

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**  
**SEDE VIÑA DEL MAR-JOSÉ MIGUEL CARRERA**

**DESARROLLO DE INTERFAZ SCADA/HMI EN WINCC PARA EL MONITOREO DE UN  
SISTEMA DE CONTROL DE NIVEL**

Trabajo de Titulación para optar al Título de  
Ingeniería de Ejecución en Control e  
Instrumentación Industrial.

Alumno:

Roberto Andrés San Martín Pino

Profesor Guía:

Mag. Loreto Marín Carcey

**2024**

## **RESUMEN**

**KEYWORDS:** PID, Factory IO, Control de nivel.

Esta memoria se enfoca en el desarrollo de interfaz SCADA/HMI en WINCC para el monitoreo de un sistema de control de nivel que se implementará de manera remota a través del programa Factory IO que permite la simulación de procesos industriales a escala real y tiempo real, esto se realizará en la Universidad Técnica Federico Santa María Sede Viña del Mar, planteada como solución a la necesidad de que tengan una experiencia práctica más cercana a la que se enfrentarán los alumnos al entrar al mundo laboral, que es cuestión fundamental en la automatización de procesos a nivel industrial.

Lo que se hará es monitorear y controlar un sistema de control de nivel en un estanque de manera virtual simulando todo el proceso en tiempo real.

Esta memoria se dividirá en tres capítulos en donde el capítulo 1 se enfocará en conocer las alternativas de soluciones que se pueden dar a la problemática inicial y la solución que se implementará. El capítulo 2 se enfocará al diseño de los programas tanto como el TIA PORTAL y Factory IO. El capítulo 3 está dedicado al funcionamiento del programa diseñado en el software Factory IO además de los costos del proyecto.

## ÍNDICE

|                                                            |    |
|------------------------------------------------------------|----|
| INTRODUCCIÓN .....                                         | 1  |
| CAPÍTULO 1: ANTECEDENTES PREVIOS.....                      | 2  |
| 1.1. UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA.....         | 3  |
| 1.2. PLANTEAMIENTO DEL PROBLEMA.....                       | 5  |
| 1.3. ALTERNATIVAS DE SOLUCIÓN.....                         | 5  |
| 1.3.1. ALTERNATIVAS                                        |    |
| 1.3.1.1. Alternativa 1 .....                               | 5  |
| 1.3.1.2. Alternativa 2 .....                               | 5  |
| 1.3.1.3. Alternativa 3 .....                               | 6  |
| 1.3.2. ALTERNATIVA SELECCIONADA .....                      | 7  |
| 1.3.3. Programmable Logic Controller (PLC) .....           | 8  |
| 1.3.3.1. Componentes .....                                 | 9  |
| 1.3.3.2. Alimentación y carcasa.....                       | 9  |
| 1.3.3.3. Unidad central de procesamiento (CPU).....        | 9  |
| 1.3.3.4. Sistemas de entrada y salida (input-output) ..... | 9  |
| 1.3.4. El Ciclo Scan .....                                 | 10 |
| 1.3.5. Human Machine Interface (HMI).....                  | 11 |
| 1.3.6. Factory IO.....                                     | 12 |
| 1.4. OBJETIVOS DEL PROYECTO .....                          | 14 |
| CAPÍTULO 2: DESARROLLO DE LA SOLUCIÓN .....                | 15 |
| 2.1. COMUNICACIÓN ENTRE PROGRAMAS .....                    | 16 |
| 2.1.1. Descarga de programa a trabajar .....               | 16 |
| 2.1.2. Elección de escena .....                            | 18 |
| 2.1.2. Configuración en Factory IO.....                    | 19 |
| 2.2. ESQUEMA DE CONTROL .....                              | 21 |
| 2.3. CONTROLADOR PID .....                                 | 22 |
| 2.4. DISEÑO DEL PROGRAMA .....                             | 23 |
| 2.4.1. Primer segmento .....                               | 23 |
| 2.4.2. Segundo segmento .....                              | 23 |
| 2.4.3. Tercer segmento .....                               | 24 |
| 2.4.4. Cuarto segmento .....                               | 27 |
| 2.4.5. Quinto segmento .....                               | 27 |
| 2.4.6. Bloque de organización .....                        | 28 |

2.5. DISEÑO DE LA INTERFAZ HMI ..... 29

CAPÍTULO 3: FUNCIONAMIENTO Y COSTOS DEL PROYECTO ..... 31

3.1. FUNCIONAMIENTO DEL PROGRAMA ..... 32

3.2. COSTOS DEL PROYECTO..... 42

CONCLUSIONES..... 43

BIBLIOGRAFÍA ..... 44

## ÍNDICE DE FIGURAS

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| Figura 1-1. Logo de UTFSM.....                                                | 4  |
| Figura 1-2. Sede Viña del Mar UTFSM.....                                      | 4  |
| Figura 1-3. Ejemplo de implementación en Factory IO .....                     | 6  |
| Figura 1-4. PLC Siemens S7-1200.....                                          | 8  |
| Figura 1-5. Ciclo Scan.....                                                   | 11 |
| Figura 1-6. HMI KTP700 .....                                                  | 12 |
| Figura 1-7. Simulación 3D Factory IO .....                                    | 13 |
| Figura 2-1. Planillas a descargar según versión del TIA PORTAL .....          | 17 |
| Figura 2-2. Bloque de comunicación entre TIA PORTAL y Factory IO.....         | 17 |
| Figura 2-3. Escenas en Factory IO.....                                        | 18 |
| Figura 2-4. Configuración en programa Factory IO primera parte .....          | 19 |
| Figura 2-5. Configuración en programa Factory IO segunda parte .....          | 20 |
| Figura 2-6. Esquema de control .....                                          | 21 |
| Figura 2-7. Esquema de control .....                                          | 21 |
| Figura 2-8. Programa realizado en el segundo segmento .....                   | 23 |
| Figura 2-9. Programa realizado en el tercer segmento .....                    | 24 |
| Figura 2-10. Programa del bloque de función “Set Point” .....                 | 25 |
| Figura 2-11. Programa del bloque de función “Señal Sensor” .....              | 25 |
| Figura 2-12. Programa del bloque de función “Salida PID” .....                | 26 |
| Figura 2-13. Programa realizado en el cuarto segmento .....                   | 27 |
| Figura 2-14. Programa realizado en el quinto segmento .....                   | 27 |
| Figura 2-15. Programa realizado en el bloque de organización .....            | 28 |
| Figura 2-16. Versión a utilizar del PID_Compact .....                         | 28 |
| Figura 2-17. Interfaz HMI.....                                                | 29 |
| Figura 2-18. Interfaz HMI.....                                                | 30 |
| Figura 3-1. Ajustes básicos.....                                              | 33 |
| Figura 3-2. Ajustes del valor real .....                                      | 34 |
| Figura 3-3. PID compact.....                                                  | 35 |
| Figura 3-4. Ventana puesta de servicio .....                                  | 36 |
| Figura 3-5. Ventana puesta de servicio con la optimización fina en curso..... | 37 |
| Figura 3-6. HMI con la optimización fina en curso .....                       | 37 |
| Figura 3-7. Parámetros PID después de optimización fina .....                 | 38 |
| Figura 3-8. Cambio manual de los parámetros PID.....                          | 39 |
| Figura 3-9. Resultados de la ventana puesta en servicio .....                 | 40 |
| Figura 3-10. Resultados de la pantalla HMI .....                              | 40 |

Figura 3-11. Costos del Proyecto.....42

**ÍNDICE DE TABLAS**

Tabla 1-1. Pauta Evaluación Alternativas ..... 7

Tabla 1-2. Evaluación de propuestas ..... 7

## **SIGLAS Y SIMBOLOGÍAS**

### **A. SIGLAS:**

CPU : Unidad central de procesamiento

HMI : Human Machine Interface

PLC : Programmable Logic Controller

### **B. SIMBOLOGÍAS:**

m : metros

V : Voltaje

VAC : Voltaje alterno

VDC : Voltaje continuo

I : Corriente

AC : Corriente alterna

DC : Corriente continua

rpm : Revoluciones por minuto

Hz : Hertz

P : Potencia activa

W : Watts

S : Potencia aparente

Q : Potencia reactiva

V<sub>máx</sub>: Voltaje máximo

I<sub>máx</sub> : Corriente máxima

mA : miliamperio

## **INTRODUCCIÓN**

Actualmente, con el avance de la tecnología en el área de la automatización, las empresas requieren mayor cantidad de equipos que faciliten la labor en la que se desempeñan, reemplazando la labor del ser humano por máquinas más eficientes que abaratan costos en el largo plazo. Esto trae consigo la necesidad de profesionales capacitados en esta área y en constante adquisición de nuevos conocimientos.

La instrumentación puede formar estructuras complejas para medir, controlar y monitorear todos los elementos de un sistema industrial con profundidad y gran exactitud, además de automatizar tales procesos y, a la vez, garantizar la exactitud de las medidas y resultados.

Es por esto que nace la necesidad de desarrollar el monitoreo de un sistema de control de nivel en el laboratorio de la carrera de manera virtual, en donde ahí se tiene que reforzar los conocimientos a los alumnos con el manejo de nuevos equipos y estrategias para estar capacitados a la nueva tecnología que se está implementando en el mundo laboral.

Este proyecto dará el paso a paso para realizar el desarrollo de la solución del monitoreo de un sistema de control de nivel en tiempo real implementado en el software Factory IO, que es un programa de simulación de una fábrica virtual para realizar monitoreo, controlar, transmitir y convertir variables, finalizando con una muestra del uso del sistema y el control aplicado en él del resultado final.

## **CAPÍTULO 1. ANTECEDENTES PREVIOS**

## **1      ANTECEDENTES PREVIOS**

El presente capítulo tiene como objetivo detallar la problemática y posibles soluciones para el monitoreo de un sistema de control de nivel.

### **1.1.    UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**

La Universidad Técnica Federico Santa María es una prestigiosa Universidad de Chile, que es conocida por formar a los Técnicos e ingenieros de gran prestigio, enfocándose en la ciencia, tecnología e ingeniería, aportando programas académicos, investigaciones en la industria junto con su colaboración. La Universidad se establece en el año 1926 en Valparaíso y con el tiempo se fue ampliando a otras ciudades como Santiago, Viña del Mar, Concepción y además de contar con una sede internacional en Ecuador.

La carrera en la que se realizará este proyecto de titulación tiene como formación a Profesionales con sólidos conocimientos en procesos industriales en los cuales se utiliza instrumentación y control automático. Su formación tecnológica e integral, les permite realizar labores de índole interdisciplinarias en el ámbito de control de procesos industriales. Este ingeniero en ejecución en control e instrumentación Industrial; como también a la asistencia técnica, asesoría y adquisición de tecnología de punta para la industria, siendo un profesional capaz de responder con eficacia a los cambios que experimenta la tecnología del área electrónica orientada al control automático pudiendo evaluar, seleccionar, y adaptar esta tecnología a los procesos.

Actualmente, en esta Sede se imparten diversas carreras tanto diurnas como vespertinas, como, por ejemplo, Ingeniería de Ejecución en Control e Instrumentación Industrial.

En la figura 1-1 se puede observar el logo de la Universidad Técnica Federico Santa María. En la figura 1-2 se puede observar la sede de Viña del mar de la Universidad Técnica Federico Santa María.



Fuente: [www.usm.cl](http://www.usm.cl).

Figura 1-1. Logo de UTFSM.



Fuente: [www.usm.cl](http://www.usm.cl).

Figura 1-2. Sede Viña del Mar UTFSM.

## **1.2. PLANTEAMIENTO DEL PROBLEMA**

Este proyecto surge como respuesta a la necesidad de la carrera de Ingeniería de Ejecución en Control e Instrumentación Industrial de la Universidad Técnica Federico Santa María, Sede José Miguel Carrera, de desarrollar un monitoreo de un sistema de control de nivel en el laboratorio de la carrera de manera virtual, para que tengan una experiencia práctica más cercana a la que se enfrentarán los alumnos al entrar al mundo laboral. Esto tiene relación específicamente de dicho conocimiento, cuestión fundamental en la automatización de procesos a nivel industrial.

## **1.3. ALTERNATIVAS DE SOLUCIÓN**

En este apartado se darán 3 soluciones al problema planteado con anterioridad, siendo la opción seleccionada la más adecuada, aplicando criterios como la innovación, beneficio y factibilidad, entre otros.

### **1.3.1 ALTERNATIVAS**

#### **1.3.1.1. Alternativa 1: Mantener condiciones existentes**

Esta propuesta consiste en no realizar ningún tipo de desarrollo de una interfaz, manteniendo la situación actual provocando la falta de conocimiento que es fundamental en la automatización de procesos a nivel industrial.

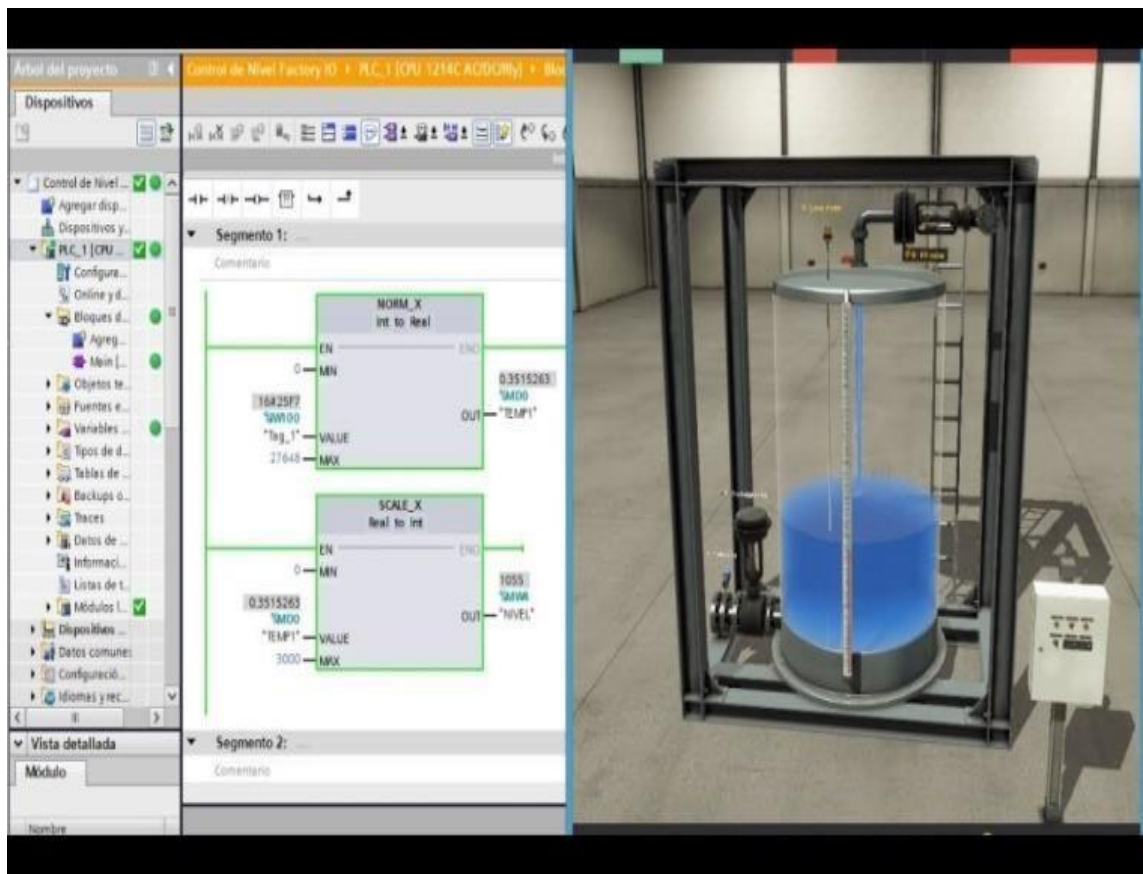
#### **1.3.1.2. Alternativa 2: Desarrollar una interfaz a través de un microcontrolador**

Esta propuesta se basa en realizar una interfaz a través de un microcontrolador, utilizando lenguaje C++ y desarrollar la implementación en programa Factory IO.

### 1.3.1.3. Alternativa 3: Desarrollar una interfaz SCADA/HMI en WINCC implementado en Factory IO.

Esta propuesta se basa en el desarrollo de una interfaz SCADA/HMI en WINCC para el monitoreo de un sistema de control de nivel implementándolo en Factory IO, para realizar una simulación del sistema real, considerando en la implementación simulada todos los materiales, equipos y artefactos para su instalación.

En la figura 1-3 se puede observar un ejemplo de implementación en el programa Factory IO.



Fuente: <https://www.youtube.com/watch?v=1PrapJbbuRM>

Figura 1-3. Ejemplo de implementación en Factory IO.

### 1.3.2. ALTERNATIVA SELECCIONADA.

La alternativa seleccionada se elegirá referente al análisis de las posibles alternativas de solución dichas con anterioridad, la elegida se basará en criterios establecidos para este proyecto que son fundamentales en el desarrollo: Innovación, Beneficio, Factibilidad y Creatividad, esto se va a evaluar con la tabla 1-1. En la tabla 1-2 se verá el resultado de los puntajes obtenidos.

Tabla 1-1. Pauta Evaluación Alternativas.

| <b>Tabla de Puntuación de Alternativas de Solución</b> |            |           |       |        |
|--------------------------------------------------------|------------|-----------|-------|--------|
| Muy deficiente                                         | Deficiente | Aceptable | Bueno | Óptimo |
| 1                                                      | 2          | 3         | 4     | 5      |

Fuente: Elaboración propia basada en escala de competencias.

Tabla 1-2. Evaluación de propuestas.

| <b>Característica/Alternativa</b> | <b><u>Alternativa 1</u></b><br><b>Mantener condiciones existentes</b> | <b><u>Alternativa 2</u></b><br><b>Desarrollar una interfaz A través de un microcontrolador</b> | <b><u>Alternativa 3</u></b><br><b>Desarrollar una interfaz SCADA/HMI en WINCC implementado en Factory IO</b> |
|-----------------------------------|-----------------------------------------------------------------------|------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| <b>Innovación</b>                 | 1                                                                     | 2                                                                                              | 5                                                                                                            |
| <b>Beneficio</b>                  | 1                                                                     | 3                                                                                              | 5                                                                                                            |
| <b>Factibilidad</b>               | 3                                                                     | 4                                                                                              | 3                                                                                                            |
| <b>Creatividad</b>                | 1                                                                     | 2                                                                                              | 5                                                                                                            |
| <b>Puntuación total</b>           | 6                                                                     | 12                                                                                             | 18                                                                                                           |

Fuente: Elaboración propia basada en escala de competencias.

La alternativa que obtuvo más puntaje fue la alternativa número 3 obteniendo 20 puntos según los criterios dados anteriormente, esta solución es la más óptima y factible.

### 1.3.3. PROGRAMABLE LOGIC CONTROLLER (PLC)

Un controlador lógico programable o PLC es un dispositivo electrónico utilizado principalmente en la automatización de procesos industriales. Es capaz de reemplazar a otros dispositivos mecánicos de control, como lo son los relés, mediante la “digitalización” de los procesos de control.

Los orígenes del PLC se remontan al año 1968, cuando la división de transmisión automática de la General Motors, buscaba reemplazar los sistemas de transmisión vigentes para esa época, basados en relés, por un sistema electrónico. La solución vino de la compañía “Bedford Associates” y se le llamó MODICON, acrónimo de Modular Digital Controller. Se considera al ingeniero e inventor Richard Dick Morley el padre del PLC, dado que su participación fue crítica en el desarrollo. En la figura 1-4 se puede observar el modelo del PLC a utilizar en el proyecto, que corresponde a la Marca Siemens SIMATIC S7-1200.



Fuente: <https://www.solucionesyservicios.biz/6ES7211-1AE31-0XB0>.

Figura 1-4. PLC Siemens SIMATIC S7-1200 1211C DC/DC/DC.

#### 1.3.3.1. Componentes

Los componentes que hacen funcionar a un PLC se pueden dividir en 3 áreas fundamentales: alimentación y carcasa, unidad central de procesamiento (CPU) y sistema de entradas y salidas (input-output).

#### 1.3.3.2. Alimentación y carcasa

La carcasa es el esqueleto del PLC, donde se acoplan los distintos módulos y donde a su vez se alimenta de energía a estos mediante la conversión, generalmente, de la red eléctrica de 220 V (en el caso de Chile) o 24 VDC.

#### 1.3.3.3. Unidad central de procesamiento (CPU)

La unidad central de procesamiento es el cerebro del PLC, en este módulo se llevan a cabo los procesos de lectura y ejecución del programa. La CPU está compuesta de un microprocesador, un chip de memoria y otros circuitos integrados para la lógica de control, monitoreo y comunicaciones.

Para llevar a cabo la lectura y posterior ejecución del programa, el PLC utiliza lo que se conoce como el ciclo SCAN, tema que se abordará más adelante.

#### 1.3.3.4. Sistema de entradas y salidas (input-output)

El sistema de entradas y salidas (input-output) es el encargado de establecer el vínculo entre la máquina y el PLC. Consta de una serie de conexiones de entrada y de salida que proveen al PLC de una conexión física con el “exterior”. Los dispositivos de entradas o salidas pueden ser digitales o analógicos; los dispositivos digitales son aquellos que son capaces de “leer” señales del tipo encendido o apagado (0 o 1), generalmente este tipo de señal es provista por artefactos tales como botoneras, sensores, interruptores limitadores, etc. Por su parte los dispositivos de entrada analógica son aquellos que son capaces de distinguir entre un rango de valores, que generalmente va entre los

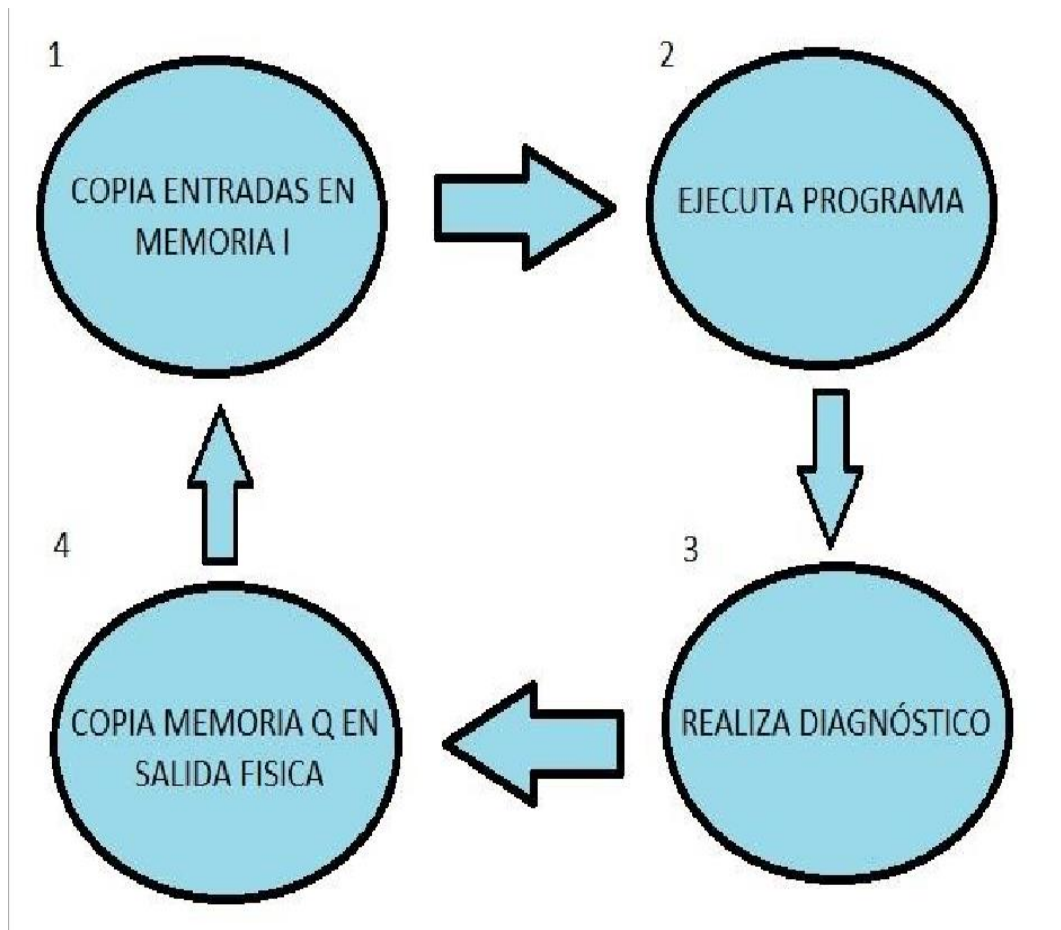
0~10V en el caso de una señal de voltaje o de los 4~20mA en el caso de una señal de corriente. Para que ese tipo de señal sea “entendida” por el PLC, los dispositivos de entrada análoga, llevan a cabo lo que se conoce como proceso de conversión análogo digital, que consiste en la transformación de una señal análoga en una serie de señales digitales comprendidas por el PLC.

#### **1.3.4. EL CICLO SCAN**

El ciclo Scan consiste en 4 pasos iterativos que se mencionan a continuación:

- Primero el PLC copia entradas en memoria I.
- Segundo el PLC ejecuta el programa.
- Tercero el PLC realiza diagnóstico.
- Cuarto el PLC copia memoria Q en salida física, finalizando esto vuelve al primer paso.

En el primer paso el PLC mira las entradas para determinar si están activadas o desactivadas. En el segundo paso ejecutan las operaciones del programa. El tercer paso realiza diagnóstico siguiendo el orden en el que se han grabado. El cuarto paso se dedica a escribir las salidas, después las actualiza de acuerdo a que entradas fueron activadas durante el primer paso, después de esta última etapa se vuelve a la primera fase. En la figura 1-5 se muestra un esquema del ciclo Scan.



Fuente: Elaboración propia, material de clases.

Figura 1-5. Ciclo Scan.

### 1.3.5. HMI

HMI significa Human Machine interface o en español Interfaz Hombre-Máquina es un dispositivo electrónico que permite al usuario comunicarse con la máquina a través del software que tiene según la marca a utilizar.

El HMI cuenta con diferentes características que van a depender y variar según el fabricante, pero lo más común es que sea de pantalla táctil, en algunos casos también con botones y con sus respectivas conexiones de comunicación.

Este dispositivo tiene como finalidad controlar, monitorear, registrar, informar, operar y/o regular los procesos en tiempo real de los elementos que están en funcionamiento dentro de los sistemas automatizados en la industria, mejorando la eficiencia, la seguridad y la productividad en entornos industriales. En la figura 1-6 se muestra HMI Siemens modelo KTP700 utilizado en la simulación de este trabajo.



Fuente: [http://www.ebay.com/itm/SIEMENS-6AV6-647-0AF11-3AX0-SIMATIC-HMI-KTP700-PN-BASIC-TOUCH-PANEL-XCLNT-/292100400907?\\_ul=PE](http://www.ebay.com/itm/SIEMENS-6AV6-647-0AF11-3AX0-SIMATIC-HMI-KTP700-PN-BASIC-TOUCH-PANEL-XCLNT-/292100400907?_ul=PE).

Figura 1-6. HMI KTP700.

### **1.3.6. FACTORY IO**

Factory IO es un programa de simulación industrial que sirve para que el usuario adquiera el aprendizaje de simular y visualizar procesos de fabricación automatizados en tiempo real. Factory IO permite crear escenarios virtuales de procesos industriales, siendo compatible con varios softwares de automatización industrial principalmente PLCs de marcas como Siemens, Allen Bradley, entre otros.

También es compatible con softwares de herramientas de simulación como lo son MATLAB y LabVIEW, permitiendo crear propósitos de entrenamiento, pruebas y desarrollo de sistemas de control industrial.

En la figura 1-7 se muestra la interfaz de simulación de Factory IO.



Fuente: <https://www.google.com/url?sa=i&url=https%3A%2F%2Fdocs.factoryio.com%2F&psig=AOvVaw383TAusm-0Nq82PbS2P2dO&ust=1621048661361000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCJiWpJybyPACFQAAAAAdAAAAABAU>

Figura 1-7. Simulación 3D Factory IO.

#### **1.4. OBJETIVOS DEL PROYECTO**

Los objetivos que se darán a conocer continuación fueron establecidos para cumplir ante la necesidad de la problemática inicial.

##### **1.4.1. Objetivo general**

Desarrollar una interfaz SCADA/HMI en WINCC para el monitoreo de un sistema de control de nivel.

##### **1.4.2. Objetivos específicos**

- Identificar las distintas variables involucradas en el proceso.
- Generar comunicación entre programas de simulación.
- Generar lógica de control PID.
- Generar interfaz HMI.
- Determinar los costos del proyecto.

## **CAPÍTULO 2. DESARROLLO DE LA SOLUCIÓN**

## **2. DESARROLLO DE LA SOLUCIÓN**

El control de nivel ocupa un lugar importante en la industria de alimentos y de la industria química es por esto se realizan en gran cantidad tesis o memorias enfocados en este tema, el control de nivel es fundamental para garantizar la eficiencia, seguridad y la calidad en los procesos industriales. En este trabajo se dará a conocer el desarrollo del programa y las configuraciones a realizar para establecer una interfaz SCADA/HMI en WINCC para el monitoreo de un sistema de control de nivel para finalmente realizar una simulación real en el programa Factory IO, dando solución a la problemática inicial.

### **2.1. COMUNICACIÓN ENTRE PROGRAMAS**

Se dará a conocer el método de comunicación entre el programa Factory IO y TIA PORTAL en dos pasos.

#### **2.1.1. Descarga de programa a trabajar**

El primer paso que se debe tomar, es descargar de la página <https://docs.factoryio.com/tutorials/siemens/setting-up-s7-plcsim-v13/> una planilla de TIA PORTAL según la versión instalada en el PC, la versión que se está utilizando en este caso es la V14, esta plantilla permite la comunicación entre el programa Factory IO Y TIA PORTAL. En la figura 2-1 se muestra las diferentes opciones de descarga según versión.

## Proyectos de plantillas del portal TIA

Tenga en cuenta que debe utilizar un proyecto de plantilla de TIA Portal al conectarse a S7-PLCSIM V13-18. **De lo contrario, Factory I/O no podrá comunicarse con S7-PLCSIM.**

Descargas para TIA Portal V13

Plantilla FactoryIO S7-1200  
 V13.zip Plantilla FactoryIO S7-1500 V13.zip

Descargas para TIA Portal V14

Plantilla FactoryIO S7-1200  
 V14.zip Plantilla FactoryIO S7-1500 V14.zip

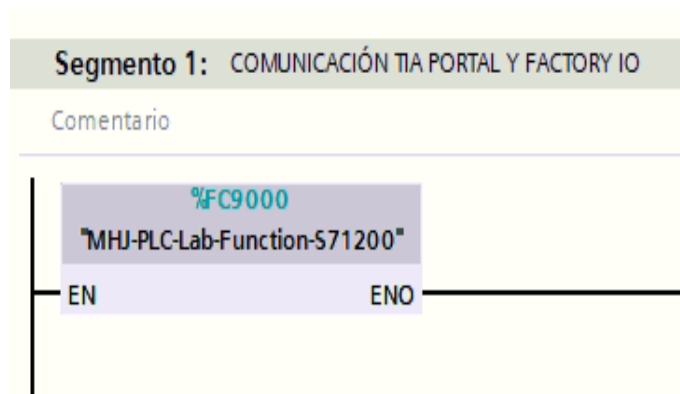
Descargas para TIA Portal V15 (compatible con V16/17/18)

Plantilla FactoryIO S7-1200  
 V15.zip Plantilla FactoryIO S7-1500 V15.zip

Fuente: <https://docs.factoryio.com/tutorials/siemens/setting-up-s7-plcsim-v13/>

Figura 2-1. Planillas a descargar según versión de TIA PORTAL.

Una vez descargada la planilla de TIA PORTAL se inicia el programa para dar inicio a la simulación del PLC virtual, apareciendo ya inserto el bloque de función que permite la comunicación con el software FACTORY IO. En la figura 2-2 se muestra el bloque de función inserto en el programa descargado.

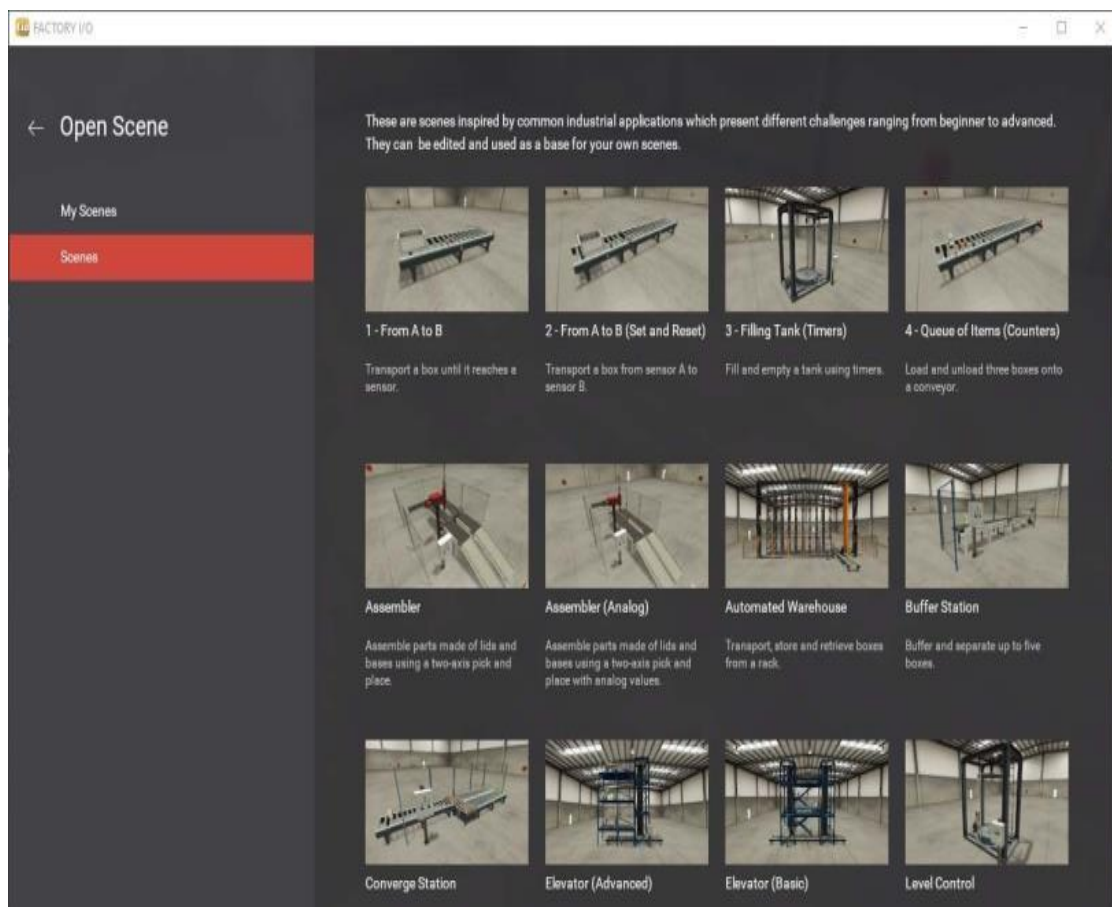


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-2. Bloque de comunicación entre TIA PORTAL y Factory IO.

### 2.1.2. Elección de escena

Al iniciar el programa se deberá elegir el escenario que el usuario quiera utilizar, en el caso de este proyecto se escogió Level Control, que tiene como objetivo regular el nivel de líquido o flujo de un tanque usando un controlador proporcional-integral-derivado (PID). En la figura 2-3 se muestra la elección de escenas a utilizar en el programa Factory IO.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-3. Escenas en Factory IO.

### 2.1.3. Configuración en Factory IO

Lo segundo es configurar en el programa Factory IO. Se asigna el PLC a utilizar definiendo entradas y salidas analógicas y digitales y también los tipos de datos numérico, que en este caso se utilizó dato binario de tipo WORD. En la figura 2-4 se muestra la elección de entradas y salidas y además de la asignación del PLC a utilizar.

The screenshot shows the 'CONFIGURATION' window in Factory IO. On the left is a list of hardware components, and on the right are configuration settings for the selected PLC.

**Left Panel (CONFIGURATION):**

- Advantech USB 4704 & USB 4750
- Allen-Bradley Logix5000
- Allen-Bradley Micro800
- Allen-Bradley MicroLogix
- Allen-Bradley SLC 5/05
- Automgen Server
- Control I/O
- MHJ
- Modbus TCP/IP Client
- Modbus TCP/IP Server
- OPC Client Data Access
- Siemens LOGO!
- Siemens S7-200/300/400
- Siemens S7-1200/1500

**Right Panel (PLC Configuration):**

**PLC**

☐ Auto connect

Model: S7-1200 (V13/V14)

**I/O Config**

Numerical Data Type: WORD

**I/O Points**

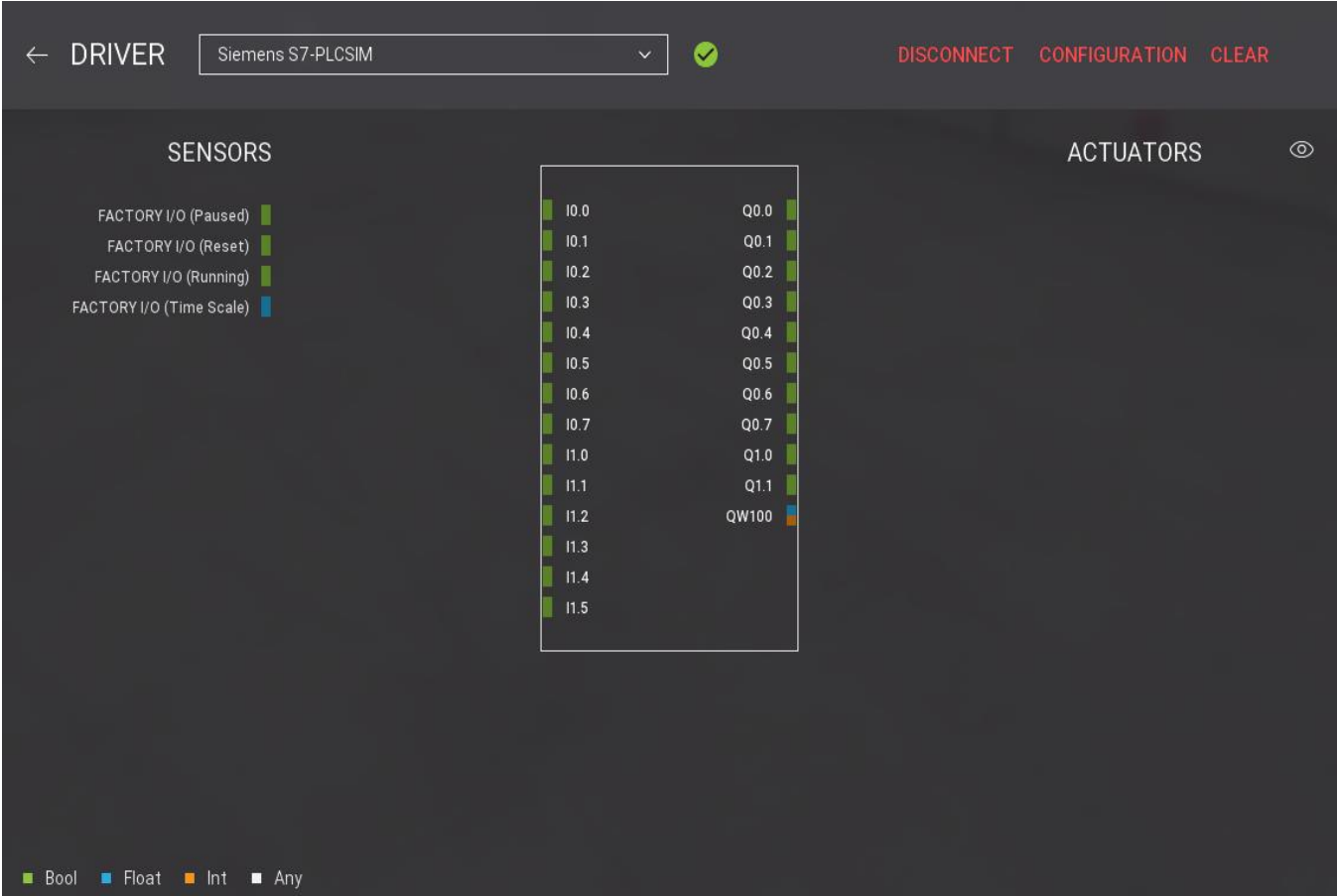
|              | Offset | Count |
|--------------|--------|-------|
| Bool Inputs  | 20     | 4     |
| Bool Outputs | 0      | 4     |
| WORD Inputs  | 30     | 4     |
| WORD Outputs | 30     | 4     |

DEFAULT

Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-4. Configuración en programa Factory IO primera parte.

Y finalmente se da a conectar y la comunicación se realiza entre los dos programas como se muestra en la figura 2-5. Para luego empezar con la programación en el programa TIA PORTAL, y empezar la simulación de lo propuesto en este proyecto.

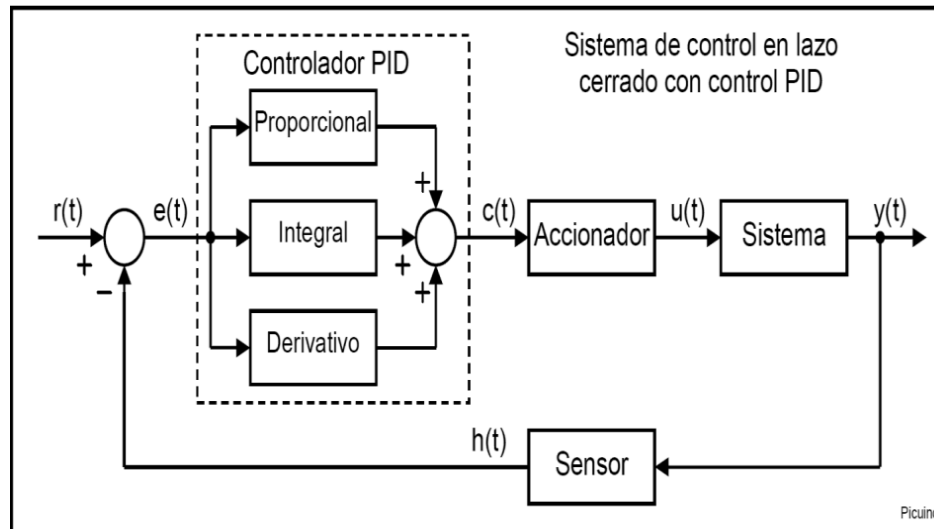


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-5. Configuración en programa Factory IO segunda parte.

## 2.2. ESQUEMA DE CONTROL.

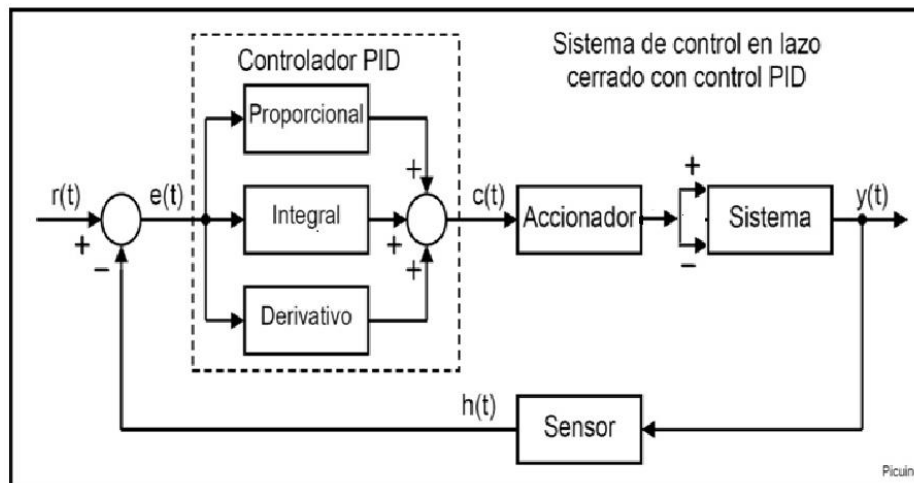
Para el diseño del programa se utiliza el siguiente esquema de la figura 2-6, que es un sistema de control en lazo cerrado con control PID que va a ser la estrategia de control a utilizar:



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-6. Esquema de control.

Pero en el caso de este proyecto se va a necesitar controlar tanto la válvula de carga y la válvula de descarga, para poder tener control de las dos válvulas, se ocupará el sistema mostrado en la figura 2-6, pero se modificará la salida como se muestra en la figura 2-7 quedando un esquema de rango partido, controlando dos variable representadas por el símbolo + Y -



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-7. Esquema de control de rango partidor.

### 2.3. CONTROLADOR PID

El controlador PID es un mecanismo de control de lazo cerrado que permite corregir un sistema de automatización a través de la retroalimentación de un sensor, que entrega el valor real medido y lo compara con la entrada o valor deseado (set point) dando como resultado el error, en este punto entra en funcionamiento el PID que va a tratar de eliminar el error con 3 acciones que son la Proporcional, Integral y Derivativa para que el valor real se iguale con el set point con la máxima rapidez y estabilidad.

La acción proporcional como su nombre lo dice es proporcional al error, lo que hace es adicionar una ganancia  $K_p$ , y así aumentar la velocidad del sistema, pero sin embargo no elimina el error.

La acción integral toma los errores pasados, los conserva en el tiempo y les suma una constante  $k_i$  que elimina el error de estado estacionario; la desventaja es que puede generar oscilaciones en el set point.

La acción derivativa permite anticiparse a los errores futuros calculando el comportamiento de los errores presentes, esto hace que disminuyan los impulsos y oscilaciones del sistema.

Para este proyecto se utilizará la instrucción PID\_Compact que funciona como controlador PID que va a calcular los parámetros P, I y D para el sistema de forma automática para su optimización inicial. Los parámetros pueden optimizarse aún más a través de una optimización fina que ofrece PID\_Compact, como también ingresar los parámetros  $K_p$ ,  $K_i$  y  $K_d$  de modo manual.

## 2.4. DISEÑO DEL PROGRAMA

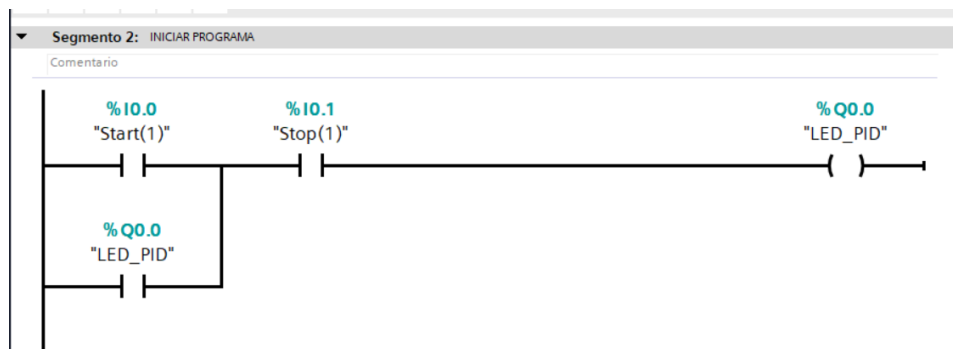
Este diseño hace énfasis a un control PID que se realizará para optimizar el proceso de llenado y vaciado del estanque, que este se controlará con una válvula de llenado y una válvula de descarga, además de un sensor de nivel, todo lo mencionado anteriormente se implementará en el programa Factory IO. Para el programa se utilizaron las funciones normalizar y escalar para poder ser legibles las señales de la válvula y del sensor de nivel y un bloque de organización que son fundamentales para la programación de la interfaz, además en el programa Factory IO se utilizará la escena de estanque de nivel para este desarrollo.

### 2.4.1. Primer Segmento

En el segmento 1 se encuentra el bloque de comunicación que permite la conexión del PLC virtual al software Factory IO como se muestra en la figura 2-2.

### 2.4.2. Segundo Segmento

En el segmento 2 se realiza la iniciación del programa, dándole partida al controlador PID. En la figura 2-8 se muestra el segmento 2 del programa.

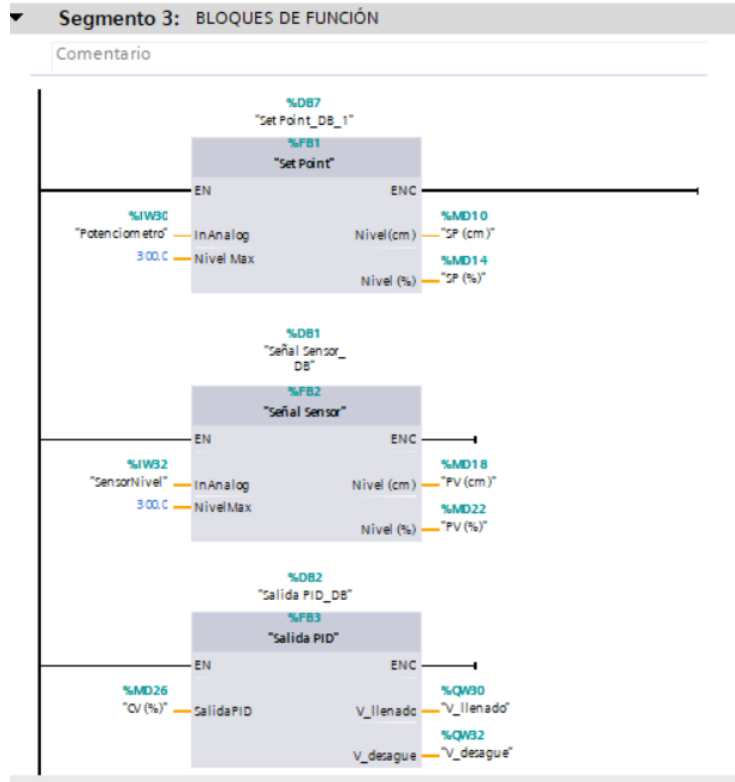


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-8. Programa realizado en el Segundo segmento.

### 2.4.3. Tercer segmento

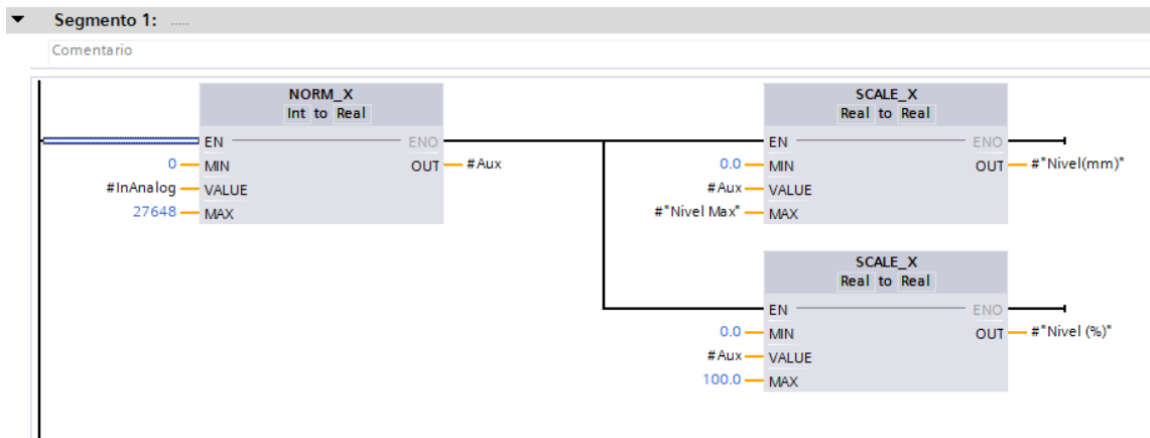
En el segmento 3 se insertó al Main (bloque de programa principal), tres bloques de función, cada uno cumpliendo el propósito de visualizar y controlar las variables a utilizar. En la figura 2-9 se muestra el segmento 3 del programa.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-9. Programa realizado en el tercer segmento.

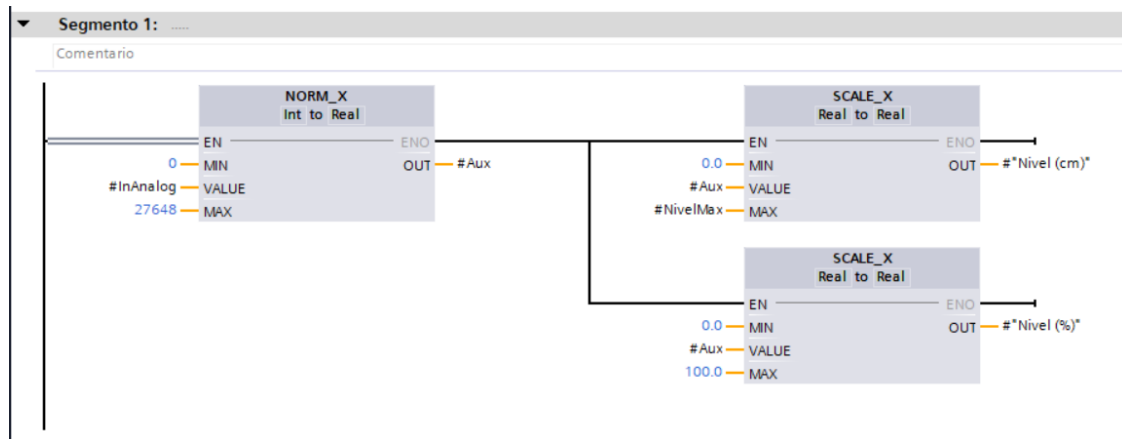
El primer bloque de función tiene por nombre "Set Point", donde se normalizó el dato proveniente del factory IO de tipo entero a real y se realizó un escalamiento para que el dato fuera entendible al valor real del sistema, quedando el SP en centímetros y porcentaje. En la figura 2-10 se visualiza el programa del bloque de función "Set Point".



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-10. Programa del bloque de función “Set Point”.

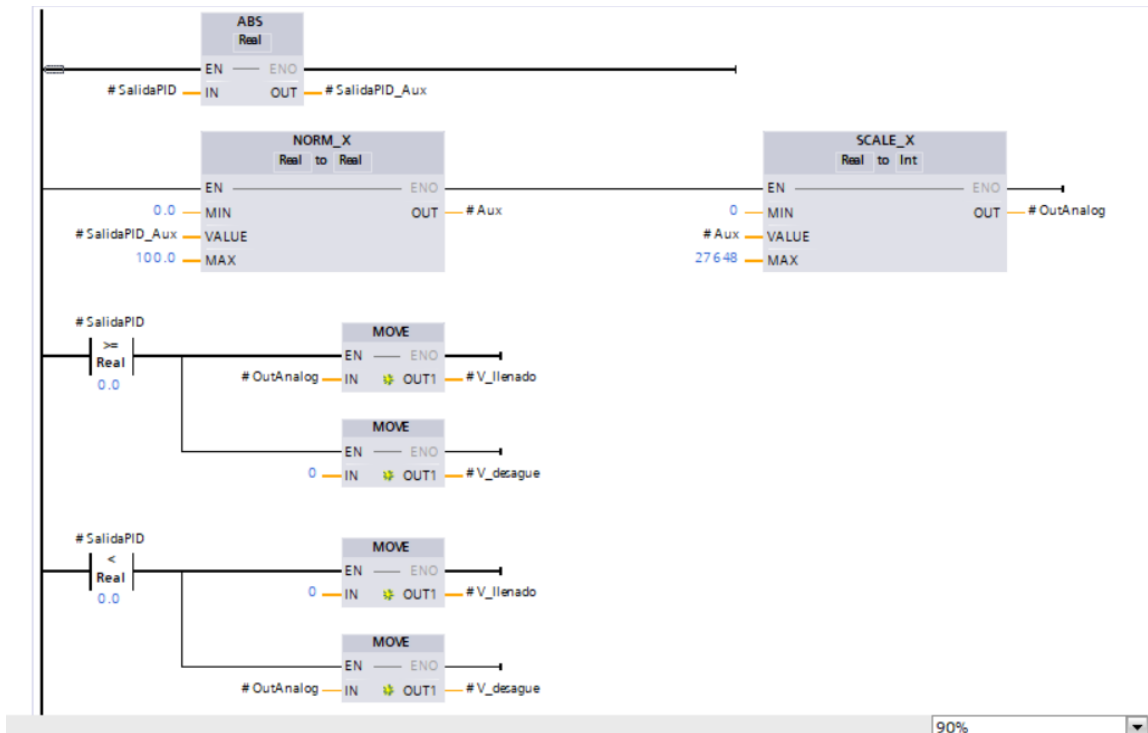
El segundo bloque de función tiene por nombre “Señal Sensor”, donde se normalizó el dato proveniente del factory IO de tipo entero a real y se realizó un escalamiento para que el dato fuera entendible al valor real del sistema quedando el PV en centímetro y porcentaje. En la figura 2-11 se visualiza el programa del bloque de función “Señal Sensor”.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-11. Programa del bloque de función “Señal Sensor”.

El tercer bloque de función tiene por nombre “Salida PID”, donde la salida del PID independiente del valor del error tanto negativo como positivo lo necesitamos como un valor absoluto, que para eso se aplicó la función matemática de valor absoluto. Luego se normalizó el dato proveniente de la función valor absoluto de tipo real a real y se realizó un escalamiento para que el dato lo leyera el PLC virtual y dando la activación de la salida hacia las válvulas, pero para distinguir cuál de las dos válvulas abrir o cerrar, se realiza una comparación entre si el valor de la salida del PID es positivo se abre la válvula de llenado y si es negativa se abre la válvula de desagüe. En la figura 2-12 se visualiza el programa del bloque de función “Salida PID”.

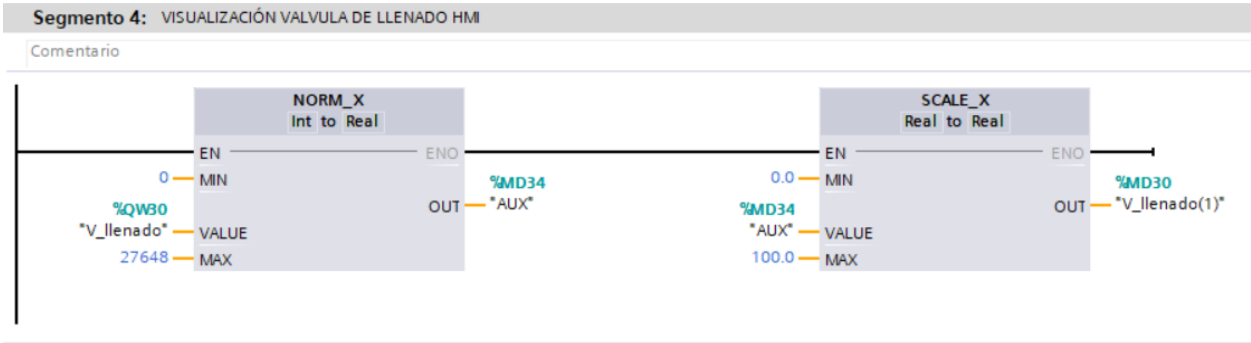


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-12. Programa del bloque de función “Salida PID”.

2.4.4. Cuarto segmento.

En el cuarto segmento se realizó una normalización y un escalamiento para poder visualizar el porcentaje de la apertura de la válvula de llenado para luego ocuparla en la interfaz HMI. En la figura 2-13 se muestra el proceso realizado.

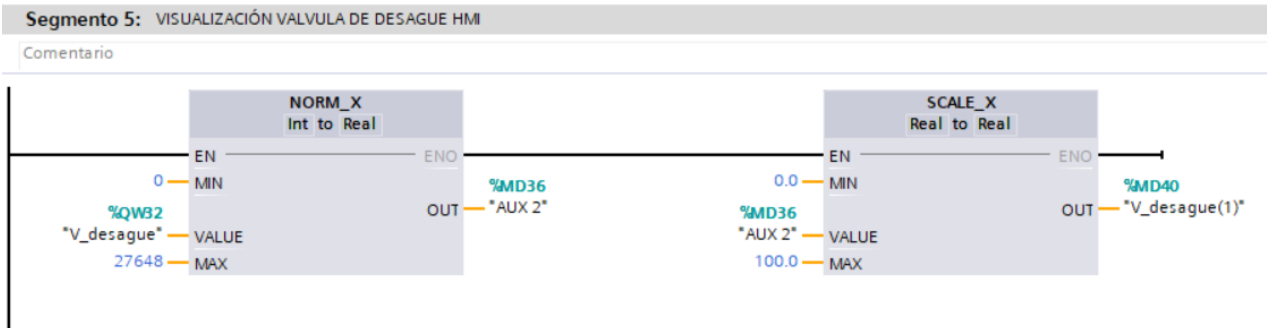


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-13. Programa realizado en el cuarto segmento.

2.4.5. Quinto segmento.

En el Quinto segmento se realizó una normalización y un escalamiento para poder visualizar el porcentaje de la apertura de la válvula de desagüe para luego ocuparla en la interfaz HMI. En la figura 2-14 se muestra el proceso realizado.

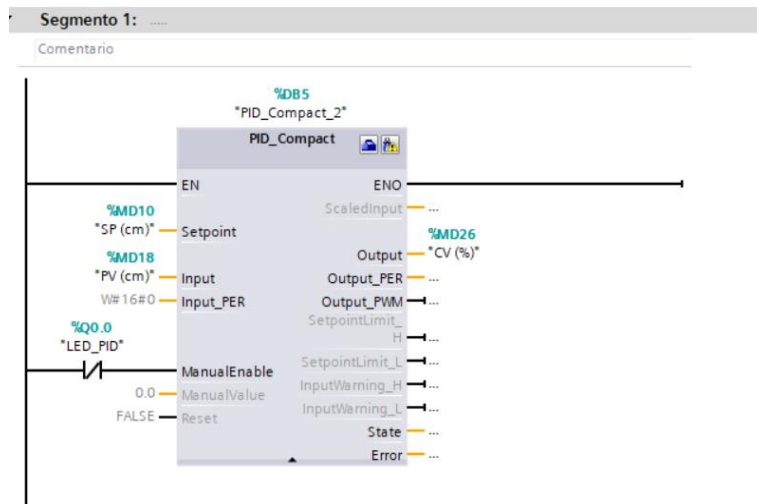


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-14. Programa realizado en el quinto segmento.

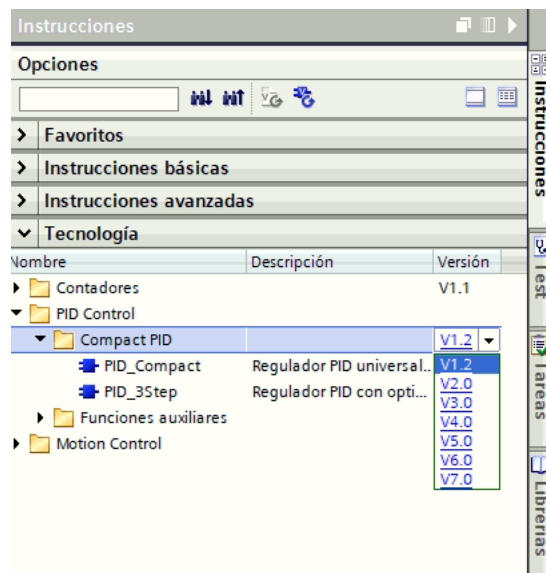
#### 2.4.6. Bloque de organización

El bloque de organización que se escogió es el interruptor cíclico, ya que funciona independiente del ciclo del programa principal, permitiendo una acción más rápida para los cambios físicos como temperatura, nivel, etc. Es por eso que se insertó el PID para configurar parámetros PID. La versión del PID\_Compact debe ser V1.2 ya que no funcionará si es con otra versión en el caso de simuladores virtuales como el PLC SIM. En la figura 2-15 se muestra la instrucción PID compact y en la figura 2-16 la versión a utilizar del PID\_Compact.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-15. Programa realizado en el bloque de organización.

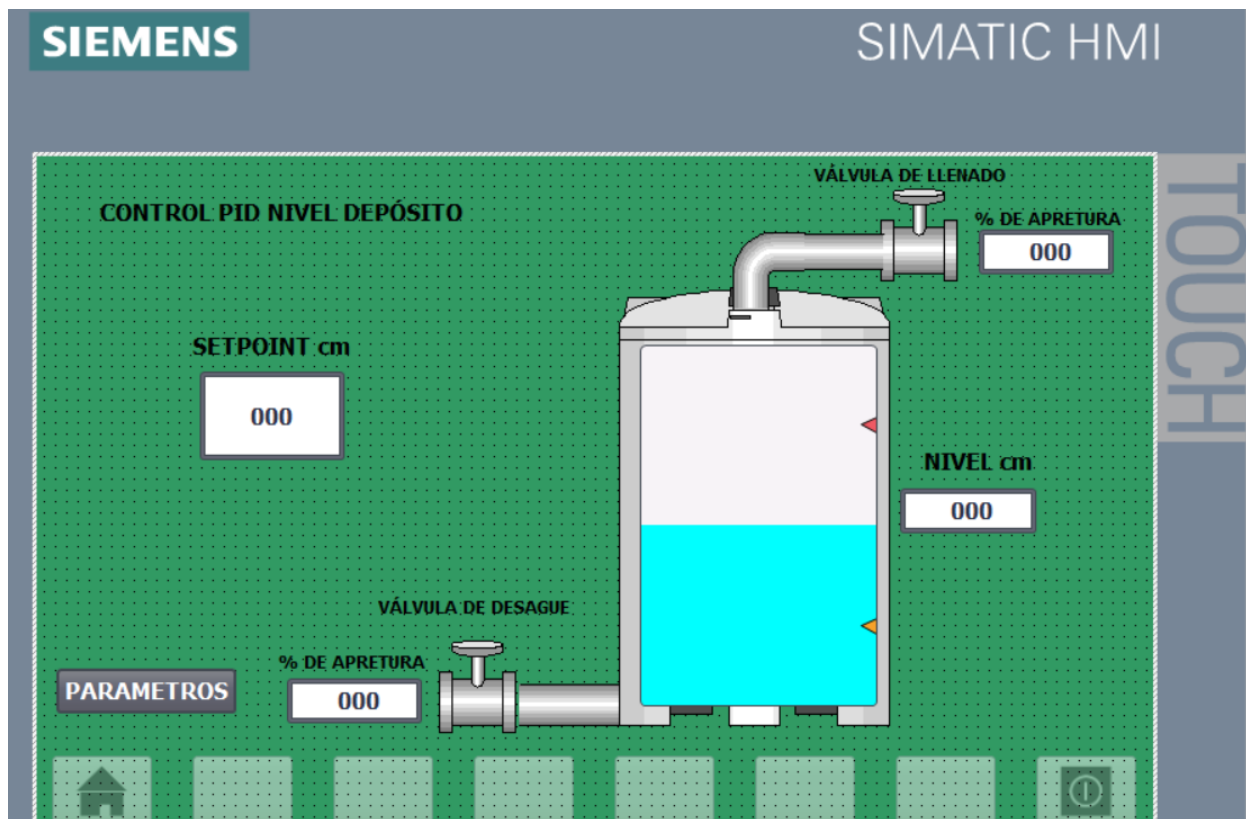


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-16. Versión a utilizar del PID\_Compact.

## 2.5. DISEÑO DE LA INTERFAZ HMI

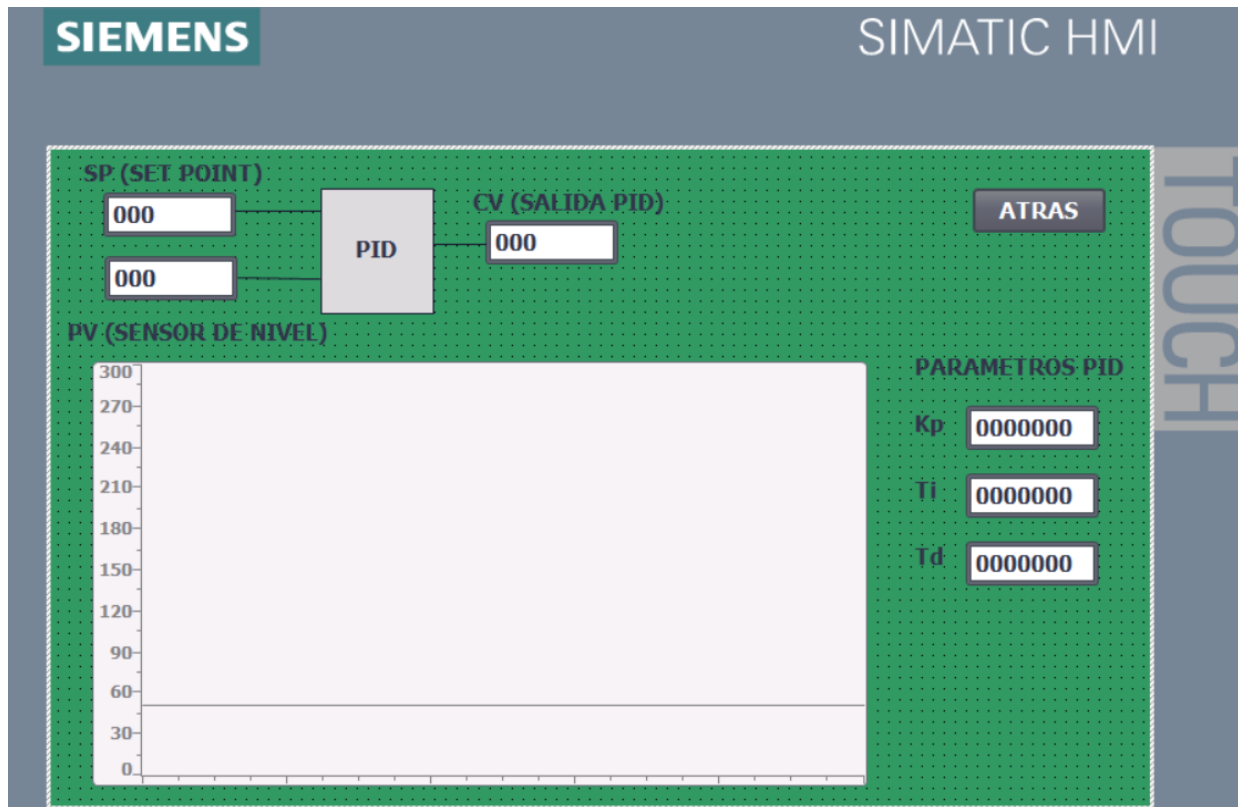
Para el diseño de la interfaz HMI en la figura 2-17 se insertó un display para ingresar el setpoint deseado del nivel del estanque, además de poder visualizar el porcentaje de la apertura de la válvula de llenado y desagüe, controlada por la salida del PID\_Compact, y la visualización del nivel en cm en un estanque simulado.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-17. Interfaz HMI.

En la segunda figura 2-18 se visualiza un gráfico en tiempo real del proceso donde se ve el setpoint y la señal del sensor de nivel escalada, además de visualizar los parámetros del PID para observar los valores finales optimizados del sistema.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 2-18. Interfaz HMI.

### **CAPÍTULO 3. FUNCIONAMIENTO Y COSTOS DEL PROYECTO**

### 3. FUNCIONAMIENTO Y COSTOS DEL PROYECTO

En este capítulo se hablará sobre el funcionamiento del programa diseñado en el TIA PORTAL e implementado en Factory IO, además de dar a conocer los costos del proyecto.

#### 3.1. FUNCIONAMIENTO DEL PROGRAMA

Para comenzar se establece si el sistema es lineal o no lineal, para eso se aplicarán dos principios el de superposición y de homogeneidad.

Superposición:

- Se establece un punto de operación, que en este caso sería 2 volts de la electroválvula, luego se ve cuanto se llena el estanque y se anotan los valores
- Se establece un segundo punto de operación que sería 3 volts y se realiza lo mismo del punto anterior, para luego realizar la aditivita de este principio

2 volts dan 55 cm

6 volts dan 190 cm

Entonces sería:

Punto de operación 1 + Punto de operación 2 dan Nivel del tanque en cm 1 + Nivel del tanque en cm 2

Reemplazando:

2 volts + 6 volts dan 55 cm + 190 cm  8 volts dan 245 cm

Pero al llevarlo a la práctica los 8 volts dan 260 cm, entonces quiere decir que el sistema no es lineal.

Homogeneidad: se establecen los mismos puntos de operación anteriores, pero se aplica de la siguiente forma:

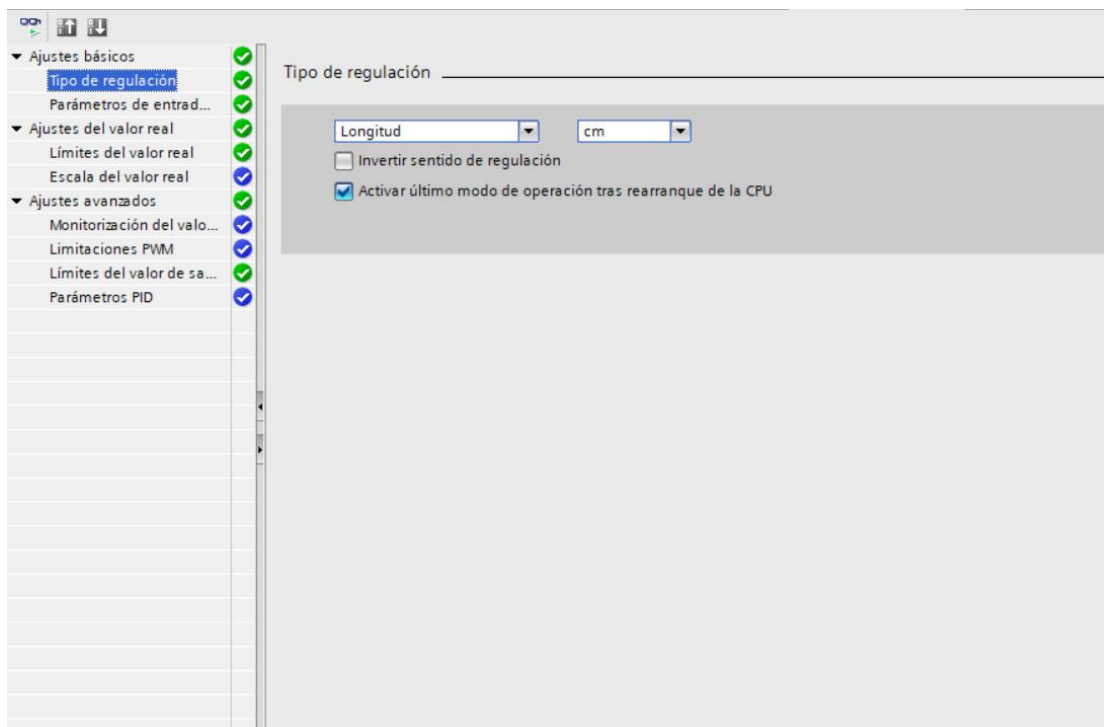
2 volts x 2 = 6 volts

55 cm x 2 = 110 cm

Pero los 6 volts daban 190 cm y no 110 cm, entonces el sistema no es lineal.

Para comenzar con el funcionamiento se realizarán algunos ajustes al bloque PID\_Compact\_2 que son los siguientes:

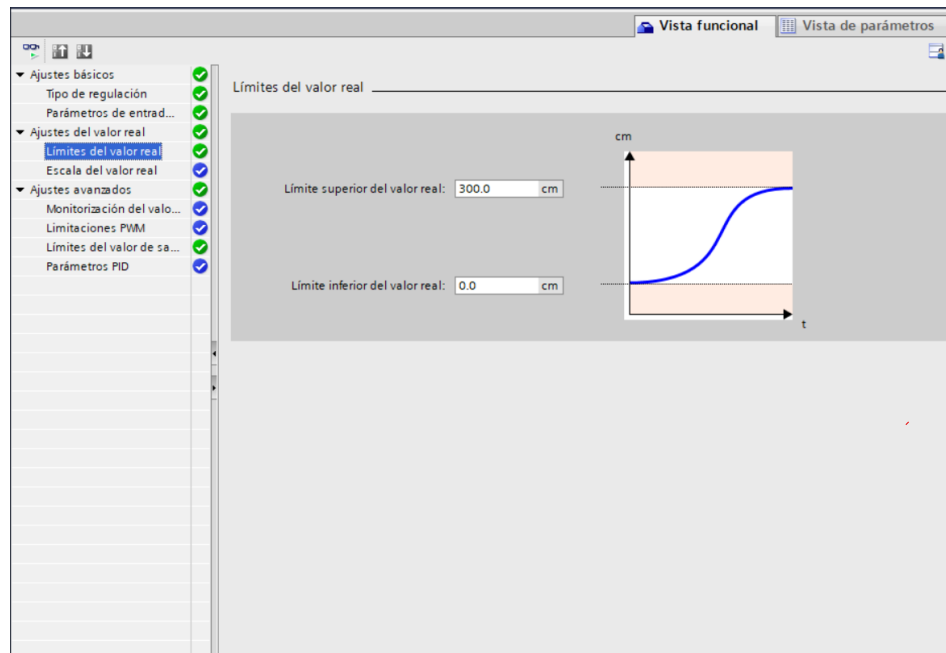
Lo primero son los ajustes básicos para asignar las variables que se realizó para este sistema y también se podrá configurar el tipo de variable que se está manipulando, en este caso se realiza una regulación de tipo longitud en cm ya que se trabajó la altura como se mide en Factory IO, pero también se podría en otros casos asignar intensidad, flujo, presión, nivel, etc. En la figura 3-1. Se muestra el menú de ajustes.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-1. Ajustes básicos.

Segundo son los ajustes de Valor Real, sirven para ajustar y escalar la variable de entrada sobre el tipo de regulación que se está haciendo en este caso es de 0 a 300, esto quiere decir que podemos elegir entre el 0 y 300 como punto de operación deseado. En la figura 3-2. Se muestra los ajustes del valor real.

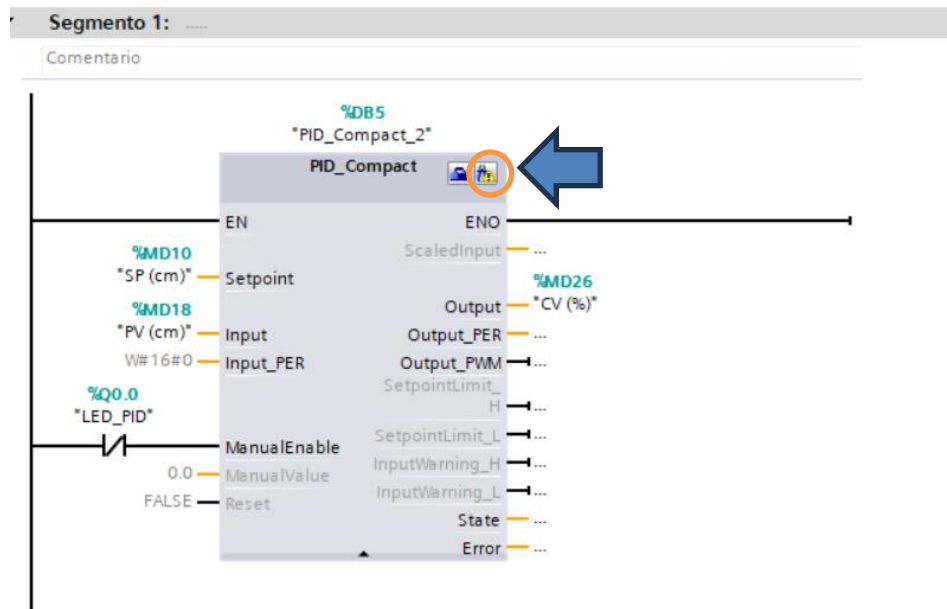


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-2. Ajustes del valor real.

Una vez realizado lo anterior se presiona Start en la ventana puesta en servicio y empieza a correr el PID compact y además el sensor comienza a medir el nivel del estanque. Una vez insertando el valor en el Setpoint, el PID compact comienza a diseñar los parámetros PID para mantener la variable de proceso lo más cerca posible del valor deseado, pero esto lo hace con oscilaciones, esto quiere decir que el sistema no está optimizado y nunca se mantendrá el valor deseado exacto.

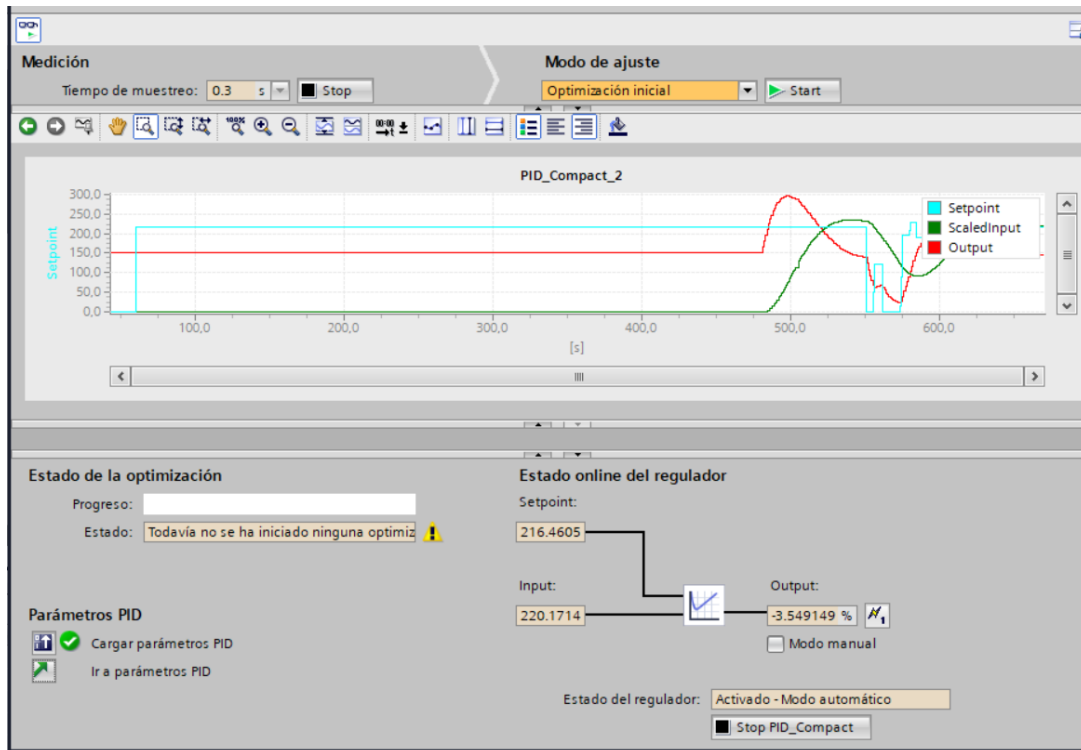
Para poder llegar al valor deseado sin oscilaciones se tiene que iniciar el proceso de optimización inicial, el método que utiliza la optimización inicial el de Ziegler and Nichols ya determinar el comportamiento del proceso a un escalón del valor de salida y busca el punto de inflexión para realizarlo, para activarlo se tiene que ir al PID compact y seleccionar ventana puesta en servicio como se muestra en la figura 3-3.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-3. PID compact.

Una vez realizado lo anterior se abrirá la ventana de puesta en servicio donde muestra la entrada (PV o variable de proceso), salida (CV) y setpoint (SP o punto de operación) del sistema, en esta ventana se podrá observar el comportamiento del sistema además de los ajustes para optimizar el sistema. En la figura 3-4. Se muestra la ventana puesta en servicio.

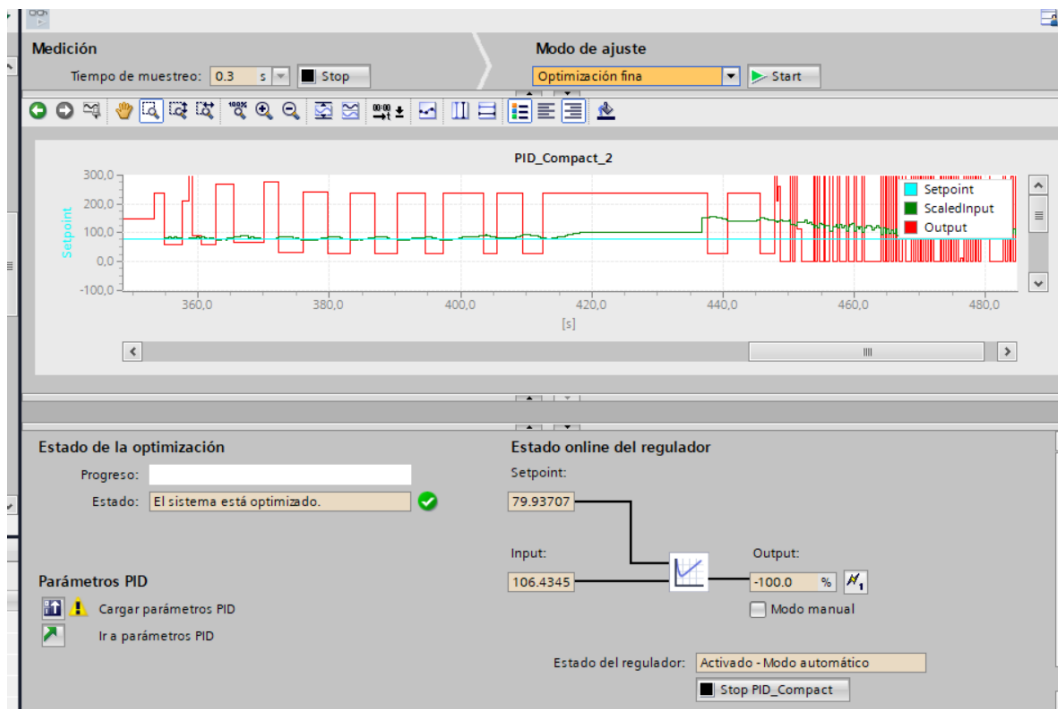


Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-4. Ventana puesta de servicio.

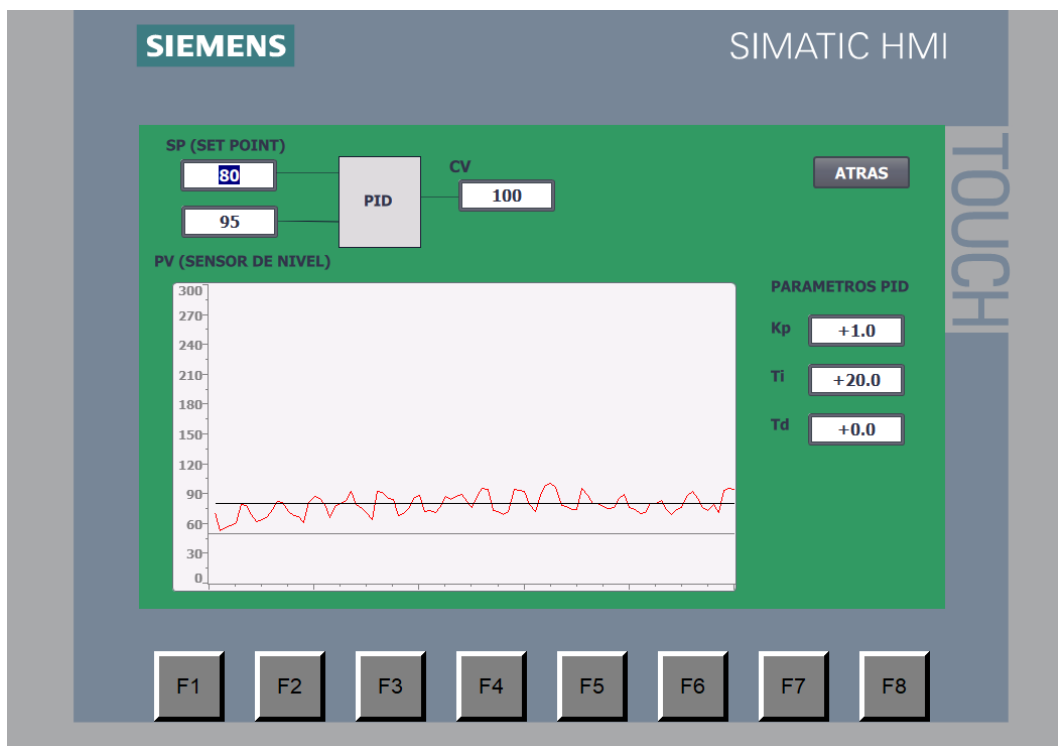
Una vez abierta la ventana puesta de servicio se da clic en iniciar optimización inicial y lo que hará es facilitar y acelerar el proceso de la configuración de los parámetros PID, asegurando un sistema optimo desde el principio para el Setpoint deseado, que para este proyecto se seleccionó 80 cm, dando el valor más cercano que pueda, una vez terminado el proceso de optimización inicial, se puede observar si el PID\_Compact dio los valores de las variables PID adecuados para el sistema, si el sistema no se estabiliza se da a iniciar al proceso de optimización fina que utiliza el método de Ziegler and Nichols, ya que los parámetros PID se optimizan para el punto de operación a partir de la amplitud y la frecuencia, para llegar al valor deseado de forma uniforme y lineal sin oscilaciones y más precisa tratando de compensar perturbaciones del sistema y referencia en el caso que se requiera cambiar el punto de operación. En esta ocasión no se realizó la optimización inicial ya que sale el siguiente error: "Error al iniciar la optimización inicial. El valor real está demasiado cerca de la consigna. Inicie la optimización fina." Es por esto que se realizó directamente la optimización fina.

En la figura 3-5 se muestra el proceso de optimización fina y en la figura 3-6 se muestra el mismo proceso, pero en la interfaz HMI.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

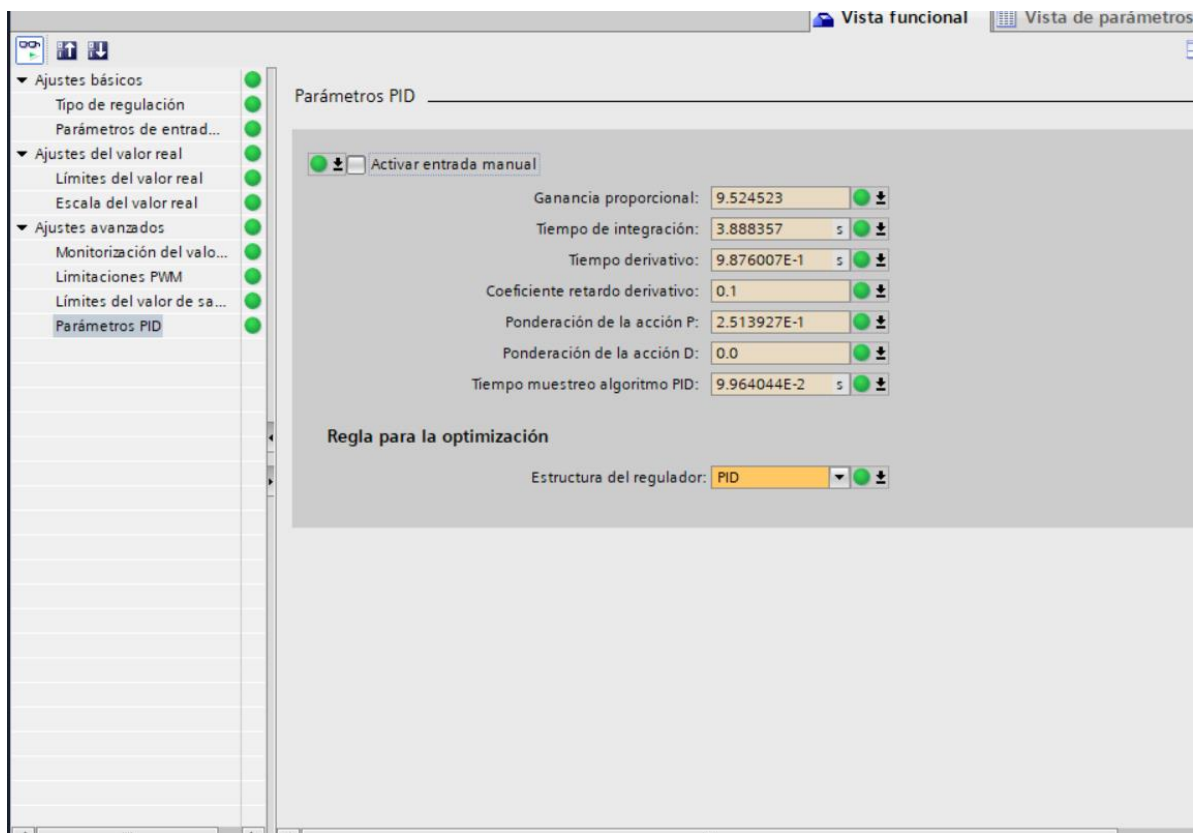
Figura 3-5. Ventana puesta de servicio con la optimización fina en curso.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

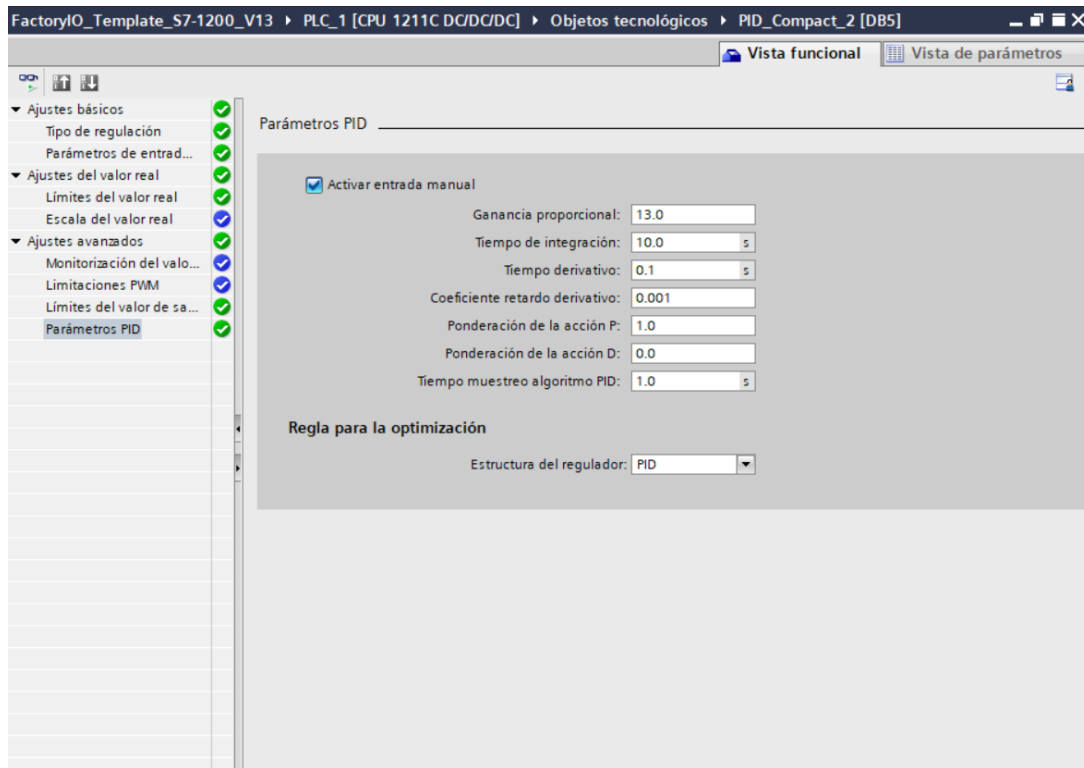
Figura 3-6. HMI con la optimización fina en curso.

Obtenido los resultados de la optimización fina da cuenta que el PID\_Compact dio los parámetros PID nuevamente pero aun así no son los que se necesitan para que sistema funcione los más estable posible ya que presenta muchas oscilaciones en la salida del PID, esto no permite que el valor de la variable de proceso llegue al punto de operación deseado. Por lo anterior se realizará cambios de los parámetros PID de forma manual, aumentando el tiempo de integración para eliminar el error en estado estacionario provocado por perturbaciones, y el tiempo derivativo se aumentará para conseguir la menor oscilación posible en el punto de operación. En la figura 3-7 se muestran los parámetros PID después de la optimización fina y en la figura 3-8 se muestra el cambio manual de los parámetros PID.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

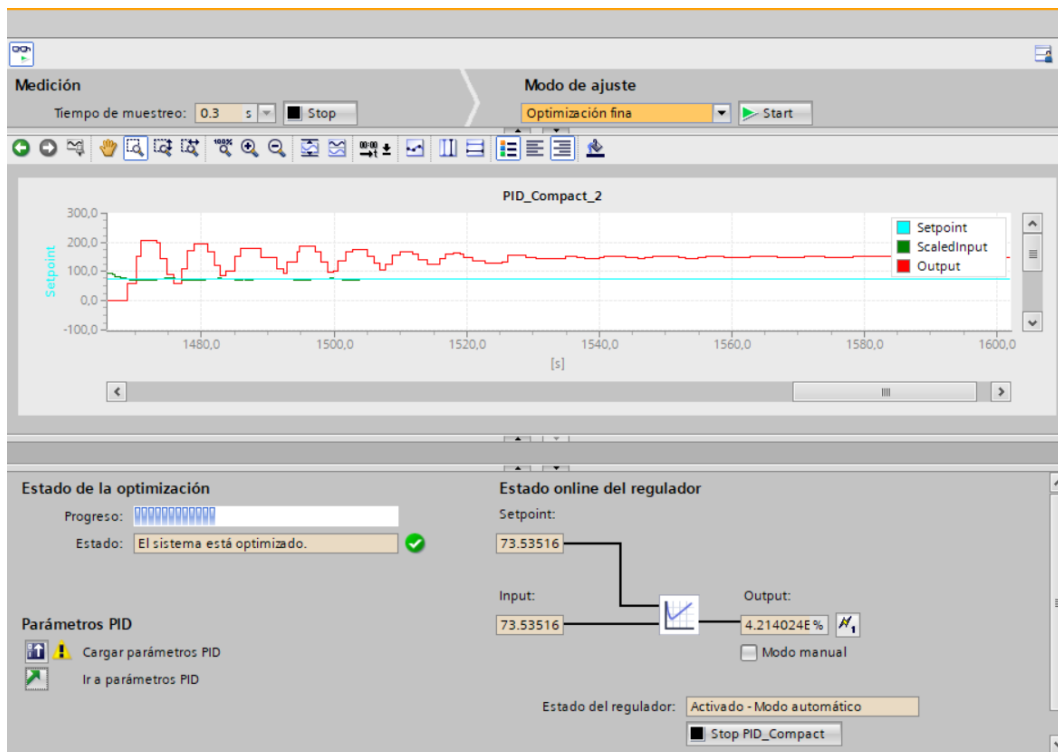
Figura 3-7. Parámetros PID después de optimización fina.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

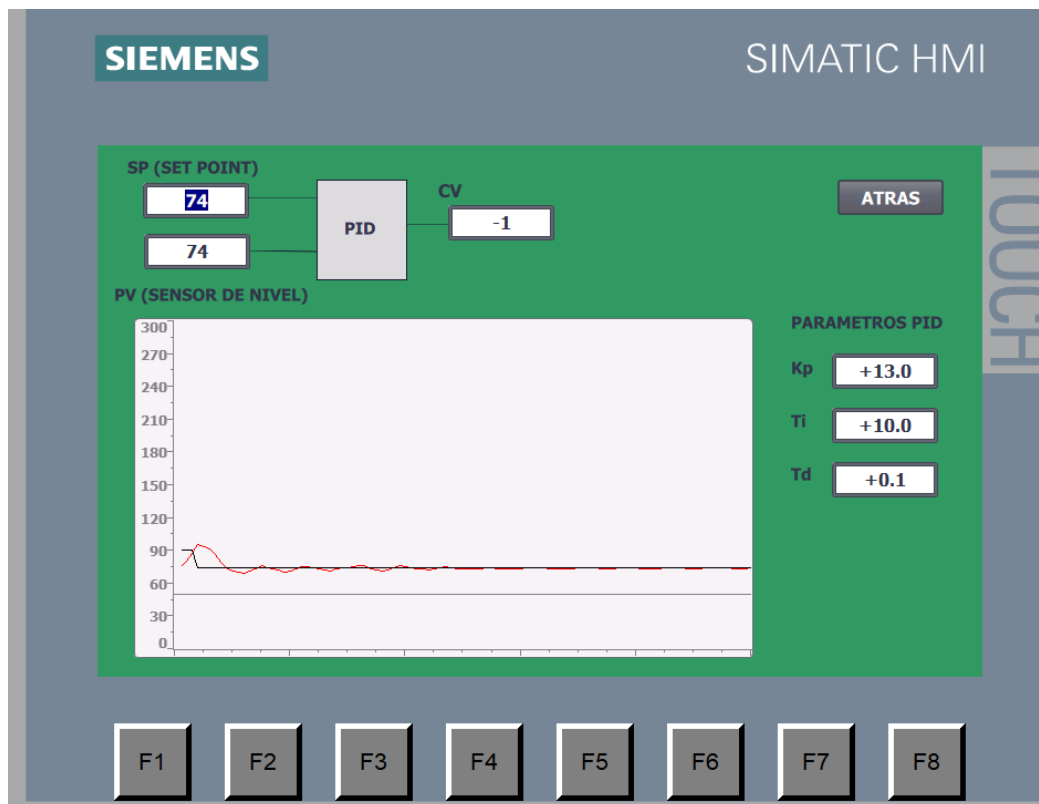
Figura 3-8. Cambio manual de los parámetros PID.

Una vez terminada la configuración de parámetros se pone en funcionamiento el PID\_Compact nuevamente para ver el comportamiento del sistema bajando el punto de operación a 74 cm para visualizar mejor el cambio. En la figura 3-9 se muestra los resultados de la ventana puesta en servicio y en la figura 3-10 los resultados de la interfaz HMI.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-9. Resultados de la ventana puesta en servicio.



Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-10. Resultados de la pantalla HMI.

Ya obteniendo los resultados finales del PID\_Compact como se puede ver en la figura 3-9 y figura 3-10, se observa que se logra optimizar el sistema consiguiendo que la variable de proceso llegue en la parte estacionaria al punto de operación deseado sin perturbaciones en las válvulas de llenado y desagüe, se cumplió con el objetivo para el que fue diseñado el PID para la variable de proceso deseada.

### 3.2. COSTOS DEL PROYECTO

Los valores y cálculos que se realizaron se basaron en la implementación de manera física en el laboratorio de control y automatización de la Universidad Técnica Federico Santa María en la sede de Viña del Mar.

El valor de las horas hombres (HH) calculadas según el Anexo A se establecieron respecto a lo que gana mensualmente en la actualidad un Ing. Recién titulado dentro de esta zona (V región), y un técnico que realizará labores de montaje, las HH varían según si es montajista o diseñador del proyecto.

Los gastos generales son costos que tiene las empresas de servicios básicos, transportes y gestiones, en este caso del proyecto es energía, agua, retiro de material, gestiones de pagos a empleadores, etc. Los valores de los materiales cotizados son valores actuales sacados de tiendas reconocidas en el mercado.

| DESCRIPCIÓN                                                                                | TIPO DE UNIDAD | CANTIDAD | PRECIO NETO | TOTAL NETO          |
|--------------------------------------------------------------------------------------------|----------------|----------|-------------|---------------------|
| <b>MATERIALES</b>                                                                          |                |          |             |                     |
| PLC SIEMENS S7-1200 1214 DC/DC/DC                                                          | C/U            | 1        | \$ 375.089  | \$ 375.089          |
| HMI SIEMENS KTP700 BASIC                                                                   | C/U            | 1        | \$ 401.181  | \$ 401.181          |
| SWITCH ETHERNET SCALANCE XB005                                                             | C/U            | 1        | \$ 128.000  | \$ 128.000          |
| FUENTE DE ALIMENTACIÓN 6EP1332-5BAC0 SIEMENS                                               | C/U            | 1        | \$ 68.000   | \$ 68.000           |
| Válvula Solenoide 3/4 Pulgada Rain C/c Flujo 24vac                                         | C/U            | 2        | \$ 14.660   | \$ 29.320           |
| Sensor De Nivel De Agua Transmisor De Nivel Liquido 0-5m                                   | C/U            | 1        | \$ 58.749   | \$ 58.749           |
| ESTANQUE AQUATANK 1000LT                                                                   | C/U            | 1        | \$ 106.408  | \$ 106.408          |
| SOPORTE ESTANQUE (ESTRUCTURAL)                                                             | C/U            | 1        | \$ 80.000   | \$ 80.000           |
| TIA Portal contiene certificado de licencia para TIA Portal Teamcenter Gateway V14 SP1, SW | C/U            | 1        | \$ 973.446  | \$ 973.446          |
| FACTORY IO LICENCIA POR SIEMPRE                                                            | C/U            | 1        | \$ 359.313  | \$ 359.313          |
| MISCELANEOS                                                                                | GLOBAL (GL)    | 1        | \$ 50.000   | \$ 50.000           |
| <b>TOTAL MATERIALES NETO</b>                                                               |                |          |             | <b>\$ 2.629.506</b> |
| <b>MANO DE OBRA Y UTILIDADES</b>                                                           |                |          |             |                     |
| MO DIRECTA DE IMPLEMENTACIÓN                                                               | HH             | 45       | \$ 3.800    | \$ 171.000          |
| DISEÑO DEL PROGRAMA                                                                        | HH             | 135      | \$ 5.200    | \$ 702.000          |
| INTALACIÓN DE PROGRAMA                                                                     | HH             | 18       | \$ 5.200    | \$ 93.600           |
| GASTOS GENERALES                                                                           | %              | 15%      | \$ 394.426  | \$ 394.426          |
| UTILIDADES                                                                                 | %              | 15%      | \$ 394.426  | \$ 394.426          |
| <b>TOTAL NETO</b>                                                                          |                |          |             | <b>\$ 4.384.958</b> |

Fuente: Extraído de programa realizado para el desarrollo del proyecto de título.

Figura 3-11. Costos del Proyecto.

## **CONCLUSIONES**

El desarrollo de una interfaz SCADA/HMI para el monitoreo de un sistema de control de nivel, ofrece una variedad de ventajas en la industria. La principal es que permite conocer las variables y procesos reales de un sistema, esto lo permite el programa Factory IO, el cual permite construir y controlar procesos industriales en tiempo real, además que el software es totalmente interactivo proporcionando un ambiente realista de una industria.

Gracias a este proyecto se aprendieron diversos aspectos como el uso del programa Factory IO y lo relevante que se puede convertir en la enseñanza del control de la industria, además de poder realizar un control PID a través del programa TIA PORTAL obteniendo resultados exitosos del sistema de nivel optimizado y estable.

Actualmente este proyecto es fundamental para el desarrollo de experiencias de laboratorio de manera virtual, ya que por la pandemia se han tomado diferentes caminos para el retorno a la educación, permitiendo educación semi presencial. Esto ayuda a los estudiantes de Ingeniería de Ejecución en Control e Instrumentación Industrial al aprendizaje de manera remota y poder controlar procesos reales en el mundo laboral.

Todo lo anterior permite poner en práctica lo aprendido durante el largo de la carrera y así poder estar más preparados para las problemáticas que se puedan presentar en trabajos como ingeniero, permitiendo aplicar los conocimientos adquiridos como lo fue en este proyecto. También se puede concluir que durante lo largo de este trabajo se fueron cumpliendo con los objetivos satisfactoriamente de la propuesta inicial de este proyecto.

**BIBLIOGRAFÍA**

Manual Controlador programable S7-1200 [en línea] 2009. S.l.: Siemens. (06/05/2021)

García, D. (2021). Simulacion de PIDcompact en Siemens TIA Portal. InfoPLC. <https://www.infoplcn.net/descargas/107-siemens/software-step7-tiaportal/tia-portal/3240-simulacion-pidcompact-siemens-tia-portal> (05/06/2021)

Programmable logic controller. En.wikipedia.org [en línea] [s.a] (06/05/2021)

Manual regulación PID compact  
[https://cache.industry.siemens.com/dl/files/036/108210036/att\\_916495/v1/s71500\\_pid\\_controller\\_function\\_manual\\_esES-es-ES.pdf](https://cache.industry.siemens.com/dl/files/036/108210036/att_916495/v1/s71500_pid_controller_function_manual_esES-es-ES.pdf) (08/07/2021)