



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**Implementación de un control PID utilizando una estación de caudal mediante el PLC
MicroLogix 1100 para prácticas de control de procesos.**

Chamorro Portilla, Jhon Alejandro

Departamento de Eléctrica y Electrónica

Carrera de Tecnología en Electrónica Mención Instrumentación y Aviónica

Trabajo de titulación, previo a la obtención del título de Tecnólogo en Electrónica
Mención Instrumentación y Aviónica

Ing. Alpúsig Cuichán, Silvia Emperatriz

25 de marzo de 2021

Latacunga



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

CERTIFICACIÓN

Certifico que la monografía, **“Implementación de un control PID utilizando una estación de caudal mediante el PLC MicroLogix 1100 para prácticas de control de procesos”** fue realizado por el señor **Chamorro Portilla, Jhon Alejandro** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 25 de marzo del 2021

Firma:

A handwritten signature in blue ink is written over a horizontal line. The signature is cursive and appears to read 'Silvia Emperatriz'.

Ing. Alpúsig Cuichán, Silvia Emperatriz.

Directora de proyecto

C.C.: 050277969-7












Análisis Urkund



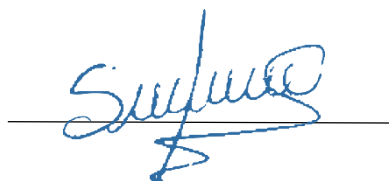
Document Information

Analyzed document	Monografia_Chamorro_Portilla_Jhon.pdf (D98799502)
Submitted	3/18/2021 6:41:00 PM
Submitted by	
Submitter email	jachamorro4@espe.edu.ec
Similarity	5%
Analysis address	sealpusig.espe@analysis.orkund.com

Sources included in the report

SA	Universidad de las Fuerzas Armadas ESPE / ESCRITO_WILMER_ABATA.pdf Document ESCRITO_WILMER_ABATA.pdf (D97781704) Submitted by: wpabata@espe.edu.ec Receiver: jpcalvopina1.espe@analysis.orkund.com	 2
W	URL: https://www.ingmecafenix.com/automatizacion/que-es-un-plc/ Fetched: 3/18/2021 6:42:00 PM	 1
SA	Universidad de las Fuerzas Armadas ESPE / HERRERA QUIMBITA MARJORIE STEPHANIA.pdf Document HERRERA QUIMBITA MARJORIE STEPHANIA.pdf (D98563660) Submitted by: msherrera5@espe.edu.ec Receiver: mlcajas.espe@analysis.orkund.com	 4
W	URL: https://core.ac.uk/download/pdf/190112637.pdf Fetched: 6/11/2020 8:18:27 PM	 1
W	URL: https://www.seika.com.mx/que-es-un-plc/ Fetched: 3/18/2021 6:42:00 PM	 1
W	URL: https://automantenimiento.net/electricidad/partes-de-un-plc/ Fetched: 3/18/2021 6:42:00 PM	 2
SA	Trabajo final de grado.docx Document Trabajo final de grado.docx (D44225265)	 1
W	URL: https://www.roydisa.es/wp-content/uploads/2013/01/controladores-programables-micro ... Fetched: 3/18/2021 6:42:00 PM	 4
W	URL: http://automation-networks.es/glossary/rslogix-500 Fetched: 3/18/2021 6:42:00 PM	 1
W	URL: https://repositorio.espe.edu.ec/bitstream/21000/7367/1/AC-ELET-ESPE-047424.pdf Fetched: 10/30/2019 7:58:13 AM	 1
W	URL: http://repositorio.upt.edu.pe/bitstream/UPT/348/1/Solorzano-Gary-Aranibar-Jorge.pdf Fetched: 12/5/2020 8:17:42 AM	 1

Firma:



Ing. Alpúsig Cuichán, Silvia Emperatriz.

Directora de proyecto

C.C.: 050277969-7



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGÍA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, **Chamorro Portilla, Jhon Alejandro**, declaro que el contenido, ideas y criterios de la monografía: **Implementación de un control PID utilizando una estación de caudal mediante el PLC MicroLogix 1100 para prácticas de control de procesos**, es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Latacunga, 17 de enero del 2021

Firma:

Chamorro Portilla, Jhon Alejandro

C.C.: 040167557-4



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE TECNOLOGIA EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

AUTORIZACIÓN DE PUBLICACIÓN

Yo, **Chamorro Portilla, Jhon Alejandro**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar la monografía: **“Implementación de un control PID utilizando una estación de caudal mediante el PLC MicroLogix 1100 para prácticas de control de procesos”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 17 de enero del 2021

Firma:

Chamorro Portilla, Jhon Alejandro

C.C.: 040167557-4

Dedicatoria

Este proyecto está dedicado principalmente a Dios por permitirme culminar un objetivo tan anhelado.

A mis padres que con su apoyo incondicional fueron mi inspiración y mis pilares fundamentales durante todo este proceso. Estoy orgulloso de ser su hijo, son los mejores.

A la universidad y a mis profesores por compartir sus conocimientos y dejarme ser parte de esta grandiosa institución.

Agradecimiento

Agradezco a Dios por bendecirme y guiarme a lo largo de mi vida y mi formación académica.

A mis padres por todo su amor, comprensión y apoyo, pero sobre todo gracias infinitas por la paciencia que me han tenido.

A mis amigos y compañeros con los que compartí muchas experiencias dentro y fuera de las aulas.

Tabla de contenidos

Carátula.....	1
Certificación.....	2
Análisis urkund	3
Autoría de responsabilidad	4
Autorización de publicación.....	5
Dedicatoria	6
Agradecimiento	7
Tabla de contenidos	8
Índice de tablas	12
Índice de figuras	13
Resumen	16
Abstract	17
Generalidades.....	18
Tema	18
Antecedentes.....	18
Planteamiento del problema	18
Justificación	19
Objetivos.....	19
<i>Objetivo general</i>	<i>19</i>
<i>Objetivos específicos</i>	<i>19</i>
Alcance.....	20
Fundamentación teórica	21
Proceso Industrial.....	21
Estación didáctica de control de procesos	21
Estación de didáctica de caudal	22
PLC (Programable Logic Controller)	23

<i>Características de los PLC</i>	23
<i>Partes de un PLC</i>	24
<i>Tipos de PLC</i>	25
<i>PLC virtual de CoDeSys</i>	27
<i>Resolución del PLC virtual de CoDeSys</i>	27
Marca comercial de PLC Allen Bradley	27
<i>MicroLogix 1100</i>	28
<i>Características MicroLogix 1100</i>	29
<i>Modelo de MicroLogix 1100</i>	29
<i>Versión Firmware MicroLogix 1100</i>	30
RSLogix 500	30
RSLinx	31
RSEmulate	32
Controlador PID	32
<i>Diagrama de bloques de un control PID</i>	33
<i>Características del control PID</i>	34
<i>Sobreimpulso</i>	34
<i>Tiempo de asentamiento</i>	35
<i>Bloque PID de RSLogix 500 para MicroLogix 1100</i>	35
<i>Descripción Bloque PID para MicroLogix 1100</i>	36
<i>Ecuación PID</i>	36
Bloque PID de CoDeSys	37
<i>Parámetros del bloque de función del control PID</i>	38
Método de sintonización de parámetros del control PID	39
Software Factory I/O	40
<i>Requisitos de Sistema</i>	40
Software CODESYS	41
<i>Requisitos para el software CoDeSys</i>	42

	10
POU'S	43
Variable Global	44
Variable local	44
Desarrollo del tema	45
Investigación del proyecto	46
Programación en el Software CoDeSys	46
<i>Creación de un nuevo proyecto en CoDeSys</i>	47
<i>Asignación de un nombre y ubicación del nuevo proyecto</i>	48
<i>Selección el dispositivo</i>	48
<i>Elección de tipos de lenguaje de programación</i>	48
<i>Asignación de variables globales</i>	50
<i>Creación de un POU</i>	51
<i>Operaciones para escalamiento de señales analógicas</i>	53
<i>Ecuación de la recta para escalamiento de señales</i>	53
<i>Programación principal</i>	55
<i>Lectura de la señal de sensor de flujo y escalamiento</i>	56
<i>Lectura de la señal de set point y escalamiento</i>	58
<i>Colocación del bloque de función para el control PID</i>	59
<i>Compilación e inicio de sesión del programa en CoDeSys</i>	61
<i>Creación de un HMI en CoDeSys</i>	61
<i>Creación de indicadores de datos en el HMI en CoDeSys</i>	68
Simulación en Factory I/O	75
Comunicación entre el Software CoDeSys y Factory I/O mediante OPC	77
Análisis de resultados	83
Conclusiones y recomendaciones	84
Conclusiones	84
Recomendaciones	85
Bibliografía	86

Anexos90

Índice de tablas

Tabla 1 <i>Parámetros del bloque de función del control PID</i>	38
Tabla 2 <i>Requisitos de sistema para Factory I/O</i>	41
Tabla 3 <i>Requisitos de sistema para software CoDeSys</i>	42

Índice de figuras

Figura 1 <i>Diagrama de bloques general del proyecto</i>	21
Figura 2 <i>Estación de caudal</i>	22
Figura 3 <i>Partes de un PLC</i>	24
Figura 4 <i>PLC Físico</i>	26
Figura 5 <i>PLC tipo software</i>	26
Figura 6 <i>PLC Virtual de CoDeSys</i>	27
Figura 7 <i>Logo Allen Bradley</i>	28
Figura 8 <i>MicroLogix 1100</i>	29
Figura 9 <i>Modelo de MicroLogix 1100</i>	30
Figura 10 <i>Software RSLogix 500</i>	31
Figura 11 <i>Logo de Software RSLinx</i>	31
Figura 12 <i>Logo RsEmulate 500</i>	32
Figura 13 <i>Gráfica del control PID</i>	33
Figura 14 <i>Diagrama de bloques de un control PID</i>	33
Figura 15 <i>Bloque PID en RSLogix 500</i>	35
Figura 16 <i>Bloque de función PID en CoDeSys</i>	37
Figura 17 <i>Logo de CoDeSys</i>	41
Figura 18 <i>POU de CoDeSys</i>	44
Figura 19 <i>Diagrama de bloques del proyecto</i>	45
Figura 20 <i>Entorno de trabajo de CoDeSys</i>	47
Figura 21 <i>Creación de un nuevo proyecto en CoDeSys</i>	47
Figura 22 <i>Nombre del proyecto</i>	48
Figura 23 <i>Elección de dispositivo</i>	48
Figura 24 <i>Elección de lenguaje de programación</i>	49
Figura 25 <i>Entorno de trabajo de CoDeSys</i>	49
Figura 26 <i>Agregar lista de variables globales en CoDeSys</i>	50
Figura 27 <i>Lista de variables globales</i>	51
Figura 28 <i>Creación de un POU en CoDeSys</i>	51
Figura 29 <i>Asignación de tipo de POU</i>	52
Figura 30 <i>Asignación de lenguaje de programación del POU</i>	53
Figura 31 <i>POU para el escalamiento de señales</i>	55
Figura 32 <i>Variables generadas en el POU</i>	55
Figura 33 <i>Enclavamiento marcha/paro</i>	56

Figura 34 Lectura de señal del sensor de flujo	56
Figura 35 Llamamiento del POU para el escalamiento de la señal del sensor de fluj.....	57
Figura 36 Guardado de señal escalada del sensor de flujo en una variable global.....	57
Figura 37 Escalamiento para datos en lt/min	58
Figura 38 Llamamiento del POU para el escalamiento de la señal del set point	58
Figura 39 Guardado de señal escalada del set point en una variable global	59
Figura 40 Control PID de la librería Oscat	60
Figura 41 Escalamiento para obtener litros\minuto del sensor de flujo	60
Figura 42 Botón de compilar de CoDeSys.....	61
Figura 43 Crear un HMI en CoDeSys.....	62
Figura 44 Asignación de nombre al HMI	62
Figura 45 Agregar imágenes en CoDeSys.....	63
Figura 46 Asignación de un nombre a la imagen a agregar	64
Figura 47 Pantalla inicial para agregar imágenes	64
Figura 48 Selección de imagen	65
Figura 49 Selección de imagen desde el computador	65
Figura 50 Imagen seleccionada en CoDeSys	66
Figura 51 V Imagen cargada con éxito	66
Figura 52 Vinculación de imágenes al HMI	67
Figura 53 Selección de imagen desde el HMI.....	67
Figura 54 Imagen insertada en el HM	68
Figura 55 Indicadores en el HMI de CoDeSys	68
Figura 56 Propiedades del indicador del set point.....	69
Figura 57 Propiedades del indicador de la variable de proceso	70
Figura 58 HMI implementado del CoDeSys	70
Figura 59 Indicadores para visualización de lt/min	71
Figura 60 Propiedades del indicador de lt/min del set point	71
Figura 61 Propiedades del indicador de lt/min de la variable de proceso	72
Figura 62 Colocación de grafica en el HMI	73
Figura 63 Asignación de la primera variable en la gráficaa	73
Figura 64 Asignación de la segunda variable en la gráfica	74
Figura 65 HMI Final	75
Figura 66 Selección de escena para control PID en Factory I/O.....	76
Figura 67 Escena de control PID en Factory I/O I	76
Figura 68 Inicio de sesión de PLC virtual	77

Figura 69 Vinculación PC con PLC virtual	77
Figura 70 OPC config	78
Figura 71 Ventana de OPC config	78
Figura 72 Configuración del OPC en Factory I/O	79
Figura 73 Selección del servidor OPC en Factory I/O.....	80
Figura 74 Conexión entre las variables Factory I/O y las variables de CoDeSy.....	80
Figura 75 Simulación del proyecto	81
Figura 76 Simulación del sistema con un SP de 51%.....	82
Figura 77 Simulación del sistema con un SP de 76 %	82
Figura 78 Gráfica de señales con un SP y PV de 76%	83

Resumen

Este proyecto describe originalmente la realización de una implementación y configuración de un control en PID en una estación de caudal del ubicada en el laboratorio de instrumentación virtual de la Universidad De Las Fuerzas Armadas-ESPE, mediante la utilización PLC y un HMI mediante un Panel View. Empleando el software RSLogix 500 para la programación en lenguaje tipo ladder del controlador y para el control PID simultáneamente. La cuarentena debido a la propagación del virus Covid-19 originó cambios a nivel global, debido a esto, el proyecto presentó cambios en su elaboración, se debió hacer todo de manera virtual, por el objetivo de salvaguardar la salud de todos, los laboratorios y aulas permanecieron sin acceso. Los programas que se empleó para llevar a cabo el proyecto fueron: CoDeSys, un software empleado para la programación y el HMI y Factory I/O para la simulación, mediante un servidor OPC propio de CoDeSys. Se empieza describiendo a detalle cada uno de los elementos y dispositivos electrónicos utilizados, luego se detalla paso a paso la programación del proyecto y su respectiva simulación que son favorables para familiarización con el tema del proyecto, se ha ocupado el control PID para el manejo del flujo de un líquido de una estación didáctica. Finalmente se presenta los resultados que dejó la elaboración del proyecto, así como también las conclusiones y recomendaciones que se sacaron en el transcurso de la realización de este trabajo de grado.

Palabras clave:

- **CONTROLADOR LÓGICO PROGRAMABLE**
- **PROPORCIONAL INTEGRAL DERIVATIVO**
- **INTERFAZ HOMBRE MÁQUINA**
- **OLE PARA CONTROL DE PROCESOS**

Abstract

This project originally describes the implementation and configuration of a PID control in a flow station located in the virtual instrumentation laboratory of the Universidad De Las Fuerzas Armadas-ESPE, using PLC and a HMI where a Panel View was used. Using RSLogix 500 software for programming the controller in ladder language and PID control simultaneously. The quarantine due to the Covid-19 pandemic originated changes at a global level, due to that this project presented changes in its elaboration, the whole project had to be done in a virtual way, since the laboratories where originally it would be carried out, were vetoed, it was impossible to enter due to the pandemic. The programs used to carry out the project were: CoDeSys, a software used for programming and the HMI and Factory I/O for the simulation, through CoDeSys' own OPC server. It begins by describing in detail each of the elements and electronic devices used, then step by step details the programming of the project and its respective simulation that are favorable for familiarization with the subject of the project, in this case the PID control has been used to manage the flow of a liquid either the filling or emptying of a tank. Finally, the results of the elaboration of the project are presented, as well as the conclusions and recommendations that were drawn during the course of this degree work.

Key words:

- **PROGRAMMABLE LOGIC CONTROLLER**
- **PROPORTIONAL INTEGRAL DERIVATIVE**
- **HUMAN MACHINE INTERFAZ**
- **OLE FOR PROCESS CONTROL**

Capítulo I

1. Generalidades

1.1. Tema

Implementación de un control PID utilizando una estación de caudal mediante el PLC MicroLogix 1100 para prácticas de control de procesos.

1.2. Antecedentes

Como lo plantea (Coba, 2005) en su publicación “El controlador PID fue patentado en 1939 por Albert Callender y Allan Stevenson de la firma Imperial Chemicals Limited (Northwich, Inglaterra). El controlador PID representó un enorme avance sobre los métodos de control automáticos previos”.

Según (García, 2013) indica en una investigación descriptiva publicada “El control PID es una excelente herramienta para lograr el ahorro de energía en sistemas de bombeo, temperatura u otras variables, ofreciéndole tecnología confiable y de vanguardia”. El control PID dispone de varias prestaciones y su investigación para poder realizar aplicaciones en otros tipos de controladores lógicos programables es favorable para los estudiantes que están actualmente en su formación profesional”.

Según (Cruz, 2010), quien publicó en su trabajo de titulación “DESARROLLO DE UN CONTROL PID WAVENET”, donde manifiesta “El control PID es una unidad para crear un lazo de control estable y alcanzar el desempeño deseado mediante tres tipos de acciones básicas de corrección de error: proporcional, integral y derivativa”.

1.3. Planteamiento del problema

La Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas “ESPE” sede en Latacunga, cuenta con laboratorios de automatización e instrumentación donde se realizan diferentes prácticas con controladores lógicos programables (PLC) satisfactoriamente gracias a la asesoría de los docentes que con su apoyo facilitan que los estudiantes lleguen a desarrollar el aprendizaje de mejor manera.

Sin embargo, la constante innovación y automatización de la tecnología de diferentes empresas industriales exige nuevos conocimientos de mejora de procesos dentro de formación profesional para ello, es necesario el uso de equipos actualizados en los

laboratorios, con el fin de que los futuros profesionales estén preparados frente a cada adversidad que se presente en el campo laboral y puede dar soluciones satisfactorias a cada caso.

Por lo tanto, la implementación de un control PID de una estación de caudal mediante un PLC y un HMI, será de gran ayuda para que los estudiantes que trabajen con controladores lógicos programables (PLC) adquieran conocimientos más amplios y por ende adquirieran más oportunidades a nivel profesional.

1.4. Justificación

Las empresas de producción hoy en día requieren profesionales capaces de desarrollar trabajos tanto de mantenimiento como de control de procesos, debido a esto es importante que los estudiantes sean preparados con bases actualizadas.

El proyecto a desarrollar permitirá a los estudiantes obtener nuevos conocimientos en el ámbito industrial, conocimientos modernos que les ayudará a desempeñarse de mejor en manera, teniendo un nivel más amplio de oportunidades de trabajo.

El proyecto pretende establecer la factibilidad de realizar un control PID con los dispositivos de control mencionados, la Panel View C600 cuenta con características novedosas además de tener un precio accesible y su funcionalidad es relativamente favorable para prácticas en laboratorio de instrumentación virtual.

1.5. Objetivos

1.5.1. Objetivo general

- Implementar un control PID utilizando una estación de caudal mediante el PLC MicroLogix 1100 para prácticas de control de procesos.

1.5.2. Objetivos específicos

- Investigar las características del PLC MicroLogix 1100 y la Panel View C600 Allen Bradley para desarrollo del proyecto.
- Desarrollar la lógica de programación de un control PID utilizando una estación de caudal para prácticas de control de procesos mediante el MicroLogix 1100.

- Elaborar HMI con el Panel View C600 Allen Bradley para el control y visualización del control PID mediante una página web.
- Verificar que el control PID esté funcionando correctamente y realizar un análisis de los resultados obtenidos.

1.6. Alcance

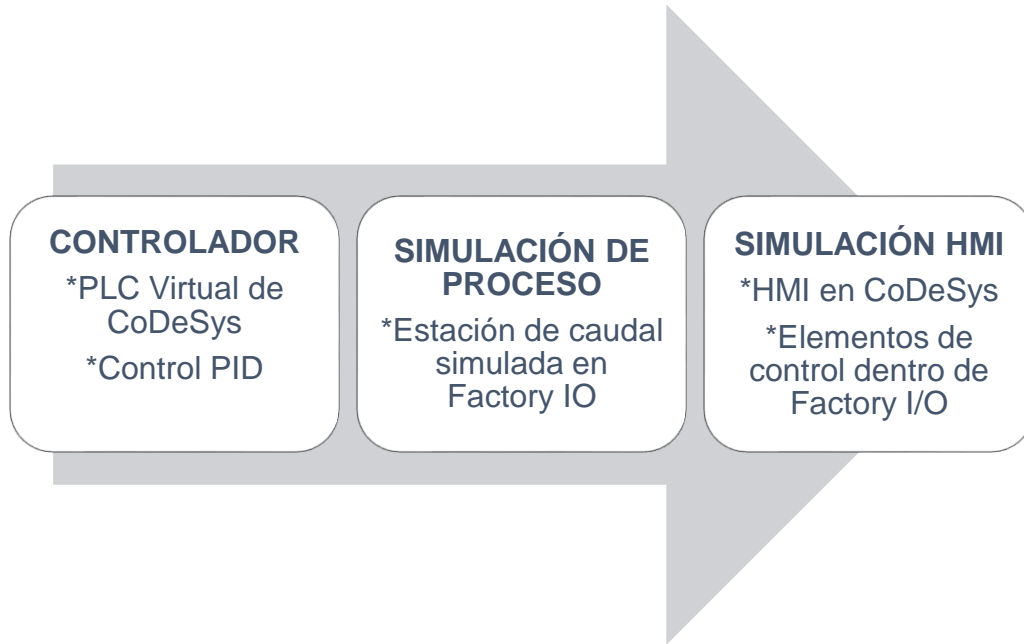
El proyecto está enfocado al control PID (Proporcional Integral Derivativo), donde se maneja el flujo de la estación de caudal ubicada en el laboratorio de instrumentación virtual de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas ESPE, mediante un controlador PLC Micrologix versión 1100 de la marca Allen Bradley y se realizará un HMI en una Panel View versión C600 de la marca Allen Bradley para el monitoreo y control de los valores de flujo y datos de la estación de caudal, el cual será programado (Micrologix 1100) desde el software RSlogix500 y una página web (Panel View C600).

Capítulo II

2. Fundamentación teórica

Figura 1

Diagrama de bloques general del proyecto



Nota. La imagen muestra el proceso del proyecto representado en un diagrama de flujo.

2.1. Proceso Industrial

Un proceso industrial se puede definir como un desarrollo sistematizado donde se transforma o transporta materia prima, en el cual se ocupan elementos electrónicos o electromecánicos como sensores, transmisores, actuadores, etc. que ocupan una tarea específica que también implica al seguimiento de pasos establecidos con el propósito de obtener un resultado esperado, ya sean productos, materiales, herramientas, sustancias que son necesarias para un público en concreto. (School O. B., 2006).

2.2. Estación didáctica de control de procesos

Es un equipo ocupado para la enseñanza en las instituciones superiores, acerca de control de procesos y automatización, donde se proporcionan las instalaciones necesarias para permitir el control de cualquier magnitud que se desee, representando distintos

procesos industriales utilizando materiales reales con el fin de promover la educación en el ámbito industrial. (Alberto, 2014)

Formada por varios elementos de toda naturaleza, ya sean elementos de control final, transmisores, sensores, actuadores, válvulas, un HMI, un PLC en su mayoría, se puede representar distintos escenarios, existen estaciones hechas por fabricantes de las marcas más conocidas o también se la puede crear desde cero. (Alberto, 2014)

El objetivo de las estaciones didácticas de control de procesos es el que los estudiantes puedan experimentar con las prácticas de control de procesos industriales, de esa manera aumentar el campo de conocimientos de cada estudiante, es una herramienta muy útil para el aprendizaje. (Atracen, 2017)

2.3. Estación de didáctica de caudal

Una estación de caudal es una instalación donde se puede observar el comportamiento de un líquido que pasa a través de una tubería para ser transportado por la misma a otro destino (ya sea un tanque o un reservorio), con el fin de poder regular la cantidad de líquido que se desea que pase por la tubería.

El proceso se realiza mediante la comunicación de dispositivos eléctricos y electrónicos hacia un controlador que recibe los datos de los transmisores y mediante la programación que se esté ejecutando manda a trabajar a las válvulas para que estas dejen pasar el líquido requerido por el usuario. Está compuesta por bombas, tuberías, tanques, transmisores, sensores, controladores, indicadores, válvulas, entre otros.

Figura 2

Estación de caudal.



Nota. La imagen muestra una estación de caudal de referencia. Tomado de (Jorge, 2017).

2.4. PLC (Programable Logic Controller)

Un PLC (Programable Logic Controller) o Control Lógico Programable en español es un dispositivo electrónico similar a una computadora industrial, la cual procesa los datos provenientes de sensores, botones, temporizadores y cualquier señal de entrada, para después controlar elementos de salida como actuadores, pistones, motores, válvulas, etcétera. De esa manera es posible que un PLC pueda controlar cualquier proceso industrial de manera automática. Es muy utilizado en procesos industriales por su capacidad de procesar datos en tiempo real, su capacidad de memoria, entre otras, tiene muchas aplicaciones en el ámbito industrial, su programación y software depende de la marca la cual lo elabora. Lo que los hace propicios para ser empleados en la industria son sus características físicas, las cuales le permiten ser resistente al ruido, las vibraciones y su capacidad de trabajar en diferentes espacios. (Méndez, Ingeniería Mecafenix, 2020).

De acuerdo con la definición de la NEMA (National Electrical Manufacturers Association) un PLC es: “Un aparato electrónico operado digitalmente, que usa una memoria programable para el almacenamiento interno de instrucciones para implementar funciones específicas, tales como lógica, secuenciación, registro y control de tiempos, conteo y operaciones aritméticas para controlar, a través de módulos de entrada/salida digitales (ON/OFF) o analógicos (1-5 VDC, 4-20 mA, etc.), varios tipos de máquinas o procesos”. (Sánchez, 2013).

Para que un PLC pueda procesar los datos que recibe y controlar cualquier sistema se necesita que este previamente programado para la tarea que va a realizar. Para poder programarlo se lo hace mediante un software que es específico dependiendo la marca del PLC y cada programa cuenta con diversos lenguajes de programación, como por la programación por lista de instrucciones, texto estructurado, tipo escalera, bloques de funciones, funciones secuenciales, en los cual se escribe instrucción por instrucción lo que se va a procesar y controlar. (Mecafenix, 2020).

2.4.1. Características de los PLC

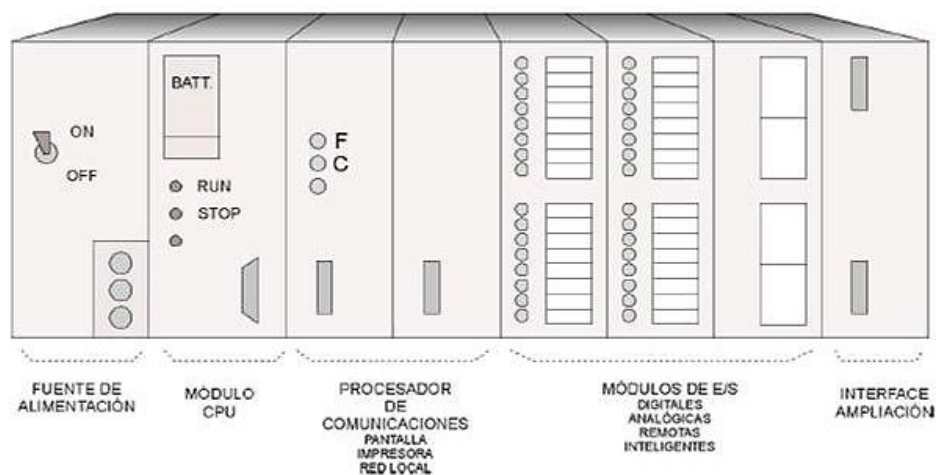
- Lectura de señales de captadores distribuidos.
- Comunicación con diferentes equipos en tiempo real.
- Interfaz que permite el uso y dialogo de los operarios.

- Conexión a un sistema supervisorio que facilita la interfaz y monitoreo.
- Reciben y ejecutan ordenes continuas por tiempos prolongados.
- Control de entradas y salidas distribuidas y ajenas al armario central autómeta mediante un cable red. (Seika, 2020).

2.4.2. Partes de un PLC

Figura 3

Partes de un PLC



Nota. La imagen muestra las partes que posee generalmente un PLC. Tomado de (Rojas, 2020).

- **Fuente de alimentación**

Encargada de proporcionar corriente continua a todos los circuitos eléctricos internos dentro del controlador lógico programable y demás tarjetas del PLC (si dispone). (Gomez, 2018)

- **Batería**

Se ocupa de provisionar de voltaje a la memoria RAM mientras el PLC este sin alimentación. Esta debe ser cambiada preventivamente, lo ideal es realizar el cambio en un periodo de 1 a 3 años con fin de evitar el desgaste y perder el programa localizado en la memoria RAM. (Gomez, 2018)

- **Módulo de memoria**

Se encarga de almacenar el programa en una memoria la cual puede ser de tipo volátil o no volátil. (Gomez, 2018)

- **CPU**

Es la unidad central de proceso, se encarga de ejecutar y controlar el programa, además de realizar todas las funciones aritméticas y logarítmicas, interpreta cada una de las instrucciones programadas, también se encarga de verificar la comunicación con los diferentes dispositivos. (Gomez, 2018).

- **Módulos de entrada**

Tienen la tarea de recibir las señales eléctricas provenientes de los equipos de la instalación que están controlando el proceso, o ya sea que los sensores conectados detectaron cambio de estado y el CPU pueda procesar los datos. (Gomez, 2018)

- **Módulo de salida**

Los datos procesados en el CPU los envía este módulo en forma de las señales eléctricas a los dispositivos actuadores que estén dentro de la instalación para la ejecución de sus tareas y el cumplimiento del proceso programado. (Gomez, 2018)

- **Puertos de comunicación**

Es canal para la comunicación entre el PLC con los otros dispositivos, periféricos, unidades de programación o con la interfaz HMI. (Gomez, 2018).

2.4.3. Tipos de PLC

2.4.3.1. PLC Físicos

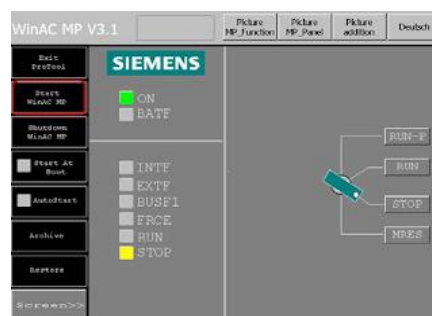
Son el tipo de controlador con más popularidad en el ámbito de la automatización, posee características que lo hacen apto para ambientes industriales como, resistencia al ruido, resistencia al polvo y humedad, su velocidad de procesamiento de datos en tiempo real, su tamaño facilita el posicionamiento del mismo en las instalaciones de sistemas de control, conforman una sola unidad, cuentan con CPU, entradas y salidas, batería, memoria, fuente de alimentación y controladores analógicos. Son fáciles de encontrar y su distribución es global, (ClasificacionDe, 2019)

Figura 4*PLC Físico*

Nota. En la imagen se muestra un PLC físico de la marca Allen Bradley. Tomado de (Automation, Rockwell Automation, 2005).

2.4.3.2. PLC tipo software

Es un tipo de PLC muy moderna, el más moderno de todos, es en resumen un PLC virtual el cual se ejecuta en un ordenador, su principal característica y se podría decir también que es novedosa es su capacidad de controlar las entradas y salidas de forma remota, para las comunicaciones se utilizan los puertos de comunicación ethernet, COM o unas tarjetas especiales. Una de las desventajas es que podría verse disminuida la velocidad de control de los procesos, también no está comprobado que el PLC virtual no genere conflicto al momento de trabajar simultáneamente con otros programas como el HMI. (Santiago, 2017).

Figura 5*PLC tipo software.*

Nota. En la imagen de muestra un ejemplo de un PLC tipo software. Tomado de (Santiago, 2017).

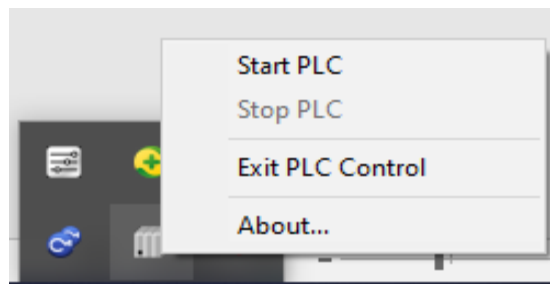
2.4.3.3. PLC virtual de CoDeSys

CoDeSys dentro de sus características cuenta con un PLC virtual, el cual viene incluido en el paquete de instalación, hace posible el poder simular los programas que se estén realizando y de esa manera el poder verificar el funcionamiento de los mismos, esa es una de las razones por la cual se escogió este software para la consecución del proyecto al mismo que tiempo que es la razón que lo hace uno de los softwares más completos para programar.

En la versión de CoDeSys V3.5.10.4 el PLC virtual se debe encenderlo desde la barra de tareas de la computadora y posteriormente entablar comunicación entre ambos, pasos que se muestran el desarrollo del proyecto.

Figura 6

PLC Virtual de CoDeSys.



Nota. En esta imagen se muestra la forma de encender el PLC virtual de CoDeSys.

2.4.3.3.1. Resolución del PLC virtual de CoDeSys

El PLC virtual de CoDeSys, de la versión 3.5.10.4 e instalado en una computadora de 64 bits, tiene una resolución de 15 bits, de acuerdo a esa resolución, el máximo valor que puede leer el PLC virtual es **32767**, dato importante a tomar en cuenta al momento de realizar el escalamiento de la señal, la resolución de bits puede cambiar con las versiones de CoDeSys y de los bits de la computadora. (Store, 2017)

2.5. Marca comercial de PLC Allen Bradley

Allen Bradley es una de las marcas que actualmente conforma la línea de equipos automatizado de Rockwell Automation. Fue fundada a principios del siglo XX por el Dr. Staton Allen y Lynde Bradley. Al ser una marca enfocada en la automatización, sus productos con mayor demanda son precisamente aquellos relacionado con esta área. PLC's, HMI, sensores, drivers y contactores son algunos de ellos. (Autracen, 2017).

Conforme la tecnología avanzó, Allen Bradley buscó dar soluciones a las nuevas exigencias que el mercado iba solicitando día con día. Uno de los problemas que tenían las empresas aquellos días eran los relacionados con el control y cableado de una inmensa cantidad de actuadores, sensores y demás dispositivos que se necesitaban implementar. La solución (como ya hemos visto en otras entradas del blog) fue la creación de redes industriales. Con aquello en mente, la empresa empezó a crear diversos tipos de redes que satisficiesen las diversas demandas que el mercado exigía. . (Autracen, 2017).

Figura 7

Logo Allen Bradley



Nota. En la imagen se muestra el logo de la marca Allen Bradley. Tomado de (Automation, Rockwell Automation, 2005)

2.5.1. MicroLogix 1100

Es un PLC (Controlador Lógico programable) elaborado por la marca Allen Bradley, es uno de los numerosos modelos que posee la marca, este PLC posee numerosas prestaciones en el ámbito industrial, entre las más comunes tenemos el control industrial.

Es el recién miembro de la familia MicroLogix, diseñado para ampliar la cobertura de aplicaciones mediante entradas analógicas incorporadas, comunicaciones Ethernet y capacidades de visualización también incluye edición en línea. Cada MicroLogix 1100 contiene dos entradas analógicas incorporadas, con 10 entradas digitales y 6 salidas digitales. La pantalla de cristal líquido incorporada le permite monitorear el estado de las E/S y del controlador, así como hacer cambios a datos de enteros y de bits. El controlador se puede actualizar fácilmente con el firmware más reciente mediante una descarga desde un sitio web. (Automation, Rockwell Automation, 2005)

Figura 8*MicroLogix 1100*

Nota. En la imagen se muestra la parte frontal del PLC MicroLogix 1100. Tomado de (Automation, Rockwell Automation, 2005)

2.5.1.1. Características MicroLogix 1100

- Incluye un puerto EtherNet/IP™ de 10/100 MBps incorporado para mensajería entre dispositivos similares.
- Proporciona una memoria de 8 KB (4 KB de programas de usuario con 4 KB de datos de usuario).
- Permite el acceso, el monitoreo y la programación desde cualquier conexión Ethernet.
- Admite la edición en línea.
- Proporciona un servidor web incorporado que permite configurar los datos del controlador para que aparezcan como una página web.
- Contiene un puerto combinado RS-232/RS-485 aislado para comunicación en serie y conectada en red.
- Permite monitorear y modificar los datos del controlador a través de una pantalla LCD incorporada.
- Compatible con módulos de expansión de E/S MicroLogix 1762 (hasta cuatro módulos por controlador).
- Admite un máximo de 144 puntos de E/S digitales. (Networks, 2009)

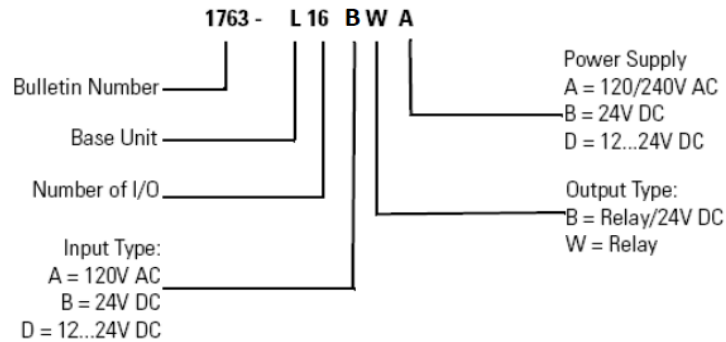
2.5.1.2. Modelo de MicroLogix 1100

El PLC MicroLogix 1100 es un microcontrolador muy utilizado y dentro de la gama 1100 hay diferentes modelos con características mínimas, dentro de estos existen cuatro:

L16AWA, L16BWA, L116BBB, L16DWD. El que se utiliza en este proyecto es el L16BWA.
(Automation, Rockwell Automation, 2005)

Figura 9

Modelo de MicroLogix 1100.



Nota. En la imagen se muestra el modelo del PLC MicroLogix 1100. Tomado de (Automation, MicroLogix Programmable Controller Selection Guide, 2008)

2.5.1.3. Versión Firmware MicroLogix 1100

El firmware es un programa informático existente y funcional en dispositivos que trabajan por hardware y software. Está ubicado en la memoria ROM del dispositivo y se encarga de controlar los circuitos electrónicos, conocida también como el nivel más bajo dentro del dispositivo.

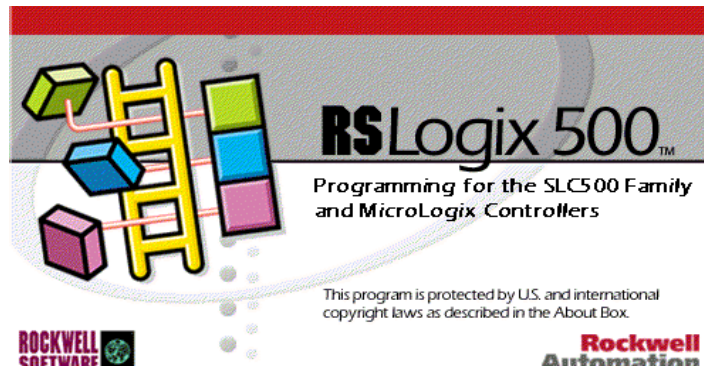
En los PLCs el firmware puede ser actualizado por personal capacitado, pero siempre existe riesgo de dañar el dispositivo. El MicroLogix 110 utilizado en este proyecto tiene de versión 16 de firmware de Allen Bradley. (Alberto, 2014)

2.6. RSLogix 500

RSLogix 500 es un software elaborado por la marca Rockwell Automation para programar PLC's Allen Bradley exclusivamente. Este software ha sido elaborado para trabajar con el sistema operativo Microsoft®, siendo compatible en muchas de sus versiones, proporciona el soporte para la programación de los PLC's de Allen Bradley tipo SLC500 y MicroLogix. RSLogix 500 fue el primer software de programación PLC® para ofrecer una incomparable productividad con una interfaz de usuario líder en la industria. (Networks, 2009)

Figura 10

Software RSLogix 500.



Nota. En la imagen se muestra la portada del software RsLogix 500. Tomado de (Gurung, 22).

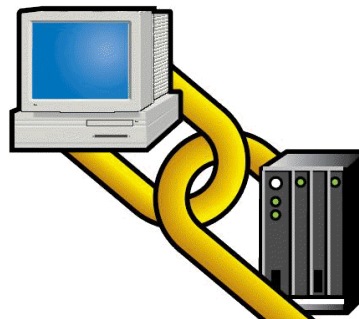
2.7. RSLinx

Es un software para controladores lógicos programables de la marca Rockwell Automation, es utilizado para realizar redes de comunicaciones industriales, y también para que los PLCs puedan acceder a una gran variedad de aplicaciones de Rockwell Software y Allen Bradley.

Entre estas aplicaciones se incluyen desde aplicaciones de configuración y programación tales como RSLogix y RSNetWorx hasta aplicaciones HMI (interfaz operador-máquina) como RSView32, hasta sus propias aplicaciones de adquisición de datos mediante Microsoft Office, páginas Web o Visual Basic. (Automation, Rockwell Automation, 2005)

Figura 11

Logo de Software RSLinx.



Nota. En la imagen se muestra el ícono del software RSLinx. Tomado de (Alberto, 2014)

2.8. RSEmulate

El software de emulación RSLogix Emulate 500 es una herramienta de solución de problemas y depuración, fue creado para mejorar los procesos de validación de los programas de usuario en el desarrollo de los proyectos de automatización industrial basados en los PLC's de Allen Bradley tipo SLC500 y Micrologix, trabajando en un entorno bajo el sistema operativo Microsoft Windows 95 o NT.

Ejecuta sus programas de lógica de escalera en su computadora, actualizando las tablas de datos de sus programas, permitiéndole aproximarse a lo que sucederá cuando descargue sus programas a procesadores físicos PLC-5. RSLogix Emulate utiliza la CPU de su computadora para escanear los peldaños en su programa de escalera. Los peldaños en su programa leen entradas y escriben salidas en la tabla de datos almacenada fuera de línea con su proyecto de lógica de escalera. (Automation, Rockwell Automation, 2005).

Figura 12

Logo RsEmulate 500



Nota. En la imagen se muestra el ícono del software RSLogix Emulated 500. Tomado de (Networks, 2009).

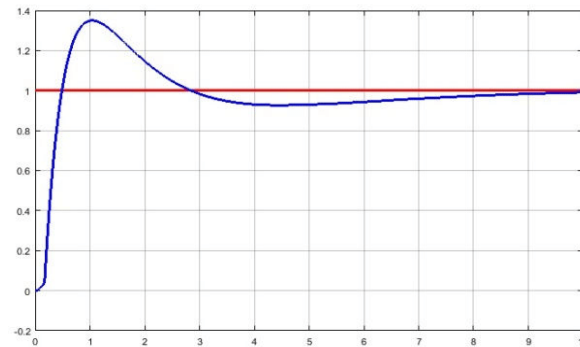
2.9. Controlador PID

Es un mecanismo que permite el manejo de diferentes variables como la temperatura, presión, flujo y otras variables más, todo esto a través de sistemas de control de lazo cerrado o de retroalimentación. Es uno de los métodos de control automático más frecuentes y precisos. El control PID está compuesto por tres modos de control que lo forman: proporcional (P), integral (I) y derivativo (D). Esta acción combinada reúne las ventajas de cada uno de los tres controles individuales.

Su uso en la actualidad ha crecido de manera significativa desde su aparición en los años treinta, Según (Moreno, 2001) afirma “En la industria el 95% de los lazos de control que existen en las aplicaciones industriales son del tipo PID, lo que muestra la preferencia del usuario en el uso de leyes de control simples”.

Figura 13

Gráfica del control PID

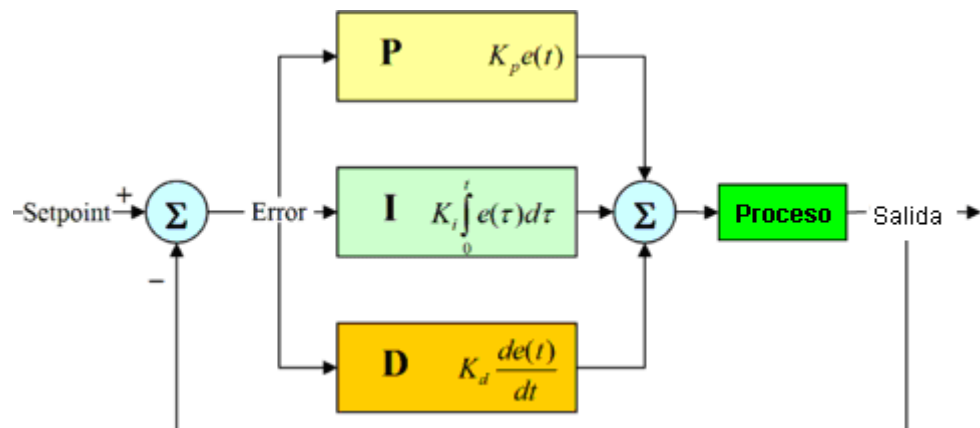


Nota. En la imagen se muestra la gráfica del control PID. Tomado de (Cero, 2016)

2.9.1. Diagrama de bloques de un control PID

Figura 14

Diagrama de bloques de un control PID



Nota. En la imagen se muestra el diagrama de bloques del control PID. Tomado de (Cando, 2008)

2.9.2. Características del control PID

- Modo automático y manual

Existen ocasiones donde el operario de la planta necesita que el sistema trabaje en modo manual, sobre escribiendo cada acción esto sucede cuando ocurren arranques o paradas en la planta inesperados. (Villajulca, 2018)

- Seguimiento de set point y de salida

Hacer el seguimiento de set point y de salida evita que cuando se presente cambios de modo automático a modo manual, no haya ningún cambio brusco en la salida, es uno de los métodos más utilizados. (Villajulca, 2018)

- **Alarmado**

Cada vez es más normal que los sistemas lleven esta función para alertar a los operarios de condiciones de proceso anormales o peligrosas. (Villajulca, 2018)

- **Límites de salida y del set point**

En algunos casos resulta poco favorable que el sistema trabaje de 0 a 100%, principalmente porque afecta la vida útil de los actuadores que se trabaje de esa manera, para ello colocar un mínimo de salida al 5% podría resultar favorable para la seguridad de los actuadores. (Villajulca, 2018)

- **Seguridad en parámetros de sintonía**

Los parámetros del control PID son la base del éxito de la eficiencia del control PID, por ello cuando se requiere cambiar los parámetros del sistema, deben hacerlo ingenieros de proceso o técnicos especializados y nunca técnicos instrumentista u operadores. (Villajulca, 2018)

2.9.3. Sobreimpulso

Como lo define (Ingeniería, 2012), “El sobreimpulso es la respuesta de un sistema a un cambio en su entrada cuando excede su objetivo”. Básicamente el sobreimpulso está

ligado a cualquier situación que desequilibre el sistema. Con respecto al control PID se requiere que este efecto cada sea menos abrupto para conseguir una sintonía equilibrada.

Según (Chuck, 2012), el sobreimpulso está descrito como:” la diferencia entre el valor máximo alcanzado y el valor de estado estacionario, la cual se pretende que no sobrepase un cierto porcentaje del valor de estado estacionario.

2.9.4. Tiempo de asentamiento

El tiempo de asentamiento es el tiempo que toma una curva de respuesta para alcanzar un rango esperado, cerca de un valor especificado y que se espera permanezca dentro de dicho valor. (Ogatha, 2010)

El tiempo de asentamiento se relaciona con la mayor constante de tiempo del sistema de control. Los objetivos del diseño del sistema en cuestión determinan cuál criterio de error en porcentaje usar, el tiempo de asentamiento es una función del factor de amortiguamiento, el cual oscila hasta finalmente alcanzar un valor donde encuentre equilibrio. (Docencia, 2011)

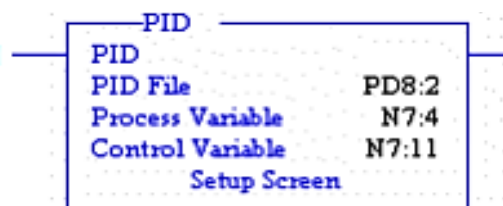
2.9.5. Bloque PID de RSLogix 500 para MicroLogix 1100

El software cuenta con una instrucción para facilitar la implementación de control PID dentro de su lenguaje de programación, es un bloque en el cual se deben llenar los parámetros y variables propios del control PID y su configuración.

La instrucción PID normalmente controla un circuito cerrado utilizando entradas de un módulo de entrada analógica y proporcionando una salida a un módulo de salida analógica como respuesta para mantener efectivamente una variable de proceso en un punto de ajuste deseado. (Automation, MicroLogix Instruction Set Help, 2010)

Figura 15

Bloque PID en RSLogix 500.



Nota. En la imagen se muestra el bloque PID en RSLogix 500. Tomado de (Automation, MicroLogix Instruction Set Help, 2010)

2.9.6. Descripción Bloque PID para MicroLogix 1100

- **PID File**

(MicroLogix 1200, 1400 y 1500 solamente) Especifique un archivo PID. Si aún no ha definido un tipo de archivo PD entre sus archivos de datos, se creará para usted. La longitud del archivo se fija en 23 palabras. El archivo PD reemplaza el antiguo bloque de control de archivos enteros. (Automation, MicroLogix Instruction Set Help, 2010)

- **Control Block Length**

Especifica un archivo entero, por ejemplo, N7: 0. La longitud del archivo se fija en 23 palabras. (Automation, MicroLogix Instruction Set Help, 2010)

- **Process Variable PV**

La dirección del elemento que almacena el valor de entrada del proceso. Esta dirección puede ser la ubicación de la palabra de entrada analógica donde se almacena el valor de la entrada A / D. También puede ingresar una dirección entera si elige pre escalar su valor de entrada al rango 0-16383. (Automation, MicroLogix Instruction Set Help, 2010).

- **Control Variable CV**

Es la dirección del elemento que almacena la salida PID. El valor de salida varía de 0-16383, con 16383 siendo el valor 100% "ON". Esta normalmente es una dirección entera, de modo que puede escalar el rango de salida PID al rango analógico particular que requiere su aplicación. (Automation, MicroLogix Instruction Set Help, 2010).

- **Setup Screen**

Pantalla de configuración que le solicita otros parámetros que debe ingresar para programar completamente la instrucción PID. (Automation, MicroLogix Instruction Set Help, 2010).

2.9.7. Ecuación PID

La ecuación del control PID es la siguiente:

$$c(t) = Kp \cdot e(t) + Ki \cdot \int e(t)dt + Kd \cdot \frac{\partial e(t)}{\partial t}$$

Donde:

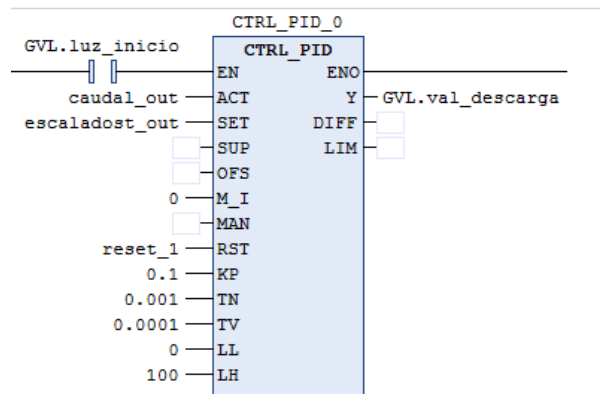
- **c(t)** = Señal de control
- **e(t)** = Señal de error
- **Kp** = Constante de proporcionalidad
- **Ki** = Constante de integración
- **Kd** = Constante de derivación

2.9.8. Bloque PID de CoDeSys

CoDeSys cuenta con una amplia gama de librerías y otras que se le pueden instalar, la librería Oscat fue creada con el fin de proporcionar ayuda a muchos programadores, dándoles un sinfín de herramientas muy útiles para todo proyecto, esta librería también cuenta con bloques de función para control PID, que se encarga de realizar todas las operaciones internas propias del controlador PID, el bloque consta de entradas y salidas las cuales se deben asignar adecuadamente.

Figura 16

Bloque de función PID en CoDeSys



Nota. En esta imagen el bloque de función del control PID y su distribución de entradas y salidas.

2.9.8.1. Parámetros del bloque de función del control PID

Tabla 1

Parámetros del bloque de función del control PID

	Pin	Tipo de dato	Descripción
Entradas	ACT	Real	Valor medido
	SET	Real	Set point
	SUP	Real	Reducción de ruido
	SFO	Real	Ajuste para la salida
	M_I	Real	Valor de entrada para operación manual
	MAN	Bool	Cambiar al modo manual, Manual = Verdadero
	RST	Bool	Entrada de reinicio asincrónica
	KP	Real	Ganancia del controlador
	TN	Real	Reinicio del controlador

	Pin	Tipo de dato	Descripción
Salidas	TV	Real	Derivado del controlador
	LL	Real	Límite de salida inferior
	LH	Real	Límite de salida superior
	Y	Real	Salida del controlador
	DIFF	Real	Desviación
	LIM	Bool	VERDADERO si la salida ha alcanzado un límite

Nota. Es la tabla se muestran los parámetros del bloque de control PID de la librería Ocat, entradas y salidas correspondientes y el tipo de variable de cada una. Tomado de (Oscat, 2013)

2.10. Método de sintonización de parámetros del control PID

La sintonización del control PID busca determinar los valores adecuados para los parámetros K_p , T_i , T_d , con el fin de obtener un comportamiento del sistema equilibrado y eficiente. (Lucas, 2012)

El método del tanteo indica poner el sistema en marcha con parámetros cercanos a cero e ir aumentando gradualmente lo valores hasta encontrar estabilidad. (Lucas, 2012)

Como lo plantea (Indriago, 2019), en su publicación, afirma que hay una serie de pasos para encontrar una sintonización de un lazo PID:

- Colocar todas las constantes en cero.

- Incrementar la constante proporcional hasta obtener una respuesta deseada.
- Ajustar la constante integral para corregir el estado estacionario.
- Ajustar la constante derivativa para reducir el sobre impulso.

2.11. Software Factory I/O

Es un programa de simulación de procesos industriales, dispone de varios escenarios donde puede simular diferentes procesos industriales con tecnologías de automatización actuales más comunes. Factory I/O puede simular dispositivos de control PLC, microcontroladores, FPGA, HMI, transmisores, sensores, de la mayoría de marcas comerciales más conocidas como: Allen Bradley, Siemens, Wind, Samsung, Volvo, Grafcet. (Games, 2012)

Tiene la capacidad de simular sistemas industriales con diferentes protocolos de comunicación como: Modbus y OPC, Ethernet, MQTT, TCP/IP, DA/UA. Como se puede deducir sus prestaciones con varias además de su utilidad. (Games, 2012)

Factory I/O utiliza una tecnología innovadora que permite una creación fácil y rápida de los sistemas industriales en 3D con solo arrastrar y soltar. Cualesquiera de los sistemas construidos se pueden controlar en tiempo real mediante la conexión de Factory I/O y equipos externos como PLC's, microcontroladores, FPGA, etc. (S.A.S., 2017).

Factory I/O al incorporar OPC DA/UA entre sus protocolos de comunicación facilita la comunicación con varios programas, uno de ellos es CoDeSys, que trae consigo un OPC Server de buena calidad que hace posible la comunicación y la compatibilidad entre estos dos programas.

2.11.1. Requisitos de Sistema

Para que el programa funcione de una manera prolija y no exista ningún problema con la computadora, debe cumplir algunos requisitos.

Tabla 2

Requisitos de sistema para Factory I/O.

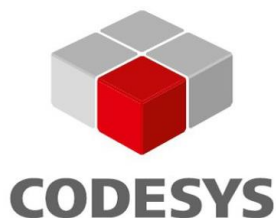
Sistema Operativo	Windows 7 SP1+ o superior
UPC	CPU con soporte de conjunto de instrucciones SSE2
GPU	NVIDIA desde 2006 (GeForce8), AMD desde 2006 (Radeon HD 2000), Intel desde 2012 (HD 4000/IvyBrige)

Nota. En la tabla se muestra los requisitos mínimos de Factory I/O. Tomado de (Games, 2012).

2.12. Software CODESYS

Figura 17

Logo de CoDeSys.



Nota. En la imagen se muestra el logo CoDeSys 3.5V. Tomado de (Electronics, Opiron Electronics, 2019)

Es un entorno de programación de dispositivos de control industrial y tecnologías de automatización industrial, su nombre proviene de las abreviaturas del inglés Controlled Development System que al español significa Sistema de Desarrollo de Controladores. Este software es producido por la marca alemana 3S-Smart Software Solutions GmbH y regulado bajo el estándar internacional IEC 61131-3, una de sus mayores características es su amplia gama de lenguajes en los cuales se puede realizar la programación, posee flexibilidad en los cinco lenguajes de programación más utilizados en el mundo de la automatización, siendo estos: Lista de Instrucciones (IL), Diagrama de Bloque de Funciones (FBD), Diagrama Ladder

(LD), Gráficos de Función Secuencial (CFC) y Texto Estructurado (ST). (Electronics, Opiron Electronics, 2019).

Su compatibilidad de programación con otras marcas cada vez es más extensa, hoy en día es compatible con fabricantes como Bosch, Schneider Electric, Festo, IFM, Factory I/O, entre otros siendo una larga lista que llega hasta más de 500 PLC's en el mercado. Las versiones más conocidas de CoDeSys son la 2.3 y la 3.5 , ambas tienen sus ventajas de desventajas, pero la última es la más recomendable debido a las nuevas implementación que posee. (Electronics, Opiron Electronics, 2019).

CoDeSys tiene consigo un potente simulador, muy útil para poder compilar y comprobar los diferentes programas que se realicen en el mismo antes de cargarlos a los PLC's, una de las razones de la gran acogida de este software libre, también cuenta con la capacidad de incorporar un HMI (Interfaz Hombre-Máquina), lo cual permite desarrollar en el mismo entorno el HMI y a la vez lo convierte en un software más versátil y efectivo. También cabe mencionar que las versiones anteriores de la versión V3.5.12.0, cuentan con un OPC server propio de CoDeSys muy apropiado para comunicaciones con otros softwares y en las versiones después de la indicada debe instalar por separado el OPC Server y este cuenta con una licencia pagada, pero se puede descargar una versión demo que dura alrededor de 30 días, totalmente gratis. (Electronics, Opiron Electronics, 2019).

2.12.1. Requisitos para el software CoDeSys

Tabla 3

Requisitos de sistema para software CoDeSys

Requisitos de sistema para software CoDeSys	
Sistema de programación	Sistema de desarrollo CODESYS V3.5.10.4
Sistema de tiempo de ejecución	Todas las versiones anteriores del tiempo de ejecución de CODESYS
Plataformas/ dispositivos compatibles	Sistema operativo Windows 8/10 (32/64 bits) Las versiones mantenidas por Microsoft son compatibles. Procesador de 2,5 GHz, 8 GB de RAM 4 GB de espacio disponible en el disco duro

Requisitos de sistema para software CoDeSys

Requerimientos adicionales	Microsoft Internet Explorer 11 o superior
Restricciones	Nota importante para los usuarios de CODESYS Static Analysis AddOn: CODESYS V3.5 SP9 Patch 2 (y superior) requiere la versión mínima 4.1.0.0 de CODESYS Static Analysis AddOn. La instalación de versiones anteriores de AddOn causa serios problemas al iniciar CODESYS
Licencia	-
Accesorios requeridos	-

Nota. En esta tabla se muestra los requisitos mínimos para el software CoDeSys. Tomado de (Store, 2017).

2.13. POU'S

Un POU (Programmable Organization Unit) en CoDeSys es un bloque de programación al que se puede asignar una tarea, ya sea una operación, cálculos. Se realiza la programación en una ventana diferente a la de la programación principal, y luego se la manda a llamar desde la programación principal donde aparece en forma de bloque de función. Cada POU puede ser programado por los cinco lenguajes de programación, Lista de Instrucciones (IL), Diagrama de Bloque de Funciones (FBD), Diagrama Ladder (LD), Gráficos de Función Secuencial (CFC) y Texto Estructurado (ST). (Electronics, Opiron Electronics, 2019)

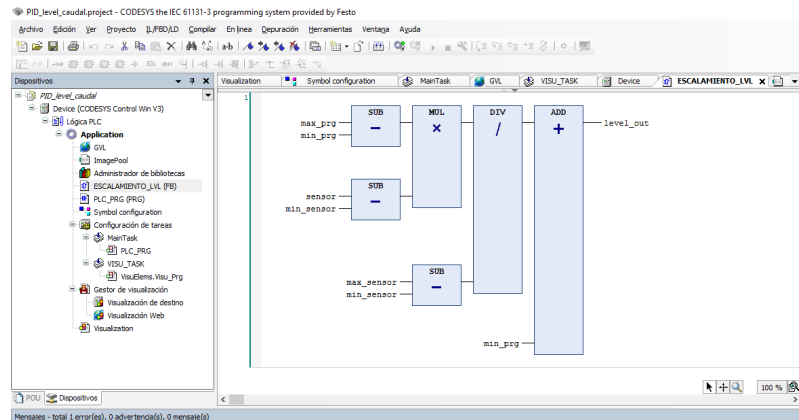
Existe tres tipos de POU's dentro de CoDeSys, y son:

- **Funciones.** - Son aquellos a los que se le asigna programas sencillos, especialmente que no ocupan memoria. Se le puede asignar tareas como multiplicaciones o acciones repetitivas. (Electronics, Opiron Electronics, 2019)
- **Bloques de función.** - Pueden almacenar datos, sus entradas y salidas están bien asignadas. Su salida puede llegar a depender de su entrada, apropiados para el control de válvulas, motores, o variadores. (Electronics, Opiron Electronics, 2019)

- **Programas.** – Es el tipo de función más complejo, puede organizar la automatización que se esté realizando y puede acceder a los bloques de función y funciones. Puede realizar una tarea de automatización directa. (Electronics, Opiron Electronics, 2019)

Figura 18

POU de CoDeSys



Nota. En esta imagen se muestra un POU elaborado en CoDeSys.

2.14. Variable Global

Las variables globales son variables ordinarias, constantes, variables externas o remanentes que se reconocen en todo el proyecto, cualquier POU puede leer y escribir en las variables globales, además tienes la capacidad para acoplarse al OPC para ser reconocidas en otros programas y mejorar el envío y recepción de datos. Se asignan al principio de proyecto, y tiene una sección propia. (Smith, 2017)

2.15. Variable local

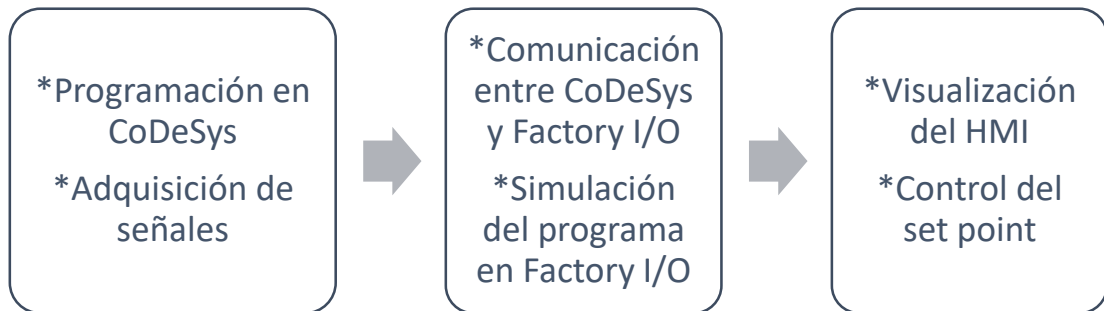
Son las variables que se asignan para ejecutarse dentro de un programa solamente o ya sea dentro de un POU, su disponibilidad se ve limitada y no se puede hacer referencia fuera de su sección, solo se la puede manipular la sección que se ha declarado. Las variables locales son importantes para la programación por la asignación de funciones y no ocupan memoria. (Smith, 2017)

Capítulo III

3. Desarrollo del tema

Figura 19

Diagrama de bloques del proyecto



Nota. En este gráfico se muestra el diagrama de bloques del proyecto.

Si bien en el tema principal de este proyecto de grado es: Implementación de un control PID utilizando una estación de caudal mediante el PLC MicroLogix 1100 para prácticas de control de procesos” y en el alcance se indica los programas a utilizar y las marcas y modelos, debido a la pandemia mundial del año 2020 del virus covid 19, dificultó la entrada a instituciones educativas y sus respectivos laboratorios, razón por la cual este proyecto se hizo completamente simulado y se utilizaron otros programas a los mencionados en el anteproyecto, la programación se realizó en el software CoDeSys y no en el RsLogix 500, por razones de compatibilidad con el simulador (Factory IO) y debido a que CoDeSys es una marca neutral y está regulado bajo el estándar internacional IEC 61131-3, además de estar programado con el mismo lenguaje indicado previamente, indica que no hay gran diferencia o alteración a los resultados concluyentes.

El proyecto básicamente empieza por recoger las señales de los sensores de flujo provenientes de Factory I/O, mediante la comunicación OPC Server, se comunica con el CoDeSys y este posteriormente realiza los cálculos y operaciones para procesar las señales de entrada y poder enviar una señal de salida con datos que le permitan a los actuadores de Factory I/O trabajar de la manera requerida.

3.1. Investigación del proyecto

El control PID surge como una necesidad de controlar los sistemas de manera automática, que manejan porcentajes de capacidad como con los de temperatura, velocidad, caudal, nivel, luz, entre otras magnitudes controladas, teniendo esto en cuenta, resulta interesante el saber cómo, a través de los años, (porque el control PID nació hace muchos años de la mano de Elmer Sperry en 1911), se ha ido perfeccionando con la ayuda de varios ingenieros a través de la historia hasta llegar a ser un control muy utilizado para control industrial y para la automatización.

Hace no mucho es posible hacer el control PID en diferentes PLC's de diferentes marcas para controlar diferentes magnitudes, lo que se convirtió en un tema importante en el aprendizaje dentro del mundo de la automatización y su uso también se volvió más común, de esa manera es importante el manejo e implementación de este control automático, ya que el realizar correctamente un control PID no solo expande los conocimientos, sino que también amplía oportunidades en el campo laboral el cual cambia con los años.

La búsqueda de softwares dentro desde proyecto fue dadas en primer lugar por las marcas comerciales que están manejando siendo esas RsLogix 500 para la programación del PLC, ya que cada marca comercial en el mundo de la automatización maneja su propio software programador, aunque los lenguajes no cambien de manera significativa.

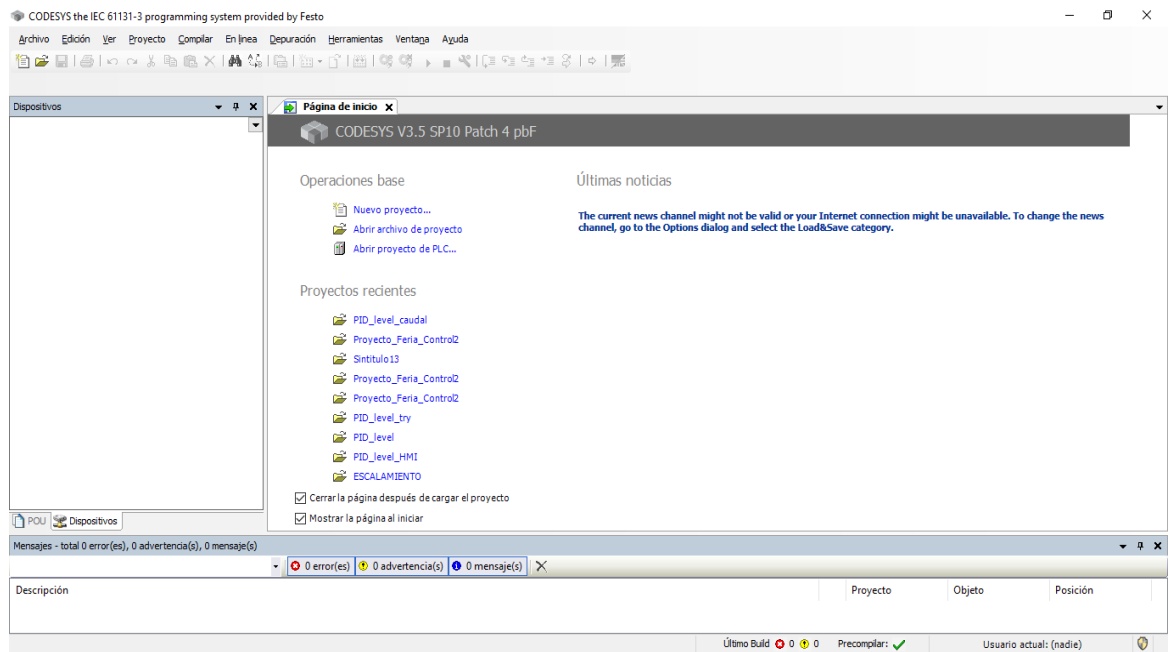
3.2. Programación en el Software CoDeSys

Para iniciar este proyecto se tomó en cuenta el software CoDeSys versión 3.5.10.4, debido a que sus prestaciones con muchas, tiene todo incluido, un PLC virtual, capacidad de realizar HMI, OPC propio de CoDeSys que permite comunicarse con otros softwares para enviar y recibir datos, y también porque es compatible con el simulador que se va a utilizar para la realización de este proyecto que es el Factory I/O.

Entonces ya contando con el software CoDeSys V3.5.10.4 instalado en la computadora, lo ejecutamos e iniciamos un nuevo proyecto.

Figura 20

Entorno de trabajo de CoDeSys.



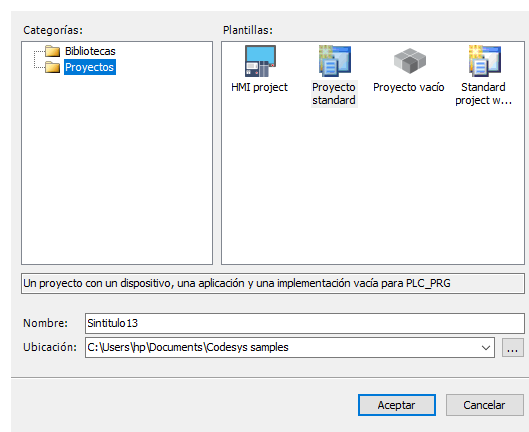
Nota. En esta imagen se muestra la pantalla de inicio al ejecutar CoDeSys.

3.2.1. Creación de un nuevo proyecto en CoDeSys

Con el programa ejecutándose, seleccionamos crear un nuevo proyecto, seleccionamos un proyecto estándar.

Figura 21

Creación de un nuevo proyecto en CoDeSys.



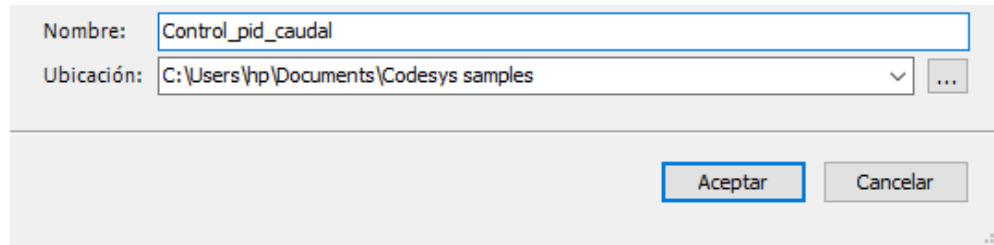
Nota. En esta imagen se muestra la manera de crear un nuevo proyecto.

3.2.2. Asignación de un nombre y ubicación del nuevo proyecto

Colocamos el nombre que deseamos para reconocer el proyecto al momento de abrir y también la ubicación le asignamos la que resulte más cómoda, y le damos aceptar.

Figura 22

Nombre del proyecto.



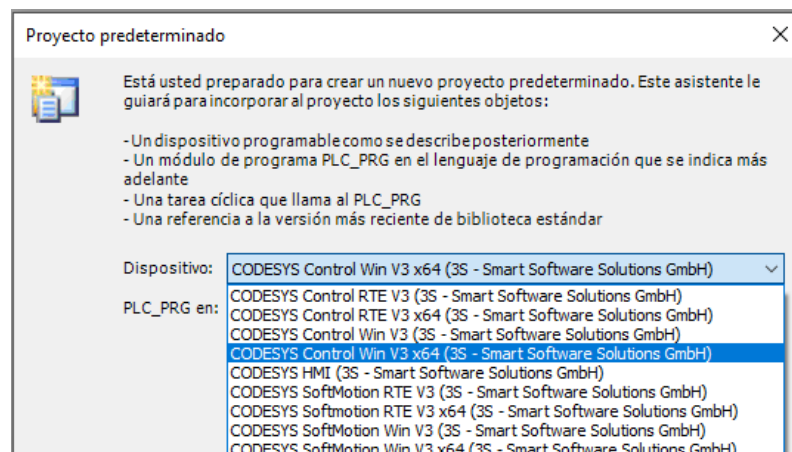
Nota. En esta imagen se muestra la asignación de un nombre al nuevo proyecto.

3.2.3. Selección el dispositivo

Aquí en este paso se debe seleccionar el tipo de dispositivo para trabajar, esto depende de la computadora y su número de bits, en este caso la computadora que se utilizó es de 64 bits y por eso se marcó la casilla señalada en la Figura 23.

Figura 23

Elección de dispositivo.



Nota. En esta imagen se muestra la elección del dispositivo para trabajar.

3.2.4. Elección de tipos de lenguaje de programación

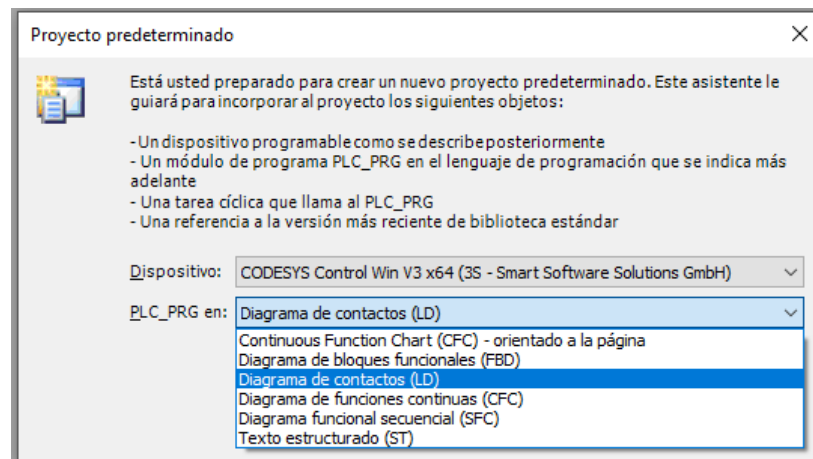
En este paso es donde se elige en que lenguaje de programación se va a realizar el proyecto, CoDeSys nos ofrece cinco alternativas para ello: Lista de Instrucciones (IL),

Diagrama de Bloque de Funciones (FBD), Diagrama Ladder (LD), Gráficos de Función Secuencial (CFC) y Texto Estructurado (ST).

Para este proyecto la opción que se escogió es lenguaje de diagrama ladder (LD).

Figura 24

Elección de lenguaje de programación.

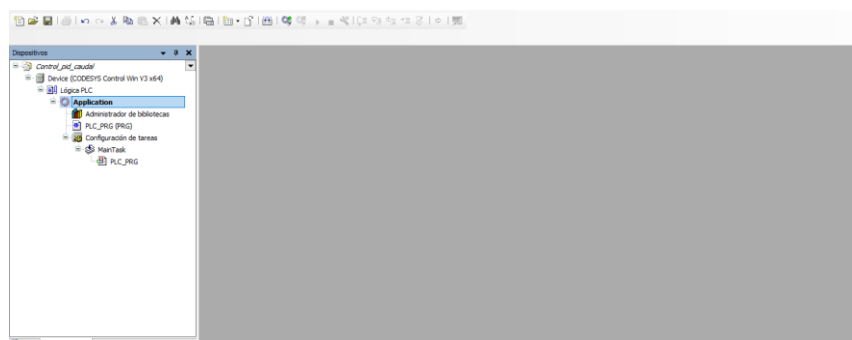


Nota. En esta imagen se muestra la elección de lenguaje de programación.

Una vez escogido el tipo de dispositivo y el tipo de lenguaje de programación, se debe dar clic en aceptar y posteriormente carga una pantalla similar a la que está en la Figura 25.

Figura 25

Entorno de trabajo de CoDeSys.



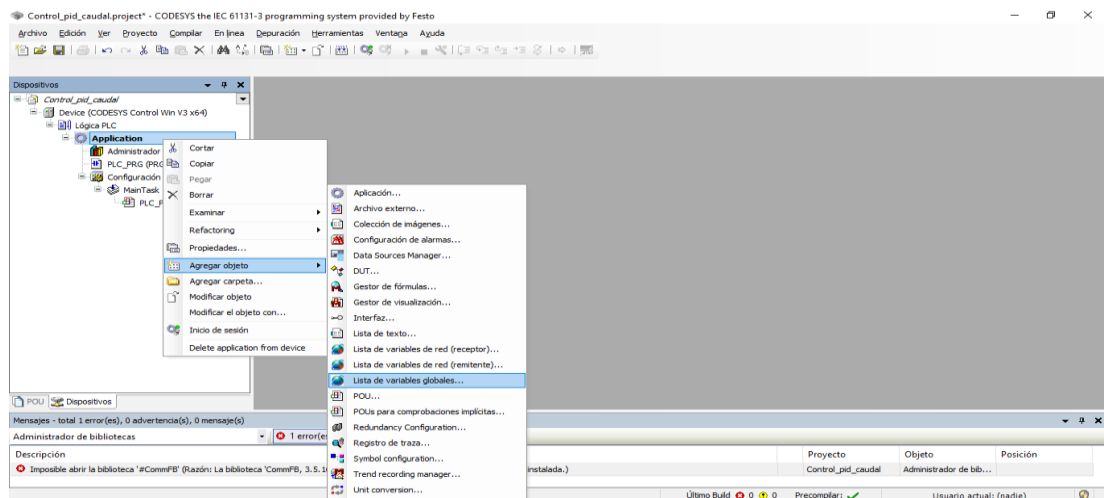
Nota. En esta imagen se muestra el entorno de trabajo de CoDeSys.

3.2.5. Asignación de variables globales

Para asignar variables globales en CoDeSys, se debe hacer clic derecho en: "Aplicación" (al lado izquierdo de la ventana principal, en la ventana de Dispositivos), en luego saldrá una lista de opciones donde se busca y selecciona la pestaña: "Agregar objeto", donde aparecerá otra lista de opciones, finalmente se busca: "Lista de variables globales", y se da clic.

Figura 26

Agregar lista de variables globales en CoDeSys.

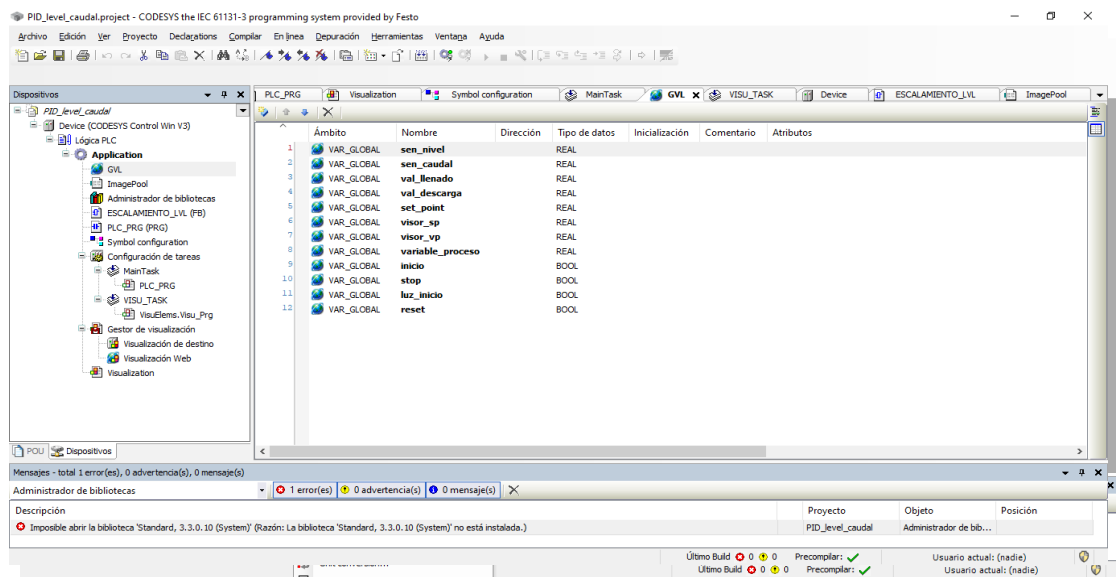


Nota. En esta imagen se muestra como agregar variables globales.

Para empezar la programación se debe tener en cuenta que algunos variables deben ser de tipo globales, están serán las que se posicionarán como entradas o salidas, dependiendo del caso, en el PLC virtual y al OPC y se podrán observar en la asignación de pines en el programa Factory I/O.

Figura 27

Lista de variables globales.



Nota. En esta imagen se muestra la elaboración de la lista de variables globales.

Algunas variables globales son de tipo “real” debido a que en esas variables se manejarán valores analógicos y colocarlas en tipo real es lo más apropiado para que puedan ser leídas, otras variables globales están asignadas dato tipo “bool”, aquellas serán destinadas a pulsadores donde solo reconocen 1 y 0.

3.2.6. Creación de un POU

Un POU en CoDeSys es un bloque de programación al que se puede asignar una tarea, y para programarlo primero hay que asignar el tipo de POU y el tipo de lenguaje de programación con el que se va a trabajar. Para ello se debe hacer clic en: Application (que está en el lado izquierdo de la ventana principal, en la ventana de Dispositivos), donde aparecerá una lista de opciones, se busca y selecciona: “Agregar objeto” donde aparecerá otra lista de opciones, se busca y selecciona: “POU”

Figura 28

Creación de un POU en CoDeSys.

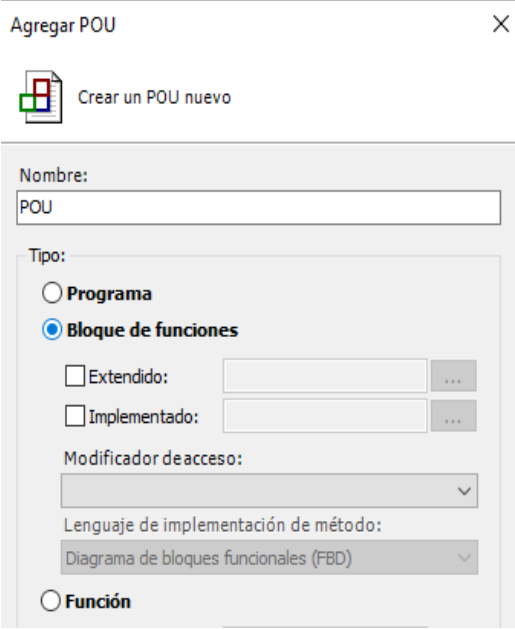
Nota. En esta imagen se muestra el procedimiento para crear un POU.

Seleccionado la creación del POU, aparece una ventana donde se debe seleccionar el tipo de POU que se requiere implementar, en este paso, para este proyecto se requiere de un POU tipo bloque de función.

También es posible, (si se prefiere) asignarle un nombre distintivo al POU, el nombre por defecto es "POU".

Figura 29

Asignación de tipo de POU



The image shows a dialog box titled "Agregar POU" with a close button (X) in the top right corner. Below the title bar, there is a button with a document icon and the text "Crear un POU nuevo". The main area of the dialog is a form with the following fields and options:

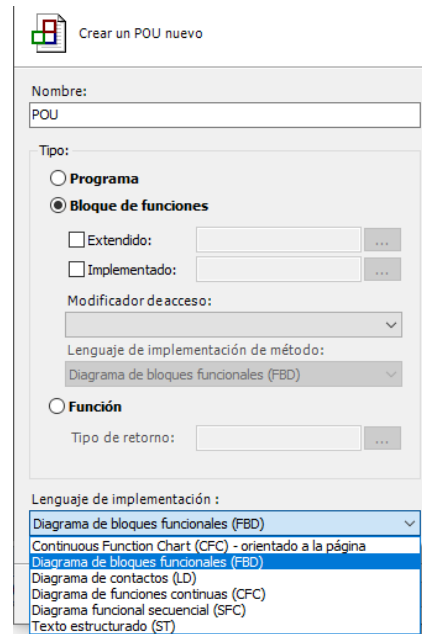
- Nombre:** A text input field containing the text "POU".
- Tipo:** A section containing several options:
 - Programa
 - Bloque de funciones**
 - Extendido: [text input] [...]
 - Implementado: [text input] [...]
 - Modificador de acceso: [dropdown menu]
 - Lenguaje de implementación de método: [dropdown menu] (selected: Diagrama de bloques funcionales (FBD))
 - Función**

Nota. En esta imagen se muestra la asignación de tipo de POU.

Luego se debe asignar, en la misma ventana, el tipo de lenguaje de programación que se va a utilizar dentro del POU, se puede elegir entre los cinco principales lenguajes de programación dentro de la automatización, según la regulación IEC 61131-3. Para este proyecto el lenguaje de programación dentro del POU, es el de Diagrama de bloques funcionales, debido a las operaciones matemáticas a realizar, este lenguaje facilita dichos cálculos.

Figura 30

Asignación de lenguaje de programación del POU.



Nota. En esta imagen se muestra la selección de lenguaje de programación del POU.

3.2.7. Operaciones para escalamiento de señales analógicas de entrada dentro del POU

La función del POU dentro de la programación principal es la de hacer los cálculos y operaciones necesarios para escalar las señales analógicas de entrada del sensor de caudal y del set point, las operaciones están basadas en la ecuación de la recta, tomando en cuenta también parámetros importantes para obtener una señal de salida que este dentro de un rango de 0 y 100.

3.2.8. Ecuación de la recta para escalamiento de señales

$$\text{Señal out} = \frac{\text{max prog} - \text{min prog}}{\text{max sensor} - \text{min sensor}} \times (\text{sensor in} - \text{min sensor}) + \text{min prog}$$

Donde:

- **Señal out.** - señal escalada.
- **Max prog.** - Máximo valor que lee la programación.

- **Min prog.** - Mínimo valor que lee la programación.
- **Max sensor.** - Máximo valor que puede leer el sensor.
- **Min sensor.** - Mínimo valor que puede leer el sensor.
- **Sensor in.** - valor medido en el sensor en ese instante.

El valor para cada una de las variables depende del alcance de tipo de sensor y de la lectura que entrega, ya que existen sensores que pueden dar lectura en voltios (0-10V) o en miliamperios (4-20mA), y también de los bits de cada PLC, ya que en cada modelo de PLC cambia el número de bits, y también hay que tomar en cuenta la relación entre a lectura y la magnitud medida.

Para este proyecto se utilizó los sensores virtuales del software Factory I/O los cuales tienen un rango de 0 a 10 Voltios y la magnitud que es el caudal será medida de 0 a 100 % que representa de 0 - 30 lt/min, el PLC virtual de CoDeSys posee 15 bits, siendo el valor máximo que puede leer el **32767** y el mínimo 0.

Dada la ecuación de la recta adecuada para el escalamiento de señales analógicas de entrada, se debe hacer toda esa ecuación en la programación del POU, con el fin de obtener una señal escalada de 0 a 100.

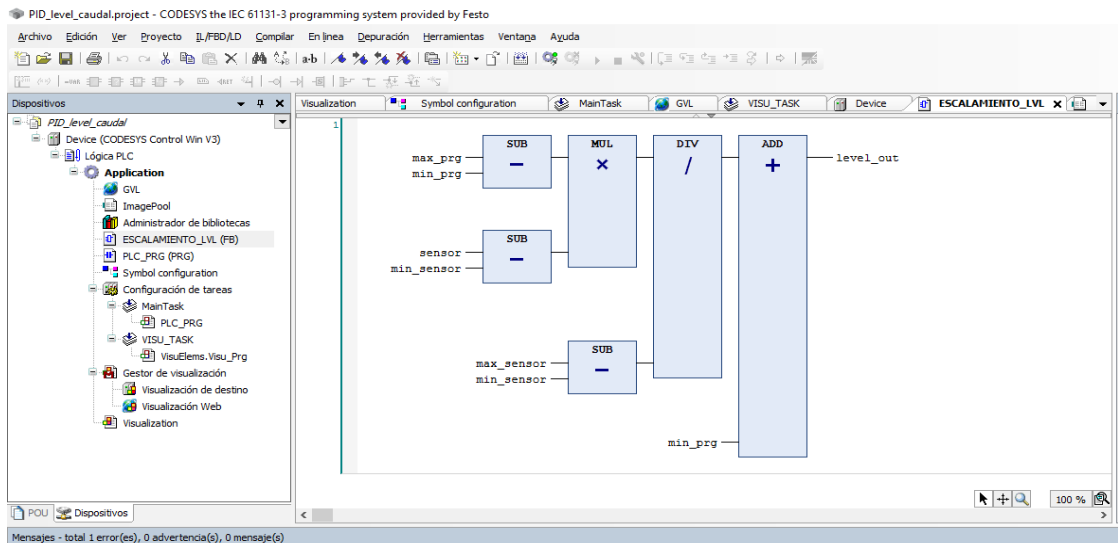
Este escalamiento se debe hacer en dos ocasiones tanto para la lectura del valor del sensor de caudal y para la lectura del valor del set point que básicamente es un potenciómetro. Interesa que estas dos entradas sean escaladas para ser procesadas y lograr hacer el control PID.

En la Figura 30, se puede observar las operaciones necesarias basadas en la ecuación de la recta en un POU denominado "ESCALAMIENTO_LVL" que luego se mandara a llamar desde el programa principal.

Las variables pueden generarse mediante la asignación de nombres las entradas de cada una de las operaciones o de manera directa en el panel de variables, que se encuentra encima del panel de programación, las variables generadas del POU "ESCALAMIENTO_LVL" se pueden observar en la Figura 31, ahí también se distribuyen tomando en cuenta su destino, si serán entradas o salidas del bloque de función en construcción

Figura 32

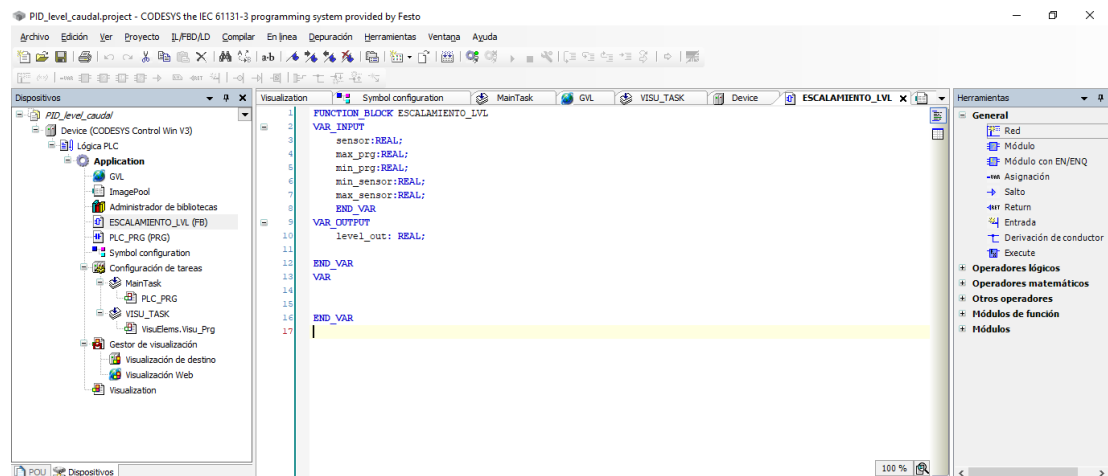
POU para el escalamiento de señales.



Nota. En esta imagen se muestra la ecuación de la recta en el POU.

Figura 31

Variables generadas en el POU



Nota. En esta imagen se muestra las variables que se utilizaron para elaborar el POU.

3.2.9. Programación principal

El lenguaje utilizado para la programación principal es el lenguaje Ladder, no es el mismo lenguaje que el escogido para la programación en el POU.

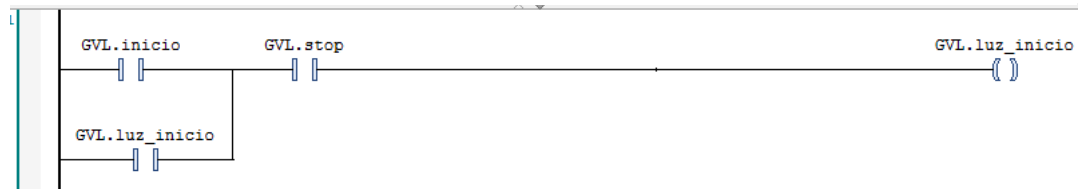
Se comenzó realizando un enclavamiento y que se encienda una luz indicadora, para saber que el sistema esta encendido, utilizando dos contactores normalmente abiertos, tanto para el inicio como para la parada, y una marca que memorice el sistema.

3.2.10. Enclavamiento Marcha/Paro

Un enclavamiento básico de marcha y paro que será útil para encender el sistema y que el sistema funcione cuando se presione el botón “inicio” desde Factory I/O y se desactive al momento de presionar el botón “stop”.

Figura 33

Enclavamiento marcha/paro



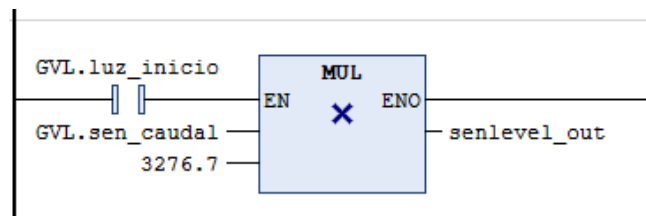
Nota. En esta imagen se muestra la elaboración del enclavamiento Marcha/Paro.

Una vez hecho el enclavamiento prosigue la lectura de las señales analógicas, primero se comenzó con la señal proveniente de sensor de flujo de Factory I/O.

3.2.11. Lectura de la señal de sensor de flujo y escalamiento

Figura 34

Lectura de señal del sensor de flujo.



Nota. En esta imagen se muestra la lectura de señales del sensor de flujo.

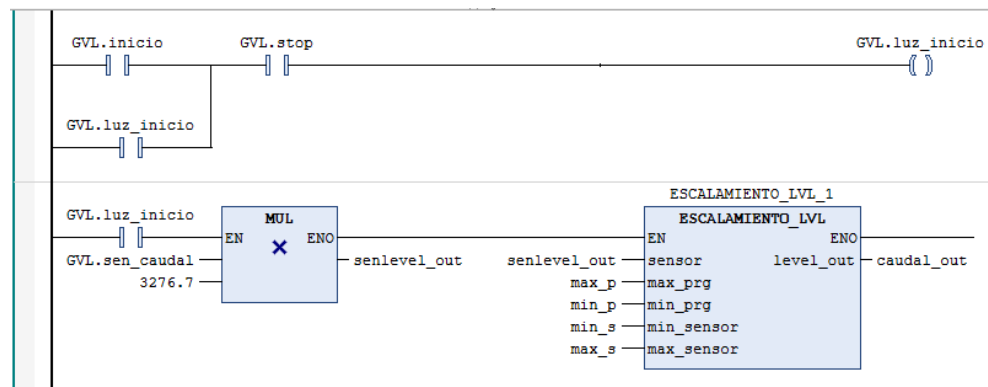
A la señal leída se le multiplica por el valor máximo que puede leer el PLC virtual que es **32767** y se divide para 10, es parte del proceso de escalamiento de la señal para que pueda dar un valor legible de 0 a 100.

Posterior se debe hacer el llamamiento de un bloque de función, del POU que se creó previamente y se llena todos los parámetros, con los valores que ya se conocen, para la salida del POU se crea una variable de tipo real.

Hecho eso la señal proveniente del sensor de flujo del software Factory I/O será una señal que fluctúe de 0 a 100

Figura 35

Llamamiento del POU para el escalamiento de la señal del sensor de flujo

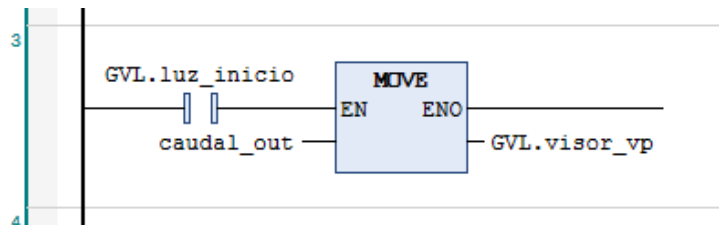


Nota. En esta imagen se muestra el llamamiento del POU para el escalamiento de señales del sensor de flujo.

Luego se colocó un MOVE para que la señal escalada se guarde en una de las variables globales, en la variable GVL.visor_vp, esto para que el valor se pueda observar en uno de los displays simulados que está en la escena del software Factory I/O.

Figura 36

Guardado de señal escalada del sensor de flujo en una variable global.

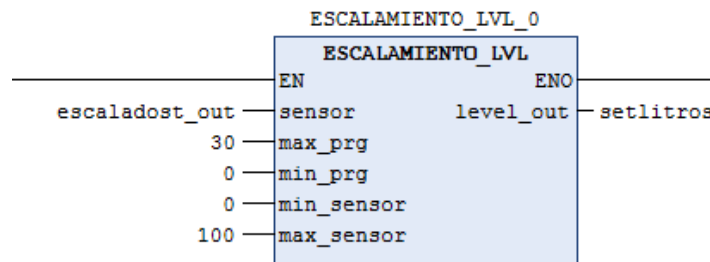


Nota. En esta imagen se muestra el guardado de la señal escalado en otra variable para ser visualizada en Factory I/O.

A continuación, se coloca un módulo para llamar al POU programado previamente, esto con el fin de poder visualizar los lt/min que se deben estar entregando de acuerdo a la posición actual del set point, siendo 30 lt/min el máximo nivel de litros\minuto que puede controlar la válvula de descarga, este valor se verá en el HMI.

Figura 37

Escalamiento para datos en litros\minuto.



Nota. En esta imagen se muestra el escalamiento para visualizar valores en litro/minuto.

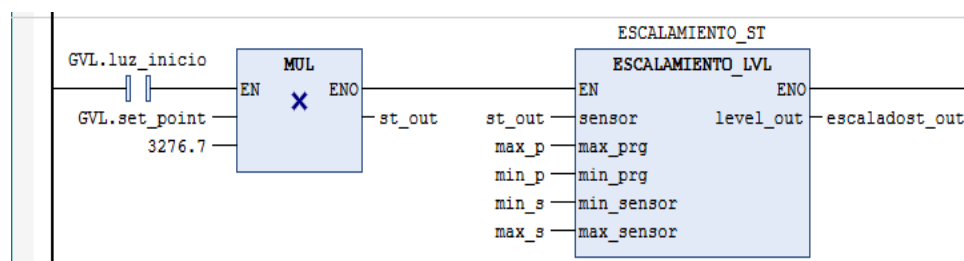
3.2.12. Lectura de la señal de set point y escalamiento

Se debe seguir los mismos pasos que en la lectura de la señal proveniente del sensor de flujo, a la señal leída se le multiplica por el valor máximo que puede leer el PLC virtual que es **32767** y se divide para 10, es parte del proceso de escalamiento de la señal para que pueda dar un valor legible de 0 a 100.

Así que primero se leyó los valores provenientes del set point, en la misma línea se mandó a llamar al bloque de función realizado en el POU, se llenó las entradas del bloque de función con los datos correspondientes y en la salida se asignó una variable de tipo real para guardar la señal escalada.

Figura 38

Llamamiento del POU para el escalamiento de la señal del set point.

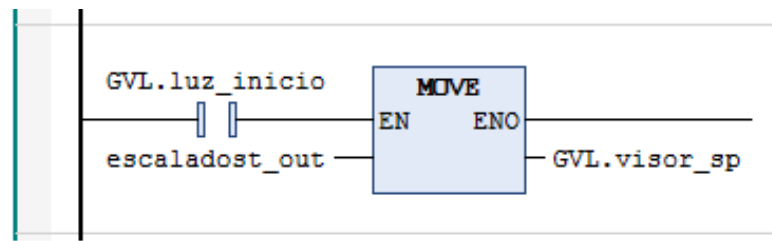


Nota. En esta imagen se muestra el llamamiento del POU para escalar la señal del set point.

Una vez escalada la señal del set point, se colocó un MOVE para que la señal escalada se guarde en una de las variables globales, en la variable GVL.visor_sp, esto para que el valor se pueda observar en uno de los displays simulados que está en la escena del software Factory I/O.

Figura 39

Guardado de señal escalada del set point en una variable global.



Nota. En esta imagen se muestra el guardado de la señal escalada en una variable global para ser visualizada en Factory I/O.

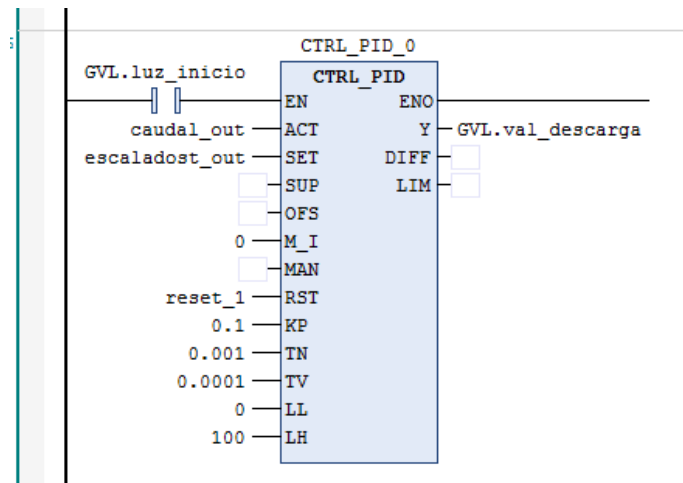
3.2.13. Colocación del bloque de función para el control PID

Para el control PID se utilizó un bloque de función perteneciente a la librería Oscat, una librería que posee muchas herramientas que facilitan la programación en CoDeSys, algunas de esas herramientas no están en CoDeSys, la librería Oscat se descargó de la página oficial de FESTO, es de acceso gratuito y solo hay que instalarla en las librerías de CoDeSys.

Ya teniendo la señal de sensor de flujo y la señal de set point escaladas, se colocó el bloque de función de control PID, se completó los parámetros correspondientes, para los parámetros propios de control PID, la constante proporcional, la constante integral y la constante derivativa, se recurrió a encontrar esas constantes utilizando el método del tanteo, es decir, cambiando los valores de los parámetros gradualmente del control PID hasta encontrar una sintonización donde exista una estabilidad aceptable.

Figura 40

Control PID de la librería Oscat.

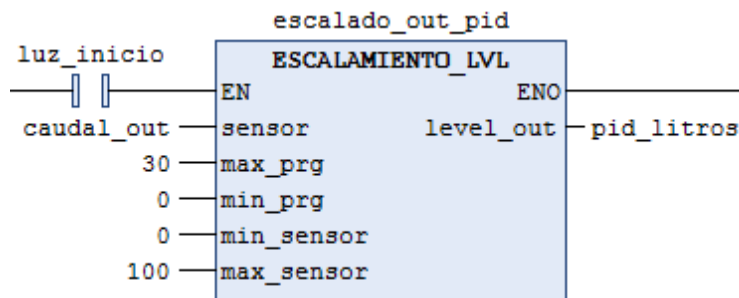


Nota. En esta imagen se muestra la colocación del bloque de función del control PID.

A continuación, se coloca un módulo para mandar a llamar al POU programado previamente, esto con el fin de poder visualizar los lt/min que se están entregando ese momento, de acuerdo a la lectura del sensor de flujo, el valor máximo que puede llegar a tener son 30 lt/min, valor que debe ser similar al dato que corresponde al del set point, este valor se verá en el HMI.

Figura 41

Escalamiento para obtener litros/minuto del sensor de flujo.



Nota. En esta imagen se muestra es escalamiento de la señal de salida del control PID para visualizar valores en lt/min.

Con respecto al controlador PID, de esa manera los valores con los que se pudo llegar a obtener una estabilidad aceptable para el control PID fueron: constante proporcional (K_p) = 0,1, constante integral (T_i) = 0,001, constante derivativa (T_d) = 0,0001.

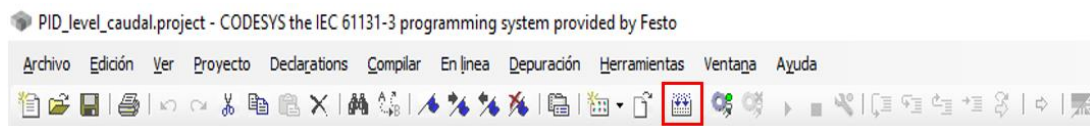
3.2.14. Compilación e inicio de sesión del programa en CoDeSys

Ya una vez sintonizado adecuadamente el control PID, se debe compilar el programa para verificar que no haya errores ni conflictos en la programación y poder proseguir con la vinculación con Factory I/O.

Para compilar se debe buscar en el panel superior de herramientas el ícono señalado en la Figura 41. Se debe hacer clic y revisar los mensajes en la parte inferior de la pantalla, donde indica si el programa tiene errores o advertencias.

Figura 42

Botón de compilar de CoDeSys.



Nota. En esta imagen se muestra señalado el botón de compilar.

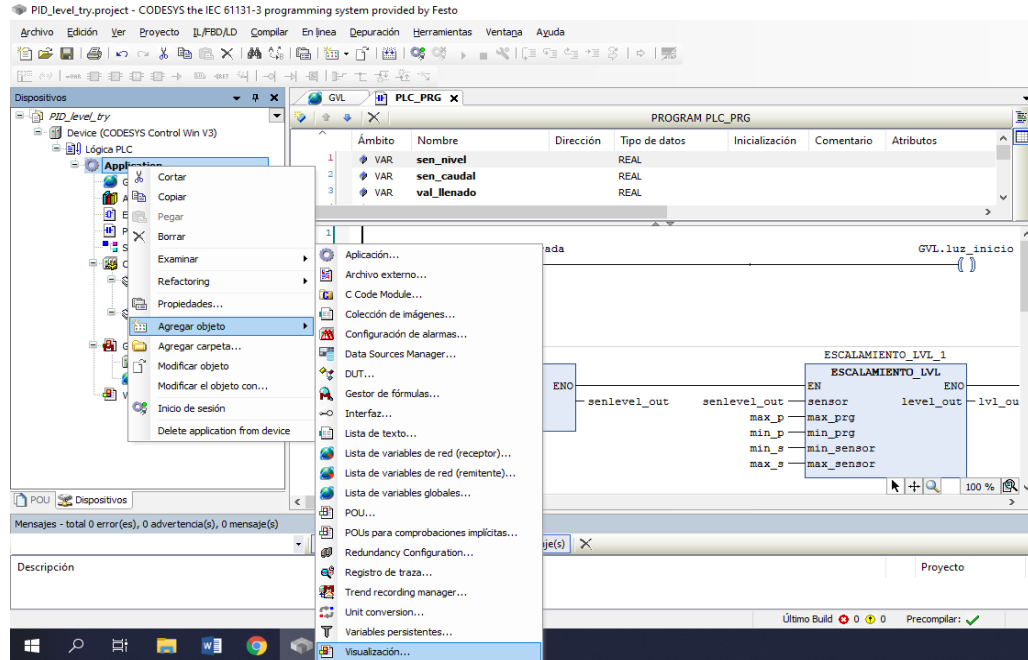
3.2.15. Creación de un HMI en CoDeSys

CoDeSys también cuenta con la capacidad de hacer HMI dentro de sus funciones, la creación de un HMI se creó con la finalidad de visualizar los datos más relevantes del proyecto como son, el set point y la variable de procesos.

Para realizar el HMI desde CoDeSys se debe ir a “Application” y se da clic derecho, entonces aparece una lista de opciones, se busca “Agregar Objeto”, aparece otra lista de opciones en la cual se busca “Visualización” y se selecciona.

Figura 43

Crear un HMI en CoDeSys.

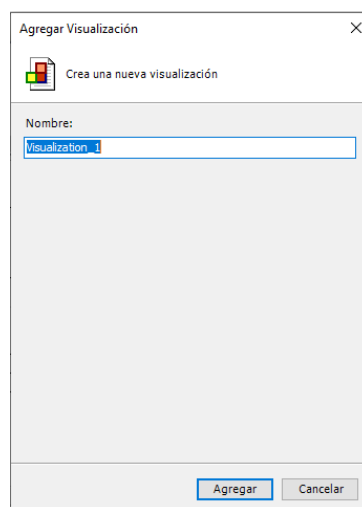


Nota. En esta imagen se muestra la selección para realizar un HMI.

Aparece una ventana, con el propósito de asignarle un nombre al HMI para distinguirlo de las otras ventanas, se le colocó por defecto “Visualización”, se le da clic en agregar para proseguir.

Figura 44

Asignación de nombre al HMI.



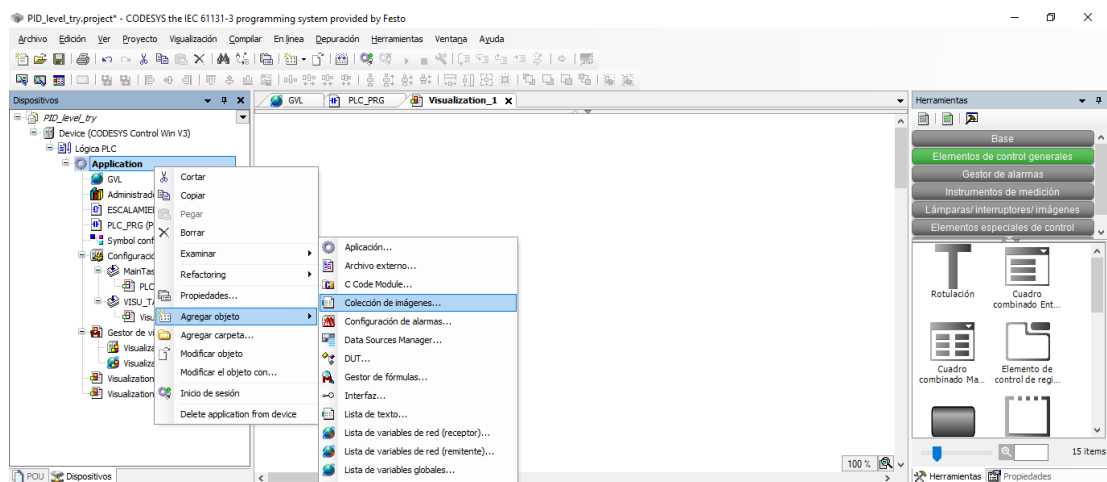
Nota. En esta imagen se muestra la asignación de nombre a la visualización del HMI.

Se crea la ventana de “Visualización”, está en blanco, se comenzó agregándole una imagen con el logotipo de la Universidad De La Fuerzas Armadas- ESPE, para ello primero se debe agregar la imagen en la base de datos de CoDeSys.

3.2.16. Adición de una imagen externa para el HMI

Figura 45

Agregar imágenes en CoDeSys



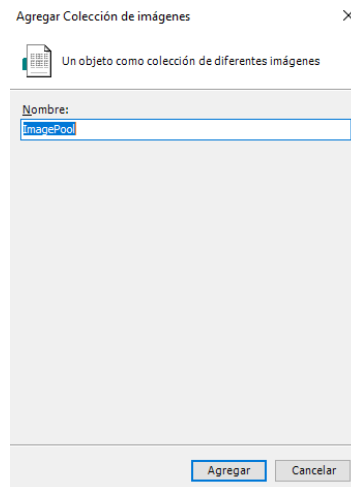
Nota. En esta imagen se muestra la opción para agregar imágenes en el HMI.

En la Figura 44 se muestra los pasos iniciales para agregar una imagen en el HMI, clic derecho en “Applications”, aparece una lista de opciones, se busca “Agregar objeto”, donde sale otra lista de opciones y finalmente se selecciona “Colección de imágenes”.

Una ventana aparece, sirve para asignar un nombre a la imagen, se le colocó por defecto el nombre inicial.

Figura 46

Asignación de un nombre a la imagen a agregar.

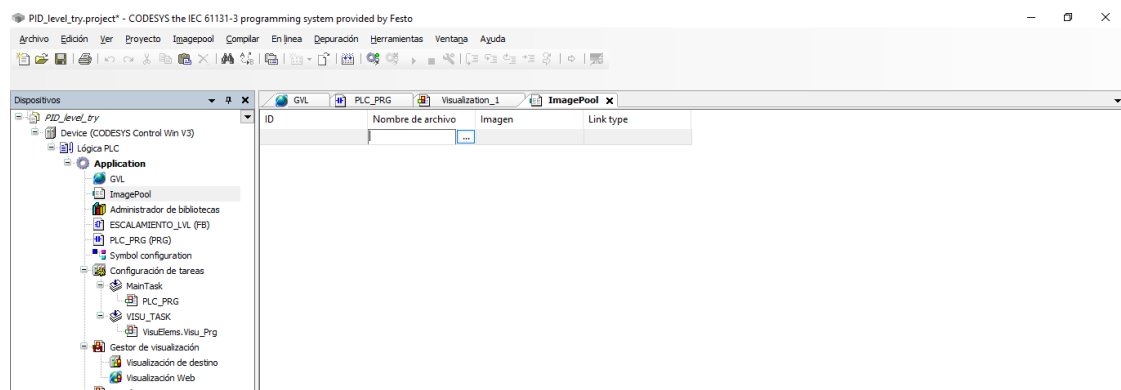


Nota. En esta imagen se muestra la asignación de nombre a la ventana de colección de imágenes.

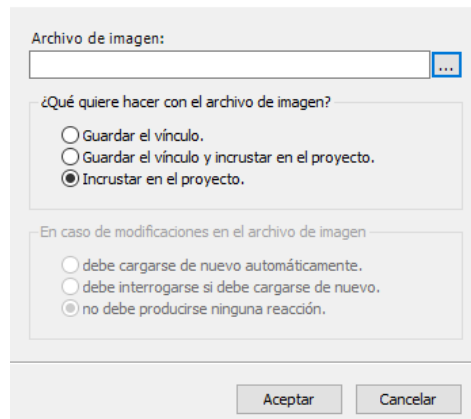
Se abre una ventana en blanco, para agregar la imagen se hace doble clic bajo cuadro llamado “Nombre del archivo”, aparecen tres puntos, se hace clic sobre esos tres puntos

Figura 47

Pantalla inicial para agregar imágenes.

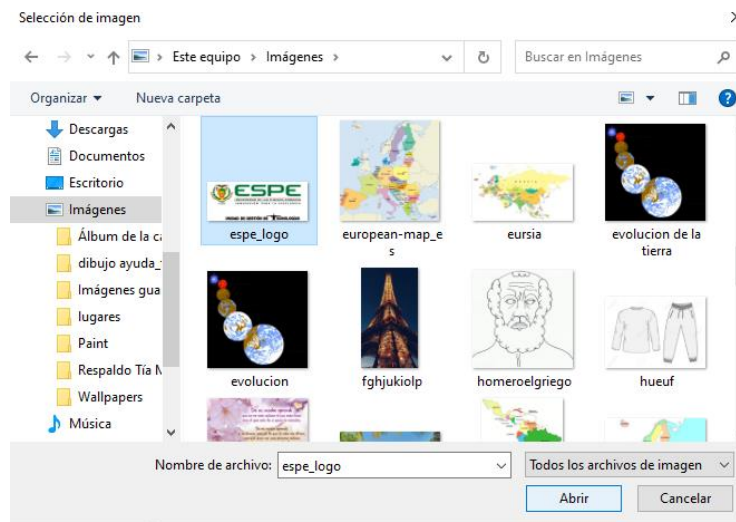


Nota. En esta imagen se muestra la pantalla inicial donde se puede agregar imágenes externas.

Figura 48*Selección de imagen*

Nota. En esta imagen se muestra la ventana donde se debe seleccionar para buscar las imágenes.

Aparece nuevamente una ventana, debajo de “Archivo de imagen” están tres puntos, se hace clic en los tres puntos, y se abrirá un buscador de los archivos personales del computador, se busca la dirección de la imagen y se la selecciona.

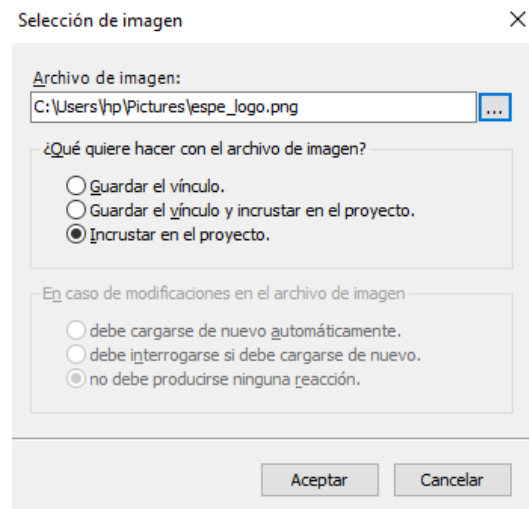
Figura 49*Selección de imagen desde el computador.*

Nota. En esta imagen se muestra la imagen selecciona que va estar en el HMI.

Seleccionada la imagen, la ventana de la Figura 49 aparece llena la casilla debajo de “Archivo de imagen”, con la dirección de la imagen seleccionada, se le da clic en “Aceptar”.

Figura 50

Imagen seleccionada en CoDeSys.

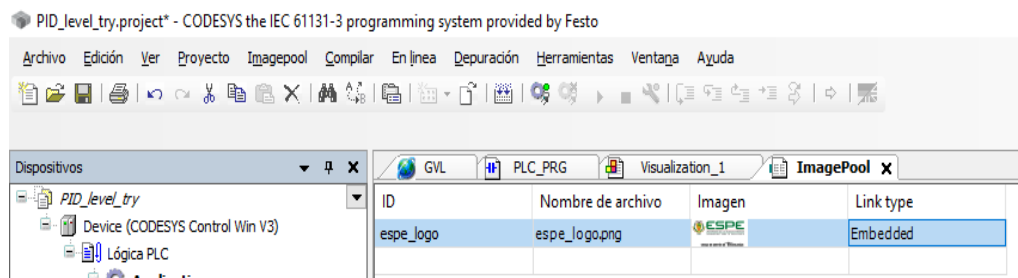


Nota. En esta imagen se muestra a imagen cuando ya está cargada.

Para verificar que la imagen está cargada en la base de datos de CoDeSys, la imagen aparece en la tabla de la ventana de imágenes, con todos los datos correspondientes.

Figura 51

Imagen cargada con éxito.



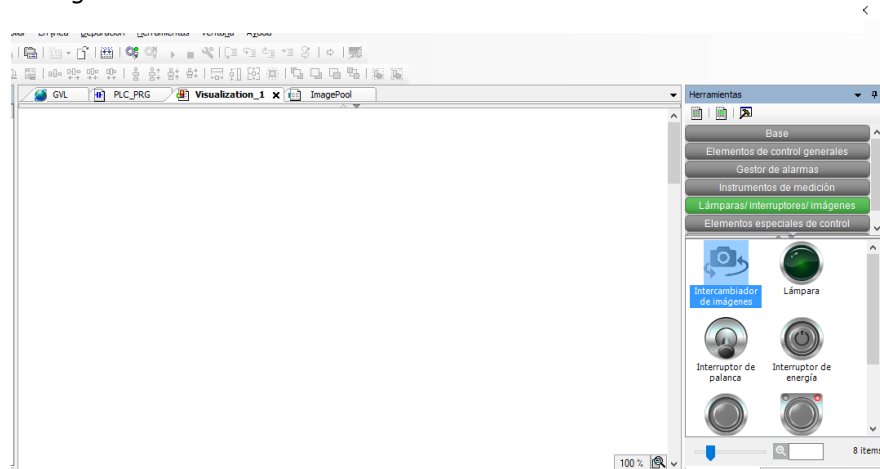
Nota. En esta imagen se muestra la verificación de la imagen cargada.

3.2.17. Vinculación de una imagen externa en el HMI de CoDeSys

En la ventana del HMI, se debe dirigir al panel de herramientas, en el lado derecho de la pantalla, en la lista de herramientas, se debe hacer clic en “Lámparas/interruptores/imágenes” debajo sale el icono de nombre “Intercambiador de imágenes” se arrastra el icono hacia el panel de trabajo

Figura 52

Vinculación de imágenes al HMI.

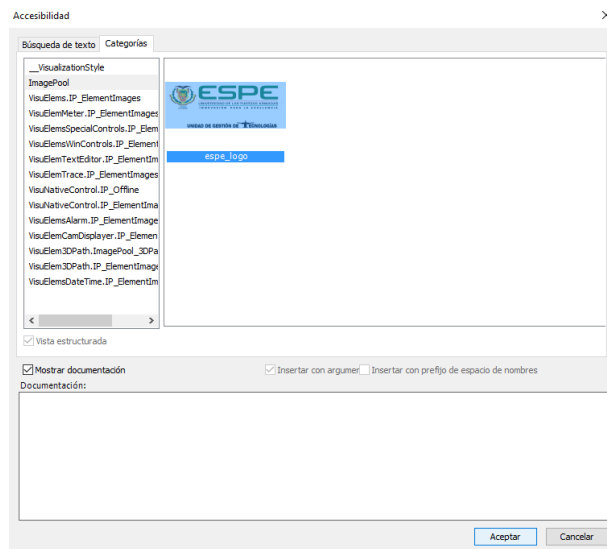


Nota. En esta imagen se muestra el espacio de trabajo del HMI.

Una ventana nueva aparece inmediatamente, tiene un buscador y diversas categorías de imágenes, algunas son propias de CoDeSys, se busca la imagen requerida y se da clic en “aceptar”.

Figura 53

Selección de imagen desde el HMI.



Nota. En esta imagen se muestra la selección de la imagen cargada.

La imagen seleccionada aparece en el espacio de trabajo del HMI, se la dimensiona, se desea que la imagen quede como título y de personalización al HMI.

3.2.18. Creación de indicadores de datos en el HMI en CoDeSys

Figura 54

Imagen insertada en el HMI.



Nota. En esta imagen se muestra la imagen insertada en el HMI.

En el panel de herramientas del lado derecho de la pantalla, esta una lista de opciones, se busca y selecciona “Base”, debajo de la lista aparecen las herramientas correspondientes a la lista “Base”, se busca la opción “Rectángulo” y se la arrastra el icono hacia el panel de trabajo.

Figura 55

Indicadores en el HMI de CoDeSys.

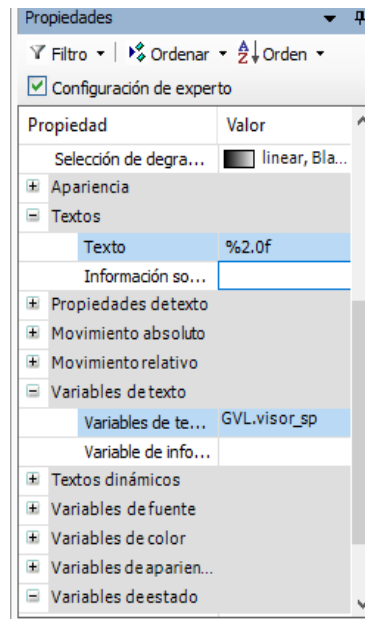


Nota. En esta imagen se muestra la colocación de indicadores en el HMI.

Ese rectángulo estará destinado para visualizar el set point, se debe cambiar algunas de las configuraciones para poder visualizar valores, en las opciones de la figura se coloca el comando “%2.0f”, este comando le permite a la figura mostrar valores, en la parte de “Variables de texto” dentro de las propiedades de la figura, se le designa la variable global que guarda los valores. La variable global para este indicador es “GVL.visor_sp”

Figura 56

Propiedades del indicador del set point.

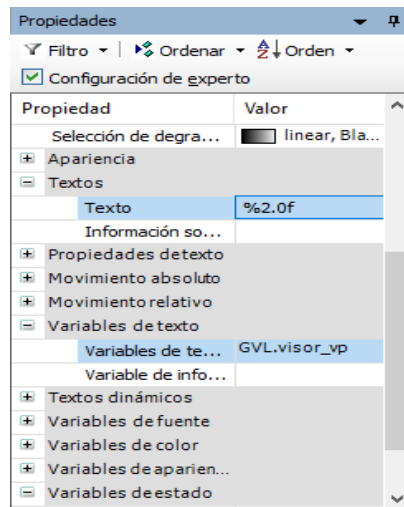


Nota. En esta imagen se muestra las propiedades de los indicadores.

Se coloca otro indicador de la misma manera que el anterior, esta vez para que se pueda visualizar la variable de proceso, se hace el mismo procedimiento que en el indicador del set point.

Figura 57

Propiedades del indicador de la variable de proceso

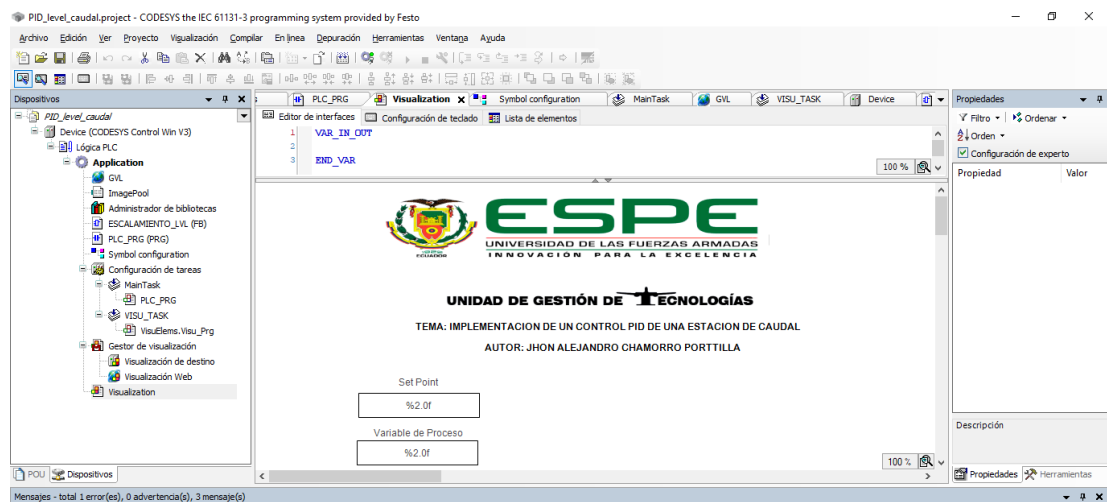


Nota. En esta imagen se muestra las propiedades del indicador de la variable de proceso.

EL HMI se muestra en la Figura 57, cuenta con los indicadores de set point y el de variable de proceso, también con la imagen insertada, el tema del proyecto y el autor,

Figura 58

HMI implementado del CoDeSys.

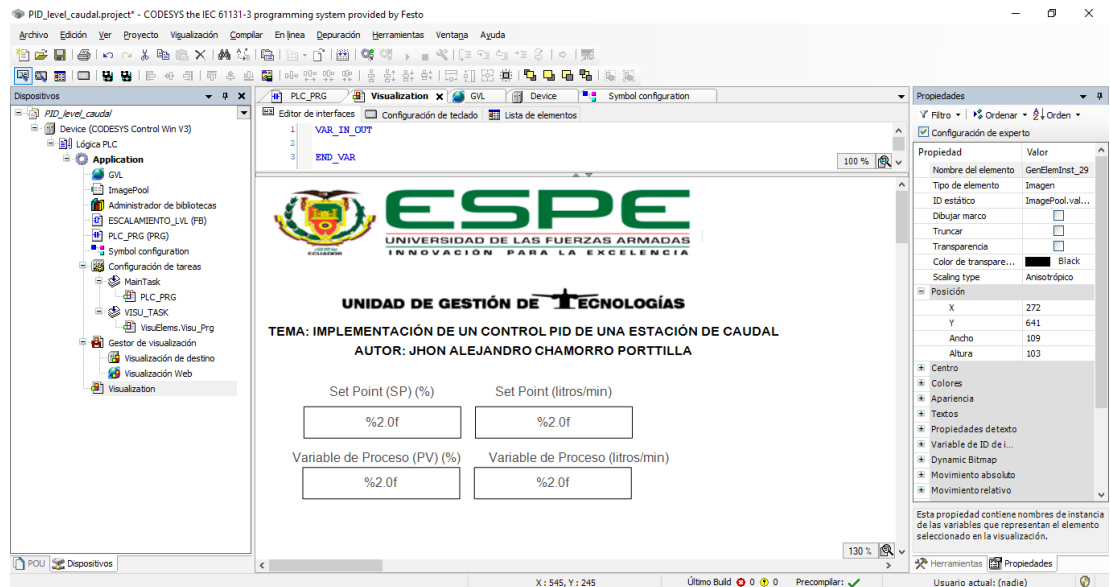


Nota. En esta imagen se muestra instalados los indicadores del set point y de la variable de proceso.

De la misma forma se instalan dos indicadores más, estos mostrarán los valores en litros/minuto, que se están entregando en ese momento, uno leído desde el sensor de flujo que indicara el comportamiento de la variable de proceso y el otro destinado para la lectura del valor en litros/minuto del set point, donde se realiza el mismo procedimiento que se utilizó para los dos indicadores primeros.

Figura 59

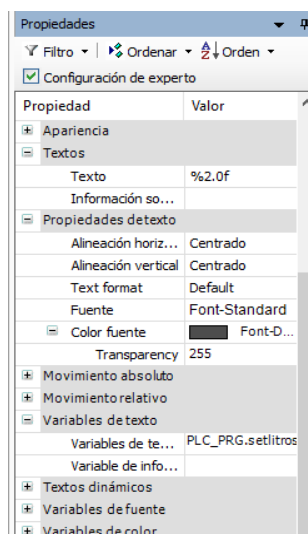
Indicadores para visualización de litros/minuto



Nota. En esta imagen se muestra los indicadores completos.

Figura 60

Propiedades del indicador de litros/minutos del set point.



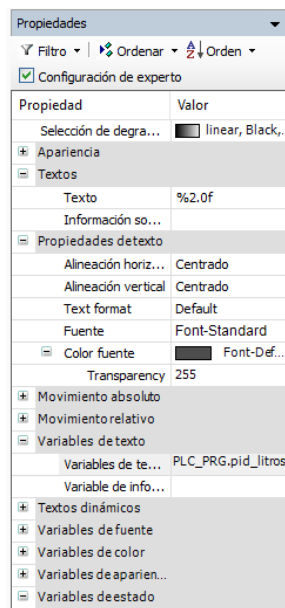
Nota. En esta imagen se muestra las propiedades de los indicadores de litros/minuto del set point. Elaboración Propia

En la Figura 59 se muestra la asignación de variables para el indicador de litros\minuto del set point, para lo cual fue necesario realizar un escalamiento.

También se puede observar en la Figura 60 se muestran las propiedades del indicador, donde se observarán los valores de la variable de proceso en litros\minuto y la asignación de la variable correspondiente.

Figura 61

Propiedades del indicador de litros\minutos de la variable de proceso

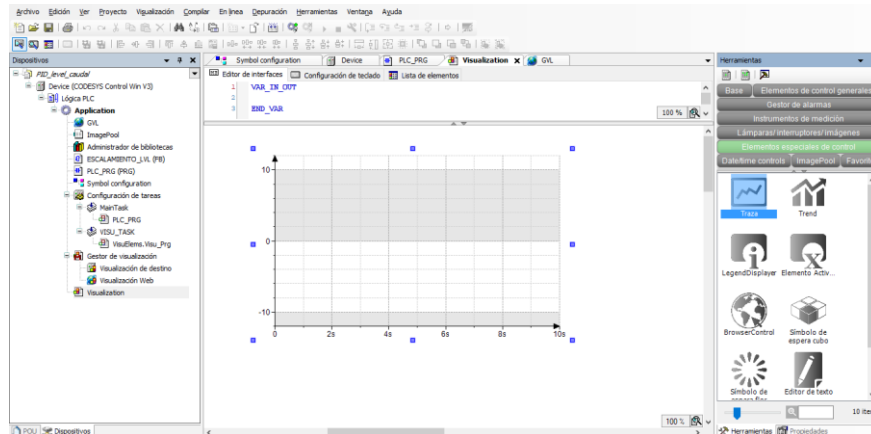


Nota. En esta imagen se muestra las propiedades del indicador de litros/minuto de la variable de proceso.

Se colocó un gráfico con el propósito de observar el comportamiento de las señales, de la variable de proceso y de la del set point, para ellos en la barra de herramientas, se escoge "Elementos especiales de control" y debajo se escoge el primer icono y se lo arrastra a la hoja de trabajo dentro de la visualización del HMI.

Figura 22

Colocación de gráfica en el HMI

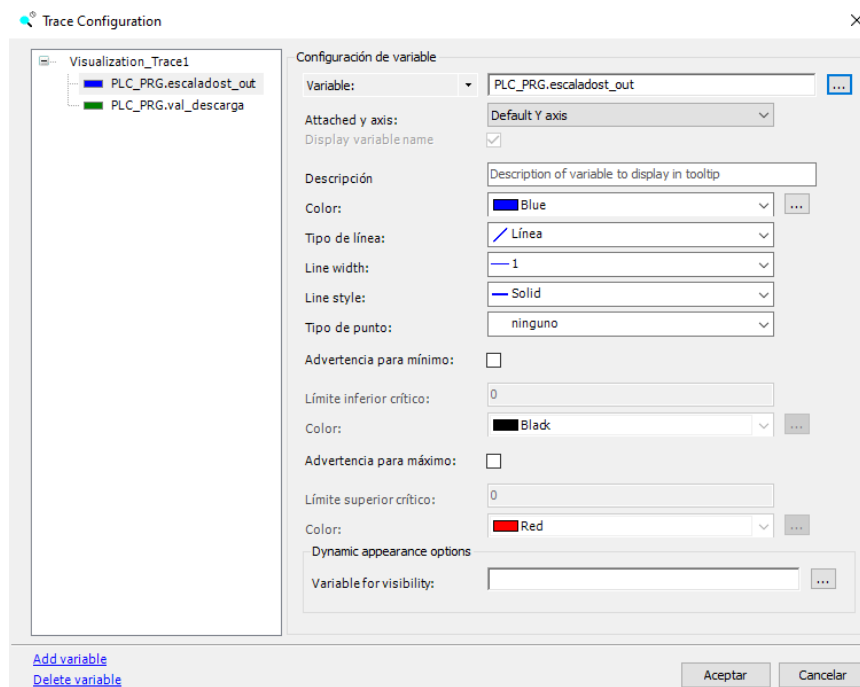


Nota. En esta imagen se muestra la colocación de la gráfica.

Con la gráfica colocada, se debe asignar las señales que se desean observar, para ello en las propiedades de la gráfica, se busca la pestaña “Traza”, se da clic y se abre una ventana, donde se debe dar clic en añadir variable.

Figura 63

Asignación de la primera variable en la gráfica

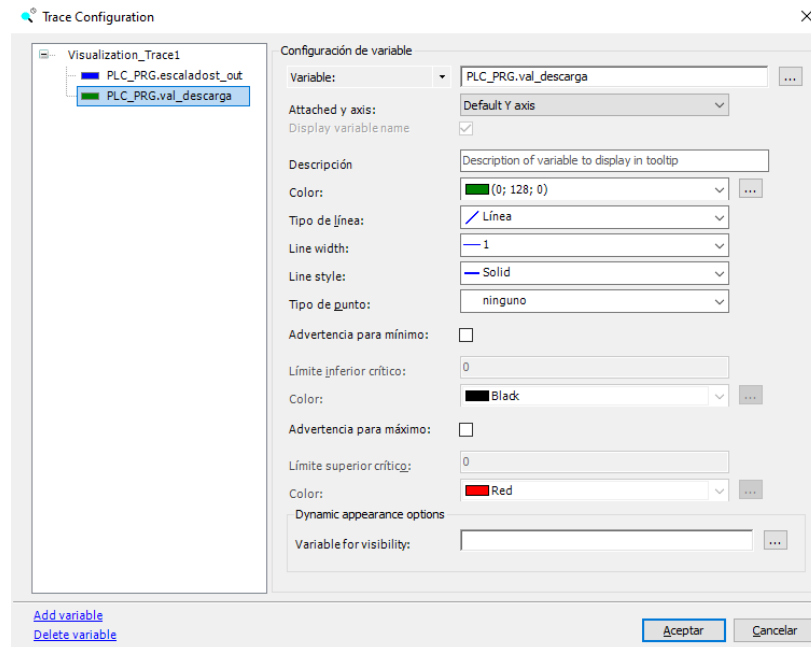


Nota. En esta imagen se muestra la asignación de la primera variable a la gráfica.

Se asigna la variable correspondiente, en la Figura 62 se muestra asignada la variable llama “escalado_out” que es la señal escalada proveniente del set point.

Figura 64

Asignación de la segunda variable en la gráfica

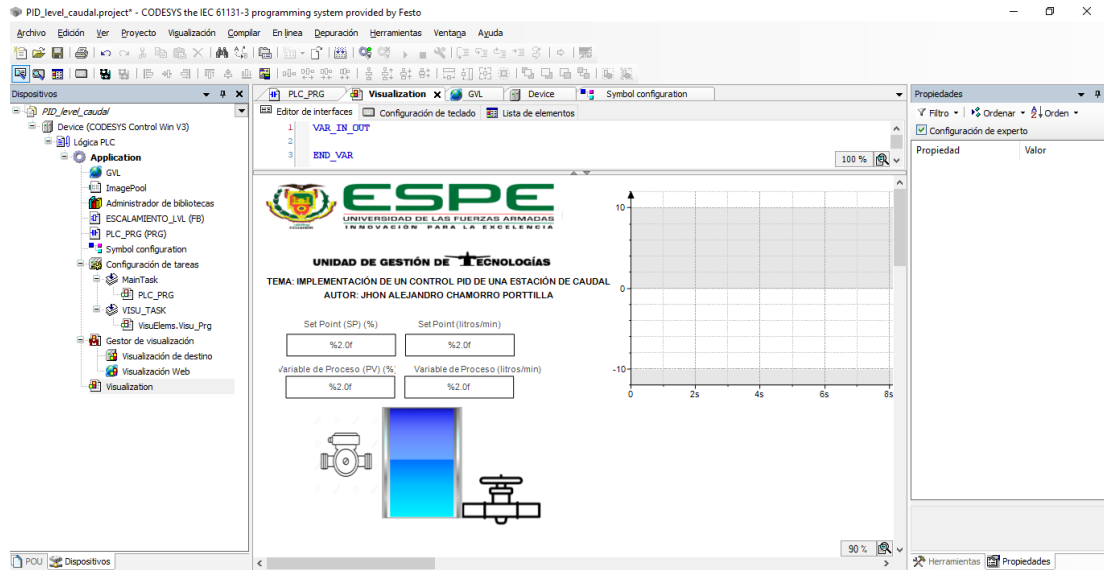


Nota. En esta imagen se muestra la asignación de la segunda variable a la gráfica.

Se agrega la segunda variable en la gráfica, como lo muestra la Figura 63, es la señal proveniente de la variable de proceso, de esa manera se observará el comportamiento de la variable de proceso con respecto a la del set point. Se añadió una animación representativa del sistema utilizando imágenes que se encuentran dentro del CoDeSys y externas también, y el resultado con todos los elementos dentro del HMI se muestra en la Figura 64.

Figura 65

HMI Final



Nota. En esta imagen se muestra el HMI final.

3.3. Simulación en Factory I/O

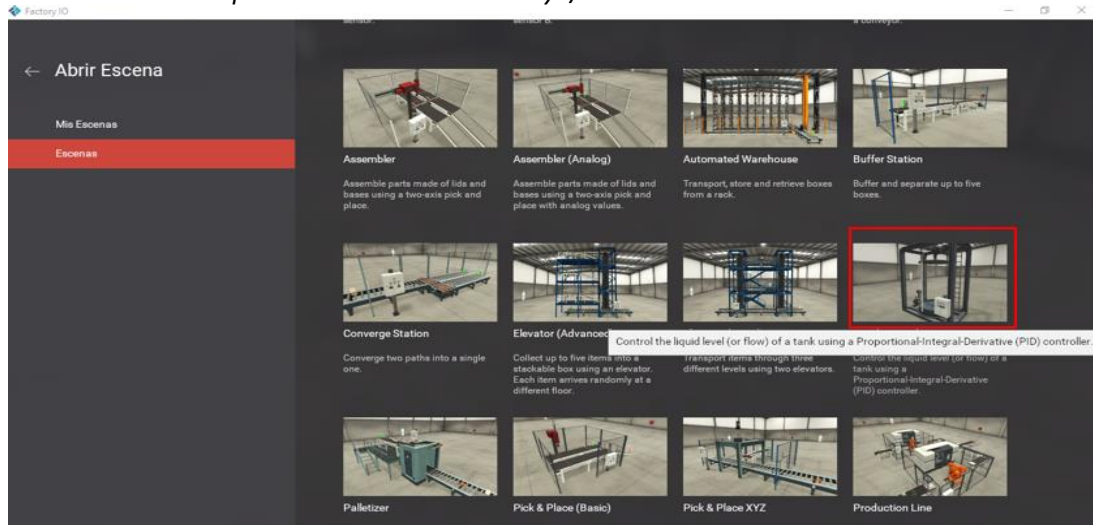
Factory I/O cuenta con varias escenas de ejemplo, implementaciones útiles para simular procesos industriales y de automatización en 3D, la escena escogida para el proyecto es la del control de nivel de un tanque perteneciente a las escenas de ejemplo.

Ejecutado el programa Factory I/O, en el menú de la izquierda se debe escoger “Escenas” y se despliega un abanico de escenas, se escoge la escena “Level Control”, escena apropiada para un control PID ya sea de nivel o de caudal.

Dentro de la escena, la válvula de descarga tiene una capacidad máxima de flujo de 30 litros\minutos, valor que depende del nivel existente en el tanque.

Figura 66

Selección de escena para control PID en Factory I/O.

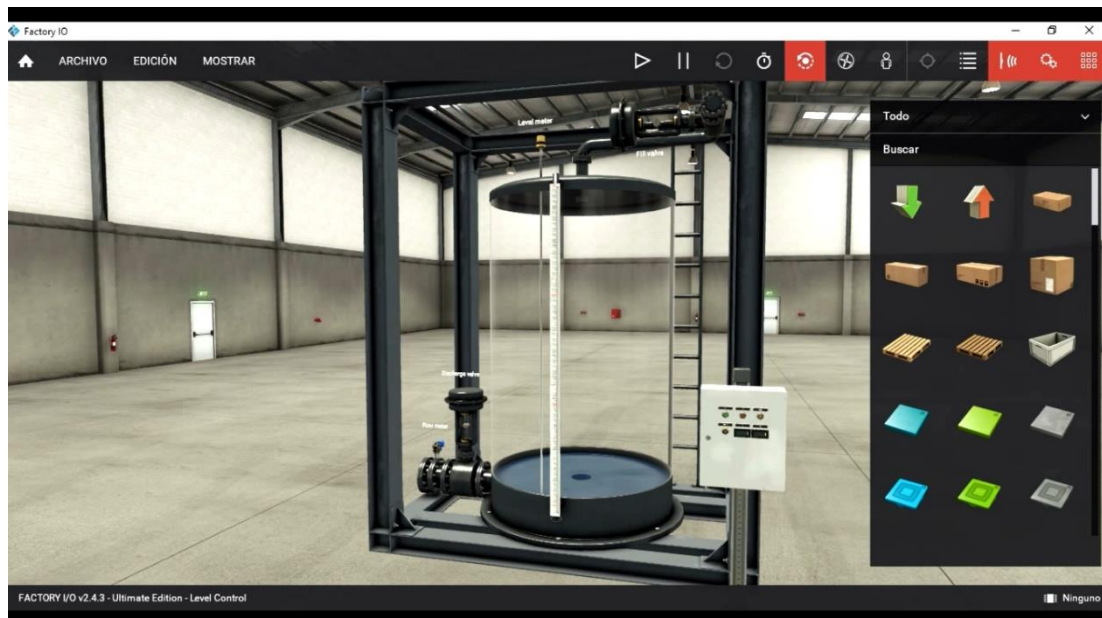


Nota. En esta imagen se muestra la selección de la escena en Factory I/O.

Una vez seleccionada la escena, se muestra la estación con todos insumos apropiados para un control PID.

Figura 67

Escena de control PID en Factory I/O



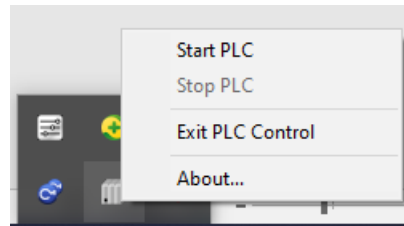
Nota. En esta imagen se muestra la escena cargada en Factory I/O.

3.4. Comunicación entre el Software CoDeSys y Factory I/O mediante OPC

El OPC es un protocolo que se usó para la comunicación entre los programas CoDeSys y Factory I/O, lo primero que ese realizo fue compilar el programa en CoDeSys, una vez verificado que no existan errores ni conflictos, se inicia sesión en línea, significa que se debe encender el PLC virtual e iniciar sesión.

Figura 68

Inicio de sesión de PLC virtual

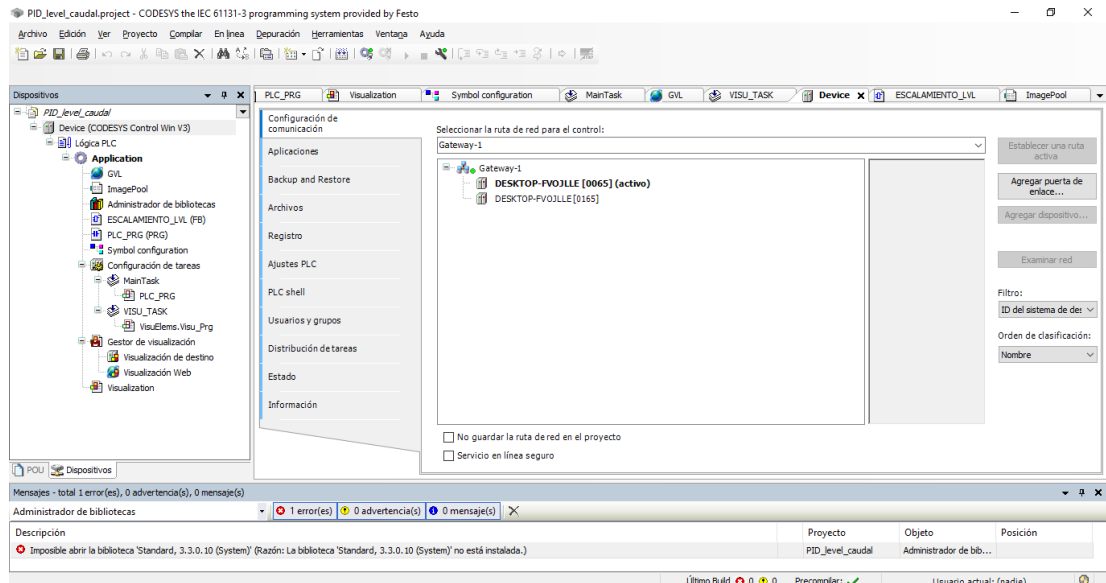


Nota. En esta imagen se muestra el encendido del PLC virtual.

Con el PLC virtual encendido, se debe verificar que el PLC virtual esté vinculado con la computadora, en CoDeSys en la pestaña “Device” se debe revisar dicha conexión.

Figura 69

Vinculación PC con PLC virtual.



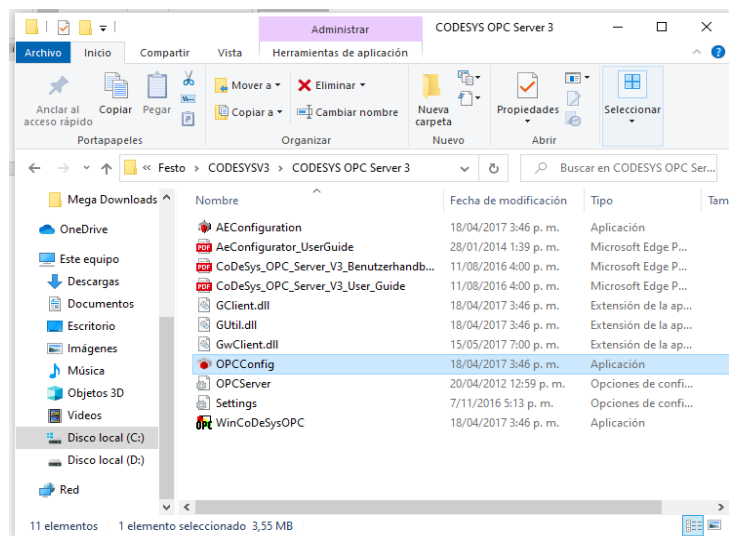
Nota. En esta imagen se muestra la vinculación con el PLC virtual.

Lo siguiente es configurar el OPC desde CoDeSys, para ello se debe ir a la carpeta donde el programa CoDeSys está instalado, y se ejecuta el programa “OPC config”.

Cuando se ejecute el programa “OPC config”, se abre una ventana similar a la de la Figura 72, se debe colocar en la casilla “Rate” 10, para que el sistema OPC se esté actualizando cada 10 milisegundos.

Figura 70

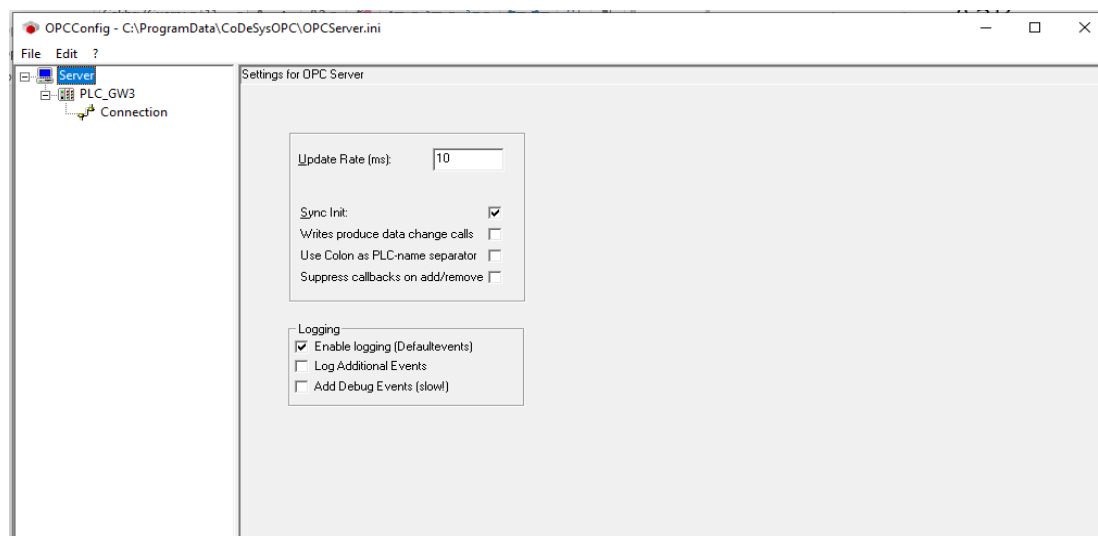
OPC config.



Nota. En esta imagen se muestra el OPC config de CoDeSys.

Figura 71

Ventana de OPC config.



Nota. En esta imagen se muestra el OPC config cargado.

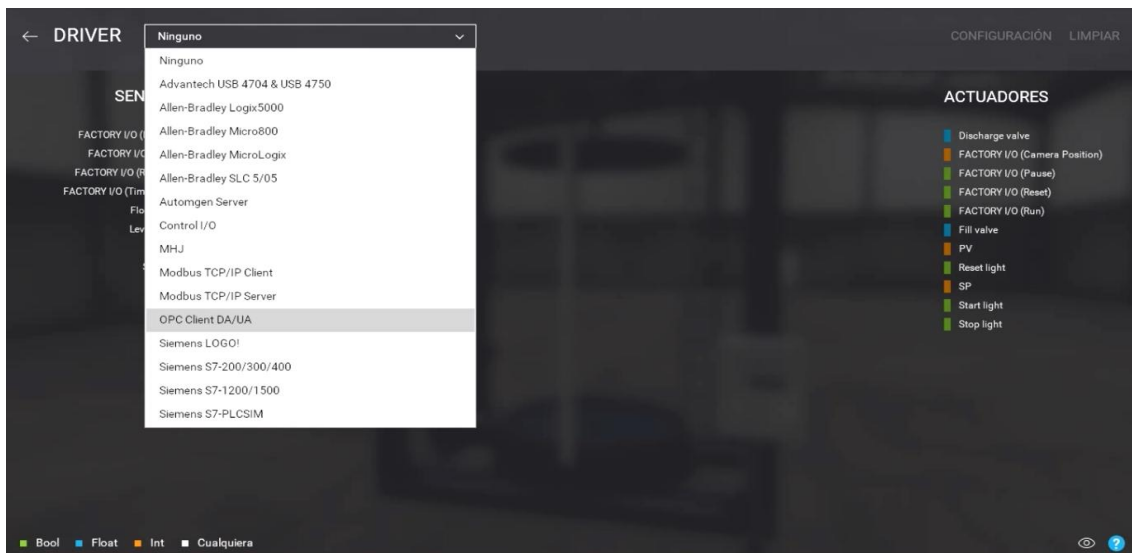
En la ventana del OPC config una vez lleno la casilla “Rate”, se manda a guardar y la ventana se debe minimizar.

Se debe verificar en el lado izquierdo de la ventana del OPC config, que esté vinculado, normalmente se vincula automáticamente.

En Factory I/O, una vez en la escena, de debe ir a las configuraciones de la escena y se selecciona Driver, aparecerá una ventana, con una lista de dispositivos compatibles, se selecciona “OPC Client DA/UA”.

Figura 72

Configuración del OPC en Factory I/O



Nota. En esta imagen se muestra el protocolo de comunicación en Factory I/O.

Se selecciona el protocolo OPC, luego se hace clic en configuración en el lado superior derecho de la pantalla de Factory I/O, aparecerá una ventana nueva.

En esta parte se selecciona “COSESYS OPC UA Server”, y se da clic en el botón “Muestra”.

Figura 73

Selección del servidor OPC en Factory I/O



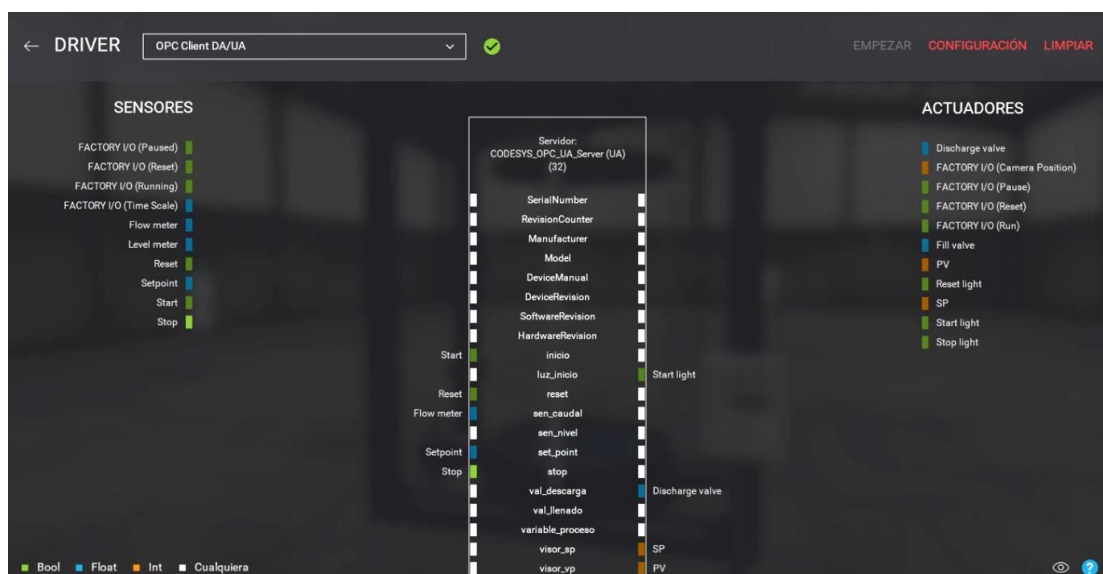
Nota. En esta imagen se muestra la selección del servidor OPC.

Aparece una figura similar a un circuito integrado donde están señaladas las variables globales que se asignó desde CoDeSys, se debe unir como corresponda, arrastrando las variables de Factory I/O a las variables de CoDeSys.

Hecha la conexión correspondiente, se debe volver a la escena y dar inicio para realizar la simulación del control PID, también se debe dar inicio desde CoDeSys.

Figura 74

Conexión entre las variables Factory I/O y las variables de CoDeSys



Nota. En esta imagen se muestra la asignación de todas las variables globales.

En la Figura 74 se muestra el HMI del proyecto cuando se está ejecutando, se puede observar los valores del SP y PV, el SP está en 25 % y refleja 7 lt/min, la PV está estabilizada en 25% y el indicador muestra 7 lt/min, en la parte derecha se puede el grafico y como las señales están en 25 mostrando la eficacia del sistema, la imagen es capturada con la cámara de un teléfono móvil, debido a la poca capacidad de procesamiento de la computadora utilizada , no permite utilizar otro programa en adición a los tres que se están ejecutando

Figura 75

Simulación del proyecto



Nota. En esta imagen se muestra la simulación del proyecto, vista del HMI.

En la Figura 75 se muestra la ejecución de la simulación del control PID, el SP está situado en el 51%, los indicadores del HMI en CoDeSys y de los displays de Factory I/O los valores son similares y muestran valores iguales a los esperados, en el indicador muestra que cuando el SP está en 51% la válvula de descarga entrega 15 lt/min.

En el tablero de control de Factory I/O se puede observar los valores del SP y de la PV, donde se relacionan a los valores mostrados en los indicadores del HMI de CoDeSys. Poseen valores similares lo que indica el correcto funcionamiento de la comunicación OPC y del sistema mismo.

Figura 76

Simulación del sistema con un SP de 51%

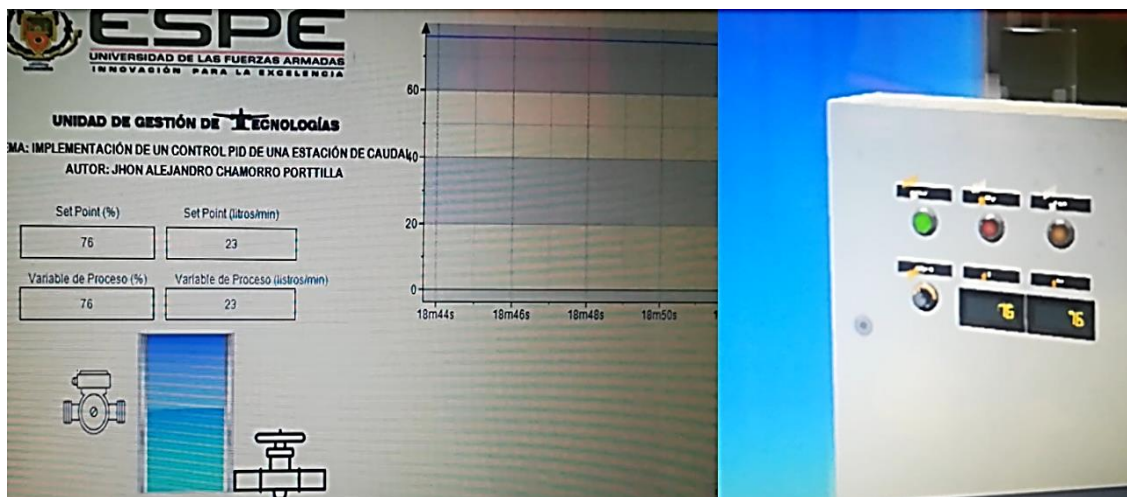


Nota. En esta imagen se muestra la simulación del proyecto con un SP de 51%, vista del HMI y del tablero de control de Factory I/O.

En la Figura 76 se muestra la ejecución de la simulación del control PID, con un SP de 76% y se puede observar como coinciden los valores de SP y PV tanto en los indicadores del HMI como en los displays del tablero de control de Factory I/O, en el HMI, los indicadores muestran que con un SP de 76% la valvula de descarga entrega 23 lt/min.

Figura 77

Simulación del sistema con un SP de 76 %



Nota. En esta imagen se muestra la simulación del proyecto con un SP de 76%, vista del HMI y del tablero de control de Factory I/O.

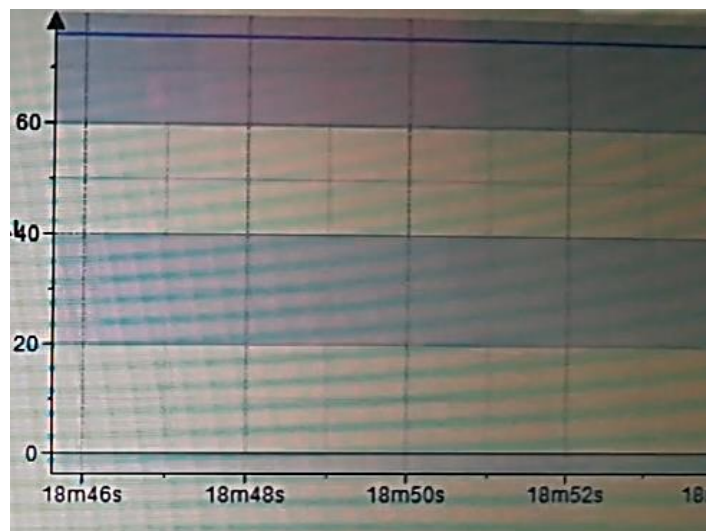
3.5. Análisis de resultados

Las variables SP y PV muestran valores similares en los indicadores de los programas CoDeSys y Factory I/O, demostrando favorablemente el envío y recepción de datos y se pudo comprobar satisfactoriamente la comunicación de estos programas mediante el protocolo OPC.

Las constantes para la sintonización de control PID, utilizando el método del tanteo, son las siguientes: constante proporcional (K_p) = 0,1, constante integral (T_i) = 0,001, constante derivativa (T_d) = 0,0001.

Figura 78

Gráfica de señales con un SP y PV de 76%



Nota. En esta imagen se muestra la simulación del proyecto con un SP y PV de 76%, vista del HMI.

Capítulo IV

4. Conclusiones y recomendaciones

4.1. Conclusiones

- Se logró desarrollar un programa para un control PID utilizando el lenguaje de programación ladder para una estación de caudal simulada en el software Factory I/O, mediante el uso de un PLC virtual perteneciente al software CoDeSys.
- Para establecer la comunicación entre el software de programación CoDeSys y el software de simulación Factory I/O es necesario utilizar el OPC server que viene instalado junto con CoDeSys, obteniendo resultados favorables en el envío y recepción de datos.
- Se diseñó un HMI en el software CoDeSys para la visualización de los valores del set point y la variable de proceso del control PID de una estación de caudal.
- Factory I/O es un software para simular procesos industriales automatizados que cuenta con escenas reales en 3D, útiles para representar procesos, tiene una amplia compatibilidad con las marcas comerciales de automatización más conocidas.
- Se verificó el funcionamiento del control PID de la estación de caudal simulada en el software Factory I/O y programada desde el software CoDeSys con el bloque de función de control PID de la librería OScat
- Se logró entablar comunicación entre los programas CoDeSys y Factory I/O, demostrando la compatibilidad, que no se logró con el software RSLogix 500, debido principalmente a que es un programa desactualizado y no contempla la compatibilidad en la comunicación OPC con el software Factory I/O, también al difícil acceso a los programas que se requería para la comunicación como es el caso del RSLinx OEM.

4.2. Recomendaciones

- El Software CoDeSys cuenta con un OPC server, en las versiones inferiores a la 3.5.12.0 el OPC está incorporado con el paquete de instalación del programa y se instala con todo el paquete de CoDeSys, en las versiones superiores a la 3.5.12.0 en OPC se instala por separado, se debe descargar desde la página CoDeSys Store, cuenta con una licencia que tiene un valor económico, pero tiene un período de 30 días de prueba.
- La librería OScat es de acceso libre, es la que se utilizó para implementar el bloque de función del control PID, existe también la librería "Util", que esta agregada en CoDeSys por defecto.
- El método de sintonización del control PID utilizado fue el método del tanteo, descrito en la fundamentación teórica donde se detalla los pasos a seguir para sintonizar el control PID, las constantes para la sintonización de control PID fueron: $K_p = 0,1$, $T_i = 0,001$, $T_d = 0,0001$, se pudo alcanzar un sistema equilibrado y eficaz.
- Desarrollar el proyecto y su simulación en general en una computadora con un procesador superior a CORE i3 puede mejorar significativamente la ejecución de los programas.
- Al momento de iniciar sesión con el PLC desde CoDeSys, aparece una ventana donde se debe seleccionar "iniciar sesión con descarga" para que la gráfica de señales pueda ejecutarse en simultáneo con el programa y no genere errores.

5. Bibliografía

Alberto, J. (24 de 12 de 2014). *noeju.com & plcpc.com*. Obtenido de <https://www.noeju.com/crear-driver-serie-en-rslnx-comunicar-a-traves-de-puerto-com-0-de-micrologix-1400/>. Recuperado el 23 de Febrero del 2021.

Automation, R. (agosto de 2005). *Rockwell Automation*. Obtenido de <https://www.roydisa.es/wp-content/uploads/2013/01/controladores-programables-micrologix-1100-rockwell.pdf>. Recuperado el 15 de Enero del 2021.

Automation, R. (2008). *MicroLogix Programmable Controller Selection Guide*. Milwaukee: Rockwell Automation. . Recuperado el 12 de Enero del 2021.

Automation, R. (2010). *MicroLogix Instruction Set Help*. Milwaukee, Wisconsin, Estados Unidos. . Recuperado el 13 de Enero del 2021.

Autracen. (31 de 10 de 2017). *Autracen*. Obtenido de <http://www.autracen.com/las-redes-industriales-allen-bradley/>. Recuperado el 13 de Enero del 2021.

AVILA, M. J. (AGOSTO de 2002). *Users*. Obtenido de Users: <file:///C:/Users/usuario/Downloads/17782.pdf>. Recuperado el 12 de Febrero del 2021.

Cando, A. (21 de 11 de 2008). *ResearchGate*. Obtenido de https://www.researchgate.net/figure/Figura-18-Diagrama-en-bloques-de-un-control-PID_fig4_27558350. Recuperado el 7 de Enero del 2021.

CARRICA, D. (21-25 de SEPTIEMBRE de 1999). Obtenido de <Users/usuario/Documents/AlgoritmoeficienteSecure.pdf>. Recuperado el 26 de Febrero del 2021.

Cero, I. (2 de 9 de 2016). *Resarch Gate*. Obtenido de https://www.researchgate.net/figure/Figura-4-Graficas-del-control-PID-anidado-aplicado-a-la-planta-definida-por-la-ecuacion_fig4_308063447. Recuperado el 9 de Febrero del 2021.

Chuck, D. (2012). *Los sistemas de primer orden y los Controladores PID*. San Juan, Argentina: Universidad de San Juan.

ClasificaciónDe. (1 de 3 de 2019). *ClasificaciónDe*. Obtenido de <https://www.clasificacionde.org/tipos-de-plc/#:~:text=El%20Plc%20se%20clasifica%20en.> Recuperado el 30 de Enero del 2021.

Coba, W. D. (2005). *Sistema de control PID con visualizadores de programación gráfica para automatización de la temperatura de un horno para prácticas*. México.D.F.: Universidad Nacional Autónoma de México.

Cruz, J. M. (2010). *Desarrollo de un control PID wavenet*. México.D.F. : Universidad Nacional Autónoma de México.

Docencia, E. (2011). *ESPECIFICACIONES DE LA RESPUESTA TRANSITORIA*. Pereira, Colombia: Universidad Tecnológica de Perreira.

Electronics, O. (24 de 12 de 2019). *Opiron Electronics*. Obtenido de <https://www.opiron.com/que-es-codesys/>. Recuperado el 24 de Febrero del 2021.

Electronics, O. (24 de 12 de 2019). *Opiron Electronics*. Obtenido de <https://www.opiron.com/pou-debo-usar/>. Recuperado el 15 de Febrero del 2021.

Games, R. (21 de 12 de 2012). *Docs. Factory I/O*. Obtenido de <https://docs.factoryio.com/system-requirements/>. Recuperado el 23 de Febrero del 2021.

García, P. (2013). *El aumento de las aplicaciones de los control PID dentro de la industria y como favorece al mercado*. Sevilla, España: Universidad de Sevilla.

Gómez, F. (26 de 3 de 2018). *AUTOMANTENIMIENTO.NET*. Obtenido de <https://automantenimiento.net/electricidad/partes-de-un-plc/>. Recuperado el 12 de Enero del 2021.

Gürocak, H. (. (25 de Marzo de 2018). *Industrial Motion Control. Vancouver*. Obtenido de Industrial Motion Control. Vancouver: NI: <http://www.ni.com/tutorial/7450/en/>. Recuperado el 13 de Febrero del 2021.

Gurung, R. (2018 de 12 de 22). *Xybernetics*. Obtenido de <http://xybernetics.com/tag/rslogix-500/>. Recuperado el 17 de Febrero del 2021.

Gutiérrez Torrijos, F. (2015). *Sistema de seguridad con visualizadores de programación gráfica para procesos de control*. México.D.F.: Universidad Nacional Autónoma de México .

Indriago, M. (6 de 9 de 2019). *Control Real*. Obtenido de <https://controlreal.com/es/pid-como-sintonizar-un-lazo-pid/>. Recuperado el 31 de Enero del 2021.

INDUSTRIAL, C. D. (s.f.). Obtenido de <https://www.cursosaula21.com/que-es-un-hmi/>. Recuperado el 30 de Febrero del 2021.

INDUSTRIAL, S. T. (s.f.). Obtenido de <https://www.cursosaula21.com/que-es-un-hmi/>. Recuperado el 30 de Febrero del 2021.

Ingeniería, R. A. (3 de 12 de 2012). *Real Academia de Ingeniería*. Obtenido de <http://diccionario.raing.es/es/lema/sobreimpulso>. Recuperado el 13 de Febrero del 2021.

Jorge, B. (6 de 10 de 2017). *ResearchGate*. Obtenido de https://www.researchgate.net/figure/Estacion-de-control-de-flujo-de-caudal-III-SOFTWARE_fig2_322214077. Recuperado el 13 de Diciembre del 2020.

Lucas, S. L. (21 de 10 de 2012). *Sintonización PID*. Obtenido de <https://es.slideshare.net/santylucas1/sintonizacin-pid#:La%20sintonizaci%C3%B3n%20>. Recuperado el 4 de Marzo del 2021.

Mecafenix, I. (22 de Junio de 2020). *Ingeniería Mecafenix*. Obtenido de <https://www.ingmecafenix.com/automatizacion/que-es-un-plc/>. Recuperado el 7 de Febrero del 2021.

Moreno, M. A. (2001). *Apuntes de control PID*. La Paz, Bolivia: Universidad Mayor de San Andrés.

Networks, A. (12 de 02 de 2009). *Automation Networks*. Obtenido de <http://automation-networks.es/glossary/rslogix-500>. Recuperado el 17 de Diciembre del 2020.

Ogatha, K. (2010). *Sistemas de control automático*. Quito,Ecuador: Escuela Politécnica Nacional.

Oscat. (4 de 3 de 2013). *Oscat*. Obtenido de https://www.tecomat.com/uploads/files/sw/Mosaic/OSCAT/oscat_basic333_en.pdf. Recuperado el 7 de Febrero del 2021.

Pardo, C. (12 de 03 de 2013). *Picuino*. Obtenido de Tecno Recursos: <https://www.picuino.com/es/arduprog/control-pid.html>. Recuperado el 19 de Febrero del 2021.

S.A.S., I. D. (12 de 02 de 2017). *ICL Didáctica S.A.S*. Obtenido de <https://www.icl-didactica.com/factory-i-o/#:~:text=Factory%20I%2FO%20es%20un,tecnolog%C3>. Recuperado el 12 de Febrero del 2021.

Sánchez, A. E. (25 de 10 de 2013). *CTIN*. Obtenido de <http://www.ctinmx.com/que-es-un-plc/>. Recuperado el 27 de Enero del 2021.

Santiago, P. (2 de 7 de 2017). *PLC Santiago*. Obtenido de <http://plcsantiago.blogspot.com/2017/07/tipos-de-plc.html>. Recuperado el 7 de Febrero del 2021.

Seika. (18 de 4 de 2020). *Seika Automation*. Obtenido de <https://www.seika.com.mx/que-es-un-plc/>. Recuperado el 17 de Febrero del 2021.

SIEMENS. (s.f.). *CONTROLADOR PROGRAMABLE S7-1200*. Obtenido de https://media.automation24.com/manual/es/91696622_s71200_system_manual_es-ES_es-ES.pdf. Recuperado el 12 de Febrero del 2021.

Smith, S. (20 de 11 de 2017). *techlandia*. Obtenido de https://techlandia.com/son-variables-instancia-java-info_197306/. Recuperado el 13 de Enero del 2021.

Store, C. (12 de 4 de 2017). *CODESYS Store*. Obtenido de https://store.codesys.com/codesys.html?__store=en#System%20Requirements. Recuperado el 30 de Marzo del 2021.

Villajulca, J. C. (7 de 11 de 2018). *Instrumentación y Control.net*. Obtenido de <https://instrumentacionycontrol.net/caracteristicas-practicas-de-controladores-pid/>. Recuperado el 13 de Marzo del 2021.

6. Anexos