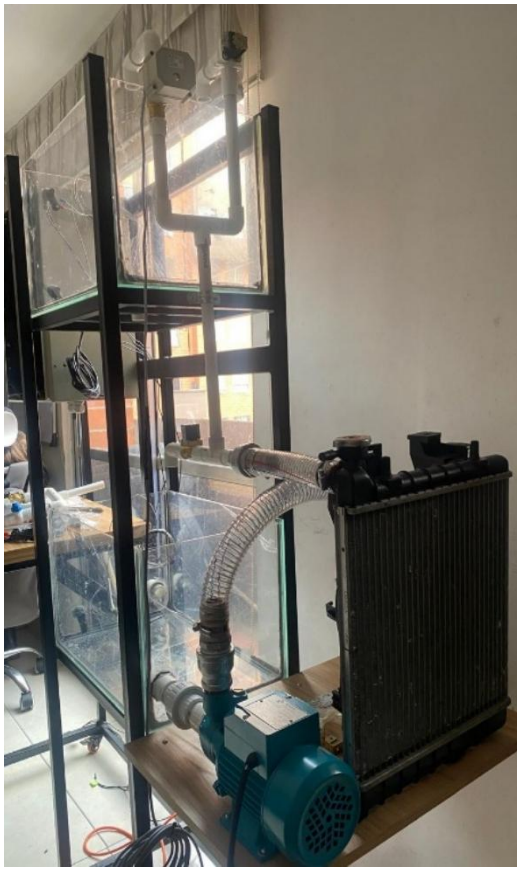


Figura 56. Montaje tanques con tablero eléctrico en estructura [Fuente propia]

Continuado con las conexiones del tablero se realiza el montaje del sistema de almacenamiento y circulación de agua. Para esta sección se realizó las aberturas en los tanques para las termocuplas, los sensores de nivel máximo y mínimo. En el caso del tanque superior se hizo la abertura de la resistencia y en el tanque inferior la abertura para la entrada de la electrobomba. Para fijar los elementos se utilizó Adhesivo de cianoacrilato y luego en la parte interior y exterior se aplicó una capa de silicona de alta temperatura con el propósito de evitar fugas de agua del sistema. De acuerdo al esquema realizado se realiza la instalación de la tubería.



Figura 57.Cableado sensores y actuadores a tablero eléctrico [Fuente propia]



Por decisión del ingeniero Edward Hernando para la salida de la electrobomba se añadió una manguera transparente flexible que se conecta a la entrada radiador. De igual forma a la salida del radiador con el sistema de circulación de agua hacia los tanques. Esta decisión se tomó por dos aspectos importantes. Uno era la ubicación de la entrada del radiador, puesto que está en la parte inferior lo cual resulta en un difícil acceso y segundo, al ser una planta con un fin educativo es importante que los estudiantes observen la circulación por ejemplo a diferentes frecuencias del variador.

Una vez realizado la instalación del sistema de circulación de agua se canalizan los cables en la parte posterior de la estructura mediante una canaleta ranurada hacia el tablero de control. Para ingresar los cables se realiza una división en la parte inferior del cofre en donde se dividen pro señales de control, instrumentación y potencia. Posteriormente se conecta a los bornes de conexión. Finalmente el montaje de la estructura está finalizada para realizar pruebas de control con todos los elementos. A continuación se presenta una imagen con la estructura completa.

Figura 58. Vista lateral sistema de circulación [Fuente propia]



Figura 59. Construcción SEDAI [Fuente propia]

7.1.6. Pruebas de funcionamiento sistema de circulacion

Estas preubas son fundamentales para determinar el funcionamiento optimo de todos los componentes que conforman el sistema. Para ello se crea un programa en cual activa todas las salidas del PLC, mediante las entradas del tablero y HMI. Para Llevar a cabo esta preuba se llena el tanque inferior con una medida considerable de agua para realizar el proceso de circulacion, posteriormente se activa el variador con la electrobomba. Esta actividad es realizada por el ingeniero Jean Carlo Leguizamon obteniedo como resultado una operación positiva de todos los componentes instalados del sistema salvo unas pequeñas fugas que se presentan y tienen que ser rapidamente corregidas.

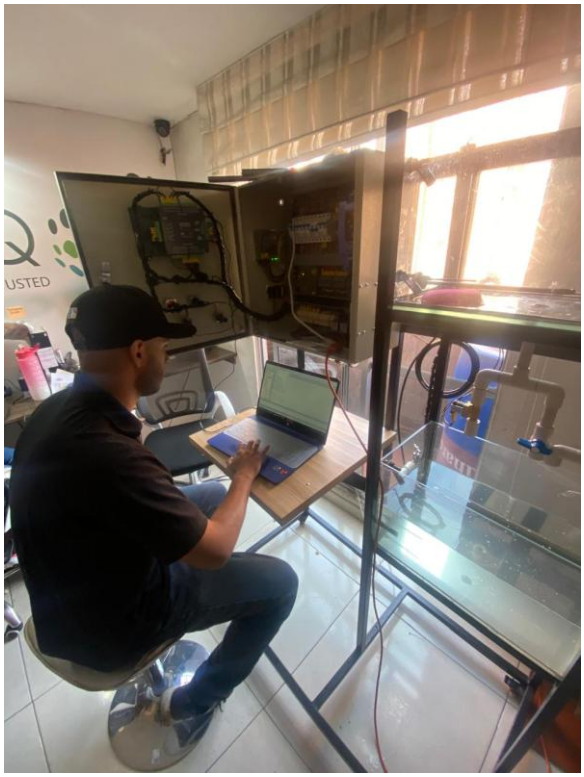


Figura 60. Ingeniero Jean Carlo Leguizamón realizando pruebas de circulación de caudal [Fuente propia]

Las preubas se realizan cambiando la frecuencia del variador progresivamtne desde el HMI y observando el comportamiento del sistema En primera instancia, se efectúa la recirculación en el tanque inferior mediante la activación del solenoide correspondiente. Posteriormente, para el tanque superior, se activa el segundo solenoide y se regula de forma gradual la apertura de la válvula proporcional. A medida que se incrementa la frecuencia de la electrobomba, se genera una mayor presión en el sistema, lo que provoca fugas en el radiador. Consecuentemente, este aumento de presión también intensifica aumenta la descarga de agua en el tanque superior. La fuga se soluciona sellando mejor el radiador y en los tanque se hace un refuerzo de silicona de alta temperatura

7.1.7. Placa de comunicación ESP32

Hasta el momento las pruebas simples de comunicación entre la ESP32 y el PLC se habían realizado con el montaje de modem nulo con protocolo de enlace de bucle hacia atrás de Lammert Bies en protoboard. El funcionamiento era correcto pero para la envergadura del proyecto era necesario fabricar un circuito que fuera capaz de contener los elementos de una forma fiable. Es por ello que una vez establecida la prueba de Lammert se diseña la PCB con el circuito siguiendo la conexiones que a continuación se describen :

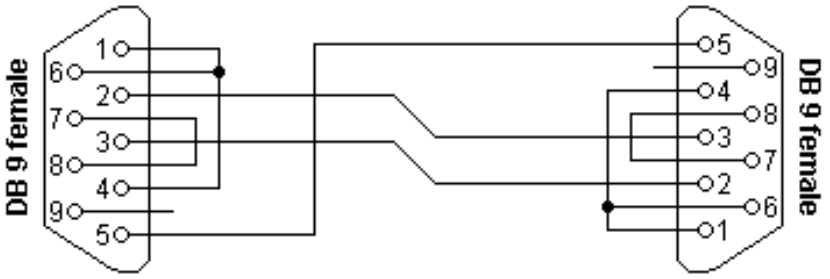


Figura 61. Conexión modem nulo con protocolo de enlace de bucle hacia atrás de Lammert Bies [41]

Conector 1	Conector 2	Función
2	3	Rx ◄ Tx
3	2	Tx ► Rx
5	5	Tierra de señal
1 + 4 + 6	–	DTR ► CD + DSR
–	1 + 4 + 6	DTR ► CD + DSR
7 + 8	–	RTS ► CTS
–	7 + 8	RTS ► CTS

Tabla 3. Conexión de modem nulo, adaptado de [41].

Los conectores DB9 enlazan la ESP32 y el PLC mediante los conversores de cada fabricante. Las conexiones propuestas de Lammert solucionan el problema de comunicación ModBus por lo que para realizar el diseño de la PCB se cerciora que cada pin tenga una conexión solida entre cada conector. Para realizar el diseño de la PCB se realiza el esquema en Kicad, un software que permite importar elementos para la creación de la placa de circuitos. permitiendo en desarrollos posteriores la incorporación de nuevos sensores y el aumento de la funcionalidad del proyecto. A continuación se presenta los esquemas realizados y la placa con los componentes soldados

- Circuito trazado de placas:

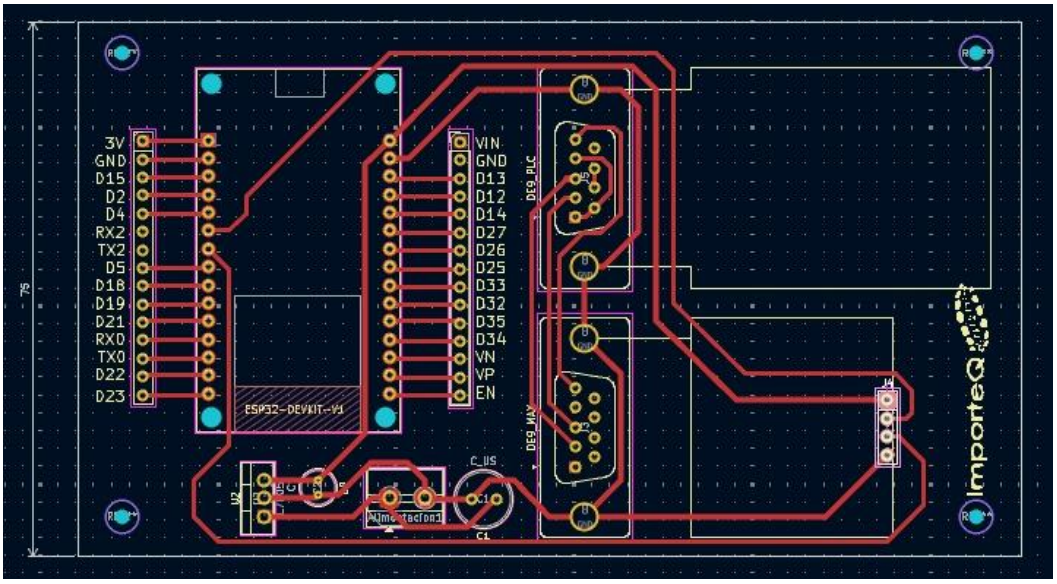


Figura 62. Circuito de trazado de placas realizado en Kicad [Fuente propia]

- Circuito en 3D:

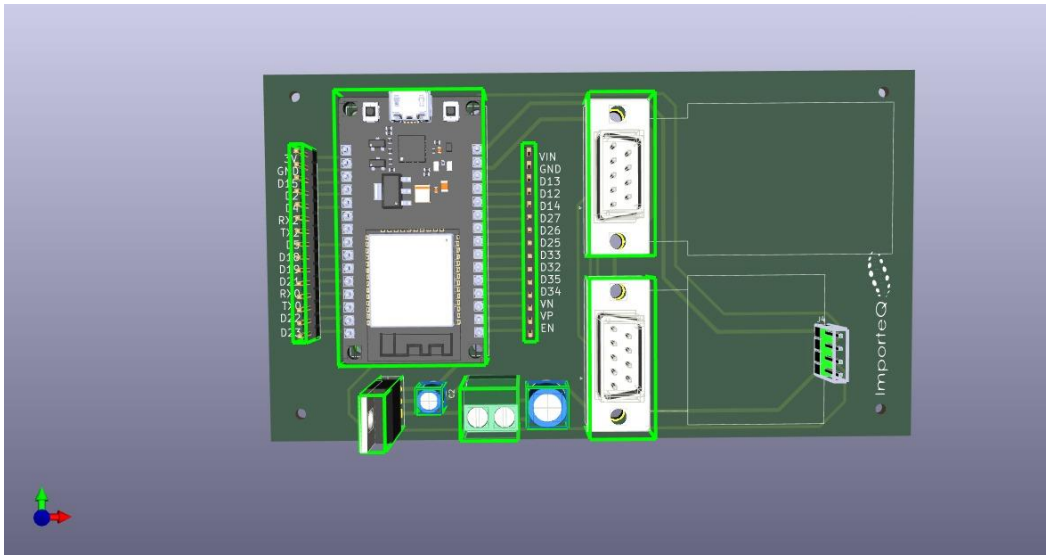


Figura 63. Esquema final tarjeta de comunicación Modbus-MQTT [Fuente propia]

- Placa de circuito soldado

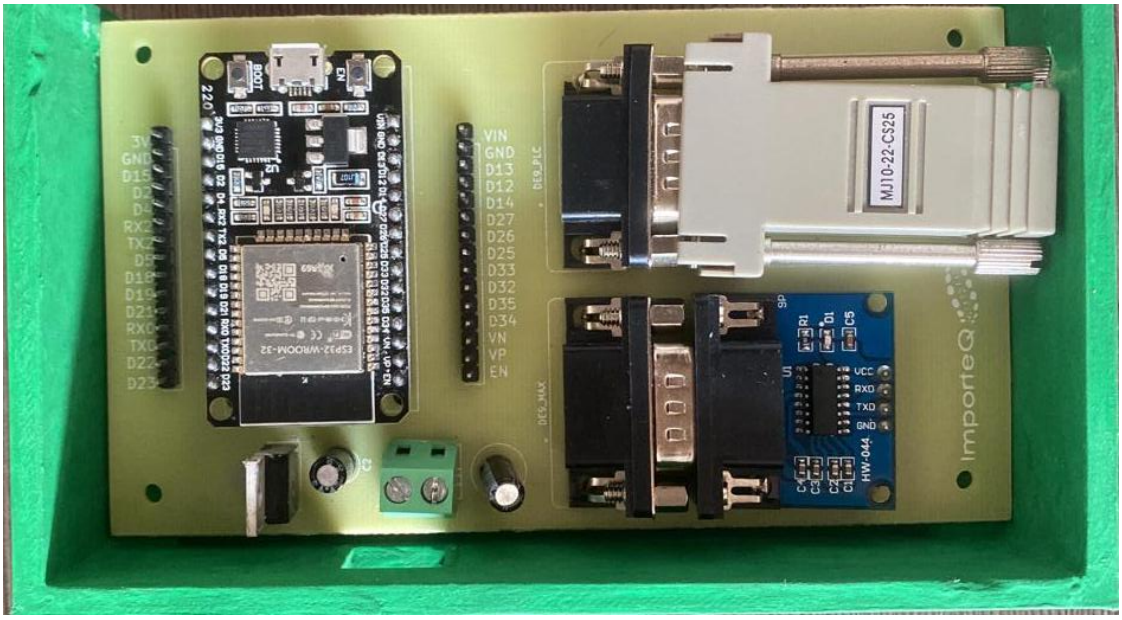


Figura 64. Circuito de comunicación soldado con conversos MAX3232 y MJ10-22-CS25 [Fuente propia]

7.1.8. Contenedor circuito de comunicación

Para almacenar el módulo de comunicación se realizó un contenedor con las dimensiones del circuito, esto con el fin de anclar el módulo al tablero de control y proteger el circuito de posibles cortos circuitos. El diseño de este contenedor se realizó en Tinkercad en dos partes, primero la caja y luego la tapa. Ambos elementos se imprimieron en PLA blanco.

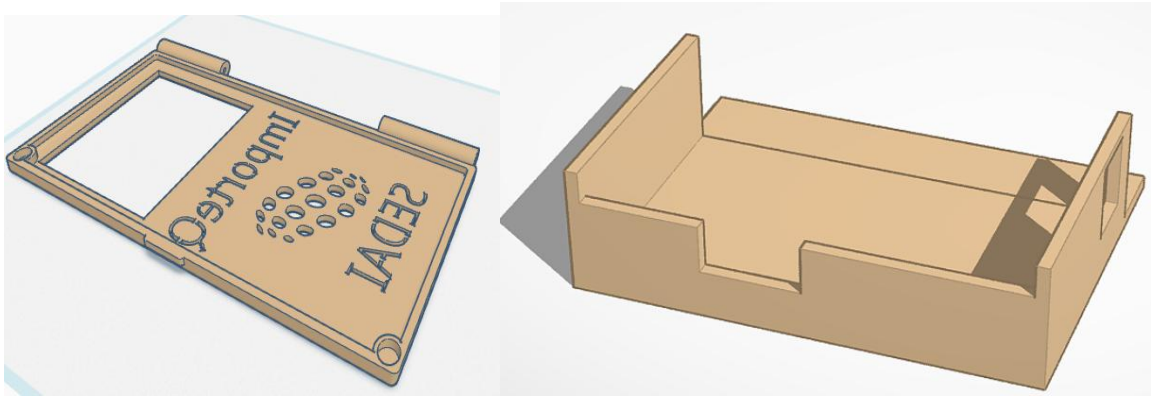


Figura 65. Diseño contenedor circuito en Tinkercad [Fuente propia]



Figura 66. Instalación módulo de comunicación en tablero eléctrico [Fuente propia]

7.1.9. Control On-Off de nivel

Antes de realizar el control de nivel fue necesario realizar una linealización del sensor, para ello y con una regla como patrón de medida se comienza a aumentar de nivel del agua en el tanque superior y para luego observar la salida en bits del sensor. El TL-136 funciona mediante presión hidrostática por lo que al aumentar el nivel el valor en bits aumenta. Estos datos se fueron tabulando en Excel para luego realizar un regresión lineal y así obtener la parametrización del sensor. Se observo que los datos de sensor variaban y no se mantenían fijo en una medida, por lo que este problema de estabilidad represento un problema para obtener una linealización precisa. Para Solucionar este inconveniente esta prueba se realizó varias veces tomando los datos de forma reiterativa y haciendo un promedio.

mm	Valor en bits
40	975
100	1088
150	1182
200	1276
250	1370

Tabla 4. Datos de linealización [Fuente propia].

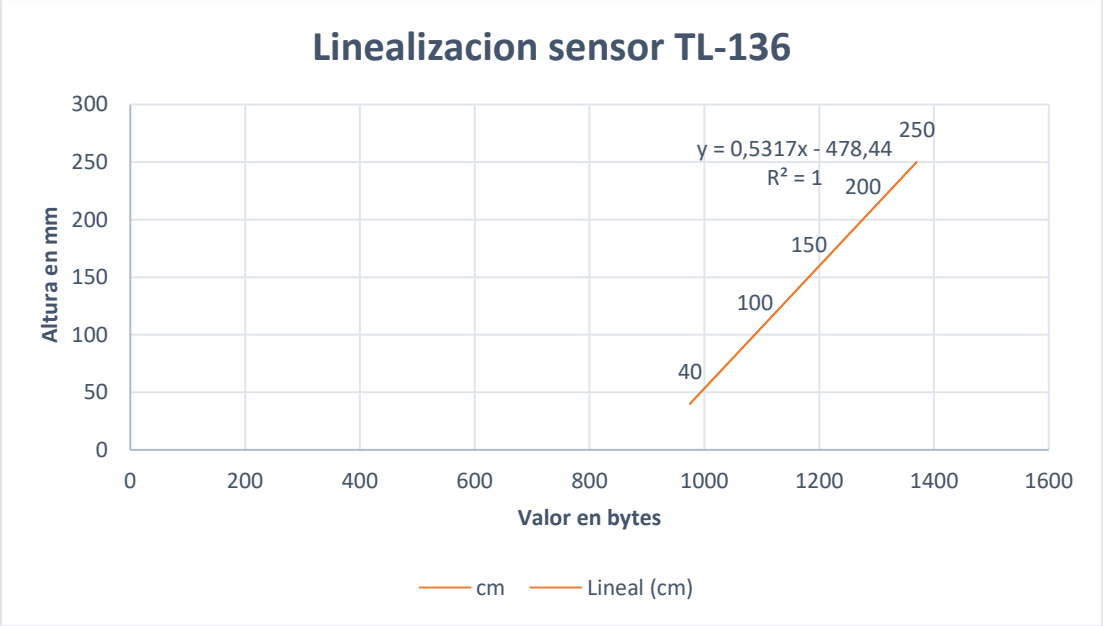


Figura 67. Grafica linealización sensor TL-136 [Fuente propia]

Este proceso de linealización es importante, ya que esta fórmula también será utilizada en el control PID. Este paso previo antes de realizar la prueba de control no soluciona el problema real del sensor: El problema de precisión y estabilidad. Para minimizar el ruido del sensor Unitronics ofrece dos soluciones que se aplicaron para este sensor. Como primera medida se aplica un filtro de alta resistencia a la entrada análoga del sensor TL-136

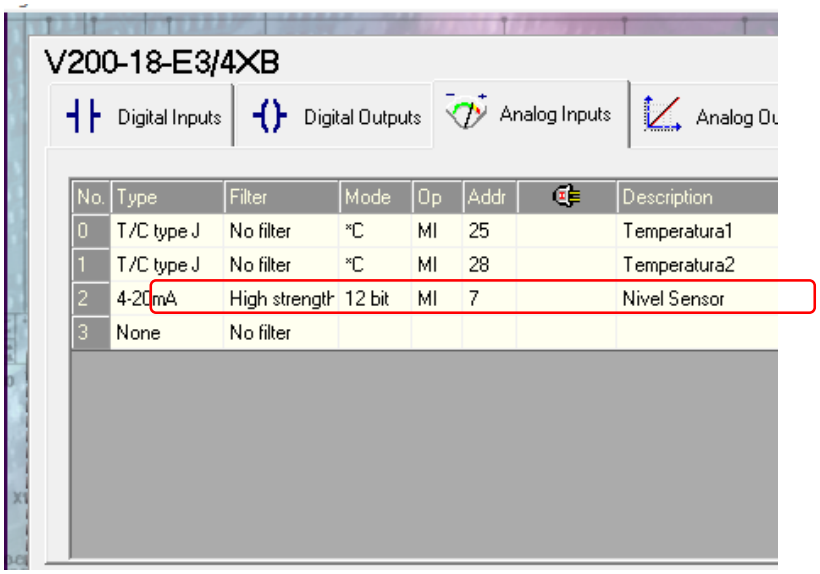


Figura 68. Configuración entrada análoga [Fuente propia]

Aunque esta configuración estabiliza los valores de los bits del sensor aun no es lo suficiente para aplicar ningún tipo de control, esto debido a que dada la linealización los valores en cm

variaban entre 2cm - 2.5cm por encima y por debajo del valor de medida original. Como segundo filtro digital se aplicó en programación mediante el bloque *"filter"*. La función de este bloque es almacenar una x cantidad de datos en un vector y luego los promedia para obtener un único valor, luego de realizar esta operación, almacena nuevos datos y repite el proceso. Con este bloque se asegura una mayor estabilidad de la señal y con las pruebas realizadas la variación es de de tan solo 1.5 mm – 2 mm por encima y por debajo del punto de referencia

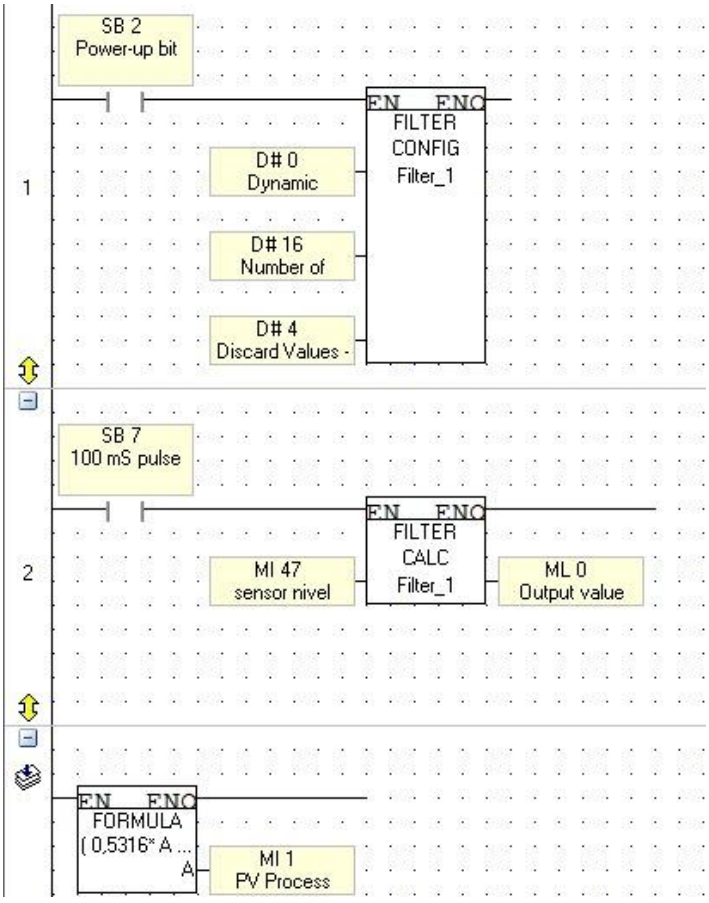


Figura 69. Filtro digital sensor nivel [Fuente propia]

Una vez finalizada la configuración del sensor para la prueba de control de nivel, se procede a realizar múltiples ensayos utilizando un patrón de medida, evaluando diferentes puntos de referencia . Como salida de control se toma el variador de frecuencia, quien regula la salida de la bomba, parando o activando el flujo del caudal hacia el tanque superior. Al ser un control On-Off la frecuencia del variador se deja estable en 40 Hz. A continuación se presentan las gráficas obtenidas de VisiLogic.

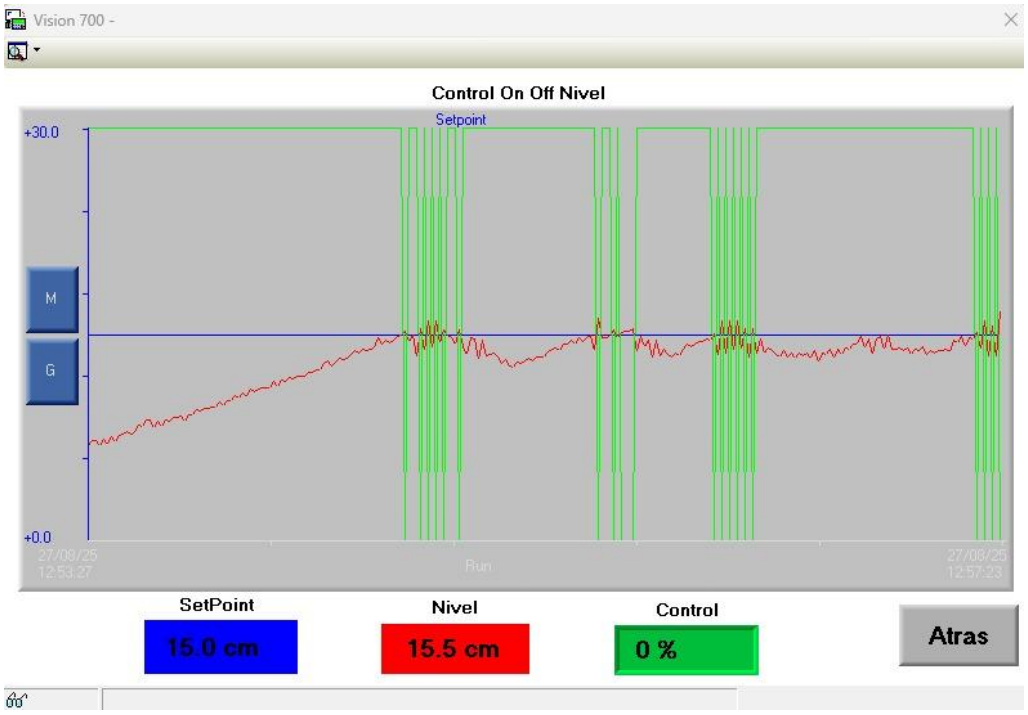


Figura 60. Grafica control nivel On-Off -1 [Fuente propia]

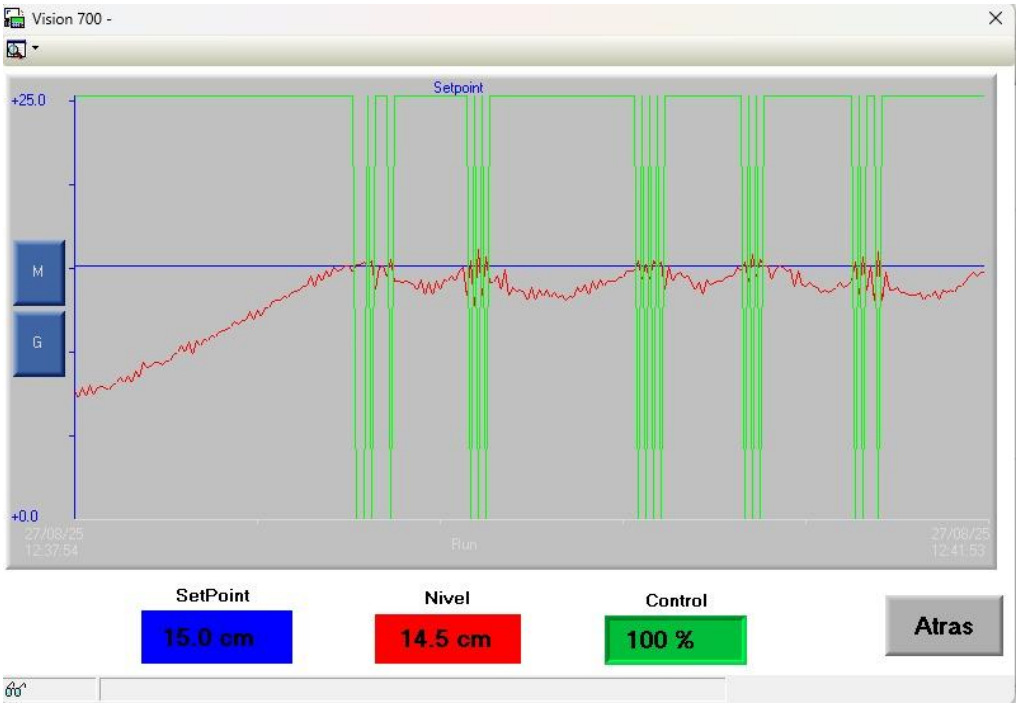


Figura 61. Grafica control nivel On-Off -2 [Fuente propia]

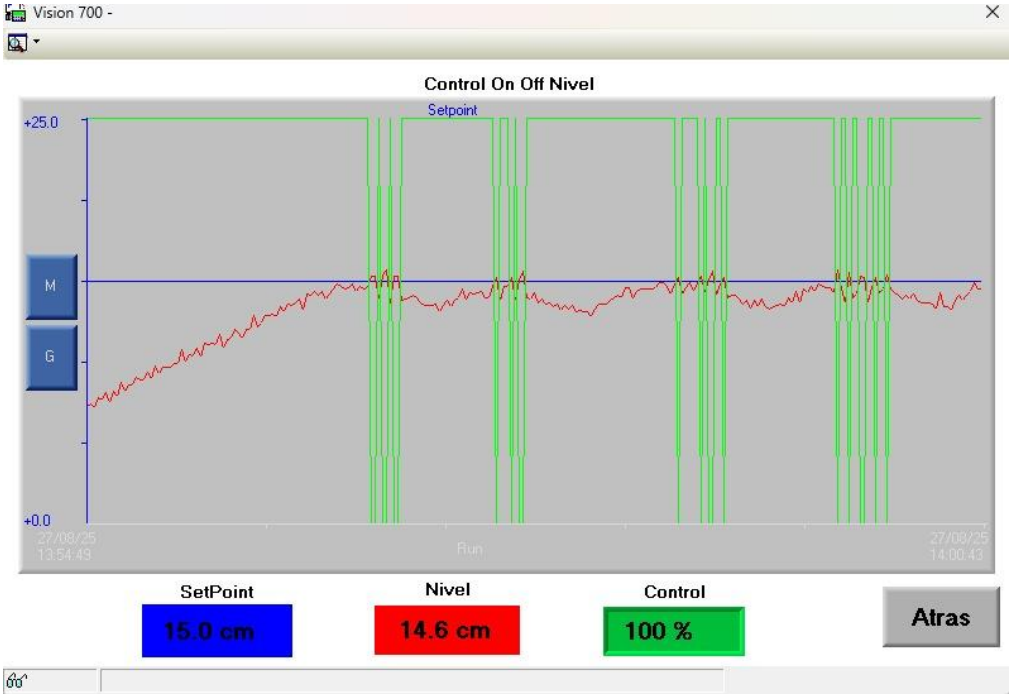


Figura 62. Grafica control nivela On-Off -3 [Fuente propia]

La señal de nivel presenta oscilaciones alrededor del setpoint con presencia de ruido de alta frecuencia, el cual se intensifica durante la conmutación del actuador, generando un efecto de *chattering*. Estas perturbaciones pueden deberse a turbulencias en el proceso, interferencias eléctricas o limitaciones del sensor. Aunque el control ON-OFF mantiene el nivel cercano a la referencia, la alta frecuencia de conmutación reduce la estabilidad y puede desgastar los actuadores. Aunque el nivel se mantiene alrededor del punto de referencia, el resultado de estas pruebas evidencia que la lectura del sensor a pesar de doble filtro no garantiza un control efectivo y estable. Estas oscilaciones de los datos sugieren que para aplicar control de nivel con el sensor TL-136 se necesita un filtrado de mayor rigurosidad

7.1.10. Control PID de nivel

Para realiza las pruebas de PID se toman los mismo puntos de referencia y se configura el bloque
Autotuning PID de Unitronics que permite ajustar automáticamente los parámetros del controlador PID para un lazo de control específico. Su funcionamiento consiste en aplicar variaciones de prueba en la señal de salida para generar oscilaciones en la variable de proceso, medir la respuesta del sistema (tiempo, amplitud y retardo) y, con base en esos datos, calcular los valores óptimos de Kp, Ti y Td mediante algoritmos como el de Ziegler–Nichols. Una vez finalizado, el autotuning almacena los parámetros obtenidos en el bloque PID, dejando el lazo listo para operar en modo automático con una configuración adecuada.

El bloque PID de VisiLogic tiene los siguientes parámetros:

- **Set Point**

Valor objetivo que se quiere alcanzar (ejemplo: 150 = 15.0 cm de nivel si está escalado a décimas).

- **PV (Process Value)**

Variable de proceso medida en tiempo real (ejemplo: el nivel actual en cm o la temperatura en °C).

- **Proportional Band**

Ancho de banda proporcional en décimas de %. Un valor más bajo implica una mayor ganancia proporcional (**respuesta más rápida, pero mayor riesgo de oscilación**).

- **Integral** Time **(Ti)**

Tiempo en segundos que tarda la acción integral en corregir el error acumulado. **Reduce error estacionario**, pero si es muy bajo puede provocar inestabilidad.

- **Derivative Time (Td)**

Tiempo en segundos para la acción derivativa. Ayuda a **anticipar cambios y suavizar oscilaciones**, aunque un valor alto puede amplificar el ruido.

- **Sample Time**

Intervalo de muestreo del lazo PID (en unidades de 10 ms). Define cada cuánto se actualiza el cálculo del PID.

- **Action (0 = Reverse, 1 = Direct)**

  - **Reverse (0):** al aumentar la salida, la PV disminuye (ejemplo: válvula de refrigeración).
  - **Direct (1):** al aumentar la salida, la PV también aumenta (ejemplo: calefactor o bomba).

- **Range Low Limit**

Valor mínimo esperado de la variable de proceso (PV). Permite escalar la señal.

- **Range High Limit**

Valor máximo esperado de la PV.

- **Output Range Low Limit**

Límite inferior de la salida del controlador (ejemplo: 0%).

- **Output Range High Limit**

Límite superior de la salida (ejemplo: 100%).

Pruebas de control PID nivel:

- Primera prueba

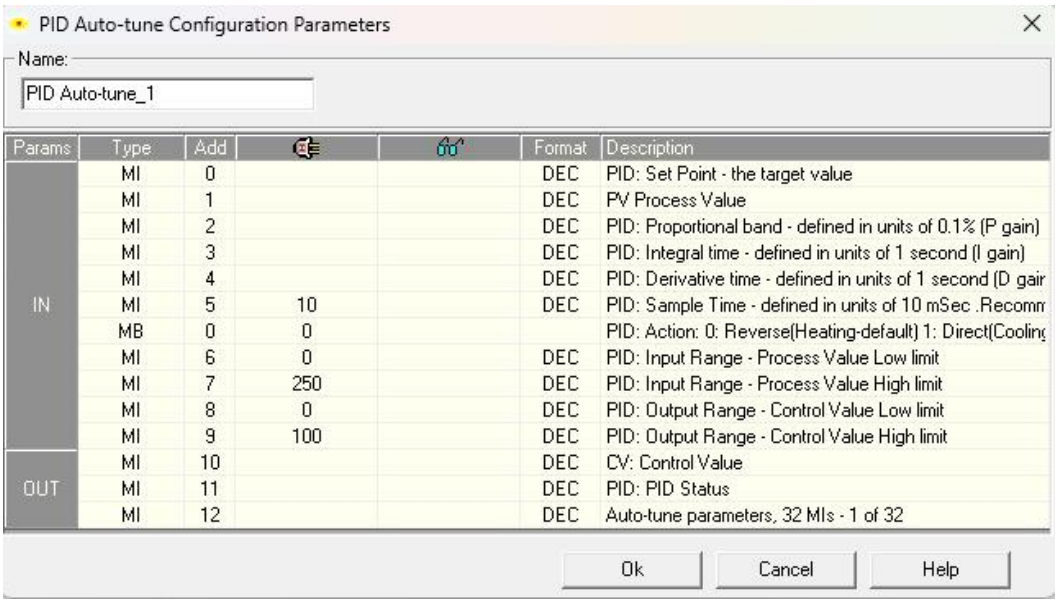


Figura 63. Configuración bloque PID control nivel [Fuente propia]

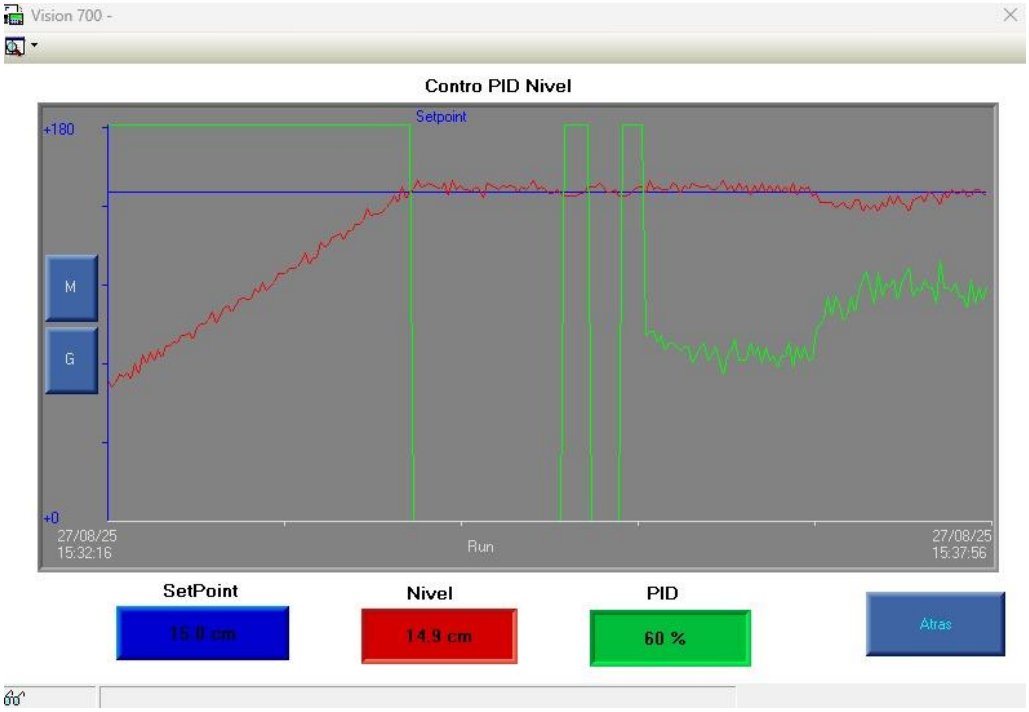


Figura 64. Control PID, primera prueba nivel-1 [Fuente propia]

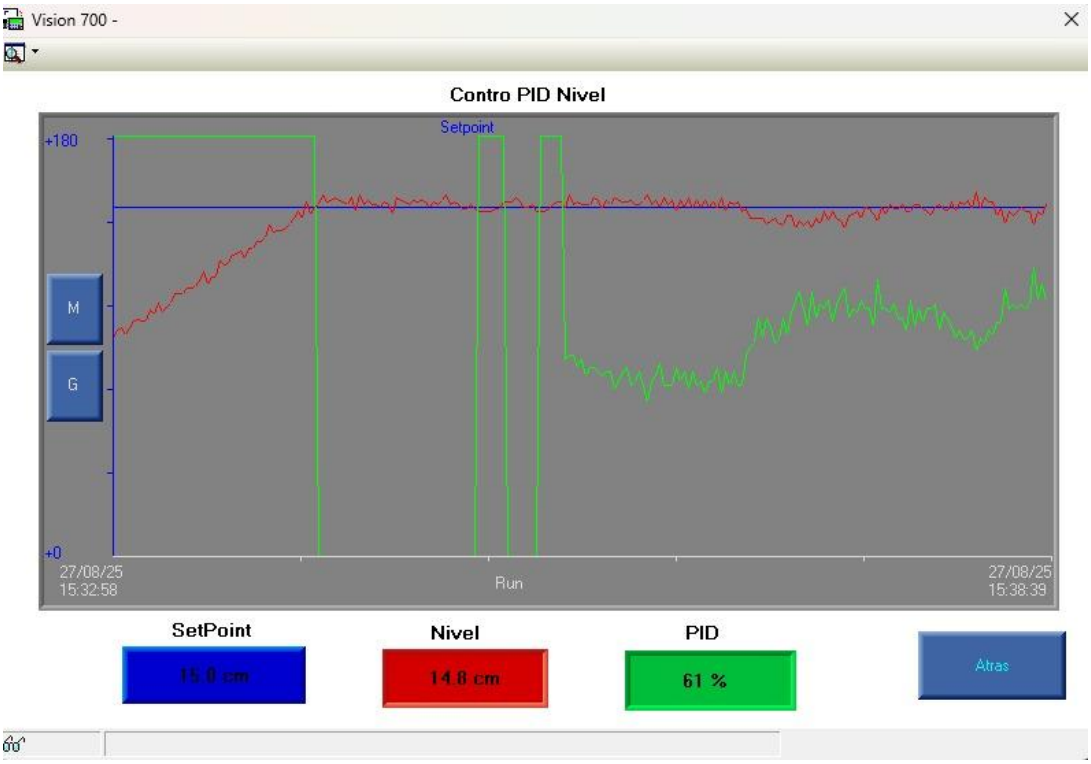


Figura 65. Control PID, primera prueba nivel-2 [Fuente propia]

- Segunda prueba

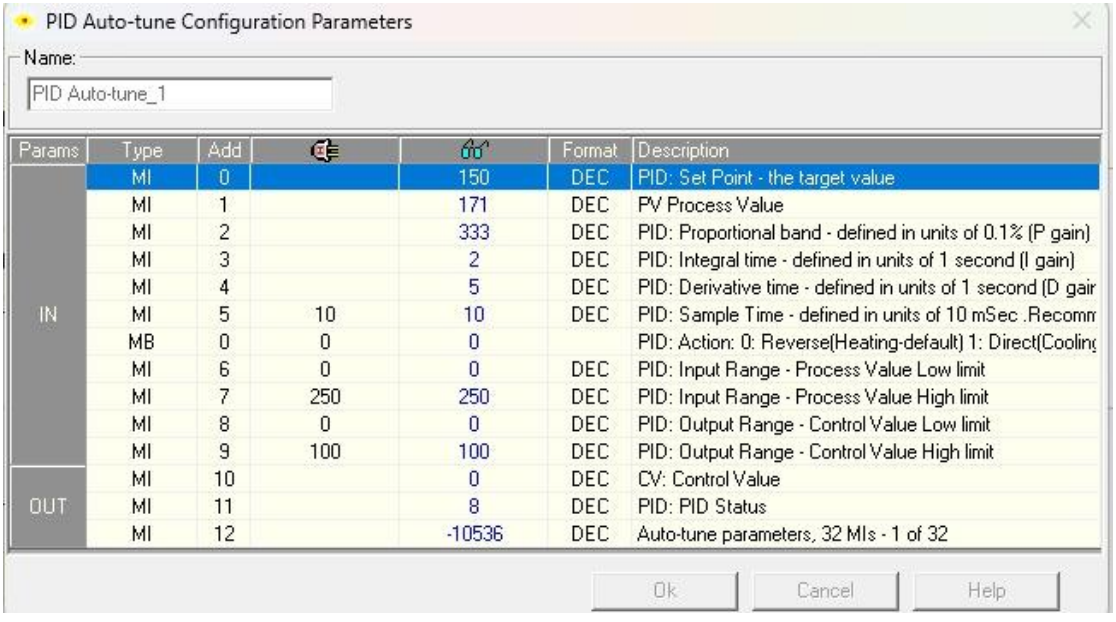


Figura 66. Configuración bloque PID, segunda prueba [Fuente propia]



Figura 67. Control PID nivel, segunda prueba-1 [Fuente propia]

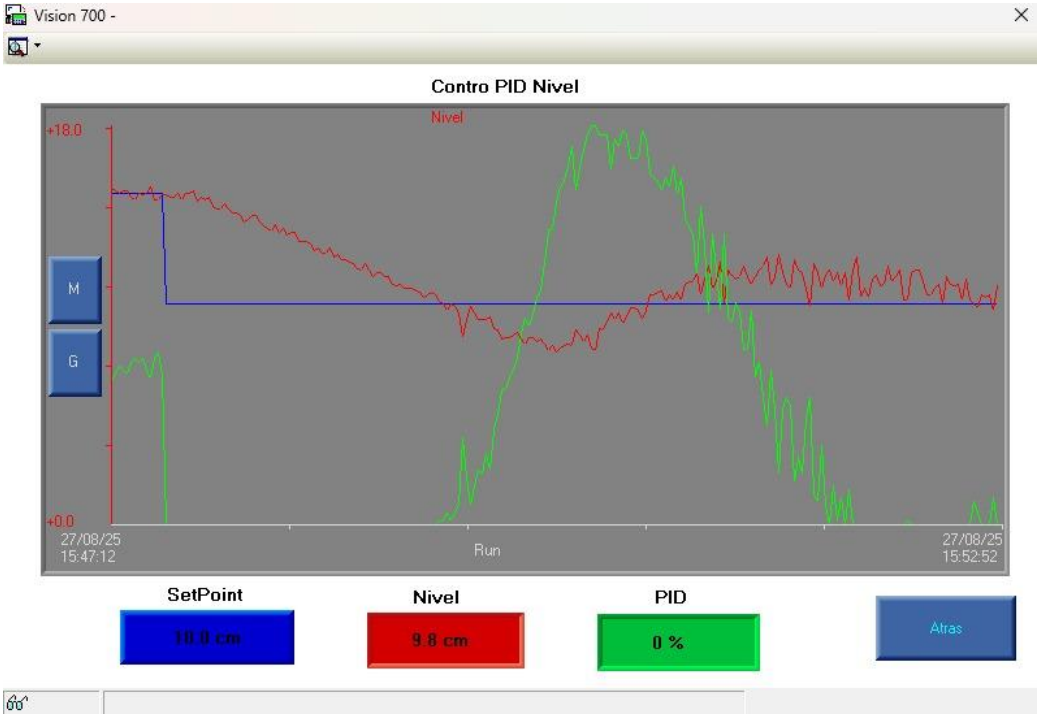


Figura 68. Control PID nivel, segunda prueba-2 [Fuente propia]

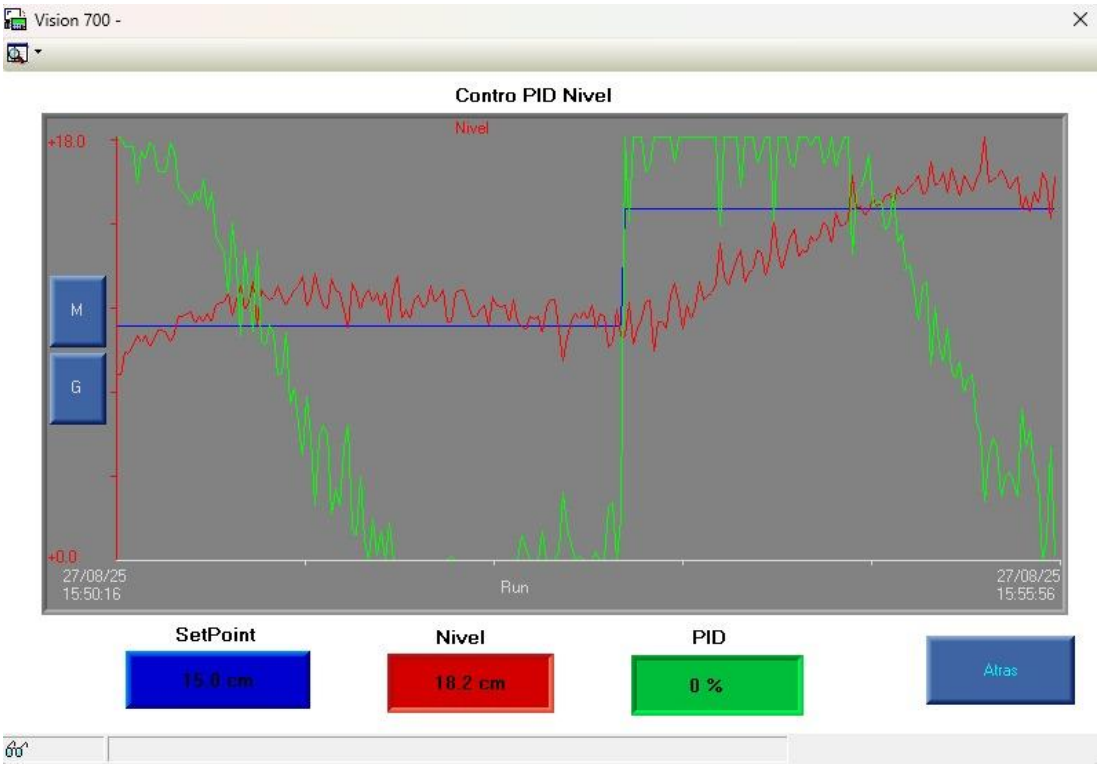


Figura 69. Control PID nivel, segunda prueba-3 [Fuente propia]

- Tercera prueba

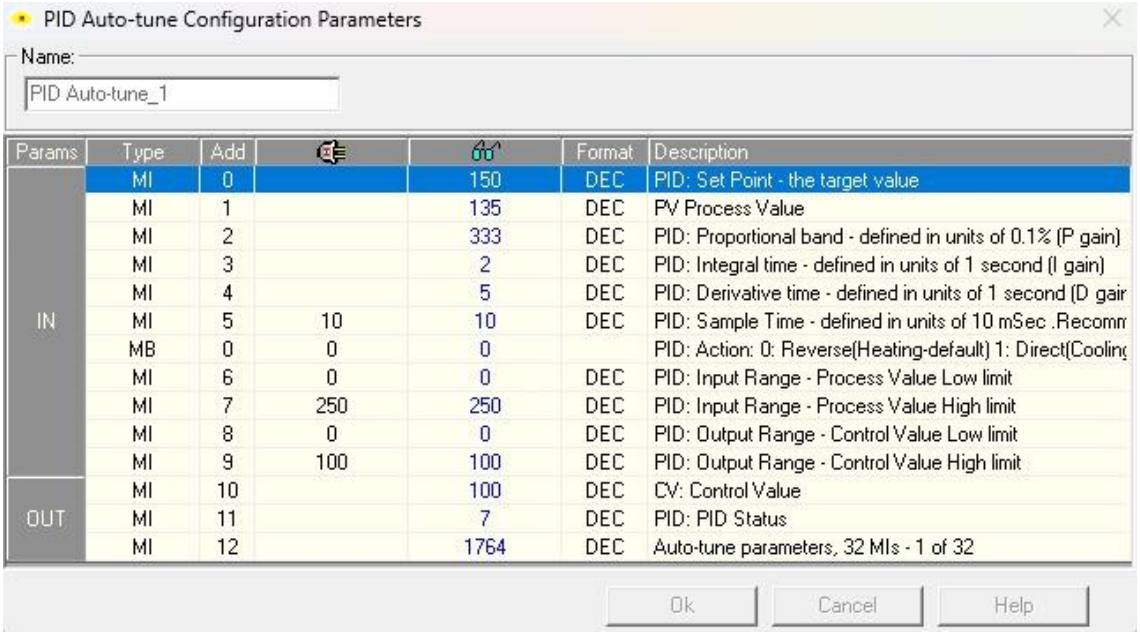


Figura 70. Grafica linealización sensor TL-136 [Fuente propia]

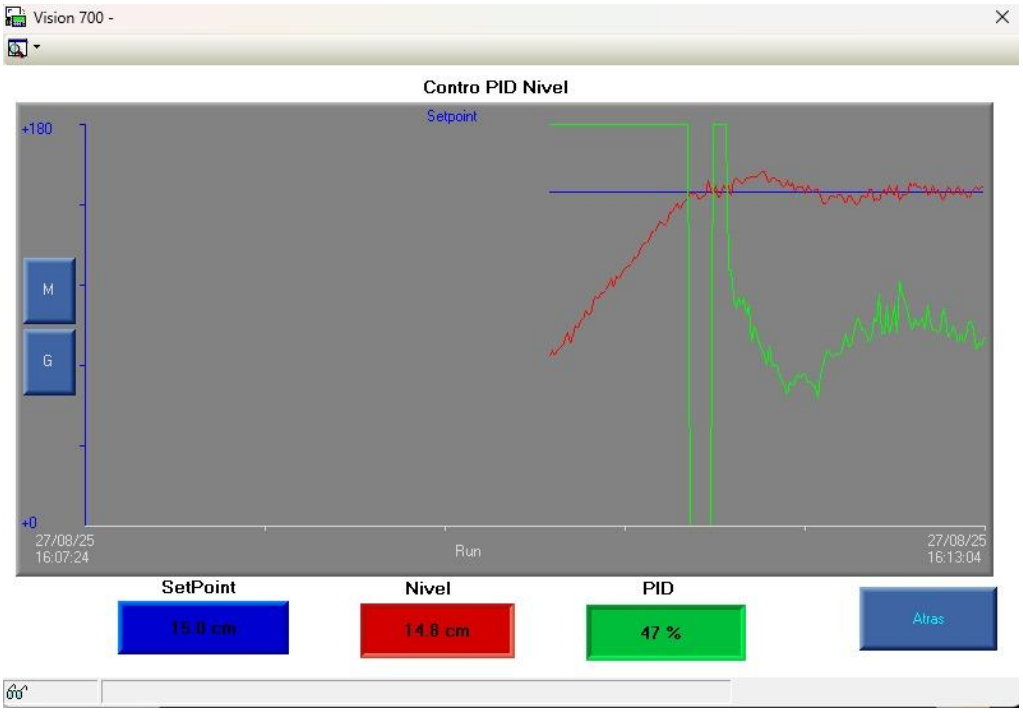


Figura 71. Control PID nivel, tercera prueba-1 [Fuente propia]



Figura 72. Control PID nivel, tercera prueba-2 [Fuente propia]

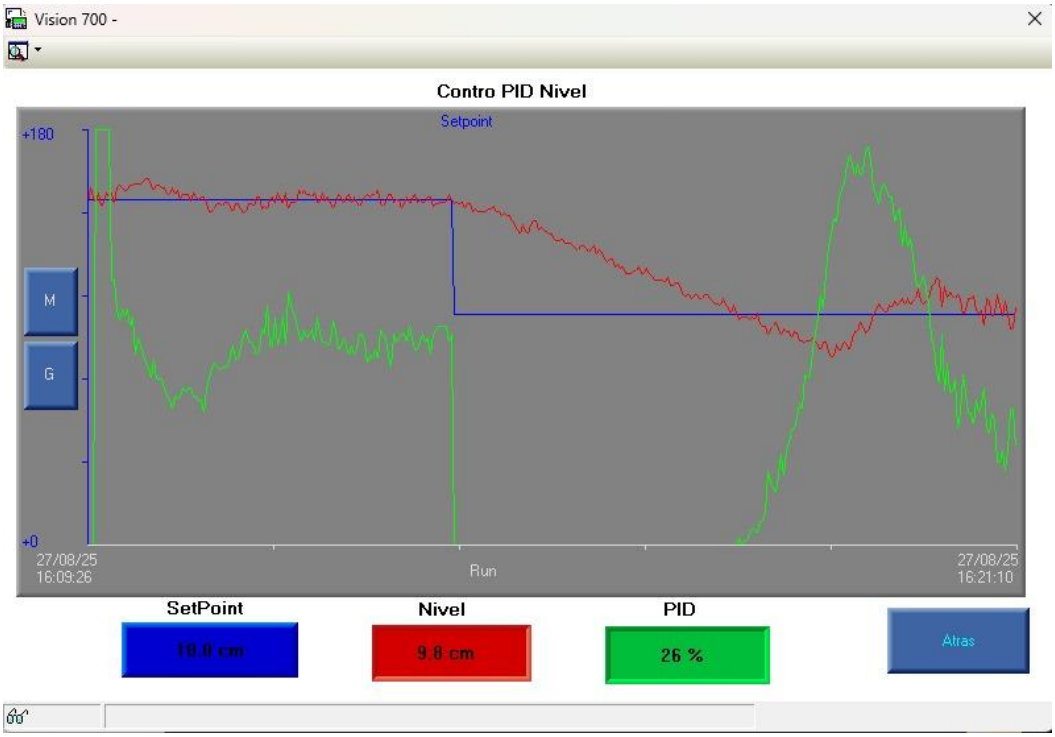


Figura 73. Control PID nivel, tercera prueba-3 [Fuente propia]

7.1.11. Control On-Off de temperatura

La linealización de la termocupla resultó ser la más sencilla. Tal como se explicó en la actividad 6.2.13, únicamente es necesario conectar la termocupla a la entrada analógica del módulo y, en VisiLogic, configurar el tipo de termocupla junto con la unidad de medida (grados Fahrenheit o Celsius). En VisiLogic, el valor de salida se presenta como un número entero debido a que no trabaja con números reales. Para interpretar la medición real, los dígitos del valor entero se leen de la siguiente manera: el último dígito representa la parte decimal, y los dígitos precedentes representan la parte entera. Por ejemplo si se obtiene una lectura de temperatura de 356, este dato se deberá interpretar como 35.6º (grados centígrados para este ejemplo).

Para ver y trabajar estos datos es importante separarlo para interpretarlo correctamente, matemáticamente se puede realizar mediante una división de la siguiente forma:

$$T = \frac{Dato}{10} \tag{1}$$

Para visualizarlo en el HMI se realiza asignado un punto decimal al valor recibido, para que así de esta forma se visualice de la forma correcta. Esta acción se realiza mediante la configuración del campo de texto asociado a la marca de memoria asignada al sensor de temperatura en el módulo de entradas y salidas

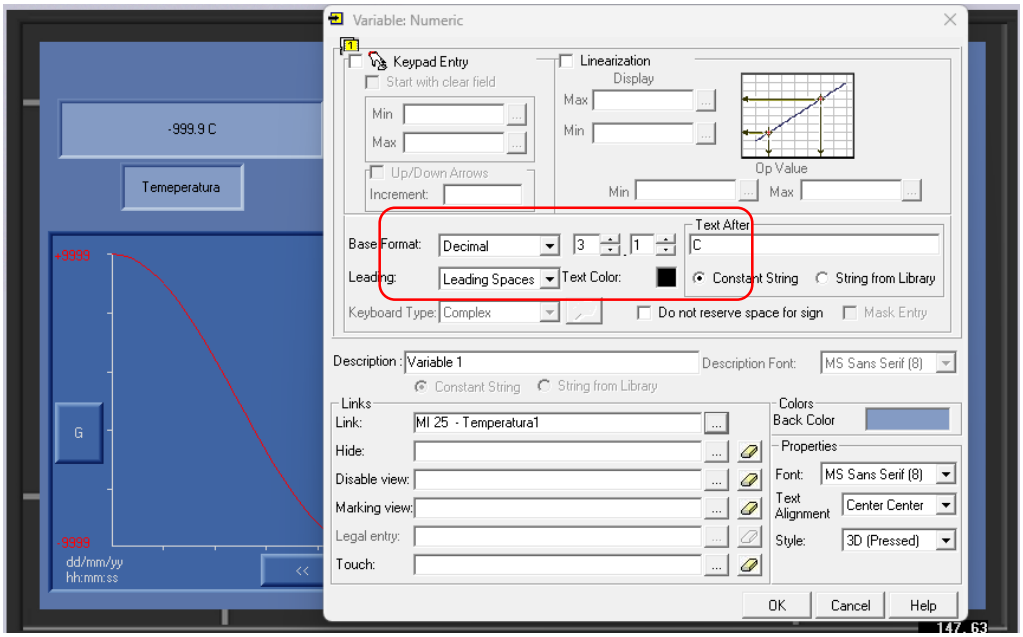


Figura 74. Configuración punto decimal en campo de texto [Fuente propia]

Para realizar la prueba de control de temperatura se suministra al tanque 10 litros de agua, esto con el fin de cubrir la resistencia en su totalidad para las practicas, no suministrar la suficiente agua para esta prueba puede causar daños irreparables a la resistencia eléctrica . A continuación se presenta la gráfica de control On-Off

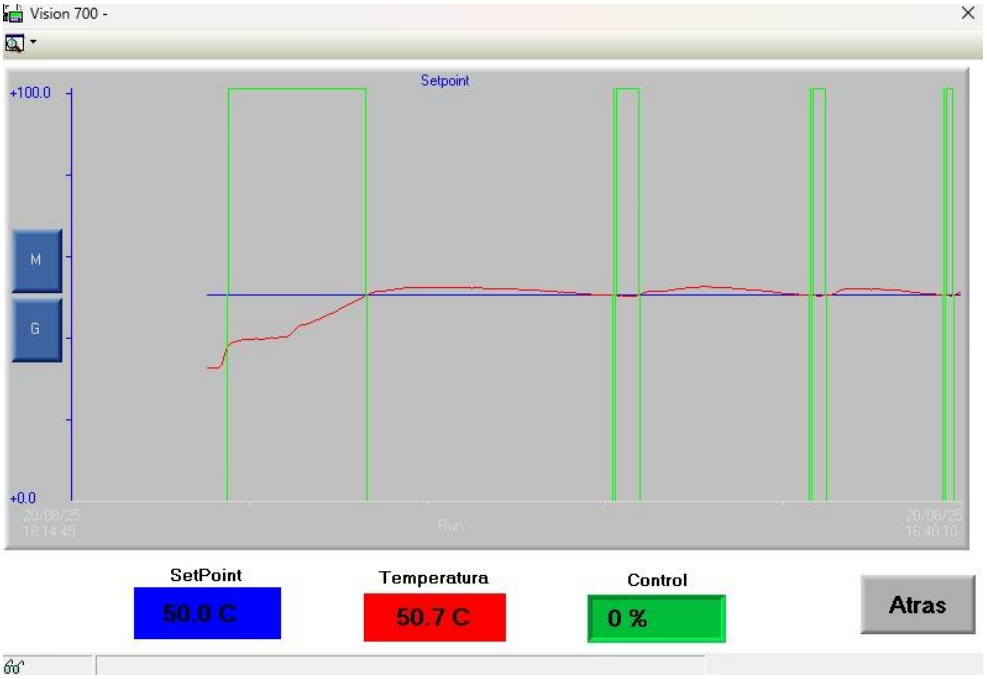


Figura 75. Control On-Off temperatura-1 [Fuente propia]

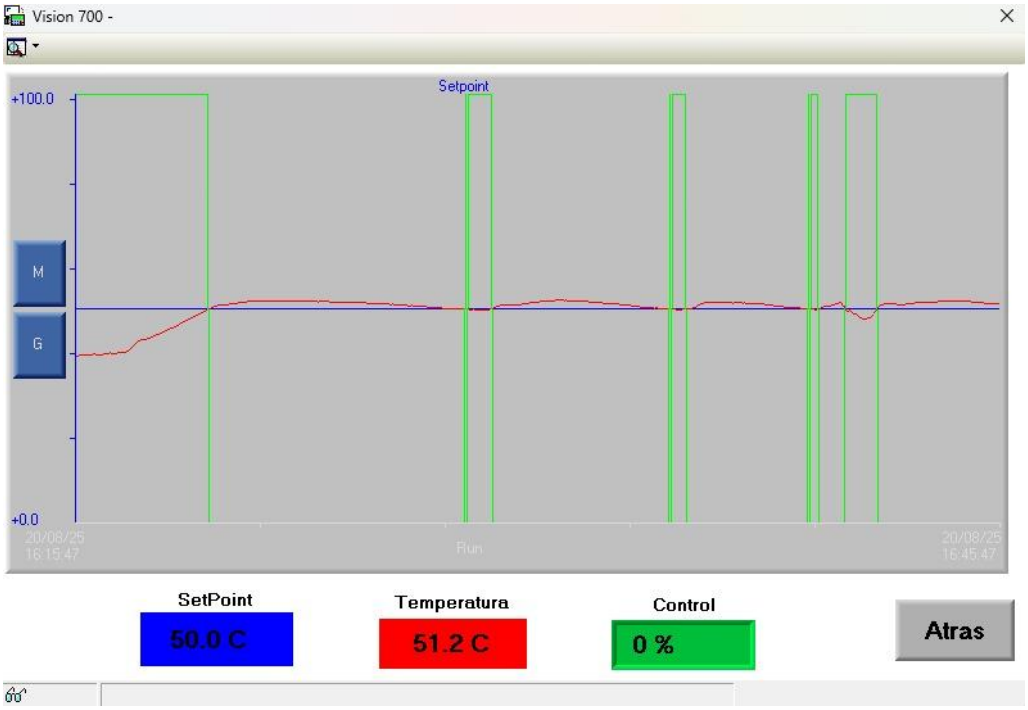
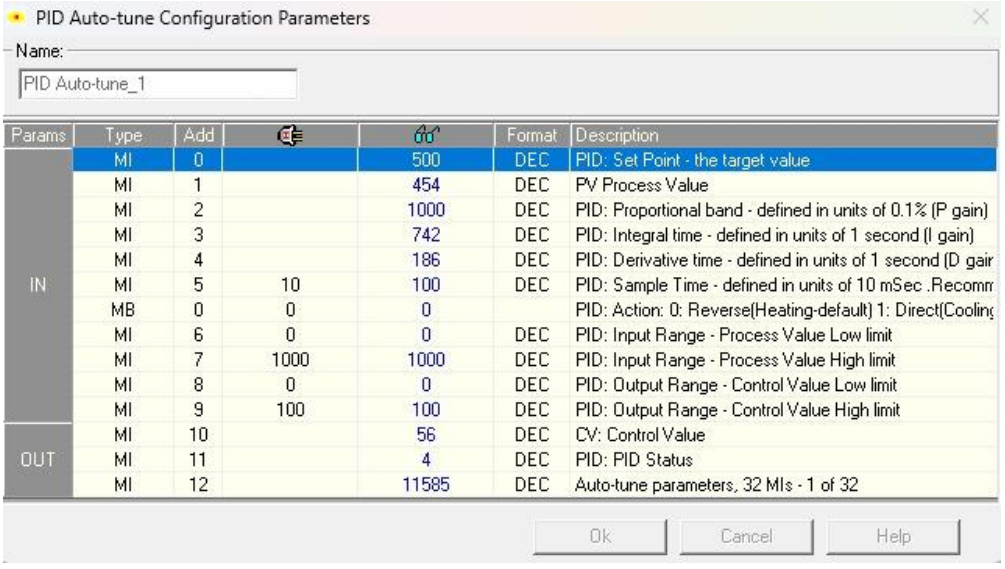


Figura 76. Control On-Off temperature-2 [Fuente propia]

7.1.12. Control PID de temperatura

Para realizar el control PID de temperatura se configura el mismo bloque que se utilizó para el control PID con los siguientes para metros:



Params	Type	Add			Format	Description
	MI	0		500	DEC	PID: Set Point - the target value
	MI	1		454	DEC	PV Process Value
	MI	2		1000	DEC	PID: Proportional band - defined in units of 0.1% (P gain)
	MI	3		742	DEC	PID: Integral time - defined in units of 1 second (I gain)
	MI	4		186	DEC	PID: Derivative time - defined in units of 1 second (D gair
IN	MI	5	10	100	DEC	PID: Sample Time - defined in units of 10 mSec .Recomm
	MB	0	0	0		PID: Action: 0: Reverse(Heating-default) 1: Direct(Cooling
	MI	6	0	0	DEC	PID: Input Range - Process Value Low limit
	MI	7	1000	1000	DEC	PID: Input Range - Process Value High limit
	MI	8	0	0	DEC	PID: Output Range - Control Value Low limit
	MI	9	100	100	DEC	PID: Output Range - Control Value High limit
	MI	10		56	DEC	CV: Control Value
OUT	MI	11		4	DEC	PID: PID Status
	MI	12		11585	DEC	Auto-tune parameters, 32 MIs - 1 of 32

Figura 77. Configuración bloque PID, control temperatura [Fuente propia]

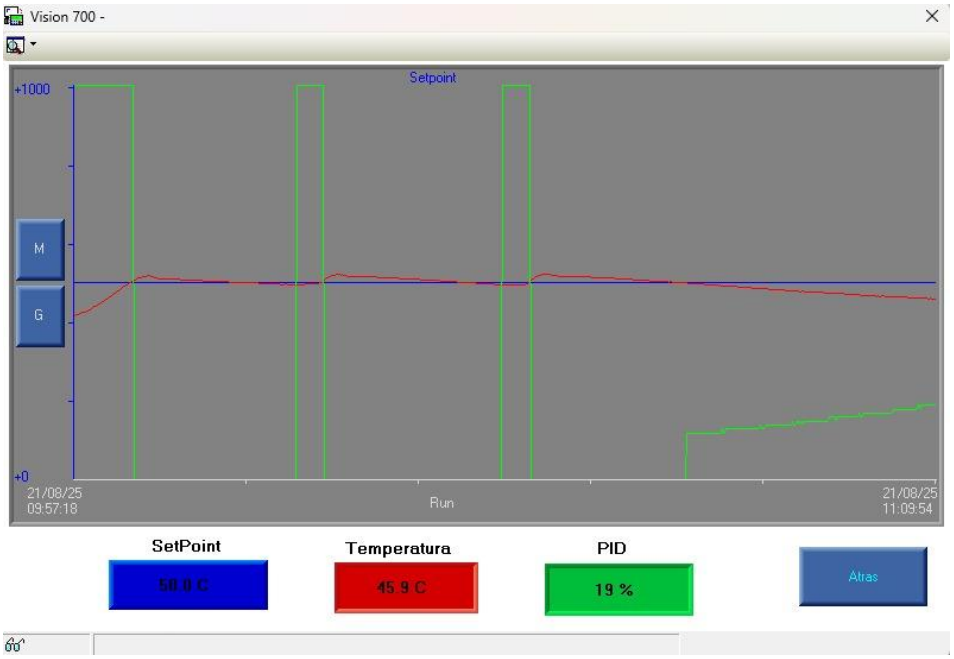


Figura 78. Control PID temperatura-1 [Fuente propia]

Como se muestra en la figura 95, el bloque PID aplica la señal de control en tres ocasiones antes de determinar las constantes de ajuste del proceso. Dado que esta señal presenta mucho menos ruido en comparación con la de nivel, el comportamiento del fenómeno se aprecia con mayor claridad. Una vez calculadas las constantes, el bloque logra mantener la temperatura estable frente a las perturbaciones presentes.

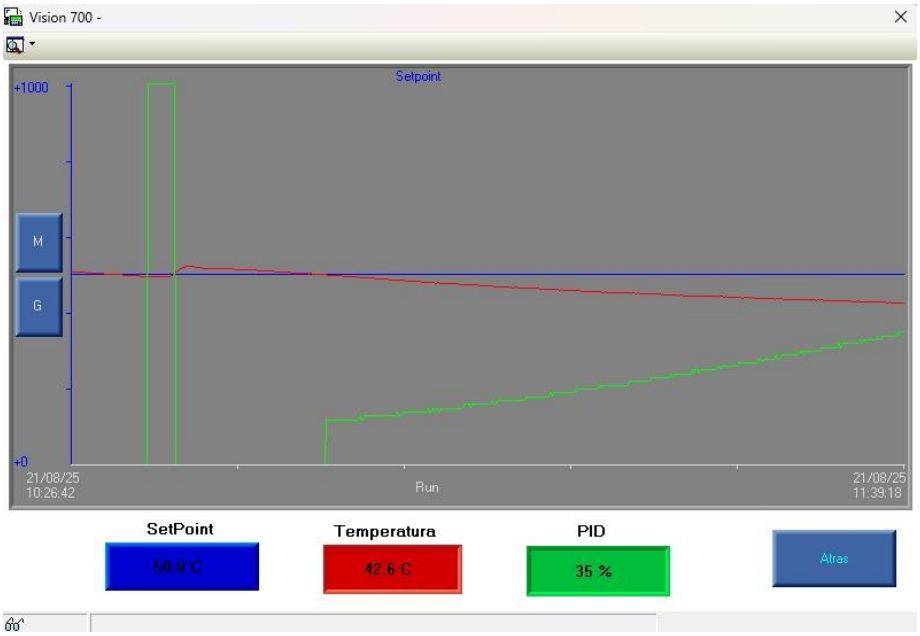


Figura 79. Control PID temperatura-2 [Fuente propia]

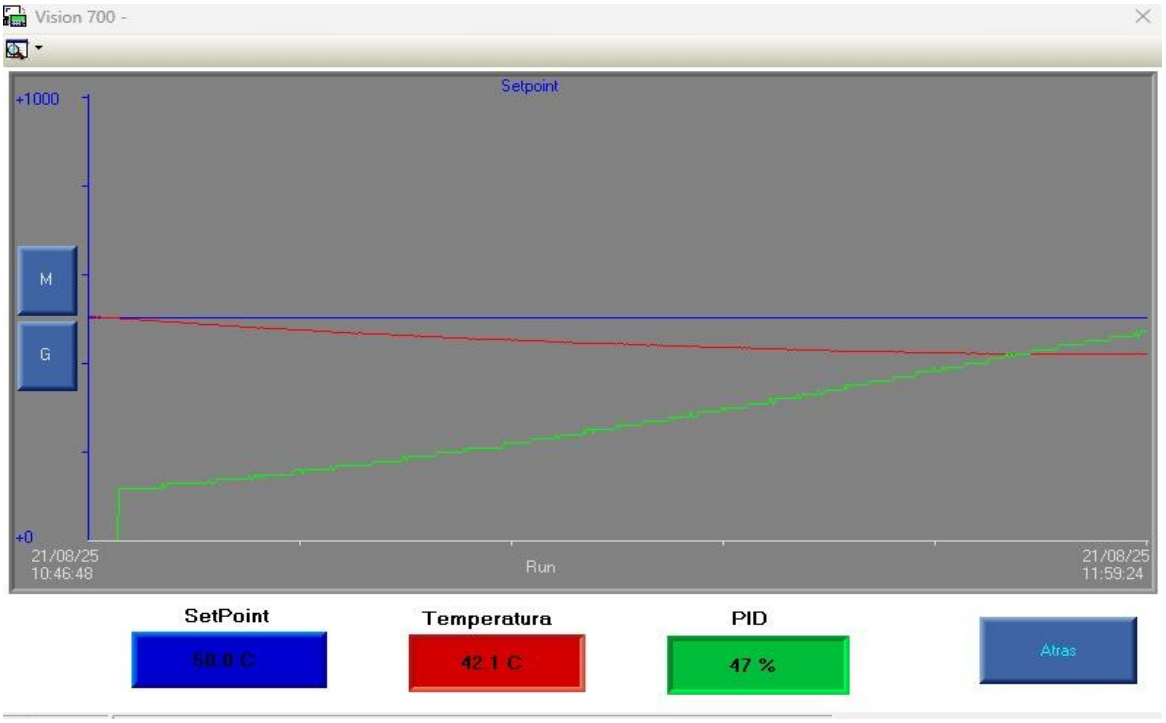


Figura 80. Control PID temperatura-3 [Fuente propia]

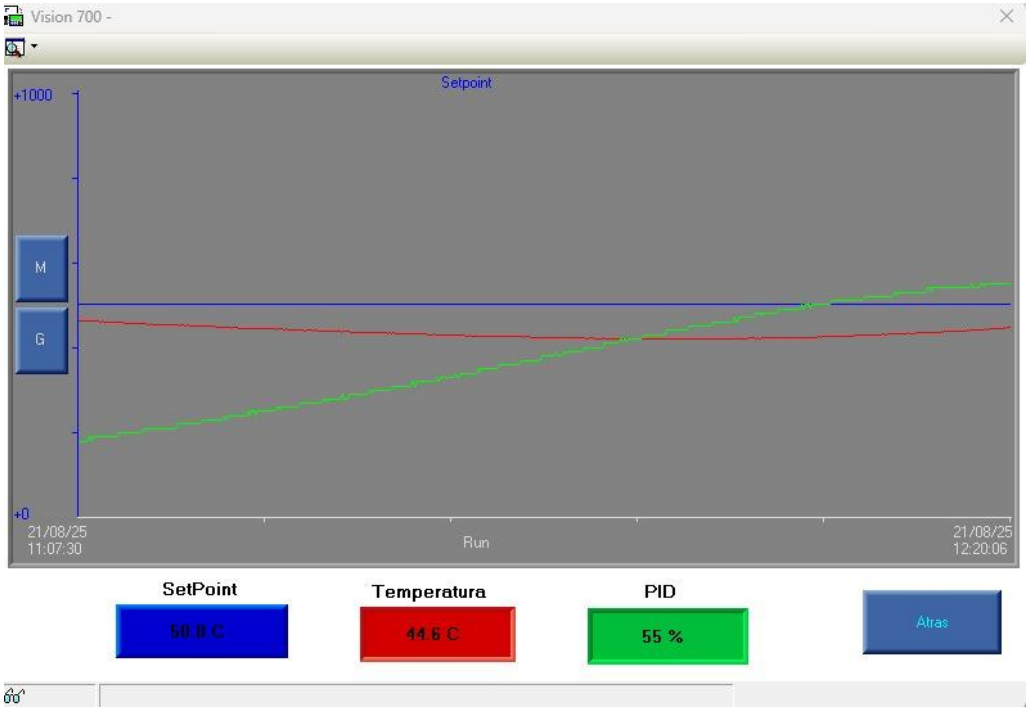


Figura 81. Control PID temperatura-4 [Fuente propia]

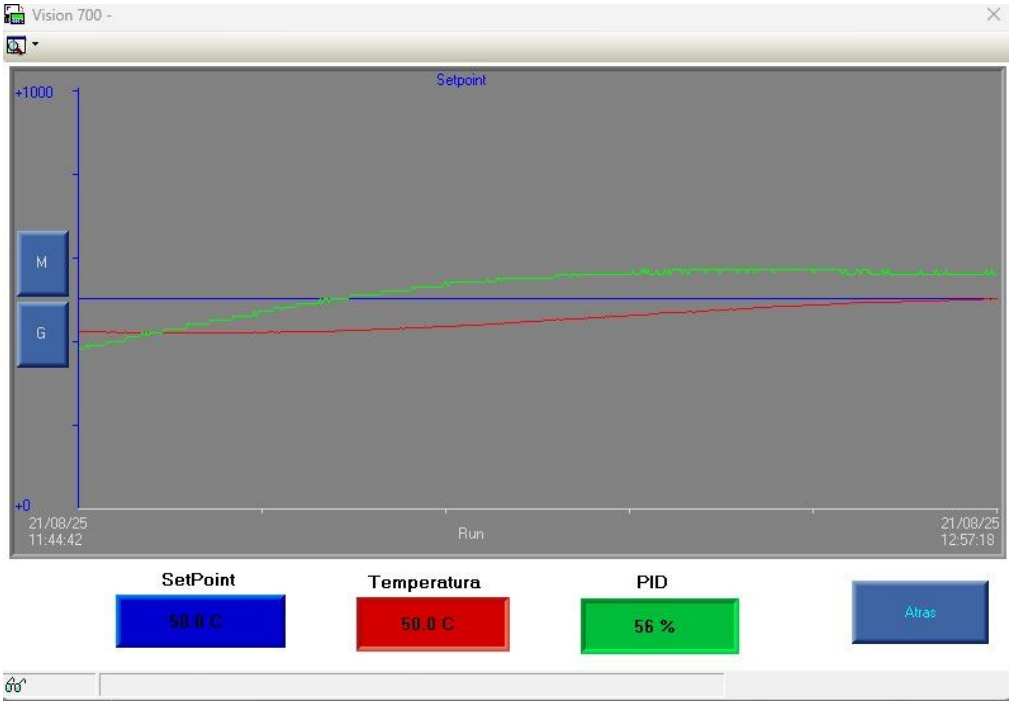


Figura 82. Control PID temperatura-5 [Fuente propia]

7.1.13. Control On-Off flujo

El sensor de flujo se programó en la placa de desarrollo ESP32, este dato se procesa y luego se envía por medio de Modbus, Se realizo de esta forma para comprobar la comunicación bidireccional del módulo de comunicación. Para ello primero se comprueba el funcionamiento optimo del sensor de flujo, esta prueba consta de conectar el sensor a una fuente de voltaje y la salida al osciloscopio, posterior a ello se inyecta un flujo de aire y se observa el comportamiento. El sensor YF-S201 posee un transductor de efecto Hall que genera pulsos eléctricos proporcionales al caudal de fluido que atraviesa el dispositivo. Cada pulso corresponde a una cantidad determinada de volumen, lo que permite calcular el flujo en litros por minuto a partir de la frecuencia de la señal. En la programación realizada en la ESP32, se emplearon rutinas de conteo de pulsos y temporización para convertir dicha señal en un valor digital representativo del caudal.

Posteriormente, el dato obtenido se escaló y se almacenó en un registro Modbus, desde donde es leído por el PLC. De esta manera se validó no solo el funcionamiento del sensor, sino también la capacidad del módulo de comunicación para intercambiar información en ambos sentidos, garantizando la transmisión confiable de datos de campo hacia el sistema de supervisión. Este valor luego se enviará al SCADA hecho en Node-red mediante el protocolo de comunicación MQTT.

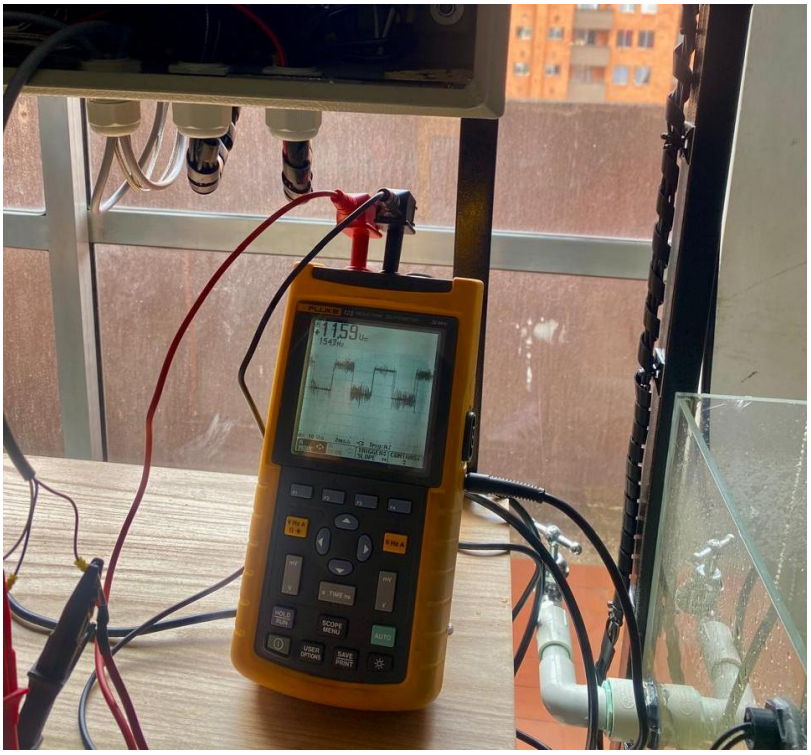


Figura 83. Señal del sensor efecto hall YF-S201 [Fuente propia]

Como se observa en la figura el comportamiento del sensor YF-S201 es el de una onda cuadrada, la salida del sensor es proporcional al voltaje de entrada del sensor. Por lo que el funcionamiento de el sensor de flujo se resume al funcionamiento de un switch que cambia de estado cuando la turbina va girando con el caudal. El caudal en litros por minuto se calcula contando los pulsos que genera el sensor en un segundo, obteniendo así la frecuencia, la cual se divide entre el factor de calibración del sensor (7.5 en el caso del YF-S201). El resultado corresponde al flujo en L/min y, para su envío al PLC mediante Modbus, se multiplica por 10 con el fin de representarlo en un valor entero. A continuación se describe el código para el proceso de flujo

```
1  #include <ModbusMaster.h>
2
3  volatile int contadorPulsos = 0;
4  const byte pinSensor = 25;   // GPIO de la ESP32 donde conectas el cable amarillo
5  unsigned long tiempoAnterior = 0;
6  float factorCalibracion = 7.5;  // Depende del sensor, el YF-S201 suele estar en ~7.5
7  int flujo=0;
8
```

Figura 84. Programación de variables [Fuente propia]

Primero se declara las variables necesarias, según el datasheet del sensor el factor a utilizar para calcular del caudal se utiliza un factor de conversión de 7.5

```
9   void IRAM_ATTR contarPulsos() {
10    contadorPulsos++;
11  }
12
13  ModbusMaster node;
14  void setup() {
15    Serial.begin(9600);
16    //Configuracion pin sensor flujo
17    pinMode(pinSensor, INPUT_PULLUP);
18    attachInterrupt(digitalPinToInterrupt(pinSensor), contarPulsos, RISING);
19    //configuración modbus
20    Serial2.begin(9600, SERIAL_8N1, 16, 17);// Puerto RS232 hacia el Vision700 UART2 en GPIO16 y 17
21    Serial2.flush();
22    // Iniciar el ModbusMaster con ID de esclavo 2 y puerto Serial1
23    node.begin(2, Serial2);
24    Serial.println("Iniciando lectura Modbus de MI5 a MI8...");
25  }
```

Figura 85. Programación de la interrupción y ModBus [Fuente propia]

Mediante estas funciones se realiza el conteo de pulsos del sensor, esta acción se realiza por medio de la interrupción con un flaco de subida el cual aumenta un contador de a una unidad. Como se observa en el código de la figura la configuración del sensor es idéntica a un switch. En esta sección también se declara la velocidad del puerto y el protocolo ModBus para enviar el doto hacia el PLC. Para este caso se está iniciando la ESP32 como maestro y el PLC como esclavo con ID 2. Para esta prueba de control se va a escribir el dato calculado desde el microcontrolador y mediante el PLC se procesará para el control PID y On-Off

```
26  void loop() {
27    unsigned long tiempoActual = millis();
28    // Calcular cada segundo
29    if (tiempoActual - tiempoAnterior >= 1000) {
30      detachInterrupt(digitalPinToInterrupt(pinSensor));
31      float frecuencia = contadorPulsos; // Pulsos por segundo
32      float caudal = frecuencia / factorCalibracion; // L/min
33      flujo = caudal*10;
34      contadorPulsos = 0;
35      tiempoAnterior = tiempoActual;
36      attachInterrupt(digitalPinToInterrupt(pinSensor), contarPulsos, RISING);
37    }
38    //Enviar dato a MI 6 del plc vision 700
39    uint8_t result = node.writeSingleRegister(6, flujo);
40    Serial.println("////// Dato enviado ///////");
41    Serial.print("Flujo: "); Serial.println(flujo);
42    result = node.readHoldingRegisters(5, 4);
43    if (result == node.ku8MBSuccess) {
44
45    } else {
46      Serial.print("Error Modbus: ");
47      Serial.println(result);
48    }
49  }
```

Figura 86. Envío de dato de flujo por vía Modbus [Fuente propia]

El proceso de calcular el caudal es algo que el programa realiza cada segundo, para ello cada intervalo de tiempo cuanta los pulsos y los divide por el factor de calibración del sensor. Realizado esta operación resetea los pulsos y multiplica el dato por 10. Este dato es el que finalmente se envara por ModBus mediante la línea *"node.writeSingleRegister"*. Este dato se almacenará en la marca de memoria M6 del PLC constantemente.

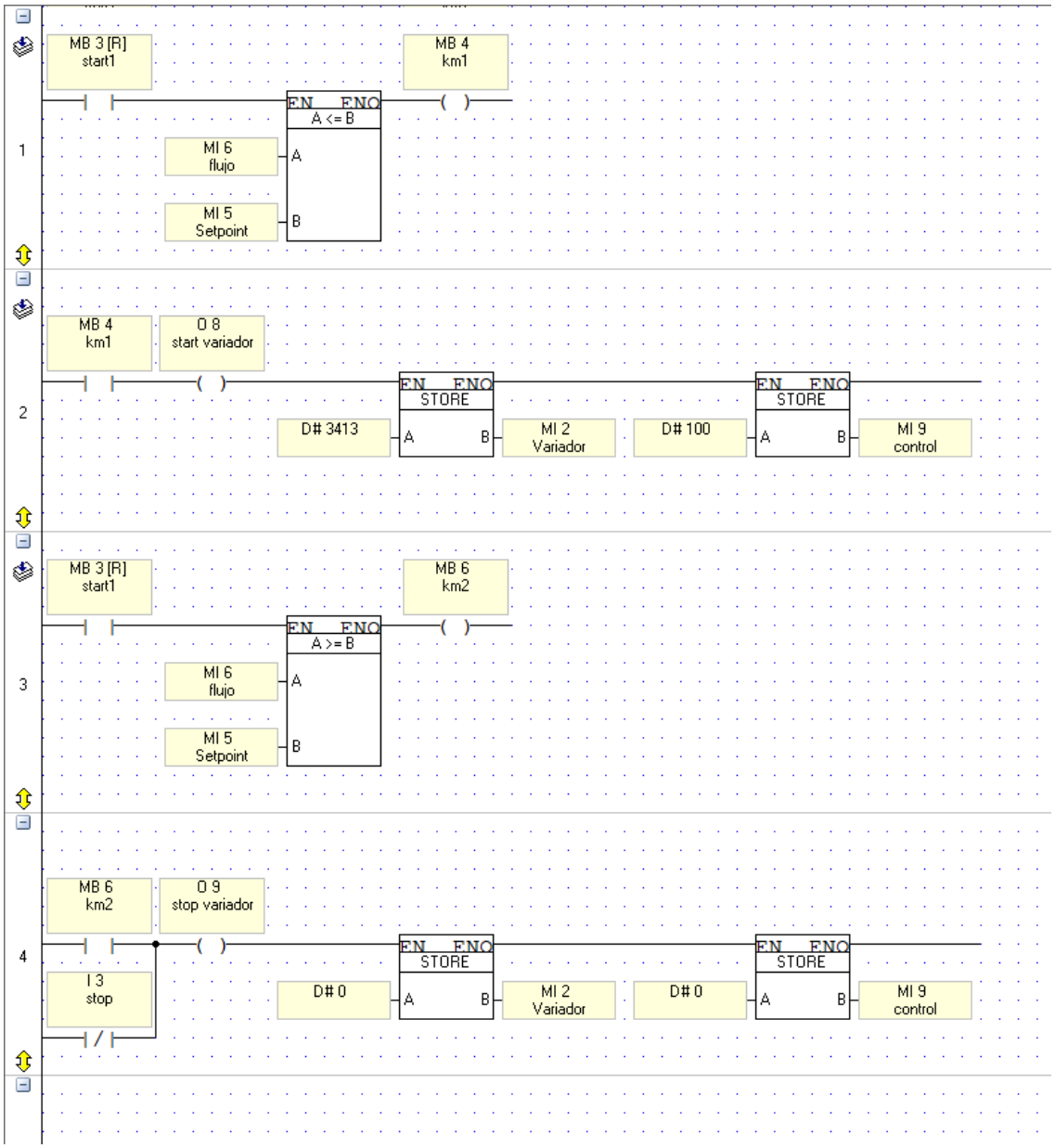


Figura 87. Programación control, On-Off, proceso flujo [Fuente propia]

El programa recibe el dato del sensor almacenado en la marca de memoria MI6 y compara si es mayor o menor que el punto de referencia asignado (Setpoint). Dependiendo de esta acción enciende o apaga el variador de frecuencia. A continuación se observa la respuesta del control On-Off

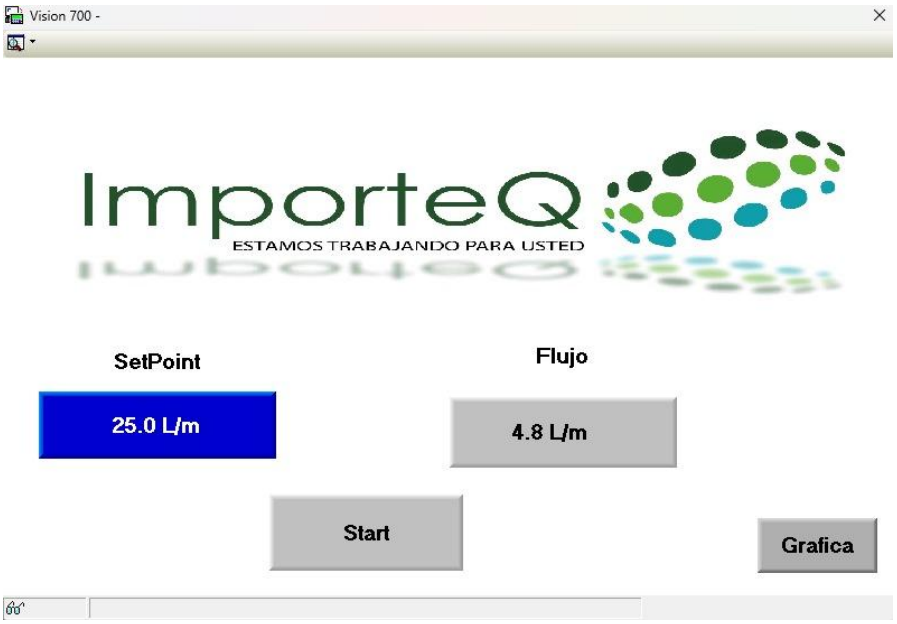


Figura 88. Diseño HMI ventana inicial [Fuente propia]

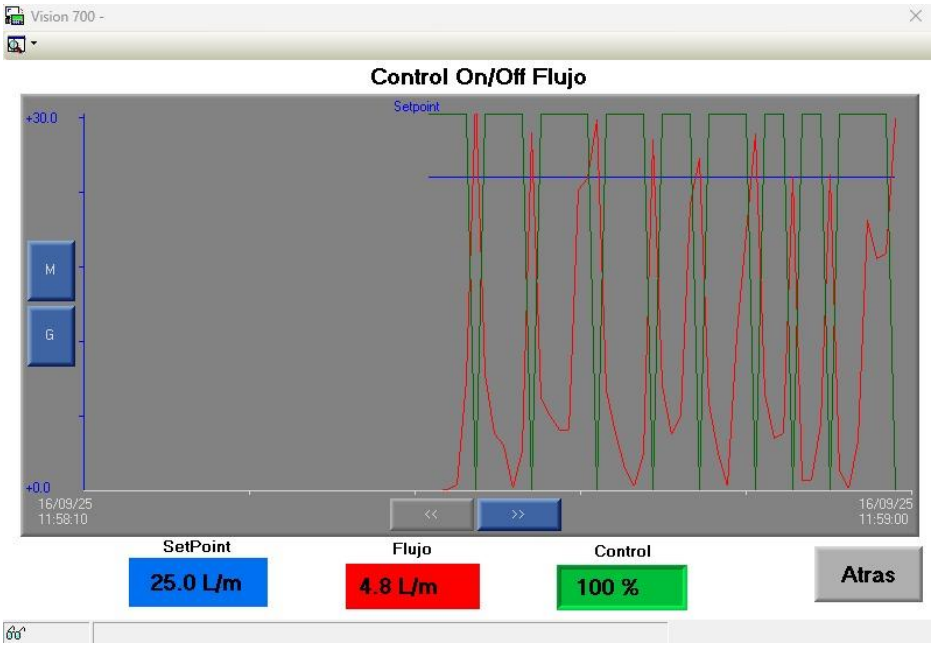


Figura 89. Control On-Off de flujo-1 [Fuente propia]

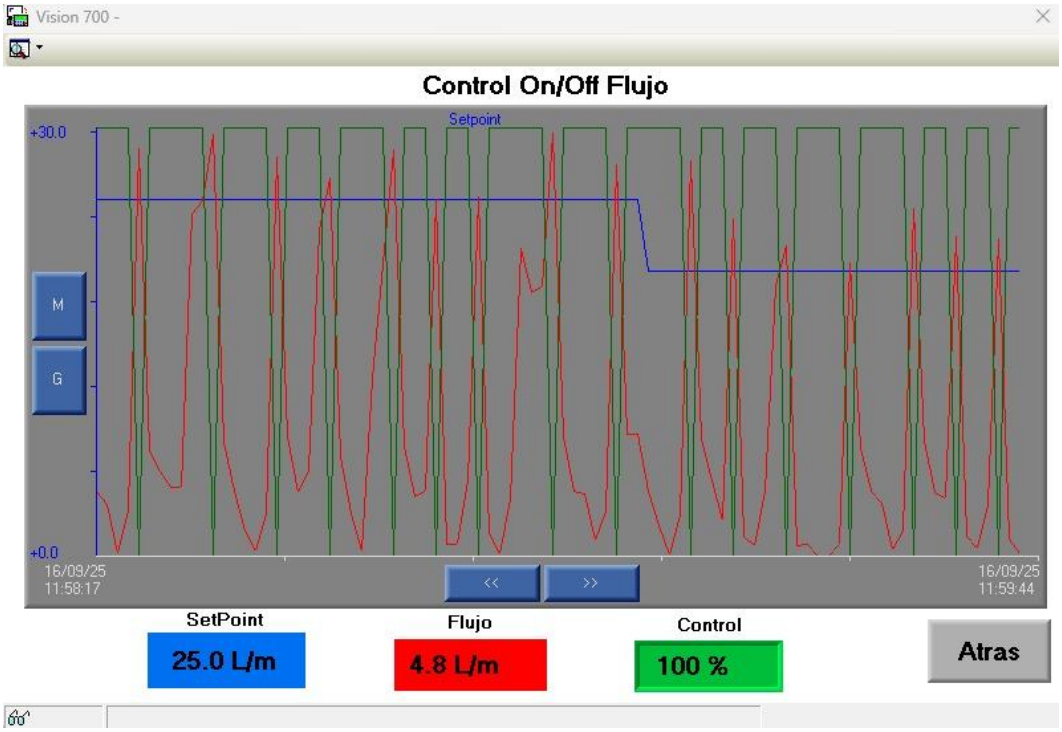


Figura 90. Control On-Off de flujo-2 [Fuente propia]

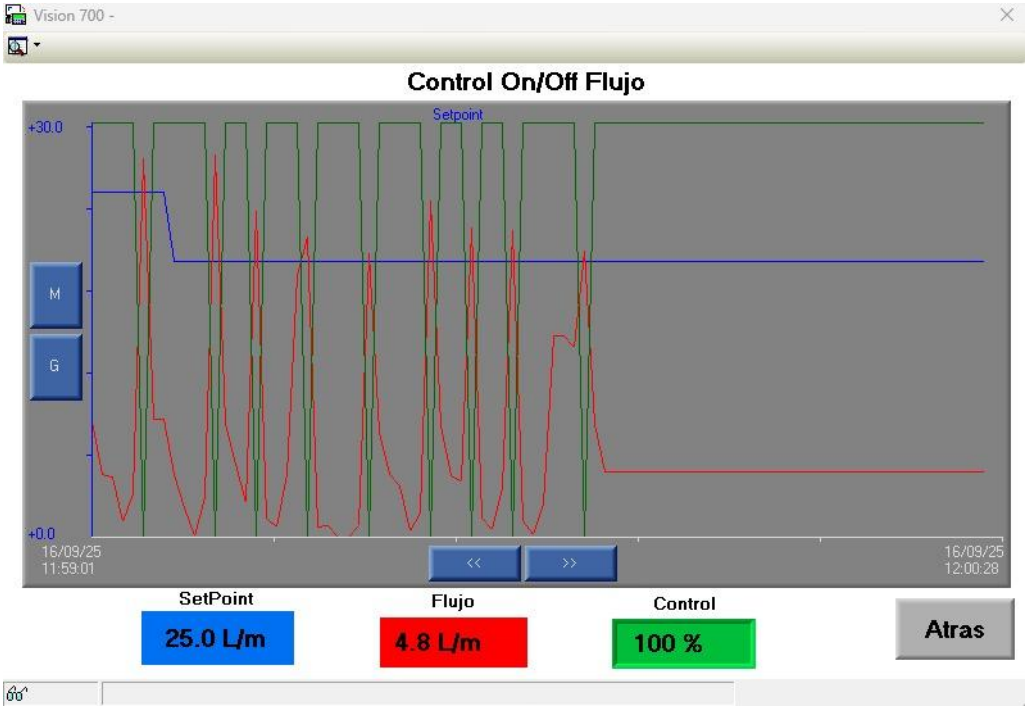


Figura 91. Control On-Off de flujo-3 [Fuente propia]

7.1.14. Control PID flujo

Como salida de control se tomó la válvula proporciona, partiendo de este elemento para realizar el control PID de nivel se configura el bloque de auto ajuste con los siguientes parámetros:

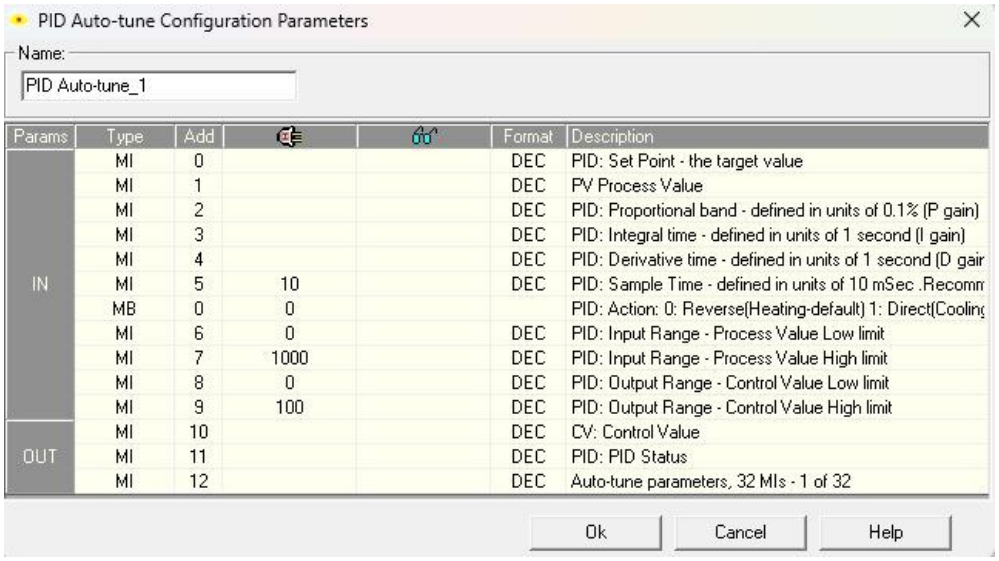


Figura 92. Configuración bloque PID, control caudal [Fuente propia]

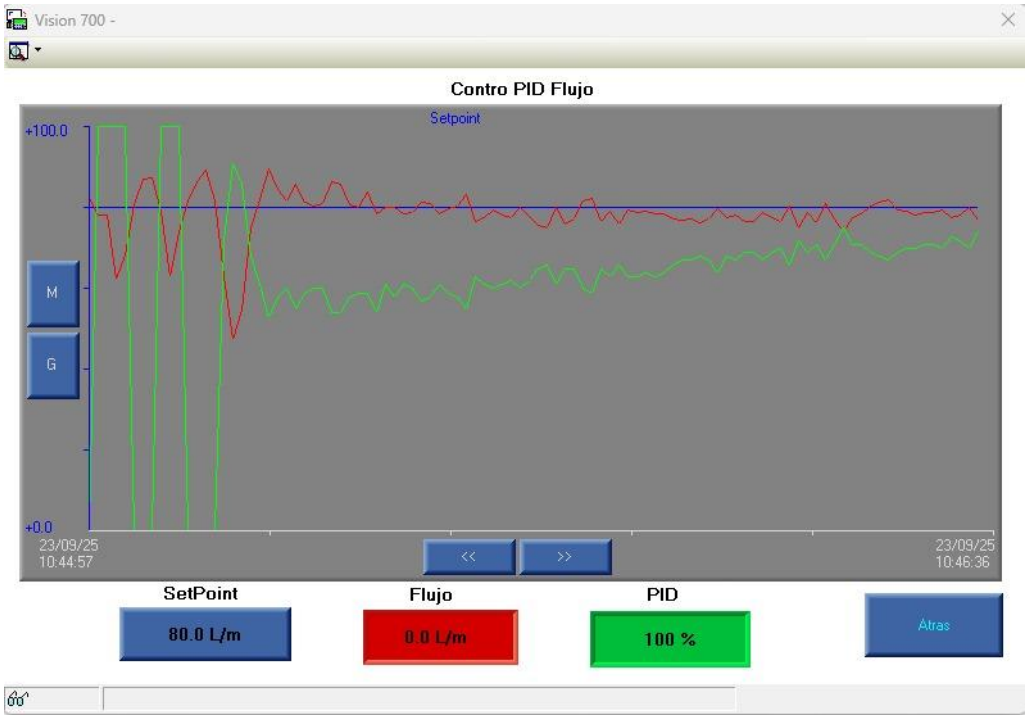


Figura 93. Control PID de flujo-1 [Fuente propia]

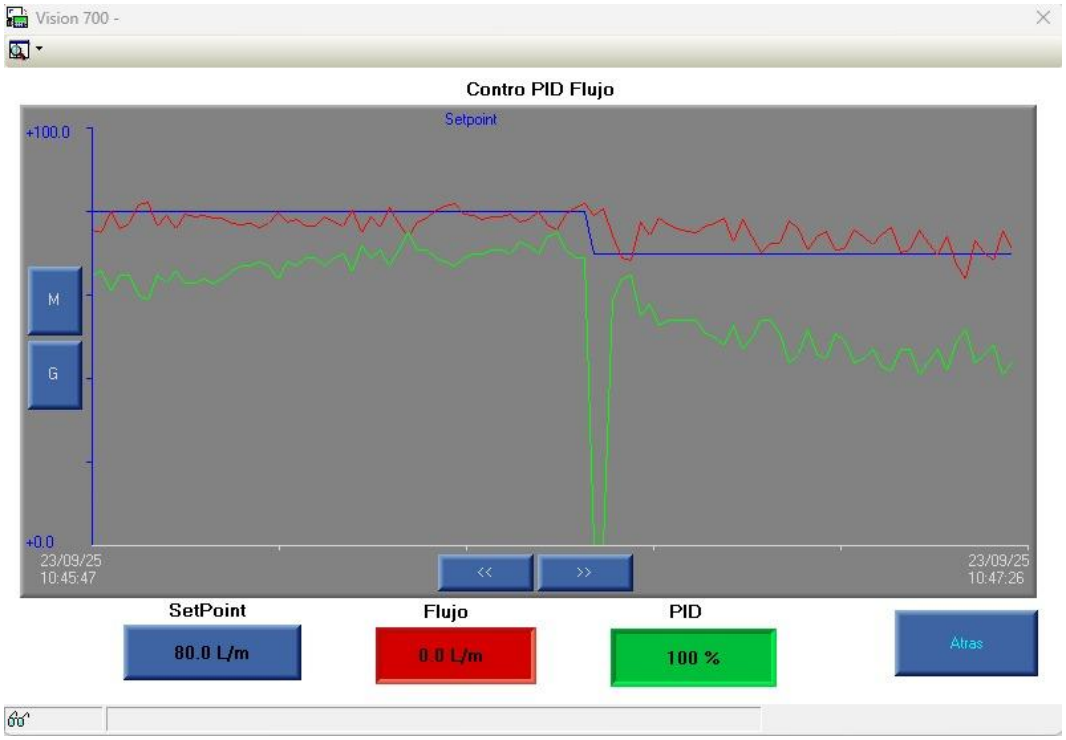


Figura 94. Control PID de flujo-2 [Fuente propia]

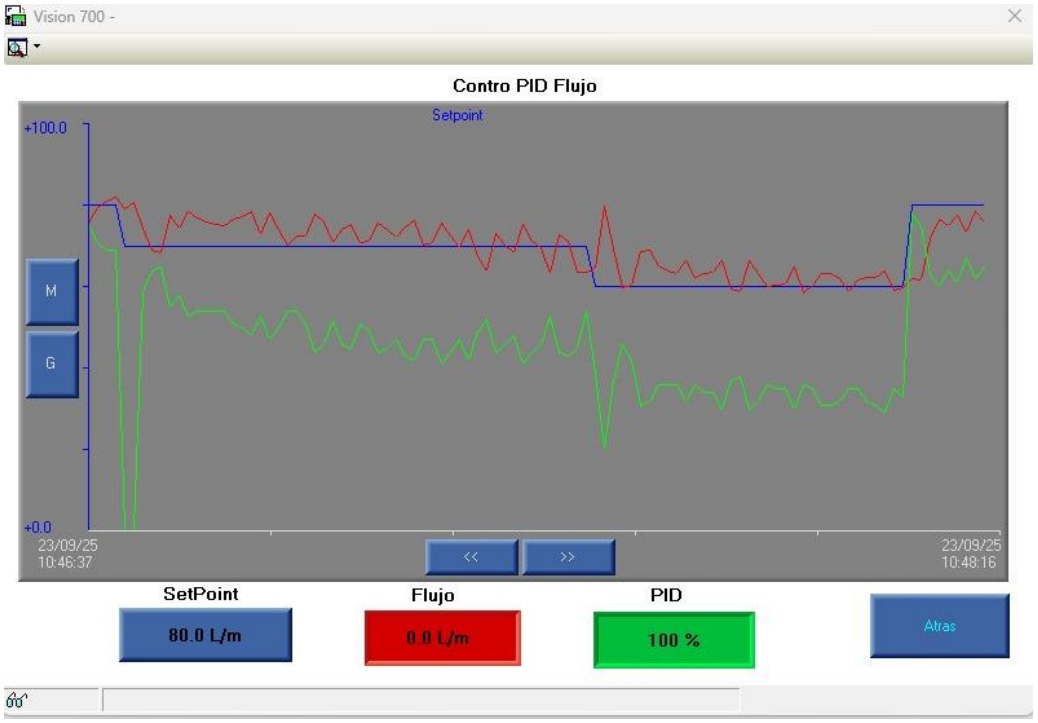


Figura 95. Control PID de flujo-3 [Fuente propia]

Como se observa en la grafica de la figura el bloque PID siempre inicia de la misma forma esto con el fin de asegurar las constantes de control de acuerdo al proceso. El comportamiento de la graficas parece normal excepto por los picos que se forman en la señal. Este fenómeno

a diferencia del sensor de nivel no es ruido, si no que corresponde debido a la ubicación vertical del sensor dentro del sistema. Como se observa la señal azul corresponde al punto de referencia y la señal roja el valor del sensor, si se realiza una comparativa la señal del sensor se mantiene dentro del rango del setpoint. Para ello, en ese proceso la válvula proporcional constantemente esta cambiando su porcentaje de apertura para poder mantener el caudal hacia los litros por minutos asignados



Figura 96. Válvula proporcional y sensor de flujo YF-S201 [Fuente propia]

Para realizar las pruebas de flujo fue necesario mantener abiertas las válvulas de desagüe hacia el tanque inferior, dado que el objetivo del control no era llenar el tanque superior, sino regular el caudal en el sistema de circulación de agua. Durante toda la prueba se mantuvo activada la válvula solenoide sin presión, mientras que la válvula proporcional, equipada con un indicador de apertura, permitió verificar su grado de funcionamiento. En esta etapa, al igual que en la prueba de control On-Off, la única válvula utilizada fue la proporcional.

7.1.15. Comunicación entre dispositivos

Para realizar recepción de los datos por parte del SCADA este proceso de comunicación inicia por el PLC-HMI quien se encarga de procesar los valores de las variables de los

sensores y actuadores que se están registrando constantemente. Estos datos se envían por ModBus RTU a la ESP32 quien funcionara como maestro. Cuando los datos se reciben inmediatamente se envían por el protocolo MQTT al servidor de HiveMQ quien posteriormente los publicara en Node-Red, a través del SCADA realizado en inskape. Este proceso se realiza a tiempo real, esto permite tener una información actualizada y constante de la planta sin pérdida de información. Para esta etapa es crucial asegurar una estabilidad de la conexión en dos momentos, cuando la planta se enciende y cuando la conexión se pierde.

Para realizar estas pruebas y obtenido la conexión y recepción de datos el proyecto se deja operativo por un tiempo de 8 horas, esto con el fin de asegurar la estabilidad de ambos protocolos de comunicación, ModBus y MQTT. Luego se realizaron pruebas de desconexión, esta etapa consistió en desconectar la ESP32 de la red wifi y volviendo reconectar a la red. Como se observó en los resultados la ESP32 no establecía una conexión de datos al cluster de HiveMQ a no ser que se reiniciara de nuevo el proyecto. Para corregir este inconveniente se realiza la reconexión mediante la programación de una verificación donde se establece una validación con el MQTT y el wifi y en caso de ocurrir una desconexión el programa vuelve a realizar el intento de vínculo con los protocolos.

El tiempo de envió de datos se establece desde la programación de la ESP32, por lo que el intervalo se puede modificar dependiendo del dato del sensor que se esté enviando. Para datos de temperatura se recomienda aproximadamente entre 3 y 4 segundos y para valores de flujo y nivel máximo un 1 segundo de intervalo. Es importe recordar que el cluster tiene un peso máximo de 10GB, es por ello que enviar muchos datos redundantes pueden llenar rápidamente el espacio asignado para el plan gratuito. Este tiempo impacta directamente en la actualización del SCADA, ya que, a mayor intervalo definido, mayor será la demora en la actualización de los datos visualizados en el sistema. A continuación se presenta un diagrama de flujo que representa este proceso.

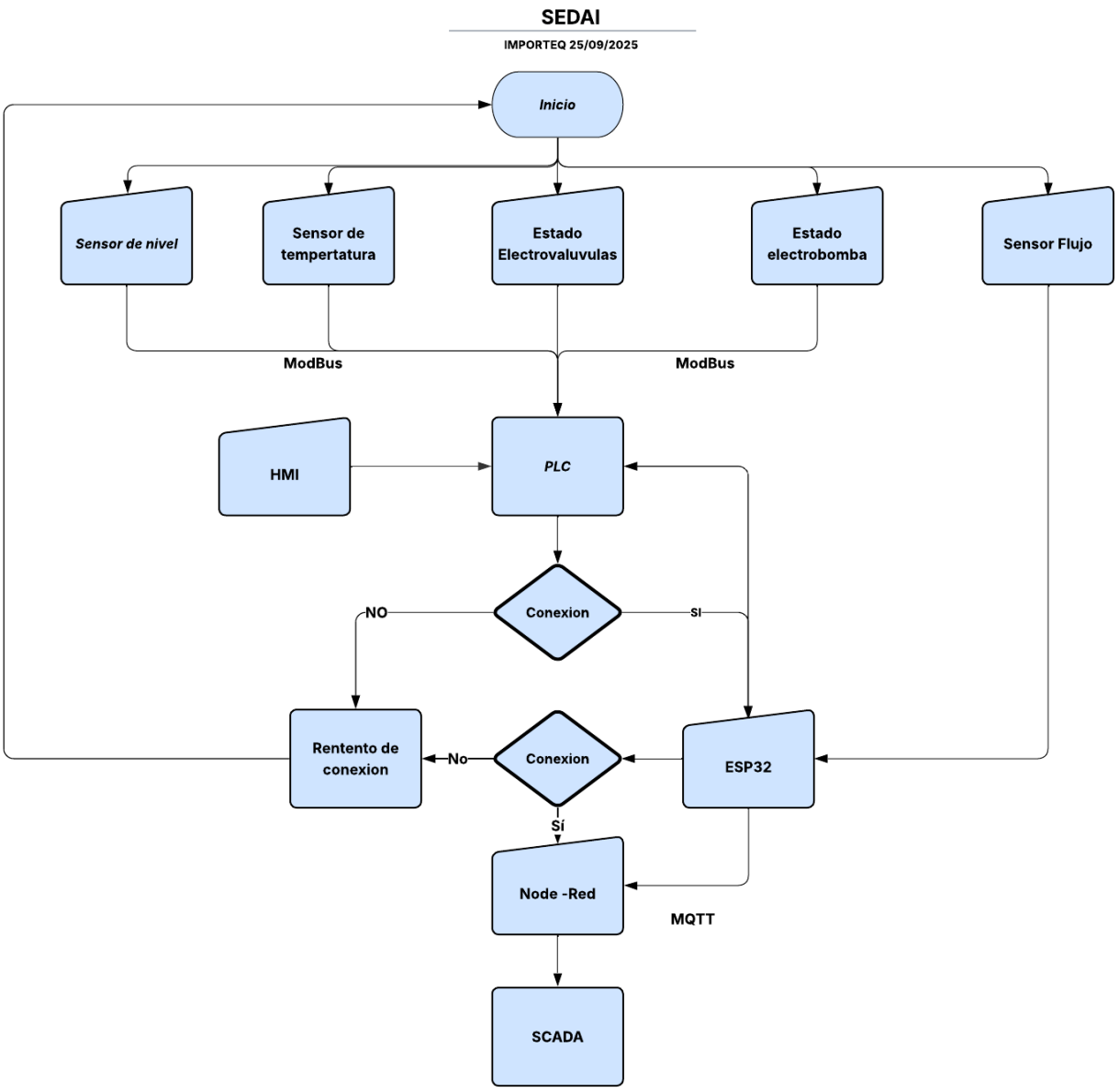


Figura 97. Diagrama de flujo SEDAI [Fuente propia]

Una vez comprobado la conexión con los códigos de prueba se procede a establecer el enlace con los sensores y actuadores. La programación en el PLC no varía, ya que como está funcionando como esclavo es la ESP32 como maestro quien se encarga de extraer los datos a través de la marca de memoria asignada. Por lo que para extraer los datos hay que tener presente dos detalles importantes. Uno es desde donde inicia la marca de memoria asignada para el primero sensor y hasta donde van las marcas a consultar. Por lo que si se desea extraer el dato de la marca Mi1 hasta el Mi10 son 10 registros a los que accede la ESP32. La segunda información es el tipo dato, si es un tipo entero o booleano ya que aunque es necesario conocer desde donde empieza y donde termina, estos datos no se procesan de la misma forma.

Identificado las marcas de memoria que se van a utilizar se asignan para los valores enteros y para los valores booleanos. Se programan los elementos de planta y se ejecutan junto a la ESP32. Antes de publicar los datos se realiza una prueba para verificar los datos en Node red a través de la consola antes de ser enlazados al SCADA. A continuación, se muestra la conexión con el sensor de temperatura, cuya lectura es visualizada en la consola de Node-RED. En esta sección se detallan la conexión ModBus y los códigos implementados para la reconexión tanto de MQTT como de la red Wifi. Esta prueba garantiza una comunicación estable y confiable entre todos los dispositivos.

Para iniciar el proceso de comunicación se configuran los bloques de Modbus, usuario y contraseña explicados en la actividad 6.1.7. Estos bloques son fundamentales y se utilizaran a lo largo de cualquier practica y actividad del proyecto SEDAI. Para programar el sensor se asignada a la variable análoga. Asignada la marca de memoria se inicia con la programación de la ESP32. Primero se establece los siguientes parámetros:

```
#include <ModbusMaster.h>
#include <WiFi.h>
#include <PubSubClient.h>
#include <WiFiClientSecure.h>// Librerias  utilizadas

ModbusMaster node; //Se declara la ESP32 como maestro
HardwareSerial mySerial(1); //Declaracion del serial

WiFiClientSecure espClient;
PubSubClient client(espClient);

const char* ssid = "Moto E7 de Nicolas";
const char* password = "Nicolas1234";  //Red Wifi

const char* mqtt_server = "98ad036b0a2d47119b8e8be5b737200b.s1.eu.hivemq.cloud"; //Url del cluste
const int mqtt_port = 8883; // Puerto MQTT

const char* mqtt_username = "Envios_datos_SEDAI"; // Usuario de la credencial (IMPORTANTE)
const char* mqtt_password = "Sedai123";          // Contraseña de la credencial (IMPORTANTE)

const char* inTopic = "LED";
const char* outTopicTemp1 = "Valor1";
```

Figura 98. Librerías y variables, módulo de comunicación [Fuente propia]

En esta sección se declaran:

- Librerías
- ESP32 como maestro Modbus
- MQTT
- Puerto serial
- Red wifi (SSID y contraseña)
- Servidor MQTT
- Credenciales del cluster
- Tópicos (Entradas y salidas)

```
60  void setup_wifi() {
61    Serial.println();
62    Serial.print("Conectando a ");
63    Serial.println(ssid);
64    WiFi.begin(ssid, password);
65    while (WiFi.status() != WL_CONNECTED) {
66      delay(500);
67      Serial.print(".");
68    }
69    Serial.println("\nWiFi conectada");
70    Serial.print("IP: ");
71    Serial.println(WiFi.localIP());
72  }
73
74  void callback(char* topic, byte* payload, unsigned int length) {
75    Serial.print("Mensaje recibido en canal: ");
76    Serial.println(topic);
77  }
78
```

Figura 99. Conexión wifi [Fuente propia]

Sin conexión a la red, el programa no se ejecuta. En las pruebas realizadas, el enlace a la red Wifi se establece de forma inmediata, siempre que las credenciales sean correctas y la calidad de la red sea adecuada. Una vez establecida la conexión se imprime un mensaje en la consola, se imprime la IP y el programa pasa luego a realizar el vínculo MQTT. La función *"callback"* corresponde a la recepción de datos desde el servidor HiveMQ o cualquier otro servidor.

```
79  void reconnect() {
80    while (!client.connected()) {
81      Serial.print("Conectando a MQTT...");
82      if (client.connect("ESP32Client", mqtt_username, mqtt_password)) {
83        Serial.println("Conectado a MQTT");
84        client.subscribe(inTopic);
85      } else {
86        Serial.print("Fallo, rc=");
87        Serial.print(client.state());
88        Serial.println(". Reintentando...");
89        delay(5000);
90      }
91    }
92  }
93  void setup() {
94    Serial.begin(9600);
95    Serial1.begin(9600, SERIAL_8N1, 16, 17);  // UART2 en GPIO16 y 17
96    Serial1.flush();
97    setup_wifi();
98
99    espClient.setCACert(root_ca);
100   client.setServer(mqtt_server, mqtt_port);
101   client.setCallback(callback);
102
103   node.begin(2, Serial1);  // ID del esclavo Modbus
104   Serial.println("Inicialización completa. Enviando datos...");
105   delay(5000);
106 }
```

Figura 100. Conexión MQTT [Fuente propia]

En la función *"reconnect"* cumple dos funciones importantes, la primera es realizar la conexión al cluster de HiveMQ con las credenciales del usuario y reconectar si hay un fallo en la red o en la comunicación MQTT. Sin la validación de estos dos protocolos de

comunicación es imposible realizar un enlace del PLC al SCADA en Node-Red es por ello que las funciones no terminan si ambas conexiones están completadas. En el setup se declaran los puertos seriales para la ESP32, los puertos MQTT. Por último indica un mensaje que hace referencia a que el envío de datos ya empezó por parte del microcontrolador a la nube.

```
108  void loop() {
109    if (!client.connected()) {
110      reconnect();
111    }
112    client.loop();
113
114    // Leer MI47 continuamente
115    uint8_t result = node.readHoldingRegisters(47, 10);
116
117    if (result == node.ku8MBSuccess) {
118      uint16_t mi47 = node.getResponseBuffer(0);
119      float mi47_1 = mi47 / 10.0;
120      //uint16_t mi29 = node.getResponseBuffer(4);
121      Serial.print("Valor MI47: ");
122      Serial.println(mi47_1);
123      // Publicar a MQTT
124      client.publish(outTopicTemp1, String(mi47_1).c_str(), true);
125      //client.publish(outTopicNivel, String(mi29).c_str(), true);
126    } else {
127      Serial.print("Error lectura Modbus: ");
128      Serial.println(result);
129    }
```

Figura 101. Lectura registros comunicación ModBus [Fuente propia]

La función *"loop"* se encarga de verificar constantemente la conexión MQTT y reconectarla si es necesario. Como se observa en la figura se leen los registros desde la marca de memoria 47 hasta la marca de memoria 57. En este caso solo se toma Mi47 y cuando se detecta un mensaje con la característica de la marca solicitada se almacena en una variable unt16_t. El programa VisiLogic no procesa valores de tipo flotante, es por es que para recibir este dato hay que dividirlo. Por ejemplo si se envía una temperatura de 34.5ºC el PLC lo envía como 345, para solucionar este inconveniente se divide por 10.0 y se almacena en una variable flotante para posteriormente ser publicado en el cluster mediante el tópico "Valor1" almacenado en "outTopicTemp1" . Si hay un problema la comunicación ModBus se imprime el mensaje con el tipo de error.

Para realizar una verificación del envío de estos datos se puede hacer desde la consola del cluster o directamente desde Node-Red con la configuración del nodo *"MQTT in"*. En este caso se realiza la verificación desde la consola de HiveMQ quien imprime los datos cada 2 segundos. En este apartado se presenta el tópico y el mensaje recibido.

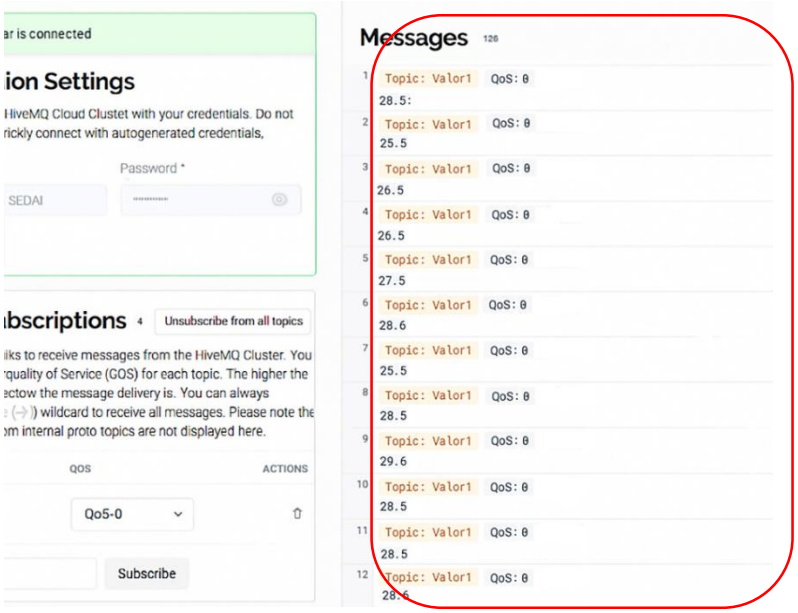


Figura 102. Recepción de temperatura en HiveMQ [Fuente propia]

Con los valores recibidos en cluster solamente se realiza una configuración de nodo *"MQTT in"* explicada en la actividad 6.1.5. Para este apartado lo único que varía es el nombre del tópico, para la termocupla se definió como Valor1. El nombre siempre debe coincidir ya que es un registro único por el cual se trasmite el mensaje.



Figura 103. Lectura temperatura de tópico "Valor1" [Fuente propia]

7.1.16. Diseño del SCADA

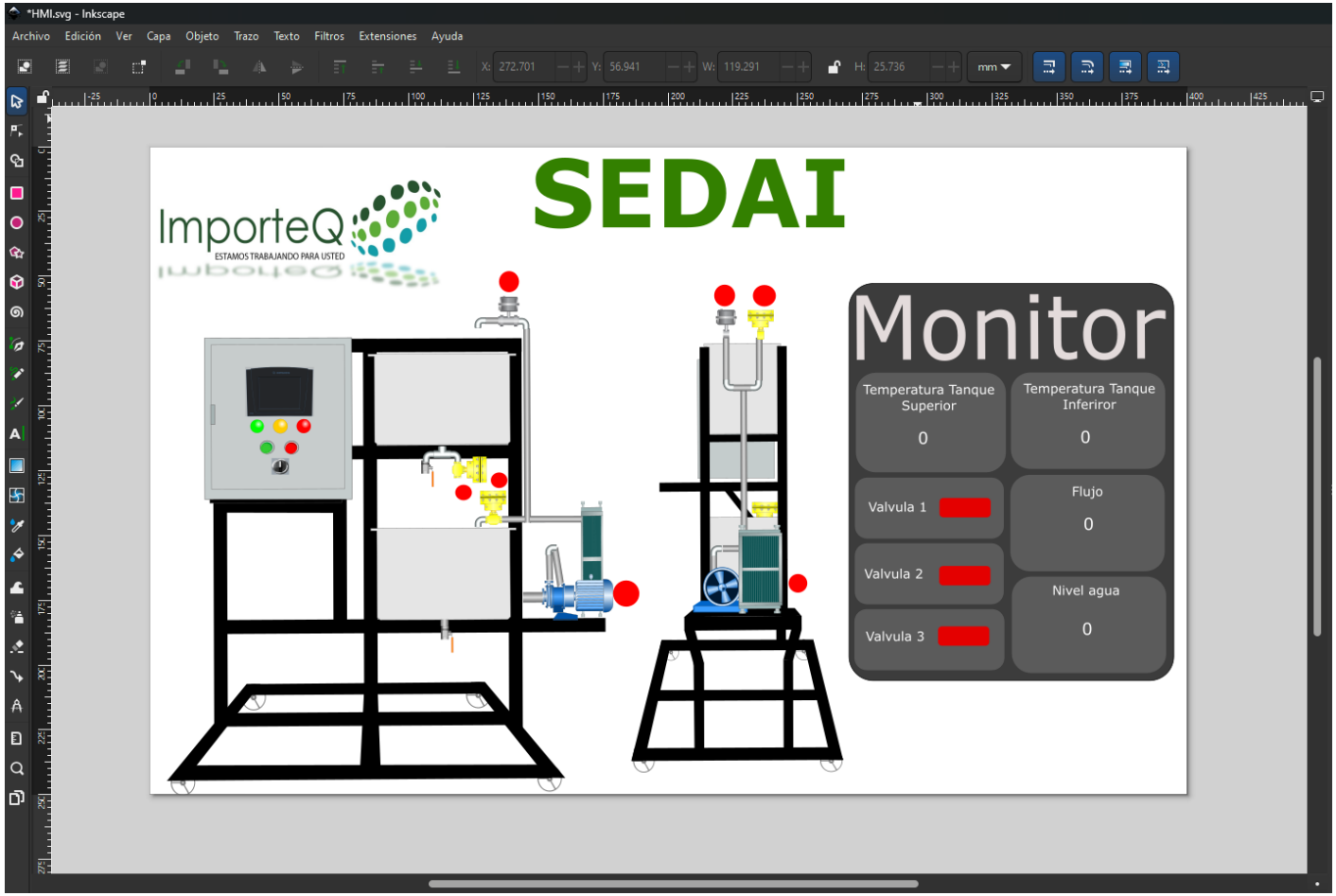


Figura 104. Efecto de una acción derivativa grande [Fuente propia]

El diseño del SCADA se realizó en Inskape, este software permite crear archivos SVG, este tipo de archivos permite interactividad y animaciones mediante JavaScript o CSS, al estar basado en XML, se puede modificar directamente el código para cambiar colores, formas o atributos. Este aspecto es importante ya que mediante los valores recibidos de la ESP32 se pueden actualizar para presentar la información de la planta mediante la actualización de valores numéricos, de color y animaciones. Para el diseño del SCADA se recreó la planta, incluyendo todos los elementos que la conforma y un panel donde se visualizaran los valores de los sensores de nivel, flujo y temperatura.

Para interactuar con las imágenes del archivo .SVG hay que hacer a través de la programación en JS. De esta forma se podrá manipular propiedades como el color, el color del texto, la actualización de los sensores y de los actuadores. Para ello se requiere de unos nodos llamados *"Funtion"* que recibirá el valor de los tópicos de Node-red y los publicará en el archivo SVG. Para manipular la imagen se realiza mediante la instalación de la librería "node-redcontrib-ui-svg-app", una vez importada quedara de la siguiente forma

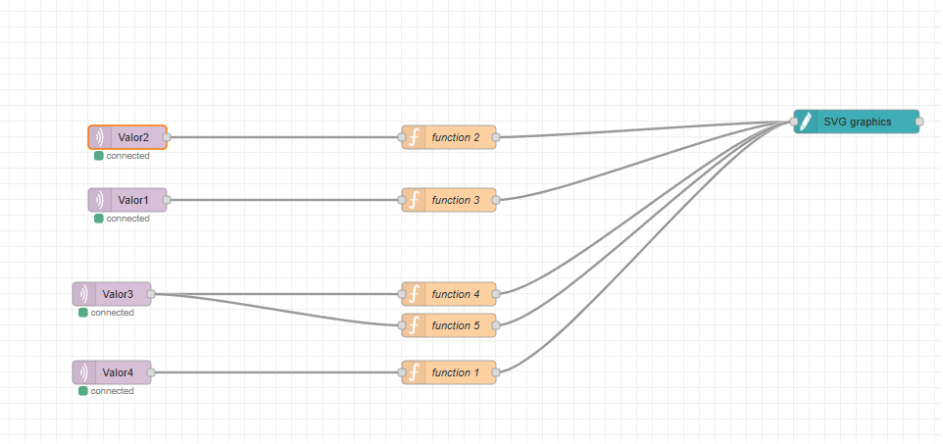


Figura 105. Nodos de tópicos enlazados al SCADA [Fuente propia]

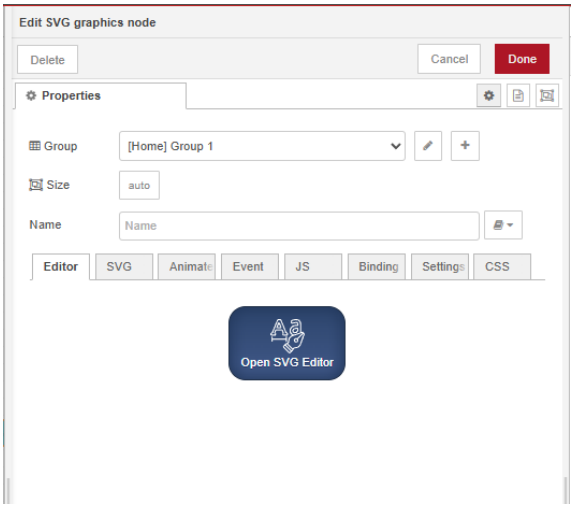


Figura 106. Configuración archivo SVG [Fuente propia]

Para la configuración se abre el editor y se cargara el diseño:

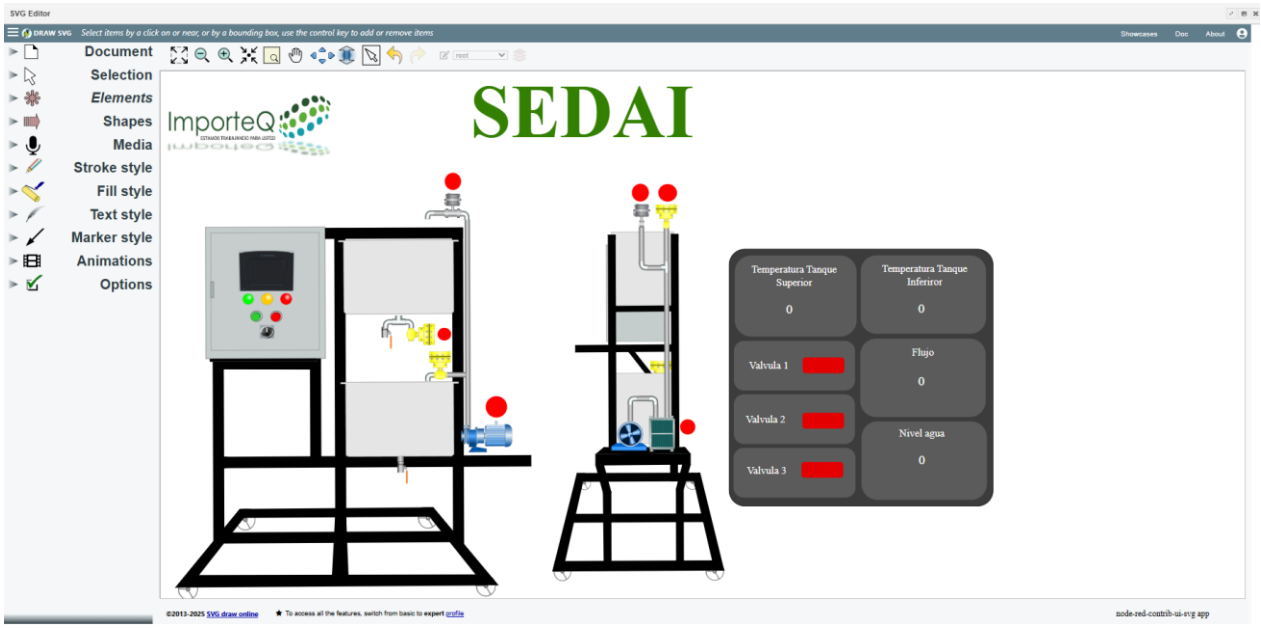


Figura 107. Configuración archivo SVG en Node-red [Fuente propia]

Una vez cargado el archivo se procederá a programar los nodos de función, quien dependiendo del tipo de información que contenga podrá realiza una acción determinada dentro del SCADA.

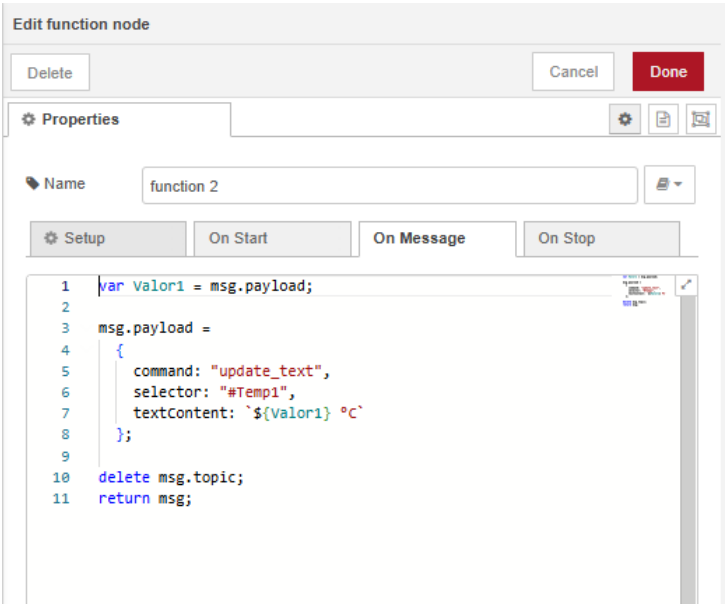


Figura 108. Programación tópico "Valor1" [Fuente propia]

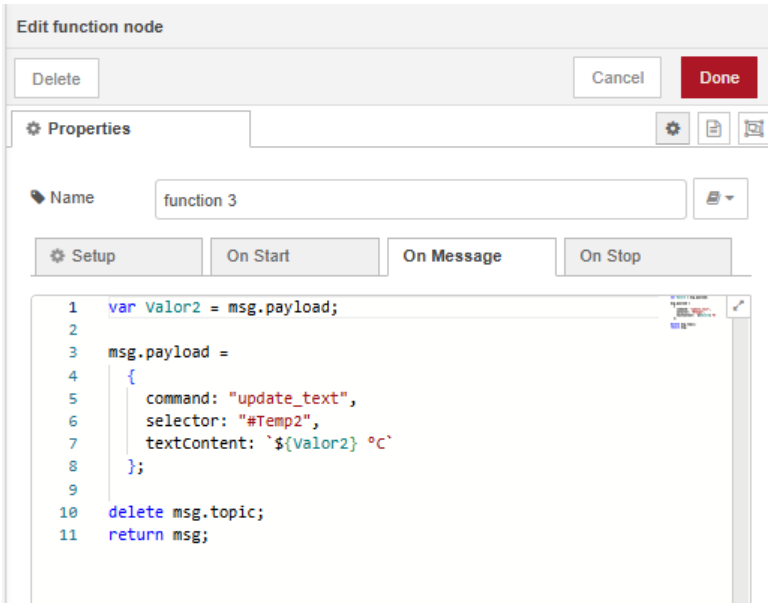


Figura 109. Programación tópico "Valor2" [Fuente propia]

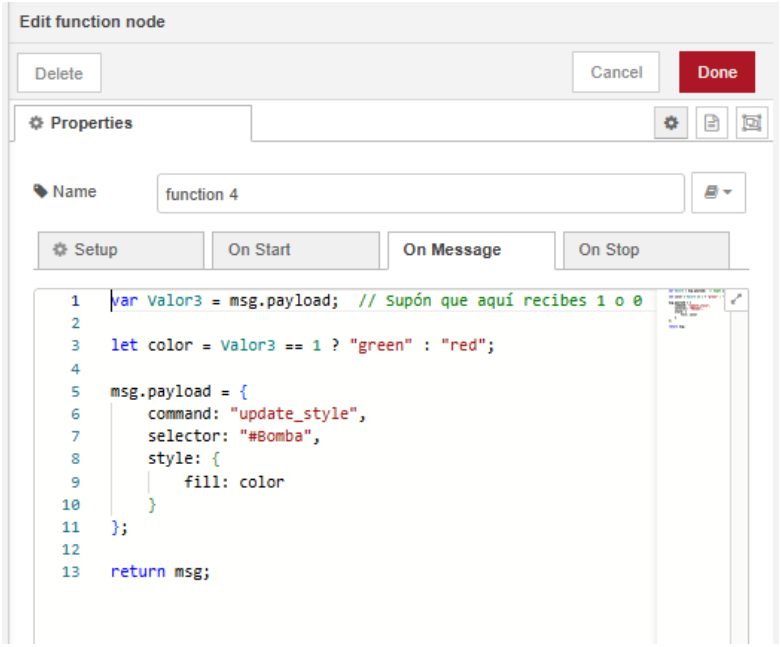


Figura 110. Programación Variable booleanas [Fuente propia]

Una vez programado los nodos requeridos para el proceso, se pone en marcha todo el sistema y se procede a observar la respuesta de los elementos en el SCADA:

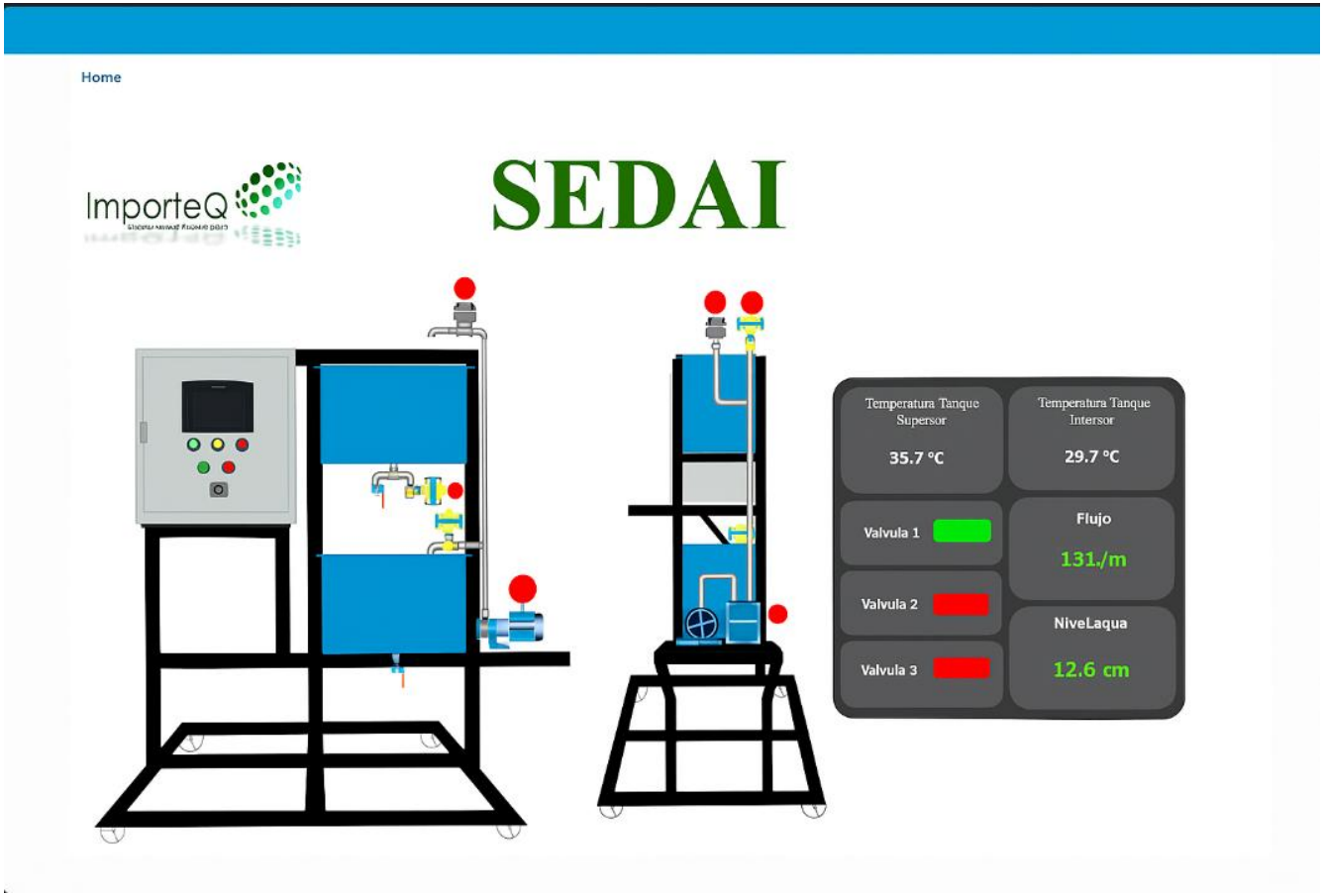


Figura 111. SCADA SEDAI [Fuente propia]

7.1.17. Diseño HMI SEDAI

Para la pantalla Vision700 que cuenta SEDAI de igual forma se realizo un HMI donde se presenta los valores de flujo, temperatura y nivel. Este archivo forma parte también del conjunto de los entregables del proyecto y funciona como base para el estudiante y docente. Para este diseño se recreo la planta con los elementos que cuenta VisiLogic.

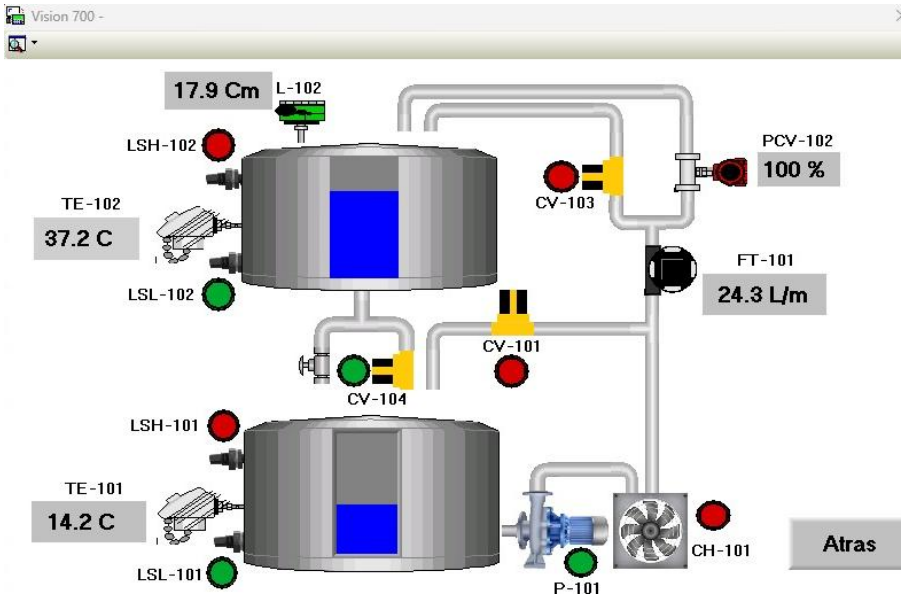


Figura 112. SCADA SEDAI [Fuente propia]



Figura 113. SCADA SEDAI [Fuente propia]

7.1.18. Actividades

Probado el sistema con todos los componentes a continuación se adjuntan actividades propuestas en la documentación de la planta SEDA.

Módulo 1 – Familiarización con señales y dispositivos

1. **Práctica 1: Entradas digitales**
    - Leer el estado de los sensores de nivel alto y bajo en ambos tanques.
    - Mostrarlo en el HMI y tablero como indicadores (luces verdes/rojas).
2. **Práctica 2: Salidas digitales**
    - Encender y apagar manualmente (desde el HMI y pulsadores) las válvulas solenoides
    - Verificar la respuesta física en la planta.
3. **Práctica 3: Entradas analógicas**
    - Calibración del sensor análogo de nivel tipo sonda.
    - Mostrar el valor en cm en el HMI.
4. **Práctica 4: Salidas analógicas**
    - Variar el valor de salida hacia las salidas análogas (variador, válvula proporcional) y ver su respuesta.

5. **Práctica 5: Termocuplas tipo J**
    - Lectura y visualización de la temperatura en cada tanque.
    - Conversión a °C y comparación con termómetro externo para verificar precisión.

Módulo 2 – Control básico ON/OFF

5. **Práctica 5: Control de nivel inferior-superior (ON/OFF)**
    - Con lógica de relé/ladder:
        - Si el nivel bajo del tanque superior se activa → prender bomba.
        - Si el nivel alto del tanque superior se activa → apagar bomba.
6. **Práctica 6: Control de temperatura ON/OFF**
    - Encender la resistencia sumergible cuando la temperatura baje de un valor mínimo (ej. 40 °C).
    - Apagar cuando supere un valor máximo (ej. 60 °C).
7. **Práctica 7: Alarma de seguridad**
    - Generar alarma visual en el HMI y piloto en tablero si el tanque superior llega a nivel máximo + temperatura máxima.

Módulo 3 – Control analógico

8. **Práctica 8: Control proporcional del nivel**
    - Usar la válvula proporcional para llenar el tanque superior suavemente, según la señal del sensor análogo de nivel.
9. **Práctica 9: Control de bomba con variador**

- Modulación de velocidad de la bomba para ajustar el caudal de llenado según la diferencia de niveles entre tanques.44

10. **Práctica 10: Curvas de respuesta**

    - Medir la respuesta del nivel en el tanque superior al aplicar diferentes velocidades de la bomba.
    - Analizar constante de tiempo y ganancia del sistema.

Módulo 4 – Control automático avanzado

11. **Práctica 11: Control PID de nivel**

    - Implementar un control PID en el PLC para mantener el nivel del tanque superior en un setpoint usando la bomba + variador.

12. **Práctica 12: Control PID de temperatura**

    - Usar la resistencia controlada por un SSR y un lazo PID para mantener la temperatura estable.
    - Comparar desempeño con el control ON/OFF.

13. **Práctica 13: Estrategia combinada (nivel + temperatura)**

    - Mantener un nivel constante en el tanque superior y al mismo tiempo una temperatura definida.
    - Coordinar el uso de la bomba, válvulas y resistencia.

14. **Práctica 14: Control en cascada**

    - PID primario: nivel en tanque superior.
    - PID secundario: velocidad de bomba (control de caudal).

15. **Práctica 15: Automatización completa de la planta**

- Secuencia automática:
    - Llenar tanque superior desde inferior.
    - Calentar agua hasta temperatura deseada.
    - Descargar con válvula proporcional a un proceso.
    - Repetir ciclo automáticamente.

Módulo 5 – Integración y proyectos finales

16. **Práctica 16: Registro de datos (Data Logging)**

- Registrar en el PLC/HMI tendencias de temperatura y nivel.

17. **Práctica 17: Comunicación industrial**

    - Lectura y escritura de variables por **Modbus RTU** desde la ESP32.
    - Leer datos de sensores en la ESP32 mediante **Modbus RTU** y subirlos a la nube.

18. **Práctica 18: Optimización de control**

    - Comparar diferentes modos de control: ON/OFF, Proporcional, PID.
    - Evaluar consumo eléctrico y eficiencia.

8. Conclusiones y Recomendaciones

El presente trabajo permitió crear un prototipo funcional que opera como un banco de pruebas para realizar control de temperatura, nivel y flujo mediante diferentes elementos de control que fueron instalados. A través de los protocolos de comunicación MQTT, ModBus y ethernet fue posible realizar una conexión a un SCADA mediante la implementación módulo IoT y la comunicación con Node-Red posibilitó el monitoreo remoto en tiempo real, de forma clara y estable, lo que alinea el proyecto con los requerimientos de la industria 4.0. El Sistema de entrenamiento SEDAI mediante su capacidad demuestra ser una herramienta de gran potencial para uso pedagógico en automatización y control. Las guías y documentación realizadas aportan un valor académico al facilitar el aprendizaje progresivo y estandarizado.

El proyecto está realizado para el estudiante, el entusiasta y toda aquella persona que desea ampliar sus conocimientos en automatización, apoyándose de una herramienta que pretende dar un énfasis a la industrial actual, sirviendo como base sólida en la construcción de conocimiento. Es por ello que el proyecto resalta la relevancia colaboración universidad-empresa, al vincular la experiencia técnica de ImporteQ con los objetivos académicos actuales, logrando un desarrollo alineado con las necesidades del sector productivo. Los elementos adquiridos por la compañía permitieron realizar una integración de tecnología dando forma a lo que es una primera versión del sistema y aunque hubo limitaciones en adquisiciones de componentes, estas no impidieron alcanzar los objetivos y por el contrario sirvieron como base para identificar oportunidades de mejora y ampliación futura del sistema.

Ahora con la adquisición del equipo a la universidad Uniminuto de Dios se espera ver el comportamiento dentro de un espacio real de trabajo, donde se incluirá dentro de las materias y servirá como apoyo a la universidad. El uso del equipo aportará información relevante a la empresa acerca del sistema, posibles mejoras y resultados en los estudiantes. Aunque las pruebas y guías se verificaron por Nicolás Sopo Bobadilla y Jean Carlo Leguizamón Martínez supervisados por el ingeniero Edward Hernando Bejarano Barreto, los resultados reales del proyecto serán corroborados por los estudiantes cuando pongan a prueba todas las capacidades que tiene SEDAI.

Recomendaciones:

Con la puesta en operación del sistema y observando los resultados obtenidos se tienen en cuenta las siguientes recomendaciones:

1. La principal mejora que tiene el proyecto es el sensor de nivel, esto es debido a que la señal que arroja el sensor TL-136 no es lo suficientemente estable para poder brindar una lectura precisa y exacta. Principalmente para este tipo de proyectos se recomienda utilizar

sensores ultrasónicos pero debido a que supera costos de presupuesto esta adquisición no se pudo realizar

2. Ampliar las variables de proceso incorporando otros módulos como presión y pH que permitan diversificar las prácticas de automatización, dando también como prioridad la expansión de entradas y salidas que ofrece la placa de desarrollo ESP32

3. Escalar el proyecto a otras instituciones educativas, promoviendo la estandarización de SEDAI como kit pedagógico accesible para la enseñanza de la automatización industrial

4. Fortalecer la supervisión de la planta mediante cámaras

5. Optimizar la documentación pedagógica con ejercicios que expandan la comunicación hacia múltiples dispositivos ampliando el funcionamiento del módulo

9. Referencias

---

1 Pérez Perdomo, A. J. (2023). *Industria 4.0: retos en la formación universitaria del ingeniero industrial*. Revista Ingenium, 1(1), 23-37. Recuperado de https://revista.uny.edu.ve/ojs/index.php/ingenium/article/view/384 (Revista Universidad Yacambú)

2 Groover, M. P. (2016). *Automation, Production Systems, and Computer-Integrated Manufacturing*. Pearson.

3 Mustard. (2024). ISA interchange. Recuperado de https://blog.isa.org/the-future-of-automation-in-the-workforce-what-do-we-need-to-know

4 Lasi, D. H. (2014). Industry 4.0. *BISE – CATCHWORD*.

5 Barberá, O. V. P. (1996). *El trabajo práctico en la enseñanza de las ciencias*. Departamento de Didáctica de las Ciencias, Universidad de Valencia.

6 Rodríguez Ruíz, O. A. M. S. M. F. J. (2021). *Desafíos en la formación de competencias y habilidades para la industria*. Universidad de Santo Tomás.

7 Vázquez-Mata, A. G.-L., & Guillamet-Lloveras, A. (s. f). El entrenamiento basado en la simulación

8 García, M. E. Á.-R. M. (s. f). *La simulación en la industria*. FICA – EISIC.

9 Amtrol. (2023). Amatrol. Recuperado de https://amatrol.com/our-story/iso-140012015-certified-amatrol/

10 Universidad Santo Tomás. (s. f). *Repositorio Institucional*. Recuperado de https://repository.usta.edu.co/server/api/core/bitstreams/8b9fb7b4-5a3b-456c-a94c-489f13f91f5f/content

11 Pirog, N. A. F. B. N. (s. f). Los trabajos prácticos como materiales didácticos para aplicaciones tecnológicas en carreras de ingeniería. Universidad Tecnológica Nacional, Facultad Regional Santa Fe.

12 Universidad Distrital Francisco José de Caldas. (s. f). Repositorio institucional. Recuperado de https://repository.udistrital.edu.co/server/api/core/bitstreams/41036bca-576a-4fcd-a6d4-8d25ff6d17da/content

14 Amatrol. (s. f). *Program, process control and instrumentation*. Recuperado de https://amatrol.com/product/program-process-control-and-instrumentation/

15 Lozano Villamizar, D. F., Girón Arbeláez, J. J., Piedrahíta, A. R., & Castellanos Garzón, J. A. (2023). Prototipo de un equipo didáctico para laboratorio de control y automatización en industria 4.0. En *Encuentro Internacional de Educación en Ingeniería*. https://doi.org/10.26507/paper.3234 (acofipapers.org)

16 Universidad del Valle / autoridad del repositorio. (s. f). *TFM I-3224*. Recuperado de https://uvadoc.uva.es/bitstream/handle/10324/77033/TFM-I-3224.pdf?sequence=1&isAllowed=y

17 Abele, E., Metternich, J., Tisch, M., Chryssolouris, G., Sihn, W., ElMaraghy, H., Hummel, V., & Ranz, F. (2017). Learning factories for research, education, and training. *Procedia CIRP, 57*, 1–6. https://doi.org/10.1016/j.procir.2016.11.002

18 Wikipedia contributors. (2024). *Learning factory*. En *Wikipedia*. Recuperado el 1 de octubre de 2025 de https://en.wikipedia.org/wiki/Learning_Factory

19 Real Games. (2024). *Features – Factory I/O*. Recuperado el 1 de octubre de 2025 de https://factoryio.com/features

20 Kramp, T., Van Kranenburg, R., & Lange, S. (2013). Chapter 1: Introduction to the Internet of Things. En D. Giusto, A. Iera, G. Morabito, & L. Atzori (Eds.), *The Internet of Things* (pp. 1–10). Springer. https://doi.org/10.1007/978-1-4614-5749-2_1

21 Kramp, T., Van Kranenburg, R., & Lange, S. (2013). Introduction to the Internet of Things. Recuperado de https://www.researchgate.net/publication/303726147_Introduction_to_the_Internet_of_Things

22 Latorre, L., Rego, E., & De Leo, L. (2025). *Reporte de tecnología: Internet de las cosas*. Banco Interamericano de Desarrollo. Recuperado de https://publications.iadb.org/es/reporte-de-tecnologia-internet-de-las-cosas

23 Vecteezy. (s. f.). *Optimizando el procesamiento de datos en IIoT con Node-R*. En *Vecteezy*. Recuperado de https://es.vecteezy.com/arte-vectorial/2298908-elementos-de-

interfaz-de-usuario-para-tecnologia-en-la-nube-aplicacion-movil-diseno-neumorphic-plantilla-elementos-de-interfaz-de-usuario

24 (s. f.). *Elementos de la interfaz de usuario para la aplicación móvil de tecnología en la nube (Plantilla de interfaz neumórfica)*. En *Vecteezy*. Recuperado de https://es.vecteezy.com/arte-vectorial/2298908-elementos-de-interfaz-de-usuario-para-tecnologia-en-la-nube-aplicacion-movil-diseno-neumorphic-plantilla-elementos-de-interfaz-de-usuario

25 Lluch, P. (2023). *La IoT industrial acelera la digitalización de América*. Orange Business. Recuperado de https://uat-ob.cdb.cdn.orange.com/es/blogs/el-industrial-iot-como-tractor-negocio-transformacion-digital-en-america-latina

26 AstralPool. (s. f.). *Guía protocolo Modbus 60363 KIT Modbus RTU ver. 1.3 rev. 0*. España: AstralPool. Recuperado de https://www.modbus.org

27. MQTT.org. (s. f.). *MQTT – The standard for IoT messaging*. Recuperado el 1 de octubre de 2025, de https://mqtt.org/messaging

28. HiveMQ. (s. f.). *MQTT essentials*. Recuperado el 1 de octubre de 2025, de https://www.hivemq.com/mqtt/

29. TechTarget. (s. f.). *SCADA (supervisory control and data acquisition)*. Recuperado el 1 de octubre de 2025, de https://www.techtarget.com/whatis/definition/SCADA-supervisory-control-and-data-acquisition

30. Prometheus Group. (s. f.). *What is SCADA system?*. Recuperado el 1 de octubre de 2025, de https://www.prometheusgroup.com/learning-center/what-is-scada-system

31 **City of Prairie Grove.** (2024, octubre). *FYI: New developments create a full time need for capacity development*. City of Prairie Grove.

32. Blevins, T., & Nixon, M. (2011). *Control loop foundation: Batch and continuous processes*. New York, NY: ISA.

33. St. Clair, D. W. (1993). *Practical process control: Tuning and troubleshooting (2nd ed.)*. USA: Straight-Line Control Co., Inc.

34. Smith, L. (2009). *Practical process control tuning and troubleshooting*. Hoboken, NJ: Wiley.

35. Dunn, W. C. (2006). *Introduction to instrumentation and process control*. Norwood, MA: Artech House.

36. Corripio, A. B. (2015). *Tuning of industrial control systems* (3rd ed.). Research Triangle Park, NC: ISA.

37. Kuphaldt, T. R. (2008–2022). *Lessons in industrial instrumentation*. Licensed under the International Public License. Recuperado de https://www.ibiblio.org/kuphaldt

38. Constitución Política de Colombia. (1991). *Constitución política de Colombia*. Bogotá, Colombia: Asamblea Nacional Constituyente.

39. Ministerio de Trabajo y Seguridad Social. (1979). *Resolución 2400 de 1979: Por la cual se establecen algunas disposiciones sobre vivienda, higiene y seguridad en los establecimientos de trabajo*. Recuperado de https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=53565

40. Ministerio de Minas y Energía de Colombia. (2004). *Reglamento Técnico de Instalaciones Eléctricas (RETIE)*. Recuperado de https://www.minenergia.gov.co/es/misional/energia-electrica-2/reglamentos-tecnicos/reglamento-t%C3%A9cnico-de-instalaciones-el%C3%A9ctricas-retie

Lammert Bies. (s. f.). *RS-232 serial null modem cable wiring and tutorial*. Recuperado de https://www.lammertbies.nl/comm/info/rs-232-null-modem

10. Anexos

10.1. Código ESP32

```
#include <ModbusMaster.h>
#include <WiFi.h>
#include <PubSubClient.h>

// ================== OBJETOS ==================
ModbusMaster node;
HardwareSerial mySerial(1);

// ================== CONFIG WIFI ==================
// const char* ssid = "Tu_Red_WiFi";       // <-- Nombre red (comentado)
// const char* password = "Tu_Password";   // <-- Contraseña red (comentado)

// ================== CONFIG MQTT ==================
const char* mqtt_server = "98ad036b0a2d47119b8e8be5b737200b.s1.eu.hivemq.cloud";
const int mqtt_port = 1883;  // sin TLS

WiFiClient espClient;
PubSubClient client(espClient);
```

```
// ================== TOPICS MQTT ==================
const char* outTopicFlujo      = "SEDAI/Flujo";
const char* outTopicNivel      = "SEDAI/Nivel";
const char* outTopicTemp       = "SEDAI/Temperatura";

const char* outTopicRadiador   = "SEDAI/Radiador";
const char* outTopicBomba      = "SEDAI/Bomba";
const char* outTopicPiloto1    = "SEDAI/Piloto1";
const char* outTopicPiloto2    = "SEDAI/Piloto2";
const char* outTopicPiloto3    = "SEDAI/Piloto3";

// ================== SENSOR DE FLUJO ==================
volatile int contadorPulsos = 0;
const byte pinSensor = 25;
unsigned long tiempoAnterior = 0;
float factorCalibracion = 7.5;
int flujo = 0;

void IRAM_ATTR contarPulsos() {
  contadorPulsos++;
}

// ================== WIFI ==================
void setup_wifi() {
  Serial.println();
  Serial.println("Conectando a WiFi...");
  // WiFi.begin(ssid, password);   // <-- Línea comentada
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("\nWiFi conectada.");
  Serial.print("IP: "); Serial.println(WiFi.localIP());
}

// ================== RECONNECT MQTT ==================
void reconnect() {
  while (!client.connected()) {
    Serial.print("Conectando a MQTT...");
    if (client.connect("ESP32_SEDAI")) {
      Serial.println("Conectado.");
    } else {
      Serial.print("Fallo, rc=");
      Serial.print(client.state());
      Serial.println(" Reintentando en 5s...");
      delay(5000);
```

```
    }
  }
}

// ================== SETUP ==================
void setup() {
  Serial.begin(9600);
  pinMode(pinSensor, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(pinSensor), contarPulsos, RISING);

  // Puerto UART2 para Modbus
  Serial2.begin(9600, SERIAL_8N1, 16, 17);
  Serial2.flush();
  node.begin(2, Serial2);

  setup_wifi();
  client.setServer(mqtt_server, mqtt_port);

  Serial.println("Sistema inicializado...");
}

// ================== LOOP ==================
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

  // ================== SENSOR DE FLUJO ==================
  unsigned long tiempoActual = millis();
  if (tiempoActual - tiempoAnterior >= 1000) {
    detachInterrupt(digitalPinToInterrupt(pinSensor));
    float frecuencia = contadorPulsos;
    float caudal = frecuencia / factorCalibracion;
    flujo = caudal * 10;
    contadorPulsos = 0;
    tiempoAnterior = tiempoActual;
    attachInterrupt(digitalPinToInterrupt(pinSensor), contarPulsos, RISING);

    // Enviar al PLC (ejemplo MI6)
    node.writeSingleRegister(6, flujo);

    Serial.print("Flujo calculado ESP32: "); Serial.println(flujo);
    client.publish(outTopicFlujo, String(flujo).c_str(), true);
  }
```

```
 // ================== LECTURA NIVEL Y TEMPERATURA ==================
  uint8_t result = node.readHoldingRegisters(47, 2);
  if (result == node.ku8MBSuccess) {
    float nivel = node.getResponseBuffer(0) / 10.0;
    float temperatura = node.getResponseBuffer(1) / 10.0;

    Serial.print("Nivel: "); Serial.println(nivel);
    Serial.print("Temperatura: "); Serial.println(temperatura);

    client.publish(outTopicNivel, String(nivel).c_str(), true);
    client.publish(outTopicTemp, String(temperatura).c_str(), true);
  } else {
    Serial.println("Error lectura variables analógicas.");
  }

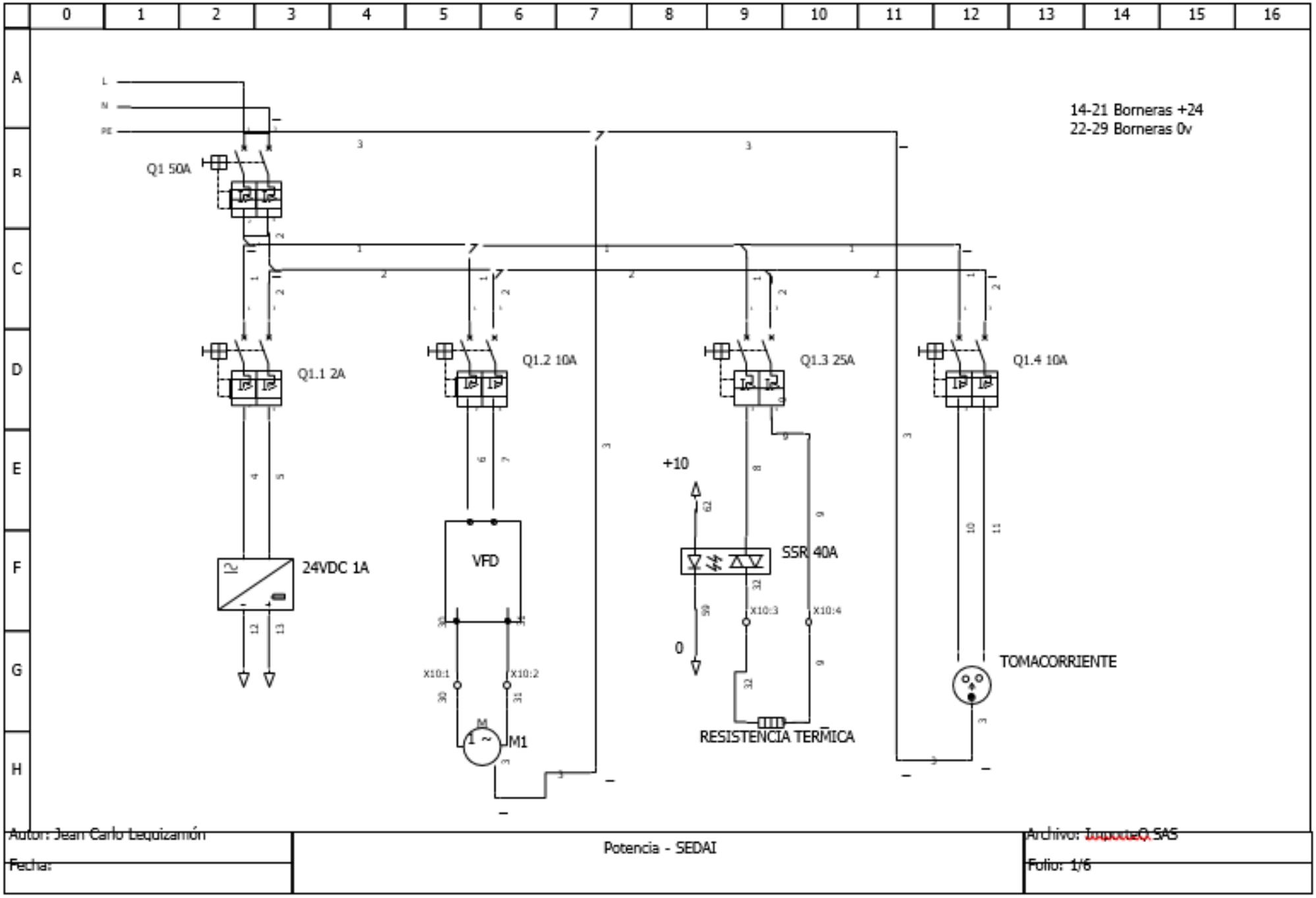
  // ================== LECTURA COILS ==================
  uint8_t resultCoils = node.readCoils(13, 5);
  if (resultCoils == node.ku8MBSuccess) {
    bool radiador = node.getResponseBuffer(0) & 0x01;
    bool bomba    = node.getResponseBuffer(0) & 0x02;
    bool piloto1  = node.getResponseBuffer(0) & 0x04;
    bool piloto2  = node.getResponseBuffer(0) & 0x08;
    bool piloto3  = node.getResponseBuffer(0) & 0x10;

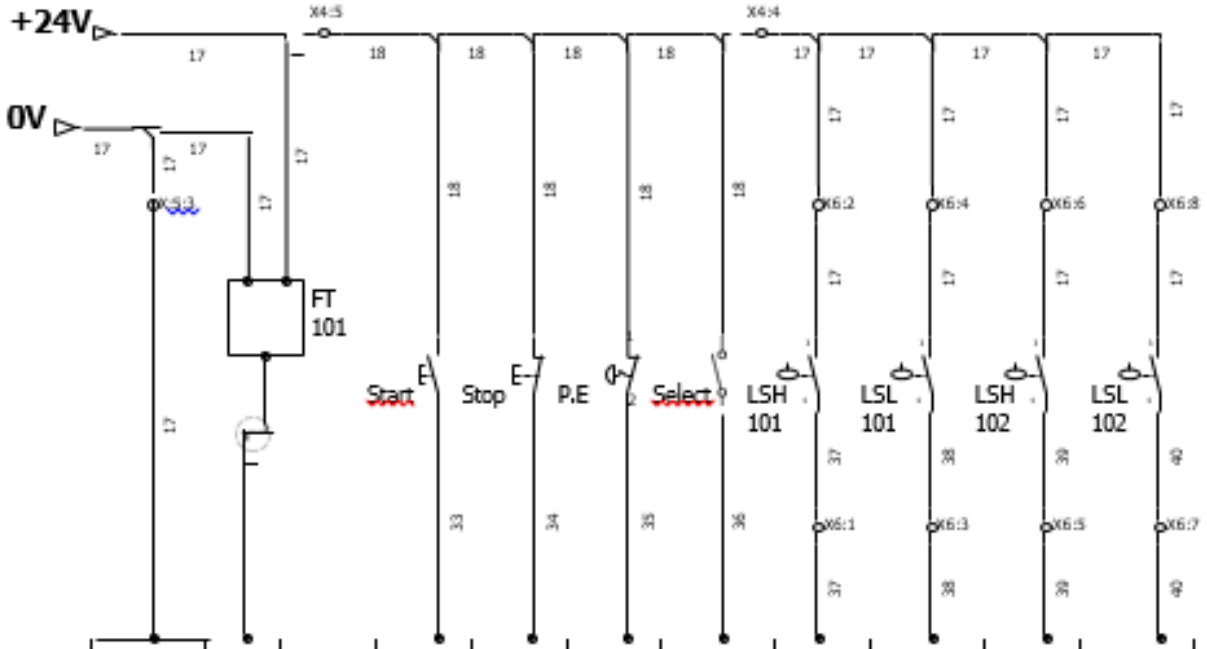
    Serial.print("Radiador: "); Serial.println(radiador);
    Serial.print("Bomba: "); Serial.println(bomba);
    Serial.print("Piloto1: "); Serial.println(piloto1);
    Serial.print("Piloto2: "); Serial.println(piloto2);
    Serial.print("Piloto3: "); Serial.println(piloto3);

    client.publish(outTopicRadiador, String(radiador).c_str(), true);
    client.publish(outTopicBomba, String(bomba).c_str(), true);
    client.publish(outTopicPiloto1, String(piloto1).c_str(), true);
    client.publish(outTopicPiloto2, String(piloto2).c_str(), true);
    client.publish(outTopicPiloto3, String(piloto3).c_str(), true);
  } else {
    Serial.println("Error lectura de coils.");
  }

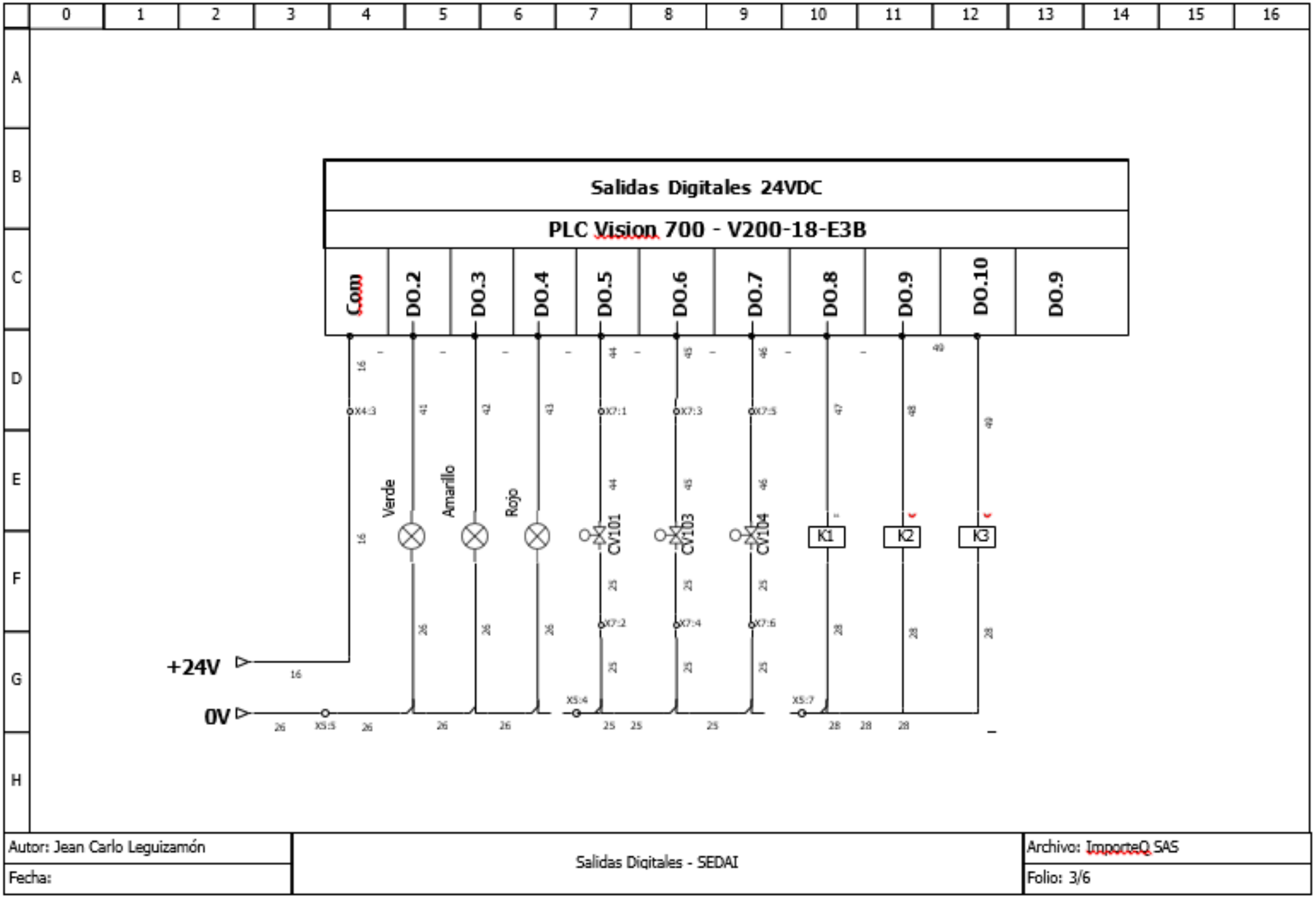
  delay(2000);
}
```

10.2. Planos Eléctricos

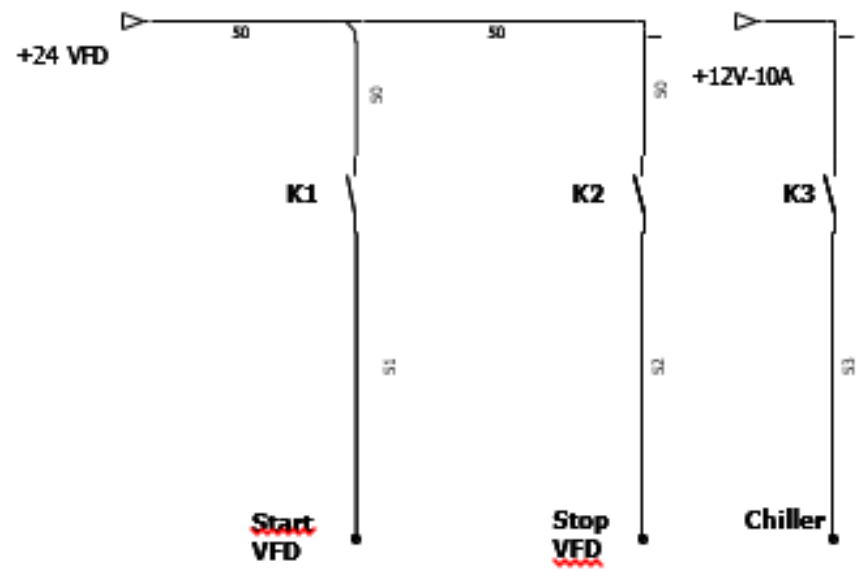

14-21 Borneras +24
22-29 Borneras 0v
L
N
PE
Q1 50A
Q1.1 2A
Q1.2 10A
Q1.3 25A
Q1.4 10A
24VDC 1A
VFD
M1
+10
0
SSR 40A
RESISTENCIA TERMICA
TOMACORRIENTE
X10:1
X10:2
X10:3
X10:4
Autor: Jean Carlo Lequizamón
Fecha:
Potencia - SEDAI
Archivo: InputteQ.SAS
Folio: 1/6



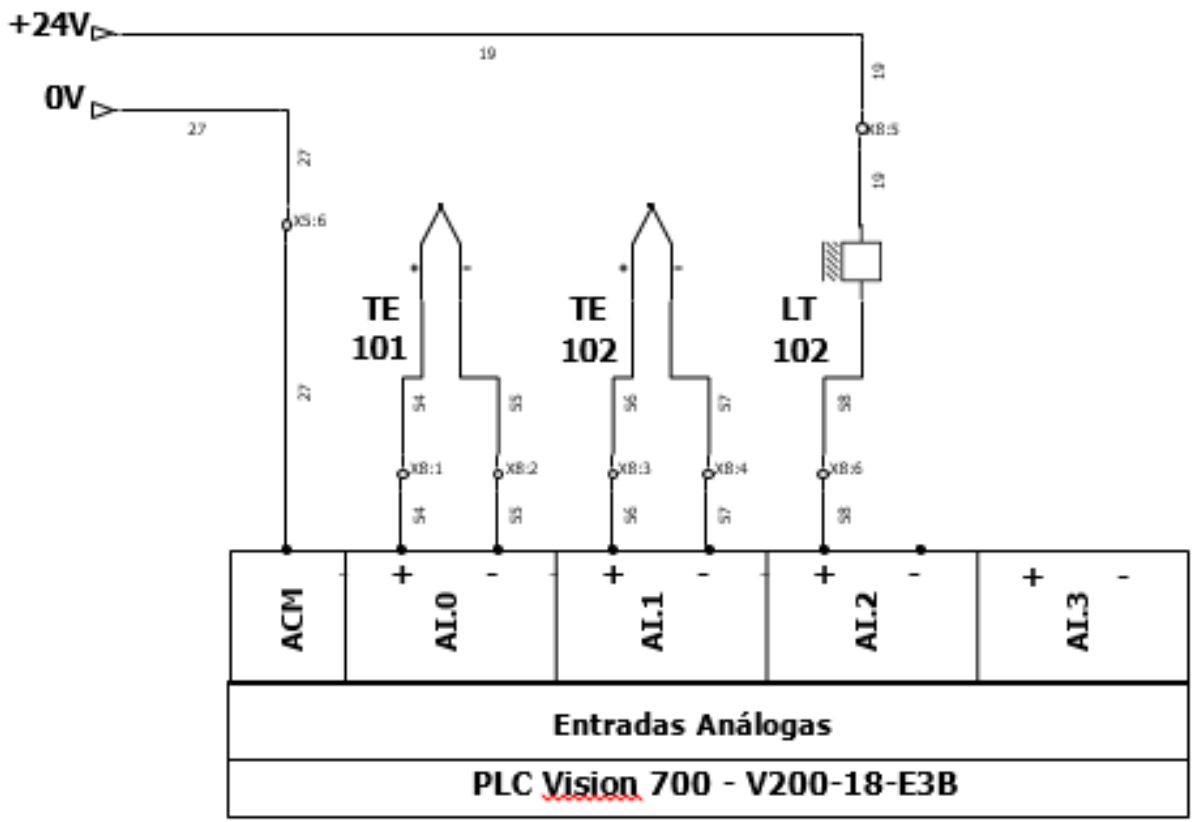
Autor: Jean Carlo Leguizamón	Entradas Digitales - SEDAI	Archivo: ImporteQ SAS
Fecha:		Folio: 2/6



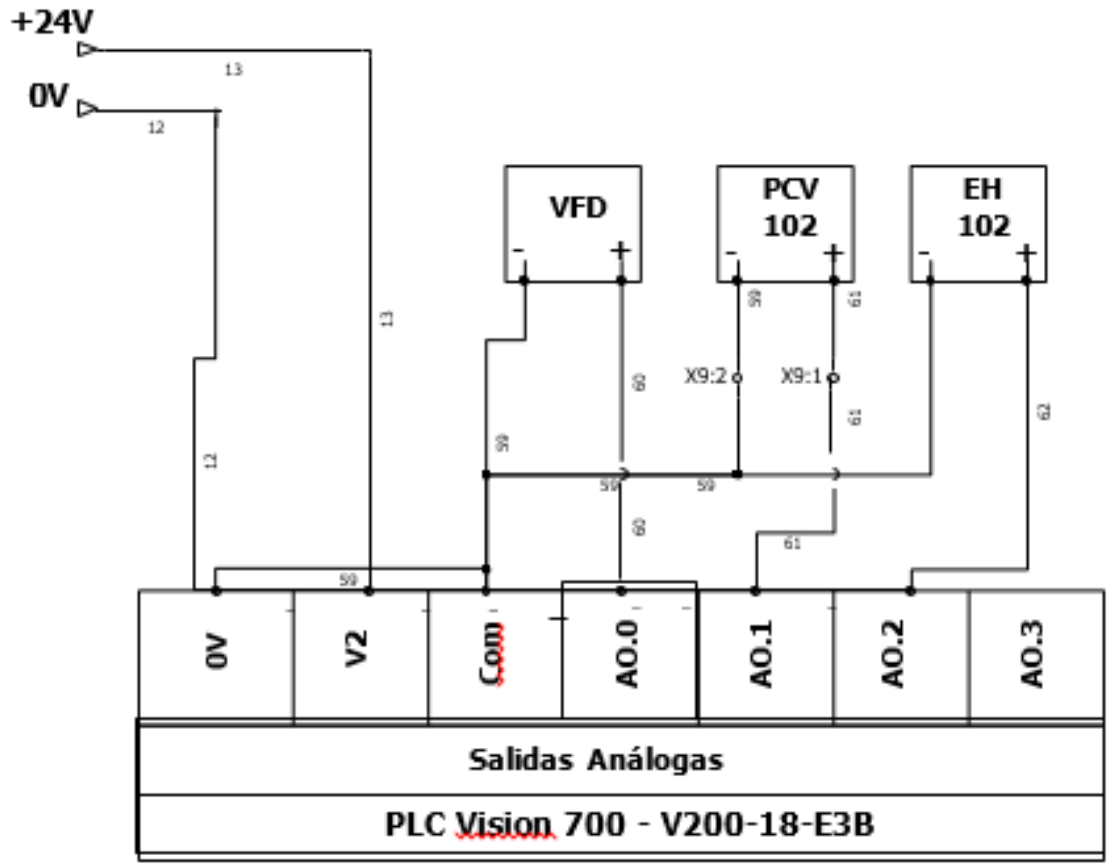
Autor: Jean Carlo Leguizamón	Salidas Digitales - SEDAI	Archivo: ImporteQ.SAS
Fecha:		Folio: 3/6



Autor: Jean Carlo Leguizamón	Salidas Digitales 2 - SEDAI	Archivo: ImporteQ SAS
Fecha:		Folio: 4/6



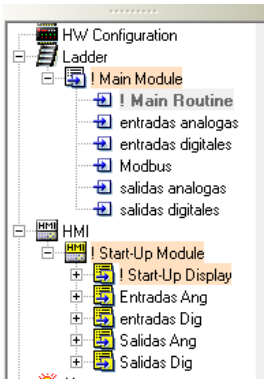
Autor: Jean Carlo Leguizamón	Entradas Analogas - SEDAI	Archivo: ImporteQ SAS
Fecha:		Folio: 5/6



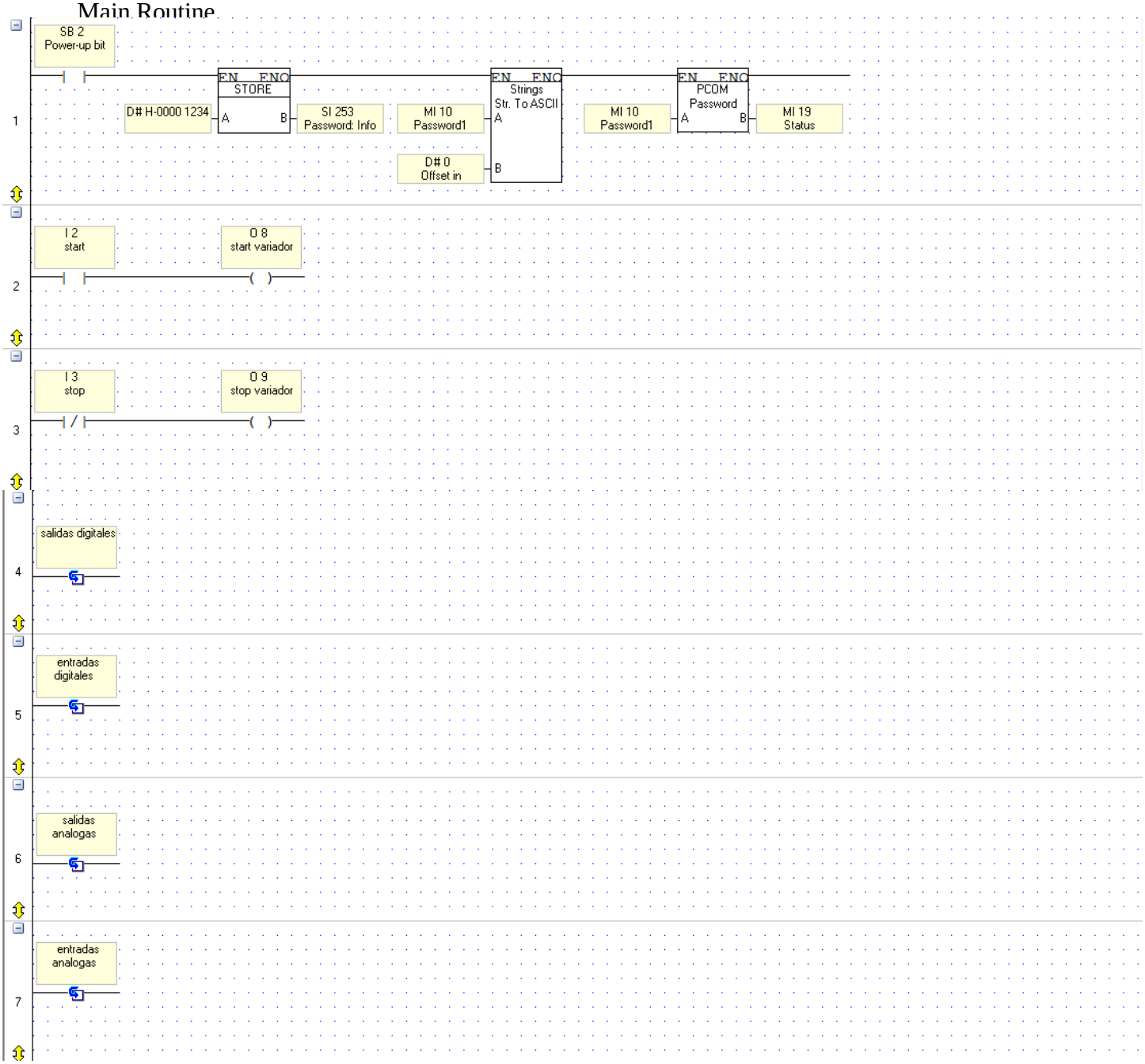
Autor: Jean Carlo Leguizamón	Salidas Analogas - SEDAI	Archivo: ImporteQ SAS
Fecha:		Folio: 6/6

10.3. Códigos Visilogic

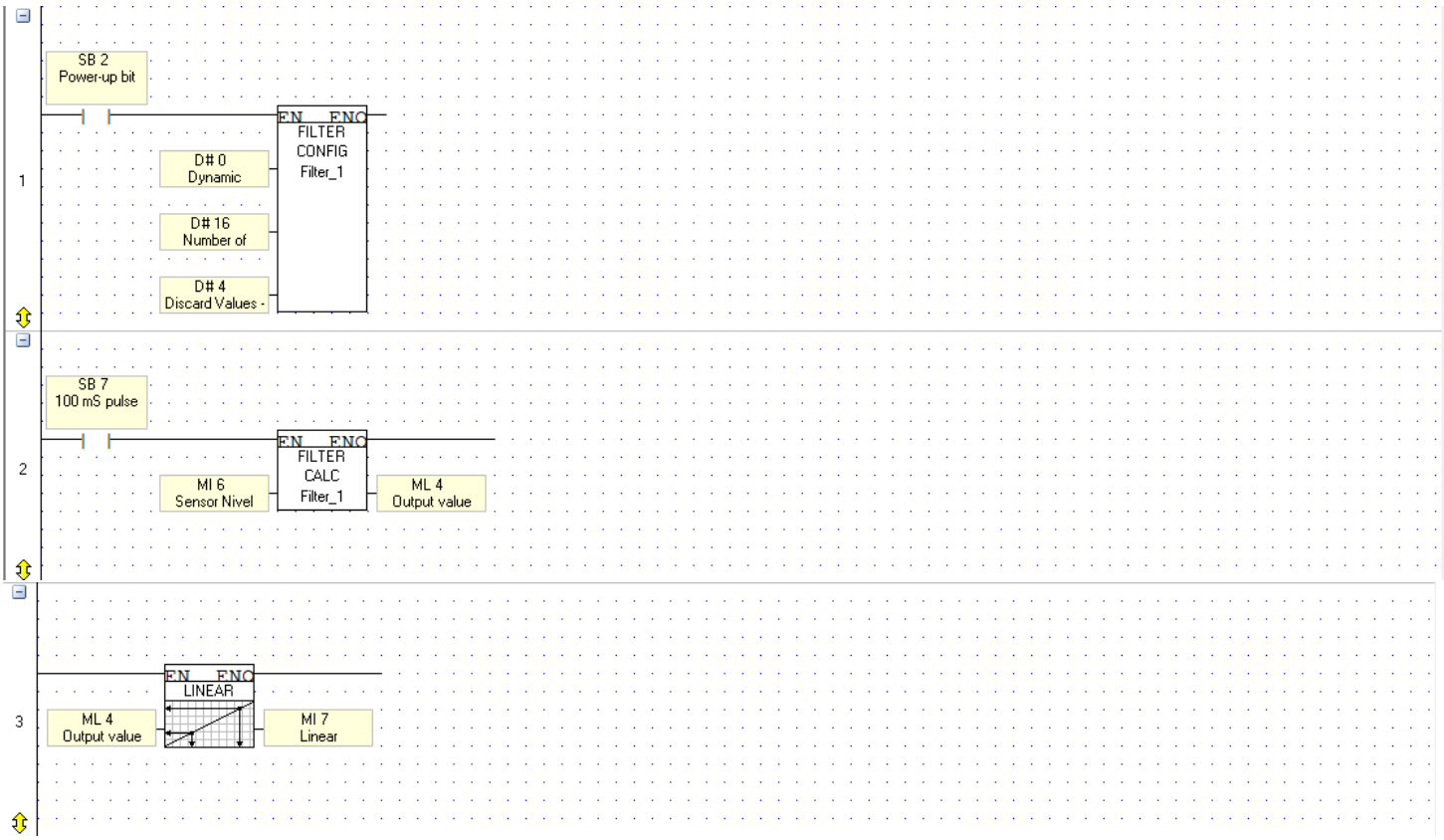
10.3.1. 1) Inicio- Monitoreo y control de señales



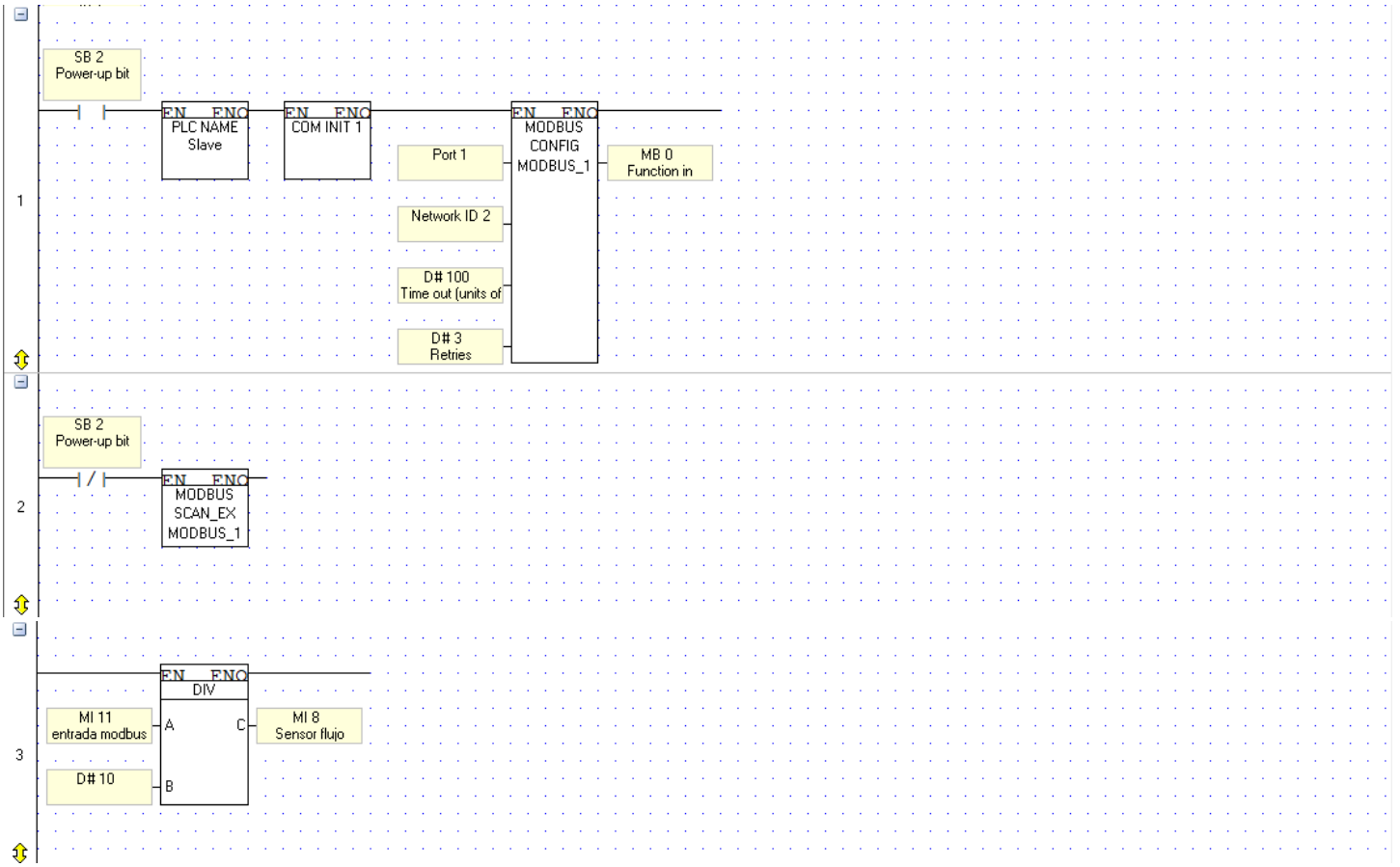
Main Routine



Entradas Análogas



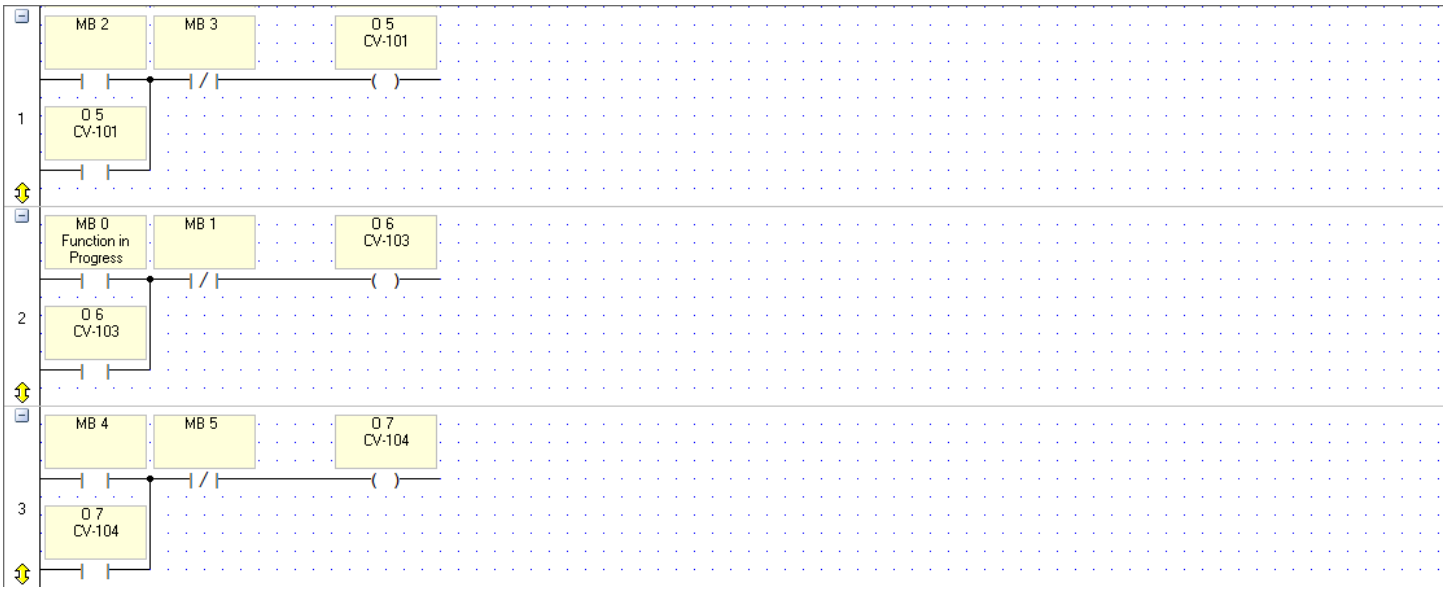
Modbus



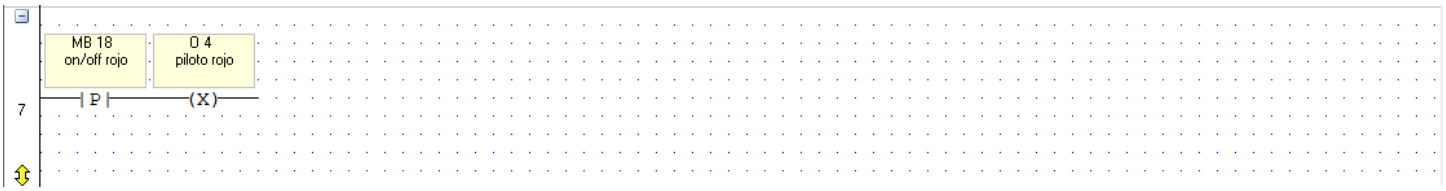
Salidas Análogas



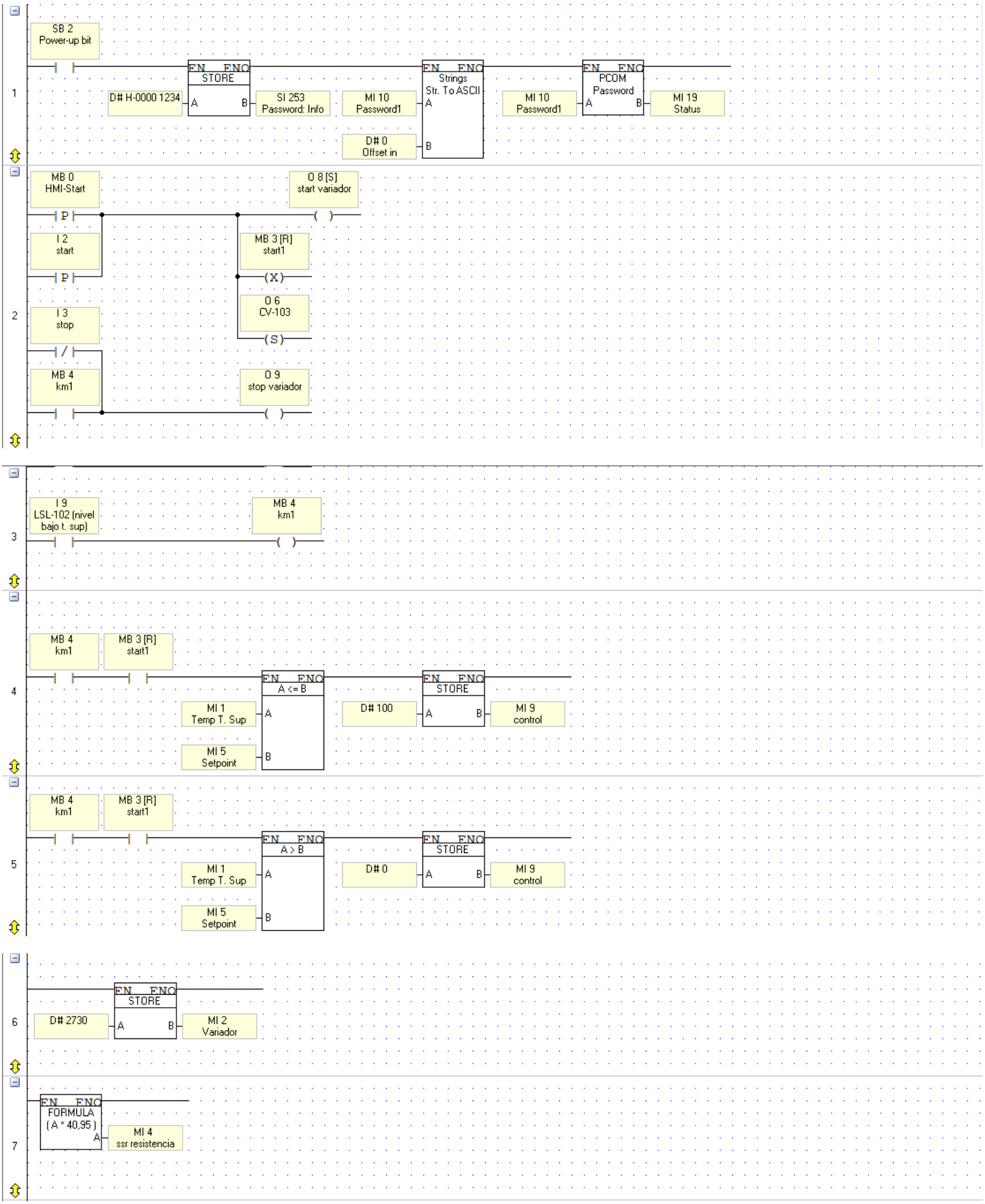
Salidas Digitale



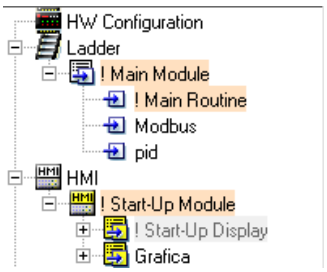



MB 18
on/off rojo
O 4
piloto rojo
7
P
(X)

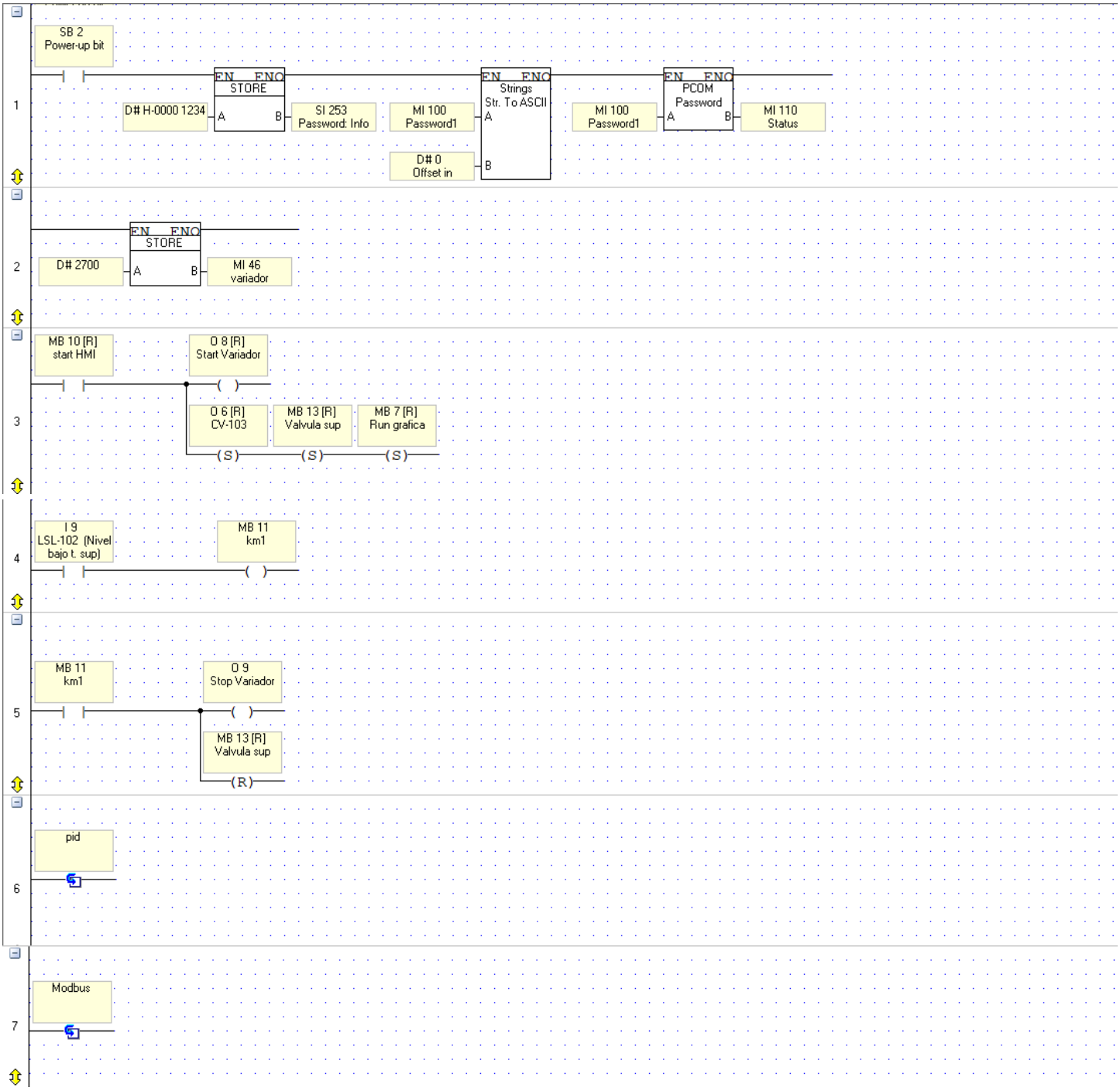
10.3.2. 2) Control ON/OFF temperatura



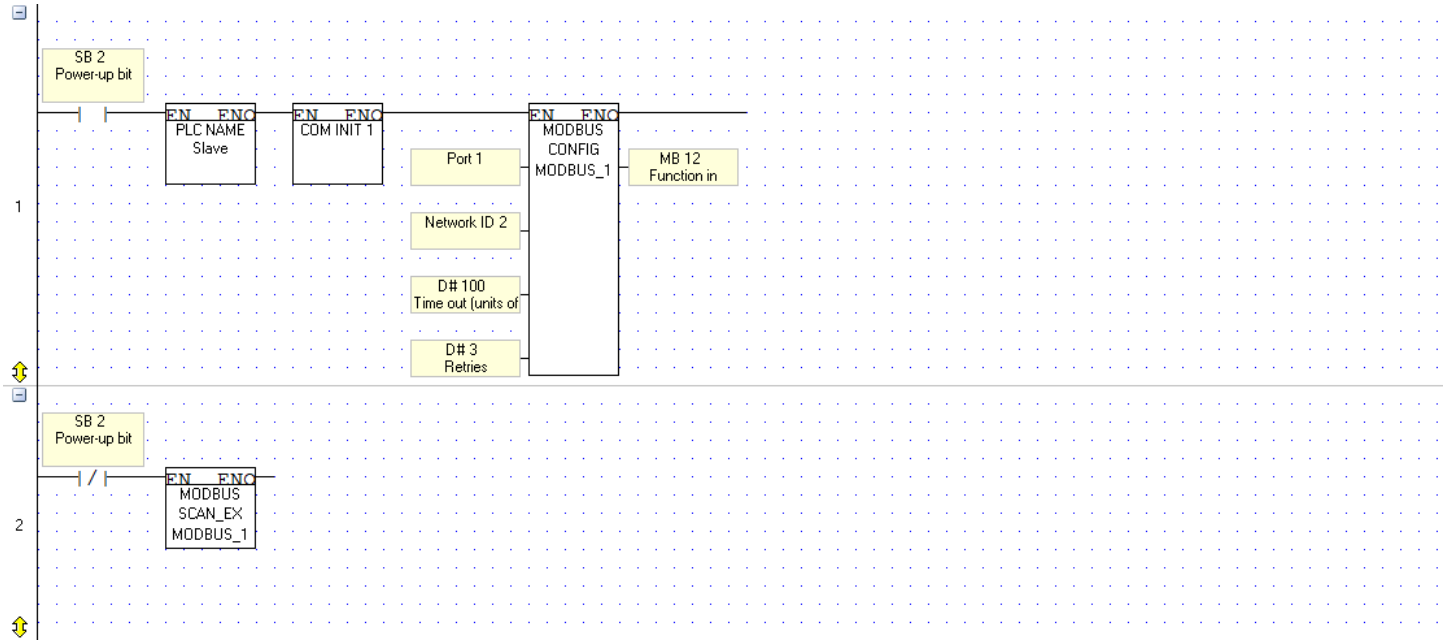
10.3.3. 3) Control PID Temperatura



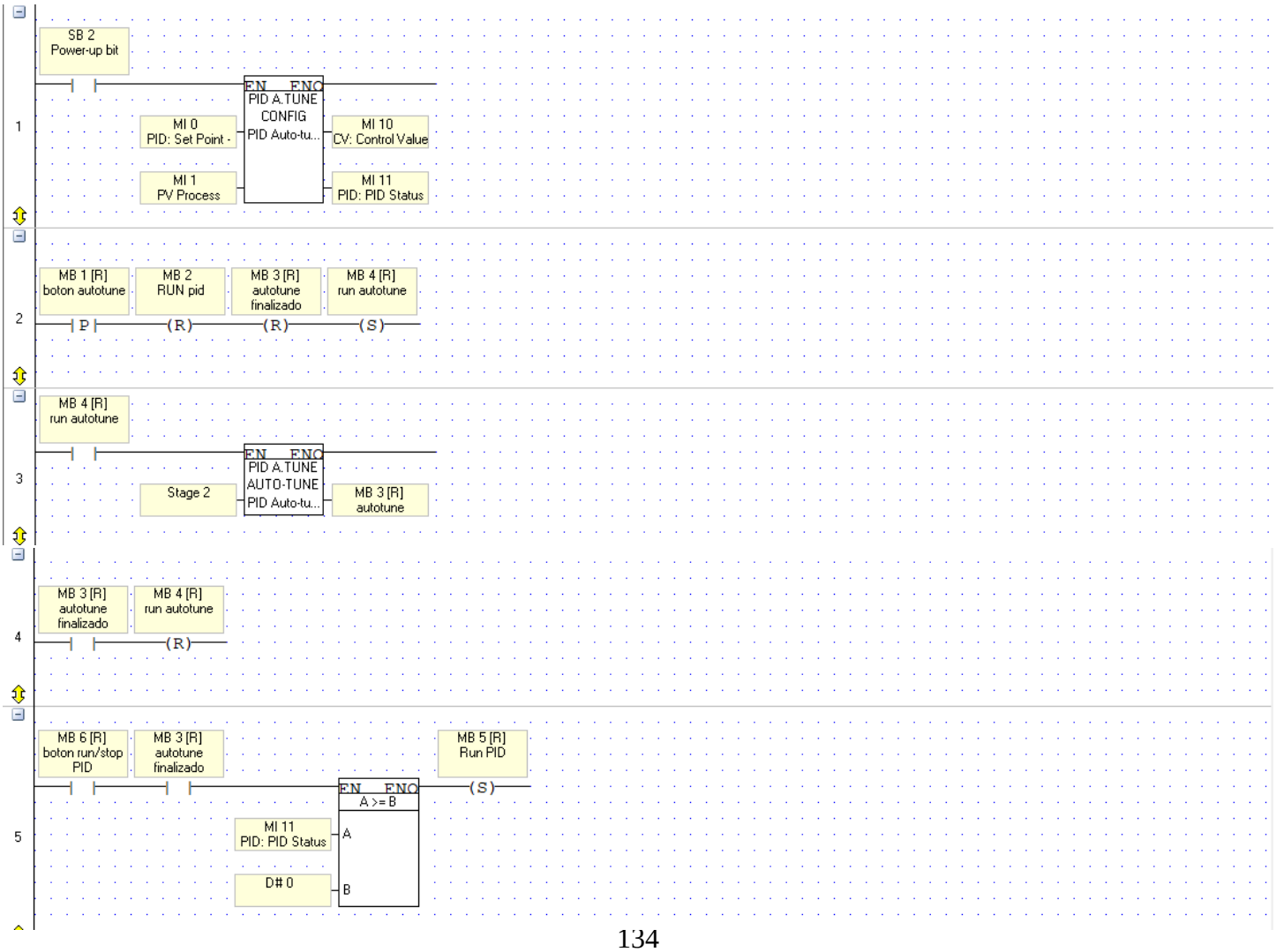
Main Routine



Modbus

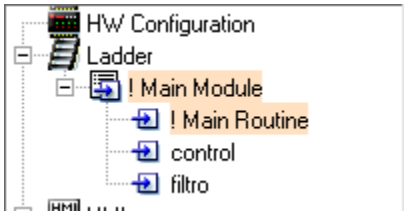

SB 2
Power-up bit
EN ENO
PLC NAME
Slave
EN ENO
COM INIT 1
Port 1
Network ID 2
D# 100
Time out (units of
D# 3
Retries
EN ENO
MODBUS
CONFIG
MODBUS_1
MB 12
Function in
1
SB 2
Power-up bit
EN ENO
MODBUS
SCAN_EX
MODBUS_1
2

PID

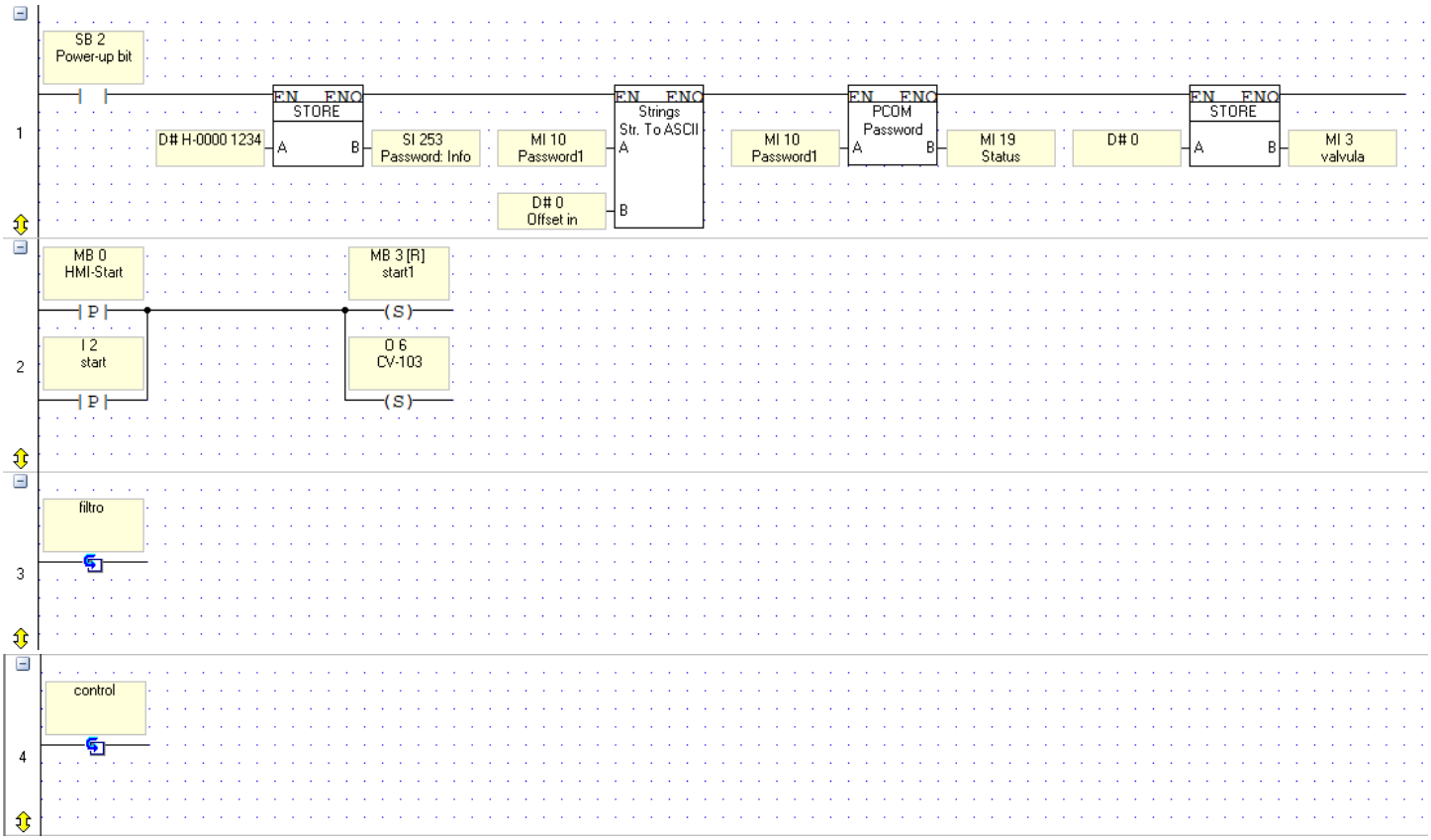

SB 2
Power-up bit
EN ENO
PID A.TUNE
CONFIG
PID Auto-tu...
MI 0
PID: Set Point -
MI 1
PV Process
MI 10
CV: Control Value
MI 11
PID: PID Status
1
MB 1 [R]
boton autotune
MB 2
RUN pid
MB 3 [R]
autotune
finalizado
MB 4 [R]
run autotune
P
(R)
(R)
(S)
2
MB 4 [R]
run autotune
EN ENO
PID A.TUNE
AUTO-TUNE
PID Auto-tu...
Stage 2
MB 3 [R]
autotune
3
MB 3 [R]
autotune
finalizado
MB 4 [R]
run autotune
(R)
4
MB 6 [R]
boton run/stop
PID
MB 3 [R]
autotune
finalizado
MB 5 [R]
Run PID
(S)
EN ENO
A >= B
MI 11
PID: PID Status
A
D# 0
B
5


MB 5 [R]
Run PID
EN ENO
PID A.TUNE
RUN
PID Auto-tu...
6
MB 11
km1
EN ENO
FORMULA
(A * 40,95)
A
MI 44
ssr resistencia
7

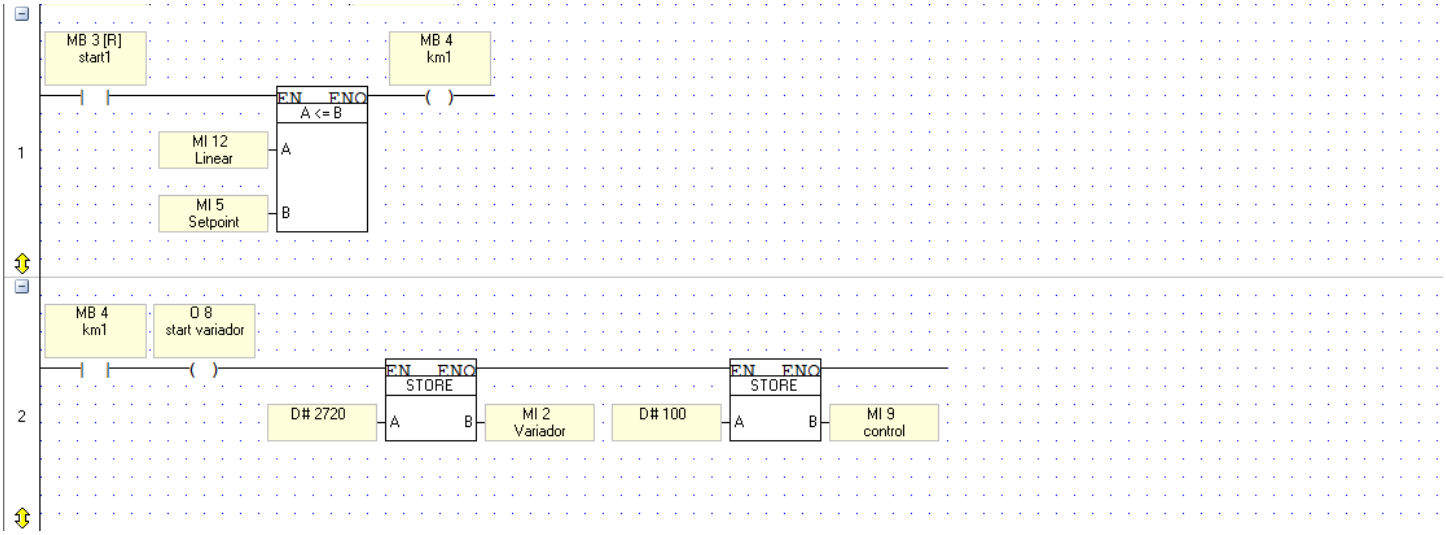
10.3.4. 4) Control ON/OFF Nivel

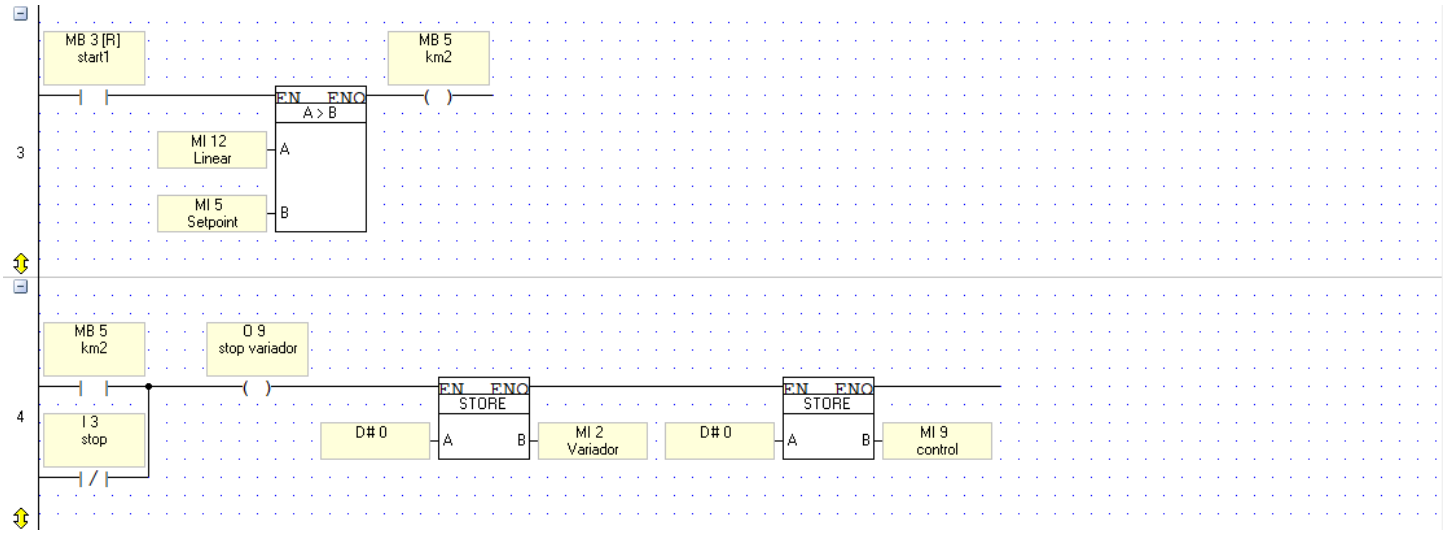


Main Routine

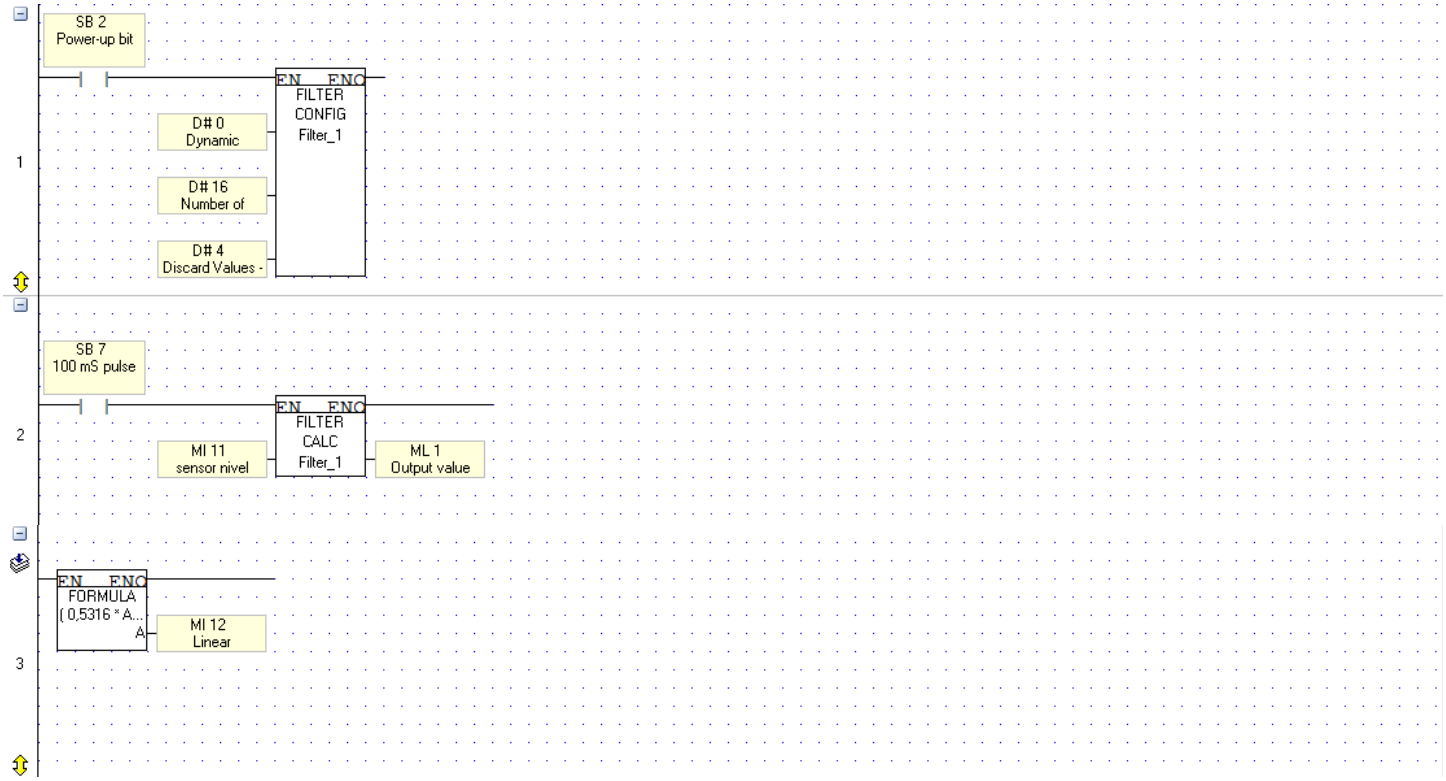


Control

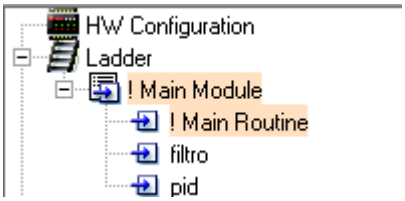



MB 3 [R]
start1
MB 5
km2
EN ENO
A > B
MI 12
Linear
A
MI 5
Setpoint
B
3
MB 5
km2
O 9
stop variador
I 3
stop
4
EN ENO
STORE
D# 0
A
B
MI 2
Variador
EN ENO
STORE
D# 0
A
B
MI 9
control

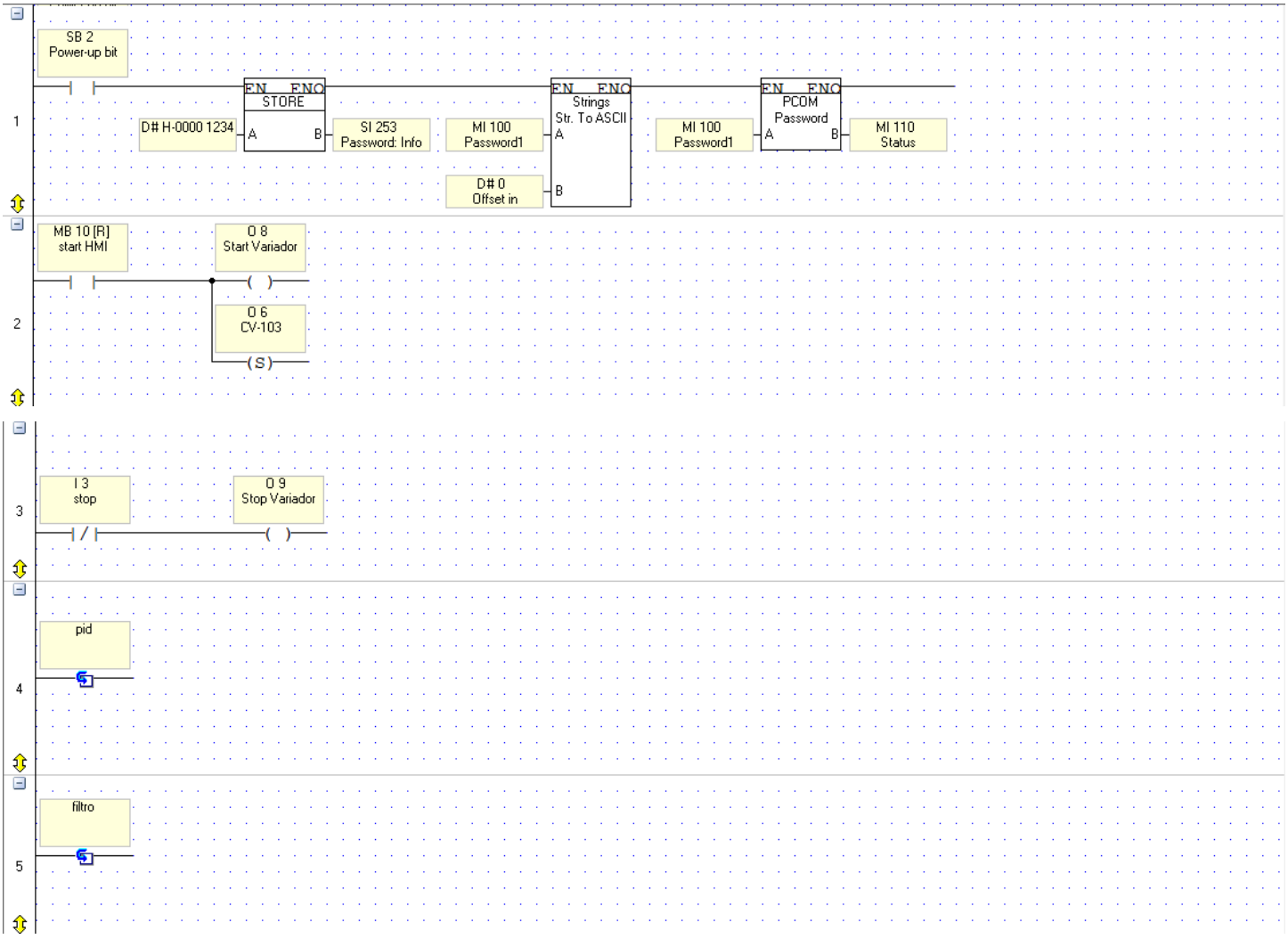
Filtro


SB 2
Power-up bit
EN ENO
FILTER
CONFIG
Filter_1
D# 0
Dynamic
D# 16
Number of
D# 4
Discard Values -
1
SB 7
100 mS pulse
EN ENO
FILTER
CALC
Filter_1
MI 11
sensor nivel
ML 1
Output value
2
EN ENO
FORMULA
(0,5316 * A...
A
MI 12
Linear
3

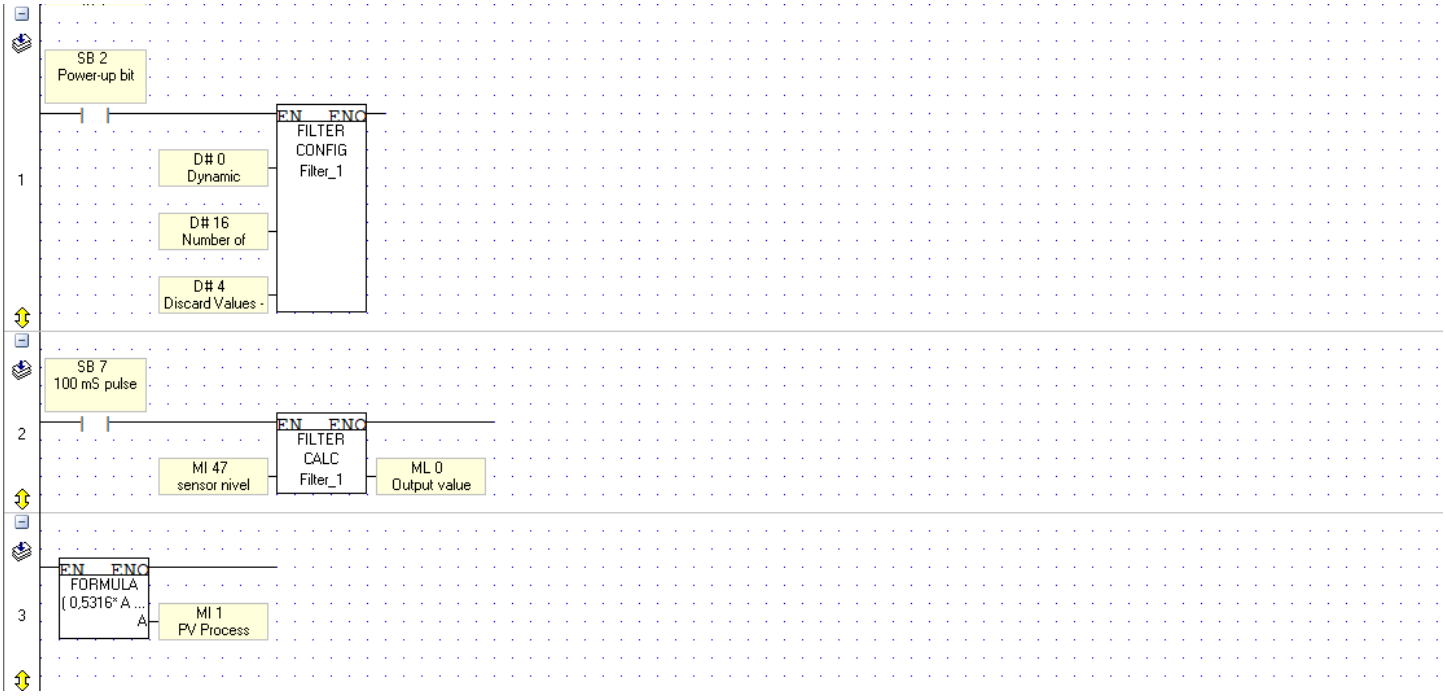
10.3.5. 5) Control PID Nivel



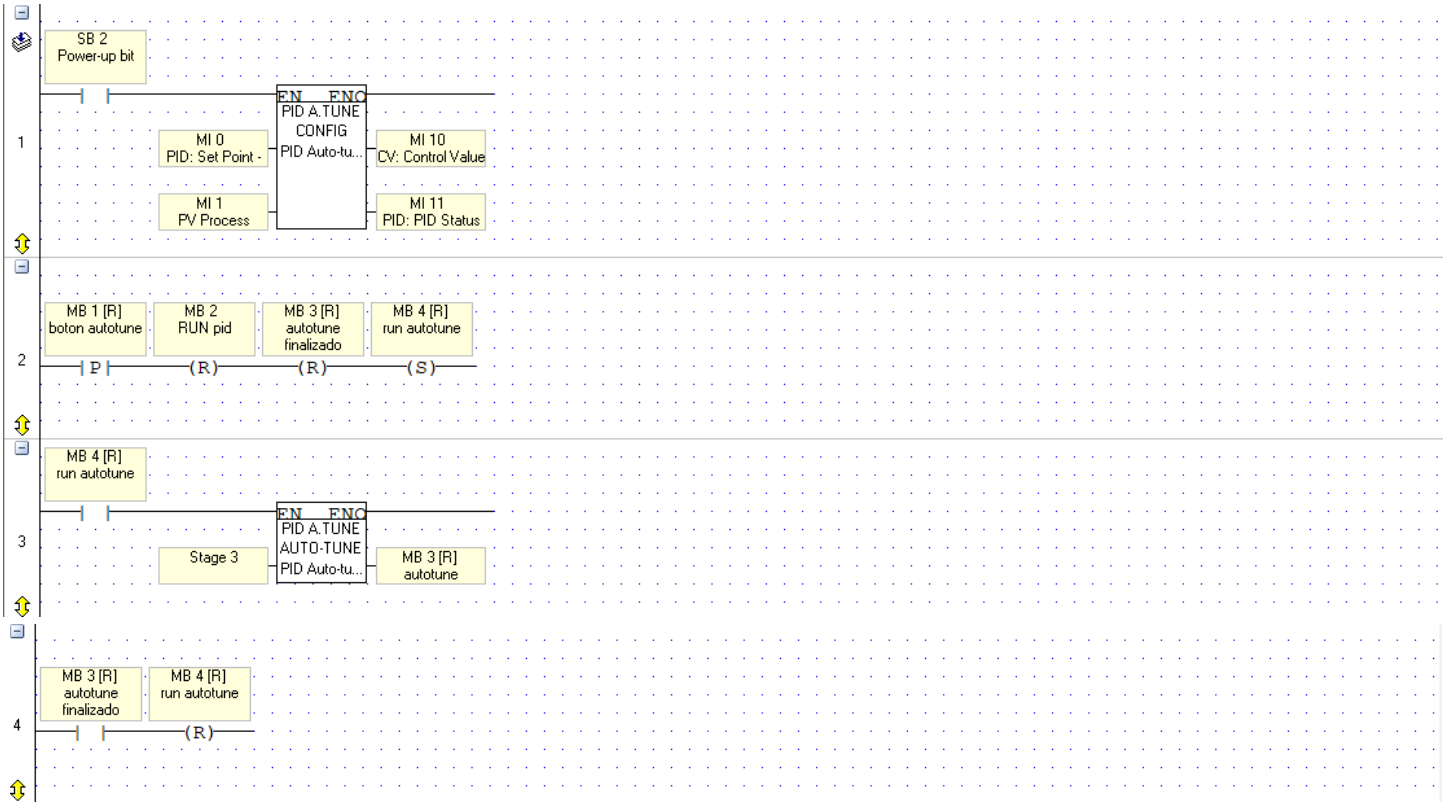
Main Routine

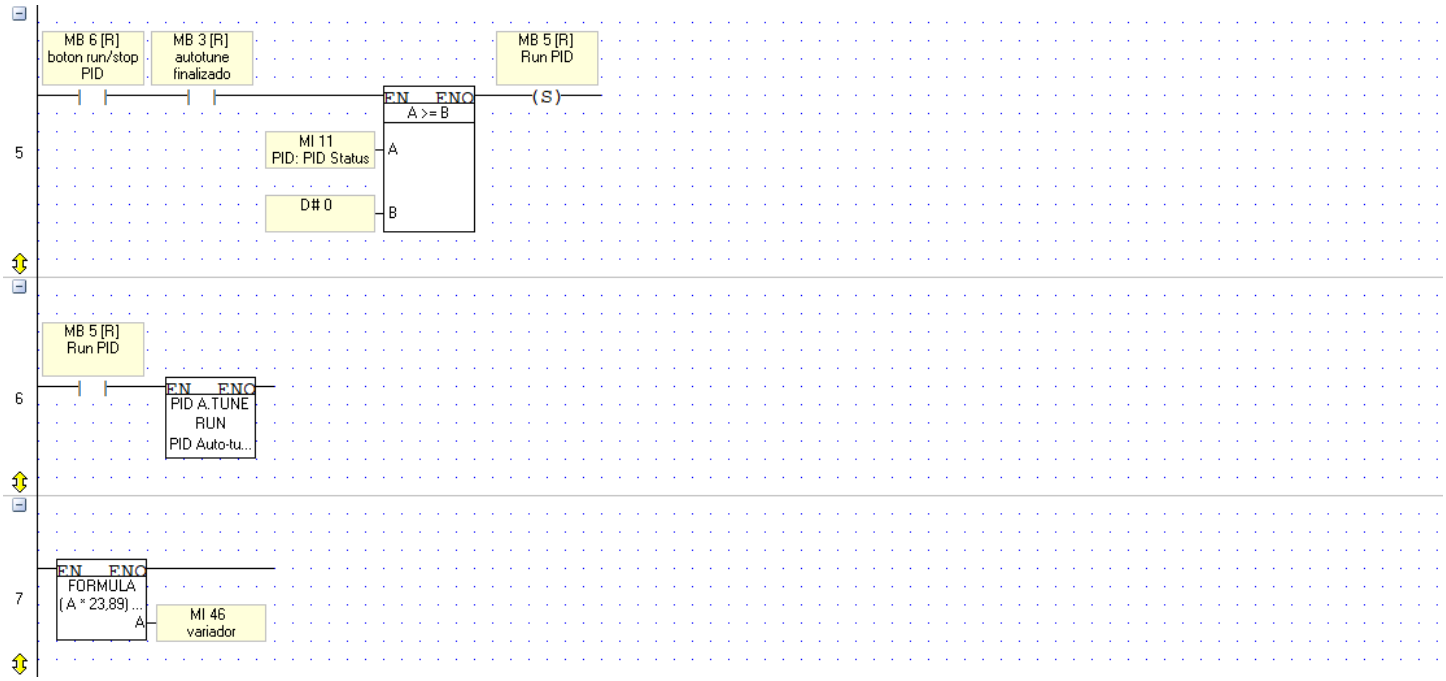


Filtro

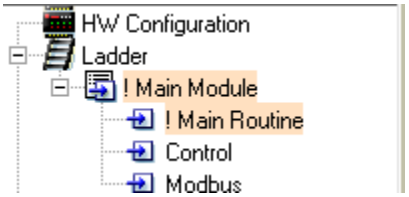

SB 2
Power-up bit
EN ENO
FILTER
CONFIG
Filter_1
D# 0
Dynamic
D# 16
Number of
D# 4
Discard Values -
1
SB 7
100 mS pulse
EN ENO
FILTER
CALC
Filter_1
MI 47
sensor nivel
ML 0
Output value
2
EN ENO
FORMULA
(0,5316* A ...
A
MI 1
PV Process
3

PID

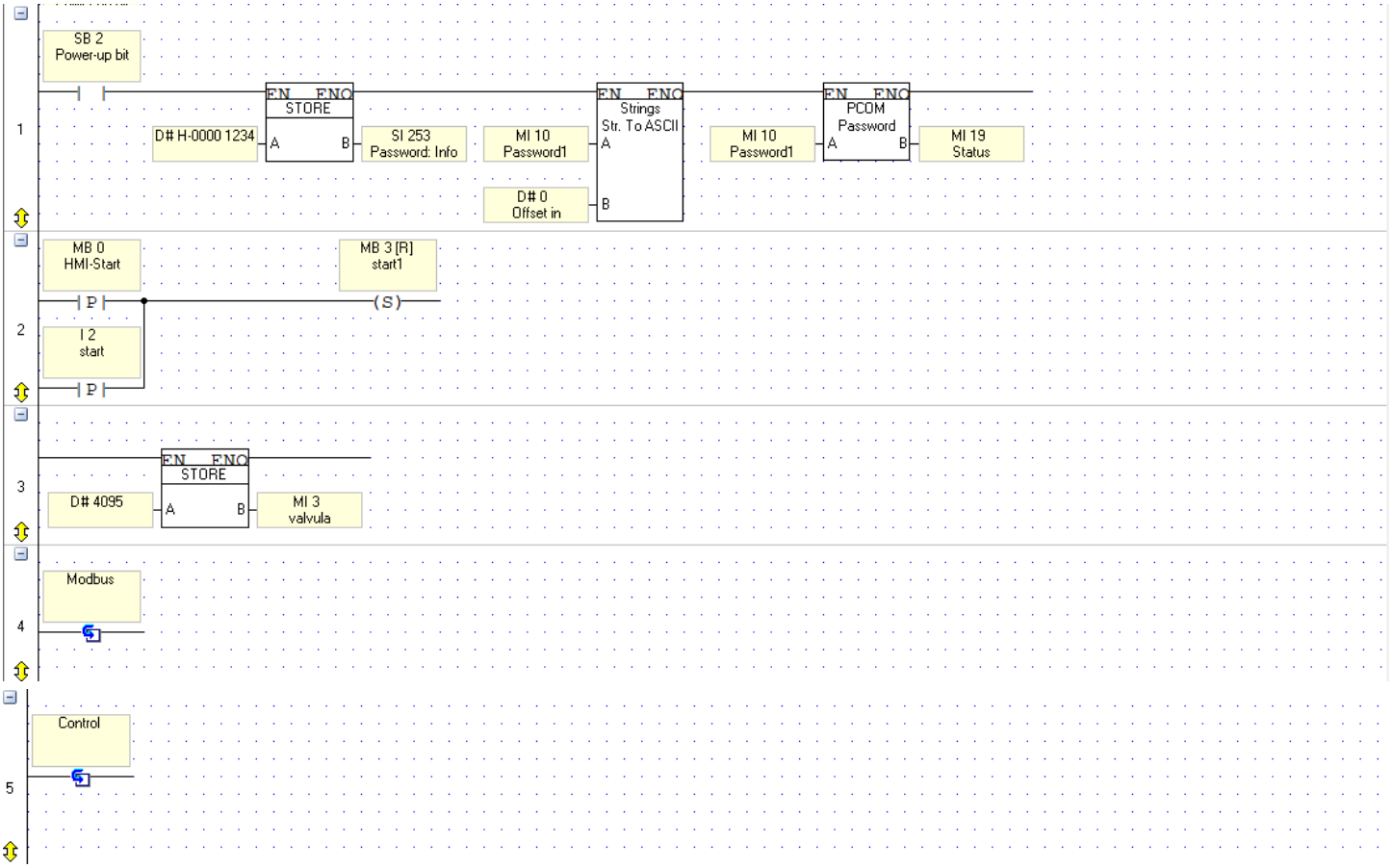

SB 2
Power-up bit
EN ENO
PID A.TUNE
CONFIG
PID Auto-tu...
MI 0
PID: Set Point -
MI 1
PV Process
MI 10
CV: Control Value
MI 11
PID: PID Status
1
MB 1 [R]
boton autotune
MB 2
RUN pid
MB 3 [R]
autotune
finalizado
MB 4 [R]
run autotune
P
(R)
(R)
(S)
2
MB 4 [R]
run autotune
EN ENO
PID A.TUNE
AUTO-TUNE
PID Auto-tu...
Stage 3
MB 3 [R]
autotune
3
MB 3 [R]
autotune
finalizado
MB 4 [R]
run autotune
(R)
4


MB 6 [R]
boton run/stop PID
MB 3 [R]
autotune finalizado
MB 5 [R]
Run PID
EN ENO
A >= B
(S)
5
MI 11
PID: PID Status
A
D# 0
B
MB 5 [R]
Run PID
6
EN ENO
PID A.TUNE
RUN
PID Auto-tu...
EN ENO
FORMULA
(A * 23,89) ...
A
7
MI 46
variador

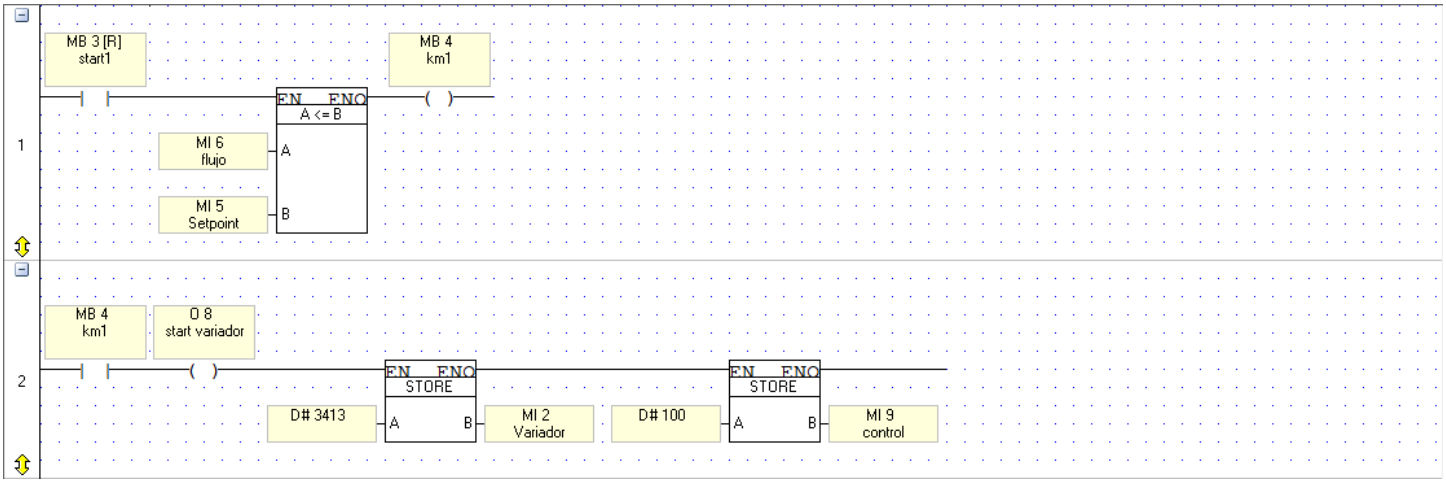
10.3.6. 6) Control ON/OFF flujo

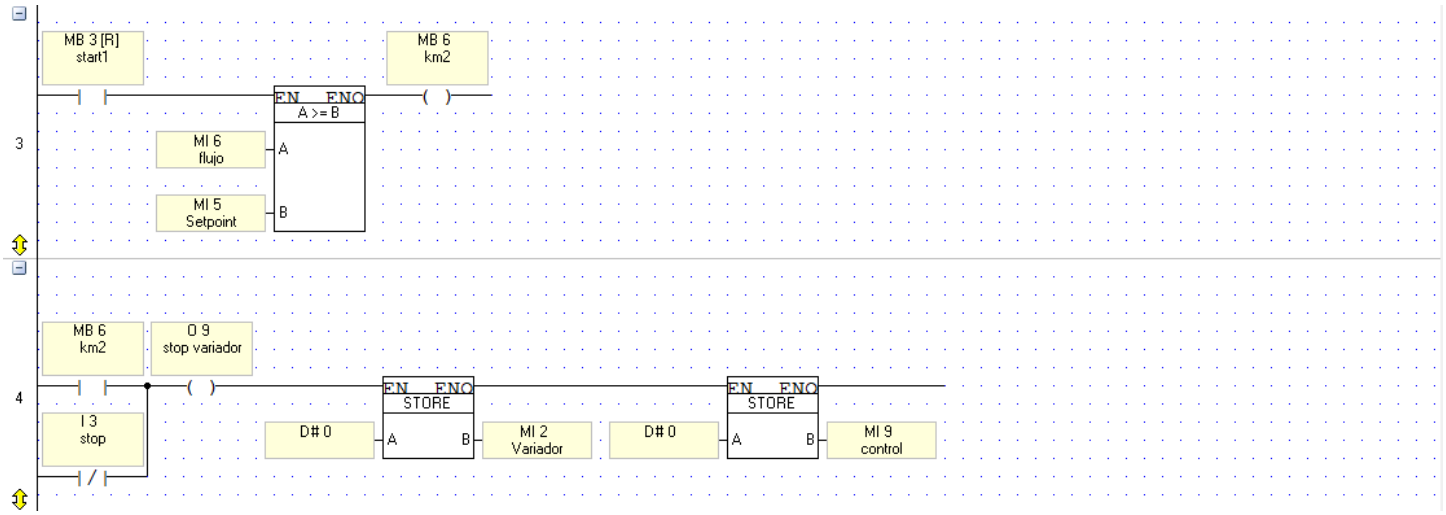


Main Routine

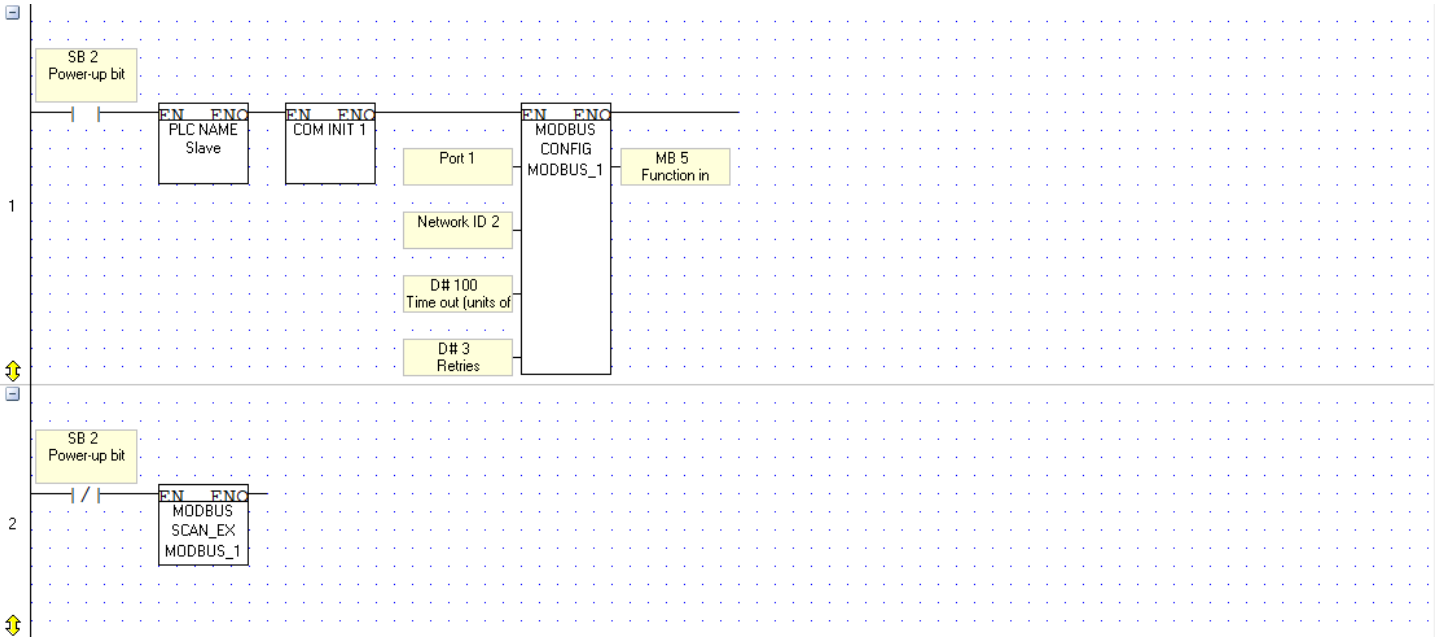


Control

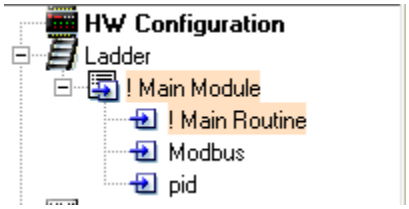



MB 3 [R]
start1
MB 6
km2
EN ENO
A >= B
3
MI 6
flujo
A
MI 5
Setpoint
B
MB 6
km2
O 9
stop variador
4
I 3
stop
D# 0
EN ENO
STORE
A
B
MI 2
Variador
D# 0
EN ENO
STORE
A
B
MI 9
control

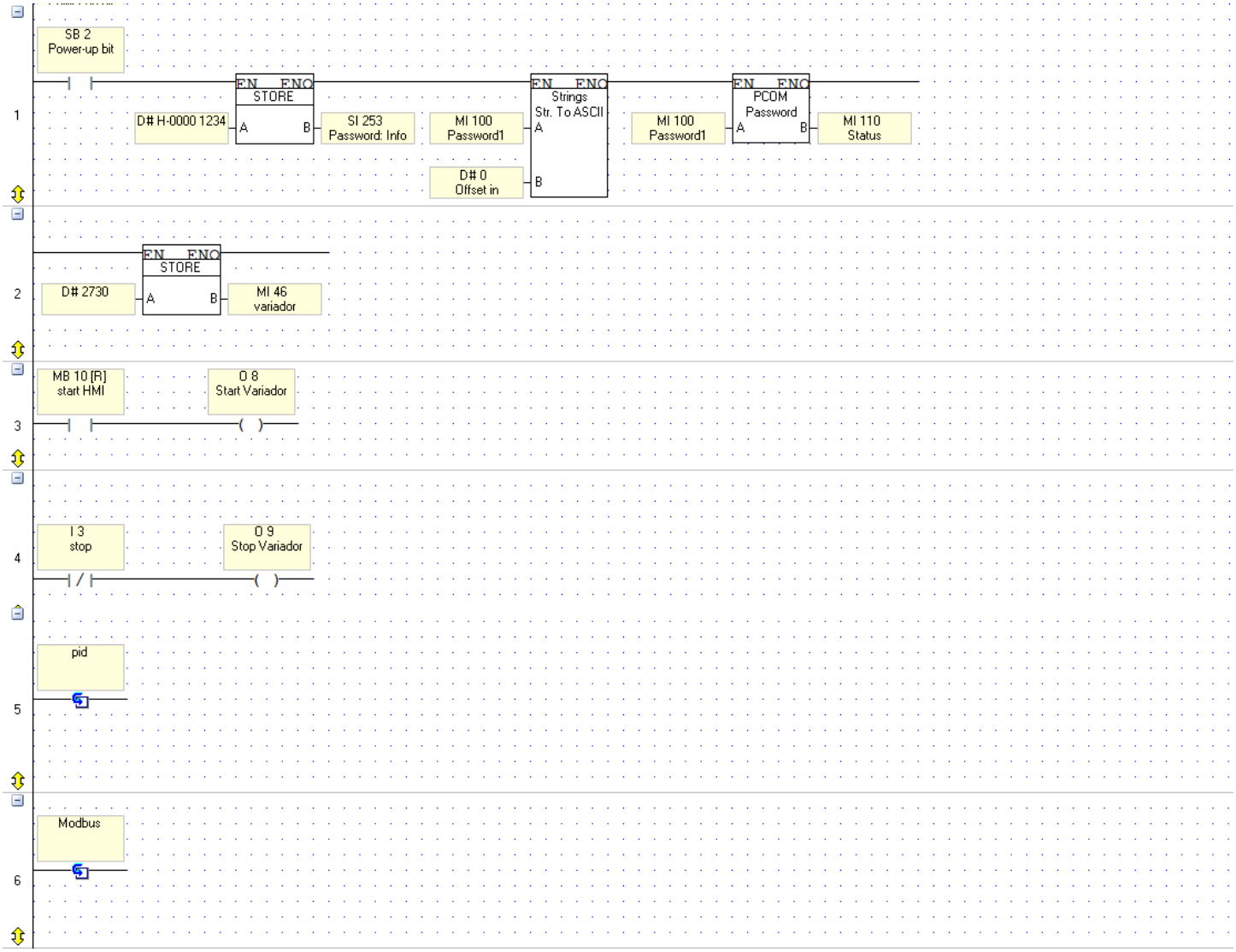
Modbus


SB 2
Power-up bit
EN ENO
PLC NAME
Slave
EN ENO
COM INIT 1
EN ENO
MODBUS
CONFIG
MODBUS_1
Port 1
MB 5
Function in
1
Network ID 2
D# 100
Time out (units of
D# 3
Retries
SB 2
Power-up bit
EN ENO
MODBUS
SCAN_EX
MODBUS_1
2

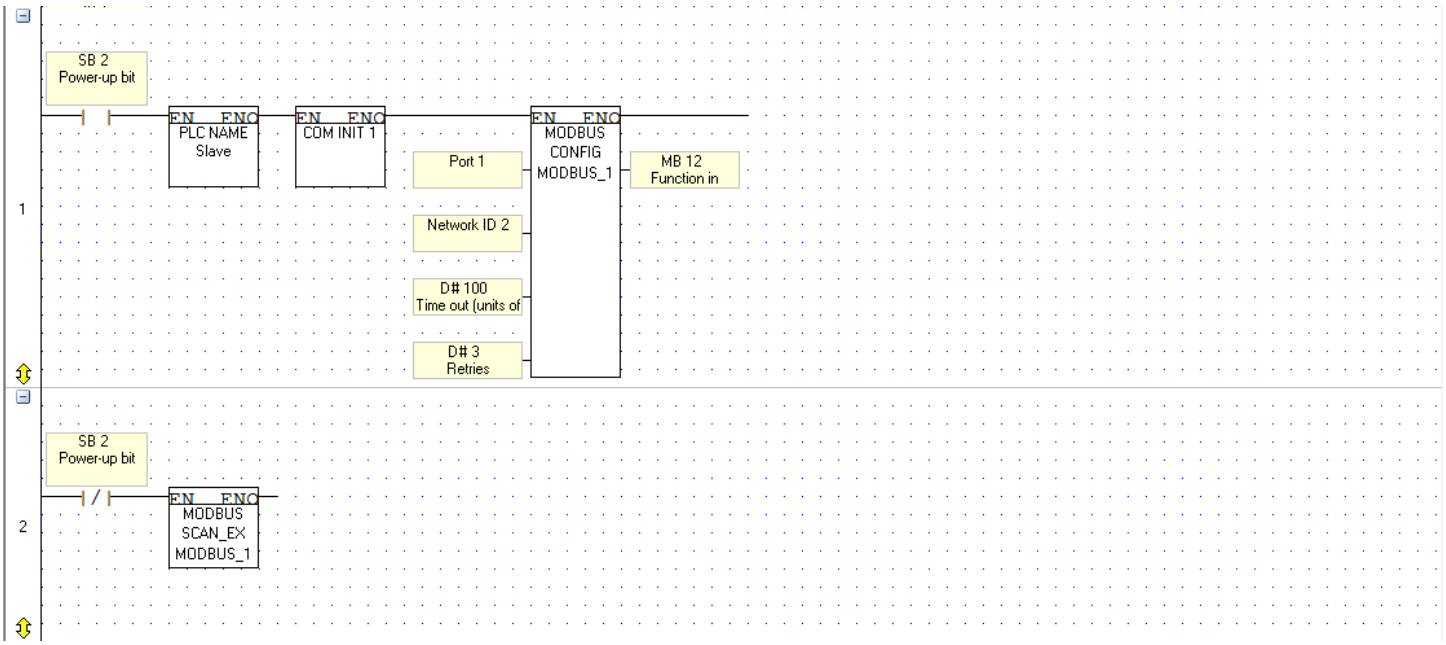
10.3.7. 7) Control PID Flujo



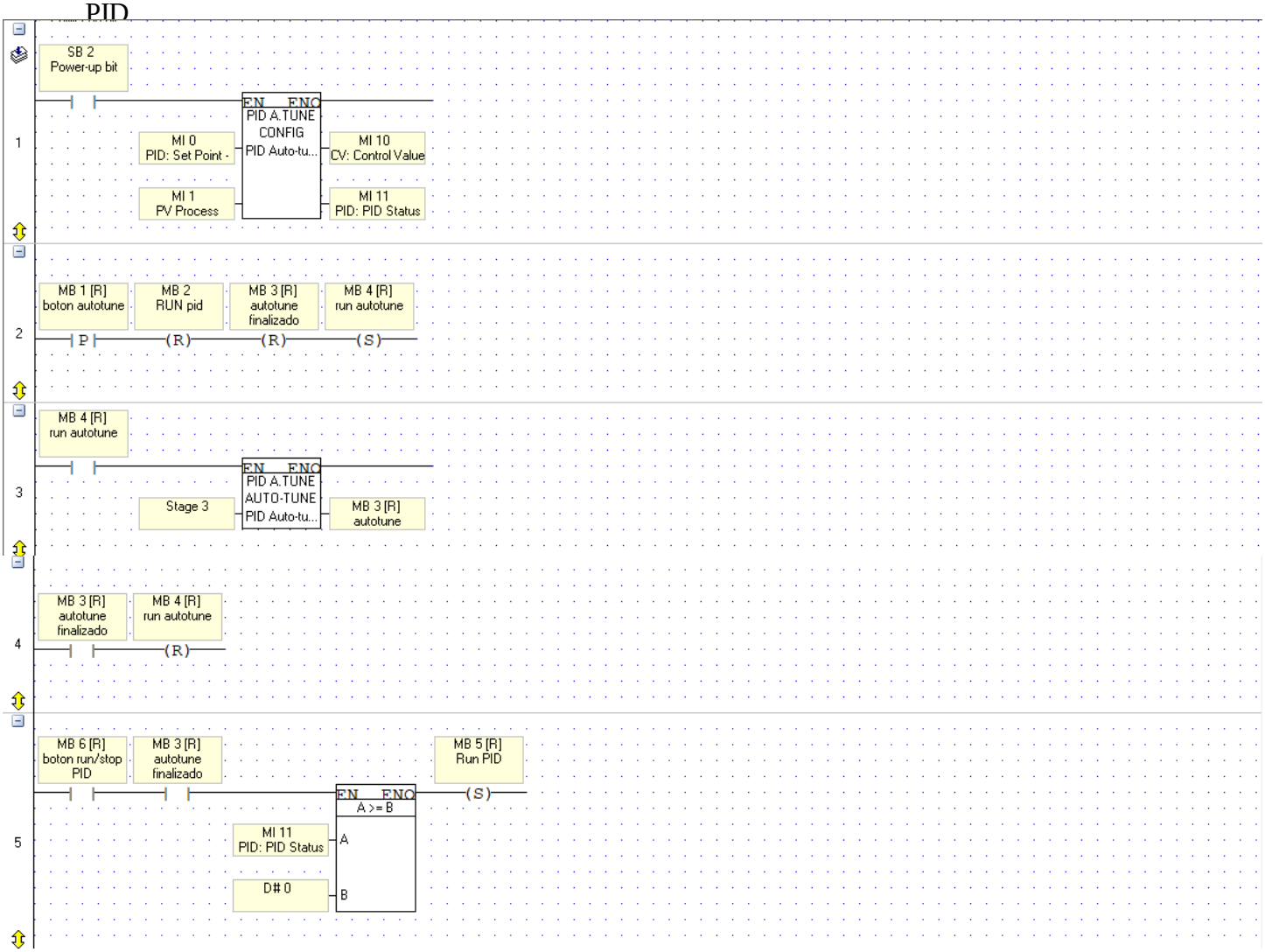
Main Routine

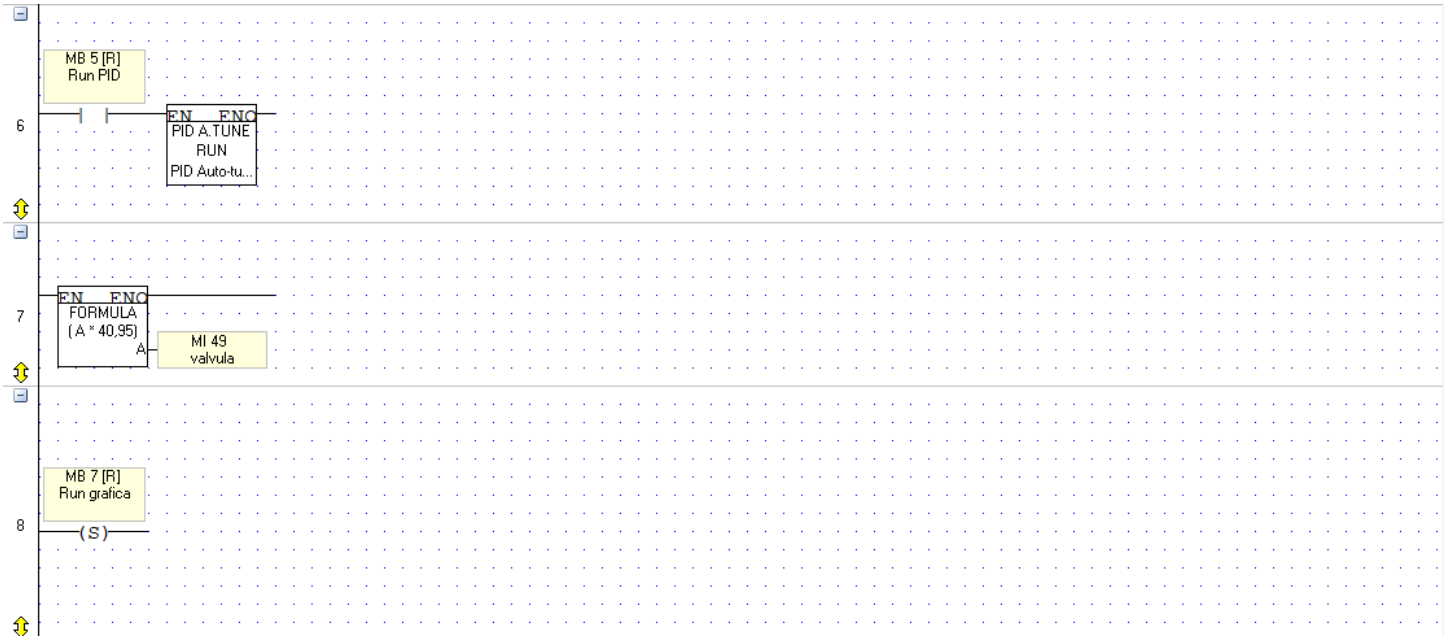


Modbus


SB 2
Power-up bit
EN ENO
PLC NAME
Slave
EN ENO
COM INIT 1
Port 1
Network ID 2
D# 100
Time out (units of
D# 3
Retries
EN ENO
MODBUS
CONFIG
MODBUS_1
MB 12
Function in
1
SB 2
Power-up bit
/
EN ENO
MODBUS
SCAN_EX
MODBUS_1
2

PID


SB 2
Power-up bit
EN ENO
PID A.TUNE
CONFIG
PID Auto-tu...
MI 0
PID: Set Point -
MI 1
PV Process
MI 10
CV: Control Value
MI 11
PID: PID Status
1
MB 1 [R]
boton autotune
MB 2
RUN pid
MB 3 [R]
autotune
finalizado
MB 4 [R]
run autotune
P
(R)
(R)
(S)
2
MB 4 [R]
run autotune
EN ENO
PID A.TUNE
AUTO-TUNE
PID Auto-tu...
Stage 3
MB 3 [R]
autotune
3
MB 3 [R]
autotune
finalizado
MB 4 [R]
run autotune
(R)
4
MB 6 [R]
boton run/stop
PID
MB 3 [R]
autotune
finalizado
MB 5 [R]
Run PID
(S)
EN ENO
A >= B
MI 11
PID: PID Status
A
D# 0
B
5


MB 5 [R]
Run PID
6
EN ENO
PID A.TUNE
RUN
PID Auto-tu...
7
EN ENO
FORMULA
(A * 40,95)
A
MI 49
valvula
MB 7 [R]
Run grafica
8
(S)