

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**TEMA:
DISEÑO DE CONTROLADORES PID DIFUSOS DE LAS VARIABLES DE
NIVEL Y TEMPERATURA DE UNA PLATAFORMA DE ENTRENAMIENTO
PARA CONTROL DE PROCESOS CONTINUOS**

**AUTORES:
JOHN JAIRO SÁNCHEZ GÓMEZ
WILLAM FERNANDO BARRERA FLORES**

**DIRECTOR:
WILLIAM MANUEL MONTALVO LÓPEZ**

Quito, mayo de 2015

**DECLARACIÓN DE RESPONSABILIDAD Y AUTORIZACIÓN DE USO
DEL TRABAJO DE TITULACIÓN**

Nosotros, autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de titulación y su reproducción sin fines de lucro.

Además, declaramos que los conceptos, análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

Quito, mayo de 2015.

John Jairo Sánchez Gómez
CI: 2200076509

Willam Fernando Barrera Flores
CI: 1718957382

DEDICATORIA

Con toda la humildad que de mi corazón puede emanar, este proyecto de titulación lo dedico principalmente a Dios; por haberme dado la vida y permitirme llegar hasta este momento tan importante de mi formación profesional. A mi amada madre Rosa Gómez, porque el día de hoy también ella se convierte en una Ingeniera, porque sin duda es que gracias a su ejemplo de superación, humildad y sacrificio he llegado hasta este gran día; además por enseñarme a levantarme y no rendirme jamás ante las adversidades, asimismo fomentar en mí el deseo de superación y de triunfo en la vida. A mi padre Orlando Sánchez, por darme la oportunidad de estudiar y enseñarme que las cosas se consiguen esforzándose y trabajando duro. A Giselle, porque te amo infinitamente hermanita y agradecerte por ser mi mejor amiga. A mi hermano Jefferson, a pesar de nuestra distancia física, siento que estás conmigo siempre y aunque nos faltaron muchas cosas por vivir juntos, sé que este momento hubiera sido tan especial para ti como lo es para mí. En general a toda mi familia y amigos, por sus muestras de alegría y apoyo incondicional al compartir conmigo buenos y malos momentos.

John Jairo

La finalización de este trabajo de titulación está dedicada a Dios y a mis Padres. A Dios porque ha estado conmigo a cada paso que he dado, protegiéndome, cuidándome y dándome fortaleza para continuar, a mis Padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Entregando su confianza en cada desafío que se me presentaba sin dudar ni en un solo momento en mi capacidad.

Al mismo tiempo dedico este proyecto a mis abuelitas, hermanos y amigos que siempre creyeron en mí y supieron brindarme el apoyo necesario en los momentos difíciles, y sobre todo por estar conmigo en los buenos y malos momentos en el transcurso de mi vida estudiantil.

Fernando

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana por brindarnos la oportunidad de cursar nuestros estudios de pregrado, logrando escalar un peldaño más en el campo del conocimiento; y a cada uno de los docentes de la carrera de Ingeniería Electrónica, que no solo caminaron a nuestro lado cultivando y abonando nuestras mentes con su sabiduría, sino guiándonos con paso firme siempre enfocados en la conquista de nuestra meta, de culminar nuestros estudios universitarios y ser excelentes profesionales.

Al Ingeniero William Manuel Montalvo López, a quien expresamos nuestra más profunda gratitud por brindarnos la oportunidad de trabajar bajo su supervisión en nuestro proyecto de tesis. Gracias a su apoyo, sugerencias y dedicación se hizo posible la pronta terminación de nuestro proyecto de titulación.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1	2
ANTECEDENTES.....	2
1.1 Tema.....	2
1.2 Justificación.....	2
1.3 Delimitación.....	3
Delimitación temporal.....	3
Delimitación espacial	3
Delimitación académica	3
1.4 Planteamiento del problema	4
1.5 Objetivos	4
1.5.1 Objetivo General	4
1.5.2 Objetivos Específicos.....	5
1.6 Beneficiarios de la propuesta de intervención	5
1.7 Documento resultante del proyecto.....	6
CAPÍTULO 2	7
FUNDAMENTACIÓN TEÓRICA.....	7
2.1 Sistemas de control Automático.....	7
2.2 Controlador	9
2.2.1 Sistemas de control en lazo abierto	10
2.2.2 Sistemas de control en lazo cerrado	11
2.3 Controlador Proporcional Integral Derivativo – PID.....	11
2.3.1 Estructura del PID	12
2.3.2 Ajuste empírico de controladores PID	13
2.3.2.1 Método de Ziegler y Nichols en lazo abierto	13
2.3.2.2 Método de Ziegler y Nichols en lazo cerrado	15
2.4 CONTROLADORES LÓGICOS PROGRAMABLES - PLC.	16
2.4.1 Introducción al PLC S7-1200.....	17
2.4.2 Software de programación TIA PORTAL	19
2.4.3 Módulos S7-1200	20
2.4.4 Comunicación Industrial	23

2.5 Programación de PLC'S.....	24
2.5.1 Programación TIA PORTAL	24
2.6 Algoritmos de Control Moderno.....	26
2.6.1 Lógica Difusa.....	27
2.6.1.1 Fundamentos de lógica difusa.....	27
2.6.1.2 Conjuntos difusos.....	27
2.6.1.3 Funciones de membresía.....	28
2.6.1.4 Operaciones difusas	30
2.6.1.5 Inferencia difusa.....	31
2.6.1.6 Implicación difusa.....	31
2.6.1.7 Variables y términos lingüísticos	32
2.6.1.8 Modelos difusos lingüísticos.....	32
2.6.1.9 Fusificación.....	33
2.6.1.10 Defusificación	33
2.6.2 Controlador difuso	33
2.6.2.1 Estructura de un controlador difuso	34
2.6.2.2 Controlador PID Difuso	35
2.7 Software de Aplicación de Algoritmos modernos	36
2.7.1 Software Matlab.....	36
2.7.2 Componentes de Matlab.....	37
2.7.3 Herramientas de Matlab	37
2.7.4 Toolbox Fuzzy	38
2.7.5 El editor del sistema difuso	39
2.7.6 Editor de las Funciones de Membrecía	39
2.7.7 Editor de las Reglas.....	40
2.7.8 OPC Toolbox	41
CAPÍTULO 3	42
DISEÑO E IMPLEMENTACIÓN DE CONTROLADORES PID DIFUSOS	42
3.1 Caracterización de la Plataforma de entrenamiento para control de procesos continuos	42
3.1.1 Descripción de los componentes de la Plataforma de entrenamiento para control de procesos continuos	43
3.1.1.1 Elementos de control.....	44
3.2 Diseño de los controladores PID difusos	50

3.2.1 Controladores PID difusos en Matlab	50
3.2.1.1 Controlador PID difuso de la variable de nivel.....	50
3.2.1.2 Controlador PID difuso de la variable de temperatura.....	71
3.2.2 Controladores PID difusos en el PLC Siemens S7-1200	75
3.2.2.1 Controladores PID difusos de las variables de nivel y temperatura.....	75
3.3 Implementación de los controladores PID difusos.....	86
3.3.1 Controladores PID Difusos en Matlab	86
3.3.2 Controladores PID Difusos en el PLC Siemens S7-1200	92
CAPÍTULO 4.....	97
PRUEBAS Y RESULTADOS	97
4.1 Pruebas del controlador PID difuso de nivel.....	97
4.1.1 Respuesta a la señal del Set Point en Matlab	97
4.1.2 Respuesta a la señal del Set Point en el PLC	100
4.1.3 Resultados del controlador PID difuso de nivel.....	101
4.1.3.1 Respuesta a la señal del Set Point en Matlab	101
4.1.3.2 Respuesta a la señal del Set Point en el PLC	102
4.1.4 Respuesta a la señal aplicando perturbación en Matlab.....	103
4.1.5 Respuesta a la señal aplicando perturbación en el PLC	103
4.2 Pruebas del controlador PID difuso de temperatura	104
4.2.1 Respuesta a la señal del Set Point en Matlab	104
4.2.2 Respuesta a la señal del Set Point en el PLC	106
4.2.3 Resultados del controlador PID difuso de temperatura.....	108
4.2.3.1 Respuesta a la señal del Set Point en Matlab	108
4.2.3.2 Respuesta a la señal del Set Point en el PLC	108
4.2.4 Respuesta a la señal aplicando perturbación en Matlab y en el PLC	109
4.3 Análisis de Resultados	109
4.3.1 Comparación de los controladores PID difusos de nivel	110
4.3.2 Comparación de los controladores PID difusos de temperatura	110
CONCLUSIONES	112
RECOMENDACIONES	114
LISTA DE REFERENCIAS	116

ÍNDICE DE FIGURAS

Figura 1. Ubicación de la Universidad Politécnica Salesiana, Sede Quito-Campus Sur.	3
Figura 2. Elementos de un sistema de control en lazo abierto. Fuente: (Kuo, 2010, pág. 9).....	10
Figura 3. Sistema de control en lazo cerrado. Fuente: (Kuo, 2010, pág. 10).....	11
Figura 4. Control Proporcional, Integral y Derivativo (PID) en bucle cerrado. Fuente: (Mandado, 2010, pág. 384)	12
Figura 5. Módulo físico del PLC Siemens S7-1200. Fuente: (Siemens)	18
Figura 6. Módulo de señales análogas para el PLC Siemens S7-1200. Fuente: (Siemens)	20
Figura 7. Módulos de comunicación para el PLC Siemens S7-1200. Fuente: (Siemens)	23
Figura 8. Ejemplo de segmento en el programa TIA PORTAL. Fuente: (Siemens) .	25
Figura 9. Bloque de Función SCL. Fuente: (Siemens)	26
Figura 10. Estructura de un Sistema Difuso. Fuente: (Ponce, 2010, pág. 71)	33
Figura 11. Estructura de un Sistema Difuso. Fuente: (Ponce, 2010, pág. 15)	35
Figura 12. Estructura de un Sistema Difuso. Fuente: (Ponce, 2010, pág. 15)	35
Figura 13. Ambiente de programación de Matlab. Fuente: (Matlab)	37
Figura 14. Herramientas de Lógica Difusa. Fuente: (Marrón, Departamento de Electrónica)	38
Figura 15. Editor FIS en Matlab. Fuente: (Escuela Politécnica Superior de Elche, Departamento de Electrónica).....	39
Figura 16. Funciones de membresía pertenecientes al test de inteligencia. Fuente: (Escuela Politécnica Superior de Elche, Departamento de Electrónica).....	40
Figura 17. Reglas implementadas para el sistema exacto. Fuente: (Escuela Politécnica Superior de Elche, Departamento de Electrónica)	41
Figura 18. OPC Toolbox.	41
Figura 19. Componentes de la plataforma de entrenamiento. Fuente: (Plataforma de entrenamiento para control de procesos continuos)	43
Figura 20. PLC Siemens S7-1200. Fuente: (Plataforma de entrenamiento para control de procesos continuos)	44

Figura 21. Transmisor de nivel. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	45
Figura 22. Sensor de temperatura - RTD. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	45
Figura 23. Transductor electro neumático. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	46
Figura 24. Servo válvula. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	46
Figura 25. Bomba de agua. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	47
Figura 26. Niquelina de 700 W. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	47
Figura 27. Niquelina de 1500 W. Fuente: (Plataforma de entrenamiento para control de procesos continuos).....	47
Figura 28. Diagrama de bloques del controlador PID difuso de la variable de nivel.	50
Figura 29. Pantalla principal del KEPServerEX. Fuente: (Programa KEPServerEx)	51
Figura 30. Configuración del KEPServerEx OPC – Pasos 1 y 2. Fuente: (Programa KEPServerEX)	52
Figura 31. Configuración del KEPServerEx OPC – Paso 3. Fuente: (Programa KEPServerEX)	52
Figura 32 Variables de intercambio para la comunicación entre Matlab y el PLC. ..	53
Figura 33. Comunicación en Launch OPC Quick Client. Fuente: (Programa KEPServerEX)	53
Figura 34. Verificación de la comunicación del KEPServerEX OPC con el PLC Siemens S7-1200.....	53
Figura 35. Bloques OPC dentro de Simulink. Fuente: (Programa Matlab/Simulink)	54
Figura 36. Bloque de configuración OPC. Fuente: (Programa Matlab/Simulink).....	55
Figura 37. Bloque de lectura OPC. Fuente : (Programa Matlab/Simulink).....	56
Figura 38. Bloque de escritura OPC. Fuente : (Programa Matlab/Simulink)	57
Figura 39. Controlador PID difuso para el nivel del líquido.....	58
Figura 40. Valor del Set Point escalado.	59
Figura 41. Lectura de la variable de proceso.	59
Figura 42. Visualización del PID Difuso.	60

Figura 43. Acción de control.....	60
Figura 44. Ventana de la Liberia Difusa. Fuente: (Programa Matlab).....	62
Figura 45. Funciones de membrecía para el error. Fuente: (Programa Matlab)	63
Figura 46. Funciones de membrecía para la derivada del error. Fuente: (Programa Matlab).....	63
Figura 47. Funciones de membrecía de la acción de control. Fuente: (Programa Matlab).....	64
Figura 48. Edición de las 25 reglas de fusificación. Fuente: (Programa Matlab).....	65
Figura 49. Ventana para crear nuevo proyecto GUIDE. Fuente: (Programa Matlab).....	65
Figura 50. Interfaz gráfica de usuario en blanco. Fuente: (Programa Matlab)	66
Figura 51. Interfaz gráfica creada en Guide. Fuente: (Programa Matlab)	66
Figura 52. Interfaz gráfica guardada con la extensión fig. Fuente: Programa Matlab)	67
Figura 53. Creación de un nuevo proyecto. Fuente: (Programa TIA PORTAL)	68
Figura 54. Selección del CPU. Fuente: (Programa TIA PORTAL).....	68
Figura 55. Módulo de expansión de entradas y salidas análogas SM-1234. Fuente: (Programa TIA PORTAL)	69
Figura 56. Configuración de los parámetros de variables de entradas y salidas análogas. Fuente: (Programa TIA PORTAL)	70
Figura 57. Proyecto en blanco configurado las entradas y salidas análogas. Fuente: (Programa TIA PORTAL)	70
Figura 58. Diagrama de bloques del controlador PID difuso de la variable de temperatura.....	71
Figura 59. Controlador PID difuso para la temperatura del líquido.	72
Figura 60. Valor del Set Point escalado.	73
Figura 61. Lectura de la variable del transmisor.....	73
Figura 62. Controlador PID difuso.....	74
Figura 63. Acción de control para el actuador.	74
Figura 64. Bloque Lógico FC1.....	76
Figura 65. Limitación del error.	76
Figura 66. Limitación de la derivada del error.....	77
Figura 67. Máquina fusificadora.	77
Figura 68. Máquina defusificadora.	78
Figura 69. Bloque de datos DB1.	78

Figura 70. Bloque Lógico FC5.....	79
Figura 71. Bloque Lógico FC2.....	79
Figura 72. Bloque Lógico FC3.....	80
Figura 73. Bloque Lógico FC4.....	80
Figura 74. Bloque de Datos DB4.	81
Figura 75. Bloque de datos DB5.	81
Figura 76. Bloque Main OB1.....	82
Figura 77. Ventana para agregar dispositivo HMI. Fuente: (Programa TIA PORTAL)	83
Figura 78. Configuración de conexión al PLC. Fuente: (Programa TIA PORTAL) .	83
Figura 79. Ventana de imagen raíz de la interfaz gráfica del controlador PID difuso.	84
Figura 80. Ventana de imagen de la interfaz gráfica del controlador PID difuso de nivel.....	85
Figura 81. Ventana de imagen de la interfaz gráfica del controlador PID difuso de temperatura.....	85
Figura 82. Proyecto en blanco. Fuente: (Programa TIA PORTAL)	86
Figura 83. Proyecto creado en KEPServerEX. Fuente: (Programa KEPServerEX)..	87
Figura 84. Comando TESIS.	87
Figura 85. Exportar archivo fis a la ventana workspace.	88
Figura 86. Interfaz Gráfica CARATULA.	88
Figura 87. Interfaz Gráfica del controlador PID difuso de nivel.	89
Figura 88. Interfaz gráfica del controlador PID difuso de temperatura.	90
Figura 89. Flujo del proceso para el control del PID Difuso con Matlab	91
Figura 90. Proyecto PID Difuso Nivel y Temperatura. Fuente: (Programa TIA PORTAL).....	92
Figura 91. Compilación del dispositivo HMI.	92
Figura 92. Interfaz Gráfica CARATULA.	93
Figura 93. Interfaz Gráfica del controlador PID difuso de Nivel.	94
Figura 94. Interfaz Gráfica del controlador PID difuso de temperatura.	95
Figura 95. Flujo del proceso para el control del PID difuso con el PLC Siemens S7- 1200.....	96
Figura 96. Cambios del Set Point.....	97
Figura 97. Respuesta del sistema al Set Point, desde cero a sesenta centímetros.....	98

Figura 98. Respuesta del sistema al Set Point, desde sesenta a cincuenta y cinco centímetros.....	99
Figura 99. Respuesta del sistema al Set Point, desde veinte a sesenta centímetros..	99
Figura 100. Respuesta del sistema al Set Point, desde diez a sesenta centímetros.	100
Figura 101. Respuesta del sistema al Set Point, desde sesenta a cincuenta y cinco centímetros.....	101
Figura 102. Respuesta del sistema aplicando perturbación, desde treinta a veinte centímetros.....	103
Figura 103. Respuesta del sistema aplicando perturbación, desde treinta a veinte centímetros.....	104
Figura 104. Respuesta del sistema al Set Point, desde cuarenta a treinta grados centígrados.	105
Figura 105. Respuesta del sistema al Set Point, desde veinte a treinta y cinco grados centígrados.	105
Figura 106. Respuesta del sistema al Set Point, desde cincuenta a cuarenta y cinco grados centígrados.....	106
Figura 107. Respuesta del sistema al Set Point, desde treinta a treinta y cinco grados centígrados.	107
Figura 108. Comparación de la dinámica de respuesta entre el controlador PID difuso en el programa Matlab y el controlador PID difuso en el PLC Siemens S7-1200.	111

ÍNDICE DE TABLAS

Tabla 1. Ajuste en lazo abierto de los parámetros de un regulador PID propuesta por Ziegler y Nichols.....	14
Tabla 2. Ajuste en lazo cerrado de los parámetros de un regulador PID propuesta por Ziegler y Nichols.....	16
Tabla 3. Módulos de señales digitales.....	21
Tabla 4. Módulos de señales analógicas	22
Tabla 5. Reglas de inferencia para el controlador difuso de nivel	62
Tabla 6. Elementos de la carátula en el software Matlab.....	89
Tabla 7. Elementos de la ventana control PID Difuso de nivel en el software Matlab	89
Tabla 8. Elementos de la ventana control PID Difuso de temperatura en el software Matlab	90
Tabla 9. Elementos de la carátula en el software TIA PORTAL.....	93
Tabla 10. Elementos de la ventana del controlador PID difuso de nivel en el software TIA PORTAL	94
Tabla 11. Elementos de la ventana del controlador PID difuso de temperatura en el software TIA PORTAL.....	95
Tabla 12. Resultado de las pruebas realizadas y cálculo de la mediana	102
Tabla 13. Resultado de las pruebas realizadas y cálculo de la mediana	102
Tabla 14. Resultado de las pruebas realizadas y cálculo de la mediana	108
Tabla 15. Resultado de las pruebas realizadas y cálculo de la mediana	109
Tabla 16. Tabla de comparación de los dos tipos de controladores en Matlab y en el PLC Siemens S7-1200	110
Tabla 17. Tabla de comparación de los dos tipos de controladores en Matlab y en el PLC Siemens S7-1200	110

ÍNDICE DE ECUACIONES

Ecuación 1. Operación de complemento.....	30
Ecuación 2. Operación de unión	30
Ecuación 3. Operación de intersección	31

ÍNDICE DE ANEXOS

Anexo 1. Programación del Controlador PID Difuso en el PLC Siemens S7-1200	119
Anexo 2. Gráficas las pruebas realizadas del Controlador PID Difuso de la variable de nivel para el Matlab.....	125
Anexo 3. Gráficas las pruebas realizadas del Controlador PID Difuso de la variable de nivel para el PLC Siemens S7-1200.....	128
Anexo 4. Gráficas las pruebas realizadas del Controlador PID Difuso de la variable de temperatura para el Matlab.....	131
Anexo 5. Gráficas las pruebas realizadas del Controlador PID Difuso de la variable de temperatura para el PLC Siemens S7-1200.	134
Anexo 6. Prácticas de laboratorio para el control PID Difuso	137

RESUMEN

El presente proyecto tiene como objetivo proporcionar a los estudiantes y docentes de la Universidad Politécnica Salesiana Sede Quito - Campus Sur la oportunidad de fortalecer los conocimientos de controles modernos, aplicados a sistemas de producción continua del tipo industrial.

Se diseña un control Proporcional Integral y Derivativo Difuso – PID FUZZY, de las variables de Nivel y Temperatura, para la “Plataforma de Entrenamiento para Control de Procesos Continuos”, globalizando los conocimientos y la teoría de Instrumentación, Automatización Industrial, Teoría de Control e Informática Industrial, estudiados en la carrera de Ingeniería Electrónica de la UPS sede Quito-Campus Sur.

Se efectúa una comparación del desempeño de este tipo de controladores, elaborados en el programa Matlab y en el PLC Siemens S7-1200, además se elabora un manual técnico de prácticas de los controles PID difusos aplicados a los procesos de Nivel y Temperatura.

ABSTRACT

This project aims to provide students and teachers of the Salesian Polytechnic University South Campus headquarters Quito opportunity to strengthen the knowledge of modern controls, continuous systems applied to industrial production.

Comprehensive Proportional and Derivative Diffuse It is designed - PID FUZZY, variable level and temperature for the "Platform Training Control Continuous Process" globalizing knowledge and theory Instrumentation, Industrial Automation, Control Theory and Industrial computer, studied the race of Electronic Engineering of the Quito-based UPS South Campus.

A comparison of the performance of these controllers, developed in Matlab and the Siemens S7-1200 PLC also a technical manual on our fuzzy PID controls over processes and temperature level is made is made.

INTRODUCCIÓN

El diseño de controladores Proporcional Integral Derivativo – modernos (PID difusos) ayudará a los estudiantes de la Carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, Sede Quito-Campus Sur; a ampliar sus conocimientos en el campo de los sistemas expertos.

El proyecto se ha organizado de la siguiente manera:

El capítulo uno, detalla el tema del proyecto desarrollado, planteamiento del problema, justificación, objetivo general, objetivos específicos y los beneficiarios del proyecto.

El capítulo dos, describe el marco teórico, principales generalidades, conceptos, características de los elementos y dispositivos a emplear en este proyecto; apoyado con literatura especializada.

El capítulo tres, presenta el diseño y desarrollo de controladores PID difusos, para el control de las variables de nivel y temperatura de una “Plataforma de Entrenamiento de Procesos Continuos”.

El capítulo cuatro, presenta las respectivas pruebas de funcionamiento del control PID difuso de las variables de nivel y temperatura, y una comparación del desempeño de su implementación en Matlab y en el PLC Siemens de una “Plataforma de Entrenamiento de Procesos Continuos”, y se detalla las conclusiones y recomendaciones del proyecto realizado.

CAPÍTULO 1

ANTECEDENTES

1.1 Tema

Diseño de controladores PID difusos de las variables de nivel y temperatura de una plataforma de entrenamiento para control de procesos continuos.

1.2 Justificación

El control PID es la estrategia más común y general para el control de procesos continuos, debido a que su ajuste de parámetros no suele ser tan complejo; además que muchos de los controladores que se manejan en la industria emplean esta herramienta.

Las variables a controlar y monitorear son muy utilizadas en la industria ecuatoriana en los diferentes procesos de manufactura como son textiles, medicinas, productos alimenticios; y puesto que los estudiantes de la Carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, Sede Quito-Campus Sur no están familiarizados con equipos de procesos para realizar control difuso; se ha presentado este proyecto para ayudarlos a comprender el control PID (Proporcional-Integrador-Derivativo) difuso real a nivel industrial.

La plataforma de entrenamiento para control de procesos continuos complementará los conocimientos teóricos adquiridos durante la formación académica en la Carrera de Ingeniería Electrónica y afianzará el aprendizaje práctico al utilizar el control PID difuso.

1.3 Delimitación

Delimitación temporal

El presente proyecto tendrá una duración de 12 meses a partir de la aprobación del plan incluyendo la evaluación del mismo.

Delimitación espacial

Este proyecto se llevará a cabo en la ciudad de Quito, en la Universidad Politécnica Salesiana, Sede Quito-Campus Sur, para el laboratorio de Teoría de Control de la Carrera de Ingeniería Electrónica.

Ubicación de la Universidad Politécnica Salesiana

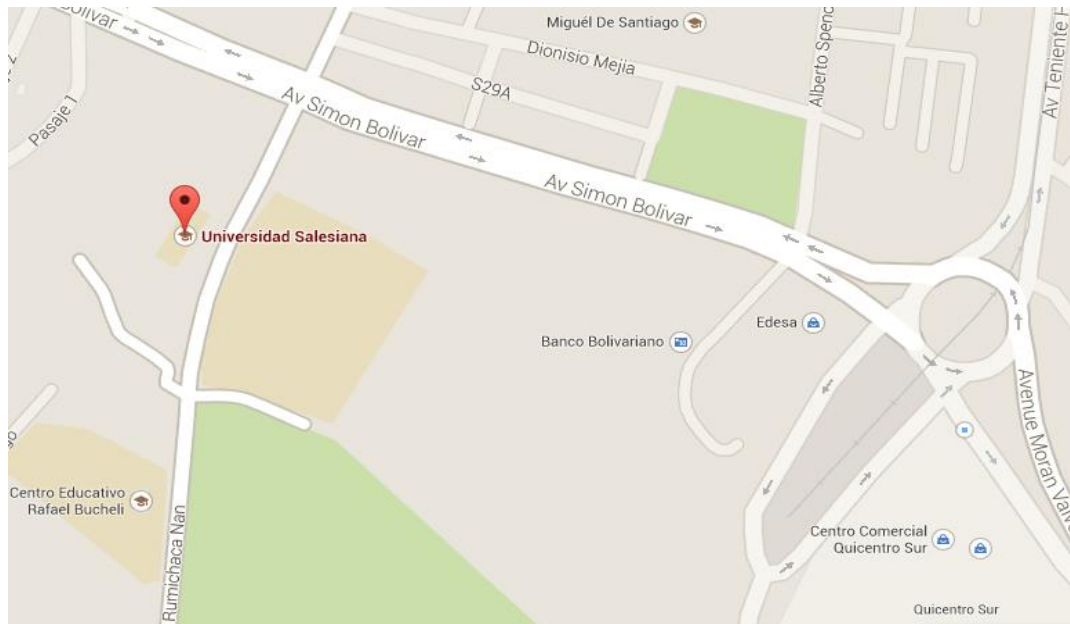


Figura 1. Ubicación de la Universidad Politécnica Salesiana, Sede Quito-Campus Sur.
Fuente: (Google Maps)

Delimitación académica

En este proyecto se desarrollarán los controles PID difusos independientes para las variables de nivel y temperatura de la plataforma de entrenamiento para control de procesos continuos mediante el controlador PLC Siemens S7-1200 el cual dispondrá del protocolo Ethernet con una comunicación OPC Server hacia el software Matlab,

empleando equipos de instrumentación y automatización industrial, además de enlazar los aspectos fundamentales aprendidos durante el estudio de la carrera de Ingeniería Electrónica.

Se efectuara una comparación del desempeño de los controladores elaborados el software Matlab y en el PLC Siemens S7-1200, además se elaborará un manual técnico de prácticas de los controles PID difusos de los procesos de nivel y temperatura.

1.4 Planteamiento del problema

Con el avance de la tecnología en la industria y su necesidad de entregar productos de alta calidad ha hecho que muchos de los controles convencionales queden obsoletos, entre estos avances se encuentran los sistemas difusos.

Los estudiantes de la Carrera de Ingeniería Electrónica de la Universidad Politécnica Salesiana, Sede Quito-Campus Sur no están familiarizados con equipos de procesos para realizar control difuso; por lo tanto se presentará este proyecto para ayudarlos a comprender el control PID (Proporcional-Integrador-Derivativo) difuso real a nivel industrial.

Gracias a la plataforma de entrenamiento para control de procesos continuos se conseguirá diseñar controladores de PID difusos y los estudiantes podrán realizar prácticas para control de variables de nivel y temperatura con el PLC Siemens S7-1200.

1.5 Objetivos

1.5.1 Objetivo general

Diseñar controladores PID difusos de las variables de nivel y temperatura de una plataforma de entrenamiento para el control de procesos continuos.

1.5.2 Objetivos específicos

- Diseñar e implementar el algoritmo de control PID difuso en el software Matlab para control de los procesos de nivel y temperatura.
- Diseñar e implementar el algoritmo de control PID difuso para el PLC Siemens S7-1200 para el control de las variables de nivel y temperatura.
- Comunicar el PLC Siemens S7-1200 con el software Matlab mediante el protocolo de comunicación Ethernet para adquirir datos, ser procesados por software y actuar sobre la planta.
- Comparar el desempeño de los controladores en el software Matlab y en el PLC para optimizar el proceso.
- Elaborar un manual técnico de prácticas con control PID difuso en la plataforma de entrenamiento de proceso de nivel y temperatura.

1.6 Beneficiarios de la propuesta de intervención

Este proyecto beneficiará a los estudiantes y docentes de la Universidad Politécnica Salesiana, Sede Quito-Campus Sur, de la Carrera de Ingeniería Electrónica para el laboratorio de Teoría de Control; ya que los ayudará a complementar los conocimientos adquiridos a lo largo de su formación estudiantil.

La Universidad Politécnica Salesiana Sede Quito-Campus Sur tendrá la capacidad de ofertar cursos y seminarios tanto a empresas públicas, privadas y estudiantes de otras universidades, ya que contará con una plataforma de entrenamiento de procesos continuos que permitirá desarrollar conocimientos en el control PID difuso de las variables de nivel y temperatura; además de ofertar cursos de actualización a los docentes para entrenarse en el control PID difuso.

El módulo quedará como inventario de la Universidad Politécnica Salesiana Sede Quito-Campus Sur incrementando su patrimonio, ayudando a mejorar los estándares de aprendizaje.

1.7 Documento resultante del proyecto

Se propone un paper con el tema “Diseño de controladores PID difusos de las variables de nivel y temperatura de una plataforma de entrenamiento para control de procesos continuos”; para la revista INGENIUS de la Universidad Politécnica Salesiana - UPS.

CAPÍTULO 2

FUNDAMENTACIÓN TEÓRICA

Este capítulo describe el marco teórico, principales generalidades, conceptos, características de los elementos y dispositivos a emplear en este proyecto, apoyado con literatura especializada.

2.1 Sistemas de control automático

En 2010, Ogata menciona que el control automático ha desempeñado un papel vital en el avance de la ingeniería y la ciencia convirtiéndose en una parte importante de los avances tecnológicos de la actualidad; los sistemas robóticos, procesos modernos de fabricación y cualquier operación industrial que requiera el control de temperatura, presión, humedad, flujo, etc, se basan en control automático.

Los sistemas de control automático son fundamentales para el manejo de los procesos de producción de las plantas industriales. Está comprobado que el aumento de la productividad está muy relacionado a la automatización de los procesos en la medida que se haga un uso eficiente de los equipos y sistemas asociados.

Antes de analizar los sistemas de control, Ogata define ciertos términos básicos.

- **Variable de proceso – PV:** es la cantidad o condición que se mide y controla, normalmente es la salida del sistema.
- **La señal de control o variable manipulada – MV:** es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada.
- **Controlar:** significa medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar la desviación del valor medido respecto del valor deseado.

- **Planta:** una planta puede ser una parte de un equipo, tal vez un conjunto de los elementos de una máquina que funcionan juntos, y cuyo objetivo es efectuar una operación particular.
- **Sistema:** es una combinación de componentes que actúan juntos y realizan un objetivo determinado.
- **Perturbación:** es una señal que tiende a afectar negativamente el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna. Mientras que una perturbación externa se genera fuera del sistema y es una entrada.
- **Sensor:** elemento primario que se encarga de captar la magnitud física o química directamente del proceso.
- **Instrumento:** es un dispositivo que se encarga de interpretar señales proporcionales a la magnitud de la variable.
- **Transductor:** dispositivo que recibe una o varias señales provenientes de la variable medida y pueden modificarla o no en otra señal.
- **Transmisor:** captan la variable de proceso a través del elemento primario y la transmiten a distancia en forma de señal neumática de 3 a 15 psi o electrónica de 4 a 20 mA.
- **Control de proceso continuo:** trata de mantener, mediante un sistema automático, las principales variables de un proceso en valores próximos a los deseados a pesar de las posibles perturbaciones.
- **Proceso continuo:** es aquello cuyo producto final, en lugar de estar formado por un conjunto de elementos separados, está constituido por un material que fluye de forma continua.

Este tipo de proceso se caracteriza por que en ello es necesario medir y controlar elementos.

- **Variable de temperatura:** es una medida del estado térmico de un cuerpo que se corresponde con el sentido del tacto, la temperatura cuantifica si está más frío o más caliente.
- **Variable de nivel:** es la altura de un fluido o producto en un tanque o columna.
- **Interfaz gráfica de control:** proporciona al operador las funciones de control y supervisión de la planta. El proceso se representa mediante gráficos.

2.2 Controlador

Los controladores fueron los primeros instrumentos en ser empleados para propósitos de control. Un algoritmo grabado en su memoria les permitía detectar el valor de una variable física y, sobre la base de un algoritmo de control, envían órdenes que tienen como destino final dispositivos tales como válvulas, motores, etc. Su característica principal, y quizás su punto más débil, es que son generalmente instrumentos de propósito específico; es decir, son diseñados para trabajar generalmente con un solo tipo de variable física como: temperatura, nivel, flujo, etc.

Los PLC's surgieron precisamente porque corrigieron la debilidad de los controladores. Estos dispositivos, como su nombre los sugiere, son también controladores pero tienen la virtud de acoplarse a casi cualquier variable física o situación gracias a que son programables. Se puede decir entonces que los PLC's son controladores de propósito general, pero, si se desea, se los puede convertir de propósito específico, con solo cambiar su programación interna. (Corrales, 2007, pág. 75).

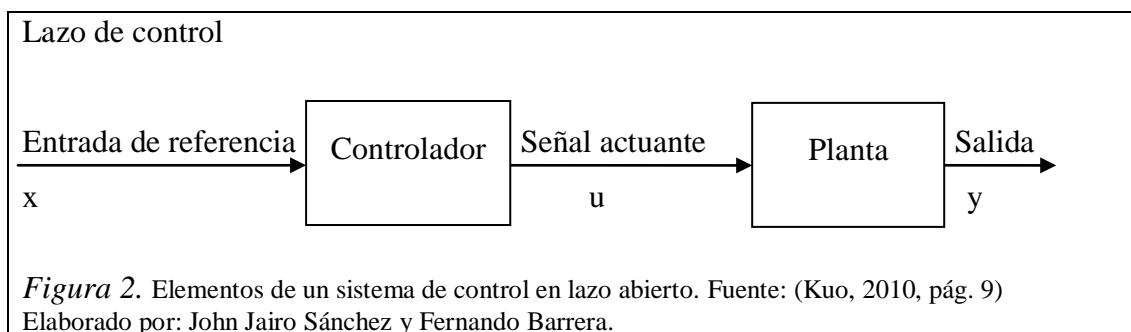
El controlador es un elemento del sistema de lazo cerrado que tiene como entrada la señal de error y determina la salida que se entregará al actuador para mejorar el funcionamiento del sistema.

La relación que existe entre la salida y la entrada del controlador se conoce normalmente como acción de control. La fijación del punto de ajuste (Set Point) en el controlador determina el valor que deberá tener la variable controladora.

2.2.1 Sistemas de control en lazo abierto

En 2010, Ogata menciona que para el control de una planta industrial existen dos tipos de lazo de control, en donde define que un sistema de control en lazo abierto no se mide la salida ni se realimenta para compararla con la entrada. Un ejemplo práctico es una lavadora. El remojo, el lavado y el centrifugado en la lavadora operan con una base de tiempo. La máquina no mide la señal de salida, que es la limpieza de la ropa. En la práctica, el control en lazo abierto sólo se usa si se conoce la relación entre la entrada y la salida y si no hay perturbaciones internas ni externas.

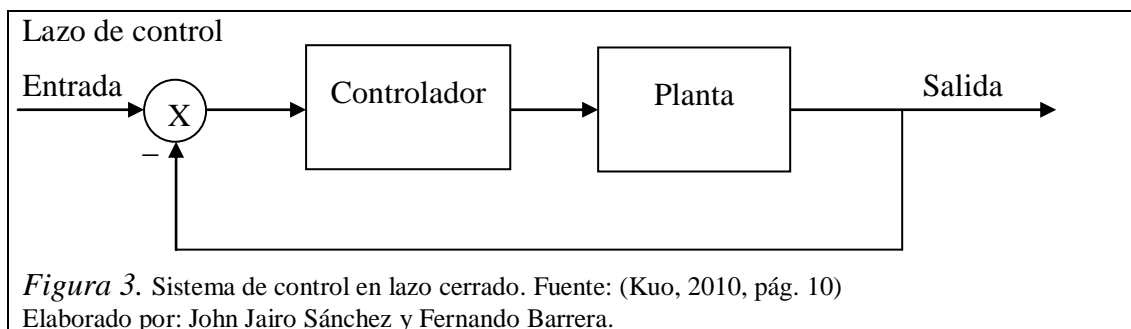
Los elementos de este sistema se pueden dividir en dos partes: el controlador y el proceso controlado, como se muestra en la Figura 2. Una señal de entrada x se aplica al controlador, cuya salida actúa como señal actuante u ; la señal actuante controla el proceso controlado de tal forma que la variable controlada y se desempeñe de acuerdo con estándares preestablecidos.



2.2.2 Sistemas de control en lazo cerrado

En 2010, Kuo menciona que los sistemas de control pueden ser en lazo abierto o lazo cerrado, en donde los más empleados son estos últimos, debido a que se obtiene un control más exacto; la señal controlada y debe ser realimentada y comparada con la entrada de referencia, y se debe enviar una señal actuante proporcional a la diferencia de la entrada y la salida a través del sistema para corregir el error. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema.

Un sistema en lazo cerrado se muestra en la Figura 3.



2.3 Controlador proporcional integral derivativo – PID

En 2010, Ogata sustenta que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID. La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos, es cuando los controles PID resultan más útiles. En el campo de los sistemas para control de procesos, es un hecho bien conocido que los esquemas de control PID básicos y modificados han demostrado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo.

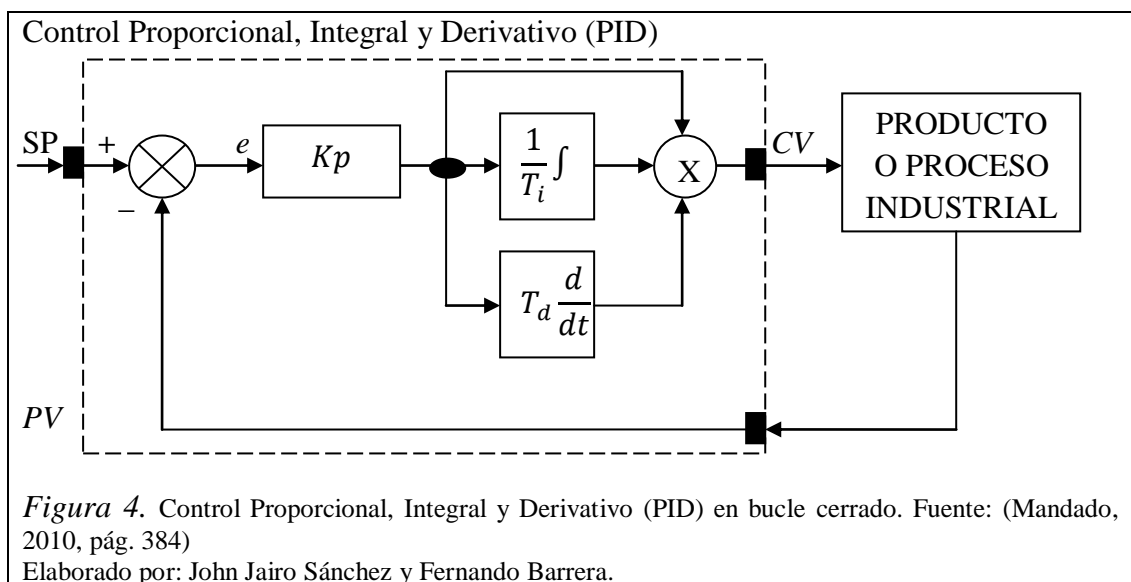
En 2010, Mandado en su trabajo introductorio señala que el regulador “Proporcional, Integral y Derivativo” conocido como PID es un regulador realimentado continuo. La acción integral hace que se anule el error en régimen permanente y la acción

derivativa proporcional al regulador capacidad para anticipar el futuro y tiene un efecto predictivo sobre la salida del proceso, que limita su oscilación en régimen transitorio, al mismo tiempo que aumenta la velocidad de respuesta.

Los controladores PID se utilizan en el 95% de los sistemas de control continuo que existen en la industria, debido en gran parte a su simplicidad, robustez, fiabilidad y a que es posible utilizar un procedimiento de prueba y error para seleccionar la proporción en la que se combinan las tres acciones P, I, y D sin que el usuario tenga que tener un dominio profundo de la teoría de control. Debido a ello en muchas ocasiones el usuario no explota toda la capacidad de control de este tipo de controladores.

2.3.1 Estructura del PID

En 2010, Mandado representa la estructura de control más usada en el medio industrial, en la Figura 4 se muestra el diagrama de bloques de un regulador PID en el que la señal de control CV se obtiene mediante la suma de los tres tipos de acciones. El parámetro T_i , pondera la acción integral y el parámetro T_d pondera la acción derivativa. Además, la señal de error se amplifica mediante el parámetro K_p que afecta también a las acciones integral y derivativa. El diseñador debe asignar a los parámetros $K_p T_i T_d$ un valor positivo o cero.



2.3.2 Ajuste empírico de controladores PID

En 2010, Ogata menciona que alrededor del año 1940 se presentaron varios métodos de ajuste empírico de controladores PID, basados en mediciones realizadas sobre una planta real. Se deduce que para realizar el ajuste o sintonía de un regulador o controlador PID, el diseñador debe conocer las características del sistema a controlar así como el comportamiento que debe tener el regulador ante cambios del punto de consigna (Set Point) o ante perturbaciones.

Utilizando la teoría de control, se pueden calcular matemáticamente los parámetros de un regulador PID, y predecir con precisión el comportamiento del proceso controlado por él. Para ello es necesario modelar la función de transferencia, es decir, obtener la relación entre la salida y la entrada del proceso a controlar, para lo cual se tienen que determinar diferentes parámetros mecánicos (como por ejemplo masa, fricción, inercia, ele.), eléctricos (como por ejemplo inductancia, capacidad, resistencia, factor de potencia, etc.), químicos, etc. La relación entre estos parámetros es difícil de determinar y, por ello, muchos diseñadores obvian esta fase y simplemente ajustan el PID utilizando alguno de los métodos desarrollados por diferentes investigadores. De acuerdo con ello, John G. Ziegler y Nathaniel B. Nichols en 1942 propusieron las denominadas reglas de sintonía de “Ziegler y Nichols”, estimando en primer lugar determinadas características del proceso mediante un experimento y a continuación calculan los parámetros del regulador mediante tablas o fórmulas; clasificándolas en método de ajuste en lazo abierto y método de ajuste en lazo cerrado.

2.3.2.1 Método de Ziegler y Nichols en lazo abierto

El método de sintonía de Ziegler y Nichols fue desarrollado en 1942 por los ingenieros J. G. Ziegler y N. B. Nichols de la empresa Taylor Instruments Company y consta de las siguientes partes:

1. Abrir el lazo de control y ajustar la ganancia proporcional $K_p = 1$, y anular las ganancias integral ($T_i \rightarrow \infty$) y derivada ($T_d = 0$).

2. Aplicar un cambio brusco (escalón) a la variable de control CV.
Con los ajustes realizados dicho cambio se puede conseguir aplicando un escalón a la variable de consigna SP.
3. Registrar gráficamente la evolución en el tiempo de la variable de proceso PV partir del instante en el que se aplica el escalón.
4. Obtener los puntos de intersección de las tres líneas siguientes:
 - La tangente a la curva PV en el punto de máxima pendiente.
 - La recta horizontal correspondiente al valor inicial de PV.
 - La recta horizontal correspondiente al valor final de PV.
5. Asignar al retardo L el valor del tiempo que transcurre desde que se aplica el escalón hasta que se cortan la primera y segunda líneas.
6. Asignar a la constante de tiempo τ el valor del tiempo que transcurre desde el punto de corte anterior hasta el punto de corte de la primera y tercera líneas.
7. Asignar a la ganancia del proceso K el cociente entre el incremento de PV y la amplitud del escalón aplicado a CV.
8. Ajustar las constantes del PID de acuerdo con los valores indicados en la Tabla 1 en la que $a = K \cdot L / \tau$. Tal como se indica en la citada tabla, el regulador resultante puede ser Proporcional (P), Proporcional - Integral (PI) o Proporcional - Integral - Derivativo (PID). (Ponce, 2010, pág. 396).

Tabla 1.

Ajuste en lazo abierto de los parámetros de un regulador PID propuesta por Ziegler y Nichols.

Regulador	K_p	T_i	T_d	T_p
P	$1/a$	-	-	$4 \cdot L$
PI	$0,9/a$	$3 \cdot L$	-	$5,7 \cdot L$
PID	$1,2/a$	$2 \cdot L$	$0,5 \cdot L$	$3,4 \cdot L$

Nota.: Sintonización del PID.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

2.3.2.2 Método de Ziegler y Nichols en lazo cerrado

El objeto que se persigue al sintonizar el regulador PID mediante el método Ziegler y Nichols en lazo cerrado, conocido también como método de la oscilación, es el mismo que en el método Ziegler y Nichols en lazo abierto, es decir, ajustar mediante el uso de tablas los parámetros del regulador para conseguir que el sistema en el lazo cerrado responda con una razón de amortiguamiento de $\frac{1}{4}$.

Mediante este método se determinan dos parámetros denominados ganancia última K_u y periodo último T_u , a partir de los cuales se obtienen los valores de K_p , T_i y T_d .

La sintonía de un PID mediante este método consiste en:

1. Ajustar, con el lazo cerrado, la ganancia proporcional K_p a un valor inicial pequeño y anular las ganancias integral ($T_i \rightarrow \infty$) y derivativa ($T_d = 0$).
2. Incrementar paulatinamente la ganancia proporcional K_p hasta que el sistema presente una oscilación permanente de amplitud constante. Puede ser necesario realizar cambios en el punto de consigna para obtener la oscilación.
3. Asignar a K_u el valor de K_p que origina la situación del punto anterior.
4. Medir el periodo de la oscilación. Este periodo es el periodo último T_u .
5. Ajustar los parámetros del regulador de acuerdo con la Tabla 2. (Ponce, 2010, pág. 404).

Tabla 2.

Ajuste en lazo cerrado de los parámetros de un regulador PID propuesta por Ziegler y Nichols.

Regulador	K_p	T_i	T_d
P	$0,5 K_u$	-	-
PI	$0,4 K_u$	$0,8 T_u$	-
PID	$0,6 K_u$	$0,5 T_u$	$0,125 T_u$

Nota: Sintonización del PID.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

2.4 Controladores lógicos programables - PLC.

En 2010, Mandado explica que los sistemas de control de procesos que utilizan algoritmos lineales de control se pueden implementar con sistemas analógicos o con sistemas digitales y estos últimos constituyen la mejor solución en la actualidad. Uno de los más utilizados es el Controlador Lógico Programable - PLC para implementar PID's en entornos industriales, porque se puede controlar en tiempo real y en medio industrial procesos secuenciales.

Hasta no hace mucho tiempo el control de procesos industriales se venía haciendo de forma cableada por medio de contactores y relees. Al operario que se encontraba a cargo de este tipo de instalaciones, se le exigía tener altos conocimientos técnicos para poder realizarlas y posteriormente mantenerlas. Además cualquier variación en el proceso suponía modificar físicamente gran parte de las conexiones de los montajes, siendo necesario para ello un gran esfuerzo técnico y un mayor desembolso económico.

En la actualidad no se puede entender un proceso complejo de alto nivel desarrollado por técnicas cableadas. El ordenador y los Controladores Lógicos Programables han intervenido de forma considerable para que este tipo de instalaciones se hayan visto sustituidas por otras controladas de forma programada.

En el Ecuador se han presentado varias marcas y modelos de PLC's, siendo los más populares los de la marca Siemens, abarcando gran parte de la industria automotriz y en área de la domótica; destacando el PLC Siemens S7-1200 por su tarea de automatización de alta precisión.

2.4.1 Introducción al PLC S7-1200

En 2012, Siemens en su manual técnico destaca que el controlador S7-1200 ofrece la flexibilidad y potencia necesarias para controlar una gran variedad de dispositivos para las distintas necesidades de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones.

La CPU incorpora un microprocesador, una fuente de alimentación integrada, circuitos de entrada y salida, PROFINET integrado, E/S de control de movimiento de alta velocidad y entradas analógicas incorporadas, todo ello en una carcasa compacta, conformando así un potente controlador. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de conteo y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

La CPU incorpora un puerto PROFINET para la comunicación en una red PROFINET. Hay disponibles módulos adicionales para la comunicación en redes PROFIBUS, GPRS, RS485 o RS232, como se muestra en la Figura 5.

PLC Siemens S7-1200

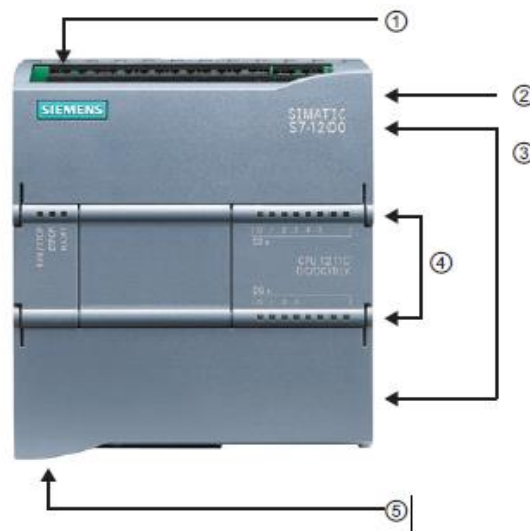


Figura 5. Módulo físico del PLC Siemens S7-1200. Fuente: (Siemens)

El CPU consta de:

1. Conector de corriente.
2. Ranura para Memory Card (debajo de la tapa superior).
3. Conectores extraíbles para el cableado de usuario (detrás de las tapas).
4. LEDs de estado para las E/S integradas.
5. Conector PROFINET (en el lado inferior de la CPU).

La familia S7-1200 ofrece diversos módulos y placas de conexión para ampliar las capacidades de la CPU con E/S adicionales y otros protocolos de comunicación.

Los PLC's operan de manera secuencial y cíclica, es decir, una vez finalizado el recorrido completo de un programa, comienza a ejecutar su primera instrucción.

Los elementos que contiene un PLC son:

1. Unidad Central de proceso.
2. Módulos de entrada.
3. Módulos de salida.
4. Fuente de Alimentación.
5. Dispositivos periféricos.
6. Interfaces.

La unidad central es el “cerebro” del PLC. Este toma las decisiones relacionadas al control de la máquina o proceso. Durante su operación, el CPU recibe entradas de diferentes dispositivos de censado, ejecuta decisiones lógicas, basadas en un programa almacenado en la memoria, y controla los dispositivos de salida de acuerdo al resultado de la lógica programada.

Los módulos de entradas y salidas son la sección del PLC en donde sensores y actuadores son conectados y a través de los cuales el PLC monitorea y controla el proceso.

La fuente de alimentación convierte altos voltajes de corriente de línea (115V 230V AC) a bajos voltajes (5V, 15V, 24V DC) requeridos por el CPU y los módulos de entradas y salidas.

2.4.2 Software de programación TIA PORTAL

En 2012, Siemens en su manual señala que TIA PORTAL ofrece un entorno confortable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como controladores y dispositivos HMI. Para poder encontrar la información necesaria, TIA PORTAL ofrece un completo sistema de ayuda en pantalla.

TIA PORTAL proporciona lenguajes de programación estándar, que permiten desarrollar de forma cómoda y eficiente el programa de control.

- **KOP (esquema de contactos):** es un lenguaje de programación gráfico. Su representación se basa en esquemas de circuitos.
- **FUP (diagrama de funciones):** es un lenguaje de programación que se basa en los símbolos lógicos gráficos empleados en el álgebra booleana.
- **SCL (lenguaje de control estructurado):** es un lenguaje de programación de alto nivel basado en texto.

Al crear un bloque lógico, se debe seleccionar el lenguaje de programación que empleará dicho bloque. El programa de usuario puede emplear bloques lógicos creados con cualquiera de los lenguajes de programación.

2.4.3 Módulos S7-1200

La familia S7-1200 ofrece diversos módulos y placas de conexión para ampliar capacidades de la CPU con E/S adicionales y otros protocolos de comunicación.

- **Módulos de señales:** las mayores CPU admiten la conexión de hasta ocho Módulos de Señales, ampliando así las posibilidades de utilizar E/S digitales o analógicas adicionales, como se muestra en la Figura 6.

PLC Siemens S7-1200



Figura 6. Módulo de señales analógicas para el PLC Siemens S7-1200. Fuente: (Siemens)

Las siguientes tablas muestran las características de los módulos de señal digital y analógica respectivamente.

Tabla 3.

Módulos de señales digitales

Tipo	Sólo Entradas	Sólo Salidas	Combinación de entradas y salidas
SB digital 1	Entrada: 4 x 24 V DC, 200 kHz Entrada: 4 x 5 V DC, 200 kHz	Salida: 4 x 24 V DC, 200 kHz Salida: 4 x 5 V DC, 200 kHz	Entrada: 2 x 24 V DC / Salida: 2 x 24 V DC Entrada: 2 x 24 V DC / Salida: 2 x 24 V DC, 200 kHz Entrada: 2 x 5 V DC / Salida: 2 x 5 V DC, 200 kHz
SM digital 1	Entrada: 8 x 24 V DC	Salida: 8 x 24 V DC 8 salidas de relé 8 salidas de relé (conmutador)	Entrada: 8 x 24 V DC / Salida: 8 x 24 V DC Entrada: 8 x 24 V DC / 8 salidas de relé 8 x 120/230VAC In / 8 x salidas de relé
	Entrada: 16 x 24 V DC	Salida: 16 x 24 V DC 16 salidas de relé	Entrada: 16 x 24 V DC / Salida: 16 x 24 V DC Entrada: 16 x 24 V DC / 16 salidas de relé

Nota: Características del módulo de señales digitales. Fuente: (Siemens)
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Tabla 4.
Módulos de señales analógicas

Tipo	Sólo Entradas	Sólo Salidas	Combinación de entradas y salidas
SB analógica	1 entrada analógica de 12 bits. RTD de 16 bits. 1 termopar de 16 bits.	1 salida analógica.	-
SM analógico	4 entradas analógicas. 4 entradas analógicas de 16 bits. 8 entradas analógicas. Termopar: – 4 TC de 16 bits – 8 TC de 16 bits RTD: – 4 RTD de 16 bits – 8 RTD de 16 bits	2 salidas analógicas. 4 salidas analógicas.	4 entradas analógicas. 2 salidas Analógicas.

Nota: Características del módulo de señales análogas. Fuente: (Siemens)
Elaborado por: John Jairo Sánchez y Fernando Barrera.

- **Módulos de comunicación:** toda CPU SIMATIC S7-1200 puede ampliarse hasta con 3 Módulos de Comunicación.

Los módulos de comunicación RS485 y RS232 son aptos para conexiones punto a punto en serie, basadas en caracteres, como se muestra en la Figura 7.

Esta comunicación se programa y configura con sencillas instrucciones, o bien con las funciones de librerías para protocolo maestro y esclavo USS Drive y Modbus RTU, que están incluidas en el sistema de ingeniería SIMATIC STEP 7 Basic.

PLC Siemens S7-1200



Figura 7. Módulos de comunicación para el PLC Siemens S7-1200. Fuente: (Siemens)

2.4.4 Comunicación industrial

En 2012, Siemens explica que la comunicación industrial con productos y sistemas de Siemens aporta una mayor eficiencia en todos los ámbitos de la empresa. Esto es así porque sus componentes de primera categoría basados en estándares probados, como por ejemplo SIMATIC NET, permiten desarrollar redes de datos homogéneos y potentes que respondan también a las necesidades futuras.

Las Comunicaciones Industriales son el área de la tecnología que estudia la transmisión de información entre circuitos y sistemas electrónicos utilizados para llevar a cabo tareas de control y gestión del ciclo de vida de los productos industriales.

Para cada uno de los modos de comunicación existe su correspondiente interfaz que independiza a las capas superiores del tipo de red utilizada, entre los cuales se puede resaltar la interface Profinet que utiliza el PLC Siemens S7-1200.

- **Interfaz profinet integrada:** el nuevo Simatic S7-1200 dispone de una interfaz Profinet integrada que garantiza una comunicación perfecta con el sistema de ingeniería Simatic Step 7 Basic Integrado. La interfaz Profinet permite la programación y la comunicación con los paneles de la gama

Simatic HMI Basic Panels para la visualización, con controladores adicionales para la comunicación de CPU a CPU y con equipos de otros fabricantes para ampliar las posibilidades de integración mediante protocolos abiertos Ethernet.

2.5 Programación de PLC'S

En 2012, Siemens en su manual técnico explica que se puede definir un programa como un conjunto de instrucciones, órdenes y símbolos reconocibles por el PLC, a través de su unidad de programación, que le permiten ejecutar una secuencia de control deseada. El Lenguaje de Programación en cambio, permite al usuario ingresar un programa de control en la memoria del PLC, usando una sintaxis establecida.

2.5.1 Programación TIA PORTAL

El PLC Siemens utiliza el programa TIA PORTAL estándar de SIMATIC para configurar sus controladores en lenguaje KOP (Esquema de contactos), FUP (Diagrama de funciones) o SCL (Lenguaje de control estructurado).

El KOP o Ladder, también denominado lenguaje de contactos o de escalera, es un lenguaje de programación gráfico muy popular dentro de los Controladores Lógicos Programables (PLC), debido a que está basado en los esquemas eléctricos de control clásicos. De este modo, con los conocimientos que todo técnico eléctrico posee, es muy fácil adaptarse a la programación en este tipo de lenguaje, es por eso que el lenguaje KOP o Ladder es el más utilizado en las industrias y es un lenguaje fácil manejo.

La representación del lenguaje de programación gráfico KOP (esquema de contactos) es similar a la de los esquemas de circuitos. Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, se agrupan en segmentos. Uno o varios segmentos constituyen el área de instrucciones de un bloque lógico, como se muestra en la Figura 8. Las operaciones que realiza el PLC pueden clasificarse en operaciones con bits, comparación, conversión, aritméticas, de transferencia, temporización, conteo, etc.

TIA PORTAL

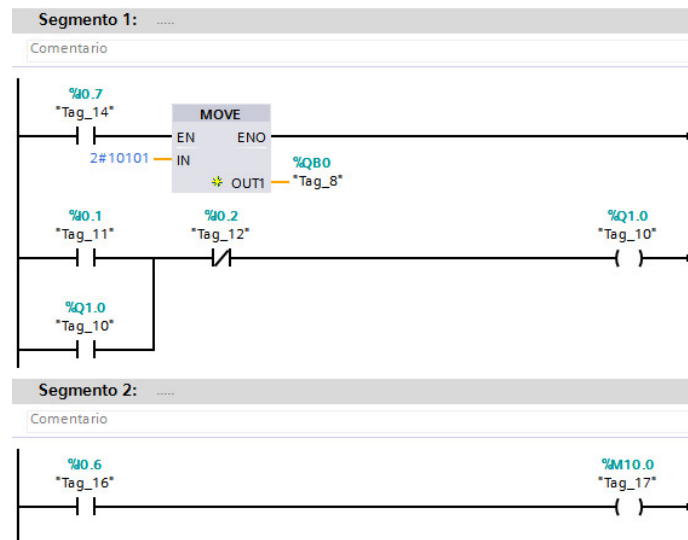


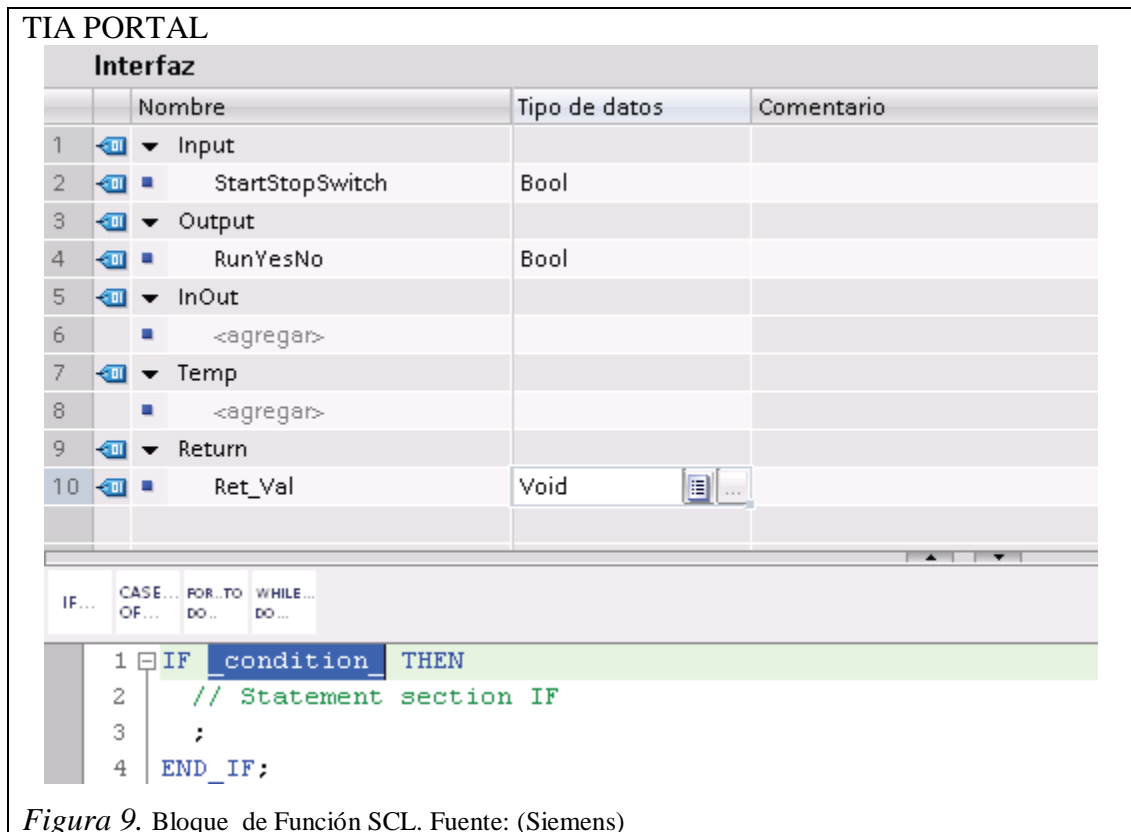
Figura 8. Ejemplo de segmento en el programa TIA PORTAL. Fuente: (Siemens)

El lenguaje de control estructurado SCL es un lenguaje de programación de alto nivel basado en PASCAL para las CPU de SIMATIC S7. SCL soporta la estructura de bloques de STEP 7. También es posible incluir bloques de programa escritos en SCL con bloques de programa escritos en KOP y FUP.

El código SCL para la instrucción se introduce directamente en la sección de código; y en la sección del bloque lógico SCL se pueden declarar los siguientes tipos de parámetros: (como se muestra en la Figura 9)

- **Input, Output, InOut y Ret_Val (corresponde a un FC):** estos parámetros definen las variables de entrada, las variables de salida y el valor de retorno del bloque lógico. El nombre de la variable introducida en este punto se emplea de forma local durante la ejecución del bloque lógico. Normalmente, no se emplea el nombre de variables globales en la tabla de variables.
- **Estáticos (solo FBs):** las variables estáticas se utilizan para almacenar resultados intermedios estáticos en el bloque de datos instancia. Los datos estáticos se retienen hasta que se sobrescriben, lo cual puede ocurrir después de varios ciclos. Los nombres de los bloques, los cuales se llaman en este bloque lógico como multiinstancia, también se almacenan en los datos locales estáticos.
- **Temp:** estos parámetros son variables temporales que se emplean durante la ejecución del bloque lógico.

Si se llama el bloque lógico SCL desde otro bloque lógico, los parámetros del bloque lógico SCL aparecen como entradas o salidas.



2.6 Algoritmos de control moderno

En la actualidad se tiene sistemas muy avanzados que pueden emular ciertas características humanas, en donde los métodos de la inteligencia artificial tienen un gran auge para el funcionamiento de varios sistemas, entre ellos los equipos electrodomésticos como lavadoras, hornos de microondas, cámaras de video, e inclusive sistemas de transporte. Los métodos destacados son la lógica difusa, las redes neuronales, los sistemas neuro-difusos y los algoritmos genéticos; entre los cuales se destaca los sistemas difusos, porque tienen la capacidad de emular la forma de la inferencia humana y además pueden almacenar la experiencia en forma lingüística.

2.6.1 Lógica difusa

Los sistemas basados en lógica difusa emulan la manera en que el cerebro razona o piensa, y en donde las aplicaciones se realizan en áreas multidisciplinarias que van desde la evolución tecnológica de los electrodomésticos, hasta programas computacionales para tomar decisiones y se han extendido a diversas áreas específicas como las cámaras de video, seguridad en base al monitoreo de la actividad volcánica, a través de controladores difusos para la calidad del agua, sistemas de operación automática de trenes, reactores nucleares y las transmisiones de automóviles, por mencionar diversos ejemplos.

2.6.1.1 Fundamentos de lógica difusa

En 2010, Ponce en su obra indica que la lógica difusa es un conjunto de principios matemáticos basados en grados de membresía o pertenencia, cuya función es modelar información. Este modelado se hace con base en reglas lingüísticas que aproximan una función mediante la relación de entradas y salidas del sistema. Esta lógica presenta rangos de membresía dentro de un intervalo entre 0 y 1, a diferencia de la lógica convencional, en la que el rango se limita a dos valores: el cero o el uno.

El ser humano muestra dificultad para tomar decisiones cuando se tiene información imprecisa. La lógica difusa fue creada para emular la lógica humana y tomar decisiones acertadas a pesar de la información.

2.6.1.2 Conjuntos difusos

En 2007, Molina menciona que en los conjuntos clásicos algo está incluido completamente en él o no lo está en absoluto, donde puede describirse asignando un 1 a todos los elementos incluidos en el conjunto y un 0 a los no incluidos. A la función que asigna estos valores la denominaremos función de inclusión o pertenencia.

Se dice que un conjunto difuso está normalizado si el máximo de su función de inclusión es 1; obviamente, un conjunto difuso puede normalizarse multiplicando su función de inclusión por un coeficiente fijo para que sea de tipo normalizado.

Cuando se habla de conjuntos nítidos, la variable típica a usar es la X . En conjuntos difusos la función de pertenencia que se utiliza es la μ . Ésta toma los valores entre cero (0) y uno (1); la forma de representación de los conjuntos difusos puede ser de dos maneras: de forma continua o discreta.

Un conjunto difuso se escribe con una tilde arriba del nombre del conjunto:

$$\tilde{A} = \{a, b, c\}$$

Ésta se utiliza para diferenciarlos de los conjuntos nítidos.

2.6.1.3 Funciones de membresía

Para la representación de los grados de pertenencia de cada uno de los elementos que conforman el conjunto difuso, se debe extraer los datos de los fenómenos que se va a representar y con ellos definir la forma de la función de membresía.

Existen funciones de membresía convencionales y no convencionales que permiten realizar un mapeo de un universo nítido a un universo difuso. Entre las funciones de membresía convencionales se tienen las siguientes:

- **Función de saturación:** es la más sencilla de ellas. Tiene un valor de 0 hasta cierto punto y después crece con pendiente constante hasta alcanzar el valor de 1, en donde se estaciona.
- **Función hombro:** que es, por decirlo de alguna manera, la contraparte de la función saturación. En este tipo de funciones se inicia en un valor unitario y se descende con constante pendiente hasta alcanzar el valor de 0.
- **Función triangular:** su forma, como su nombre lo indica, consta de una parte de pendiente positiva constante hasta alcanzar la

unidad, y una vez que lo ha logrado desciende de manera uniforme. Es muy adecuada para definir situaciones en las que se tiene un valor óptimo central, el cual se va perdiendo conforme uno se aleja de él. Un ejemplo de esta situación es la temperatura corporal, que tiene un valor óptimo de 37° centígrados, pero que por debajo de 35° por encima de 39° se considera peligrosa, es decir, el nivel de pertenencia al conjunto de temperaturas seguras en el cuerpo humano es 0.

- **Función trapecio o Pi:** una generalización de la función triangular es la función trapecio o función Pi. En el caso de esta función de membresía, no sólo se tiene un valor para el cual la pertenencia es unitaria. La forma de esta función es muy utilizada, ya que como se mencionó se emplea cuando hay un rango de valores óptimos, alrededor de los cuales las condiciones no son adecuadas. Un buen ejemplo de esto es la iluminación de un salón de clases. Existe un rango en el cual la iluminación es agradable para las personas, pero por debajo de dicho rango la luz no es suficiente para leer el pizarrón, y por encima de él es molesto para la vista de los estudiantes.
- **Función “S” o sigmoidal:** finalmente tenemos la función “S”. La forma de esta función es similar a la de saturación. Sin embargo, como su nombre lo indica, el segmento de subida no es una línea recta, sino una curva de segundo orden, la cual cambia la concavidad en un punto dado, y una vez que llega a 1 se mantiene en este valor. (Ponce, 2010, pág. 49).

Las funciones características más comúnmente utilizadas por su simplicidad matemática y su manejabilidad son las anteriormente mencionadas; pero siendo la función triangular la forma más utilizada para una función de membresía.

2.6.1.4 Operaciones difusas

Al aplicar un operador sobre un solo conjunto difuso se obtiene otro conjunto difuso; de la misma manera al combinar dos o más subconjuntos mediante alguna operación, se obtendrá otro conjunto.

Sean los subconjuntos difusos identificados por las etiquetas A y B, asociadas a una variable lingüística x, para ellos pueden definirse tres operaciones básicas: complemento, unión e intersección. Estas operaciones básicas pueden expresarse de la siguiente manera en términos de las funciones de pertenencia de los conjuntos difusos A y B:

- **Complemento:** para un conjunto difuso A definido sobre un Universo de discurso U, y cuya función de pertenencia es $\mu_A(x)$; el resultado de efectuar la operación de complemento (que en lógica binaria es el equivalente de la operación NOT), es un nuevo conjunto difuso \bar{A} definido sobre el mismo Universo, y con función de pertenencia $\mu_{\bar{A}}(x)$; dada por la Ecuación 1.

Ecuación 1. Operación de complemento

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

- **Unión:** para dos conjuntos difusos A y B definidos sobre el mismo Universo de discurso U, y cuya función de pertenencia es $\mu_A(x)$ y $\mu_B(x)$; el resultado de efectuar la operación de unión entre estos dos conjuntos (que en lógica binaria es el equivalente de la operación OR), es un nuevo conjunto difuso $A \cup B$ definido sobre el mismo Universo, y con función de pertenencia $\mu_{A \cup B}(x)$; dada por la Ecuación 2.

Ecuación 2. Operación de unión

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)]$$

- **Intersección:** para dos conjuntos difusos A y B definidos sobre el mismo Universo de discurso U , y cuya función de pertenencia es $\mu_A(x)$ y $\mu_B(x)$; el resultado de efectuar la operación de intersección entre estos dos conjuntos (que en lógica binaria es el equivalente de la operación AND), es un nuevo conjunto difuso $A \cap B$ definido sobre el mismo Universo, y con función de pertenencia $\mu_{A \cap B}(x)$; dada por la Ecuación 3.

Ecuación 3. Operación de intersección

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]. \text{ (Molina, 2010, pág. 257).}$$

Es importante resaltar que el funcionamiento de estas operaciones básicas coincide con el de las correspondientes a las de la teoría clásica de conjuntos; de hecho, la teoría de conjuntos difusos se reduce a la teoría clásica si reducimos la incertidumbre a 0, y admitimos sólo los valores 0 y 1 para las funciones de pertenencia a un conjunto (0, no pertenece; 1, pertenece).

2.6.1.5 Inferencia difusa

En 2007, Molina indica que la lógica difusa se ocupa del razonamiento formal con proposiciones, pero a diferencia de ésta, los valores de las proposiciones pueden tomar valores intermedios entre verdadero y falso.

El objeto de la lógica difusa es proporcionar un soporte formal al razonamiento basado en el lenguaje natural, que se caracteriza por tratarse de un razonamiento de tipo aproximado, que hace uso de unas proposiciones que a su vez expresan información de carácter impreciso.

2.6.1.6 Implicación difusa

En 2007, Molina enseña que el operador lógico de implicación \Rightarrow , combina dos proposiciones con la expresión SI... Entonces... (If... Then...), y que es el fundamento de las inferencias realizadas en sistemas de lógica difusa.

El operador lógico de implicación \Rightarrow , permite encontrar un camino matemático para evaluar proposiciones como las siguientes: “Si los ingresos del cliente son bajos Entonces su capacidad de endeudamiento es poca”. A la relación entre dos proposiciones a través del operador lógico de implicación, se las llama también reglas, las mismas que servirán en lo posterior para definir la base de reglas de los sistemas de control difuso.

En términos de lógica difusa se puede decir que la proposición “Si u es A , entonces v es B ”, donde $u \in U$ y $v \in V$, tiene asociada una función de membresía $\mu_{A \rightarrow B}(x, y)$ que toma valores en el intervalo $[0,1]$. Es decir, cada una de las reglas o proposiciones if-then es a su vez un conjunto difuso con su función de membresía, que mide el grado de verdad de la relación de implicación entre x e y . }

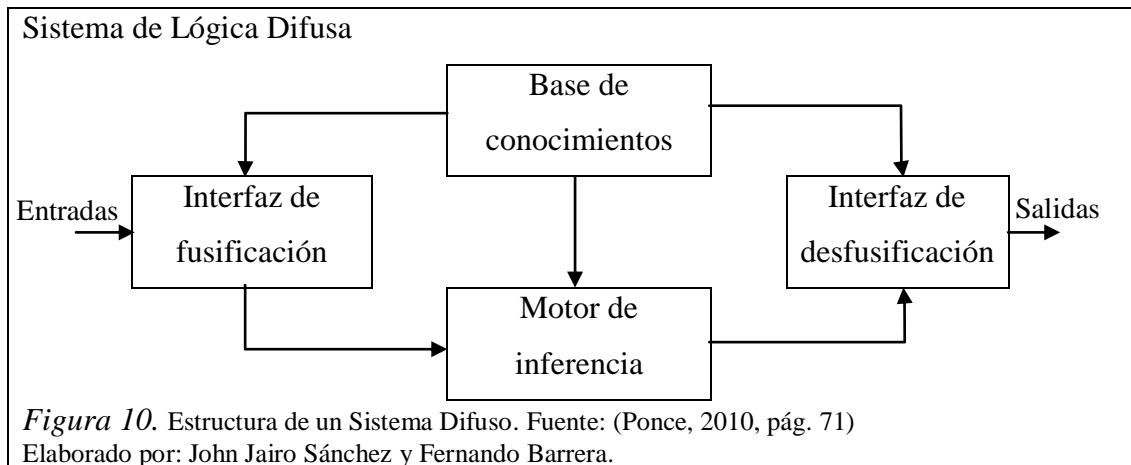
2.6.1.7 Variables y términos lingüísticos

En 2007, Molina denomina variable lingüística a aquella que puede tomar por valor términos del lenguaje natural, como mucho, poco, positivo, negativo, etc., que son las palabras que desempeñan el papel de etiquetas en un conjunto difuso. Aunque el objetivo principal de este concepto es expresar de manera formal el hecho de que pueden asignarse como valor de una variable palabras tomadas del lenguaje natural, no obstante a una variable lingüística podrán asignarse también valores numéricos. Así, en una expresión como la temperatura es fría, la variable temperatura debe ser entendida como una variable lingüística, pues se le asigna como valor el conjunto difuso fría, pero además esta variable puede también tomar valores numéricos como la temperatura es 4°C.

2.6.1.8 Modelos difusos lingüísticos

En 2010, Ponce indica que los modelos se basan en un conjunto de reglas heurísticas donde las variables lingüísticas de las entradas y salidas se representan por conjuntos difusos.

La estructura básica de un Sistema de Lógica Difusa se muestra en la Figura 10, el sistema recibe varias entradas numéricas y entrega varias salidas numéricas.



2.6.1.9 Fusificación

En 2010, Ponce indica que en esta etapa se transforma las variables de entrada del modelo (u) en variables difusas, donde a cada variable de entrada se le asigna un grado de pertenencia a cada uno de los conjuntos difusos que se han considerado, mediante las funciones de membresía asociadas a estos conjuntos difusos.

2.6.1.10 Defusificación

En 2007, Molina indica que en este bloque a partir del conjunto difuso obtenido en el bloque de inferencia y mediante métodos matemáticos de defusificación, se obtiene el valor concreto de la variable de salida.

2.6.2 Controlador difuso

El uso de lógica difusa en los sistemas de control ha sido motivado por el deseo satisfacer uno o más de los siguientes objetivos:

1. Mejorar la robustez que se obtiene con los métodos clásicos de control lineales.
2. Diseño de control simplificado para modelos complejos.

3. También se obtiene una implementación simplificada.
4. Autonomía.
5. Adaptabilidad.
6. En el caso del control difuso, no es necesario un modelo matemático de la planta. (Ponce, 2010, pág. 36)

Una de las bondades de los sistemas difusos es que no necesitan el modelado de una planta ni la ejecución de la identificación en tiempo real. La esencia del control difuso es que convierte la estrategia de control lingüístico, la cual se basa en el conocimiento de un experto, en una estrategia de control automático.

Es muy importante tener presente el hecho de que los controles difusos se basan en reglas de control empíricas. Para este punto es bien sabido que el modelado difuso es un método para describir las características de un sistema usando reglas de inferencia difusas. Se hace especial referencia en el manejo de las reglas de control difuso, comúnmente extraídas de un experto, y de la sintonización de éstas.

2.6.2.1 Estructura de un controlador difuso

La anatomía básica de un controlador difuso consta de tres partes:

- **Reglas:** estas son reglas que dictan la acción de control que se va a tomar. Éstas se derivan de un experto. Dichas reglas tienen la estructura de relaciones. La lógica difusa se basa en relaciones, las cuales se determinan por medio de cálculo de reglas “SI-ENTONCES” (con las cuales se puede modelar aspectos cualitativos del conocimiento humano, así como los procesos de razonamiento sin la necesidad de un análisis cuantitativo de precisión).

Un ejemplo de una regla sería:

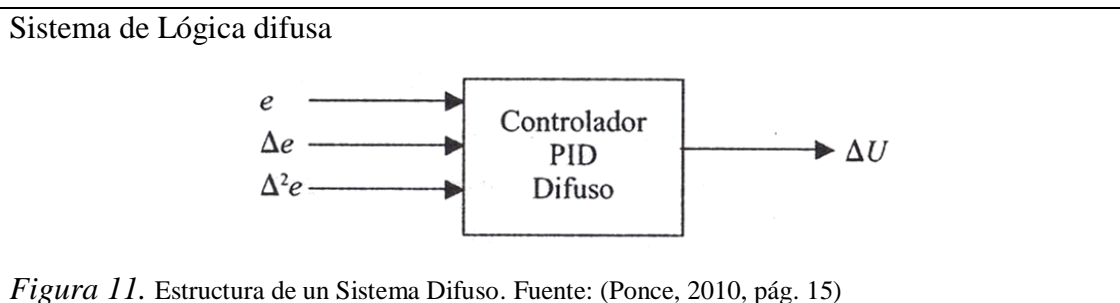
Si la temperatura es alta entonces se debe de encender el ventilador.

- **Difusificador:** es el nexo entre las entradas reales y difusas. Todas las entradas necesitan ser mapeadas a una forma en que las reglas puedan utilizarlas.

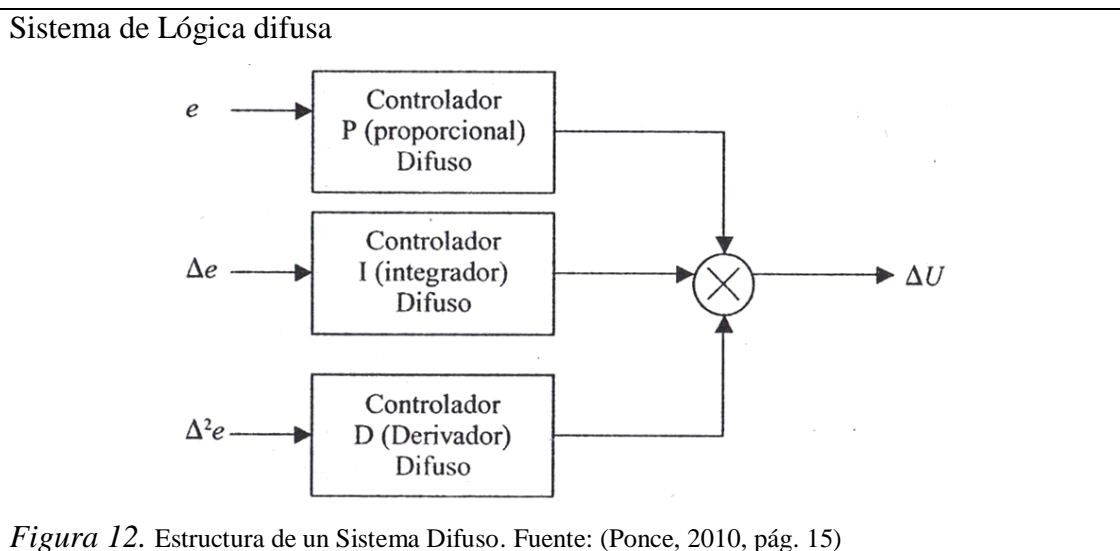
- **Desfusificador:** toma un valor difuso de las reglas y genera una salida real. (Ponce, 2010, pág. 34).

2.6.2.2 Controlador PID difuso

En 2010, Ponce menciona que existen dos maneras de armar un controlador PID difuso, la primera propuesta es colocando en un solo bloque las variables, error, la derivada del error y la integral del error, como se muestra en la Figura 11.



La segunda propuesta es creando cada variable por separado, sumando los bloques y obtener así el PID, como se muestra en la Figura 12.



2.7 Software de aplicación de algoritmos modernos

Existe varios softwares de aplicación para algoritmos modernos basados esencialmente en el manejo numérico de la información, entre los cuales se destaca el software Matlab el cual se basa en programación por matrices. A diferencia de otros programas como Mathematica o MathCad, no permite trabajar con expresiones matemáticas simbólicas (salvo librerías que lo simulan). Esto que en un principio parece una desventaja, termina siendo una de las mayores virtudes de Matlab.

2.7.1 Software Matlab

En 2012, Reyes en su obra literaria “Matlab Aplicado a Robótica y Mecatrónica” menciona que Matlab es un lenguaje de programación matemático de alto nivel integrado con entorno gráfico amigable, visualización de datos, funciones, gráficas 2D y 3D, procesamiento de imágenes, video, computación numérica para desarrollar algorítmicas matemáticos con aplicaciones de ingeniería y ciencias exactas. Particularmente, en ingeniería es una herramienta muy poderosa para realizar aplicaciones en mecatrónica, robótica, control y automatización.

Matlab es un acrónimo que proviene de matrix laboratory (laboratorio matricial) creado por el profesor y matemático Cleve Moler en 1970. La primera versión de Matlab fue escrita en lenguaje fortran la cual contempló la idea de emplear subrutinas para los cursos de álgebra lineal y análisis numérico de los paquetes Linpack y Eispack; posteriormente se desarrolló software de matrices para acceder a esos paquetes sin necesidad de usar programas en fortran. Actualmente, el lenguaje de programación de Matlab proporciona un sencillo acceso a algoritmos numéricos que incluye matrices.

Matlab dispone de dos herramientas adicionales que expanden sus prestaciones: plataforma de simulación multidominio Simulink y Guide editor de interfaces de usuario. (Se recomienda visitar el siguiente link: www.mathworks.com/productos/matlab).

2.7.2 Componentes de Matlab

En 2012, Reyes indica que el ambiente de programación de Matlab es amigable al usuario, y está compuesto por una interface gráfica con varias herramientas distribuidas en ventanas que permiten programar, revisar, analizar, registrar datos, utilizar funciones, historial de comandos y desarrollar diversas aplicaciones.

La pantalla principal Matlab se presenta en la Figura 13, la cual contiene la interface gráfica de usuario con varias ventanas como la de comandos (Command Window fx >>), manejo de archivos, espacio de trabajo (Workspace), historial de comandos (Command History), directorio de trabajo y aplicaciones como Simulink.

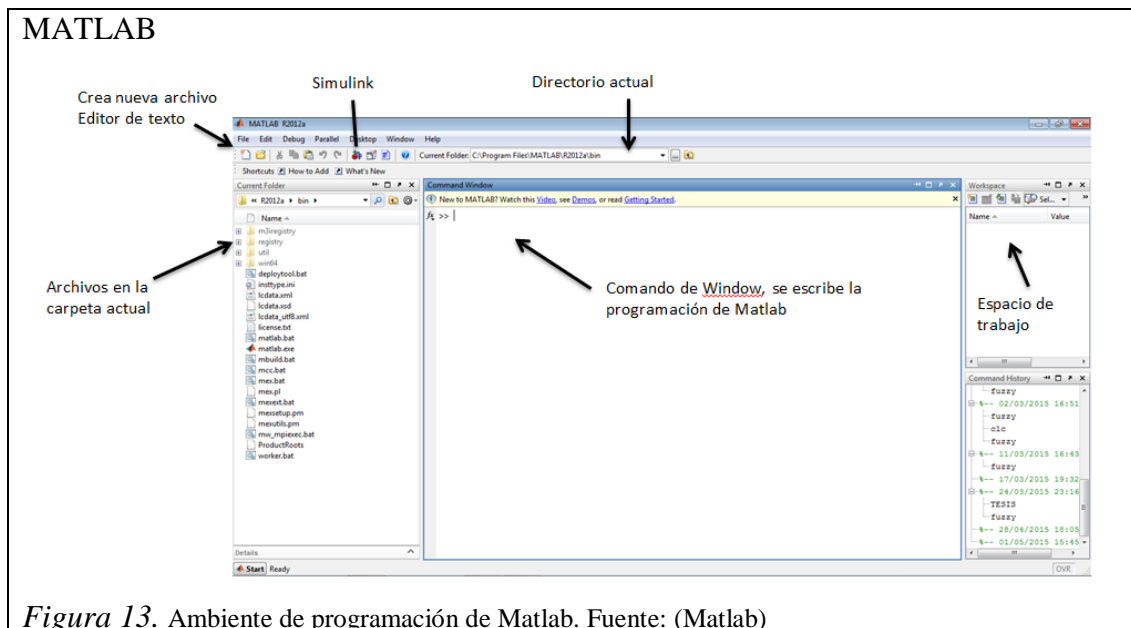


Figura 13. Ambiente de programación de Matlab. Fuente: (Matlab)

Es recomendable personalizar el ambiente de programación de Matlab para una fácil interacción con manejos de archivos, datos y herramientas de programación.

2.7.3 Herramientas de Matlab

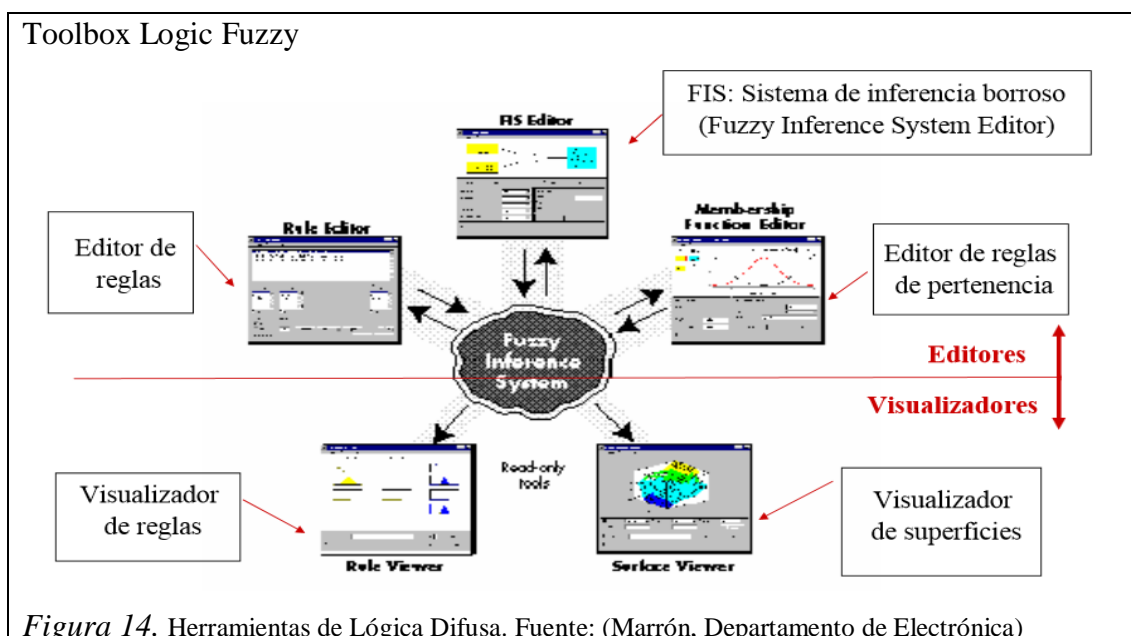
En 2012, Reyes explica que un sistema difuso puede simularse e implementarse directamente usando un lenguaje de programación como el que incorpora Matlab, la misma que contiene herramientas para simular procesos de control, comunicación, lógica difusa de redes neuronales, entre otros. Destacando la lógica difusa, la cual

utiliza un toolbox ya construido que permite definir e implementar distintos sistemas difusos; por este motivo es necesario comenzar introduciendo el manejo de esta herramienta que permite analizar el comportamiento de un sistema difuso y su utilización como regulador en un bucle de control.

2.7.4 Toolbox Fuzzy

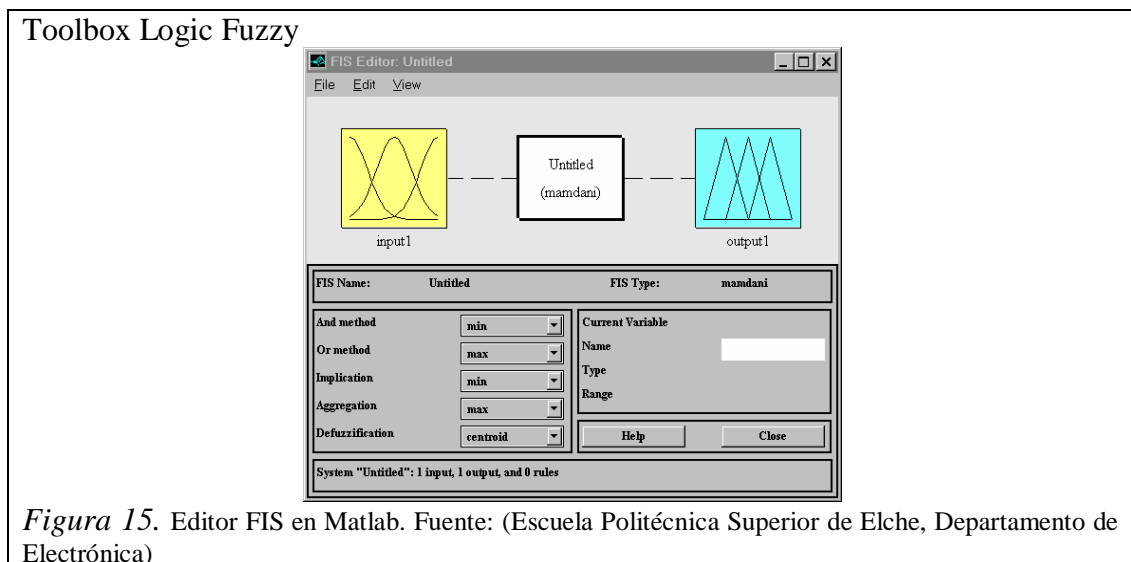
Tolbox Fuzzy ofrece funciones, aplicaciones, y un bloque de Simulink para el análisis, el diseño y los sistemas basados en lógica difusa. El programa le guía a través del proceso de diseño de los sistemas de inferencia difusos. Las funciones se proporcionan para los muchos métodos comunes, incluyendo clustering fuzzy y aprendizaje neuroborroso adaptativa. (Se recomienda visitar el siguiente link: <http://www.mathworks.com/products/fuzzy-logic/>)

Aunque es posible utilizar la caja de herramientas de la lógica difusa trabajando de la línea de comando, en general es mucho más fácil construir un sistema gráficamente. Hay cinco herramientas primarias del GUI, y observando sistemas difusos de la inferencia en la caja de herramientas de la lógica difusa: el sistema difuso de la inferencia o editor de FIS, el editor de las funciones de membrecía, el editor de las reglas, el visualizador de las reglas, y el visualizador de la superficie, como se muestra en la Figura 14.



2.7.5 El editor del sistema difuso

Para abrir la ventana de edición del sistema difuso, puede ejecutarse el comando fuzzy, si se quiere crear un sistema nuevo, o fuzzy (nombre_fis) si se quiere editar el sistema difuso definido en la matriz almacenada en la variable nombre_fis, o fuzzy ('nombre_fichero') si se quiere editar el sistema almacenado en el fichero nombre_fichero. Aparece la ventana, que se muestra en la Figura 15.



2.7.6 Editor de las funciones de membrecía

El editor de las funciones de membrecía comparte algunos rasgos con el Editor de FIS. De hecho, todas las cinco herramientas de GUI tienen opciones del menú similares, el estado en línea, ayuda y botones para cerrar.

El editor de las funciones de membrecía, es la herramienta que permite desplegar y editar todas las funciones de membrecía asociadas con todas las variables de entrada y de salida para entrar al sistema de la inferencia Fuzzy, como se muestra en la Figura 16.

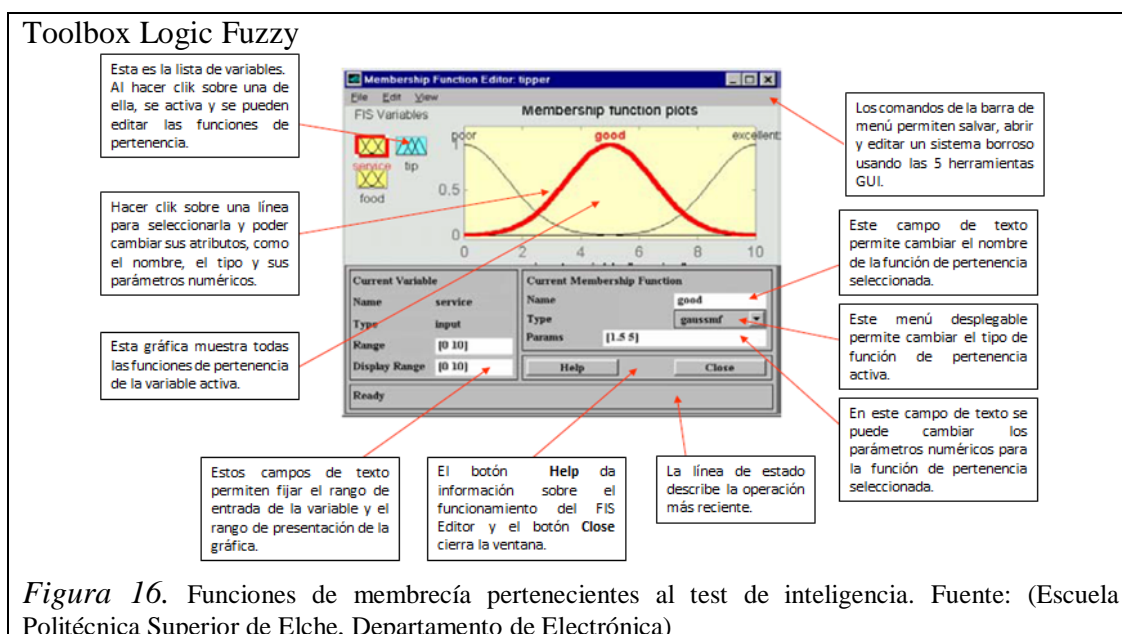


Figura 16. Funciones de membrecía pertenecientes al test de inteligencia. Fuente: (Escuela Politécnica Superior de Elche, Departamento de Electrónica)

2.7.7 Editor de las reglas

La interface gráfica del editor de reglas es bastante fácil para ir construyendo las reglas que se van a usar, como se observa la Figura 17. Basado en las descripciones de las variables de entrada y variables de salida definidas con el editor FIS, el editor de las reglas permite construir las declaraciones de las reglas automáticamente, pulsando el botón y seleccionando un ítem en cada variable de entrada en la caja, un ítem en cada caja de salida, y un ítem de conexión. Pueden cambiarse las reglas, pueden anularse, o pueden agregarse más, pulsando el botón apropiado. El editor de reglas también tiene algunos hitos familiares, similar a aquéllos en el editor FIS y el editor de funciones de membrecía, incluso la barra de menú y la barra de estado.

Toolbox Logic Fuzzy

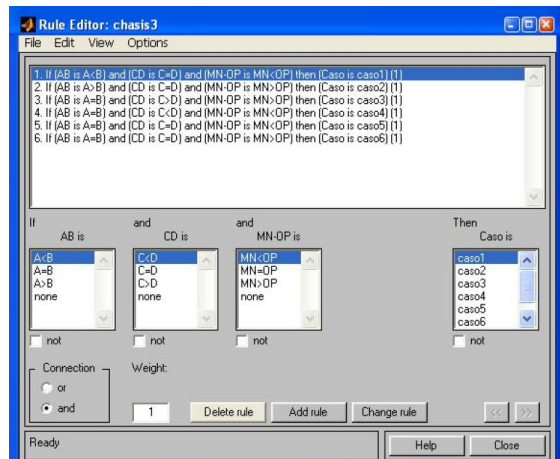


Figura 17. Reglas implementadas para el sistema exacto. Fuente: (Escuela Politécnica Superior de Elche, Departamento de Electrónica)

2.7.8 OPC Toolbox

OPC Toolbox, proporciona una conexión con servidores OPC DA y OPC HDA, que le da acceso a transferir datos históricos OPC directamente desde MATLAB y Simulink, como se muestra en la Figura 18. Usted puede leer, escribir y registrar datos OPC de dispositivos, tales como los sistemas de control distribuido, control de supervisión y adquisición de datos y controladores lógicos programables, que cumplen con el estándar OPC Fundación Data Access. (Se recomienda visitar el siguiente link: <http://www.mathworks.com/products/opc/>)

Toolbox Logic Fuzzy

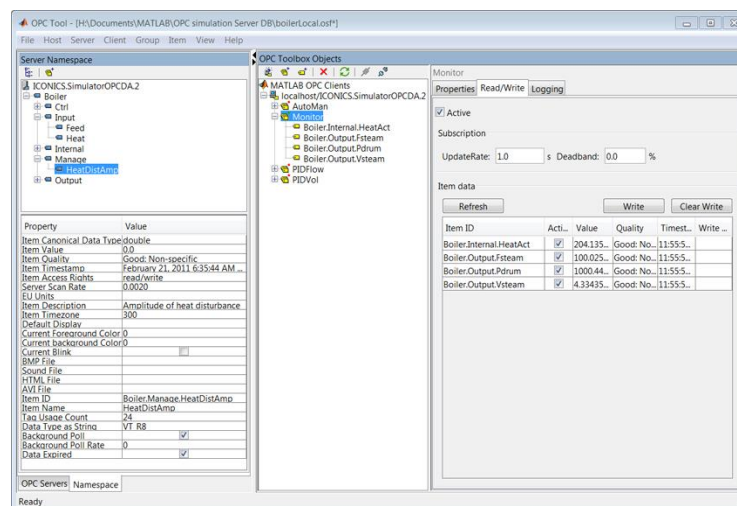


Figura 18. OPC Toolbox.
Fuente: (Mathworks)

CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN DE CONTROLADORES PID DIFUSOS

Este capítulo describe el diseño e implementación de los controladores PID difusos de las variables de nivel y temperatura para la plataforma de entrenamiento de procesos continuos (ver pág. 8), en el programa Matlab y en el PLC Siemens S7-1200.

Para el funcionamiento de la plataforma es necesario la elaboración de una interfaz gráfica de control, con el cual se podrá interactuar de forma directa con el prototipo, para cumplir con este fin se emplearon los siguientes programas:

- **MATLAB:** creación de algoritmos para control con el método PID-Difuso para las variables de nivel y temperatura.
- **KEPSERVEREX:** comunicación entre PLC y Matlab.
- **TIA PORTAL:** programación para el PLC.

Con el fin de diseñar e implementar los controladores PID Difusos, es necesario caracterizar la Plataforma de entrenamiento para control de procesos continuos.

3.1 Caracterización de la plataforma de entrenamiento para control de procesos continuos

La plataforma permite realizar el control de las variables de nivel y temperatura; ya que posee un sensor que toma la señal de salida y la compara con un punto de consigna (Set Point) dado, el controlador que recibe la diferencia entre el Set Point y la salida, la procesa para dar una señal adecuada al actuador, el cual aplica un mando a la plataforma y de este modo mantener el sistema en las condiciones deseables.

Esta plataforma es un sistema híbrido, el cual trabaja con un sistema neumático y eléctrico.

Con el afán de usar y dar un adecuado manejo a la plataforma de entrenamiento para control de procesos continuos, es beneficioso describir e identificar cada una de sus partes y componentes fundamentales.

3.1.1 Descripción de los componentes de la plataforma de entrenamiento para control de procesos continuos

La plataforma consta de varios componentes, como se muestra en la Figura 19.

Plataforma de entrenamiento para control de procesos continuos



Figura 19. Componentes de la plataforma de entrenamiento. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Para una correcta identificación de los componentes de la plataforma de entrenamiento se lo ha dividido en dos partes:

- Elementos de control.
- Elementos auxiliares.

3.1.1.1 Elementos de control

Cumple con la función de controlar el proceso, se detallan a continuación:

- **Controlador lógico programable - PLC siemens S7-1200:** es el dispositivo que se encarga de monitorear todas las señales del proceso de nivel y temperatura, es donde se podrá generar las comparaciones necesarias con el fin de que la acción de control sobre el sistema sea más eficaz, como se muestra en la Figura 20.

Controlador Lógico Programable - PLC



Figura 20. PLC Siemens S7-1200. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- **Sensor de nivel:** instrumento basado en la presión hidrostática, puesto que estos tipos de instrumentos son utilizados en la industria para tanques abiertos y para tanques cerrados. El instrumento seleccionado es el transmisor electrónico FOXBORO 823DP, como se muestra en la Figura 21.

Transmisor



Figura 21. Transmisor de nivel. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- **Sensor de temperatura RTD:** es un detector de temperatura resistivo, más conocido como Pt100; es decir que mientras aumenta la temperatura también aumenta la resistencia. Consiste en un alambre de platino que a 0 °C tiene 100 ohms y que al aumentar la temperatura aumenta su resistencia eléctrica, como se muestra en la Figura 22.

Sensor



Figura 22. Sensor de temperatura - RTD. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

El incremento de la resistencia no es lineal pero si creciente y característico del platino, de tal forma que mediante tablas es posible encontrar la temperatura exacta a la que corresponde.

- **Transductor electro neumático:** se activa por una señal eléctrica de 4-20 mA para controlar la servo válvula mediante presión de aire de 3-15 psi. Es de la marca Fisher modelo 546, las señales tanto de entrada como de salida

son estandarizadas, además no se necesita que la presión de entrada sea elevada, únicamente necesita 20 psi para funcionar, como se muestra en la Figura 23.

Transductor



Figura 23. Transductor electro neumático. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- **Servo válvula:** es de la marca Fisher, consiste en un diafragma con resorte que trabaja con presión de aire 3-15 psi; al aplicar una cierta presión sobre el diafragma, el resorte se comprime; de tal modo que el mecanismo empieza a moverse, como se muestra en la Figura 24.

Servo válvula



Figura 24. Servo válvula. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- **Bomba:** genera presión y caudal respectivamente en el fluido. La bomba es de 0.5 HP = $\frac{1}{2}$ HP, lo cual asegura que la bomba no va trabajar al 100% de la carga nominal y se extenderá la vida útil de la misma, como se muestra en la Figura 25.

Bomba

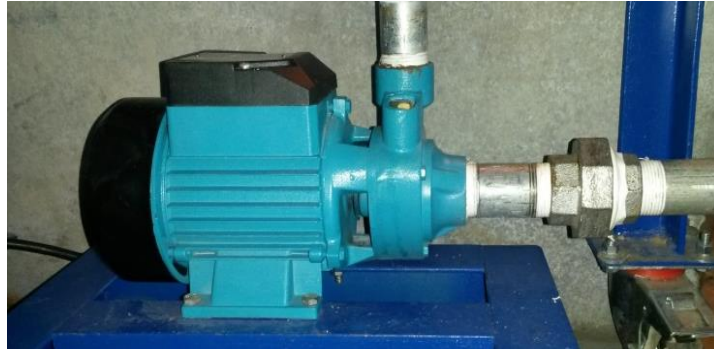


Figura 25. Bomba de agua. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- **Niquelinas:** la plataforma está provista de 2 calentadores de inmersión de diferente potencia uno es de 1500 W y la otra es de 700 W, los dos calentadores funcionan a 110V, como se muestra en la Figuras 26 y 27.

Niquelina 1



Figura 26. Niquelina de 700 W. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Niquelina 2



Figura 27. Niquelina de 1500 W. Fuente: (Plataforma de entrenamiento para control de procesos continuos)

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Para la activación de la niquelina de 1500W se utilizó la salida digital número 3 del PLC la cual activa un relé y este a su vez la niquelina.

Para la activación de la niquelina de 700 W se realizó la construcción de una tarjeta de control de fase.

3.1.1.2 Elementos auxiliares

La plataforma cuenta con varios elementos auxiliares los cuales son encargados de asegurar el óptimo funcionamiento de los elementos de control antes mencionados, además brindan las protecciones necesarias para evitar que los elementos de control se averíen.

A continuación se detallan todos los elementos y su función.

- **Electroválvula:** interviene principalmente en el proceso de temperatura, porque es el encargado de sacar el agua caliente para que ingrese agua fría y mantener la temperatura deseada. También se la puede utilizar en el proceso de nivel para simular un consumo adicional al que entrega la válvula manual.
- **Bomba Sumergible:** aumenta la presión de salida del agua desde el tanque de almacenamiento hacia la electroválvula lo cual garantiza que el flujo de agua de la electroválvula sea constante.
- **Módulo de Mantenimiento:** regula el suministro de aire, puesto que el transmisor electro-neumático para el control de la servo válvula debe ser alimentado con 20 PSI. Cuenta con una válvula reguladora de 0 a 25 PSI.
- **Computador:** posee programas utilitarios para programar al PLC y el monitor que permite visualizar el proceso de control.
- **Tarjeta del control de fase:** realiza un control proporcional de la niquelina de 700 W, funciona con una señal análoga de 1-5 VDC que es entregada a través de la salida análoga del PLC.
- **Sensor de nivel:** detecta el nivel de agua en el tanque cisterna, bajo la siguiente condición, si el agua de la cisterna es demasiado bajo no se podrá activar la bomba lo cual evita que la bomba pueda quemarse.

- **Switch:** para la comunicación entre el computador y el PLC se precisa de un switch de 8 puertos, los cuales están distribuidos de la siguiente manera:
 - Puerto 1: PLC.
 - Puerto 2: Face-plate RJ-45.
 - Puerto 3: Computador.
- **Luz piloto verde:** indica que el PLC está encendido.
- **Luz piloto amarilla:** indica que el tanque reservorio no tiene el nivel de agua requerido.
- **Paro de emergencia:** realiza el paro/bloqueo de control de la plataforma de entrenamiento, apagando los actuadores y sensores.
- **Mixer:** se lo utiliza en el proceso de temperatura para mezclar el agua y tener una temperatura más uniforme dentro del tanque la alimentación es a 110VAC, la potencia es de 250W.
- **Uninterrupted Power Supply (UPS):** la función principal del equipo UPS es dar tiempo de respaldo para que en caso de un corte energético se pueda guardar los datos y apagar correctamente el computador.
- **Face-plate RJ-45:** es un accesorio para el montaje de redes estructuradas, permite programar al PLC Siemens S7-1200 desde otro computador personal.

Las protecciones eléctricas están distribuidas de la siguiente manera:

- **Breaker (B1) 20A:** alimenta todos los tomacorrientes de la cabina de la plataforma.
- **Breaker (B2) 6A:** alimenta el Controlador Lógico Prográmale PLC, electroválvula, mixer y bomba sumergible.
- **Breaker (B3) 32A:** alimenta a las niquelinas de 1500 y 700 watios.
- **Breaker (B4) 6A:** alimenta Bomba de agua de ½ HP.
- **Fusibles:** protección de para el control a 24 VDC.
- **Contactores:** protección para elementos de alta corriente, empleado para la bomba, niquelinas y el mixer.
- **Relés auxiliares:** se instaló los relés auxiliares por seguridad puesto que en caso de que exista un corto circuito en los elementos que están siendo alimentados únicamente se averíe los relés auxiliares y el PLC no se vea afectado.

3.2 Diseño de los controladores PID difusos

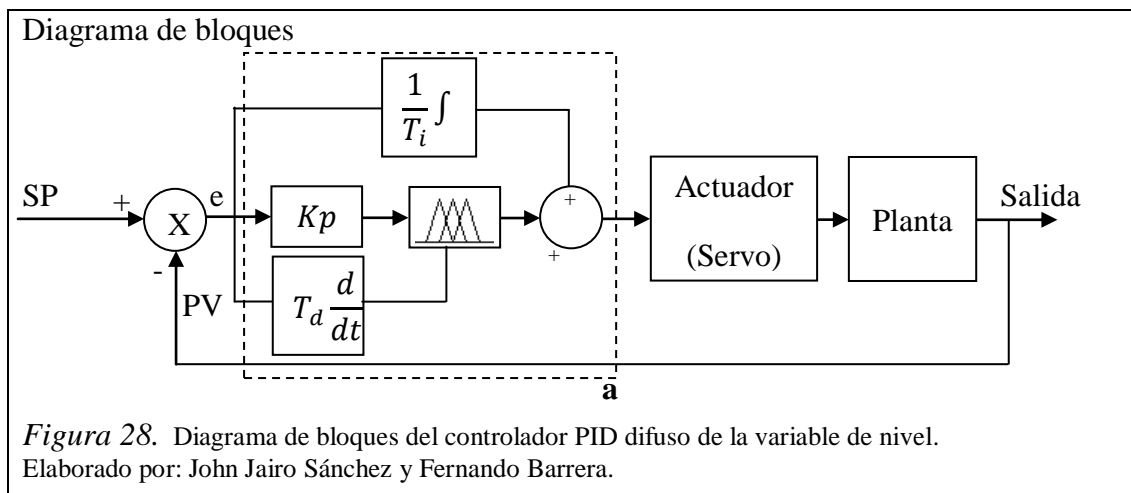
Se procede a realizar el diseño de los controladores PID difusos tanto para el programa Matlab como para el PLC Siemens S7-1200 basándose en la información obtenida.

3.2.1 Controladores PID difusos en Matlab

El fundamento principal es realizar el control interactivo en tiempo real del modelo de un proceso en Simulink a través de un sistema de control. El tipo de control digital utilizado es a base de computador personal, mediante el programa Matlab.

3.2.1.1 Controlador PID difuso de la variable de nivel

La Figura 28, muestra el diagrama de bloques empleado para realizar el controlador PID difuso de la variable de nivel.



Ponce, menciona que existen varias maneras de armar un controlador PID Difuso, se selecciona la propuesta que consiste en crear en un solo bloque las variables error y la derivada del error, y en otro bloque la integral del error; sumando los bloques y así obtener el PID Difuso, como se muestra en la Figura 28 sub-bloque a.

Para enlazar el Matlab con el PLC Siemens S7-1200, utilizar el programa KEPServerEX; trabajando el PLC como una tarjeta de adquisición de datos.

- **KEPServerEX**

KEPServerEX OPC, es un programa propietario de la empresa Kepware Technologies, el mismo que proporciona el intercambio de datos entre clientes OPC y Siemens S7-200, S7-300, S7-400 y S7-1200 PLC utilizando TCP / protocolo IP Ethernet, como se muestra en la Figura 29.

El OPC (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

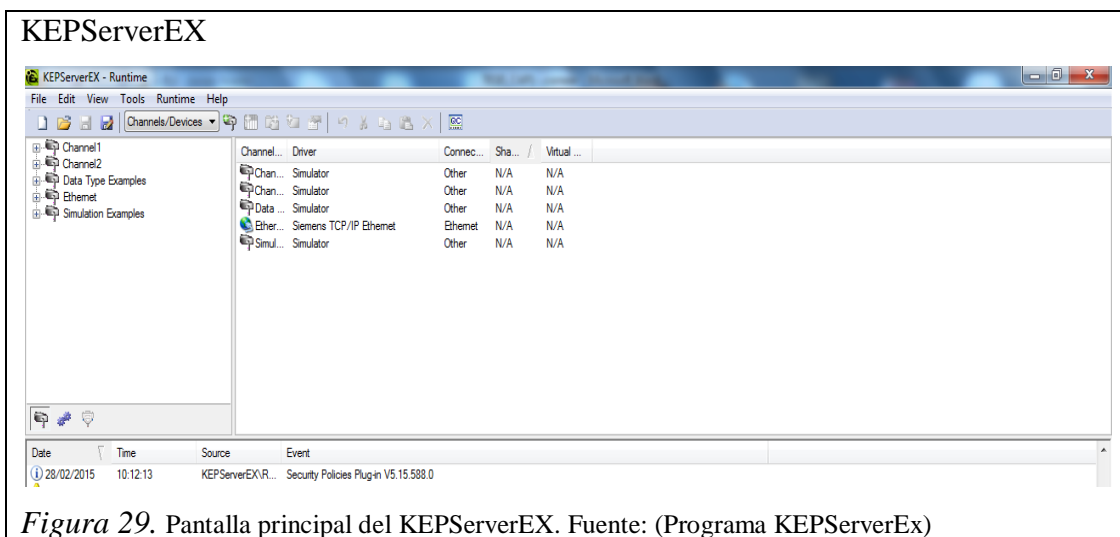


Figura 29. Pantalla principal del KEPServerEX. Fuente: (Programa KEPServerEx)

Para configurar la comunicación del OPC se realiza de la siguiente manera:

1. Crear un nuevo proyecto, como se muestra en la Figura 30.
2. Crear un canal con el identificador Ethernet, seleccionar el controlador de dispositivo Siemens TCP/IP Ethernet, seleccionar la tarjeta de red del computador utilizado, clic izquierdo en siguiente y luego finalizar, como se muestra en la Figura 30.

KEPServerEx OPC



Figura 30. Configuración del KEPServerEx OPC – Pasos 1 y 2. Fuente: (Programa KEPServerEX)

3. Añadir un nuevo dispositivo con el identificador S7-1212C, clic izquierdo en siguiente, seleccionar el modelo del dispositivo S7-1200, clic izquierdo en siguiente, ingresamos la dirección IP del PLC, por cuatro ocasiones dar clic izquierdo en siguiente, definir las direcciones de transporte Local TSAP (Transport Service Acces Point, punto de acceso al servicio de transporte) en 1000 y Remoto TSAP en 0301, por dos ocasiones clic izquierdo en siguiente y luego finalizar, como se muestra en la Figura 31.

KEPServerEx OPC



Figura 31. Configuración del KEPServerEx OPC – Paso 3. Fuente: (Programa KEPServerEX)

4. Crear las variables de intercambio, para las variables de entrada con la dirección I y variables de salidas con la dirección Q, como se muestra en la Figura 32.

KEPServerEX OPC

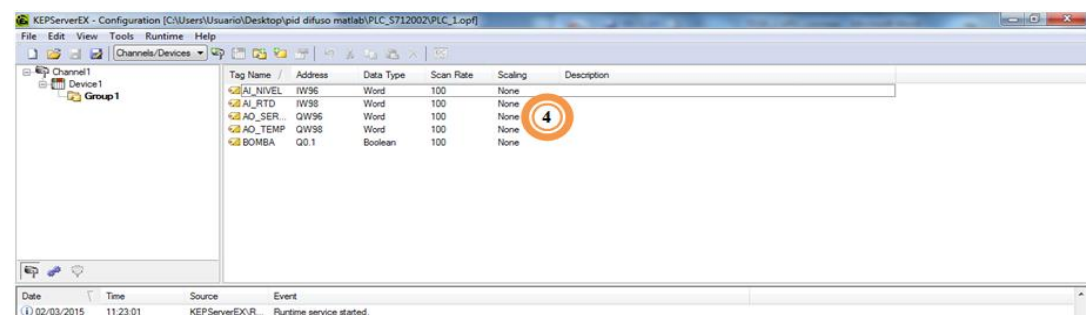


Figura 32 Variables de intercambio para la comunicación entre Matlab y el PLC.
Fuente: (Programa KEPServerEX)

5. Verificar la comunicación mediante el KEPServerEX OPC con el PLC Siemens S7-1200 dando clic izquierdo en Launch OPC Quick Client, como se muestra en la Figura 33.

Launch OPC Quick Client

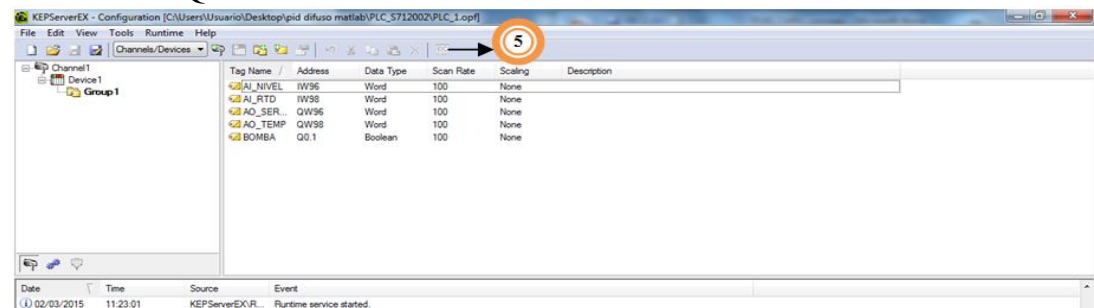


Figura 33. Comunicación en Launch OPC Quick Client. Fuente: (Programa KEPServerEX)

6. Verificar la comunicación de los sensores y actuadores dentro de la ventana del Launch OPC Quick Client, siempre y cuando esté conectado el PLC Siemens S7-1200 con la computadora personal, como se observa en la Figura 34.

KEPServerEX OPC

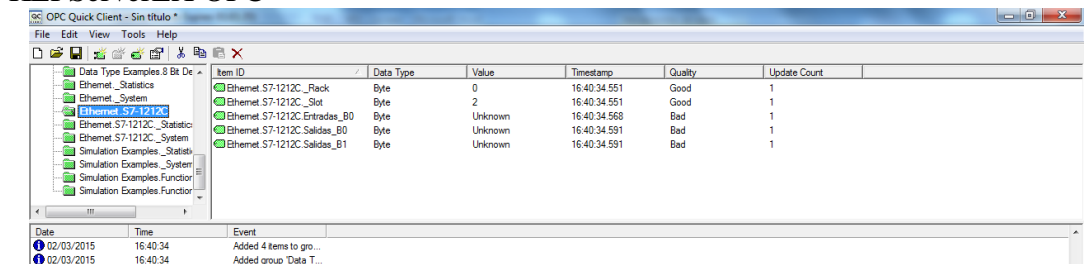


Figura 34. Verificación de la comunicación del KEPServerEX OPC con el PLC Siemens S7-1200.
Fuente: (Programa KEPServerEX)

7. El proyecto creado dentro del KEPServerEX se utiliza tanto para el controlador de nivel como para el de temperatura.

- **Matlab/Simulink**

Configurar el OPC KEPServerEx dentro de Simulink, realizando los siguientes pasos:

1. Agregar el bloque OPC de configuración, el bloque OPC de lectura y el bloque OPC de escritura, como se muestra en la Figura 35.

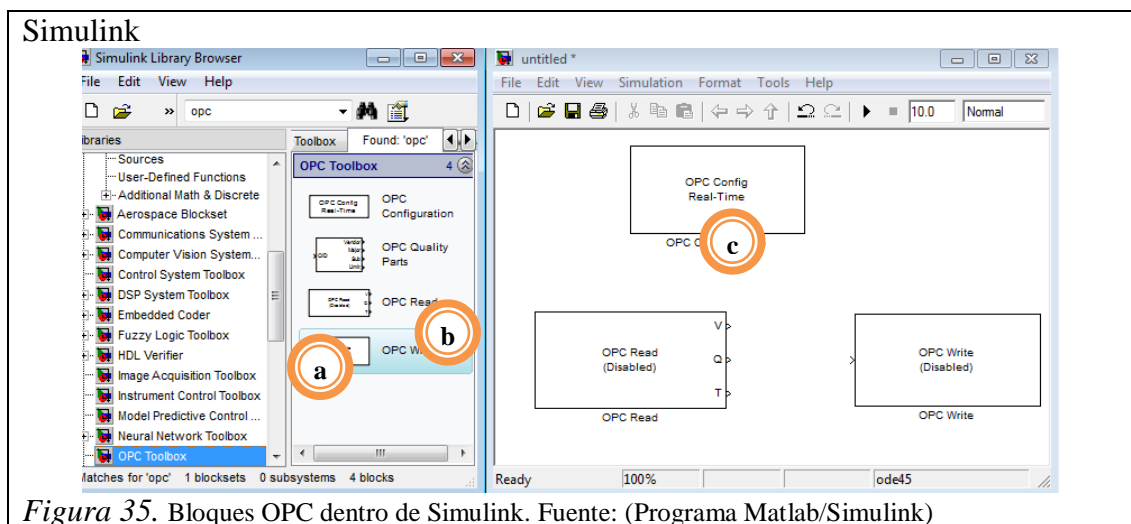


Figura 35. Bloques OPC dentro de Simulink. Fuente: (Programa Matlab/Simulink)

Los pasos para agregar los bloques OPC, se detallan a continuación:

- a. Seleccionar la librería OPC Toolbox.
 - b. Seleccionar cada uno de los bloques OPC.
 - c. Añadir los bloques OPC en el nuevo modelo de simulink.
2. Dentro del bloque OPC de configuración definir el cliente OPC utilizado, que en este caso es el PLC Siemens S7-1200, como se muestra en la Figura 36.

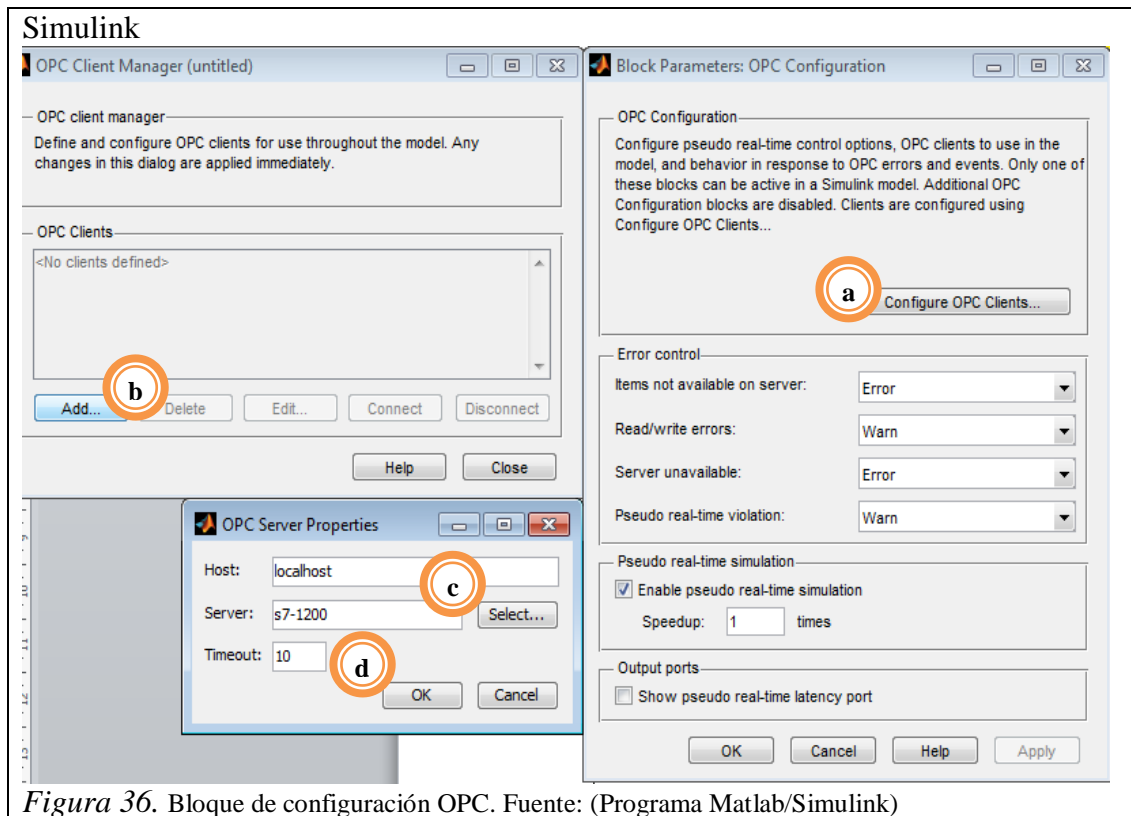


Figura 36. Bloque de configuración OPC. Fuente: (Programa Matlab/Simulink)

Los pasos para agregar los parámetros del bloque de configuración OPC, se detallan a continuación:

- a. Seleccionar Configure OPC Clients.
 - b. En la ventana OPC Client Manager, seleccionar Add.
 - c. Seleccionar el servidor s7-1200, generado del programa KEPServerEX.
 - d. Seleccionar OK en cada una de las ventanas abiertas.
3. Dentro del bloque OPC de lectura añadir las variables a leer, transmisores conectados al PLC Siemens S7-1200, como se muestra en la Figura 37.

Simulink

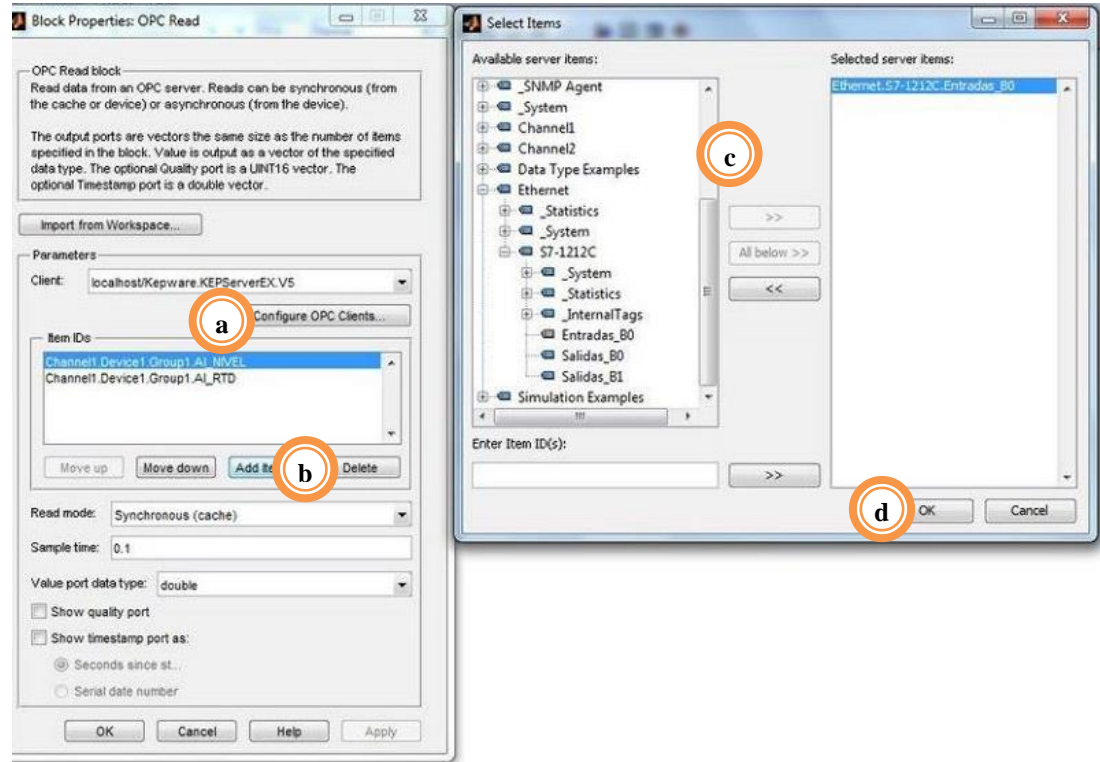


Figura 37. Bloque de lectura OPC. Fuente : (Programa Matlab/Simulink)

Los pasos para agregar los parámetros del bloque de lectura OPC, se detallan a continuación:

- a. Seleccionar Configure OPC Clients, se realiza la configuración de acuerdo a los literales b, c y d, del paso 2.
 - b. Seleccionar Add Items.
 - c. Agregar los ítems que se utilizan como variables de lectura.
 - d. Seleccionar OK.
4. Dentro del bloque OPC de escritura añadir las variables a controlar, actuadores conectados al PLC Siemens S7-1200, como se observa en la Figura 38.

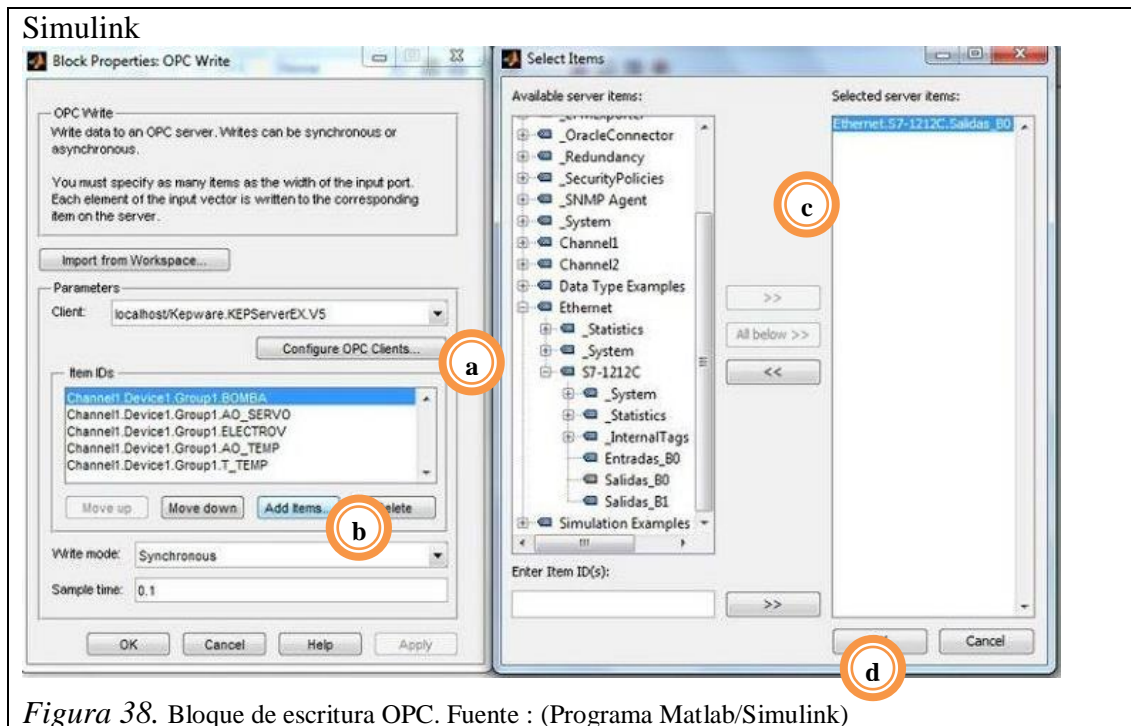
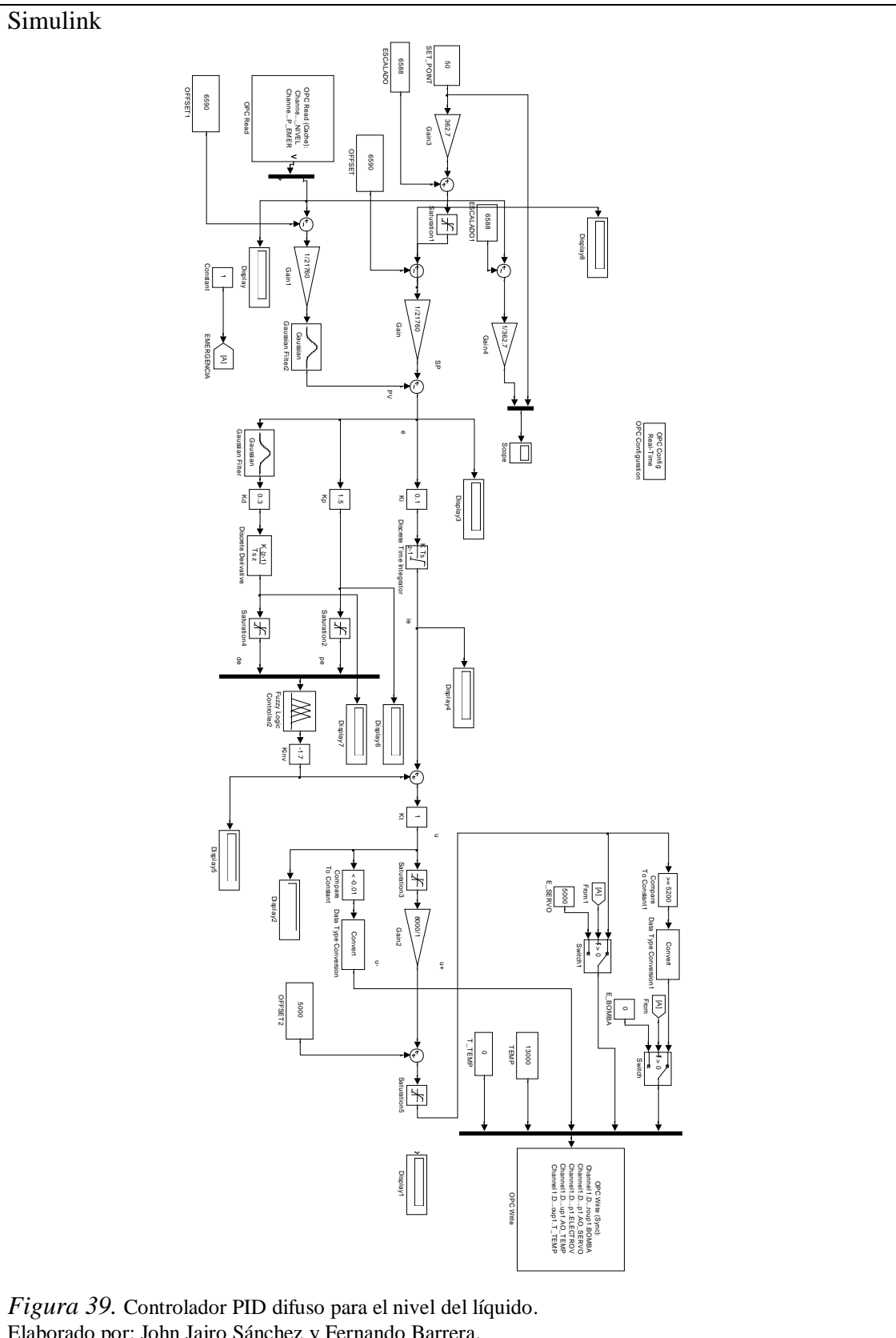


Figura 38. Bloque de escritura OPC. Fuente : (Programa Matlab/Simulink)

Los pasos para agregar los parámetros del bloque de escritura OPC, se detallan a continuación:

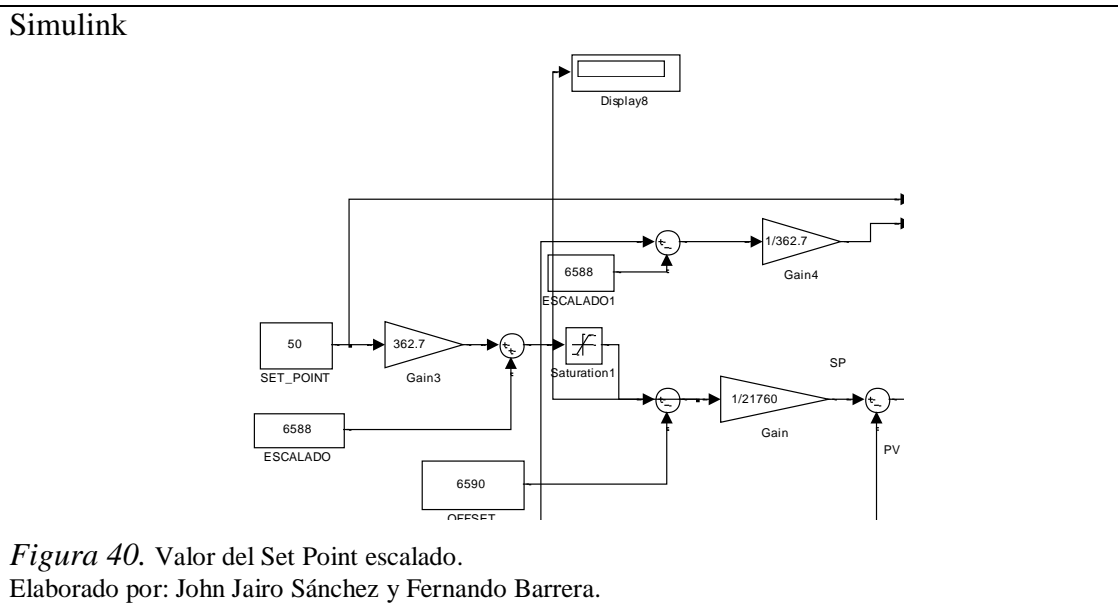
- Seleccionar Configure OPC Clients, se realiza la configuración de acuerdo a los literales b, c y d, del paso 2.
- Seleccionar Add Items.
- Agregar los ítems que se utilizan como variables de escritura.
- Seleccionar OK.

De acuerdo con el diagrama de bloques de la Figura 28, programar el controlador PID difuso dentro del entorno Simulink, como se muestra en la Figura 39.

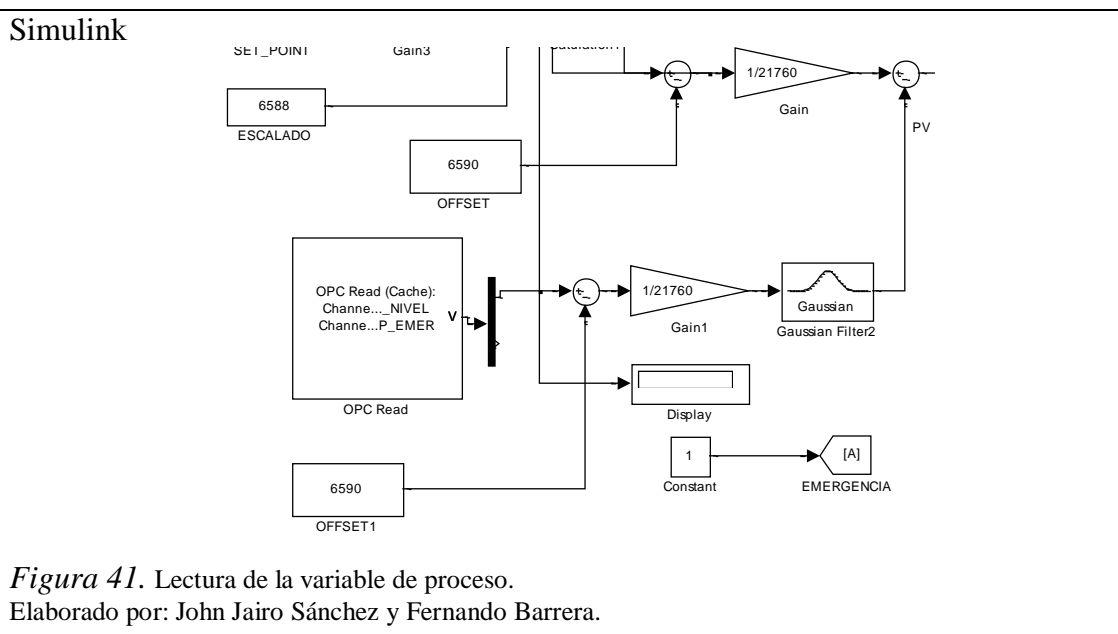


A continuación se detalla la programación en bloques realizada en Simulink del controlador PID difuso de la variable de nivel:

1. La Figura 40 muestra el valor del Set Point escalado a una magnitud de longitud de 0 cm. a 60 cm.



2. Lectura de la variable de proceso del transmisor de nivel, como se observa en la Figura 41.



-

- [illegible]

Para realizar el diseño del controlador PID difuso se va a utilizar el Fuzzy Logic Toolbox de Matlab que crea un archivo de extensión fis, el cual se carga en el bloque difuso dentro del entorno Simulink, para ello se realiza los siguientes pasos:

1. Crear las variables y términos lingüísticos que se utilizan para el controlador PID difuso.

Las variables lingüísticas son el error, la derivada del error y la acción de control, mientras que los términos lingüísticos son los conjuntos que pertenecen a las variables lingüísticas, detallados a continuación.

- Conjuntos difusos para el error
Egn: error grande negativo.
Epn: error pequeño negativo.
Ec: error cero.
Epp: error pequeño positivo.
Egp: error grande positivo.
- Conjuntos difusos para la derivada del error
Degn: derivada del error grande negativo.
Depn: derivada del error pequeño negativo.
Dec: derivada del error cero.
Depp: derivada del error pequeño positivo.
Degp: derivada del error grande positivo.
- Conjuntos difusos de la acción de control
Ugn: salida grande negativo.
Upn: salida pequeño negativo.
Uc: salida cero.
Upp: salida pequeño positivo.
Ugp: salida grande positivo.

2. Crear la base de reglas de inferencia para el controlador difuso de nivel, que se muestra en la Tabla 5.

Tabla 5.

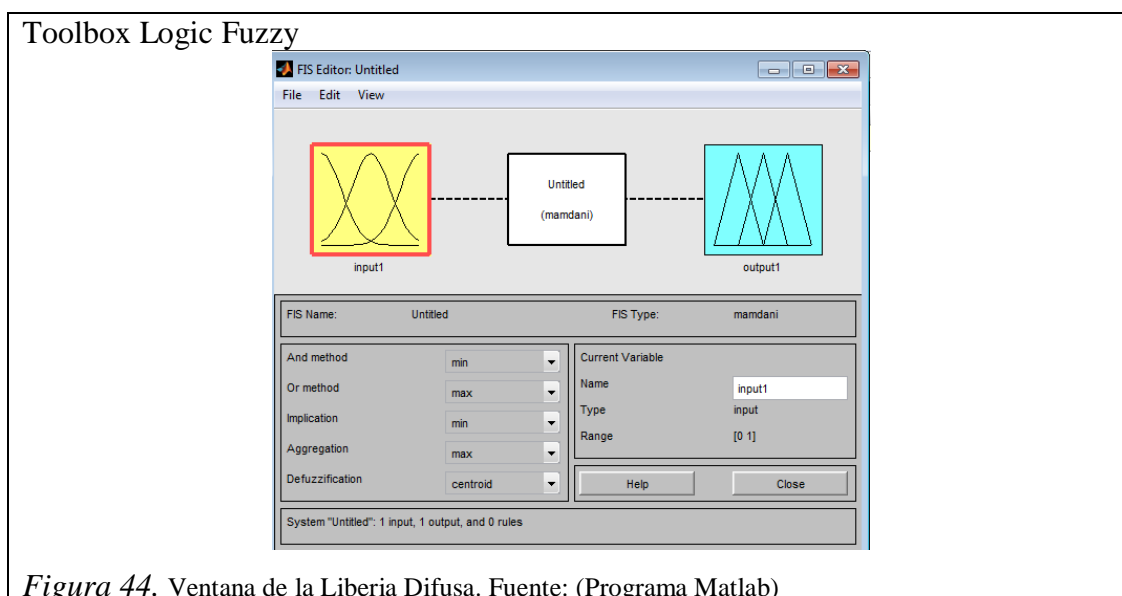
Reglas de inferencia para el controlador difuso de nivel

Error \ Derivada del error	Egn	Epn	Ec	Epp	Egp
Degn	Ugn	Ugn	Ugn	Upn	Uc
Depn	Ugn	Upn	Upn	Uc	Upp
Dec	Ugn	Upn	Uc	Upp	Ugp
Depp	Upn	Uc	Upp	Ugp	Ugp
Degp	Uc	Upp	Ugp	Ugp	Ugp

Nota: Abreviatura de los conjuntos de las variables lingüísticas.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

3. En el espacio de trabajo de Matlab escribir el comando “fuzzy”, posterior se visualiza la ventana de la librería difusa de Matlab, como lo muestra la Figura 44.



4. Establecer los parámetros de los conjuntos difusos del error para las funciones de membrecía, como se muestra en la Figura 45.

Toolbox Logic Fuzzy

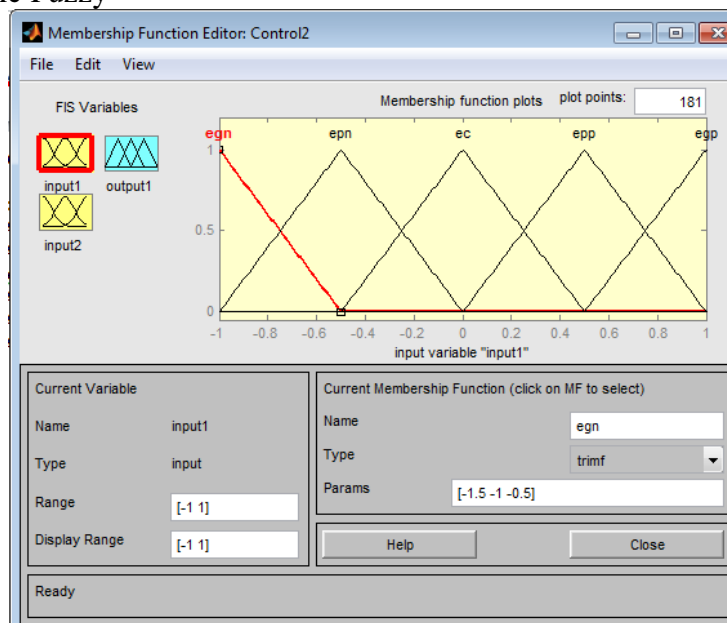


Figura 45. Funciones de membrecía para el error. Fuente: (Programa Matlab)

5. Establecer los parámetros de los conjuntos difusos de la derivada del error para las funciones de membrecía, como se muestra en la Figura 46.

Toolbox Logic Fuzzy

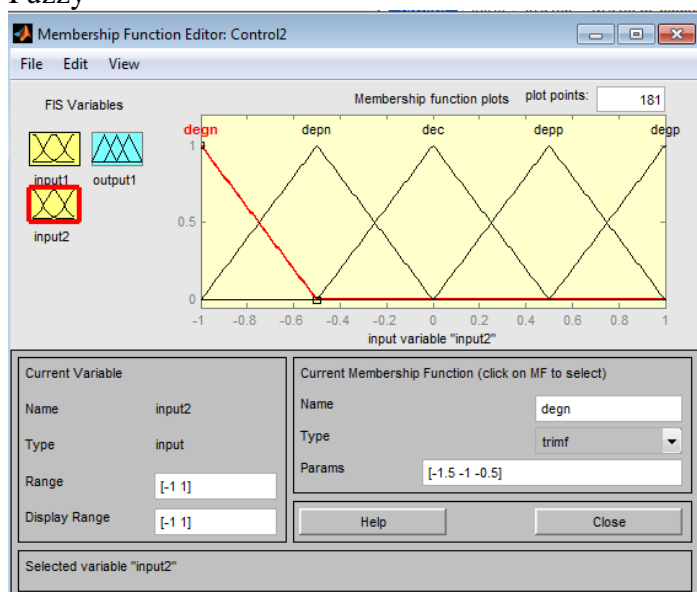
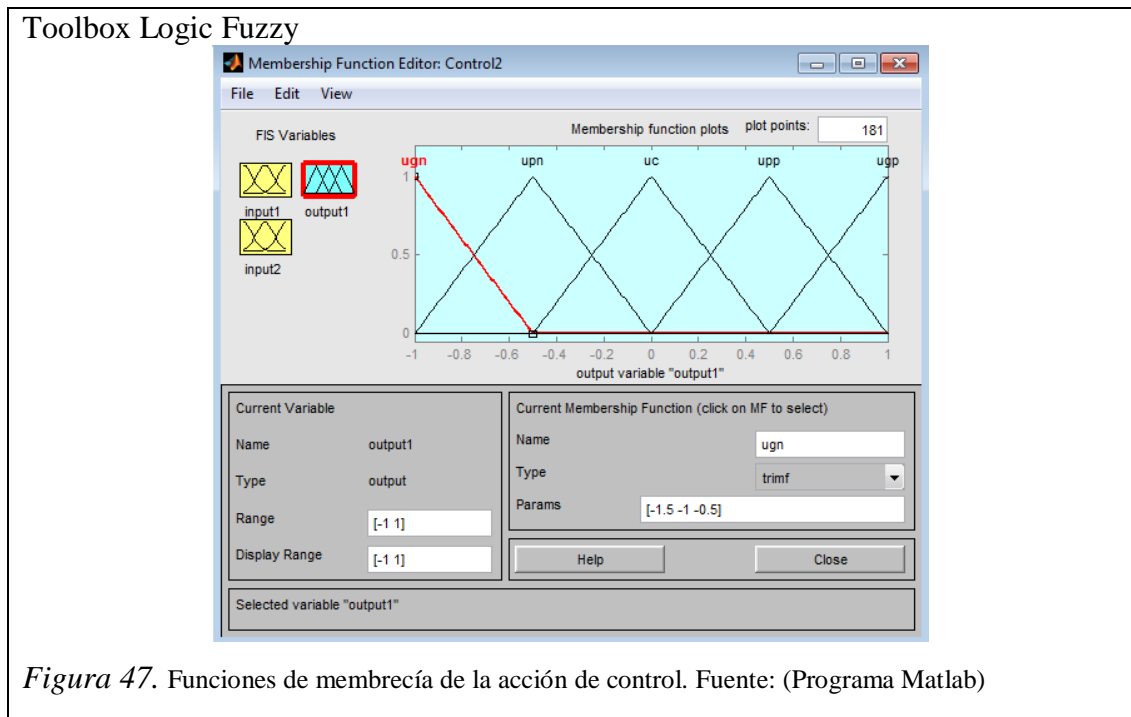


Figura 46. Funciones de membrecía para la derivada del error. Fuente: (Programa Matlab)

6. Establecer los parámetros de los conjuntos difusos de la acción de control para las funciones de membrecía, como se muestra en la Figura 47.



7. Ingresar la base de reglas creadas para el controlador difuso de la Tabla 5, como se muestra en la Figura 48.

El número de conjuntos es de 5 por lo tanto, el número de reglas es de 25, generando así el hecho de contemplar más posibilidades y obtener una superficie más lineal.

Toolbox Logic Fuzzy

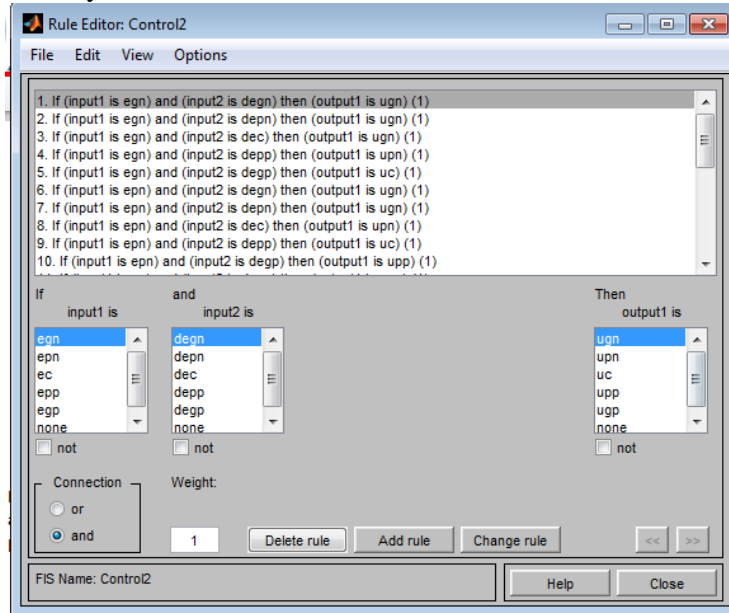


Figura 48. Edición de las 25 reglas de fusificación. Fuente: (Programa Matlab)

Para realizar el diseño de la interfaz gráfica del controlador PID difuso tanto para la variable de nivel como de temperatura, se utiliza la interfaz gráfica de usuario en Matlab GUIDE; se sigue los siguientes pasos:

1. Para crear un nuevo proyecto, en el espacio de trabajo de Matlab se escribe el comando “guide”, desplegando la ventana GUIDE Quick Start, como lo muestra la Figura 49.

GUIDE Quick Start

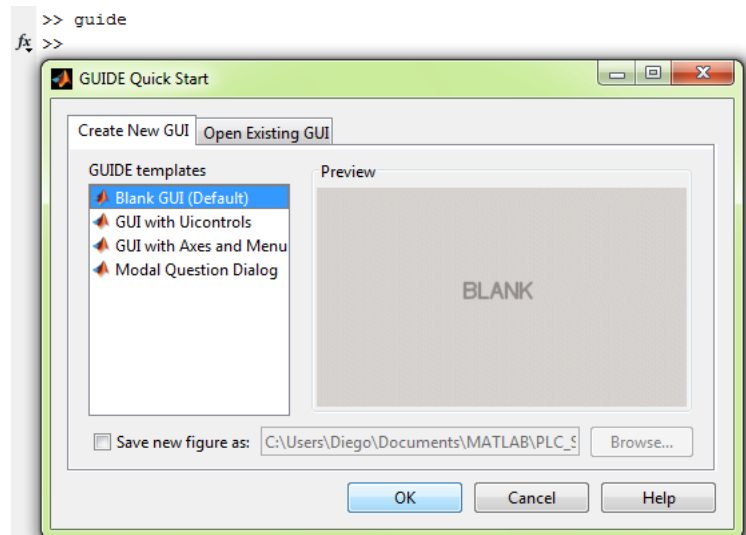


Figura 49. Ventana para crear nuevo proyecto GUIDE. Fuente: (Programa Matlab)

2. Abrir la interfaz gráfica de usuario en blanco, la que presenta un formulario nuevo en el que permite diseñar un nuevo programa, como se muestra en la Figura 50.

GUIDE Quick Start

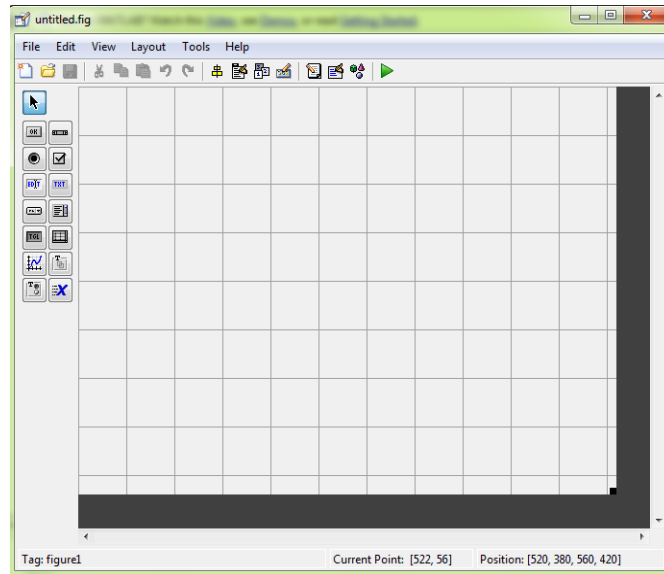


Figura 50. Interfaz gráfica de usuario en blanco. Fuente: (Programa Matlab)

3. En el entorno de diseño de GUIDE añadir cada uno de los componentes que se emplea en el diseño de la interfaz gráfica del controlador PID difuso, como se muestra en la Figura 51.

GUIDE Quick Start

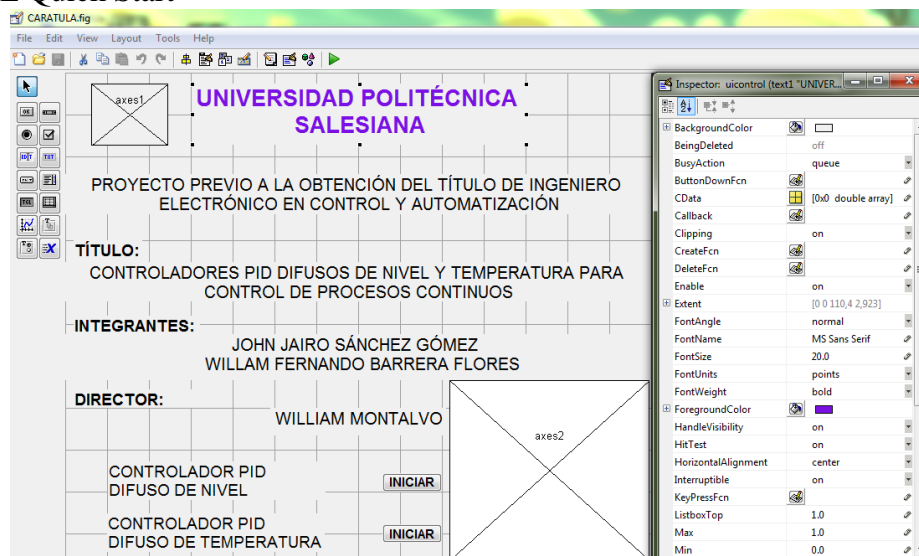


Figura 51. Interfaz gráfica creada en Guide. Fuente: (Programa Matlab)

4. Al finalizar el diseño de la interfaz gráfica guardar con el nombre CARATULA con la extensión fig, como se muestra en la Figura 52.

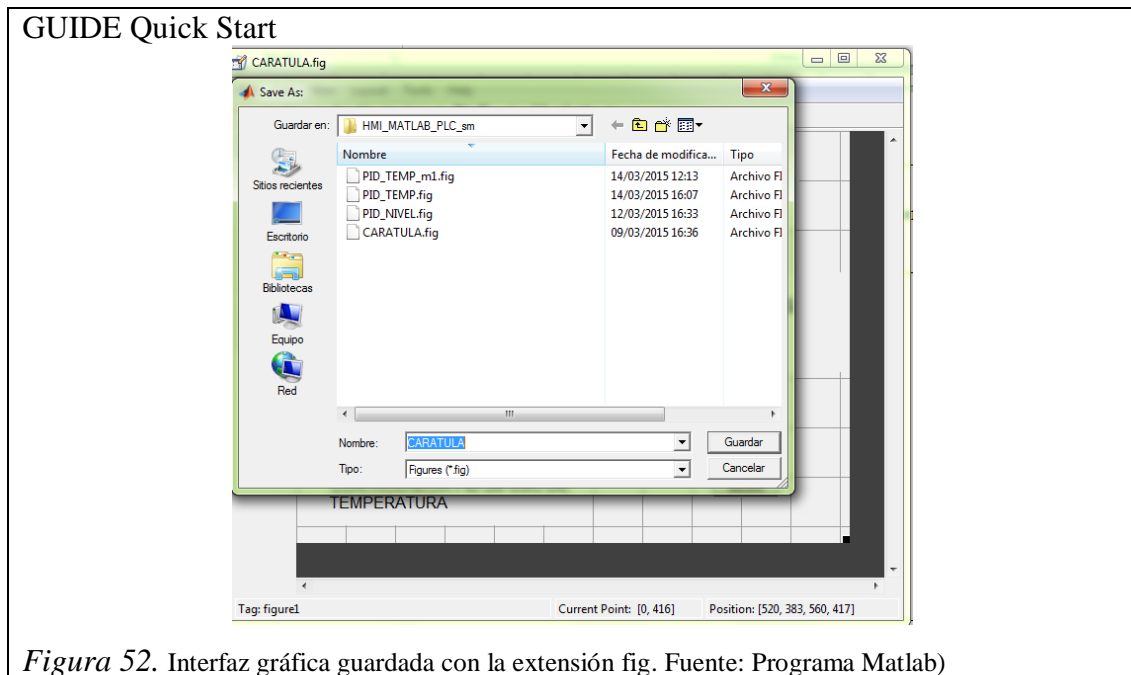


Figura 52. Interfaz gráfica guardada con la extensión fig. Fuente: Programa Matlab)

- **TIA PORTAL**

Para que el PLC Siemens S7-1200, actúe como una tarjeta de adquisición de datos se procede a realizar los siguientes pasos:

1. Abrir el programa TIA PORTAL V12, donde se selecciona crear proyecto, a este proyecto se le detalla tanto el nombre, la ruta donde va estar guardado el archivo, el autor o los detalles que realiza el proyecto y un comentario referente al proyecto; por último seleccionar en crear, como se muestra en la Figura 53.

TIA PORTAL

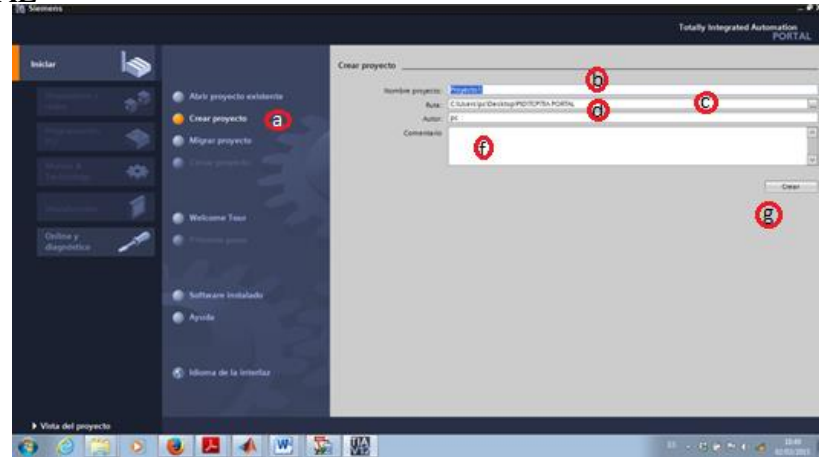


Figura 53. Creación de un nuevo proyecto. Fuente: (Programa TIA PORTAL)

2. Agregar el CPU en este caso el 1212C AC/DC/Rly, como se muestra en la Figura 54.

TIA PORTAL



Figura 54. Selección del CPU. Fuente: (Programa TIA PORTAL)

Para agregar el CPU requerido, se realiza los siguientes pasos:

- a. Seleccionar la pestaña dispositivos y redes.
- b. Seleccionar agregar dispositivo.
- c. En la pestaña controladores elegir el modelo de CPU, y dar clic izquierdo en agregar.

3. Agregar el módulo de expansión de entradas y salidas análogas SM-1234, como se observa en la Figura 55.

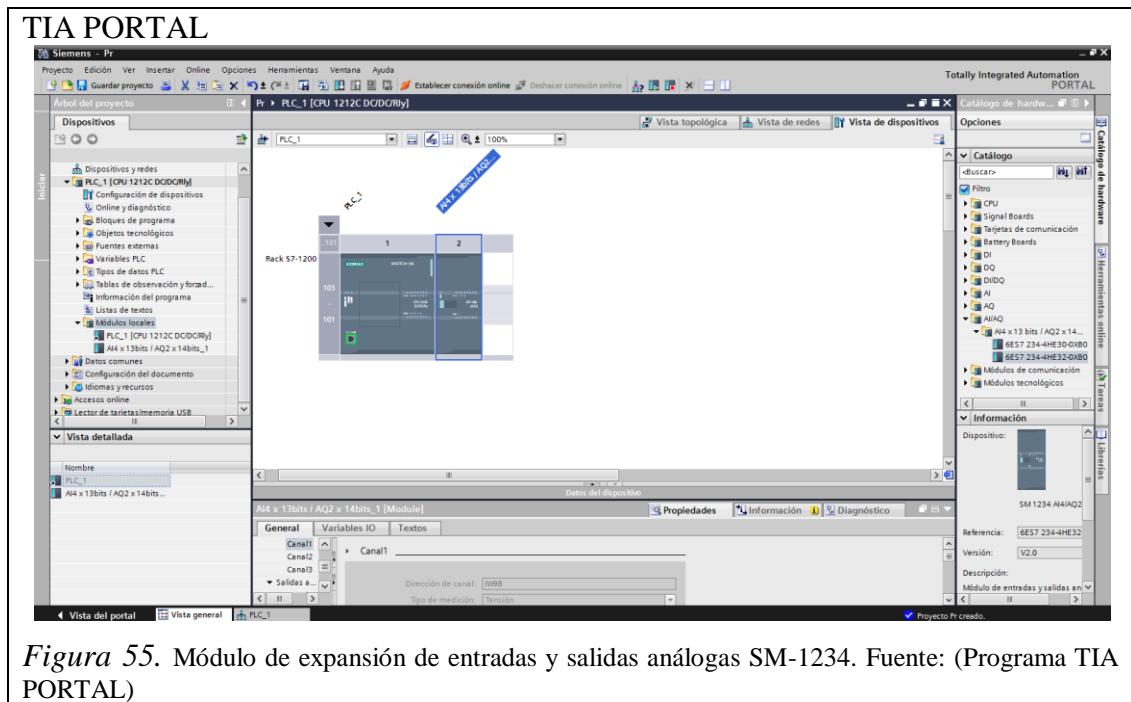


Figura 55. Módulo de expansión de entradas y salidas análogas SM-1234. Fuente: (Programa TIA PORTAL)

Para agregar el módulo de expansión requerido, se realiza los siguientes pasos:

- a. Seleccionar el espacio de módulo vacío.
 - b. Seleccionar el módulo de entradas y salidas análogas.
 - c. Dar doble clic izquierdo sobre el módulo de entradas y salidas análogas.
4. Establecer los parámetros de las entradas y salidas análogas, ya sean por tensión o corriente, como se observa en la Figura 56.

TIA PORTAL

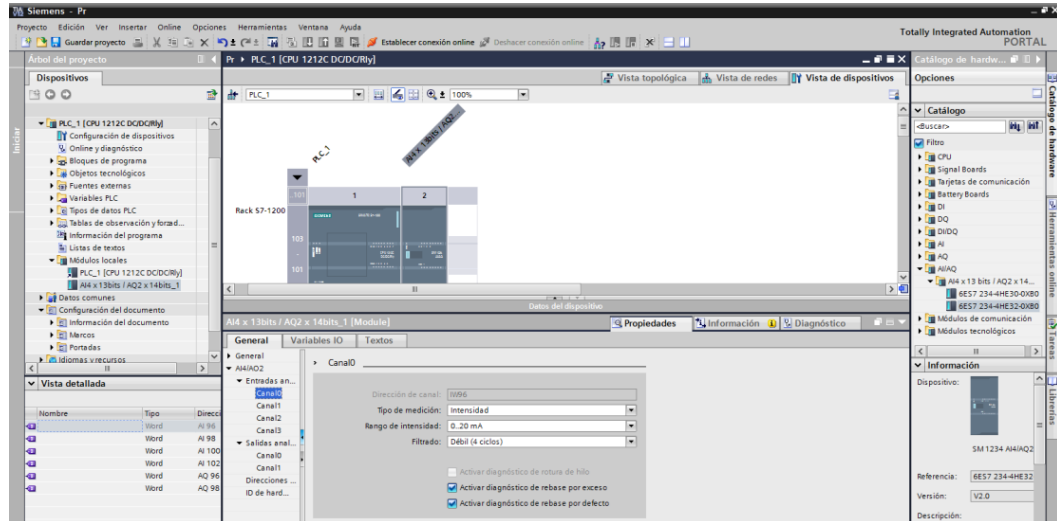


Figura 56. Configuración de los parámetros de variables de entradas y salidas analógicas. Fuente: (Programa TIA PORTAL)

5. Para el funcionamiento de la planta con el controlador PID difuso en Matlab, se carga un archivo en blanco, con los segmentos sin programar, como se observa en la Figura 57.

TIA PORTAL

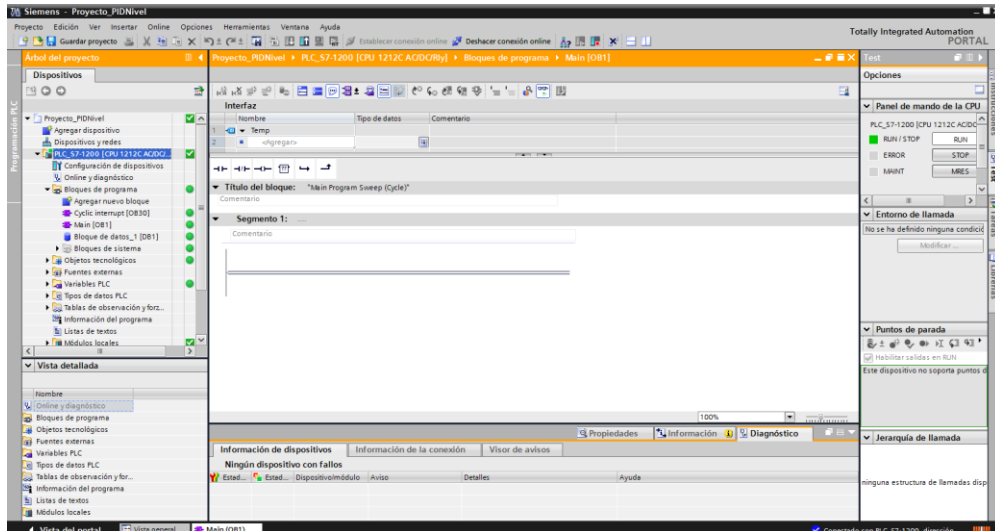
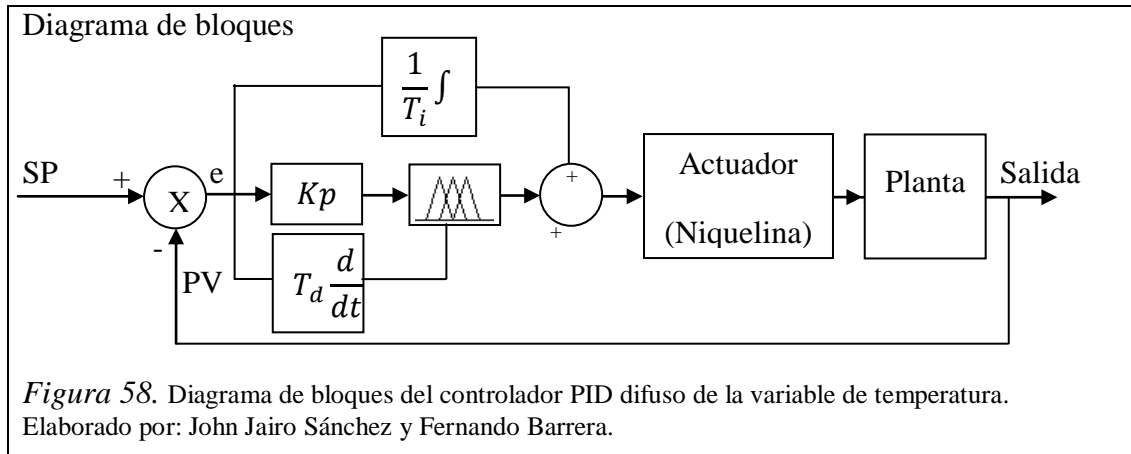


Figura 57. Proyecto en blanco configurado las entradas y salidas analógicas. Fuente: (Programa TIA PORTAL)

3.2.1.2 Controlador PID difuso de la variable de temperatura

En la Figura 58, se detalla el diagrama de bloques para realizar el controlador PID difuso de la variable de temperatura.



- **Matlab/Simulink**

De acuerdo con el diagrama de bloques de la Figura 58, programar el controlador PID difuso dentro del entorno Simulink, como se visualiza en la Figura 59.

Simulink

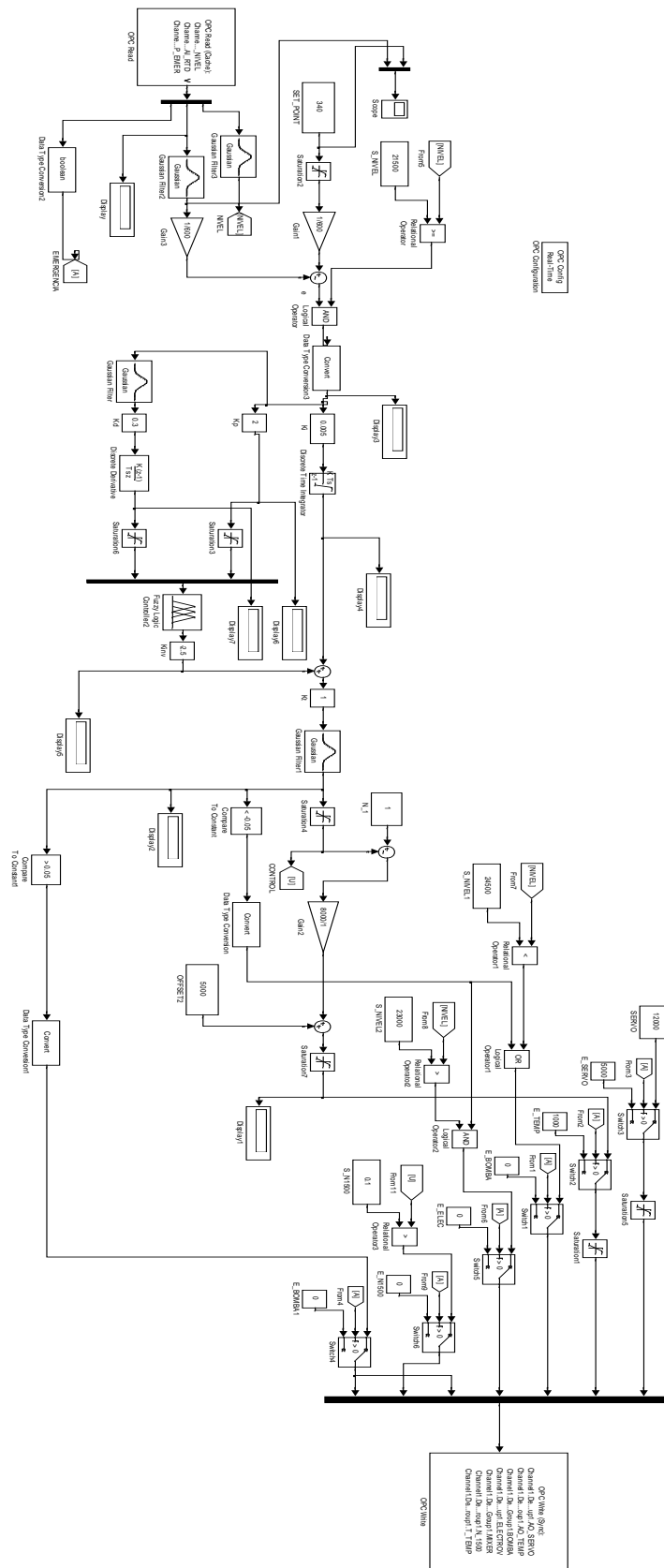
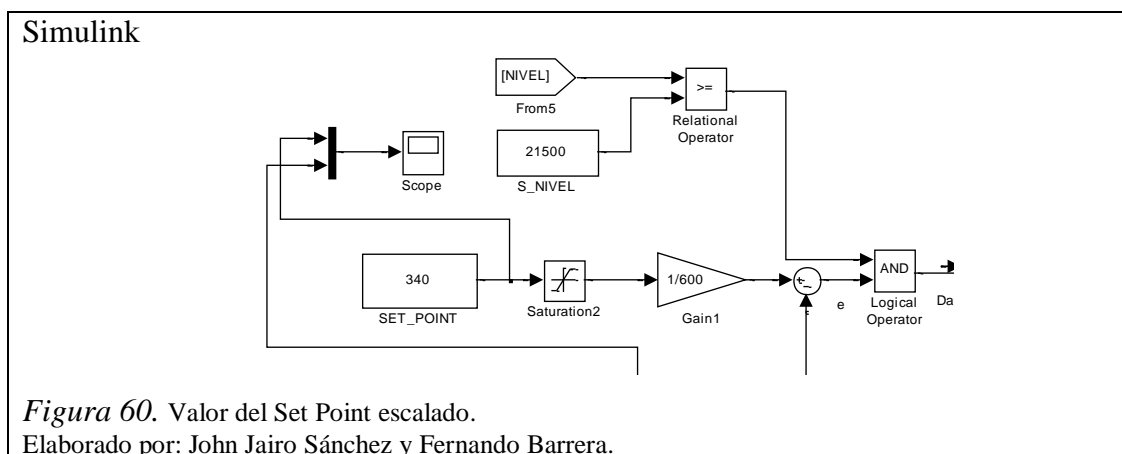


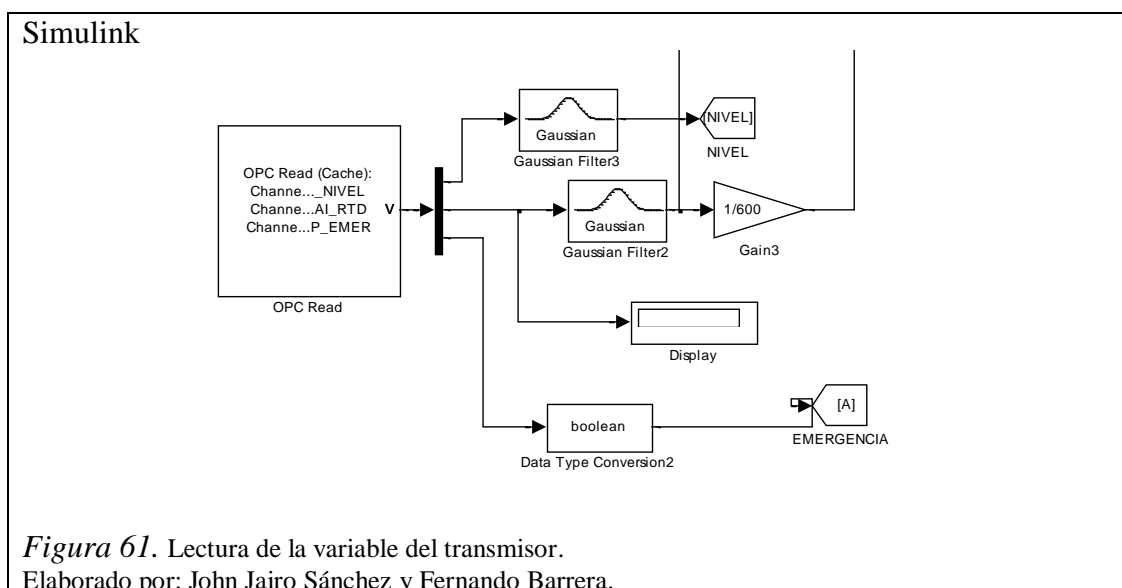
Figura 59. Controlador PID difuso para la temperatura del líquido.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

A continuación se detalla la programación en bloques realizada en Simulink del controlador PID difuso de la variable de temperatura:

1. La Figura 60 muestra el valor del Set Point escalado a una magnitud de temperatura en grados centígrados.



2. Lectura de la variable de proceso del transmisor de temperatura, como se observa en la Figura 61.



3. Figura 62 se muestra el controlador PID difuso, en un bloque el parámetro K_p y K_d , y sumado el parámetro K_i .

4. La acción de control es leída por el actuador, en este caso la tarjeta de control de la niquelina, como se observa en la Figura 63.

3.2.2 Controladores PID difusos en el PLC siemens S7-1200

Los PLC's operan de manera secuencial y cíclica, es decir, una vez finalizado el recorrido completo de un programa, comienza a ejecutar su primera instrucción.

3.2.2.1 Controladores PID difusos de las variables de nivel y temperatura

Para realizar los controladores PID difusos de las variables de nivel y temperatura se basan a los diagramas de bloques de la Figura 28 y Figura 58.

El controlador PID difuso que se diseña para el control del proceso de nivel con el PLC Siemens S7-1200; emplea el programa TIA PORTAL, donde se realiza la programación en lenguaje de programación de alto nivel basado en texto SCL y lenguaje de programación gráfico KOP.

- **TIA PORTAL**

Para la programación del PID Difuso de la variable de nivel, se realiza los siguientes pasos:

1. Crear un nuevo proyecto seleccionando el CPU y el módulo de expansión de entradas y salidas análogas; establecer los parámetros de tensión o corriente de las entradas y salidas análogas.
2. Crear un bloque lógico FC llamado FUZZY, donde se programa la lógica difusa utilizando lenguaje de programación SCL, como se observa en la Figura 64.

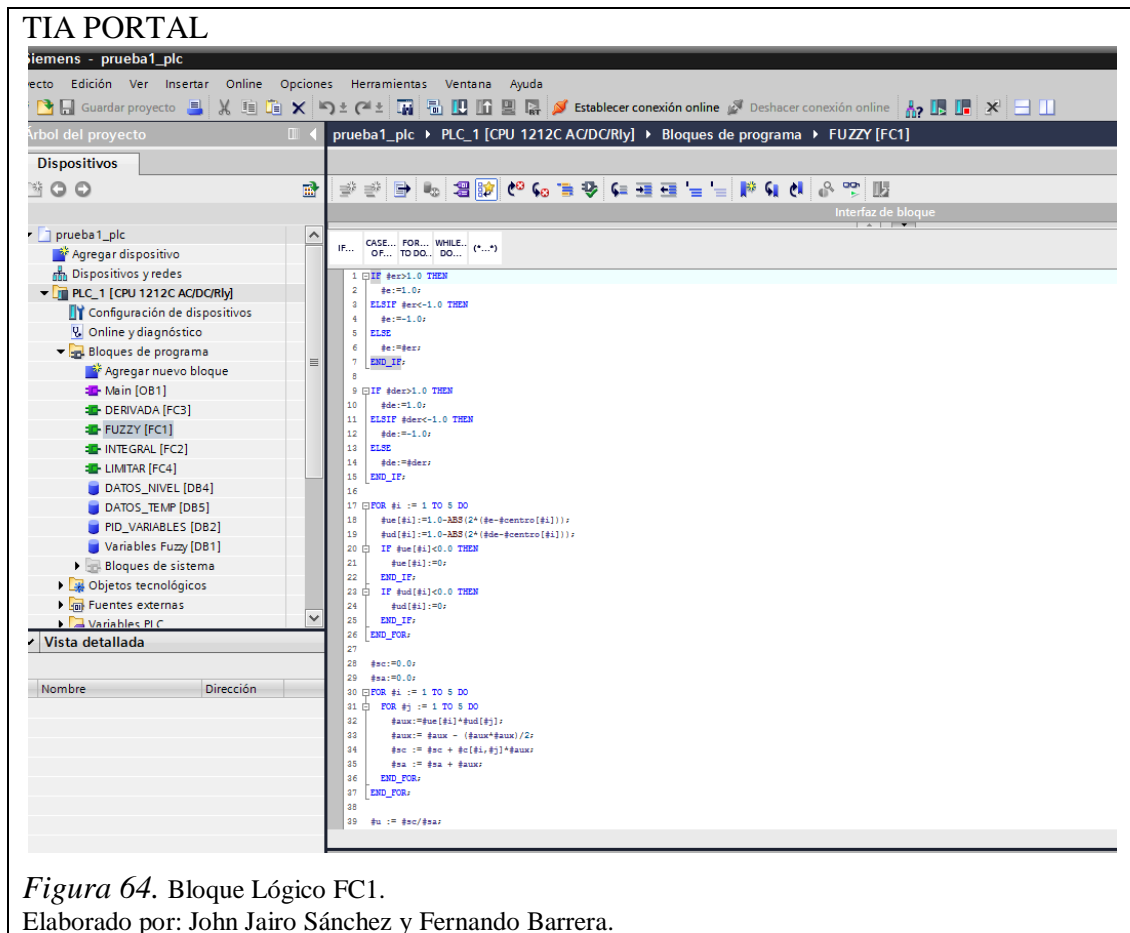


Figura 64. Bloque Lógico FC1.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Los pasos de programación del bloque FUZZY, se detallan a continuación:

- La Figura 65 muestra el código de programación que efectúa la limitación del error en los valores de menos uno a uno.

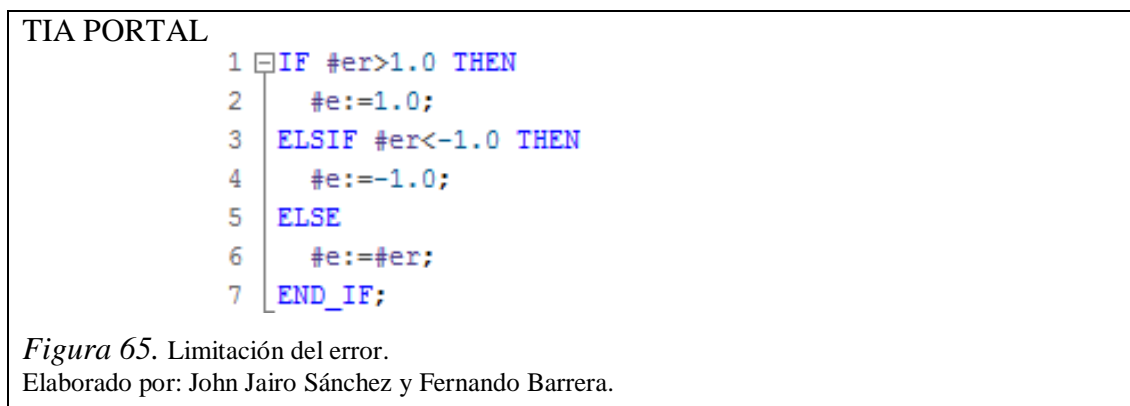


Figura 65. Limitación del error.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- b. La Figura 66 muestra el código de programación que efectúa la limitación de la derivada del error en los valores de menos uno a uno.

TIA PORTAL

```
9 IF #der>1.0 THEN
10     #de:=1.0;
11 ELSIF #der<-1.0 THEN
12     #de:=-1.0;
13 ELSE
14     #de:=#der;
15 END_IF;
```

Figura 66. Limitación de la derivada del error.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- c. La Figura 67 muestra el código de programación que efectúa la máquina fusificadora, en la cual se evalúa el error y la derivada del error en las cinco funciones de pertenencia.

TIA PORTAL

```
17 FOR #i := 1 TO 5 DO
18     #ue[#i]:=1.0-ABS(2*(#e-#centro[#i]));
19     #ud[#i]:=1.0-ABS(2*(#de-#centro[#i]));
20 IF #ue[#i]<0.0 THEN
21     #ue[#i]:=0;
22 END_IF;
23 IF #ud[#i]<0.0 THEN
24     #ud[#i]:=0;
25 END_IF;
26 END_FOR;
```

Figura 67. Máquina fusificadora.

Elaborado por: John Jairo Sánchez - Fernando Barrera.

- d. La Figura 68 muestra el código de programación que efectúa la máquina defusificadora, en donde calcula la acción de control en base a que función de pertenencia se encuentran el error y derivada del error, respectivamente.

TIA PORTAL

```

28 #sc:=0.0;
29 #sa:=0.0;
30 FOR #i := 1 TO 5 DO
31   FOR #j := 1 TO 5 DO
32     #aux:=#ue[#i]*#ud[#j];
33     #aux:= #aux - (#aux*#aux)/2;
34     #sc := #sc + #c[#i,#j]*#aux;
35     #sa := #sa + #aux;
36   END_FOR;
37 END_FOR;
38
39 #u := #sc/#sa;

```

Figura 68. Máquina desfusificadora.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

3. Crear un bloque de datos DB1 llamado Variables Fuzzy, donde se crea una matriz llamada **c**, con la base de reglas; y otra matriz llamada **centro**, con los centros de todas las funciones de pertenencia, como se observa en la Figura 69.

TIA PORTAL

Variables Fuzzy		Nombre	Tipo de datos	Valor de arranq...	Remanen...	Visible en ...	Valor de a...
1	Static						
2	Static	c	Array [1..4, ...]			<input checked="" type="checkbox"/>	
3		c[1,1]	Real	1.0		<input checked="" type="checkbox"/>	
4		c[1,2]	Real	1.0		<input checked="" type="checkbox"/>	
5		c[1,3]	Real	1.0		<input checked="" type="checkbox"/>	
6		c[1,4]	Real	0.0		<input checked="" type="checkbox"/>	
7		c[2,1]	Real	1.0		<input checked="" type="checkbox"/>	
8		c[2,2]	Real	1.0		<input checked="" type="checkbox"/>	
9		c[2,3]	Real	0.5		<input checked="" type="checkbox"/>	
10		c[2,4]	Real	0.0		<input checked="" type="checkbox"/>	
11		c[3,1]	Real	1.0		<input checked="" type="checkbox"/>	
12		c[3,2]	Real	0.5		<input checked="" type="checkbox"/>	
13		c[3,3]	Real	0.0		<input checked="" type="checkbox"/>	
14		c[3,4]	Real	0.0		<input checked="" type="checkbox"/>	
15		c[4,1]	Real	0.0		<input checked="" type="checkbox"/>	
16		c[4,2]	Real	0.0		<input checked="" type="checkbox"/>	
17		c[4,3]	Real	0.0		<input checked="" type="checkbox"/>	
18		c[4,4]	Real	0.0		<input checked="" type="checkbox"/>	
19	centro		Array [1..5] of Real			<input checked="" type="checkbox"/>	
20		centro[1]	Real	-1.0		<input checked="" type="checkbox"/>	
21		centro[2]	Real	-0.5		<input checked="" type="checkbox"/>	
22		centro[3]	Real	0.0		<input checked="" type="checkbox"/>	
23		centro[4]	Real	0.5		<input checked="" type="checkbox"/>	

Figura 69. Bloque de datos DB1.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

4. Crear un bloque lógico FC5 llamado CALCULAR_ERROR, donde se programa el cálculo del error porcentual, como se observa en la Figura 70.

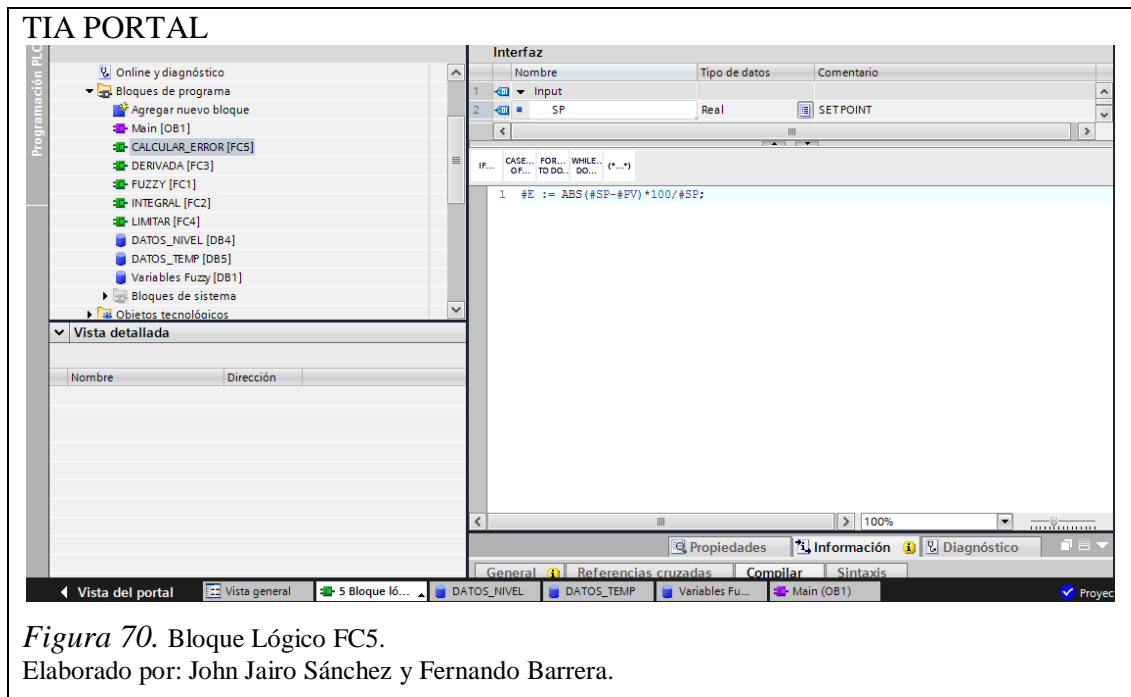


Figura 70. Bloque Lógico FC5.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

5. Crear un bloque lógico FC2 llamado INTEGRAL, donde se programa el cálculo de la integral discreta del error, en un tiempo de 100 milisegundos, como se observa en la Figura 71.

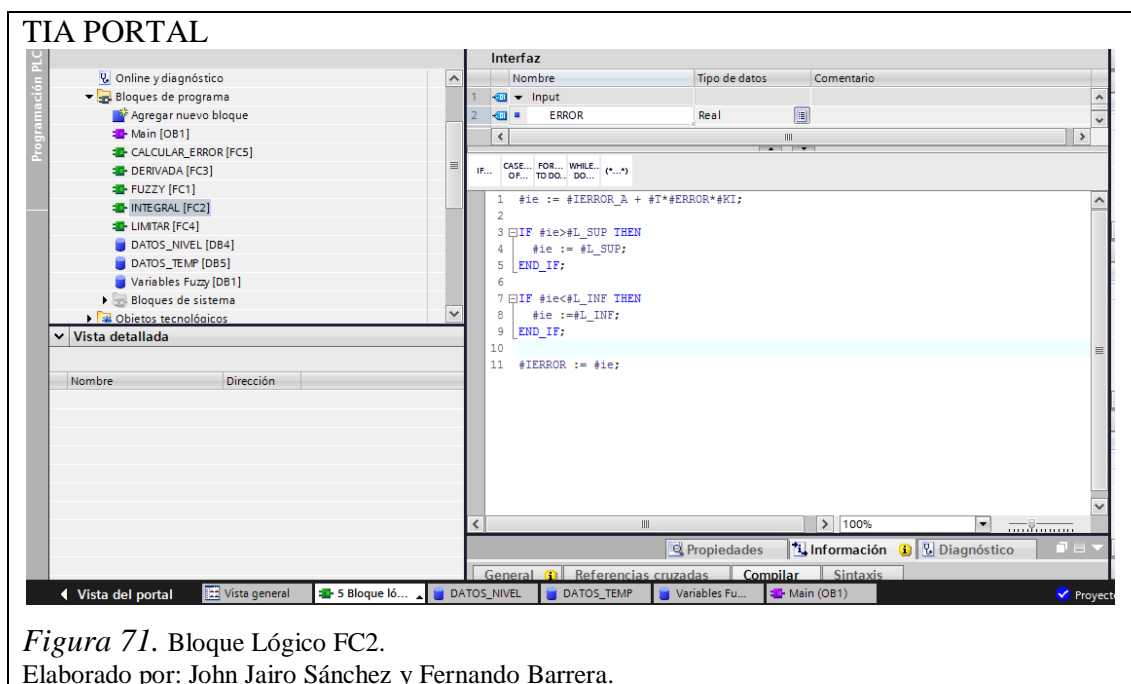


Figura 71. Bloque Lógico FC2.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

-
- The screenshot displays the TIA Portal software interface. On the left, the 'Objetos tecnológicos' (Technological Objects) tree is visible, showing various blocks like 'Online y diagnóstico', 'Bloques de programa', 'Agregar nuevo bloque', 'Main [OB1]', 'CALCULAR_ERROR [FC5]', 'DERIVADA [FC3]', 'FUZZY [FC1]', 'INTEGRAL [FC2]', 'LIMITAR [FC4]', 'DATOS_NIVEL [DB4]', 'DATOS_TEMP [DB5]', and 'Variables Fuzzy [DB1]'. The main editor area shows a single step with a 'Real' data type and the variable 'ERROR'. The right pane shows the 'Interfaz' (Interface) window with a table for the step's interface, including columns for 'Nombre', 'Tipo de datos', and 'Comentario'. The table has one row for 'ERROR' with type 'Real' and comment 'ERROR ACTUAL'. The bottom status bar shows the project name 'Proyecto' and the current step 'Main (OB1)'.

-
- The screenshot displays the TIA Portal software interface for editing a Ladder Logic (LAD) function block, specifically FC4. The interface is divided into several panes:
- Left Pane (Project Tree):** Shows the project structure under 'Programación PLC'. The 'Objetos tecnológicos' (Technological Objects) folder is expanded, showing various data objects like 'DATOS_NIVEL [DB4]', 'DATOS_TEMP [DB5]', and 'Variables Fuzzy [DB1]'. The 'Vista detallada' (Detailed View) tab is active.
 - Top Pane (Interfaz):** Displays the variable declaration for the function block. It shows a variable 'AUX' of type 'Real'.
 - Main Editor:** Contains the Ladder Logic code for FC4. The code is as follows:


```

1  #AUX := #UK;
2
3  IF #AUX > 1 THEN
4      // Statement section IF
5      #AUX := 1;
6  END_IF;
7
8  IF #AUX < 0 THEN
9      // Statement section IF
10     #AUX := 0;
11  END_IF;
12
13  #U := #AUX;
      
```
 - Bottom Pane (Status Bar):** Shows the current view is 'Vista general' (General View). Other tabs include 'Vista del portal', 'Vista general', '5 Bloque l6...', 'DATOS_NIVEL', 'DATOS_TEMP', 'Variables Fu...', and 'Main (OB1)'. The 'Proy' (Project) button is also visible.

8. Crear un bloque de datos DB4 llamado DATOS_NIVEL, donde se crea todas las variables utilizadas para el controlador PID difuso de nivel, como se observa en la Figura 74.

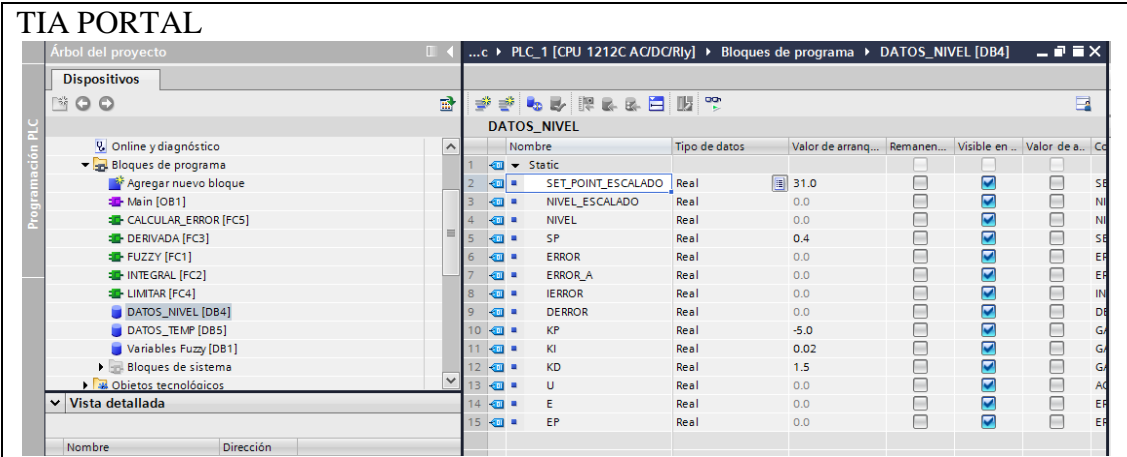


Figura 74. Bloque de Datos DB4.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

9. Crear un bloque de datos DB5 llamado DATOS_TEMP, donde se crea todas las variables utilizadas para el controlador PID difuso de temperatura, como se observa en la Figura 75.

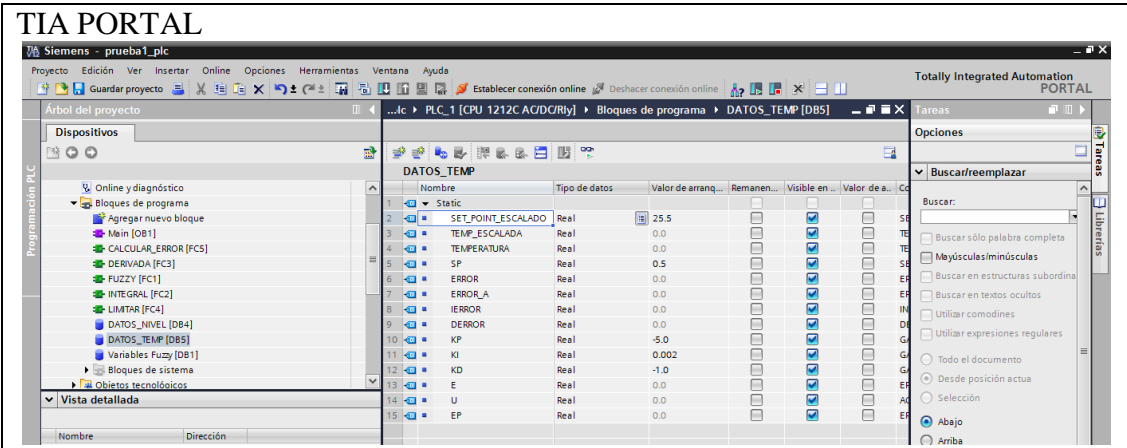


Figura 75. Bloque de datos DB5.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

10. En el bloque de organización Main (OB1) programar los controladores PID Difusos, donde se puede abrir los bloques creados FC y el DB; como se muestra en la Figura 76.

En el Anexo 1, se detalla la programación de los controladores PID Difusos de las variables de nivel y temperatura para el PLC Siemens S7-1200.

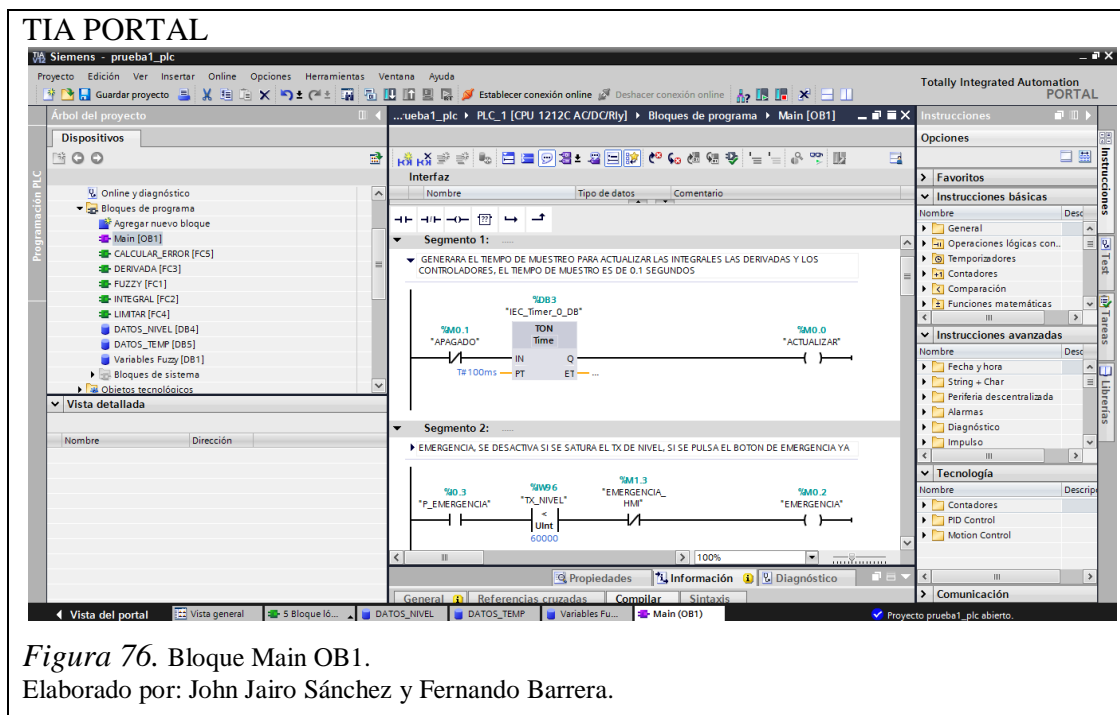


Figura 76. Bloque Main OB1.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Para realizar el diseño de la interfaz gráfica del controlador PID difuso tanto para la variable de nivel como de temperatura, se utiliza la interfaz gráfica propia del programa TIA PORTAL; se sigue los siguientes pasos:

1. En el programa TIA PORTAL, agregar un dispositivo HMI, como se muestra en la Figura 77.

TIA PORTAL

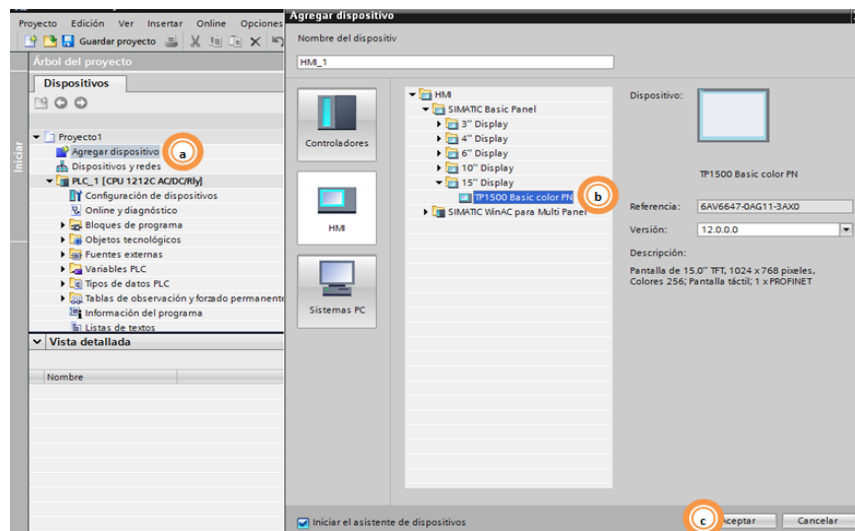


Figura 77. Ventana para agregar dispositivo HMI. Fuente: (Programa TIA PORTAL)

Los pasos para agregar el dispositivo HMI, se detallan a continuación:

- Seleccionar Agregar nuevo dispositivo.
- Seleccionar la pestaña HMI y escoger el modelo del dispositivo.
- Seleccionar Aceptar.

- En la ventana asistente del panel de operador, se configura la conexión al PLC, como se muestra en la Figura 78.

TIA PORTAL

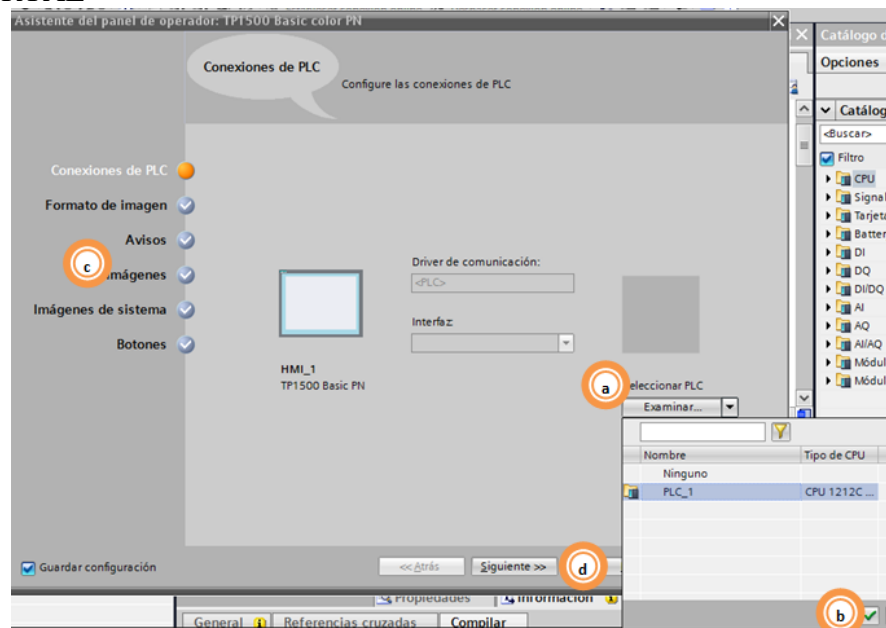


Figura 78. Configuración de conexión al PLC. Fuente: (Programa TIA PORTAL)

Los pasos para las conexiones del PLC, se detallan a continuación:

- a. Dar clic izquierdo en Seleccionar PLC.
- b. Seleccionar el PLC a utilizar y el icono de visto.
- c. Seleccionar el formato del HMI de acuerdo al requerimiento del proyecto a utilizarse.
- d. Seleccionar Finalizar.

3. En la ventana de imagen raíz del HMI añadir cada uno de los componentes que se emplea en el diseño de la interfaz gráfica del controlador PID difuso, como se muestra en la Figura 79.

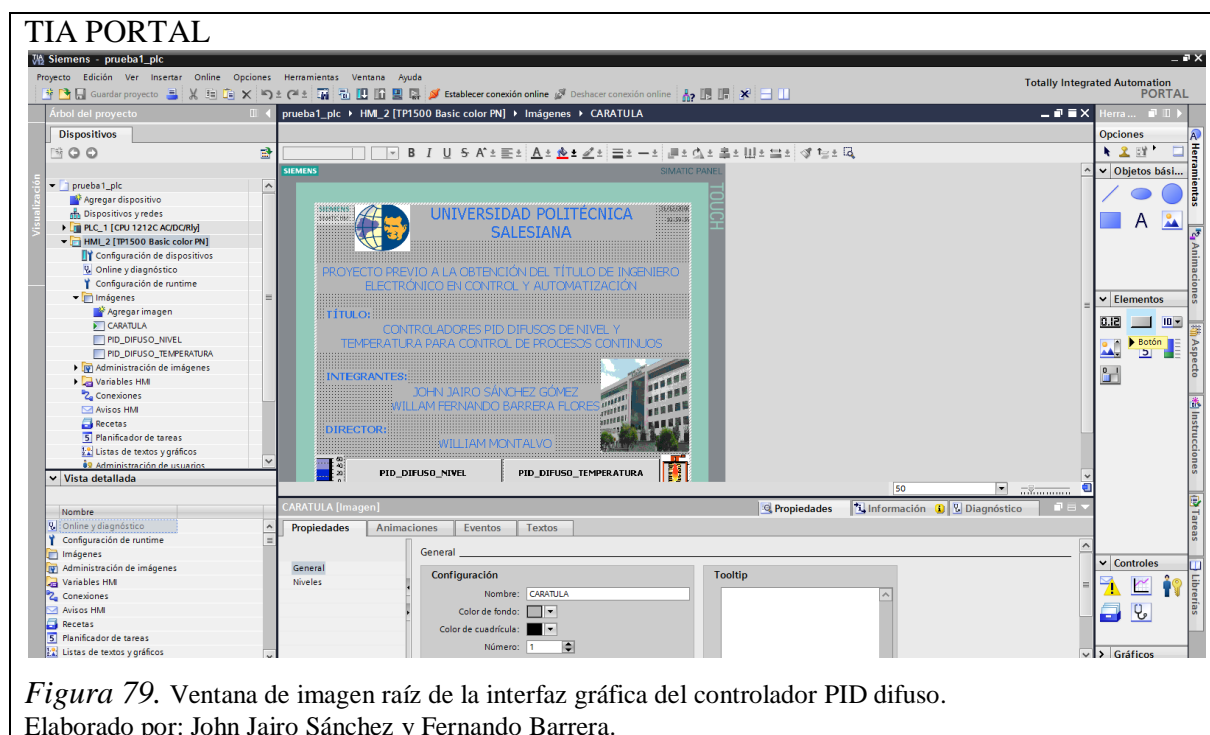
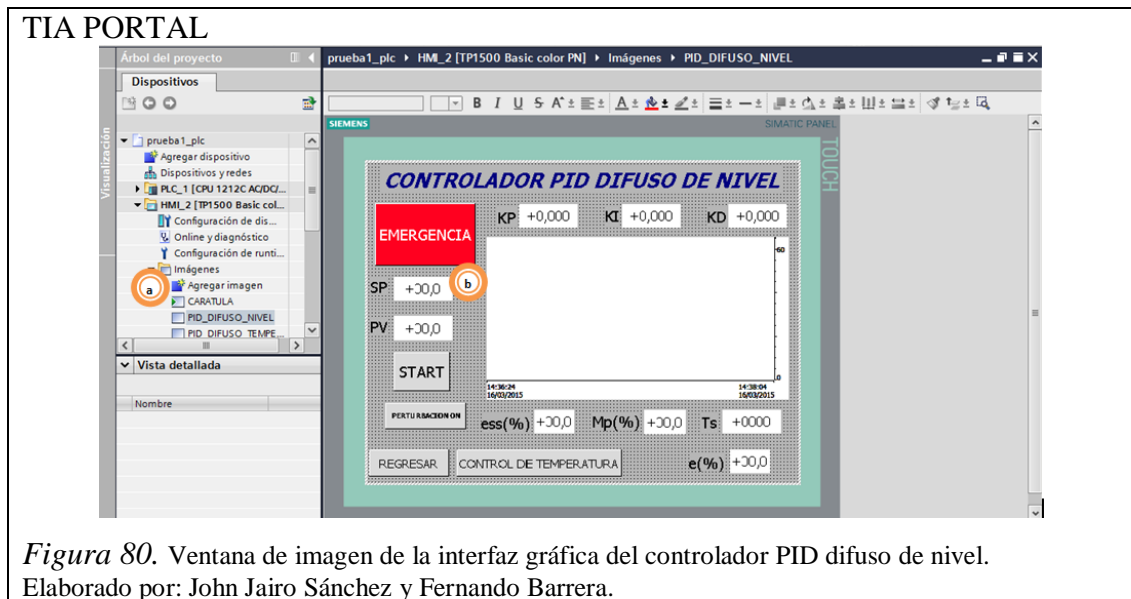


Figura 79. Ventana de imagen raíz de la interfaz gráfica del controlador PID difuso.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

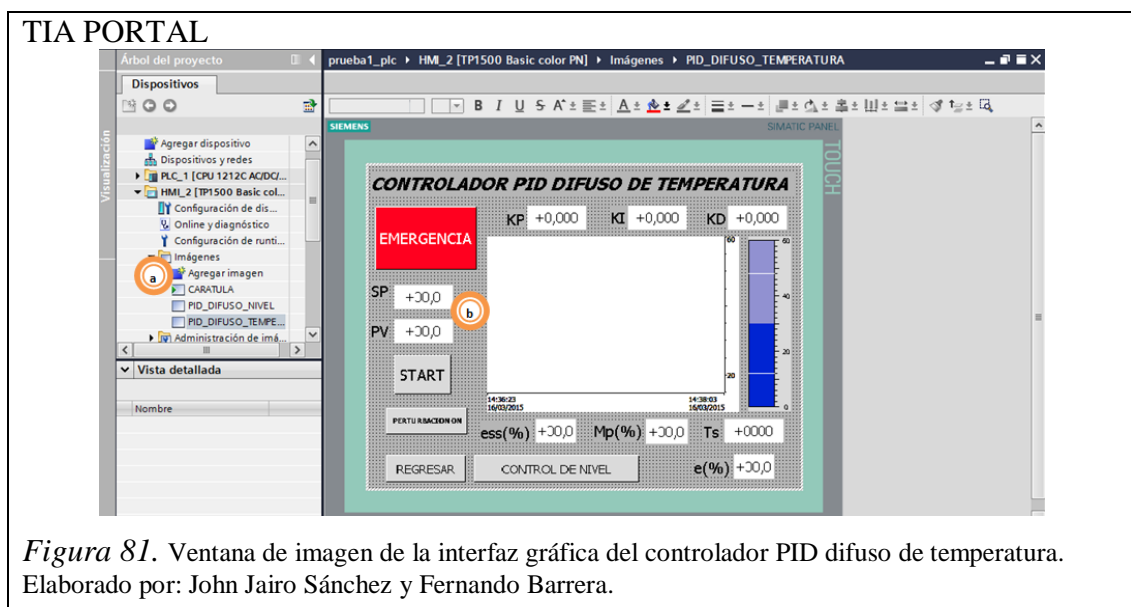
4. Agregar una imagen para la interfaz gráfica controlador PID difuso de nivel, como se muestra en la Figura 80.



Los pasos de la interfaz gráfica de nivel, se detallan a continuación:

- a. En la pestaña del dispositivo HMI dar doble clic izquierdo en agregar imagen.
- b. Añadir cada uno de los componentes que se emplea en el diseño del controlador PID difuso de nivel.

5. Agregar una nueva imagen para la interfaz gráfica controlador PID difuso de temperatura, como se muestra en la Figura 81.



Los pasos de la interfaz gráfica de temperatura, se detallan a continuación:

- En la pestaña del dispositivo HMI dar doble clic izquierdo en agregar imagen.
- Añadir cada uno de los componentes que se emplea en el diseño del controlador PID difuso de temperatura.

3.3 Implementación de los controladores PID difusos

Se procede a la implementación de los controladores PID Difusos tanto para el programa Matlab como para el PLC Siemens S7-1200.

3.3.1 Controladores PID difusos en Matlab

Para la implementación de los controladores PID difusos tanto de nivel como de temperatura, se detalla los siguientes pasos:

- Abrir el programa TIA PORTAL, con el proyecto creado.
- Cargar el proyecto del TIA PORTAL al PLC Siemens S7-1200 y establecer al PLC en línea, como se observa en la Figura 82.

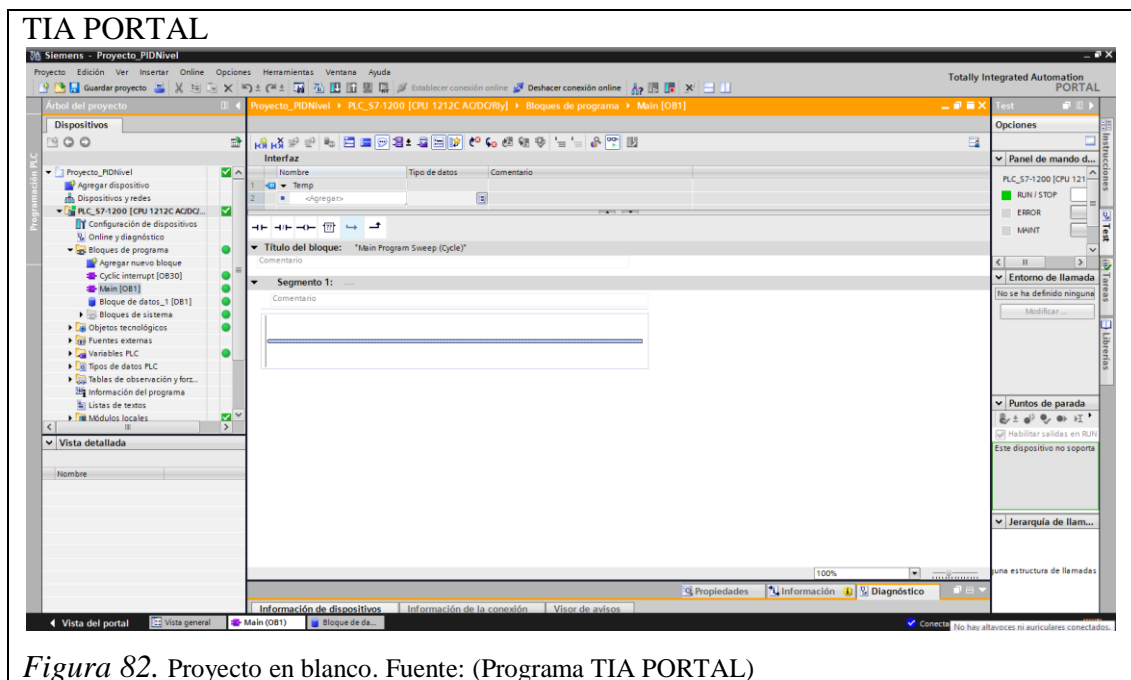


Figura 82. Proyecto en blanco. Fuente: (Programa TIA PORTAL)

3. Abrir el programa KEPServerEx, con el proyecto creado.
4. En el KEPServerEx abrir la ventana OPC Quick Client, para comprobar el funcionamiento de la conexión, como se observa en la Figura 83.

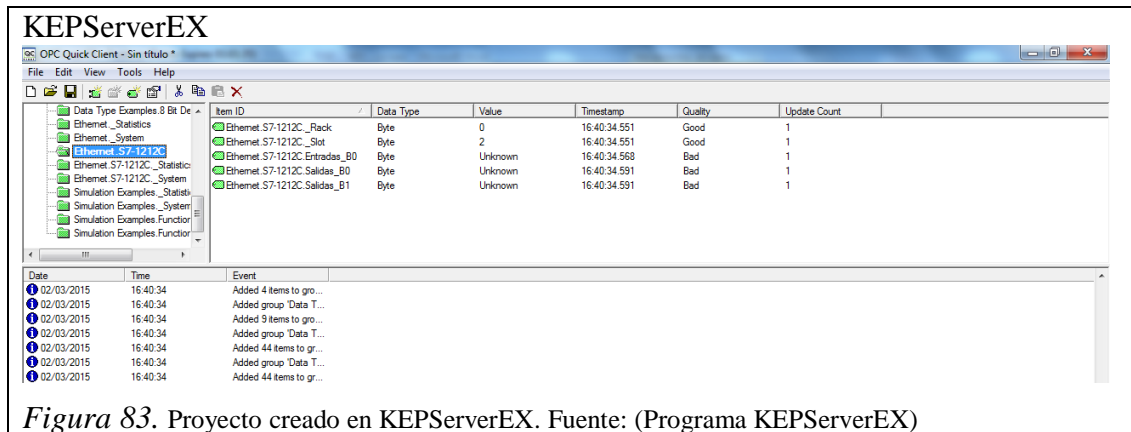


Figura 83. Proyecto creado en KEPServerEX. Fuente: (Programa KEPServerEX)

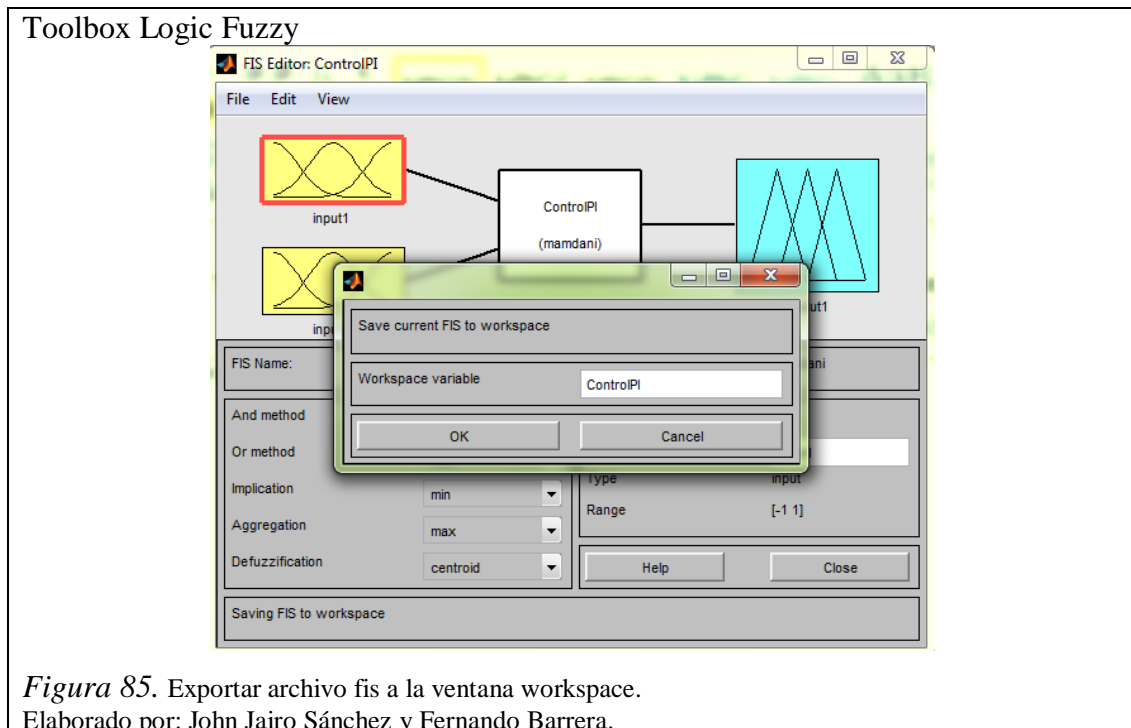
5. Abrir el programa Matlab, en el espacio de trabajo escribir el comando “tesis”, añadiendo automáticamente todas las variables definidas a la ventana Workspace, también desplegando las ventanas de la interfaz gráfica y Fuzzy Logic Toolbox, como se muestra en la Figura 84.



Figura 84. Comando TESIS.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

6. En la ventana de lógica difusa exportar el archivo con la extensión .fis a la ventana Workspace de Matlab, como se muestra en la Figura 85.



7. La interfaz gráfica desplegada llamada CARATULA, permite seleccionar el control a ejecutarse, como se observa en la Figura 86.



Tabla 6.
Elementos de la carátula en el software Matlab

Elemento	Detalle
Controlador PID difuso de nivel	Despliega la ventana del Controlador de nivel
Controlador PID difuso de temperatura	Despliega la ventana del Controlador de temperatura

Nota: Detalles de los elementos que conforma la carátula del controlador PID difuso en Matlab.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

8. La interfaz gráfica verifica el funcionamiento del controlador PID difuso de las variables de nivel y temperatura, como se muestran en las Figuras 87 y 88.



Figura 87. Interfaz Gráfica del controlador PID difuso de nivel.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Tabla 7.
Elementos de la ventana control PID Difuso de nivel en el software Matlab

Elemento	Detalle
SP	Set Point punto de consigna a controlar
INICIAR	Inicia el control de nivel
PARAR	Finaliza el control de nivel
PERT	Activa la perturbación del sistema
Ts(s)	Tiempo de establecimiento
Mp(%)	Máximo sobre impulso
e(%)	Error en estado estable
MEDIR	Despliega los valores Ts, Mp, e.
REGRESAR	Retorna a la venta de Carátula

Nota: Detalles de los elementos que conforma la ventana del controlador PID difuso de nivel en Matlab.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

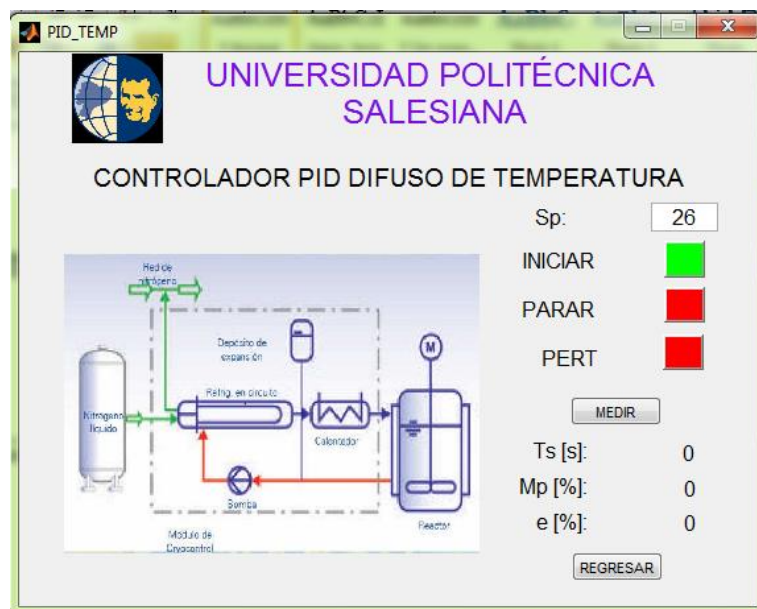


Figura 88. Interfaz gráfica del controlador PID difuso de temperatura.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Tabla 8.

Elementos de la ventana control PID Difuso de temperatura en el software Matlab

Elemento	Detalle
SP	Set Point punto de consigna a controlar
INICIAR	Inicia el control de temperatura
PARAR	finaliza el control de temperatura
PERT	Activa la perturbación del sistema
Ts(s)	Tiempo de establecimiento
Mp(%)	Máximo sobre impulso
e(%)	Error en estado estable
MEDIR	Despliega los valores Ts, Mp, e.
REGRESAR	Retorna a la venta de Carátula

Nota: Detalles de los elementos que conforma la ventana del controlador PID difuso de temperatura en Matlab.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Diagrama de Flujo del controlador PID difuso en Matlab

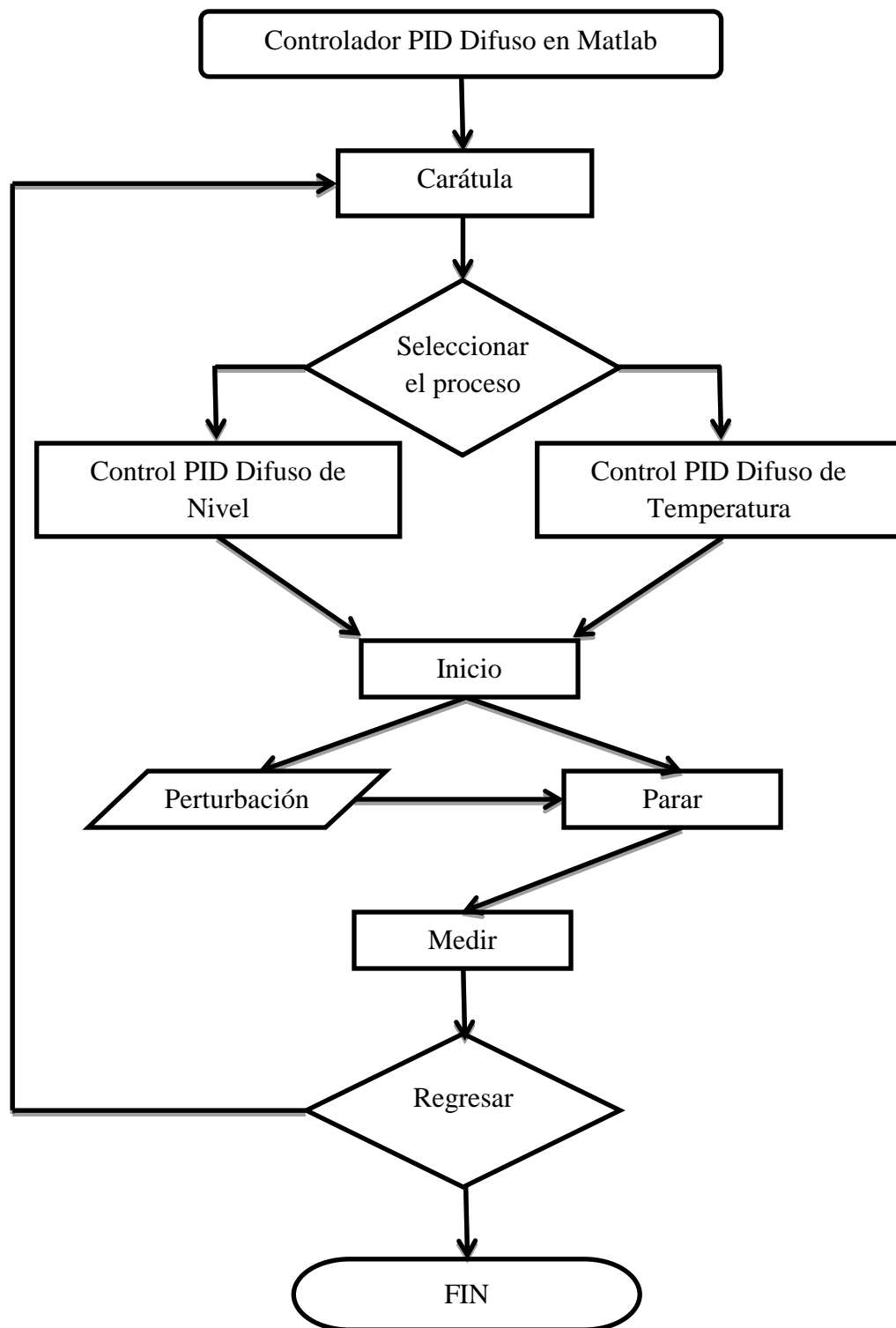
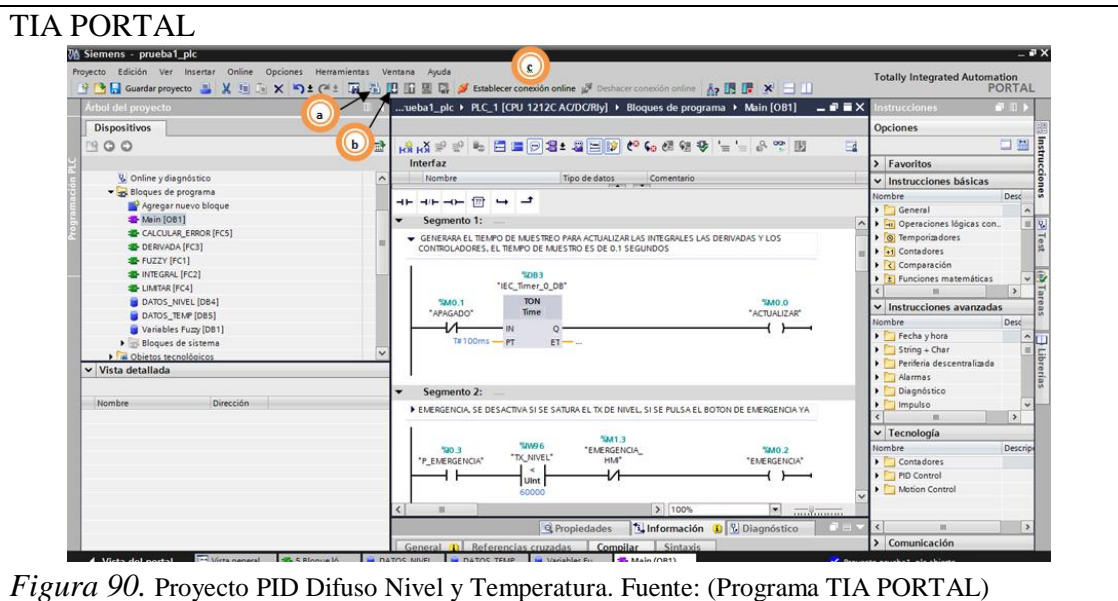


Figura 89. Flujo del proceso para el control del PID Difuso con Matlab
Elaborado por: John Jairo Sánchez y Fernando Barrera.

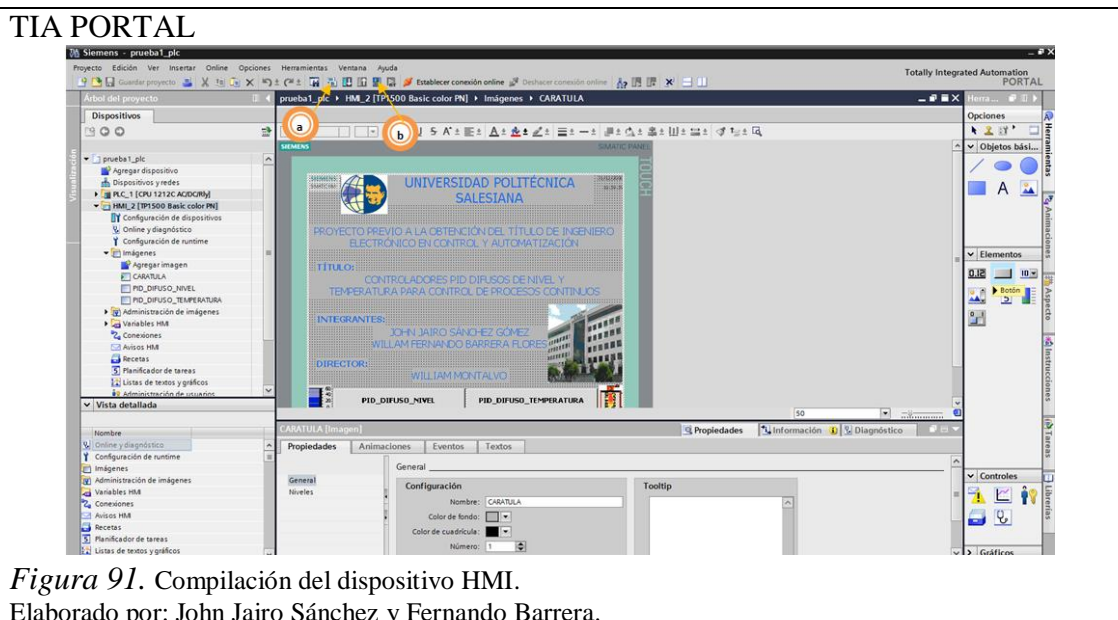
3.3.2 Controladores PID difusos en el PLC siemens S7-1200

Para la implementación de los controladores PID difusos tanto de nivel como de temperatura, se detalla los siguientes pasos:

1. Abrir el programa TIA PORTAL, con el proyecto creado.
2. Compilar y cargar el proyecto del TIA PORTAL al PLC Siemens S7-1200 y establecer al PLC en línea, como se muestra en la Figura 90.



3. Compilar la interfaz gráfica del dispositivo HMI, luego iniciar la simulación del HMI, como se muestra en la Figura 91.



- La interfaz gráfica desplegada llamada CARATULA, permite seleccionar el control a ejecutarse, como se observa en la Figura 92.



Tabla 9.
Elementos de la carátula en el software TIA PORTAL

Elemento	Detalle
PID Difuso de Nivel	Despliega la ventana del controlador PID difuso de nivel
PID Difuso de Temperatura	Despliega la ventana del controlador PID difuso de temperatura

Nota: Detalles de los elementos que conforma la carátula del controlador PID difuso en el PLC.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

- La interfaz gráfica verifica el funcionamiento del controlador PID difuso de las variables de nivel y temperatura, como se muestran en las Figuras 93 y 94.

TIA PORTAL

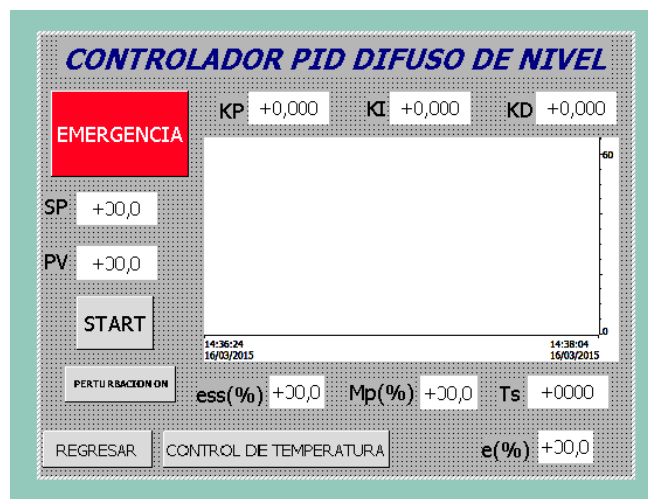


Figura 93. Interfaz Gráfica del controlador PID difuso de Nivel.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Tabla 10.

Elementos de la ventana del controlador PID difuso de nivel en el software TIA PORTAL

Elemento	Detalle
EMERGENCIA	Para todo el sistema
KP	Ganancia proporcional
KI	Ganancia integral
KD	Ganancia derivativa
SP	Set Point punto de consigna a controlar
START	Inicia o detiene el control de nivel
PERTURBACIÓN	Activa la perturbación del sistema
Ts(s)	Tiempo de establecimiento
Mp(%)	Máximo sobre impulso
ess(%)	Error en estado estable
CONTROL DE TEMPERATURA	Despliega la ventana del controlador PID difuso de temperatura
REGRESAR	Retorna a la ventana de Carátula

Nota: Detalles de los elementos que conforma la ventana del controlador PID difuso de nivel en TIA PORTAL.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

TIA PORTAL

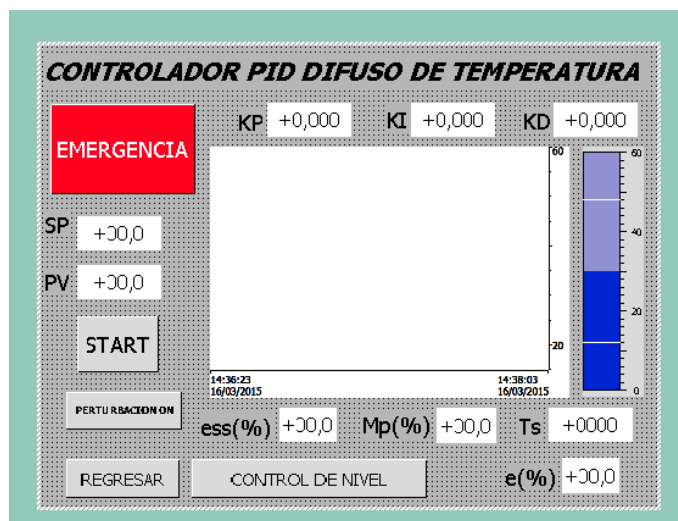


Figura 94. Interfaz Gráfica del controlador PID difuso de temperatura.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Tabla 11.

Elementos de la ventana del controlador PID difuso de temperatura en el software TIA PORTAL

Elemento	Detalle
EMERGENCIA	Para todo el sistema
KP	Ganancia proporcional
KI	Ganancia integral
KD	Ganancia derivativa
SP	Set Point punto de consigna a controlar
START	Inicia o detiene el control de temperatura
PERTURBACIÓN	Activa la perturbación del sistema
Ts(s)	Tiempo de establecimiento
Mp(%)	Máximo sobre impulso
ess(%)	Error en estado estable
CONTROL DE NIVEL	Despliega la ventana del controlador PID difuso de nivel
REGRESAR	Retorna a la venta de Carátula

Nota: Detalles de los elementos que conforma la ventana del controlador PID difuso de temperatura en TIA PORTAL.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Diagrama de Flujo del controlador PID difuso en el PLC Siemens S7-1200

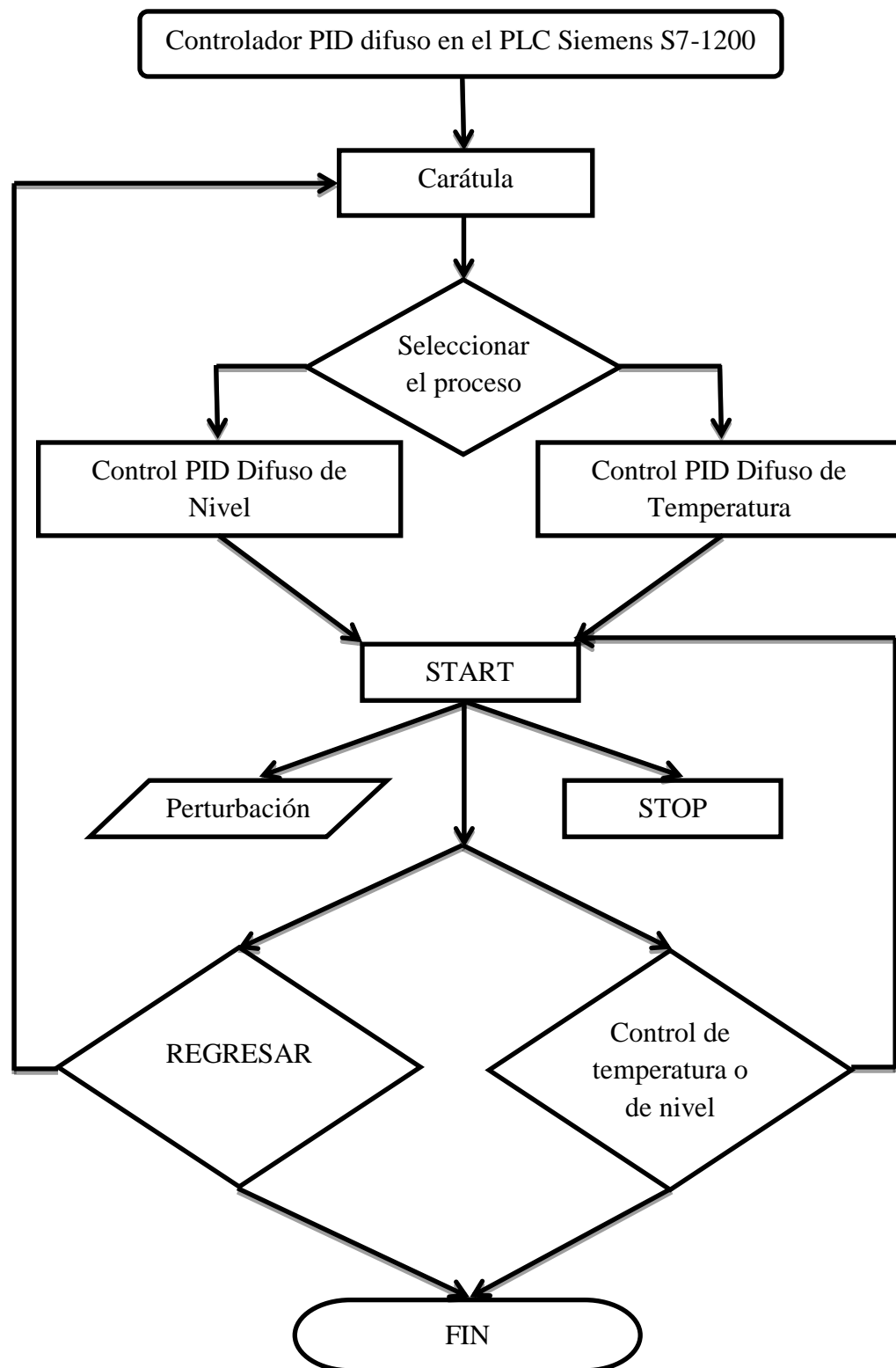


Figura 95. Flujo del proceso para el control del PID difuso con el PLC Siemens S7-1200
Elaborado por: John Jairo Sánchez y Fernando Barrera.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

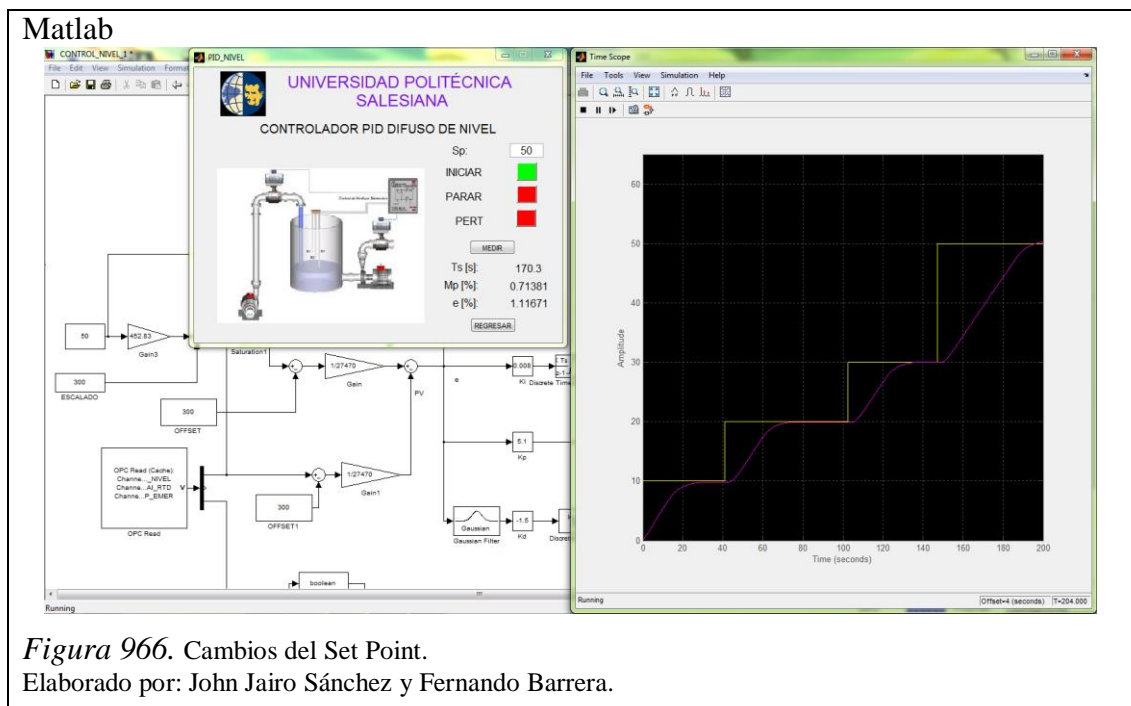
Este capítulo detalla las pruebas realizadas al controlador PID difuso de las variables de nivel y temperatura, para evaluar el comportamiento del controlador en el programa Matlab y en el PLC Siemens S7-1200.

4.1 Pruebas del controlador PID difuso de nivel

Las pruebas que se realizan tanto para el Matlab como para el PLC Siemens S7-1200 son la respuesta del sistema al cambio de la señal del Set Point.

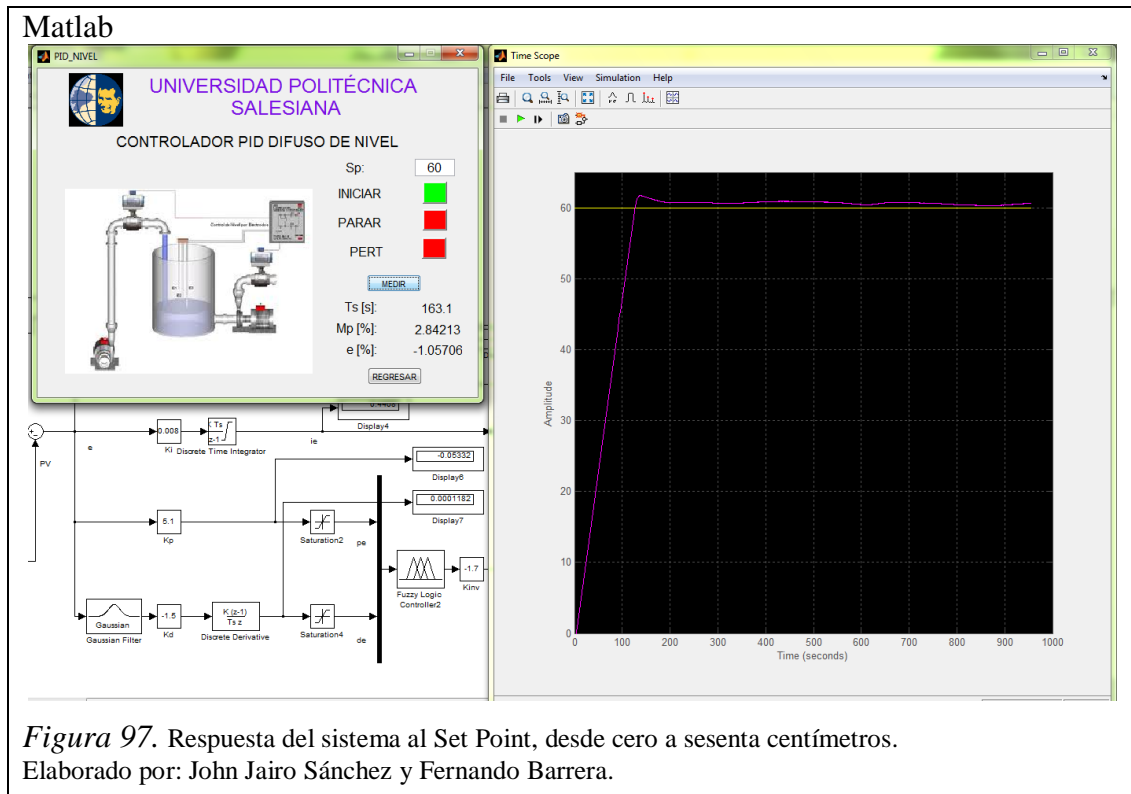
4.1.1 Respuesta a la señal del Set Point en Matlab

La prueba consiste en observar la respuesta del sistema en lazo cerrado ante diferentes cambios del Set Point, los resultados obtenidos se muestran en la Figura 96.



Como se observa en la Figura 96, el actuador cambia su estado para compensar el nuevo valor del Set Point.

Los valores del Set Point son tomados desde diferentes puntos, es decir, desde cero a un valor máximo, del valor máximo a cero, desde un valor medio al valor máximo. Los resultados obtenidos se muestran a continuación.



Como se observa en la Figura 97, el Set Point arranca desde cero hasta sesenta centímetros, la señal de salida responde adecuadamente, con un pequeño sobreimpulso y un pequeño error en estado estable, lo que corrobora la eficiencia del controlador PID difuso; también se observa un tiempo de establecimiento, que es el tiempo que tarda la señal real en alcanzar a la señal del Set Point.

Matlab

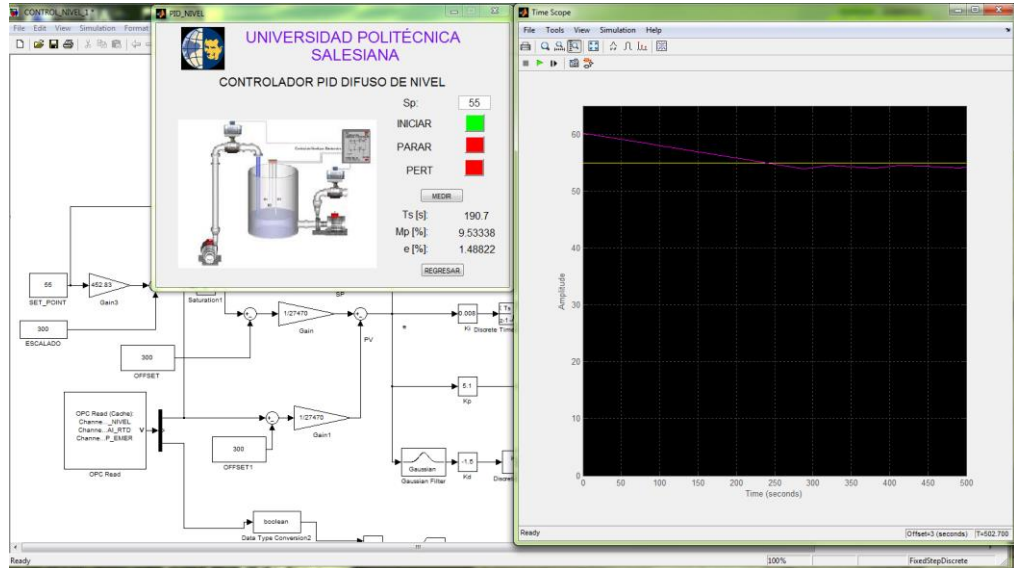


Figura 98. Respuesta del sistema al Set Point, desde sesenta a cincuenta y cinco centímetros.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 98, el Set Point arranca desde sesenta hasta cincuenta y cinco, la señal de salida responde adecuadamente, no existe sobre-impulso y un pequeño error en estado estable, pero se observa un tiempo de establecimiento muy alto, debido a que el consumo de líquido es pequeño.

Matlab

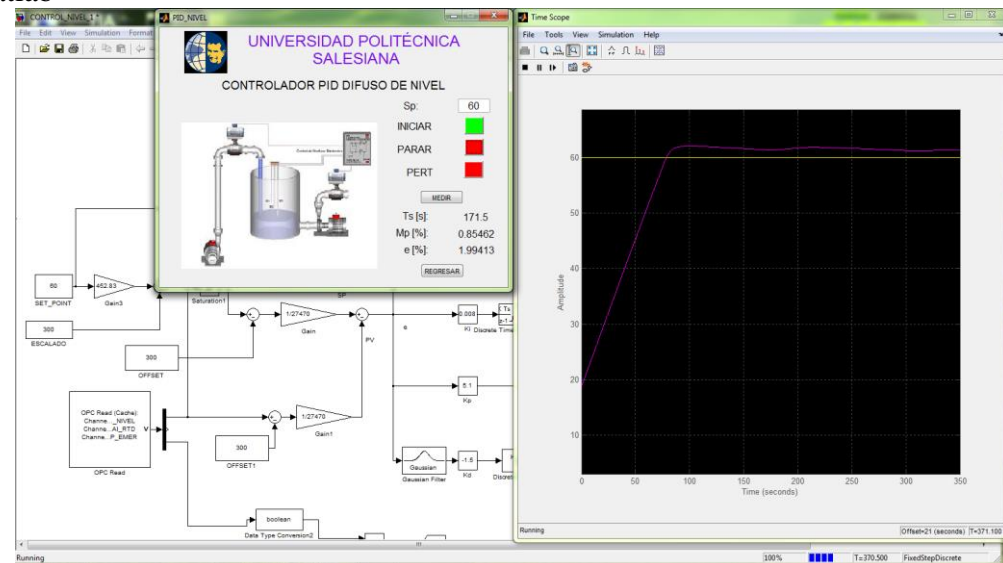


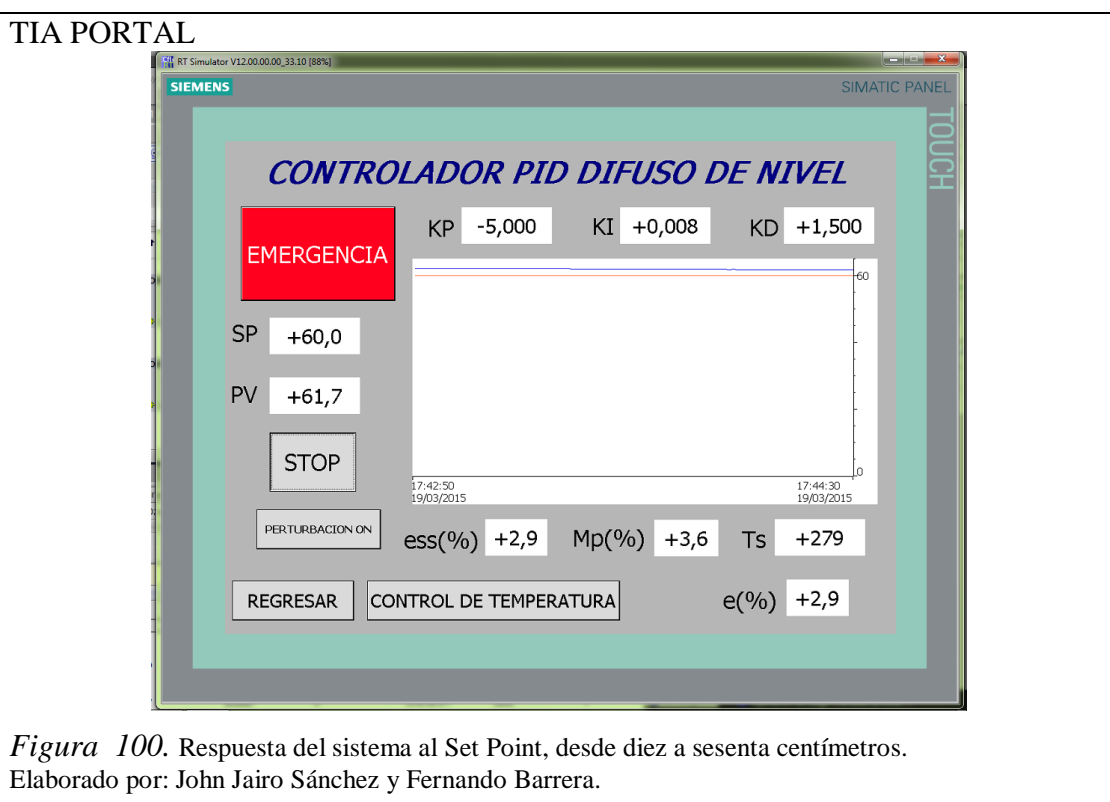
Figura 99. Respuesta del sistema al Set Point, desde veinte a sesenta centímetros.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 99, el Set Point arranca desde veinte hasta sesenta centímetros, la señal de salida responde adecuadamente, con un pequeño sobreimpulso y un pequeño error en estado estable; se observa un tiempo de establecimiento alto.

4.1.2 Respuesta a la señal del Set Point en el PLC

La prueba consiste en observar la respuesta del sistema en lazo cerrado ante diferentes cambios del Set Point.

Los valores del Set Point son tomados desde diferentes puntos, es decir, desde cero a un valor máximo, del valor máximo a cero, desde un valor medio al valor máximo. Los resultados obtenidos se muestran a continuación.



Como se observa en la Figura 100, el Set Point arranca desde diez hasta sesenta centímetros, la señal de salida responde adecuadamente, existe un pequeño sobreimpulso y un pequeño error en estado estable; también se observa un tiempo de establecimiento muy alto.



Figura 101. Respuesta del sistema al Set Point, desde sesenta a cincuenta y cinco centímetros.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 101, el Set Point arranca desde sesenta hasta cincuenta y cinco, la señal de salida responde adecuadamente, no existe sobre-impulso y un pequeño error en estado estable; pero se observa un tiempo de establecimiento muy alto, debido a que el consumo de líquido es pequeño.

4.1.3 Resultados del controlador PID difuso de nivel

Las pruebas que se realizan al controlador PID Difuso de nivel permiten calcular el máximo sobre-impulso, error en estado estable y el tiempo de establecimiento de la señal de salida.

4.1.3.1 Respuesta a la señal del Set Point en Matlab

Las pruebas realizadas permiten calcular los valores del tiempo de establecimiento, máximo sobre-impulso y el error en estado estable de la señal de salida, como se muestra en la Tabla 12.

Tabla 12.

Resultado de las pruebas realizadas y cálculo de la mediana

Pruebas	Set Point	Ts(s)	Mp%	e%
Prueba 1	0cm – 10cm	173.3(s)	0%	2%
Prueba 2	10cm – 20cm	105(s)	0%	2%
Prueba 3	20cm – 40cm	45.8(s)	0%	2%
Prueba 4	40cm – 60cm	43.3(s)	0%	2%
Prueba 5	60cm – 55cm	252.6(s)	0%	2%
MEDIANA		105(s)	0%	2%

Nota: Abreviatura del tiempo de establecimiento, máximo sobre-impulso y error.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

En el Anexo 2, se muestran en gráficas las pruebas realizadas del controlador PID difuso de la variable de nivel para el Matlab.

4.1.3.2 Respuesta a la señal del Set Point en el PLC

Las pruebas realizadas permiten calcular los valores del tiempo de establecimiento, máximo sobre-impulso y error en estado estable de la señal de salida, como se muestra en la Tabla 13.

Tabla 13.

Resultado de las pruebas realizadas y cálculo de la mediana

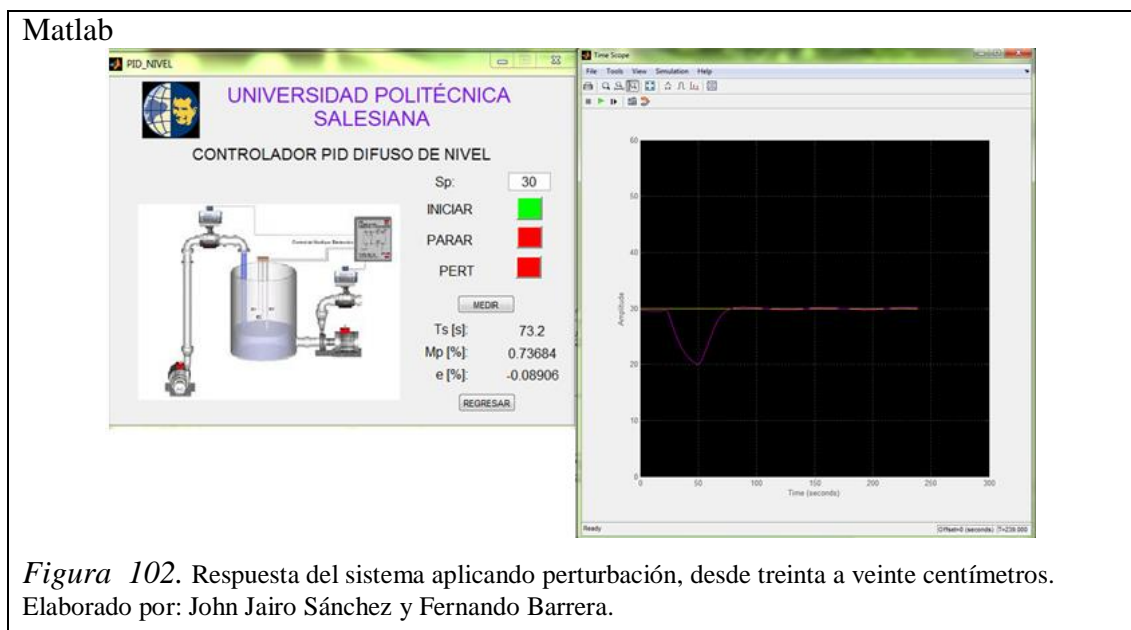
Pruebas	Set Point	Ts(s)	Mp%	e%
Prueba 1	0cm – 10cm	126(s)	0%	2%
Prueba 2	10cm – 20cm	61(s)	0%	2%
Prueba 3	20cm – 40cm	59(s)	0%	2%
Prueba 4	40cm – 60cm	54(s)	0%	2%
Prueba 5	60cm – 55cm	253(s)	0%	2%
MEDIANA		61(s)	0%	2%

Nota: Abreviatura del tiempo de establecimiento, máximo sobre-impulso y error.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

En el Anexo 3, se muestran en gráficas las pruebas realizadas del controlador PID difuso de la variable de nivel para el PLC Siemens S7-1200.

4.1.4 Respuesta a la señal aplicando perturbación en Matlab

La perturbación es realizada de forma manual es decir, al momento que el sistema esté controlado, se aplica un desfogue de líquido y se observa que el sistema compensa y estabiliza el nivel. Los resultados obtenidos se muestran en la Figura 102.



Como se observa en la Figura 102, la perturbación arranca desde treinta centímetros y baja a veinte centímetros, la señal responde adecuadamente, existe un pequeño sobre-impulso y un pequeño error en estado estable; también se observa un tiempo de establecimiento corto.

4.1.5 Respuesta a la señal aplicando perturbación en el PLC

La perturbación es realizada de forma manual es decir, al momento que el sistema esté controlado, se aplica un desfogue de líquido y se observa que el sistema compensa y estabiliza el nivel. Los resultados obtenidos se muestran en la Figura 103.

TIA PORTAL



Figura 103. Respuesta del sistema aplicando perturbación, desde treinta a veinte centímetros
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 103, la perturbación arranca desde treinta centímetros y baja a veinte centímetros, la señal responde adecuadamente, no existe sobreimpulso y un pequeño error en estado estable; también se observa un tiempo de establecimiento corto.

4.2 Pruebas del controlador PID difuso de temperatura

Las pruebas que se realizan tanto para el Matlab como para el PLC Siemens S7-1200 son la respuesta del sistema al cambio de la señal del Set Point.

4.2.1 Respuesta a la señal del Set Point en Matlab

Los valores del Set Point son tomados desde diferentes puntos, es decir, desde un valor alto a un valor medio, del valor medio a un valor alto. Los resultados obtenidos se muestran a continuación.

Matlab

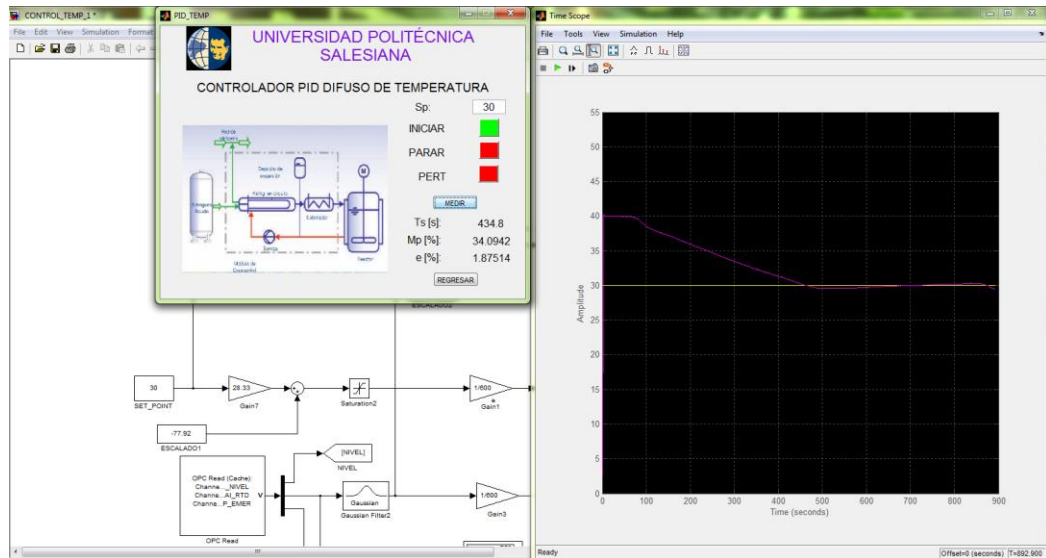


Figura 104. Respuesta del sistema al Set Point, desde cuarenta a treinta grados centígrados.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 104, el Set Point arranca desde cuarenta y baja a treinta grados centígrados, la señal de salida responde adecuadamente, no existe sobre-impulso y un pequeño error en estado estable; también se observa un tiempo de establecimiento muy alto, debido a que la única forma enfriar el líquido es sacar el líquido caliente e ingresando líquido a temperatura ambiente.

Matlab

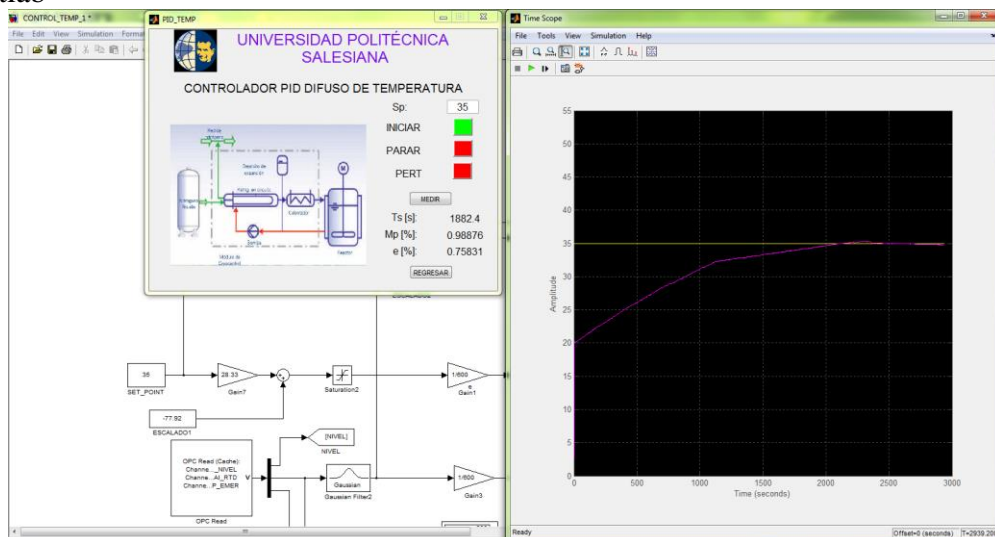


Figura 105. Respuesta del sistema al Set Point, desde veinte a treinta y cinco grados centígrados.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 105, el Set Point arranca desde veinte hasta treinta y cinco grados centígrados, la señal de salida responde adecuadamente, existe un pequeño sobre-impulso y un pequeño error en estado estable, lo que corrobora la eficiencia del controlador PID difuso; pero se observa un tiempo de establecimiento muy alto, debido a que la potencia de la niquelina no es muy eficiente para la cantidad del líquido utilizado.

4.2.2 Respuesta a la señal del Set Point en el PLC

Los valores del Set Point son tomados desde diferentes puntos, es decir, desde un valor alto a un valor medio, del valor medio a un valor alto. Los resultados obtenidos se muestran a continuación.

TIA PORTAL

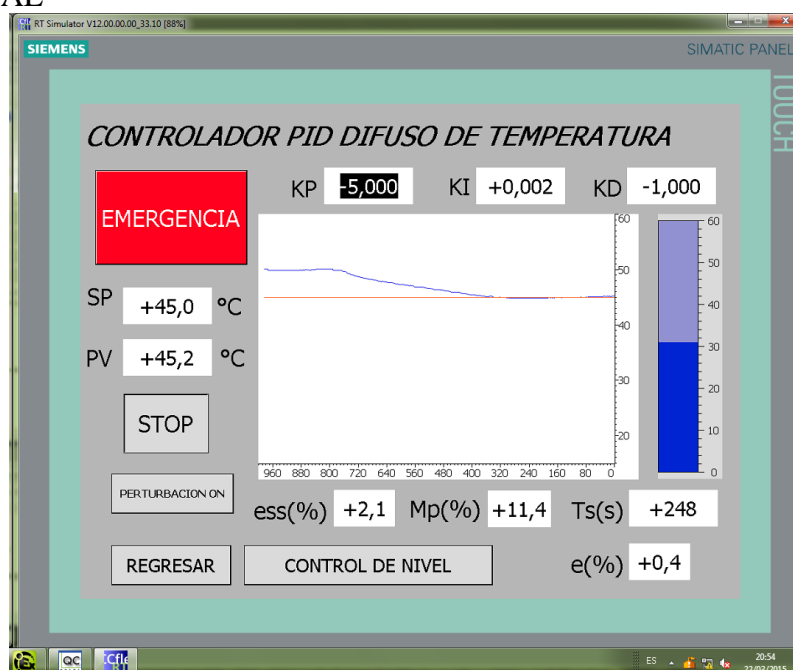


Figura 106. Respuesta del sistema al Set Point, desde cincuenta a cuarenta y cinco grados centígrados.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 106, el Set Point arranca desde cincuenta y baja a cuarenta y cinco grados centígrados, la señal de salida responde adecuadamente, no existe sobre-impulso y un pequeño error en estado estable; también se observa un tiempo de establecimiento muy alto, debido a que la única forma enfriar el líquido es sacar el líquido caliente e ingresar líquido a temperatura ambiente.

TIA PORTAL

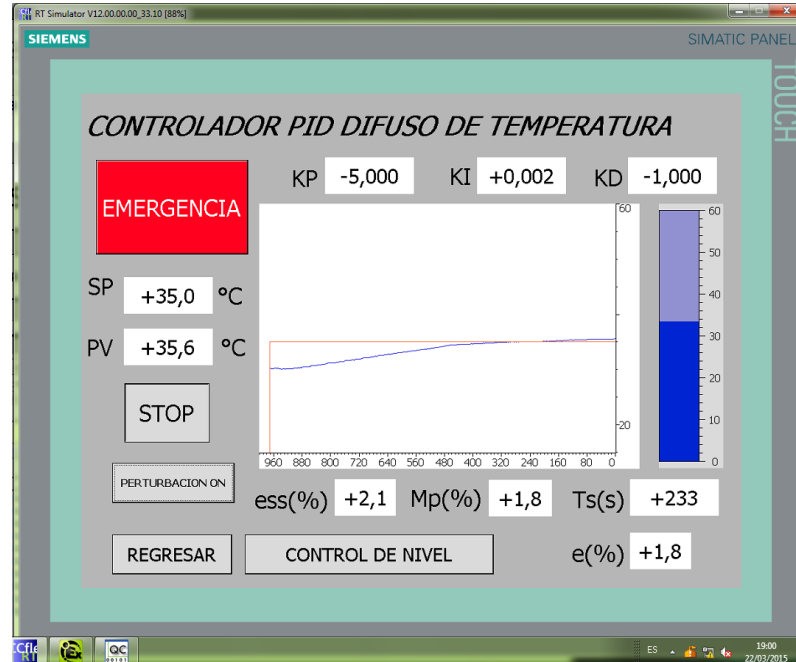


Figura 107. Respuesta del sistema al Set Point, desde treinta a treinta y cinco grados centígrados. Elaborado por: John Jairo Sánchez y Fernando Barrera.

Como se observa en la Figura 107, el Set Point arranca desde treinta hasta treinta y cinco grados centígrados, la señal de salida responde adecuadamente, existe un pequeño sobre-impulso y un pequeño error en estado estable; pero se observa un tiempo de establecimiento muy alto, debido a que la potencia de la niquelina no es muy eficiente para la cantidad del líquido utilizado.

De los datos obtenidos se calcula la mediana de cada uno de ellos, para la comparación del controlador PID difuso realizado en el programa Matlab y en el PLC Siemens S7-1200.

La mediana se usa cuando los valores de los datos no son muy confiables, como son valores de transmisores análogos. Debido a que la planta está diseñada con transmisores análogos se realiza el cálculo de la mediana.

Sí se calcula el promedio, que es una medida que aprovecha todos los valores, se obtendrá un valor errado. En cambio sí se calcula la mediana como medida de tendencia central o posición, el valor será confiable.

4.2.3 Resultados del controlador PID difuso de temperatura

Las pruebas que se realizan al controlador PID difuso de temperatura permiten calcular el máximo sobre-impulso, error en estado estable y el tiempo de establecimiento de la señal de salida.

4.2.3.1 Respuesta a la señal del Set Point en Matlab

Las pruebas realizadas permiten calcular los valores del tiempo de establecimiento, máximo sobre-impulso y error en estado estable de la señal de salida, como se muestra en la Tabla 14.

Tabla 14.

Resultado de las pruebas realizadas y cálculo de la mediana

Pruebas	Set Point	Ts(s)	Mp%	e%
Prueba 1	0cm – 10cm	731(s)	4.65%	2%
Prueba 2	10cm – 20cm	576(s)	3.93%	2%
Prueba 3	20cm – 40cm	957.9(s)	2.62%	2%
Prueba 4	40cm – 60cm	944.3(s)	0%	2%
Prueba 5	60cm – 55cm	3760.5(s)	0%	2%
MEDIANA		944.3(s)	2.62%	2%

Nota: Abreviatura del tiempo de establecimiento, máximo sobre-impulso y error.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

En el Anexo 4, se muestran en gráficas las pruebas realizadas del controlador PID difuso de la variable de temperatura para el Matlab.

4.2.3.2 Respuesta a la señal del Set Point en el PLC

Las pruebas realizadas permiten calcular los valores del tiempo de establecimiento, máximo sobre-impulso y error en estado estable de la señal de salida, como se muestra en la Tabla 15.

Tabla 15.
Resultado de las pruebas realizadas y cálculo de la mediana

Pruebas	Set Point	Ts(s)	Mp%	e%
Prueba 1	0cm – 10cm	709(s)	3.8%	2%
Prueba 2	10cm – 20cm	381(s)	3.1%	2.1%
Prueba 3	20cm – 40cm	881(s)	2.1%	2.1%
Prueba 4	40cm – 60cm	864(s)	0%	2.2%
Prueba 5	60cm – 55cm	1743(s)	0%	2%
MEDIANA		881(s)	2.1%	2%

Nota: Abreviatura del tiempo de establecimiento, máximo sobre-impulso y error.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

En el Anexo 5, se muestran en gráficas las pruebas realizadas del controlador PID difuso de la variable de temperatura para el PLC Siemens S7-1200.

4.2.4 Respuesta a la señal aplicando perturbación en Matlab y en el PLC

La plataforma de entrenamiento para control de procesos continuos, consta de una niquelina de 1500 vatios para la aplicación de disturbio, al instante que la niquelina se activa como disturbio el controlador PID difuso actúa sobre el sistema y lo enfría rápidamente, no se puede observar variaciones en la señal; por esta razón se determina que la plataforma no consta de un actuador que genere un disturbio significativo.

4.3 Análisis de resultados

Las pruebas realizadas en la plataforma de entrenamiento para el control de procesos continuos presentan los resultados mostrados en las Tablas 12, 13, 14, 15 obteniendo el error porcentual en estado estable, el tiempo de establecimiento en segundos y el máximo sobre impulso.

4.3.1 Comparación de los controladores PID difusos de nivel

La comparación se realiza entre en controlador PID difuso de nivel del Matlab con el Controlador del PLC Siemens S7-1200.

Tabla 16.

Tabla de comparación de los dos tipos de controladores en Matlab y en el PLC Siemens S7-1200

Controlador PID Difuso	Ts(s)	Mp%	e%
MATLAB	105(s)	0%	2%
PLC	61(s)	0%	2%

Nota: Los datos obtenidos son la mediana de las 5 pruebas realizadas en las tablas 6,7, 8, y 9

4.3.2 Comparación de los controladores PID difusos de temperatura

La comparación se realiza entre en controlador PID difuso de temperatura del Matlab con el Controlador del PLC Siemens S7-1200.

Tabla 17.

Tabla de comparación de los dos tipos de controladores en Matlab y en el PLC Siemens S7-1200

Controlador PID Difuso	Ts(s)	Mp%	e%
MATLAB	944.3(s)	2.62%	2%
PLC	881(s)	2.1%	2%

Nota: Los datos obtenidos son la mediana de las 5 pruebas realizadas en las tablas 6,7, 8, y 9

Dada la complejidad de la planta, no resulta sencillo determinar si la dinámica obtenida con el controlador PID difuso en el PLC o con el controlador PID difuso en Matlab es el adecuado, motivo por el cual se realiza una comparación entre la respuesta del sistema ajustado con control PID difuso en el PLC y la respuesta con el controlador PID difuso en Matlab, lógicamente bajo las mismas condiciones. Esto puede ser apreciado en la Figura 108, donde se observa que ambos métodos presentan dinámicas muy similares, validando como consecuencia la calidad del resultado obtenido con el Control PID Difuso en el PLC.

Dinámica de respuesta de los controladores PID difusos

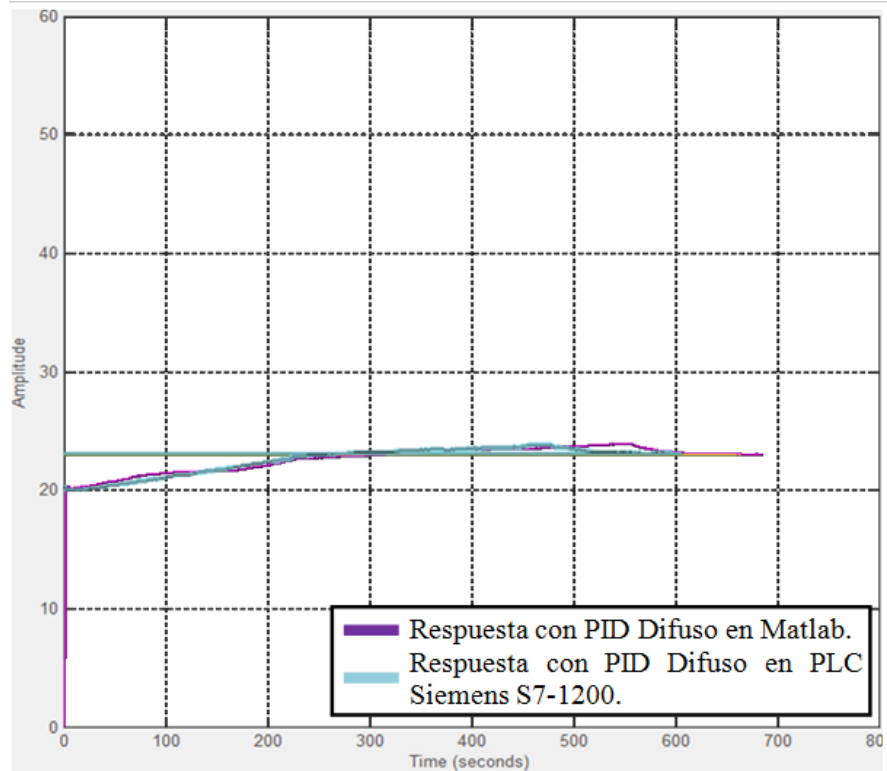


Figura 108. Comparación de la dinámica de respuesta entre el controlador PID difuso en el programa Matlab y el controlador PID difuso en el PLC Siemens S7-1200.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

CONCLUSIONES

- Como conclusión principal del presente proyecto de titulación es de haber cumplido satisfactoriamente el objetivo principal de diseñar controladores PID difusos de las variables de nivel y temperatura de una plataforma de entrenamiento para control de procesos continuos, tanto en el programa Matlab como en el PLC Siemens S7-1200.
- Como resultado al diseño de controladores PID difusos de la variable de nivel en el programa Matlab y en el PLC Siemens S7-1200, se cumple que el control PID difuso provee una respuesta rápida y estable para el sistema, cuando éste es sometido a diferentes variaciones del Set Point.
- Los controladores PID difusos de la variable de temperatura en el programa Matlab y en el PLC Siemens S7-1200, presentan una respuesta con un pequeño sobre-impulso, tiempo alto en alcanzar al Set Point, estado estacionario oscilatorio de muy pequeña amplitud, pero de poca incidencia en las acciones de control; debido a que la planta consta de una niquelina de baja potencia como actuador y la variable en si es de respuesta lenta.
- El diseño del controlador PID difuso realizado para la variable de nivel se asemeja para la variable de temperatura, ya que el principio de funcionamiento emplea la misma lógica de control en ambos procesos.
- El controlador PID difuso al no necesitar el modelo del proceso a controlar, permite una gran facilidad de implementación porque incorpora un lenguaje común al diseño de sistemas de control.
- Para el ajuste del controlador PID difuso se empleó el método empírico de prueba y error, ya que da la posibilidad de modificar el desempeño del controlador al variar las ganancias K_p , K_i y K_d .

- El programa Matlab 2012 contiene todas las herramientas para diseñar y simular, de manera estática y dinámica, un controlador PID difuso.
- Las herramientas Toolbox Fuzzy Logic simula el controlador PID difuso en una forma continua.
- La comparación de los resultados obtenidos en los dos controladores PID difusos de nivel y temperatura, realizados en Matlab y en el PLC Siemens S7-1200, demuestra que el control realizado en el PLC es más rápido que el control realizado en el programa Matlab y a su vez tiene un sobre impulso menor.
- Se realizó un manual técnico de prácticas que permite al estudiante o docente de la Universidad Politécnica Salesiana de la carrera de Ingeniería Electrónica, realizar pruebas de funcionamiento de los controladores PID difusos de los procesos de nivel y temperatura para verificar el control en la plataforma de procesos continuos (Anexo 6).
- En el control PID difuso se observa que la salida del controlador no presenta grandes oscilaciones alrededor del Set Point, ya que por trabajar de manera más humana (inteligente), va alcanzando el resultado mediante incremento o decremento, desde un valor alto hasta cada vez valores más pequeños que le permiten alcanzar el valor deseado; sin esos saltos bruscos que se ve en el control tradicional.
- El algoritmo de programación para el controlador PID difuso de las variables de nivel y temperatura es más fácil implementarlo en el programa Matlab que en el PLC Siemens S7-1200; ya que el programa Matlab posee el Toolbox Fuzzy Logic, donde permite crear sistemas de lógica difusa de forma intuitiva, debido a que tiene programación gráfica en todos los elementos que conforman al sistema.

RECOMENDACIONES

- Para obtener un eficiente controlador PID difuso de la variable de nivel, se debe calibrar el transmisor de nivel respecto al escalamiento de los datos mínimos y máximos del tanque que almacena el líquido de la planta, con el fin de evitar errores en la lectura de la variable de entrada.
- Se recomienda complementar las prácticas de laboratorio en el módulo de entrenamiento procesos continuos, usando el manual técnico de prácticas del diseño de controladores PID difusos para la plataforma de entrenamiento de procesos continuos; para reforzar los conocimientos de los estudiantes.
- Al no existir una metodología que indique con exactitud la cantidad de términos lingüísticos necesarios dentro de una variable lingüística, en términos generales es recomendable tener entre 5 y 11 términos lingüísticos, tanto para la variable de entrada como para la variable de salida.
- Para realizar una interfaz gráfica con el programa Matlab, se requiere estar familiarizado con la interfaz gráfica de usuario GUIDE de Matlab.
- Al trabajar con los elementos de campo principales que posee la plataforma de procesos continuos, como son los actuadores; la servo válvula para el control de nivel y la tarjeta de control de la niquelina para el control de temperatura, se debe asegurar el nivel de voltaje asignado para realizar la medida o control: de 1 – 5VDC. De no ser así, el equipo está expuesto al deterioro o daño irreversible de los elementos principales y más costosos de la plataforma de entrenamiento para control de procesos continuos.
- Cuando se trabaje con transmisores o actuadores, se debe consultar que señal de control soporta o envía, ya sea señal de tensión o corriente; para establecer los parámetros de tensión o corriente en las entradas y salidas análogas del PLC Siemens S7-1200.

- Para el desarrollo de las pruebas de los controladores PID difusos de las variables de nivel y temperatura, se recomienda leer el manual de uso y mantenimiento de la plataforma de entrenamiento para control de procesos continuos, para establecer las conexiones básicas con las que se debe inicializar el funcionamiento de la plataforma.
- Para la comunicación entre el programa Matlab y el PLC Siemens S7-1200 se emplea el OPC KEPServerEX, evitando buscar e instalar drivers del programa Matlab para dialogar con múltiples fuentes de datos, en este caso el PLC Siemens S7-1200.

LISTA DE REFERENCIAS

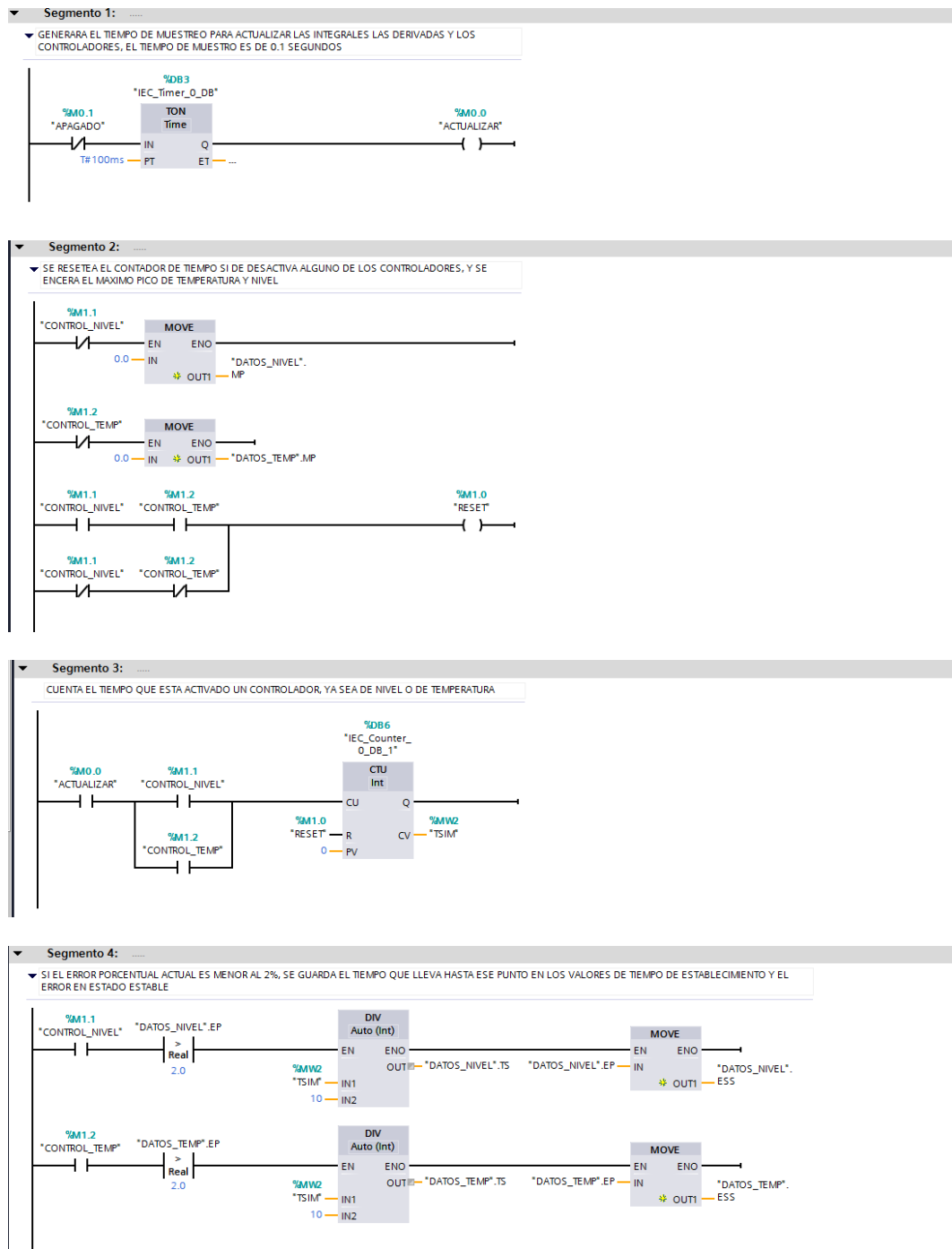
- Acevedo, J. (2007). *Comunicación entre Simulink y sistemas de control*. Recuperado el 16 de Septiembre de 2014, de <http://bibing.us.es/proyectos/abreproy/11477/fichero/Memoria%252F00.pdf>
- Adewuyi, P. (2013). *DC Motor Speed Control: A Case between PID Controller and Fuzzy Logic Controller*. Recuperado el 23 de Junio de 2014, de <http://www.ijmse.org/Volume4/Issue4/paper6.pdf>
- ARIAN. (2014). *Operación del Pt100*. Recuperado el 15 de Octubre de 2014, de <http://www.arian.cl/downloads/nt-004.pdf>
- Aström, K. & Hägglund, T. (2009). *Control PID Avanzado*. Madrid: Prentice Hall Hispanoamericana, S.A.
- Bolton, W. (2013). *Mecatrónica. Sistemas de Control Electrónico en la Ingeniería Mecánica y Eléctrica*. México: Alfaomega Grupo Editor, S.A.
- Cooper, W. & Helfrick, A. (2010). *Instrumentación Electrónica Moderna y Técnicas de Medición*. Mexico: Prentice Hall Hispanoamericana, S.A.
- Creus, A. (2011). *Instrumentación Industrial*. México: Alfaomega Grupo Editor, S.A.
- Creus, A. (2011). *Instrumentación Industrial*. México D.F.: Alfaomega Grupo Editor, S.a.
- Equipo de comunicación del Centro para la elaboración de trabajos de grado. (2014). *Ejemplos de citas y referencias bajo las normas APA 6ta. Edición*. Recuperado el 14 de Enero de 2015, de <https://trabajodegradoups.files.wordpress.com/2014/05/manual-prc3a1ctico-de-apa.pdf>
- Gonzalez, E. (2010). *Control fuzzy pid*. Recuperado el 23 de Junio de 2014, de <http://www.buenastareas.com/ensayos/Control-Fuzzy-Pid/208703.html>
- Henao, C. (2013). *Tutorial Simatic S7-1200*. Recuperado el 6 de Noviembre de 2014, de <https://www.youtube.com/watch?v=a9SlvCNoGE8>
- InfoPLC. (2011). *Kepserver OPC*. Recuperado el 15 de Octubre de 2014, de <http://www.infoplcn.net/descargas/13-omron/821-scada-cx-superviso-con-servidor-opc-keppure-kepserverex>
- Juárez, L. & Álvarez, A. (2011). *Comunicación OPC para Monitoreo de Datos Analógicos en Tiempo Real*. Recuperado el 15 de Septiembre de 2014, de <http://www.mecamex.net/anterior/cong10/trabajos/art04.pdf>

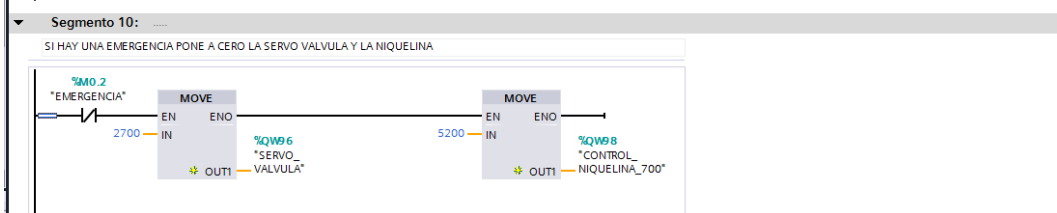
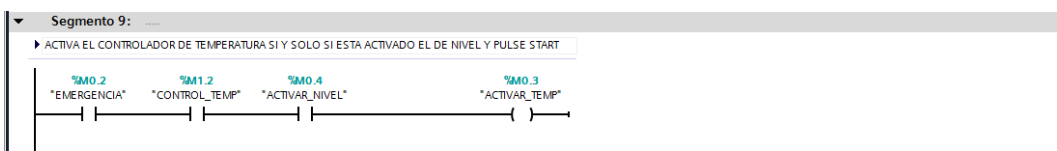
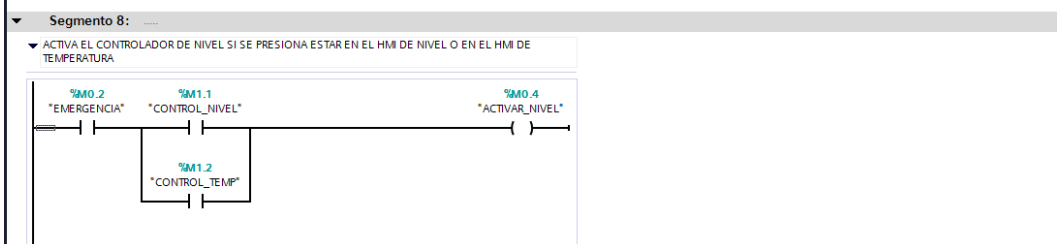
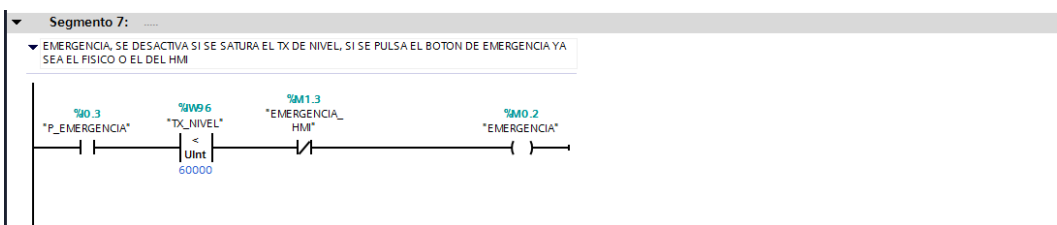
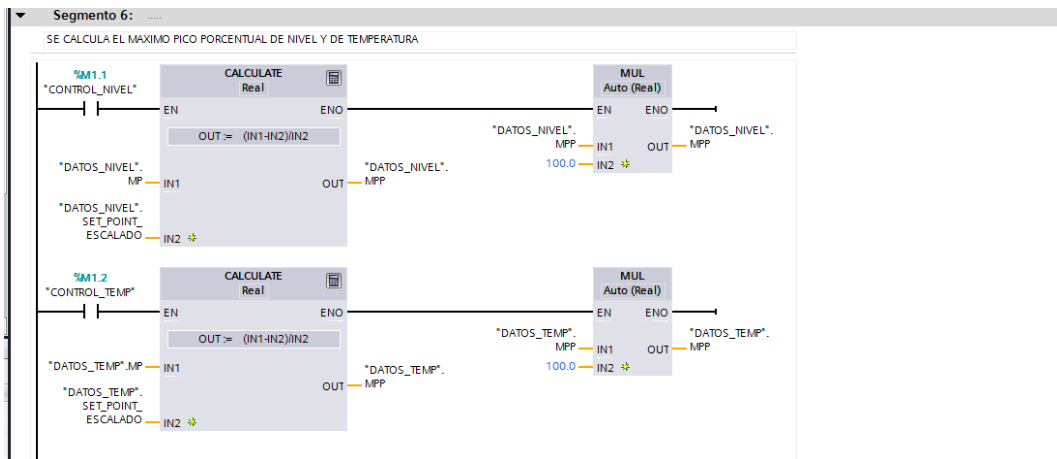
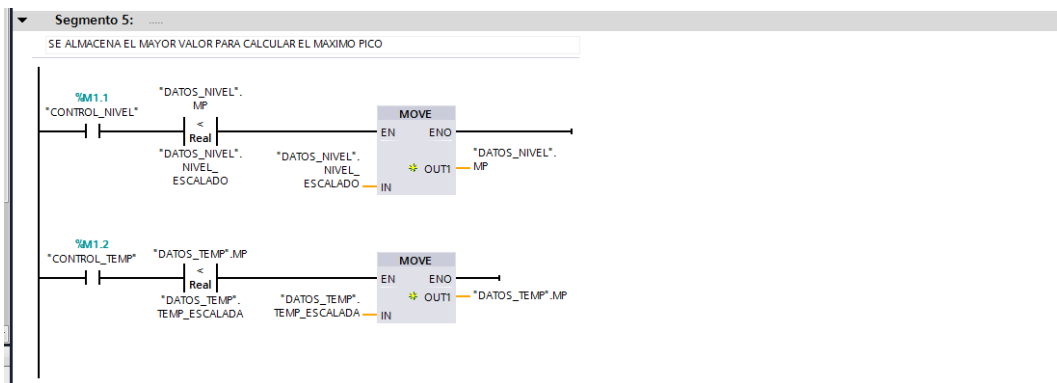
- Kepware, Technologies. (2014). *OPC Server Siemens*. Recuperado el 15 de Octubre de 2014, de http://www.kepware.com/Spec_Sheets/Siemens_S7_200300400_TCPIPEthernet.asp
- Kepware, Technologies. (2014). *Siemens TCP/IP Ethernet Driver Help*. Recuperado el 15 de Octubre de 2014, de <http://www.kepware.com/support/manuals/siemens-tcp-ip-ethernet-manual.pdf>
- Kouro, S. & Musalem, R. (2002). *Control Mediante Lógica Difusa*. Recuperado el 23 de Junio de 2014, de <http://www2.elo.utfsm.cl/~elo377/documentos/Fuzzy.pdf>
- Kuo, B. (2013). *Sistemas de control automático*. México: Prentice Hall Hispanoamericana, S.A.
- Lorandi, A. (2011). *Controladores PID y Controladores Difusos*. Recuperado el 16 de Junio de 2014, de <http://academiajournals.com/downloads/Lorandi2011IE.pdf>
- Mandado, E., Marcos, J., Fernandez, C. & Armesto, J. (2009). *Autómatas programables y sistemas de automatización*. México: Alfaomega Grupo Editio, S.A.
- Martín del Brío, B. & Molina, A. (2007). *Redes Neuronales y Sistemas Borrosos*. México: Alfaomega Grupo Editor, S.A.
- Martín, J. (2013). *Comunicación de un OPC con un controlador S7-1200*. Recuperado el 3 de Octubre de 2014, de <https://www.youtube.com/watch?v=j3cLFLBMW6Q>
- Martins, J. &. (2010). *A Matlab/Simulink framework for PLC controlled processes*. Recuperado el 16 de Septiembre de 2014, de <http://www.intechopen.com/books/matlab-modelling-programming-and-simulations/plc-control-and-matlab-simulink-simulations-a-translation-approach>
- Matlab. (2013). *Fuzzy Logic Controller in Simulink*. Recuperado el 3 de Octubre de 2014, de https://www.youtube.com/watch?v=Jbh3oEsI3_E&index=403&list=UUgdHSFcXvkN6O3NXvif0-pA
- Matlab. (2013). *Getting Started with Fuzzy Logic Toolbox*. Recuperado el 3 de Octubre de 2014, de <https://www.youtube.com/watch?v=O348HnWPm7A>
- Mazzone, V. (2002). *Controladores PID*. Recuperado el 16 de Junio de 2014, de <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>

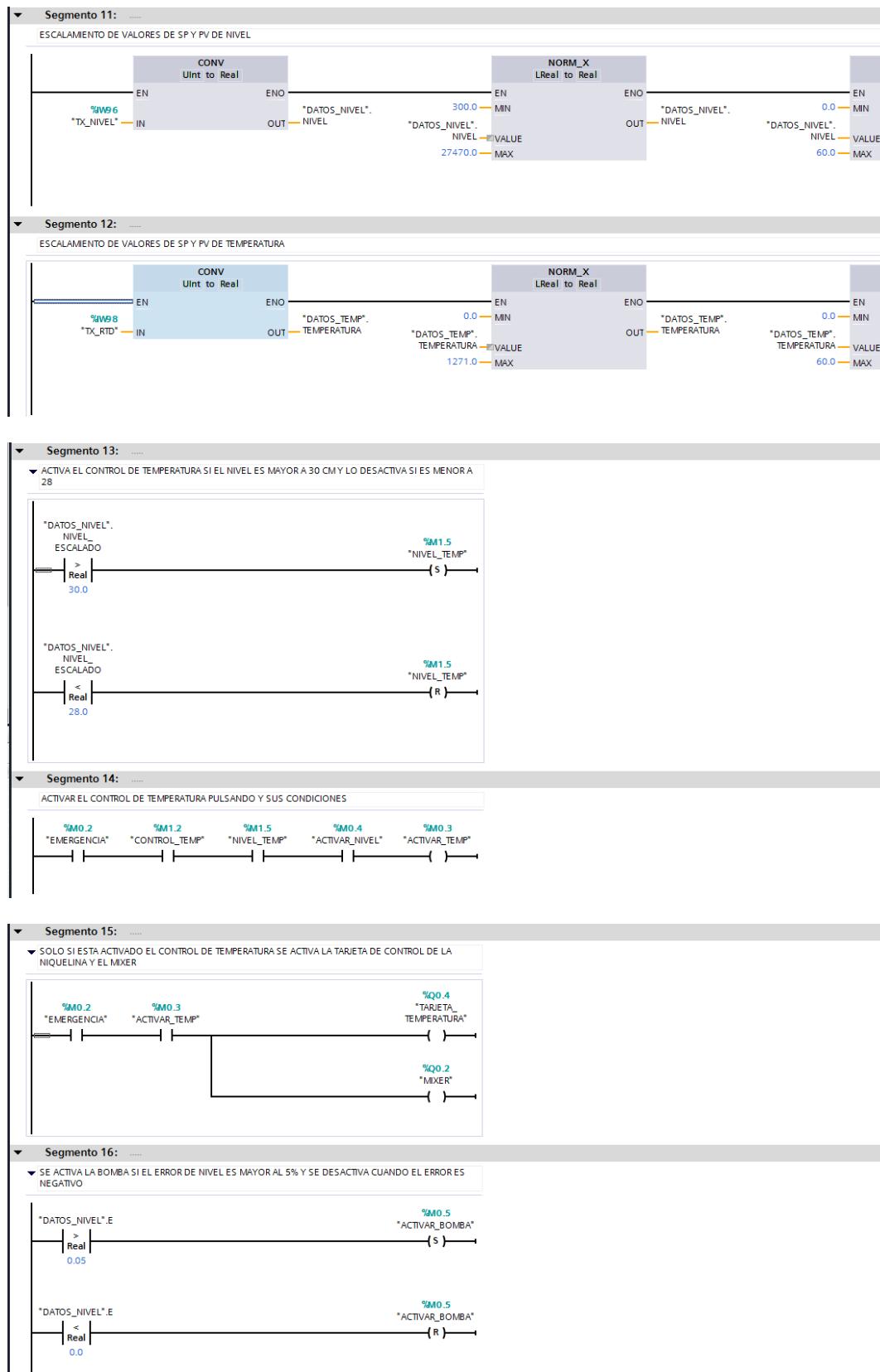
- Mora, J., García, P. & Ardanuy, J. (2013). *Instrumentación Aplicada a la Ingeniería*. Madrid: Ibergarceta Publicaciones, S.L.
- Mosvel, E. (s.f.). *Control Difuso*. Recuperado el 23 de Junio de 2014, de <http://es.scribd.com/doc/3200501/Control-Difuso>
- Ogata, K. (2010). *Ingeniería de Control Moderna*. Madrid: Prentice Hall Hispanoamericana, S.A.
- Pablo, U. S. (2006). *Interfaces de usuario en Matlab*. Recuperado el 28 de Octubre de 2014, de <http://biocomp.cnb.csic.es/~coss/Docencia/MATLAB/ApuntesGrande.pdf>
- Pérez, M. & Vernon, J. (2014). *Sistemas de Lógica Difusa*. Recuperado el 16 de Junio de 2014, de <http://control-systems-principles.co.uk/whitepapers/spanishwp/13fuzzylogicSP.pdf>
- Pinto, E.& Espada, F. (2010). *Fundamentos de Control con Matlab*. Madrid: Prentice Hall Hispanoamericana, S.A.
- PLC, P. (2010). *Configuración de panel HMI*. Recuperado el 6 de Noviembre de 2014, de <https://www.youtube.com/watch?v=mC8CAySNUNg&list=PLBE1493484BD99F5&index=6>
- Ponce, P. (2010). *Inteligencia Artificial con Aplicaciones a la Ingeniería*. México: Alfaomega Grupo Editor, S.A.
- Reyes, F. (2012). *Matlab Aplicado a Robótica y Mecatrónica*. México: Alfaomega Grupo Editor, S.A.
- Ruiz, M. (2012). *Análisis e Implementación de un Control Fuzzy PID de iluminación usando Matlab*. Recuperado el 16 de Septiembre de 2014, de <http://es.slideshare.net/diegomejia230290/fuzzy-pid>
- Siemens. (2012). *Manual PLC Siemens S7-1200*. Recuperado el 6 de Noviembre de 2014, de <https://www.swe.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200%20Manual%20Sistema%20Abr12.pdf>
- Siemens, Simulink. (2013). *Integration of Simatic S7-1200 with the Matlab Simulink environment Configuration*. Recuperado el 3 de Octubre de 2014, de <https://www.youtube.com/watch?v=PNXhHBIXywc>
- UNIFE. (2014). *Tuning of Fuzzy PID Controllers*. Recuperado el 28 de Octubre de 2014, de <http://www.unife.it/ing/lm.infoauto/tecnico-controllo/FuzzyPIDTuning.pdf>

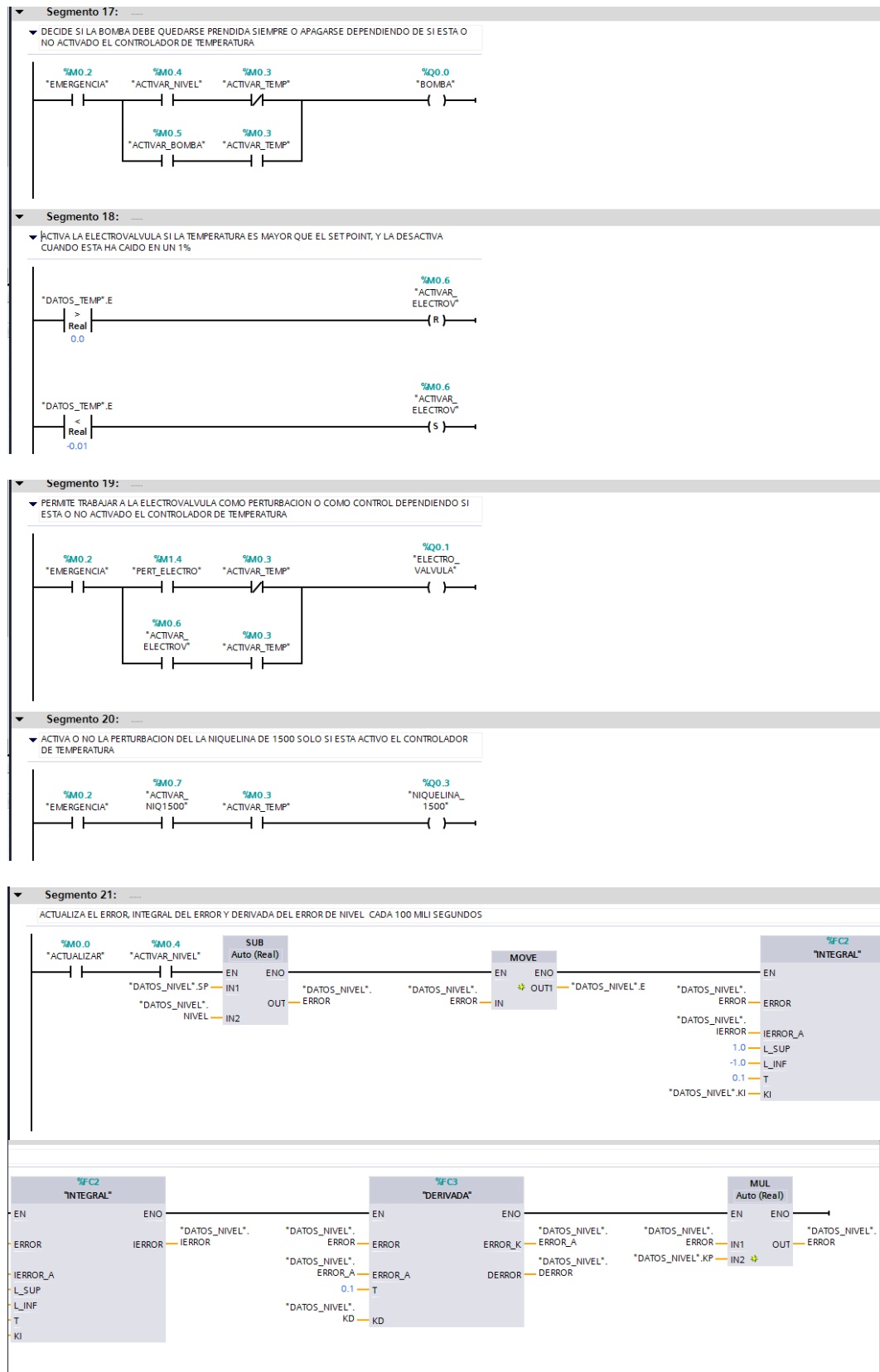
ANEXOS

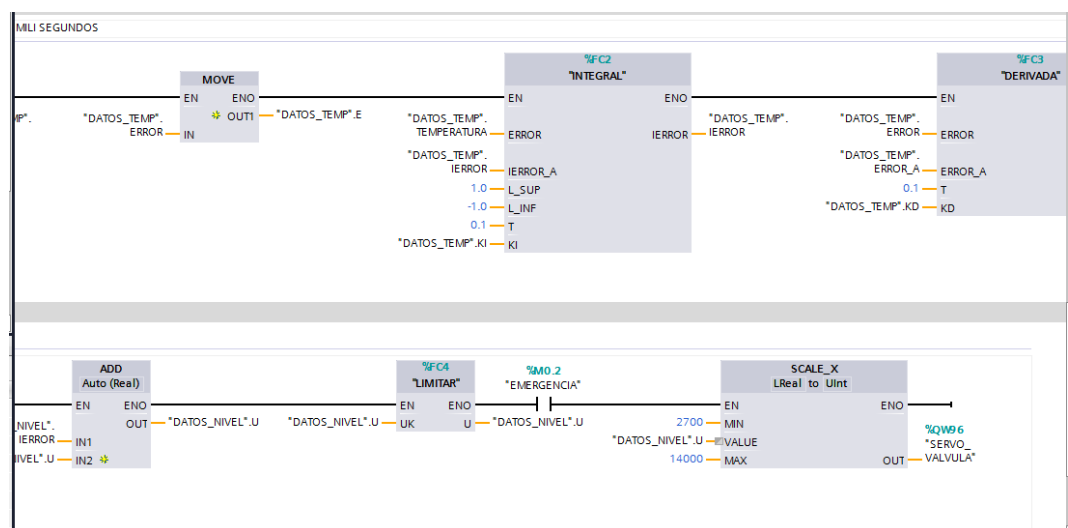
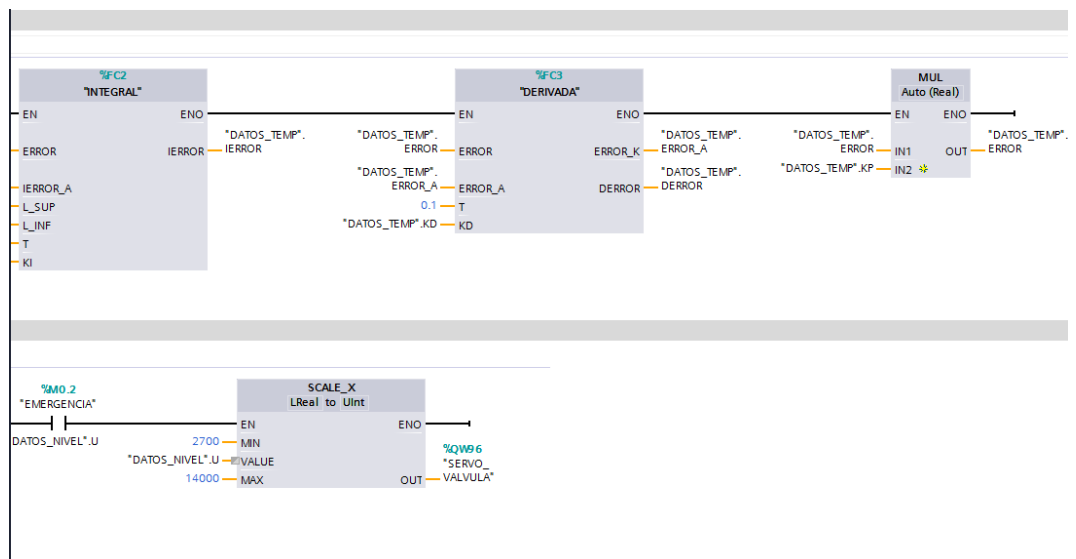
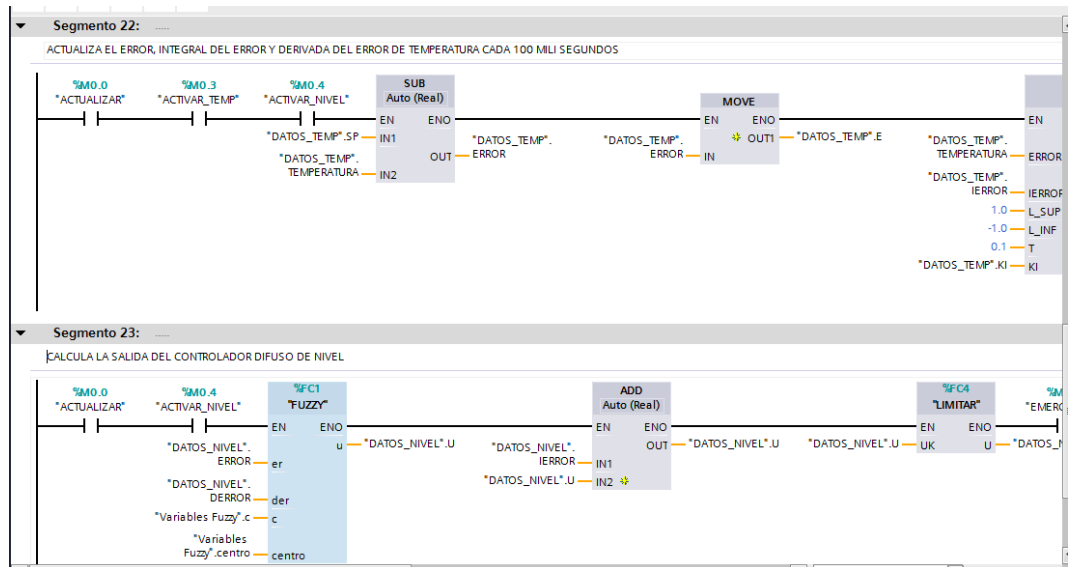
Anexo 1. Programación del controlador PID difuso en el PLC Siemens S7-1200.

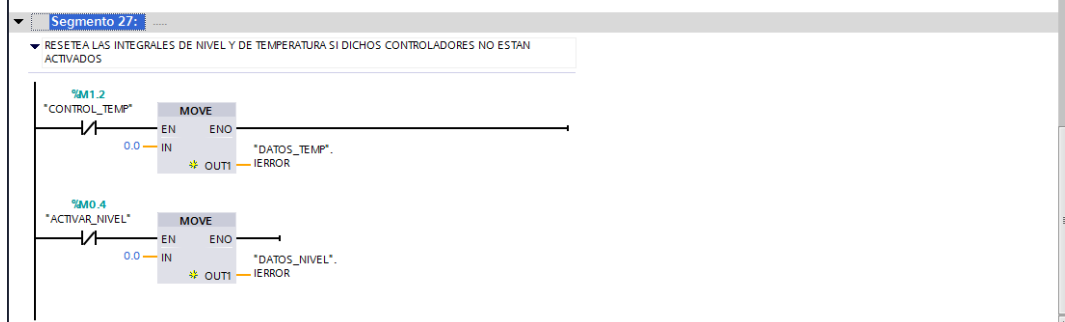
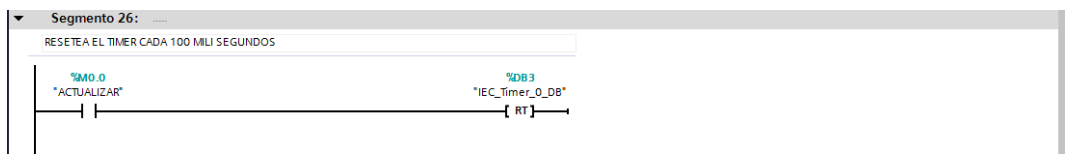
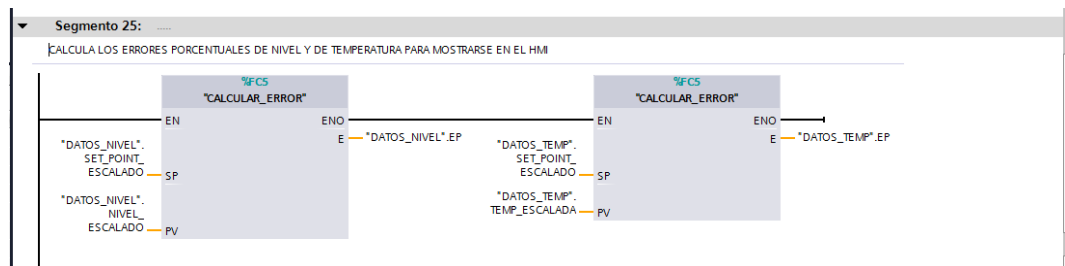
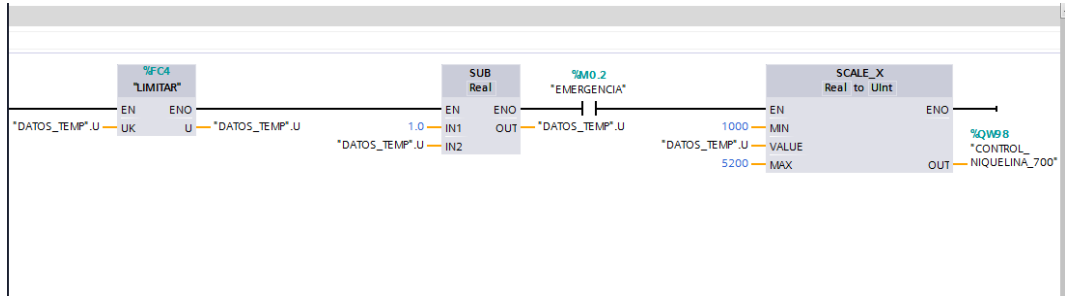
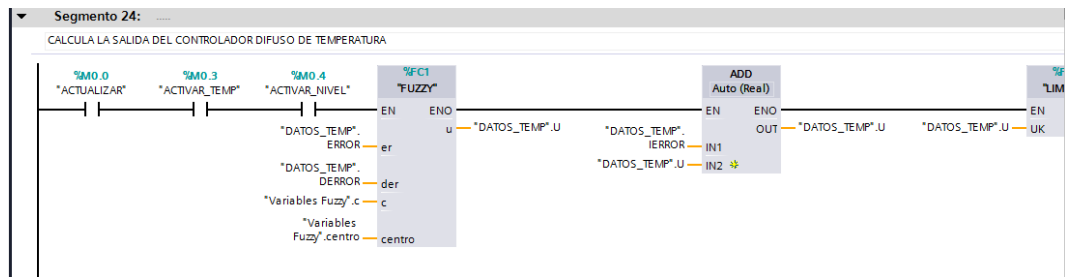






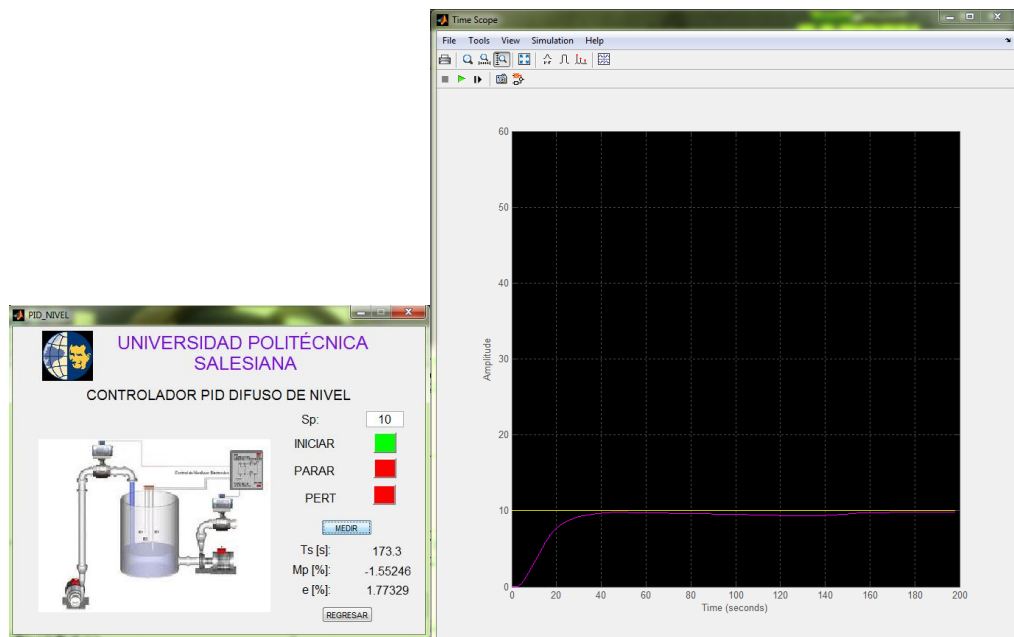




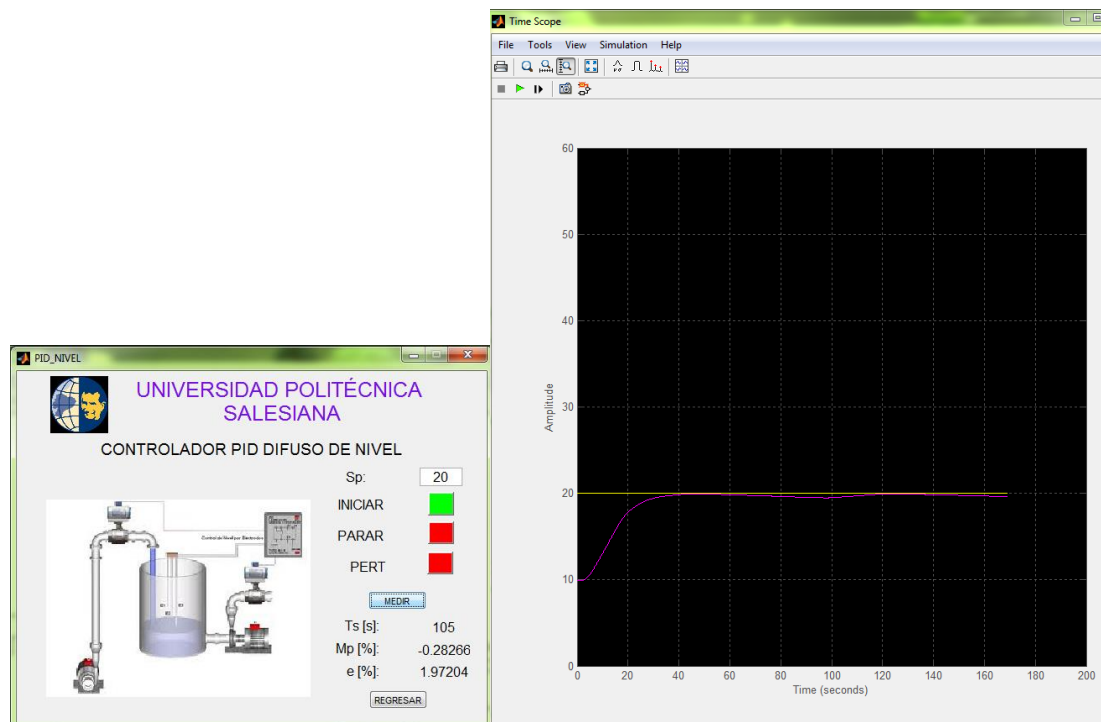


Anexo 2. Gráficas de las pruebas realizadas del controlador PID difuso de la variable de nivel para el Matlab.

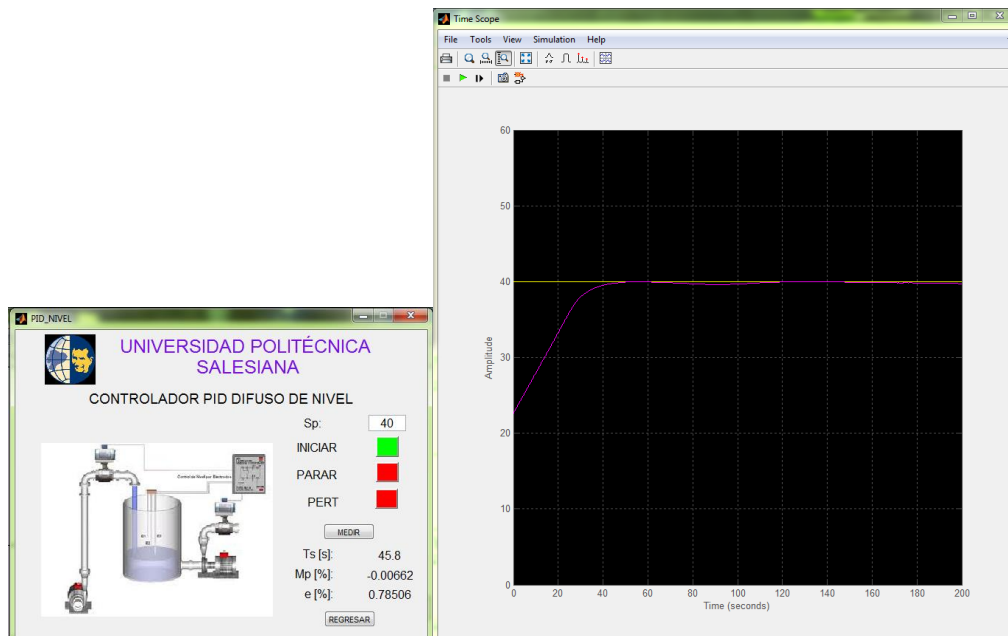
Prueba 1. 0cm a 10cm



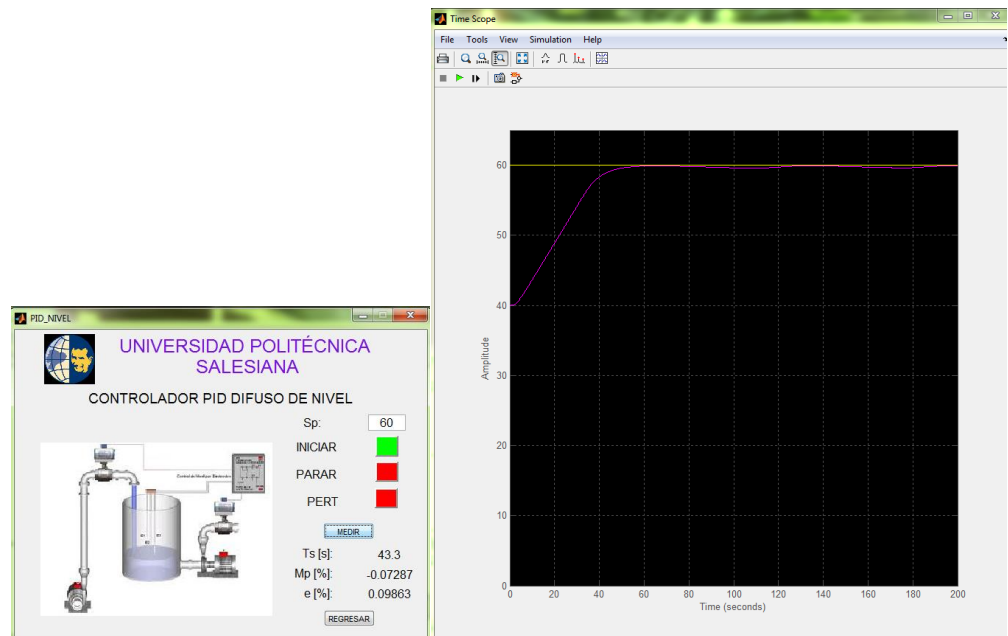
Prueba 2. 10cm a 20cm



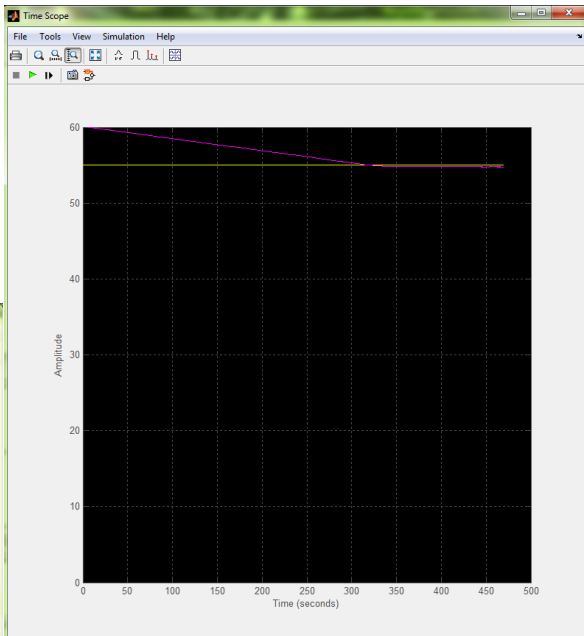
Prueba 3. 20cm a 40cm



Prueba 4. 40cm a 60cm

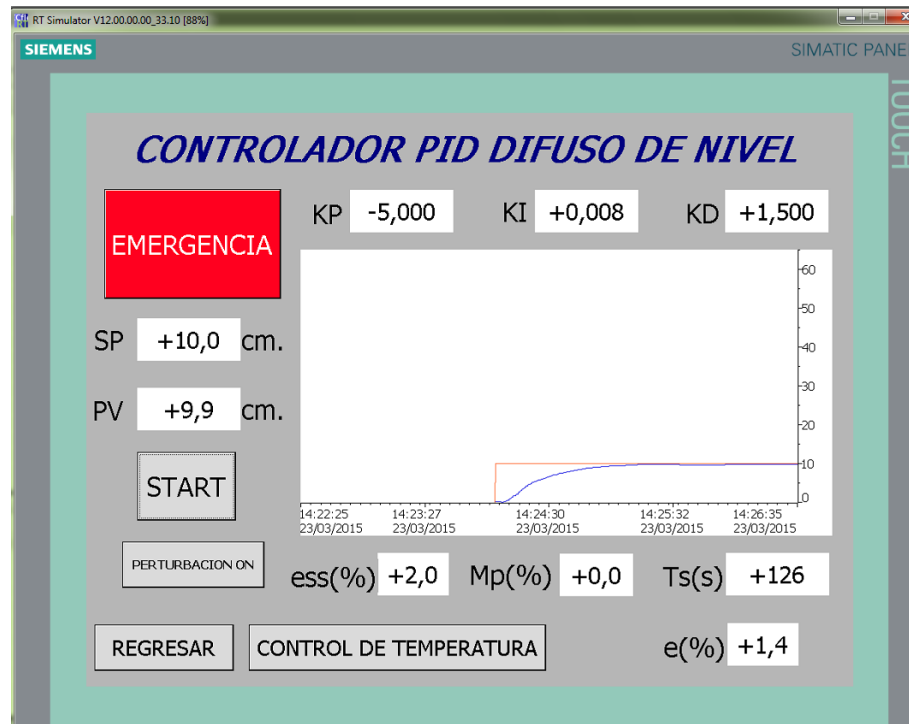


Prueba 5. 60cm a 55cm



Anexo 3. Gráficas de las pruebas realizadas del controlador PID difuso de la variable de nivel para el PLC Siemens S7-1200.

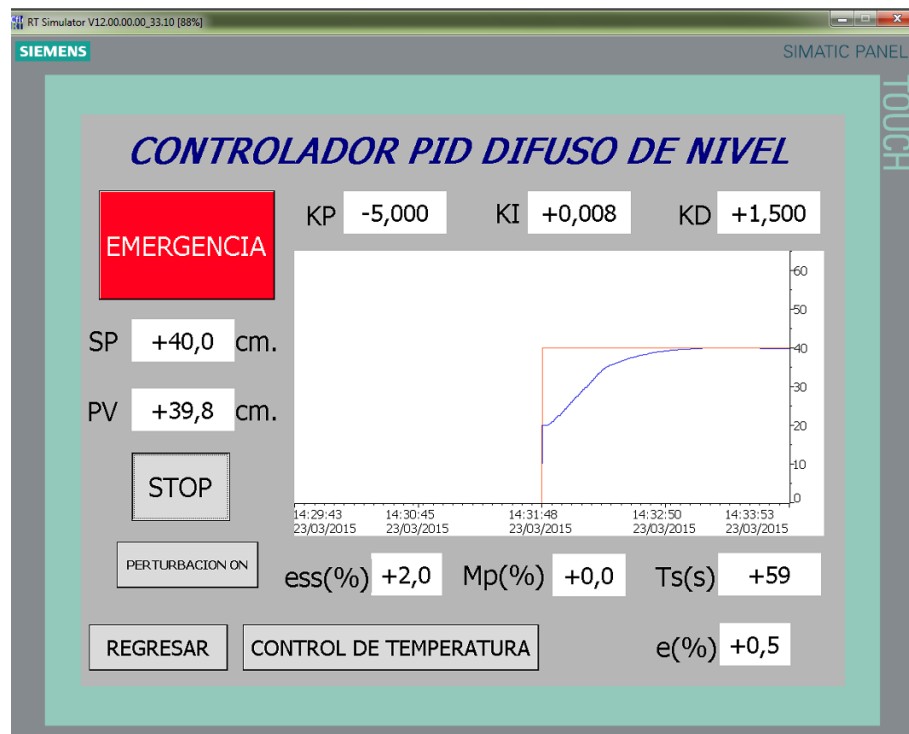
Prueba 1. 0cm a 10cm



Prueba 2. 10cm a 20cm



Prueba 3. 20cm a 40cm



Prueba 4. 40cm a 60cm

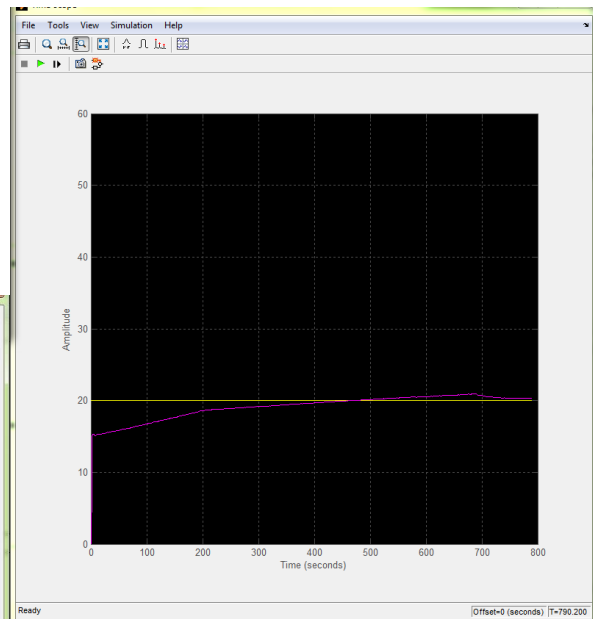


Prueba 5. 60cm a 55cm

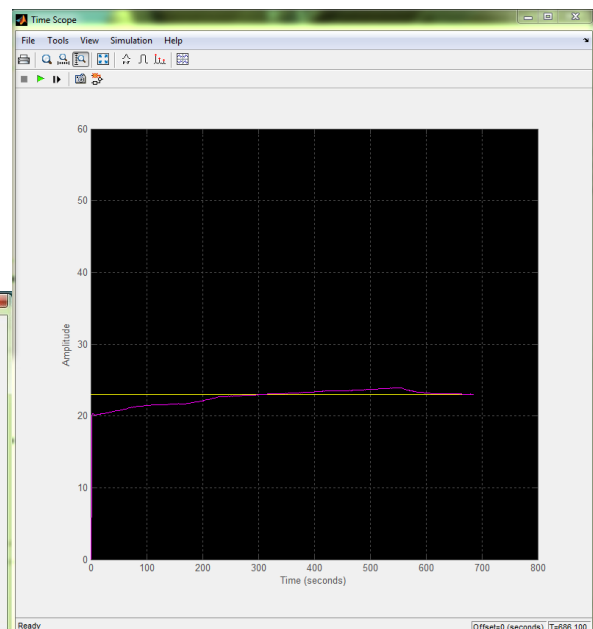
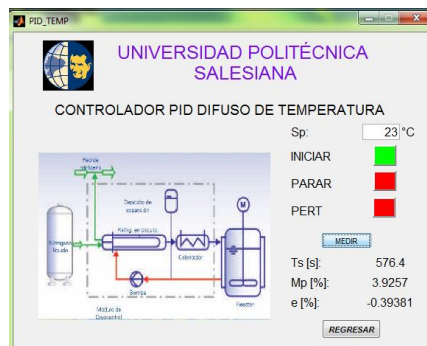


Anexo 4. Gráficas de las pruebas realizadas del controlador PID difuso de la variable de temperatura para el Matlab.

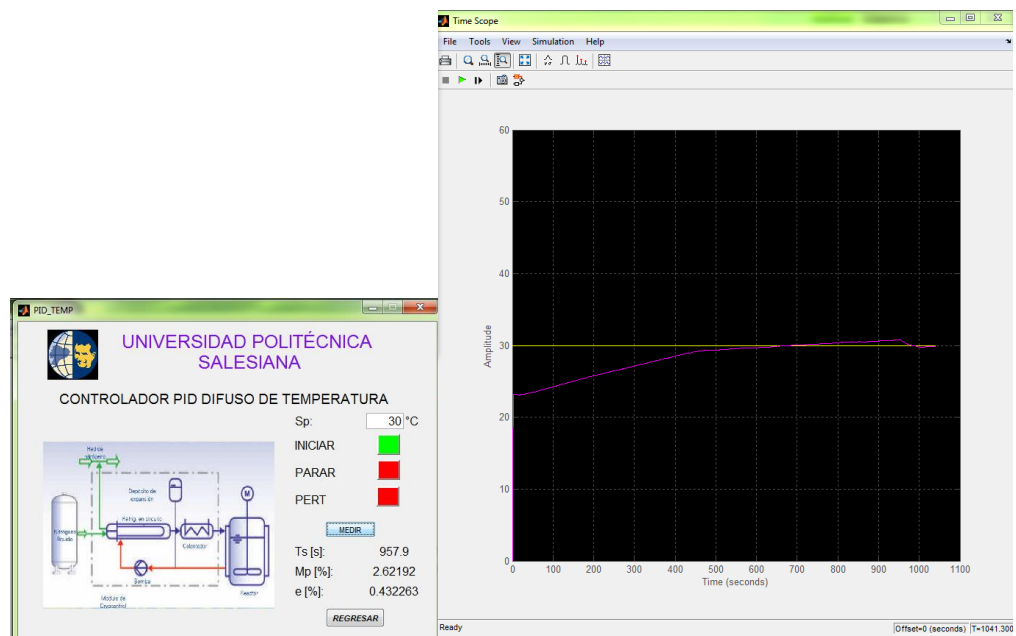
Prueba 1. 15° - 20°



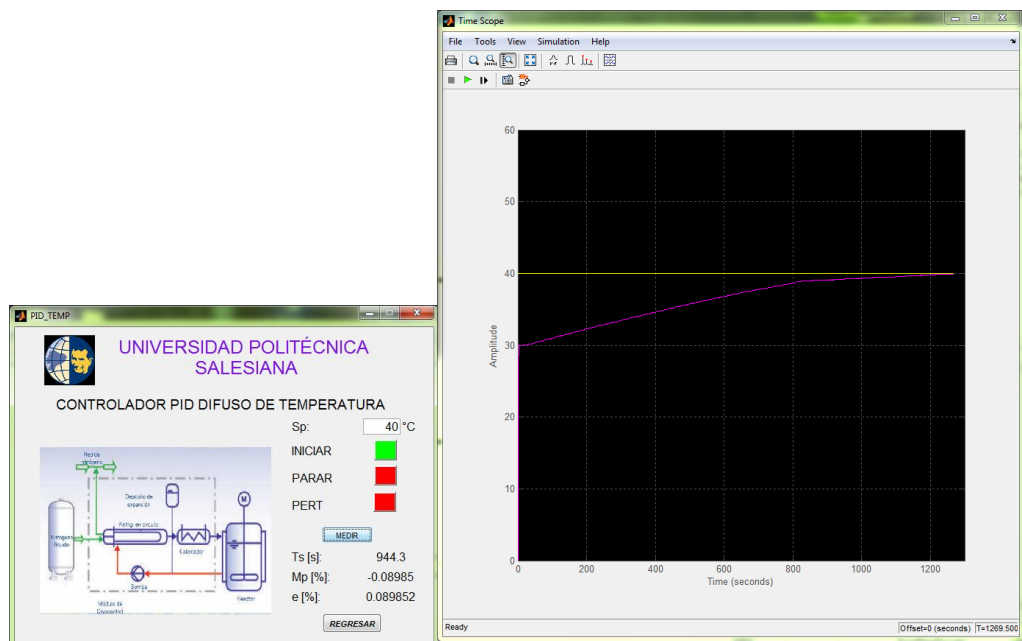
Prueba 2. 20° - 23°



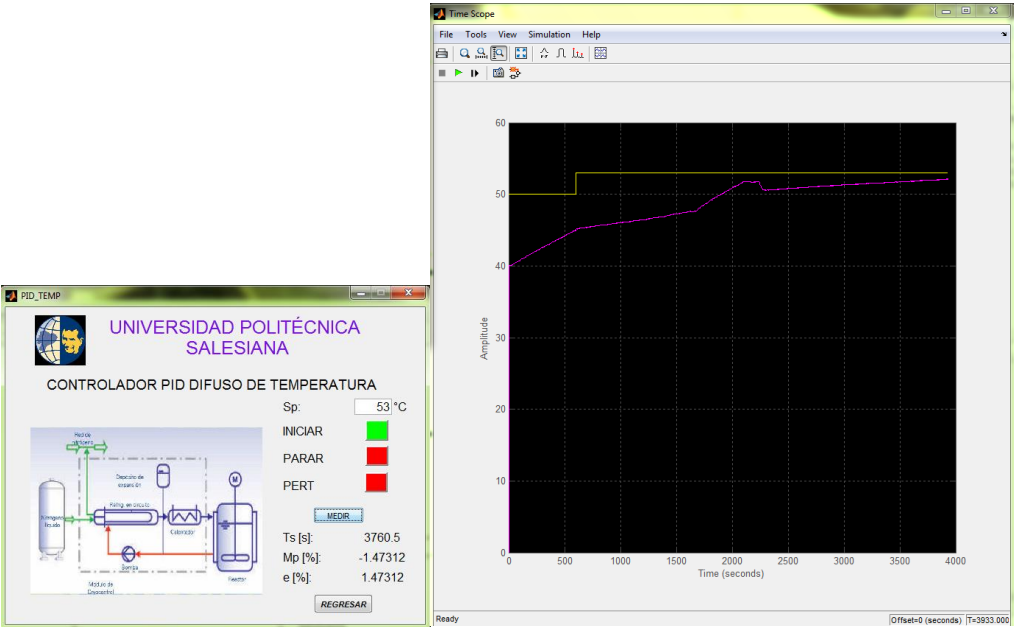
Prueba 3. 23° - 30°



Prueba 4. 30° - 40°

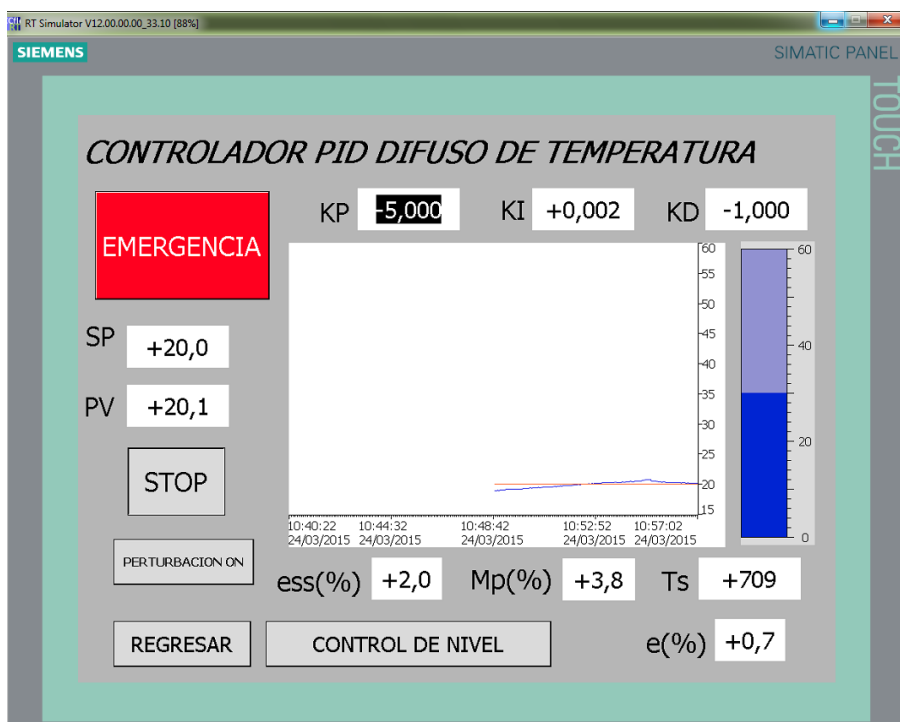


Prueba 5. 30° - 53°

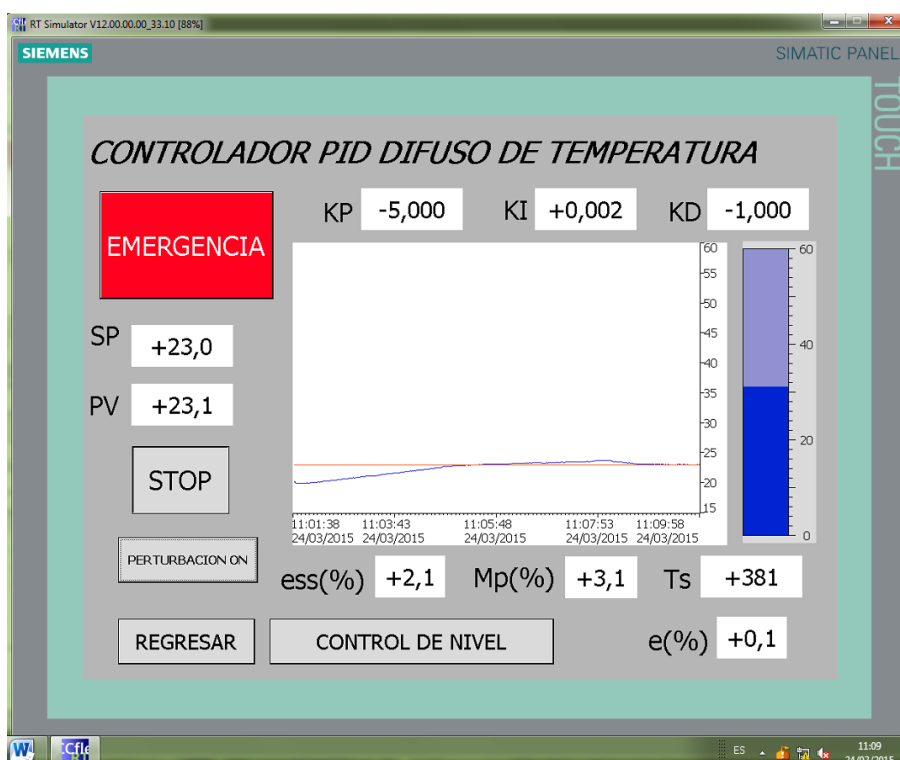


Anexo 5. Gráficas de las pruebas realizadas del controlador PID difuso de la variable de temperatura para el PLC Siemens S7-1200.

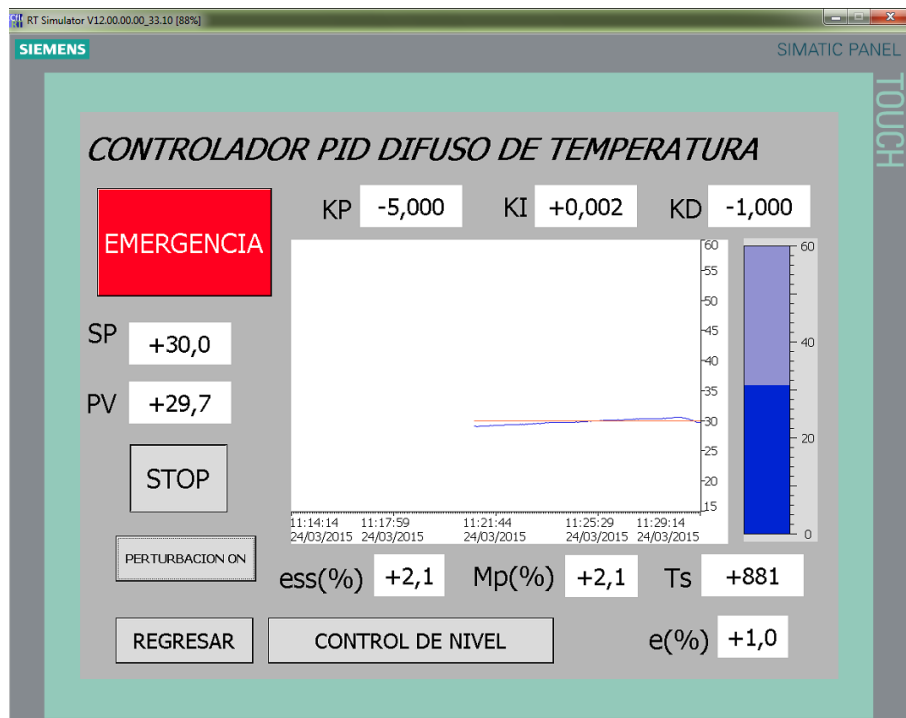
Prueba 1. 15° - 20°



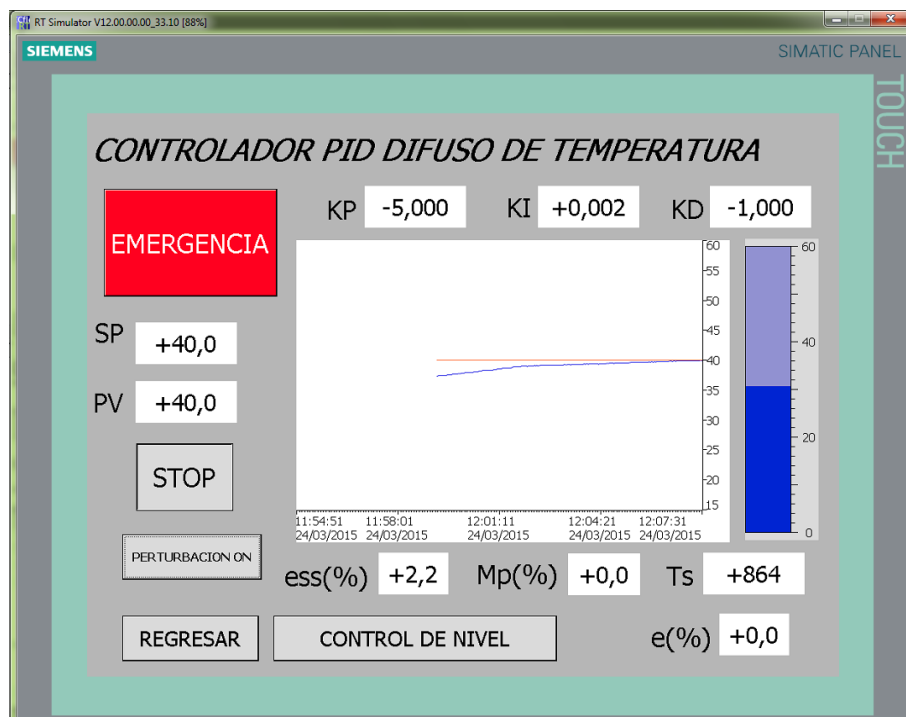
Prueba 2. 20° - 23°



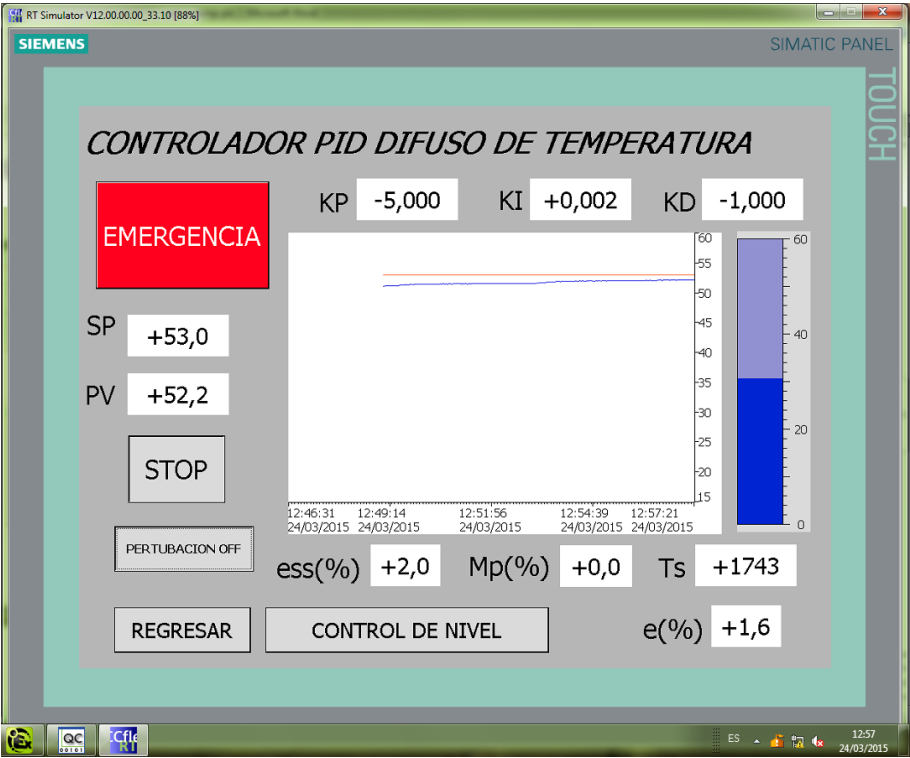
Prueba 3. 23° - 30°




Prueba 4. 30° - 40°



Prueba 5. 40° - 53°



Anexo 6. Prácticas de laboratorio para el control PID difuso.

 UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR	PRÁCTICA NÚMERO 1
	COMUNICACIÓN ENTRE EL PLC SIEMENS S7-1200 Y EL PROGRAMA MATLAB
	CONTROL PID
	ELECTRÓNICA
LABORATORIO	
CARRERA	
SEDE	QUITO

DATOS INFORMATIVOS

- a) MATERIA:
- b) No. DE PRÁCTICA: 1
- c) NÚMERO DE ESTUDIANTES:
- d) NOMBRE DEL INSTRUCTOR:

DATOS DE LA PRÁCTICA

a) TEMA:

Comunicación entre el PLC Siemens S7-1200 y el programa Matlab.

b) OBJETIVO GENERAL:

- Comunicar el PLC Siemens S7-1200 con el programa Matlab por medio del programa KEPServerEX para manipular variables físicas del PLC desde el Matlab y viceversa.

c) OBJETIVOS ESPECÍFICOS:

- Crear las variables de lectura y escritura en el programa KEPServerEX para la interacción entre el dispositivo y el programa.
- Agregar y configurar los bloques OPC en la herramienta Simulink de Matlab con la ayuda de las librerías que posee el Simulink para establecer la interacción con el PLC.
- Configurar las entradas y salidas análogas del PLC siemens S7-1200 con el programa TIA PORTAL para lectura y escritura en el PLC.
- Verificar el correcto funcionamiento de la comunicación mediante la observación de la lectura y escritura de variables para poder realizar prácticas futuras.

d) MARCO TEÓRICO

TIA PORTAL

Siemens, en su manual señala que TIA PORTAL ofrece un entorno confortable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como controladores y dispositivos HMI. Para poder encontrar la información necesaria, TIA PORTAL ofrece un completo sistema de ayuda en pantalla.

Matlab

Matlab es un acrónimo que proviene de matrix laboratory (laboratorio matricial) creado por el profesor y matemático Cleve Moler en 1970. La primera versión de Matlab fue escrita en lenguaje fortran la cual contempló la idea de emplear subrutinas para los cursos de álgebra lineal y análisis numérico de los paquetes Linpack y Eispack; posteriormente se desarrolló software de matrices para acceder a esos paquetes sin necesidad de usar programas en fortran. Actualmente, el lenguaje de programación de Matlab proporciona un sencillo acceso a algoritmos numéricos que incluye matrices.

KEPServerEx

KEPServerEX OPC, es un programa propietario de la empresa Kepware Technologies, el mismo que proporciona el intercambio de datos entre clientes OPC y Siemens S7-200, S7-300, S7-400 y S7-1200 PLC utilizando TCP / protocolo IP Ethernet.

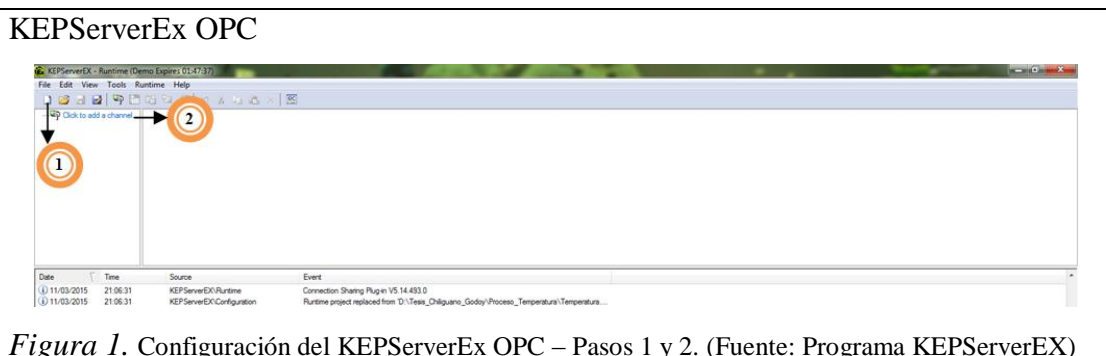
El OPC (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos. Este estándar permite que diferentes fuentes de datos envíen datos a un mismo servidor OPC, al que a su vez podrán conectarse diferentes programas compatibles con dicho estándar. De este modo se elimina la necesidad de que todos los programas cuenten con drivers para dialogar con múltiples fuentes de datos, basta que tengan un driver OPC.

e) PROCEDIMIENTO

Crear las variables en el KEPServerEX

Para configurar la comunicación del OPC se realiza de la siguiente manera:

1. Crear un nuevo proyecto, como se muestra en la Figura 1.
2. Crear un canal con el identificador Ethernet, seleccionar el controlador de dispositivo Siemens TCP/IP Ethernet, seleccionar la tarjeta de red del computador utilizado, clic en siguiente y luego finalizar, como se muestra en la Figura 1.



3. Añadir un nuevo dispositivo con el identificador S7-1212C, clic siguiente, seleccionar el modelo del dispositivo S7-1200, clic siguiente, ingresamos la dirección IP del PLC, por cuatro ocasiones clic siguiente, definir las direcciones de transporte Local TSAP (Transport Service Acces Point, punto de acceso al servicio de transporte) en 1000 y Remoto TSAP en 0301, por dos ocasiones clic siguiente y luego finalizar, como se muestra en la Figura 2.



4. Crear las variables de intercambio, para las variables de entrada con la dirección I y variables de salidas con la dirección Q, como se observa en la Figura 3.

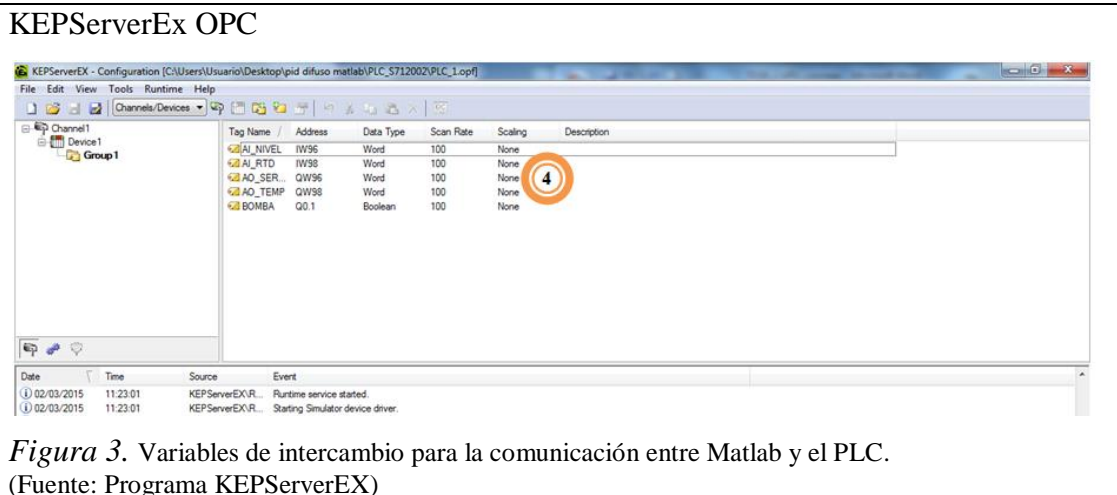


Figura 3. Variables de intercambio para la comunicación entre Matlab y el PLC.

(Fuente: Programa KEPServerEX)

5. Verificar la comunicación mediante el KEPServerEX OPC con el PLC Siemens S7-1200 dando clic en Launch OPC Quick Client, como se observa en la Figura 4.

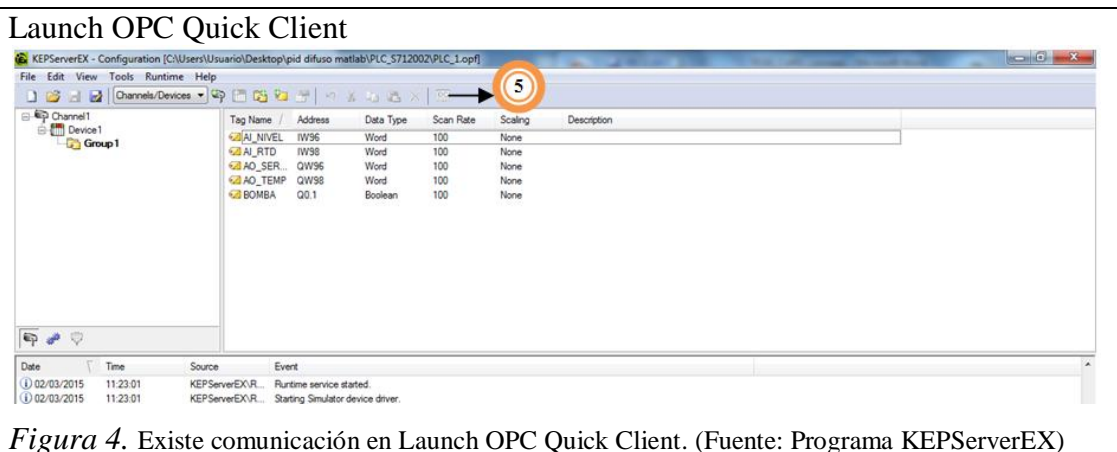


Figura 4. Existe comunicación en Launch OPC Quick Client. (Fuente: Programa KEPServerEX)

6. La ventana del Launch OPC Quick Client se verifica la comunicación de los sensores y actuadores, como se observa en la Figura 5.

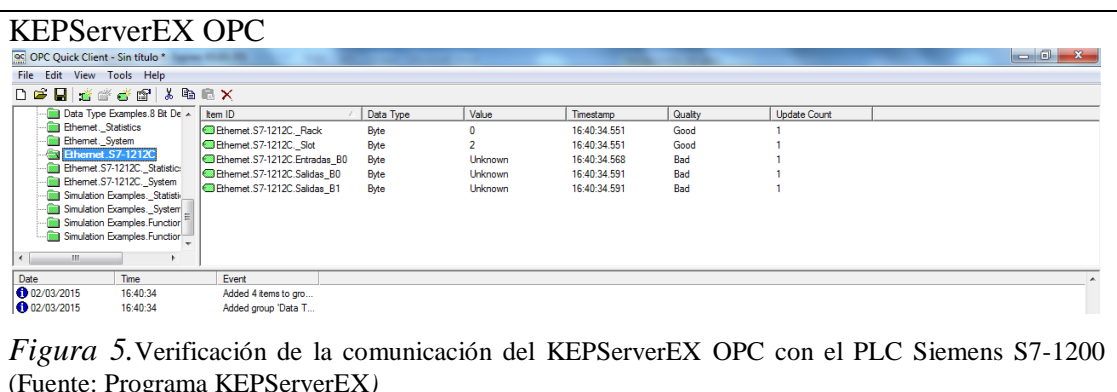


Figura 5. Verificación de la comunicación del KEPServerEX OPC con el PLC Siemens S7-1200 (Fuente: Programa KEPServerEX)

Matlab

Se configura el OPC KEPServerEx dentro de Simulink, realizando los siguientes pasos:

1. Agregar el bloque OPC de configuración, el bloque OPC de lectura y el bloque OPC de escritura, como se observa en la Figura 6.

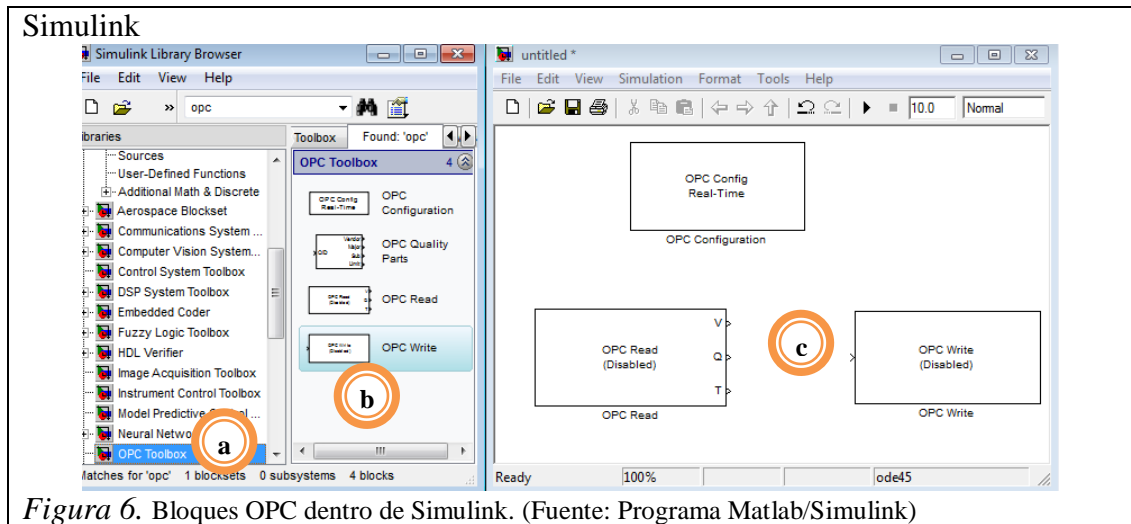


Figura 6. Bloques OPC dentro de Simulink. (Fuente: Programa Matlab/Simulink)

Los pasos para agregar los bloques OPC, se detallan a continuación:

- a. Seleccionar la librería OPC Toolbox.
 - b. Seleccionar cada uno de los bloques OPC.
 - c. Añadir los bloques OPC en el nuevo modelo de simulink.
2. Dentro del bloque OPC de configuración se define el cliente OPC utilizado, que en este caso es el PLC Siemens S7-1200, como se observa en la Figura 7.

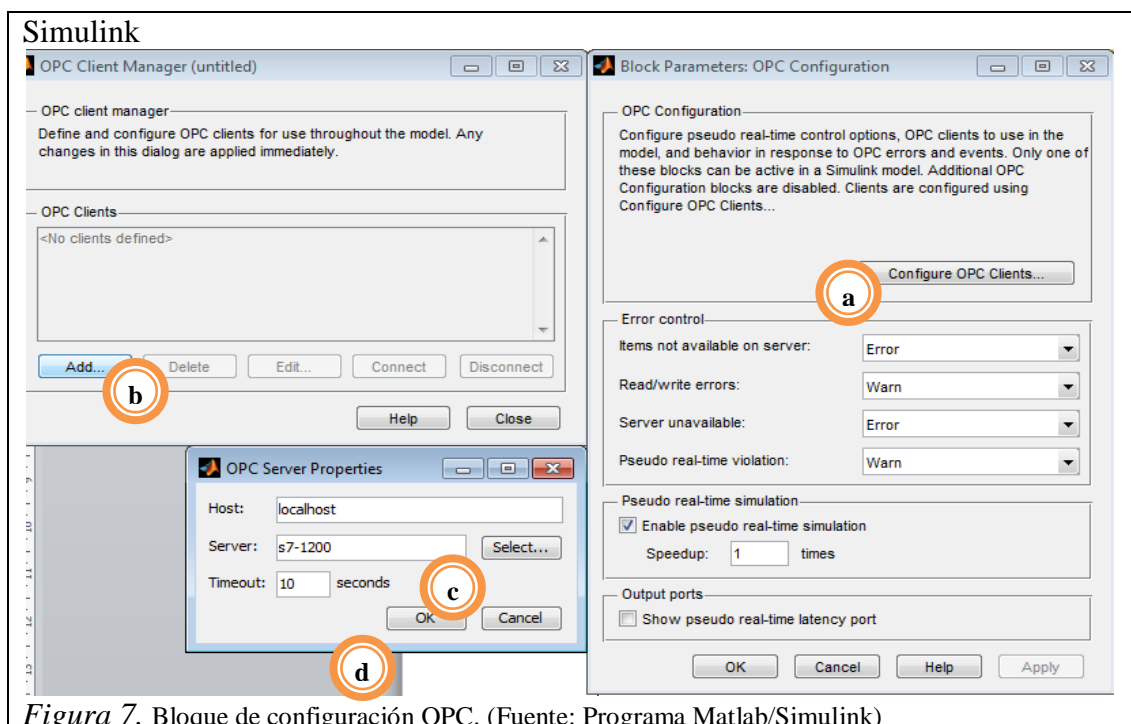


Figura 7. Bloque de configuración OPC. (Fuente: Programa Matlab/Simulink)

Los pasos para agregar los parámetros del bloque de configuración OPC, se detallan a continuación:

- a. Seleccionar Configure OPC Clients.
 - b. En la ventana OPC Client Manager, seleccionar Add.
 - c. Seleccionar el servidor s7-1200, generado del programa KEPServerEX.
 - d. Seleccionar OK en cada una de las ventanas abiertas.
3. Dentro del bloque OPC de lectura se añade las variables a leer, transmisores conectados al PLC Siemens S7-1200, como se observa en la Figura 8.

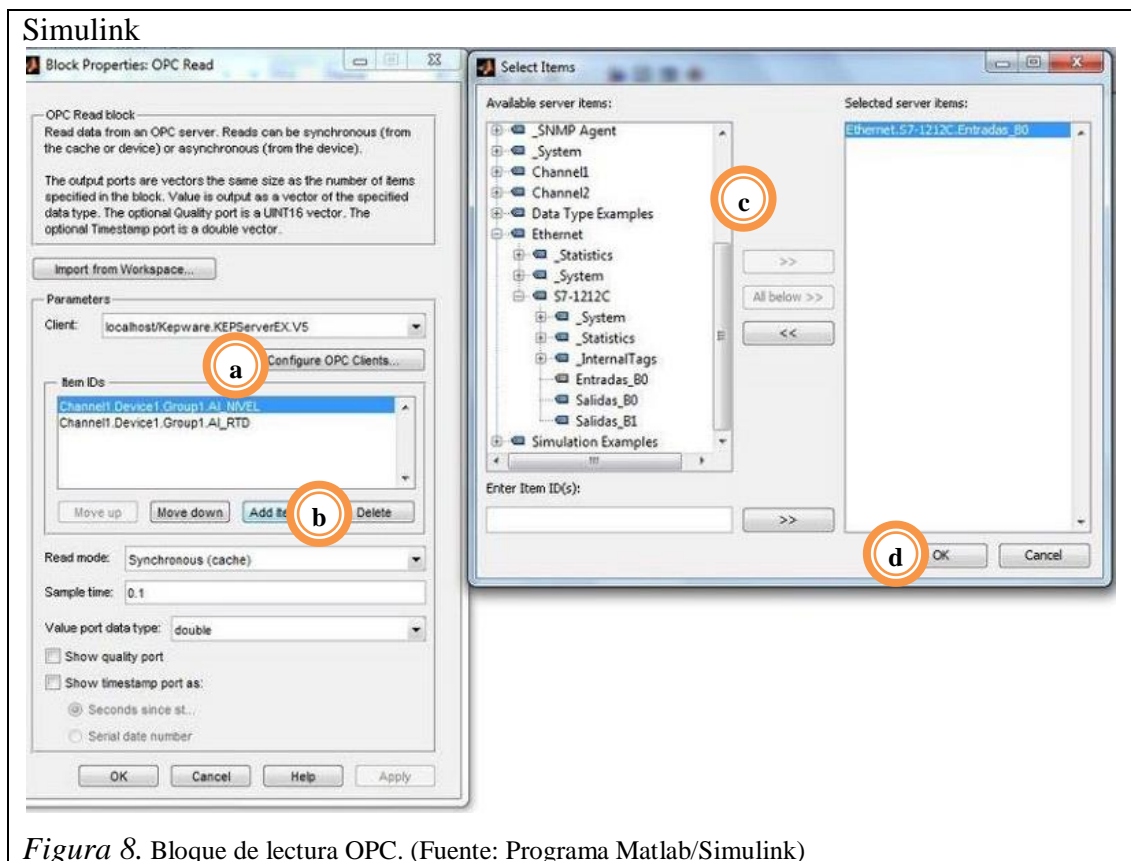


Figura 8. Bloque de lectura OPC. (Fuente: Programa Matlab/Simulink)

Los pasos para agregar los parámetros del bloque de lectura OPC, se detallan a continuación:

- a. Seleccionar Configure OPC Clients, se realiza la configuración de acuerdo a los literales b, c y d, del paso 2.
- b. Seleccionar Add Items.
- c. Agregar los ítems que se utilizan como variables de lectura.
- d. Seleccionar OK.

4. Dentro del bloque OPC de escritura se añaden las variables a controlar, actuadores conectados al PLC Siemens S7-1200, como se observa en la Figura 9.

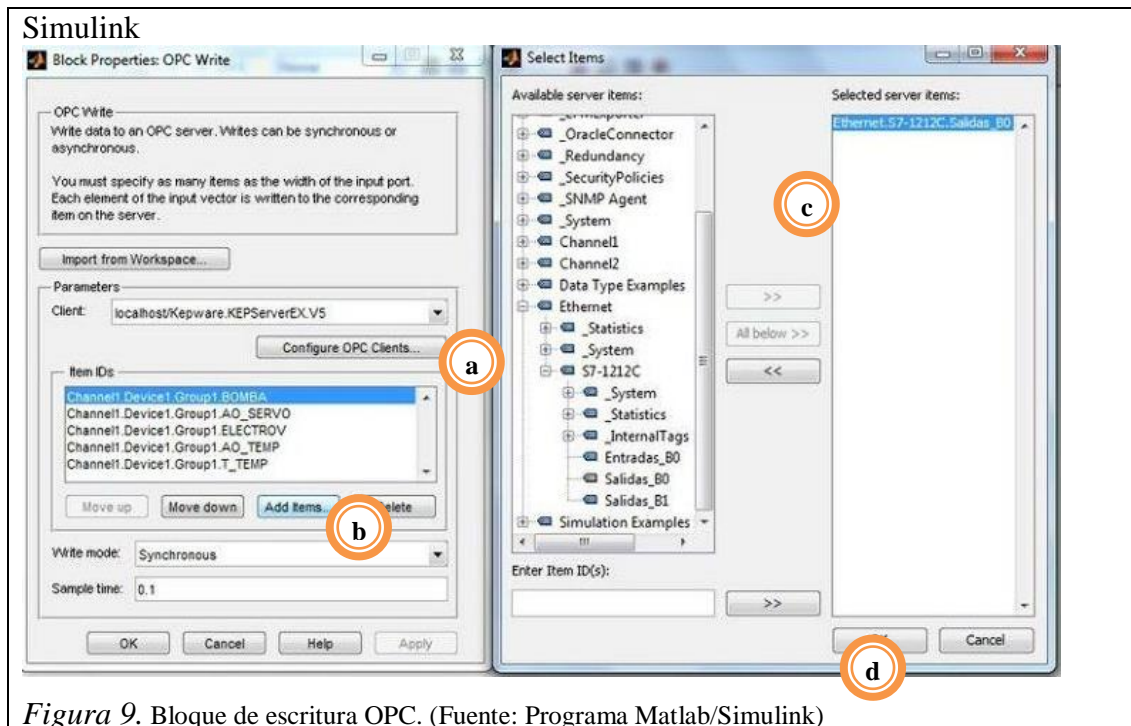


Figura 9. Bloque de escritura OPC. (Fuente: Programa Matlab/Simulink)

Los pasos para agregar los parámetros del bloque de escritura OPC, se detallan a continuación:

- a. Seleccionar Configure OPC Clients, se realiza la configuración de acuerdo a los literales b, c y d, del paso 2.
- b. Seleccionar Add Items.
- c. Agregar los ítems que se utilizan como variables de escritura.
- d. Seleccionar OK.

TIA PORTAL

Para que el PLC Siemens S7-1200, actúe como una tarjeta de adquisición de datos se procede a realizar los siguientes pasos:

1. Se abre el programa TIA PORTAL V12, donde se selecciona crear proyecto, a este proyecto se le detalla tanto el nombre, la ruta donde va hacer guardado el archivo, el autor o los detalles que realiza el proyecto y un comentario referente al proyecto. Y por último se selecciona crear, como se observa en la Figura 10.

TIA PORTAL

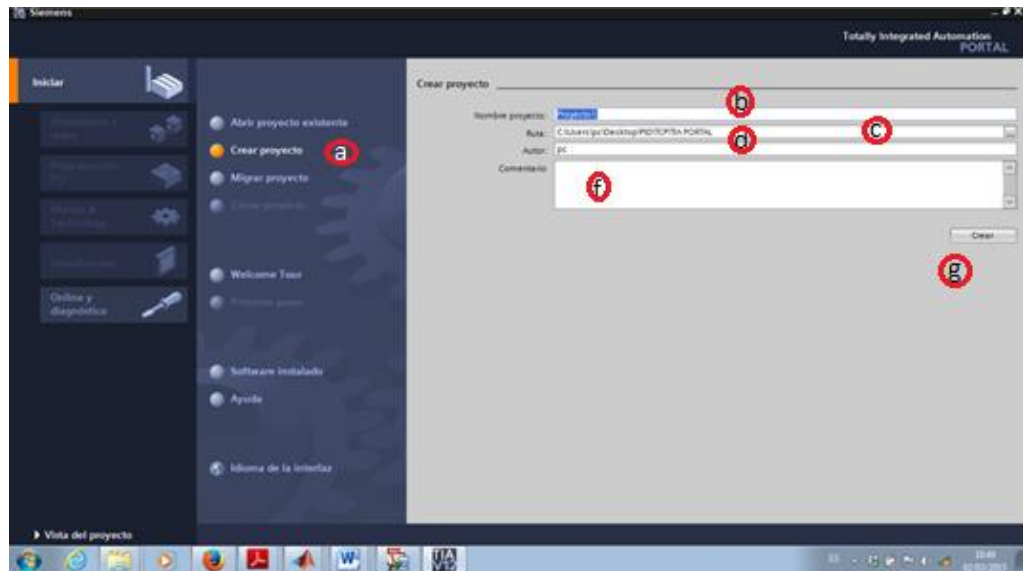


Figura 10. Creación de un nuevo proyecto. (Fuente: Programa TIA PORTAL)

2. Se agrega el CPU en este caso el 1212C AC/DC/Rly, como se muestra en la Figura 11.

TIA PORTAL



Figura 11. Selección del CPU. (Fuente: Programa TIA PORTAL)

3. Se agrega el módulo de expansión de entradas y salidas analógicas SM-1234, como se observa en la Figura 12.

TIA PORTAL

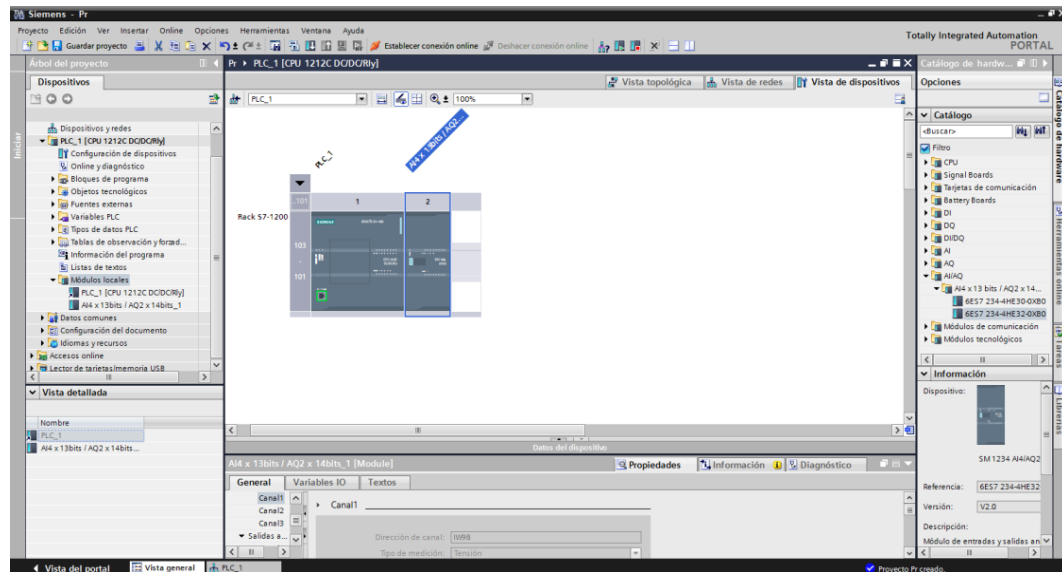


Figura 12. Módulo de expansión de entradas y salidas analógicas SM-1234. (Fuente: Programa TIA PORTAL)

4. Se establece los parámetros de las entradas y salidas analógicas, ya sean por tensión o corriente, como se observa en la Figura 13.

TIA PORTAL

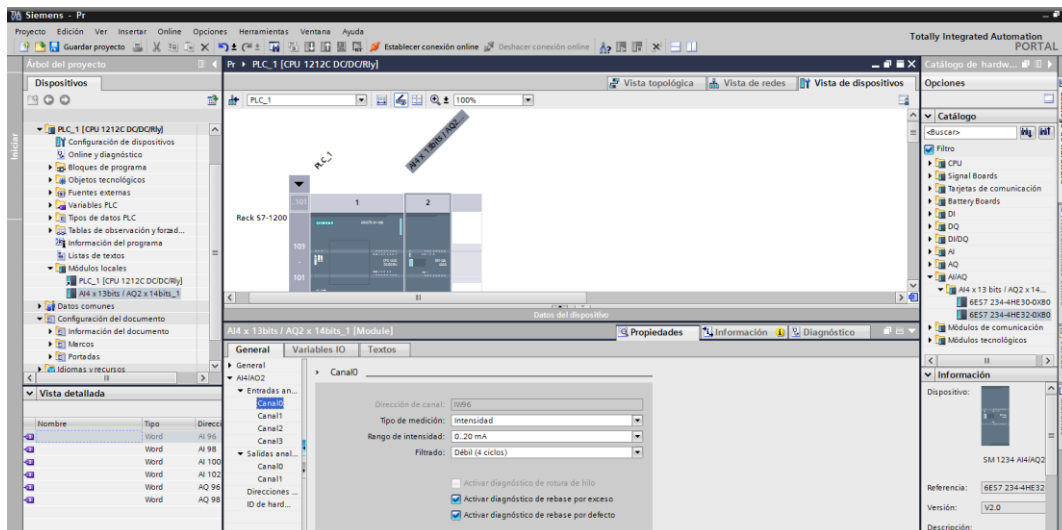
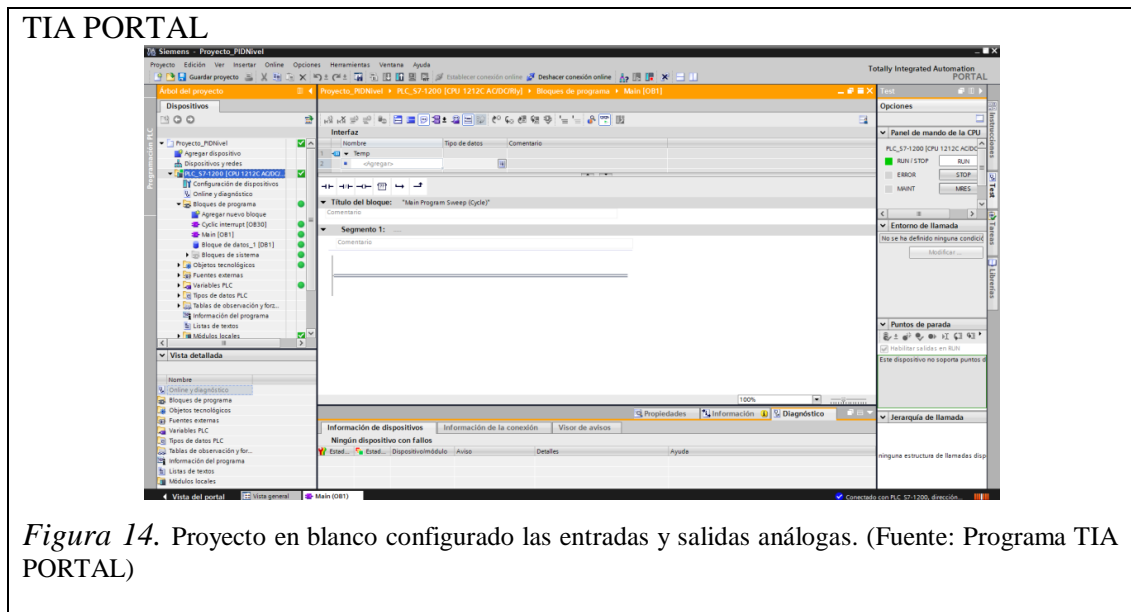



Figura 13. Configuración de los parámetros de variables de entradas y salidas analógicas. (Fuente: Programa TIA PORTAL)

5. Para el funcionamiento de la planta con el PID Difuso en Matlab, se carga un archivo en blanco, con los segmentos sin programar, como se observa en la Figura 14.



 UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR	PRÁCTICA NÚMERO 2
	CONTROL PID DIFUSO DE LAS VARIABLES DE NIVEL Y TEMPERATURA EN EL PROGRAMA MATLAB
LABORATORIO	CONTROL PID
CARRERA	ELECTRÓNICA
SEDE	QUITO

DATOS INFORMATIVOS

- a) MATERIA:
- b) No. DE PRÁCTICA: 2
- c) NÚMERO DE ESTUDIANTES:
- d) NOMBRE DEL INSTRUCTOR:

DATOS DE LA PRÁCTICA

a) TEMA:

Diseño del controlador PID Difuso de las variables de nivel y temperatura en el programa Matlab para el control de procesos continuos.

b) OBJETIVO GENERAL:

- Diseñar un controlador PID Difuso de las variables de nivel y temperatura en el programa Matlab para el control de procesos continuos.

c) OBJETIVOS ESPECÍFICOS:

- Diseñar e implementar el algoritmo de control PID difuso en el software Matlab para control de los procesos de nivel y temperatura.
- Comunicar el PLC Siemens S7-1200 con el software Matlab mediante el protocolo de comunicación Ethernet para adquirir datos, ser procesados por software y actuar sobre la planta.
- Verificar el correcto funcionamiento de la comunicación mediante la observación de la lectura y escritura de variables para el control PID Difuso de la planta de procesos continuos.

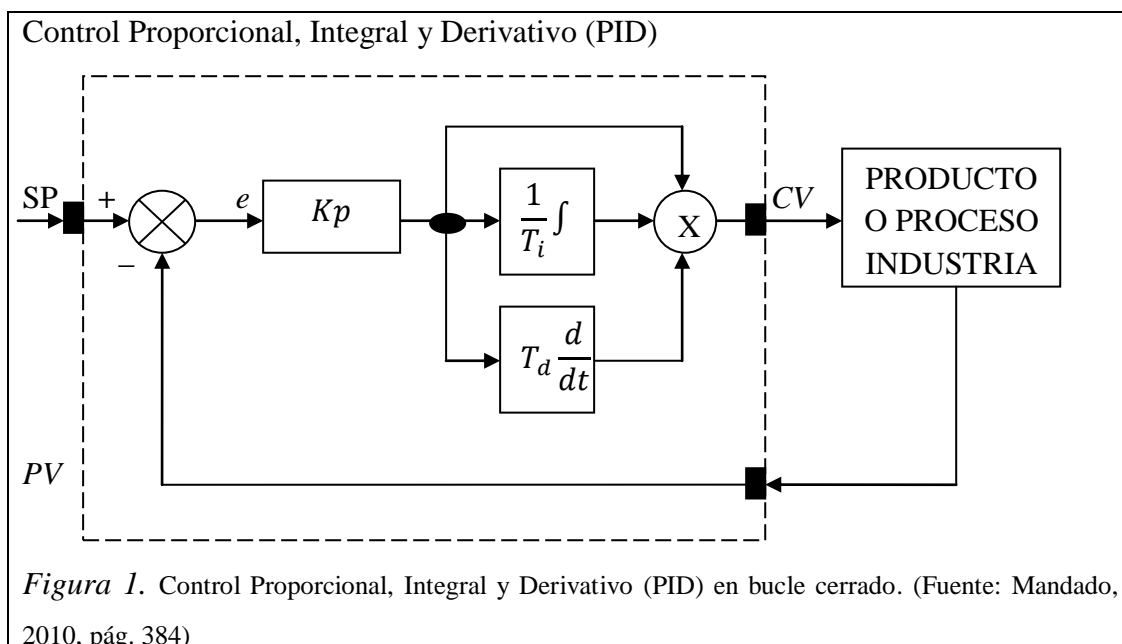
d) MARCO TEÓRICO

Controlador Proporcional Integral Derivativo – PID

Ogata, sustenta que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID. La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos, es cuando los controles PID resultan más útiles. En el campo de los sistemas para control de procesos, es un hecho bien conocido que los esquemas de control PID básicos y modificados han demostrado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo. (Ogata, 2010).

Estructura del PID

Mandado, representa la estructura de control más usada en el medio industrial, mostrando en la Figura 1 el diagrama de bloques de un regulador PID en el que la señal de control CV se obtiene mediante la suma de los tres tipos de acciones. El parámetro T_i , pondera la acción integral y el parámetro T_d pondera la acción derivativa. Además, la señal de error se amplifica mediante el parámetro K_p que afecta también a las acciones integral y derivativa. El diseñador debe asignar a los parámetros K_p T_i T_d un valor positivo o cero. (Mandado, 2010).



Lógica Difusa

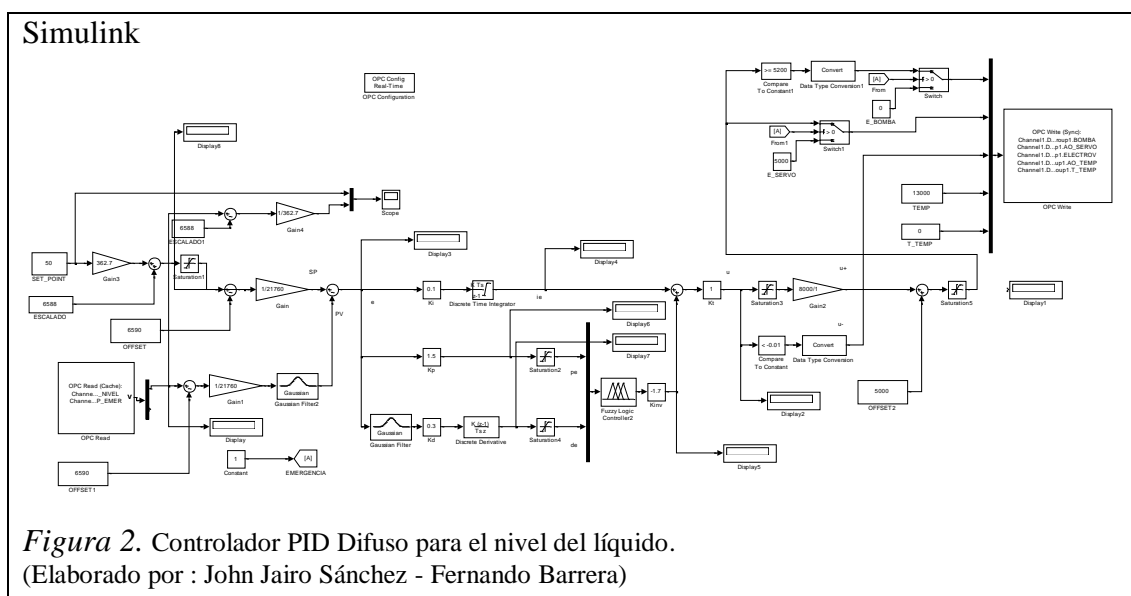
Los sistemas basados en lógica difusa emulan la manera en que el cerebro razona o piensa, y en donde las aplicaciones se realizan en áreas multidisciplinarias que van desde la evolución tecnológica de los electrodomésticos, hasta programas computacionales para tomar decisiones y se han extendido a diversas áreas específicas como las cámaras de video, seguridad en base al monitoreo de la actividad volcánica, a través de controladores difusos para la calidad del agua, sistemas de operación automática de trenes, reactores nucleares y las transmisiones de automóviles.

Toolbox Fuzzy

Tolbox Fuzzy ofrece funciones, aplicaciones, y un bloque de Simulink para el análisis, el diseño y los sistemas basados en lógica difusa simulando. El Software le guía a través del proceso de diseño de los sistemas de inferencia difusos. Las funciones se proporcionan para los muchos métodos comunes, incluyendo clustering fuzzy y aprendizaje neuroborroso adaptativa.

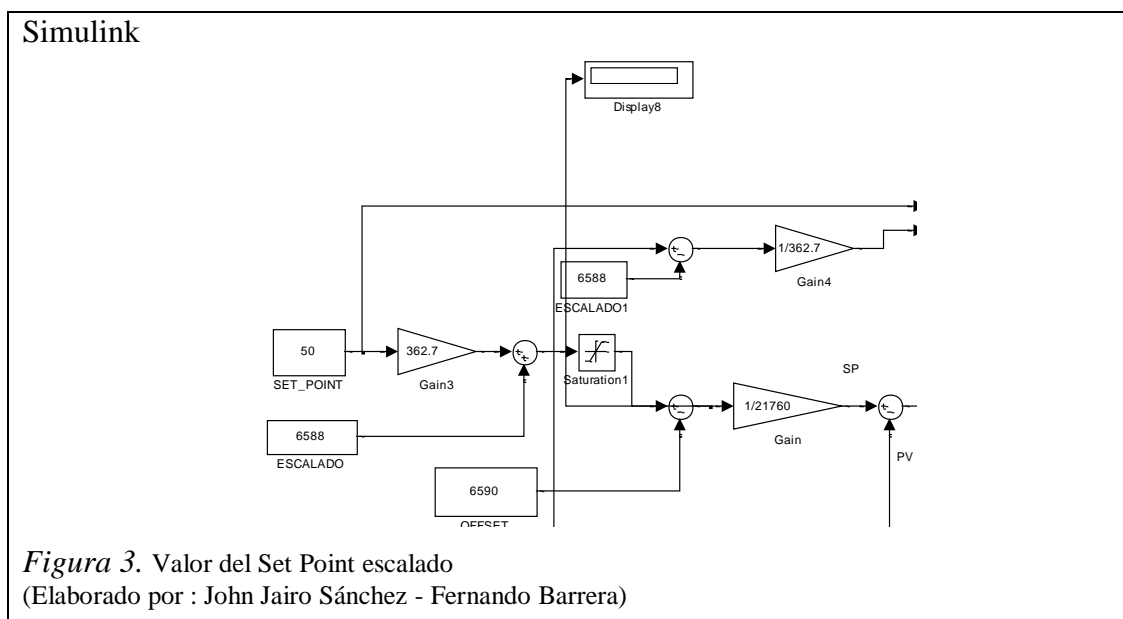
e) PROCEDIMIENTO

De acuerdo con el diagrama de bloques se programa el control PID difuso dentro del entorno Simulink, como se visualiza en la Figura 2.

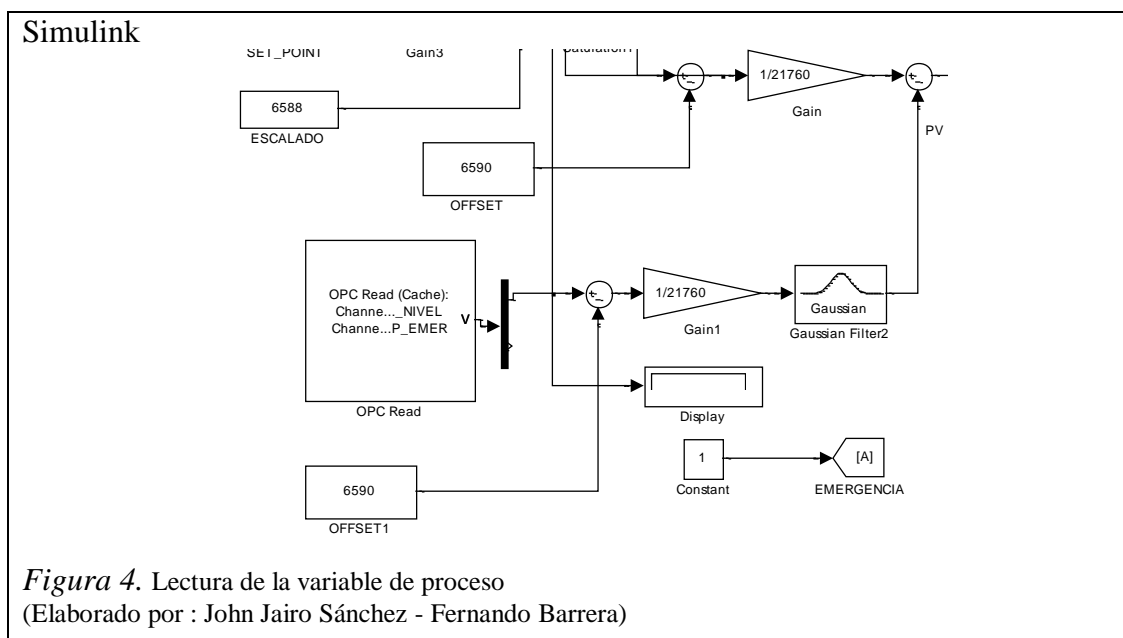


A continuación se detalla la programación en bloques realizada en Simulink del controlador PID difuso de la variable de nivel:

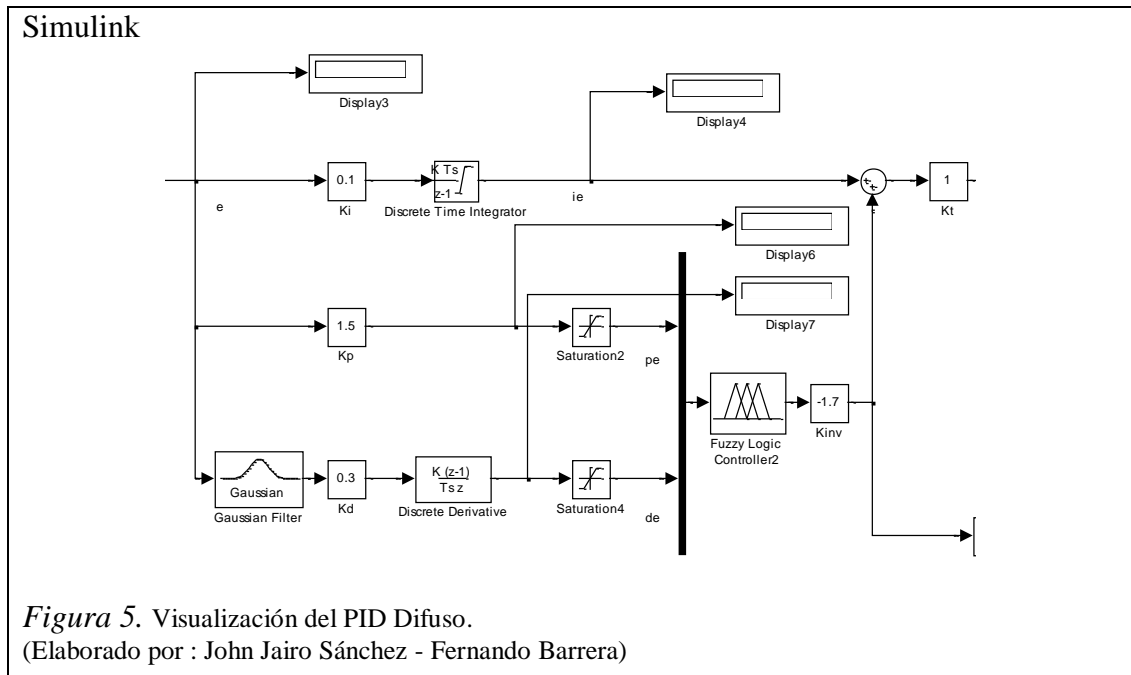
- En la Figura 3 se muestra el valor del Set Point escalado a una magnitud de longitud de 0 cm. a 60 cm.



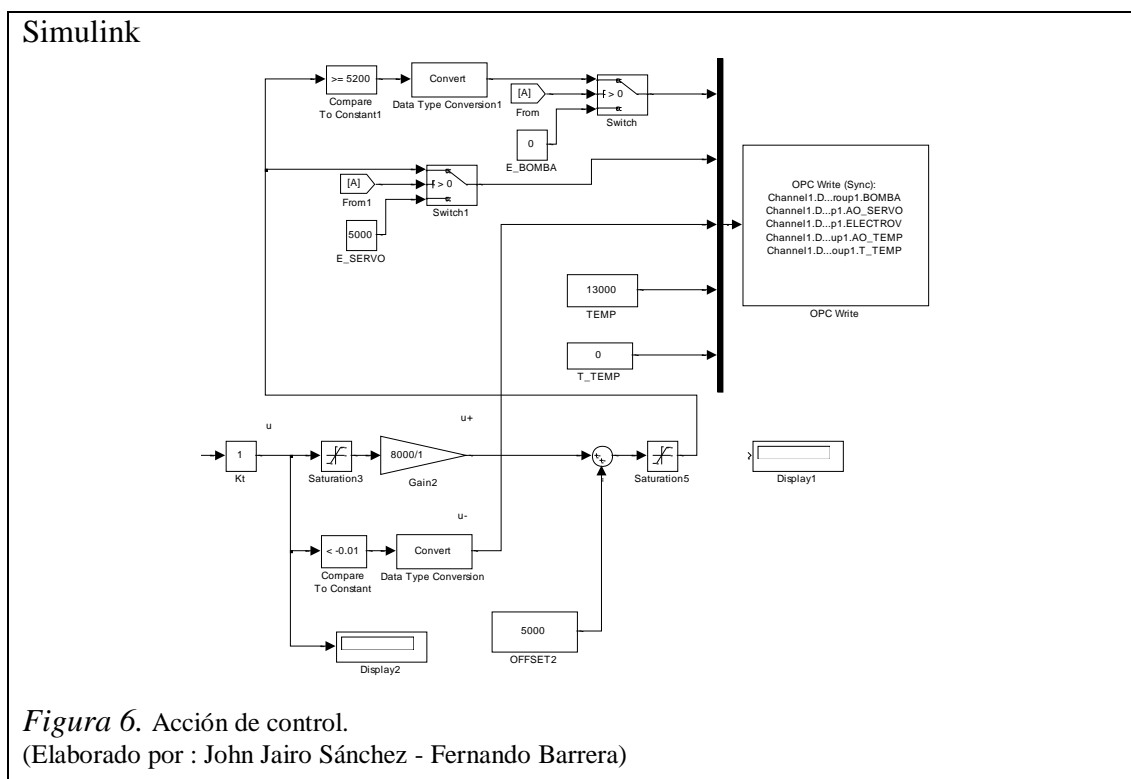
- Lectura de la variable de proceso del transmisor de nivel, como se observa en la Figura 4.



7. En la Figura 5 se visualiza el controlador PID difuso, en un bloque el parámetro K_p y K_d , y sumando el parámetro K_i .



8. La acción de control es leída por el actuador, en este caso la servo-válvula, como se observa en la Figura 6.



Para realizar el diseño del controlador PID difuso se va a utilizar el Fuzzy Logic Toolbox de Matlab que crea un archivo de extensión fis, el cual se carga en el bloque difuso dentro del entorno Simulink, para ello se realiza los siguientes pasos:

1. Se crea las variables y términos lingüísticos que se utilizan para el controlador PID difuso.
 - Conjuntos difusos para el error
 - Egn: Error grande negativo.
 - Epn: Error pequeño negativo.
 - Ec: Error cero.
 - Epp: Error pequeño positivo.
 - Egp: Error grande positivo.
 - Conjuntos difusos para la derivada del error
 - Degn: Derivada del error grande negativo.
 - Depn: Derivada del error pequeño negativo.
 - Dec: Derivada del error cero.
 - Depp: Derivada del error pequeño positivo.
 - Degp: Derivada del error grande positivo.
 - Conjuntos difusos de la acción de control
 - Ugn: Salida grande negativo.
 - Upn: Salida pequeño negativo.
 - Uc: Salida cero.
 - Upp: Salida pequeño positivo.
 - Ugp: Salida grande positivo.

- Se crea la base de reglas de inferencia para el controlador difuso de nivel, que se muestra en la Tabla 1.

Tabla 1.

Reglas de inferencia para el controlador difuso de nivel

Error \ Derivada del error	Egn	Epn	Ec	Epp	Egp
Degn	Ugn	Ugn	Ugn	Upn	Uc
Depn	Ugn	Upn	Upn	Uc	Upp
Dec	Ugn	Upn	Uc	Upp	Ugp
Depp	Upn	Uc	Upp	Ugp	Ugp
Degp	Uc	Upp	Ugp	Ugp	Ugp

Nota: Abreviatura de los conjuntos de las variables lingüísticas

- En el espacio de trabajo de Matlab se introduce el comando “fuzzy”, se visualiza la ventana de la librería difusa de Matlab, como lo muestra la Figura 7.

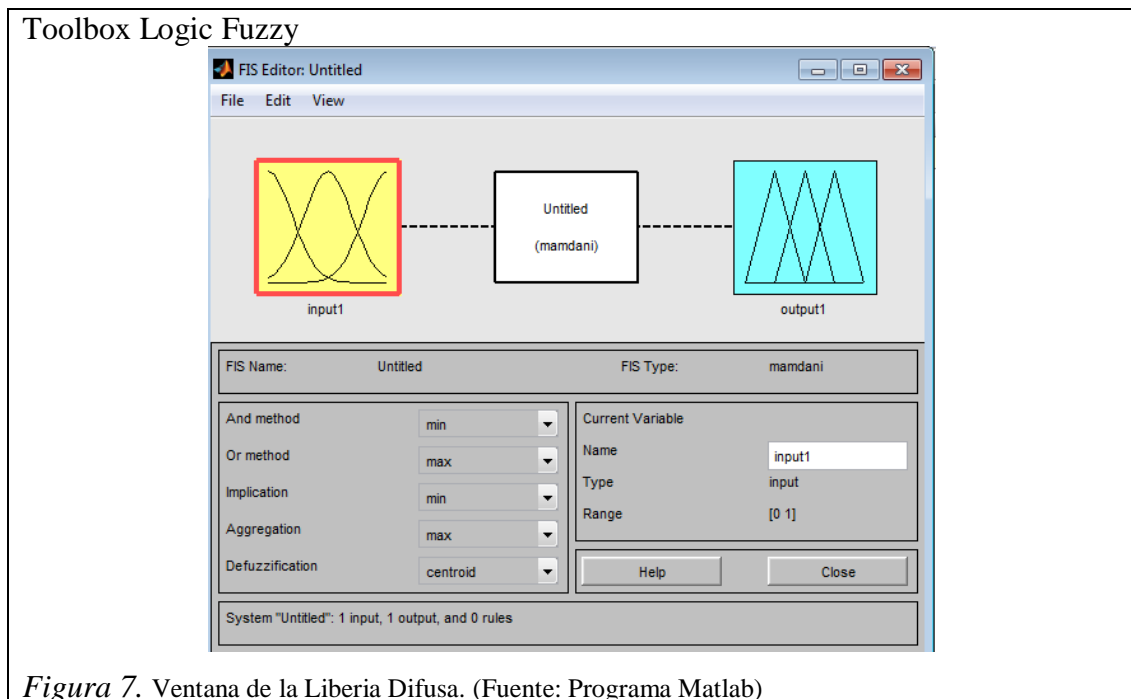


Figura 7. Ventana de la Librería Difusa. (Fuente: Programa Matlab)

4. Se establece los parámetros de los conjuntos difusos del error para las funciones de membrecía, como se muestra en la Figura 8.

Toolbox Logic Fuzzy

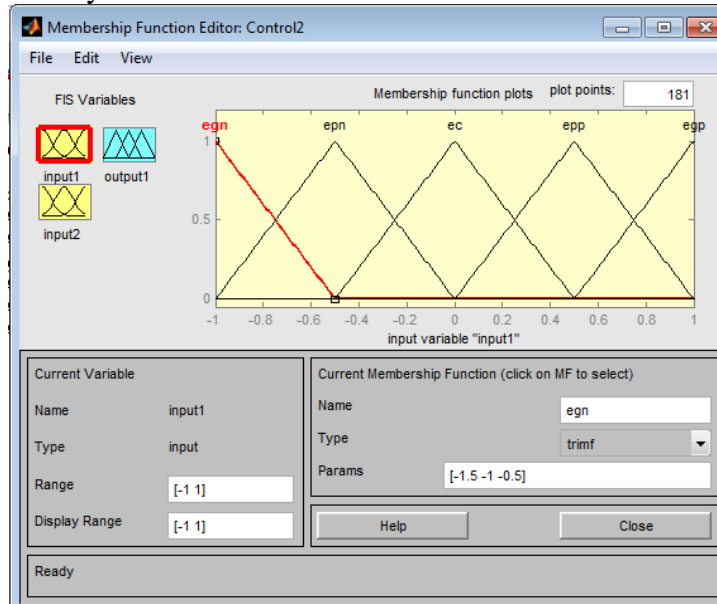


Figura 8. Funciones de membrecía para el error. (Fuente: Programa Matlab)

5. Se establece los parámetros de los conjuntos difusos de la derivada del error para las funciones de membrecía, como se muestra en la Figura 9.

Toolbox Logic Fuzzy

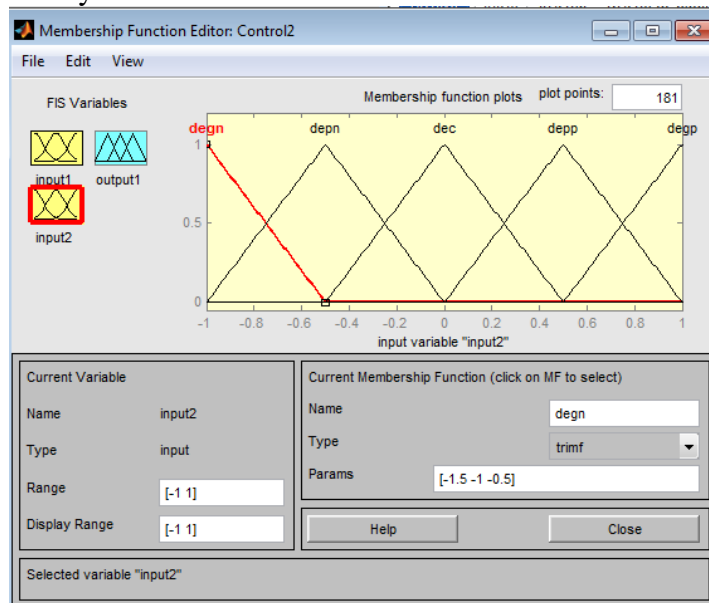
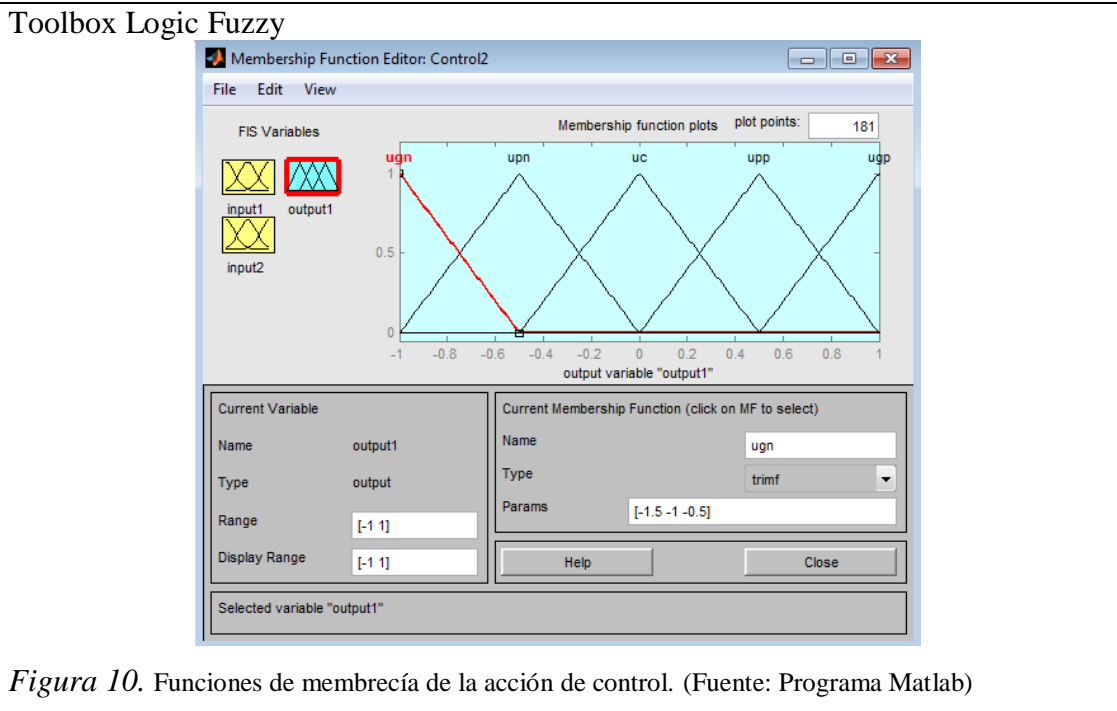


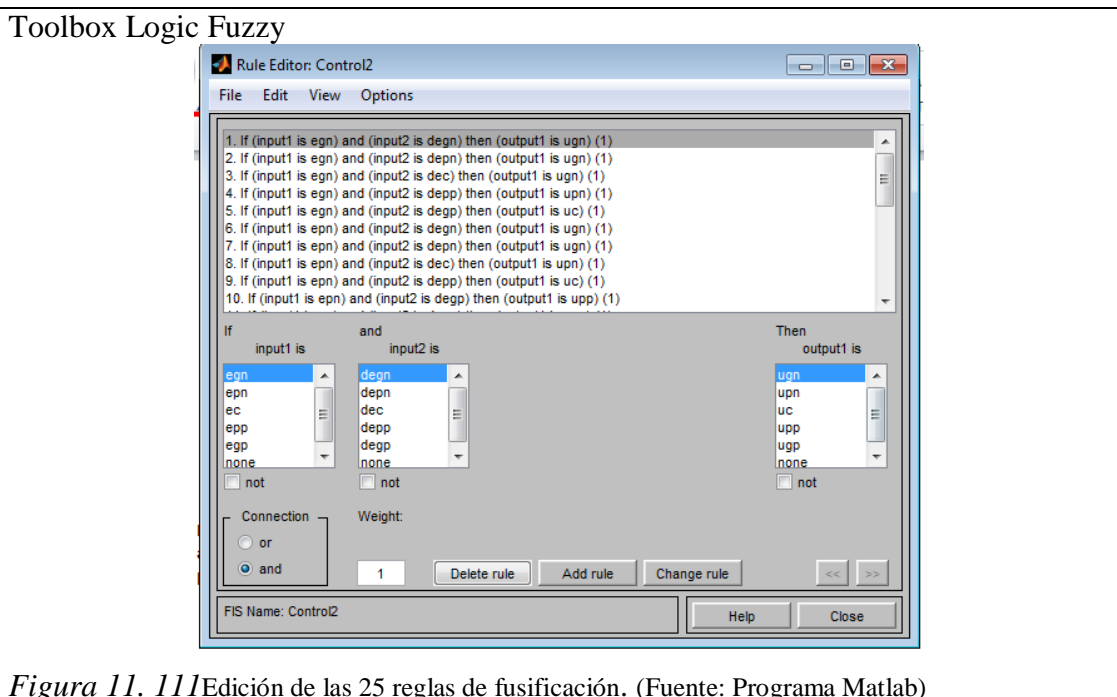
Figura 9. 110Funciones de membrecía para la derivada del error. (Fuente: Programa Matlab)

6. Se establece los parámetros de los conjuntos difusos de la acción de control para las funciones de membrecía, como se muestra en la Figura 10.



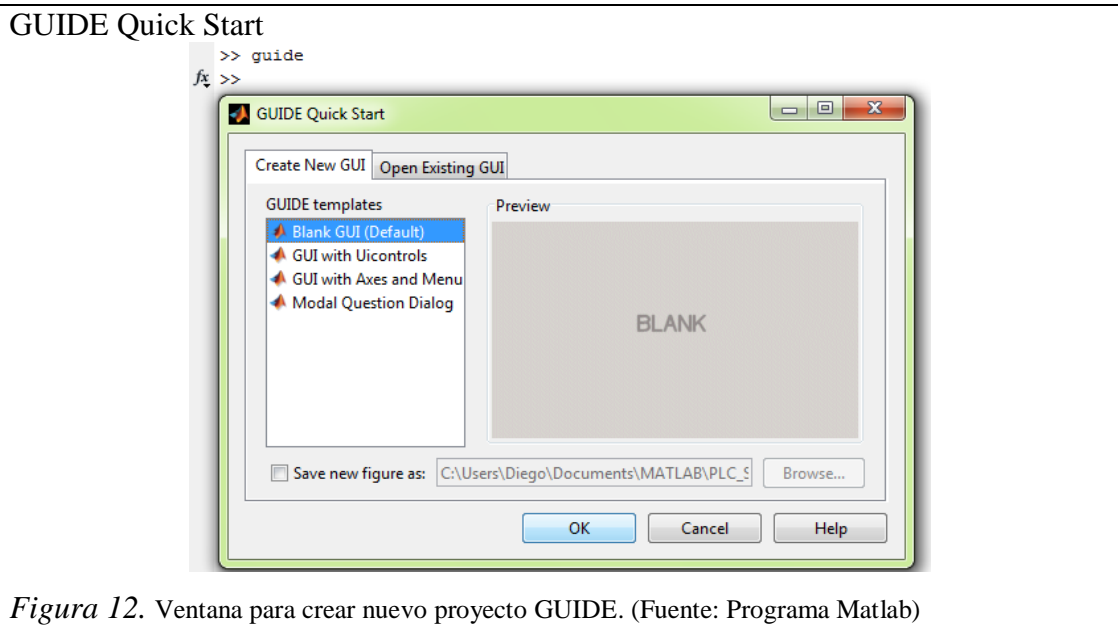
7. Se ingresa la base de reglas creadas para el controlador difuso, como se muestra en la Figura 11.

El número de conjuntos es de 5 por lo tanto, el número de reglas es de 25, generando así el hecho de contemplar más posibilidades y obtener una superficie más lineal. Un parámetro que se consideró como base para tomar otros conjuntos.

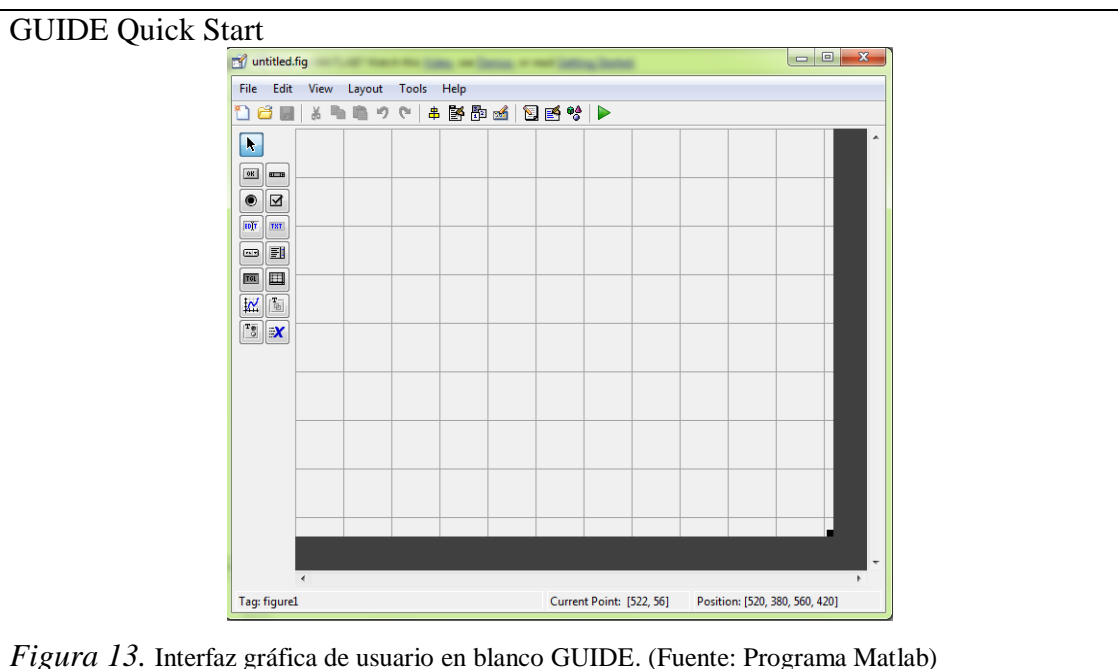


Para realizar el diseño del HMI del controlador PID difuso tanto para la variable de nivel como de temperatura, se utiliza la interfaz gráfica de usuario en Matlab GUIDE; se sigue los siguientes pasos:

1. Para crear un nuevo proyecto, en el espacio de trabajo de Matlab se introduce el comando “guide”, desplegando la ventana GUIDE Quick Start, como lo muestra la Figura 12.



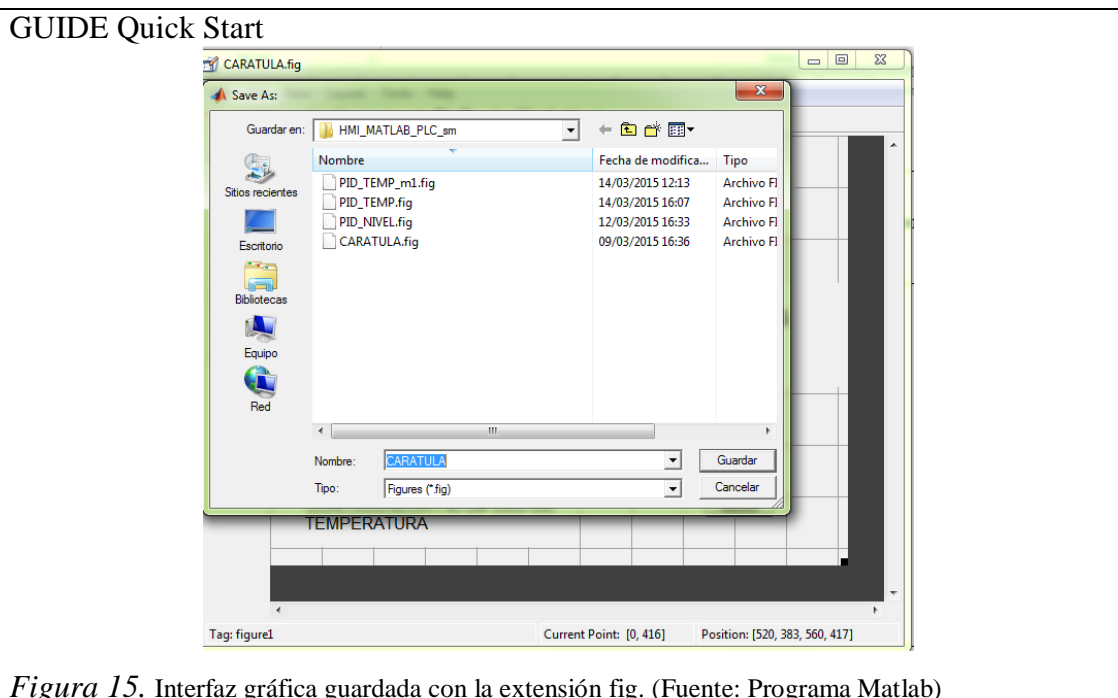
2. Abrir la interfaz gráfica de usuario en blanco, la que presenta un formulario nuevo en el que se puede diseñar un nuevo programa, como se muestra en la Figura 13.




3. En el entorno de diseño de GUIDE se añade cada uno de los componentes que se emplea en el diseño del controlador PID difuso, como se muestra en la Figura 14.



4. Al finalizar el diseño del HMI se guarda con el nombre CARATULA con la extensión fig, como se muestra en la Figura 15.



 UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR	PRÁCTICA NÚMERO 3
	CONTROL PID DIFUSO DE LAS VARIABLES DE NIVEL EN EL PLC SIEMENS S7-1200
LABORATORIO	CONTROL PID
CARRERA	ELECTRÓNICA
SEDE	QUITO

DATOS INFORMATIVOS

- a) MATERIA:
- b) No. DE PRÁCTICA: 3
- c) NÚMERO DE ESTUDIANTES:
- d) NOMBRE DEL INSTRUCTOR:

DATOS DE LA PRÁCTICA

a) TEMA:

Diseño del controlador PID Difuso de las variables de nivel y temperatura en el PLC Siemens S7-1200 para el control de procesos continuos.

b) OBJETIVO GENERAL:

- Diseñar un controlador PID Difuso de las variables de nivel y temperatura en el PLC Siemens S7-1200 para el control de procesos continuos.

c) OBJETIVOS ESPECÍFICOS:

- Diseñar e implementar el algoritmo de control PID difuso en el PLC Siemens S7-1200 para control de los procesos de nivel y temperatura.
- Comunicar el PLC Siemens S7-1200 con el TIA PORTAL mediante el protocolo de comunicación Ethernet para adquirir datos, ser procesados por el programa y actuar sobre la planta.
- Verificar el correcto funcionamiento de la comunicación mediante la observación de la lectura y escritura de variables para el control PID difuso de la planta de procesos continuos.

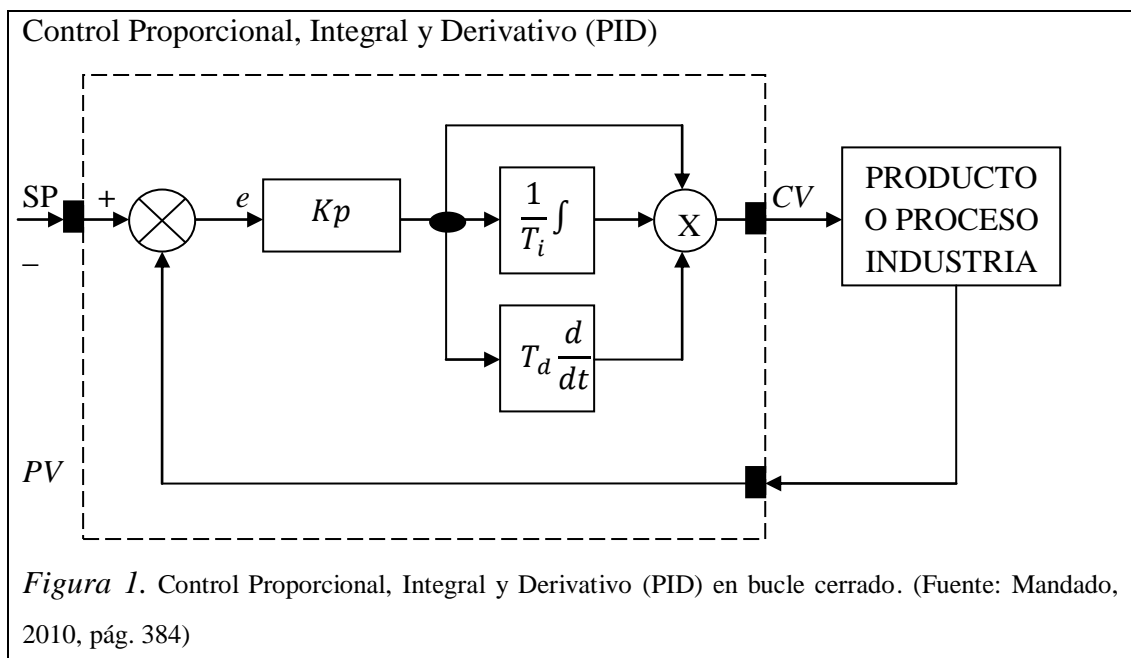
d) MARCO TEÓRICO

Controlador Proporcional Integral Derivativo – PID

Ogata, sustenta que más de la mitad de los controladores industriales que se usan hoy en día utilizan esquemas de control PID. La utilidad de los controles PID estriba en que se aplican en forma casi general a la mayoría de los sistemas de control. En particular, cuando el modelo matemático de la planta no se conoce y, por lo tanto, no se pueden emplear métodos de diseño analíticos, es cuando los controles PID resultan más útiles. En el campo de los sistemas para control de procesos, es un hecho bien conocido que los esquemas de control PID básicos y modificados han demostrado su utilidad para aportar un control satisfactorio, aunque tal vez en muchas situaciones específicas no aporten un control óptimo. (Ogata, 2010).

Estructura del PID

Mandado, representa la estructura de control más usada en el medio industrial, mostrando en la Figura 1 el diagrama de bloques de un regulador PID en el que la señal de control CV se obtiene mediante la suma de los tres tipos de acciones. El parámetro T_i , pondera la acción integral y el parámetro T_d pondera la acción derivativa. Además, la señal de error se amplifica mediante el parámetro K_p que afecta también a las acciones integral y derivativa. El diseñador debe asignar a los parámetros K_p T_i T_d un valor positivo o cero. (Mandado, 2010).



Lógica Difusa

Los sistemas basados en lógica difusa emulan la manera en que el cerebro razona o piensa, y en donde las aplicaciones se realizan en áreas multidisciplinarias que van desde la evolución tecnológica de los electrodomésticos, hasta programas computacionales para tomar decisiones y se han extendido a diversas áreas específicas como las cámaras de video, seguridad en base al monitoreo de la actividad volcánica, a través de controladores difusos para la calidad del agua, sistemas de operación automática de trenes, reactores nucleares y las transmisiones de automóviles.

Software de programación TIA PORTAL

Siemens, en su manual señala que TIA PORTAL ofrece un entorno confortable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como controladores y dispositivos HMI. Para poder encontrar la información necesaria, TIA PORTAL ofrece un completo sistema de ayuda en pantalla. (<http://www.siemens.com/automation/>).

TIA PORTAL proporciona lenguajes de programación estándar, que permiten desarrollar de forma cómoda y eficiente el programa de control.

- **KOP** (esquema de contactos): es un lenguaje de programación gráfico. Su representación se basa en esquemas de circuitos.
- **FUP** (diagrama de funciones): es un lenguaje de programación que se basa en los símbolos lógicos gráficos empleados en el álgebra booleana.
- **SCL** (lenguaje de control estructurado): es un lenguaje de programación de alto nivel basado en texto.

Al crear un bloque lógico, se debe seleccionar el lenguaje de programación que empleará dicho bloque. El programa de usuario puede emplear bloques lógicos creados con cualquiera de los lenguajes de programación.

El lenguaje de control estructurado SCL es un lenguaje de programación de alto nivel basado en PASCAL para las CPU de SIMATIC S7. SCL soporta la estructura de bloques de STEP 7. También es posible incluir bloques de programa escritos en SCL con bloques de programa escritos en KOP y FUP.

e) PROCEDIMIENTO

Para la programación del PID Difuso de la variable de nivel, se realiza los siguientes pasos:

1. Crear un nuevo proyecto seleccionando el CPU y el módulo de expansión de entradas y salidas análogas; establecer los parámetros de tensión o corriente de las entradas y salidas análogas.
2. Crear un bloque lógico FC llamado FUZZY, donde se programa la lógica difusa utilizando lenguaje de programación SCL, como se observa en la Figura 2.

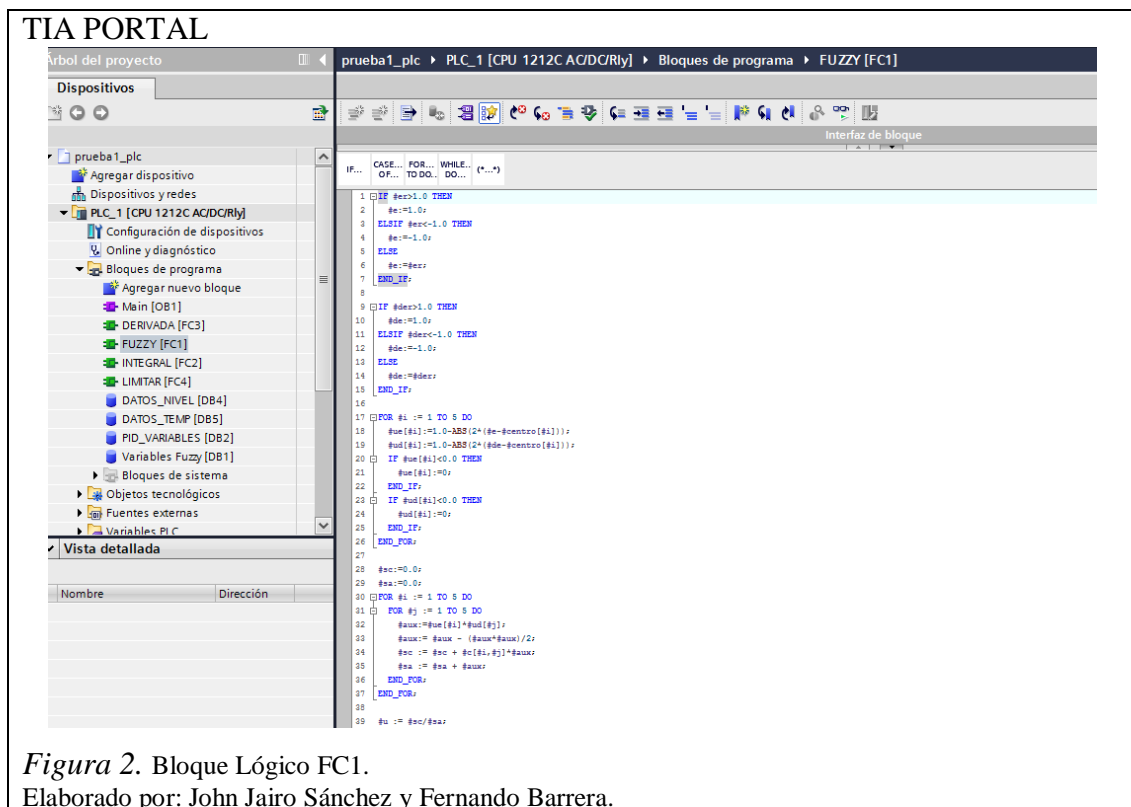


Figura 2. Bloque Lógico FC1.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Los pasos de programación del bloque FUZZY, se detallan a continuación:

- a. La Figura 3 muestra el código de programación que efectúa la limitación del error en los valores de menos uno a uno.

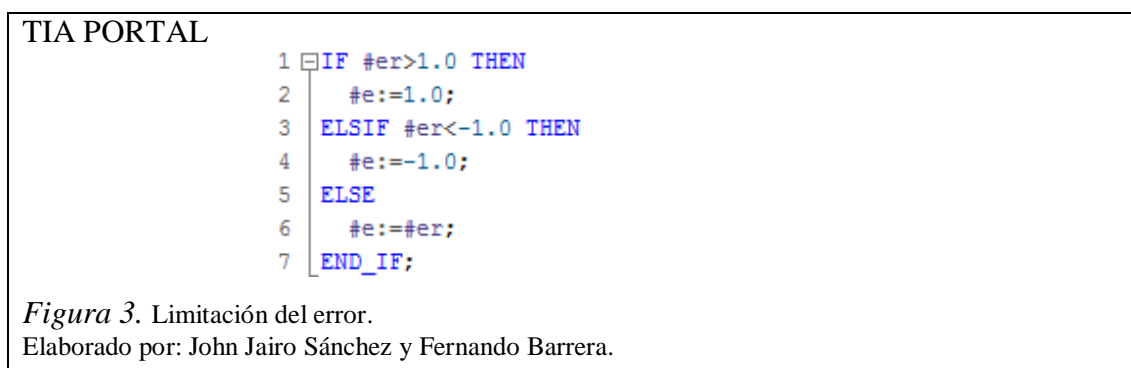


Figura 3. Limitación del error.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- b. La Figura 4 muestra el código de programación que efectúa la limitación de la derivada del error en los valores de menos uno a uno.

TIA PORTAL

```

9 IF #der>1.0 THEN
10   #de:=1.0;
11 ELSIF #der<-1.0 THEN
12   #de:=-1.0;
13 ELSE
14   #de:=#der;
15 END_IF;

```

Figura 4. Limitación de la derivada del error.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

- c. La Figura 5 muestra el código de programación que efectúa la máquina fusificadora, en la cual se evalúa el error y la derivada del error en las cinco funciones de pertenencia.

TIA PORTAL

```

17 FOR #i := 1 TO 5 DO
18   #ue[#i]:=1.0-ABS(2*(#e-#centro[#i]));
19   #ud[#i]:=1.0-ABS(2*(#de-#centro[#i]));
20 IF #ue[#i]<0.0 THEN
21   #ue[#i]:=0;
22 END_IF;
23 IF #ud[#i]<0.0 THEN
24   #ud[#i]:=0;
25 END_IF;
26 END_FOR;

```

Figura 5. Máquina fusificadora.

Elaborado por: John Jairo Sánchez - Fernando Barrera.

- d. La Figura 6 muestra el código de programación que efectúa la máquina desfusificadora, en donde calcula la acción de control en base a que función de pertenencia se encuentran el error y derivada del error, respectivamente.

TIA PORTAL

```

28 #sc:=0.0;
29 #sa:=0.0;
30 FOR #i := 1 TO 5 DO
31   FOR #j := 1 TO 5 DO
32     #aux:=#ue[#i]*#ud[#j];
33     #aux:= #aux - (#aux*#aux)/2;
34     #sc := #sc + #c[#i,#j]*#aux;
35     #sa := #sa + #aux;
36   END_FOR;
37 END_FOR;
38
39 #u := #sc/#sa;

```

Figura 6. Máquina desfusificadora.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

3. Crear un bloque de datos DB1 llamado Variables Fuzzy, donde se crea una matriz llamada **c**, con la base de reglas; y otra matriz llamada **centro**, con los centros de todas las funciones de pertenencia, como se observa en la Figura 7.

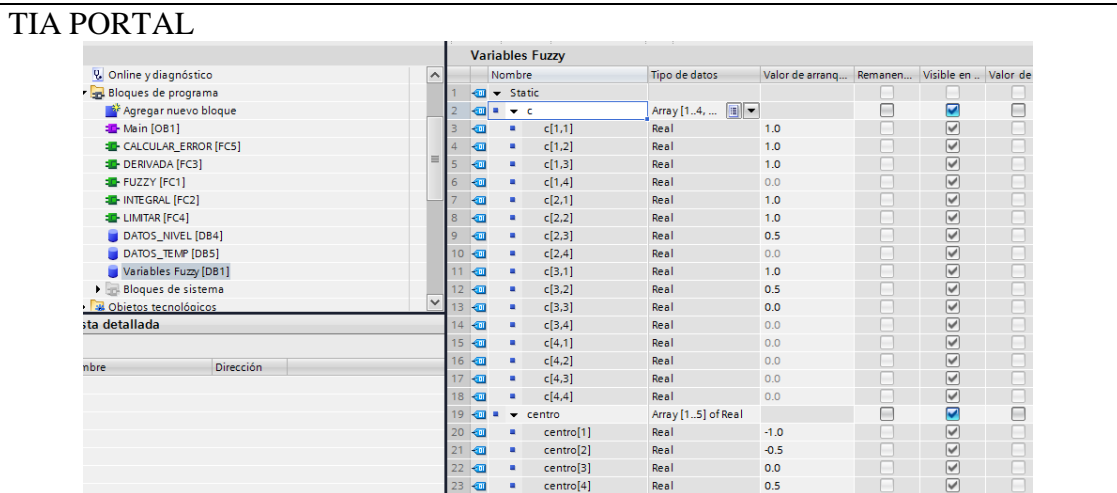


Figura 7. Bloque de datos DB1.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

4. Crear un bloque lógico FC5 llamado CALCULAR_ERROR, donde se programa el cálculo del error porcentual, como se observa en la Figura 8.

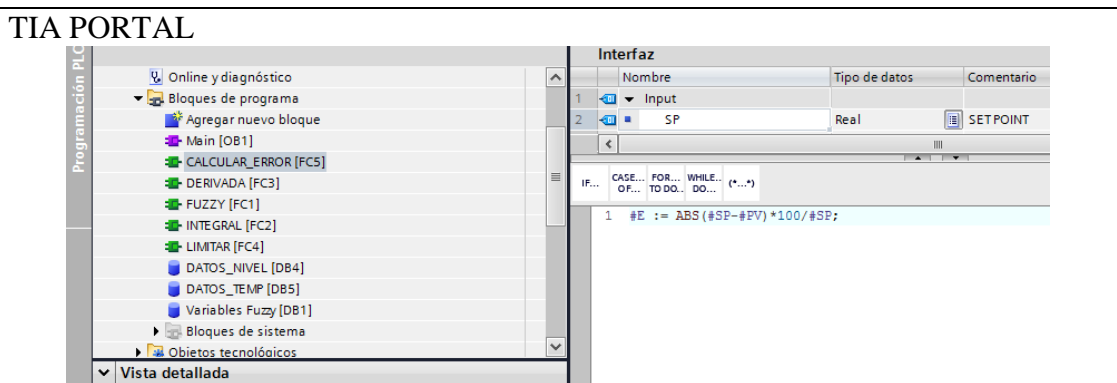


Figura 8. Bloque Lógico FC5.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

5. Crear un bloque lógico FC2 llamado INTEGRAL, donde se programa el cálculo de la integral discreta del error, en un tiempo de 100 milisegundos, como se observa en la Figura 9.

TIA PORTAL

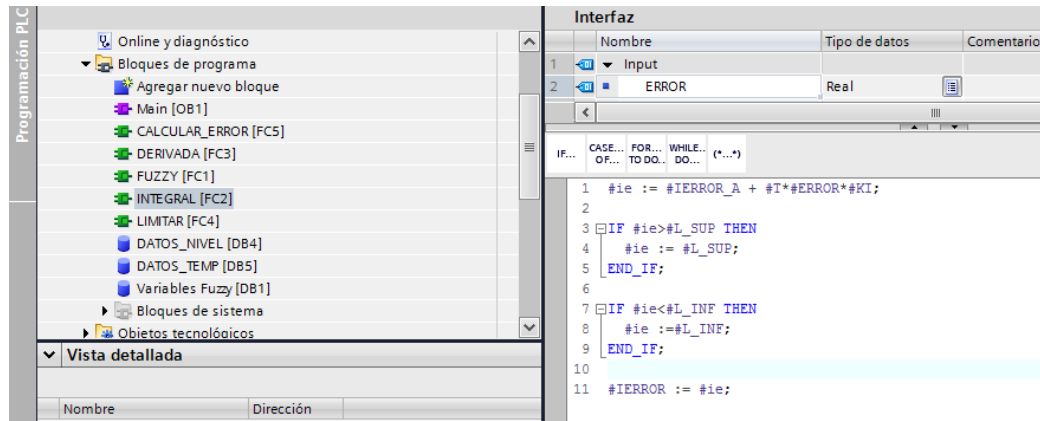


Figura 9112. Bloque Lógico FC2.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

6. Crear un bloque lógico FC3 llamado DERIVADA, donde se programa el cálculo de la derivada discreta del error, en un tiempo de 100 milisegundos, como se observa en la Figura 10.

TIA PORTAL

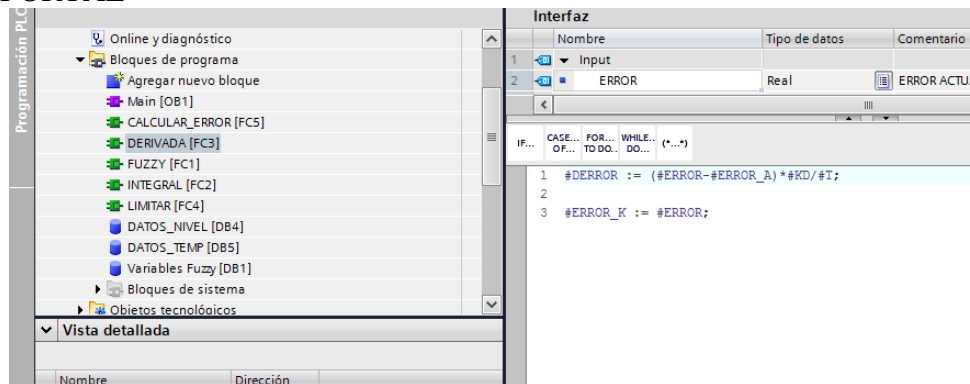


Figura 10113. Bloque Lógico FC3.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

7. Crear un bloque lógico FC4 llamado LIMITAR, donde se programa el límite de la señal de cero a uno, como se observa en la Figura 11.

TIA PORTAL

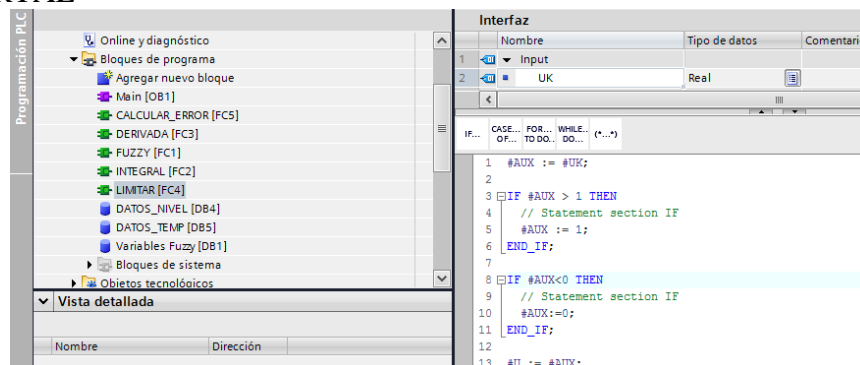


Figura 11. Bloque Lógico FC4.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

8. Crear un bloque de datos DB4 llamado DATOS_NIVEL, donde se crea todas las variables utilizadas para el controlador PID difuso de nivel, como se observa en la Figura 12.

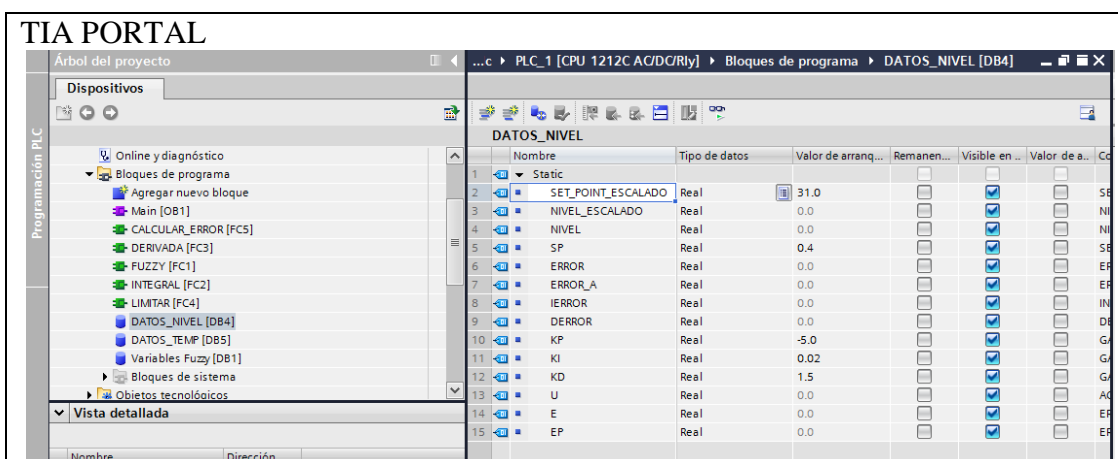


Figura 12/14. Bloque de Datos DB4.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

9. Crear un bloque de datos DB5 llamado DATOS_TEMP, donde se crea todas las variables utilizadas para el controlador PID difuso de temperatura, como se observa en la Figura 13.

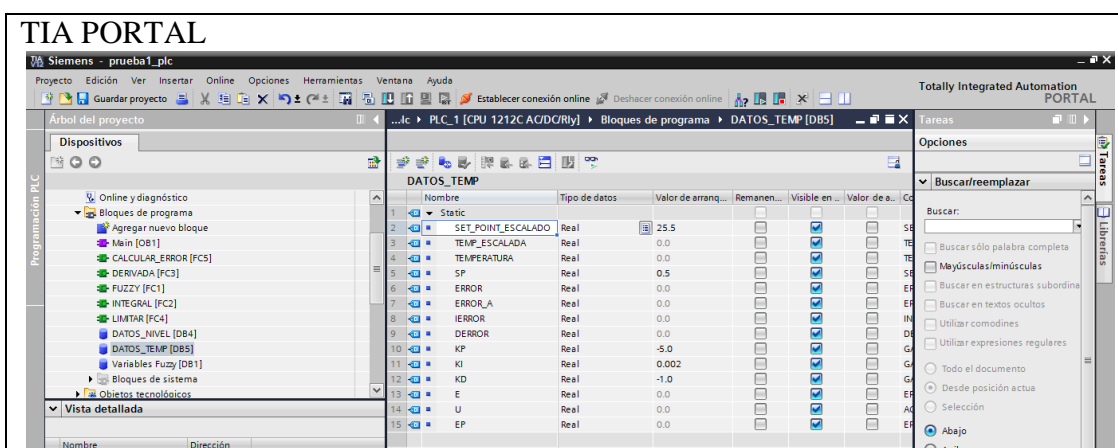


Figura 13. Bloque de datos DB5.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

10. En el bloque de organización Main (OB1) programar los controladores PID Difusos, donde se puede abrir los bloques creados FC y el DB; como se muestra en la Figura 14.

En el Anexo 1, se detalla la progración de los controladores PID Difusos de las variables de nivel y temperatura para el PLC Siemens S7-1200.

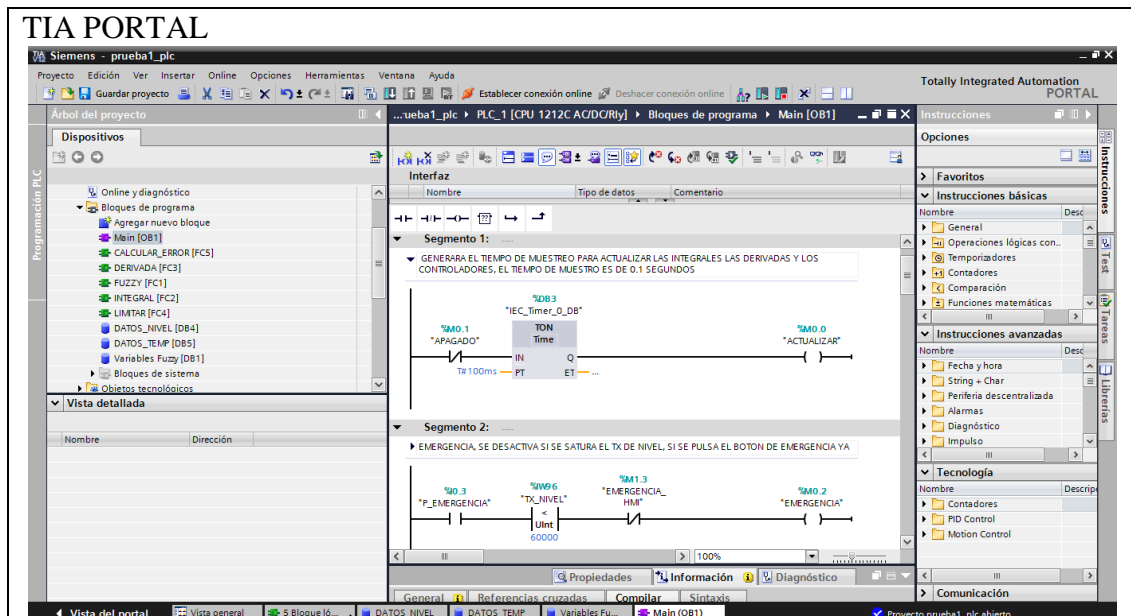


Figura 14. Bloque Main OB1.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Para realizar el diseño de la interfaz gráfica del controlador PID difuso tanto para la variable de nivel como de temperatura, se utiliza la interfaz gráfica propia del programa TIA PORTAL; se sigue los siguientes pasos:

1. En el programa TIA PORTAL, agregar un dispositivo HMI, como se muestra en la Figura 15.

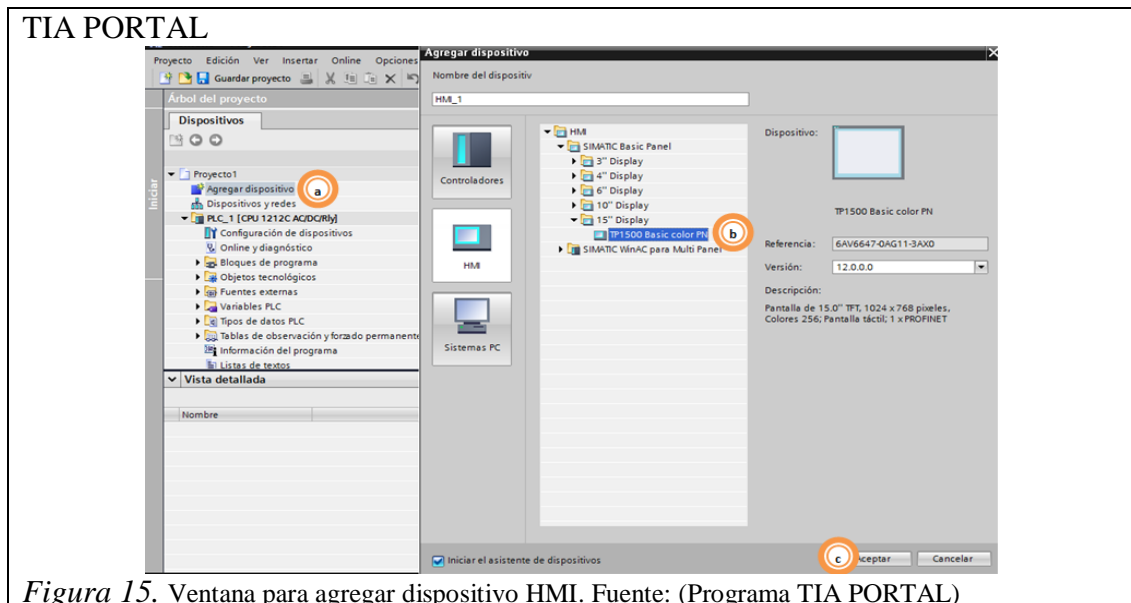


Figura 15. Ventana para agregar dispositivo HMI. Fuente: (Programa TIA PORTAL)

Los pasos para agregar el dispositivo HMI, se detallan a continuación:

- a. Seleccionar Agregar nuevo dispositivo.
- b. Seleccionar la pestaña HMI y escoger el modelo del dispositivo.
- c. Seleccionar Aceptar.

2. En la ventana asistente del panel de operador, se configura la conexión al PLC, como se muestra en la Figura 16.

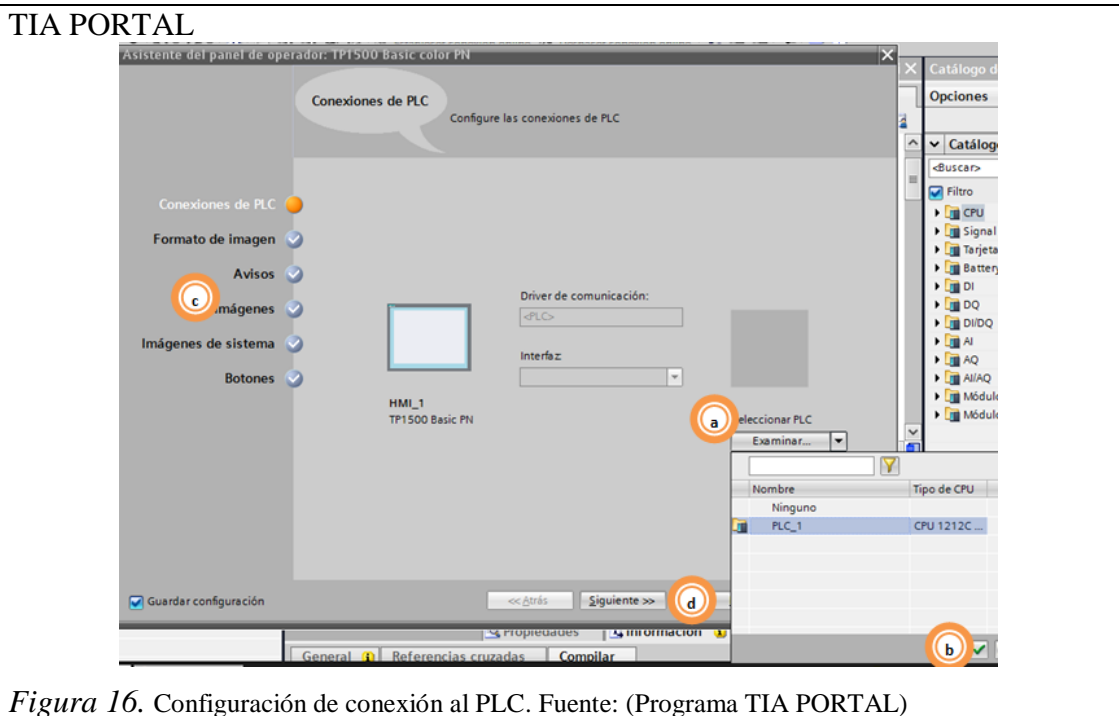


Figura 16. Configuración de conexión al PLC. Fuente: (Programa TIA PORTAL)

Los pasos para las conexiones del PLC, se detallan a continuación:

- a. Dar clic izquierdo en Seleccionar PLC.
 - b. Seleccionar el PLC a utilizar y el icono de visto.
 - c. Seleccionar el formato del HMI de acuerdo al requerimiento del proyecto a utilizarse.
 - d. Seleccionar Finalizar.
3. En la ventana de imagen raíz del HMI añadir cada uno de los componentes que se emplea en el diseño de la interfaz gráfica del controlador PID difuso, como se muestra en la Figura 17.

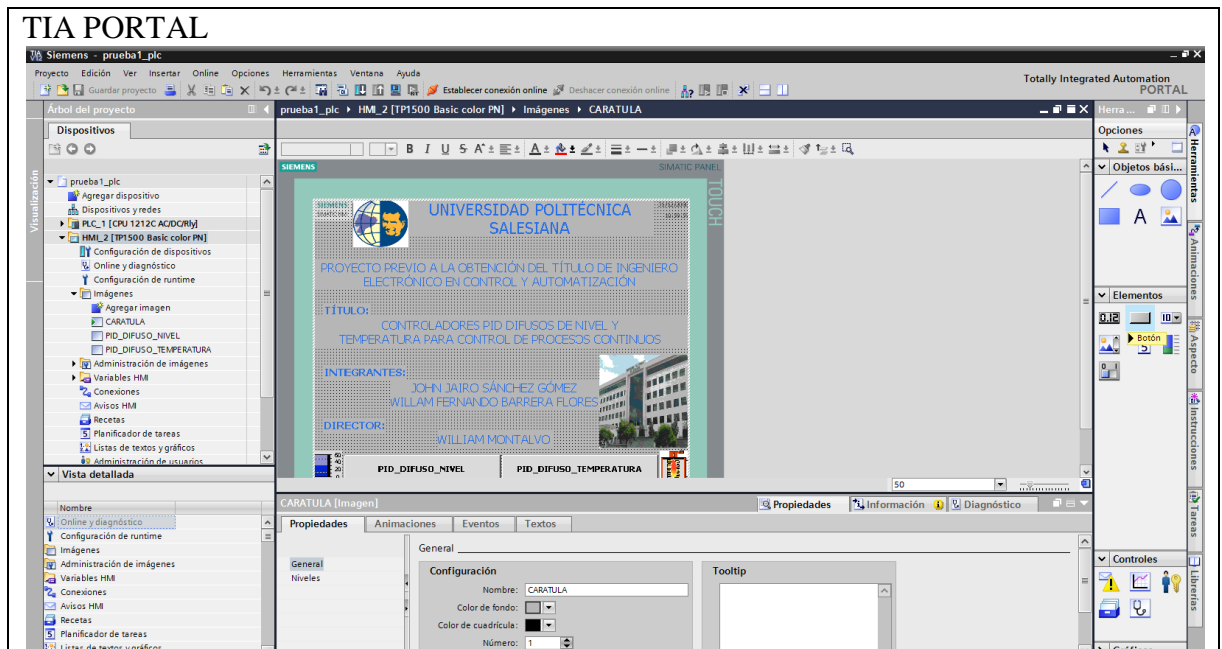


Figura 17. Ventana de imagen raíz de la interfaz gráfica del controlador PID difuso.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

4. Agregar una imagen para la interfaz gráfica controlador PID difuso de nivel, como se muestra en la Figura 18.

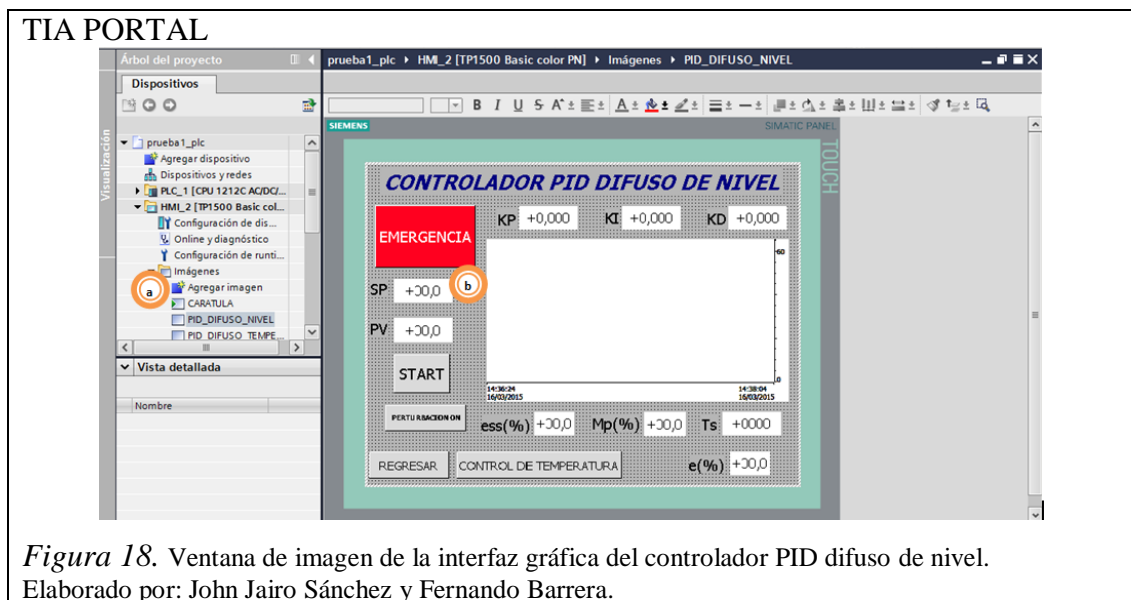


Figura 18. Ventana de imagen de la interfaz gráfica del controlador PID difuso de nivel.

Elaborado por: John Jairo Sánchez y Fernando Barrera.

Los pasos de la interfaz gráfica de nivel, se detallan a continuación:

- a. En la pestaña del dispositivo HMI dar doble clic izquierdo en agregar imagen.
- b. Añadir cada uno de los componentes que se emplea en el diseño del controlador PID difuso de nivel.

5. Agregar una nueva imagen para la interfaz gráfica controlador PID difuso de temperatura, como se muestra en la Figura 19.

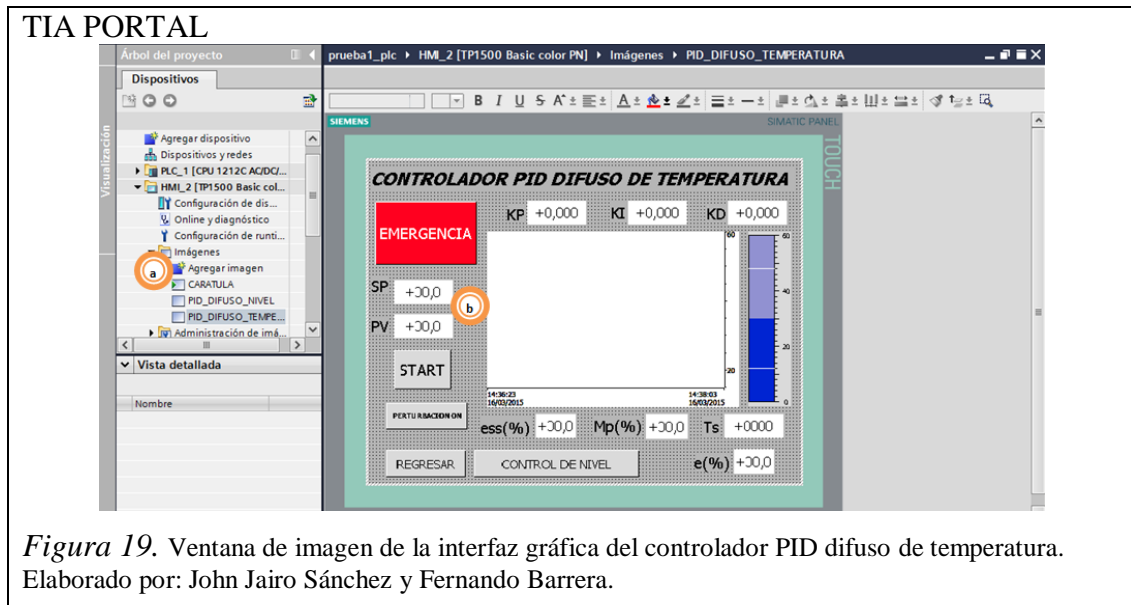


Figura 19. Ventana de imagen de la interfaz gráfica del controlador PID difuso de temperatura.
Elaborado por: John Jairo Sánchez y Fernando Barrera.

Los pasos de la interfaz gráfica de temperatura, se detallan a continuación:

- a. En la pestaña del dispositivo HMI dar doble clic izquierdo en agregar imagen.
- b. Añadir cada uno de los componentes que se emplea en el diseño del controlador PID difuso de temperatura.