



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL MAESTRO
ESCLAVO UTILIZANDO PLC PARA LA SINCRONIZACIÓN DE LA VELOCIDAD DE
MOTORES TRIFÁSICOS.**

Trabajo de titulación previo a la obtención del

Título de Ingeniero en Electrónica

AUTORES: OSWALDO LEANDRO GÓMEZ BEJARANO

RONNY ANTONIO SANTANA VERA

TUTOR: ING. GEOVANNY GARCÍA FLOR, MSC.

Guayaquil – Ecuador

2025

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Oswaldo Leandro Gómez Bejarano con documento de identificación N°, 0925582389 y Ronny Antonio Santana Vera con documento de identificación N°, 0950669358 manifiesto que:

Somos los autores y responsables del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Guayaquil, 6 de marzo del año 2025.

Atentamente,



Oswaldo Leandro Gómez Bejarano
0925582389



Ronny Antonio Santana Vera
0950669358

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO
DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, Oswaldo Leandro Gómez Bejarano con documento de identificación N°, 0925582389 y Ronny Antonio Santana Vera con documento de identificación N°, 0950669358 expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del artículo académico: "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL MAESTRO ESCLAVO UTILIZANDO PLC PARA LA SINCRONIZACIÓN DE LA VELOCIDAD DE MOTORES TRIFÁSICOS." el cual ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 6 de marzo del año 2025.

Atentamente,



Oswaldo Leandro Gómez Bejarano



Ronny Antonio Santana Vera

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Geovanny García Flor con documento de identificación N° 0922357702, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL MAESTRO ESCLAVO UTILIZANDO PLC PARA LA SINCRONIZACIÓN DE LA VELOCIDAD DE MOTORES TRIFÁSICOS, realizado por Oswaldo Leandro Gómez Bejarano N°, 0925582389 y Ronny Antonio Santana Vera N° 0950669358, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 6 de marzo del año 2025.

Atentamente,

A handwritten signature in blue ink, reading "Geovanny García Flor", is written over a horizontal line.

Ing. Geovanny García Flor, MSc
0922357702

DEDICATORIA

Para Dios que tuve salud y sabiduría durante todo el proceso académico, mi madre y mi madre por darme apoyo emocional todo el tiempo para estar a mi lado en cada paso que doy para dar fuerza para tomar decisiones para avanzar y enseñarme los valores de responsabilidad.

Ronny Santana.

Dedico este trabajo a mi madre, mi hermana por estar cada día a mi lado dando la fuerza para continuar en los momentos más difíciles de la carrera, a mi abuela que, aunque ya no está con nosotros, me cuido y me crio para poder siempre llevar a cabo cada meta que me pongo, aparte de enseñarme que la familia es lo que más importa.

Oswaldo Gómez.

AGRADECIMIENTO

Para Dios, que tengo el poder del trabajo duro a lo largo de mi carrera, para la salud y la bendición, llegando a este proceso en mi proyecto de vida.

Mi familia tuvo un momento difícil durante el viaje y siempre me motivó a seguir adelante.

Mis maestros sobre todo el conocimiento compartido en mi proceso académico. Mi profesor para todo el apoyo proporcionado para este título y apoyo durante el proceso de aprendizaje.

Ronny Santana.

En primer lugar, expreso mi agradecimiento a mi tutor de tesis, Ing. Geovanny por su guía en este trabajo ya que sus conocimientos fueron fundamentales para la culminación de este trabajo.

Quiero agradecer a mi madre por apoyarme en este camino de experiencia que es el estudiar ya que siempre a estado ayudándome como puede en estos años de estudio.

Agradezco a mi hermana que siempre me a aguantando y esta temporada con tantos problemas me a ayudando mucho con sus palabras y su apoyo emocional.

Oswaldo Gomez.

Resumen

Esta tesis desarrolla un sistema de control maestro-esclavo utilizando PLCs para sincronizar la velocidad de motores trifásicos en líneas de producción de empresas empacadoras, con el fin de mantener una velocidad constante en las bandas transportadoras sin detener la planta, se diseña una arquitectura de control donde un PLC maestro coordina varios PLCs esclavos, cada uno controlando un motor trifásico, utilizando una red Profinet para garantizar una comunicación eficiente.

El control de la velocidad del motor se introduce a través de cambios de frecuencia utilizando algoritmos PID y PI. Además, se crea una interfaz HMI centralizada para facilitar el monitoreo y la configuración del sistema. A través de pruebas experimentales, se verificará la efectividad del sistema en la sincronización de los motores, mostrando que mejora la coordinación y reduce el desgaste mecánico.

Este sistema optimiza recursos, reduce costos operativos al evitar nuevas líneas de producción y aumenta la flexibilidad de la empresa para adaptarse a los cambios de la demanda. En resumen, ofrece una solución eficiente, rentable y robusta, fortaleciendo la productividad de la empresa en el mundo laboral.

Palabras claves: Motores trifásicos, Bandas transportadoras, Profinet, Control PID, Control PI

Abstract

This thesis develops a master-slave control system using PLCs to synchronize the speed of three-phase motors in production lines of packaging companies, in order to maintain a constant speed on the conveyor belts without stopping the plant, a control architecture is designed where a master PLC coordinates several slave PLCs, each controlling a three-phase motor, using a Profinet network to ensure efficient communication.

The speed control of the motors is implemented by frequency inverters, using PID and PI algorithms. In addition, a centralized HMI interface is created that facilitates monitoring and configuration of the system. Through experimental tests, the effectiveness of the system in synchronizing the motors will be verified, showing that it improves coordination and reduces mechanical wear.

This system optimizes resources, reduces operating costs by avoiding new production lines and increases the company's flexibility to adapt to changes in demand. In short, it offers an efficient, cost-effective and robust solution, strengthening the company's competitiveness in the market.

Key words: Three-phase motors, Conveyor belts, Profinet, PID Control, PI Control, TIA Portal

ÍNDICE GENERAL

I.	INTRODUCCIÓN	1
II.	PROBLEMA.....	2
III.	JUSTIFICACIÓN	3
IV.	OBJETIVOS	4
4.1	Objetivo general.....	4
4.2	Objetivos específicos	4
V.	FUNDAMENTO TEÓRICO	5
5.1	SIMATIC HMI	5
5.2	PLC	5
5.3	Motores trifásicos de inducción tipo ardilla.....	7
5.3.1	Principio de Funcionamiento	7
5.4	Encoder	8
5.4.1	Tipos de Encoder	9
5.4.1.1	Encoders magnéticos	9
5.4.1.2	Encoders ópticos	10
5.4.1.3	Encoders inductivos	11
5.5	Variador de velocidad V20	12
5.6	Variador G120	12
5.7	Tía portal.....	13

5.8	Controlador PID.....	14
5.9	Controlador PI.....	14
5.10	PID compact.....	15
5.11	Comunicación PROFINET	15
5.12	Protocolos que usan los variadores.	16
5.12.1	Comunicación de interfaz serie universal (USS)	16
5.12.2	Protocolo Modbus RTU.....	17
VI	MARCO METODOLÓGICO.....	19
6.1	Diseño de un sistema de control maestro esclavo utilizando PLC para la sincronización de la velocidad de motores trifásicos utilizando TIA PORTAL V18.....	19
6.1.1	Investigación sobre el programa para la creación de la lógica del proyecto.	19
6.1.2	Creación del proyecto en TIA PORTAL V18	20
6.1.3	Guardar proyecto creado	21
6.1.4	Creación del hardware del PLC S71200 en el árbol del proyecto del TIA PORTAL V18. 22	
6.1.5	Configuración de red del PLC.	24
6.1.6	Programación de los PLCs en Tia PORTAL V18.	25
6.1.6.1	Parametrización del bloque de datos en PLC “Master” Datos_HMI[DB2] y RECETAS[DB1].....	26
6.1.6.2	Parametrización del bloque de datos en PLC “Slave_1” Datos_Slave_1[DB5].	27
6.1.6.3	Parametrización del bloque de funcion en PLC “Slave_1” Filtro[FB1].....	28

6.1.6.4	Parametrización de función en PLC “Slave_1” Control_VFD_1[FC2].	28
6.1.6.5	Parametrización del bloque de datos en PLC “Slave_2” Datos_Slave_2[DB5].	29
6.1.6.6	Parametrización del bloque de función en PLC “Slave_2” Filtro[FB1].	30
6.1.6.7	Parametrización de función en PLC “Slave_2” Control_VFD_2[FC1].	30
6.1.6.8	Parametrización de los bloques PUT y GET.	31
6.2	Diseñar un sistema de control PID y PI en TIA PORTAL para el control de velocidad de los motores trifásicos usando variadores de frecuencia.	33
6.2.1	Parametrización del Variador G120.	34
6.3	Elaborar un HMI centralizado para el monitoreo y configuración de la sincronización de velocidad de los motores trifásicos.	37
6.3.1	Configuración de la comunicación del KTP700 con el PLC S7-1200.	37
6.3.2	Creación de las pantallas.	38
6.3.3	Diseño de pantallas.	40
6.3.4	Insertar gráficos.	41
6.3.5	Insertar objetos básicos.	41
6.3.6	Insertar botones.	42
6.3.7	Insertar lista de texto.	43
6.3.8	Insertar campos de entrada/salida.	44
6.3.9	Creación de animaciones.	46
6.3.10	Creación de recetas.	46
6.3.11	Creación de tendencias.	48

VII	RESULTADOS.....	50
7.1	Ejecucion del proyecto en TIA PORTAL V18 para la revision de la comunicaci3n entre los PLCs.....	50
7.2	Comparativa de controlador PID y PI en el sistema de control de velocidad para motores trifasicos.	53
7.3	HMI centralizada para sincronizacion de motores trifasicos.	56
VIII	CRONOGRAMA DE ACTIVIDADES	64
IX	PRESUPUESTO	65
X	CONCLUSIONES	66
XI	RECOMENDACIONES.....	67
XII	REFERENCIA BIBLIOGRÁFICAS	68
XIII	ANEXOS	73
13.1	Anexo 1. Programaci3n de PLC Master (Main_Master [OB1]).....	73
13.2	Programaci3n de PLC Slave_1	76
13.2.1	Anexo 2. Programaci3n del Main_Slave1[OB1]	76
13.2.2	Anexo 3. Programaci3n del bloque Cyclic interrupt [OB30].	78
13.2.3	Anexo 4. Programaci3n del bloque Control_VDF_1 [FC2].....	79
13.2.4	Anexo 5. Programaci3n del bloque filtro [FB1].	80
13.3	Programaci3n de PLC Slave_2	81
13.3.1	Anexo 6. Programaci3n del Main_Slave2[OB1]	81
13.3.2	Anexo 7. Programaci3n del Cyclic interrupt[OB30]	83
13.3.3	Anexo 8. Programaci3n del Control_VFD_2[FC1].....	84

13.3.4	Anexo 9. Programación del Filtro [FB1]	85
--------	--	----

ÍNDICE DE FIGURAS

Figura 1.	Pantalla HMI.	5
Figura 2.	PLC S7-1200.	6
Figura 3.	Motor trifásico.	8
Figura 4.	Encoder.	8
Figura 5.	Encoder compuesto por discos y optoacoplador.	9
Figura 6.	Encoder Magnético.	10
Figura 7.	Encoder óptico.	10
Figura 8.	Encoder inductivo.	11
Figura 9.	Variador V20.	12
Figura 10.	Variador G120.	13
Figura 11.	Programa Tia Portal.	13
Figura 12.	Control PID.	14
Figura 13.	Compact PID.	15
Figura 14.	Comunicación PROFINET.	16
Figura 15.	Protocolo USS.	17
Figura 16.	Protocolo Modbus RTU.	18
Figura 17.	Icono del Software.	20
Figura 18.	Creación de nuevo proyecto en TIA PORTAL V18.	20
Figura 19.	Seleccionar destino final de un proyecto en TIA PORTAL V18.	21
Figura 20.	Guardar un proyecto creado en TIA PORTAL V18.	21
Figura 21.	Configuración de dispositivos en TIA PORTAL V18.	22

Figura 22. Selección de dispositivos de trabajo.....	22
Figura 23. Elección de los dispositivos a usar.	23
Figura 24. Visualización de dispositivos en árbol de proyectos.	24
Figura 25. <i>Configuración del PLC “Master”</i>	25
Figura 26. Creación de bloques de funciones.	26
Figura 27. Parametrización del DB “Datos_HMI”.....	27
Figura 28. Parametrización del DB “Datos_Slave_1”.....	27
Figura 29. Parametrización del FB “Filtro”.....	28
Figura 30. Parametrización del FC “Control_VFD_1”.....	29
Figura 31. Parametrización del DB “Datos_Slave_2”.....	29
Figura 32. Parametrización del FB “Filtro”.....	30
Figura 33. Parametrización del FC “Control_VFD_2”.....	31
Figura 34. Bloque PUT en “Master”.....	31
Figura 35. Bloque PUT en “Master”.....	32
Figura 36. Bloque PUT en “Slave_1”.....	32
Figura 37. Bloque PUT en “Slave_1”.....	33
Figura 38. Parametrización del Variador G120.	34
Figura 39. Parametrización del Variador G120.	34
Figura 40. Configuración del Variador V20.	35
Figura 41. Configuración del tipo de control del PID_Compact.	36
Figura 42. <i>Configuración del rango límite del PID_Compact.</i>	36
Figura 43. Configuración del tipo de controlador PID o PI del PID_Compact.	36
Figura 44. Añadimos el HMI KTP700.	37
Figura 45. Configuración de la comunicación del HMI KTP700.....	38
Figura 46. Configuración de la comunicación del HMI.	38
Figura 47. Creación de pantallas.....	39

Figura 48. Configuración de la plantilla de las pantallas.....	39
Figura 49. Configuración de la plantilla de las pantallas.....	40
Figura 50. Cambio de nombre de pantalla.	40
Figura 51. Creación de un gráfico.....	41
Figura 52. Colocar un objeto.	42
Figura 53. Colocación de botones.....	42
Figura 54. Colocación de botones.....	43
Figura 55. Creación de lista de texto.	43
Figura 56. Parametrización de lista de texto.	44
Figura 57. Colocación de los campos de entrada/salida.	45
Figura 58. Configuración de los campos de entrada/salida.	45
Figura 59. Creación de animación para gráficos.	46
Figura 60. Configuración del bloque de datos “RECETAS”.....	47
Figura 61. Creación de recetas.....	48
Figura 62. Colocación del “Recipe view”.....	48
Figura 63. Creación de tendencias.	49
Figura 64. Icono del Proyecto en TIA PORTAL V18.	50
Figura 65. Device & Networks conexión de los PLCs.	51
Figura 66. Ubicación de Download to device.....	51
Figura 67. Cargar Programa a PLC.....	52
Figura 68. Sistema apagado	52
Figura 69. Chequeo de encendido del sistema y comunicación correcta entre PLCs.....	53
Figura 70. Como ingresar a la herramienta para crear controladores.	53
Figura 71. Proceso creación un controlador para el sistema.....	54
Figura 72. Grafica del control PI en motor 1.	55
Figura 73. Grafica del control PI en motor 2.	56

Figura 74. Pantalla de Presentación.	57
Figura 75. Pantalla de Inicio.	57
Figura 76. Pantalla de Proceso.	58
Figura 77. Pantalla de Motor 1.	58
Figura 78. Pantalla de Grafico Motor 1.	59
Figura 79. Pantalla de Motor 2.	59
Figura 80. Pantalla de Grafico Motor 1.	60
Figura 81. Pantalla de Recetas Operario.	60
Figura 82. Pantalla de Rectas para el Administrador.	61
Figura 83. Pantalla de Proceso en funcionamiento.	61
Figura 84. Pantalla en funcionamiento de Motor 1.	62
Figura 85. Pantalla en funcionamiento de Motor 2.	62
Figura 86. Pantalla en funcionamiento de Recetas Administrador.	63
Figura 87. Proyecto Físico Final.	63

ÍNDICE DE TABLAS

Tabla 1. Tabla de datos de los Controladores PID.....	54
Tabla 2. Tabla de datos de los Controladores PI.....	55
Tabla 3, Cronograma de actividades para el desarrollo del Proyecto de Titulación.	64
Tabla 4. Presupuesto estimado para el desarrollo del Proyecto de Titulación.....	65

I. INTRODUCCIÓN

La industria empacadora enfrenta desafíos significativos al intentar expandir su catálogo de productos sin incurrir en altos costos de infraestructura, una solución común es utilizar las líneas de producción existentes para manejar una variedad de productos en diferentes presentaciones, en este contexto, las bandas transportadoras accionadas por motores trifásicos son esenciales.

La sincronización de los motores trifásicos es crucial para evitar problemas mecánicos y mantener la operatividad. Una falta de sincronización puede provocar variación de tensiones y desequilibrios que dañan tanto las bandas transportadoras como los productos. Estos problemas, si no se abordan de manera oportuna, pueden llevar a tiempos de inactividad prolongados y costosos, afectando la rentabilidad de la empresa.

Para enfrentar estos desafíos, esta tesis propone el diseño e implementación de un sistema de control maestro-esclavo utilizando Controladores Lógicos Programables (PLC). Este sistema permite la sincronización de la velocidad de motores trifásicos, asegurando una operación estable entre los motores.

La implementación del control de velocidad se realiza mediante algoritmos PID y PI programados en TIA Portal, utilizando variadores de frecuencia para ajustar la velocidad de los motores con precisión. Además, Se ha desarrollado una interfaz HMI centralizada para permitir a los trabajadores monitorear y configurar el sistema en tiempo real, lo que facilita la desviación y la corrección.

II. PROBLEMA

En la producción, los sistemas de manufactura son clave para mejorar la eficiencia y competitividad. La optimización de procesos, a través de la implementación inteligente de tecnología y gestión, permite maximizar la productividad y transformar la manera en que se ejecuta la fabricación. Este enfoque no solo mejora la operatividad, sino que también impulsa la competitividad y rentabilidad de las empresas en un mercado dinámico y exigente. (LONDON CONSOLTING GROUP, 2024)

Los motores trifásicos son un elemento esencial en la industria empacadora, en la actualidad se puede observar en prácticamente en todas las distintas áreas de producción, en combinación con diversos mecanismos electrónicos forman enormes máquinas que facilitan el trabajo humano. (Cuzcano Rivas, 2023)

Al no tener una correcta sincronización de las velocidades de los motores se pueden producir fallos uno o a los dos rotores, es posible que se pueda perder las bandas transportadoras conectadas a los motores, estos fallos en muchos casos, causan retraso en la productividad que causan pérdidas financieras debido a las reparaciones de los diferentes componentes dañados. (Quispe Mendoza, 2021)

Teniendo en cuenta esto un sistema de bandas transportadoras al tener un fallo en sus motores tendría que detenerse por completo si existe un problema en algún motor lo más pronto posible para evitar daños en las bandas o en el producto. (Hwang Cárdenas & Iñiguez Avila, 2023)

III. JUSTIFICACIÓN

En el ámbito de las empacadoras la sincronización de los motores trifásicos podría ser un aspecto crítico para el correcto funcionamiento de los procesos de transporte de los diversos productos que manejan, La introducción de un sistema de control basado en PLC se mostrará como una posible solución, este sistema, preconfigurado con distintas velocidades asociadas a las variadas presentaciones de productos en las bandas transportadoras.

La adopción de este sistema proporcionaría un entorno de monitoreo y control centralizado, lo que facilitaría la detección y corrección de posibles desviaciones en tiempo real, esto podría minimizar los fallos en los procesos y a mantener la eficiencia operativa de la planta, al controlar la velocidad de los motores se podrían evitar desequilibrios en la carga de trabajo, lo que podría conducir a una reducción del consumo de energía, por ende, a una disminución de los costos operativos.

Desde una perspectiva económica, la implementación de este sistema tendría un beneficio, al eliminar la necesidad de invertir en nuevas líneas de bandas transportadoras para productos diferentes. Esto conllevaría a evitar los costos asociados con la infraestructura y los equipos adicionales requeridos para tal fin. Como resultado, la empresa puede optimizar sus recursos y mejorar su rentabilidad.

Además de los beneficios económicos, esta medida también podría implementarse en la implementación de un resumen en este sistema de control de visión maestra para mejorar las operaciones y reducir los costos, pero también para fortalecer la posición de la empresa en el mercado y su capacidad para enfrentar desafíos futuros.

IV. OBJETIVOS

4.1 Objetivo general

Diseñar e implementar un sistema de control maestro-esclavo utilizando PLC para la sincronización de la velocidad de motores trifásicos.

4.2 Objetivos específicos

- Establecer una red de comunicación Profinet entre los PLCs esclavos y el PLC maestro para lectura y control de velocidad.
- Diseñar un sistema de control PID y PI en TIA PORTAL para el control de velocidad de los motores trifásicos usando variadores de frecuencia.
- Elaborar un HMI centralizado para el monitoreo y configuración de la sincronización de velocidad de los motores trifásicos.

V. FUNDAMENTO TEÓRICO

5.1 SIMATIC HMI

La interfaz HMI o Man-Machine es un sistema diseñado para facilitar la convivencia entre el usuario humano y la máquina. También llamado MMI (interfaz de hombres-máquina), HMI actúa como una conexión de comunicación entre el usuario y el sistema, proporcionando una forma clara y efectiva de controlar, monitorear y ajustar los dispositivos y procesos. Se cree que este es el valor básico de la informática industrial. (IronTech, 2023).

Figura 1. *Pantalla HMI.*



Nota: Parte delantera Pantalla Siemens (Bouchakri Zaamy, 2021).

El objetivo principal del HMI es reducir la complejidad de las operaciones automatizadas. Ofrece una interfaz gráfica intuitiva que facilita a los usuarios la interacción con sistemas, máquinas o procesos complejos de forma clara y accesible. (IronTech, 2023)

5.2 PLC

El PLC puede ser visto como un sistema integral que engloba todos los elementos esenciales para su funcionamiento, y se adecua a entornos industriales adversos gracias a sus

propiedades diseñadas específicamente para este tipo de ambiente, el PLC está vinculado con la maquinaria empleada en procesos de fabricación y se responsabiliza de regular su operatividad. (GSL Industrias, 2021).

Figura 2. PLC S7-1200.



Nota: Estructura externa del PLC. (Peláez, 2024)

La sigla PLC deriva del concepto en inglés "Programmable Logic Controller" (Controlador Lógico Programable), la implementación de los PLC ha tenido un efecto notable en la industria al simplificar la adaptabilidad de los procesos de fabricación y supervisión, los PLC son empleados no solo en ambientes industriales, sino también en hogares, centros comerciales, hospitales y centros educativos. (GSL Industrias, 2021)

5.3 Motores trifásicos de inducción tipo ardilla

Los motores de inducción de jaula de ardilla son ampliamente empleados en diversas aplicaciones industriales y comerciales debido a su robustez, eficiencia y diseño sencillo. Su nombre se debe a la forma característica de su rotor, que recuerda a una jaula de ardilla debido a su estructura. (Magnetism, 2024)

5.3.1 Principio de Funcionamiento

El motor de jaula de ardilla es un tipo de motor de inducción, lo que implica que opera según los principios del electromagnetismo. En su estructura, se distinguen principalmente dos componentes: el estator y el rotor. (Magnetism, 2024)

Estator: Es una parte sólida y actúa como un fondo de un motor que consiste en una casa con una corona de acero de hierro o silicio, donde hay algunas ranuras porque es un motor de tres fases, encontramos tres bobinas y tres cadenas diferentes, cada circuito tiene la misma cantidad de bobinas. (S&P, 2024)

Rotor: El rotor, ubicado dentro del estator, está compuesto por barras conductoras, usualmente de aluminio o cobre, que se insertan en ranuras y se conectan en cortocircuito en ambos extremos, dándole la forma de una jaula de ardilla. Cuando el campo magnético giratorio del estator interactúa con las barras del rotor, se induce una corriente en ellas según la ley de Faraday de la inducción electromagnética. (Magnetism, 2024)

Figura 3. Motor trifásico.



Nota: ejemplo de motor trifásico. (Farina, 2018)

5.4 Encoder

En entornos industriales, es frecuente encontrarse con la tarea de medir distancias para verificar la correcta ubicación de herramientas o productos, lo cual es fundamental para la seguridad en los sistemas industriales, la medición puede ser absoluta o relativa y es crucial para garantizar la precisión en la electrónica, donde incluso pequeñas disparidades pueden acarrear consecuencias significativas. (ROSALENY, 2021)

Figura 4. Encoder.

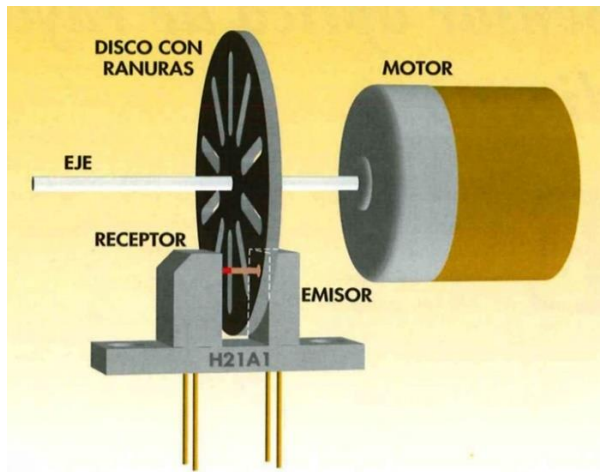


Nota: ejemplo de Encoder. (Sensor, 2021)

La operación fundamental de un Encoder se basa en los optoacopladores, que son elementos integrados conformados por un diodo emisor de luz y un transistor receptor, los

optoacopladores tienen la función de identificar si hay presencia o disminución de luz a través de los discos concéntricos al eje del Encoder, los cuales cuentan con muescas que posibilitan el paso de la luz de forma codificada, lo que ayuda a obtener la medida final requerida. Este proceso se ilustra en la Figura 7. (ROSALENY, 2021)

Figura 5. *Encoder compuesto por discos y optoacoplador.*



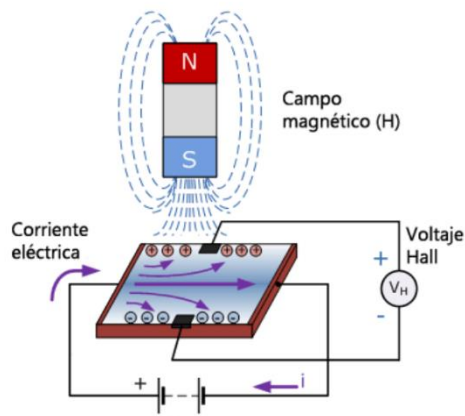
Nota: ejemplo de discos y optoacoplador. (Plata, Ingenieria Creativa, 2014)

5.4.1 Tipos de Encoder

5.4.1.1 Encoders magnéticos

Hay Encoders magnéticos que usan este principio físico para leer algunas de las marcas premagnetizadas, la regla de medición se polariza para tener un orden continuo de norte-sur; De esta manera, el lector que se mueve por encima de la regla puede determinar el aumento en la condición, afectando los campos magnéticos externos y las propiedades endógenas del magnetismo, como la histéresis, reduciendo su persistencia y repetición. (SINADRIVES, 2020)

Figura 6. *Encoder Magnético.*



Nota: ejemplo funcionamiento de encoder magnético. (PP, 2020)

5.4.1.2 Encoders ópticos

Los Encoders ópticos utilizan la refracción de un haz de luz sobre la regla de medición. Dicha regla contiene una combinación de tramos reflectantes y opacos, que provocan que el haz de luz del cabezal se refleje y sea detectado, o que no haya detección. De esta forma el cabezal puede contar los incrementos de posición. La tecnología óptica permite alcanzar precisiones muy altas, y es inmune a los campos magnéticos. Sin embargo, es sensible a la suciedad y, puesto que ofrece unas prestaciones muy altas, el precio acostumbra a ser elevado. (SINADRIVES, 2020)

Figura 7. *Encoder óptico.*

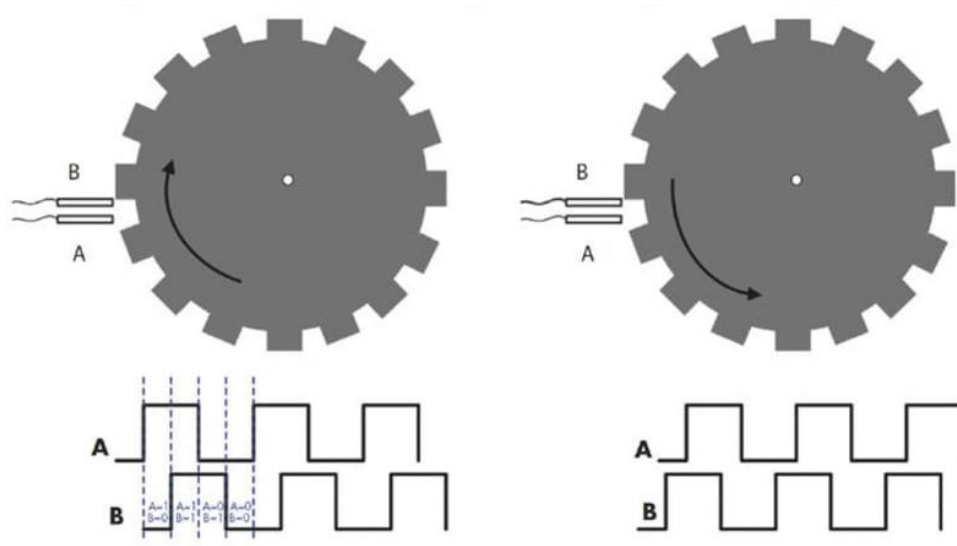


Nota: ejemplo de Encoder óptico. (ARROW, 2019)

5.4.1.3 Encoders inductivos

Como tercera opción tenemos los Encoders inductivos, el principio de funcionamiento no difiere de los dos anteriores en forma, pero si en el principio de detección. En este caso la regla es metálica y tiene ranuras mecanizadas a intervalos constantes; y el cabezal lector detecta la inductancia, que varía su valor en función de si en la regla hay metal o una ranura. (SINADRIVES, 2020)

Figura 8. Encoder inductivo.



Nota: ejemplo de Encoder inductivo. (Tech Briefs, 2019)

Este sistema es increíblemente robusto, debido a su inmunidad a los campos magnéticos, a la suciedad e incluso a los golpes, no alcanza las precisiones de los sistemas ópticos, pero se acerca, lo cual le convierte en un firme candidato en la mayoría de las aplicaciones de automatización. (SINADRIVES, 2020)

5.5 Variador de velocidad V20

El variador V20 es un equipo de control empleado en entornos industriales para regular la velocidad de motores eléctricos, se utiliza en diversos sectores industriales, como la manufactura, la automatización de procesos y la energía, este dispositivo destaca por su capacidad para mejorar la eficiencia energética al adecuar la velocidad del motor, permitiendo adaptarse a distintas cargas y optimizar el desempeño del sistema en el que se aplica. (Loayza Rodríguez & Mendiola Villanueva, 2020)

Figura 9. Variador V20.



Nota: ejemplo de variador V20. (Eltra Trade, 2023)

5.6 Variador G120

La Sinámica G120 es un variador universal que responde a varios requisitos industriales y comerciales para bombear, ventilar, comprimirse, mover o procesar. La industria automotriz, textiles, envases y la industria automotriz generalmente depende de las soluciones probadas de Sinámica G120. (Eltra Trade, 2023)

Su diseño modular que consiste en un dispositivo de ajuste (control del dispositivo, CU) y el módulo de potencia (modo de alimentación, PM) en el rango de potencia de 0.37 kW de hasta 250 kW lo convierte en el sistema perfecto para aplicaciones estandarizadas. La gran cantidad de componentes disponibles le permite crear la opción óptima para sus necesidades. (Granda Rivera Jasson Stalin, 2025)

Figura 10. Variador G120.



Nota: ejemplo de variador G120. (Escuale Control+, 2023)

5.7 Tía portal

El Portal de Automatización Totalmente Integrada es una plataforma de ingeniería desarrollada por Siemens para automatizar dispositivos que brinda una amplia gama de opciones. A través del TIA Portal se pueden acceder a diversas aplicaciones, siendo utilizadas en este proyecto las que se detallan en las secciones siguientes. (Satorres Mira, 2021)

Figura 11. Programa Tia Portal.

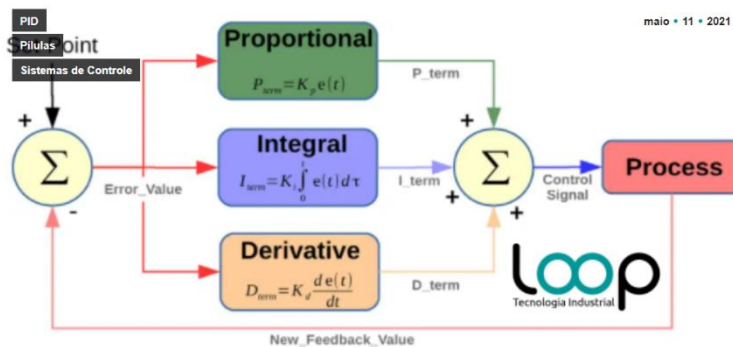


Nota: logotipo de programa Tia portal. (siemens, 2021)

5.8 Controlador PID

Un controlador PID es un sistema de regulación que emplea un actuador para mantener una variable o proceso en un valor deseado dentro del rango medido por un sensor. Es uno de los métodos más utilizados y precisos en el campo del control automático. Este tipo de control proporciona una solución eficiente para sistemas que necesitan ajustar su respuesta tanto durante el transitorio como en el estado estacionario. (Smith, 2024)

Figura 12. Control PID.



Nota: estructura interna del PID. (Itiengenharia, 2021)

5.9 Controlador PI

Un controlador PI es descrito por la función de transferencia:

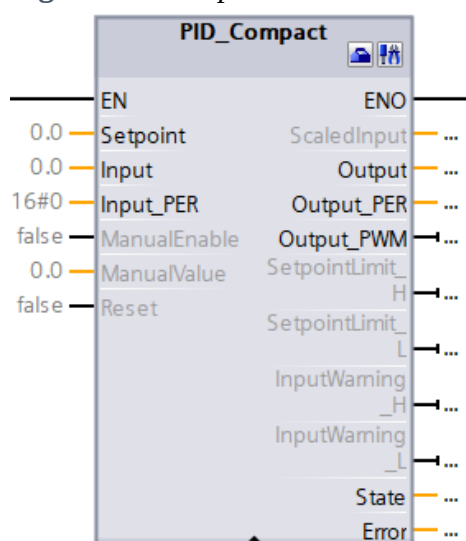
$$K(s) = k_p + \frac{k_i}{s} = \frac{k_p(s + k_i/k_p)}{s}$$

El protocolo fue creado y publicado por Modicon en 1979 para ser utilizado con controladores lógicos programables. Desde entonces, ha sido adoptado y se utiliza de manera extensa en la construcción de sistemas de gestión de edificios (BMS) y sistemas de automatización industrial (IAS). Su amplia aceptación se debe a su facilidad de uso, fiabilidad y al hecho de ser un protocolo de código abierto, lo que permite su uso gratuito en cualquier dispositivo. (Smith, 2024)

5.10 PID compact

El objeto tecnológico compacto ofrece un regulador PID continuo con optimización integrada. El objeto determina continuamente la medición del bucle de ajuste real y se compara con el eslogan deseado. Desde el error regulatorio resultante, las instrucciones PID_Compact calculan el valor de salida, que está nivelado con un eslogan a la velocidad y estabilidad máxima. (infopl, 2021)

Figura 13. Compact PID.



Nota: Bloque de PID_compact. (Thanh, 2021)

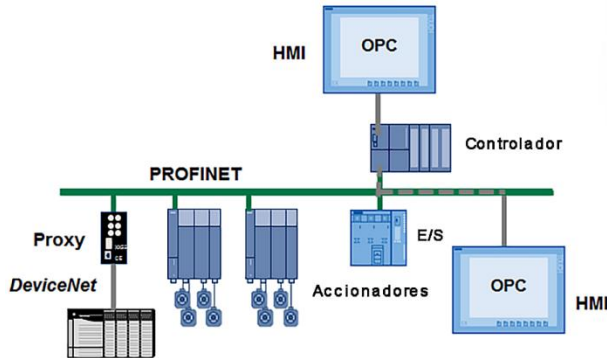
5.11 Comunicación PROFINET

Profinet (Process Field Network) es un protocolo de comunicación creado por la organización Profibus & Profinet International (PI) para interconectar dispositivos industriales con diversos tipos de equipos de producción. Este equipo fue diseñado para ser independiente del usuario fabricante, lo que permite que equipos de diferentes marcas puedan funcionar juntos sin problemas. (Profibus.Profinet, 2023)

Profinet puede ser utilizado por TCP/IP (o UDP/IP) para notificar a ciertas tareas que no son críticas, como la configuración, la parametrización y los diagnósticos enviando datos

usando TCP/IP, agregando información adicional al paquete cuando el "Pase" a través del modelo ISO/OSI es relativamente grande. (Bowne, PROFIBUS.PROFINET, 2020)

Figura 14. *Comunicación PROFINET.*



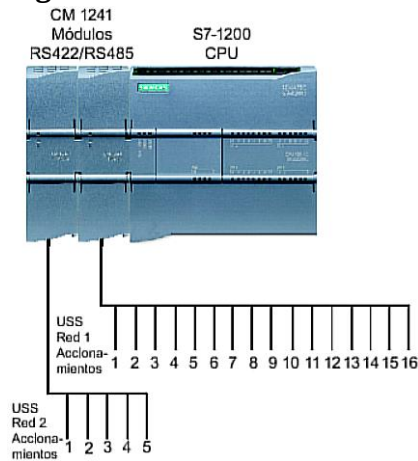
Nota: Esquema de red PROFINET (Bowne, PROFIBUS.PROFINET, 2020)

5.12 Protocolos que usan los variadores.

5.12.1 Comunicación de interfaz serie universal (USS)

Las instrucciones USS gestionan el funcionamiento de los sistemas de accionamiento motorizados que emplean el protocolo de interfaz serie universal (USS). Este protocolo se basa en una estructura de comunicación maestro-esclavo a través de un bus serie. El maestro envía un mensaje a un esclavo específico utilizando un parámetro de dirección, y un esclavo nunca puede transmitir sin haber recibido previamente una solicitud. No está permitida la transferencia directa de mensajes entre los distintos esclavos. (Siemens, 2018)

Figura 15. *Protocolo USS.*



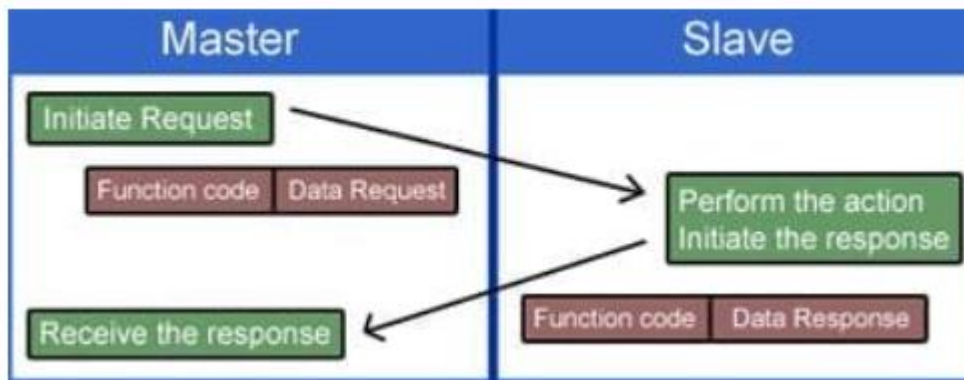
Nota: Esquema del Protocolo de comunicación USS (Siemens, 2018)

5.12.3 Protocolo Modbus RTU.

El protocolo Modbus consiste en una armadura en serie para transmitir y recibir datos de control que toma la arquitectura principal/esclava entre la unidad y el control de los instrumentos de campo. Como uno de los más antiguos, se ha convertido en uno de los más utilizados en aplicaciones industriales debido a su funcionalidad y bajos costos de mercado en comparación con otros protocolos. (Universidad Autonoma del Estado de Hidalgo, 2017)

El protocolo fue desarrollado y publicado en Modicon en 1979 para usarlo con controladores lógicos programables, este protocolo se ha adoptado y se utiliza ampliamente para la construcción de sistemas de gestión de edificios (BMS) y sistemas de automatización industrial (IAS). Su aceptación ha sido facilitada por su facilidad de uso, confiabilidad y el hecho de que es de código abierto y puede usarse de forma gratuita para cualquier unidad. (Weis, 2023)

Figura 16. Protocolo Modbus RTU.



Nota: Esquema del Protocolo de comunicación *Modbus RTU* (Logicbus, 2019)

COMENTARIO

6.12.3 Sincronización de motores.

La sincronización es contar con un sistema que permita obtener movimientos simultáneos y repetitivos de una máquina que realiza un proceso de producción, los motores pueden ser sincronizados mediante el uso de autómatas programables y variadores de frecuencia, favoreciendo así un proceso secuencial a través de una serie de estados de funcionamiento y operación. (YUQUILEMA TRUJILLO RICARDO XAVIER, 2019)

VI MARCO METODOLÓGICO

6.1 Diseño de un sistema de control maestro esclavo utilizando PLC para la sincronización de la velocidad de motores trifásicos utilizando TIA PORTAL V18.

El diseño de un sistema de control maestro-esclavo con PLC para sincronizar la velocidad de motores trifásicos utilizando TIA PORTAL V18 implica la configuración de un controlador que actúa como maestro y varios controladores esclavos, en este sistema, el PLC maestro coordina y ajusta la velocidad de los motores trifásicos conectados a los PLC esclavos, asegurando que todos los motores operen a la misma velocidad y mantengan una sincronización precisa.

Utilizando TIA PORTAL V18, se programan las estrategias de control, se configuran los parámetros de comunicación entre los PLCs y se desarrollan algoritmos de control para mantener la coherencia en la velocidad de los motores.

6.1.1 Investigación sobre el programa para la creación de la lógica del proyecto.

En esta etapa del proyecto se hace un análisis del proceso de las bandas transportadoras, los componentes que se usan en su funcionamiento, diagramas eléctricos y los niveles de comunicación que existen en el momento del funcionamiento de las bandas transportadoras.

Después del análisis, se recomiendan requisitos técnicos como el tipo de sensor y controlador que se utilizará en el proyecto, se recomienda un sistema de control automático con información de monitoreo de información y recopilación de información sobre el motor que se utilizará, HP máximas, variaciones óptimas para su operación y el PLC correcto para controlar el variador.

6.1.2 Creación del proyecto en TIA PORTAL V18

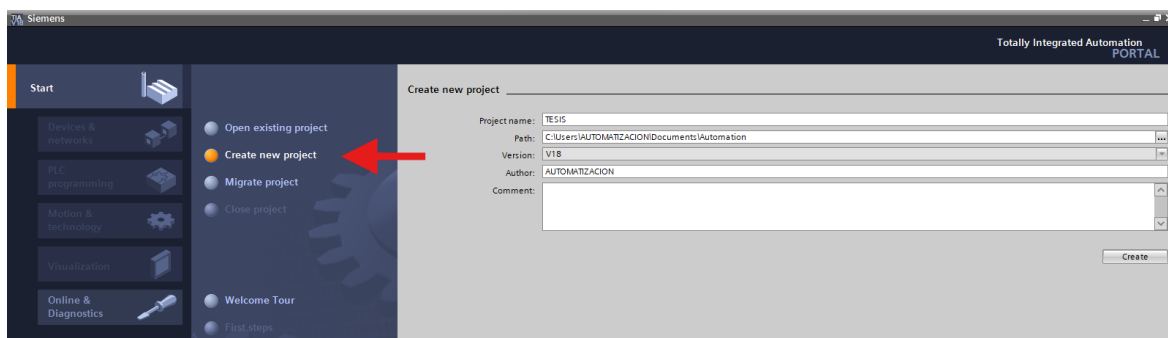
Para comenzar haga clic en el icono de acceso de escritorio de Windows para iniciar el proyecto. Luego cree una nueva configuración del proyecto que esté cerrada cuando terminemos, necesitamos nombrar el proyecto en el cuadro llamado "Nombre del proyecto" como se describe en la Figura 17.

Figura 17. *Icono del Software.*



Nota. En la figura, se muestra el icono del software TIA PORTAL V18.

Figura 18. *Creación de nuevo proyecto en TIA PORTAL V18.*

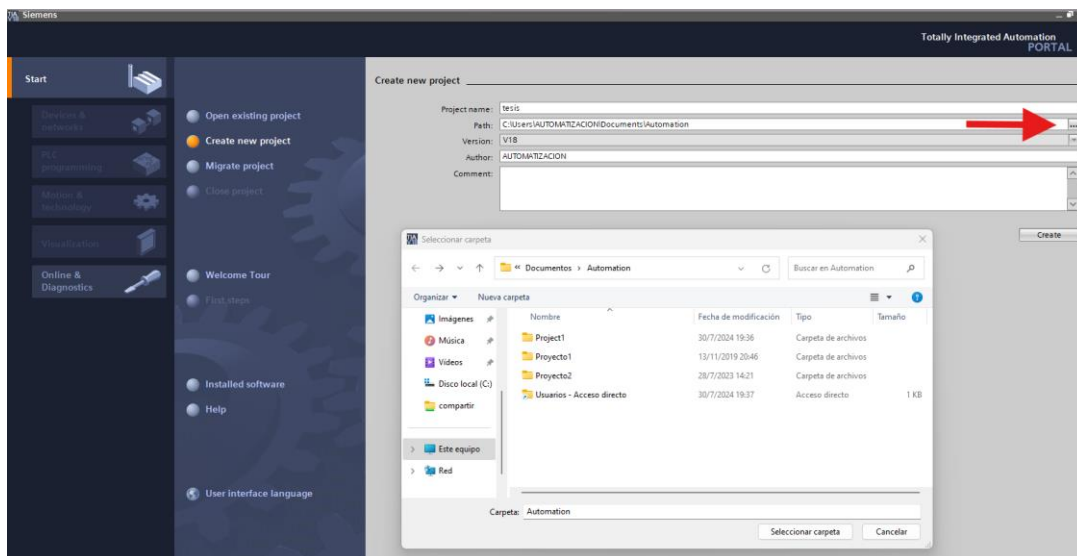


Nota. En la ventana de inicio del programa TIA PORTAL V18, se muestra una flecha de color rojo que indica la opción a seleccionar para iniciar un nuevo proyecto.

6.1.3 Guardar proyecto creado

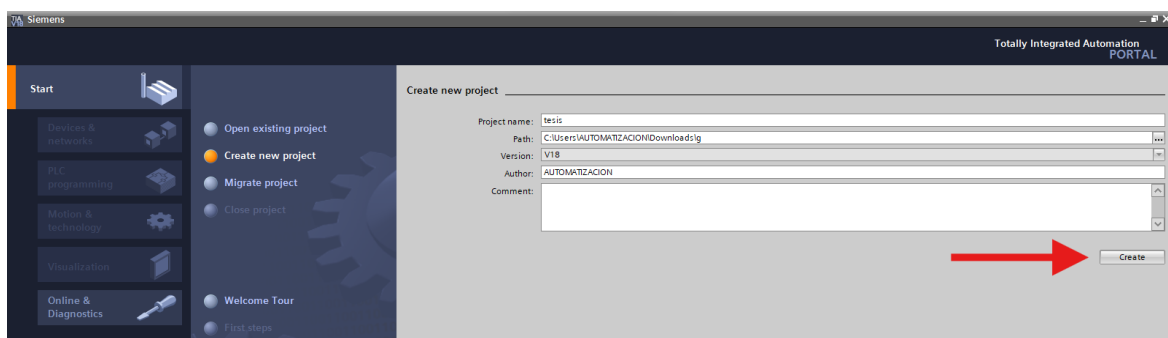
Después de crear un nuevo proyecto, seleccionamos la ruta donde se guardará el proyecto, visualizamos el cuadro llamado "path", hacemos clic y busca preferencias, seleccione la carpeta y finalmente haga clic en la opción "Crear" para guardar el proyecto en la ruta elegida, como se visualiza en la figura 18.

Figura 19. Seleccionar destino final de un proyecto en TIA PORTAL V18.



Nota. En la imagen con una flecha roja, resulta que la oportunidad de ser seleccionada para buscar un lugar dentro de la computadora donde desea guardar el proyecto.

Figura 20. Guardar un proyecto creado en TIA PORTAL V18.



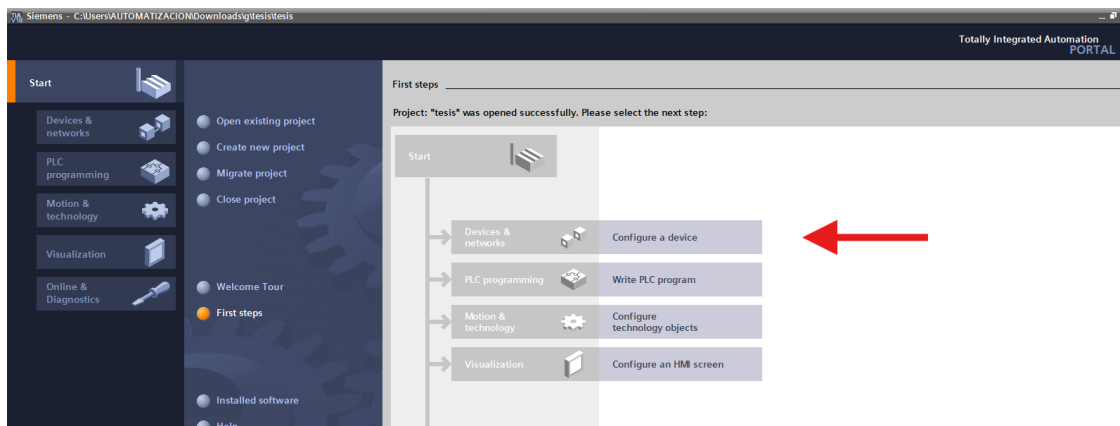
Nota. Se muestra la opción para crear y almacenar el proyecto realizado.

6.1.4 Creación del hardware del PLC S71200 en el árbol del proyecto del TIA PORTAL V18.

Una vez creado el proyecto, procedemos a realizar la configuración del dispositivo a utilizar, damos clic en “Configure a device”, luego “Add new device”, elegimos la CPU de nuestra preferencia, en nuestro caso es CPU 1214C DC/DC/DC, 6ES7 214-1AG40-0XB0 V4.4, después clic en “Add” para añadir uno por uno los dispositivos que se necesitan.

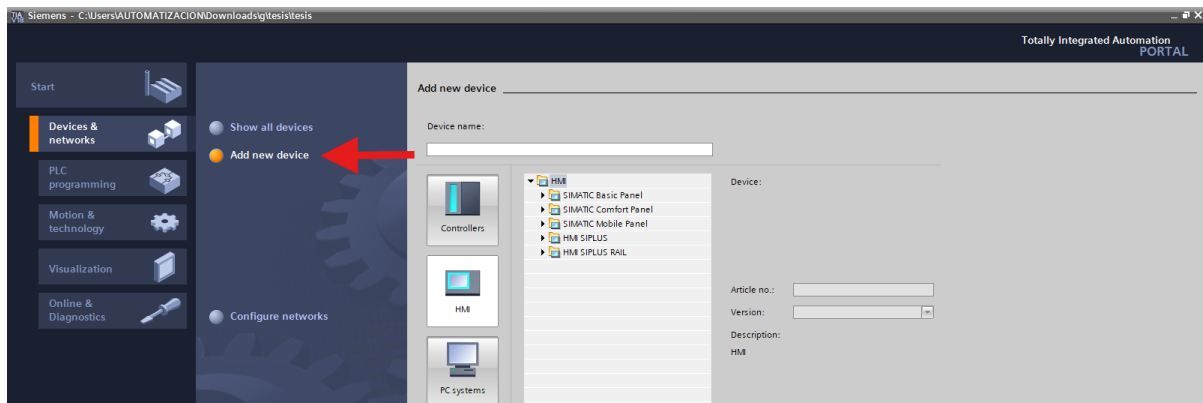
En nuestro caso se añaden 2 PLCs más con las mismas características anteriormente mencionadas, aparte se añade un variador es un G120 CU240E-2 PN, 6SL3244-0BB12-1FA0 V4.6 y un HMI KTP700 Basic PN, 6AV2 123-1GB03-0AX0 V15.1.0.0. como se visualiza en la figura 20 y figura 21.

Figura 21. Configuración de dispositivos en TIA PORTAL V18.



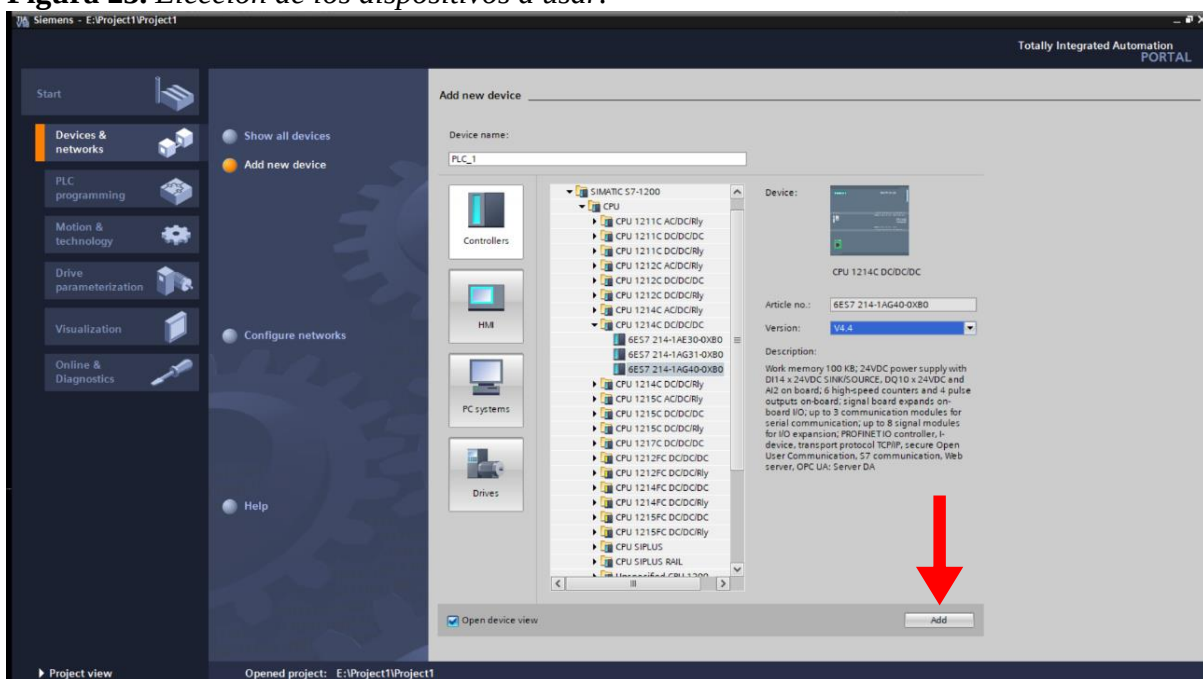
Nota. Se visualiza con la flecha de color roja, la opción para configurar el dispositivo con el cual vamos a trabajar.

Figura 22. Seleccionamos el dispositivo de trabajo.



Nota. En la imagen, se nos abre la opción para insertar los dispositivos requeridos.

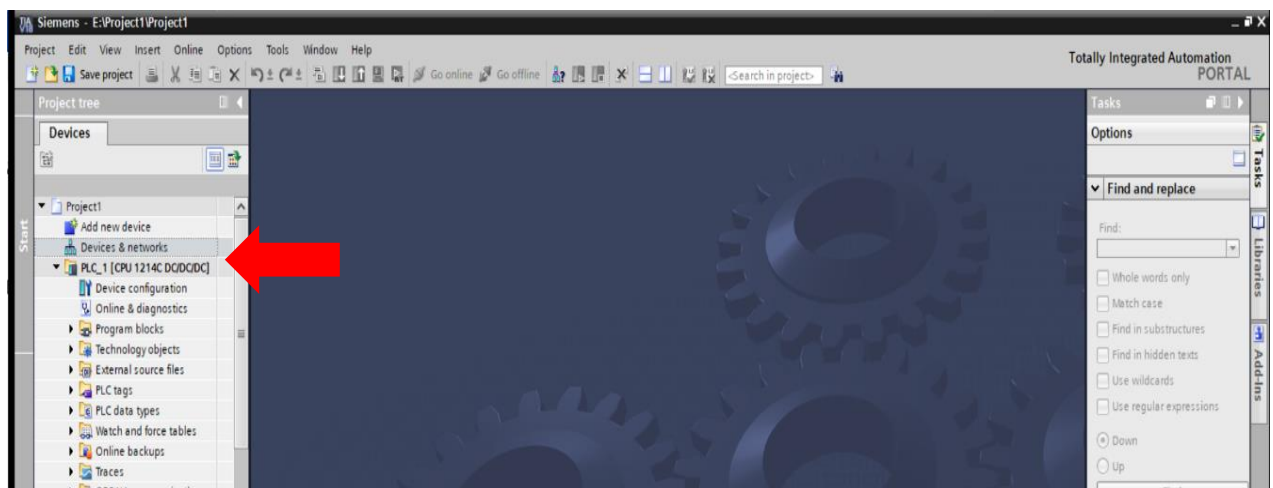
Figura 23. Elección de los dispositivos a usar.



Nota. En la figura, se detallan las características de los dispositivos con los cuales se trabajará.

Lo siguiente a la selección de los dispositivos que se usaran en nuestro trabajo se desplegara la opción de interfaz de programación y se verá el dispositivo creado con asunto “PLC_1, PLC_2, PLC_3”, se les cambian los nombres por “Master, Slave_1, Slave_2” respectivamente, visible en el árbol de proyectos a la izquierda de la ventana.

Figura 24. Visualizar dispositivos en árbol de proyectos.



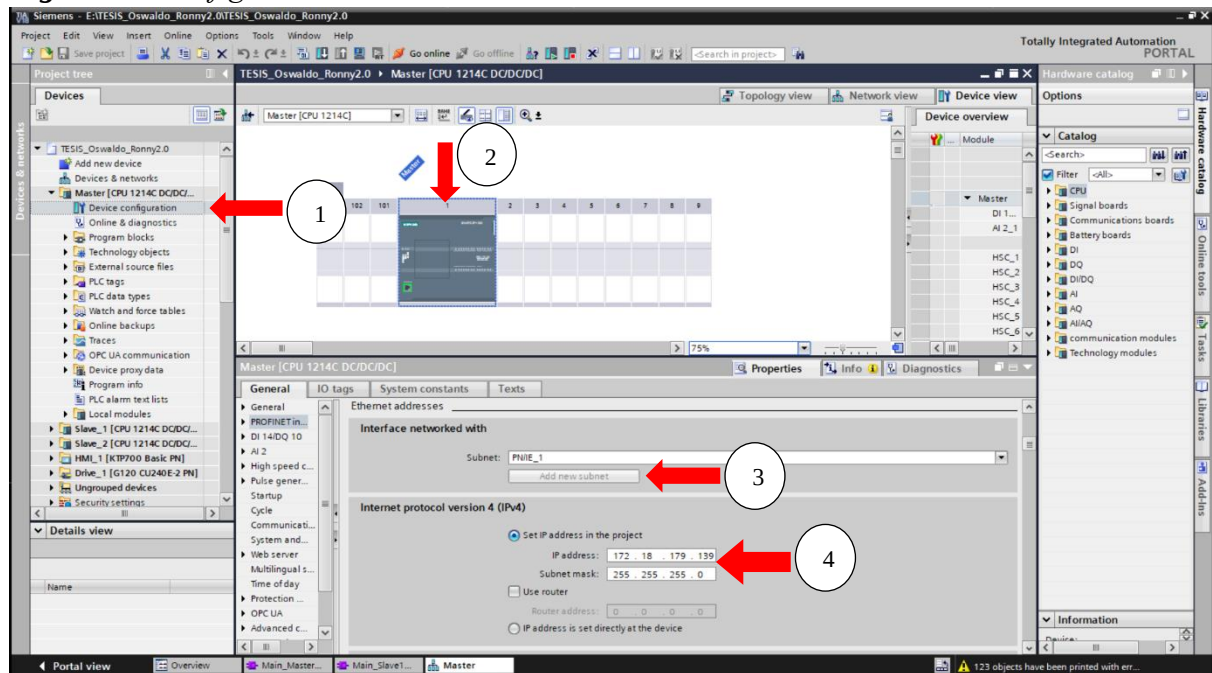
Nota. En la imagen, se visualiza el dispositivo indicado para trabajar, ubicado en la columna del lado izquierdo en el árbol de proyectos.

6.1.5 Configuración de red del PLC.

Para crear la configuración de red de nuestro PLC Maestro, damos clic en “Device configuration” como se muestra en la figura 24, seleccionamos en la CPU “Master” y otra vez seleccionamos el botón “Add new subnet”, así creamos la subnet PN/IE_1, luego definimos la IP del PLC los cuales serían 172.18.192.139 y subnet mask 255.255.255.0.

Los cambios anterior mente dichos también se realizan en el Slave_1 y Slave_2, en este caso la IP de cada PLC serían: 172.18.179.143, 172.18.179.145, la IP del HMI es 172.18.179.140, la IP del variador G120 es 172.18.179.11 y en el subnet mask será 255.255.255.0. cómo se visualiza en la figura 24.

Figura 25. Configuración del PLC “Master”.



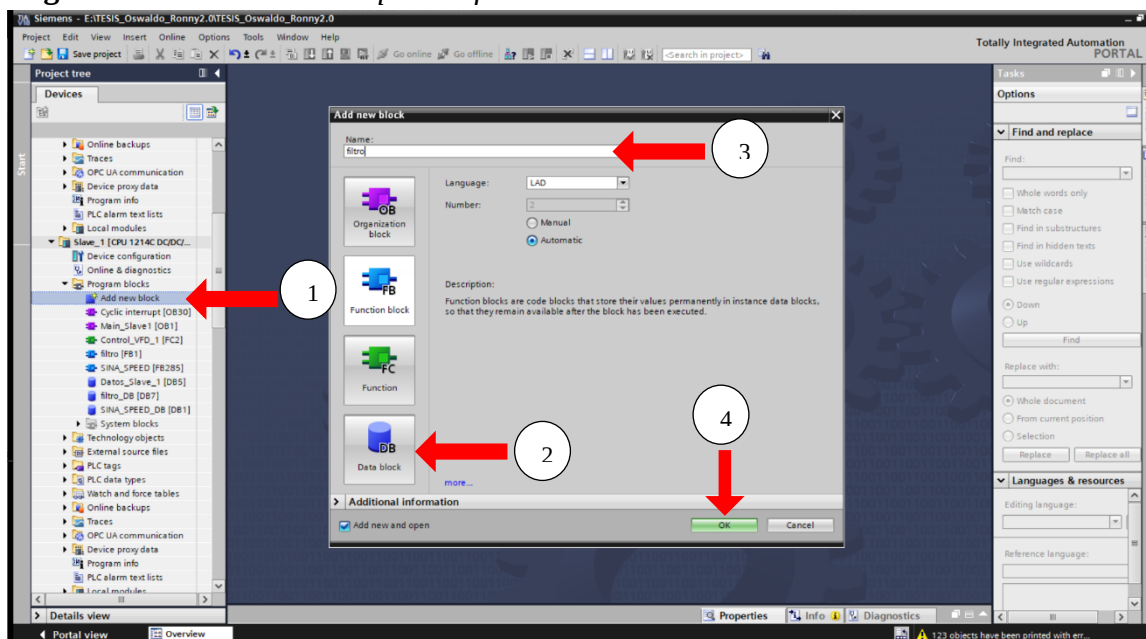
Nota. En la figura, se muestra el dispositivo seleccionado previamente para trabajar, en el cual se ven los cambios a realizar.

6.1.6 Programación de los PLCs en Tia PORTAL V18.

En este paso comenzamos a crear los bloques necesarios para la programación en este caso usaremos bloques de funciones y bloques de datos, tenemos que escoger la opción “ADD new block” ítem 1, luego la de “Function block” ítem 2, añadimos un nombre al bloque, el cual sería “Master” se usará un bloque de datos que se llamara: “Datos_HMI”.

En el caso del “Slave_1” en el cual se añaden bloques de datos de nombre: “Datos_Slave_1”, un bloque de función llamado: “filtro” y una función llamada: “Control_VFD_1”, el “Slave_2” tenemos un bloque de datos con los nombres: “Datos_Slave_2”, un bloque de función llamado: “Filtro” y una función llamada: “Control_VFD_2”.

Figura 26. Creación de bloques de funciones.



Nota. Se muestran las opciones para la creación de bloques de funciones en el programa TIA PORTAL V18.

6.1.6.1 Parametrización del bloque de datos en PLC “Master”

Datos_HMI[DB2] y RECETAS[DB1].

El bloque de datos “Datos_HMI” luego de haber creado se desliza automáticamente en la interfaz del TIA PORTAL V18 y podemos ver la interfaz del bloque donde se define cada variable que guardará la información que se recibirá de los PLCs esclavos, el bloque de datos de “RECETAS” tendrá las recetas que se crearan para el control de los motores con diferentes casos velocidades dependiendo de lo que se requiera.

Figura 27. Parametrización del DB “Datos_HMI”.

	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static									
2	Setpoint_VFD_1	Int	...	0		☒	☒	☒	☐	
3	Setpoint_VFD_2	Int	...	0		☒	☒	☒	☐	
4	RPM_VFD_1	Real	...	0.0		☒	☒	☒	☐	
5	RPM_VFD_2	Real	...	0.0		☒	☒	☒	☐	
6	Enable_VFD_1	Bool	...	false		☒	☒	☒	☐	
7	ON_VFD_2	Bool	...	false		☒	☒	☒	☐	
8	OFF_VFD_2	Bool	...	false		☒	☒	☒	☐	
9	HABILITAR_VFD_2	Bool	...	false		☒	☒	☒	☐	
10	Encendido	Bool	...	false		☒	☒	☒	☐	
11	Luz_online_M1	Bool	...	false		☒	☒	☒	☐	
12	Luz_online_M2	Bool	...	false		☒	☒	☒	☐	

Nota. Se describe la interfaz del bloque de datos “Datos_HMI” con las variables a usar en la lógica del proyecto.

6.1.6.2 Parametrización del bloque de datos en PLC “Slave_1”

Datos_Slave_1[DB5].

El bloque de datos “Datos_Slave_1” una vez creado se despliega automáticamente en la interfaz del TIA PORTAL V18 y podemos ver la interfaz del bloque donde se define cada variable que guardará la información que se enviará y la información que se recibirá del “Master”.

Figura 28. Parametrización del DB “Datos_Slave_1”.

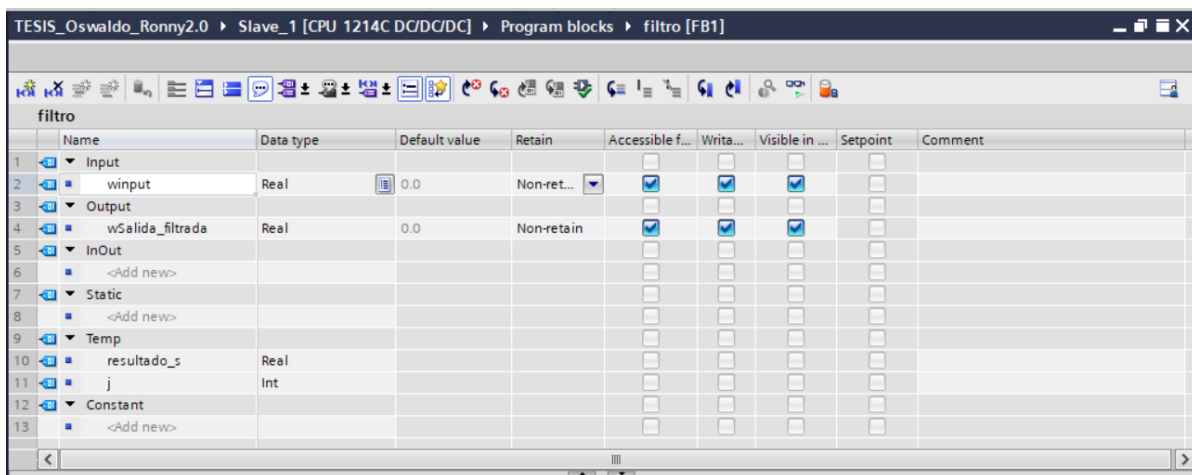
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static									
2	RPM_Encoder_VFD_1	Real	...	0.0		☒	☒	☒	☐	
3	Setpoint_RPM_VFD_1	Int	...	580		☒	☒	☒	☐	
4	Enable_VFD1	Bool	...	0		☒	☒	☒	☐	
5	Encendido	Bool	...	false		☒	☒	☒	☐	
6	Luz_online	Bool	...	false		☒	☒	☒	☐	
7	Registro	Array[0..4] of Real	...			☒	☒	☒	☐	
8	Registro[0]	Real	...	0.0		☒	☒	☒	☐	
9	Registro[1]	Real	...	0.0		☒	☒	☒	☐	
10	Registro[2]	Real	...	0.0		☒	☒	☒	☐	
11	Registro[3]	Real	...	0.0		☒	☒	☒	☐	
12	Registro[4]	Real	...	0.0		☒	☒	☒	☐	

Nota. Se describe la interfaz del bloque de datos “Datos_Slave_1” con las variables a usar en la lógica del proyecto.

6.1.6.3 Parametrización del bloque de función en PLC “Slave_1” filtro[FB1].

El bloque de función “filtro” Cuando se crea automáticamente, se abre automáticamente la interfaz Portal V18 de la tía, y podemos observar una descripción de la interfaz de bloque de programación en la parte superior de la pantalla de programación, donde se definirán las variables utilizadas en nuestra lógica, continuaremos con la programación del bloque de funciones utilizando variables definidas en la interfaz (revisar anexo 5).

Figura 29. Parametrización del FB “Filtro”.

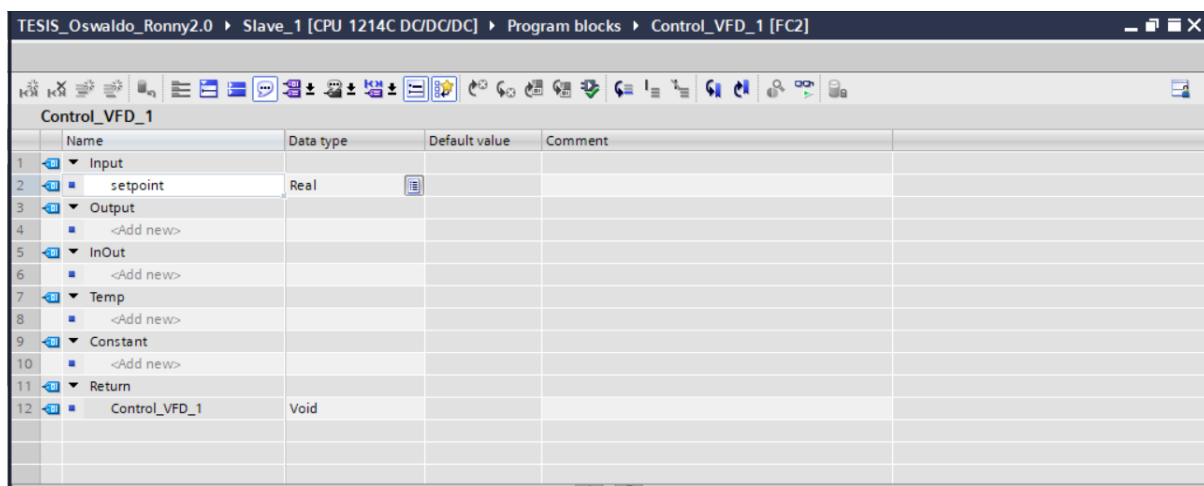


Nota. En la imagen, se visualiza la interfaz del bloque de opción “Filtro” con las variables a usar en la lógica del proyecto.

6.1.6.4 Parametrización de función en PLC “Slave_1” Control_VFD_1[FC2].

La función “Control_VFD_1” Cuando se crea se abre automáticamente en la interfaz Tía portal V18, y podemos observar una descripción de la interfaz de bloque de programación en la parte superior de la pantalla de programación, donde se definirán las variables utilizadas en nuestra lógica, continuaremos con la programación del bloque funcional utilizando las variables definidas en la interfaz (revisar anexo 4).

Figura 30. Parametrización del FC “Control_VFD_1”.

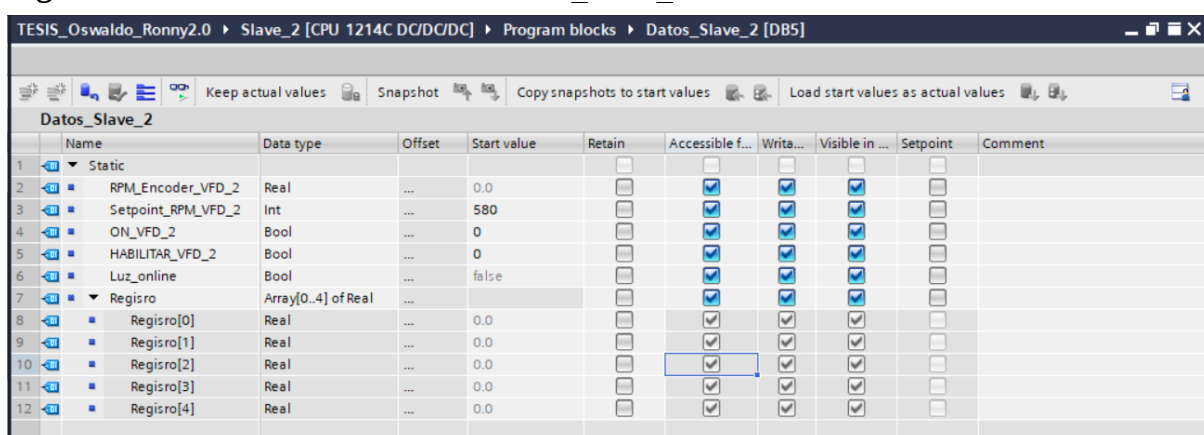


Nota. En la imagen, se visualiza la interfaz de función “Control_VFD_1” con las variables a usar en la lógica del proyecto.

6.1.6.5 Parametrización del bloque de datos en PLC “Slave_2” Datos_Slave_2[DB5].

El bloque de datos “Datos_Slave_2” cuando ya hemos creado se abre automáticamente en la interfaz del TIA PORTAL V18 y podemos ver la interfaz del bloque donde se define cada variable que guardará la información que el “Master” leerá y enviará al “Slave_2”.

Figura 31. Parametrización del DB “Datos_Slave_2”.

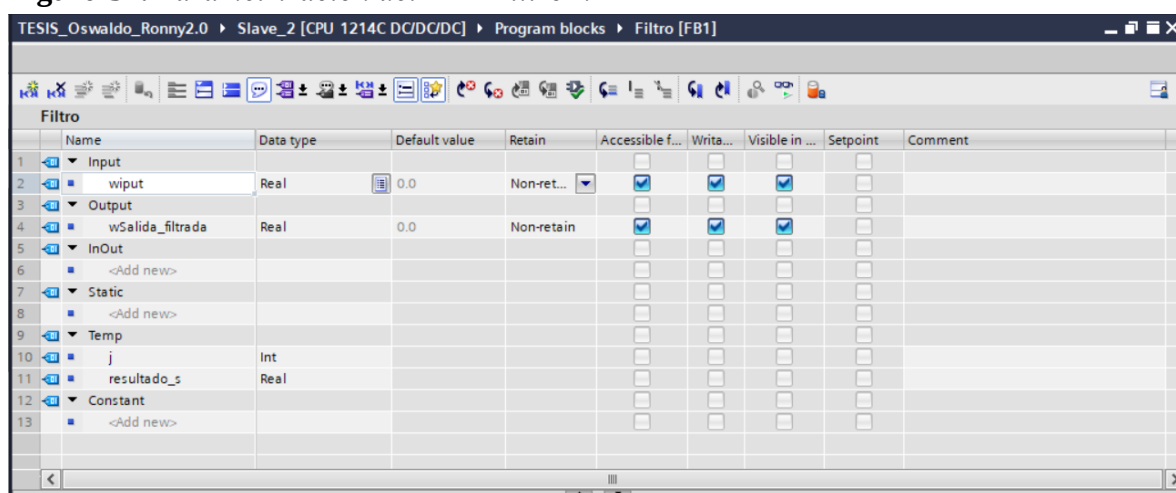


Nota. Se describe la interfaz del bloque de datos “Datos_Slave_2” con las variables a usar en la lógica del proyecto.

6.1.6.6 Parametrización del bloque de función en PLC “Slave_2” Filtro [FB1].

El bloque de función “Filtro” Cuando se crea se abre automáticamente en la interfaz Tía portal V18, y podemos observar una descripción de la interfaz de bloque de programación en la parte superior de la pantalla de programación, donde se definirán las variables utilizadas en nuestra lógica, continuaremos con la programación del bloque funcional utilizando las variables definidas en la interfaz (revisar anexo 9).

Figura 32. Parametrización del FB “Filtro”.

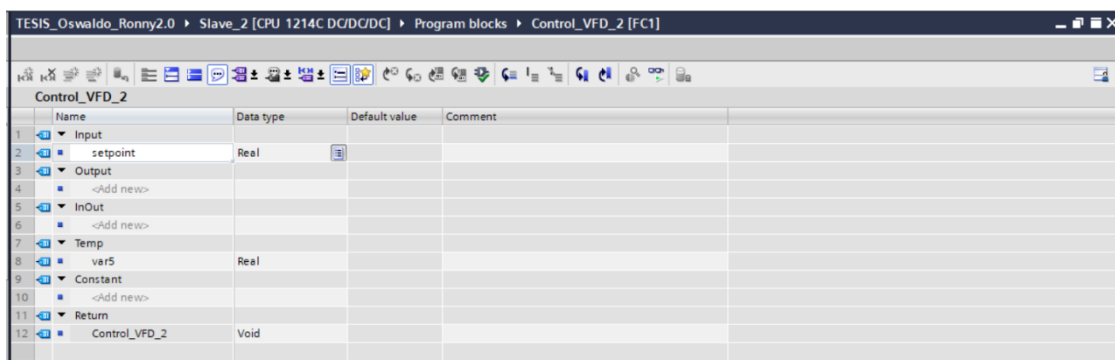


Nota. En la figura, se expresa la interfaz del bloque de función “Filtro” con las variables a utilizar en la lógica del proyecto.

6.1.6.7 Parametrización de función en PLC “Slave_2” Control_VFD_2[FC1].

La función “Control_VFD_2” Cuando se crea se abre automáticamente en la interfaz Tía portal V18, y podemos observar una descripción de la interfaz de bloque de programación en la parte superior de la pantalla de programación, donde se definirán las variables utilizadas en nuestra lógica, continuaremos con la programación del bloque funcional utilizando las variables definidas en la interfaz (revisar anexo 8).

Figura 33. Parametrización del FC “Control_VFD_2”.

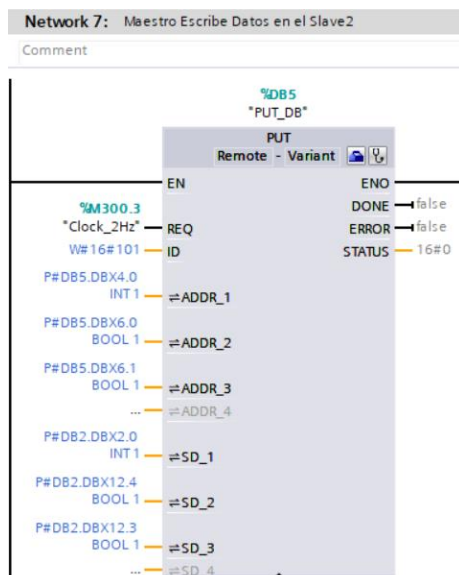


Nota. En la imagen, se visualiza la interfaz de función “Control_VFD_2” con las variables a usar en la lógica del proyecto.

6.1.6.8 Parametrización de los bloques PUT y GET.

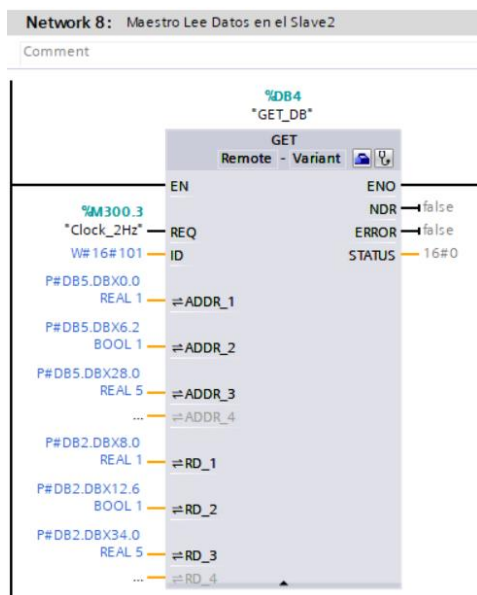
La comunicación entre los PLCs se lleva a cabo gracias al bloque PUT que se usa para escribir los datos del bloque de datos del PLC emisor al bloque de datos del PLC receptor y el bloque GET que permite la lectura de los datos en el PLC que necesitamos en este caso el “Master” y se escriben en el PLC que lo lee.

Figura 34. Bloque PUT en “Master”.



Nota. En la imagen, se puede visualizar la configuración del bloque PUT en “Master” para la comunicación con “Slave_2”.

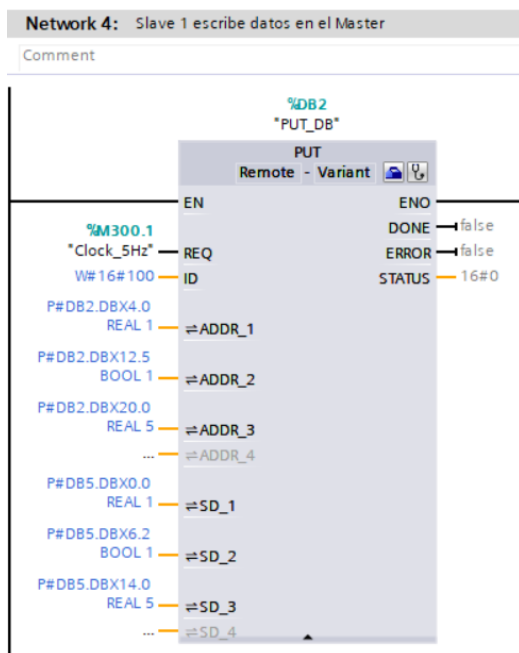
Figura 35. Bloque PUT en “Master”.



Nota. En la imagen, se visualiza la configuración del bloque GET en “Master” para la comunicación con “Slave_2”.

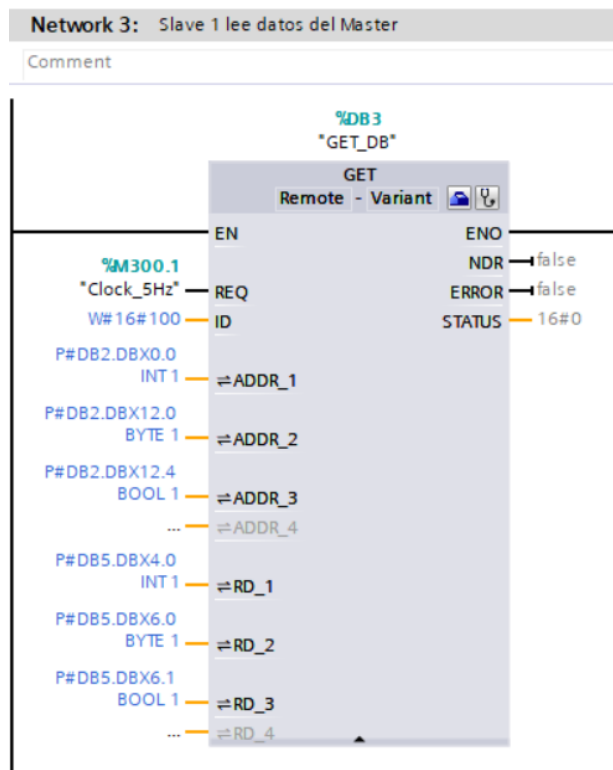
Se puede ver que en los dos casos los bloques usados necesitan saber la dirección a la cual van a enviar la información o de donde leer es información y en qué lugar debería ir cada una de esas variables anteriormente guardadas, así cada vez que existe un pulso en el bloque comienza su trabajo.

Figura 36. Bloque PUT en “Slave_1”.



Nota. En la figura, se puede observar la configuración del bloque PUT en “Slave_1” para la comunicación con “Master”.

Figura 37. Bloque PUT en “Slave_1”.



Nota. En la imagen, observaremos las configuraciones del bloque GET en “Slave_1” para la comunicación con “Master”.

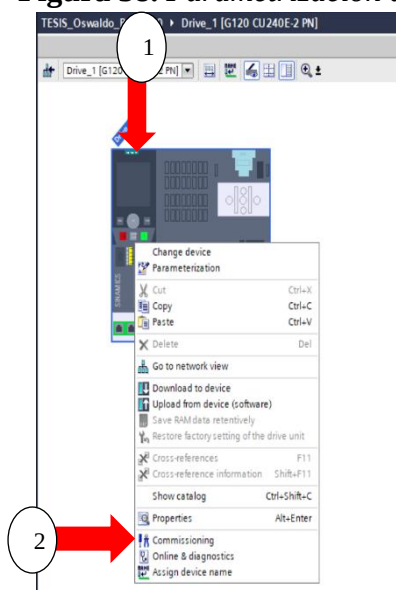
6.2 Diseñar un sistema de control PID y PI en TIA PORTAL para el control de velocidad de los motores trifásicos usando variadores de frecuencia.

El diseño de un sistema de control PID y PI para sincronizar la velocidad de motores trifásicos utilizando el bloque PID_Compact, el cual nos permite el poder crear un PID o PI, este bloque da la opción de detectar el funcionamiento del sistema y calcular los parámetros necesarios para un buen control PID o PI a su vez también nos deja ingresar los valores manualmente.

6.2.1 Parametrización del Variador G120.

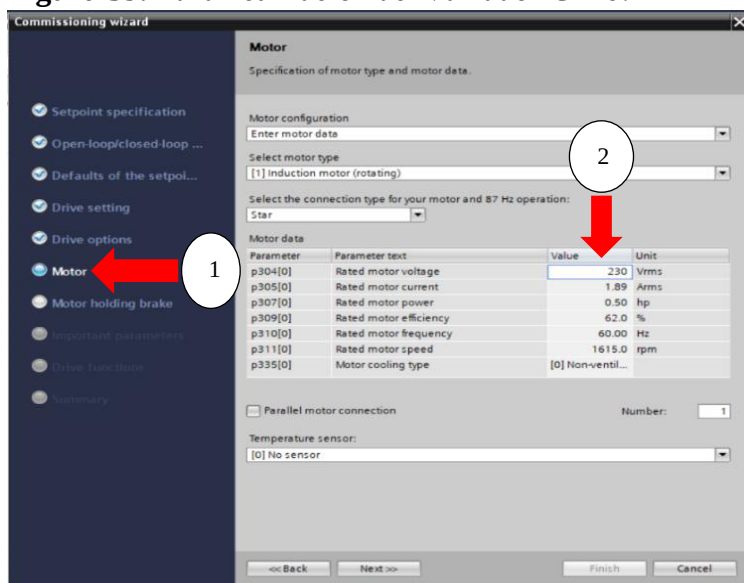
Al momento de agregar el variador G120 debemos elegir el variador con las características del que tenemos físicamente, luego se debe colocar la fuente conectada al variador, al tener todo listo damos clic derecho sobre el variador y vamos a commissioning donde configuraremos el variador con la información del motor que controlara.

Figura 38. Parametrización del Variador G120.



Nota. En la figura, se puede observar los pasos para llegar a configuración del variador.

Figura 39. Parametrización del Variador G120.

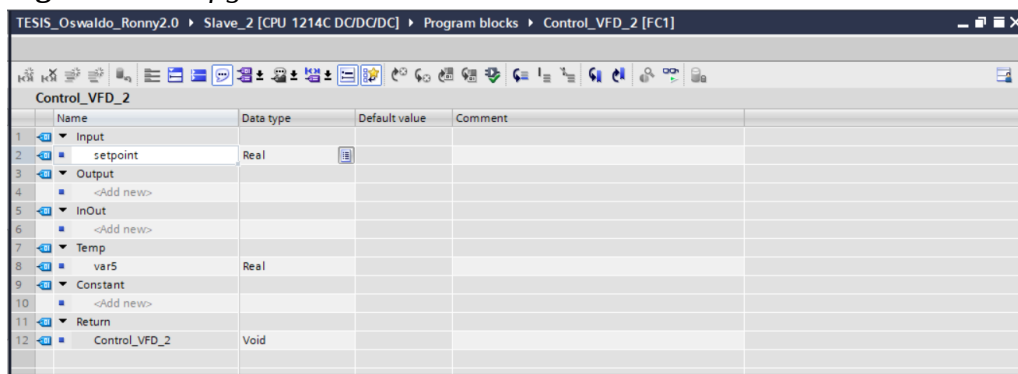


Nota. En la figura, se puede observar los pasos para llegar a configuración del variador.

6.2.2 Parametrización de Entradas para Variador V20 en “Slave_2”.

En el caso del Variador V20 se usarán salidas analógicas para su control, la cual estará normalizada y escalada para este variador, lo que se realiza por medio de los bloques de “NORM_X” y “SCALE_X”, se usarán salidas analógicas y digitales para su control. (revisar anexo 8).

Figura 40. Configuración del Variador V20.



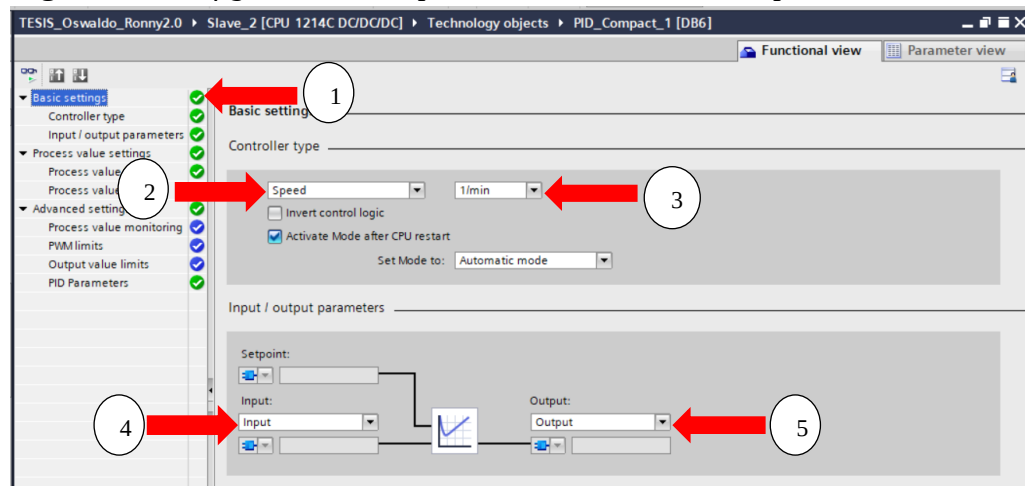
	Name	Data type	Default value	Comment
1	Input			
2	setpoint	Real		
3	Output			
4	<Add new>			
5	InOut			
6	<Add new>			
7	Temp			
8	var5	Real		
9	Constant			
10	<Add new>			
11	Return			
12	Control_VFD_2	Void		

Nota. En la figura, observamos los parámetros los cuales cambian en el variador.

6.2.3 Parametrización del bloque PID_Compact que se encuentra en “Slave_1” y “Slave_2”.

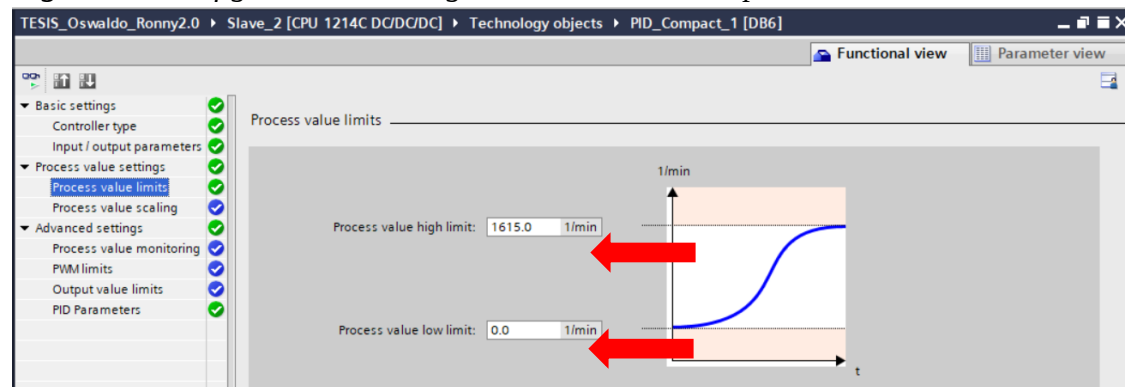
En este paso primero se debe añadir el bloque cyclic interrupt donde colocaremos el bloque PID_Compact, al momento de colocar el bloque debemos configurarlo para eso se debe dar doble clic a la caja de herramientas azul al lado derecho superior del bloque, allí cambiamos parámetros que usaremos en este caso controlaremos la velocidad y cambiamos los límites de 0 a 1625 RPM, esto se realiza en los dos bloques usados en “Slave_1” y “Slave_2”.

Figura 41. Configuración del tipo de control del PID_Compact.



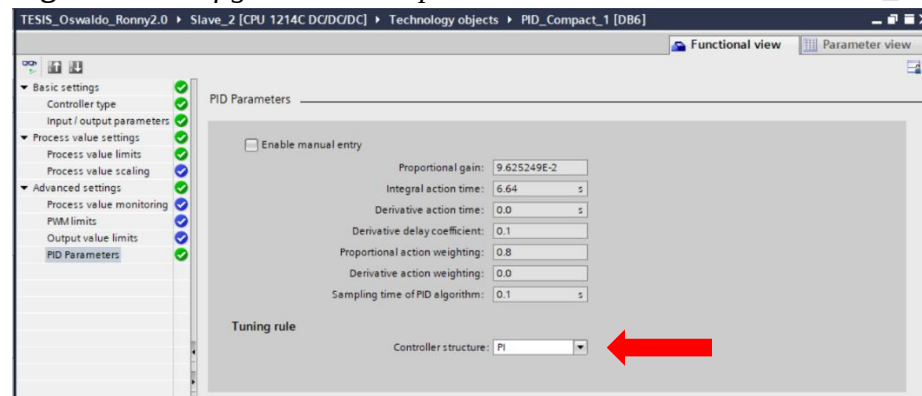
Nota. En la figura, se puede observar los parámetros que se cambian en el variador.

Figura 42. Configuración del rango límite del PID_Compact.



Nota. En la figura, se puede observar los límites del variador.

Figura 43. Configuración del tipo de controlador PID o PI del PID_Compact.

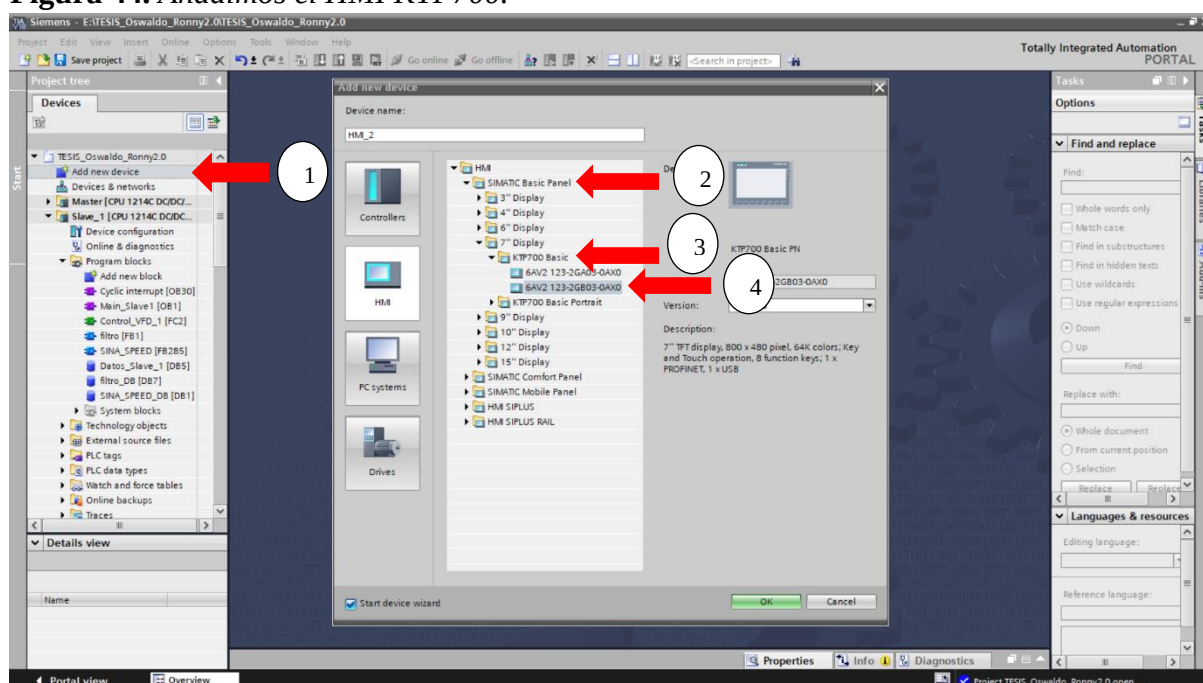


Nota. En la figura, se puede observar donde cambiar el tipo de controlador PID o PI.

6.3 Elaborar un HMI centralizado para el monitoreo y configuración de la sincronización de velocidad de los motores trifásicos.

Dentro del programa vamos a comenzar primero con el proyecto, al tener el proyecto finalizado podemos empezar con la creación del HMI para lo cual primero damos doble clic en “Add new device”, en nuestro caso el HMI que se usara es el KTP700 Basic PN, 6AV2 123-2GB03-0AX0, V15.1.0.0, para este caso la IP que se usara es 172.18.179.140 con subnet mask de 255.255.255.0.

Figura 44. Añadimos el HMI KTP700.

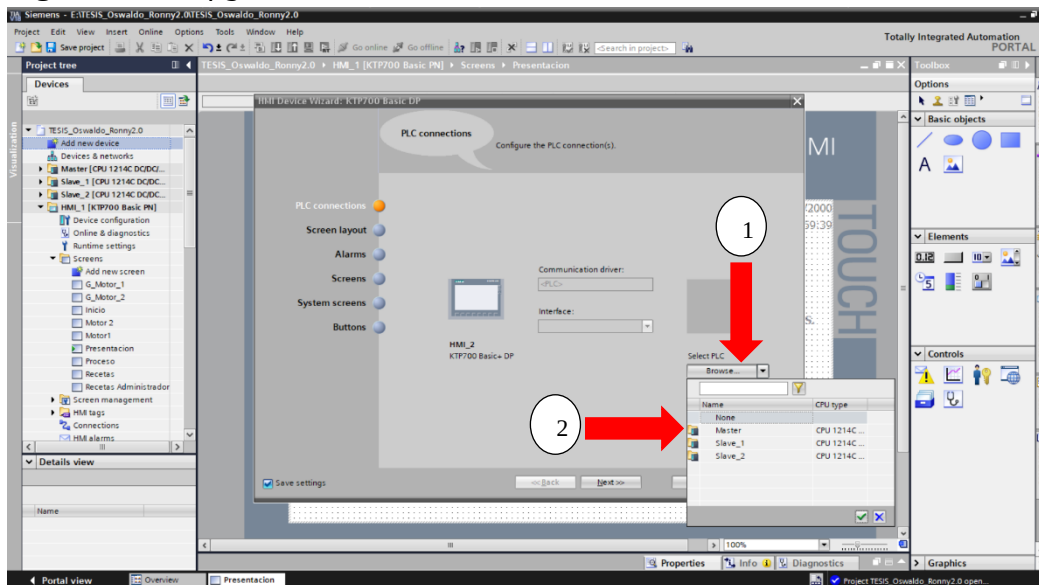


Nota. En la figura, se observa como añadir el HMI respectivo.

6.3.1 Configuración de la comunicación del KTP700 con el PLC S7-1200.

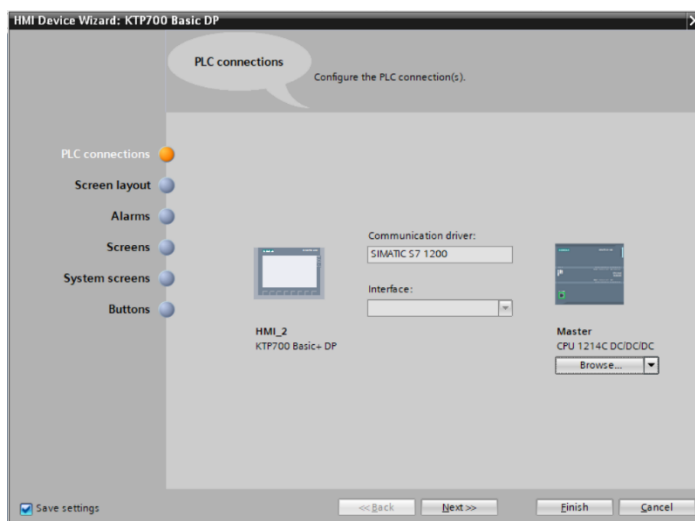
Luego de agregar el dispositivo, visualizaremos una ventana llamada HMI Device Wizard; KTP700 Basic PN, en esta buscaremos el PLC “Master” y daremos clic en aceptar, luego visualizaremos la ventana de confirmación y seleccionamos finalizar.

Figura 45. Configuración de la comunicación del HMI KTP700.



Nota. En la imagen, se observa como añadir el HMI respecto.

Figura 46. Configuración de la comunicación del HMI.



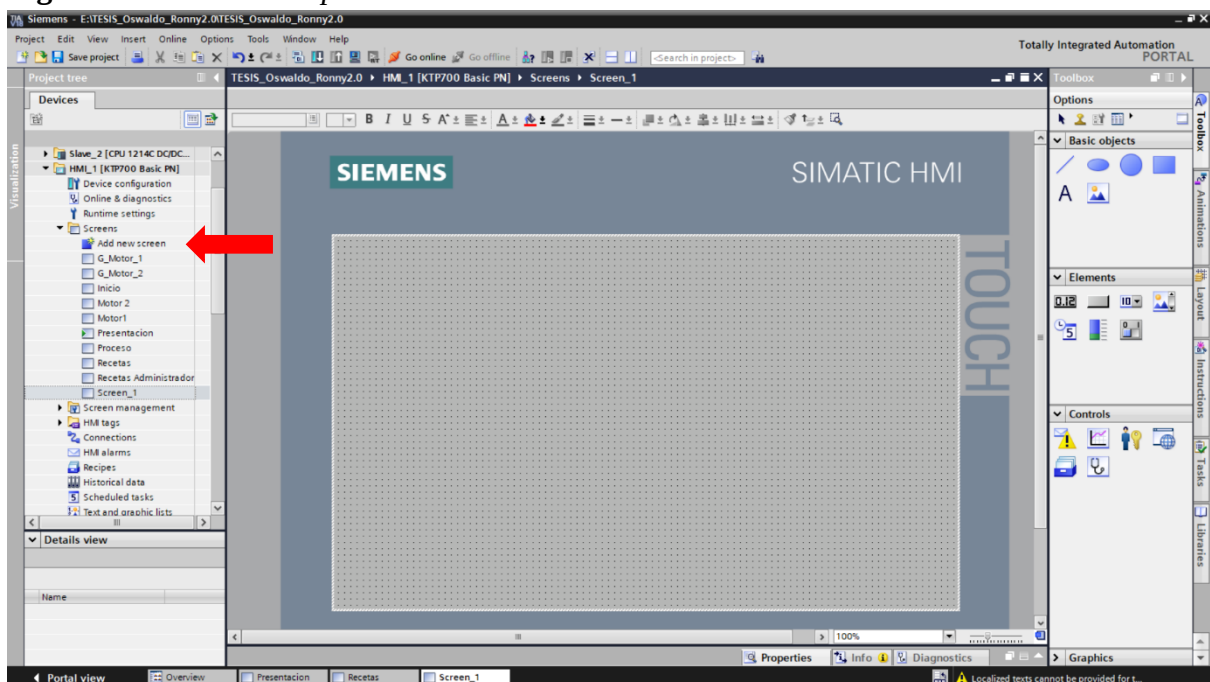
Nota. En la imagen, se observa la comunicación del HMI con el PLC “Master”.

6.3.2 Creación de las pantallas.

Al terminar la configuración del HMI podemos ingresar al mismo para crear la cantidad de pantallas que sean necesarias, primero desplegamos la carpeta Screens para luego dar clic en “Add new screen”, apenas le damos sobre clic se puede ver la nueva pantalla (revisar imagen 46).

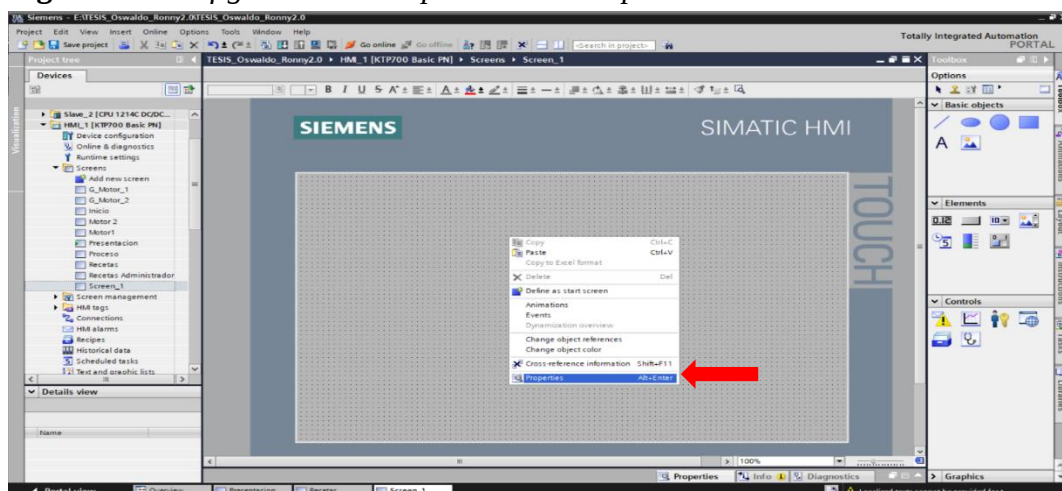
Posteriormente a crear la pantalla es necesario configurar las plantillas de las diferentes pantallas que se han creado, daremos clic derecho en el centro de la imagen de la pantalla y nos iremos a propiedades (revisar figura 47), luego damos clic en el cuadro de texto definido como “Template” lo desplegamos, para identificar “Screen template” y seleccionamos “Template_1” (revisar figura 48).

Figura 47. Creación de pantallas.



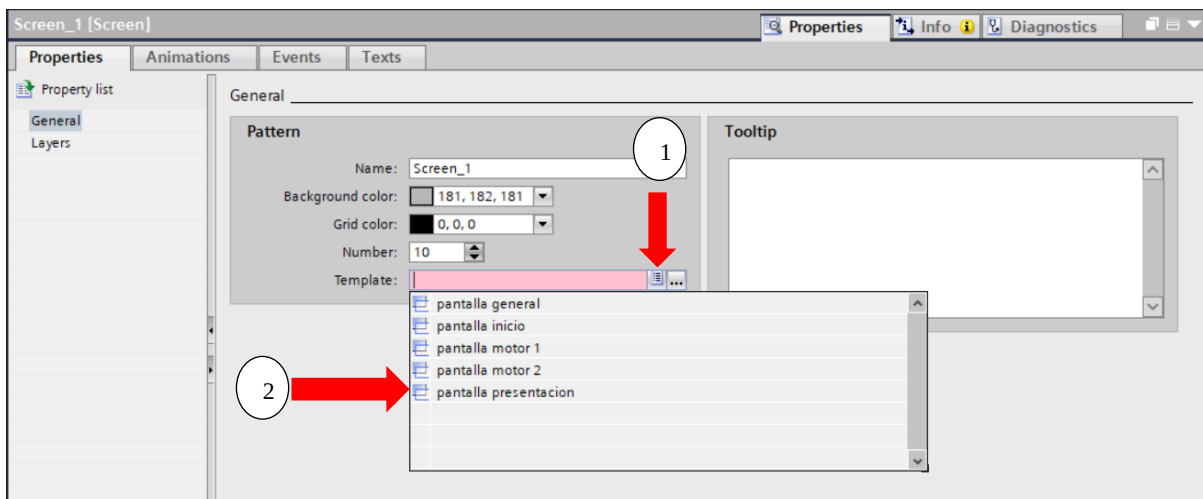
Nota. En la imagen, se ve dónde está la opción para crear pantallas con una flecha roja. .

Figura 48. Configuración de la plantilla de las pantallas.



Nota. En la imagen, se ve dónde está la opción para configurar la plantilla de las pantallas.

Figura 49. Configuración de la plantilla de las pantallas.

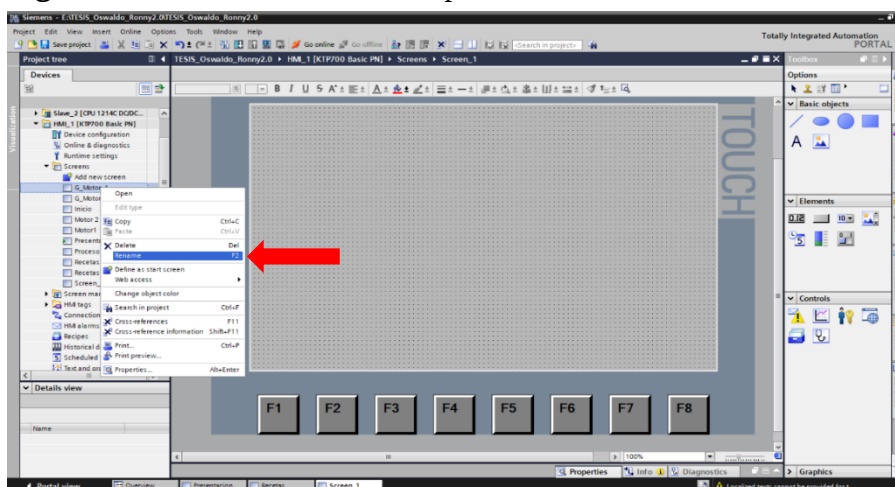


Nota. En la imagen, se ve dónde está la opción para configurar la plantilla de las pantallas.

6.3.3 Diseño de pantallas.

Al tener ya configurado y colocado el template en las pantallas correspondientes, es momento de editar las pantallas de la aplicación, procedemos a reescribirle el nombre a las pantallas respectivamente para lo que la necesitemos en este caso damos clic derecho en la pantalla que se desea renombrar y seleccionamos “Rename”. En este caso tendremos pantallas de visualización general del proceso con animación, botones, campos de entrada/salida y animación.

Figura 50. Cambio de nombre de pantalla.

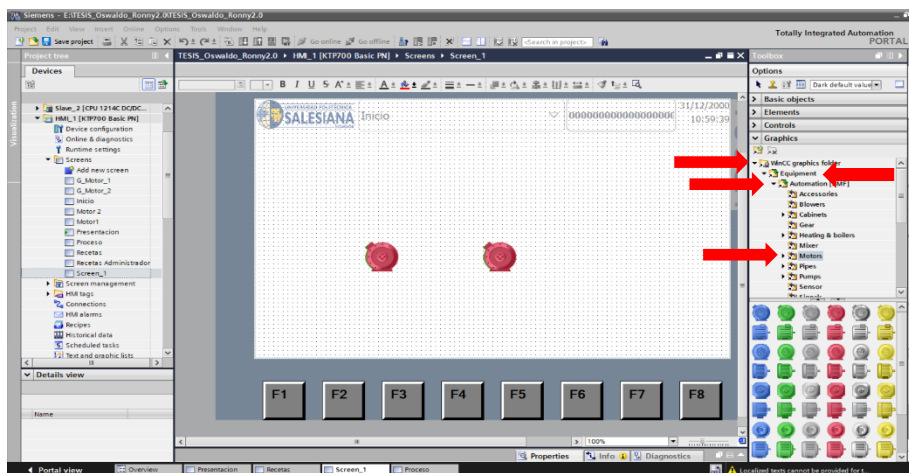


Nota. La imagen se observa donde se puede cambiar el nombre de la pantalla.

6.3.4 Insertar gráficos.

Para el ingreso de graficas se necesita hacer uso de la parte llamada “toolbox” en la cual se encuentra la sección de graficas, al dar clic allí se desplegará varias carpetas, en nuestro caso vamos a usar motores por lo cual entramos a la carpeta “WinCC graphics folder”, luego “Equipment”, “Automation [EMF]”, por último, dar clic sobre “Motors” y seleccionamos el motor que más nos guste (revisar figura 50).

Figura 51. Creación de un gráfico.

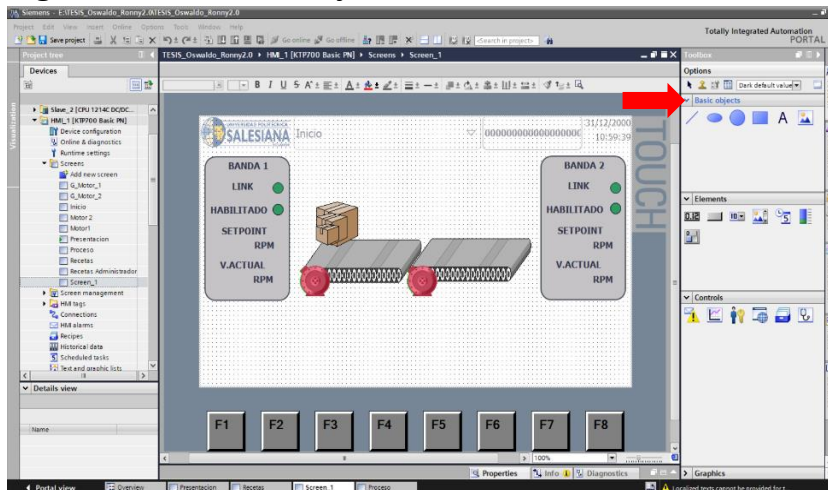


Nota. En la imagen, representaremos el uso de un gráfico.

6.3.5 Insertar objetos básicos.

Para poder insertar objetos básicos, debemos dar clic en la parte llamada “Basic objects” que está incluida en el programa en la misma sección anterior mente mencionada llamada “toolbox”, el objetivo para agregar estos objetos es resaltar y mostrar los datos que nosotros requerimos en el trabajo, para esto damos clic en los objetos dentro de la Liberia y arrastramos el objeto a la pantalla (revisar figura 51).

Figura 52. Colocar un objeto.



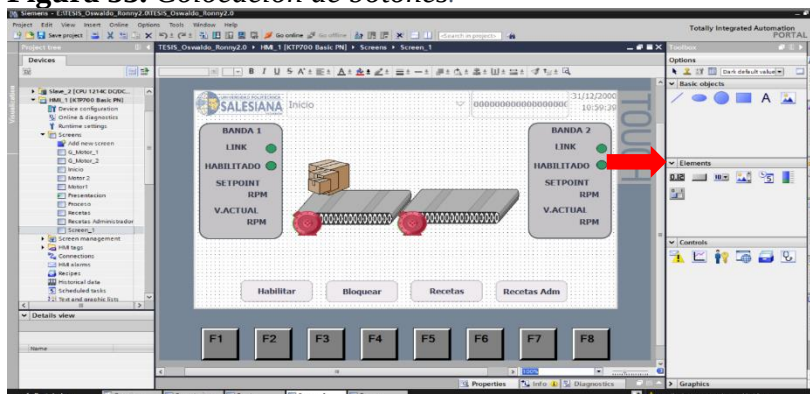
Nota. En la figura, se representa la selección y el uso de un objeto.

6.3.6 Insertar botones.

Si se desea controlar variables digitales en el proyecto se deben ingresar botones, para esto se debe ir a la parte de “toolbox” en la sección llamada “Elements” que se incluye en el programa, damos clic en los botones dentro de la librería y arrastramos los elementos que se va a usar en la pantalla (revisar figura 52)

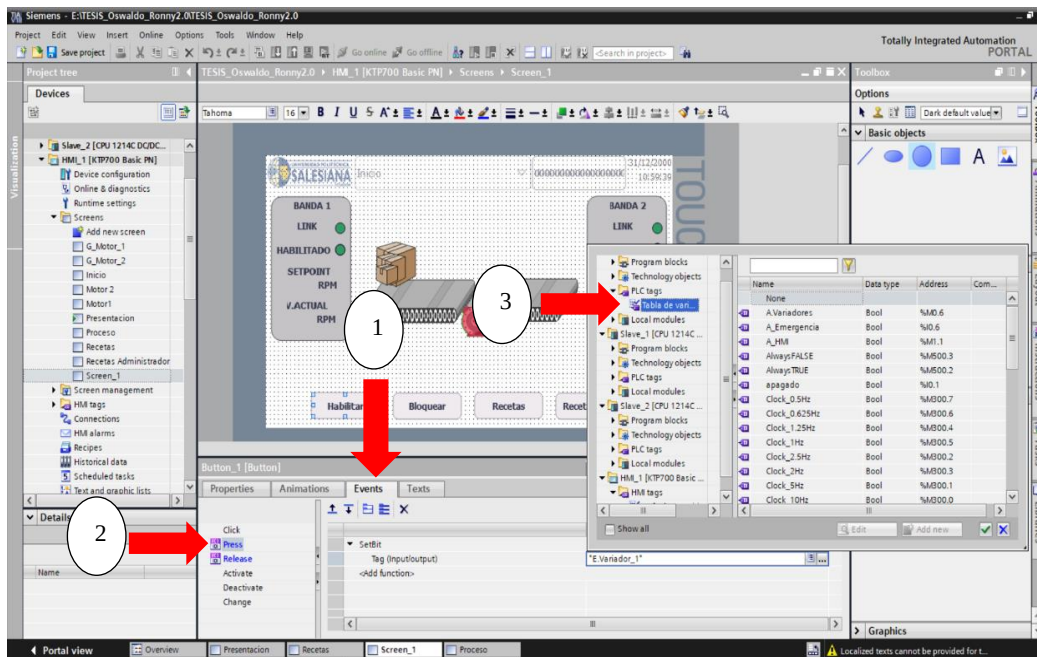
Para el correcto uso de los botones definimos la variable que controlara el botón, damos clic derecho sobre el botón, damos clic en propiedades, luego vamos a “Events” de la fila de propiedades y daremos clic para seleccionar el tipo de evento a realizar, seleccionamos “Press” y “Release”, definimos el mismo tag a utilizar en ambos eventos (revisar figura 53).

Figura 53. Colocación de botones.



Nota. En la imagen, observaremos la selección de botones.

Figura 54. Colocación de botones.

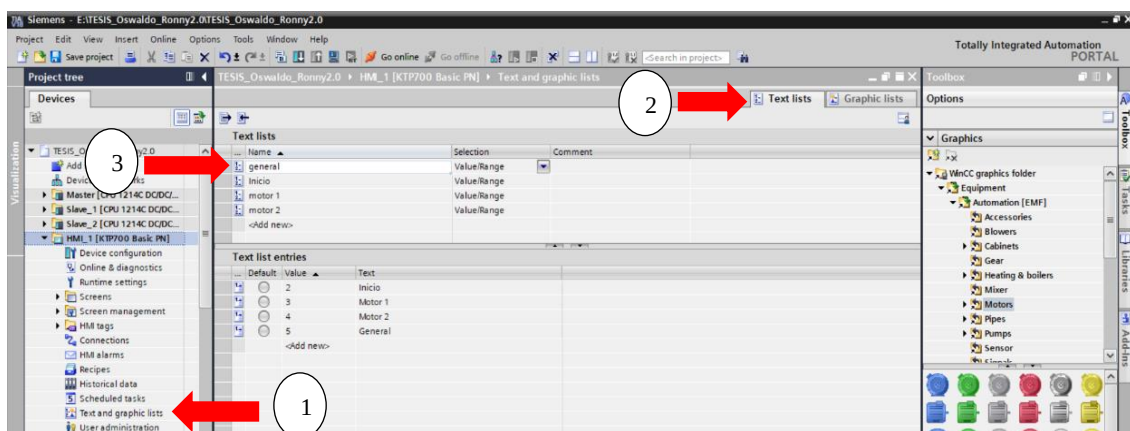


Nota. En la imagen, se observa la configuración del botón y la selección del tag del botón.

6.3.7 Insertar lista de texto.

Las listas de texto se pueden dinamizar con etiquetas o listas de texto, para insertar una lista de texto y controlar variables digitales, se realizará la configuración dando clic en “Text and graphic list” en el listado principal del proyecto, seleccionamos “Text list” y creamos las listas que se quiere en este caso llevará el nombre de “Inicio” (revisar figura 54).

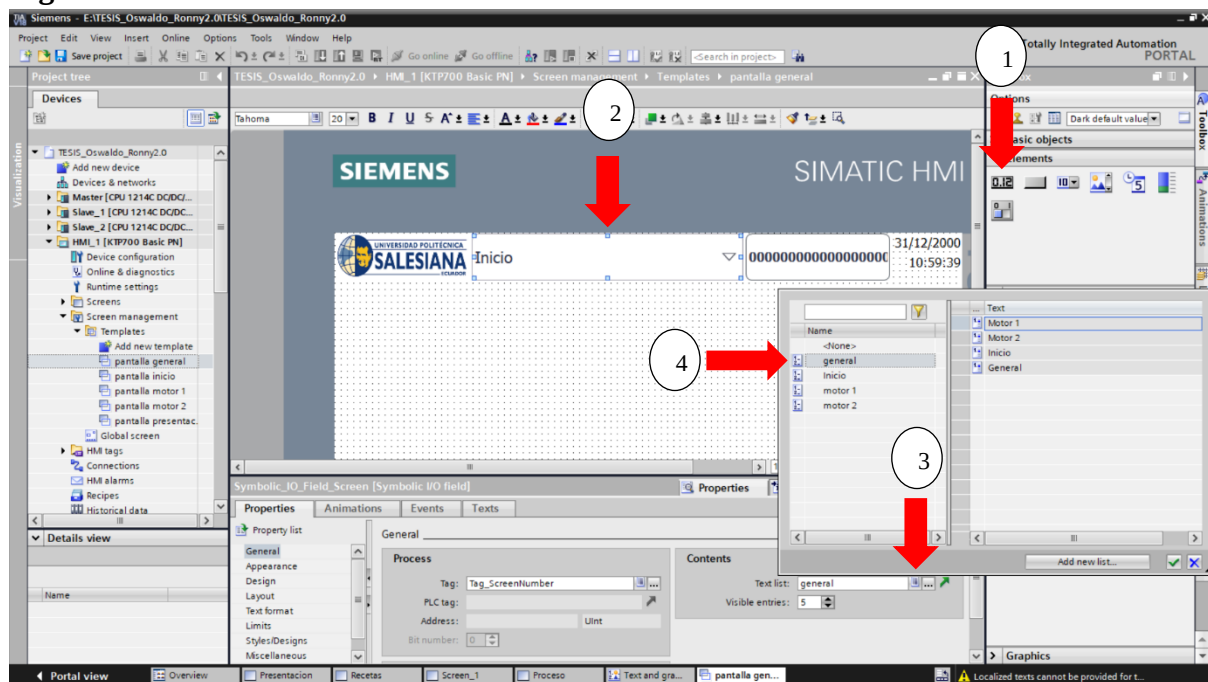
Figura 55. Creación de lista de texto.



Nota. En la imagen, visualizaremos los pasos para la creación de una lista de texto en el programa TIA PORTAL V18.

En el template de la pantalla de proceso escogeremos la opción “Symbolic I/O field” que la veremos en el “toolbox”, lo seleccionamos hasta la ubicación que queramos, abrimos las propiedades del campo recién creado y le asignamos la lista de texto en el apartado “Contents”, después elegimos la lista que se desea en esa pantalla o pantallas donde se usara el template que se está configurando (revisar figura 55).

Figura 56. Parametrización de lista de texto.



Nota. La imagen muestra la configuración de un “Symbolic i/o field” utilizando lista de texto.

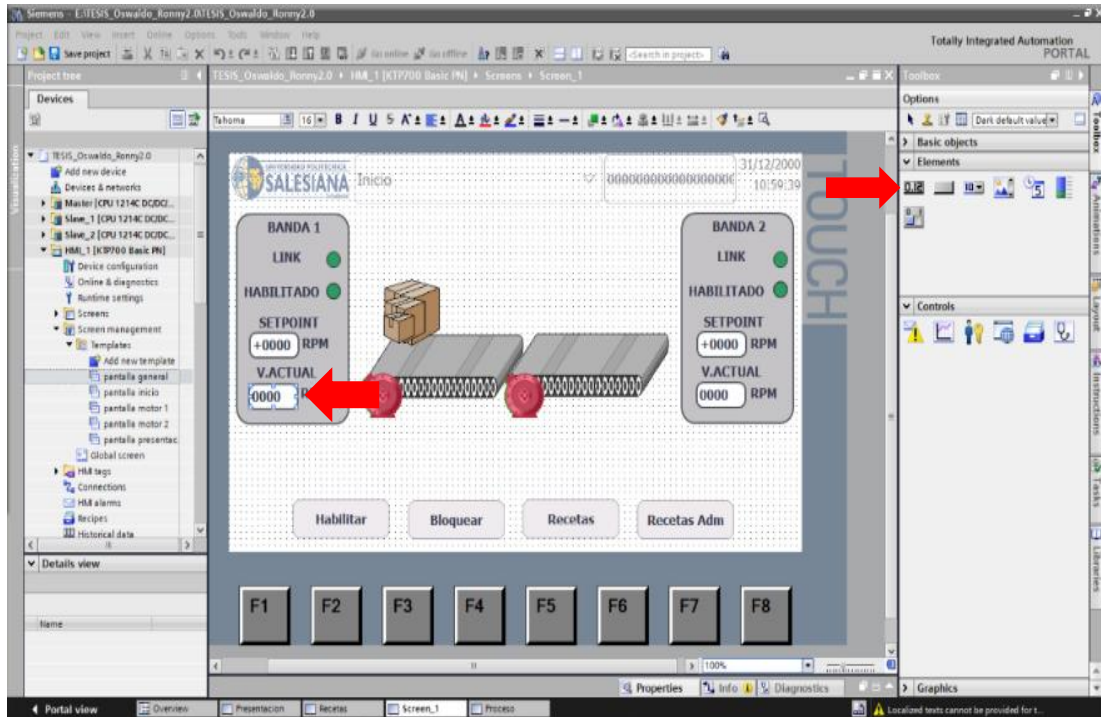
6.3.8 Insertar campos de entrada/salida.

Para acceder a un mundo de entrada/salida que establezca un control de los procesos con algún valor de referencia, en nuestro proyecto vamos a utilizar la librería “Elements”, incluida dentro del “toolbox”, daremos clic en la opción entrada/salida denominada “I/O field”, escogemos hasta la pantalla en edición (revisar figura 56).

Para definir la variable en la entrada/salida damos clic derecho sobre él y entramos a propiedades, damos clic en el apartado general de la lista, seleccionamos el tag de la variable

que queremos en nuestro caso es una variable del bloque de datos “Datos_HMI” donde están asignada las variables del proyecto (revisar figura 57).

Figura 57. Colocación de los campos de entrada/salida.

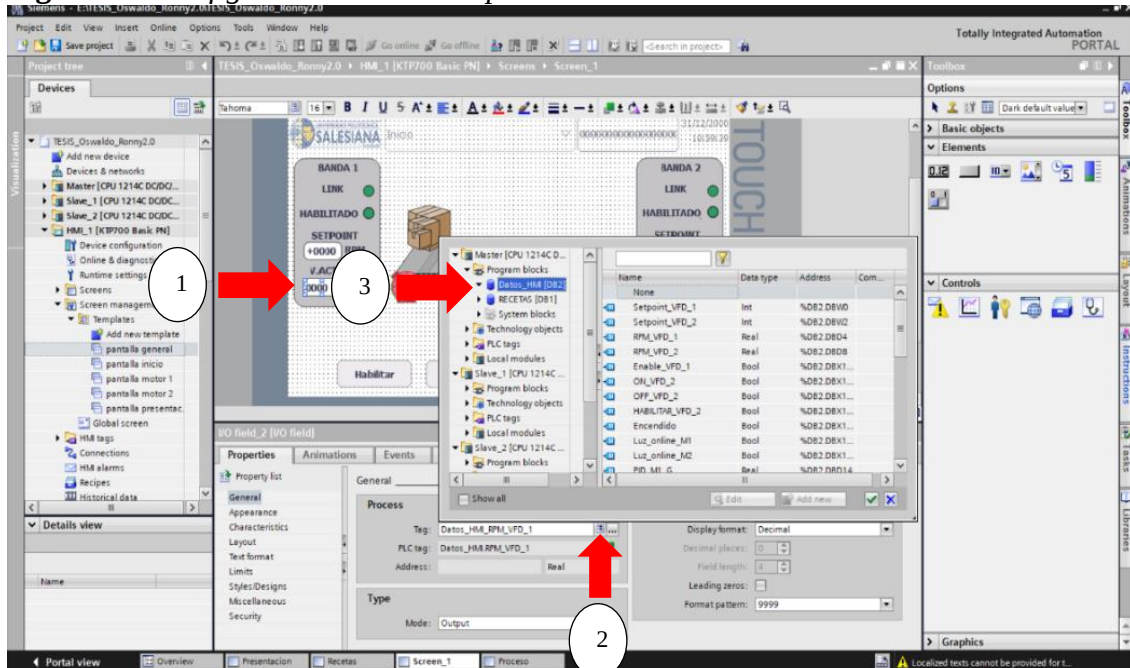


Nota.

La
figura

muestra la ubicación de las entradas/salidas usadas en la pantalla proceso.

Figura 58. Configuración de los campos de entrada/salida.

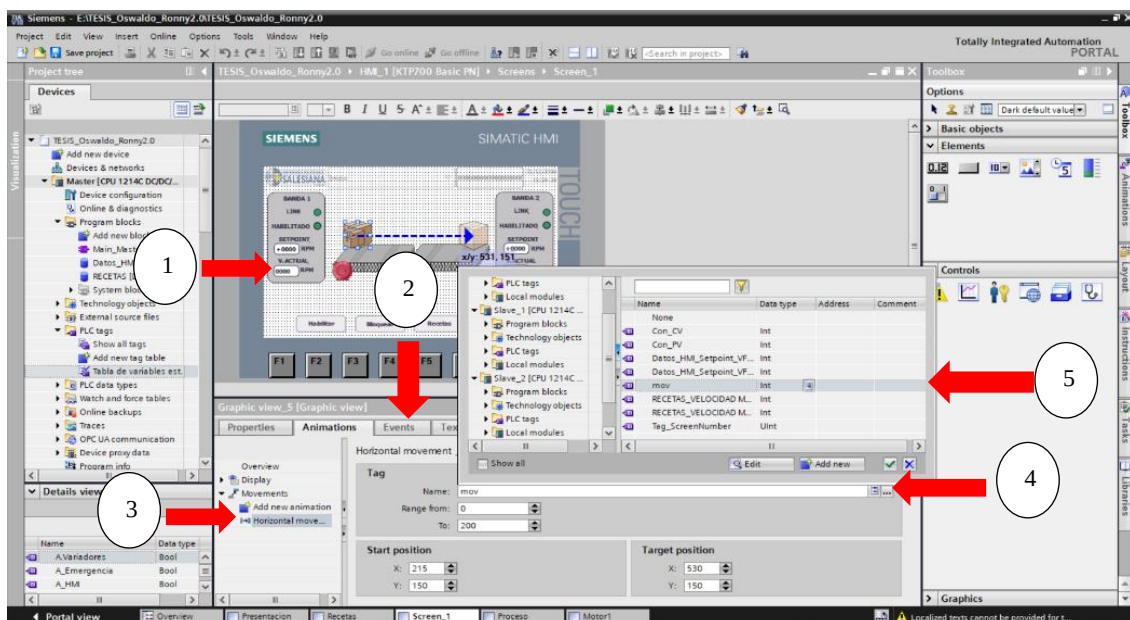


Nota. En la imagen, se muestra la configuración del campo entrada/salida.

6.3.9 Creación de animaciones.

Para animar un objeto, se usarán las gráficas anterior mente colocados, vamos a propiedades y nos dirigimos a la parte que se llama “Animations”, luego nos dirigimos a la sección “Movements”, damos clic en “Add new animation”, en este caso se usará un simple movimiento en X, al momento de crear la animación cambiamos hasta donde se moverá el objeto, después colocamos el tag que hará mover la caja en este caso es un contador incremental (revisar figura 59).

Figura 59. Creación de animación para gráficos.



Nota. En la imagen se detalla la creación de las animaciones en los gráficos.

6.3.10 Creación de recetas.

Para la creación de recetas primero tenemos que crear un bloque de datos donde guardar las recetas creadas, este bloque se llamara “RECETAS” dentro del bloque se colocara una variable Struct en este caso llamada “VELOCIDAD MOTORES” y dos String llamada “Nombre” y “Registro”, en la variable Struct se colocara los ingredientes que se modificaran en este caso son “Motor1” y “Motor2”.

luego de crear el bloque donde guardar la información vamos a la sección de “Recipes” en el HMI, al ingresar podemos crear diferentes parámetros primero definimos dos tipos de recetas y luego en la parte de “Elements” colocamos los ingredientes que se usaran en este caso las bandas, vamos a tag y encontramos en el bloque “Recetas” la variable respectiva (revisar figura 60).

Figura 60. Configuración del bloque de datos “RECETAS”.

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static			☐	☐	☐	☐	☐	
2	VELOCIDAD MOTORES	Struct		☐	☒	☒	☒	☐	
3	Motor 1	Int	0	☐	☒	☒	☒	☐	
4	Motor 2	Int	0	☐	☒	☒	☒	☐	
5	Nombre	String	"	☐	☒	☒	☒	☐	
6	Registro	String	"	☐	☒	☒	☒	☐	

Nota. Se describe la interfaz del bloque de datos “RECETAS” con las variables a utilizar en la lógica del proyecto.

Al finalizar la configuración en recipe podemos ir al HMI y ya podemos hacer uso del “recipe view” en el área de “toolbox”, en la sección de “Controls”, el cual se va a arrastrar a la pantalla y colocarlo donde nos sea más útil así poder crear y usar las recetas que vamos a definir en el (revisar figura 61).

Figura 61. Creación de recetas.

Name	Display name	Number	Version	Type	Maximum number of d..	Communication type	Tooltip
Tamaño	Tamaño Cajas	1	12/23/2024 2:4...	Limited	100	Tags	
peso	Peso (Kg)	2	12/23/2024 2:4...	Limited	100	Tags	
<Add new>							

Name	Display name	Tag	Data type	Data length	Default value	Minimum value	Maximum value
Banda 1	Banda 1 (RPM)	RECETAS_VELOCID...	Int	2	0	-32768	32767
Banda 2	Banda 2 (RPM)	RECETAS_VELO...	Int	2	0	-32768	32767
<Add new>							

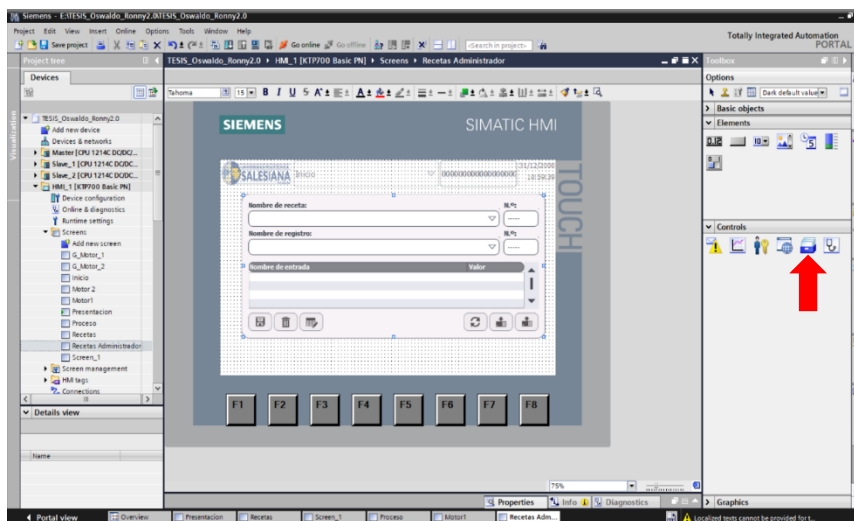
Nota. Se describe la interfaz de “Recipes”.

6.3.11 Creación de tendencias.

Para observar el comportamiento variable dependiendo del tiempo del proceso automatizado, es una herramienta de aplicación de gestión importante para evitar problemas previos a la evitación asociada con una función de control defectuoso, en este caso se puede usar “Trend view” que nos da una tabla donde se visualizara las variables que deseamos.

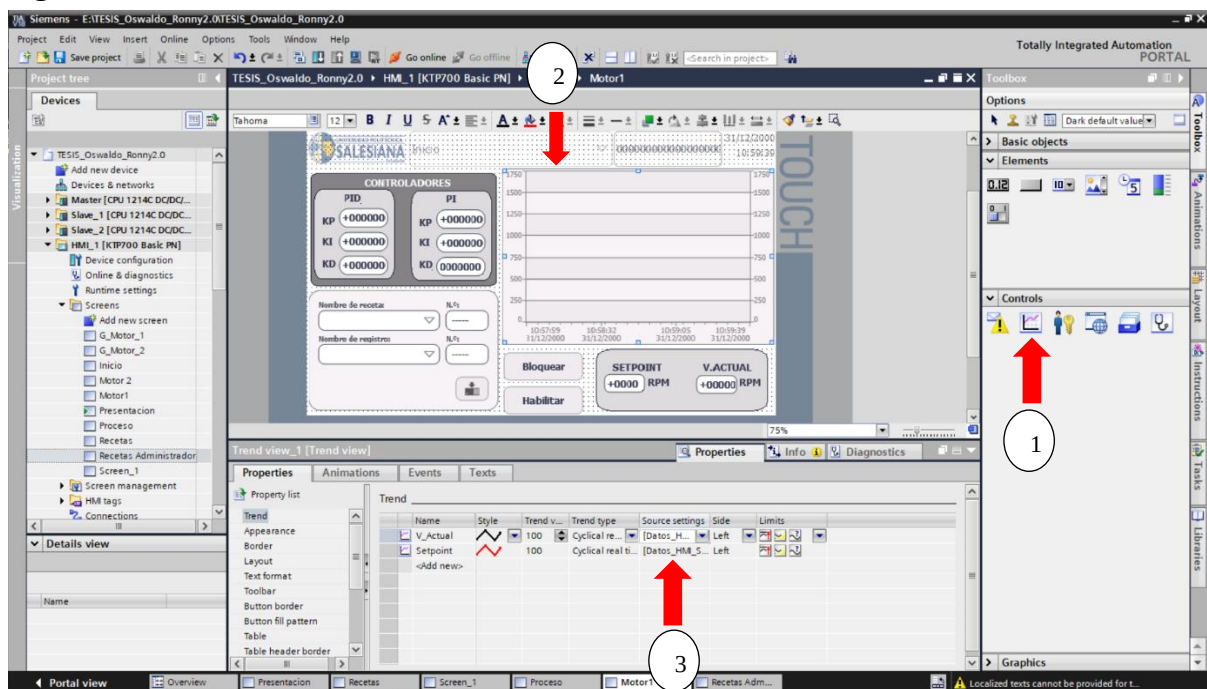
Para configurarlo se debe arrastrar el “Trend view” del área de control, luego ir a propiedades del mismo y añadir una variable para poder observarla en la gráfica, se nombra y se coloca el tag de la variable (revisar figura 62).

Figura 62. Colocación del “Recipe view”.



Nota. Se coloca el “Recipe view” de la sección Control a la pantalla.

Figura 63. Creación de tendencias.



Nota. En la imagen se observa la configuración de “Trend view”.

VII RESULTADOS

7.1 Ejecución del proyecto en TIA PORTAL V18 para la revisión de la comunicación entre los PLCs.

Lo primero que debemos hacer ubicar el proyecto que desarrollamos en TIA PORTAL V18 designado con el nombre de TESIS_Oswaldo_Ronny2.0, como se observa en el figura 63, luego damos doble clic sobre el icono del proyecto el cual nos abrirá TIA PORTAL V18 con el proyecto y cargado.

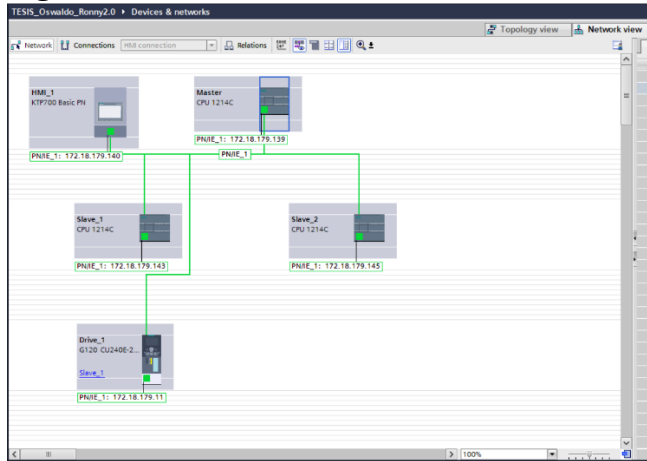
Al ingresar ya al proyecto lo que hacemos a continuación es dar clic sobre el PLCs que se desea cargar, nos dirigimos a la parte superior a “Download to device” le damos clic y se comenzara a buscar el PLC con la IP que configuramos, al encontrarlo se procede a subir la programación, proseguimos con los demás PLCs y HMI, como se puede visualizar en las siguientes figura 64.

Figura 64. Icono del Proyecto en TIA PORTAL V18.

Nombre	Fecha de modificación	Tipo	Tamaño
 AdditionalFiles	28/2/2025 16:19	Carpeta de archivos	
 IM	28/2/2025 16:19	Carpeta de archivos	
 Logs	28/2/2025 16:19	Carpeta de archivos	
 System	28/2/2025 16:19	Carpeta de archivos	
 TMP	28/2/2025 16:19	Carpeta de archivos	
 UserFiles	28/2/2025 16:19	Carpeta de archivos	
 Vci	28/2/2025 16:19	Carpeta de archivos	
 XRef	28/2/2025 16:19	Carpeta de archivos	
 TESIS_Oswaldo_Ronny2.0	5/3/2025 13:20	Siemens TIA Portal...	8 KB

Nota. En la imagen se puede observar el archivo del proyecto.

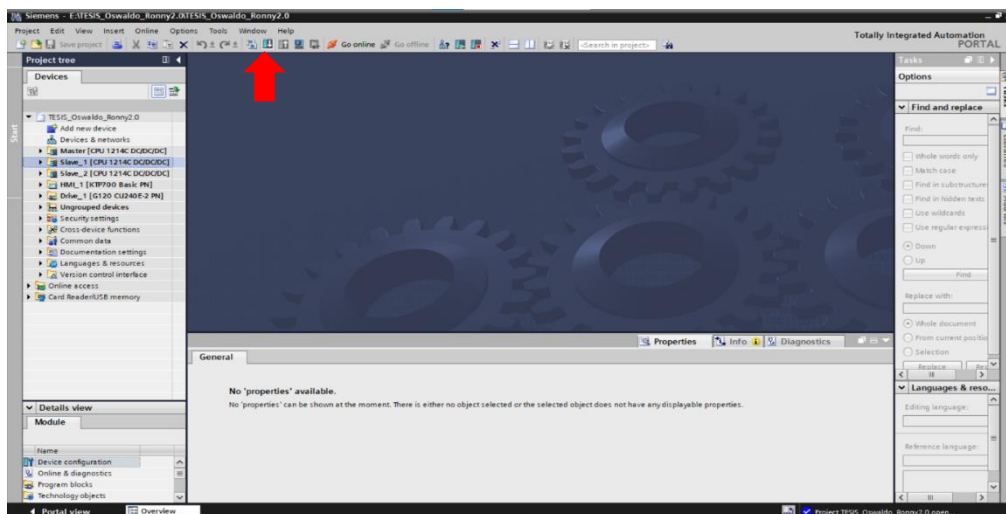
Figura 65. Device & Networks conexión de los PLCs.



Nota. Se observa la topología del proyecto y la conexión que se ha realizado.

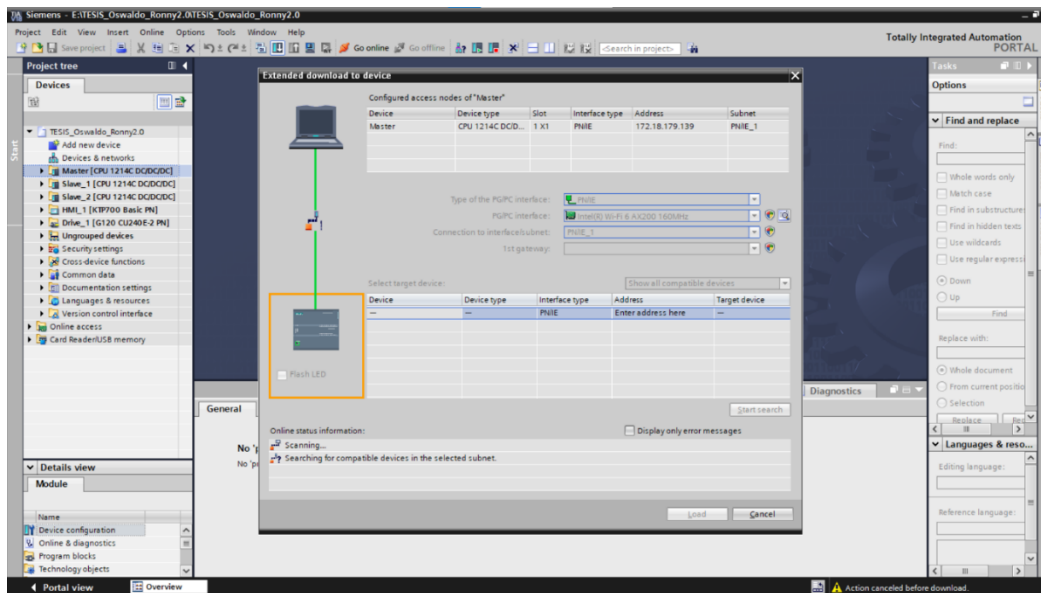
En la figura 65 tenemos a la topología que se implementó en el proyecto junto a los IP de todos los componentes conectados a la red lo cual nos da a entender que están conectados y listos para trabajar.

Figura 66. Ubicación de Download to device.



Nota. Se puede observar la pantalla inicial del programa y señalado con la flecha roja donde se encuentra el Download to device.

Figura 67. Cargar Programa a PLC.



Nota. Podemos observar los parámetros que se usan para la búsqueda y por qué red lo está realizando.

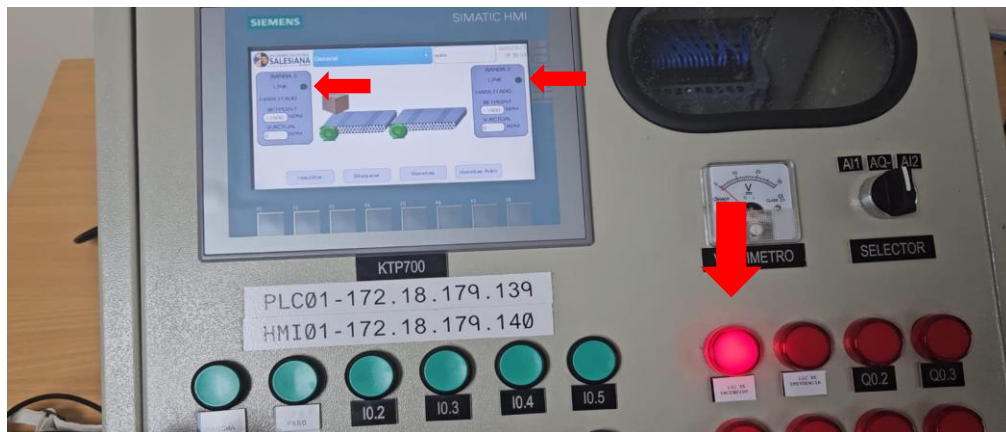
Después de cargar el programa a los PLCs y el HMI, en nuestro caso tenemos en la programación que los PLCs esclavos no manden una señal al momento de encender el proceso lo cual se ilumina una luz parpadeante en la pantalla principal de Proceso, en las siguientes figuras podremos observar cómo al momento de encender el master da la orden de ponerse en línea a los PLCs esclavos.

Figura 68. Sistema apagado



Nota. Se observa que el sistema está apagado en su totalidad.

Figura 69. Chequeo de encendido del sistema y comunicación correcta entre PLCs.

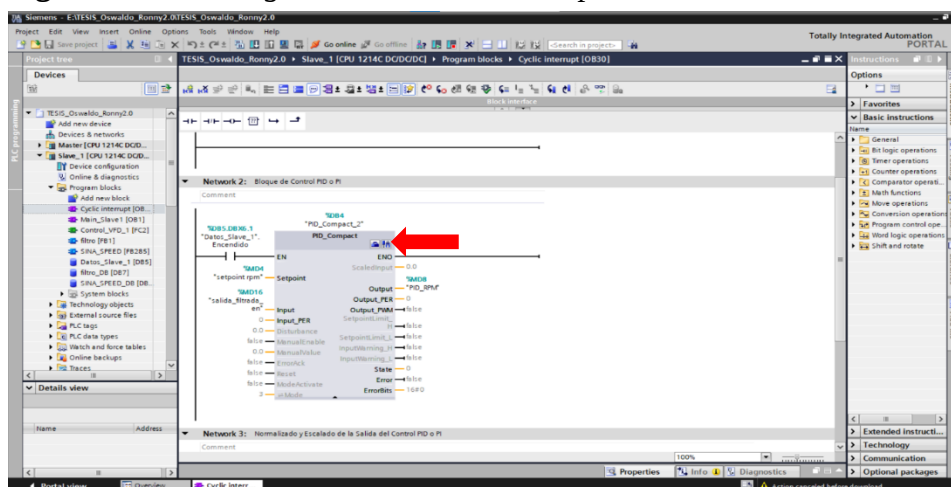


Nota. Se observa que al encender el sistema se enciende la luz de inicio y los PLCs esclavos mandan señal de estar en línea al PLC master que se observa mediante las luces de link.

7.2 Comparativa de controlador PID y PI en el sistema de control de velocidad para motores trifásicos.

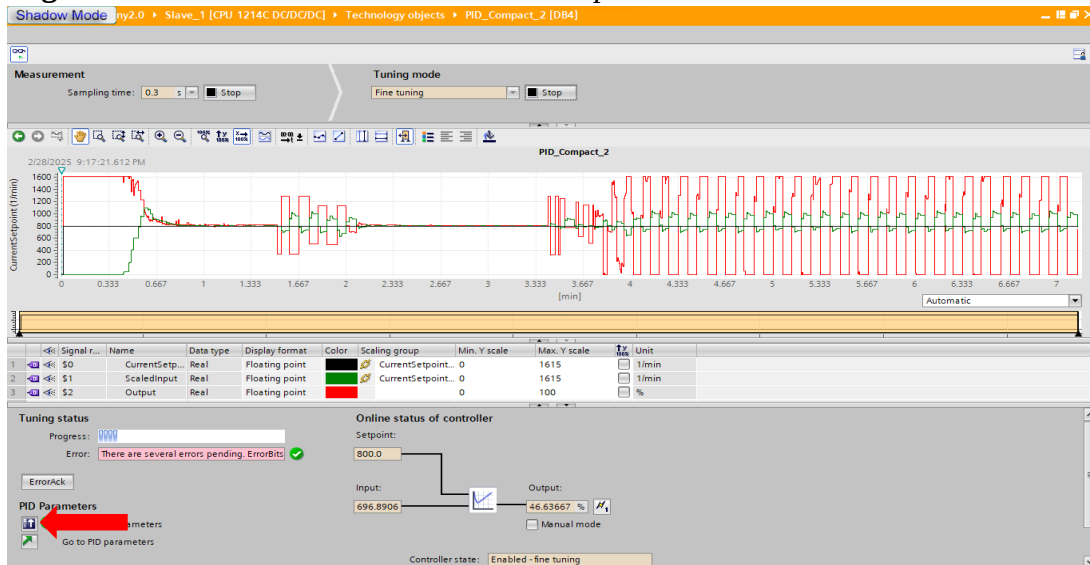
Al momento de tener ya listo la comunicación entre PLCs podemos centrarnos en la creación de un controlador en este caso usamos el bloque PID_Compact, este bloque nos permite hacer pruebas ya siendo un controlador PI o PID, en la figura 69 se verá el uso de la herramienta y como trabaja para la creación de los mejores parámetros KP, KI y KD para el sistema. A continuación, se puede ver como ingresar.

Figura 70. Como ingresar a la herramienta para crear controladores.



Nota. En la figura se puede ver donde ingresar a la herramienta de creación de controladores que está indicado con una flecha roja.

Figura 71. Proceso creación un controlador para el sistema.



Nota. Se puede observar cómo trabaja el bloque para crear el Controlador y señalado con la flecha roja enviar los parámetros creados al bloque PID_Compact.

Al tener los valores para KP, KI, KD podemos comparar los dos controladores entre si primero los dividimos en dos tablas en este caso es tabla 1 para los controladores PID y tabla 2 para los controladores PI, en este caso la variable que nos importa es el tiempo en el que se estabiliza la velocidad de los motores, en las figuras 70 y 71, tenemos una mejor visualización de la gráfica de estabilidad de uno de los motores de donde se obtuvo la información.

Tabla 1. Tabla de datos de los Controladores PID.

Controlador PID	Tiempo [Min]
Slave 1-Variador G120	0.82
Slave 2-Variador V20	0.3

Nota. En la siguiente tabla, se muestran el tiempo que demora en llegar a la estabilidad cada moto con el control PID.

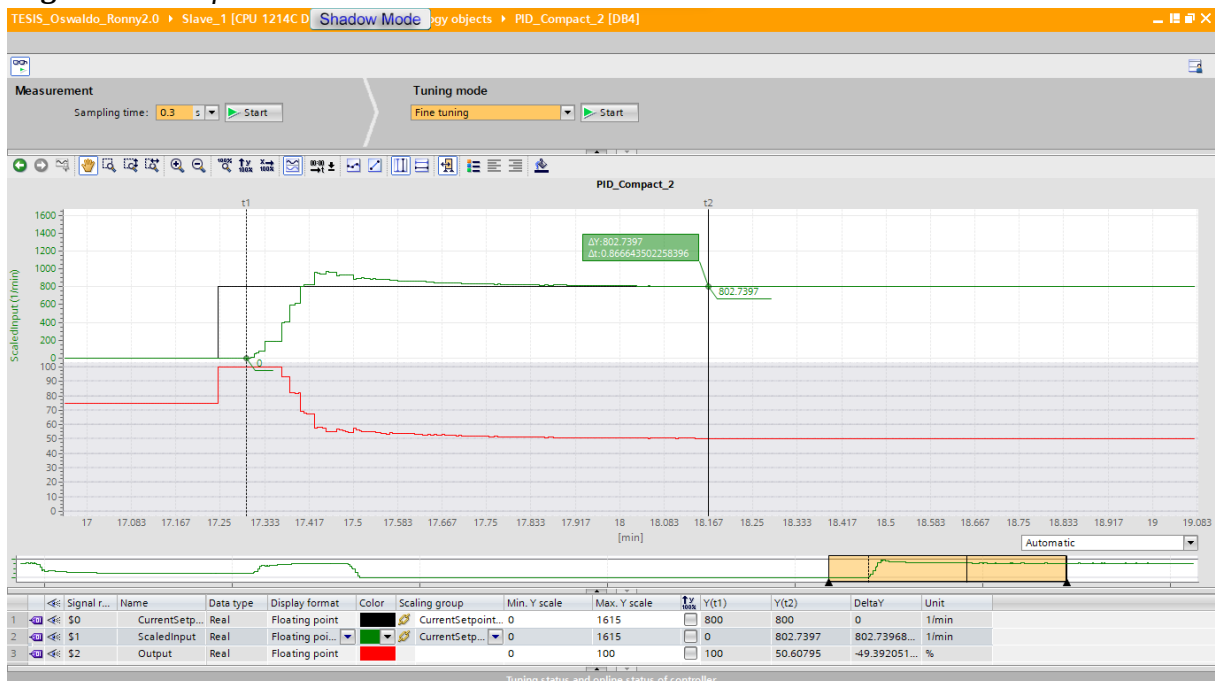
Tabla 2. Tabla de datos de los Controladores PI.

<i>Controlador PI</i>	<i>Tiempo [Min]</i>
Slave 1-Variador G120	0.87
Slave 2-Variador V20	0.51

Nota. En la siguiente tabla, se muestran el tiempo que demora en llegar a la estabilidad cada moto con el control PI.

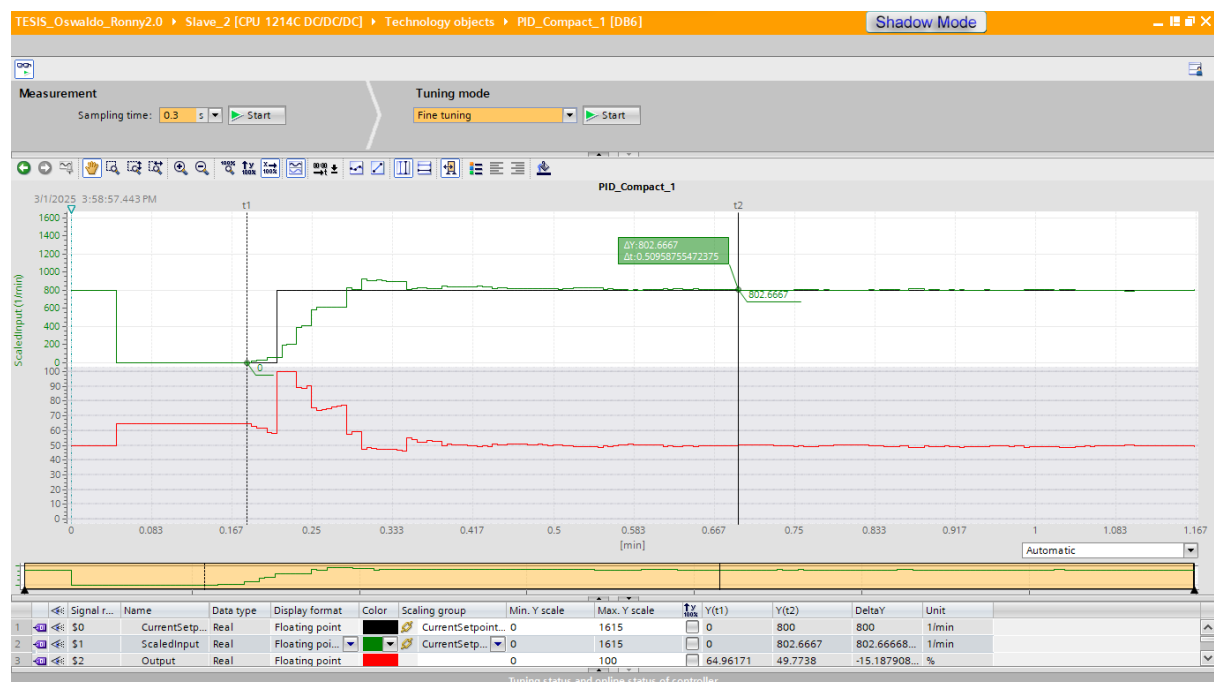
Tenemos en la figura 72 el trabajo de los motores ya implementando el control PID del bloque PID_Compact del Slave_1 que controla el variador g120 y a su vez el motor 1 para establecer su velocidad por medio del variador, vemos que la estabilidad es alcanzable en un tiempo relativamente corto.

Figura 72. Grafica del control PI en motor 1.



Nota. En la gráfica se puede observar cuando demora en estabilizarse la velocidad en motor 1.

Figura 73. Grafica del control PI en motor 2.



Nota. En la gráfica se puede observar cuando demora en estabilizarse la velocidad en motor 2.

Al revisar los datos se obtuvo que el mejor controlador en este caso sería el control PID ya que nos dan los mejores tiempos de estabilidad al comparar la tabla 1 con la tabla 2 siendo el variador v20 el que tiene una gran variación de tiempo entre el PID y PI, mientras que con el variador g120 existe una diferencia solo de 5 segundos entre el PID y PI, por lo cual llegamos a la conclusión que el controlador PID es el mejor.

7.3 HMI centralizada para sincronización de motores trifásicos.

Una vez cargado los programas a los PLCs se debe seguir cargando las pantallas en el HMI para observar y controlar los motores por medio de las mismas, a continuación, vamos a ir observando cada una de las pantallas programadas en el HMI para tener presente el funcionamiento de la misma la cual en el espacio físico trabaja sin problemas encendiendo, controlando y monitoreando cada las velocidades de los motores.

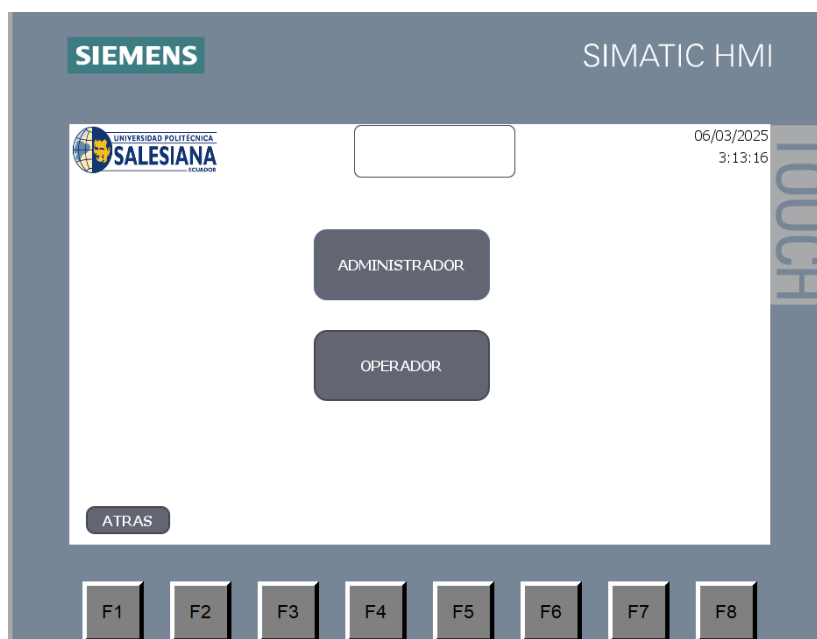
Figura 74. Pantalla de Presentación.



Nota. Pantalla de Presentación del proyecto.

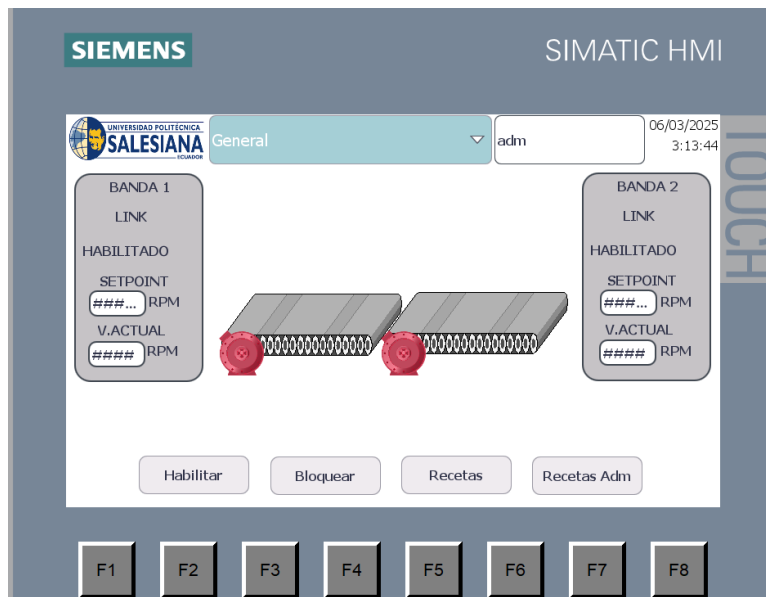
En la figura 75 observamos la pantalla para iniciar sesión como administrador o operario la cual nos podrá dar acceso a poder cambiar o modificar recetas al administrador y solo elegir de las recetas creadas al operario.

Figura 75. Pantalla de Inicio.



Nota. Pantalla de Inicio en el cual iniciamos sesión siendo administrador u operario.

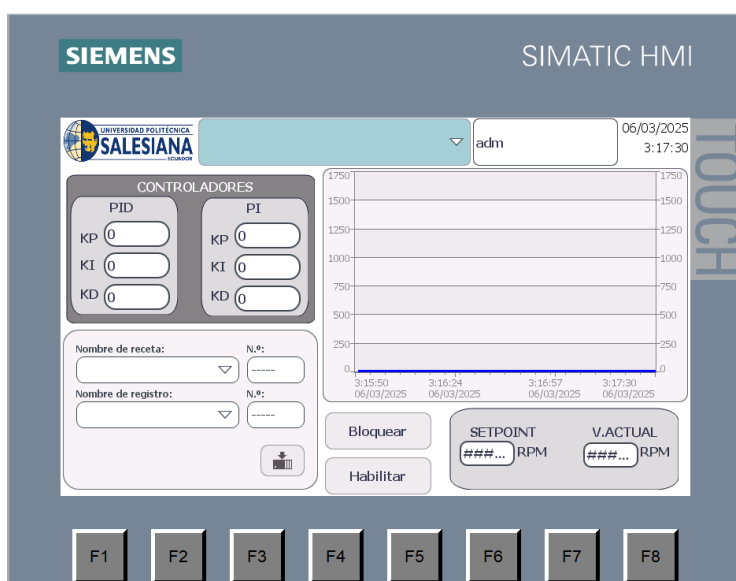
Figura 76. Pantalla de Proceso.



Nota. Pantalla de Proceso donde vemos información básica de cada banda y el inicio del sistema.

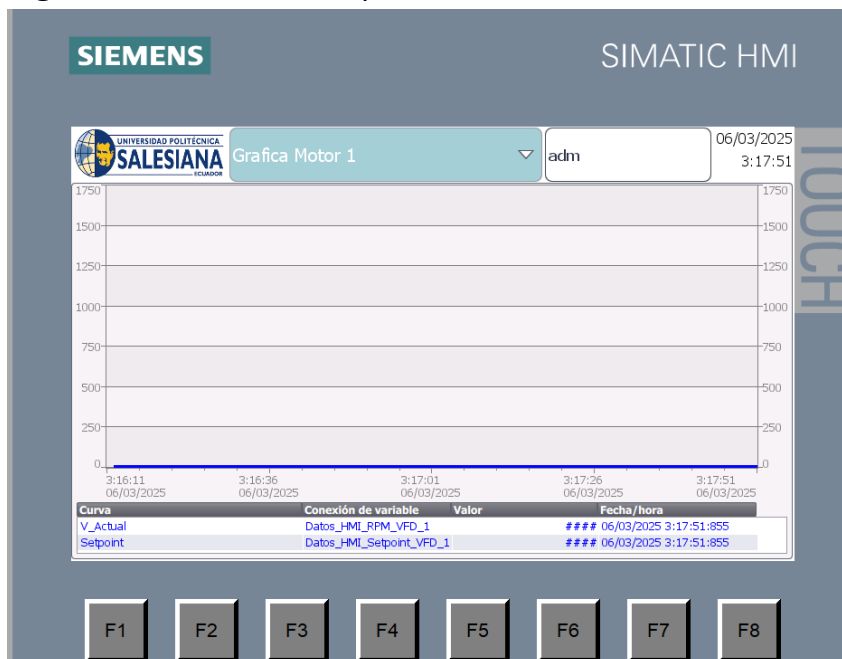
La gráfica que se puede observar en la figura 77 es donde se puede ver el setpoint y la velocidad actual del motor que nos dará el encoder conectado al mismo, estos dos valores al momento de ser iguales nos darán el tiempo que demora en estabilizarse y comprobar si el controlador es bueno o se puede mejorar.

Figura 77. Pantalla de Motor 1.



Nota. Pantalla de donde se puede controlar y revisar los valores del Motor 1.

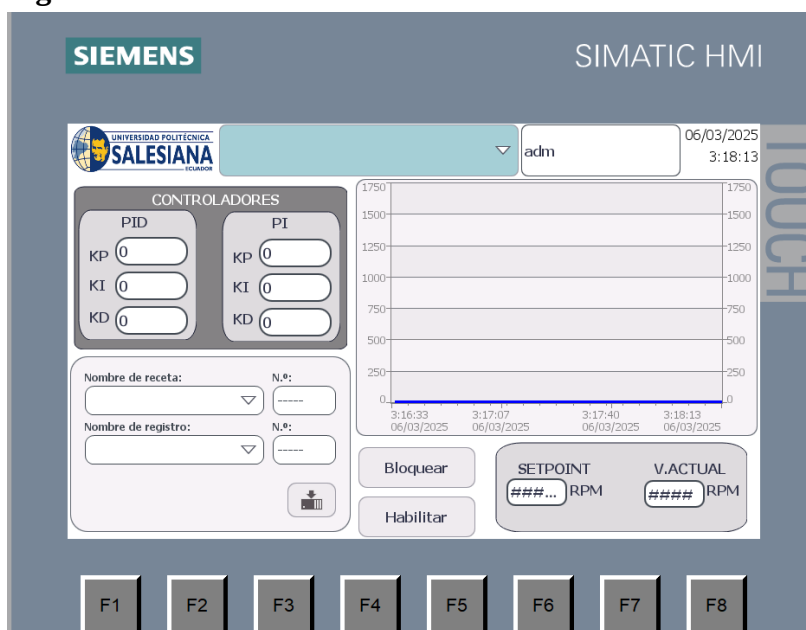
Figura 78. Pantalla de Grafico Motor 1.



Nota. Pantalla en la cual se ve con mayor detalle la gráfica del motor 1.

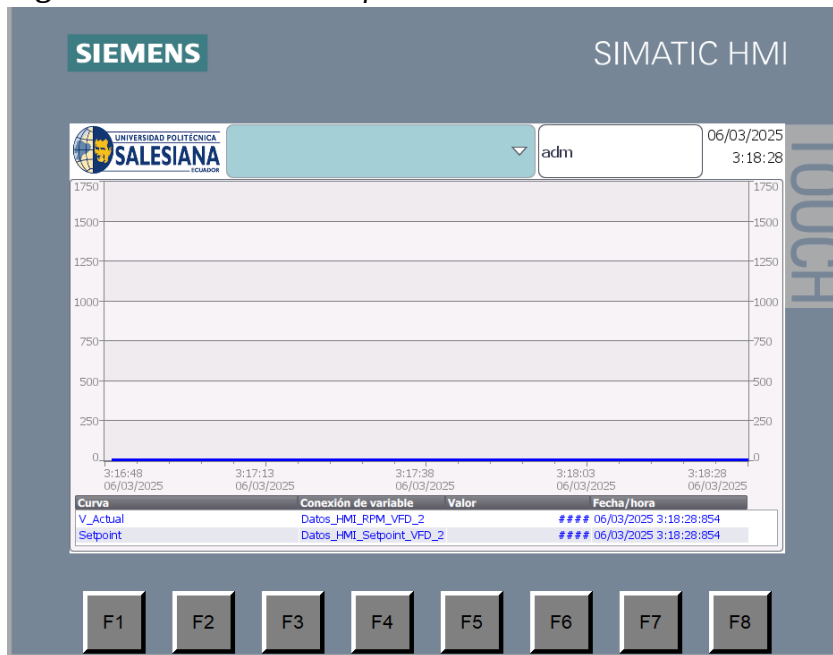
La figura 79 podemos observar lo valores del controlador PI y PID que estamos usando, el funcionamiento del mismo se puede observar en la grafica que se encuentra al lado derecho de la pantalla donde se compraran el setpoint y el valor actual de la velocidad.

Figura 79. Pantalla de Motor 2.



Nota. Pantalla de donde se puede controlar y revisar los valores del Motor 1.

Figura 80. Pantalla de Grafico Motor 1.



Nota. Pantalla en la cual se ve con mayor detalle la gráfica del motor 2.

Podemos observar en la figura 81 la pantalla de recetas en este caso sería para los operarios el cual solo se permite elegir las recetas ya configuradas previamente por el administrador, aparte de poder cargar la receta elegida y observar la velocidad en la que se pondrá el setpoint de las bandas.

Figura 81. Pantalla de Recetas Operario.

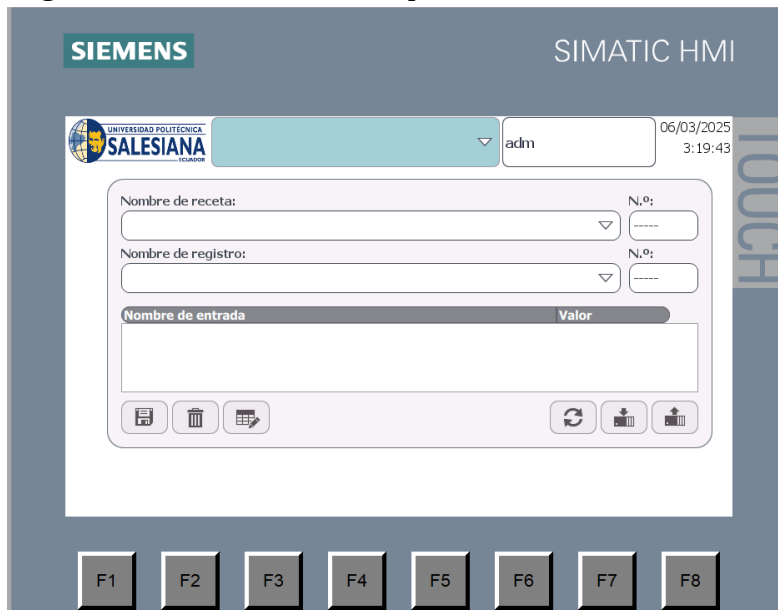
Nombre de receta: [dropdown] N.º: [input]

Nombre de registro: [dropdown] N.º: [input]

Nombre de entrada [dropdown] Valor [input]

Nota. Pantalla donde el operario puede cambiar las recetas que el administrador configure.

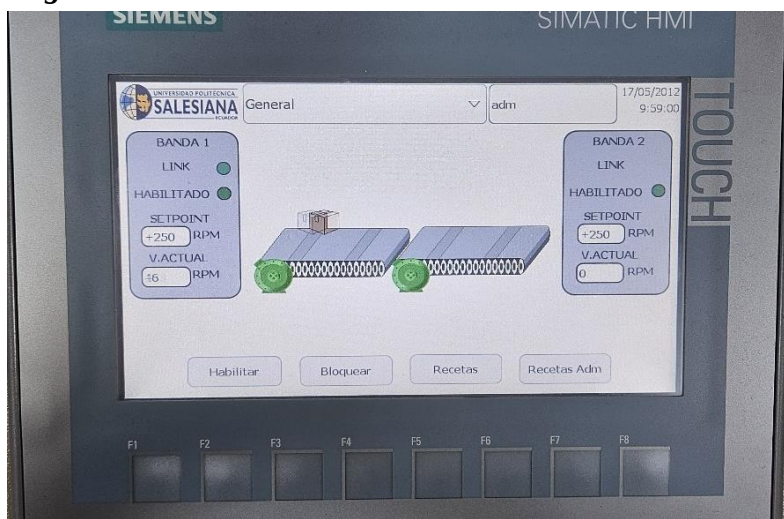
Figura 82. Pantalla de Rectas para el Administrador.



Nota. Pantalla donde el administrador puede crear recetas para cada producto que quiera.

Al ya tener las pantallas definidas para su control, tenemos el funcionamiento de las misma en el HMI en la parte física, donde se puede observar las diferentes variables y cambios que se pueden notar por medio de esta y tener el control de las bandas en un solo lugar, en las siguientes figuras vamos a ver las pantallas en funcionamiento y al finaliza el proyecto en su totalidad.

Figura 83. Pantalla de Proceso en funcionamiento.



Nota. Pantalla en funcionamiento del Proceso donde se ve los PLCs esclavos en línea y los datos entrantes junto a su animación.

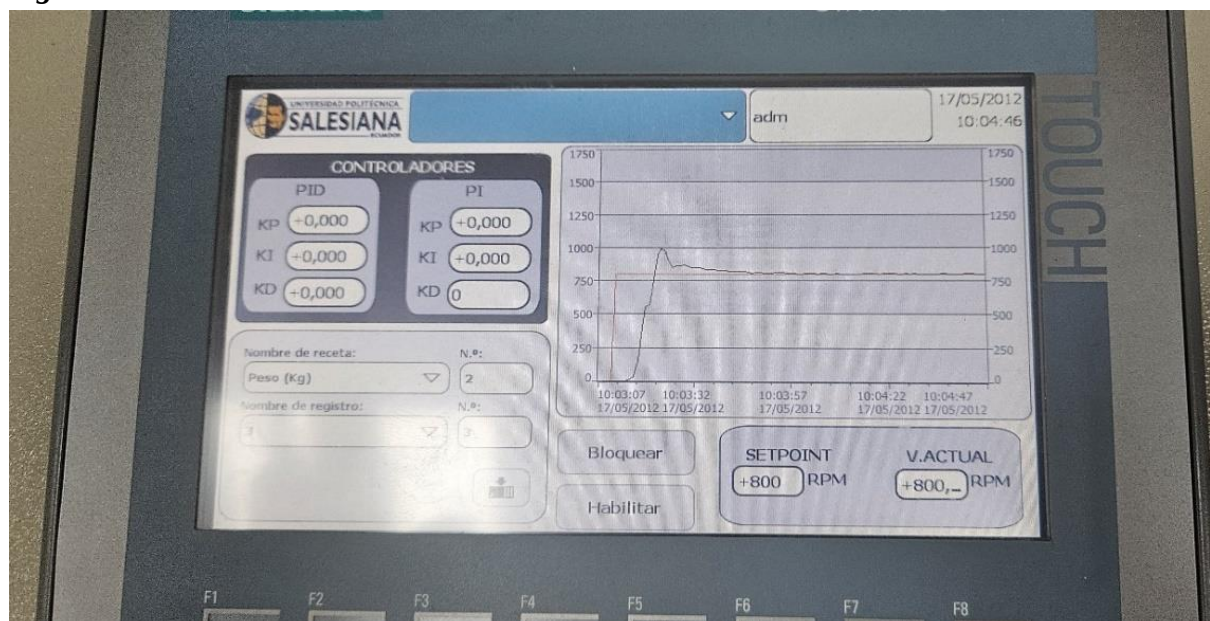
Figura 84. Pantalla en funcionamiento de Motor 1.



Nota. Pantalla en funcionamiento donde se puede observar los valores del motor 1 cambiando, dando nos la información adecuada del motor.

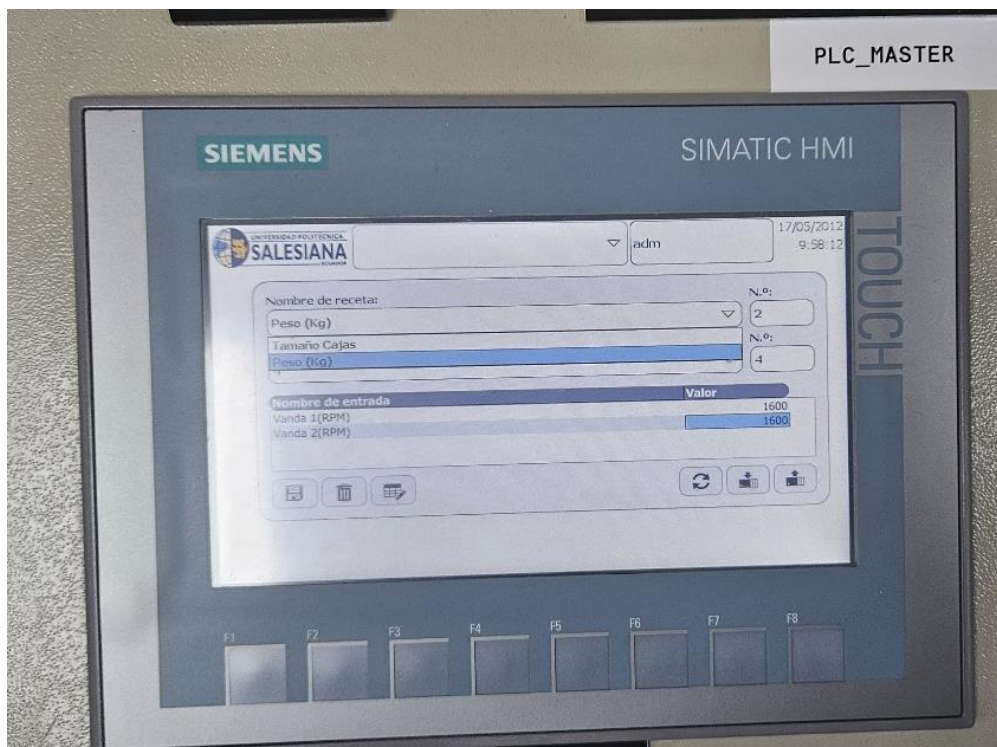
Podemos darnos cuenta que la grafica ya nos da un resultado de como trabaja el controlador puesto dentro del Slave_2, esto se ve en la figura 85, donde se puede visualizar también la receta y los valores de setpoint junto a la velocidad actual.

Figura 85. Pantalla en funcionamiento de Motor 2.



Nota. Pantalla en funcionamiento donde se puede observar los valores del motor 2 cambiando, dando nos la información adecuada del motor.

Figura 86. Pantalla en funcionamiento de Recetas Administrador.



Nota. Pantalla en funcionamiento de Recetas Administrador donde se pueden crear las recetas.

En la figura 87 se puede observar en su totalidad el proyecto en físico y como esta conectado aparte de la distribución de los componentes para el funcionamiento del mismo, en este caso el PLC master estaría al fondo y seguido por el Slave 1 y 2 respectivamente.

Figura 87. Proyecto Físico Final.



Nota. Figura donde se observa el proyecto físico final en su totalidad

VIII CRONOGRAMA DE ACTIVIDADES

En esta sección tenemos el cronograma de actividades que se estiman para el cumplimiento del proyecto en su totalidad, está formado por 7 fases, con fecha de inicio en el mes de diciembre de 2024 y fecha de finalización en el mes de febrero de 2025, se detallan a continuación en la siguiente tabla:

Tabla 3, *Cronograma de actividades para el desarrollo del Proyecto de Titulación.*

Actividades	Tiempo de duración												
	Meses	DICIEMBRE				ENERO				FEBRERO			
	Semanas	1	2	3	4	1	2	3	4	1	2	3	4
Investigación sobre el proyecto actual	X												
Compra de materiales		X											
Ensamble de los componentes			X										
Programación en TIA PORTAL					X	X							
Prueba del prototipo sin controlador						X	X						
Creación del controlador con los datos obtenidos de la prueba del prototipo							X	X	X				
Pruebas del prototipo con controlador										X	X	X	
Presentación de proyecto													X

Nota. En la siguiente tabla, se muestra el cronograma de actividades ajustado por fechas.

IX PRESUPUESTO

Considerando que el proyecto será desarrollado con varios componentes ya existen en la universidad dentro del presupuesto se ha considerado posibles gastos generados en las reuniones de trabajo e implementos de trabajo, así como impresiones del proyecto de titulación para revisiones y presentaciones del mismo, valores que se detallan en la Tabla 2, sumando un estimado de \$265,72.

Tabla 4. *Presupuesto estimado para el desarrollo del Proyecto de Titulación.*

<i>DETALLE</i>	<i>CANTIDAD</i>	<i>COSTO</i>	<i>COSTO TOTAL</i>
		<i>UNITARIO</i>	
<i>Horas de ingeniería</i>	<i>40h</i>	<i>\$2,81</i>	<i>\$112,00</i>
<i>Adaptador encoder a motor trifásico</i>	<i>2 unidad</i>	<i>\$40,00</i>	<i>\$80,00</i>
<i>Encoder</i>	<i>2 unidad</i>	<i>\$20,00</i>	<i>\$40,00</i>
<i>Prueba de funcionamiento en laboratorio UPS</i>	<i>10h</i>	<i>\$2,81</i>	<i>\$28,10</i>
<i>Sustentación el sistema propuesto</i>	<i>2h</i>	<i>\$2,81</i>	<i>\$5,62</i>
<i>TOTAL</i>			<i>\$265,72</i>

Nota. Con la presente tabla, se detalla el presupuesto referencial del diseño del proyecto de titulación.

X CONCLUSIONES

Se ha creado una red de comunicaciones Profinet entre lectura y control, lo que le permite controlar más interacciones entre dispositivos. La implementación de Profinet proporciona comunicación en tiempo real, asegurando que los datos de velocidad se actualicen y controlen efectivamente al optimizar el rendimiento y la confiabilidad del sistema.

El sistema de control PID y PI se ha diseñado e implementado con éxito en el portal de tía para probar la velocidad de los motores de tres fases con variaciones de frecuencia. El uso de estos controladores garantiza el ajuste exacto y estable de intercambio de motor, lo que mejora el rendimiento y la eficiencia del sistema en diferentes condiciones de funcionamiento.

Se ha desarrollado un HMI centralizado para monitorear y configurar de manera efectiva la sincronización del revestimiento del motor de tres fases. Este sistema proporciona una interfaz intuitiva para controlar y personalizar los parámetros de tiempo real, mejorar el monitoreo y la gestión del rendimiento de la luz.

XI RECOMENDACIONES

Es aconsejable monitorear los resultados de la red Profinet y prevenir periódicamente la confiabilidad y estabilidad del sistema. También es importante capacitar a los empleados responsables de la operación y mantenimiento del sistema para identificar rápidamente posibles errores o anomalías en la comunicación.

Se recomienda que el controlador PID y PI ajusten la prueba de prueba en varias cargas y condiciones de funcionamiento para optimizar su rendimiento. Además, se debe realizar un mantenimiento preventivo regular del sistema y el personal puede garantizar la gestión correcta y la detección temprana de posibles errores.

Se recomienda un uso apropiado y las pruebas de funcionalidad para garantizar que HMI sea intuitivo y fácil de operar en diferentes condiciones. Además, se propone integrar ansiedad e informes en caso de errores o inconsistencias en el caso de sincronización y educar a los empleados sobre el uso de HMI para optimizar su rendimiento.

XII REFERENCIA BIBLIOGRÁFICAS

- ARROW. (31 de OCTUBRE de 2019). *ARROW*. Obtenido de <https://www.arrow.com/es-mx/research-and-events/articles/capacitive-magnetic-and-optical-encoders-comparing-the-technologies>
- Bouchakri Zaamy, A. (2021). *Diseño e implementación de una aplicación de control de proceso industrial sobre pantalla HMI*.
- Bowne, M. (1 de Agosto de 2020). *PROFIBUS.PROFINET*. Obtenido de <https://us.profinet.com/canales-de-comunicacion-profinet/#:~:text=PROFINET%20es%20un%20estándar%20abierto,industrial%20más%20usados%20del%20mundo>.
- Bowne, M. (4 de Agosto de 2020). *PROFIBUS.PROFINET*. Obtenido de <https://us.profinet.com/profinet-versus-opc-2/>
- Cuzcano Rivas, A. B. (2023). *CONTROL DIGITAL DE MOTORES TRIFÁSICOS USANDO FREERTOS*.
- Eltra Trade. (29 de OCTUBRE de 2023). *Eltra Trade*. Obtenido de <https://eltra-trade.com/blog/sinamics-v20-fault-codes>
- Escuale Control+. (24 de ENERO de 2023). *Escuale Control+*. Obtenido de <https://miembros-escuelacontrolmas.com/courses/variadores-industriales-siemens-sinamics-g120/>
- Farina, A. (2018). *Motores eléctricos trifásicos: usos, componentes y funcionamiento. Suplemento instaladores*.
- Granda Rivera Jasson Stalin, B. S. (2025). *Integración de sistemas de comunicación Modbus serial RS-485 entre HMI KINCO y variador SIEMENS V20*. Santo Domingo: Instituto Tecnológico Superior Tsachila.

GSL Industrias. (1 de JUNIO de 2021). *Diseño y construcción del módulo de entrenamiento# 15 con el PLC S7-1500 con comunicación ethernet*. Obtenido de https://industriasgsl.com/blogs/automatizacion/que-es-un-plc-y-como-funciona?srsltid=AfmBOooveFKqey6Frlp8HgXvOZiEjuizv2_9bYx9QuLPuTlLjw8gP6-O

Hwang Cárdenas, J. H., & Iñiguez Avila, J. A. (2023). *SISTEMA DE CONTROL Y MONITOREO DE MOTORES TRIFÁSICOS CON SOPORTE DE LabVIEW*.

infopl. (11 de MARZO de 2021). *infopl*. Obtenido de <https://www.infopl.net/descargas/107-siemens/software-step7-tiaportal/tiaportal/3240-simulacion-pidcompact-siemens-tia-portal>

IronTech. (3 de AGOSTO de 2023). *IronTech*. Obtenido de <https://irontech-group.com/es/que-es-un-hmi-el-significado-y-la-importancia-del-human-machine-interface/>

Jaime, A. M. (2008). *Control PID de la velocidad de una banda transportadora para la clasificación de objetos*. MEXICO.

Loayza Rodríguez, & Mendiola Villanueva. (2020). *Diseño e implementación de un módulo de control de nivel de líquido usando un variador de velocidad sinamics V20 y arduino para la adquisición de señales mediante comunicación Modbus RTU y TCP*.

Logicbus. (14 de JUNIO de 2019). *Logicbus*. Obtenido de <https://www.logicbus.com.mx/blog/modbus-rtu/>

LONDON CONSOLTING GROUP. (4 de Junio de 2024). *LONDON CONSOLTING GROUP*. Obtenido de <https://londoncg.com/blog/como-optimizar-los-procesos-con-un-sistema-de-manufactura>

lتي Engenharia. (11 de MAYO de 2021). *lتي Engenharia*. Obtenido de <https://www.lتي Engenharia.com.br/2021/05/11/sistemas-de-controle-controlador-pid-parte-02/>

Magnetism, E. . (2024). *Electricity - Magnetism*. Obtenido de <https://www.electricity-magnetism.org/es/motores-de-induccion-de-jaula-de-ardilla/>

Peláez, A. (16 de OCTUBRE de 2024). *ubidots*. Obtenido de <https://es.ubidots.com/blog/siemens-simatic-s7-modbus/>

Plata, C. S. (10 de Noviembre de 2014). *Ingenieria Creativa*. Obtenido de <https://carolina-sp.blogspot.com/2014/11/contador-de-pulsos-de-encoder-con.html>

Plata, C. S. (s.f.). *Ingenieria Creativa*.

PP, J. (30 de JULIO de 2020). *IESPV-STEAM*. Obtenido de <https://iespedrodevaldivia.net/steam/2020/07/30/encoder-magnetico/>

Profibus.Profinet. (15 de Noviembre de 2023). *Profibus.Profinet*. Obtenido de <https://profibus.com.ar/profinet-que-es-y-como-funciona/#:~:text=El%20estándar%20PROFINET%20es%20una,costes%20y%20mejora%20del%20rendimiento.>

Quispe Mendoza, E. (2021). *Implementación de un módulo electrónico relé de sincronización en tablero eléctrico para mejorar el sistema de arranque de motor síncrono en la empresa Mixercon S.A-2020*.

ROSALENY, R. T. (2021). *Encoders Ópticos*.

S&P. (9 de Julio de 2024). *solerpalau*. Obtenido de <https://www.solerpalau.com/es-es/blog/motor->

trifasico/#:~:text=El%20estator%20es%20la%20parte,que%20están%20presentes%20unas%20ranuras.

Satorres Mira, S. (2021). *Automatización de un paletizador industrial. Programación con TIA Portal y virtualización con Factory IO*.

Sensor, C. (2021). *SSI Output Single-Turn Absolute Encoder CAS38 Series - CALT Sensor*.

SIEMENS. (2017). *SINAMICS G120*. Atlanta: Siemens Industry, Inc.

Siemens. (25 de Julio de 2018). *Siemens*. Obtenido de <https://support.industry.siemens.com/cs/mdm/109759862?c=90079397515&lc=es-HN>

siemens. (2021). *siemens*. Obtenido de <https://www.siemens.com/ec/es.html>

siemens, P. (2013). *Programacion siemens*. Obtenido de https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQmS3RFxGWRFXI00ui32QQh9Y09N7R7VxJuAR_57LFHL-rWRL4pdlnQjZsBqTxnoeZZcLU&usqp=CAU

SINADRIVES. (12 de 2020). *SINADRIVES*. Obtenido de En relación con el punto anterior, los encoders absolutos utilizan un bus de comunicación digital, lo que ha dado lugar a infinidad de protocolos de comunicación. Cada fabricante, ya sea de encoder o de driver, intenta fomentar su propio protocolo, hacien

Smith, G. M. (19 de JUNIO de 2024). *Introducción a los Sistemas de Control*. Obtenido de <https://dewesoft.com/es/blog/que-es-un-controlador-pid>

Smith, J. (2020). *Optimización de procesos en la industria manufacturera*. Editorial Industrial.

Tech Briefs. (1 de Octubre de 2019). *Tech Briefs*. Obtenido de <https://www.techbriefs.com/component/content/article/35291-how-to-use-rotary-encoders-to-quickly-convert-mechanical-rotation-into-digital-signals>

Thanh, P. H. (JUNIO de 2021). *RESEARCHGATE*. Obtenido de https://www.researchgate.net/figure/PID-Compact-function-in-TIA-Portal-To-open-the-configuration-window-for-the-PID-Compact_fig2_352777970

Universidad Autonoma del Estado de Hidalgo. (2017). *BOLETÍN CIENTÍFICO InvESTigium de La Escuela Superior de Tizayuca*. Estado de Hidalgo: InvESTigium.

Weis, O. (26 de Septiembre de 2023). *Virtual Serial Port*. Obtenido de <https://www.virtual-serial-port.org/es/articles/modbus-rtu-guide/#1>

YUQUILEMA TRUJILLO RICARDO XAVIER, R. B. (2019). *IMPLEMENTACIÓN DE UN MÓDULO DE SINCRONIZACIÓN Y*. Riobamba: ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

XIII ANEXOS

13.1 Anexo 1. Programación de PLC Master (Main_Master [OB1])

Totally Integrated Automation Portal		
--------------------------------------	--	--

TESIS_Oswaldo_Ronny2.0 / Master [CPU 1214C DC/DC/DC] / Program blocks

Main_Master [OB1]

Main_Master Properties							
General							
Name	Main_Master	Number	1	Type	OB	Language	LAD
Numbering	Automatic						
Information							
Title	"Main Program Sweep (Cycle)"	Author		Comment		Family	
Version	0.1	User-defined ID					

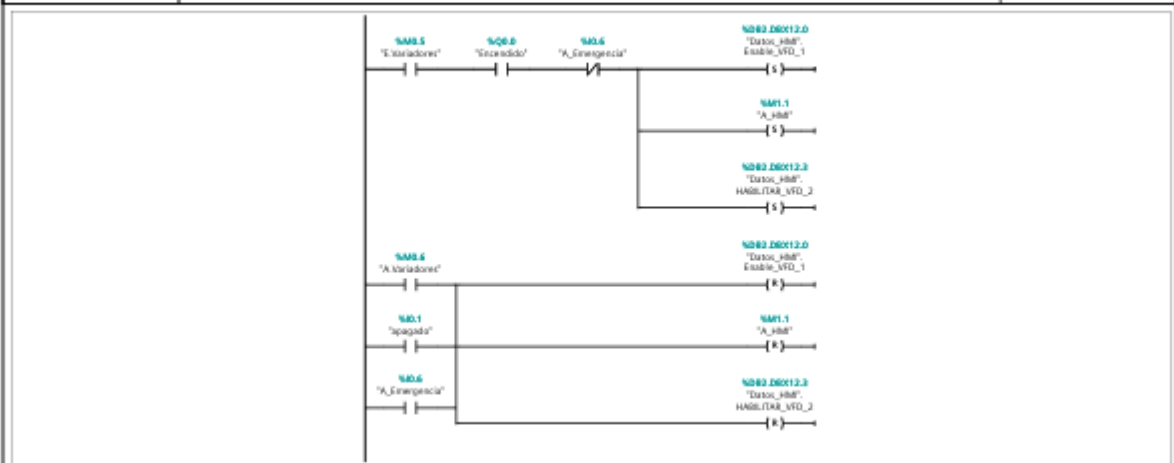
Name	Data type	Default value	Comment
▼ Input			
Initial_Call	Bool		Initial call of this OB
Remanence	Bool		=True, if remanent data are available
Temp			
Constant			

Network 1: Encendido del sistema

Network 2: Mover valor de setpoint

Network 3: Cambio de Tamaño para Animacion HMI

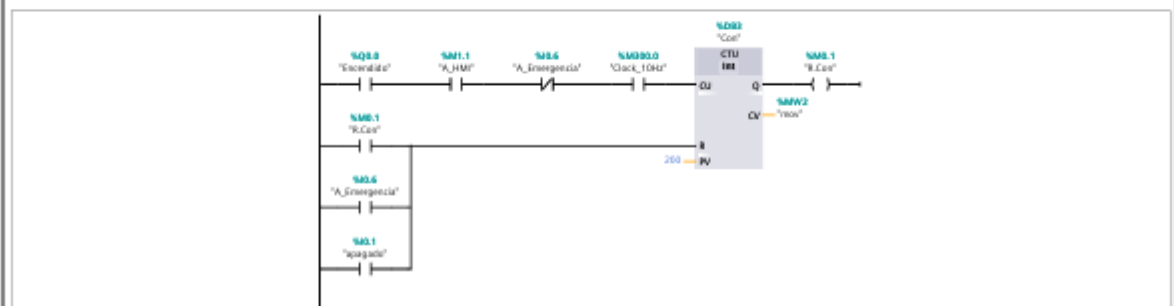
Network 4: Habilitacion de Motores



Network 5: Luces de Emergencia y de Linea de PLCs Slaves

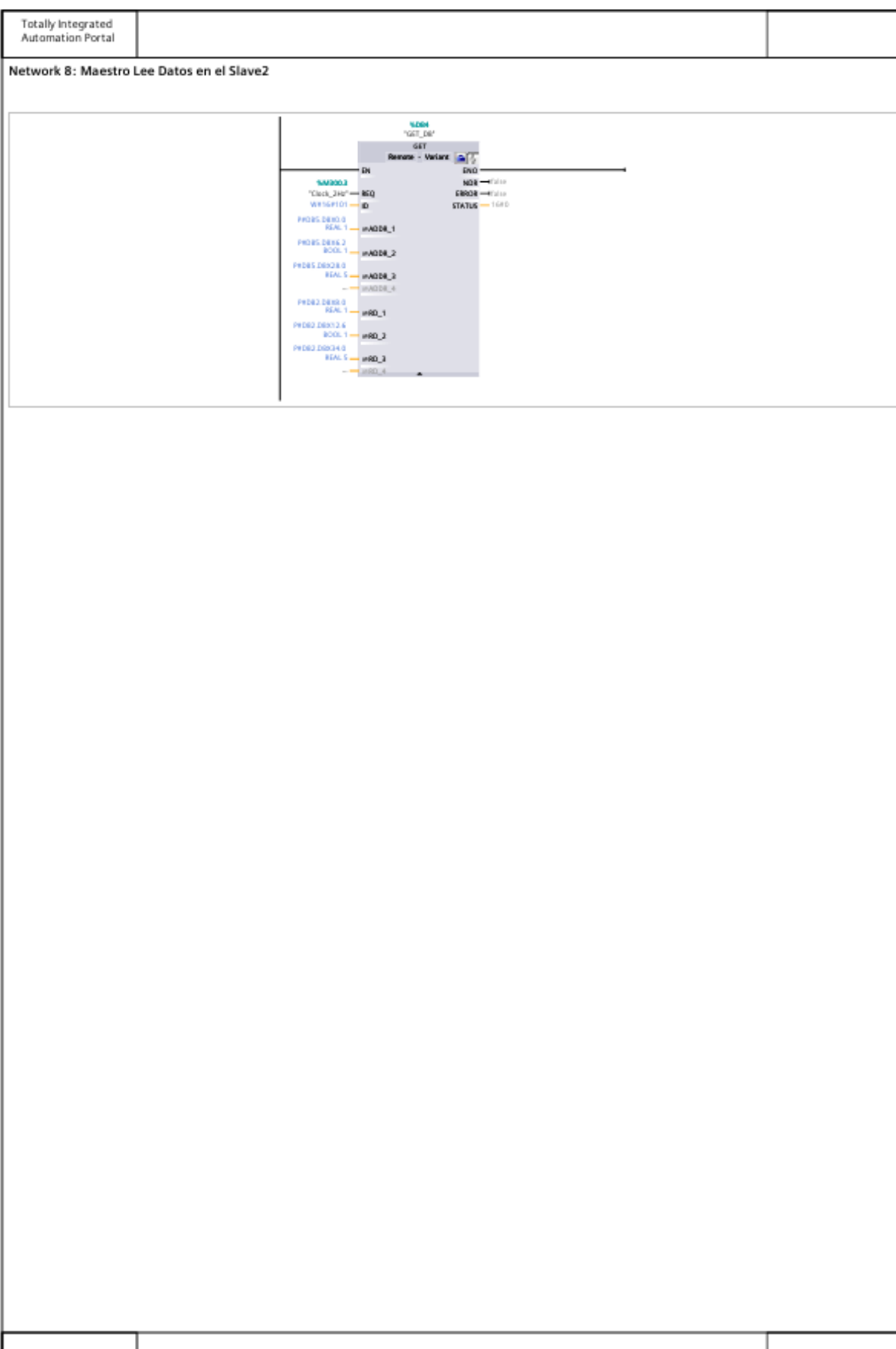


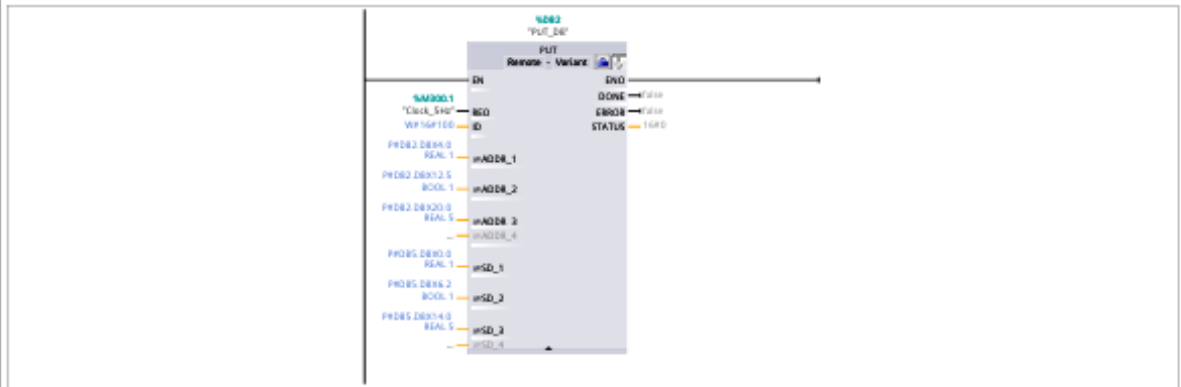
Network 6: Contador para Animacion de HMI



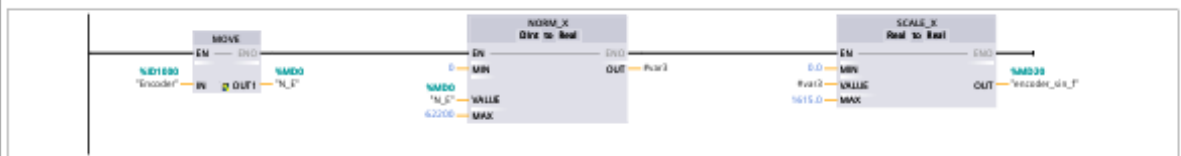
Network 7: Maestro Escribe Datos en el Slave2



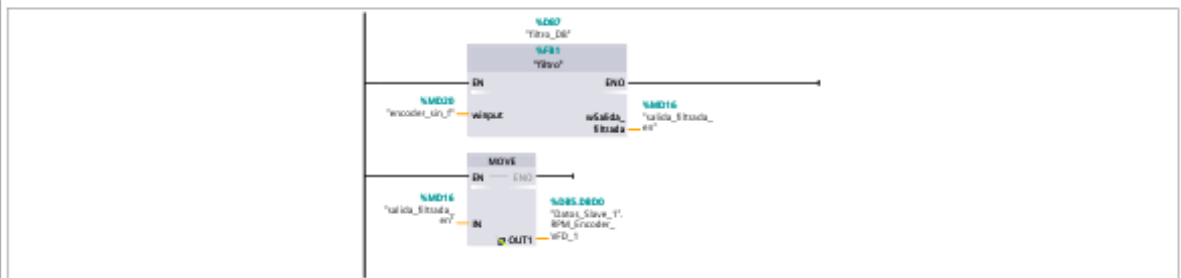




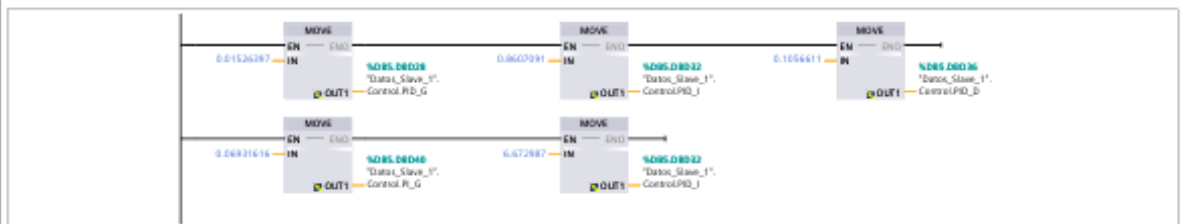
Network 5: Normalizar y Escalar la Señal Entrante del Encoder



Network 6: Filtrado de la Señal del Encoder



Network 7:



13.2.2 Anexo 3. Programación del bloque Cyclic interrupt [OB30].

Totally Integrated Automation Portal	
--------------------------------------	--

TESIS_Oswaldo_Ronny2.0 / Slave_1 [CPU 1214C DC/DC/DC] / Program blocks

Cyclic interrupt [OB30]

Cyclic interrupt Properties			
General			
Name	Cyclic interrupt	Number	30
Type	OB	Language	LAD
Numbering			
Automatic			
Information			
Title	Author	Comment	Family
Version	0.1	User-defined ID	

Name	Data type	Default value	Comment
Input			
Initial_Call	Bool		Initial call of this OB
Event_Count	Int		Events discarded
Temp			
VAR4	Real		
Constant			

Network 2: Bloque de Control PID o PI

Network 3: Normalizado y Escalado de la Salida del Control PID o PI

13.2.3 Anexo 4. Programación del bloque Control_VDF_1 [FC2].

Totally Integrated Automation Portal		
--------------------------------------	--	--

TESIS_Oswaldo_Ronny2.0 / Slave_1 [CPU 1214C DC/DC/DC] / Program blocks

Control_VFD_1 [FC2]

Control_VFD_1 Properties							
General							
Name	Control_VFD_1	Number	2	Type	FC	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Comment
Input			
setpoint	Real		
Output			
InOut			
Temp			
Constant			
Return			
Control_VFD_1	Void		

Network 1: Inicio secuencia Driver 1

Network 2: Se Convierte el Setpoint de Int a Real

13.2.4 Anexo 5. Programación del bloque filtro [FB1].

Totally Integrated Automation Portal					
--------------------------------------	--	--	--	--	--

TESIS_Oswaldo_Ronny2.0 / Slave_1 [CPU 1214C DC/DC/DC] / Program blocks

filtro [FB1]

filtro Properties

General							
Name	filtro	Number	1	Type	FB	Language	LAD
Numbering		Automatic					
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Write-Visible in HMI engineering	Setpoint	Supervision	Comment
▼ Input								
winput	Real	0.0	Non-retain	True	True	False		
▼ Output								
wSalida_filtrada	Real	0.0	Non-retain	True	True	False		
InOut								
Static								
▼ Temp								
resultado_s	Real							
j	Int							
Constant								

Network 1: Contador Incremental para variable J

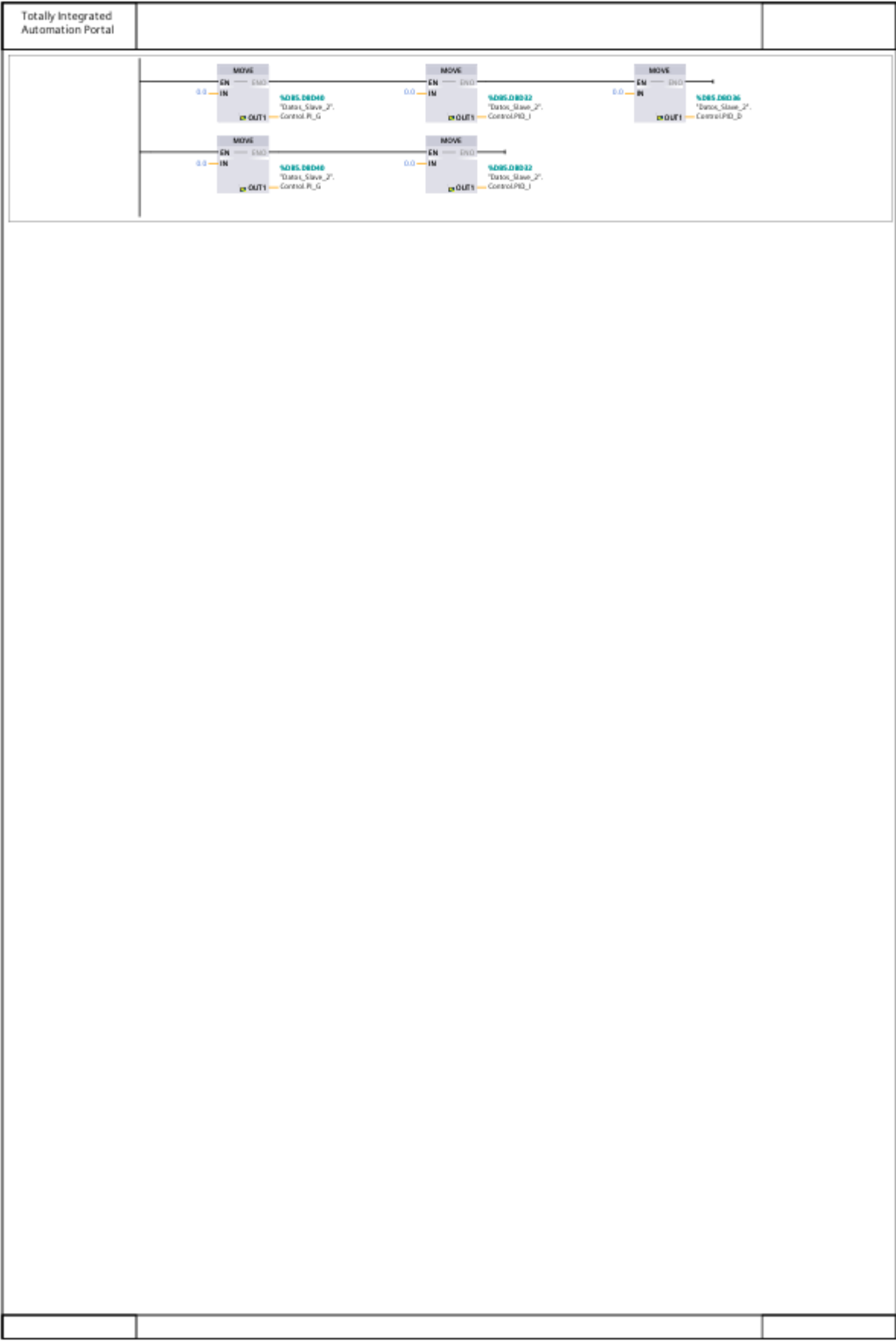
Network 2: Se Mueve el Dato de Entrada al Registro de J

Network 3: Promediado del Registro de J

13.3 Programación de PLC Slave_2

13.3.1 Anexo 6. Programación del Main_Slave2[OB1]

Totally Integrated Automation Portal					
TESIS_Oswaldo_Ronny2.0 / Slave_2 [CPU 1214C DC/DC/DC] / Program blocks					
Main_Slave2 [OB1]					
Main_Slave2 Properties					
General					
Name	Main_Slave2	Number	1	Type	OB
Numbering	Automatic				
Information					
Title	"Main Program Sweep (Cycle)"	Author		Comment	
Version	0.1	User-defined ID			
Family					
Name	Data type	Default value	Comment		
Input					
Initial_Call	Bool		Initial call of this OB		
Remanence	Bool		=True, if remanent data are available		
Temp					
var3	Real				
Constant					
Network 1: Encendido del Sistema, envia señal a master					
Network 2: Habilitacion del motor para trabajar					
Network 3: Llamar al Bloque que Controla el Variador de Frecuencia					
Network 4: Normalizar y Escalar la Señal Entrante del Encoder					
Network 5: Filtrado de la Señal del Encoder					
Network 6:					



13.3.2 Anexo 7. Programación del Cyclic interrupt[OB30]

Totally Integrated Automation Portal		
TESIS_Oswaldo_Ronny2.0 / Slave_2 [CPU 1214C DC/DC/DC] / Program blocks		
Cyclic interrupt [OB30]		
Cyclic interrupt Properties		
General		
Name	Cyclic interrupt	
Number	30	
Type	OB	
Language	LAD	
Numbering		
Automatic		
Information		
Title	Author	Comment
Version	User-defined ID	Family
Inputs		
Name	Data type	Default value
▼ Input		
Initial_Call	Bool	
Event_Count	Int	
▼ Temp		
VAR5	Real	
Constant		
Network 1:		
Network 2: Bloque de Control PID o PI		
Network 3: Normalizado y Escalado de la Salida del Control PID o PI		

13.3.3 Anexo 8. Programación del Control_VFD_2[FC1]

Totally Integrated Automation Portal		
--------------------------------------	--	--

TESIS_Oswaldo_Ronny2.0 / Slave_2 [CPU 1214C DC/DC/DC] / Program blocks

Control_VFD_2 [FC1]

Control_VFD_2 Properties

General			
Name	Control_VFD_2	Number	1
Type	FC	Language	LAD
Numbering			
Automatic			
Information			
Title		Author	
Version	0.1	User-defined ID	
Comment		Family	

Name	Data type	Default value	Comment
▼ Input			
setpoint	Real		
Output			
InOut			
▼ Temp			
var5	Real		
Constant			
▼ Return			
Control_VFD_2	Void		

Network 2: Normaliza y Escala la el Setpoint

Network 3: Se Convierte el Setpoint de Int a Real

13.3.4 Anexo 9. Programación del Filtro [FB1]

Totally Integrated Automation Portal		
--------------------------------------	--	--

TESIS_Oswaldo_Ronny2.0 / Slave_2 [CPU 1214C DC/DC/DC] / Program blocks

Filtro [FB1]

Filtro Properties							
General							
Name	Filtro	Number	1	Type	FB	Language	LAD
Numbering	Automatic						
Information							
Title		Author		Comment		Family	
Version	0.1	User-defined ID					

Name	Data type	Default value	Retain	Accessible from HMI/OPC UA/Web API	Write-Visible from HMI engineering	Setpoint	Supervision	Comment
▼ Input								
winput	Real	0.0	Non-retain	True	True	True	False	
▼ Output								
wSalida_filtrada	Real	0.0	Non-retain	True	True	True	False	
InOut								
Static								
▼ Temp								
j	Int							
resultado_s	Real							
Constant								

Network 1: Contador Incremental para variable J

Network 2: Se Mueve el Dato de Entrada al Registro de J

Network 3: Promediado del Registro de J