



**UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



**PREFACTIBILIDAD TÉCNICA DE LA IMPLEMENTACIÓN DE  
PROGRAMACIÓN DE PLC EN GREEN I&C**

POR

**Matías Francisco Martínez Vergara**

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción  
para optar al título profesional de Ingeniero Civil Electrónico

Profesor Guía  
Juan Pablo Segovia

Profesional Supervisor  
Jonathan Olivares (Green I&C)  
Oliver Sáez (Green I&C)

Diciembre 2023  
Concepción (Chile)

©2023 Matías Francisco Martínez Vergara

©2023 Matías Francisco Martínez Vergara

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

## **Agradecimientos**

Quisiera expresar mi profundo agradecimiento a todas las personas que me han acompañado a lo largo de mi camino universitario. En primer lugar, a mis padres, cuyo apoyo incondicional ha sido fundamental, sobre todo en los momentos más difíciles de mi vida. Gracias a ellos, he tenido las herramientas necesarias para superar los obstáculos que se presentaron en el camino. A mi hermana y hermano, quienes son una constante inspiración como ingeniera e ingeniero, les estoy sumamente agradecido. También quiero reconocer a mis amigos de la universidad, tanto los que compartieron conmigo la etapa inicial en plan común como aquellos que fueron compañeros de carrera. Gracias a ellos la vida universitaria se convirtió en una de las mejores épocas de mi vida, llena de gratos recuerdos. Además, agradezco el apoyo incondicional de mis amigos fuera del ámbito universitario, así como también el amor infinito que me brindaron mis mascotas que siempre me recibieron con la mayor de las alegrías cuando llegaba cansado de la universidad. Agradezco también a mis profesores por su dedicación y compromiso en la formación de excelentes profesionales. En particular, quiero mencionar a mi profesor Guía, Juan Pablo Segovia, por su valiosa ayuda en el desarrollo de esta memoria de título. También a la empresa Green I&C por confiar en mí para la realización de este trabajo proporcionándome las herramientas necesarias para avanzar sin contratiempos. Un agradecimiento especial a Christofer Olivares por su invaluable apoyo durante mi práctica profesional y la elaboración de esta memoria, siendo un pilar fundamental en mis primeros pasos en el ámbito profesional. A todas y todos los que formaron parte de este largo camino, ¡muchas gracias!

## **Sumario**

En el contexto de la empresa Green I&C se identificó la oportunidad de evaluar la viabilidad técnica de ofrecer la programación de Controladores Lógicos Programables (PLC) como nuevo servicio. El problema radicaba en la falta de este servicio, lo cual presenta una oportunidad de negocio ya que la empresa cuenta con un área de Instrumentación y sistemas de control. Para abordar esta problemática, se propuso una investigación exhaustiva que incluye la evaluación de software de PLC, la definición de nuevos servicios, la identificación de la infraestructura tecnológica necesaria y la investigación de metodologías eficientes de programación y diseño, además de la adhesión a normas y estándares en el diseño de pantallas HMI. Como resultado se desarrolló un informe que destaca los softwares seleccionados, la infraestructura requerida, flujos de trabajo eficientes y ejemplos prácticos ilustrativos, demostrando así la viabilidad técnica de implementar con éxito la programación de PLC como nuevo servicio en Green I&C, lo que potencialmente aumentará su competitividad en el mercado.

## **Summary**

In the context of the Green I&C company, an opportunity was identified to evaluate the technical feasibility of offering Programmable Logic Controller (PLC) programming as a new service. The problem stemmed from the absence of this service, which presents a business opportunity as the company has expertise in Instrumentation and Control Systems. To address this issue, a comprehensive investigation was proposed, including the evaluation of PLC software, the definition of new services, the identification of necessary technological infrastructure, and the research of efficient programming and design methodologies, in addition to adherence to standards and regulations in HMI (Human-Machine Interface) design. As a result, a report was developed highlighting the selected software, the required infrastructure, efficient workflow processes, and illustrative practical examples, thus demonstrating the technical feasibility of successfully implementing PLC programming as a new service at Green I&C, potentially enhancing its competitiveness in the market.

# Tabla de Contenidos

<b>1.</b>	<b>INTRODUCCIÓN .....</b>	<b>1</b>
1.1.	INTRODUCCIÓN GENERAL.....	1
1.2.	TRABAJOS PREVIOS .....	1
1.2.1	<i>Importancia y aplicaciones de PLC's en la industria.....</i>	<i>1</i>
1.2.2	<i>Importancia de pruebas FAT.....</i>	<i>3</i>
1.2.3	<i>Aplicaciones de modelación de procesos. ....</i>	<i>3</i>
1.2.4	<i>Discusión .....</i>	<i>4</i>
1.3.	OBJETIVOS .....	5
1.3.1	<i>Objetivo General .....</i>	<i>5</i>
1.3.2	<i>Objetivos Específicos.....</i>	<i>5</i>
1.4.	ALCANCES Y LIMITACIONES .....	5
1.5.	TEMARIO Y METODOLOGÍA.....	5
<b>2.</b>	<b>MARCO TEÓRICO.....</b>	<b>7</b>
2.1.	PLC EN LA INDUSTRIA .....	7
2.1.1	<i>Introducción al PLC.....</i>	<i>7</i>
2.1.2	<i>Estructura interna y externa del PLC.....</i>	<i>7</i>
2.1.3	<i>Ventajas y desventajas de la utilización de PLC .....</i>	<i>8</i>
2.2.	MARCAS DOMINANTES .....	9
2.3.	REVISIÓN DEL MERCADO.....	9
2.4.	NECESIDADES EN LA PROGRAMACIÓN DE PLC.....	10
<b>3.</b>	<b>ANÁLISIS TÉCNICO.....</b>	<b>12</b>
3.1.	INTRODUCCIÓN .....	12
3.2.	PROGRAMACIÓN DE PLC. ....	12
3.2.1	<i>Tipos de lenguaje.....</i>	<i>12</i>
3.2.2	<i>Softwares .....</i>	<i>13</i>
3.3.	COMPARACIÓN DE MARCAS .....	14
3.3.1	<i>Comparación a nivel de Hardware.....</i>	<i>15</i>
3.3.2	<i>Comparación a nivel de Software.....</i>	<i>16</i>
3.3.3	<i>Requerimientos tecnológicos .....</i>	<i>20</i>
<b>4.</b>	<b>DESARROLLO DEL SERVICIO.....</b>	<b>23</b>
4.1.	INTRODUCCIÓN .....	23
4.2.	METODOLOGÍA PARA DESARROLLO DEL SERVICIO .....	23
4.2.1	<i>Requerimientos del Cliente.....</i>	<i>23</i>
4.2.2	<i>Descripción del sistema.....</i>	<i>23</i>
4.2.3	<i>Modelación del proceso con uso de simuladores .....</i>	<i>24</i>
4.2.4	<i>Validación del modelo .....</i>	<i>25</i>
4.2.5	<i>Diseño del sistema de control .....</i>	<i>27</i>
4.2.6	<i>Programación PLC.....</i>	<i>28</i>
4.2.7	<i>Depuración del programa .....</i>	<i>28</i>
4.2.8	<i>Realización de pruebas FAT.....</i>	<i>28</i>
4.2.9	<i>Puesta en marcha .....</i>	<i>30</i>
<b>5.</b>	<b>PANTALLAS HMI.....</b>	<b>32</b>
5.1.	INTRODUCCIÓN .....	32
5.2.	FUNCIONES PRINCIPALES DE UNA HMI .....	32
5.3.	DISEÑO .....	33
<b>6.</b>	<b>EJEMPLOS PRÁCTICOS .....</b>	<b>36</b>
6.1.	INTRODUCCIÓN .....	36
6.2.	PROCESO DE NEUTRALIZACIÓN QUÍMICA .....	36

6.2.1	<i>Descripción del sistema</i> .....	36
6.2.2	<i>Modelación del proceso</i> .....	38
6.2.3	<i>Modelación del proceso con uso de simuladores</i> .....	40
6.2.4	<i>Pruebas en Lazo Abierto</i> .....	42
6.2.5	<i>Diseño del sistema de control</i> .....	43
6.2.6	<i>Programación PLC</i> .....	44
6.2.7	<i>Sintonización</i> .....	47
6.2.8	<i>Realización de pruebas FAT</i> .....	51
6.2.9	<i>Pantalla HMI</i> .....	53
6.2.10	<i>Puesta en marcha</i> .....	53
6.3.	<b>PLANTA DE TRATAMIENTO DE RILES</b> .....	54
6.3.1	<i>Descripción del problema</i> .....	54
6.3.2	<i>Filosofía de control y operación</i> .....	54
6.3.3	<i>Modelación del proceso con uso de simuladores</i> .....	57
6.3.4	<i>Programación PLC</i> .....	59
6.3.5	<i>Diseño de pantalla HMI</i> .....	59
6.3.6	<i>Realización de pruebas FAT</i> .....	62
6.3.7	<i>Puesta en marcha</i> .....	64
<b>7.</b>	<b>CONCLUSIÓN</b> .....	<b>65</b>
	<b>ABREVIACIONES</b> .....	<b>66</b>
	<b>REFERENCIAS</b> .....	<b>67</b>
	<b>ANEXO A. COTIZACIONES</b> .....	<b>A.1</b>
	<b>ANEXO B. NORMATIVA HMI</b> .....	<b>B.1</b>
	<b>ANEXO C. EJEMPLO NEUTRALIZACIÓN QUÍMICA</b> .....	<b>C.1</b>
	<b>ANEXO D. EJEMPLO PLANTA DE RILES</b> .....	<b>D.1</b>
	<b>ANEXO E. CÓDIGOS RSLOGIX5000</b> .....	<b>E.1</b>
	CÓDIGO INTERPOLACIÓN PH .....	E.1
	LAZO DE PH .....	E.2
	ENCLAVAMIENTO B-01 Y M-01 .....	E.3
	ENCLAVAMIENTOS EJEMPLO PLANTA DE RILES.....	E.4

## Lista de Tablas

Tabla 3.1 Comparación a nivel de Hardware .....	15
Fig. 3.2 Tia Portal vs Studio5000.....	16
Tabla 3.2 Requisitos Técnicos .....	21
Tabla 6.1 Parámetros del Reactor .....	38
Tabla 6.2 Parámetros del Proceso .....	38
Tabla 6.3 Parámetros de sintonización Kc por zona .....	50
Tabla 6.4 Parámetros Ti de sintonización por zona .....	50
Tabla 6.5 Alarmas de Nivel.....	51
Tabla 6.6 Alarmas de pH.....	51
Tabla 6.7 Enclavamiento de seguridad .....	51
Tabla 6.8 Enclavamientos .....	55
Tabla 6.9 Alarmas .....	56
Tabla 6.10 Usuarios Creados .....	61



## Lista de Figuras

Fig. 2.1 Estructura interna de un Plc .....	8
Fig. 3.1 Diagrama de comparación de marcas .....	14
Fig. 3.3 Studio5000 Layout.....	17
Fig. 3.4 Tia Portal Layout .....	18
Fig. 3.5 Ejemplo de Licencia Allen-Bradley.....	19
Fig. 4.1 Diagrama de Flujo Para Modelación De Procesos .....	25
Fig. 4.2 Diagrama de Flujo propuesto para diseño de controlador .....	27
Fig. 4.3 Diagrama de Flujo para Pruebas FAT .....	29
Fig. 5.1 Representación de Sistemas Interactivos Centrados En Humanos .....	34
Fig. 6.1 Curva de pH de Neutralización con NaOH.....	37
Fig. 6.2 P&ID Proceso de Neutralización.....	37
Fig. 6.1 Vista General Modelo Simulink .....	41
Fig. 6.4 Lazo de Nivel en Simulink .....	41
Fig. 6.5 Lazo de pH en Simulink .....	42
Fig. 6.6 Prueba en Lazo Abierto .....	42
Fig. 6.7 Diagrama SAMA .....	43
Fig. 6.8 Controller Tag's.....	44
Fig. 6.9 Creación de nueva rutina en Rslogix5000 .....	44
Fig. 6.10 Textos Estructurados.....	44
Fig. 6.11 Enclavamientos en Diagrama Ladder .....	46
Fig. 6.12 Enclavamientos B-01 y M-01 .....	47
Fig. 6.13 Depuración en Rslogix5000.....	47
Fig. 6.14 Sintonización Lazo de Flujo .....	48
Fig. 6.15 Sintonización Lazo de Nivel.....	49
Fig. 6.16 Lazo de pH sin Gain-Scheduler .....	49
Fig. 6.17 Lazo de pH Sintonizado.....	50
Fig. 6.18 Lazo de Nivel con Activación de Enclavamiento K-01.....	52
Fig. 6.19 Lazo de Ph con Activación de Enclavamiento K-01 .....	52
Fig. 6.20 Lazo de Flujo con Activación de Enclavamiento K-01 .....	53
Fig. 2.21 OPC Write.....	57
Fig. 6.22 Señales de nivel de estanques en Simulink.....	58
Fig. 6.23 Modelo simplificado en Simulink.....	58

Fig. 6.24 Código de Enclavamiento P-01 .....	59
Fig. 6.25 HMI de referencia .....	60
Fig. 6.26 Creación de usuarios en Factorytalk .....	61
Fig. 6.27 Lógica de Alarma de Nivel .....	62
Fig. 6.28 Vista general de Alarmas .....	62
Fig. 6.29 Funcionamiento de Enclavamiento P-01 en HMI.....	63
Fig. 6.30 Vista De Estados de Bombas en HMI.....	64
Fig. A.1 Cotización Precision de PLC de prueba Controllogix .....	A.1
Fig. A.2 Módulo Ethernet Controllogix .....	A.2
Fig. A.3 Cotización De Opciones - Workstation HP .....	A.3
Fig. A.4 Cotización De Lureye Softwares Siemens.....	A.4
Fig. A.5 Costo Rslinx Classic .....	A.5
Fig. A.6 Costo Studio5000 Professional Edition .....	A.6
Fig. A.7 Costo Factorytalk View Se .....	A.7
Fig. C.1 Pestaña De Inicio.....	C.1
Fig. C.2 Pestaña P&ID .....	C.2
Fig. C.3 Pestaña Diagrama Sama.....	C.3
Fig. C.4 Pestaña Interlock .....	C.4
Fig. C.5 Pestaña Tendencias .....	C.5
Fig. C.6 Pestaña Control Ph .....	C.6
Fig. C.7 Pestaña Control Nivel.....	C.7
Fig. C.8 Pestaña Control De Flujo .....	C.8
Fig. C.9 Pestaña De Alarmas .....	C.9
Fig. C.10 Matlab Function 1 .....	C.10
Fig. C.11 Matlab Function 2 .....	C.11
Fig. D.1 P&Id Planta De Riles .....	D.1
Fig. D.2 Pestaña De Inicio .....	D.2
Fig. D.3 Pestaña Tendencias Estanques A .....	D.3
Fig. D.4 Pestaña Tendencias Estanques B .....	D.4
Fig. D.5 Pestaña De Tendencias Reactores A.....	D.5
Fig. D.6 Pestaña De Tendencias Reactores B.....	D.6
Fig. D.7 Pestaña De Enclavamientos .....	D.7
Fig. D.8 Pop-Up De Parámetros.....	D.8

Fig. D.9 Pestaña De Alarmas ..... D.9

# 1. Introducción

---

## 1.1. Introducción General

En el contexto de la automatización industrial, el continuo desarrollo de la tecnología emplea un papel fundamental en el desarrollo y mejora de las operaciones industriales. Adaptarse a soluciones innovadoras es esencial para seguir siendo competitivo en un mercado en constante cambio. En este sentido la programación de PLC representa una herramienta fundamental en la automatización de procesos y sistemas industriales.

La empresa Green I&C, con su enfoque en el área de procesos con apoyo de otras áreas como instrumentación y sistemas de control, se encuentra en una posición estratégica para explorar nuevas oportunidades de negocio. La introducción de la programación de PLC como un servicio adicional en Green I&C se presenta como una posibilidad prometedora. Esta memoria de título se centra en la evaluación de la viabilidad técnica de este posible nuevo servicio.

A lo largo de este trabajo se ha realizado una exhaustiva investigación que abarca desde la evaluación de los softwares necesarios para llevar a cabo el servicio hasta la definición de los servicios específicos, la identificación de la tecnología requerida y la investigación de metodologías eficientes. El cumplimiento de normativas y estándares relevantes en el diseño de interfaces HMI.

## 1.2. Trabajos Previos

A continuación se presentarán los trabajos previos relevantes que han contribuido al desarrollo del campo de investigación de interés de esta memoria de título. Estos trabajos proporcionan antecedentes, teorías, enfoques metodológicos y resultados que servirán como punto de partida para la formulación de hipótesis, objetivos y metodología en el estudio.

### 1.2.1 Importancia y aplicaciones de PLC's en la industria

- ♣ M. V. Tenorio, "Breve Ensayo Sobre el PLC y su Importancia en la Industria," vol. 5, no. 2, pp. 150-165, 2008.

Este artículo es un breve ensayo sobre la importancia del PLC en la industria donde se describe la evolución de la ingeniería de automatización y control, desde los sistemas rudimentarios hasta los modernos PLC's que han revolucionado la producción industrial. Se destaca cómo el PLC ha mejorado la eficiencia, la seguridad y la calidad en la industria. Además, se menciona la importancia de la interfaz hombre-máquina (HMI) en la implementación de un sistema de control.

- ♣ [2] W. J. Torres Casas, "Aplicación de PLC's a nivel industrial," vol. 8, no. 4, pp. 225-240, 2012.

Es una guía sobre la aplicación de PLC's en la industria electrónica. Contiene información sobre qué es un PLC, cómo se utiliza en la industria, las ventajas de su uso en la automatización industrial y cómo se puede implementar un sistema de control de PLC's.

- ♣ [3] J. G. C. Lugo, J. J. P. Ybarra, y E. Romero, Metodología para realizar una automatización utilizando PLC, Impulso, vol. 18, 2005.

Este artículo presenta una metodología para generar automatizaciones eficientes utilizando PLC utilizando matrices de decisiones y diagramas de flujo, con el objetivo de aumentar la productividad de las empresas. También se habla de las ventajas de la automatización en términos de eficiencia y se presentan conclusiones a partir de la metodología propuesta.

- ♣ [4] P. Aguilera Martínez, Programación de PLC'S, Tesis doctoral, Universidad Autónoma de Nuevo León, 2002.

Este paper presenta antecedentes históricos sobre la industria, ofrece una descripción de la automatización, ventajas y desventajas de la utilización de PLC's, estructura interna y externa del PLC a nivel de hardware y distintos tipos de lenguaje de programación de PLC.

- ♣ [5] J. P. Ramírez, G. G. D. Conti, y J. J. P. Ybarra, La automatización modular: una nueva opción para procesos automatizables, Impulso, vol. 1, 2005.

En esta revista se destaca la sección "Metodología para realizar una automatización utilizando PLC" donde se expone una metodología que ayuda con pasos propuestos a generar una automatización de manera más sencilla, utilizando PLC's. Su importancia radica en la evaluación de las diferentes técnicas de administración de la tecnología para de una manera eficaz poder dar solución de forma sistematizada a la toma de decisiones y así incrementar la productividad de los sistemas empleados por las compañías.

### 1.2.2 Importancia de pruebas FAT

- ♣ [6] Baking Industrial Forum, "Factory Acceptance Testing," vol. 8, no. 3, pp. 210-225, 2015.

El enfoque de este paper se centra en las instrucciones de trabajo para FAT y SAT, así como en los documentos de puesta en marcha para sistemas de automatización. El objetivo del autor es proporcionar plantillas estandarizadas para estas pruebas con el fin de garantizar la consistencia en los diferentes procedimientos de prueba. Esta tesis será de ayuda para estudiar la realización de pruebas FAT en las industrias

- ♣ [7] J. P. Segovia Vera, "Aplicación de Simuladores al Diseño de Sistemas de Control," IITUDEC, 2012.

Este artículo resultará fundamental para estudiar la metodología de la aplicación de simuladores al diseño del sistema de control del proceso. Será útil para estudiar los aportes de la simulación al diseño de procesos y estrategias de control y comprender los objetivos de las pruebas FAT con simulación.

- ♣ [8] M. Kopcek, T. Skulavik, P. Tanuska y D. Mudroncik, Systematic approach to factory acceptance test planning, in Computer Aided Chemical Engineering, vol. 33, pp. 1597-1602, Elsevier, 2014.

En este artículo se aborda la propuesta del marco básico de un enfoque sistemático para las pruebas de aceptación en fábrica utilizando un ejemplo de estudio se propone el marco básico de un enfoque sistemático para la planificación, ejecución, evaluación y documentación de FAT, y finalmente, se especifica el plan detallado de FAT.

### 1.2.3 Aplicaciones de modelación de procesos.

- ♣ [9] R. Tyagi and E. J. Davison, Control of pH in a continuous stirred tank reactor (CSTR), IFAC Proceedings Volumes, vol. 26, no. 2, pp. 115-126, 1993.

En este artículo científico se estudia el problema del control de pH en un reactor continuo agitado desde un punto de vista teórico y aplicado. Se utiliza un modelo invariante de reacción para un sistema

ácido-base, que es una de las descripciones más completas para el problema de control de pH. Este artículo será útil para el entender los principios y modelación del lazo de pH para el proceso de neutralización.

- ♣ [10] M. Meza García, Disturbios del estado ácido-básico en el paciente crítico, Acta Médica Peruana, vol. 28, no. 1, pp. 46-55, 2011.

Este artículo resultará fundamental para estudiar conceptos claves de este proyecto tales como la definición de pH, electroneutralidad, ácidos, bases entre otros.

- ♣ [11] S. K. Al-Dawery, H. A. O. Al-Anbari, and D. F. Ahmed, "Nonlinear Adaptive Control of a pH Process," Iraqi Journal of Chemical and Petroleum Engineering, vol. 10, no. 1, pp. 9-16, 2009.

Este artículo es importante para este proyecto ya que presenta un estudio teórico y experimental del comportamiento dinámico del proceso de neutralización en un reactor continuo agitado tipo tanque (CSTR, por sus siglas en inglés). Este artículo describe los desafíos del control del pH en el tratamiento de residuos industriales y destaca la necesidad de utilizar un control no lineal adaptativo debido a las no linealidades e incertidumbres del proceso. Además, se introduce y aplica un controlador no lineal adaptativo especial al sistema de neutralización de pH. El artículo también describe un estudio de simulación realizado para evaluar el rendimiento del controlador adaptativo propuesto, esto será de ayuda al momento de seleccionar y diseñar el sistema de control para el lazo de pH.

#### *1.2.4 Discusión*

La revisión bibliográfica revela el estado actual del tema, destacando las ventajas de la programación de PLC's en la industria, como la mejora en la eficiencia y calidad de la producción industrial [1][2][4]. Sin embargo, también se identifican limitaciones y desafíos, como la falta de enfoques sistemáticos en la implementación de PLC y la necesidad de estandarización [6][8]. Además, se evidencia la importancia de las pruebas de Factory Acceptance Testing (FAT) para garantizar la consistencia en los procedimientos de prueba [6]. Estas deficiencias en los enfoques existentes destacan la necesidad de abordar la programación de PLC y las pruebas FAT de manera más eficiente y sistemática, lo que motiva la investigación de esta memoria de título para ofrecer una guía completa sobre el correcto desarrollo del servicio de programación de PLC.

### **1.3. Objetivos**

#### *1.3.1 Objetivo General*

Analizar la viabilidad técnica de implementar programación de Controladores Lógicos Programables (PLC) como servicio en la empresa Green I&C.

#### *1.3.2 Objetivos Específicos*

- Evaluar software de PLC adecuado.
- Definir nuevos servicios relacionados a la programación en PLC.
- Definir metodologías de programación y diseño.
- Investigar metodologías de diseño de pantallas HMI, normativas y estándares pertinentes.
- Analizar infraestructura tecnológica necesaria.
- Crear instructivos y ejemplos prácticos.
- Elaborar conclusiones basadas en el análisis técnico.

### **1.4. Alcances y Limitaciones**

Los alcances de esta memoria de título están delimitados por los objetivos específicos establecidos anteriormente. Estos alcances incluyen la evaluación de software de PLC, la definición de nuevos servicios, la identificación de la infraestructura tecnológica necesaria, la investigación de metodologías eficientes de programación y diseño, y el cumplimiento de normativas y estándares relevantes en el diseño de interfaces HMI. Se asume que los resultados y conclusiones se basarán en una investigación teórica y práctica pero no incluirán una implementación física a gran escala ni pruebas a nivel industrial. Además, se reconoce que esta investigación se realiza en el contexto de la empresa Green I&C por lo que la aplicabilidad de los resultados puede estar sujetas a las particularidades del entorno empresarial chileno.

### **1.5. Temario y Metodología**

La metodología para la memoria de título implica una serie de pasos que guiarán la investigación y análisis. En primer lugar se realizará una revisión de la literatura relacionada con la programación PLC y la automatización industrial. Esto incluirá el estudio de marcas dominantes y análisis de softwares de programación PLC.

Posteriormente, se realizará un análisis comparativo de software PLC enfocado en Studio 5000



de Allen-Bradley y TIA Portal de Siemens. También se incluye una investigación detallada sobre la tecnología requerida y el equipamiento necesario para implementar la programación PLC como servicio en Green I&C, la exploración de enfoques eficientes para programación y diseño de PLC, el estudio de normas y estándares aplicables al diseño de interfaces HMI para asegurar la calidad y el cumplimiento normativo. Además, se desarrollarán ejemplos prácticos como casos de estudio que respaldarán las conclusiones finales de la memoria.

## 2. Marco teórico

---

### 2.1. PLC en la industria

#### 2.1.1 *Introducción al PLC*

El PLC o controlador lógico programable por sus siglas en inglés es un dispositivo que ha revolucionado la producción industrial al permitir el funcionamiento de máquinas y sistemas de forma automatizada gracias a la lógica de control previamente programada.

El primer controlador lógico programable (PLC) apareció en 1968 diseñado para reemplazar los complicados circuitos de relés electromecánicos los cuales eran muy limitados y requerían procesos complejos de implementación, configuración y constante soporte técnico, además de ser ineficientes y presentar fallas constantes. El PLC fue creado para ser programado fácilmente por ingenieros y técnicos de planta que ya estaban familiarizados con la lógica del relé y los esquemas de control.

Su importancia radica en que ha mejorado la eficiencia, seguridad y calidad de la industria. Los PLC modernos son herramientas de trabajo habituales en los sistemas de automatización de procesos industriales para controlar dispositivos individuales para diversos fines. Estos dispositivos se caracterizan por una mayor confiabilidad, facilidad de integración y actualización, y la capacidad de operar durante largos períodos de tiempo con una baja necesidad de mantenimiento. El uso de PLC ha aportado grandes beneficios a la industria moderna y están presentes en prácticamente todos los procesos automatizados a nivel industrial.

#### 2.1.2 *Estructura interna y externa del PLC*

Los PLC poseen la siguiente estructura interna con elementos esenciales para su funcionamiento:

- **Entradas:** líneas de entradas, pueden ser del tipo análoga o digital y la cantidad de entradas depende del modelo del PLC y del tipo de fabricante.
- **Salidas:** Son una serie de línea de salida, pueden ser del tipo análoga o digital. A estas líneas de salidas se le conectan los actuadores.
- **CPU:** Se encarga de procesar el programa introducido por el usuario, para ello se disponen zonas de memorias e instrucciones de programas y registros.
- **Unidad de alimentación:** Componente esencial para proporcionar la energía necesaria

para el funcionamiento del PLC.

- **Dispositivos periféricos:** unidades adicionales de entradas o salidas, nuevas unidades de comunicación, etc.

La estructura interna del PLC se puede representar con el siguiente esquema:

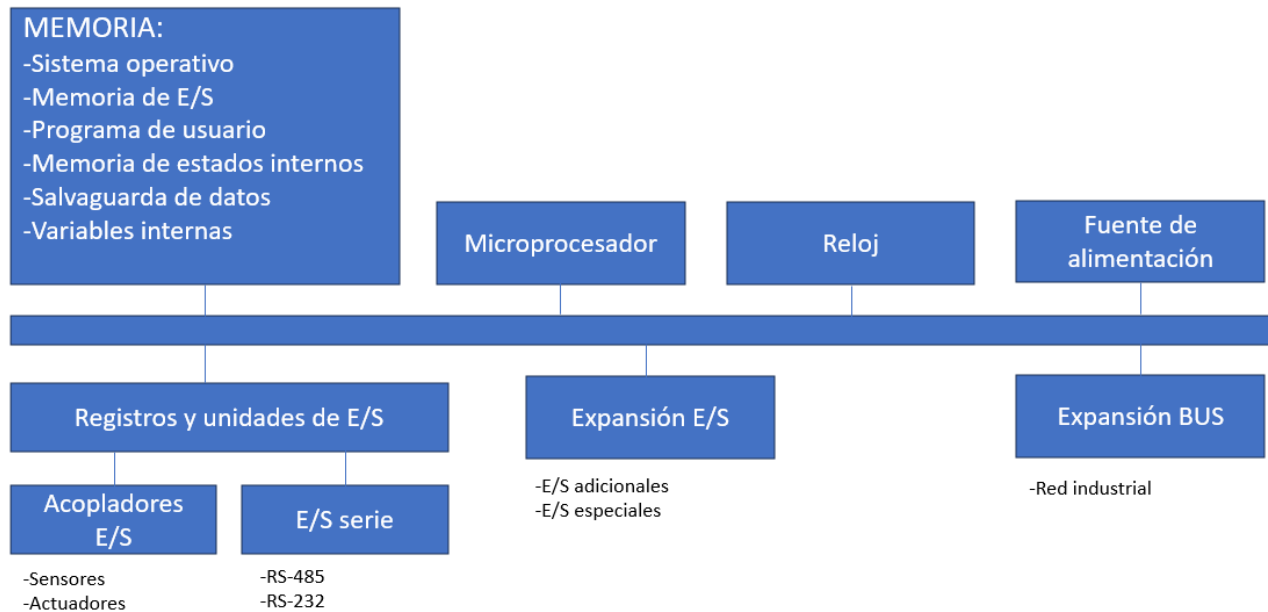


Fig. 2.1 Estructura interna de un PLC

Además, todos los PLC pueden poseer dos tipos de estructura externa:

- **Compacta:** Todos los elementos del PLC se encuentran en un solo bloque.
- **Modular:** cada módulo es una función (CPU, E/S, fuente de alimentación, etc.).

### 2.1.3 Ventajas y desventajas de la utilización de PLC

Debido a la variedad de fabricantes y modelos de PLC no todos estos dispositivos ofrecen las mismas ventajas pero a forma general se pueden identificar los siguientes puntos:

- Menor tiempo empleado en la elaboración de proyectos.
- Posibilidad de introducir modificaciones sin necesidad de cambiar cableado o añadir equipos.
- Menor espacio de ocupación.
- Mayor fiabilidad y eficiencia del sistema, lo cual como consecuencia implica mayor economía en mantenimiento. Además, estos mismos dispositivos pueden indicar y

detectar fallas u averías.

Dentro de las desventajas solo se puede mencionar que se necesita un programador especialista pero hoy en día no es un gran inconveniente ya que hay carreras universitarias especializadas en el tema así como también capacitaciones técnicas para trabajadores del área automática.

## **2.2. Marcas Dominantes**

En el mercado de la automatización industrial y la programación de PLC's la presencia de marcas dominantes es un factor relevante que influye en las decisiones de los usuarios finales y los integradores de sistemas. Aunque la falta de datos específicos sobre las ventas de cada marca puede dificultar una evaluación precisa. La experiencia y el conocimiento del campo indican que existen dos marcas principales que comparten la mayor parte del mercado.

**Allen-Bradley** es una marca asociada a Rockwell Automation, se destaca como una de las marcas más reconocidas y utilizadas en el sector. Su amplia gama de productos incluidos PLC como ControlLogix y CompactLogix, así como su software de programación, como RSLogix o Studio5000 han ganado una base sólida de usuarios por su fiabilidad. Algunas empresas integradoras de esta marca en Chile son Precision, Telconet, Sertronik e ITC Ingeniería.

**Siemens** es otro jugador clave en el mercado con su línea de productos SIMATIC que incluye PLC como SIMATIC S7 y software de programación como TIA Portal que incluye diversos softwares dentro del mismo los cuales serán detallados más adelante. Siemens ha establecido una presencia global significativa y es ampliamente utilizado en una variedad de aplicaciones industriales. Algunas empresas integradoras de esta marca en Chile son Lureye, Emeltec, Termodinámica, Pymatek y Vignola entre otras. Los integradores oficiales de Siemens en Chile se pueden encontrar directo en la página web del fabricante.

Además de Allen-Bradley y Siemens, existen numerosas otras marcas que abarcan el 33% restante del mercado lo que refleja la diversidad y la competencia en la industria. Marcas como **Mitsubishi, Schneider Electric, Omron y ABB**, entre otras, ofrecen soluciones y sistemas de automatización que se adaptan a necesidades específicas.

## **2.3. Revisión del mercado**

La revisión de mercado revela un panorama diverso y competitivo en el ámbito de la programación de PLC y sistemas de automatización en la industria. Empresas como Sertronik y ControlSim ofrecen una amplia gama de servicios que incluyen la programación de PLC,

actualizaciones de equipos, migraciones, y desarrollo de pantallas HMI, lo que refleja su enfoque en la eficiencia y mejora de sistemas existentes. Por otro lado, VOLTA y CAISE se centran en el diseño de ingeniería y programación de PLC/HMI/SCADA, lo que sugiere una orientación hacia proyectos de automatización desde su fase inicial. Finalmente RP Ingeniería se destaca por su enfoque en el desarrollo de especificaciones y propuestas de sistemas de automatización, lo que indica su papel en la planificación y diseño de soluciones personalizadas. En este entorno competitivo, la capacidad de ofrecer soluciones eficientes y adaptadas a las necesidades específicas de los clientes se convierte en un factor crítico para el éxito en el mercado de la automatización industrial.

#### **2.4. Necesidades en la programación de PLC**

Las necesidades en la programación de PLC se extienden en múltiples disciplinas para garantizar el éxito en el servicio. En el caso de Green I&C es una empresa que cuenta con especialistas en procesos lo que presenta una ventaja destacada.

Una de las necesidades fundamentales es la creación de modelos confiables para la simulación de procesos industriales. La simulación desempeña un papel crucial en la optimización y validación de sistemas de control y contar con especialistas en procesos facilita la modelación precisa de estos sistemas. Esto asegura que los PLC sean programados y probados en ambientes simulados que reflejan con precisión las condiciones reales de la planta, lo que es esencial para la realización de pruebas de FAT efectivas.

Además, la necesidad de eficiencia y precisión en la programación de PLC se ve respaldada por la experiencia en procesos de Green I&C. La comprensión de los procesos industriales permite una programación más precisa y eficiente de los controladores.

En el ámbito del mercado nacional los PLCs se utilizan en una amplia gama de industrias incluyendo la minería, manufactura, energía, alimentos y bebidas, petróleo y gas, entre otros. Por ejemplo, en la industria minera los PLCs se utilizan para el control de procesos de extracción y procesamiento de minerales, lo que es fundamental en un país con una industria minera significativa. Además, no son pocas las compañías que se encuentran en la actualidad probando e implementando equipos y procesos automatizados. Un ejemplo concreto de esta aplicación se encuentra en la reciente implementación de un sistema SCADA en Radomiro Tomic. Este sistema que fue incorporado este año brinda la capacidad de reunir y analizar datos en tiempo real sobre el rendimiento eléctrico de palas y perforadoras en las operaciones mineras. Esto se traduce en decisiones más informadas para los operadores mejorando la eficiencia y el rendimiento de estos equipos. Además, ha simplificado la

programación de mantenimiento eléctrico y habilitado la operación a distancia de los sistemas de despacho eléctrico en un rango de ocho a seis kilómetros. Adicionalmente, se ha añadido una función de bloqueo digital para reforzar los procedimientos de seguridad tanto para el personal como para los equipos involucrados.

En la industria manufacturera, son esenciales para la automatización de líneas de producción lo que abarca desde la fabricación de productos electrónicos hasta la producción de alimentos. Además, el sector de energías renovables, como la energía solar y eólica, está experimentando un crecimiento significativo en Chile, la generación con renovables en Chile creció 40% interanual en abril de este año y además el Ministerio de Energía chileno estableció como meta que en 2030 el 60% de la electricidad provenga de fuentes de energía renovable. Estos sistemas requieren una programación especializada para la integración y monitoreo de generación distribuida y almacenamiento de energía. Esto impulsará aún más la demanda de programación de PLCs ya que se busca un control eficiente y la optimización de la generación y distribución de energía en un contexto de creciente demanda.

## 3. Análisis técnico

---

### 3.1. Introducción

La programación de PLC's representa un ámbito esencial en la automatización industrial. Esencialmente, la programación PLC se trata de dotar de inteligencia a las máquinas y procesos industriales permitiéndoles así ejecutar tareas específicas de manera autónoma y precisa. Esta inteligencia se traduce en una secuencia de instrucciones lógicas que dictan cómo los PLC deben responder a diferentes condiciones y entradas del proceso.

Para realizar el análisis técnico es importante conocer en profundidad la programación de PLC's y las alternativas de programación que ofrecen las marcas dominantes para poder seleccionar el software adecuado según las necesidades de Green I&C.

### 3.2. Programación de PLC.

#### 3.2.1 Tipos de lenguaje

Los tipos de lenguaje para programar PLC's se componen de símbolos, caracteres y reglas diseñadas para poder establecer comunicación entre el usuario y la máquina. Actualmente existe el estándar IEC 61131-3 el cual busca estandarizar los principales lenguajes de programación en PLC para así definir e identificar sus principales características, a pesar de la existencia del estándar IEC 61131-3, los PLC pueden incluir lenguajes adicionales propios de cada marca.

Dentro de los lenguajes principales de programación en PLC se encuentran:

- **Instruction List (IL):** Es el lenguaje de programación de texto más antiguo en comparación al resto de los lenguajes. Consiste en listas de instrucciones a través de llamado de funciones.
- **Ladder (LD):** Conocido como lógica de escalera, es un lenguaje gráfico el cuál se compone de niveles verticales los cuáles pueden ser divididos en lado izquierdo y derecho. El lado izquierdo corresponde a las instrucciones y el lado derecho a las salidas del sistema. El PLC interpreta este lenguaje de abajo hacia arriba y de izquierda a derecha.
- **Function Block Diagram (FDB):** Lenguaje de programación gráfico el cual define funciones entre variables de entrada y salida. Utiliza bloques de símbolo lógico.
- **Structured Text (ST):** Lenguaje de programación de texto gráfico el cuál es capaz de

definir funciones entre las variables de entrada y las de salida. Se compone de series de instrucciones que pueden ser ejecutadas.

### 3.2.2 *Softwares*

Los Softwares para programación PLC son esenciales en la automatización industrial ya que permiten a los ingenieros diseñar y programar los sistemas de control destinados a diversas aplicaciones industriales. Existe una amplia variedad de Softwares para programación PLC en el mercado, cada uno proporcionado por fabricantes específicos y diseñados exclusivamente para sus modelos de PLC. De acuerdo con lo estudiado en la revisión de mercado se examinarán en detalle los programas de las marcas líderes del mercado correspondientes a Siemens y Allen-Bradley.

En cuanto a la interfaz de desarrollo integrado (IDE) de cada software, varía según el fabricante pero la mayoría comparte herramientas en común, como editores de código, herramientas de depuración y entornos de simulación. Estas funciones permiten a los programadores poner a prueba sus programas de control antes de cargarlos en un PLC real. Además, estos softwares permiten la comunicación con el PLC, lo que permite cargar y descargar programas así como también controlar el PLC en tiempo real. Esta conexión depende del protocolo de comunicación la cual depende del tipo de PLC, pero el protocolo Ethernet y los puertos seriales suelen estar presentes en la mayoría de los casos.

En el caso de la marca líder Allen-Bradley posee los siguientes Softwares de interés para esta memoria:

- **Studio5000:** Utilizado para la programación y configuración de Controladores Logix. Este entorno proporciona herramientas integrales para diseñar, simular y mantener sistemas de control avanzados.
- **FactoryTalk View Studio:** Software utilizado para diseñar y configurar HMI.
- **Emulate 5000:** Permite simular el comportamiento de Controladores Logix y sistemas de automatización industrial en un entorno virtual, lo que facilita el desarrollo y la depuración de aplicaciones antes de implementarlas en hardware real.
- **Rslinx:** Es un software utilizado para establecer comunicación entre dispositivos y softwares de Allen-Bradley.



En el caso de la marca líder Siemens posee los siguientes Softwares de interés para esta memoria:

- **Step7:** Entorno de programación principal de Siemens utilizado para programar y configurar PLC's de la familia SIMATIC.
- **WinCC:** Software utilizado para diseñar y configurar HMI.
- **PLCSim:** Software que permite simular el comportamiento de Controladores SIMATIC en un entorno virtual. Facilita el desarrollo, la prueba y la depuración de programas de control antes de su implementación en hardware real.

Es importante mencionar que estos softwares vienen integrados en TIA Portal, la cual Proporciona una plataforma unificada para la programación, simulación, supervisión y mantenimiento de sistemas de control industrial.

### 3.3. Comparación de marcas

La comparación será entre dos de los gigantes de la industria, Allen-Bradley y Siemens. Ambas marcas han establecido estándares de excelencia en cuanto a hardware y software y esta comparación tiene como objetivo explorar las características distintivas de cada una. Desde la robustez de sus controladores hasta la versatilidad de sus entornos de programación, esta evaluación proporcionará una visión detallada de las fortalezas y consideraciones clave para ayudar a tomar decisiones esenciales para el desarrollo del servicio de programación PLC en Green I&C.

Para realizar la comparación entre Allen-Bradley y Siemens, se identificaron los siguientes puntos de comparación:

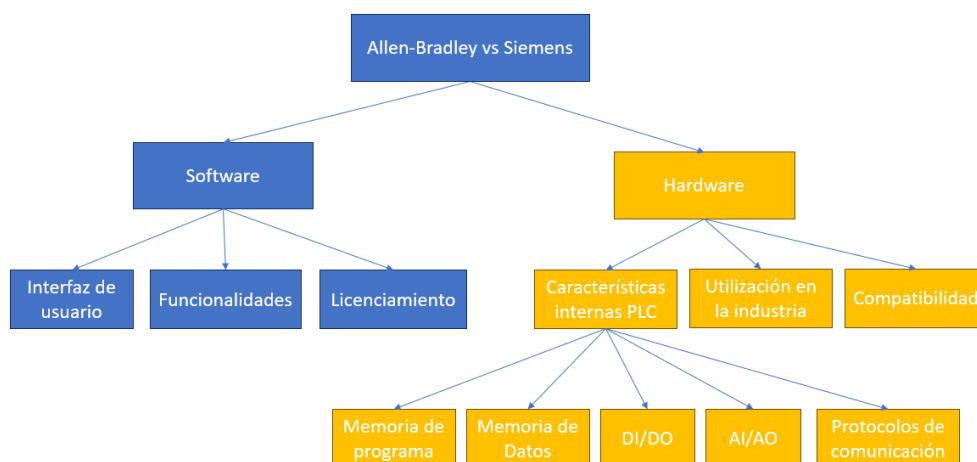


Fig. 3.1 Diagrama de comparación de marcas

### 3.3.1 Comparación a nivel de Hardware

A nivel de hardware en cuanto a robustez y confiabilidad ambas opciones ofrecen controladores igualmente resistentes y capaces de operar en condiciones industriales. Sin embargo, de igual manera se pueden apreciar diferencias en cuanto a las características internas de algunos modelos de PLC de ambas marcas las cuales se pueden observar en la siguiente tabla comparativa.

**Tabla 3.1 Comparación a nivel de Hardware**

<b>Descripción</b>	<b>Allen-Bradley</b>	<b>Siemens</b>
Memoria de programa	1-2 MB	1 MB
Memoria de datos	750 KB	512 KB
AI/AO	16/16	16/16
DI/DO	32/16	4/4
Protocolos de comunicación	Ethernet/IP, DeviceNet,ControlNET, DH+, DH485, RS-232/485, MODBUS, Profibus.	Ethernet, RS-232/485, Profibus, ASI, MODBUS TCP/IP

En cuanto a la utilización en la industria Allen-Bradley es comúnmente utilizada en Estados Unidos y es considerada una marca estándar de la industria. Es común que en la industria estadounidense se pida dentro de sus requerimientos la utilización de PLC de Allen-Bradley ya que dichos PLC utilizan protocolos de comunicación típicos de Estados Unidos tales como DeviceNet, ControlNet, EthernetIP, DH+ y DH485.

Por otro lado Siemens es más utilizados en empresas europeas que utilizan protocolos de comunicación europeos como Profibus, ASI, MODBUS o MODBUS TCP/IP. Aunque ambos equipos permiten la integración de módulos de comunicación para cualquiera de los protocolos de comunicación anteriormente mencionados.

Con respecto a la compatibilidad Siemens presenta mayor flexibilidad con el uso de accesorios de otras marcas, por ejemplo, Siemens no requiere un rack ni una fuente de alimentación propia de la marca, sino que funciona con cualquier fuente de 24 Vcc externa, por otro lado, Allen-Bradley requiere el uso exclusivo de accesorios de la misma marca.

### 3.3.2 Comparación a nivel de Software

En la búsqueda de la selección del software óptimo para la programación de PLC, la comparación entre Studio 5000 de Allen-Bradley y TIA Portal de Siemens es clave. Estos dos entornos de desarrollo representan los softwares más utilizados en aplicaciones industriales, ambos fabricantes ofrecen softwares robustos y confiables para la programación de PLC por lo que encontrar puntos de comparación de ambos softwares es imprescindible para la correcta selección de acuerdo con las necesidades de la empresa.

Como primer punto de comparación, es importante mencionar que según testimonios de especialistas programadores de PLC la interfaz de usuario de Allen-Bradley es más amigable e intuitiva que la interfaz de TIA Portal. También se menciona que la curva de aprendizaje en el software ofrecido por Siemens es más lenta en comparación a la de Studio5000 y que en el caso de TIA Portal, es necesario que el programador sea especialista en el software de Siemens.

En cuanto a los programas utilizados, TIA Portal tiene la ventaja de que ofrece herramientas como software de diseño de HMI y diseño de arquitectura de control integrados dentro del mismo software, en cambio en Allen-Bradley se requieren softwares adicionales, como por ejemplo FactoryTalk o View Designer.



**Fig. 3.2 Tia Portal vs Studio5000**

TIA Portal al incluir softwares integrados dentro del mismo al exportar el proyecto se exportan múltiples archivos en múltiples carpetas, en cambio en Allen-Bradley se exporta un archivo ACD único. También es importante mencionar que Allen-Bradley ofrece la posibilidad de exportar datos en formato CSV o en L5K el cual se utiliza para exportar proyectos de manera que puedan ser

compartidos o transferidos a otros sistemas o dispositivos de programación. Esto facilita la colaboración entre programadores y permite cargar proyectos en diferentes Controladores Logix sin tener que reconstruir la lógica de control desde cero.

Dentro de las características de cada Software también se diferencian por la cantidad de controladores permitidos por proyecto, TIA Portal ofrece la posibilidad de trabajar con múltiples controladores por proyecto, mientras que en Studio5000 solo se puede trabajar con un controlador por proyecto.

En temas de comunicación Studio5000 requiere RSLinx como software de comunicación, en cambio TIA PORTAL no requiere softwares adicionales. Además, en Studio5000 la estación de programación debe estar en la misma subred para comunicaciones Ethernet, mientras que en TIA Portal se añade una dirección IP adicional a la estación de programación para permitir comunicaciones Ethernet inmediatas.

En cuanto a la distribución del campo de trabajo de cada software igual se pueden encontrar diferencias las cuales se pueden apreciar en las siguientes imágenes.

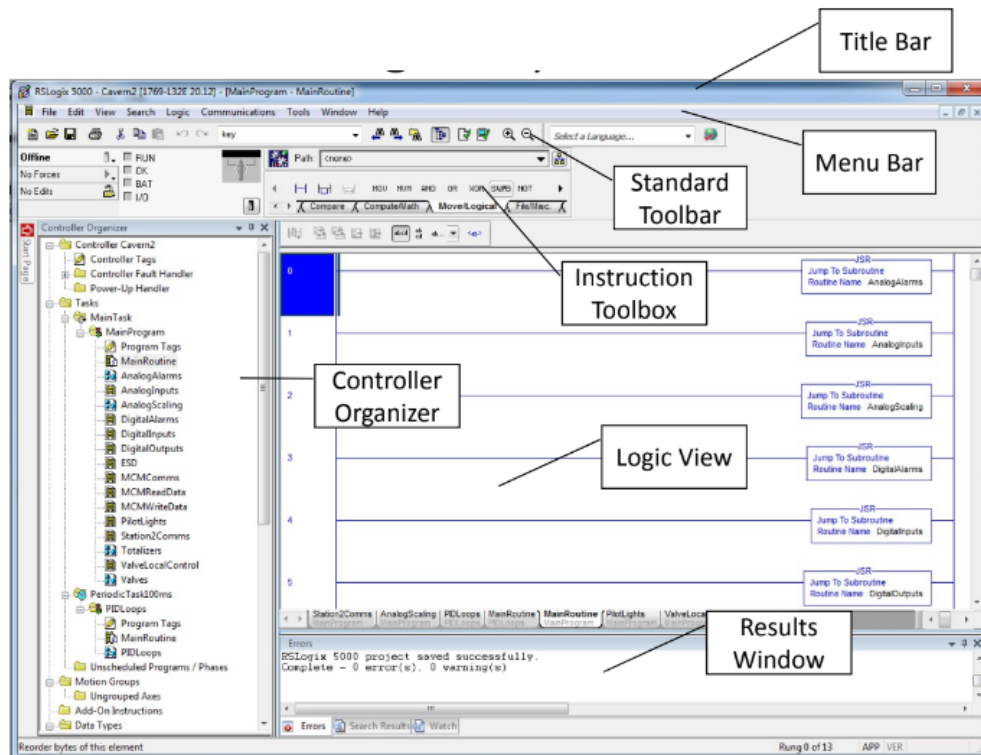
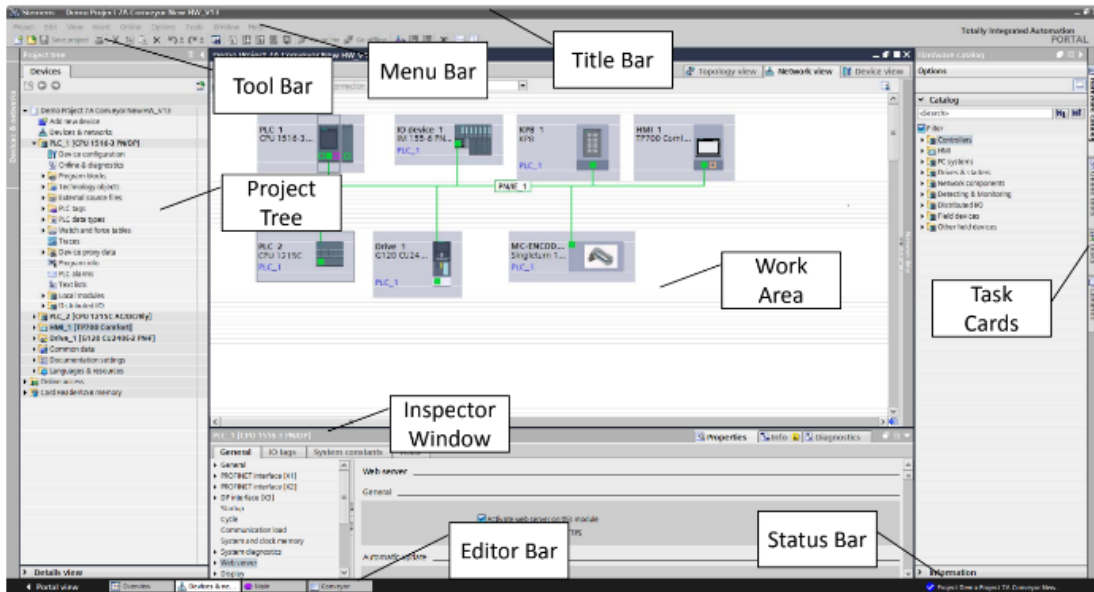


Fig. 3.3 Studio5000 Layout



**Fig. 3.4 TIA Portal Layout**

En cuanto al licenciamiento de cada software, en el caso de Siemens existen distintos tipos de licencia ya sea individual, flotante, de alquiler, de prueba, flotante combinada, PowerPack o de actualización, donde principalmente varía la limitación de tiempo y la cantidad de equipos en las que se puede utilizar el software.

También la licencia puede ser del tipo Basic o Profesional dependiendo de la serie del PLC, La licencia Basic permite programar cualquier PLC de la serie S7-1200 y las pantallas Basic. Si se necesita programar PLC más potentes, de las series S7-300, S7-400 o S7-1500, se requiere la licencia profesional.

En el caso de Allen-Bradley existe la licencia perpetua con mantenimiento o la suscripción. Una licencia perpetua es un tipo de licencia de software que otorga al usuario el derecho de usar una versión específica del software de manera indefinida sin restricciones de tiempo y por otro lado la suscripción permite a los usuarios acceder y utilizar los softwares durante un período de tiempo específico, generalmente en ciclos de suscripción anuales. Durante la suscripción los usuarios tienen acceso a las últimas versiones del software y en muchos casos a soporte técnico y actualizaciones.

	Perpetual With Maintenance	Subscription
License Flexibility ?	N/A	Yes
Intelligent Activation ?	N/A	Yes
Customizable Bundles	N/A	Yes
Reinstatement Fees	Yes	No
Phone Support	Optional	Included(8-5;M-F)
Software Updates	Included	Included
Knowledgebase	Yes	Yes
System Support	No	No
One Time Cost	\$USD 969.96 /license	-
Annual Cost	\$USD 194.66 /user/year	\$USD 399.51 /user/year
Product Selection	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Fig. 3.5 Ejemplo de licencia Allen-Bradley**

Finalmente, la distinción fundamental entre ambos softwares se encuentra en su enfoque en la programación. Siemens utiliza un tipo de lenguaje de programación llamado STL, AWG o también llamado lenguaje ensamblador de Siemens. Dicho lenguaje se basa en listas de instrucciones en un nivel muy bajo de programación orientado a la máquina. Este lenguaje permite realizar instrucciones de operaciones a nivel de bit.

Algunas de las ventajas de este lenguaje son:

- Costo computacional de la CPU bajo ya que al ser lenguaje tipo ensamblador el compilador lo entiende a la perfección y tarda menos tiempo en procesarlo por lo que no se necesita una CPU con componentes tan robustos.
- Es uno de los mejores lenguajes a la hora de trabajar con bucles y saltos.
- No tiene limitaciones a la hora de mover datos y trabajar con ellos.

Algunas de las desventajas de este lenguaje son:

- Curva de aprendizaje lenta y costosa.
- Es el lenguaje más complejo de seguir visualmente.
- Mientras más complejo es el proceso el tamaño del código se hace más grande.
- Cuando se requiere trabajar una lógica compleja el trabajo de seguimiento y depuración es difícil y susceptible a errores.
- En las puestas en marcha puede ser un problema si no se maneja bien este lenguaje.

- Es el menos preferido por los programadores de PLC y por eso, cuando entran a un proyecto ajeno y se encuentran un bloque de AWL suele ser problemático sobre todo en las puestas en marcha.
- Requiere mayor tiempo de desarrollo, lo que impacta directamente en los costos del proyecto.

Teniendo esto en cuenta podemos observar que programar en TIA Portal presenta un riesgo si no se cuenta con un programador especialista en Siemens, ya que el lenguaje ensamblador puede ser susceptible a errores y problemático en las puestas en marchas.

Dado que Green I&C es una empresa nueva en el campo de la programación de PLC y considerando que Allen-Bradley ofrece una menor probabilidad de riesgos, una curva de aprendizaje más rápida y tiempos de desarrollo de proyectos más reducidos, además de contar con un especialista en Instrumentación y Control con experiencia en Allen-Bradley, se percibe que Allen Bradley emerge como la opción más adecuada para brindar el nuevo servicio de programación de PLC.

### 3.3.3 *Requerimientos tecnológicos*

Para abordar los requerimientos tecnológicos, se llevará a cabo el análisis de dos alternativas viables. La primera opción implica la utilización de un PLC físico de prueba que comprenda elementos como chasis, CPU, fuente de poder y un módulo de comunicación Ethernet.

Esta alternativa permitiría redireccionar entradas y salidas físicas desde el mapa de I/O del PLC de prueba a un mapa de memoria traspasable vía un driver OPC o DDE existente en un PC conectado a la misma red. En el PC conectado a la red se pueden ocupar softwares de simulación para interactuar con el drive OPC o DDE para leer o escribir de acuerdo a los resultados del modelo, esto permitiría realizar pruebas FAT sobre el código final sobre la CPU y sería programada tal cual como serán instaladas en el proceso [7]

La segunda alternativa se centra en la viabilidad de operar en un entorno totalmente simulado. En este caso, se necesitará emplear un equipo robusto que asegure un rendimiento fluido durante las simulaciones. Para implementar esta alternativa se contempla el uso adicional del software **Emulate 5000** de Rockwell para simular un PLC virtual. Este PLC virtual se puede comunicar con programas de simulación mediante conexiones OPC o DDE, aprovechando el servidor OPC proporcionado por el software **Rslinx**. Este enfoque proporciona un entorno de simulación completo permitiendo una validación del sistema mediante pruebas FAT antes de proceder con la implementación física.

Para ambas alternativas se necesita un PC robusto que sea capaz de llevar a cabo simulaciones de manera fluida, para eso debemos tener en cuenta las especificaciones técnicas mínimas para poder ejecutar los softwares de Rockwell. Las especificaciones técnicas para dichos softwares se pueden encontrar en la siguiente tabla:

**Tabla 3.2 Requisitos técnicos**

<b>Requisitos</b>	
Sistema operativo	Windows 10 (64 bits) o Windows Server 2012 R2 (64 bits)
Procesador	Procesador de varios núcleos a 3.0 GHz o superior
RAM	16 GB de RAM o superior
Espacio en disco	20 GB de espacio libre en disco
Tarjeta gráfica	Resolución de pantalla de 1920 x 1080 o superior y Adaptador de gráficos compatible con DirectX 9 con soporte para hardware de controlador WDDM 2.0
Otros	Conexión a Internet para la activación del software y las actualizaciones. Se recomienda una conexión de red Ethernet para la comunicación con controladores y dispositivos.

Para este tipo de aplicación se recomienda una Workstation. Son computadoras diseñadas para aplicaciones profesionales y tareas que requieren un rendimiento intensivo, como diseño gráfico, modelado 3D, simulaciones, desarrollo de software y otras aplicaciones especializadas. Estos equipos incluyen potentes procesadores, tarjetas gráficas dedicadas de alto rendimiento, grandes cantidades de memoria RAM y opciones de almacenamiento rápido como unidades de almacenamiento SSD.

Dentro de las Workstation se recomienda la HP ZBOOK FIREFLY 14 G8 la cual incluye un procesador Intel Core i7, 16GB de RAM, tarjeta gráfica Nvidia T500 y capacidad de disco duro de 512 GB SSD dentro de sus características.

Ambas alternativas cumplen su propósito fundamental, el cual es facilitar la realización de pruebas FAT antes de la puesta en marcha. No se observan ventajas sustanciales de una sobre la otra



en cuanto a la función específica de validar el sistema. Sin embargo, se puede considerar el aspecto económico al evaluar estas opciones. La implementación de un PLC de prueba implica una inversión significativamente mayor en comparación con la alternativa de trabajar completamente en un entorno simulado. Este último enfoque ofrece una solución más rentable, eliminando la necesidad de adquirir hardware adicional. En el anexo de este documento se adjuntarán las cotizaciones detalladas de los equipos y softwares asociados, lo que proporcionará una visión más completa de los costos asociados con cada alternativa.

## 4. Desarrollo del servicio

---

### 4.1. Introducción

En el desarrollo del servicio de programación de PLC en Green I&C se busca estandarizar la prestación de servicios a través de la implementación de diagramas de flujo detallados y la creación de instructivos. Estos elementos son esenciales para garantizar un enfoque eficiente en la programación de PLC. Dentro de este mismo contexto se explorarán métodos de validación de modelos para asegurar la fiabilidad de estos mismos. Además, se presentarán diagramas de flujo que servirán como guías visuales que abarcarán cada etapa del servicio, desde la recepción del proyecto hasta la puesta en marcha.

### 4.2. Metodología para desarrollo del servicio

#### 4.2.1 *Requerimientos del Cliente*

Los requerimientos de los clientes abarcan un espectro amplio de aspectos fundamentales para el éxito de un proyecto. Estos incluyen las características de operación, que detallan cómo debe funcionar el sistema o proceso y cuáles son sus parámetros críticos. Además, es esencial comprender las características de los equipos involucrados, incluyendo especificaciones técnicas y requisitos particulares de cada componente. Los rangos de operación, que establecen los límites de funcionamiento del sistema. En algunos casos, se incluye el rango de costos de los equipos a utilizar, lo que proporciona una guía valiosa para la selección de componentes. Estos requerimientos son la base sobre la cual se construirá el diseño y la programación del PLC, asegurando que el sistema cumpla con las expectativas del cliente de manera precisa y eficiente.

#### 4.2.2 *Descripción del sistema*

Tras la recepción del proyecto es importante realizar una descripción del sistema o proceso junto con sus requerimientos con el propósito de obtener una total comprensión de la operación en su totalidad. Para lograrlo se inicia el proceso de recopilación de la información esencial que respaldará el desarrollo del servicio. Algunos de los documentos importantes para este propósito son los siguientes:

- **Filosofía de Control y Operación:** Este documento reviste gran importancia, ya que proporciona una comprensión profunda del funcionamiento del sistema o proceso.

También, ofrece información como los parámetros de operación, enclavamientos y aspectos relativos a la seguridad de la planta.

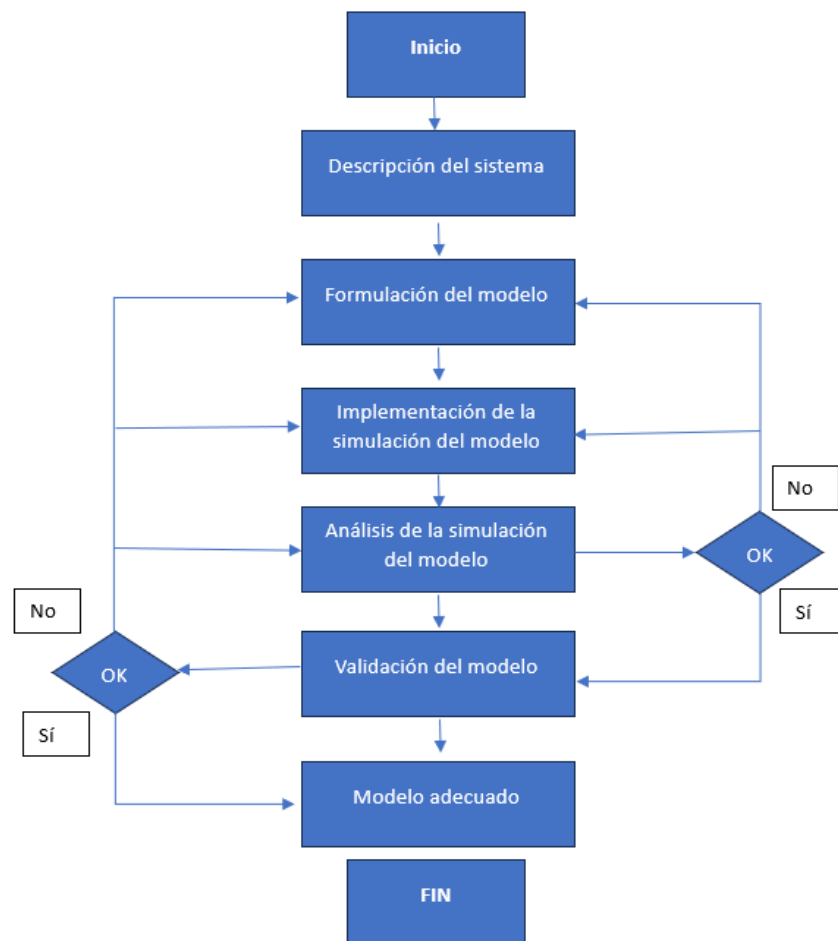
- **Listado de equipos:** Este documento es esencial para identificar y enumerar los equipos que intervienen en el proceso permitiendo una visión completa de los componentes involucrados.
- **Listado de señales:** La obtención de un listado de señales es fundamental para conocer las entradas y salidas del sistema, lo que facilita el dimensionamiento adecuado de los módulos de entrada/salida del PLC.
- **Variables de interés:** Identificar las variables de interés es crucial para el diseño del sistema de control ya que estas variables son las que se supervisarán y controlarán durante el proceso.
- **P&ID:** Es una herramienta esencial que muestra la representación gráfica de tuberías, equipos e instrumentación en el proceso y ayudará a facilitar la comprensión de la disposición física de los elementos y su interconexión.
- **SAMA:** Este diagrama permite comprender a través de lenguaje gráfico el funcionamiento de los lazos de control ayudando al especialista a analizar, diseñar o modificar la lógica de control de forma estructurada.
- **Arquitectura del sistema de control:** En situaciones donde el proyecto implica la modificación de una filosofía de control existente, la recopilación de información acerca de la arquitectura del sistema de control actual es imprescindible. Los planos de la arquitectura del sistema proporcionan una visión general de los equipos involucrados en el proceso, sus interconexiones y su funcionamiento.
- **Diagramas de lazos:** Los diagramas de lazos son herramientas gráficas que representan las interacciones y relaciones entre los componentes del sistema de control. Estos diagramas son esenciales para comprender cómo se comunican y cooperan los elementos en el sistema de control.

#### *4.2.3 Modelación del proceso con uso de simuladores*

La modelación de sistemas es un proceso esencial en la automatización industrial y la programación de PLC. Implica la creación de representaciones matemáticas o lógicas de un sistema físico o proceso con el propósito de comprender su comportamiento, predecir su respuesta a diferentes condiciones y, diseñar y controlar eficazmente dicho sistema.

La modelación se basa en la recopilación de datos y la identificación de variables clave que influyen en el sistema. Estas variables pueden ser entradas (como señales de sensores), salidas (como acciones de control), o parámetros internos del sistema. A través de ecuaciones, diagramas de bloques u otros métodos matemáticos, se establecen relaciones entre estas variables, lo que permite simular el comportamiento del sistema en diferentes situaciones.

Para la modelación del proceso ocupando simuladores se propone el siguiente diagrama de flujo para la modelación de procesos. Este diagrama visual representa de manera clara y organizada los pasos necesarios para llevar a cabo la modelación de un proceso. Utilizando símbolos y conexiones, el diagrama muestra la secuencia de las actividades y decisiones involucradas en el proceso de modelación.



**Fig. 4.1 Diagrama de flujo para modelación de procesos**

#### 4.2.4 Validación del modelo

Uno de los problemas más difíciles a los que nos enfrentamos es obtener un modelo que sea

una representación apropiada del sistema real estudiado, es decir, la obtención un modelo válido.

Una guía conocida para el proceso de validación fue introducida en el artículo [12] en la cual se destacan 3 puntos para validar un sistema:

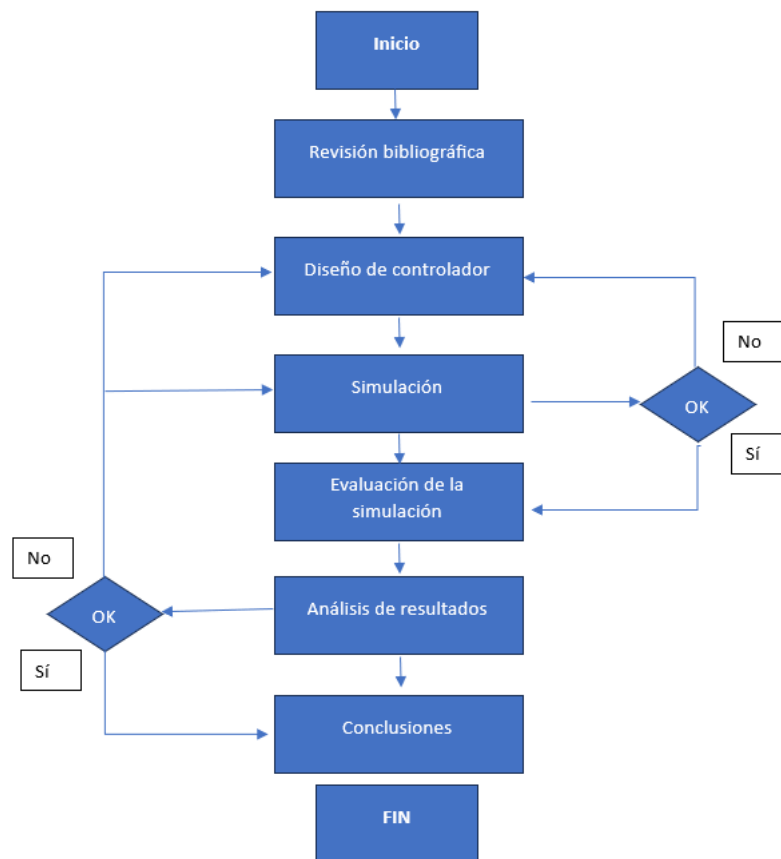
1. **Construir un modelo con un aspecto sumamente razonable:** El desarrollo de un modelo efectivo en la modelación de sistemas implica que el modelo refleje de manera precisa y razonable la realidad del sistema. Este proceso no se trata de una mera abstracción del modelador, sino más bien de una síntesis de la experiencia y la intuición de expertos que conocen el sistema a fondo. Por lo tanto, es fundamental que el modelador colabore estrechamente con los expertos para asegurar que el modelo sea fiel a la operación real del sistema. Esta interacción constante y regular con expertos es crucial para garantizar la calidad y precisión del modelo resultante.
2. **Comprobar empíricamente las suposiciones del modelo:** La validación de un modelo implica una evaluación cuantitativa de sus supuestos desde el principio. Esto puede incluir la verificación de suposiciones sobre la distribución de probabilidad de las variables de entrada mediante técnicas como Ji-cuadrada o Kolmogorov-Smirnov. En el caso de múltiples muestras de observaciones de un mismo fenómeno aleatorio, es útil justificar el uso combinado de datos mediante pruebas de Kruskal-Wallis para determinar la homogeneidad de poblaciones. Además, se emplea el análisis de sensibilidad para evaluar cómo varían los resultados de la simulación cuando se modifican los parámetros de entrada, se cambian las distribuciones de probabilidad o se altera el nivel de detalle del modelo en relación al sistema real. La sensibilidad del modelo a ciertos aspectos del sistema real destaca áreas que requieren un estudio más detallado.
3. **Determinar si los resultados del modelo son representativos:** La validación de un modelo de simulación se basa en asegurarse de que sus resultados sean comparables a lo que ocurre en el sistema real o lo que se esperaría en situaciones similares. Un buen modelo debe pronosticar con precisión el comportamiento del sistema real no solo para un conjunto de datos específico, sino para un rango completo de valores de entrada. Aunque existen pruebas estadísticas para comparar datos reales y simulados, en sistemas estocásticos no estacionarios, donde las distribuciones de probabilidad cambian con el tiempo, y con resultados del modelo que presentan autocorrelación, las técnicas estadísticas clásicas pueden no ser aplicables. En tales casos, la Prueba de

Turing, originada en la inteligencia artificial, puede ser útil. Esta prueba implica la comparación de resultados del modelo y del sistema real por expertos conocedores del sistema, sin que se les revele cuáles datos son reales y cuáles son simulados. Si los expertos pueden distinguir entre los datos reales y simulados, sus explicaciones ayudan a mejorar el modelo.

#### 4.2.5 *Diseño del sistema de control*

Una vez reunidos los antecedentes en los pasos anteriormente descritos se procede a definir una estrategia de control que describe cómo se lograrán los objetivos. Esto puede incluir la elección de algoritmos de control (como control PID o Fuzzy logic por ejemplo), la definición de consignas y límites, y la estructura general del sistema de control.

Para el diseño del sistema de control se propone el siguiente diagrama de flujo. Este diagrama visual proporciona una representación clara y organizada de los pasos necesarios en el proceso de diseño de sistemas de control. Utilizando símbolos y conexiones, el diagrama muestra la secuencia de las actividades y decisiones involucradas en el diseño del sistema.



**Fig. 4.2 Diagrama de flujo propuesto para diseño de controlador**

#### 4.2.6 Programación PLC

Una vez diseñada la estrategia de control se procede a escribir el código de control, que consiste en una serie de instrucciones lógicas que determinan cómo el PLC debe operar en función de las entradas del sistema y las condiciones especificadas. Esto implica la creación de rutinas de control, secuencias de arranque y parada, lógica de seguridad, y más. Para esto se pueden combinar los distintos lenguajes de programación de PLC existentes definidos por la norma IEC 61131-3.

#### 4.2.7 Depuración del programa

La depuración de un programa de PLC es un proceso crítico en el desarrollo de sistemas de control industrial ya que garantiza que el programa se ejecute correctamente, sin errores ni fallos. En esta etapa se busca y se identifican posibles errores en el programa. Los errores pueden incluir problemas de lógica, errores de tipeo, malentendidos en las instrucciones, problemas de comunicación, configuración incorrecta de E/S y más.

#### 4.2.8 Realización de pruebas FAT

Al escoger sistemas de control basados en controladores programables para dar solución a procesos se enfrenta el problema de no tener la posibilidad de probar la lógica de programación hasta no hacer la puesta en marcha física lo que puede resultar en daños para los equipos.

La sigla FAT se refiere a la Prueba de Aceptación en Fábrica, la cual es una etapa importante en el proceso de compra y puesta en marcha de equipos o sistemas. Su objetivo principal es asegurarse de que cumplen con las especificaciones de diseño acordadas antes de su implementación. Lo ideal es que estas pruebas se realicen antes de que los equipos lleguen a las instalaciones del cliente. Esto permite identificar y resolver cualquier problema tanto en la ubicación del proveedor como antes de la producción, utilizando la experiencia técnica y los recursos disponibles en la fábrica. De esta manera, se evita cualquier contaminación o interferencia con las instalaciones del cliente y se mejora la calidad general del producto entregado al cliente.

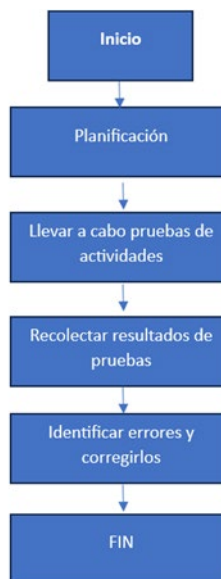
la aplicación de simuladores permite simular el comportamiento del proceso frente a diversas situaciones o escenarios como por ejemplo situaciones extremas para comprobar el correcto funcionamiento de los enclavamientos o códigos de seguridad.

De acuerdo con lo estudiado en [7] las ventajas más importantes de las pruebas FAT usando simuladores que aplican para este proceso son:

- Ahorro de tiempo permitiendo puesta en marcha más rápida del sistema de control.

- Verificación del proyecto total, revisión de la programación, observar posibles problemas de instalación y antes del desarme de control actual.
- Elimina defectos de software anterior a la instalación del sistema.
- Localizar y corregir problemas software más rápidamente en ambiente simulado.
- Reducir el riesgo de problemas de lógica de control al eliminar errores.
- Identificar y corregir problemas antes de que causen retrasos y aumentos de costo excesivo.
- Someter al sistema a pruebas de desconexión/reconexión controlada de sensores para detectar fallas, probar alarmas y rutinas de reinicio.
- Permitir setear los límites de alarma y seguridad en valores seguros típicos tal que no generen alarmas falsas (durante transientes de partida) o avalanchas de datos.
- Verificar si el sistema es capaz de capturar, almacenar, procesar y generar un reporte que explique el escenario de falla simulada.

Para garantizar una prueba de aceptación de fábrica exitosa, es importante comunicarse de manera efectiva durante todo el proceso, establecer expectativas claras y aclarar roles mucho antes de que se realice la FAT. Se propone el siguiente diagrama de flujo para pruebas FAT de alto nivel:



**Fig. 4.3 Diagrama de flujo para pruebas FAT**

Donde cada etapa se describe a continuación:

- **Planificación:** Se determina y documentan las pruebas que se llevarán a cabo, los criterios de prueba, protocolos, herramientas necesarias, errores a identificar y aceptabilidad. También es importante determinar el personal que debe estar



involucrado durante el proceso.

- Llevar a cabo pruebas de actividades: Se realizan pruebas de acuerdo con los requisitos y especificaciones planificados anteriormente.
- Recolectar resultados de pruebas: Se recopilan los resultados y se miden de acuerdo a lo planificado, se determina si se ha cumplido o no con los objetivos de las pruebas.
- Identificar errores y corregirlos: Se identifican las fallas, errores y peligros de acuerdo a los criterios determinados en la planificación, los problemas que necesiten solución se corrigen para su aceptación, este último paso puede requerir una nueva prueba.
- Fin: Se aprueba la aceptación de los equipos o sistemas dando por finalizadas las pruebas.

#### 4.2.9 *Puesta en marcha*

La puesta en marcha es la fase culminante del proyecto, donde se traslada la planificación y el diseño teórico a la realidad física del entorno industrial. Este proceso implica una serie de etapas críticas para garantizar que el sistema funcione de manera óptima y cumpla con las expectativas del cliente.

Previo al inicio de la puesta en marcha, se lleva a cabo una planificación exhaustiva que incluye la definición de protocolos de seguridad y el establecimiento de procedimientos para afrontar eventuales problemas durante la implementación, sin embargo la realización de pruebas FAT realizadas antes de la puesta en marcha contribuye a mitigar posibles riesgos.

Luego se lleva a cabo una exhaustiva verificación de todos los equipos, conexiones eléctricas y comunicaciones para asegurar que estén correctamente instalados y funcionando según las especificaciones. Esto incluye procedimientos como la revisión de E/S (Entradas/Salidas) del PLC y la conexión adecuada de sensores y actuadores.

Luego se configuran y ajustan los parámetros del sistema según los requisitos del cliente y las condiciones específicas del entorno. Esto puede incluir la configuración de umbrales de alarma y límites de operación, así como ajustes en los parámetros de sintonización del sistema de control.

Posteriormente se realiza la ejecución del programa de manera progresiva, lo que implica activar gradualmente las diferentes secciones del proceso. Se monitorea cada paso para garantizar que no haya anomalías y que el sistema responda como se espera. Además se verifica el funcionamiento de enclavamientos, respuestas a eventos de alarma, tiempos de respuesta y cualquier otra funcionalidad crítica.

Durante la puesta en marcha también se proporciona capacitación al personal que operará y mantendrá el sistema. Esto incluye la instrucción sobre el uso de la interfaz HMI, la interpretación de alarmas y la ejecución de procedimientos de emergencia.

Al finalizar la puesta en marcha, se genera la documentación final que incluye manuales de operación, listas de equipos, esquemas eléctricos actualizados y cualquier otra información relevante dando por finalizado el servicio.

## 5. Pantallas HMI

---

### 5.1. Introducción

Las pantallas HMI (Human-Machine-Interface) desempeñan un papel fundamental en la automatización industrial ya que sirven como ventana principal a través de la cual los operadores y técnicos interactúan con el sistema de control y monitorean los procesos en tiempo real. Estas interfaces gráficas permiten una comunicación efectiva entre humanos y máquinas, simplificando la supervisión y el control de equipos industriales complejos.

La importancia de abordar el diseño de pantallas HMI en esta memoria radica en su estrecha relación con la programación de PLC. A menudo, estos dos elementos suelen trabajar en conjunto para controlar y supervisar sistemas industriales complejos. Al desarrollar el servicio de programación PLC es también importante comprender cómo diseñar y desarrollar interfaces HMI efectivas.

### 5.2. Funciones principales de una HMI

Dentro de las funciones principales que podemos encontrar dentro de una HMI se encuentran:

- **Monitoreo:** dentro de una HMI podemos encontrar las variables de la planta en tiempo real los cuales pueden ser mostrados en gráficos, números, textos u otros elementos que faciliten el entendimiento al operador.
- **Alarmas:** Es importante que una HMI sea capaz de reconocer y reportar eventos excepcionales dentro del proceso para que el operador pueda tomar las acciones correctas para prevenir accidentes o mal funcionamiento del proceso.
- **Control:** Se refiere a la capacidad de aplicar algoritmos que regulan y mantienen los valores del proceso dentro de límites aceptables. Este enfoque de control busca reducir o incluso eliminar la necesidad de intervención humana en la toma de decisiones relacionadas con los procedimientos.
- **Supervisión:** Junto con el monitoreo se pueden lograr ajustes en las condiciones de trabajo del proceso directo desde la pantalla.
- **Histórico:** Es la capacidad de muestrear y almacenar datos del proceso, esta función es una herramienta esencial ya que puede ser utilizada para la optimización y corrección de procesos.

### 5.3. Diseño

El diseño de HMI se basa en la normativa ISA-101, y se enfoca en crear sistemas interactivos centrados en el usuario para mejorar la toma de decisiones y la seguridad en el entorno industrial, la normativa detallada se puede encontrar en el **Anexo 8.2**. Los principios básicos para el desarrollo de una HMI son mitigar los errores y optimizar las herramientas y elementos que conforman la HMI. El uso de colores, formas y tamaños de los elementos estáticos, fondo de pantalla, uso de líneas, representación de alarmas y elementos de control son importantes para hacer una representación lo más adecuada para el operador con la intención de facilitar su entendimiento y minimizar el esfuerzo visual ya que letras muy pequeñas o colores muy fuertes pueden ser molestos para la vista al estar varias horas frente a la pantalla. Además, se deben establecer pautas de diseño, estrategias de conducta y análisis para el procesamiento de la información suministradas por las HMI.

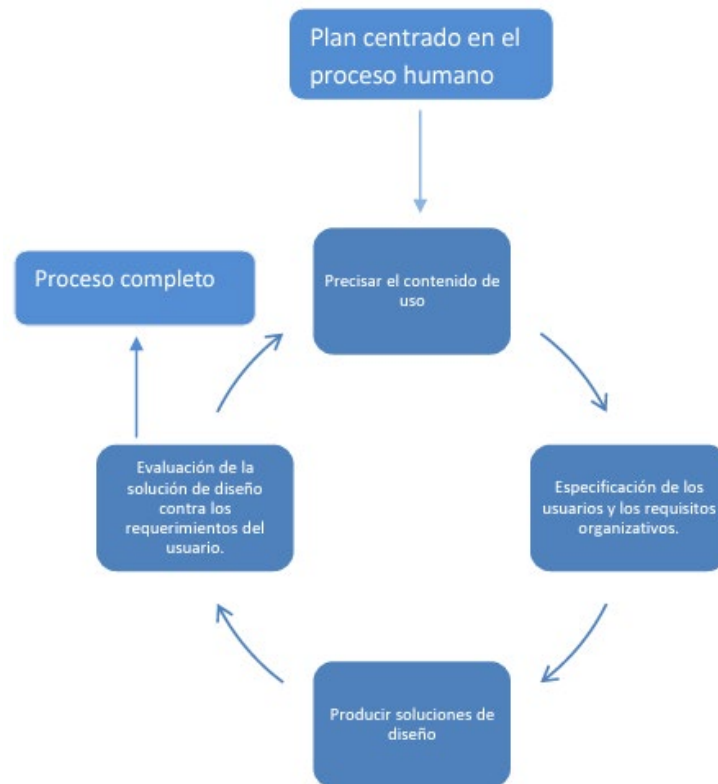
Una de las etapas más importantes para el diseño de HMI es el plan centrado en el proceso humano donde se recopilan los objetivos, compromisos, filosofías de control, metodologías, características específicas y requerimientos del cliente. Para esto es posible basarse en la norma **ISO 13470**, esta norma proporciona una guía sólida para garantizar que los sistemas y productos interactivos estén diseñados considerando las necesidades, capacidades y expectativas de los usuarios finales. Se enfoca en mejorar la usabilidad, la eficacia y la satisfacción del usuario.

Para desarrollar el servicio de acuerdo con la norma ISO 13470 se proponen las siguientes actividades:

- **Precisar el contenido de uso:** Al momento de diseñar una HMI es fundamental planificar las actividades necesarias para el desarrollo del servicio por lo que es importante comprender y determinar los detalles del contenido del diseño para proporcionar estrategias y bases para la evaluación del de la HMI. Para este punto es importante que la participación activa del cliente para precisar el contenido y los requerimientos del servicio.
- **Especificación de los usuarios y los requisitos organizativos:** En este punto se desarrollan los perfiles de los usuarios (visita, operador, programador, etc.) y se le otorgan los permisos correspondientes para operar o realizar modificaciones en las características del sistema.
- **Producir soluciones de diseño:** En esta etapa se materializa la HMI a partir de la información recopilada en los puntos anteriores permitiendo la comunicación de la HMI con el proceso.

- Evaluación de la solución de diseño:** En este punto se controla la calidad y usabilidad de la HMI diseñada. Se compara la solución elaborada durante el servicio con respecto a los requerimientos del cliente. Esta actividad se puede apoyar de simulaciones creando perturbaciones al sistema para observar si el comportamiento es el deseado. Si el diseño no cumple con los requerimientos del cliente entonces se debe volver a el precisamiento de las condiciones de uso para realizar los ajustes necesarios para cumplir con las necesidades del cliente. Si el diseño cumple con los requerimientos del cliente entonces se completará la etapa de diseño de HMI.

A continuación se detallan las actividades mencionadas con anterioridad a través de un diagrama de flujo:



**Fig. 5.1 Representación de sistemas interactivos centrados en humanos**

También es importante al momento del diseño considerar algunos de los principios básicos de diseño.

Es importante la claridad para facilitar la operación utilizando la HMI, que los gráficos sean entendibles, y los mensajes sean de fácil lectura, también las condiciones y estado actual del proceso. El contenido de la HMI debe ser preciso sin contener información irrelevante para el proceso, además las alarmas e indicadores deben ser claros, resaltados y distinguibles.

La HMI debe ser coherente, es decir que sus funciones establezcan un manejo lógico. Las funciones gráficas deben ser estandarizadas, sencillas y que sus funciones faciliten la manipulación de la HMI.

Además, de acuerdo con la norma ISA-101 el diseño debe ser enfocado en minimizar la fatiga del usuario, ya sea utilizando colores tenues, letras adecuadas para no realizar esfuerzo visual extra.

Por último, las acciones importantes del proceso como por ejemplo paradas de emergencias deben tener mecanismos de confirmación para evitar acciones equivocadas.

## 6. Ejemplos prácticos

---

### 6.1. Introducción

En este capítulo, se presentarán ejemplos prácticos destinados a ilustrar la teoría investigada a lo largo de la memoria de título. El propósito principal es apoyar el entendimiento de la programación de PLC, para esto se abordarán dos casos prácticos específicos, cada uno enfocado en aspectos clave de este campo.

El primer ejemplo se centrará en el diseño de un sistema de control para supervisar y controlar un proceso complejo. Se analizarán las estrategias de programación PLC empleadas para controlar un sistema altamente no lineal.

El segundo ejemplo se enfocará en la simulación de un servicio especializado en la programación de enclavamientos. Se expondrán realizaciones de pruebas FAT efectivas que validen y verifiquen el correcto funcionamiento de los enclavamientos en un entorno simulado.

Ambos ejemplos se abordarán de manera detallada, destacando los principios teóricos aplicados y proporcionando un análisis práctico que fortalezca la comprensión en relación con la aplicación de la programación PLC.

### 6.2. Proceso de neutralización química

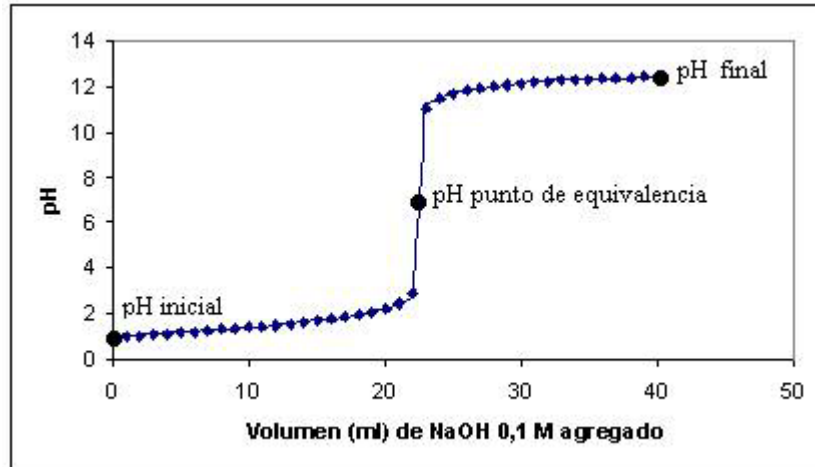
#### 6.2.1 Descripción del sistema

Diseñar y supervisar un proceso de neutralización química representa un desafío de considerable complejidad, demandando el entendimiento de los procesos químicos involucrados. En el marco de los lazos de control que serán abordados, se incluyen variables de proceso como el flujo, nivel y pH.

La neutralización del pH emerge como un procedimiento altamente delicado, caracterizado por la no linealidad del proceso. Esta complejidad se atribuye principalmente a la imposibilidad de medir directamente la cantidad de iones presentes. En su lugar, se recurre al pH como indicador, el cual refleja la acidez o alcalinidad de una solución. Cabe destacar que el pH está directamente vinculado a la concentración de iones de hidrógeno, denotados como  $[H^+]$  presentes en la solución.

$$pH = -\log_{10}[H^+] \quad (\text{Ec. 1.1})$$

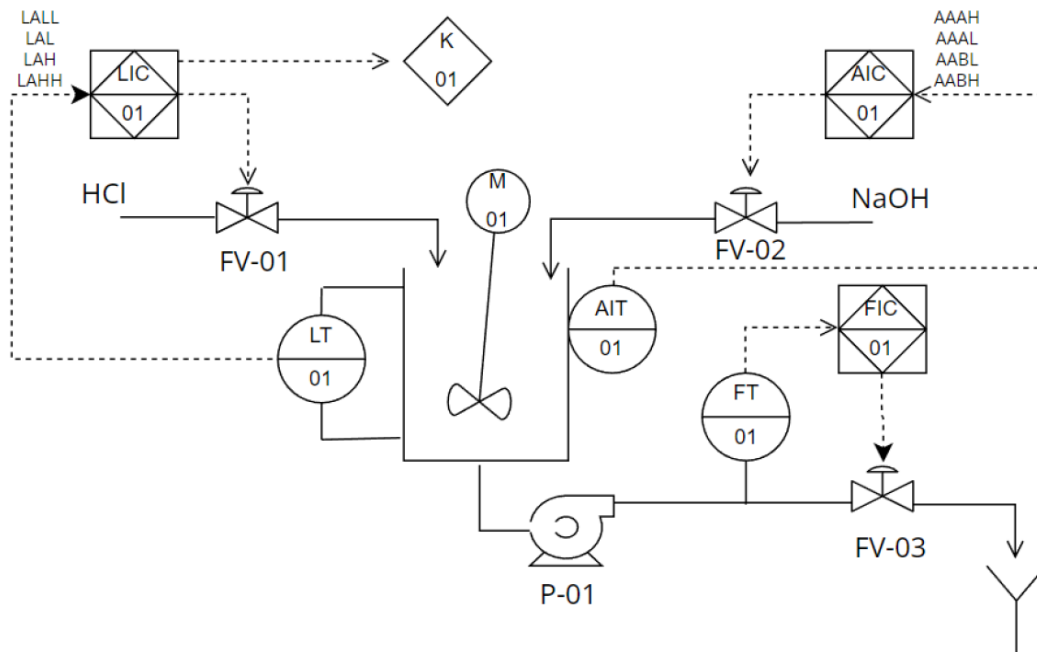
Esto no solo hace que el proceso sea altamente no lineal, sino que su relación logarítmica lo vuelve altamente sensible a pequeños cambios de concentración en torno al pH 7. La forma en "S" de la Figura X ilustra esta sensibilidad y no linealidad con una pendiente casi infinita en el punto de equivalencia.



**Fig. 6.1 Curva de pH de neutralización con NaOH**

A continuación se reúne información importante para la comprensión adecuada del proceso, así como datos esenciales que deben recopilarse en el rol de programador:

- P&ID: A continuación se muestra el P&ID del proceso.



**Fig. 6.2 P&ID Proceso de neutralización**

- Parámetros del Reactor: A continuación se muestran los parámetros de diseño del



Reactor

Tabla 6.1 Parámetros del reactor

Parámetro	Valor	Unidad de medida
Diámetro interno	11,48	<i>ft</i>
Long. Interna	18	<i>ft</i>
Volumen	0,08	<i>ft</i> <sup>3</sup>
Espesor de la Chaqueta	0.5	<i>ft</i>
Altura de la Chaqueta	18	<i>ft</i>

- Constantes del proceso: A continuación se presentan las constantes involucradas en el proceso.

Tabla 6.2 Parámetros del proceso

Parámetro	Valor	Unidad de medida
Concentración de HCl	0.0056	<i>mol/l</i>
Concentración de NaOH	0.0045	<i>mol/l</i>
Constante de disociación del agua	$1 \times 10^{-14}$	<i>mol</i> <sup>2</sup> / <i>l</i> <sup>2</sup>

### 6.2.2 Modelación del proceso

El proceso de neutralización utilizado en este proyecto es la neutralización de un ácido fuerte (*HCl*) en un reactor CSTR mediante una base fuerte (*NaOH*). Este proceso se modela utilizando un modelo de primer orden con la curva de titulación como no linealidad. Las reacciones que ocurren son:



La cantidad de efluente a neutralizar se puede definir por la concentración de los reactantes, hay que considerar que en el reactor la mezcla es perfecta e instantánea, así la concentración de iones de  $[Cl^-]$  y  $[Na^+]$  puede ser relacionada con las concentraciones de entrada al reactor dada las siguientes ecuaciones:

$$V \frac{d}{dt} [Cl^-] = [Cl^-]_{in} \cdot Q_a - [Cl^-] \cdot Q_{out} \quad (\text{Ec. 1.4})$$

$$V \frac{d}{dt} [Na^+] = [Na^+]_{in} \cdot Q_b - [Na^+] \cdot Q_{out} \quad (\text{Ec. 1.5})$$

Donde:

$V$ : Volumen del fluido al interior del reactor CSTR.

$Q_a$ =Tasa de flujo del ácido.

$Q_b$ =Tasa de flujo de la base.

$Q_{out}$ =Flujo de salida del reactor.

Las concentraciones de efluente también deben satisfacer la ecuación de electroneutralidad:

$$[Na^+] + [H^+] = [Cl^-] + [OH^-] \quad (\text{Ec. 1.6})$$

En conjunto con la ecuación de disociación para el agua:

$$[H^+] \cdot [OH^-] = K_w = 10^{-14} \quad (\text{Ec. 1.7})$$

Relaciona las concentraciones con la concentración de iones de hidrógeno  $[H^+]$  y, por lo tanto, con el pH. Esta relación se expresa en términos de la diferencia de las concentraciones iónicas X:

$$X = [OH^-] - [H^+] \quad (\text{Ec. 1.8})$$

Combinado con la ecuación (1.6) da como resultado:

$$X = [Na^+] - [Cl^-] \quad (\text{Ec. 1.9})$$

Luego, combinando las ecuaciones (1.7) y (1.8) da como resultado las ecuaciones:

$$[H^+] = \frac{X}{2} \left( \sqrt{1 + \frac{4K_w}{X^2}} - 1 \right) \quad \text{si } X > 0 \quad (\text{Ec. 1.10})$$

$$[H^+] = -\frac{X}{2} \left( \sqrt{1 + \frac{4K_w}{X^2}} + 1 \right) \quad \text{si } X < 0 \quad (\text{Ec. 1.11})$$

$$[H^+] = \sqrt{K_w} \quad \text{si } X = 0 \quad (\text{Ec. 1.12})$$

Así utilizando la ecuación (1.1) podemos obtener el pH en el reactor. Por último la ecuación que describe la dinámica del proceso es obtenida restando las ecuaciones (1.4) y (1.5) y utilizando la

ecuación (1.9) obteniendo como resultado:

$$V \frac{d}{dt} X = [Na_{in}^+] \cdot Q_b - [Cl^-] \cdot Q_a - X \cdot Q_{out} \quad (\text{Ec. 1.13})$$

Por otro lado el nivel del Reactor CSTR está modelado a través del balance de masa dentro del reactor:

$$\rho \frac{dV}{dt} = A \cdot \rho \cdot \frac{dh}{dt} = \rho_a Q_a + \rho_b Q_b - \rho Q_{out} \quad (\text{Ec. 1.14})$$

Donde:

$A$  = Área del reactor.

$\rho$  = Es la masa específica al interior del reactor.

$\rho_a$  = Es la masa específica del ácido.

$\rho_b$  = Es la masa específica de la base.

Para las válvulas de control FV-01, FV-02 y FV-03 se decidió modelar válvulas lineales, es decir que a un 10% de apertura haya 10% de caudal lo cual corresponde a 0.1 litros por metro cúbico, por lo que obtuvimos la siguiente función de transferencia de primer orden con retardo que modela el comportamiento dinámico de las válvulas:

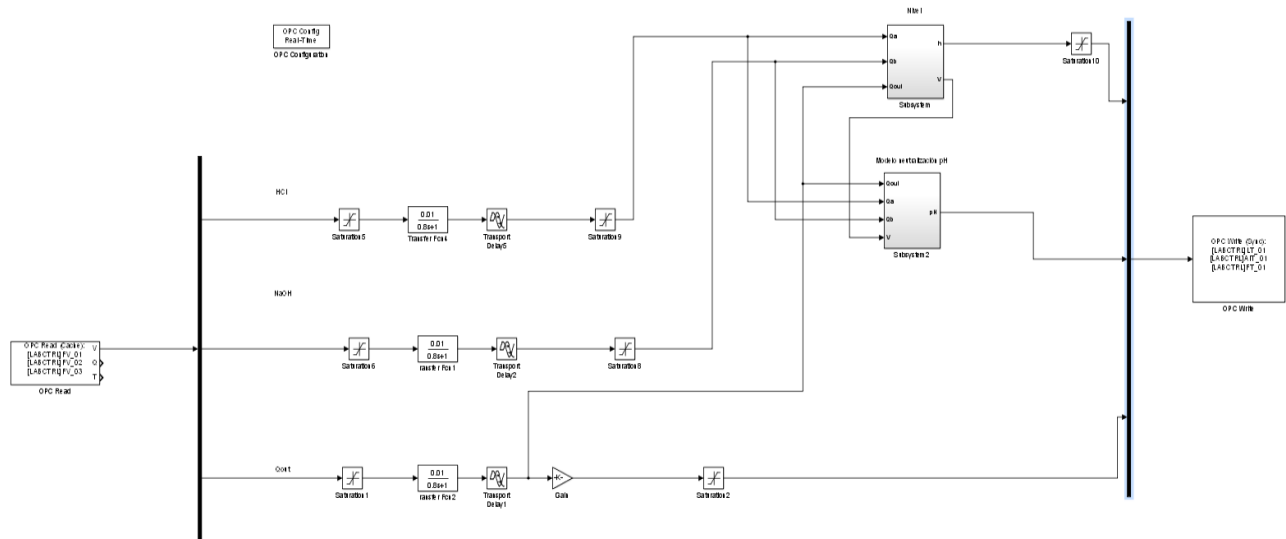
$$f_{valve}(s) = \frac{0.01}{0.8s + 1} e^{-0.4s} \quad (\text{Ec. 1.15})$$

### 6.2.3 Modelación del proceso con uso de simuladores

A partir del modelo matemático obtenido anteriormente se procede a modelar la dinámica del proceso en Simulink de Matlab el cuál será conectado vía OPC a Rslogix5000 para la programación PLC.

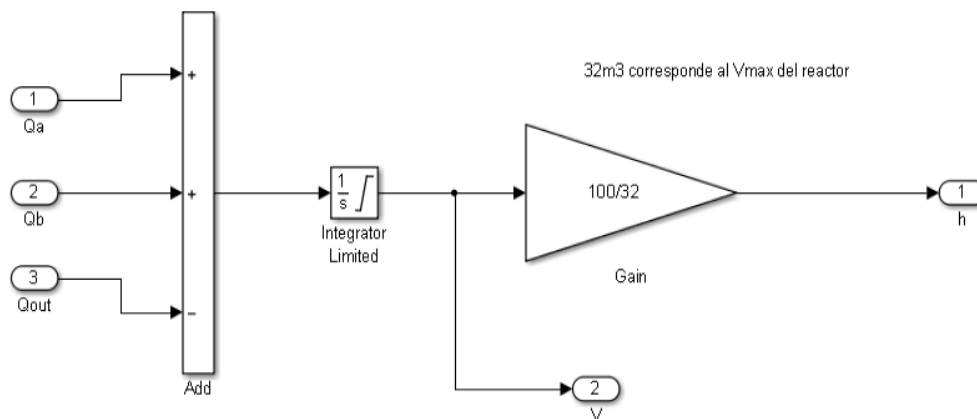
La conexión OPC entre RSLogix y Simulink permite la comunicación y el intercambio de datos entre estos dos entornos de programación y simulación. Para establecer una conexión OPC entre RSLogix y Simulink, se requiere un servidor OPC que actúe como intermediario entre los dos sistemas. El servidor OPC se encarga de obtener los datos del controlador RSLogix y ponerlos a disposición de Simulink, y viceversa. Una vez que se establece una conexión OPC, Simulink puede acceder a los datos del controlador RSLogix y usarlos en simulaciones de sistemas, como implementar y probar algoritmos de control o verificar sistemas de control industrial. De manera similar, RSLogix puede recibir datos de Simulink para usarlos en la programación y el control del sistema de automatización.

La conexión OPC entre RSLogix y Simulink permite la integración e integración de procesos y la comparación de las capacidades de los dos entornos, facilitando la creación y configuración de sistemas de control en entornos industriales. Así se obtuvo el siguiente modelo:



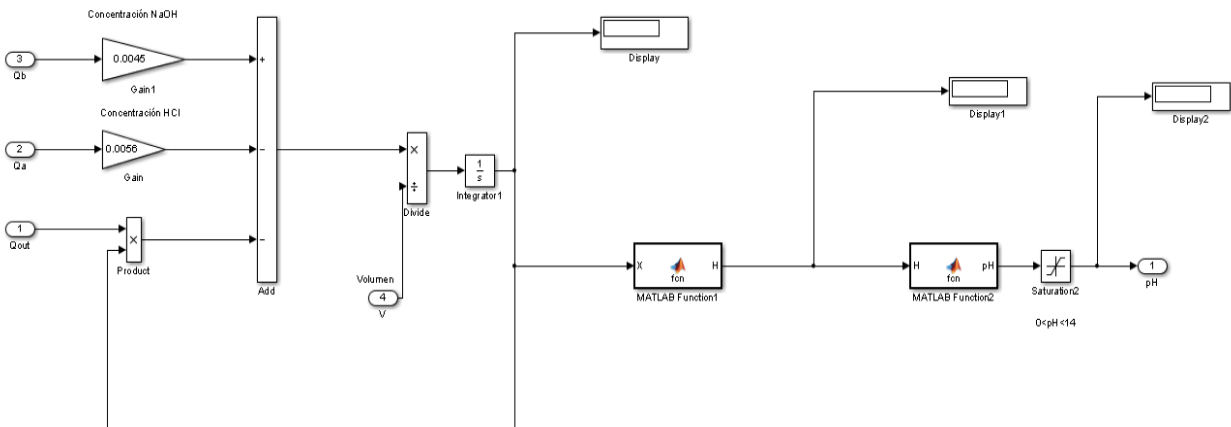
**Fig. 6.1 Vista general Modelo Simulink**

Donde dentro de los subsistemas tenemos el lazo de nivel:



**Fig. 6.4 Lazo de Nivel en Simulink**

Dentro del segundo subsistema tenemos el lazo de pH:



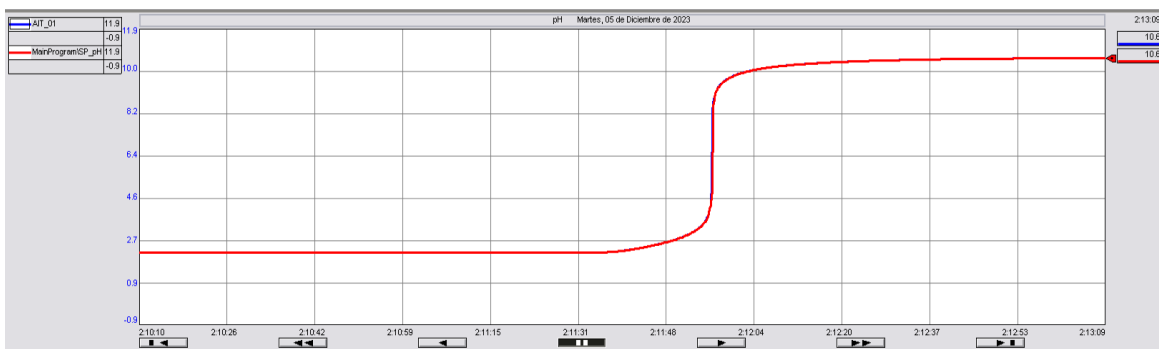
**Fig. 6.5 Lazo de pH en Simulink**

Dentro de las funciones Matlab se tiene una función que selecciona la ecuación adecuada dependiendo del valor de X y En el segundo bloque se tiene una función para calcular el valor del pH a partir de los iones de hidrógeno, el contenido de estos bloques se encuentra detallado en el Anexo.

#### 6.2.4 Pruebas en Lazo Abierto

Las pruebas en lazo abierto en un proceso de pH se llevan a cabo con el propósito de observar la dinámica del sistema y comprender su respuesta ante cambios en las variables de entrada, como la concentración de iones de hidrógeno ( $[H^+]$ ). Estas pruebas permiten identificar parámetros clave del modelo del proceso, como la velocidad de reacción ante adiciones de una base fuerte. La información obtenida de estas pruebas en lazo abierto es fundamental para diseñar estrategias efectivas de control, optimizar los puntos de operación del sistema y asegurar un rendimiento robusto y preciso en la regulación del pH.

Para esto se realizó una adición de un 50% de flujo de carga de NaOH obteniendo la siguiente respuesta en el pH:



**Fig. 6.6 Prueba en Lazo Abierto**

Mediante la prueba en Lazo Abierto se ha podido observar que el sistema exhibe un pH mínimo de 2.6 y un pH máximo de 10.6. Además, se observa una pendiente casi infinita en el rango de pH 3 a 9. En vista de esta alta ganancia de la planta en estos intervalos, resulta necesario diseñar un controlador con una ganancia adecuada que compense correctamente dicha dinámica del sistema.

### 6.2.5 Diseño del sistema de control

Para este proyecto se decidió utilizar el control PI en el diseño del sistema de control para el lazo de flujo, nivel y pH en el estanque CSTR del proceso de neutralización debido a su capacidad para manejar sistemas lineales y su robustez ante perturbaciones y variaciones en las condiciones de operación. El control PI proporciona un enfoque versátil y confiable para regular y mantener los niveles de flujo y nivel deseados, permitiéndome ajustar los parámetros del controlador para lograr una respuesta óptima en función de las características específicas del proceso de neutralización.

El diseño de un sistema de control PI para un lazo de control de posición y flujo de un tanque CSTR en modo neutral incluye la implementación de un control proporcional-integral (PI) para controlar y mantener el flujo y la posición del tanque. El objetivo principal del control PI es generar correctamente la salida del sistema de control para mantener el flujo, nivel y pH dentro del reactor deseados, independientemente de las perturbaciones y las condiciones de entrada cambiantes.

A continuación se muestra el diagrama SAMA que detalla el sistema de control diseñado para este ejemplo práctico de neutralización química:

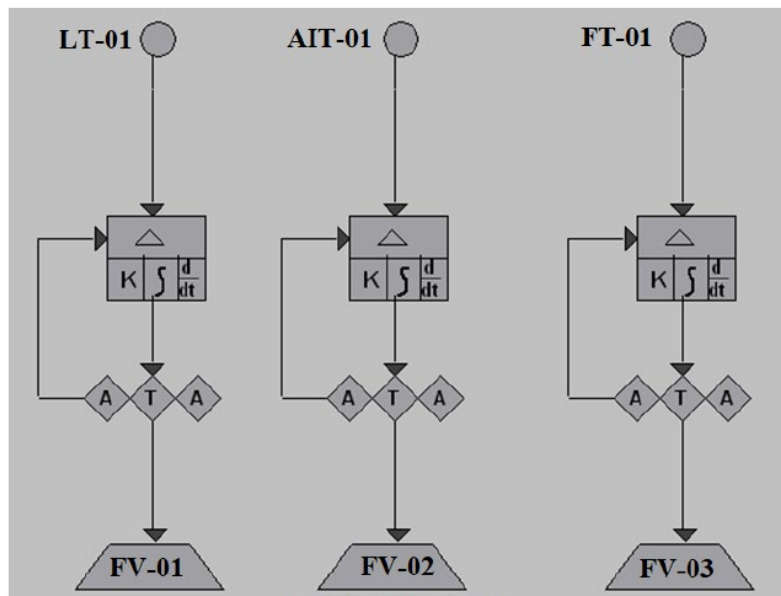
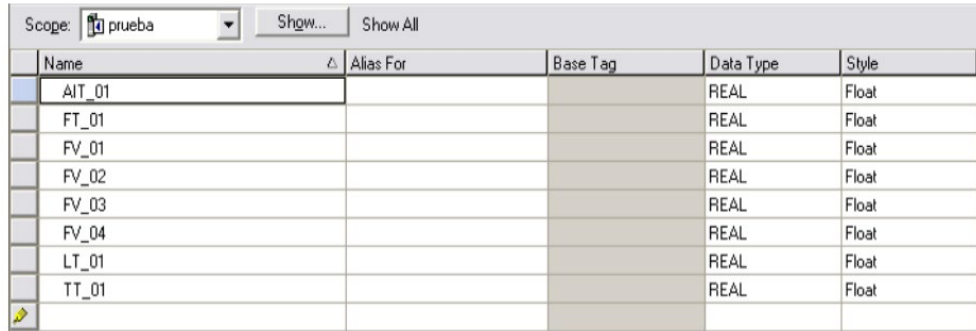


Fig. 6.7 Diagrama SAMA

### 6.2.6 Programación PLC

Para la programación PLC del sistema de control de este proceso se utilizaron dos lenguajes de programación, Ladder y texto estructurado, además de bloques Add-on propios de RsLogix.

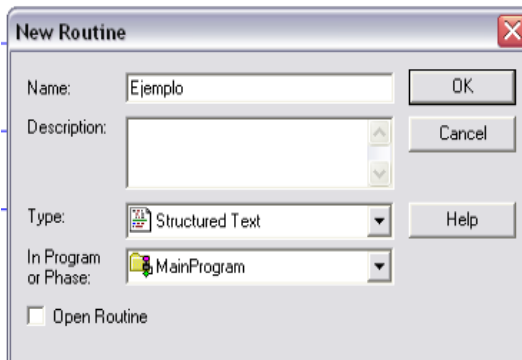
El primer paso es ingresar los Controller tags, las cuales son nuestras variables de escritura y lectura que están enlazadas con los bloques de OPC de Simulink.



Name	Alias For	Base Tag	Data Type	Style
AIT_01			REAL	Float
FT_01			REAL	Float
FV_01			REAL	Float
FV_02			REAL	Float
FV_03			REAL	Float
FV_04			REAL	Float
LT_01			REAL	Float
TT_01			REAL	Float

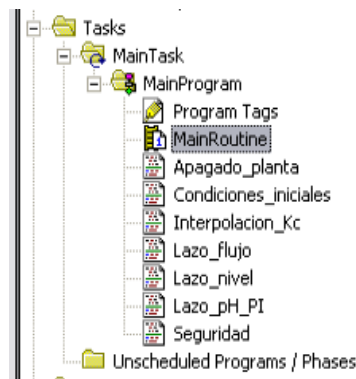
**Fig. 6.8 Controller Tag's**

Con los controller tags procedemos a crear textos estructurados para programar el controlador de cada lazo:



**Fig. 6.9 Creación de nueva rutina en Rslogix5000**

Obteniendo así la siguiente vista:



**Fig. 6.10 Textos Estructurados**

Luego para programar el controlador para cada lazo se deja como ejemplo la programación del controlador del lazo de flujo, el código completo se encuentra en el Anexo.

- **Coefficientes equivalentes discretos PID:** Se calcula los coeficientes utilizados en la acción de control del controlador PID. Los coeficientes se calculan en función de las ganancias proporcionales ( $k_c\_f$ ), integrales ( $T_i\_f$ ) y derivativas ( $T_d\_f$ )
- **Cálculo del error:** Calcula el error entre el valor de referencia ( $SP\_f$ ) y el valor de la variable proceso PV ( $FT\_01$ ).
- **Acción de Control:** Multiplica el error ( $ek\_f$ ) por el switch de acción de control de flujo ( $sw\_acc\_f$ ) para obtener la acción de control correspondiente.
- **Banda de ruido:** Verifica si el valor absoluto del error está dentro de una banda de ruido predefinida ( $Nb\_f$ ). Si es así, se establece el error a cero para evitar una acción de control innecesaria.
- **Cálculo del delta de control:** Calcula el cambio en la acción de control ( $duk\_f$ ) utilizando los coeficientes calculados anteriormente y los valores previos del error y la salida del proceso ( $yk1\_f$ ,  $yk2\_f$ ).
- **Cálculo del signo de duk:** Determina el signo (positivo, negativo o cero) del cambio en la acción de control.
- **Limitador de variación:** Limita el cambio en la acción de control ( $duk\_f$ ) para evitar cambios bruscos y mantenerlo dentro de un rango máximo permitido ( $dumax\_f$ ).
- **Set point tracking:** Si se activa el interruptor ( $sw\_am\_f$ ), actualiza el valor de salida del actuador ( $FV\_03$ ) sumando el cambio en la acción de control ( $duk\_f$ ) al valor previo. Si se cumple una condición específica ( $sw\_st\_f$ ), restablece el valor del actuador ( $uk1\_f$ ) al valor de referencia ( $SP\_f$ ).
- **Limitador de mínimo y máximo en el actuador:** Limita el valor de salida del actuador ( $FV\_03$ ) para asegurarse de que esté dentro de los límites mínimo ( $umin\_f$ ) y máximo ( $umax\_f$ ) permitidos.
- **Memoria del actuador:** Almacena el valor de salida del actuador ( $FV\_03$ ) para su uso en el próximo ciclo de control.
- **Memoria del error:** Almacena los valores anteriores del error ( $ek\_f$ ) para su uso en el próximo ciclo de control.
- **Memoria del PV:** Almacena los valores anteriores del PV ( $FT\_01$ ) para su uso en el



próximo ciclo de control.

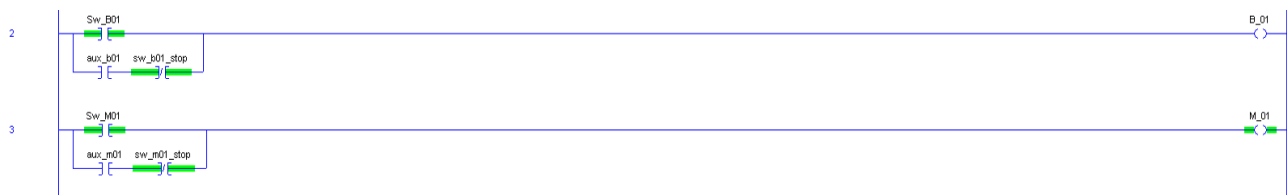
Una vez programados los textos estructurados se procede a agregarlos al diagrama Ladder el cual se puede encontrar en el Anexo.

Adicionalmente se programó un Time Scheduler utilizando un Add-on de RsLogix5000 para cada lazo, a continuación se explica brevemente el código, sin embargo el código final se encuentra en el Anexo de este documento.

- **"if ContOut\_seg<=0 then sw\_TimeScheduler:=1;"** - Esta línea de código verifica si la variable "ContOut\_seg" es menor o igual a cero. Si esto se cumple, significa que el tiempo programado ha expirado y establece la variable "sw\_TimeScheduler" en 1.
- **"ContIn\_seg:=Delta\_Seg;"** - Esta línea de código asigna el valor de la variable "Delta\_Seg" a la variable "ContIn\_seg".
- **"else sw\_TimeScheduler:=0; end\_if;"** - Si la condición en la línea 1 no se cumple, es decir, "ContOut\_seg" es mayor que cero, la variable "sw\_TimeScheduler" se establece en 0, lo que indica que no se debe activar ninguna tarea programada en ese momento.
- **"ContOut\_seg:=ContIn\_seg-Tm\_lazo;"** - En esta línea de código se calcula el tiempo restante para la próxima activación programada. Resta el valor del tiempo de muestreo del lazo "Tm\_lazo" de la variable "ContIn\_seg" y guarda el resultado en la variable "ContOut\_seg". Esto permite mantener un seguimiento del tiempo hasta la siguiente activación.

Finalmente, se fijan condiciones iniciales donde se le asignan valores a las bandas de ruido, se procede a dejar los lazos en automático o manual según se desee, se fijan valores de Set-Point, se define la tasa de variación máxima para el actuador, valores mínimos y máximos de salida entre otros parámetros clave. El código se puede encontrar en el Anexo de este documento.

En el diagrama Ladder se procede a programar un enclavamiento para el correcto encendido y apagado de la bomba B-01 y el motor M-01 los cuales poseen botones de partida y parada los cuáles son momentáneos, por lo que para mantener el estado de la bomba o del motor encendidos y apagados se realiza la siguiente lógica de enclavamientos en el Ladder:



**Fig. 6.11 Enclavamientos en diagrama Ladder**

Donde aux\_b01 y aux\_m01 contienen el valor actual de la bomba y del motor respectivamente. Así también se programa un código en texto estructurado, el cuál se puede encontrar en el anexo de este documento, para que los equipos puedan ser encendidos y apagados dependiendo de las condiciones operacionales, en este caso para la bomba de descarga B-01 se necesita que haya nivel en el estanque y que la válvula de descarga FV-03 este abierta y para el caso del motor M-01 se necesita que haya nivel en el estanque y que haya flujo de carga. Obteniendo así la siguiente lógica de enclavamiento:

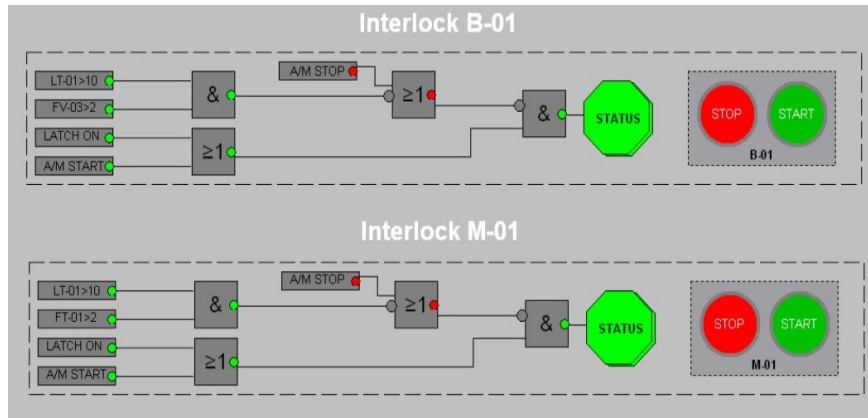


Fig. 6.12 Enclavamientos B-01 y M-01

Donde se deben cumplir las condiciones lógicas para poder encender los equipos y en caso contrario los equipos se apagarían y no se podrían volver a encender hasta que se cumplan las condiciones operacionales.

Una vez programados los textos estructurados y el diagrama Ladder se procede a depurar el programa para verificar que no haya error de sintaxis para así proceder a la etapa de sintonización.

```

Verifying routine Condiciones_iniciales of Program MainProgram
Verifying routine Interpolacion_Kc of Program MainProgram
Verifying routine Lazo_flujo of Program MainProgram
Verifying routine Lazo_nivel of Program MainProgram
Verifying routine Lazo_pH_PI of Program MainProgram
Verifying routine MainRoutine of Program MainProgram
Verifying routine Seguridad of Program MainProgram
Accepting the Pending Edits of program 'MainProgram'...
Accepting the Pending Edits of routine 'Seguridad' in program 'MainProgram'...
Testing edits for program 'MainProgram' .....
Assembling edits for program 'MainProgram'....
Complete - 0 error(s), 0 warning(s)

```

Fig. 6.13 Depuración en Rslogix5000

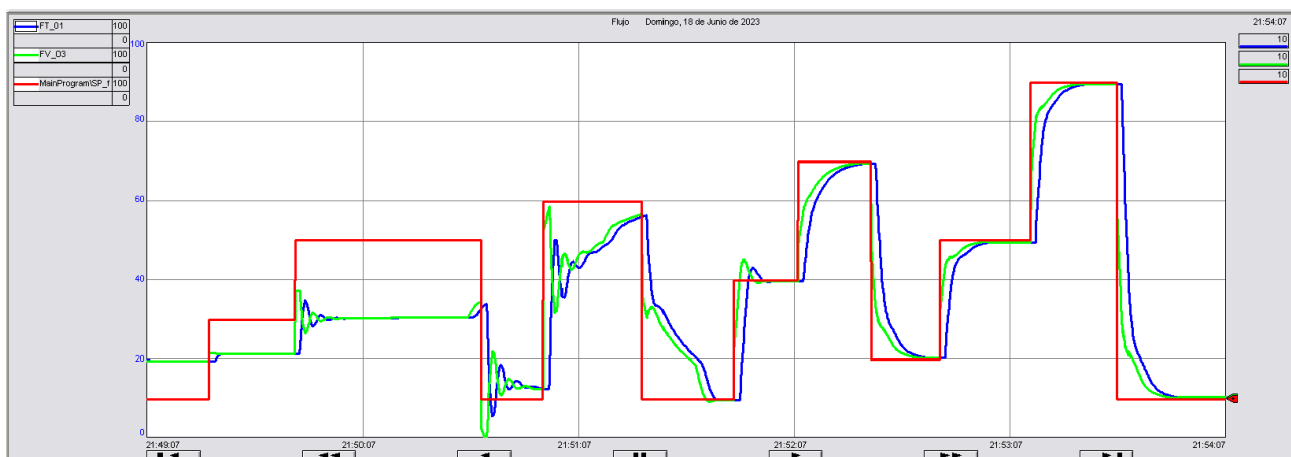
### 6.2.7 Sintonización

La sintonización PI se refiere al proceso de ajustar los parámetros del controlador proporcional

integral (PI) para lograr un rendimiento óptimo en un sistema de control. Consiste en determinar los valores adecuados de la ganancia proporcional ( $K_c$ ) y el tiempo integral ( $T_i$ ). Para esto ocuparemos el método Ultimate o también conocido como prueba y error donde el procedimiento es el siguiente:

- a) Quitar toda acción derivativa e integral del controlador ( $T_d=0$ ,  $T_i = \infty$ ).
- b) Poner una baja ganancia, o lo que es equivalente, una alta banda proporcional (del orden de 300%).
- c) Aumentar la ganancia (por ejemplo en factores de 2) y hacer una prueba de respuesta al escalón.
- d) Repetir c) hasta que la respuesta sea oscilatoria (casi) sin amortiguación.
- e) Disminuir la ganancia a la mitad del valor obtenido en d).
- f) Poner acción integral comenzando desde el máximo valor del controlador y observar la respuesta escalón.
- g) Reducir  $T_i$  (en factores de 2) hasta que la respuesta sea muy oscilatoria.
- h) Aumentar a  $T_i$  al doble del valor determinado en 9).
- i) Poner acción derivativa hasta que el ruido aparezca en la salida del controlador.
- j) Reducir  $T_D$  a la mitad del valor determinado en i).
- k) Varíe la ganancia en torno al valor puesto y examine la respuesta hasta obtener una aceptable.

Así aplicando el método ultimate se obtuvieron los siguientes resultados para el lazo de flujo:



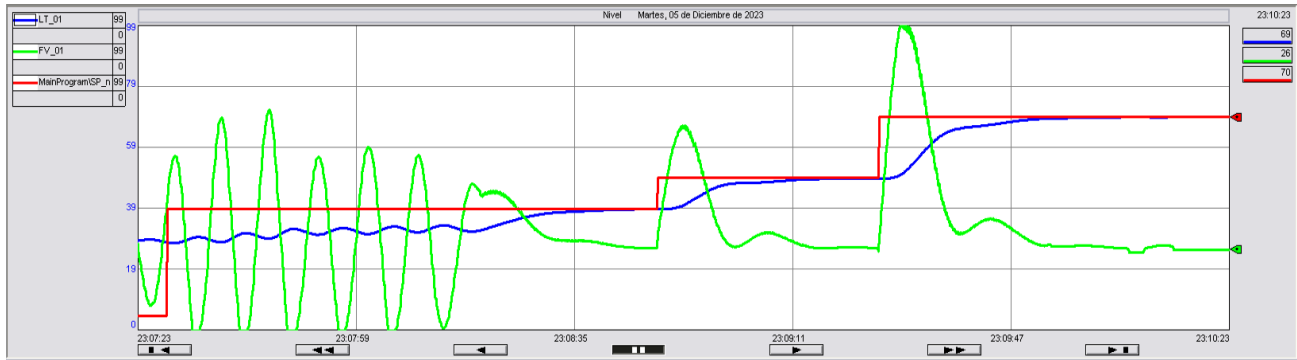
**Fig. 6.14 Sintonización Lazo de Flujo**

$$k_{c_f} = 0.4 \quad T_{i_f} = 1.5$$

Donde se puede observar el proceso de sintonización a lo largo de la gráfica, donde se inicia

con alto tiempo integrativo y una baja ganancia, posteriormente se aumenta la ganancia por lo que se observa un pequeño sobrepaso el cuál fue compensado disminuyendo  $K_c$  hasta obtener la respuesta deseada.

Análogamente para el lazo de Nivel

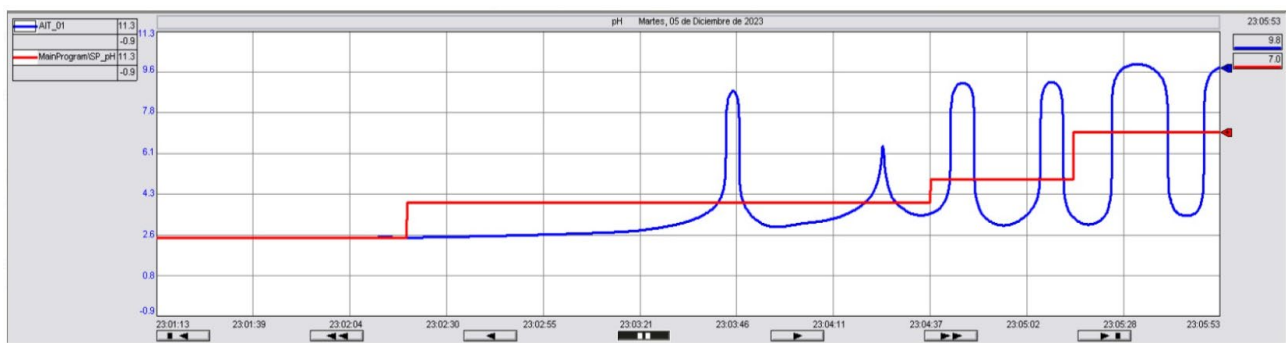


**Fig. 6.15 Sintonización Lazo de Nivel**

$$k_{c_n} = 12 \quad T_{i_f} = 20$$

En el caso del lazo de pH se necesita una ganancia  $K_c$  y  $T_i$  variable para poder compensar las zonas de alta ganancia del proceso, por lo que se procedió a sintonizar por zonas con el método ultimate ocupando Gain-Scheduler y el método de interpolación para que al cambiar de una zona a otra mediante interpolación lineal estimar  $K_c$  para puntos intermedios entre las zonas conocidas, proporcionando así una transición suave y continua. Para el caso del  $T_i$  se decidió ocupar el método por tabla.

Si no se ocupara Gain-Scheduler y el método por tablas y mantuviéramos un  $K_c$  y  $T_i$  constante obtendríamos una respuesta oscilatoria al entrar en la zona de alta ganancia del proceso como se puede observar a continuación:



**Fig. 6.16 Lazo de pH sin Gain-Scheduler**

Por lo que se procedió a sintonizar por zonas ocupando el método ultimate obteniendo así los siguientes valores:

**Tabla 6.3 Parámetros de sintonización Kc por zona**

Zona pH	Kc
1 - 3	8
3 - 3.5	4
3.5 - 4	1
4 - 4.5	0.3
4.5 - 5.5	0.2
5.5 - 6	0.1
6 - 7	0.08
7 - 9	0.5
9 -14	1

En el caso de Ti se obtuvieron los siguientes valores:

**Tabla 1.4 Parámetros Ti de sintonización por zona**

Zona pH	Ti
1 - 5	25
5 - 6	28
6 - 7	20
7 - 8	35
8 - 9	30
9 - 14	25

Obteniendo así la siguiente sintonización:



**Fig. 6.17 Lazo de pH sintonizado**

Donde se puede observar como el ácido logra ser neutralizado en un pH=7 con algunos

sobrepasos dentro del rango aceptable y finalmente se procede al apagado de planta donde se detiene el proceso y se vacía el reactor por lo que el pH oscila entre 6.5 y 7 ya que se encuentra en un ambiente neutro.

### 6.2.8 Realización de pruebas FAT

Una de las ventajas de los simuladores es que podemos hacer pruebas FAT de forma segura, podemos probar que la lógica de control funcione correctamente, así como también los códigos de seguridad en situaciones extremas.

La seguridad en una planta es de suma importancia para garantizar la protección de los equipos, la integridad del proceso y la seguridad del personal involucrado. Para los lazos de flujo, nivel y pH, se implementarán enclavamientos y alarmas como medidas de seguridad para prevenir situaciones peligrosas y mitigar riesgos. Para los lazos de Nivel y pH tenemos las siguientes alarmas en distintos niveles:

**Tabla 6.5 Alarmas de Nivel**

<b>Alarmas Nivel</b>	<b>Rango [%]</b>
Nivel bajo-bajo (LALL)	0-20
Nivel bajo (LAL)	20-50
Nivel alto (LAH)	50-95
Nivel Alto Alto (LAHH)	95-100

**Tabla 6.6 Alarmas de pH**

<b>Alarmas pH</b>	<b>Rango</b>
Nivel ácido alto (AAAH)	2-4
Nivel ácido medio (AAAL)	4-6
Nivel básico medio (AABL)	8-9
Nivel básico alto (AABH)	9-10

Y también tenemos el siguiente enclavamiento de seguridad:

**Tabla 6.7 Enclavamiento de seguridad**

<b>Enclavamiento</b>	<b>Evento</b>	<b>Acción</b>
K-01	Alto Nivel	Detención completa del proceso y de todos los equipos

Para la realización de pruebas FAT se seguirá el diagrama de flujo propuesto anteriormente en el capítulo 4 Para verificar que la lógica de nuestro código de seguridad funciona correctamente.

- **Planificación:** Se realizarán pruebas de llenado del reactor para verificar correcto funcionamiento de código de seguridad en caso de posible rebalse del reactor, posteriormente se neutralizará el ácido fuerte (HCl) para verificar la correcta sintonización de cada lazo. Los errores a identificar son irregularidades en el funcionamiento de las acciones de control, mala sintonización y mal funcionamiento del código de seguridad.
- **Llevar a cabo pruebas de actividades:** Se realizaron las pruebas planificadas anteriormente a través de su HMI correspondiente.
- **Recolectar resultados de pruebas:** Se recopilaron los resultados guardando los gráficos del proceso de neutralización y de las pruebas del código de seguridad para ser analizados posteriormente.
- **Identificar errores y corregirlos:** Se identificaron errores de tipeo en el código los cuáles producían acciones peligrosas para la válvula por lo que se tomaron acciones para corregir los errores de programación, por otro lado el código de seguridad funcionó correctamente.
- **Fin:** Se aprueba la aceptación del sistema dando por finalizadas las pruebas

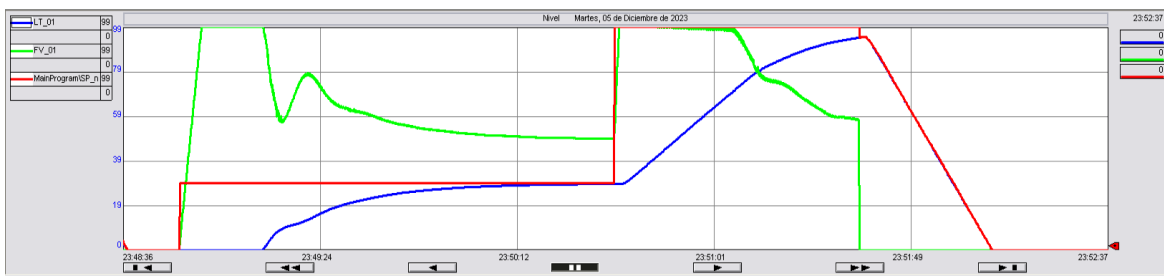


Fig. 6.18 Lazo de nivel con activación de enclavamiento K-01

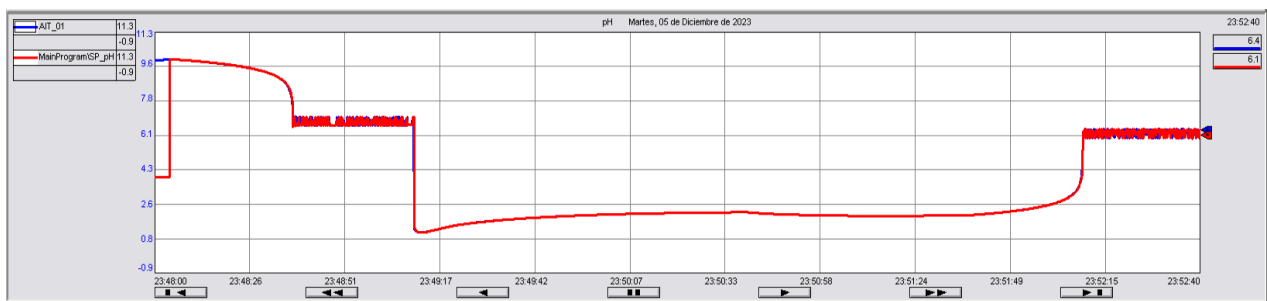
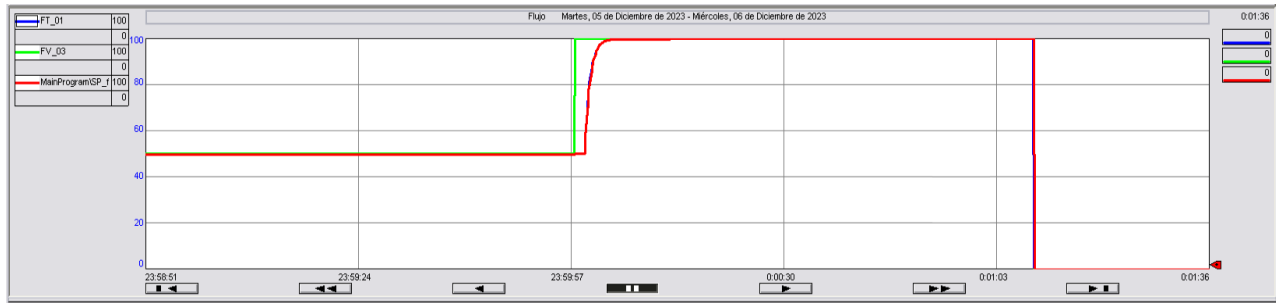


Fig. 6.19 Lazo de pH con activación de enclavamiento K-01



**Fig. 6.20 Lazo de Flujo con activación de enclavamiento K-01**

En las figuras expuestas anteriormente se puede observar que antes de que el nivel del reactor llegue a 100% el lazo de nivel y de flujo pasan a manual y la válvula de descarga procede a abrirse al 100% hasta vaciar el estanque. Finalmente una vez vaciado el reactor la válvula FV-03 procede a cerrarse. Por lo tanto se puede concluir que la lógica del código de seguridad funciona correctamente.

### 6.2.9 Pantalla HMI

Con el propósito de presentar los resultados obtenidos, se ha desarrollado una Interfaz Gráfica de Usuario (HMI) mediante FactoryTalk View Studio. Esta HMI permite visualizar gráficos de tendencia, así como representaciones gráficas de cada lazo individual con sus parámetros de sintonización respectivos. Esto posibilita el control del proceso directamente desde la pantalla, brindando una interfaz intuitiva para monitorear y ajustar los parámetros en tiempo real. Sin embargo, el diseño de HMI's será abarcado en el siguiente ejemplo práctico. Las imágenes detalladas de esta pantalla se encuentran disponibles en el anexo adjunto a este documento.

### 6.2.10 Puesta en marcha

Con las pruebas FAT completadas y aprobadas, se logró la presintonización de los lazos, validando de manera efectiva el funcionamiento adecuado de los enclavamientos y códigos de seguridad. En este punto el único paso pendiente para concluir el servicio es la fase de puesta en marcha, la cual se ejecutará según lo detallado en la sección 4.2.9 del documento.

Como parte de la entrega final, se proporcionará la documentación necesaria, incluyendo el programa Rslogix o Studio5000 en formato .ACD y para la HMI se entregará un archivo en formato .MER.



### **6.3. Planta de tratamiento de RILES**

#### *6.3.1 Descripción del problema*

En este segundo ejemplo práctico, se desarrollará un proyecto específico que destaca la importancia y la aplicabilidad directa de la programación PLC en entornos industriales. Enfocándonos en una planta de tratamiento de RILES, abordaremos la programación de alarmas y enclavamientos para la gestión eficiente de bombas. Este proyecto se centra especialmente en los estanques de almacenamiento TK-10A/B/C/D y los reactores RC-03/04/05/06, el P&ID se puede encontrar en el Anexo de este documento. La tarea clave será la programación de los enclavamientos que regulan el encendido y apagado de las bombas de acuerdo a las condiciones operacionales, garantizando un funcionamiento seguro y eficiente de la planta. Además, desarrollaremos una interfaz personalizada diseñada para satisfacer los requisitos específicos del cliente.

#### *6.3.2 Filosofía de control y operación.*

En el contexto de la filosofía de control, los RILes filtrados provenientes del Filtro Rotatorio (FL-01) serán dirigidos a los Tanques de Almacenamiento de RILes (TK-10A/B/C/D/E), equipados con sistemas de medición y transmisión de nivel (LE/LT-03/04/05/06/07) para prevenir posibles sobrelLENADOS. En cuanto al tratamiento en los Reactores de RILes Simples (RC-03/04), se llevará a cabo en modo batch, recibiendo RILes desde los filtros (FL-01/02), desde CITA/RSU, desde el Separador de Fases, o desde los Tanques mencionados.

Los RILes se descargarán desde los Tanques (TK-10A/B/C/D/E) hacia los reactores por medio de la Bomba de Descarga Almacenamiento RILes (P-15). Dicha bomba tendrá el Enclavamiento (I-05) cuando se detecte alto nivel en los Reactores (RC-03/04/05/06) o cuando exista bajo nivel en los Tanques (TK-10A/B/C/D/E).

Los Reactores (RC-03/04) contarán con un medidor y transmisor de nivel (LE/LT-08/09), que detendrá la bomba correspondiente (P-01, P-02, P-03, P-04, P-15, dependiendo del origen del RIL), cuando se llegue al nivel establecido de llenado según el tratamiento de cada RIL.

Por otra parte, los reactores (RC-05/06) contarán con un medidor y transmisor de nivel (LE/LT-10/11), que detendrá la bomba correspondiente (P-01, P-02, o P-05, dependiendo del origen del RIL), cuando se llegue al nivel establecido de llenado según el tratamiento de cada RIL.

Se considera también la detención de las bombas P-27/28/29/30/31/32/33 por alto nivel en los reactores (RC-03/04/05/06) (Enclavamiento I-06).

Además, cada Reactor (RC-03/04/05/06) contará con un instrumento de medición y transmisión de pH (AE/AT-02/03/04/05) y contarán también con una medición y transmisión de temperatura (TE/TIT-01/02/03/04), en conjunto con la protección ante el proceso de vainas protectoras de sondas de temperatura (TW-01/02/03/04), para monitorear el comportamiento del RIL tratado.

Se considera el Enclavamiento (I-15) cuando exista alta temperatura en los reactores (RC-03/04/05/06).

Junto con lo anterior, se tiene el encendido y apagado remoto de las Bombas (P-28/30/31) desde la Pantalla HMI PTFQ.

Así se pueden resumir los enclavamientos en la siguiente tabla:

**Tabla 6.8 Enclavamientos**

<b>Enclavamiento</b>	<b>Descripción</b>
I-01	Bombas P-01/02 se detendrán cuando exista alto nivel en Tanque de Almacenamiento de RILes (TK-10A/B/C/D/E).
I-02	Bombas P-01/02 se detendrán cuando exista alto nivel (de RIL) en los Reactores (RC-03/04/05/06)
I-05	Bombas P-03/04 se detendrán cuando exista nivel alto (de RIL) en Reactores (RC-03/04)
I-05	Bomba P-15 se detendrá cuando exista bajo nivel en Tanque de Almacenamiento de RILes (TK-10A/B/C/D/E).
I-06	Bomba P-15 se detendrá cuando exista alto nivel en los Reactores (RC-03/04/05/06).
I-06	Bomba P-27/28/29/30/31 se detendrá cuando exista alto nivel en los Reactores (RC-03/04/05/06).
I-07	Bomba P-28 se detendrá cuando el pH en el Reactor (RC-03/04/05/06) se encuentre entre 6 - 8.
I-09	Bomba P-30 se detendrá cuando el pH en el Reactor (RC-03/04/05/06) se encuentre entre 6 - 8.
I-10	Bomba P-31 se detendrá cuando el pH en el Reactor (RC-03/04/05/06) se encuentre entre 6 - 8.
I-15	Bomba P-28/29/30/31 se detendrá cuando exista alta temperatura en los Reactores RC-03/04/05/06.

Y además, también se cuenta con el siguiente listado de alarmas:

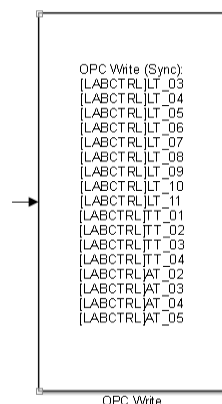
**Tabla 6.9 Alarmas**

<b>Descripción</b>	<b>Alarma</b>	<b>Valor</b>	<b>Enclavamiento</b>
Nivel en Tanque de Almacenamiento de RILes (TK-10A).	LAH-03	Alto	I-01
	LAL-03	Bajo	I-05
Nivel en Tanque de Almacenamiento de RILes (TK-10B).	LAH-04	Alto	I-01
	LAL-04	Bajo	I-05
Nivel en Tanque de Almacenamiento de RILes (TK-10C).	LAH-05	Alto	I-01
	LAL-05	Bajo	I-05
Nivel en Tanque de Almacenamiento de RILes (TK-10D).	LAH-06	Alto	I-01
	LAL-06	Bajo	I-05
Nivel en Tanque de Almacenamiento de RILes (TK-10E).	LAH-07	Alto	I-01
	LAL-07	Bajo	I-05
Nivel (de RIL) en Reactor de RILes Simples (RC-03)	LAH-08	Alto	I-01, I-02, I-06 y I-15
	LAL-08	Bajo	-
Nivel (de RIL) en Reactor de RILes Simples (RC-04)	LAH-09	Alto	I-01, I-02, I-06 y I-15
	LAL-09	Bajo	-
Nivel (de RIL) en Reactor de RILes Simples (RC-05)	LAH-10	Alto	I-01, I-02, I-06 y I-15
	LAL-10	Bajo	-
Nivel (de RIL) en Reactor de RILes Simples (RC-06)	LAH-11	Alto	I-01, I-02, I-06 y I-15
	LAL-11	Bajo	-
Temperatura en Reactor de RILes Simples (RC-03)	TAH-01	Alta	
	TAHH-01	Muy Alta	
Temperatura en Reactor de RILes Simples (RC-04)	TAH-02	Alta	
	TAHH-02	Muy Alta	I-15
Temperatura en Reactor de RILes Simples (RC-05)	TAH-03	Alta	
	TAHH-03	Muy Alta	
Temperatura en Reactor de RILes Simples (RC-06)	TAH-04	Alta	
	TAHH-04	Muy Alta	

pH en Reactor RILes Simples (RC-03)	AAH-02	Básico	I-7, I-09 y I-10
	AAL-02	Ácido	
pH en Reactor RILes Simples (RC-04)	AAH-03	Básico	
	AAL-03	Ácido	
pH en Reactor RILes Simples (RC-05)	AAH-04	Básico	
	AAL-04	Ácido	
pH en Reactor RILes Simples (RC-06)	AAH-5	Básico	
	AAL-05	Ácido	

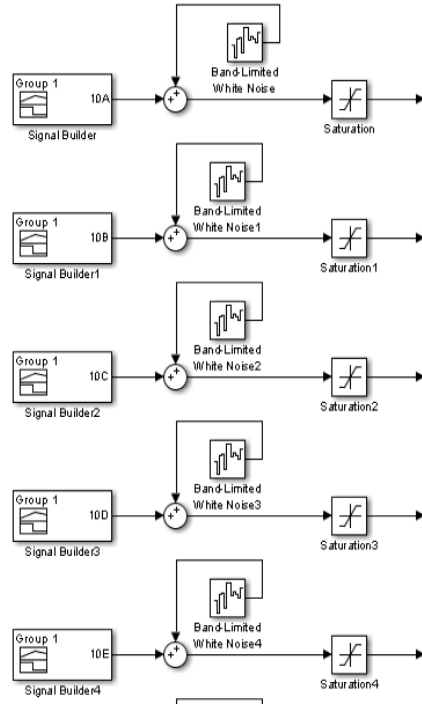
### 6.3.3 Modelación del proceso con uso de simuladores

Este ejemplo se centra en la programación de enclavamientos y alarmas, así como en la verificación de su correcto funcionamiento. Debido a las limitaciones de software que surgieron a raíz de la considerable cantidad de variables involucradas en este proyecto, se ha tomado la decisión de emplear únicamente un bloque OPC Write . En este enfoque, se suministrarán señales de interés para los enclavamientos, tales como nivel de los tanques y de nivel, pH y temperatura de los reactores mediante un bloque Signal Builder de Simulink. Para añadir realismo a las señales se ha introducido ruido lo que permitirá obtener gráficas más representativas. Este modelo simplificado proporciona un entorno adecuado para probar la lógica de programación en el contexto de enclavamientos y alarmas.



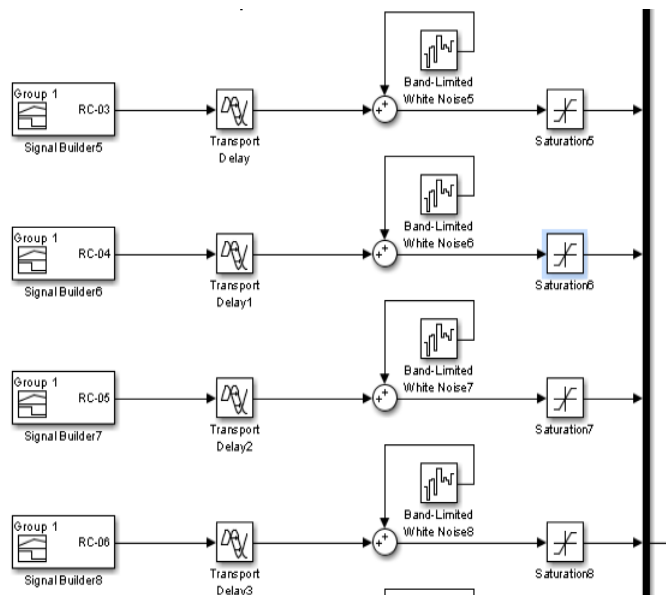
**Fig. 2.21 OPC Write**

Obteniendo así la siguiente vista general para las señales de Nivel de los estanques de almacenamiento:



**Fig. 6.22** Señales de Nivel de estanques en Simulink

En el caso de las señales de los Reactores se agregó un retardo para que al momento de que los estanques de almacenamiento comiencen a vaciarse los reactores empiecen a llenarse y se realizó el mismo procedimiento para la temperatura y el pH de los reactores obteniendo así la siguiente vista general:



**Fig. 6.23** Modelo simplificado en Simulink

#### 6.3.4 Programación PLC

La lógica de programación para el enclavamiento de cada bomba se llevó a cabo en un texto estructurado y se fundamenta en un conjunto de condicionales (IF) que responden a diversas situaciones operativas y de seguridad. Las bombas cuentan con un botón de partida y de parada digital que modifican el valor actual del Hand Switch físico de cada bomba correspondiente. Además, se han establecido condiciones de seguridad que apagan la bomba en caso de que no se cumplan las condiciones operativas de Nivel, pH o temperatura correspondiente a cada enclavamiento y dichas bombas no se pueden encender hasta que se cumplan las condiciones adecuadas para operar, además cabe mencionar que como dentro de los requerimientos del cliente no se encuentran definidos los valores mínimos y máximos estos valores podrán ser modificados desde la HMI a criterio del programador. Así cada enclavamiento asegura un control preciso y seguro de la bomba, evitando situaciones de sobrellenado y garantizando un funcionamiento eficiente en el sistema de tratamiento de RILES.

La programación de los enclavamientos se presenta mediante una serie de condicionales IF, que especifican las condiciones operativas para la activación del enclavamiento de cada bomba. A continuación, se detalla el código de enclavamiento de la bomba P-01, indicando las condiciones de apagado en caso de niveles elevados en los tanques de almacenamiento o los reactores.

```
// P-01
if P01_start then HS_P01:=1; P01_start:=0; end_if; // SE PRESIONA LA BOTONERA DE PARTIDA LA BOMBA PARTE
if P01_stop then HS_P01:=0; P01_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE APAGADO LA BOMBA SE APAGA

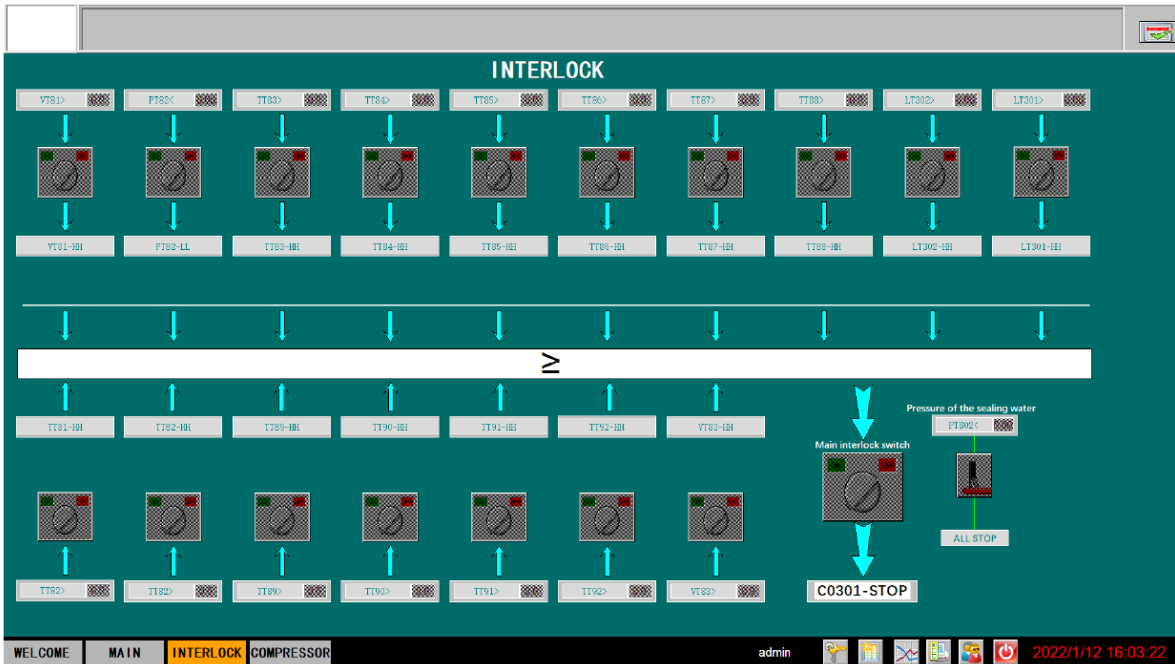
IF LT_03 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_04 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_05 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_06 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_07 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_08 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
```

**Fig. 6.24 Código de enclavamiento P-01**

#### 6.3.5 Diseño de pantalla HMI

La HMI diseñada para este proyecto se ha estructurado de manera específica para facilitar la verificación efectiva del funcionamiento de los enclavamientos y alarmas del sistema. La primera pestaña permite observar en tiempo real los niveles de los reactores y tanques, proporcionando una representación visual clara de las variables críticas del proceso. La segunda pestaña se centra en los enclavamientos, presentando de manera detallada el estado de las condiciones de arranque y parada de la bomba. La tercera pestaña se dedica a la visualización de alarmas, permitiendo una rápida

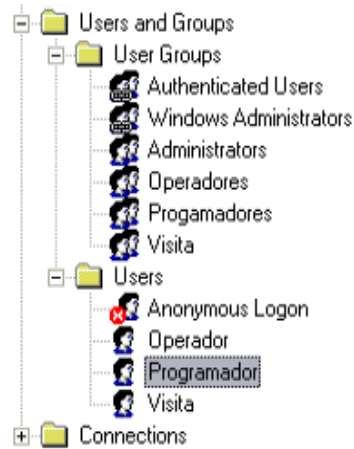
identificación y respuesta ante posibles problemas en el sistema. Cabe destacar que el diseño de la HMI se basa en las directrices de la normativa ISA 101, garantizando una interfaz ergonómica y eficiente. Además se tomaron como referencia imágenes de HMIs implementadas anteriormente en aplicaciones similares facilitadas por el cliente para así optimizar la experiencia del usuario y mantener la consistencia con estándares del cliente.



**Fig. 6.25 HMI de referencia**

No obstante se realizaron mejoras significativas en el diseño de la HMI. Se ajustó el tamaño de la letra para lograr una legibilidad óptima, se sustituyeron los interruptores por un sistema más intuitivo que evalúa de manera clara el cumplimiento de las condiciones, ofreciendo una interfaz más comprensible y agradable para el operador. Además se integró una función de visualización de alarmas en las pestañas de enclavamientos, proporcionando una referencia inmediata y mejorando la capacidad de respuesta ante situaciones críticas. Estas modificaciones optimizan la experiencia del usuario y fortalecen la eficiencia operativa al simplificar la interpretación de la información clave en tiempo real.

Además se crearon usuarios con distintos privilegios dentro de la interfaz, esto se realizó desde FactoryTalk View Studio:



**Fig. 6.26 Creación de usuarios en FactoryTalk**

Donde se crearon usuarios de Visita, Operador y programador. Los usuarios, contraseñas y privilegios se pueden resumir en la siguiente Tabla:

**Tabla 6.10 Usuarios creados**

Usuario	Contraseña	Privilegios
Visita	visita123	Sólo vistas de gráficos sin acceso a las pestañas donde hay parámetros modificables como enclavamientos y alarmas.
Operador	GIC123	Acceso a todas las pestañas de la HMI pero sin acceso a la modificación de parámetros mínimos y máximos de los enclavamientos.
Programador	GIC2023	Acceso a todas las pestañas de la HMI y acceso a la modificación de parámetros mínimos y máximos de los enclavamientos.

También desde la HMI se programaron las alarmas de Nivel, pH y temperatura ocupando condicionales para que cuando se cumpliera la condición la alarma fuera desplegada en el panel de Alarmas.



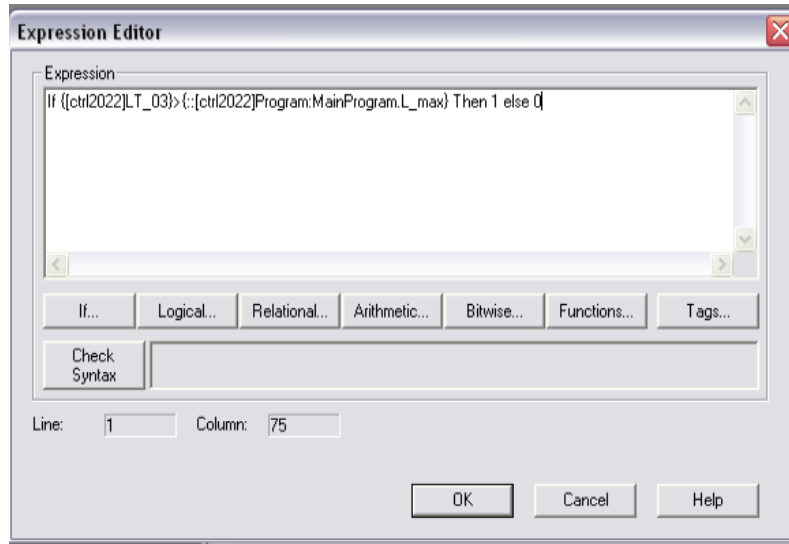


Fig. 6.27 Lógica de alarma de nivel

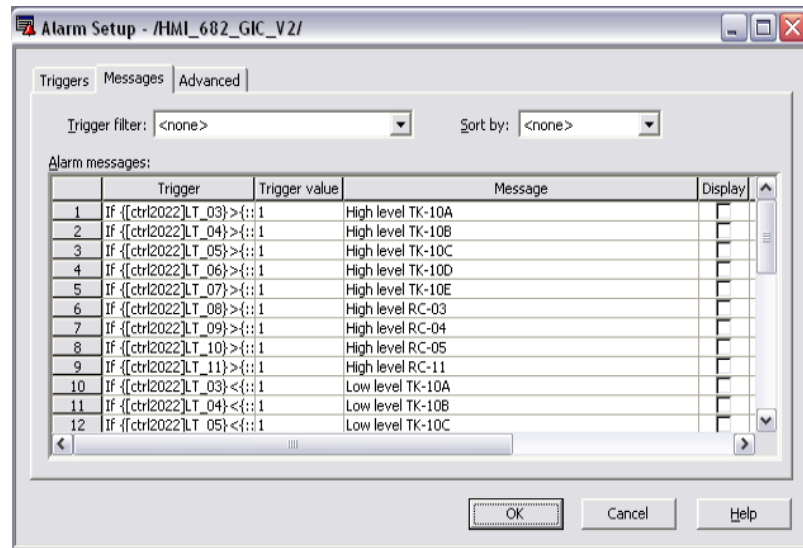


Fig. 6.28 Vista general de alarmas

### 6.3.6 Realización de pruebas FAT

Para la realización de pruebas FAT se seguirá el diagrama de flujo propuesto en el capítulo 4.

- **Planificación:** Se realizarán pruebas de llenado del reactor para verificar correcto funcionamiento de los enclavamientos de cada bomba, para esto se crearon señales en Signal Builder para cumplir las condiciones de alto nivel, alta temperatura y pH entre 6 y 8 para así activar los enclavamientos y alarmas programados para el proceso. Se verificará que las bombas no puedan ser encendidas mientras el enclavamiento esté activado y el encendido y apagado manual mientras el enclavamiento no esté activado.
- **Llevar a cabo pruebas de actividades:** Se realizaron las pruebas planificadas con

anterioridad a través de su HMI correspondiente.

- **Recolectar resultados de pruebas:** Se recopilaron los resultados guardando los gráficos de tendencias, imágenes de los apagados de bombas y sus estados actuales, además de la activación de alarmas en su momento correspondiente.
- **Identificar errores y corregirlos:** Para este caso no se identificaron errores en la lógica de control de los enclavamientos.
- **Fin:** Se aprueba la aceptación del sistema dando por finalizadas las pruebas

A continuación, se puede observar que los transmisores de nivel LT-03 y LT-04 de los estanques de almacenamiento TK-10A y TK-10E tiene un nivel superior a 60% el cuál fue fijado como límite máximo dentro de las condiciones iniciales por lo que el círculo verde indica que se está cumpliendo una de las condiciones del enclavamiento accionando el enclavamiento I-01 teniendo como consecuencia el apagado automático de la bomba.

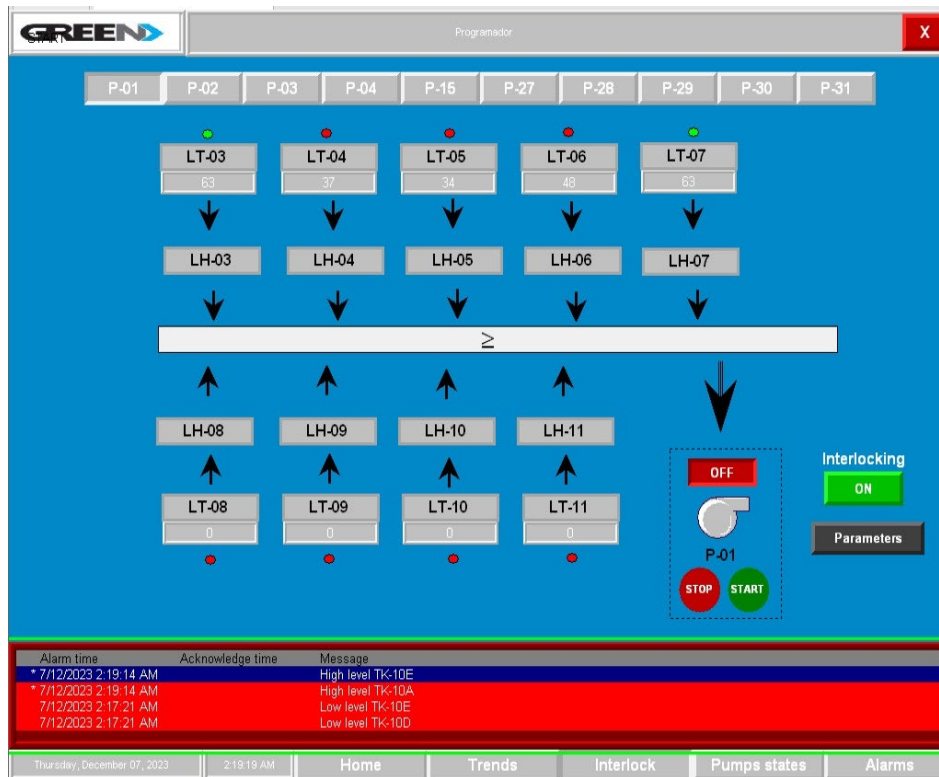
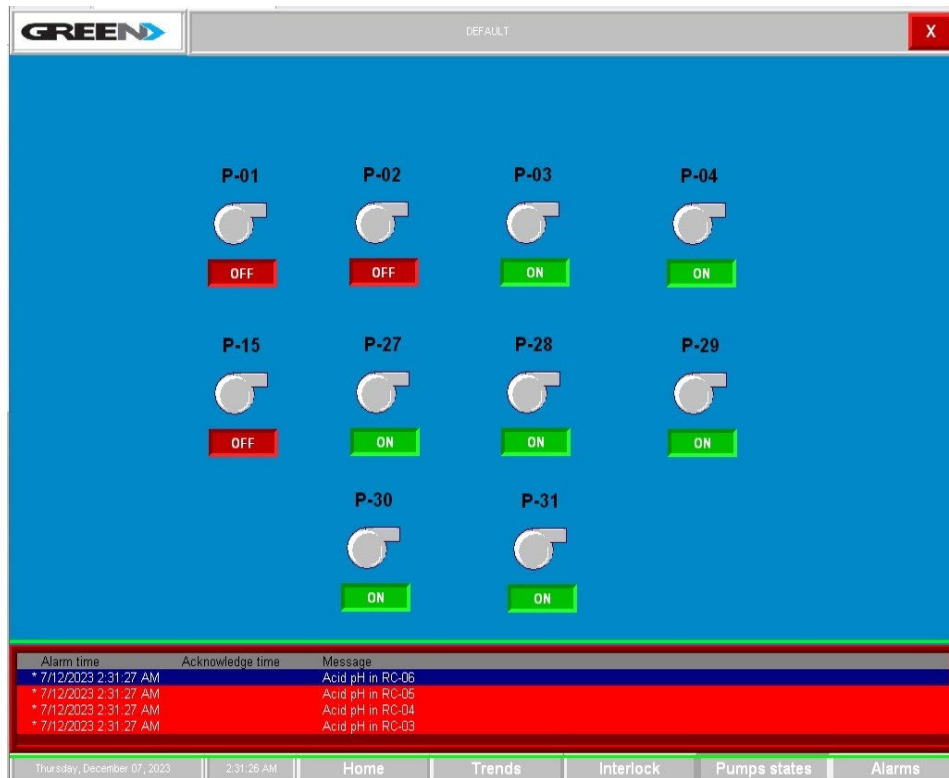


Fig. 6.29 Funcionamiento de enclavamiento P-01 en HMI



**Fig. 6.30 Vista de estados de bombas en HMI**

### 6.3.7 Puesta en marcha

Con las pruebas FAT exitosas, se ha verificado con éxito el correcto funcionamiento de los enclavamientos y la activación adecuada de las alarmas. El siguiente paso para concluir el servicio sería la puesta en marcha, la cual se llevará a cabo según lo detallado en la sección 4.2.9 Puesta en marcha.

Como parte de la entrega final, se proporcionará la documentación necesaria, incluyendo el programa Rslogix o Studio5000 en formato .ACD y para la HMI se entregará un archivo en formato .MER.

## 7. Conclusión

---

La conclusión extraída del análisis técnico presente en esta memoria de título destaca varios aspectos fundamentales. En primer lugar, se confirma que Green cumple con los requisitos necesarios para ofrecer el servicio, respaldado por su especialización en el área de procesos lo cual presenta una ventaja al momento de llevar a cabo pruebas FAT mediante modelos confiables, además se ve fortalecida por la presencia de un especialista en instrumentación y control brindando un respaldo esencial durante el desarrollo del servicio.

En el análisis de marcas, se llega a la conclusión de que, siendo una empresa nueva en el servicio de programación PLC la elección de la marca Allen-Bradley de Rockwell representa una opción de menor riesgo. La menor susceptibilidad a errores durante la programación y la puesta en marcha, inherente a los lenguajes de programación propios del software posiciona a Allen-Bradley por sobre su competencia directa Siemens. Además, se presentan dos opciones para configurar el set up del servicio, ambas robustas y confiables, aunque la alternativa de trabajar completamente en un entorno simulado destaca por su menor costo.

Se enfatiza la importancia de realizar pruebas FAT como un componente crítico para asegurar una ejecución exitosa del servicio. Estas pruebas no solo mitigan riesgos en la puesta en marcha, sino que también reducen errores de lógica de programación, optimizando tiempos de puesta en marcha y como consecuencia los costos asociados. En el desarrollo, se detallan aspectos cruciales como la recopilación exhaustiva de información sobre el proceso y los requerimientos del cliente, además de flujos de trabajo buscando así estandarizar el desarrollo por etapas fundamentales para que el programador pueda desarrollar el servicio de forma satisfactoria.

Finalmente, se llevaron a cabo ejemplos prácticos aplicando la teoría y los flujos de trabajo presentados en los capítulos anteriores. Estas simulaciones, que abarcan desde la verificación de la lógica de control de enclavamientos y alarmas hasta la pre-sintonización de un proceso altamente no lineal como la neutralización química, arrojaron resultados satisfactorios obteniendo los resultados esperados en ambos casos.

# Abreviaciones

## Mayúsculas

L.A.	: lazo abierto.
L.C.	: lazo cerrado.
L.D.	: lazo directo.
L.I.T.	: lineal invariante en el tiempo.
S.P.I.	: semi-plano izquierdo.
S.P.D.	: semi-plano derecho.
F. de T.	: función de transferencia.
F.D.	: función descriptora.
M. de T.	: matriz de transferencia.
B.W.	: ancho de banda.
E.S.	: entrada/salida.
S.S.	: estado estacionario.
SISO	: sistema de una entrada y una salida (single input single output).
MIMO	: sistema de varias entradas y varias salidas (multiple inputs multiple outputs).
L.G.R.	: lugar geométrico de las raíces.
P.I.D.	: controlador proporcional integral derivativo.
<i>S.P.</i>	: sobrepaso.
<i>M.G.</i>	: margen de ganancia.
<i>M.F.</i>	: margen de fase.
FCD	: forma canónica diagonal.
FCC	: forma canónica controlable.
FCO	: forma canónica observable.
FCJ	: forma canónica de Jordan.
T.L.	: Transformada de Laplace.
T.F.	: Transformada de Fourier.
T.F.F.D.	: Transformada de Fourier de Frecuencia Discreta.
T.Z.	: Transformada $\mathcal{Z}$ .
T.F.T.D.	: Transformada de Fourier de Tiempo Discreta.
T.F.D.	: Transformada de Fourier Discreta.
D. de B.	: Diagrama de Bode.
CSTR	: Continuous stirred tank reactor.

## Minúsculas

c.i.	: condiciones iniciales.
l.i.	: linealmente independiente.
l.d.	: linealmente dependiente.
c.c.	: corriente continua (en Inglés es d.c.).
c.a.	: corriente alterna (en Inglés es a.c.).
a.c.a.	: abscisa de convergencia absoluta.

## Referencias

- [1] M. V. Tenorio, "Breve Ensayo Sobre el PLC y su Importancia en la Industria," vol. 5, no. 2, pp. 150-165, 2008.
- [2] W. J. Torres Casas, "Aplicación de PLC's a nivel industrial," vol. 8, no. 4, pp. 225-240, 2012.
- [3] J. G. C. Lugo, J. J. P. Ybarra, y E. Romero, Metodología para realizar una automatización utilizando PLC, Impulso, vol. 18, 2005.
- [4] P. Aguilera Martínez, Programación de PLC'S, Tesis doctoral, Universidad Autónoma de Nuevo León, 2002.
- [5] J. P. Ramírez, G. G. D. Conti, y J. J. P. Ybarra, La automatización modular: una nueva opción para procesos automatizables, Impulso, vol. 1, 2005.
- [6] Baking Industrial Forum, "Factory Acceptance Testing," vol. 8, no. 3, pp. 210-225, 2015.
- [7] J. P. Segovia Vera, "Aplicación de Simuladores al Diseño de Sistemas de Control," IITUDEC, 2012.
- [8] M. Kopcek, T. Skulavik, P. Tanuska y D. Mudroncik, Systematic approach to factory acceptance test planning, in Computer Aided Chemical Engineering, vol. 33, pp. 1597-1602, Elsevier, 2014.
- [9] R. Tyagi and E. J. Davison, Control of pH in a continuous stirred tank reactor (CSTR), IFAC Proceedings Volumes, vol. 26, no. 2, pp. 115-126, 1993.
- [10] M. Meza García, Disturbios del estado ácido-básico en el paciente crítico, Acta Médica Peruana, vol. 28, no. 1, pp. 46-55, 2011.
- [11] S. K. Al-Dawery, H. A. O. Al-Anbari, and D. F. Ahmed, "Nonlinear Adaptive Control of a pH Process," Iraqi Journal of Chemical and Petroleum Engineering, vol. 10, no. 1, pp. 9-16, 2009.
- [12] T. H. Naylor y J.M. Finger "Verification of Computer Simulation Models" Management Science, vol. 14, no. 1, pp. 92-101, Oct. 1967.

## Anexo A. Cotizaciones

Código	Nombre	Cantidad	Marca	Precio unitario	Descuento unitario	Total	Tiempo de entrega
1756-L81E	PROCESADOR L81E PARA CONTROLLOGIX	1	ALLEN	USD		USD	3 Días
			BRADLEY	\$8.949,60		\$8.949,60	
1606-XLS240E	1606-XLS240E- PERFORMANCE POWER SUPPLY, 24-28V DC, 240 W, 120/240V AC / 100-150V DC INPUT VOLTAGE	1	ALLEN	USD \$826,59		USD	3 Días
			BRADLEY			\$826,59	
1756-A4	CHASIS ESTÁNDAR PARA CONTROLLOGIX- 4 ESPACIOS	1	ALLEN	USD \$549,43		USD	3 Días
			BRADLEY			\$549,43	
<b>Subtotales</b>							
One-time Subtotales				USD \$10.325,62			

**Fig. A.1 Cotización Precision de PLC de prueba Controllogix**

ITEM	UNI	QTY	DELIVERY	PRICE
<b>1) 1756-EN4TR Allen-Bradley - NEW</b> Módulo de Comunicaciones, ControlLogix, EtherNet/IP, Puerto Doble, Par Trenzado 10M/100M/1G, 512 Conexiones TCP 256 Ejes	3.389,10USD	1	In stock 7-10 days	3.389,10USD
			Subtotal	3.389,10USD
			Shipping to Chile	52,35USD
			<b>Total</b>	<b>3.441,45USD</b>

**Fig. A.2 Módulo Ethernet Controllogix**

Importes visualizados en Chile Peso

Descripción	IVA	P.U.	Cant.	Total (Base imp).
WorkStation HP Zbook Firefly 14 G8 Intel Core i7-1165G7. 16 GB (1x16GB) DDR4-3200 nECC SODIMM. 512 GB. Pantalla 14" Full HD (1920 x 1080) Tarjeta video NVIDIA Quadro T500. Windows 10 Profesional 64-bit. Garantía 1 año (1/1/0)	19%	1.350.000	1	1.350.000
<b>Total (Base imp).</b>				1.350.000
<b>Total IVA 19%</b>				256.500
<b>Total</b>				1.606.500

**Fig. A.3 Cotización de Opciones - Workstation HP**

Item	Q	Descripción	P. Venta (un)	P. Venta (t)	Entrega (Semanas)
1	1	SIMATIC STEP 7 Basic V18 SIEMENS, descarga de licencia flotante; software de ingeniería en el TIA Portal; software, documentación y clave de licencia para descargar; clase A; 9 idiomas: de,en,zh incluido, fr,es,it,ru,ja,ko como descarga; ejecutable en Windows 10; Windows 11; Servidor Windows 2016/2019/2022; para la configuración de SIMATIC S7-1200, SIMATIC Basic Panels	USD 1021	USD 1021	Descarga en línea
				<b>USD 1021</b>	
Item	Q	Descripción	P. Venta (un)	P. Venta (t)	Entrega (Semanas)
1	1	IMATIC STEP 7 Professional V18 SIEMENS, descarga de licencia flotante; software de ingeniería en el TIA Portal; software, documentación y clave de licencia para descargar; clase A; 9 idiomas: de,en,zh incluido, fr,es,it,ru,ja,ko como descarga; ejecutable en Windows 10; Windows 11; Servidor Windows 2016/2019/2022; para configuración de SIMATIC S7-1200/1500, SIMATIC S7-300/400/WinAC, SIMATIC Basic Panels	USD 2.998	USD 2.998	Descarga en línea
				<b>USD 2.998</b>	

**Fig. A.4 Cotización de Lureye Softwares Siemens**



1. Choose Your Package

RSLinx Classic Single Node Subscription <input checked="" type="checkbox"/> <b>\$USD 296.87</b> /license/year <small>RSLinx Classic Single Node provides an OPC DA interface to a single controller by a local OPC Client.</small>		
	Perpetual With Maintenance	Subscription
License Flexibility ?	N/A	Yes
Intelligent Activation ?	N/A	Yes
Customizable Bundles	N/A	Yes
Reinstatement Fees	Yes	No
Phone Support	Optional	Included(8-5;M-F)
Software Updates	Included	Included
Knowledgebase	Yes	Yes
System Support	No	No
One Time Cost	\$USD 741.15 /license	-
Annual Cost	\$USD 148.44 /license/year	\$USD 296.87 /license/year
Product Selection	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Fig. A.5 Costo RSLinx Classic

### Studio5000 Professional Edition



**\$USD 3,632.77** /user/year

The Studio 5000® Professional Design Environment is a scalable application solution, used to program and configure any of the Logix5000™ family of controller products and PanelView™ 5000 products. Studio 5000® Professional includes everything you need including the following editors: Ladder Diagram, Function Block, Structured Text, Sequential Function Chart, GuardLogix Safety Editor, Phase Manager, and Sequence Manager. Professional Edition also includes Logix Emulate, Architect, and RSNetwork applications. This Studio5000 offering provides an activation for versions 20.05 and above.

For detailed selection information, click [here](#)

	Perpetual With Maintenance	Subscription
License Flexibility ?	N/A	Yes
Intelligent Activation ?	N/A	Yes
Customizable Bundles	N/A	Yes
Reinstatement Fees	Yes	No
Phone Support	Optional	Included(8-5;M-F)
Software Updates	Included	Included
Knowledgebase	Yes	Yes
System Support	No	No
One Time Cost	<b>\$USD 9,006.02</b> /license	-
Annual Cost	<b>\$USD 1,681.12</b> /user/year	<b>\$USD 3,632.77</b> /user/year
Product Selection	<input type="checkbox"/>	<input checked="" type="checkbox"/>

**Fig. A.6 Costo Studio5000 Professional Edition**

FactoryTalk View Studio Enterprise		
<input checked="" type="checkbox"/> <b>\$USD 1.060,30</b> /license/año		
Software de configuración para el desarrollo y pruebas de las aplicaciones FactoryTalk View SE y FactoryTalk View ME. Para más información sobre la selección, haga clic aquí		
	Perpetual With Maintenance	Subscription
License Flexibility ?	N/A	Yes
Intelligent Activation ?	N/A	Yes
Customizable Bundles	N/A	Yes
Reinstatement Fees	Yes	No
Phone Support	Optional	Included(8-5;M-F)
Software Updates	Included	Included
Knowledgebase	Yes	Yes
System Support	No	No
Costo No Recurrente	<b>\$USD 2.650,75</b> /license	-
Costo Anual	<b>\$USD 530,15</b> /license/año	<b>\$USD 1.060,30</b> /license/año
Selección De Productos	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Fig. A.7 Costo FactoryTalk View SE

## Anexo B Normativa HMI

---

La normativa **ISA 101** también conocida como **ISA-101.01-2015** surge como respuesta a la necesidad de establecer estándares y mejores prácticas en el diseño de interfaces HMI en sistemas de automatización de procesos industriales. Antes de la creación de esta norma, existían desafíos significativos en el diseño y uso de HMI en entornos industriales, lo que llevó a la identificación de la necesidad de una guía estándar que asegurara proporcionar una orientación para diseñar una HMI para así mejorar la capacidad de detección del usuario.

Los sistemas de control en la industria son cada vez más complejos, lo que hace que la supervisión y operación de estos sistemas sea una tarea desafiante. La normativa ISA 101 busca proporcionar pautas claras para simplificar la interacción entre operadores humanos y sistemas de control.

La seguridad y el rendimiento son fundamentales en entornos industriales. Un diseño deficiente de HMI puede llevar a riesgos de seguridad y problemas de rendimiento. La norma ISA 101 busca mejorar la seguridad y el rendimiento a través de directrices de diseño. Antes de la norma ISA 101 no existía una estandarización en el diseño de HMI. La falta de consistencia dificultaba la formación de operadores y la comprensión de sistemas de control por parte de diferentes fabricantes.

La norma reconoce la importancia de diseñar HMI centrados en las necesidades y capacidades de los operadores humanos. Busca mejorar la toma de decisiones y la eficacia operativa al poner al operador en el centro del diseño.

Apegarse a la norma ISA 101 en el diseño de interfaces HMI en sistemas de automatización es fundamental por varias razones. En primer lugar, garantiza la seguridad de las operaciones al proporcionar directrices claras para el diseño reduciendo así los riesgos asociados con fallos o errores en la interfaz. Además, mejora el rendimiento al optimizar la interacción entre operadores humanos y sistemas de control lo que se traduce en una mayor eficiencia operativa. Cumplir con esta norma también facilita la formación y comprensión de sistemas de control por parte de diferentes fabricantes al establecer una base común de diseño por lo que es importante que al momento de realizar un servicio de diseño de pantallas HMI el especialista tenga conocimientos sobre la normativa ISA 101 y consulte los estándares al momento de realizar el diseño.

## Anexo C Ejemplo Neutralización Química

---

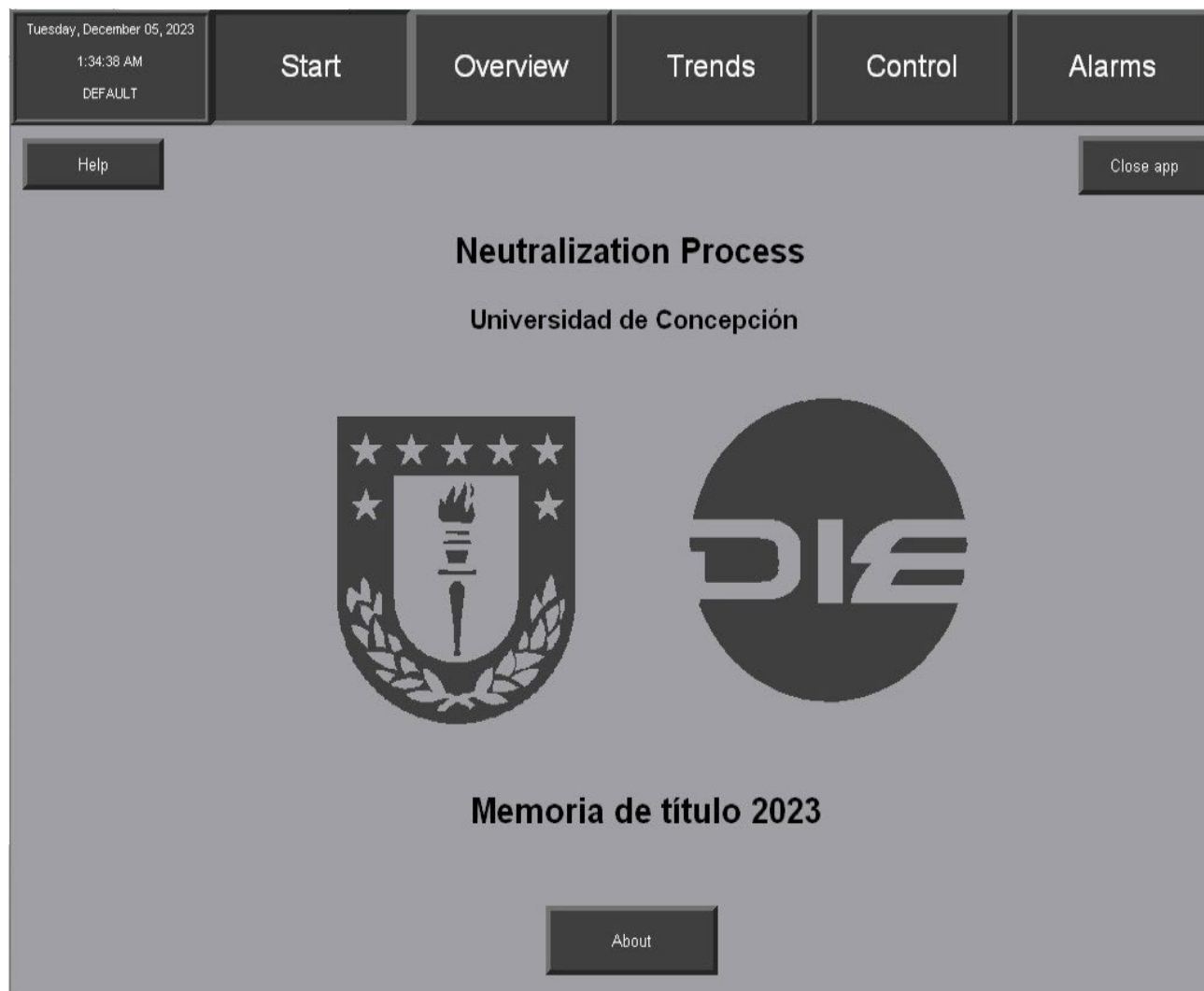


Fig. C.1 Pestaña de inicio

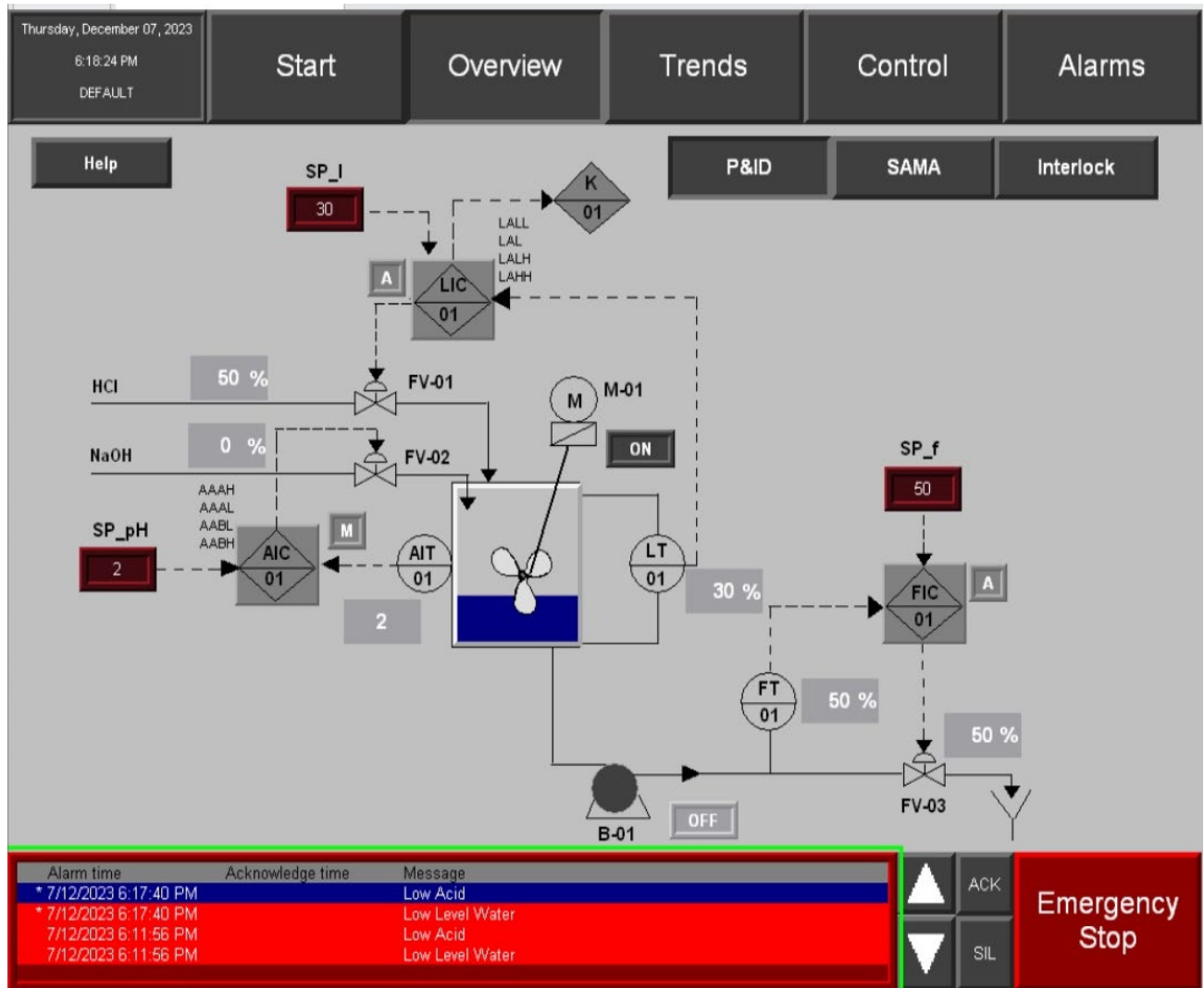


Fig. C.2 Pestaña P&ID

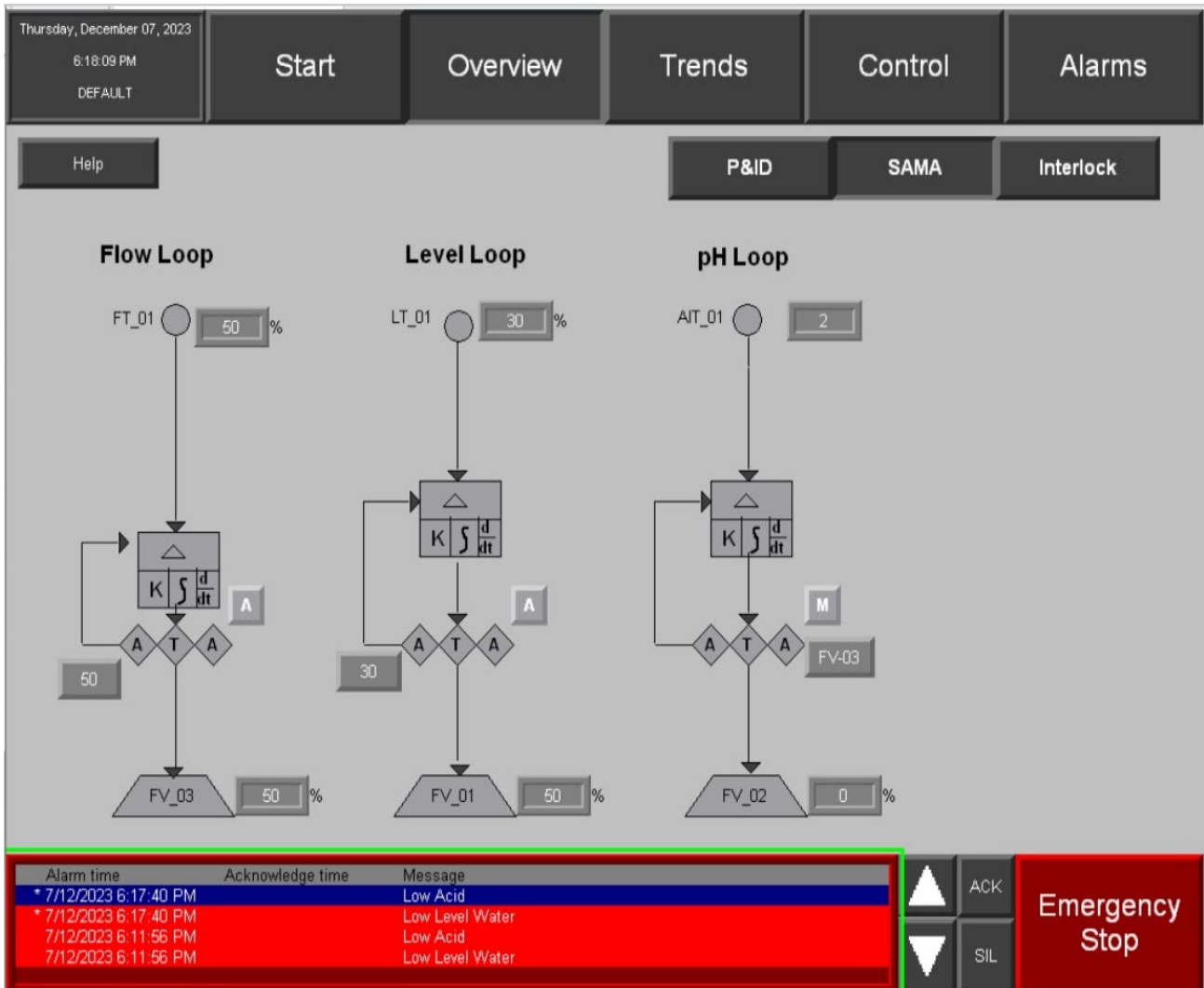


Fig. C.3 Pestaña diagrama SAMA

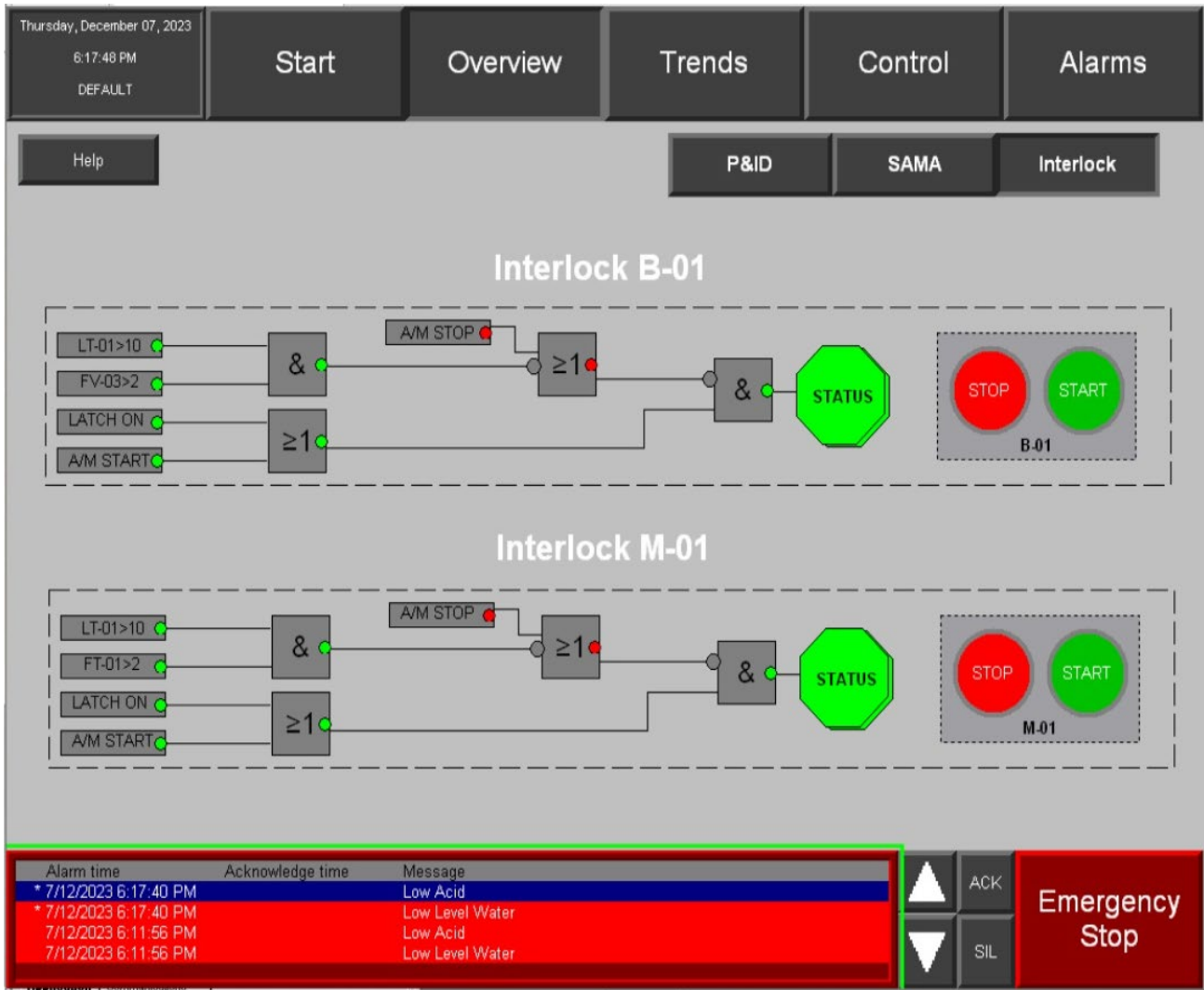


Fig. C.4 Pestaña Interlock





Fig. C.5 Pestaña tendencias

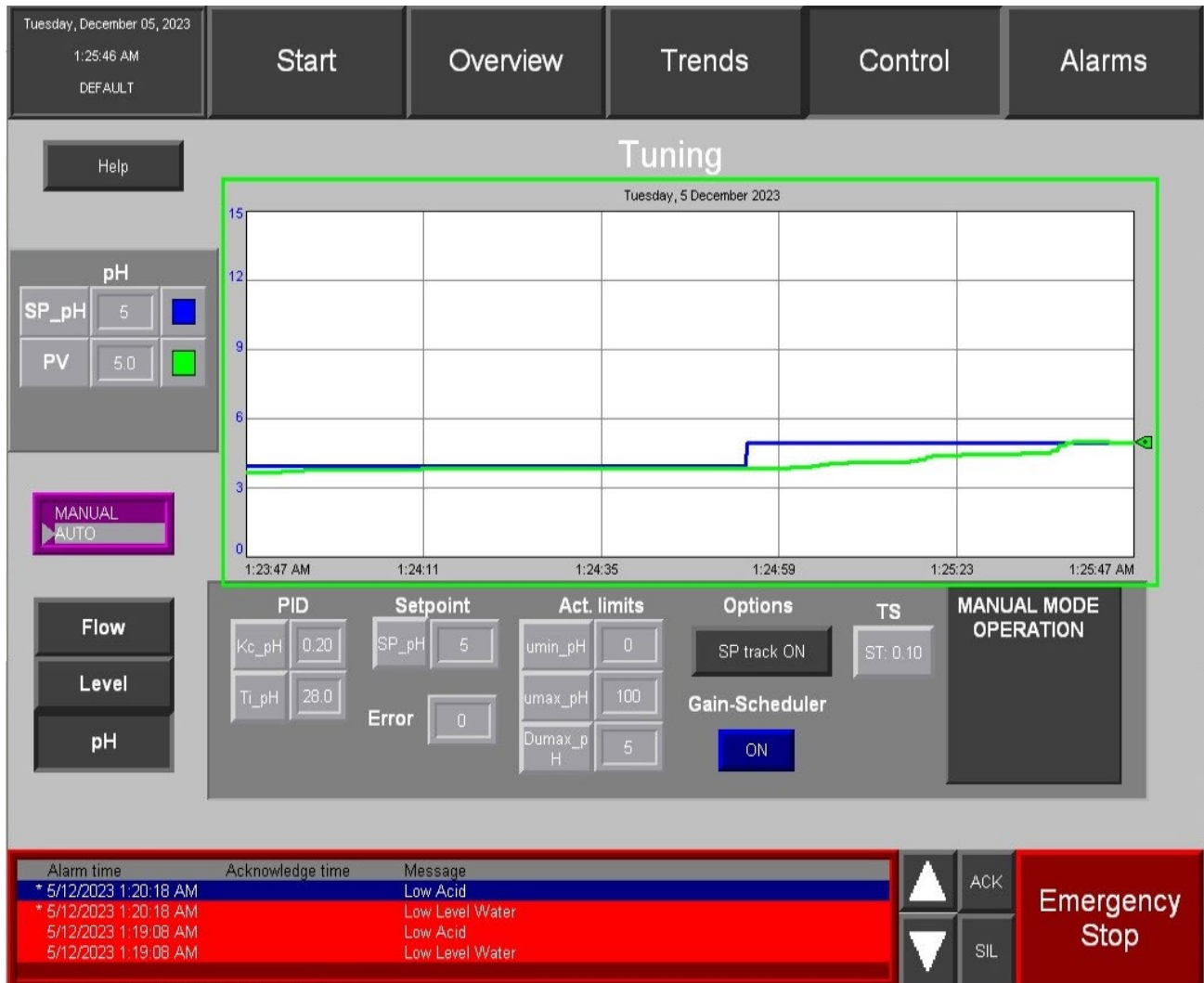


Fig. C.6 Pestaña Control pH



Fig. C.7 Pestaña control Nivel

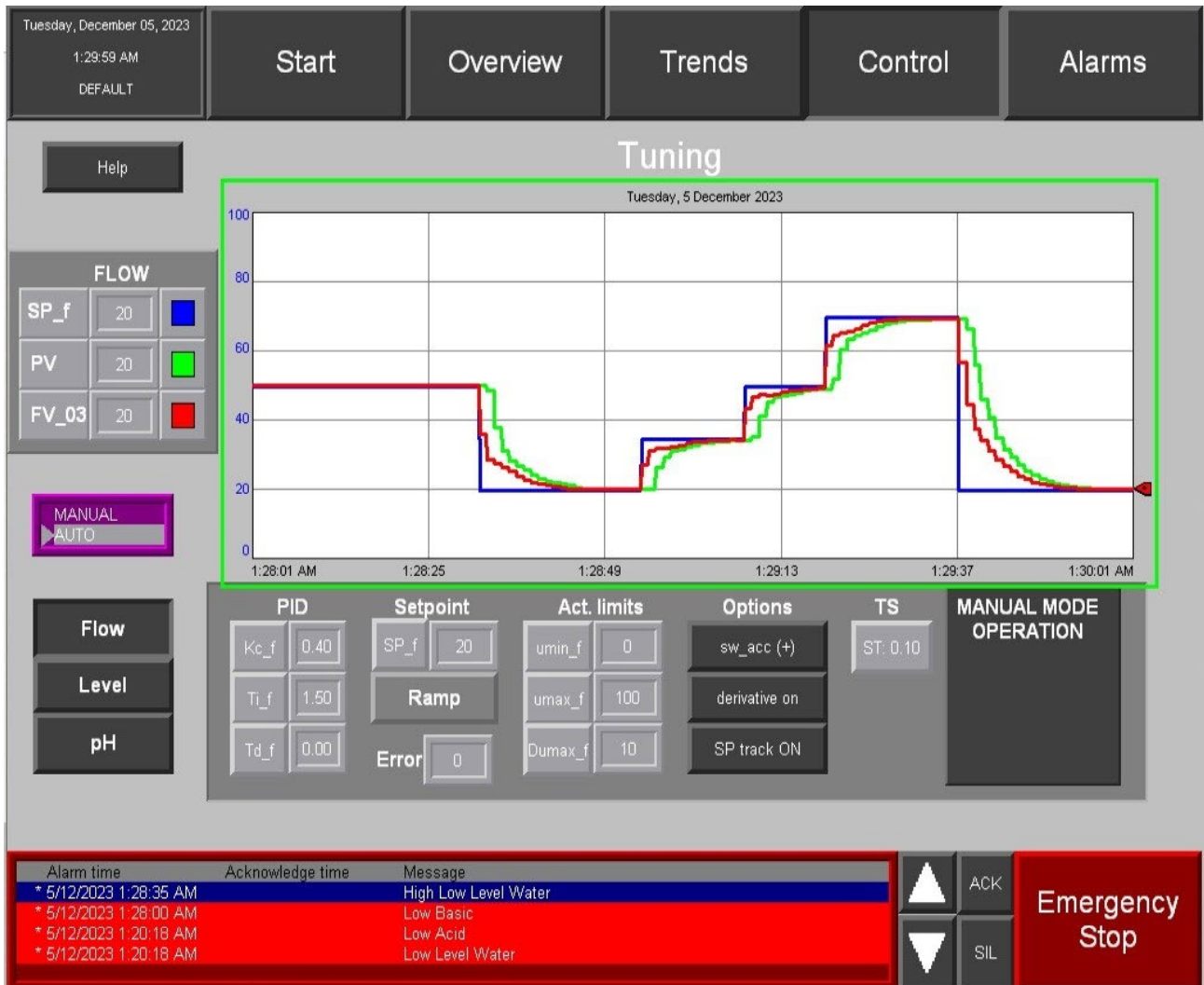


Fig. C.8 Pestaña control de flujo

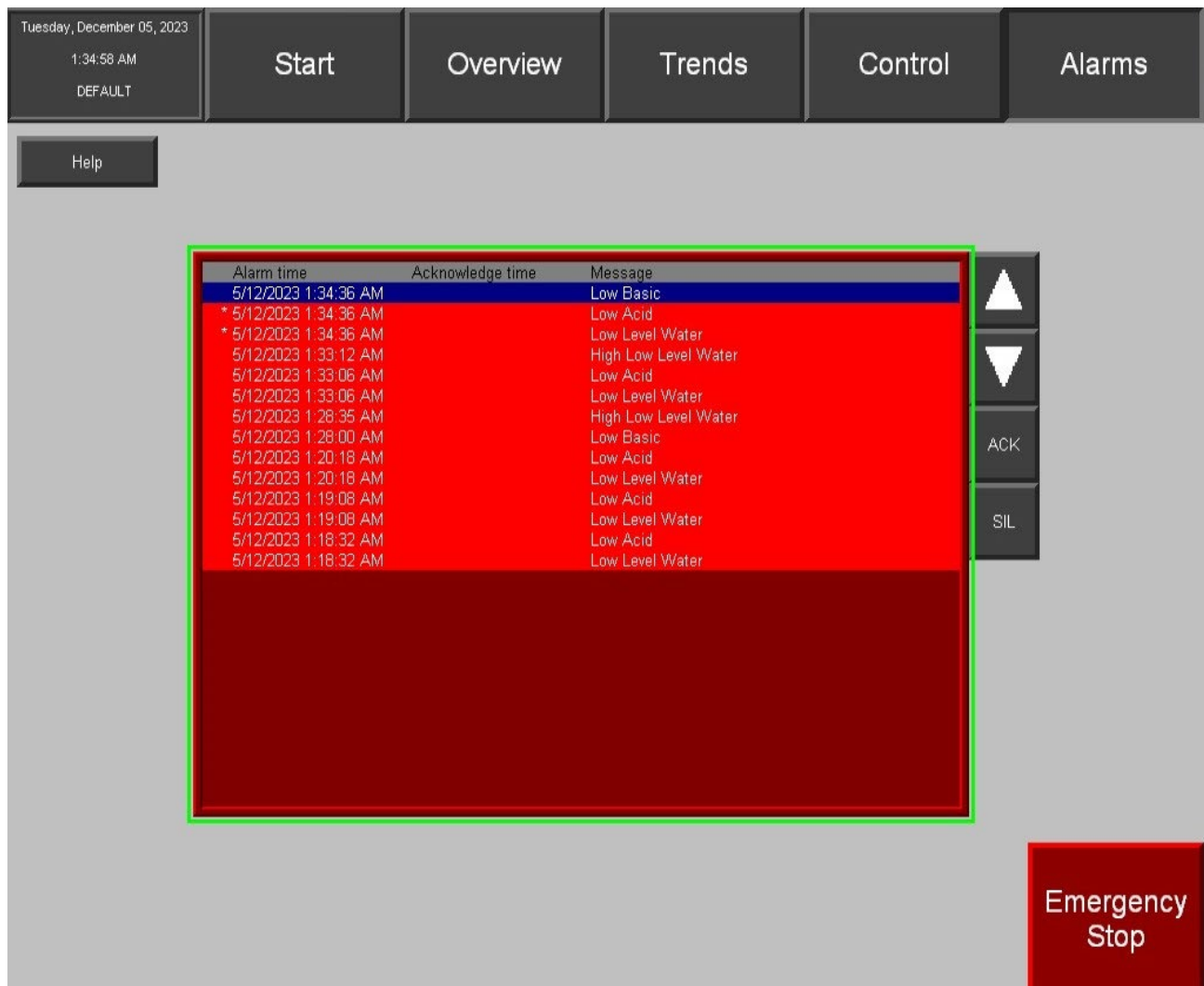


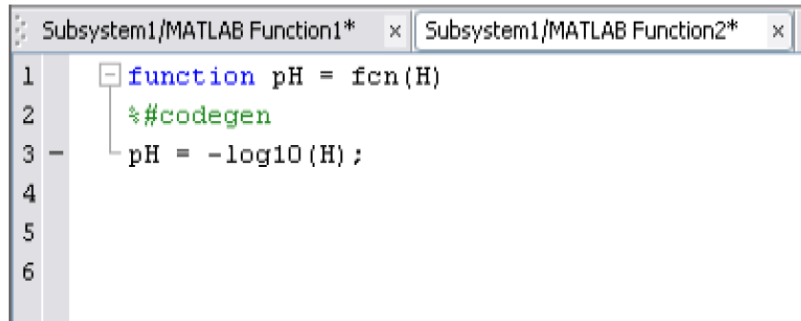
Fig. C.9 Pestaña de alarmas

```

Subsystem1/MATLAB Function1* x
1  function H = fcn(X)
2  %#codegen
3  Kw = 10e-14;
4
5  if X > 0
6      H = (X/2) * (sqrt(1 + (4*Kw/X^2)) - 1);
7  elseif X == 0
8      H = sqrt(Kw);
9  else
10     H = (-X/2) * (sqrt(1 + (4*Kw/X^2)) + 1);
11 end

```

Fig. C.10 Matlab function 1



```
Subsystem1/MATLAB Function1* x Subsystem1/MATLAB Function2* x
1 function pH = fcn(H)
2     %#codegen
3     pH = -log10(H);
4
5
6
```

**Fig. C.11 Matlab function 2**

# Anexo D Ejemplo Planta de RILes

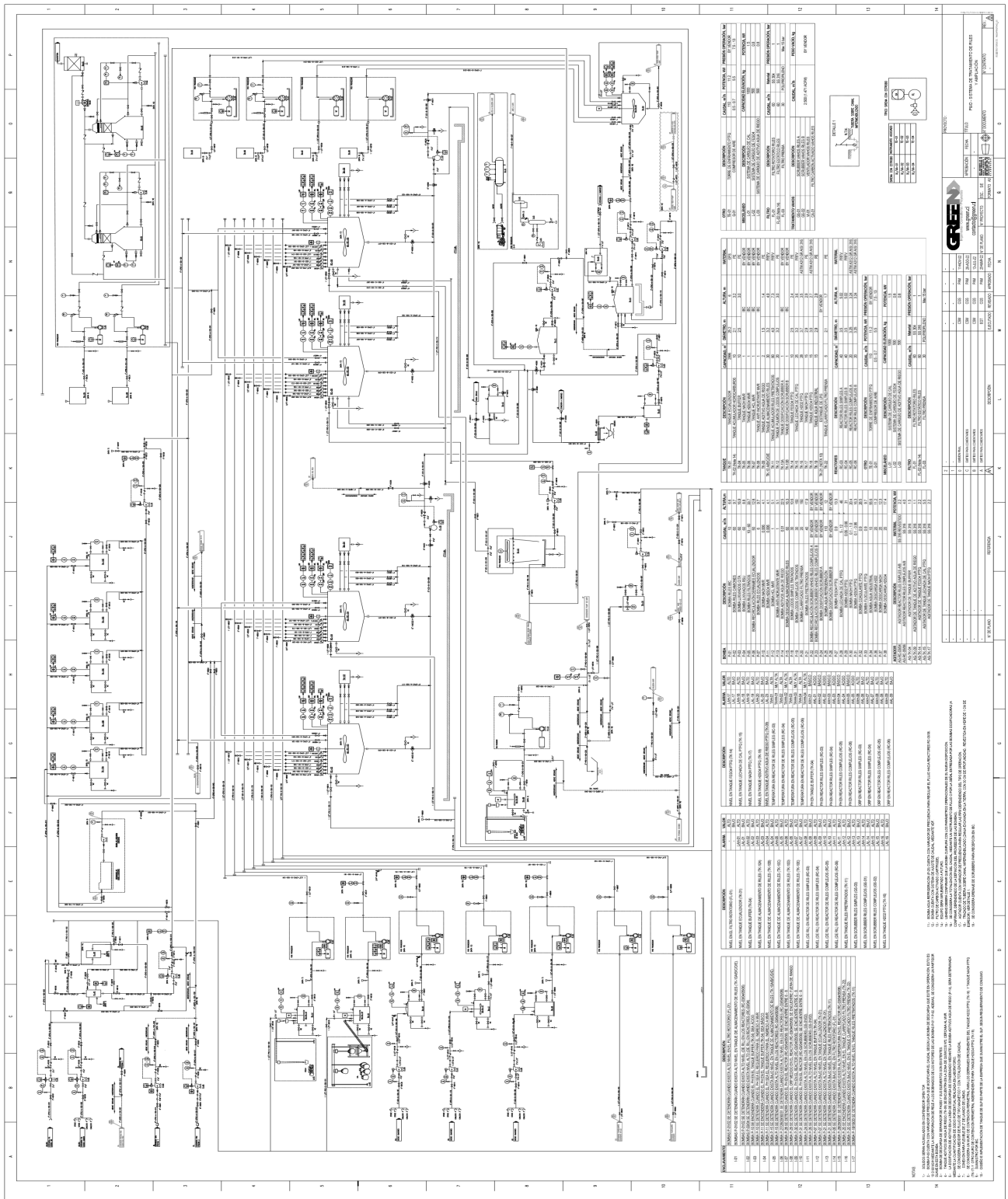


Fig. D.1 P&ID Planta de RILes



**Fig. D.2** Pestaña de inicio



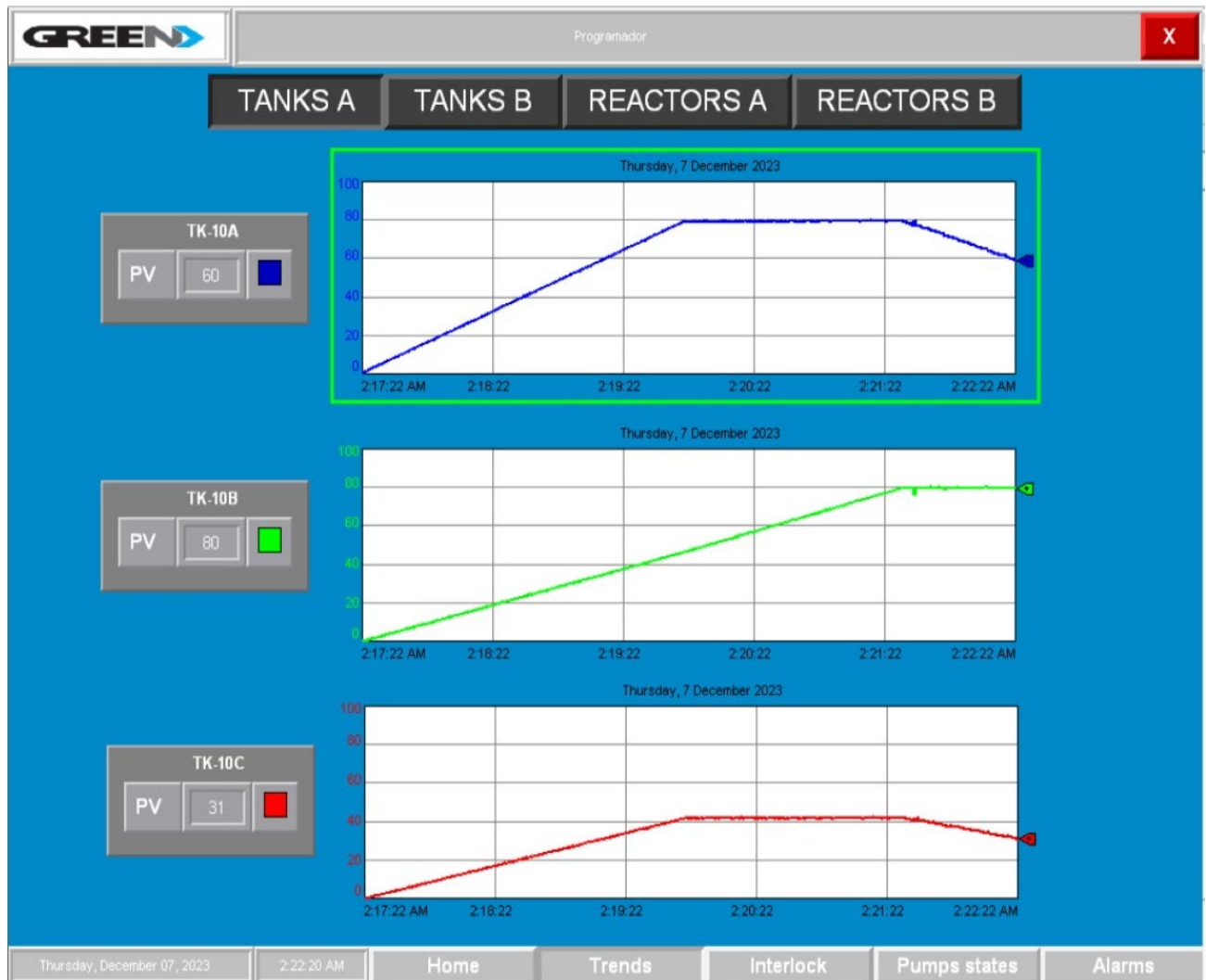


Fig. D.3 Pestaña tendencias estanques A

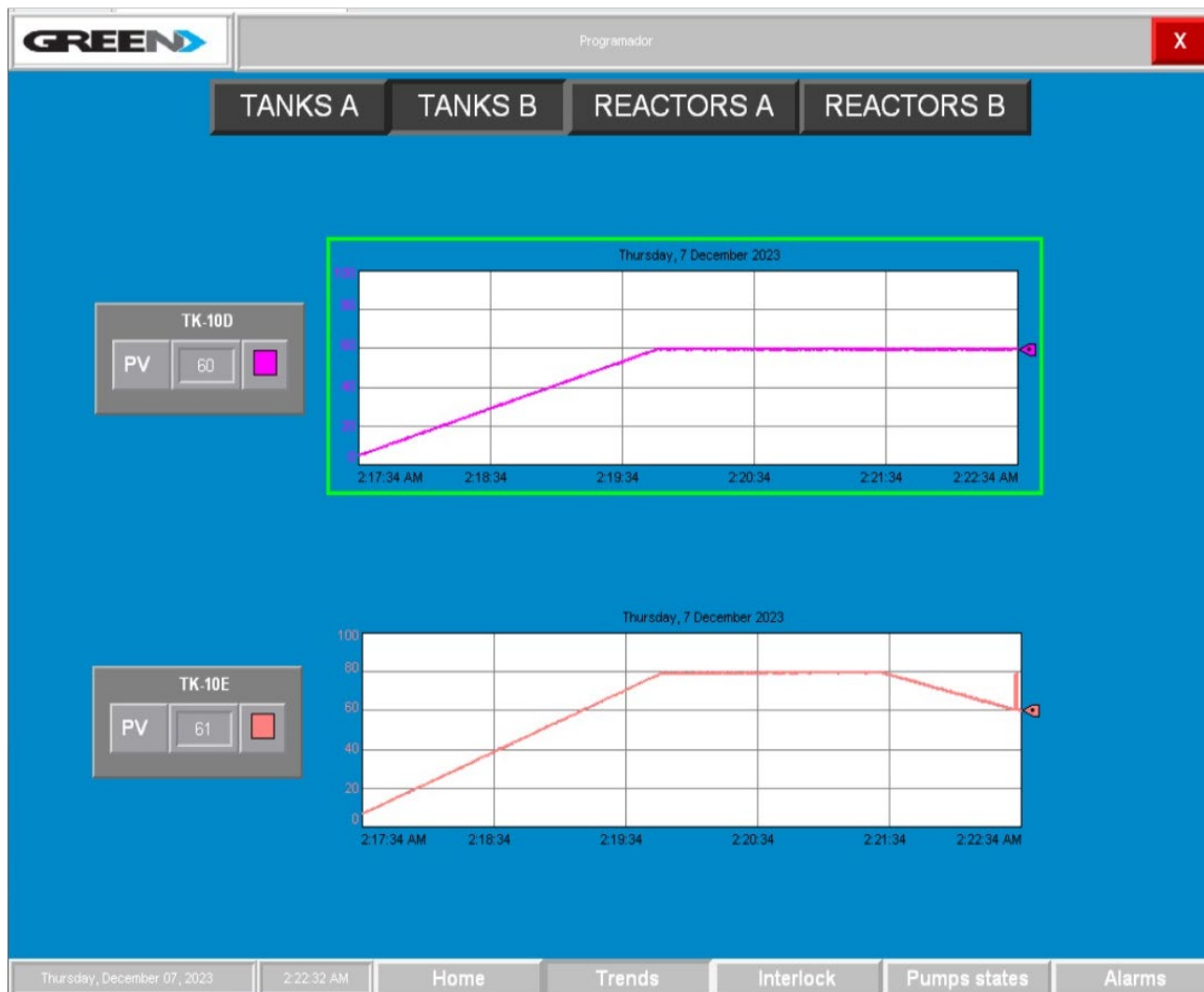


Fig. D.4 Pestaña tendencias estanques B

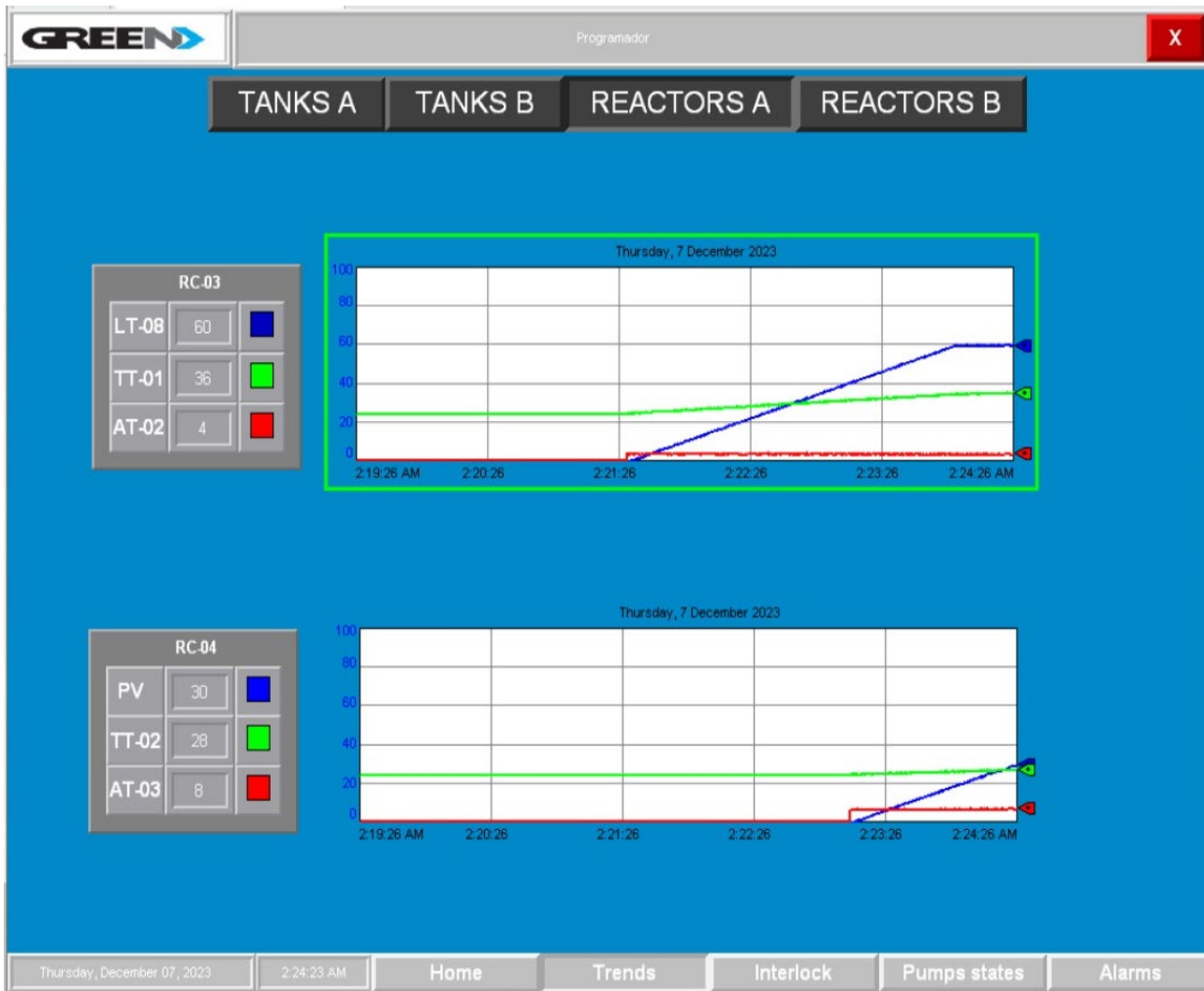


Fig. D.5 Pestaña de tendencias reactores A



**Fig. D.6 Pestaña de tendencias reactores B**

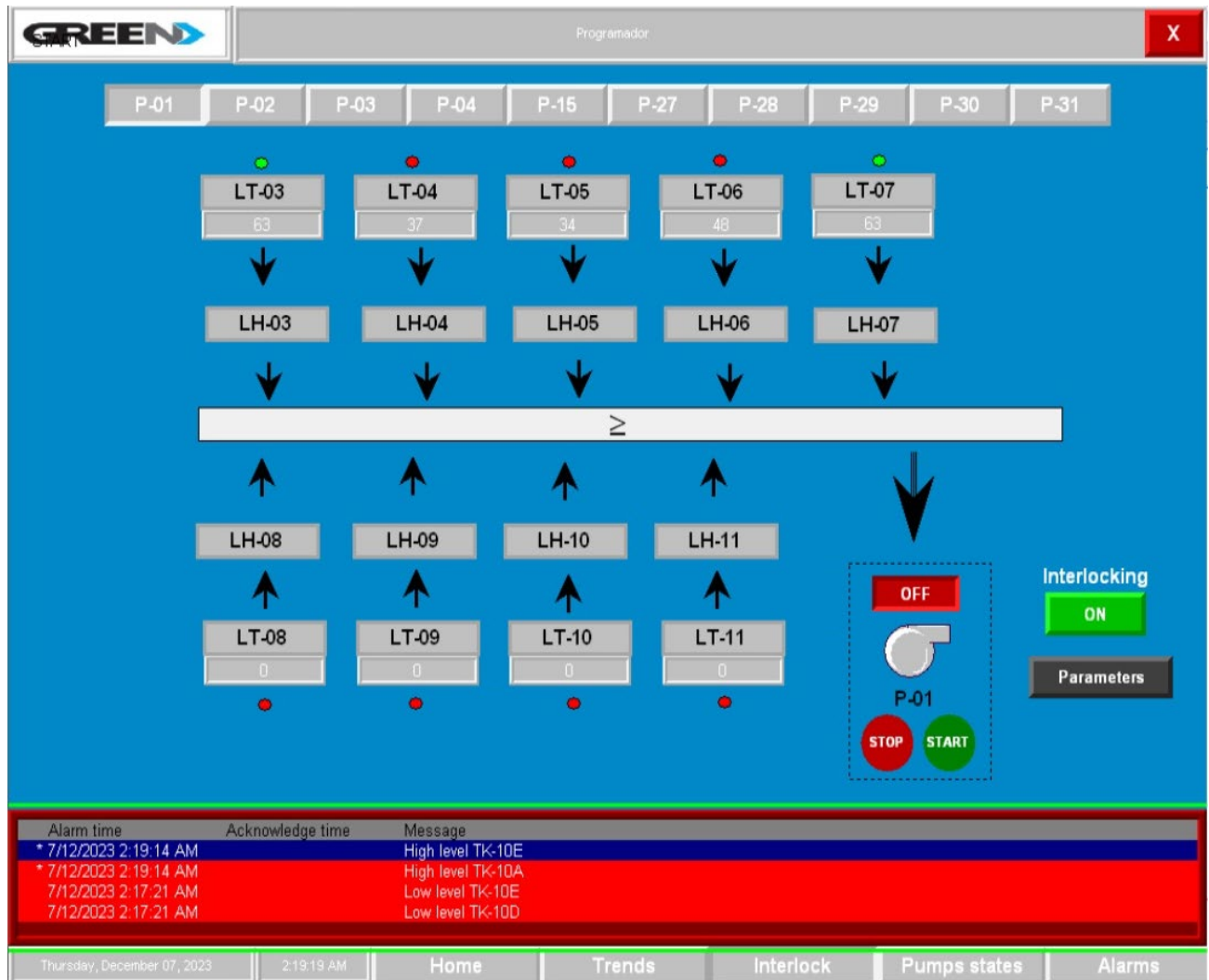


Fig. D.7 Pestaña de enclavamientos

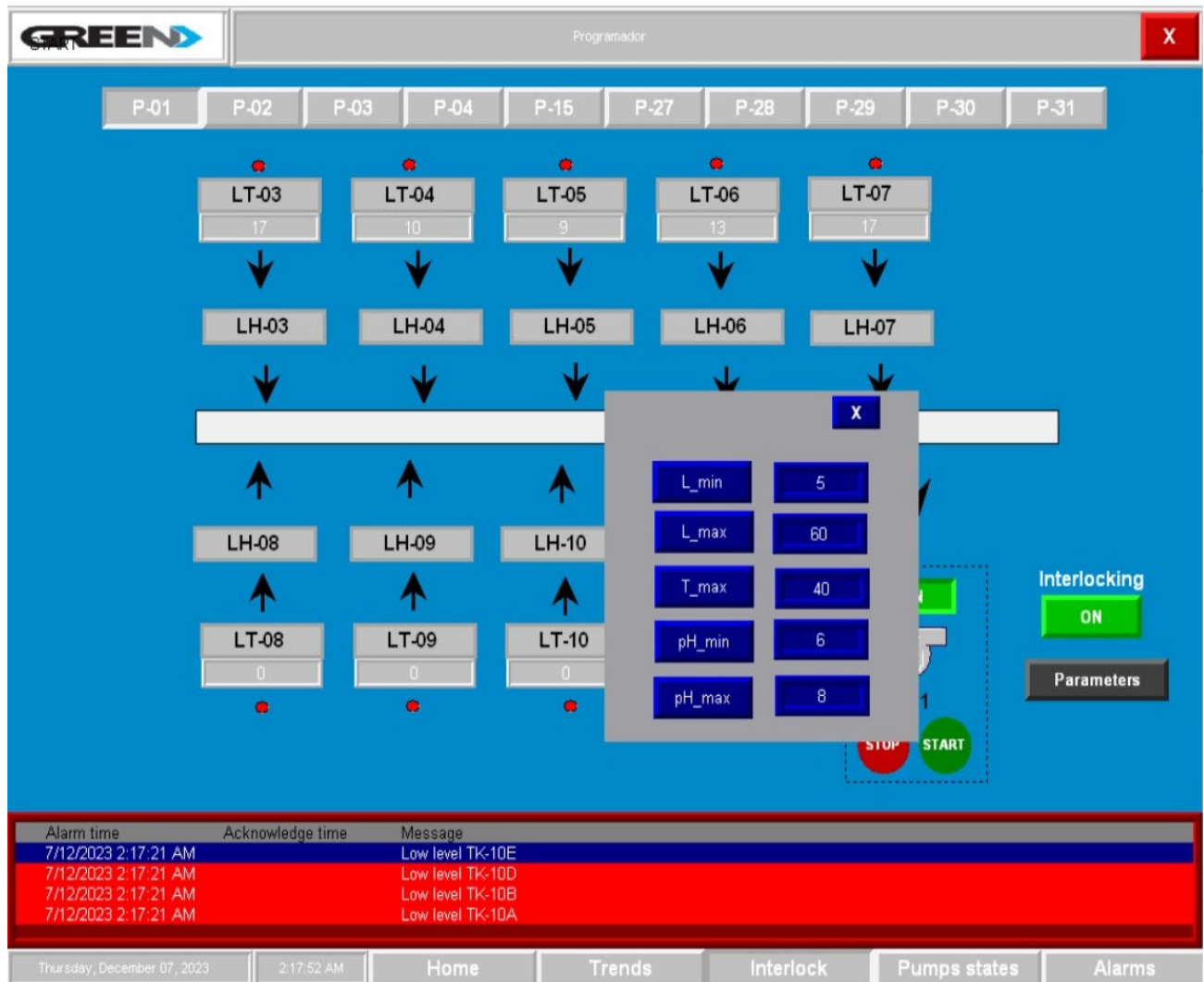


Fig. D.8 Pop-up de parámetros

GREEN Programador X

Alarm time	Acknowledge time	Message
7/12/2023 2:17:21 AM		Low level TK-10E
7/12/2023 2:17:21 AM		Low level TK-10D
7/12/2023 2:17:21 AM		Low level TK-10B
7/12/2023 2:17:21 AM		Low level TK-10A
* 7/12/2023 2:16:53 AM		Acid pH in RC-06
* 7/12/2023 2:16:53 AM		Acid pH in RC-05
* 7/12/2023 2:16:53 AM		Acid pH in RC-04
* 7/12/2023 2:16:53 AM		Acid pH in RC-03
* 7/12/2023 2:16:53 AM		Low level RC-06
* 7/12/2023 2:16:53 AM		Low level RC-05
* 7/12/2023 2:16:53 AM		Low level RC-04
* 7/12/2023 2:16:53 AM		Low level RC-03
7/12/2023 2:16:53 AM		Low level TK-10C
7/12/2023 2:09:24 AM		Acid pH in RC-06
7/12/2023 2:09:24 AM		Acid pH in RC-05
7/12/2023 2:09:24 AM		Acid pH in RC-04
7/12/2023 2:09:24 AM		Acid pH in RC-03
7/12/2023 2:09:24 AM		Low level RC-06
7/12/2023 2:09:24 AM		Low level RC-05
7/12/2023 2:09:24 AM		Low level RC-04
7/12/2023 2:09:24 AM		Low level RC-03
7/12/2023 2:09:24 AM		Low level TK-10E
7/12/2023 2:09:24 AM		Low level TK-10D
7/12/2023 2:09:24 AM		Low level TK-10C
7/12/2023 2:09:24 AM		Low level TK-10B

▲  
▼  
ACK  
SIL

Thursday, December 07, 2023 | 2:18:10 AM | Home | Trends | Interlock | Pumps states | Alarms

**Fig. D.9** Pestaña de alarmas

## Anexo E Códigos Rslogix5000

---

### Código Interpolación pH

```
sbr(alpha);

//list
list[0,0] := 1;
list[1,0] := 3;
list[2,0] := 3.5;
list[3,0] := 4;
list[4,0] := 5;
list[5,0] := 6;
list[6,0] := 7;
list[7,0] := 9;
list[8,0] := 12;

list[0,1] := 8;
list[1,1] := 4;
list[2,1] := 1;
list[3,1] := 0.3;
list[4,1] := 0.2;
list[5,1] := 0.1;
list[6,1] := 0.08;
list[7,1] := 0.5;
list[8,1] := 1;

nf:=9;

for id:=0 to nf-2 by 1 do
    IF alpha>=list[id,0] & alpha<= list[id+1,0] THEN
        ph[id] := (alpha-list[id+1,0])/(list[id,0]-list[id+1,0]);
        ph[id+1] := (alpha-list[id,0])/(list[id+1,0]-list[id,0]);

        ELSE ph[id+1] :=0;
        END_IF;
end_for;

interp :=0.0;

for j:=0 to nf-1 by 1 do
    interp := interp+ph[j]*list[j,1];
end_for;

RET(interp);
```



## Lazo de pH

IF sw\_GS then

//Gain Scheduler

```
if aux = 0 then alpha := SP_pH; end_if;  
if aux = 1 then alpha := AIT_01; end_if;  
if aux = 2 then alpha := FV_02; end_if;
```

//Interpolación Kc

```
jsr(Interpolacion_Kc , 1 , alpha , interp);
```

```
kc_pH :=interp;
```

//Metodo por tabla

```
if AIT_01 >=1 & AIT_01 < 5 THEN Ti_pH:=25; end_if;  
if AIT_01 >=5 & AIT_01 < 6 THEN Ti_pH:=28; end_if;  
if AIT_01 >=6 & AIT_01 < 7 THEN Ti_pH:=20; end_if;  
if AIT_01 >=7 & AIT_01 < 8 THEN Ti_pH:=35; end_if;  
if AIT_01 >=8 & AIT_01 < 9 THEN Ti_pH:=30; end_if;  
if AIT_01 >=9 & AIT_01 < 14 THEN Ti_pH:=25; end_if;
```

else

```
kc_pH:=0.8;
```

```
Ti_pH:=10;
```

end\_if;

//Coeficientes equivalente discreto PID

```
q0_pH := kc_pH*(1+(Tm_pH/(2*Ti_pH))+(Td_pH/Tm_pH)*(1-swd_pH));
```

```
q1_pH :=kc_pH*(-1+(Tm_pH/(2*Ti_pH))-(2*Td_pH/Tm_pH)*(1-swd_pH));
```

```
q2_pH :=kc_pH*Td_pH/Tm_pH*(1-swd_pH);
```

```
p0_pH := -sw_acc_pH*kc_pH*(Td_pH/Tm_pH)*swd_pH;
```

```
p1_pH := sw_acc_pH*2*(Td_pH/Tm_pH)*kc_pH*swd_pH;
```

```
p2_pH := -sw_acc_pH*kc_pH*(Td_pH/Tm_pH)*swd_pH;
```

//Cálculo del error

```
ek_pH :=SP_pH-AIT_01;
```

//Acción de Control

```
ek_pH := sw_acc_pH*ek_pH;
```

//Banda de ruido

```

if abs(ek_pH) < Nb_pH then ek_pH :=0; end_if;

//Cálculo delta de control
duk_pH
:=q0_pH*ek_pH+q1_pH*ek1_pH+q2_pH*ek2_pH+p0_pH*AIT_01+p1_pH*yk1_pH+p2_pH*yk2_pH;

//Cálculo signo duk
if duk_pH > 0 then sign_pH :=1;
elsif duk_pH < 0 then sign_pH :=-1;
else sign_pH :=0; end_if;

//Limitador de variación
if abs(duk_pH) > dumax_pH then duk_pH :=dumax_pH; end_if;

//Set point tracking
if sw_am_pH then FV_02 :=uk1_pH+sign_pH*abs(duk_pH);
else
    if sw_st_pH then
        uk1_pH := FV_02; SP_pH := AIT_01; end_if;
    end_if;

//Limitador de mínimo y máximo en actuador
if FV_02 < umin_pH then FV_02 :=umin_pH; end_if;
if FV_02 > umax_pH then FV_02 :=umax_pH; end_if;

//Memoria Actuador
uk1_pH:= FV_02;

//Memoria error
ek2_pH :=ek1_pH;
ek1_pH := ek_pH;

//Memoria PV
yk2_pH :=yk1_pH;
yk1_pH := AIT_01;

```

### **Enclavamiento B-01 y M-01**

```

// b01
if LT_01>10 then b01_aux_N:=1; else b01_aux_N:=0; end_if;
if FV_03>2 then b01_aux_F:=1; else b01_aux_F:=0; end_if;
b01_latch := B_01;
b01_and1 := b01_aux_N & b01_aux_F;
b01_or1 := b01_latch OR Sw_B01;
b01_or2 := sw_b01_stop OR NOT(b01_and1);

```

```
b01_and2 := NOT(b01_or2)& b01_or1;
Sw_B01 := b01_and2;
```

```
// m01
if LT_01>10 then m01_aux_N:=1; else m01_aux_N:=0; end_if;
if FT_01>1 then m01_aux_F:=1; else m01_aux_F:=0; end_if;
m01_latch := M_01;
m01_and1 := m01_aux_N & m01_aux_F;
m01_or1 := m01_latch OR Sw_M01;
m01_or2 := sw_m01_stop OR NOT(m01_and1);
m01_and2 := NOT(m01_or2)& m01_or1;
Sw_M01 := m01_and2;
```

```
aux_b01:=B_01;
aux_m01:=M_01;
```

### **Enclavamientos ejemplo Planta de RILes**

```
// Enclavamiento Bombas
```

```
// P-01
```

```
if P01_start then HS_P01:=1; P01_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P01_stop then HS_P01:=0; P01_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA
```

```
IF LT_03 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_04 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_05 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_06 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_07 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_08 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_P01 := 0; END_IF; // Apagado de bomba por alto nivel
```

```
// P-02
```

```
if P02_start then HS_P02:=1; P02_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P02_stop then HS_P02:=0; P02_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA
```

```
IF LT_03 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_04 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_05 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_06 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
```

```

IF LT_07 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_08 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_P02 := 0; END_IF; // Apagado de bomba por alto nivel

// P-03
if P03_start then HS_05:=1; P03_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P03_stop then HS_05:=0; P03_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA

IF LT_08 > L_max THEN HS_05 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_05 := 0; END_IF; // Apagado de bomba por alto nivel

// P-04

if P04_start then HS_06:=1; P04_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P04_stop then HS_06:=0; P04_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA

IF LT_08 > L_max THEN HS_06 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_06 := 0; END_IF; // Apagado de bomba por alto nivel

// P-15

if P15_start then HS_15:=1; P15_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P15_stop then HS_15:=0; P15_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA

IF LT_03 < L_min THEN HS_15 := 0; END_IF; // Apagado de bomba por bajo nivel
IF LT_04 < L_min THEN HS_15 := 0; END_IF; // Apagado de bomba por bajo nivel
IF LT_05 < L_min THEN HS_15 := 0; END_IF; // Apagado de bomba por bajo nivel
IF LT_06 < L_min THEN HS_15 := 0; END_IF; // Apagado de bomba por bajo nivel
IF LT_07 < L_min THEN HS_15 := 0; END_IF; // Apagado de bomba por bajo nivel

IF LT_08 > L_max THEN HS_15 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_15 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_15 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_15 := 0; END_IF; // Apagado de bomba por alto nivel

//P-27

if P27_start then HS_27:=1; P27_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P27_stop then HS_27:=0; P27_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA

```

```
IF LT_08 > L_max THEN HS_27 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_27 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_27 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_27 := 0; END_IF; // Apagado de bomba por alto nivel
```

```
// P-28
```

```
if P28_start then HS_28:=1; P28_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P28_stop then HS_28:=0; P28_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA
```

```
IF AT_02 >= pH_min AND AT_02 <= pH_max THEN HS_28 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_03 >= pH_min AND AT_02 <= pH_max THEN HS_28 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_04 >= pH_min AND AT_02 <= pH_max THEN HS_28 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_05 >= pH_min AND AT_02 <= pH_max THEN HS_28 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
```

```
IF LT_08 > L_max THEN HS_28 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_28 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_28 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_28 := 0; END_IF; // Apagado de bomba por alto nivel
```

```
IF TT_01 > T_max THEN HS_28 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_02 > T_max THEN HS_28 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_03 > T_max THEN HS_28 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_04 > T_max THEN HS_28 := 0; END_IF; // Apagado de bombas por alta temperatura
```

```
// P-29
```

```
if P29_start then HS_29:=1; P29_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P29_stop then HS_29:=0; P29_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA
```

```
IF TT_01 > T_max THEN HS_29 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_02 > T_max THEN HS_29 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_03 > T_max THEN HS_29 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_04 > T_max THEN HS_29 := 0; END_IF; // Apagado de bombas por alta temperatura
```

```
IF LT_08 > L_max THEN HS_29 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_29 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_29 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_29 := 0; END_IF; // Apagado de bomba por alto nivel
```

// P-30

```
if P30_start then HS_30:=1; P30_start:=0; end_if; // SE PRESIONA LA BOTONERA DE
PARTIDA LA BOMBA PARTE
if P30_stop then HS_30:=0; P30_stop:=0; end_if; // SE PRESIONA LA BOTONERA DE
APAGADO LA BOMBA SE APAGA
```

```
IF AT_02 >= pH_min AND AT_02 <= pH_max THEN HS_30 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_03 >= pH_min AND AT_02 <= pH_max THEN HS_30 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_04 >= pH_min AND AT_02 <= pH_max THEN HS_30 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_05 >= pH_min AND AT_02 <= pH_max THEN HS_30 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
```

```
IF LT_08 > L_max THEN HS_30 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_30 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_30 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_30 := 0; END_IF; // Apagado de bomba por alto nivel
```

```
IF TT_01 > T_max THEN HS_30 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_02 > T_max THEN HS_30 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_03 > T_max THEN HS_30 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_04 > T_max THEN HS_30 := 0; END_IF; // Apagado de bombas por alta temperatura
```

// P-31

```
IF AT_02 >= pH_min AND AT_02 <= pH_max THEN HS_31 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_03 >= pH_min AND AT_02 <= pH_max THEN HS_31 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_04 >= pH_min AND AT_02 <= pH_max THEN HS_31 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
IF AT_05 >= pH_min AND AT_02 <= pH_max THEN HS_31 := 0; END_IF; // Apagado de bomba
por pH entre 6 y 8
```

```
IF LT_08 > L_max THEN HS_31 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_09 > L_max THEN HS_31 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_10 > L_max THEN HS_31 := 0; END_IF; // Apagado de bomba por alto nivel
IF LT_11 > L_max THEN HS_31 := 0; END_IF; // Apagado de bomba por alto nivel
```

```
IF TT_01 > T_max THEN HS_31 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_02 > T_max THEN HS_31 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_03 > T_max THEN HS_31 := 0; END_IF; // Apagado de bombas por alta temperatura
IF TT_04 > T_max THEN HS_31 := 0; END_IF; // Apagado de bombas por alta temperatura
```

**UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA  
RESUMEN DE MEMORIA DE TITULO**

**Departamento** : Departamento de Ingeniería Eléctrica  
**Carrera** : Ingeniería Civil Electrónica  
**Nombre del memorista** : Matías Francisco Martínez Vergara  
**Título de la memoria** : Prefactibilidad Técnica de la Implementación de Programación de PLC en Green I&C  
**Fecha de la presentación oral** : 07/Marzo/2024  
**Profesor(es) guía** : Juan Pablo Segovia Vera  
**Profesor(es) revisor(es)** : José Espinoza, Mario Medina Carrasco  
**Concepto** :  
**Calificación** :

**Resumen (máximo 200 palabras)**

En el contexto de la empresa Green I&C se identificó la oportunidad de evaluar la viabilidad técnica de ofrecer la programación de Controladores Lógicos Programables (PLC) como nuevo servicio. El problema radicaba en la falta de este servicio, lo cual presenta una oportunidad de negocio ya que la empresa cuenta con un área de Instrumentación y sistemas de control. Para abordar esta problemática, se propuso una investigación exhaustiva que incluye la evaluación de software de PLC, la definición de nuevos servicios, la identificación de la infraestructura tecnológica necesaria y la investigación de metodologías eficientes de programación y diseño, además de la adhesión a normas y estándares en el diseño de pantallas HMI. Como resultado se desarrolló un informe que destaca los softwares seleccionados, la infraestructura requerida, flujos de trabajo eficientes y ejemplos prácticos ilustrativos, demostrando así la viabilidad técnica de implementar con éxito la programación de PLC como nuevo servicio en Green I&C, lo que potencialmente aumentará su competitividad en el mercado.