



Desarrollo de un sistema de supervisión y control avanzado de la estación de nivel ND-0704, basado en IIoT y PLC M-DUINO, en el Laboratorio de Redes Industriales y Control de Procesos.

Catota Camacho, Marco Vinicio y Jiménez León, Marlon Bolívar

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica e Instrumentación

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica e Instrumentación

Ing. Pruna Panchi, Edwin Patricio

23 de agosto del 2023

Latacunga

Reporte de verificación de contenido



Trabajo de titulación Jiménez Catota....

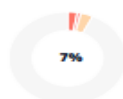
Scan details

Scan time:
August 14th, 2023 at 13:41 UTC

Total Pages:
61

Total Words:
15241

Plagiarism Detection



Types of plagiarism		Words
Identical	2.6%	400
Minor Changes	1.1%	164
Paraphrased	3.3%	509
Omitted Words	0%	0

AI Content Detection



Text coverage
 AI text
 Human text

Plagiarism Results: (51)

<p>4. CAPÍTULO 4</p> <p>https://biblus.us.es/bibing/proyectos/abreproy/12070/ficher...</p> <p>Rafael Pajarón Pérez</p> <p>Capítulo 4 Control clásico aplicado a la planta La primera estrategia que se prueba esta basada en un control clásico Proporcional, Inte...</p>	0.8%
<p>¿Qué es el IIoT? Definición y detalles</p> <p>https://www.paessler.com/es/it-explained/iiot</p> <p>EspañolEnglish Deutsch Français Italiano Português Русский Acceder ProductosIT & OT Monitoring Paessler PRTGMonit...</p>	0.7%
<p>Qué es el PLC Arduino y su uso en la industria</p> <p>https://www.cadenzaelectric.com/blog/que-es-el-plc-arduino...</p> <p>× Uso de cookies Utilizamos cookies propias y de terceros, para analizar la navegación de los usuarios. Si con...</p>	0.6%

Certified by

About this report
help.copyleaks.com

copyleaks.com

Ing. Pruna Panchi, Edwin Patricio

C. C.: 0502651003



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica e Instrumentación

Certificación

Certifico que el trabajo de titulación: **“Desarrollo de un sistema de supervisión y control avanzado de la estación de nivel ND-0704, basado en IloT y PLC M-DUINO, en el Laboratorio de Redes Industriales y Control de Procesos”** fue realizado por los señores **Catota Camacho, Marco Vinicio y Jiménez León, Marlon Bolívar**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 23 de agosto del 2023

Firma:

Ing. Pruna Panchi, Edwin Patricio

C. C.: 0502651003



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica e Instrumentación

Responsabilidad de autoría

Nosotros, **Catota Camacho, Marco Vinicio**, con cédula de ciudadanía N° 1726777384 y **Jiménez León, Marlon Bolívar** con cédula de ciudadanía N° 0503961310, declaramos que el contenido, ideas y criterios del trabajo de titulación: “**Desarrollo de un sistema de supervisión y control avanzado de la estación de nivel ND-0704, basado en IloT y PLC M-DUINO, en el Laboratorio de Redes Industriales y Control de Procesos**”, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 23 de agosto del 2023

Firmas:



Catota Camacho, Marco Vinicio
C.C.: 1726777384



Jiménez León, Marlon Bolívar
C.C.: 0503961310



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica e Instrumentación

Autorización de publicación

Nosotros, **Catota Camacho, Marco Vinicio**, con cédula de ciudadanía N° 1726777384 y **Jiménez León, Marlon Bolívar** con cédula de ciudadanía N° 0503961310, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “**Desarrollo de un sistema de supervisión y control avanzado de la estación de nivel ND-0704, basado en IIoT y PLC M-DUINO, en el Laboratorio de Redes Industriales y Control de Procesos**”, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 23 de agosto del 2023

Firmas:

Catota Camacho, Marco Vinicio

C.C.: 1726777384

Jiménez León, Marlon Bolívar

C.C.: 0503961310

Dedicatoria

Dedico el resultado de este trabajo a mis padres Segundo Catota y María Camacho en conjunto con mis hermanas María y Germania, gracias a su apoyo incondicional y motivación que me han brindado para seguir adelante y no rendirme jamás ante cualquier circunstancia que se presente.

Marco Vinicio Catota Camacho

Dedicatoria

Quiero dedicar este trabajo a mi madre Fabiola León, quien gracias a su trabajo y perseverancia de verme triunfar nunca desmayo de darme lo mejor para poder lograr mis objetivos.

A mi padre Bolívar Jiménez que, aunque físicamente no se encuentre aquí, desde el cielo me ha cuidado, guiado y bendecido durante todo este largo proceso de mi vida.

A mis hermanas Jenny Jiménez y Lorena Jiménez, quienes fueron un pilar fundamental y claro ejemplo de superación y motivación para cumplir mis sueños.

A mi esposa Karla Ramírez y mi hija Kaitlyn Jiménez, quienes con su apoyo incondicional se convirtieron en la principal inspiración para lograr esta meta trazada en vida.

Marlon Bolívar Jiménez León

Agradecimiento

Agradezco a Dios, por ser la luz que ilumina mi camino y ayudarme a tomar la decisión correcta en todo el transcurso de mi vida.

A mis padres y hermanas por apoyarme en los estudios e impulsarme para culminarlos y seguir adelante.

A mi tutor de tesis, por compartir sus conocimientos, amistad, durante el desarrollo del trabajo de investigación.

Al interprovincial Canario # 1, que todos los días me ayudaba trasladándome de Quito a Latacunga a las 5 am para llegar puntual a la Universidad.

Marco Vinicio Catota Camacho

Agradecimiento

Deseo expresar mi agradecimiento a Dios por brindarme aliento para seguir adelante cumpliendo objetivos en esta vida.

A mis padres por regalarme lo mejor de sus años y darme alas para volar, inculcándome siempre los mejores valores.

Así mismo agradezco a mi esposa, a mi hija, a mis hermanas y familia en general quienes con su apoyo incondicional me han dado fuerzas para seguir adelante a pesar de las dificultades.

Por su orientación y atención a nuestras consultas, mi agradecimiento infinito al Ing. Edwin Pruna nuestro tutor de tesis, quien nos ha brindado valiosos conocimientos para la realización del presente trabajo.

A mis compañeros de trabajo de Lactalis quienes me han ayudado con la facilidad horaria con la finalidad de permitirme cumplir este gran sueño.

Marlon Bolívar Jiménez León

ÍNDICE DE CONTENIDOS

Carátula.....	1
Reporte de verificación de contenido	2
Certificación	3
Responsabilidad de autoría.....	4
Autorización de publicación.....	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento.....	8
Agradecimiento.....	9
Índice de contenidos	10
Índice de tablas	17
Índice de figuras	18
Índice de ecuaciones.....	23
Resumen	24
Abstract.....	25
Capítulo I: Introducción	26
Antecedentes	26
Planteamiento del problema	27
Justificación e importancia	27
Objetivos de la investigación.....	28

<i>Objetivo general</i>	28
<i>Objetivos específicos</i>	28
Variables de la investigación	29
<i>Variable independiente</i>	29
<i>Variable dependiente</i>	29
Hipótesis	29
Capítulo II: Marco teórico	30
Algoritmos de control	30
<i>Controladores clásicos PID</i>	30
Métodos de sintonía para controladores PID	31
<i>Método de sintonía Ziegler-Nichols</i>	31
<i>Método de sintonía Cohen Coon</i>	31
<i>Método de sintonía Lambda</i>	32
<i>Método de sintonía Haalman</i>	33
Control en cascada	34
Modelamiento matemático	35
Estaciones de nivel	35
Transmisor de nivel Rosemount 5400 Series	36
Caudalímetro George Fischer 8150 series	37
Pantalla thinker touch (Industrial Shield)	38
Obtención del modelo matemático	39
Controladores avanzados	40
<i>Control predictivo (MPC)</i>	40
<i>Control adaptativo</i>	41
<i>Control difuso (Fuzzy)</i>	41

<i>Control por redes neuronales</i>	41
PLC M-Duino	41
Tecnología IloT.....	43
Protocolos de comunicación IloT	44
<i>HTTP (REST/JSON)</i>	44
<i>MQTT</i>	44
<i>CoAP</i>	44
<i>DDS</i>	45
<i>AMQP</i>	45
<i>APK-Tablet</i>	45
Linux.....	46
Router TP-Link.....	47
Modbus serial.....	48
Modbus TCP-IP	49
Node-RED	49
Convertor de 4-20mA a 2-10 V.....	50
<i>Resistencia de carga:</i>	50
Cable Ethernet Directo y Cruzado estándar TIPO A y TIPO B	51
<i>Cable directo ethernet:</i>	52
<i>Cable cruzado ethernet:</i>	52
Proceso autoregulado e integrante.....	52
<i>Proceso autoregulado:</i>	53
<i>Proceso integrante:</i>	53
Software Matlab	53
Software LabWiew	54

Capítulo III: Diseño e implementación	56
Identificación de los puertos de comunicación del PLC MDUINO 38AR+.....	57
Conexión entre los dispositivos y red de alimentación principal.	58
Instalación del Software e instalación de tarjetas Industrial Shields	59
<i>Pasos a seguir para instalar tarjetas Industrial Shields:.....</i>	<i>59</i>
Pasos para observar las placas Industrial Shields:	62
Instalación del Software Node Red en la thinker touch.....	63
Instalación de librerías Dashboard en Node Red	65
Pruebas de comunicación entre dispositivos mediante Modbus TCP/IP y MQTT	68
<i>Pruebas de comunicación al puerto.</i>	<i>68</i>
<i>Pruebas de comunicación Modbus TCP/IP.....</i>	<i>70</i>
Identificación de la estación de nivel ND-0704	72
Diseño del diagrama P&ID.....	74
Dinámica del proceso y modelo matemático de la estación de Nivel ND-0704.....	75
Configuración de los software previos a la obtención del modelo matemático.	76
<i>Configuración del LabVIEW</i>	<i>76</i>
<i>Configuración del Kepserver</i>	<i>76</i>
<i>Procesamiento de datos mediante el uso de Matlab y Excel.....</i>	<i>78</i>
Modelo matemático de la variable Caudal.....	81
Modelo matemático de la variable nivel.....	82
Modelo matemático de la variable nivel en cascada.....	82
Diseño de los controladores clásicos basado en métodos de sintonía.....	82
<i>Método de sintonía Cohen-Coon.....</i>	<i>83</i>
Caudal.....	83
<i>Controlador PI</i>	<i>84</i>

<i>Controlador PID</i>	84
Nivel	86
<i>Controlador PI</i>	86
<i>Controlador PID</i>	87
Método de sintonía de Haalman	88
Caudal	88
<i>Controlador PI</i>	89
Nivel	89
<i>Controlador PI</i>	89
Método de sintonía Lambda Agresivo	90
Caudal	90
<i>Controlador PI</i>	90
<i>Controlador PID</i>	91
Nivel	92
<i>Controlador PI</i>	93
<i>Controlador PID</i>	93
Método de sintonía Lambda Robusto	94
Caudal	94
<i>Controlador PI</i>	95
<i>Controlador PID</i>	95
Nivel	96
<i>Controlador PI</i>	97

Controlador PID.....	97
Diseño del controlador avanzado Fuzzy.....	98
Implementación de los controladores en el PLC M-duino	100
<i>Controlador simple de la variable caudal.....</i>	<i>100</i>
<i>Controlador simple de la variable nivel.....</i>	<i>101</i>
<i>Controlador de la técnica de control avanzado (Cascada).....</i>	<i>102</i>
<i>Controlador avanzado Fuzzy</i>	<i>102</i>
<i>Controlador técnica de control avanzado cascada (Fuzzy Caudal-PID Nivel)..</i>	<i>103</i>
Creación del HMI del Sistema	104
<i>Dashboard utilizados para la creación del HMI</i>	<i>105</i>
<i>Diagrama 3D.....</i>	<i>105</i>
<i>Diagrama 2D.....</i>	<i>106</i>
<i>Tendencias</i>	<i>106</i>
Vinculación de las variables entre el PLC y la ThinkerTouch.....	107
<i>Código de programación arduino</i>	<i>107</i>
<i>Código de programación Node-RED.....</i>	<i>108</i>
Creación de cuenta en flowforge para acceso global.....	112
Capítulo IV: Pruebas y análisis de resultados	113
Pruebas experimentales	113
<i>Funcionamiento del sistema de Caudal</i>	<i>113</i>
Desempeño del controlador PID en el sistema de Caudal.....	113
Desempeño del controlador Fuzzy en el sistema de Caudal.....	121
Desempeño del mejor controlador PID vs Fuzzy (sistema de Caudal)	123
<i>Funcionamiento del sistema de Nivel.....</i>	<i>124</i>

Desempeño del controlador PID en el sistema de nivel.....	124
<i>Funcionamiento del sistema en cascada (Caudal PID-Nivel PID)</i>	<i>132</i>
Desempeño del controlador (Caudal PID-Nivel PID)	132
<i>Funcionamiento del sistema en cascada (Caudal fuzzy-Nivel PID)</i>	<i>134</i>
Desempeño del controlador (Caudal fuzzy-Nivel PID)	134
<i>Funcionamiento del sistema de supervisión y control IloT</i>	<i>135</i>
Costos Automatas, pantallas y módulos IloT de Industrial Shields Vs Siemens....	138
Capítulo V: Conclusiones y recomendaciones.....	139
Conclusiones	139
Recomendaciones	141
Bibliografía	142
Anexos	145

ÍNDICE DE TABLAS

Tabla 1 <i>Ecuaciones para calcular los parámetros de sintonía de Ziegler-Nichols</i>	31
Tabla 2 <i>Ecuaciones para calcular los parámetros de sintonía de Cohen Coon</i>	32
Tabla 3 <i>Ecuaciones para calcular los parámetros de sintonía Lambda</i>	33
Tabla 4 <i>Ecuaciones para calcular los parámetros de sintonía Haalman</i>	34
Tabla 5 <i>Tabla de identificación de puertos del PLC</i>	57
Tabla 6 <i>Tabla FAM para definición de reglas de inferencia</i>	99
Tabla 7 <i>Parámetros controlador PID en el sistema de Caudal</i>	120
Tabla 8 <i>Parámetros del controlador Fuzzy en el sistema de Caudal</i>	122
Tabla 9 <i>Parámetros del controlador PID y Fuzzy en el sistema de Caudal</i>	124
Tabla 10 <i>Parámetros de los controladores PID en el sistema de nivel</i>	131
Tabla 11 <i>Costos de dispositivos</i>	138

ÍNDICE DE FIGURAS

Figura 1 <i>Diagrama de bloques de un sistema de control PID.</i>	30
Figura 2 <i>Diagrama de la técnica de control en cascada.</i>	35
Figura 3 <i>Estación de nivel ND-0704.</i>	36
Figura 4 <i>Transmisor de Nivel Rosemount 5400 series</i>	37
Figura 5 <i>Caudalímetro George Fischer 8150 series</i>	38
Figura 6 <i>Pantalla Thinker Touch (Industrial Shields)</i>	39
Figura 7 <i>Variables de un proceso.</i>	40
Figura 8 <i>PLC-M-DUINO</i>	43
Figura 9 <i>Aplicación APK.</i>	46
Figura 10 <i>Sistema operativo debian.</i>	47
Figura 11 <i>Router TP-Link</i>	48
Figura 12 <i>Software Node-RED.</i>	50
Figura 13 <i>Estándar Tipo A y Tipo B.</i>	51
Figura 14 <i>Estación de nivel (LabVIEW)</i>	54
Figura 15 <i>Sistema de supervisión y control IIoT de la estación de nivel ND-0704.</i>	56
Figura 16 <i>Diagrama de puertos utilizados del PLC M-DUINO</i>	58
Figura 17 <i>Conexión entre dispositivos y red de alimentación principal.</i>	59
Figura 18 <i>Arduino IDE "Archivo -> Preferencias".</i>	60
Figura 19 <i>URL y Aceptar.</i>	60
Figura 20 <i>Herramientas -> Tarjetas -> Administrador de tableros.</i>	61

Figura 21 <i>“industrialshields” en el navegador.</i>	61
Figura 22 <i>Selección de la familia Ardbox o la familia M-Duino.</i>	62
Figura 23 <i>Selección del modelo específico M-Duino 38AR+.</i>	62
Figura 24 <i>Terminal LX</i>	63
Figura 25 <i>Actualización de paquetes</i>	63
Figura 26 <i>Instalación del Node.js en Debian.</i>	64
Figura 27 <i>Instalación del Administrador de Paquetes.</i>	64
Figura 28 <i>Desactivar certificados SSL.</i>	64
Figura 29 <i>Instalación de Node-Red.</i>	65
Figura 30 <i>Comprobación de la instalación de Node-Red</i>	65
Figura 31 <i>Proceso de instalación de librerías.</i>	66
Figura 32 <i>Realizar búsqueda de librerías.</i>	66
Figura 33 <i>Instalación de librería ModBus.</i>	67
Figura 34 <i>Instalación de librería dashboard.</i>	67
Figura 35 <i>Instalación de librería mqtt.</i>	68
Figura 36 <i>Comunicación Tinker Touch.</i>	69
Figura 37 <i>Comunicación PLC-MDuino.</i>	69
Figura 38 <i>Comunicación Laptop.</i>	69
Figura 39 <i>Bloques de programación Modbus write y Modbus Read (Node-Red).</i>	70
Figura 40 <i>Configuración de programación Modbus Write y Modbus Read (Node-Red).</i>	71
Figura 41 <i>Configuración del bloque Function Node.</i>	72

Figura 42 Estación de Nivel ND-0704.	73
Figura 43 Módulo didáctico IloT.	74
Figura 44 Diagrama P&ID estación de Nivel ND-0704.	75
Figura 45 Programación del VI para la toma de datos de la estación.	76
Figura 46 Configuración del Kep Server.	77
Figura 47 Configuración del OPC Quick Client.	77
Figura 48 Importación de vectores desde Matlab hacia Excel.	78
Figura 49 Ejecución del comando “ident” en Matlab.	78
Figura 50 Estimación del modelo del proceso.	79
Figura 51 Click en “Estimate” para obtener el modelo matemático.	79
Figura 52 Seleccionar “Model output” para obtener el porcentaje de estimación.	80
Figura 53 Porcentaje de aproximación del proceso de nivel.	80
Figura 54 Porcentaje de aproximación del proceso de caudal (Lazo secundario).	81
Figura 55 Declaración de los rangos de entrada y salida.	99
Figura 56 Triangulación de las entradas y las salidas.	100
Figura 57 Definición de las 49 reglas Fuzzy.	100
Figura 58 Implementación control PID variable Caudal en el software Arduino.	101
Figura 59 Implementación control PID variable Nivel en el software Arduino.	101
Figura 60 Implementación de la técnica de control avanzado Cascada.	102
Figura 61 Implementación control Fuzzy variable Caudal en el software Arduino.	103
Figura 62 Control Fuzzy variable Caudal y PID variable Nivel en el software Arduino.	104

Figura 63 <i>Tag principal SVG graphics node.</i>	105
Figura 64 <i>Diagrama 3D.</i>	105
Figura 65 <i>Diagrama P&ID.</i>	106
Figura 66 <i>Tendencias Nivel – Caudal.</i>	106
Figura 67 <i>Registros y librerías Modbus.</i>	107
Figura 68 <i>Funciones para definir alarmas.</i>	108
Figura 69 <i>Funciones para la variable Nivel-Caudal utilizadas en el diagrama 3D.</i>	108
Figura 70 <i>Funciones para la variable Nivel-Caudal utilizadas en el diagrama 2D.</i>	109
Figura 71 <i>Setpoint Nivel, iiot/spniveInube, Modbus Write, función dato-tendencia nivel.</i>	109
Figura 72 <i>liot/spcaudal, Setpoint Caudal, función dato-tendencia caudal, iiot/caudal.</i>	110
Figura 73 <i>Modbus Read, function 2.</i>	110
Figura 74 <i>Tendencias CV1 y CV2.</i>	111
Figura 75 <i>Actualizar constantes kp, ki y kd.</i>	111
Figura 76 <i>Creación de cuenta en flowforge.</i>	112
Figura 77 <i>Creación de cuenta en flowforge (ingreso de datos solicitados).</i>	112
Figura 78 <i>Respuesta del controlador PI, método de sintonía Cohen Coon (Caudal).</i>	114
Figura 79 <i>Respuesta del controlador PID, método de sintonía Cohen Coon (Caudal).</i>	115
Figura 80 <i>Respuesta del controlador PI, método de sintonía Haalman (Caudal).</i>	116
Figura 81 <i>Respuesta del controlador PI, método de sintonía Lambda Agresivo (Caudal).</i>	117
Figura 82 <i>Respuesta del controlador PID, método de sintonía Lambda Agresivo (Caudal).</i>	118
Figura 83 <i>Respuesta del controlador PI, método de sintonía Lambda Robusto (Caudal).</i>	119

Figura 84 <i>Comparación de los métodos de sintonía PID en el sistema de Caudal.</i>	120
Figura 85 <i>Respuesta del controlador Fuzzy del sistema de Caudal.</i>	122
Figura 86 <i>Respuesta del controlador PID vs Fuzzy del sistema de Caudal.</i>	123
Figura 87 <i>Respuesta del controlador PI, método de sintonía Cohen Coon (Nivel).</i>	125
Figura 88 <i>Respuesta del controlador PID, método de sintonía Cohen Coon (Nivel).</i>	126
Figura 89 <i>Respuesta del controlador PI, método de sintonía Haalman (Nivel).</i>	127
Figura 90 <i>Respuesta del controlador PI, método de sintonía Lambda Agresivo (Nivel).</i>	128
Figura 91 <i>Respuesta del controlador PID, método de sintonía Lambda Agresivo (Nivel).</i>	129
Figura 92 <i>Respuesta del controlador PI, método de sintonía Lambda Robusto (Nivel).</i>	130
Figura 93 <i>Comparación de los métodos de sintonía PID en el sistema de nivel.</i>	131
Figura 94 <i>Respuesta del controlador PID, método de control avanzado en cascada.</i>	133
Figura 95 <i>Respuesta del controlador (Caudal fuzzy-Nivel PID) en cascada.</i>	134
Figura 96 <i>Sistema de supervisión y control IIoT</i>	135
Figura 97 <i>Supervisión y control en la Tinker Touch a un punto de consigna de 30cm.</i>	135
Figura 98 <i>Supervisión y control en la Tinker Touch a un punto de consigna de 30cm.</i>	136
Figura 99 <i>Diagrama P&ID visualizado en la Tinker Touch</i>	136
Figura 100 <i>Supervisión y control desde una laptop a un punto de consigna de 45cm.</i>	137
Figura 101 <i>Supervisión y control desde la APK a un punto de consigna de 30 LPMS</i>	137

ÍNDICE DE ECUACIONES

Ecuación 1 <i>Modelo matemático de la variable caudal.</i>	81
Ecuación 2 <i>Modelo matemático de la variable nivel.</i>	82
Ecuación 3 <i>Modelo matemático de la variable nivel en método cascada</i>	82
Ecuación 4 <i>Constantes para cálculos del método Cohen-Coon (Caudal)</i>	83
Ecuación 5 <i>Fórmulas del controlador Cohen-Coon (PI)-Caudal</i>	84
Ecuación 6 <i>Fórmulas del controlador Cohen-Coon (PID)-Caudal</i>	85
Ecuación 7 <i>Fórmulas de las constantes del controlador Cohen-Coon (Nivel)</i>	86
Ecuación 8 <i>Fórmulas del controlador Cohen-Coon (PI)-Nivel</i>	87
Ecuación 9 <i>Fórmulas del controlador Cohen-Coon (PID)-Nivel</i>	88
Ecuación 10 <i>Fórmulas del controlador Haalman (PI)-Caudal</i>	89
Ecuación 11 <i>Fórmulas del controlador Haalman (PI)-Nivel</i>	90
Ecuación 12 <i>Fórmulas del controlador Lambda Agresivo (PI)-Caudal</i>	91
Ecuación 13 <i>Fórmulas del controlador Lambda Agresivo (PID)-Caudal</i>	92
Ecuación 14 <i>Fórmulas del controlador Lambda Agresivo (PI)-Nivel</i>	93
Ecuación 15 <i>Fórmulas del controlador Lambda Agresivo (PID)-Nivel</i>	94
Ecuación 16 <i>Fórmulas del controlador Lambda Robusto (PI)-Caudal</i>	95
Ecuación 17 <i>Fórmulas del controlador Lambda Robusto (PID)-Caudal</i>	96
Ecuación 18 <i>Fórmulas del controlador Lambda Robusto (PI)-Nivel</i>	97
Ecuación 19 <i>Fórmulas del controlador Lambda Robusto (PID)-Nivel</i>	98
Ecuación 20 <i>Fórmula cálculo del sobre impulso.</i>	113

Resumen

En el presente trabajo de titulación se desarrolló un sistema de supervisión y control avanzado para la variable nivel, en la estación ND-0704 del Laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas ESPE- sede Latacunga, mismo que tiene un enfoque a la industria 4.0 utilizando nuevas tecnologías basadas en IIoT, se utiliza una Tinker Touch, un controlador (M-Duino), donde se desarrollan y sintonizan los algoritmos de control, tales como PID y Fuzzy, utilizando la estrategia de control avanzado cascada, además mediante la utilización de Node Red, aplicación (APK para Android) y software Arduino, se realizó el enlace con la nube (Tinker Touch-Nube) mediante MQTT y (Tinker Touch-Autómata), mediante Modbus TCP/IP utilizando librerías proporcionadas por desarrolladores de software. Como resultado se presenta el monitoreo y supervisión de las variables del proceso (Caudal y Nivel), de manera local y remota. Mediante el análisis se determinó el controlador más eficiente en lazo simple para el sistema de caudal es el controlador PI método de sintonía Lambda Robusto y para el sistema de nivel es el controlador PI con el método de sintonía Lambda Agresivo. Sin embargo, en cascada el mejor controlador para el sistema de caudal es el fuzzy y para el sistema de nivel es el PI Lambda Agresivo.

Palabras clave: Control Avanzado, IIoT, M-Duino, Tinker Touch, APK, MQTT

Abstract

In this degree work, an advanced monitoring and control system was developed for the level variable, in the ND-0704 station of the Industrial Networks and Process Control Laboratory of the University of the Armed Forces ESPE- Latacunga, which has a focus on industry 4.0 using new technologies based on IIoT, a Tinker Touch is used, a controller (M-Duino), where the control algorithms are developed and tuned, such as PID and Fuzzy, using the advanced cascade control strategy, also by using Node Red, application (APK for Android) and Arduino software, the link was made with the cloud (Tinker Touch-Cloud) through MQTT and (Tinker Touch-Automaton), through Modbus TCP/IP using libraries provided by software developers. As a result, the monitoring and supervision of the process variables (Flow and Level) is presented, locally and remotely. Through the analysis it was determined that the most efficient controller in simple loop for the flow system is the PI controller with the Robust Lambda tuning method and for the level system is the PI controller with the Aggressive Lambda tuning method. However, in cascade the best controller for the flow system is the fuzzy controller and for the level system is the PI Lambda Aggressive controller.

Keywords: Advanced Control, IIoT, M-Duino, Tinker Touch, APK, MQTT.

Capítulo I

Introducción

Antecedentes

En la actualidad los sistemas de control automático son de vital importancia en la industria y en la formación técnico profesional, ya que principalmente se busca aumentar la eficiencia de los procesos, implementando nuevas tecnologías que nos permitan monitorear el comportamiento de las variables de interés de forma remota. El control automático de nivel, utilizando los distintos tipos de algoritmos de control nace por la necesidad de automatizar el proceso, es así que dada la necesidad de monitorear, controlar y adquirir los diferentes datos del proceso en la actualidad la tendencia es a la implementación de sistemas de monitoreo y supervisión remotos.

Los sistemas de control de nivel son muy utilizados en la industria para aplicaciones como: llenado de tanques, tratamiento y recirculación de agua, medición y control de flujo, su funcionamiento se basa principalmente en la manipulación de las variables flujo y nivel para así garantizar el correcto desarrollo del proceso, evitando así el desperdicio del agua. Hay que tener en cuenta que a medida que aumenta el flujo, el nivel del tanque se alcanza con mayor rapidez.

Con el avance de las nuevas tecnologías, en la actualidad existen formas de implementar diversos sistemas de monitoreo, control y supervisión, con la aparición de la industria 4.0 el IIoT (Internet Industrial de las cosas) permite que la conectividad de Internet se extienda a dispositivos industriales facilitando el intercambio de información mediante la nube con otros dispositivos utilizando sensores, software y diversos sistemas inteligentes que nos permitirán el acceso en cualquier instante de tiempo (Toro, 2017). El Internet industrial de las cosas (IIoT) tiene el potencial para ser la novedad más influyente y disruptiva en automatización, esto ha provocado la reconciliación e integración con los sistemas de

automatización existentes, sin embargo, el IIoT busca mejorar la operación y gestión de los procesos de producción industriales. (Editores, 2022)

Planteamiento del problema

En el laboratorio de Redes Industriales y Control de Procesos las aplicaciones IIoT implementadas son pocas, lo que dificulta la planificación de actividades prácticas para el desarrollo de sistemas de Supervisión y Control avanzado basado en IIoT. Por lo que es de gran importancia la implementación de nuevas soluciones industriales de vanguardia, para el conocimiento de esta nueva tecnología por parte de los estudiantes. Así también no se ha implementado en la actualidad una solución de automatización industrial de bajo costo como es el PLC M-DUINO, para verificar su funcionalidad y respuesta en ambientes industriales, y tener una propuesta de menor costo que los equipos comerciales más comunes en nuestro mercado (Siemens y Allen Bradley).

Justificación e importancia

Este proyecto es una alternativa de bajo costo, para la supervisión y control avanzado de la variable nivel, basado en un PLC Arduino, pero con características industriales. Así también en la actualidad los profesionales en el área de automatización y control deben tener conocimiento sobre comunicaciones IIOT, este proyecto permitirá conocer las nuevas soluciones industriales que existen, así como sus beneficios.

Al desarrollar el sistema de supervisión y control avanzado de la estación de nivel ND-0704, basado en IIoT y PLC M-DUINO, los autores adquieren conocimientos prácticos necesarios para su desenvolvimiento en la vida profesional, ya que aplicarán conocimientos desde conexiones de sensores y actuadores, diseño de controladores, diseño de un HMI hasta una configuración para enviar las variables de interés a la nube.

Objetivos de la investigación

Objetivo general

- Desarrollar el sistema de supervisión y control avanzado de la estación de nivel ND-0704, basado en IIOT y PLC M-DUINO, en el Laboratorio de Redes Industriales y Control de Procesos.

Objetivos específicos

- Investigar sobre el funcionamiento de la estación de nivel ND-0704, para obtener los modelos matemáticos correspondientes.
- Diseñar e implementar el sistema de control en cascada mediante controladores clásicos PI-PID.
- Diseñar e implementar el sistema de control en cascada mediante controladores difusos.
- Desarrollar la interfaz humano máquina en un touch panel que incluyan todas las variables de interés, para el monitoreo local del sistema de control en cascada.
- Desarrollar el algoritmo de comunicación en el PLC, para la comunicación IIoT.
- Investigar y programar la aplicación (APK) con la cual se realizará la supervisión.
- Realizar pruebas del funcionamiento del sistema de control en cascada, para verificar el funcionamiento de los equipos e instrumentos industriales, así como para determinar el controlador más eficiente.
- Realizar pruebas del funcionamiento del sistema de supervisión, para verificar su funcionamiento y comunicación.

Variables de la investigación

Variable independiente

Sistema IIOT para la supervisión y el control avanzado de la estación de nivel ND-0704.

Variable dependiente

Control automático de la estación de nivel ND-0704.

Monitoreo local y remoto de la estación de nivel ND-0704.

Hipótesis

En función de los antecedentes técnicos, planteamiento del problema, objetivos generales y específicos se plantea la siguiente hipótesis: “El sistema IIOT aplicado a una estación de nivel ND-0704 permitirá la supervisión del proceso de manera remota, así como también el control automático del proceso en tiempo real mediante dispositivos que se encuentren conectados a la red”

Capítulo II

Marco teórico

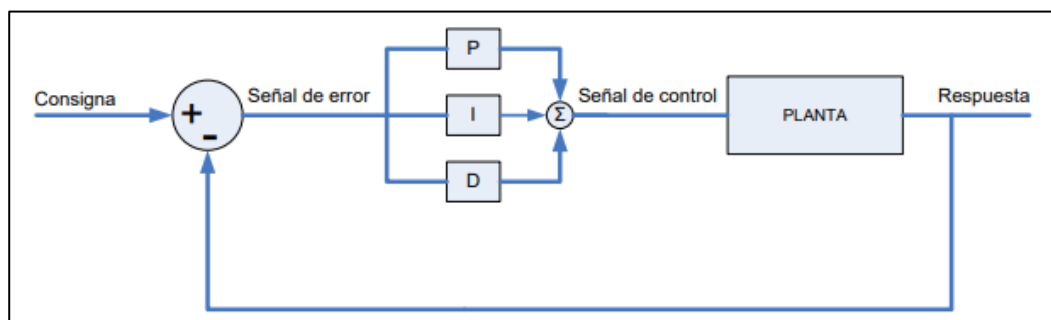
Algoritmos de control

Controladores clásicos PID

Más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID (Proportional-Integral-Derivative) o PID modificado. Debido a esto se ha pensado que un buen punto de partida para empezar a aplicar estrategias de control es empezar por los controladores más utilizados en la industria. La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. El controlador proporcional-integral-derivativo (PID) ha sido ampliamente usado en los procesos industriales por ser simple, robusto y confiable (Rafael, 2014). Como se aprecia en la Figura 1.

Figura 1

Diagrama de bloques de un sistema de control PID.



Nota. En la figura se aprecia el diagrama de bloques que tiene un controlador PID en lazo cerrado donde se muestra la acción proporcional, integral y derivativa. Tomado de Diseño de controladores PID por (Zayas Gato & Quintian Pardo, 2020, pág. 9).

Métodos de sintonía para controladores PID.

Método de sintonía Ziegler-Nichols.

El método inicial propuesto por Ziegler-Nichols se basa en obtener experimentalmente la respuesta del proceso ante una entrada de escalón unitario. Si dicha respuesta muestra una curva característica en forma de S, se considera fiable aplicar este método. Esta curva posee dos parámetros esenciales: el tiempo de retardo L y la constante de tiempo T (Ogata, 2010).

Tabla 1

Ecuaciones para calcular los parámetros de sintonía de Ziegler-Nichols

Tipo de controlador	Kp	Ti	Td
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

Nota. La tabla indica la regla del primer método de sintonía Ziegler-Nichols. Tomado de *Ingeniería de control moderna*, por K. Ogata, 2010, Pearson Educación.

Método de sintonía Cohen Coon.

Este tipo de controlador se destaca por su capacidad para contrarrestar las perturbaciones generadas por la carga en el sistema (Astrom & Hagglund, 2009). A continuación, se enumeran los parámetros de ajuste necesarios:

Tabla 2

Ecuaciones para calcular los parámetros de sintonía de Cohen Coon

Controlador	aK	T_i/L	T_d/L
PI	$0.9 \left(1 + \frac{0.092\tau}{1-\tau} \right)$	$\frac{3.3 - 3.0\tau}{1 + 1.2\tau}$	
PD	$1.24 \left(1 + \frac{0.13\tau}{1-\tau} \right)$		$\frac{0.27 - 0.36\tau}{1 - 0.87\tau}$
PID	$1.35 \left(1 + \frac{0.18\tau}{1-\tau} \right)$	$\frac{2.5 - 2.0\tau}{1 - 0.39\tau}$	$\frac{0.37 - 0.37\tau}{1 - 0.81\tau}$

Nota. Esta tabla indica las ecuaciones del método de sintonía Cohen Coon. Tomado de *Control PID avanzado*, por K. Astrom, T. Hagglund, 2009, Pearson Educación.

Donde los parámetros:

$$a = KL/T \text{ y } \tau = L/(L + T).$$

Método de sintonía Lambda.

Según (Astrom & Hagglund, 2009), se menciona que la sintonización Lambda es una técnica específica de asignación de polos que se utiliza ampliamente en la industria de procesos. Para determinar las constantes de sintonía K_p , T_i y T_d del controlador PID, el método de sintonización Lambda emplea las ecuaciones siguientes, que se detallan en la Tabla 3.

Tabla 3

Ecuaciones para calcular los parámetros de sintonía Lambda

Controlador	K_p	T_i	T_d
PI	$\frac{1}{K} \frac{T}{L + T_{cl}}$	T	
PID	$\frac{1}{K} \frac{L/2 + T}{L/2 + T_{cl}}$	$T + L/2$	$\frac{TL}{L + 2T}$

Nota. La tabla indica las ecuaciones del método de sintonía Lambda. Tomado de *Control PID avanzado*, por K. Astrom, T. Hagglund, 2009, Pearson Educación.

Donde K es la ganancia del proceso, T es la constante de tiempo, L es tiempo de retardo y T_{cl} es el tiempo de respuesta en lazo cerrado. Para obtener un controlador robusto $T_{cl} = 3T$ y para una sintonía agresiva $T_{cl} = T$ (Astrom & Hagglund, 2009).

Método de sintonía Haalman.

Este enfoque de ajuste se emplea en sistemas con un tiempo de retardo L considerable (Astrom & Hagglund, 2009). Para determinar los parámetros de sintonía de un controlador PI, se aplican las siguientes fórmulas matemáticas que se aprecian en la Tabla 4.

Tabla 4

Ecuaciones para calcular los parámetros de sintonía Haalman

Controlador	K_p	T_i	T_d
PI	$\frac{2T}{3KL}$	T	0

Nota. Esta tabla indica las ecuaciones del método de sintonía Haalman. Tomado de *Control PID avanzado*, por K. Astrom, T. Haggglund, 2009, Pearson Educación.

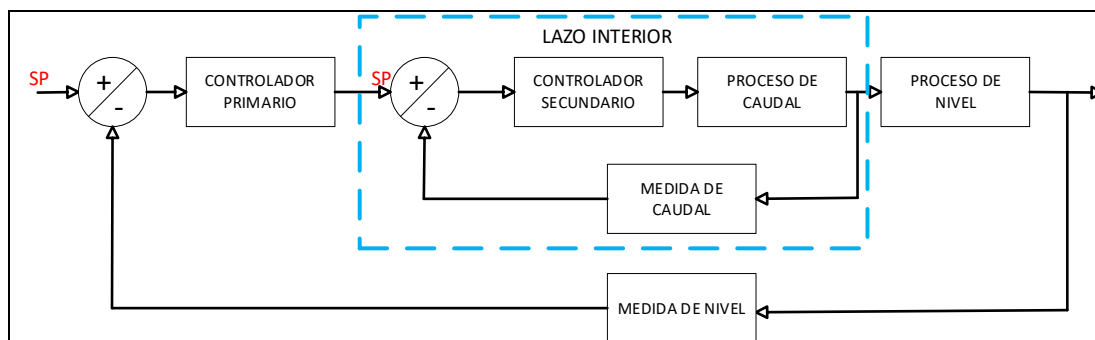
Control en cascada

Existen diferentes procesos con lazos simples de control realimentados entre los que se produce interacción. Para eliminarlas es necesario medirla y controlarla por medio de un sistema de control denominado cascada (Sanchez, 2003).

El control en cascada es uno de los métodos más exitosos para mejorar el rendimiento del control de bucle único. Puede mejorar drásticamente el rendimiento de las estrategias de control, reduciendo tanto la desviación máxima como el error integral para las respuestas de perturbación. Dado que los cálculos requeridos son simples, el control en cascada se puede implementar con una amplia variedad de equipos analógicos y digitales, como se muestra en la Figura 2.

Figura 2

Diagrama de la técnica de control en cascada.



Nota. En la figura se observa cómo está conformado un sistema de control avanzado en forma específica el control en cascada. Tomado de Control Avanzado de Procesos. (Sanchez, 2003, pág. 245).

Modelamiento matemático

Es aquel donde se utiliza fórmulas matemáticas para representar la relación que tienen las variables, parámetros y restricciones en un proceso, dependiendo del diseño pueden servir para realizar predicciones del valor de las variables y evaluar efectos de una determinada actividad (Roldan, 2019).

Estaciones de nivel

La estación de nivel ND-0704 de la Universidad de las Fuerzas Armadas "ESPE", consta de un sistema que controla la variable nivel, en la cual se aprecia los siguientes elementos: una bomba centrífuga de 3 HP, una válvula de control proporcional, un rotámetro, un transmisor de nivel y un transmisor de caudal, la cual permite realizar diferentes técnicas de control clásico y avanzado como se muestra en la Figura 3.

Figura 3

Estación de nivel ND-0704



Nota. En la figura se observa la estación de nivel ND-0704. Tomado de Laboratorio de redes Industriales y control de procesos de la “ESPE”.

Transmisor de nivel Rosemount 5400 Series

El transmisor de nivel Rosemount 5400 Series es un dispositivo utilizado para medir y controlar el nivel de líquidos en tanques y recipientes. Está diseñado por Emerson, una empresa líder en tecnología de automatización industrial. (Emerson, 2012).

El transmisor de nivel Rosemount 5400 Series utiliza tecnología de radar de onda guiada (TDR, por sus siglas en inglés) para medir con precisión el nivel de líquidos en aplicaciones industriales. El TDR emite pulsos de microondas a lo largo de una guía de onda que se sumerge en el líquido. Estos pulsos se reflejan cuando encuentran una interfaz entre medios de diferente permitividad, como el líquido y el aire o el líquido y un sólido. El tiempo que tarda en recibir el pulso reflejado se utiliza para determinar la distancia hasta la interfaz y, por lo tanto, el nivel del líquido. (Emerson, 2012).

El transmisor de nivel Rosemount 5400 Series ofrece diversas ventajas, como una medición precisa y confiable, incluso en condiciones difíciles con presencia de vapor, polvo, turbulencias o cambios de temperatura. También es resistente a la corrosión y puede utilizarse en una amplia variedad de líquidos, como agua, productos químicos, aceites, productos alimenticios y productos farmacéuticos. Además, el Rosemount 5400 Series cuenta con opciones de configuración y comunicación flexibles, lo que permite integrarse fácilmente en sistemas de control y monitoreo existentes. (Emerson, 2012). Como se muestra en la Figura 4.

Figura 4

Transmisor de Nivel Rosemount 5400 series



Nota. En la figura se aprecia el Transmisor de Nivel Rosemount 5400 series, el cual es un dispositivo para medir y controlar el Nivel de líquidos en tanques y recipientes. Tomado de Process Management por (Emerson, 2012).

Caudalímetro George Fischer 8150 series

El medidor de caudal Georg Fischer 8150 es un dispositivo utilizado para medir y monitorear el flujo de líquidos en aplicaciones industriales. Georg Fischer es un fabricante líder en sistemas de tuberías y soluciones de flujo de fluidos. (Fischer, 2014).

El medidor de caudal 8150 de Georg Fischer utiliza el principio de medición de ultrasonido para determinar el caudal del líquido que fluye a través de una tubería. Este

dispositivo mide el tiempo que tarda un pulso ultrasónico en viajar en contra y a favor del flujo del líquido, y utiliza esta información para calcular la velocidad y el volumen del flujo. (Fischer, 2014). Como se muestra en la Figura 5.

Figura 5

Caudalímetro George Fischer 8150 series



Nota. En la figura se aprecia el Caudalímetro George Fischer 8150 series, el cual es un dispositivo para medir y controlar Caudal. Tomado de +GF+ por (Fischer, 2014).

Pantalla thinker touch (Industrial Shield)

La pantalla thinker touch, de Industrial Shield, es una pantalla táctil diseñada específicamente para aplicaciones industriales y de automatización. Es producida por Industrial Shields, una empresa especializada en soluciones de automatización y control industrial. (Shields, 2019).

La pantalla thinker touch se caracteriza por ser robusta y resistente, lo que la hace adecuada para entornos industriales exigentes. Está diseñada para funcionar de manera confiable en condiciones ambientales adversas, como altas temperaturas, vibraciones y presencia de polvo o humedad. Además, posee una pantalla que cuenta con una interfaz táctil que permite la interacción con el sistema de control o la aplicación industrial correspondiente. (Shields, 2019). Como se muestra en la Figura 6.

Figura 6

Pantalla Thinker Touch (Industrial Shields)



Nota. En la figura se aprecia la pantalla Thinker Touch, la cual es una pantalla táctil diseñada específicamente para aplicaciones industriales. Tomado de Cadenza Electric por (Shields, 2019).

Obtención del modelo matemático

El modelo matemático se utiliza para representar la relación que tienen las variables, parámetros y restricciones en un proceso, dependiendo del diseño pueden servir para realizar predicciones del valor de las variables y evaluar efectos de una determinada actividad (Roldan, 2019).

Dentro de la obtención del modelo el objetivo que se persigue es hallar las ecuaciones del modelo matemático representadas en el Dominio de Laplace (s), esto se lo realiza mediante la utilización de herramientas de software de ingeniería como LabVIEW (Instruments, LABVIEW, 2023) para la obtención de datos tanto de entrada como de salida y Matlab (MathWorks, 2021) para procesar la ecuación del modelo matemático como se observa en la Figura 7.

Figura 7

Variables de un proceso



Nota. En la figura se observa el diagrama de bloques de manera general de las variables de un proceso. Tomado de Sistema de Control para Ingeniería (Nise, 2006).

Controladores avanzados

A cualquier algoritmo o estrategia que se desvíe del control clásico lineal, se entiende por Control Avanzado. Estos controladores aparecieron cuando se originó un mayor grado de complejidad que presentan las plantas modernas de procesos y los avances computacionales y tecnológicos que permiten la implementación de sofisticados algoritmos que no se podían observar usando tecnología analógica. (Cubillos, 2019).

Los controladores avanzados son sistemas de control más complejos y sofisticados que van más allá de los controladores clásicos, como los controladores proporcionales, integrales y derivativos (PID). Estos controladores avanzados se utilizan en aplicaciones donde se requiere un control más preciso, rápido y adaptable para satisfacer los requerimientos específicos de los sistemas, existen diferentes tipos los cuales son:

Control predictivo (MPC)

El control predictivo, también conocido como controlador de modelo predictivo, utiliza un modelo matemático del sistema para predecir su comportamiento futuro. Utiliza esta información para calcular las acciones de control óptimas y anticiparse a posibles perturbaciones o cambios en el proceso. El MPC es especialmente efectivo en sistemas con restricciones y en aquellos que requieren ajustes de control frecuentes.

Control adaptativo

Los controladores adaptativos ajustan automáticamente sus parámetros de control en función de las condiciones cambiantes del sistema. Estos controladores utilizan algoritmos y técnicas para identificar y adaptarse a las características y variaciones del proceso en tiempo real. El control adaptativo es útil en sistemas con comportamientos no lineales o que están sujetos a cambios o incertidumbre.

Control difuso (Fuzzy)

El control difuso, o controlador fuzzy, utiliza la lógica difusa para tratar con la incertidumbre y la imprecisión presentes en muchos sistemas. En lugar de emplear ecuaciones matemáticas precisas, utiliza variables lingüísticas y reglas lingüísticas para describir y controlar el comportamiento del sistema. El control difuso es especialmente útil en sistemas complejos o con información incompleta o vaga.

Control por redes neuronales

Los controladores basados en redes neuronales emplean algoritmos de aprendizaje automático para modelar y controlar sistemas complejos. Estas redes neuronales artificiales aprenden de los datos y la retroalimentación para adaptarse y mejorar el control del sistema. El control por redes neuronales es especialmente útil en sistemas no lineales, con múltiples variables y en aquellos donde el modelo matemático preciso es desconocido o difícil de obtener.

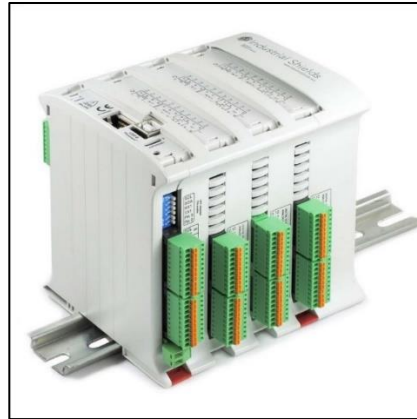
PLC M-Duino

El PLC M-Duino es una placa de control basada en Arduino que combina las capacidades de programación y la flexibilidad de un microcontrolador Arduino con las funcionalidades propias de un controlador lógico programable (PLC). Diseñado específicamente para aplicaciones de automatización industrial, el PLC M-Duino ofrece un control confiable y versátil para una amplia gama de sistemas, se destaca por su integración de Arduino, lo que

permite aprovechar las ventajas de su entorno de programación y la gran comunidad de usuarios y desarrolladores que respaldan esta plataforma. Al utilizar el entorno Arduino IDE, los usuarios pueden aprovechar las librerías y recursos disponibles, así como su experiencia previa en programación Arduino (Shields, 2019).

Este PLC cuenta con una amplia variedad de entradas y salidas digitales y analógicas. Las entradas digitales permiten interactuar con sensores y dispositivos externos, mientras que las salidas digitales se utilizan para controlar actuadores y dispositivos de salida. Por otro lado, las entradas y salidas analógicas permiten leer y controlar señales analógicas, como sensores de temperatura, señales de voltaje, entre otros, también ofrece opciones de comunicación, como puertos RS485 y Ethernet, que facilitan la integración del PLC en una red industrial. Esto permite la comunicación con otros dispositivos y sistemas para realizar tareas de supervisión y control remoto (Shields, 2019).

Además de su flexibilidad, el PLC M-Duino también es escalable, lo que significa que se puede ampliar con módulos y tarjetas de expansión adicionales según las necesidades específicas del sistema de control. Esto permite agregar funcionalidades adicionales, como comunicación inalámbrica, almacenamiento de datos o control PID, para adaptarse a los requerimientos particulares de cada aplicación (Shields, 2019). Como se observa en la Figura 8.

Figura 8*PLC-M-DUINO*

Nota. En la figura se aprecia el PLC-M-duino, el cual es un equipo industrial basado en Arduino. Tomado de Cadenza Electric por (Shields, 2019).

La versatilidad que ofrece el PLC de Arduino también lo podemos encontrar en la posibilidad que presenta de poder usar distintos sistemas operativos como Linux, Windows o Android. Esto resulta fundamental para conseguir una digitalización eficiente que pueda ofrecer los mejores resultados.

También permite la conexión mediante USB con todas las utilidades que esto presenta sobre todo en lo referente a la transmisión de datos. Es una de las múltiples facilidades que permite el PLC de Arduino y que permiten además que su configuración sea más accesible. También existe la posibilidad de conectar una tarjeta de memoria externa. (Shields, 2019).

Tecnología IIoT

IIoT es una abreviatura que significa "Internet Industrial de las cosas", la cual se refiere a una red de dispositivos interconectados en el sector industrial. Esta red es un subconjunto del Internet de las cosas (IoT). Una de las características más destacadas de los dispositivos conectados en la red IIoT es que son capaces de transmitir datos sin la necesidad de interacción persona a persona o persona a computadora. Para lograr esto, los dispositivos

conectados se comunican a través de una puerta de enlace, que es un servidor físico que se encarga de filtrar y enviar los datos a otros dispositivos y aplicaciones de software. (AG, 2020)

Protocolos de comunicación IIoT

Los protocolos tienen el potencial de conectar dispositivos industriales con plataformas IIoT (Semle, 2016). Dentro de los más comunes tenemos:

HTTP (REST/JSON)

HTTP (Protocolo de Transferencia de Hipertexto, por sus siglas en inglés) es un protocolo ampliamente utilizado en las tecnologías de la información y en la web, que opera en un modelo cliente/servidor sin conexión. Debido a la amplia disponibilidad de numerosas herramientas de código abierto que utilizan HTTP, así como las bibliotecas HTTP disponibles en todos los lenguajes de programación, el acceso a este protocolo es muy accesible.

MQTT

MQTT (Message Queuing Telemetry Transport, 'Transporte de Telemetría mediante Cola de Mensajes') es un protocolo de estilo publicar/suscribir que ha sido diseñado para aplicaciones de SCADA y redes remotas. Este protocolo se caracteriza por tener un encabezado mínimo de tan solo dos bytes y por ofrecer comunicaciones confiables. Además, MQTT se destaca por su simplicidad. Al igual que HTTP, el contenido de los mensajes MQTT es específico de la aplicación, y la mayoría de las implementaciones utilizan un formato personalizado en JSON o en binario.

CoAP

El protocolo CoAP (Constrained Application Protocol, 'Protocolo de Aplicación Restringida') fue desarrollado por IETF (Internet Engineering Task Force, 'Grupo de Trabajo de Ingeniería de Internet') con el objetivo de ofrecer compatibilidad con HTTP utilizando una carga mínima.

DDS

El protocolo DDS (Data Distribution Service, 'Servicio de Distribución de Datos') es un enfoque de publicación/suscripción que se centra en la comunicación en el límite de la red. DDS es un estándar abierto gestionado por OMG (Object Management Group, 'Grupo de Gestión de Objetos').

AMQP

AMQP (Advanced Message Queuing Protocol) es un protocolo adicional de tipo publicar/suscribir que se originó en el sector de servicios financieros. Aunque tiene cierta presencia en las tecnologías de la información y las comunicaciones (TIC), su adopción en la industria es relativamente limitada.

APK-Tablet

APK es una abreviatura que significa "Paquete de Aplicaciones para Android". Es un archivo ejecutable que contiene aplicaciones para dispositivos Android. Los archivos con extensión .APK son paquetes diseñados para el sistema operativo Android, y son una variante del formato JAR de Java. Estos paquetes se utilizan para distribuir e instalar componentes empaquetados para la plataforma Android, tanto en smartphones como en tablets. En resumen, APK es un formato de archivo que se utiliza para distribuir aplicaciones o juegos para Android, y nos permite instalarlos en nuestro dispositivo sin necesidad de utilizar Play Store. Como se observa en la Figura 9.

Figura 9

Aplicación APK.



Nota. En la figura se aprecia las APK, el cual es un tipo de formato de archivo para instalar en Android. Tomado de TD por (TechinDroid, 2023).

Linux

Linux es un sistema operativo de código abierto basado en Unix que fue creado en 1991 por Linus Torvalds. Debian es una distribución de Linux que se enfoca en la estabilidad y la seguridad del sistema. Debian se utiliza en una variedad de dispositivos, desde servidores hasta sistemas embebidos y computadoras personales. (Debian, 2021).

Una de las principales características de Linux es su capacidad para personalizar y adaptar el sistema operativo a las necesidades específicas del usuario. El código abierto permite a los desarrolladores y usuarios experimentados modificar y personalizar el sistema para adaptarse a sus necesidades. (Debian, 2021).

Debian es una distribución de Linux que se enfoca en la estabilidad y la seguridad del sistema. Debian utiliza un sistema de paquetes de software que facilita la instalación, actualización y eliminación de software. Los paquetes de software son gestionados por un sistema de gestión de paquetes llamado Advanced Package Tool (APT). (Debian, 2021).

La comunidad de usuarios y desarrolladores de Debian es muy activa y se enfoca en mantener el sistema operativo actualizado y seguro. Además, Debian cuenta con una gran

cantidad de software disponible, lo que permite a los usuarios personalizar el sistema operativo para adaptarse a sus necesidades específicas. (Debian, 2021).

En resumen, Linux es un sistema operativo de código abierto basado en Unix que se enfoca en la personalización y la adaptabilidad. Debian es una distribución de Linux que se enfoca en la estabilidad y la seguridad del sistema, y utiliza un sistema de paquetes de software para facilitar la instalación, actualización y eliminación de software. Ambos son muy populares y utilizados en todo el mundo debido a su flexibilidad y seguridad (Debian, 2021). Como se observa en la Figura 10.

Figura 10

Sistema operativo debian.



Nota. En la figura se aprecia a debian, el cual es la distribución Linux más popular en servidores. Tomado de MCPRO por (Pomeyrol, 2012).

Router TP-Link

El router TP-Link es un dispositivo de red confiable y de alta calidad que se utiliza para establecer conexiones y facilitar la comunicación en entornos industriales que emplean el protocolo Modbus TCP/IP. Con su sólida capacidad de enrutamiento y sus características avanzadas, el router TP-Link permite la interconexión eficiente y segura de dispositivos Modbus en una red industrial. Este router admite la funcionalidad Modbus TCP/IP, que es un protocolo de comunicación ampliamente utilizado en sistemas de automatización industrial y monitoreo.

Con su interfaz de administración intuitiva, el router TP-Link permite una fácil configuración de los parámetros de red necesarios para establecer y mantener conexiones Modbus TCP/IP. Además, su rendimiento y estabilidad garantizan una comunicación confiable y sin interrupciones, asegurando que los dispositivos Modbus puedan intercambiar datos de manera eficiente en el entorno industrial (Aguilar & Steenkiste, 2020). Como se observa en la Figura 11.

Figura 11

Router TP-Link.



Nota. En la figura se aprecia el Router TP-Link, el cual es un dispositivo de red que se utiliza para establecer conexiones y comunicación en entornos industriales. Tomado de Soporte Kartyy por (Kartyy, 2022).

Modbus serial

Modbus Serial es un protocolo de comunicación utilizado ampliamente en la automatización industrial para la transferencia de datos entre dispositivos electrónicos.

Fue desarrollado en la década de 1970 por Modicon (ahora parte de Schneider Electric) y se basa en una arquitectura maestro/esclavo, donde un dispositivo maestro (por ejemplo, un controlador lógico programable) solicita datos a uno o varios dispositivos esclavos (por ejemplo, sensores o actuadores). Utiliza una conexión serie (RS-232 o RS-485) para la comunicación, lo que significa que los datos se transmiten bit a bit a través de un único cable. Además, utiliza una estructura de mensajes simple y eficiente, con funciones predefinidas para leer y escribir

registros en los dispositivos esclavos. El protocolo Modbus Serial puede operar a diferentes velocidades de transmisión, como 1200, 2400, 4800, 9600, 19200, 38400, 57600 y 115200 baudios. (Modbus, 2020).

Modbus TCP-IP

Modbus TCP/IP es una variante del protocolo Modbus que utiliza Ethernet y TCP/IP para la comunicación en lugar de conexiones seriales. Permite la transmisión de datos a través de redes locales (LAN) e Internet, lo que facilita la integración de dispositivos Modbus en sistemas de automatización más amplios. En Modbus TCP/IP, los datos se encapsulan en paquetes TCP/IP y se transmiten a través de conexiones de red utilizando direcciones IP. Modbus TCP/IP permite la comunicación punto a punto y punto a multipunto, lo que significa que múltiples dispositivos maestros pueden comunicarse con varios dispositivos esclavos simultáneamente. (Modbus, 2020).

El uso de TCP/IP proporciona una mayor velocidad de transmisión y una mayor flexibilidad en comparación con las limitaciones de velocidad y distancia de las conexiones seriales. Es importante tener en cuenta que tanto Modbus Serial como Modbus TCP/IP son protocolos ampliamente utilizados en la industria, pero tienen diferencias en términos de la capa física y de transporte utilizada para la comunicación. (Modbus, 2020)

Node-RED

Node-RED es una herramienta de programación visual de código abierto que se utiliza para establecer conexiones y automatizar sistemas y servicios en línea. Su objetivo principal es facilitar la creación de flujos de trabajo y aplicaciones relacionadas con el Internet de las cosas (IoT), como se observa en la Figura 12. (Nodo-RED, 2013). Principio del formulario

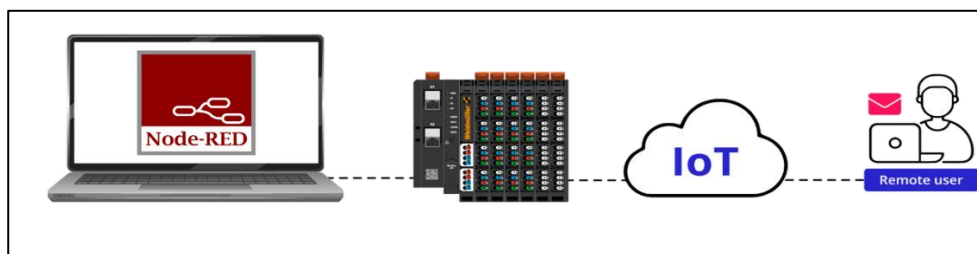
En Node-RED, los flujos se construyen utilizando nodos predefinidos que representan diferentes funciones o servicios. Estos nodos se arrastran y se conectan entre sí para formar un flujo lógico. Cada nodo realiza una acción específica, como recibir datos, transformarlos,

ejecutar una operación o enviar información a otro nodo. Se ejecuta en un servidor Node.js y se accede a través de un navegador web y proporciona una interfaz gráfica intuitiva que permite a los usuarios crear flujos de manera visual, sin necesidad de escribir código tradicional. (Node-RED, 2013).

Además, Node-RED cuenta con una amplia gama de extensiones y bibliotecas, lo que facilita la integración con diversos servicios y dispositivos. Esto lo convierte en una herramienta versátil para la creación rápida de aplicaciones y automatizaciones en entornos IoT. (Node-RED, 2013).

Figura 12

Software Node-RED.



Nota. En la figura se aprecia el Node-RED, el cual es una herramienta de programación visual de código abierto que se utiliza para establecer conexiones y automatizar sistemas y servicios en línea. Tomado de Realpars por (Realpars, 2023).

Convertor de 4-20mA a 2-10 V

Un convertor de corriente a voltaje (mA a V) se utiliza para convertir una señal de corriente de 4-20 mA a una señal de voltaje de 2-10 V. Este tipo de convertor es comúnmente utilizado en aplicaciones industriales donde se necesita transmitir una señal de corriente a un sistema o dispositivo que requiere una señal de voltaje. (Carter, 2002).

Resistencia de carga: Se conecta una resistencia de carga en serie con la señal de corriente de 4-20 mA. La resistencia debe tener un valor adecuado para generar el rango de voltaje deseado. Si consideramos un rango de voltaje de 2-10 V, se puede calcular el valor de la

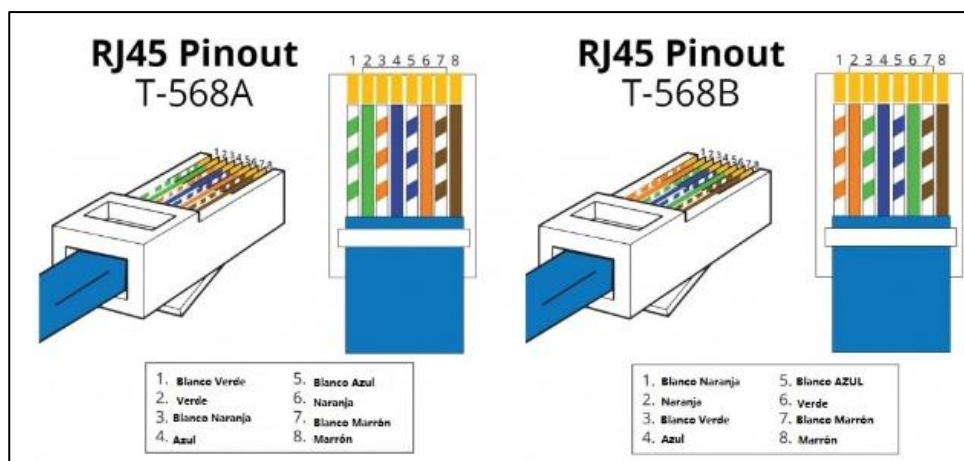
resistencia utilizando la Ley de Ohm ($V = I * R$), donde V es el voltaje deseado (10 V) e I es la corriente máxima (20 mA). En este caso, la resistencia sería de 500 ohmios. Es importante tener en cuenta que esta es solo una configuración básica y pueden existir otras opciones y circuitos más complejos para implementar un convertor de 4-20 mA a 2-10 V, dependiendo de los requisitos específicos y las características de los componentes utilizados. (Carter, 2002).

Cable Ethernet Directo y Cruzado estándar TIPO A y TIPO B

El cable directo y cruzado Ethernet se refiere a las dos variantes de cables utilizados para conectar dispositivos de red, como computadoras, switches, routers, y otros equipos de red. Los cables Ethernet transmiten datos a través de pares de cables trenzados en una configuración específica. (tragovom, 2013). Como se observa en la Figura 13.

Figura 13

Estándar Tipo A y Tipo B.



Nota. En la figura se aprecia el estándar Tipo A y Tipo B de un conector RJ45, el cual se utiliza en cada aplicación específica. Tomado de FS Community por (Worton, 2021).

Cable directo ethernet:

El cable directo se utiliza para conectar dispositivos de red diferentes, como una computadora a un switch o un router. En un cable directo, los pines en un extremo del cable se conectan directamente a los mismos pines en el otro extremo. El estándar de cableado utilizado en un cable directo Ethernet es conocido como TIA/EIA-568B para Ethernet tipo A y TIA/EIA-568A para Ethernet tipo B. (tragovom, 2013).

Cable cruzado ethernet:

El cable cruzado se utiliza para conectar dispositivos de red similares, como una computadora a otra computadora o un switch a otro switch. En un cable cruzado, los pines en un extremo del cable se cruzan o intercambian con los pines correspondientes en el otro extremo. Sin embargo, muchos dispositivos modernos, como los switches, tienen la capacidad de realizar el cruce automáticamente, lo que hace que el uso de cables cruzados sea menos común en la actualidad. Al igual que en el cable directo, el estándar de cableado utilizado en un cable cruzado Ethernet puede ser TIA/EIA-568B para Ethernet tipo A o TIA/EIA-568A para Ethernet tipo B. (tragovom, 2013)

Es importante destacar que los estándares de cableado mencionados (TIA/EIA-568A y TIA/EIA-568B) se refieren a las configuraciones de pines utilizadas en los extremos del cable. Ambos estándares son igualmente válidos, pero es importante asegurarse de que ambos extremos del cable estén terminados en la misma configuración (A o B) para garantizar la correcta comunicación. (tragovom, 2013)

Proceso autoregulado e integrante

Los términos "proceso autorregulado" y "proceso integrante" se refieren a conceptos utilizados en el ámbito del control de procesos y sistemas dinámicos.

Proceso autoregulado:

Un proceso autoregulado es aquel en el que existe un mecanismo o control interno que ajusta automáticamente las variables del sistema para mantenerlo en un estado deseado o estable. Este ajuste se basa en la retroalimentación o información del propio sistema. En un proceso autoregulado, el sistema puede realizar cambios internos para compensar perturbaciones o cambios externos y mantener su comportamiento dentro de ciertos límites establecidos. (Ogata, 2010).

Proceso integrante:

Un proceso integrante se refiere a un tipo de comportamiento o dinámica de un sistema en el cual la salida del sistema depende de la integración de la entrada a lo largo del tiempo. En un proceso integrante, la salida acumula la influencia de la entrada a medida que pasa el tiempo, lo que puede dar lugar a una respuesta gradual o acumulativa. (Ogata, 2010).

En resumen, un proceso autorregulado implica la capacidad de un sistema para ajustarse automáticamente y mantenerse dentro de ciertos límites deseados, mientras que un proceso integrante describe un comportamiento donde la salida acumula la influencia de la entrada a lo largo del tiempo. Estos conceptos son fundamentales en el diseño y análisis de sistemas de control y son utilizados para comprender y modelar diferentes tipos de sistemas dinámicos. (Ogata, 2010).

Software Matlab

"Según (MathWorks, 2021), MATLAB es una herramienta que integra un entorno de escritorio optimizado para análisis iterativos y procesos de diseño, junto con un lenguaje de programación que permite expresar directamente las operaciones matemáticas en matrices y arreglos."

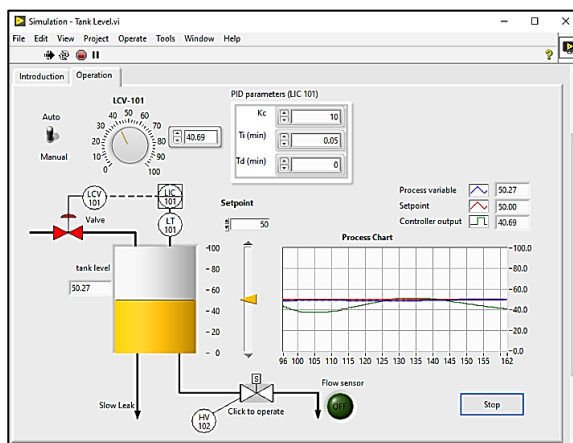
Software LabVIEW

El software LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un entorno de desarrollo integrado y un lenguaje de programación visual utilizado en la investigación científica y la ingeniería. LabVIEW se basa en un enfoque gráfico de programación, lo que facilita la creación de programas complejos mediante la conexión de bloques de función gráficos. Esta representación visual permite a los investigadores diseñar soluciones personalizadas sin necesidad de una formación extensa en programación (Instruments, LabVIEW, 2023)

LabVIEW ofrece capacidades de adquisición y procesamiento de datos. Permite adquirir datos de una amplia variedad de fuentes, como instrumentos de medición y sensores, mediante la integración de controladores y herramientas de comunicación. Además, proporciona una variedad de herramientas y algoritmos para el procesamiento y análisis de datos, incluyendo cálculos complejos, técnicas estadísticas y visualización de datos, como se muestra en la Figura 14.

Figura 14

Estación de nivel (LabVIEW)



Nota. En la figura se puede observar un ejemplo de control PID de una estación de nivel tomado desde el software LabVIEW (Instruments, LabVIEW, 2023).

En resumen, LabVIEW es una herramienta versátil y eficiente para la adquisición y procesamiento de datos en la investigación científica. Su enfoque visual de programación, integración con hardware diverso y capacidades de procesamiento de datos hacen de LabVIEW una opción atractiva para los investigadores que buscan automatizar tareas, obtener resultados precisos y avanzar en sus investigaciones. (Instruments, LabVIEW, 2023)

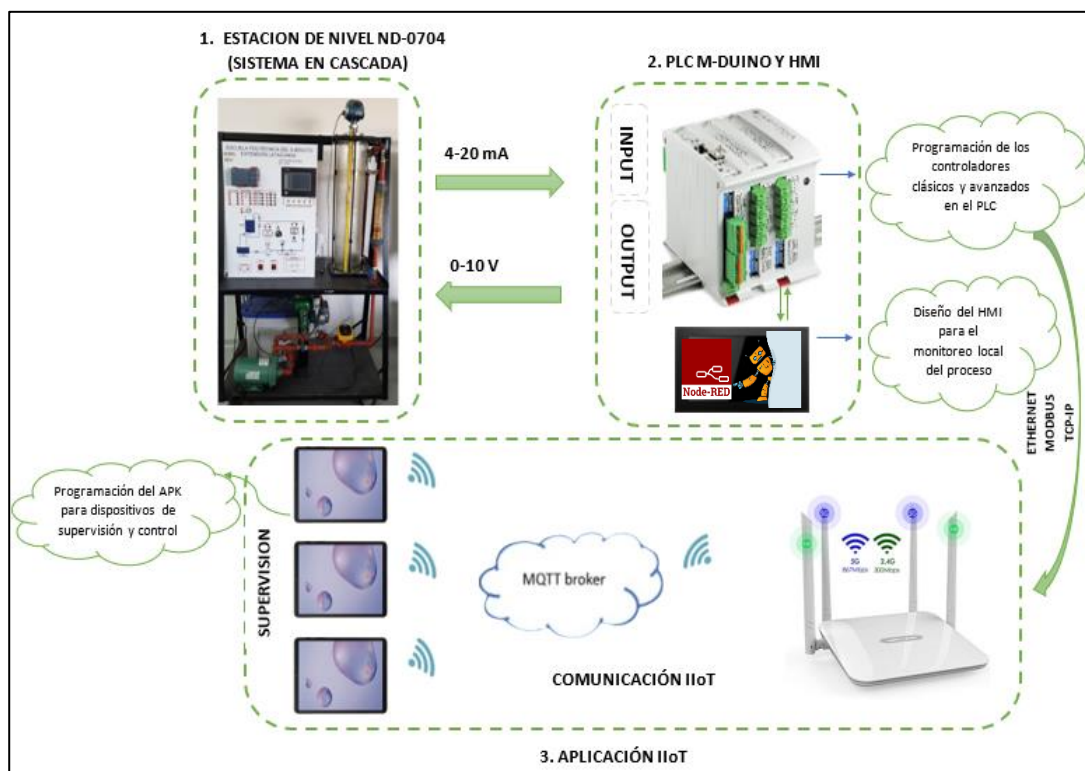
Capítulo III

Diseño e implementación

En el presente capítulo se detalla los pasos para la creación del sistema de supervisión y control IIoT de la estación de nivel ND-0704 (sistema de nivel y caudal) mediante la utilización de un PLC (M-DUINO 38AR+ Ethernet, I/Os Analógico/Digital/ Relé), para lo cual se diseña el esquema general de la implementación en el cual se muestra la comunicación entre los dispositivos utilizando protocolos industriales (Modbus TCP/IP) mediante un router TP /Link, pantalla ThinkerTouch, posteriormente la conexión a la nube mediante protocolo MQTT y la utilización del software Node-Red, como se aprecia en la Figura 15.

Figura 15

Sistema de supervisión y control IIoT de la estación de nivel ND-0704.



Identificación de los puertos de comunicación del PLC MDUINO 38AR+

A continuación, se detalla las entradas y salidas analógicas utilizadas:

Tabla 5

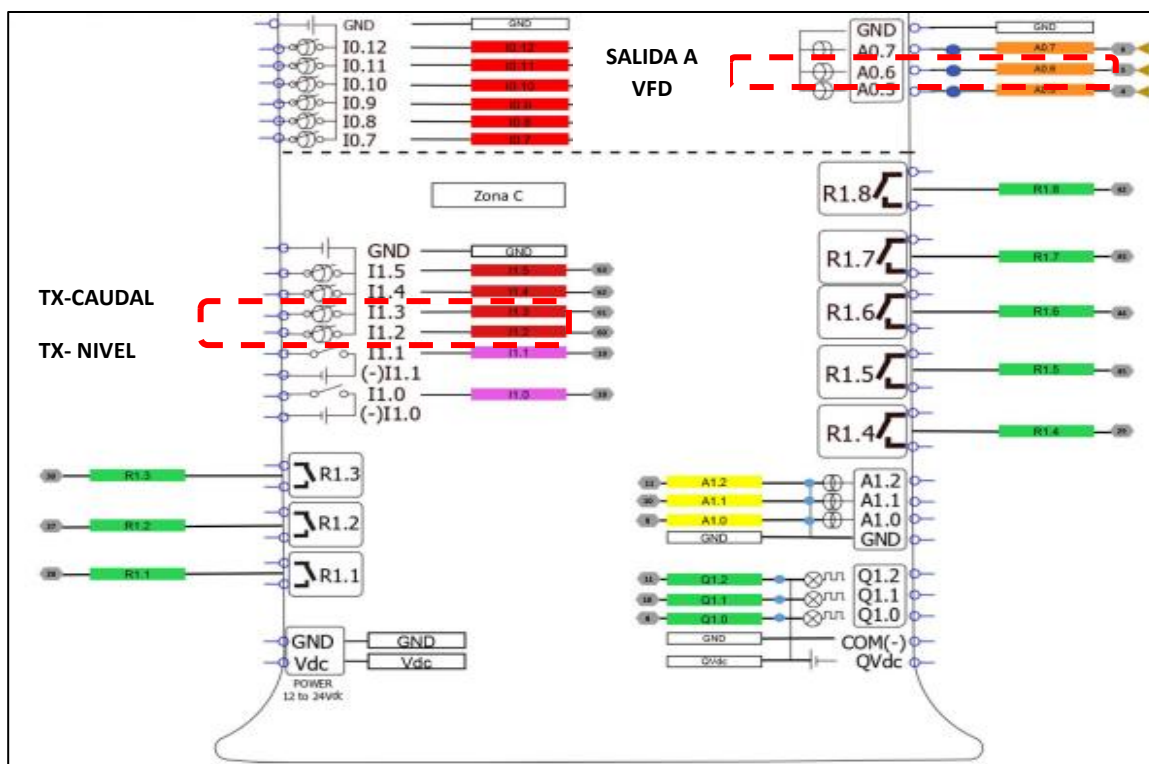
Tabla de identificación de puertos del PLC

PUERTO	DESCRIPCIÓN
GND	GND
I1.4	ENTRADA DEL TRANSMISOR DE NIVEL
I1.5	ENTRADA DEL TRANSMISOR DE CAUDAL
GND	GND
A0.6	SALIDA HACIA VFD

Como se muestra en la Figura 16.

Figura 16

Diagrama de puertos utilizados del PLC M-DUINO

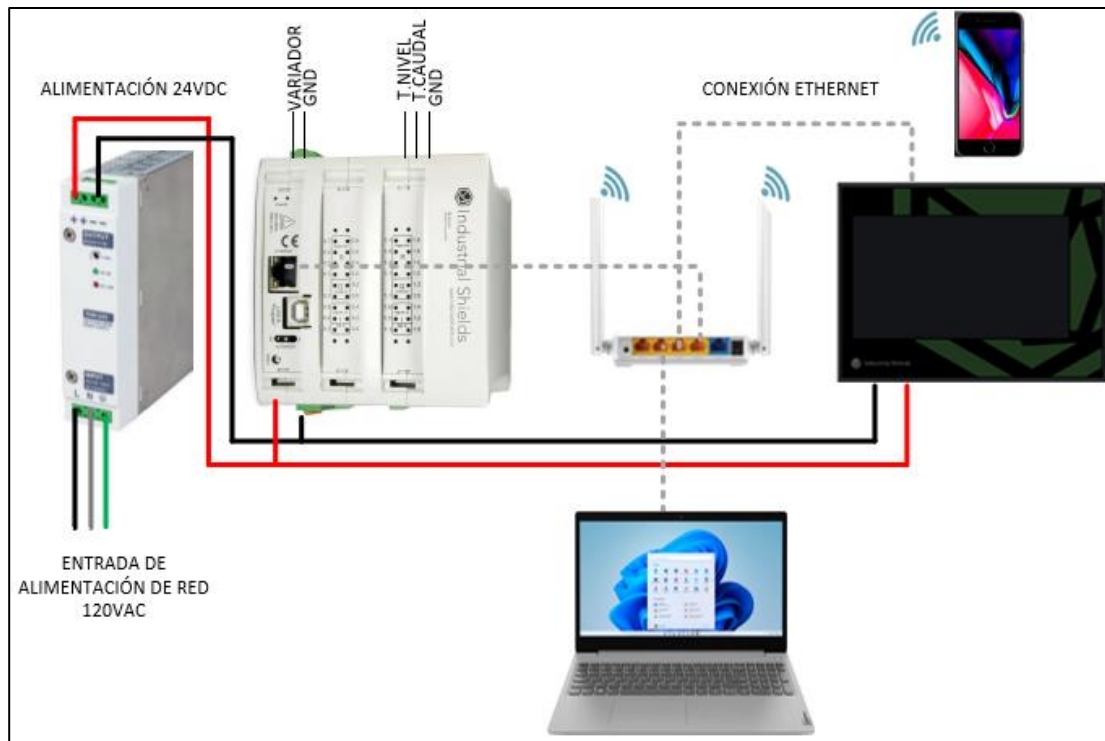


Conexión entre los dispositivos y red de alimentación principal.

Se realiza la conexión entre la fuente alimentación de 24VDC, para polarizar al M-Duino y la pantalla Tinker Touch, se tiene la salida analógica para el variador, las entradas para el transmisor de Nivel, transmisor de Caudal y GND respectivamente. Además, se tiene la conexión entre el M-Duino, el Router y la Tinker Touch y una Laptop por medio de cable Ethernet. Adicional se conecta dispositivos de manera inalámbrica por medio de Wifi. Después de completar la conexión apropiada, se continúa con la conexión principal a la red de alimentación de 120VAC, como se aprecia en la Figura 17.

Figura 17

Conexión entre dispositivos y red de alimentación principal.



Instalación del Software e instalación de tarjetas Industrial Shields

Para programar el PLC M-DUINO de Industrial Shields se utilizará Arduino IDE, mismo que está basado en lenguaje de programación C, además de proporcionar librerías que facilitan programación.

Industrial Shields brinda placas con pines y puertos definidos para programar, mediante la utilización de tarjetas que proporcionan los desarrolladores.

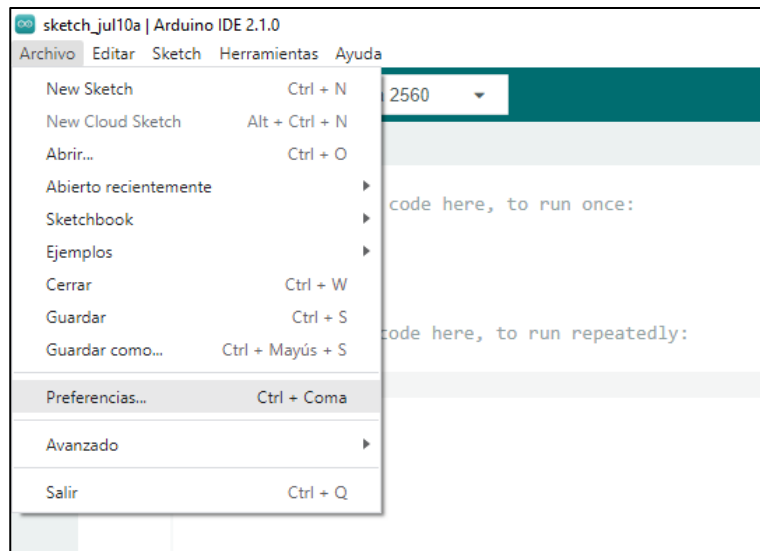
El requisito principal es tener instalado Arduino IDE 1.8.0 o superior, se recomienda tener siempre la última versión.

Pasos a seguir para instalar tarjetas Industrial Shields:

Abrir Arduino IDE y vaya a "Archivo -> Preferencias", esto se encuentra ubicado en la esquina superior izquierda. Como se observa en la Figura 18.

Figura 18

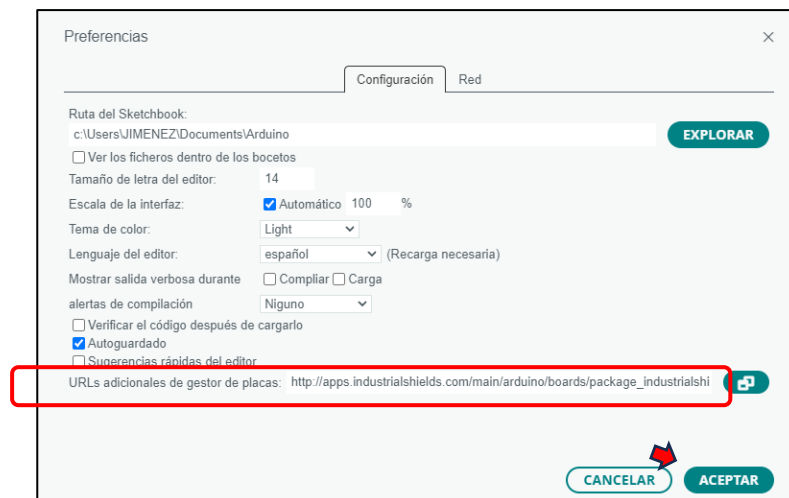
Arduino IDE "Archivo -> Preferencias".



Escribir en las URL del administrador de tarjetas adicionales y pulsar aceptar, como se aprecia en la figura 19

Figura 19

URL y Aceptar.



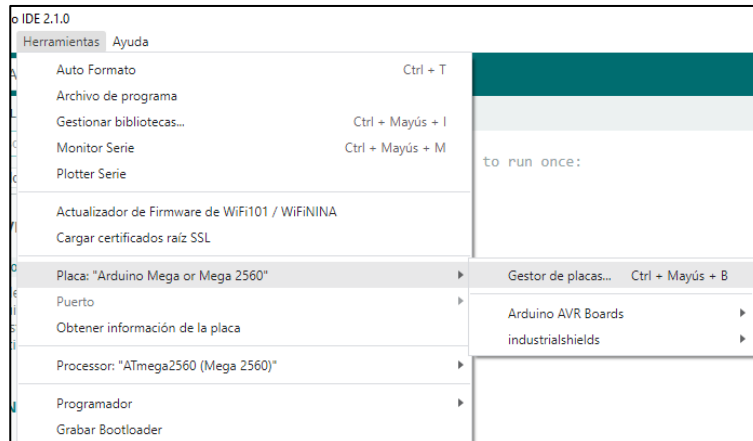
http://apps.industrialshields.com/main/arduino/boards/package_industrialshields_index.js

on

Ir a: Herramientas -> Tarjetas -> Administrador de tarjetas como se observa en la Figura 20.

Figura 20

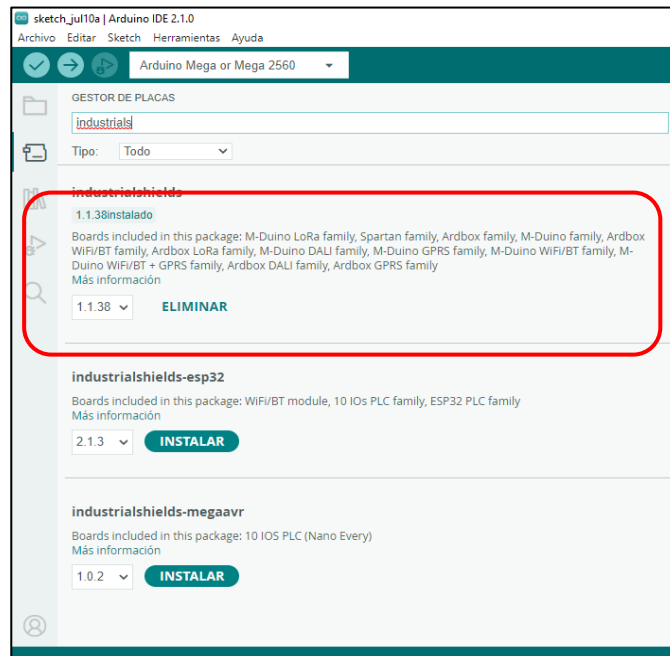
Herramientas -> Tarjetas -> Administrador de tableros.



Buscar "industrialshields" en el navegador, como se observa en la Figura 21.

Figura 21

"industrialshields" en el navegador.



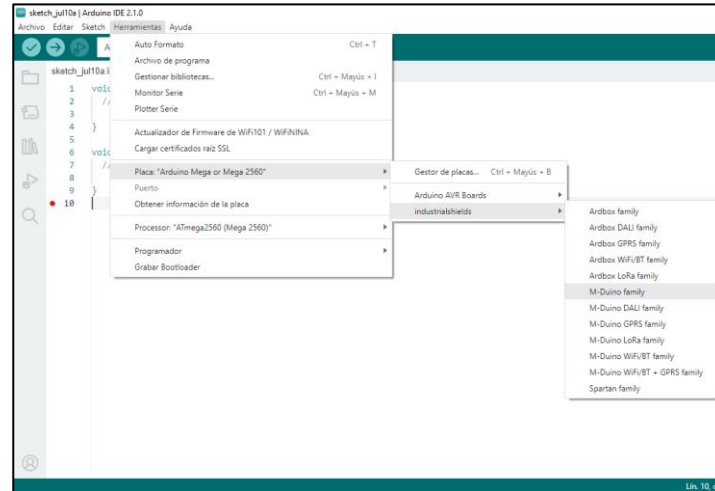
Seleccionar la última versión y hacer clic en instalar.

Pasos para observar las placas Industrial Shields:

Seleccionar la familia Ardbox o la familia M-Duino, aparecerá una opción adicional en Herramientas, como se observa en la Figura 22.

Figura 22

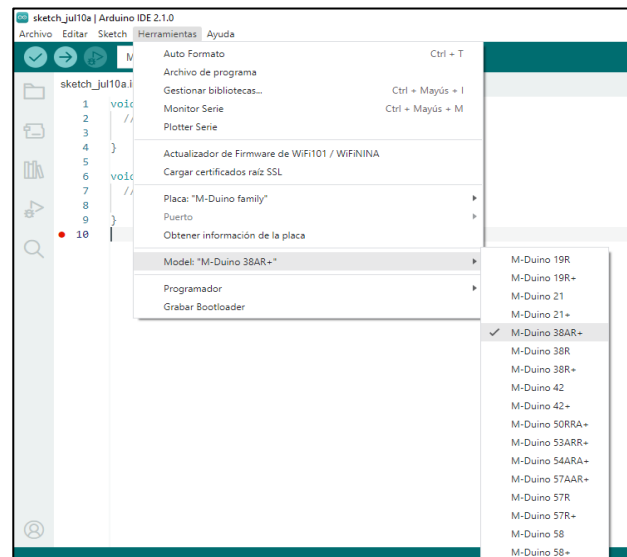
Selección de la familia Ardbox o la familia M-Duino.



Se selecciona el modelo específico para cada familia, en este caso es M-Duino 38AR+, como se observa en la Figura 23.

Figura 23

Selección del modelo específico M-Duino 38AR+.



Además, existen algunas bibliotecas adicionales que se las puede encontrar en Industrial Shields github en el siguiente link:

<https://github.com/IndustrialShields/>

Instalación del Software Node Red en la thinker touch

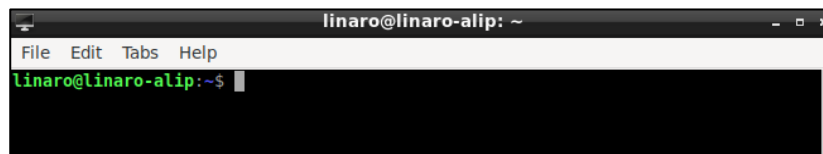
Para realizar la instalación localmente se dirige al siguiente link para revisar los requisitos previos a la instalación.

<https://nodered.org/docs/getting-started/local>

Abrir una Terminal LX en Linux (Debian), como se muestra en la Figura 24.

Figura 24

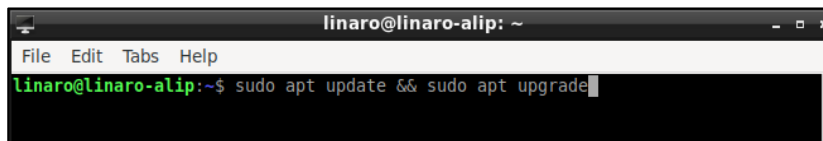
Terminal LX



Actualización de los paquetes del sistema operativo mediante el siguiente comando, como se observa en la Figura 25.

Figura 25

Actualización de paquetes



Nota. Este comando se utiliza siempre que se desee instalar un nuevo paquete o librerías.

Instalación del Node.js en Debian mediante el siguiente comando, como se observa en la Figura 26.

Figura 26

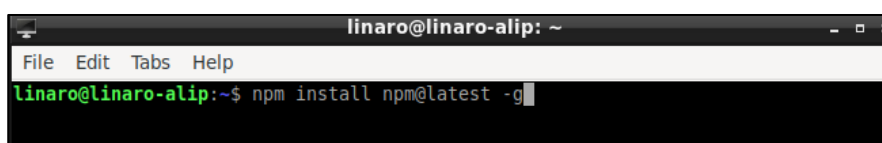
Instalación del Node.js en Debian.

```
sudo apt-get install -y nodejs
```

Instalación del Administrador de Paquetes (npm) en Debian mediante el siguiente comando, como se observa en la Figura 27.

Figura 27

Instalación del Administrador de Paquetes



```
linaro@linaro-alip: ~  
File Edit Tabs Help  
linaro@linaro-alip:~$ npm install npm@latest -g
```

Desactivar certificados SSL en ThinkerTouch mediante los siguientes comandos, como se muestra en la Figura 28.

Figura 28

Desactivar certificados SSL.



```
linaro@linaro-alip: ~  
File Edit Tabs Help  
linaro@linaro-alip:~$ sudo -i  
root@linaro-alip:~# curl_easy_setopt(curl, CURLOPT_SSL_VERIFYPEER, FALSE)
```

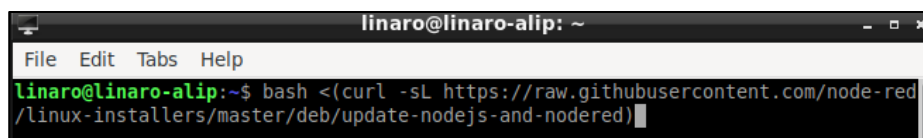
```
linaro@linaro-alip: ~  
File Edit Tabs Help  
linaro@linaro-alip:~$ sudo -i  
root@linaro-alip:~# -k/ --insecure
```

Nota. Este comando es importante para permitir el acceso a la instalación de Node Red IDE, ya que desbloquea un certificado de seguridad SSL que no viene incluido en el paquete de instalación.

Instalación y actualización del IDE Node-Red, como se muestra en la Figura 29, mediante el siguiente comando.

Figura 29

Instalación de Node-Red



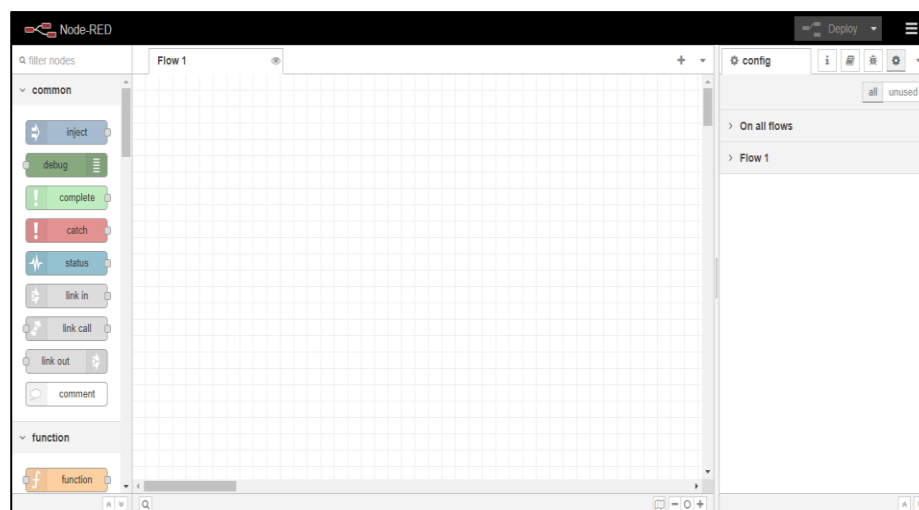
```
linaro@linaro-alip: ~  
File Edit Tabs Help  
linaro@linaro-alip:~$ bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

Abrir la interfaz de Node-Red mediante el siguiente comando y comprobarlo en un navegador Web, como se muestra en la Figura 30.

Figura 30

Comprobación de la instalación de Node-Red

<http://localhost:1880/>



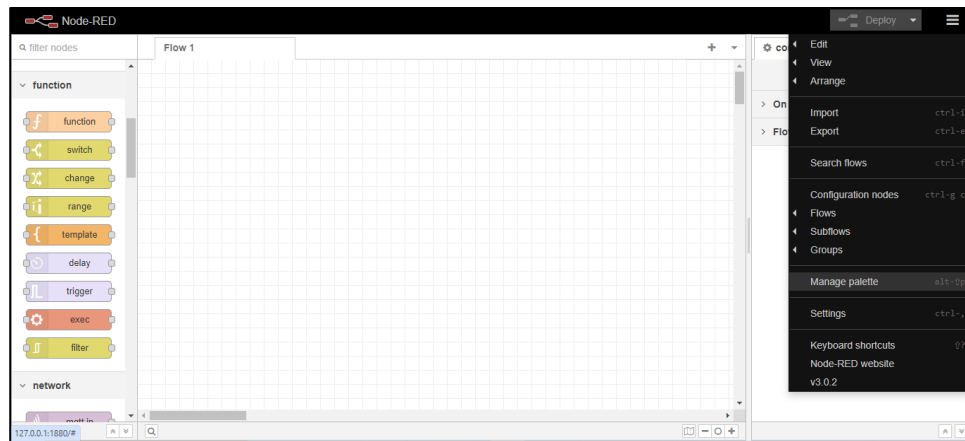
Instalación de librerías Dashboard en Node Red

Para la instalación de las diferentes librerías en Node-Red realizamos el siguiente proceso.

Se dirige a la opción de menú principal y seleccionamos Manage Palette, como se muestra en la Figura 31.

Figura 31

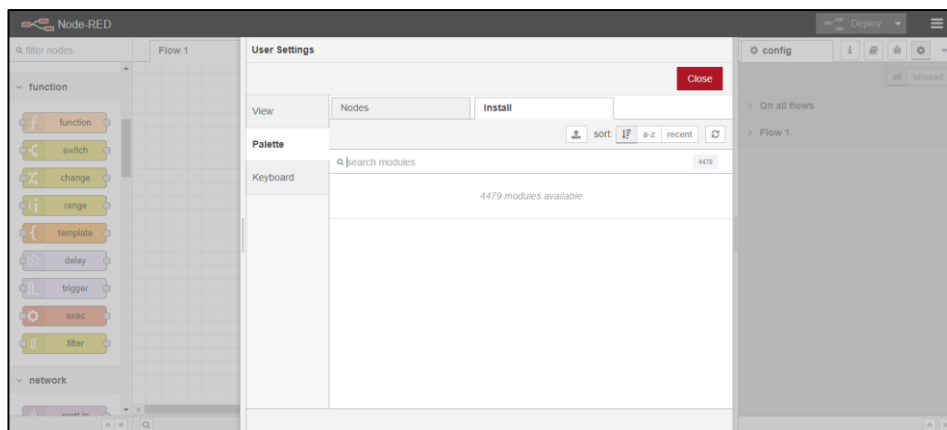
Proceso de instalación de librerías.



Seleccionar Install para buscar las diferentes funciones a instalarse, como se muestra en la Figura 32.

Figura 32

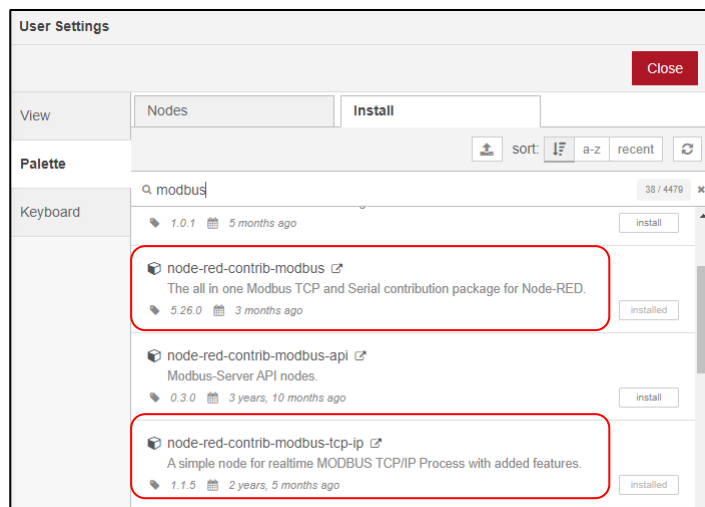
Realizar búsqueda de librerías.



Buscar la librería ModBus TCP-IP, como se muestra en la Figura 33.

Figura 33

Instalación de librería ModBus.

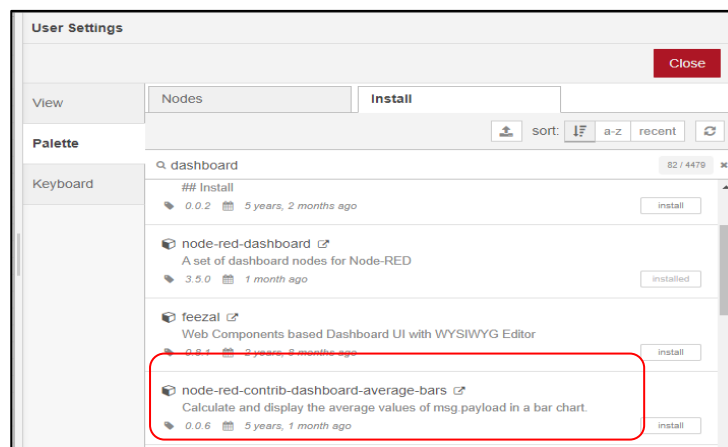


Nota. Las siguientes librerías se usan para realizar la comunicación ModBus entre el PLC M-DUINO y la THINKERTOUC.

Instalación de la librería dashboard, como se muestra en la Figura 34.

Figura 34

Instalación de librería dashboard.

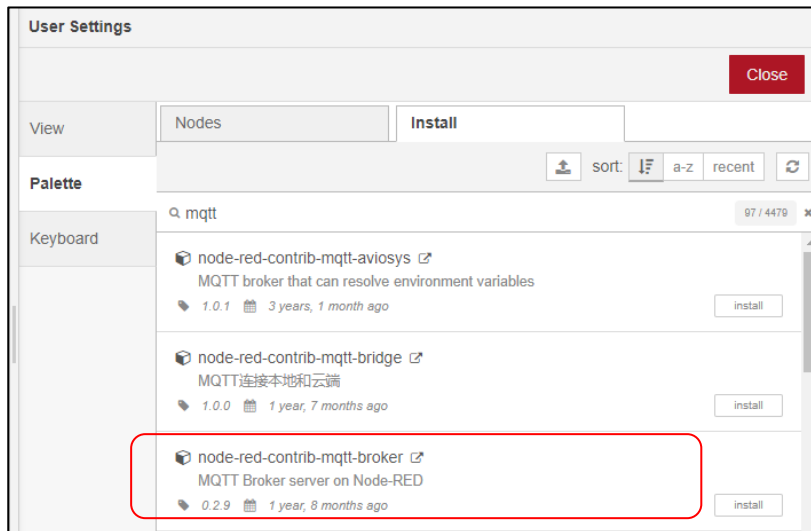


Nota. La siguiente librería se utiliza para realizar la creación de HMI mediante programación de imágenes SVG.

Instalación de la librería mqtt, como se muestra en la Figura 35.

Figura 35

Instalación de librería mqtt.



Nota. La siguiente librería se usa para realizar la comunicación con la nube mediante el protocolo mqtt.

Pruebas de comunicación entre dispositivos mediante Modbus TCP/IP y MQTT

Se realiza las pruebas mediante la comprobación de un ping al puerto de cada dispositivo para posteriormente realizar la comunicación Modbus TCP/IP y MQTT, para esto se necesita las direcciones IP en este caso de la Tinker Touch, del PLC-MDuino y de una laptop que este conectada a la red, además de crear una cuenta en la nube en FlowForge para obtener un acceso global hacia las variables de la planta.

Pruebas de comunicación al puerto.

Para comprobar la conexión al puerto se realiza un ping con la siguiente IP 192.168.0.102 que corresponde a la Tinker Touch, como se observa en la Figura 36.

Figura 36*Comunicación Tinker Touch*

```

File Edit Tabs Help
linaro@linaro-alip:~$ ping 192.168.0.102
PING 192.168.0.102 (192.168.0.102) 56(84) bytes of data.
64 bytes from 192.168.0.102: icmp_seq=1 ttl=64 time=0.117 ms
64 bytes from 192.168.0.102: icmp_seq=2 ttl=64 time=0.129 ms
64 bytes from 192.168.0.102: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.0.102: icmp_seq=4 ttl=64 time=0.129 ms
64 bytes from 192.168.0.102: icmp_seq=5 ttl=64 time=0.111 ms
64 bytes from 192.168.0.102: icmp_seq=6 ttl=64 time=0.083 ms
64 bytes from 192.168.0.102: icmp_seq=7 ttl=64 time=0.117 ms
64 bytes from 192.168.0.102: icmp_seq=8 ttl=64 time=0.127 ms
64 bytes from 192.168.0.102: icmp_seq=9 ttl=64 time=0.105 ms

```

Se realiza la prueba de comunicación al puerto con el comando ping y la siguiente IP 192.168.0.103 que corresponde al PLC-MDuino, como se observa en la Figura 37.

Figura 37*Comunicación PLC-MDuino.*

```

linaro@linaro-alip:~$ ping 192.168.0.103
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
64 bytes from 192.168.0.103: icmp_seq=1 ttl=128 time=0.424 ms
64 bytes from 192.168.0.103: icmp_seq=2 ttl=128 time=0.393 ms
64 bytes from 192.168.0.103: icmp_seq=3 ttl=128 time=0.428 ms
64 bytes from 192.168.0.103: icmp_seq=4 ttl=128 time=0.407 ms
64 bytes from 192.168.0.103: icmp_seq=5 ttl=128 time=0.408 ms
64 bytes from 192.168.0.103: icmp_seq=6 ttl=128 time=0.435 ms
64 bytes from 192.168.0.103: icmp_seq=7 ttl=128 time=0.427 ms
64 bytes from 192.168.0.103: icmp_seq=8 ttl=128 time=0.412 ms

```

Se realiza la prueba de comunicación al puerto con el comando ping y la siguiente IP 192.168.0.104 que corresponde a una laptop, como se observa en la Figura 38.

Figura 38*Comunicación Laptop.*

```

linaro@linaro-alip:~$ ping 192.168.0.104
PING 192.168.0.104 (192.168.0.104) 56(84) bytes of data.
64 bytes from 192.168.0.104: icmp_seq=1 ttl=128 time=4.18 ms
64 bytes from 192.168.0.104: icmp_seq=2 ttl=128 time=1.16 ms
64 bytes from 192.168.0.104: icmp_seq=3 ttl=128 time=2.31 ms
64 bytes from 192.168.0.104: icmp_seq=4 ttl=128 time=27.4 ms
64 bytes from 192.168.0.104: icmp_seq=5 ttl=128 time=1.65 ms

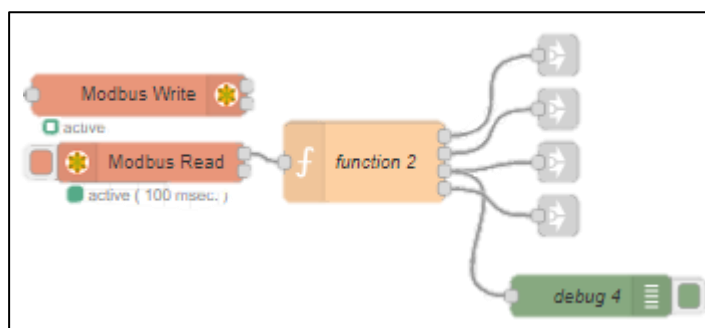
```

Pruebas de comunicación Modbus TCP/IP

Para verificar la comunicación Modbus TCP/IP entre el autómata M-DUINO y la ThinkerTouch hacemos uso de las tags Modbus Write y Modbus Read, como se observa en la Figura 39, el bloque "Modbus Write" se utiliza para enviar datos o escribir información en dispositivos Modbus, mientras que el bloque "Modbus Read" se utiliza para leer datos desde dispositivos Modbus. Ambos bloques son componentes clave en Node-RED para interactuar con dispositivos y sistemas basados en el protocolo Modbus.

Figura 39

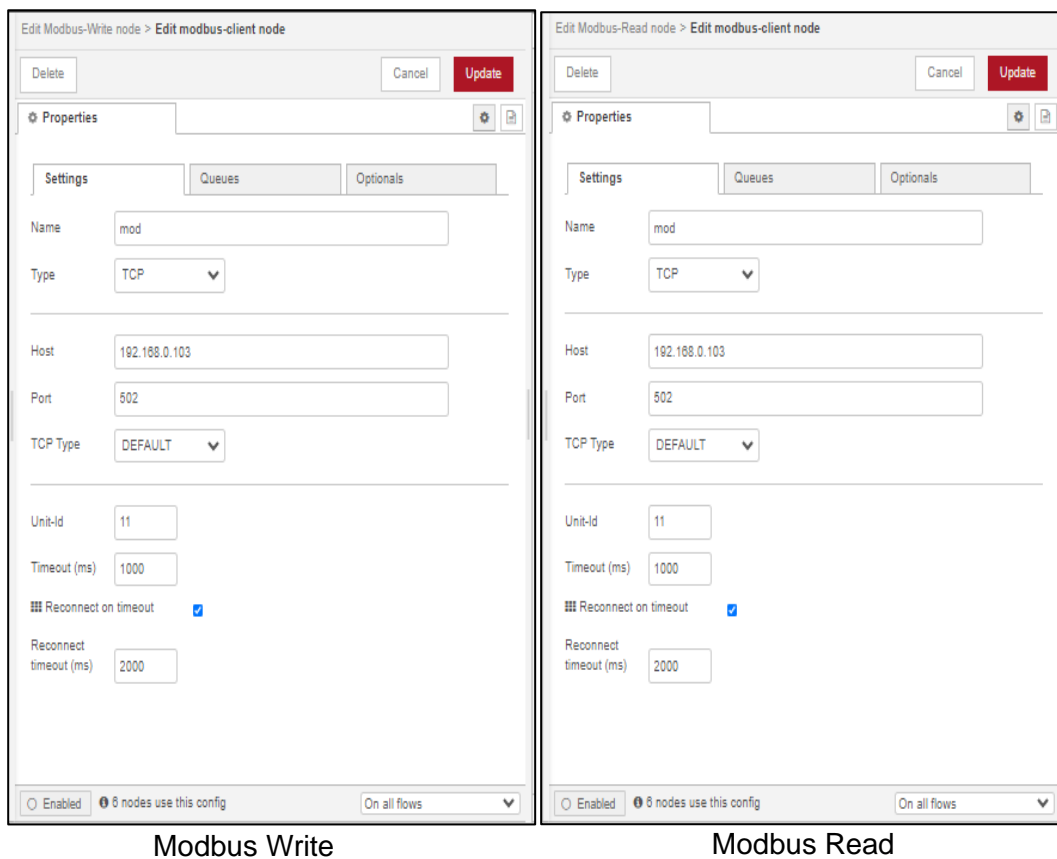
Bloques de programación Modbus write y Modbus Read (Node-Red)



En la configuración de parámetros del bloque Modbus Write y Modbus Read se debe ingresar la dirección IP que apunta al autómata, el puerto de comunicación que se está utilizando, y el tipo de protocolo de comunicación que para el presente caso sería Modbus TCP/IP, como se muestra en la Figura 40.

Figura 40

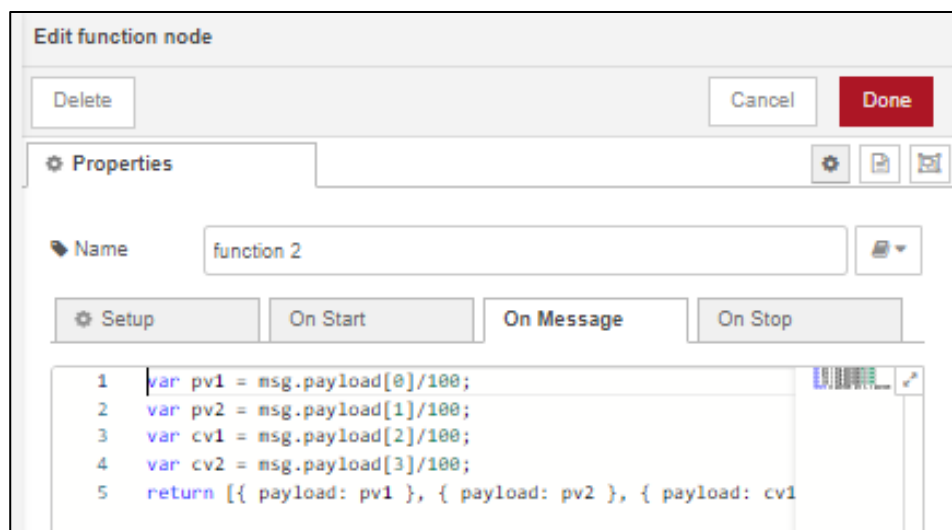
Configuración de programación Modbus Write y Modbus Read (Node-Red)



El bloque "Function Node" en Node-RED permite agregar código personalizado para realizar transformaciones de datos, lógica condicional, manipulación de mensajes, integración de bibliotecas, implementar lógica condicional, acceder a bibliotecas externas. Es una herramienta flexible que amplía la funcionalidad de los flujos de trabajo y brinda un mayor control sobre el procesamiento de datos adicional, como se observa en la Figura 41, donde se procesa los datos de la variable del proceso (PV) y la variable de control (CV).

Figura 41

Configuración del bloque Function Node.

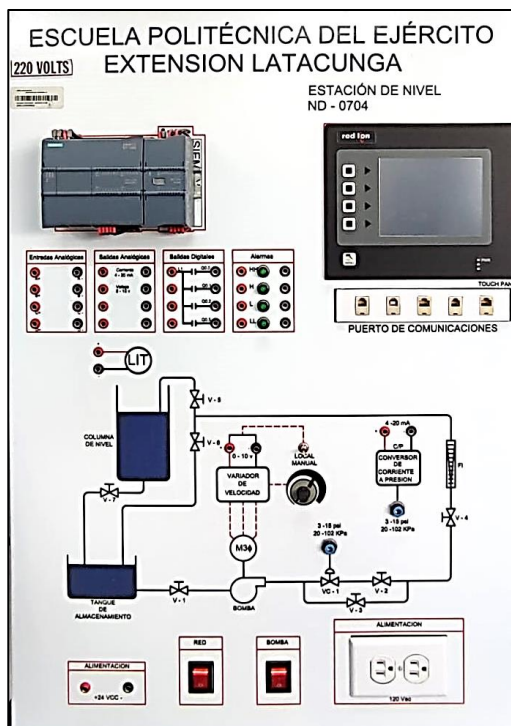


Identificación de la estación de nivel ND-0704

La estación de nivel ND-0704, tiene la posibilidad de controlar dos variables como son caudal y nivel, misma que consta de un sistema de tuberías interconectadas y control eléctrico mediante varios elementos como son: variador de frecuencia, una bomba de 3 HP marca Thebe, un PLC S7-1200, un transmisor de nivel Rosemount 5400, un transmisor de caudal George Fisher 8550, una pantalla Red Lion y una válvula de control Fisher, como se muestra en la Figura 42.

Figura 42

Estación de Nivel ND-0704.



El presente proyecto se basa en la implementación del sistema de supervisión y control IIOT basado en el PLC M-DUINO 38AR+ y una ThinkerTouch Industrial Shields, por medio de los cuales se realizará el respectivo control de las variables mencionadas (Nivel-Caudal), utilizando ciertos elementos de la estación de nivel (Variador, Bomba, Transmisores), mediante los cuales se recibirá valores de las señales de los elementos primarios y se actuará de acuerdo a la respuesta del controlador, como se muestra en la Figura 43.

Figura 43

Módulo didáctico IIoT.



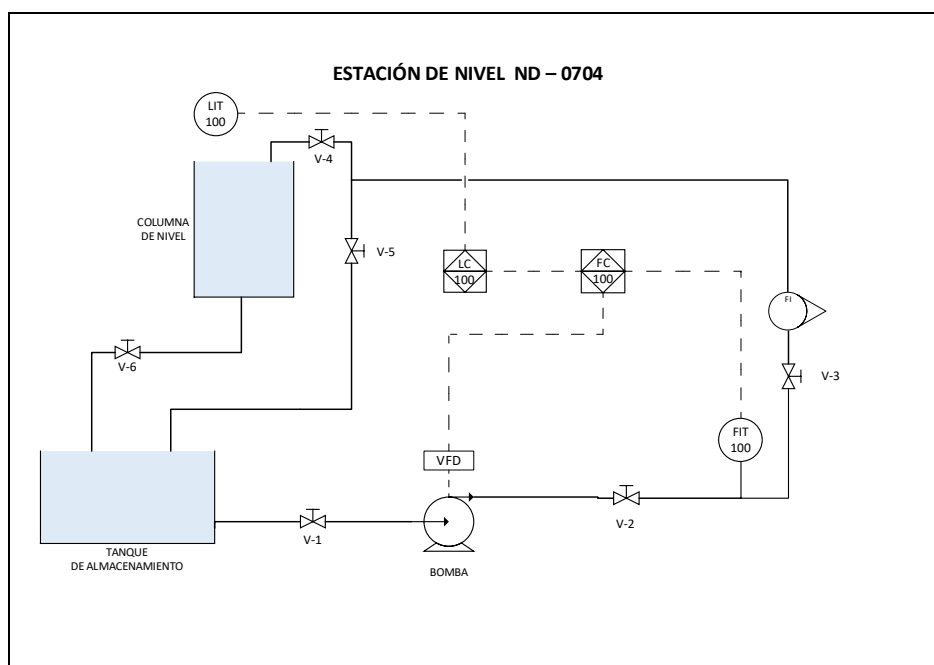
Diseño del diagrama P&ID

Para el diseño del diagrama P&ID de la estación de nivel ND-0704, se basó en las Normas ISA 5.1 e ISA 5.3, con el fin de establecer estándares y buenas prácticas en la industria.

En el diseño del sistema de nivel y caudal se consideran: tanques cerrados, tanque reservorio, tuberías, válvulas manuales, bomba, también se incluyen equipos e instrumentos industriales como, un variador de frecuencia, un transmisor-indicador de Nivel, un transmisor-indicador de Caudal y un controlador en este caso es un PLC M-DUINO 38AR+, como se observa en la Figura 44.

Figura 44

Diagrama P&ID estación de Nivel ND-0704.



Dinámica del proceso y modelo matemático de la estación de Nivel ND-0704.

Para comprender la dinámica del proceso debemos diferenciar entre un proceso integrante y un autorregulado ya que el método de sintonía difiere mucho en cada caso, para el presente proyecto las variables se encuentran dentro de un proceso autorregulado por lo cual procedemos a obtener el modelo matemático de las variables nivel y caudal.

El modelo matemático se obtuvo mediante el uso de software que sirven para obtención y procesamiento de datos de la planta como son: LabView, Matlab, Kepserver EX6, Excel y ModBus Poll para comprobar que se efectúe la comunicación y verificación de la llegada de los Bits, este proceso se utiliza para obtener el modelo matemático de los controladores simples (Nivel-Caudal) y cascada.

Configuración de los software previos a la obtención del modelo matemático.

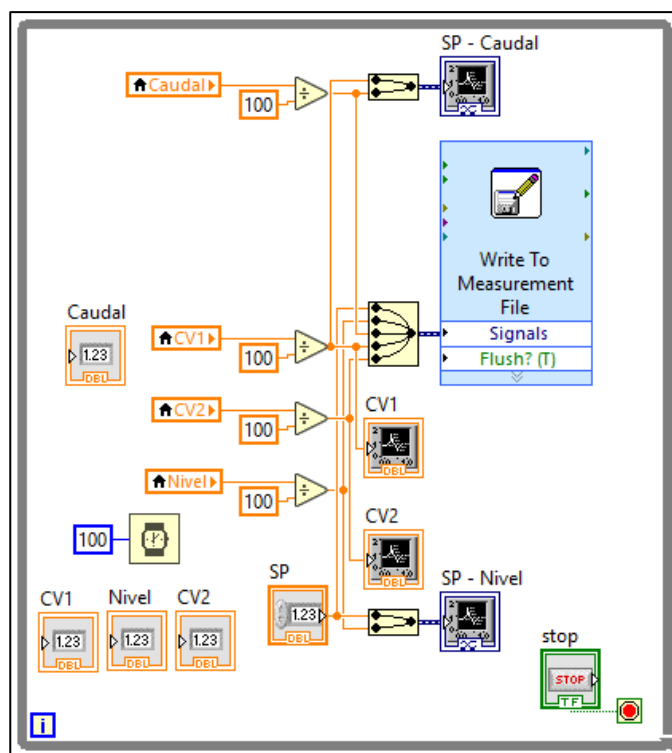
Configuración del LabVIEW

Para realizar la toma de datos mediante el programa de LabVIEW se realizó un VI previamente utilizando la función Write to Measurement.

La función "Write to Measurement File" en LabVIEW se utiliza para almacenar datos adquiridos o generados en un archivo de medición. Esto permite la captura, organización, análisis y posterior uso de los datos en aplicaciones de medición y adquisición de datos, como se muestra en la Figura 45.

Figura 45

Programación del VI para la toma de datos de la estación.



Configuración del Kepserver

Kep Server, también conocido como OPC Server, es una herramienta utilizada para conectar dispositivos industriales y sistemas de automatización mediante el estándar OPC (OLE for Process Control). OPC es un estándar de comunicación ampliamente utilizado en la

industria para facilitar el intercambio de datos entre diferentes equipos y software, como se muestra en la Figura 46 y 47 respectivamente.

Figura 46

Configuración del Kep Server.

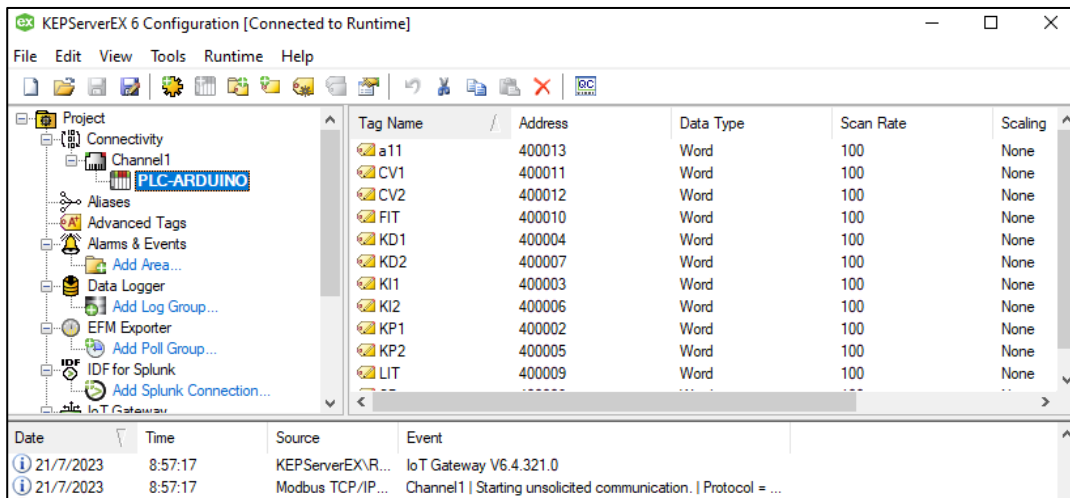
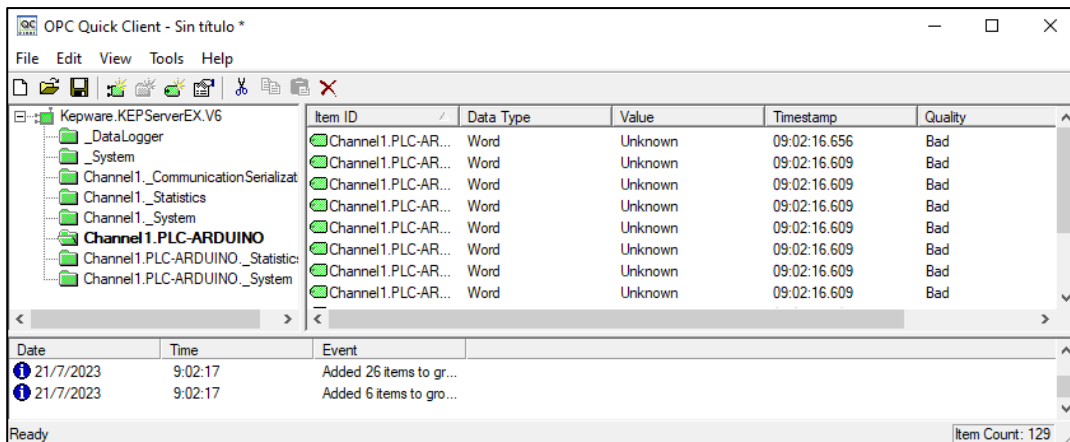


Figura 47

Configuración del OPC Quick Client.



Procesamiento de datos mediante el uso de Matlab y Excel

El uso de Matlab y Excel es indispensable para el procesamiento de datos, Excel se utilizó para guardar los datos en tipo vectores y luego exportarlos hacia el software Matlab donde mediante el comando IDENT se realiza la estimación del modelo como se muestra en las Figuras 48,49,50,51,52,53 y 54 respectivamente.

Figura 48

Importación de vectores desde Matlab hacia Excel.

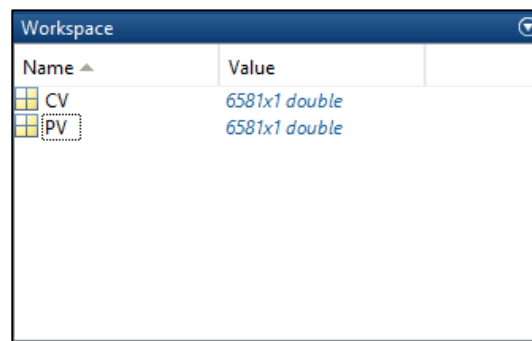
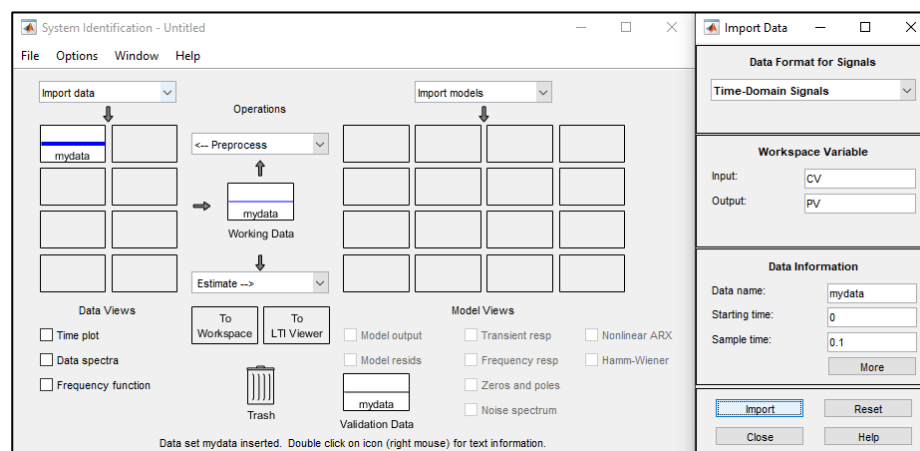


Figura 49

Ejecución del comando "ident" en Matlab



Nota. Los vectores se deben importar como dominio del tiempo.

Figura 50

Estimación del modelo del proceso

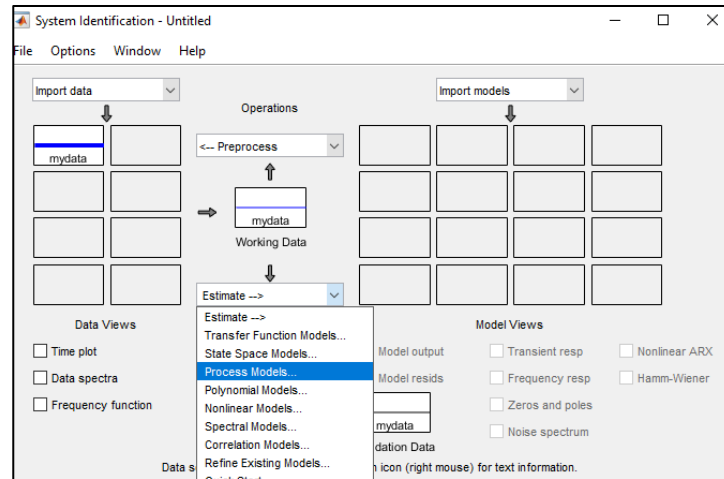


Figura 51

Click en "Estimate" para obtener el modelo matemático.

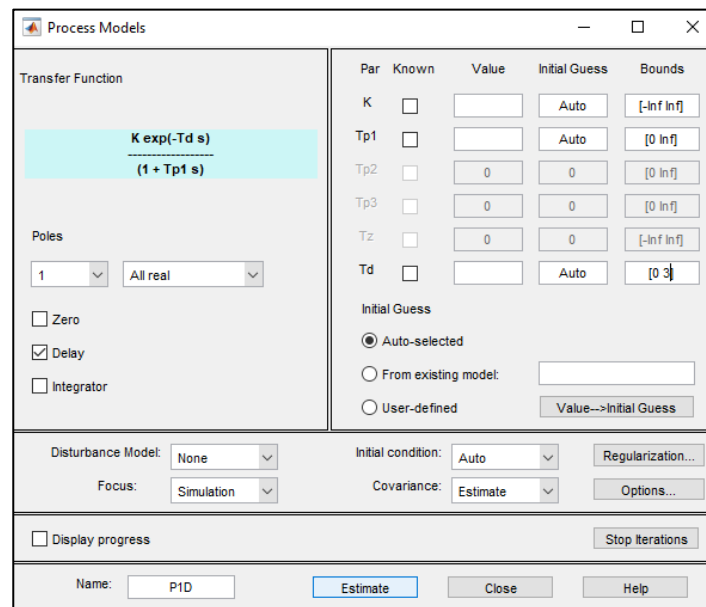


Figura 52

Seleccionar "Model output" para obtener el porcentaje de estimación.

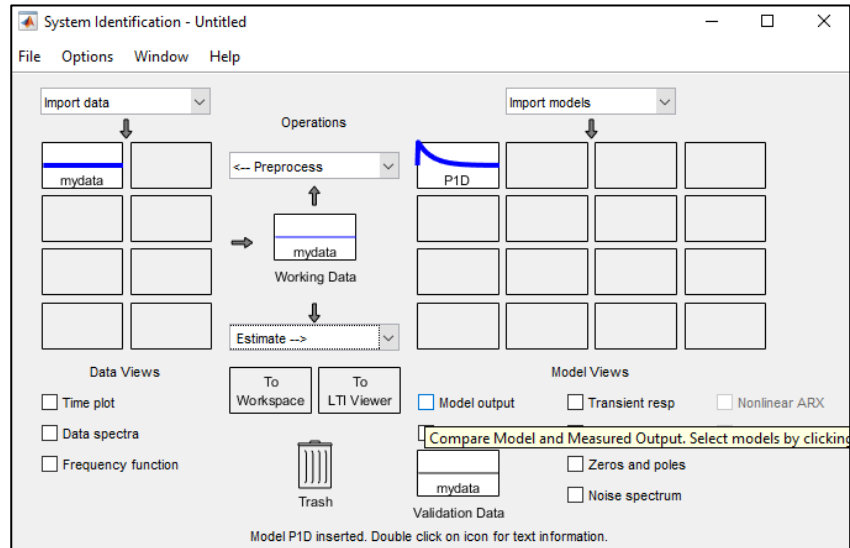


Figura 53

Porcentaje de aproximación del proceso de nivel

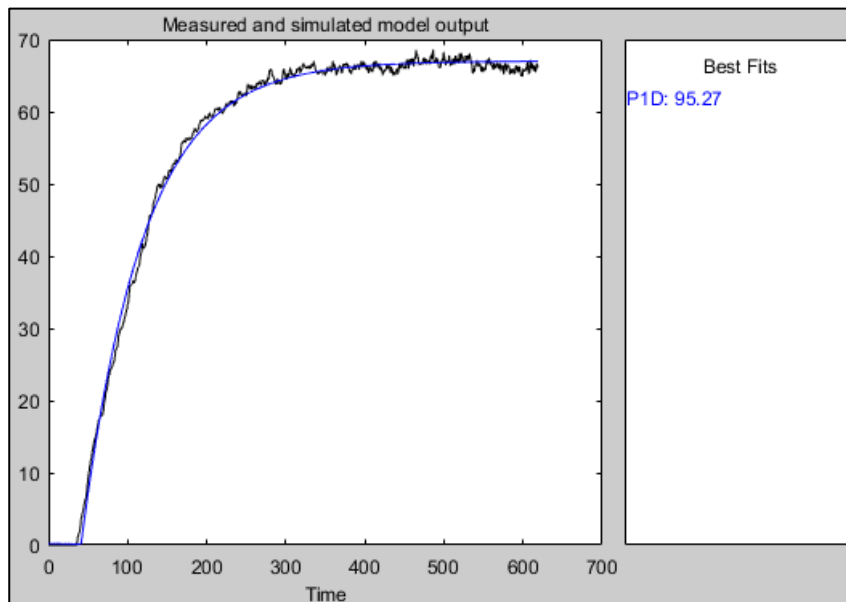
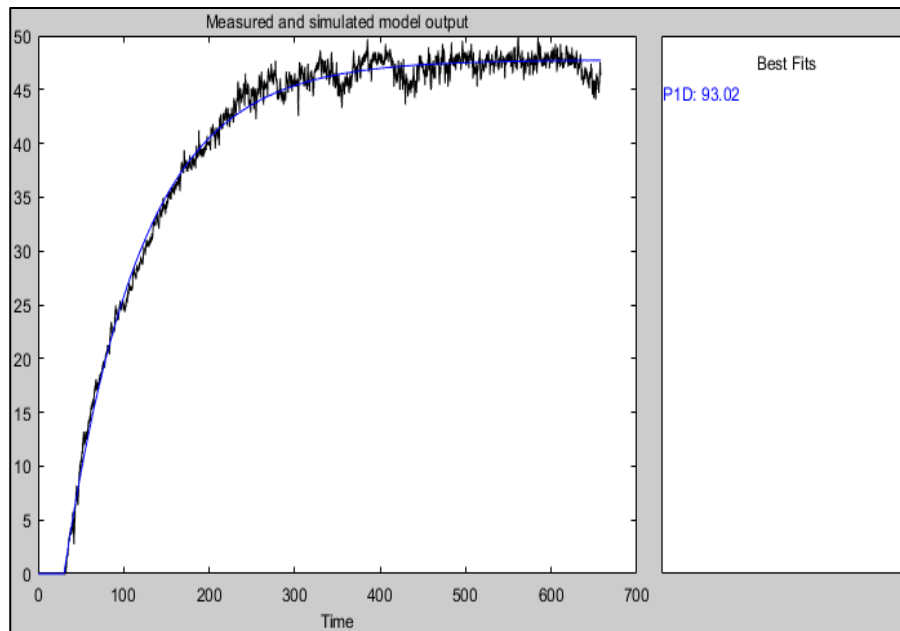


Figura 54

Porcentaje de aproximación del proceso de caudal (Lazo secundario)



Nota. La aproximación presentada corresponde al proceso de nivel, funcionando el controlador secundario en modo Fuzzy.

Modelo matemático de la variable Caudal

Para la obtención del modelo matemático a implementarse, se obtuvo mediante la experimentación en la estación de control de Caudal ND-0704 del laboratorio de redes industriales y control de procesos. La función de transferencia obtenida se presenta en la Ecuación 1.

$$G(s) = \frac{1.1477}{1 + 1.0102s} e^{-0.9666s}$$

Ecuación 1

Modelo matemático de la variable caudal.

Modelo matemático de la variable nivel

El modelo matemático se obtuvo de la estación de Nivel ND-0704 del laboratorio de redes industriales y control de procesos, mismo que se obtuvo enviando un CV y esperando a que la variable reaccione y se estabilice. La función de transferencia obtenida se presenta en la Ecuación 2.

$$G(s) = \frac{2.3344}{1 + 117.24s} e^{-1.0235s}$$

Ecuación 2

Modelo matemático de la variable nivel

Modelo matemático de la variable nivel en cascada.

Para obtener los datos y posteriormente la ecuación del modelo matemático de la variable nivel, el lazo interno se debe encontrar funcionando correctamente y sintonizado, como se representa en la Ecuación 3.

$$G(s) = \frac{1.4892}{1 + 78.058s} e^{-36.958s} \quad G(s) = \frac{0.5332}{1 + 111.27s} e^{-1.0235s}$$

MODELO EN FUZZY

MODELO EN PID

Ecuación 3

Modelo matemático de la variable nivel en método cascada

Diseño de los controladores clásicos basado en métodos de sintonía

El diseño de controladores clásicos mediante métodos de sintonía es una práctica común en el campo de la ingeniería de control. Estos métodos se utilizan para ajustar los parámetros de los controladores, ya sea de forma manual o mediante algoritmos específicos, con el objetivo de obtener un rendimiento óptimo del sistema de control. La sintonización de estos controladores busca encontrar un equilibrio entre la estabilidad del sistema y su

capacidad de respuesta frente a cambios y perturbaciones. Estas técnicas han demostrado su eficacia en diversos sistemas y procesos industriales, brindando soluciones efectivas y robustas para el control y la regulación de sistemas dinámicos.

Método de sintonía Cohen-Coon

Caudal

Las constantes proporcionales por el modelo matemático de caudal son las siguientes:

$$Kp = 1,1477$$

$$Tp = 1,0102$$

$$Td = 0,9666$$

$$a = \frac{Kp * L}{T}$$

$$a = \frac{1,1477 * 0,9666}{1,0102} = 1,0981$$

$$\tau = \frac{L}{(L + T)}$$

$$\tau = \frac{0,9666}{(0,9666 + 1,0102)} = 0,4889$$

Ecuación 4

Constantes para cálculos del método Cohen-Coon (Caudal)

Controlador PI

$$ak = 0.9 \left(1 + \frac{0,092\tau}{1 - \tau} \right)$$

$$ak = 0.9 \left(1 + \frac{0,092 * 0,4889}{1 - 0,4889} \right) = 0,9792$$

$$k = \frac{0,9792}{1,0981} = \mathbf{0,8917}$$

$$\frac{Ti}{L} = \frac{3,3 - 3,0\tau}{1 + 1,2\tau}$$

$$\frac{Ti}{L} = \frac{3,3 - 3,0 * 0,4889}{1 + 1,2 * 0,4889} = 1,1554$$

$$Ti = 1,1554 * 0,9666$$

$$Ti = \mathbf{1,1168s}$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,8917}{1,1168} = \mathbf{0,7984}$$

Ecuación 5

Fórmulas del controlador Cohen-Coon (PI)-Caudal

Controlador PID

$$ak = 1,35 \left(1 + \frac{0,18\tau}{1 - \tau} \right)$$

$$ak = 1,35 \left(1 + \frac{0,18 * 0,4889}{1 - 0,4889} \right) = 1,5824$$

$$k = \frac{1,5824}{1,0981} = \mathbf{1,4410}$$

$$\frac{Ti}{L} = \frac{2,5 - 2\tau}{1 - 0,39\tau}$$

$$\frac{Ti}{L} = \frac{2,5 - 2 * 0,4889}{1 - 0,39 * 0,4889} = 1,8808$$

$$Ti = 1,8808 * 0,9666 = \mathbf{1,8179s}$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{1,4410}{1,8179} = \mathbf{0,7926}$$

$$\frac{Td}{L} = \frac{0,37 - 0,37\tau}{1 - 0,81\tau}$$

$$\frac{Td}{L} = \frac{0,37 - 0,37(0,4889)}{1 - 0,81(0,4889)} = 0,3130$$

$$Td = 0,3130 * 0,9666 = \mathbf{0,3025s}$$

$$Kd = K * Td$$

$$Kd = 1,4410 * 0,3025$$

$$Kd = \mathbf{0,4359}$$

Ecuación 6

Fórmulas del controlador Cohen-Coon (PID)-Caudal

Nivel

Las constantes proporcionales por el modelo matemático de nivel son las siguientes:

$$Kp = 2,3344$$

$$Tp = 117,24$$

$$Td = 1,0235$$

$$a = \frac{Kp * L}{T}$$

$$a = \frac{2,3344 * 1,0235}{117,24} = 0,0203$$

$$\tau = \frac{L}{(L + T)}$$

$$\tau = \frac{1,0235}{(1,0235 + 117,24)} = 0,0086$$

Ecuación 7

Fórmulas de las constantes del controlador Cohen-Coon (Nivel)

Controlador PI

$$ak = 0.9 \left(1 + \frac{0,092\tau}{1 - \tau} \right)$$

$$ak = 0.9 \left(1 + \frac{0,092 * 0,0086}{1 - 0,0086} \right) = 0,9007$$

$$k = \frac{0,9007}{0,0203} = 44,3694$$

$$\frac{Ti}{L} = \frac{3,3 - 3,0\tau}{1 + 1,2\tau}$$

$$\frac{Ti}{L} = \frac{3,3 - 3,0 * 0,0086}{1 + 1,2 * 0,0086} = 3,2407$$

$$Ti = 3,2407 * 1,0235$$

$$Ti = 3,3168s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{44,3694}{3,3168} = 13,3771$$

Ecuación 8

Fórmulas del controlador Cohen-Coon (PI)-Nivel

Controlador PID

$$ak = 1,35 \left(1 + \frac{0,18\tau}{1 - \tau} \right)$$

$$ak = 1,35 \left(1 + \frac{0,18 * 0,0086}{1 - 0,0086} \right) = 1,3521$$

$$k = \frac{1,3521}{0,0203} = 66,6059$$

$$\frac{Ti}{L} = \frac{2,5 - 2\tau}{1 - 0,39\tau}$$

$$\frac{Ti}{L} = \frac{2,5 - 2 * 0,0086}{1 - 0,39 * 0,0086} = 2,4911$$

$$Ti = 2,4911 * 1,0235 = 2,5496s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{66,6059}{2,5496} = 26,1240$$

$$\frac{Td}{L} = \frac{0,37 - 0,37\tau}{1 - 0,81\tau}$$

$$\frac{Td}{L} = \frac{0,37 - 0,37(0,0086)}{1 - 0,81(0,0086)} = 0,3693$$

$$Td = 0,3693 * 1,0235 = \mathbf{0,3779s}$$

$$Kd = K * Td$$

$$Kd = 66,6059 * 0,3779$$

$$Kd = \mathbf{25,1703}$$

Ecuación 9

Fórmulas del controlador Cohen-Coon (PID)-Nivel

Método de sintonía de Haalman

Caudal

Las constantes proporcionales por el modelo matemático de caudal son las siguientes:

$$Kp = 1,1477$$

$$Tp = 1,0102$$

$$Td = 0,9666$$

Controlador PI

$$k = \frac{2T}{3kpL}$$

$$k = \frac{2 * 1,0102}{3 * 1,1477 * 0,9666} = 0,60$$

$$Ti = T$$

$$Ti = 1,0102s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,60}{1,0102} = 0,5939$$

Ecuación 10

Fórmulas del controlador Haalman (PI)-Caudal

Nivel

Las constantes proporcionales por el modelo matemático de nivel son las siguientes:

$$Kp = 2,3344$$

$$Tp = 117,24$$

$$Td = 1,0235$$

Controlador PI

$$k = \frac{2T}{3kpL}$$

$$k = \frac{2 * 117,24}{3 * 2,3344 * 1,0235} = 32,7130$$

$$T_i = T$$

$$T_i = 117,24s$$

$$K_i = \frac{K}{T_i}$$

$$K_i = \frac{32,7130}{117,24} = 0,2790$$

Ecuación 11

Fórmulas del controlador Haalman (PI)-Nivel

Método de sintonía Lambda Agresivo

Caudal

Las constantes proporcionales por el modelo matemático de caudal son las siguientes:

$$K_p = 1,1477$$

$$T = 1,0102$$

$$L = 0,9666$$

$$T_{cl} = T = 1,0102$$

Controlador PI

$$K = \frac{1}{K_p} \frac{T}{L + T_{cl}}$$

$$K = \frac{1}{1,1477} \frac{1,0102}{0,9666 + 1,0102} = 0,4452$$

$$T_i = T$$

$$Ti = 1,0102s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,4452}{1,0102} = 0,4407$$

Ecuación 12

Fórmulas del controlador Lambda Agresivo (PI)-Caudal

Controlador PID

$$K = \frac{1}{Kp} \frac{\frac{L}{2} + T}{\frac{L}{2} + Tcl}$$

$$K = \frac{1}{1,1477} \frac{\frac{0,9666}{2} + 1,0102}{\frac{0,9666}{2} + 1,0102} = 0,8713$$

$$Ti = T + \frac{L}{2}$$

$$Ti = 1,0102 + \frac{0,9666}{2}$$

$$Ti = 1,4935s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,8713}{1,4935} = 0,5833$$

$$Td = \frac{TL}{L + 2T}$$

$$Td = \frac{1,0102 * 0,9666}{0,9666 + 2 * 1,0102} = 0,3269s$$

$$Kd = K * Td$$

$$Kd = 0,8713 * 0,3269$$

$$Kd = 0,2848$$

Ecuación 13

Fórmulas del controlador Lambda Agresivo (PID)-Caudal

Nivel

Las constantes proporcionales por el modelo matemático de caudal son las siguientes:

$$Kp = 2,3344$$

$$T = 117,24$$

$$L = 1,0235$$

$$Tcl = T = 117,24$$

Controlador PI

$$K = \frac{1}{K_p} \frac{T}{L + TcI}$$

$$K = \frac{1}{2,3344} \frac{117,24}{1,0235 + 117,24} = 0,4246$$

$$Ti = T$$

$$Ti = 117,24s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,4246}{117,24} = 0,0036$$

Ecuación 14

Fórmulas del controlador Lambda Agresivo (PI)-Nivel

Controlador PID

$$K = \frac{1}{K_p} \frac{\frac{L}{2} + T}{\frac{L}{2} + TcI}$$

$$K = \frac{1}{2,3344} \frac{\frac{1,0235}{2} + 117,24}{\frac{1,0235}{2} + 117,24} = 0,4283$$

$$Ti = T + \frac{L}{2}$$

$$Ti = 117,24 + \frac{1,0235}{2}$$

$$Ti = 117,75s$$

$$K_i = \frac{K}{T_i}$$

$$K_i = \frac{0,4283}{117,75} = 0,0036$$

$$T_d = \frac{TL}{L + 2T}$$

$$T_d = \frac{117,24 * 1,0235}{1,0235 + 2 * 117,24} = 0,5095s$$

$$K_d = K * T_d$$

$$K_d = 0,4283 * 0,5095$$

$$K_d = 0,2182$$

Ecuación 15

Fórmulas del controlador Lambda Agresivo (PID)-Nivel

Método de sintonía Lambda Robusto

Caudal

Las constantes proporcionales por el modelo matemático de caudal son las siguientes:

$$K_p = 1,1477$$

$$T = 1,0102$$

$$L = 0,9666$$

$$T_{cl} = 3T = 3(1,0102) = 3,0306$$

Controlador PI

$$K = \frac{1}{K_p} \frac{T}{L + Tc}$$

$$K = \frac{1}{1,1477} \frac{1,0102}{0,9666 + 3,0306} = 0,2202$$

$$Ti = T$$

$$Ti = 1,0102s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,2202}{1,0102} = 0,2179$$

Ecuación 16

Fórmulas del controlador Lambda Robusto (PI)-Caudal

Controlador PID

$$K = \frac{1}{K_p} \frac{\frac{L}{2} + T}{\frac{L}{2} + Tc}$$

$$K = \frac{1}{1,1477} \frac{\frac{0,9666}{2} + 1,0102}{\frac{0,9666}{2} + 3,0306} = 0,3703$$

$$Ti = T + \frac{L}{2}$$

$$Ti = 1,0102 + \frac{0,9666}{2}$$

$$Ti = 1,4935s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,3703}{1,4935} = 0,2479$$

$$Td = \frac{TL}{L + 2T}$$

$$Td = \frac{1,0102 * 0,9666}{0,9666 + 2 * 1,0102} = 0,3269s$$

$$Kd = K * Td$$

$$Kd = 0,3703 * 0,3269$$

$$Kd = 0,1210$$

Ecuación 17

Fórmulas del controlador Lambda Robusto (PID)-Caudal

Nivel

Las constantes proporcionales por el modelo matemático de nivel son las siguientes:

$$Kp = 2,3344$$

$$T = 117,24$$

$$L = 1,0235$$

$$Tcl = 3T = 3(117,24) = 351,72$$

Controlador PI

$$K = \frac{1}{Kp} \frac{T}{L + Tcl}$$

$$K = \frac{1}{2,3344} \frac{117,24}{1,0235 + 351,72} = 0,1423$$

$$Ti = T$$

$$Ti = 117,24s$$

$$Ki = \frac{K}{Ti}$$

$$Ki = \frac{0,1423}{117,24} = 0,0012$$

Ecuación 18

Fórmulas del controlador Lambda Robusto (PI)-Nivel

Controlador PID

$$K = \frac{1}{Kp} \frac{\frac{L}{2} + T}{\frac{L}{2} + Tcl}$$

$$K = \frac{1}{2,3344} \frac{\frac{1,0235}{2} + 117,24}{\frac{1,0235}{2} + 351,72} = 0,1432$$

$$Ti = T + \frac{L}{2}$$

$$T_i = 117,24 + \frac{1,0235}{2}$$

$$T_i = 117.75s$$

$$K_i = \frac{K}{T_i}$$

$$K_i = \frac{0,1432}{117.75} = 0,0012$$

$$T_d = \frac{TL}{L + 2T}$$

$$T_d = \frac{117,24 * 1,0235}{1,0235 + 2 * 117,24} = 0,5095s$$

$$K_d = K * T_d$$

$$K_d = 0,1432 * 0,5095$$

$$K_d = 0,0729$$

Ecuación 19

Fórmulas del controlador Lambda Robusto (PID)-Nivel

Diseño del controlador avanzado Fuzzy

Para el diseño del controlador Fuzzy se basa en el método de Mamdani, donde se utiliza el software Matlab y el Toolkit Fuzzy para poder definir las reglas requeridas para el proceso en base a la experiencia, las etapas del controlador son: fusificación, defusificación, universo del discurso, funciones de pertenencia, variables lingüísticas de entrada y salida como se define en la Tabla 6.

Tabla 6 *Tabla FAM para definición de reglas de inferencia.*

e/de	deNG	deNM	deNP	deZ	dePP	dePM	dePG
eNG	UFNG	UFNG	UFNG	UFNG	UFNM	UFNP	UFZ
eNM	UFNG	UFNG	UFNG	UFNM	UFNP	UFZ	UFPP
eNP	UFNG	UFNG	UFNM	UFNP	UFZ	UFPP	UFPM
eZ	UFNG	UFNM	UFNP	UFZ	UFPP	UFPM	UFGP
ePP	UFNM	UFNP	UFZ	UFPP	UFPM	UFGP	UFGP
ePM	UFNP	UFZ	UFPP	UFPM	UFGP	UFGP	UFGP
ePP	UFZ	UFPP	UFPM	UFGP	UFGP	UFGP	UFGP

Nota. En la tabla se observa las variables lingüísticas.

Para el diseño del controlador se declara las variables de entrada y salida hacia el controlador, como entradas se declara al error, derivada del error y como salida se tiene la variable de control, definición de las 49 reglas Fuzzy, como se muestra en la Figura 55, 56 y 57 respectivamente.

Figura 55

Declaración de los rangos de entrada y salida

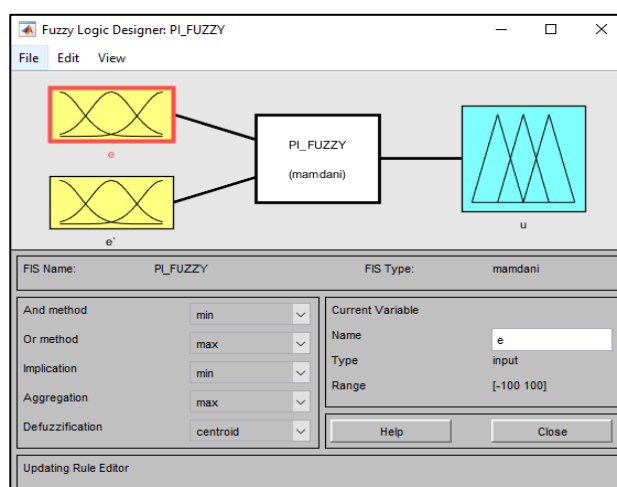


Figura 56

Triangulación de las entradas y las salidas.

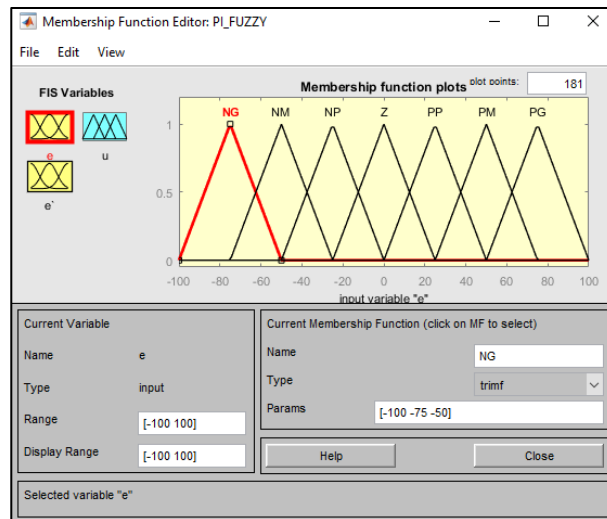
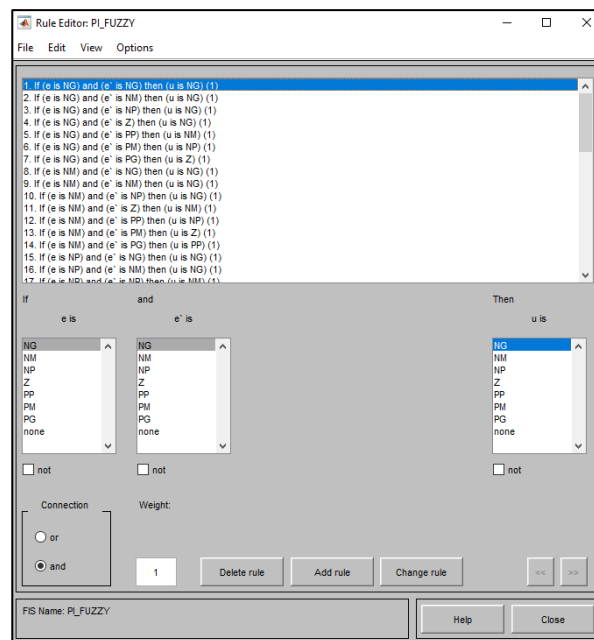


Figura 57

Definición de las 49 reglas Fuzzy



Implementación de los controladores en el PLC M-duino

Controlador simple de la variable caudal

En la Figura 58 se observa la implementación del control simple de la variable Caudal.

Figura 58

Implementación control PID variable Caudal en el software Arduino.

```

// I/O pins
const int PIN_INPUT2 = I1_4;          // Pin de entrada para el
controlador secundario(Caudal)
const int PIN_OUTPUT = A0_6;

// Control variables
double Setpoint1, Input1, Output1,Setpoint2;
double Input2, Output2,02;
double ScaledInput, ScaledOutput;

// PID controller parameters
double consKp2 = 0.2202, consKi2 = 0.2179, consKd2 = 0.0; //
Parámetros del controlador secundario (Caudal)
PID myPID2(&Input2, &Output2, &Setpoint2, consKp2, consKi2,
consKd2, DIRECT); // Controlador PID secundario

```

Controlador simple de la variable nivel

En la Figura 59 se observa la implementación del control simple de la variable nivel.

Figura 59

Implementación control PID variable Nivel en el software Arduino.

```

// I/O pins
const int PIN_INPUT1 = I1_5; // Pin de entrada para el
controlador primario (Nivel)
const int PIN_OUTPUT = A0_6;

// Control variables
double Setpoint1, Input1, Output1,Setpoint2;
double Input2, Output2,02;
double ScaledInput, ScaledOutput;

// PID controller parameters
double consKp1 = 0.8713, consKi1 = 0.583, consKd1 = 0.284; //
Parámetros del controlador primario nivel
PID myPID1(&Input1, &Output1, &Setpoint2, consKp1, consKi1,
consKd1, DIRECT);

```

Controlador de la técnica de control avanzado (Cascada)

Se implementa la técnica de control avanzado en Cascada en la cual se consideró el lazo primario a la variable Nivel por ser más lento y el lazo secundario a la variable Caudal por ser el más rápido respectivamente en este proceso, lo cual es una característica fundamental en este tipo de técnica de control avanzado.

En la Figura 60 se observa la implementación de la técnica de control avanzado (Caudal-Nivel).

Figura 60

Implementación de la técnica de control avanzado Cascada.

```
// Control variables
double Setpoint1, Input1, Output1,Setpoint2;
double Input2, Output2,O2;
double ScaledInput, ScaledOutput;

// PID controller parameters
double consKp1 = 0.65, consKi1 = 0.0036, consKd1 = 0.0;
double consKp2 = 0.2202, consKi2 = 0.2179, consKd2 = 0.0; //
Parámetros del controlador secundario
PID myPID1(&Input1, &Output1, &Setpoint2, consKp1, consKi1, consKd1,
DIRECT);
PID myPID2(&Input2, &Output2, &Output1, consKp2, consKi2, consKd2,
DIRECT); // Controlador PID secundario
```

Controlador avanzado Fuzzy

En la Figura 61 se observa la implementación del control avanzado Fuzzy de la variable Caudal.

Figura 61

Implementación control Fuzzy variable Caudal en el software Arduino.

```

//Calculo del control Fuzzy
lu = myfuzzy(e[kE], der, discurso_e, discurso_de, discurso_u);

//Integral
u[kU] = lu * 10 + u[kU - 1];

// Anti - Windup
if (u[kU] >= 100.0)
    u[kU] = 100.0;
if (u[kU] <= 0.0)
    u[kU] = 0.0;

H1 = u[kU];
//Aplica la acción de control en el PWM
analogWrite(PIN_OUTPUT, map(H1, 0, 100, 0, 255)); //Max= 100,
Min=0
}

```

Controlador técnica de control avanzado cascada (Fuzzy Caudal-PID Nivel)

En la Figura 62 se observa la implementación de la técnica de control avanzado Cascada (Fuzzy Caudal-PID Nivel).

Figura 62

Control Fuzzy variable Caudal y PID variable Nivel en el software Arduino.

```

// Control variables
double Setpoint1, Input1, Output1, Setpoint2;
double Input2, Output2, O2;
double ScaledInput, ScaledOutput;
double consKp1 = 0.55572, consKi1 = 0.010458, consKd1 = 0.006;

PID myPID1(&Input1, &Output1, &Setpoint2, consKp1, consKi1, consKd1,
DIRECT);

//Calculo del control Fuzzy
lu = myfuzzy(e[kE], der, discurso_e, discurso_de, discurso_u);

//Integral
u[kU] = lu * 10 + u[kU - 1];

// Anti - Windup
if (u[kU] >= 100.0)
    u[kU] = 100.0;
if (u[kU] <= 0.0)
    u[kU] = 0.0;

H1 = u[kU];
//Aplica la acción de control en el PWM
analogWrite(PIN_OUTPUT, map(H1, 0, 100, 0, 255)); //Max= 100,
Min=0
}

```

Creación del HMI del Sistema

Se realiza el HMI del sistema, de manera intuitiva y fácil de usar para permitir la interacción eficiente entre los operadores humanos y los sistemas automatizados, teniendo una disposición clara y lógica de la información, tamaño y legibilidad una navegación intuitiva, colores y gráficos adecuados.

Dashboard utilizados para la creación del HMI

En la Figura 63 se observa el tag principal utilizada en Node-RED es el SVG graphics node.

Figura 63

Tag principal SVG graphics node.

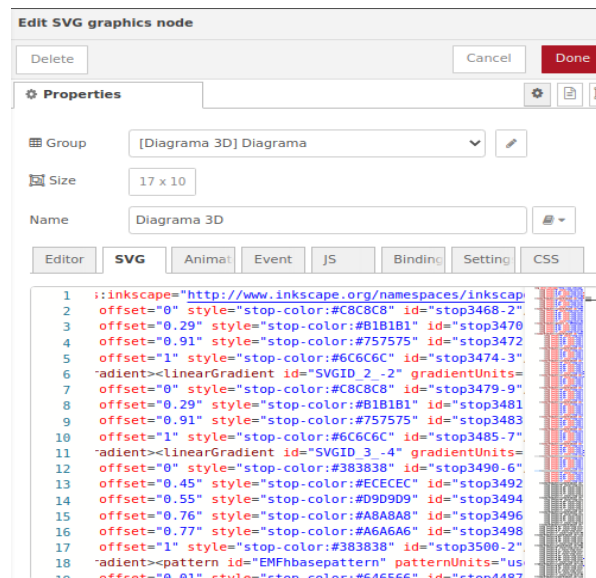


Diagrama 3D

En la Figura 64 se observa el diagrama en 3D.

Figura 64

Diagrama 3D.

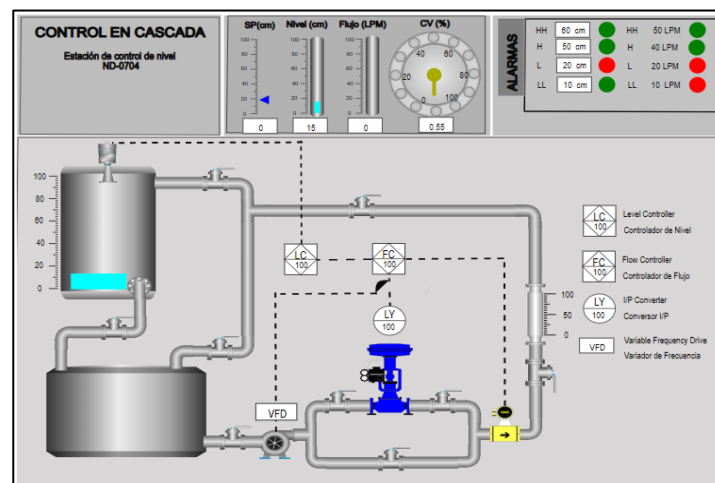
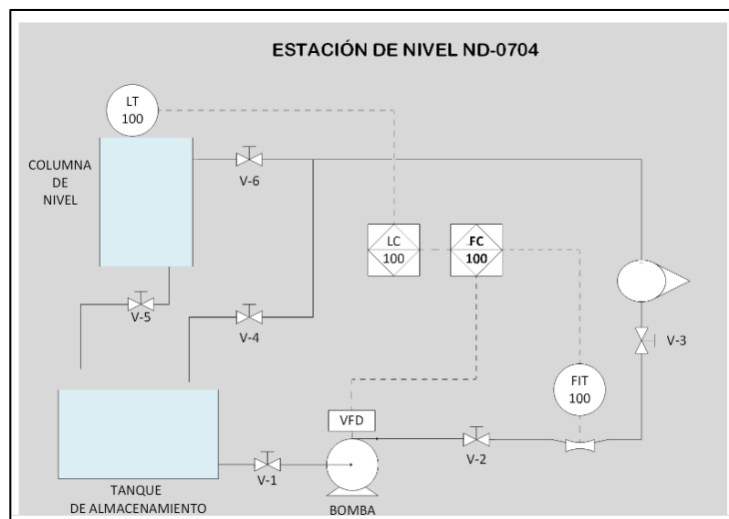


Diagrama 2D

En la Figura 65 se observa el diagrama en 2D.

Figura 65

Diagrama P&ID.

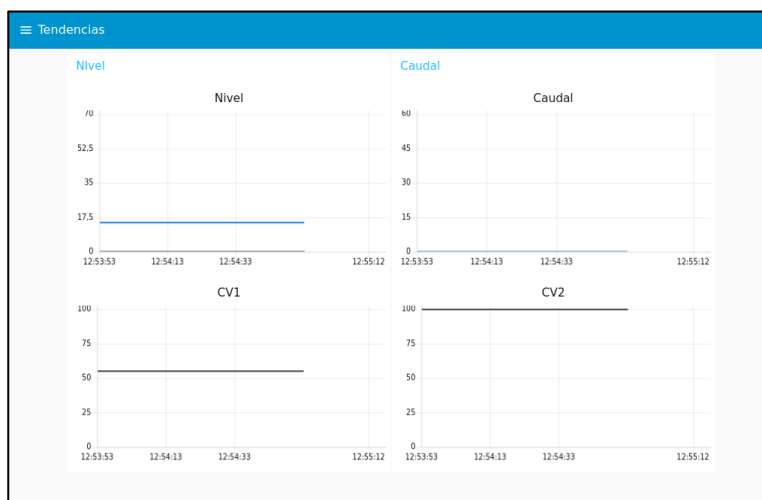


Tendencias

En la Figura 66 se observa las tendencias correspondientes a Nivel y Caudal, con sus respectivos CV1 Y CV2 respectivamente.

Figura 66

Tendencias Nivel – Caudal.



Vinculación de las variables entre el PLC y la ThinkerTouch

Para la vinculación entre el PLC y la Thinker Touch, esto se accede mediante Modbus.

Código de programación arduino

En la Figura 67 se observa los registros modbus creados que son utilizados en conjunto con las librerías Modbus.

Figura 67

Registros y librerías Modbus.

```
#include "Modbus.h"
#include "ModbusIP2.h"

// Modbus registers
const int Kc1 = 1;           // Ganancia proporcional del controlador
primario
const int Ki1 = 2;           // Ganancia integral del controlador
primario
const int Kd1 = 3;           // Ganancia derivativa del controlador
primario
const int Kc2 = 4;           // Ganancia proporcional del controlador
secundario
const int Ki2 = 5;           // Ganancia integral del controlador
secundario
const int Kd2 = 6;           // Ganancia derivativa del controlador
secundario
const int Sp1 = 7;           // Valor deseado (Setpoint) del
controlador primario
const int PV1 = 8;           // Variable de proceso (Process
Variable) del controlador primario
const int PV2 = 9;           // Variable de proceso (Process
Variable) del controlador secundario
const int CV1 = 10;          // Variable de control (Control
Variable) del controlador primario
const int CV2 = 11;          // Variable de control (Control
Variable) del controlador secundario
const int ControlType = 12;  // Tipo de control
const int UpdateTunings = 13; // Registro para actualizar las
variables de sintonización
const int LAMP1_COIL = 1;
```

Código de programación Node-RED

En Node-RED se utiliza una interfaz de programación visual basada en bloques y nodos, que representan diferentes funciones o acciones, además definen la lógica del flujo de datos y la ejecución de las distintas tareas asignadas como se observa en las Figuras 68,69,70,71,72,73,74 y 75 respectivamente.

Figura 68

Funciones para definir alarmas.

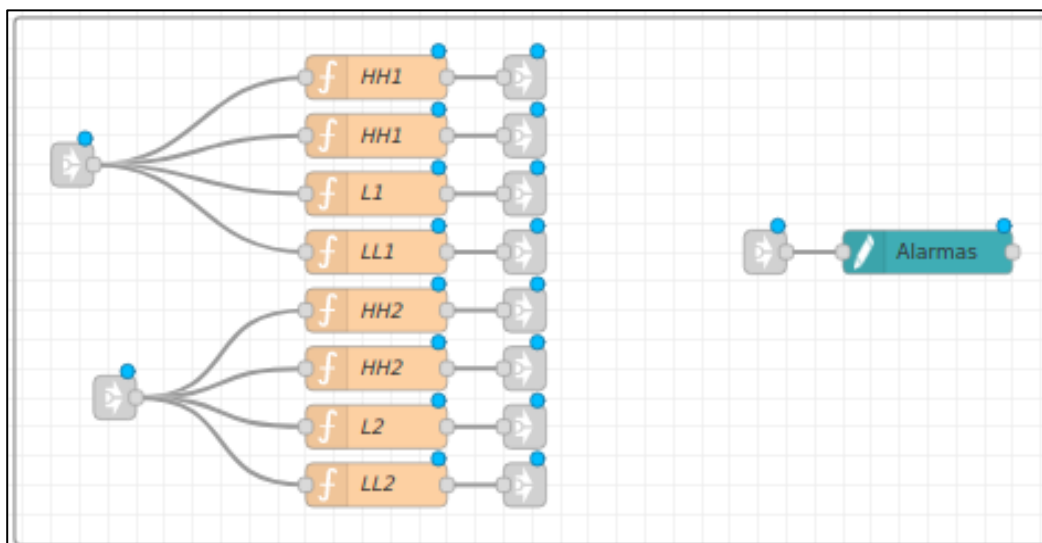


Figura 69

Funciones para la variable Nivel-Caudal utilizadas en el diagrama 3D.

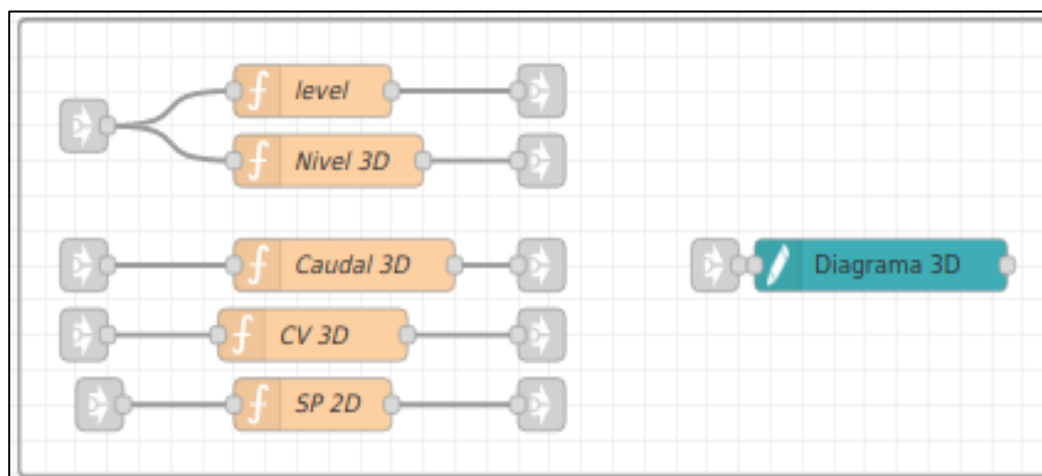


Figura 70

Funciones para la variable Nivel-Caudal utilizadas en el diagrama 2D.

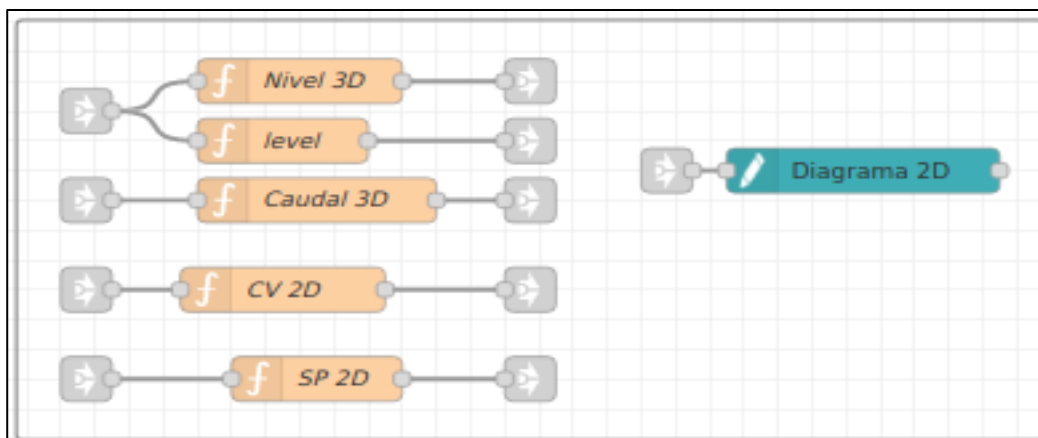


Figura 71

Setpoint Nivel, iiot/spniveLnube, Modbus Write, función dato-tendencia nivel.

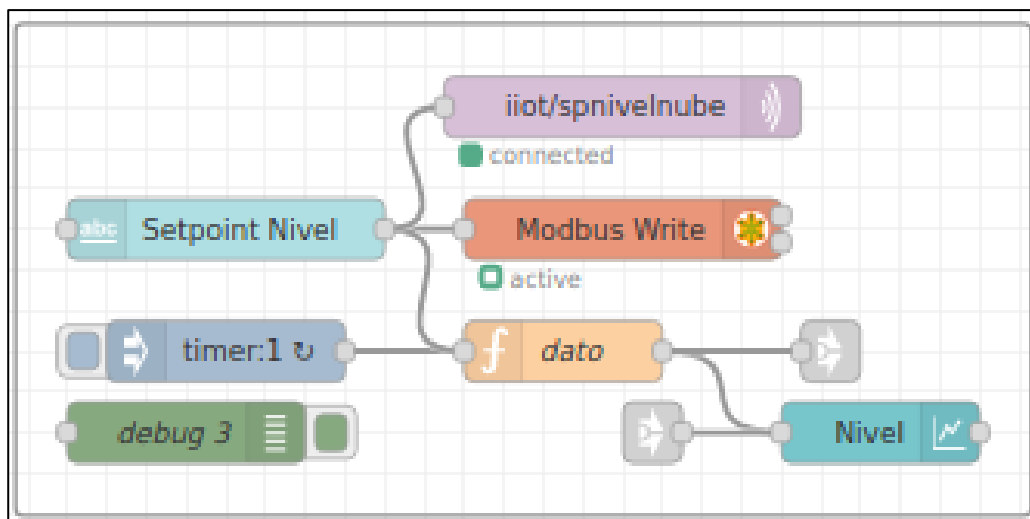


Figura 72

iiot/spcaudal, Setpoint Caudal, función dato-tendencia caudal, iiot/caudal.

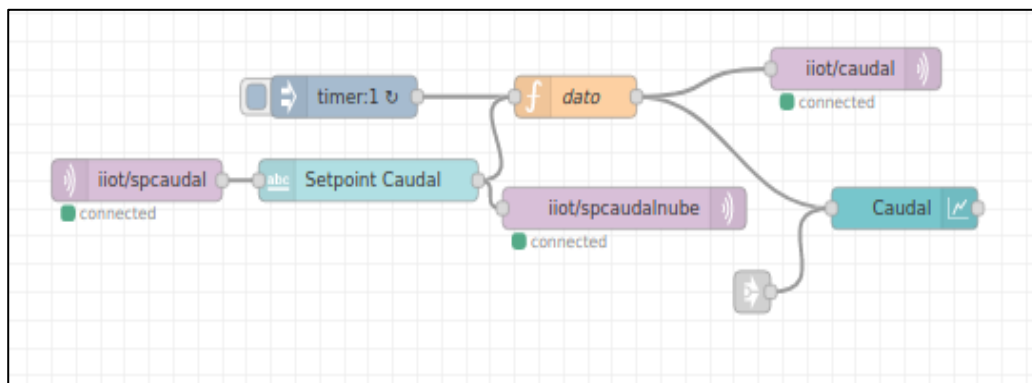


Figura 73

Modbus Read, function 2.

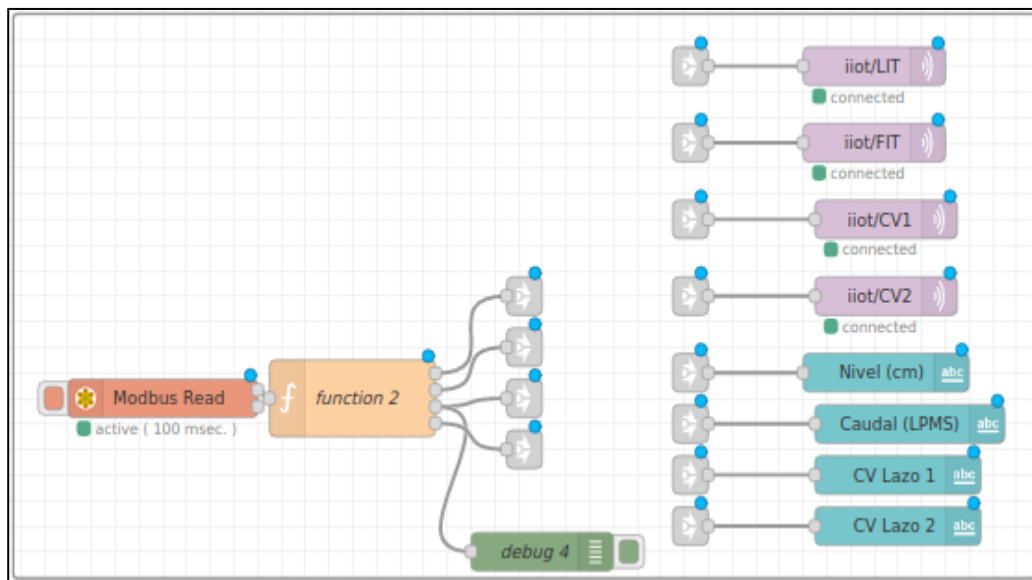
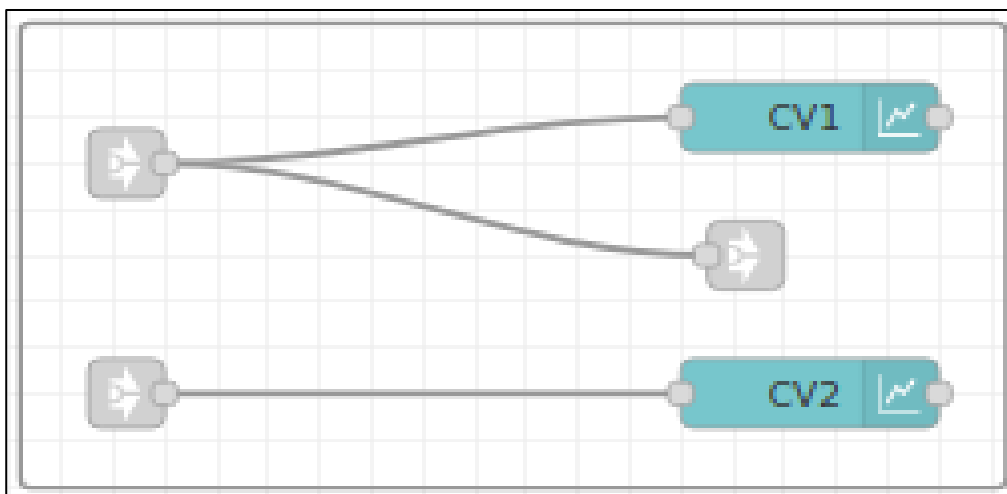
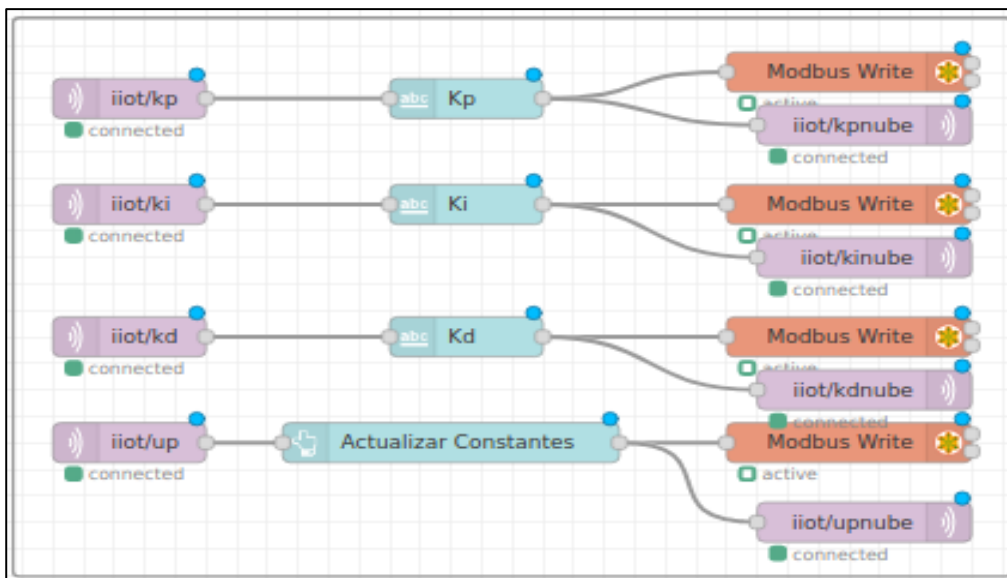


Figura 74

Tendencias CV1 y CV2.

**Figura 75**

Actualizar constantes k_p , k_i y k_d .

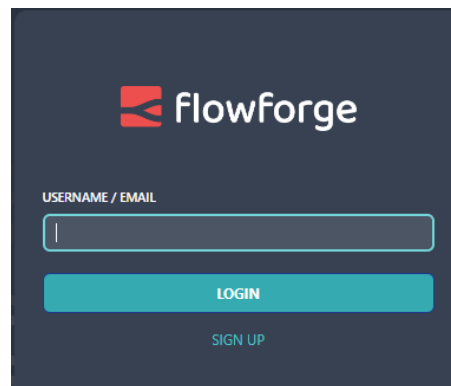


Creación de cuenta en flowforge para acceso global

Se necesita crear una cuenta en FlowForge para tener un acceso global y vinculación con la nube mediante MQTT, lo cual se realiza de la siguiente manera como se observa en la Figura 76 y 77 respectivamente.

Figura 76

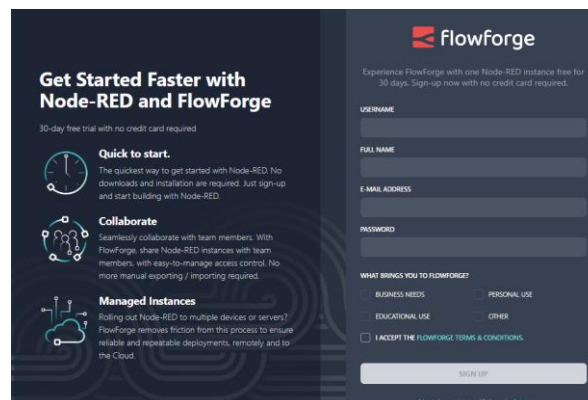
Creación de cuenta en flowforge.



Nota. En la figura se puede observar que se inicia a crear la cuenta en FlowForge. Tomado de FlowForge por (Nick, 2023).

Figura 77

Creación de cuenta en flowforge (ingreso de datos solicitados).



Nota. En la figura se ingresa todos los datos necesarios que se solicitan en la creación de la cuenta. Tomado de FlowForge por (Nick, 2023).

Capítulo IV

Pruebas y análisis de resultados

Pruebas experimentales

Para evaluar el desempeño que tenga un controlador se calcula el tiempo de establecimiento y el porcentaje de sobre impulso.

Para obtener el tiempo de establecimiento se debe tener en cuenta que una muestra se toma cada 100ms, este valor se multiplica por el número de muestras donde la señal se estabiliza.

Adicional para calcular el porcentaje de sobre impulso, se utiliza la siguiente ecuación.

$$\%OS = \left(\frac{OS - SP}{SP} \right) * 100\%$$

Ecuación 20

Fórmula cálculo del sobre impulso.

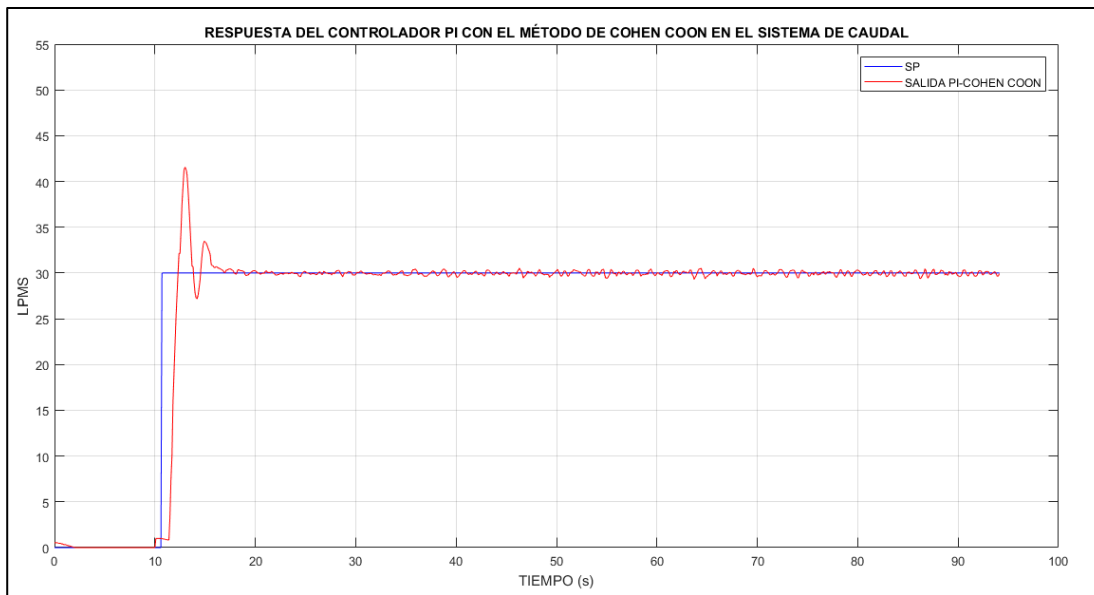
Funcionamiento del sistema de Caudal

Desempeño del controlador PID en el sistema de Caudal

Se observa el desempeño que tendrá el sistema de caudal al ser implementado el controlador PID con los métodos de sintonía: Cohen Coon, Haalman, Lambda agresivo y Lambda robusto; en este proceso se pone en funcionamiento la planta con un set point de 30 LPM, como se observa en las Figuras 78,79,80,81,82 y 83 respectivamente.

Figura 78

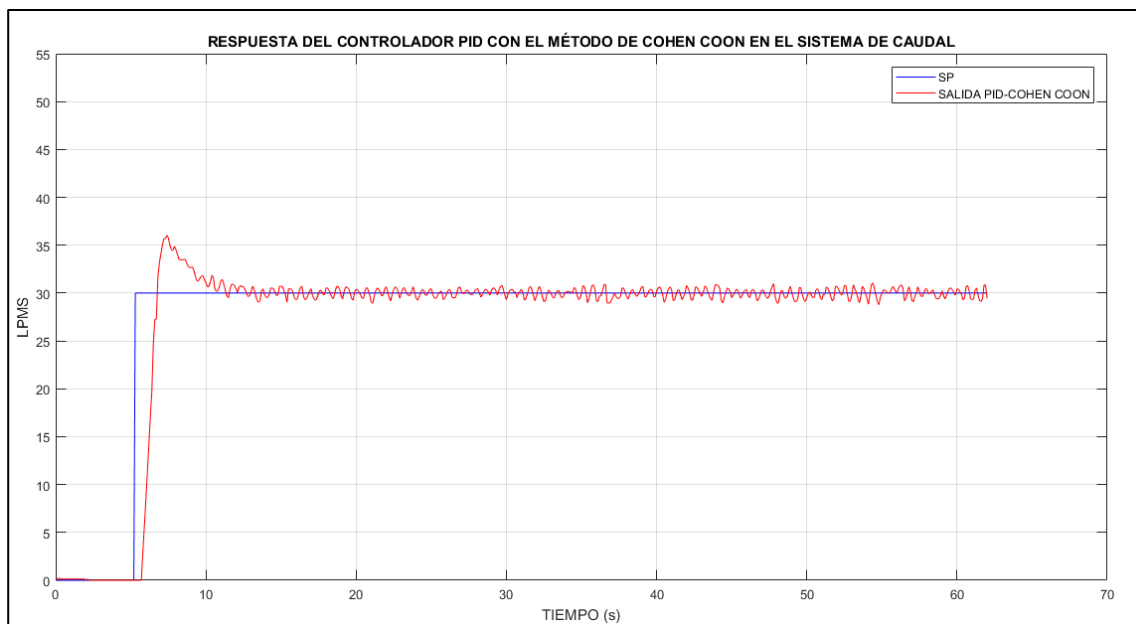
Respuesta del controlador PI, método de sintonía Cohen Coon (Caudal).



Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía de Cohen Coon, presenta un sobre impulso, su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 79

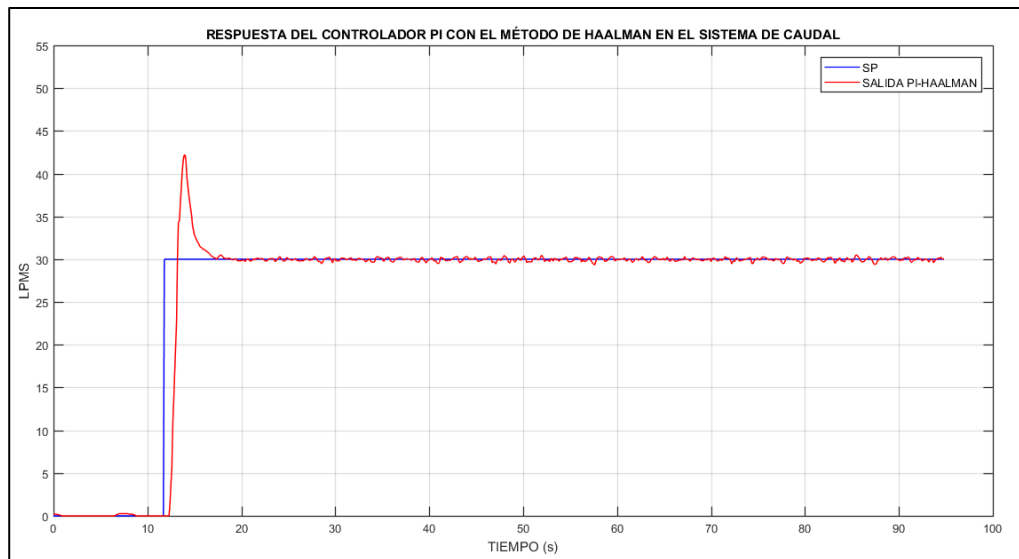
Respuesta del controlador PID, método de sintonía Cohen Coon (Caudal).



Nota. En la figura se observa la respuesta del controlador PID con el método de sintonía de Cohen Coon, presenta un sobre impulso, se tiene oscilaciones en estado estable, su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 80

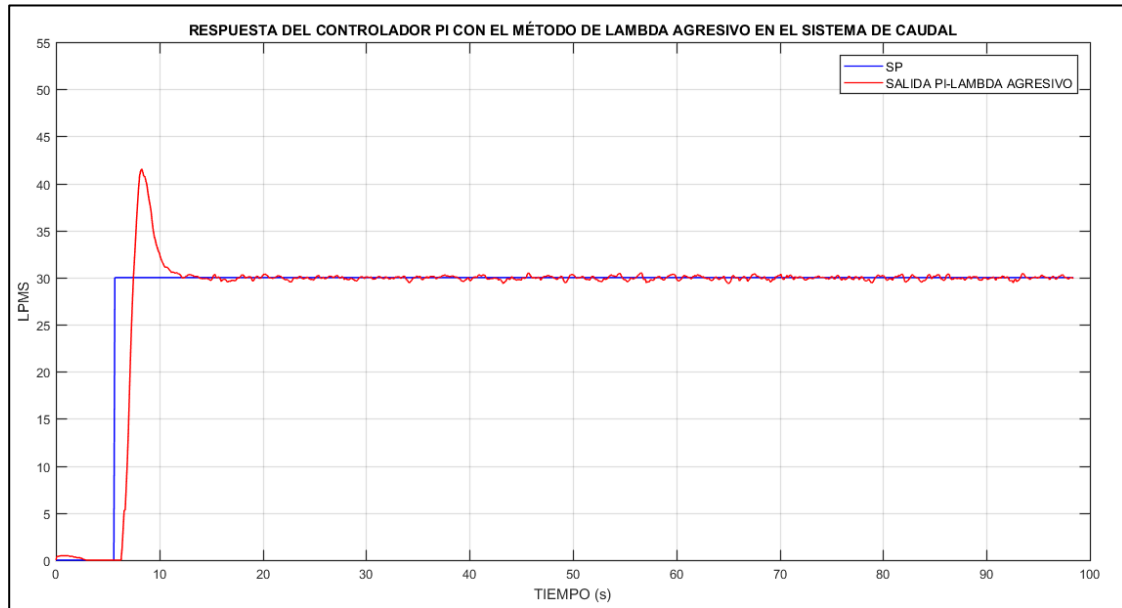
Respuesta del controlador PI, método de sintonía Haalman (Caudal).



Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía de Haalman, presenta un sobre impulso, su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 81

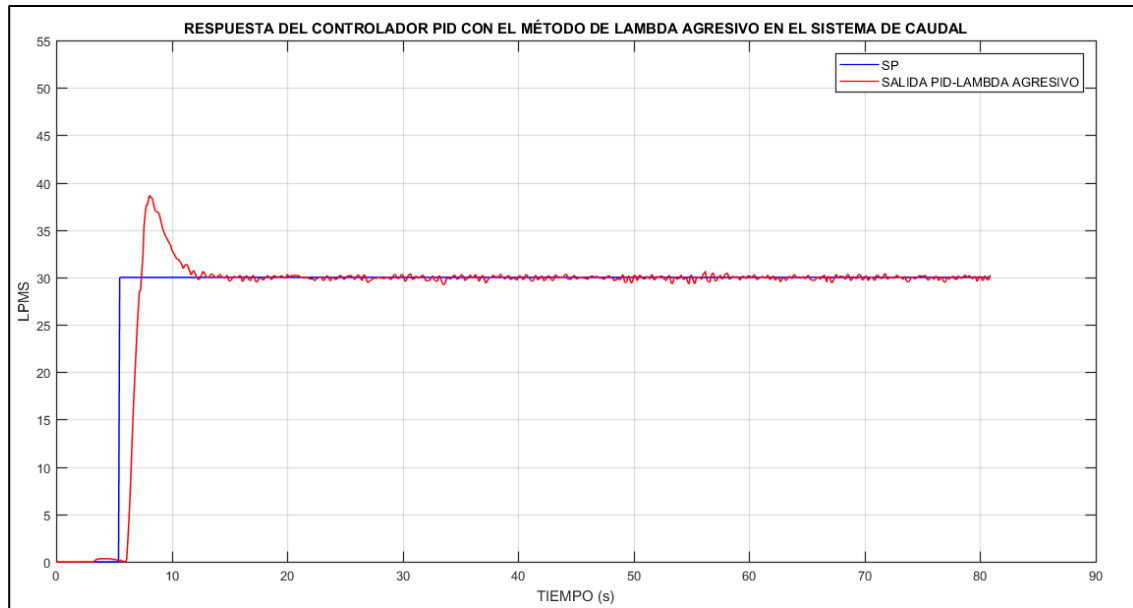
Respuesta del controlador PI, método de sintonía Lambda Agresivo (Caudal).



Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía de Lambda Agresivo, su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 82

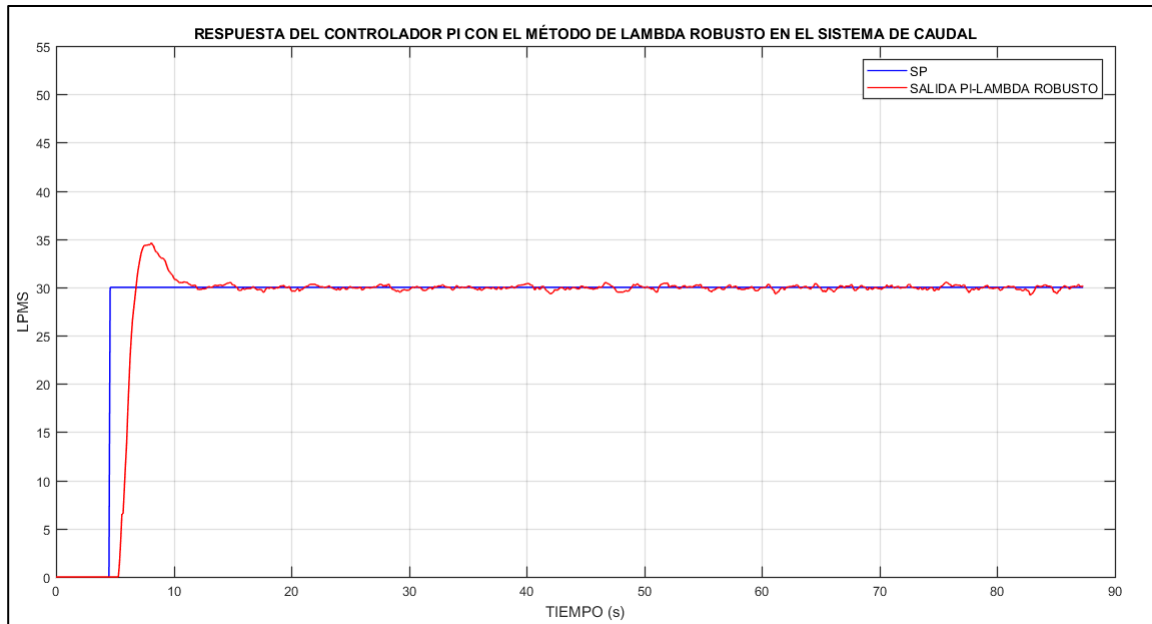
Respuesta del controlador PID, método de sintonía Lambda Agresivo (Caudal).



Nota. En la figura se observa la respuesta del controlador PID con el método de sintonía de Lambda Agresivo, presenta un sobre impulso su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 83

Respuesta del controlador PI, método de sintonía Lambda Robusto (Caudal).

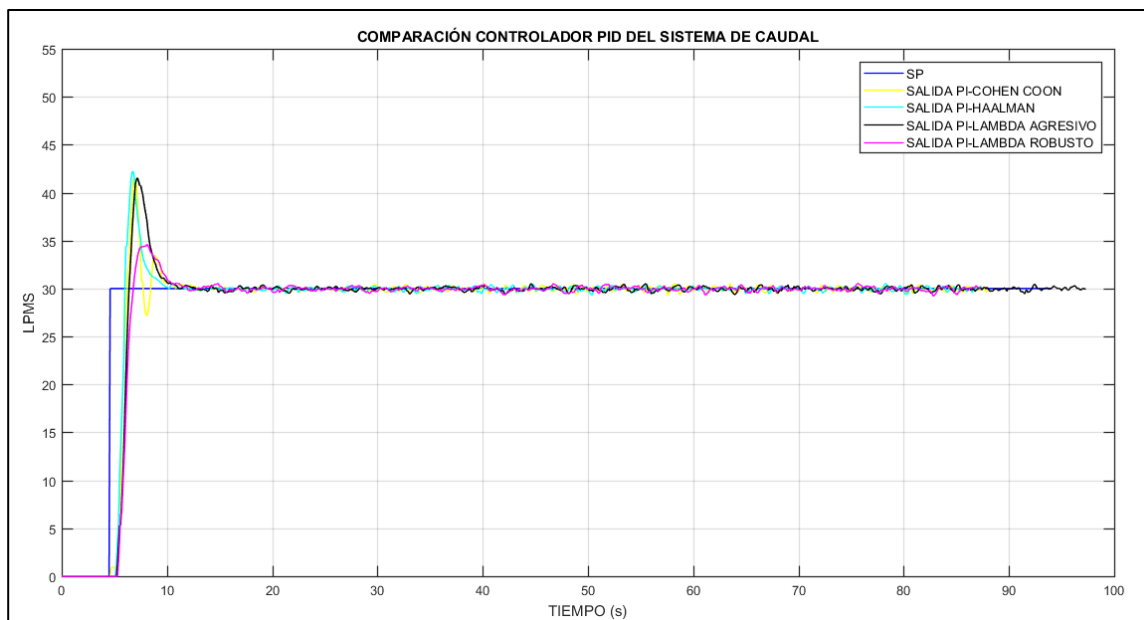


Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía de Lambda Robusto, su señal de control es estable por ende no presenta cambios bruscos, por lo que fue considerada ya que aumentaría considerablemente la vida útil de los elementos de control final.

Una vez que el controlador PID con sus cuatro métodos de sintonía ha sido implementado, se procede a realizar una comparación para determinar cuál de estos métodos brinda la respuesta óptima en el sistema, como se muestra en la Figura 84.

Figura 84

Comparación de los métodos de sintonía PID en el sistema de Caudal.



Nota. Con las gráficas obtenidas del desempeño del controlador PID con sus diferentes métodos de sintonía se calcula el porcentaje de sobre impulso y el tiempo de establecimiento, obteniendo los resultados de la Tabla 7.

Tabla 7

Parámetros controlador PID en el sistema de Caudal.

Método	%OS	Ts(s)
Cohen Coon	36.66	13
Haalman	43.33	9
Lambda Agresivo	40	11
Lambda Robusto	13.33	12

Nota. En la Tabla se observa que el método de sintonía Cohen Coon, Haalman y Lambda Agresivo presentan un gran porcentaje de sobre impulso; el método de sintonía Lambda Robusto presenta un menor porcentaje de sobre impulso y un tiempo de establecimiento aceptable, además su señal de control es estable, por ende, no presenta cambios bruscos y alarga la vida útil del elemento de control final.

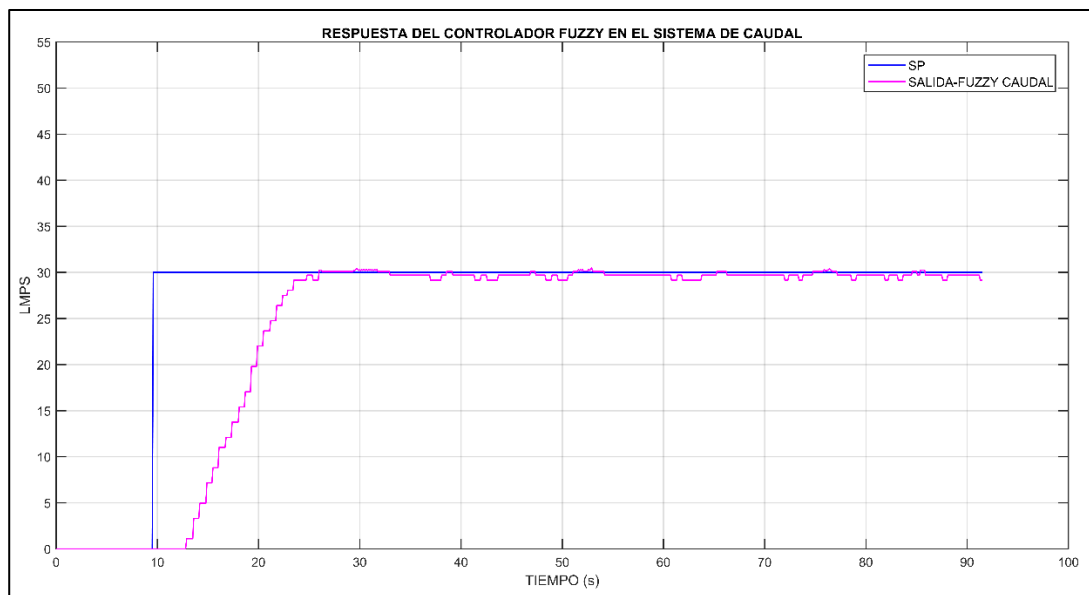
Del análisis realizado en el sistema de control de caudal, se determina que el mejor controlador es el proporcional integral con el método de sintonía Lambda Robusto, además se realizó sintonía fina a partir del mismo.

Desempeño del controlador Fuzzy en el sistema de Caudal

Se observa el desempeño que tendrá el sistema de caudal al ser implementado el controlador Fuzzy; en este proceso se pone en funcionamiento la planta con un set point de 30 LPM, como se observa en la Figura 85.

Figura 85

Respuesta del controlador Fuzzy del sistema de Caudal.



Nota. En la figura se observa la respuesta del controlador Fuzzy, la señal de control tiene mayor estabilidad por ende no presenta cambios bruscos, alargando así la vida útil de los elementos de control final, obteniendo los resultados de la Tabla 8.

Tabla 8

Parámetros del controlador Fuzzy en el sistema de Caudal.

Método	%OS	Ts(s)
Fuzzy	0.2	25

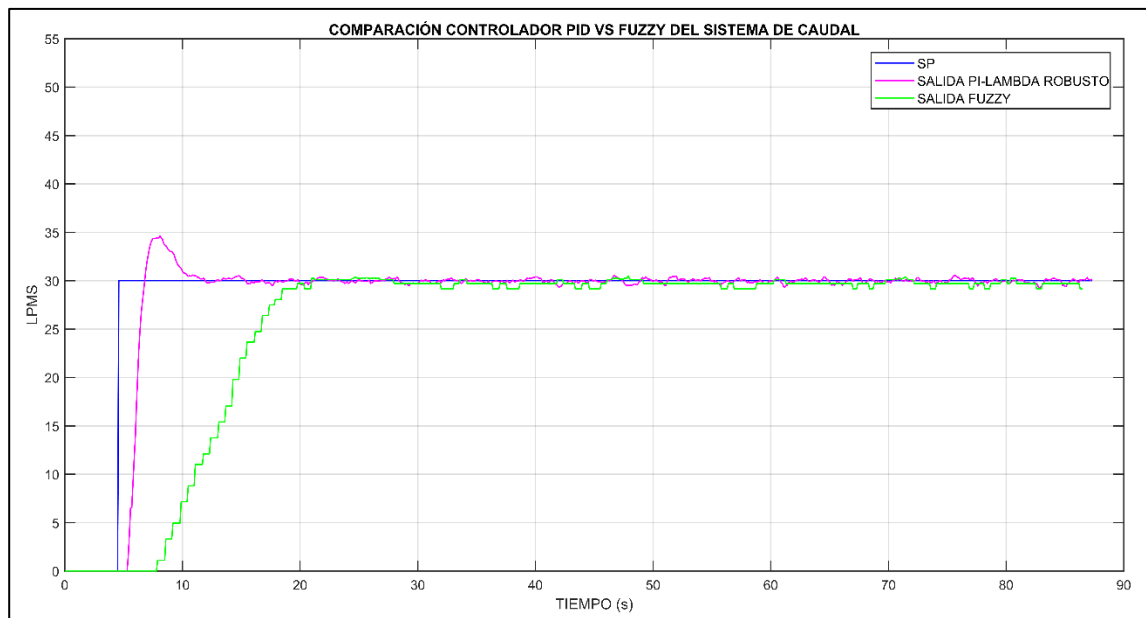
Nota. En la Tabla se observa que el controlador fuzzy presenta un pequeño porcentaje de sobre impulso y un tiempo de establecimiento aceptable, además su señal de control es estable, por ende, no presenta cambios bruscos y alarga la vida útil del elemento de control final.

Desempeño del mejor controlador PID vs Fuzzy (sistema de Caudal)

Se observa el desempeño que tendrá el sistema de caudal al ser implementado el controlador Fuzzy; en este proceso se pone en funcionamiento la planta con un set point de 30 LPM, como se observa en la Figura 86.

Figura 86

Respuesta del controlador PID vs Fuzzy del sistema de Caudal.



Nota. En la figura se observa la respuesta del controlador PI método de sintonía Lambda Robusto vs Fuzzy en el sistema de Caudal, obteniendo los resultados de la Tabla 9.

Tabla 9

Parámetros del controlador PID y Fuzzy en el sistema de Caudal.

Método	%OS	Ts(s)
Lambda Robusto PI	13,33	12
Fuzzy	0,2	25

Nota. En la Tabla se observa que el controlador PID el cual presenta un porcentaje de sobre impulso mientras que el controlador fuzzy presenta un pequeño porcentaje de sobre impulso y un tiempo de establecimiento aceptable, además su señal de control es estable en ambos casos, por ende, no presenta cambios bruscos y alarga la vida útil del elemento de control final.

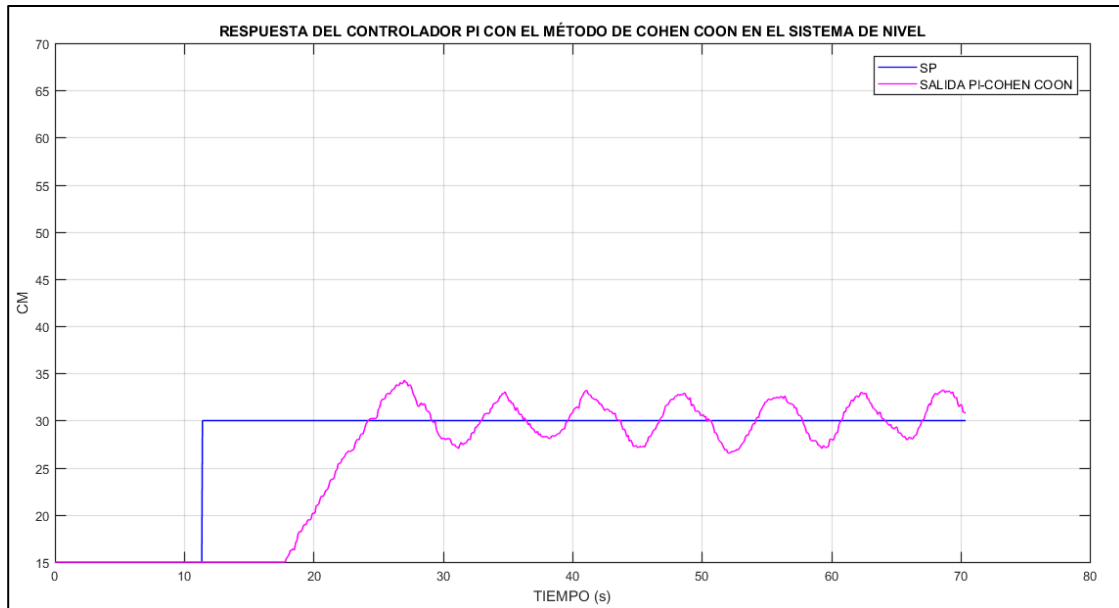
Funcionamiento del sistema de Nivel

Desempeño del controlador PID en el sistema de nivel

Se observa el desempeño que tendrá el sistema de nivel al ser implementado el controlador PID con los métodos de sintonía: Cohen Coon, Haalman, Lambda agresivo y Lambda robusto; en este proceso se pone en funcionamiento la planta con un set point de 30 Cm, como se observa en las Figuras 87,88,89,90,91 y 92 respectivamente.

Figura 87

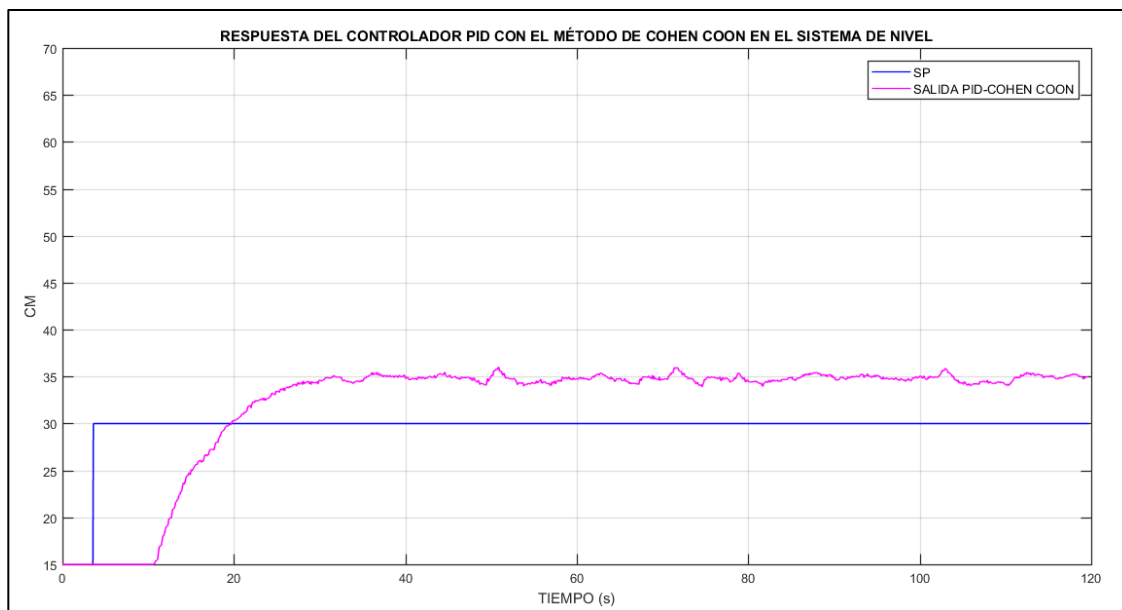
Respuesta del controlador PI, método de sintonía Cohen Coon (Nivel).



Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía de Cohen Coon, presenta oscilaciones y no se estabiliza en el punto de consigna, su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 88

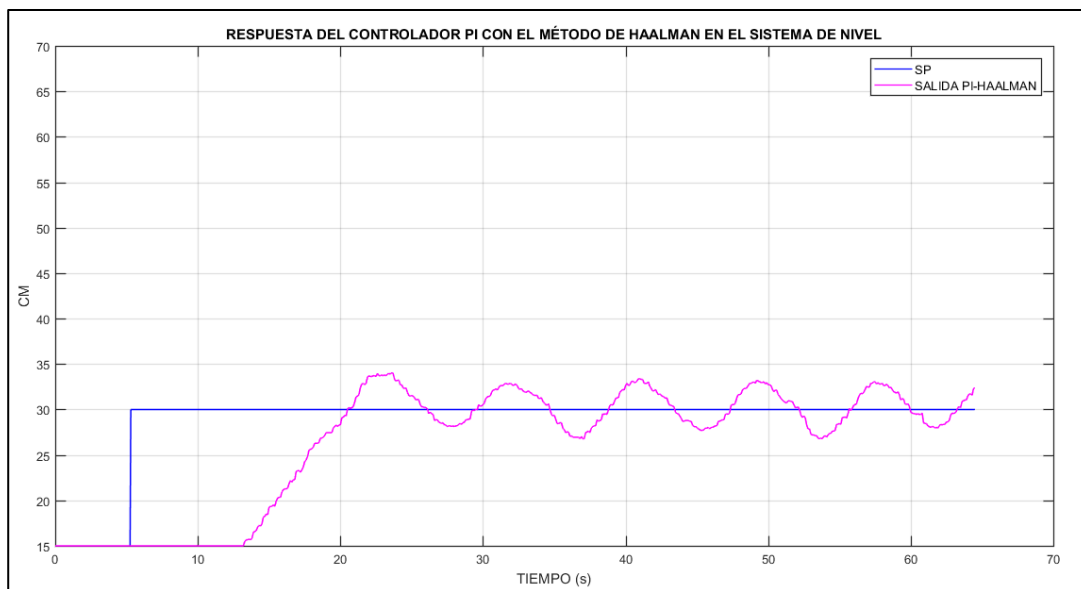
Respuesta del controlador PID, método de sintonía Cohen Coon (Nivel).



Nota. En la figura se observa la respuesta del controlador PID con el método de sintonía de Cohen Coon, y no llega a estabilizarse en el punto de consigna.

Figura 89

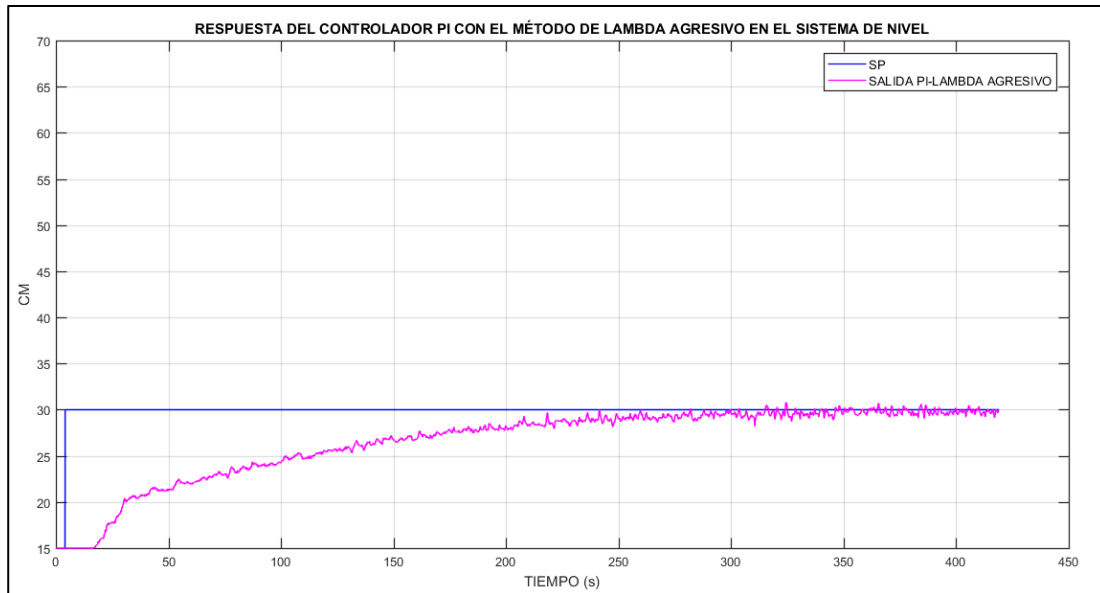
Respuesta del controlador PI, método de sintonía Haalman (Nivel).



Nota. En la figura se observa la respuesta del controlador PID con el método de sintonía de Haalman, y no llega a estabilizarse en el punto de consigna.

Figura 90

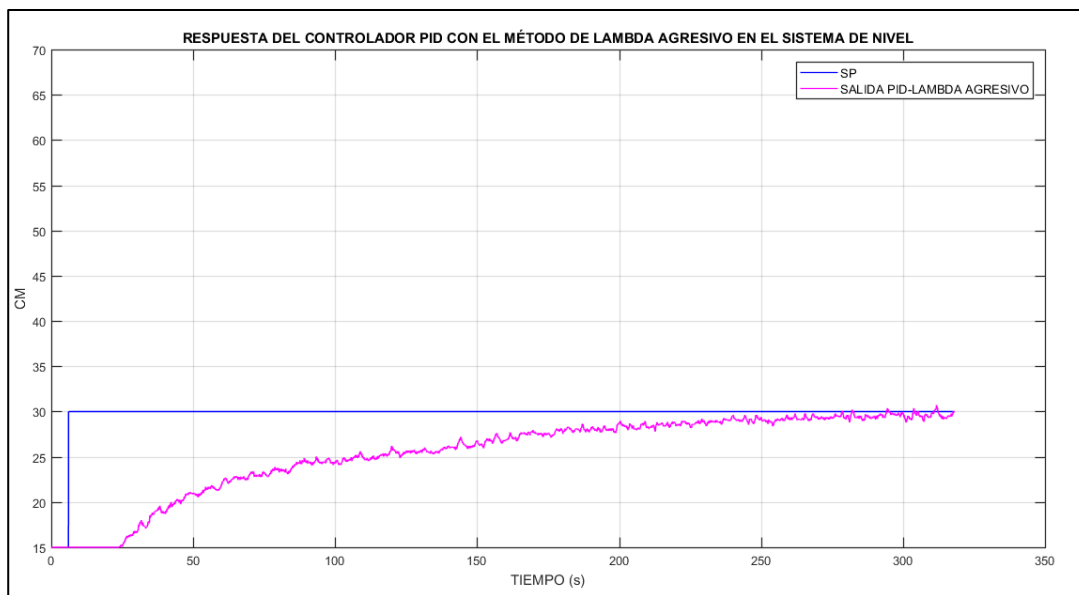
Respuesta del controlador PI, método de sintonía Lambda Agresivo (Nivel).



Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía de Lambda Agresivo, su señal de control es estable por ende no presenta cambios bruscos, por lo que fue considerada ya que aumentaría considerablemente la vida útil de los elementos de control final.

Figura 91

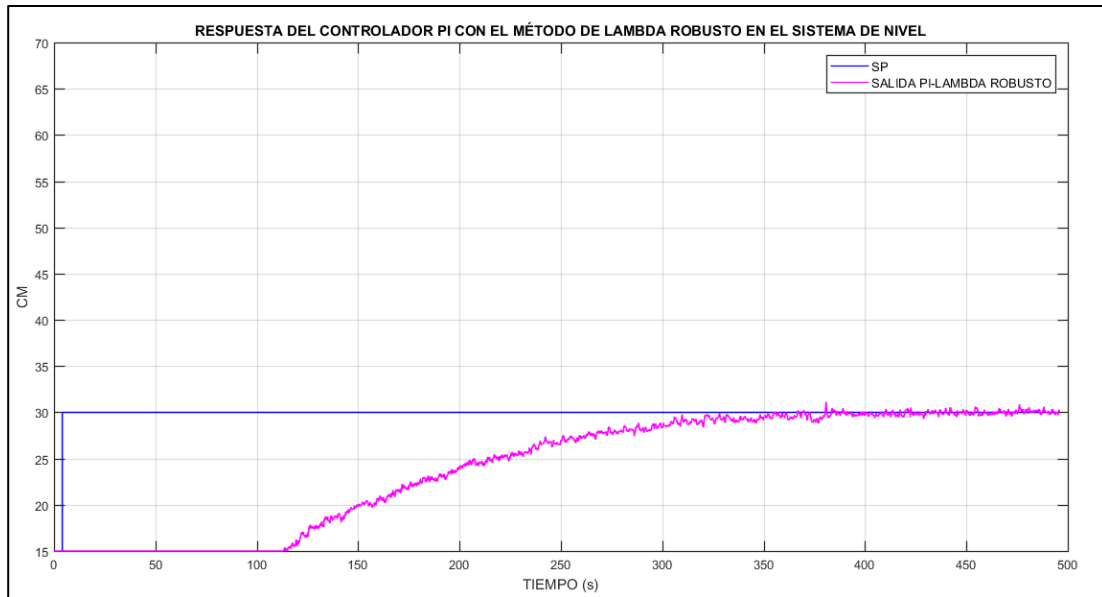
Respuesta del controlador PID, método de sintonía Lambda Agresivo (Nivel).



Nota. En la figura se observa la respuesta del controlador PID con el método de sintonía Lambda Agresivo, su señal de control presenta cambios bruscos, por lo que no fue considerada ya que disminuiría considerablemente la vida útil de los elementos de control final.

Figura 92

Respuesta del controlador PI, método de sintonía Lambda Robusto (Nivel).

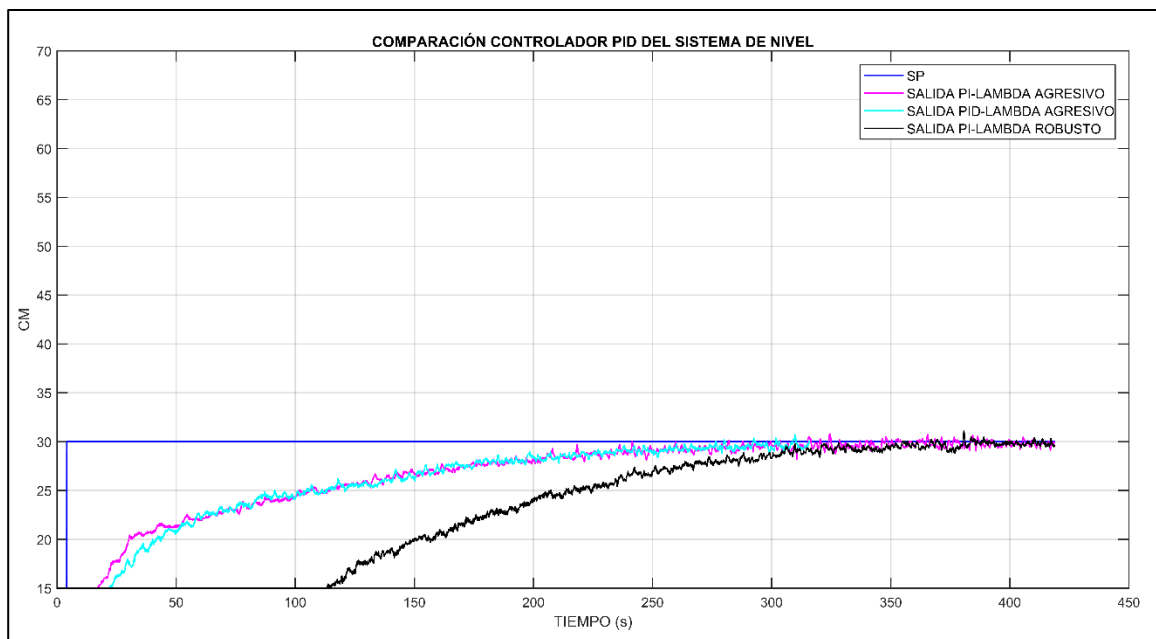


Nota. En la figura se observa la respuesta del controlador PI con el método de sintonía Lambda Robusto, su señal de control es estable por ende no presenta cambios bruscos, por lo que fue considerada ya que aumentaría considerablemente la vida útil de los elementos de control final.

Una vez que el controlador PID con sus cuatro métodos de sintonía ha sido implementado, se procede a realizar una comparación para determinar cuál de estos métodos brinda la respuesta óptima en el sistema, como se muestra en la Figura 93.

Figura 93

Comparación de los métodos de sintonía PID en el sistema de nivel.



Nota. Con las gráficas obtenidas del desempeño del controlador PID con sus diferentes métodos de sintonía se calcula el porcentaje de sobre impulso y el tiempo de establecimiento, obteniendo los resultados de la Tabla 10.

Tabla 10

Parámetros de los controladores PID en el sistema de nivel.

Método	%OS	Ts(s)
Lambda Agresivo PI	0	325
Lambda Agresivo PID	0	350
Lambda Robusto PI	0	375

Nota. En la Tabla se observa que el método de sintonía Lambda Agresivo PI, Lambda Agresivo PID y Lambda Robusto PI, no presentan sobre impulso; los otros métodos de sintonía no se los

consideró en la comparación porque no realizaron control. El método de sintonía Lambda Agresivo PI no presenta sobre impulso y tiene un menor tiempo de establecimiento, además su señal de control es estable, por ende, no presenta cambios bruscos y alarga la vida útil del elemento de control final.

Del análisis realizado en el sistema de control de nivel, se determina que el mejor controlador es el proporcional integral con el método de sintonía Lambda Agresivo, además se realizó sintonía fina a partir del mismo.

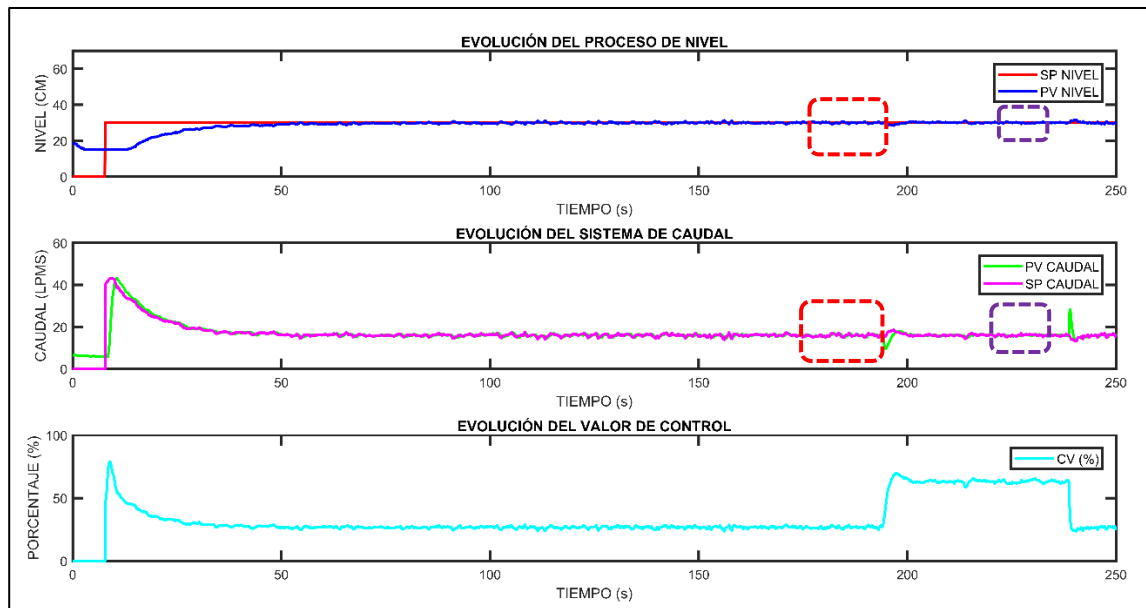
Funcionamiento del sistema en cascada (Caudal PID-Nivel PID)

Desempeño del controlador (Caudal PID-Nivel PID)

Se observa el desempeño que tendrá el sistema de Caudal-Nivel al ser implementado el controlador PID Nivel y Caudal PID mediante el método de control avanzado en cascada; en este proceso se pone en funcionamiento la planta con un set point de 30 Cm, como se observa en la Figura 94.

Figura 94

Respuesta del controlador PID, método de control avanzado en cascada.



Nota. En la figura se observa la respuesta del controlador en cascada (Caudal PID – Nivel PID), se observa que la señal de color azul corresponde a la salida de PID nivel(PV NIVEL) y la señal de color verde es la salida PID Caudal (PV-CAUDAL), para observar la eficiencia del controlador se realiza dos perturbaciones como se aprecia en el recuadro rojo y morado en los tiempos de 195s y 240s respectivamente; lo cual se puede observar claramente que el lazo interno(Caudal) compensa estas perturbaciones para así mantener un mínimo cambio en el lazo externo (Nivel) y así mantenerse en su punto de consigna que en este caso es 30.

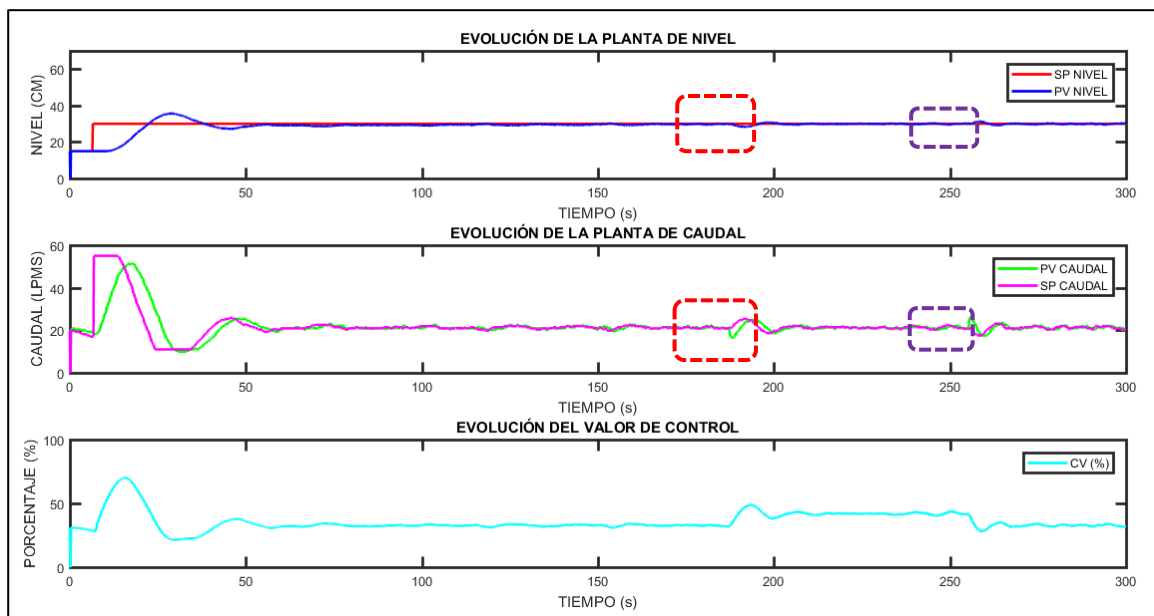
Funcionamiento del sistema en cascada (Caudal fuzzy-Nivel PID)

Desempeño del controlador (Caudal fuzzy-Nivel PID)

Se observa el desempeño que tendrá el sistema de Caudal-Nivel al ser implementado el controlador PID Nivel y Caudal Fuzzy mediante el método de control avanzado en cascada; en este proceso se pone en funcionamiento la planta con un set point de 30 Cm, como se observa en la Figura 95.

Figura 95

Respuesta del controlador (Caudal fuzzy-Nivel PID) en cascada.



Nota. En la figura se observa la respuesta del controlador en cascada (Caudal Fuzzy – Nivel PID), se observa que la señal de color azul corresponde a la salida de PID nivel(PV NIVEL) y la señal de color verde es la salida fuzzy Caudal (PV-CAUDAL), para observar la eficiencia del controlador se realiza dos perturbaciones como se aprecia en el recuadro rojo y morado en los tiempos de 190s y 260s respectivamente; lo cual se puede observar claramente que el lazo interno(Caudal) compensa estas perturbaciones para así mantener un mínimo cambio en el lazo externo (Nivel) y así mantenerse en su punto de consigna que en este caso es 30.

Funcionamiento del sistema de supervisión y control IloT

Se observa el desempeño que tendrá la estación de nivel ND-0704 con la integración del sistema de supervisión y control mediante IloT, como se observa en las Figuras 96,97,98,99,100 y 101 respectivamente.

Figura 96

Sistema de supervisión y control IloT

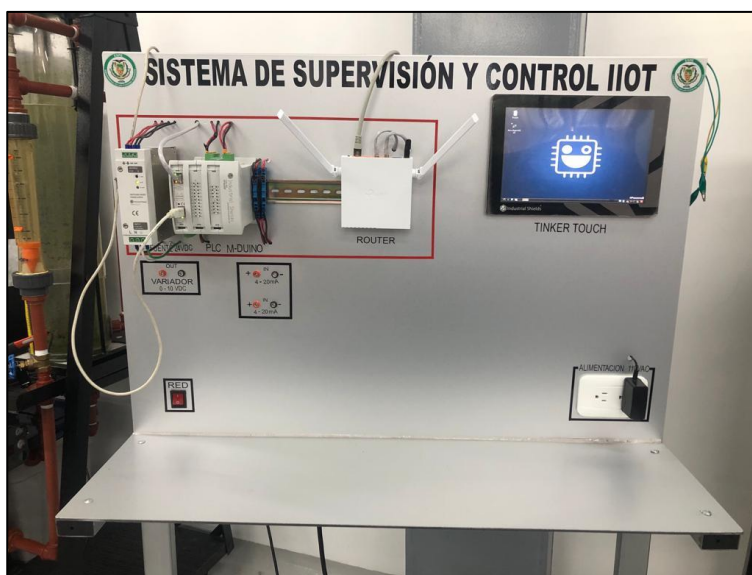


Figura 97

Supervisión y control en la Tinker Touch a un punto de consigna de 30cm

Control	
Control	
Setpoint Nivel - Caudal 30	
Nivel (cm)	27.45
Caudal (LPMS)	16.4
CV Lazo 1	16.94
CV Lazo 2	27.84

Figura 98

Supervisión y control en la Tinker Touch a un punto de consigna de 30cm

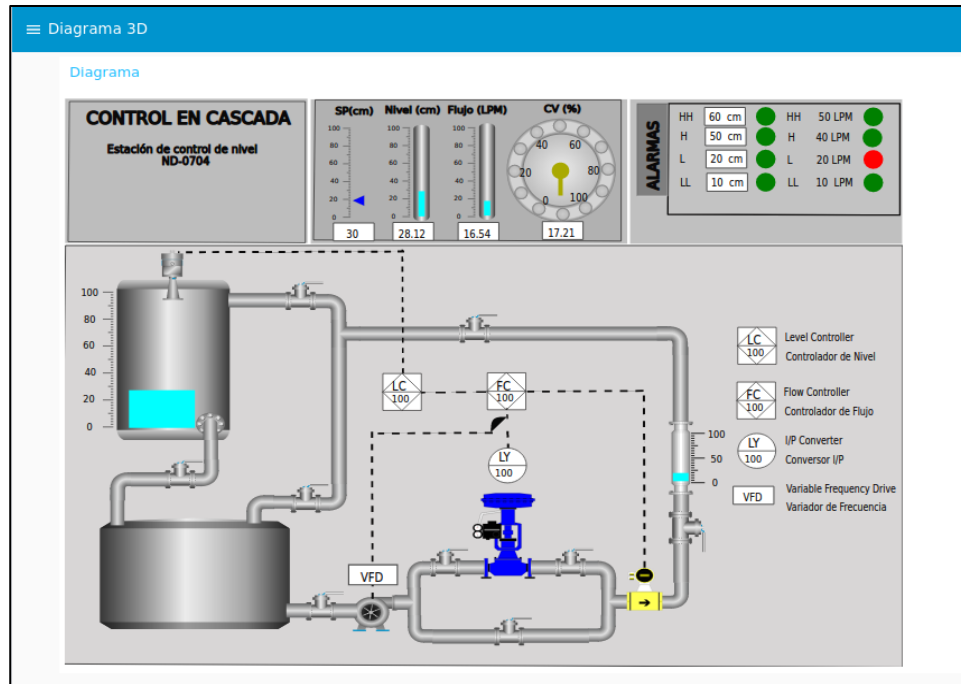


Figura 99

Diagrama P&ID visualizado en la Tinker Touch

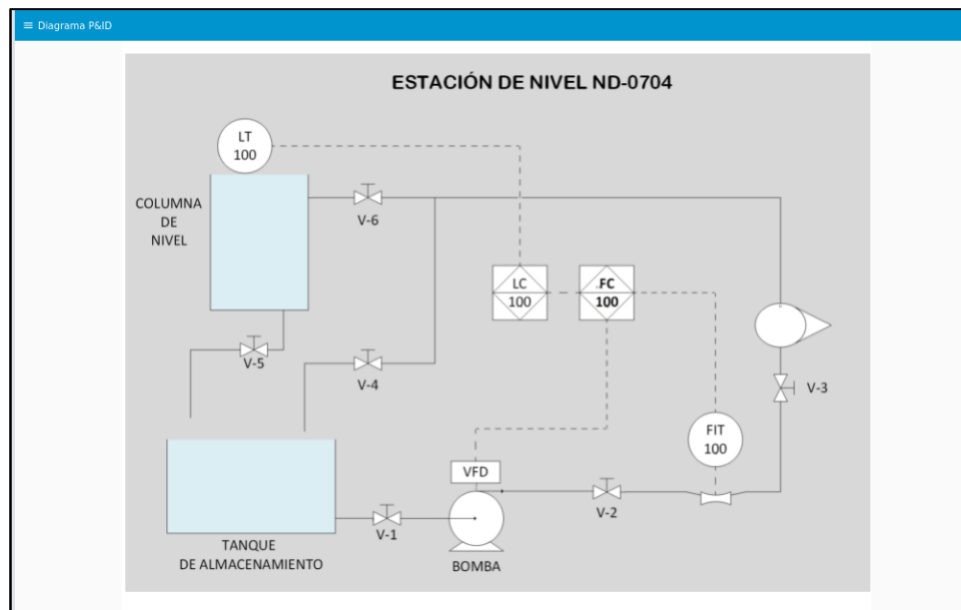
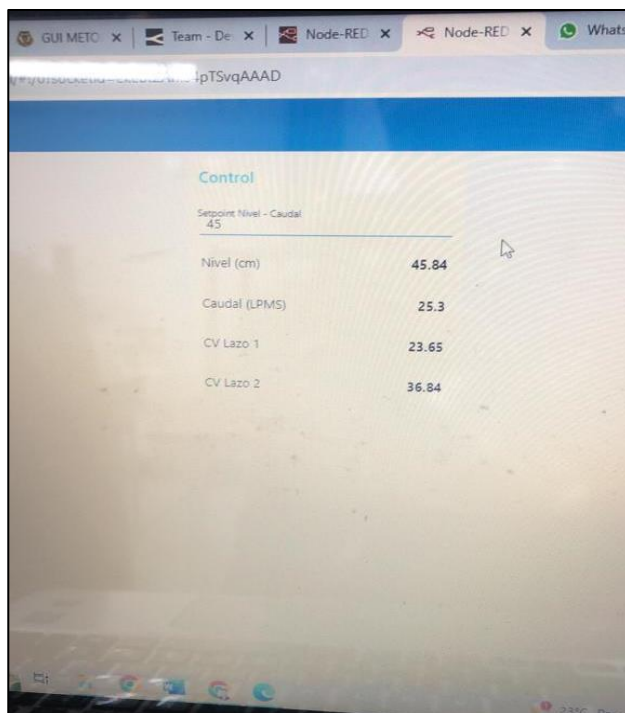
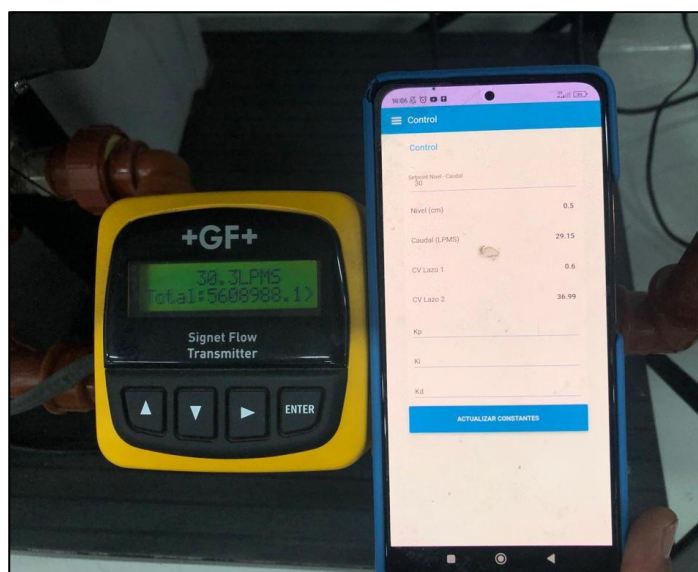


Figura 100

Supervisión y control desde una laptop a un punto de consigna de 45cm

**Figura 101**

Supervisión y control desde la APK a un punto de consigna de 30 LPMS



Costos Autómatas, pantallas y módulos IIoT de Industrial Shields Vs Siemens.

Tabla 11

Costos de dispositivos.

SIEMENS	COSTO (\$)	INDUSTRIAL SHIELDS	COSTO (\$)
PLC S7		PLC M-DUINO	
1200(6ES7214 - 1BG40-0XB0)	862	38AR+	350
SIMATIC HMI SIEMENS 7"	2851	TINKER TOUCH 10"	300
MÓDULO IIoT (SIMATIC IOT 2000)	800		
MODULO DE SALIDAS ANALÓGICAS	650		
TOTAL:	5163		650

Nota. En la Tabla se observa que el costo del PLC M-DUINO 38AR+ en conjunto con la Tinker Touch 10" adquiridos en Industrial Shields es de \$650, en comparación a los costos del PLC S7 1200, Simatic HMI Siemens 7", Simatic IOT 2000 y un módulo de salidas analógicas, todos en marca Siemens es de \$5163.

Por lo tanto, se justifica que el trabajo de titulación se lo realizó con dispositivos de bajo costo, pero demostrando la eficiencia de los mismos.

Capítulo V

Conclusiones y recomendaciones

Conclusiones

- La implementación de un sistema de supervisión y control avanzado en la estación de nivel ND-0704 ha demostrado ser un enfoque eficiente y efectivo a la industria 4.0 para optimizar el funcionamiento y el rendimiento de la estación. La integración de tecnologías como IIoT y PLC M-DUINO ha permitido obtener un control más preciso y una supervisión en tiempo real, lo que es fundamental para mejorar la eficiencia operativa del proceso.
- Se implementaron los algoritmos de control PI, PID mediante el método analítico lo que permitió el uso de los métodos de sintonía Cohen Coon, Haalman y Lambda para los sistemas de caudal y nivel. Del análisis realizado en función del controlador que presente menor sobre impulso (%OS) y menor tiempo de establecimiento (T_s) y una señal de control estable, se obtiene el controlador PI Lambda Robusto para el sistema de caudal (%OS=13.33 y $T_s=12s$) y el controlador PI Lambda Agresivo para el sistema de nivel (%OS=0 y $T_s=325s$).
- Se implementó el algoritmo de control Fuzzy Logic (Mamdani), realizando la fusificación, mecanismo de inferencia, reglas y defusificación; para el sistema de caudal se utilizaron 49 reglas. Del análisis realizado al controlador difuso para el sistema de caudal se obtiene (%OS=0.2 y $T_s=25s$).
- El diseño del HMI en la Tinker Touch ha facilitado el monitoreo local del sistema de control, brindando a los operadores una visión completa y accesible de todas las variables relevantes. Esto ha aumentado la habilidad para tomar decisiones y la respuesta rápida en responder a situaciones inesperadas.

- El desarrollo del algoritmo de comunicación en el PLC para la comunicación IIoT ha permitido una integración fluida con la aplicación de supervisión (APK), lo que ha posibilitado el acceso remoto y la monitorización en tiempo real del sistema de control desde ubicaciones externas.
- Las pruebas realizadas para verificar el funcionamiento del sistema de control en cascada (Caudal Fuzzy- Nivel PID), han proporcionado información valiosa sobre el rendimiento de los equipos e instrumentos industriales y han permitido identificar el controlador más eficiente para este caso particular fue el proporcional integral Lambda Agresivo para el proceso de nivel. Se logró observar que, al realizar perturbaciones, el lazo secundario las compensaba rápidamente para que el lazo primario sufra un mínimo cambio y así mantener el punto de consigna.
- Las pruebas de funcionamiento del sistema de supervisión han confirmado su adecuado desempeño y comunicación, asegurando que los datos relevantes se transmitan con precisión y en tiempo real para su análisis y toma de decisiones.
- El desarrollo de la comunicación industrial con la nube se logró con la ayuda de la utilización de la herramienta Node-Red para la programación en bloques mediante librerías de protocolos de comunicación como es el MQTT (Tinker Touch- Nube y viceversa) y Modbus TCP/IP para la comunicación con el autómeta (Tinker Touch- Autómata y viceversa).

Recomendaciones

- Para el correcto funcionamiento del PLC M-DUINO y la Tinker Touch se deben considerar el manual de los mismos, proporcionados por Industrial Shields para así evitar posibles inconvenientes en el futuro.
- Se recomienda tener en cuenta que tipo de proceso se va a controlar, ya que difieren entre si los métodos de sintonía para los controladores, en este caso específico se consideró un proceso auto regulado tanto para caudal y nivel.
- Para la programación en el software Arduino se recomienda tener conocimiento básico en programación del lenguaje #C.

Bibliografía

- AG, P. (2020). *Paessler*. Obtenido de <https://www.paessler.com/es/it-explained/iiot>
- Aguilar , A., & Steenkiste, P. (10 de Septiembre de 2020). *IEEE Xplore*. Obtenido de <https://ieeexplore.ieee.org/document/9191643>
- Astrom, K., & Hagglund, T. (2009). *Control PID avanzado*. Madrid, España: Pearson Educación.
Recuperado el 28 de Febrero de 2021
- Azuaje, I., Moira, M., & Iglesias, E. (2018). *Controlador difuso mejorado para estrategias de control en cascada y por acción anticipada* (Vol. 39). Venezuela: Universidad de los Andes. Recuperado el 25 de Febrero de 2021
- Carter, B. (2002). Obtenido de <https://www.ti.com/lit/an/sloa097/sloa097.pdf>
- Cubillos, F. (2019). *Labcontrol.cl*. Recuperado el 26 de Febrero de 2021, de <https://www.labcontrol.cl/sites/labcontrol/files/controlavanzado.pdf>
- Debian, P. (2021). *Debian*. Obtenido de <https://www.debian.org/>
- Editores, I. (2022). *IIoT Editores*. Recuperado el 27 de Febrero de 2021, de <https://editores-srl.com.ar/taxonomy/term/13609>
- Emerson. (2012). *Process Management*. Obtenido de <https://www.emerson.com/documents/automation/hoja-de-datos-del-producto-rosemount-serie-5400-transmisor-de-nivel-por-radar-sin-contacto-de-dos-hilos-y-con-las-mejores-prestaciones-es-es-88842.pdf>
- Fischer, G. (2014). *Pureaqua*. Obtenido de <https://es.pureaqua.com/content/Pdf%20Files/signet-totalizador-de-flujo-8150.pdf>

- Hughes, j. (2018). *Pick Data*. Obtenido de https://www.pickdata.net/es/noticias/node-red-programacion-visual-iot?gclid=EAlalQobChMlp67MvO7Q_wlVx0d_AB2pyQIMEAAYASAAEgIWB_D_BwE
- Instruments, N. (2023). *LabVIEW*. Obtenido de <https://www.ni.com/es-cr/shop/labview.html>
- Instruments, N. (2023). *LABVIEW*. Recuperado el 24 de Febrero de 2021, de <https://www.ni.com/en-us/shop/labview.html>
- Kartyy. (2022). *Kartyy*. Obtenido de <https://www.kartyy.com/categorias/7563-router-tp-link-wr844n-wireless-24ghz-300mbps-2-antenas-de-5-dbi-tecnologia-.html>
- MathWorks. (2021). *Matlab*. Recuperado el 28 de Febrero de 2021, de MathWorks: <https://la.mathworks.com/products/matlab.html>
- Modbus*. (2020). Obtenido de <https://www.modbus.org/specs.php>
- Nick. (2023). *FlowForge*. Obtenido de <https://flowforge.com/>
- Nise, N. (2006). *Sistemas de control para Ingeniería*. Mexico: Continental.
- Nodo-RED*. (Septiembre de 2013). Obtenido de <https://nodered.org/about/>
- Ogata, K. (2010). *Ingeniería de control moderna*. Madrid, España: Pearson Educación.
- Pomeyrol, J. (2012). *MCPRO*. Recuperado el 28 de Febrero de 2021, de <https://www.muycomputerpro.com/2012/01/11/debian-vuelve-a-ser-la-distribucion-linux-mas-popular-en-servidores>
- Rafael, P. P. (2014). *E-Reading*. Recuperado el 25 de Febrero de 2021, de <https://biblus.us.es/bibing/proyectos/abreproy/12070/fichero/4.+CAP%C3%8DTULO>
- Realpars. (2023). *Realpars*. Obtenido de <https://realpars.com/node-red/>

Roldan, P. N. (03 de 01 de 2019). *Economipedia*. Recuperado el 20 de Febrero de 2021, de <https://economipedia.com/definiciones/modelo-matematico.html>

Sáez, D. (2006). *Fundamentos de la lógica difusa*. Recuperado el 28 de Febrero de 2021, de https://www.u-cursos.cl/ingenieria/2008/1/EL42D/1/material_docente/bajar?id_material=163394

Sanchez, J. A. (2003). *Control Avanzado de Procesos*. Madrid, España: Díaz de Santos S.A.

Semle, A. (2016). Obtenido de https://www.editores-srl.com.ar/revistas/aa/2/semle_protocolos_ilot

Shields, I. (2019). *Cadenza Electric*. Obtenido de https://www.cadenzaelectric.com/plc-arduino-industrial_c632470/

Smith, C., & Corripio, A. (1991). *CONTROL AUTOMÁTICO DE PROCESOS*. Mexico: Noriega Editores.

TechinDroid. (23 de 6 de 2023). *TechinDroid*. Recuperado el 27 de Febrero de 2021, de <https://techindroid.com/install-apk-file-android/>

Toro, A. (2017). *Redalyc*. Obtenido de <https://www.redalyc.org/journal/5075/507555085001/507555085001.pdf>

tragovom. (2013). Obtenido de <https://www.tra.gov.om/pdf/565-nternalcablingen.pdf>

Worton. (15 de 7 de 2021). *FS Community*. Obtenido de <https://community.fs.com/es/blog/t568a-vs-t568b-difference-between-straight-through-and-crossover-cable.html>

Zayas Gato, F., & Quintian Pardo, H. (2020). *RUC*. Recuperado el 24 de Febrero de 2021, de <https://ruc.udc.es/dspace/handle/2183/25824>

Anexos